

Thomas Løkkeborg
Nikolai Mork
Anders Hoelseth Rebner
Petter Sagvold

S0I-tool

Mai 2020

NTNU

Norges teknisk-naturvitenskapelige universitet.
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Bacheloroppgave

2020

Thomas Løkkeborg
Nikolai Mork
Anders Hoelseth Rebner
Petter Sagvold

S0I-tool

Bacheloroppgave
Mai 2020

NTNU

Norges teknisk-naturvitenskapelige universitet.
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk



Abstract

Title:	SOI-tool
Date:	19.05.2020
Participants:	Thomas Løkkeborg Nikolai Mork Anders Hoelseth Rebner Petter Sagvold
Supervisor:	Frode Haug
Employer:	Norwegian Home Guard, Oppland district 05 by/ Major Leif Tanum
Contact:	Espen Torseth
Keywords:	Software development, SOI, Python, Qt, Rectpack
Number of pages:	100
Number of appendixes:	11
Availability:	Open

Abstract:	"Signal Operating Instructions", or SOI, is a critical document required for secure communications during the Norwegian Home Guard's operations. Today the document is created by hand, which is time-consuming and error-prone. This thesis presents the development of a software prototype that streamlines the production and management of SOI. Streamlining is achieved through automatic placement of SOI-components and automatic generation of content, presented in a user-friendly interface.
-----------	--

Sammendrag

Tittel:	SOI-tool
Dato:	19.05.2020
Deltakere:	Thomas Løkkeborg Nikolai Mork Anders Hoelseth Rebner Petter Sagvold
Veileder:	Frode Haug
Oppdragsgiver:	Opplandske Heimevern distrikt 05, G6-seksjonen v/ Major Leif Tanum
Kontaktperson:	Espen Torseth
Nøkkelord:	Programvareutvikling, SOI, Python, Qt, Reactpack
Antall sider:	100
Antall vedlegg:	11
Tilgjengelighet:	Åpen

Sammendrag:	"Samband- og operasjonsinstruks", eller SOI, er et kritisk dokument nødvendig for sikker kommunikasjon under Heimevernets operasjoner. Dokumentet blir i dag laget for hånd, noe som er tidkrevende og utsatt for feil. Denne oppgaven omhandler utvikling av en prototype på programvare som effektiviserer produksjon og behandling av SOI. Effektivisering oppnås ved automatisk plassering av SOI-komponenter og autogenering av innhold, presentert i et brukervennlig grensesnitt.
-------------	--

Forord

Gruppen vil takke Major Leif Tanum og Opplandske Heimevernsdistrikt 05, G6-seksjonen for muligheten til å gjennomføre en spennende bacheloroppgave. Vi ønsker i tillegg å gi en spesiell takk til kontaktperson Espen Torseth for hyggelige og engasjerende møter, og for å ha vært tilgjengelig under prosjektperioden. Videre ønsker vi å takke Jon Tjøstel Ree-Lindstad for å ha møtt oss for brukertesting av prototypen.

Gruppen ønsker også å utrekke en stor takk til vår prosessveileder Frode Haug. Gjennom regelmessig oppfølging har Frode hjulpet oss å styre prosjektarbeidet på riktig kurs, og med sin positive utstråling har han hjulpet oss å holde motivasjonen oppe.

Til slutt vil gruppemedlem Thomas utrekke en personlig takk til resten av gruppen, Anders, Nikolai og Petter: Jeg er veldig takknemlig for å ha blitt inkludert i gruppen, og føler meg heldig som har blitt bedre kjent med dere.

Innhold

Abstract	iii
Sammendrag	v
Forord	vii
Innhold	ix
Figurer	xv
Tabeller	xvii
Kodelister	xix
1 Introduksjon	1
1.1 Fagområde	1
1.2 Avgrensning	1
1.3 Oppgavebeskrivelse	2
1.4 Formål	3
1.4.1 Effektmål	3
1.4.2 Læringsmål	3
1.5 Målgruppe	3
1.6 Bakgrunn og kompetanse	4
1.7 Rammer	4
1.8 Roller	5
1.9 Rapportstruktur	5
1.10 Introduksjon til SOI-moduler	6
1.10.1 Autentiseringstavle	6
1.10.2 Forhåndsavtalte koder	6
1.10.3 Frekvenstabell	7
1.10.4 Kodefraser	7
1.10.5 Sambandsdiagram	7
1.10.6 SOP	7
1.10.7 Subtraktorkoder	7
1.10.8 Telefonliste	8
1.11 Introduksjon til kodebok	8
2 Kravspesifikasjon	9
2.1 Krav til SOI	9
2.1.1 Krav til toppsegment	9
2.1.2 Krav til SOI-moduler	9
2.2 Krav til kodebok	11

2.3	Funksjonelle krav	11
2.3.1	Use cases	13
2.4	Ikke-funksjonelle krav	17
2.4.1	Brukervennlighet	17
2.4.2	Operasjonelle krav	17
2.4.3	Leveranse	17
2.4.4	Lisensiering	18
2.4.5	Kodekvalitet	18
2.5	Domenemodell	18
3	Teknisk design	21
3.1	Modul-arkitektur	21
3.2	SOI på fil	22
3.2.1	Serialisering	22
3.2.2	Komprimering	23
3.3	Database	24
4	Design av brukergrensesnitt	27
4.1	Hovedvinduet	27
4.1.1	Toppmeny	28
4.1.2	Modullister	32
4.1.3	Ny modul	32
4.1.4	Oppsett	33
4.2	SOI	34
4.2.1	Redigering av SOI	34
4.2.2	Toppsegment til SOI	35
4.2.3	SOI-Moduler	35
4.2.4	Font og kontrast	39
5	Utviklingsprosess	41
5.1	Utviklingsmodell	41
5.2	Planlegging og estimering	41
5.3	Research	42
5.4	Designprosess	42
5.5	Arbeidsmetodikk	43
6	Implementasjon	45
6.1	Valgte teknologier og verktøy	45
6.2	Oversikt over brukergrensesnitt	46
6.3	SOI-klassen	47
6.4	SOI-redigering	47
6.4.1	Redigering av SOI-moduler	48
6.4.2	Redigering av toppsegment	49
6.5	SOI-moduler	49
6.5.1	Modul-base	49
6.5.2	Autentiseringstavle og subtraktorkoder	50
6.5.3	Frekvenstabell og telefonliste	51
6.5.4	Forhåndsavtalte koder	52

6.5.5	Kodefraser	54
6.5.6	Fritekst	54
6.5.7	Generisk tabell	54
6.6	Automatisk modulplassering i SOI	55
6.6.1	Introduksjon til rectpack	55
6.6.2	Anvendelse av rectpack	56
6.6.3	Brukerinnflytelse på plassering	56
6.7	SOI på fil	58
6.7.1	Serialisering og deserialisering	58
6.7.2	Komprimering og dekomprimering	59
6.8	Eksport av SOI til PDF	59
6.9	Dialoger	60
6.10	Kodegenerering	61
6.11	Kodebok	61
6.11.1	Grensesnitt mot kodebok	62
6.11.2	Eksport av kodebok	64
6.12	Leveranse	65
7	Testing	67
7.1	Automatisert testing	67
7.1.1	Linting	68
7.1.2	Unit testing	68
7.2	Manuell testing	69
7.3	Brukertesting	70
8	Diskusjon	71
8.1	Valg av teknologier	71
8.1.1	Programmeringsspråk	71
8.1.2	GUI-bibliotek	71
8.1.3	Database	72
8.1.4	Serialisering	73
8.1.5	Komprimering	73
8.1.6	PDF-behandling	74
8.1.7	Kodegenerering	77
8.1.8	System for versjonskontroll	77
8.1.9	Dokumentasjon	77
8.1.10	Produksjon av eksekverbar fil	79
8.2	Modul-arkitektur	79
8.3	Redigering av moduler	80
8.4	Drøfting av implementasjon av SOI-redigering	80
8.5	GUI-testing i Qt via QTest	81
8.5.1	Testing av modulære dialoger	81
8.5.2	Tilgang til omfavnede widgets	82
8.6	Skjermoppløsningsproblemer	83
8.7	Heftelse ved komprimeringsløsning	83
8.8	Databasesynkronisering	84

8.9	Refleksjon over fremdrift	84
8.9.1	Covid-19s effekt på fremdriften	85
8.10	Mangler	85
8.10.1	Sambandsdiagram	86
8.10.2	Feil ved innlesing av konfigurasjonsfiler	88
8.11	Utbedringer	88
8.11.1	Fjerning av moduler i SOI	88
8.11.2	Manuell plassering av moduler	88
8.11.3	Automatisk modulplassering med økt brukerinnflytelse	89
8.11.4	Hindre bruk av like kodeord	90
8.11.5	Evne til å oppdage fonetisk like kodeord	91
8.11.6	Søkefunksjonalitet i kodebok	91
8.11.7	Endre papirretning	91
8.11.8	Valg av emblem i toppsegmentet	91
8.11.9	Legge til kjente snarveier for redigering	92
8.11.10	Farger på moduler	92
8.11.11	Benytte posisjon til QGraphicsWidgetProxy direkte	92
8.11.12	Undersøke nødvendighet av ABC og metaklasse i modul-arkitektur	92
8.11.13	Gjøre komprimering lønnsom	92
8.12	Kjente programvarefeil	93
8.12.1	Formatering av PDF produsert av QPrinter	93
8.12.2	Mulig å trykke på SOI-side gjennom modul	94
8.12.3	Frekvenstabellens overskrifter forsvinner	94
8.12.4	Menyer i Frekvenstabell og Telefonliste kan havne bak and- re moduler	94
8.12.5	Laste inn SOI fra fil på Linux	95
9	Konklusjon	97
9.1	Kritikk av oppgaven	97
9.2	Videre arbeid	98
9.2.1	Synkronisering av data	98
9.2.2	Verktøy for å lage sambandsdiagram	98
9.3	Evaluerer av gruppens arbeid	98
9.3.1	Hva vi er fornøyde med	98
9.3.2	Hva vi ville gjort annerledes	99
9.4	Drøfting av resultater	99
9.4.1	Effektmål	100
9.4.2	Læringsmål	100
9.4.3	Resultatmål	100
9.5	Konklusjon	100
	Bibliografi	101
A	Ordliste	103
B	Gantt-diagram	107
C	Papirprototype	109

D	Script	115
E	Prosjektplan	123
F	Prosjektavtale	149
G	Statusrapporter	153
H	Møtereferater	161
I	Logg	185
J	Eksempler på eksisterende SOI-er	191
J.1	"SOI 2BN"	191
J.2	"SOI ENGRBN"	194
J.3	"SOI HV-05 Generell"	197
J.4	"SOI for HVS SBSK"	200
K	SOI fra programvaren	203
K.1	Eksportert SOI-PDF med to eksemplarer	203
K.2	SOI i ukomprimert form	208
K.3	SOI i komprimert form	209

Figurer

2.1	Use case-diagram: SOI-behandling	12
2.2	Use case-diagram: Kodebokbehandling	13
2.3	Domenemodell	19
3.1	Database-skjema	24
4.1	Hovedvindu	28
4.2	Valgene i toppmenyen	28
4.3	Eksport av PDF	29
4.4	Likhet mellom Fil og SOI	29
4.5	Utklipp av grensesnitt mot kodebok	30
4.6	Utklipp fra eksportert utgave av stor kodebok	31
4.7	Hurtigtaster og enkel bruk	31
4.8	Modulliste	32
4.9	Oversikt modulvalg	32
4.10	Innstillinger til SOI	33
4.11	SOI-toppsegment	33
4.12	SOI med moduler	34
4.13	Generisk tabell og fritext	35
4.14	Opprettelsesdialog	35
4.15	Autentiseringstavle og subtraktorkoder	36
4.16	Frekvenstabell	36
4.17	Telefonliste	37
4.18	Kodefraser	37
4.19	Forhåndsavtalte koder	38
5.1	Trello-tavle	43
6.1	Sammenheng mellom QGraphicsScene, QGraphicsView og elementer i SOI	48
6.2	Forhåndsavtalte koder: Høy og bred utgave	53
6.3	Anvendning av rectpack. Fra kaotisk plassering av moduler til optimalisert plassering av moduler	57
6.4	Ulike plasseringer av tre moduler basert på plasseringsinnstillinger	58

7.1	Pipeline i GitLab CI	68
8.1	Visualisering av mulige valg for PDF-behandling	75
8.2	Gantt med fremdrift sett i ettertid	85
8.3	Illustrasjon av modifisering av Guillotine-algoritmen for å låse posisjon til en gitt modul	90
B.1	Gantt-skjema fra prosjektplan	108
D.1	Output fra scriptet D.7	122

Tabeller

1.1	Eksempel på tom frekvenstabell med to radiotyper	7
2.1	Use case: Se SOI	13
2.2	Use case: Opprett SOI	13
2.3	Use case: Rediger SOI-oppsett	13
2.4	Use case: Legg til SOI-modul	14
2.5	Use case: Rediger SOI-modul	14
2.6	Use case: Endre modulplassering i SOI	14
2.7	Use case: Eksporter SOI til PDF	14
2.8	Use case: Lagre SOI som kompakt fil	15
2.9	Use case: Åpne SOI fra kompakt fil	15
2.10	Use case: Lagre SOI i lokal database	15
2.11	Use case: Åpne SOI fra lokal database	15
2.12	Use case: Legg til ord i kodebok	15
2.13	Use case: Fjern ord fra kodebok	15
2.14	Use case: Rediger oppføring i kodebok	16
2.15	Use case: Oppdater kodebok-koder	16
2.16	Use case: Eksporter kodebok til PDF	16
6.1	Effektive kodelinjer. Hentet via verktøyet ”pygount”.	45
8.1	Skjermopløsninger og skaleringer gruppen vet med stor sikkerhet programvaren fungerer godt på	83

Kodelister

3.1	Serialisert SOI-schema	23
3.2	SQL-statement for SOI	24
3.3	SQL-statement for Codebook	25
3.4	SQL-statement for LastUpdated	25
3.5	SQL-statement for CategoryWords	25
6.1	Utdrag fra ModuleBase-klassen	49
6.2	Eksempel på konfigurasjonsfil for frekvenstabell som gir strukturen vist i tabell 1.1	52
6.3	Hoveddelen av funksjonen <i>create_table_layout</i>	53
6.4	Formatering ved dekomprimering	59
6.5	Funksjon som genererer en ”kode”	61
6.6	Funksjon som genererer flere <i>koder</i>	61
6.7	Utdrag fra <i>CodebookWidget</i>	62
6.8	Utdrag fra <i>CodebookTableModel</i>	62
6.9	Utdrag fra funksjonen <i>insert_row</i> i <i>CodebookRowAdder</i>	63
6.10	Tabellstil	64
6.11	Utdrag fra funksjonen <i>generate_codebook_pdf</i> i <i>codebook_to_pdf.py</i>	64
7.1	Bash-script for sjekk av kodekvalitet	69
8.1	Numpydoc sammenlignet med Sphinx	78
8.2	Optimal løsning	79
8.3	Valgt løsning	80
8.4	Ikke-fungerende test av modulær dialog	81
8.5	Fungerende test av modulær dialog	82
8.6	Fungerende modul for bilder på SOI	87
D.1	Powershell-script for sjekk av kodekvalitet	115
D.2	Steg for å konfigurere Ubuntu GitLab Runner	116
D.3	Steg for å konfigurere Windows GitLab Runner	116
D.4	Docker-image for testing	117
D.5	Installasjonsinstrukser fra leverbar ZIP	118
D.6	Avhengigheter til programvaren	119
D.7	Script brukt for å sammenligne kompresjonsbiblioteker	120
K.1	Ukomprimert SOI	208
K.2	Komprimert SOI	209

Kapittel 1

Introduksjon

Dette kapittelet beskriver oppgaven og introduserer prosjektgruppen og rapportstruktur. Det gis i tillegg en introduksjon til sentrale begreper som er nyttige å ha kunnskap om for videre lesing av rapporten.

1.1 Fagområde

”Kommunikasjon er det å formidle og dele idéer og informasjon.”¹ Mulighetene for kommunikasjon har utviklet seg med tiden. Radio, telefoni og satellitt har muliggjort nesten momentan kommunikasjon, og med internett er det mulig å distribuere store mengder informasjon. Når mange parter skal utføre koordinert arbeid er det nødvendig med god kommunikasjon for å holde hverandre oppdatert.

I militær sammenheng stilles det særlige krav til kommunikasjon. Militære radioer varierer i størrelse, rekkevidde, frekvensområde og bruk, og kommunikasjon mellom disse omtales som samband. Militære operasjoner er avhengige av godt samarbeid og det er lite rom for feil når mye står på spill. Forsyninger må fraktes og distribueres rett, angrep må koordineres, og sanitet må rykke inn der det er nødvendig. Samarbeid er essensielt for å lykkes, og kommunikasjon via samband spiller her en essensiell rolle.

1.2 Avgrensning

For at kommunikasjon via samband skal foregå korrekt og sikkert brukes et dokument kalt SOI, ”Samband- og operasjonsinstruks”. En SOI inneholder alle nødvendige opplysninger for å etablere og opprettholde samband, samt prosedyrer for operasjonen. En SOI består av flere ”moduler” som presenterer ulike typer informasjon. Moduler kan blant annet være frekvenstabell for samband, autentiseringstavle og subtraktorkoder for kryptografi og sikkerhet, og SOP, ”Standard Operating Procedure”, for prosedyrer ved gitte situasjoner.

¹<https://snl.no/kommunikasjon>, aksessert 16.05.2020.

Heimevernet bruker i dag svært mye tid på å produsere dette dokumentet fordi det i stor grad gjøres manuelt. Dette er tid som gjerne skulle blitt brukt på andre viktige oppgaver som sambandsplanlegging, og vedlikehold og drift av systemene.

Oppgaven er avgrenset til å håndtere generering, produksjon og administrasjon av SOI-er og omfatter ikke distribusjon av dem. Resultatet av oppgaven skal være en så godt som mulig fungerende prototype, ikke et produkt klart for daglig drift.

1.3 Oppgavebeskrivelse

Oppgaven er å utvikle en applikasjon for produksjon av SOI-er. Applikasjonen skal fungere som en prototype for en fullstending versjon. Applikasjonens grafiske grensesnitt skal være sentralisert rundt sluttproduktets utseende, altså hvordan utskrevet SOI vil se ut. Brukervennlighet er svært viktig siden tidspress for å produsere en ny SOI kan være meget høyt, og feil i en SOI kan ha store konsekvenser. Programvaren skal være modulbasert slik at det er mulig å utvikle nye moduler ved behov.

SOI-er må produseres som godt leselige PDF-er for enkel utskrift og elektronisk distribusjon til høyere og sideliggende avdelinger. En SOI skal kunne representeres av en fil med kompakt filformat som egner seg for overføring over kommunikasjonskanaler med lav båndbredde. Mottaker av filen skal kunne gjenskape SOI gjennom programvaren, og dermed ha tilgang til all funksjonalitet slik som PDF-eksportering og videre redigering.

Systemet skal også inneholde en kodebok, samt et grensesnitt mot denne slik at den kan ses og redigeres. Det skal være mulig å eksportere kode- og dekodebok som PDF for at den enkelt skal kunne skrives ut.

Følgende SOI-moduler er nødvendige:

- Kodefraser
- Sambandsdiagram
- Autentiseringstavle
- Subtraktorkoder
- Forhåndsavtalte koder
- Frekvenstabell
- Telefonliste

Følgende SOI-moduler er ønskelige:

- Nøkkeldata
- Kallesignaler
- SOP - "Standard Operating Procedure"

For å avlaste brukeren og effektivisere SOI-produksjon skal modulers innhold, delvis eller fullstendig, autogenereres der det er hensiktsmessig. En introduksjon

til hver modul gis i seksjon 1.10. Gjennomførbarhetsbevis på synkronisering av data skal også leveres.

Dersom det blir tid, kan oppgaven utvides til å omfatte:

- Evnen til å fange opp kodeord som er fonetisk like.
- Verktøy for å lage sambandsdiagrammer. Nærmere beskrevet i 9.2.

1.4 Formål

Oppgavens formål er å lage en prototype på programvare for SOI-behandling.

1.4.1 Effektmål

Heimevernet skal kunne bruke prototypen til å vurdere, og eventuelt argumentere for, allokering av ressurser til utvikling av en fullversjon av programvaren.

Utviklet prototype skal:

- Demonstrere at produksjon av SOI i programvare utformet for nettopp dette er tidsbesparende.
- Demonstrere at egen programvare for SOI kan hindre gjenbruk av koder og kodeord og dermed øke sikkerhet ved anvendelse av SOI.
- Demonstrere nye muligheter egen programvare for SOI kan gi.

1.4.2 Læringsmål

Gruppemedlemmene skal tilegne seg:

- Kunnskap innenfor valgt programmeringsspråk, samt rammeverk for grafisk brukergrensesnitt og database.
- Grunnleggende forståelse av kryptografi innenfor militær sambandskommunikasjon.
- Erfaring vedrørende systematisk planlegging av, og metodisk arbeid med, et større prosjekt.
- Erfaring med å samle og anvende kunnskap fra utdanningen.
- Erfaring i samarbeid med oppdragsgiver og kontaktperson om en reell oppgave.

1.5 Målgruppe

Hovedmålgruppen for denne rapporten er oppdragsgiver Heimevernet. Her inngår både utviklere som potensielt skal videreutvikle programvaren og ansatte som skal produsere SOI-er.

Videre kan rapporten også være aktuell lesing for ytterlige ansatte i Forsvaret, medstudenter og ansatte ved NTNU, og øvrige personer som er interessert i utvikling av programvare for SOI-behandling.

1.6 Bakgrunn og kompetanse

Gruppen består av fire studenter på bachelornivå ved NTNU Gjøvik. Alle studentene har vært gjennom generelle informatikkfag innen programmering i C++ og Python, relasjonsdatabaser, algoritmer, systemutvikling, operativsystemer og brukergrensesnittdesign. Under følger en utdyping av bakgrunn og kompetanse til hvert av grupped medlemmene:

Thomas Løkkeborg studerer Bachelor i IT-drift og Informasjonssikkerhet. I tillegg til kompetansegrunnlaget beskrevet over har han kompetanse innen drift av systemer i skymiljø, utvikling av skybaserte tjenester i Golang, utvikling av webapplikasjoner i Polymer og PHP, datanettverk, samt noe kompetanse i utvikling av brukergrensesnitt med Swing, Java. Han har tilbrakt ett år ved CERN som "Technical Student" som en del av sin utdanning, hvor fokusområdet var å flytte Java applikasjoner til skyteknologien "Kubernetes".

Nikolai Mork studerer Bachelor i ingeniørfag, data. I tillegg til kompetansegrunnlaget beskrevet over har han kompetanse i applikasjonsutvikling innen både Java, Python og ActionScript 3, samt grunnleggende kunnskaper innenfor HTML og CSS. Han gjennomførte førstegangstjenesten i Artilleribataljonen i Hæren hvor han fikk erfaring med både HF og MRR gjennom sin stilling på kommandoplass.

Anders H. Rebner studerer Bachelor i ingeniørfag, data. I tillegg til kompetansegrunnlaget beskrevet over har han kompetanse innen applikasjonsutvikling med Java/JavaFX, Python/TkInter og grunnleggende HTML/CSS. Han har gjennomført førstegangstjeneste som vaktsoldat i Sjøforsvaret hvor han fikk erfaring i bruk av samband.

Petter Sagvold studerer Bachelor i ingeniørfag, data. I tillegg til kompetansegrunnlaget beskrevet over har han kompetanse innen applikasjonsutvikling med Java/JavaFX, Python og grunnleggende HTML og CSS.

1.7 Rammer

Fysiske rammer

- Gruppen har ikke fast arbeidsrom, og er avhengig av møterom ved NTNU Gjøvik og bruk av egne boliger.
- Tidsrammen for prosjektet er satt fra 08.01.2020 til 20.05.2020.

- Gruppen har 4 medlemmer, og er gjennom gruppereglene, se prosjektplan vedlegg E, pliktet å jobbe minst 25 timer i uken. Med 18 arbeidsuker utgjør dette til sammen minst 1800 timer for gruppen.

Teknologiske rammer

- Programvaren skal kunne kjøre på 64-bit maskiner med nyere versjon av operativsystemet Windows 10. Med nyere versjon menes en versjon ikke eldre enn ved starten av prosjektperioden.
- Gruppen vil ikke bli supplert med maskinvare og må bruke det de selv har tilgjengelig og det NTNU kan tilby.
- Gruppen vil ikke bli supplert med programvare og må bruke det de selv har tilgjengelig og det NTNU kan tilby.

Økonomiske rammer

- I utgangspunktet skal åpen programvare brukes. Om ytterligere lisensbeskyttet programvare må benyttes skal støtte for dette diskuteres med oppdragsgiver.

1.8 Roller

Relevante roller for prosjektet er:

- **Oppdragsgiver:** Opplandske Heimevernsdistrikt 05, G6-seksjonen.
- **Kontaktpersoner:** Major Leif Tanum og seniorrådgiver ved NTNU Espen Torseth.
- **Prosjektleder:** Gruppemedlem Anders H. Rebner.
- **Prosessveileder:** Universitetslektor ved NTNU Frode Haug.

1.9 Rapportstruktur

Rapporten består av følgende kapitler:

Introduksjon: Introducerer prosjektet. Leser gis nok kontekst til å få en viss domenekunnskap før kravspesifikasjon går i detaljer på hva oppgaven konkret går ut på.

Kravspesifikasjon: Beskriver krav til programvaren.

Teknisk design: Beskriver overordnet design av arkitektur og tekniske løsninger, uten å gå i implementasjonsdetaljer.

Design av brukergrensesnitt: Beskriver utforming av brukergrensesnitt, uten å gå i implementasjonsdetaljer.

Utviklingsprosess: Beskriver fremgangsmåte fulgt under utvikling.

Implementasjon: Beskriver hvordan programvaren ble implementert, med bakgrunn i grafiske og teknologimessige valg avlagt i tidligere kapitler. Kapitlet beskriver også leveransen av programvaren.

Testing: Beskriver hvordan programvaren ble testet for å kvalitets sikre sluttresultatet.

Diskusjon: Inneholder en samling diskusjoner rundt ulike aspekter ved prosjektet. Et eget diskusjonskapittel ble opprettet slik at resten av rapporten kunne presentere resultatet av valgene som ble gjort, uten å også måtte legge frem alternative løsninger og begrunnelser.

Konklusjon : Legger frem resultatet av prosjektet, og inkluderer avsluttende kommentarer.

I tillegg kommer rapporten med en rekke vedlegg som refereres til der de er relevante. Hvert vedlegg kommer med en forklarende tekst.

Rapporten bruker kryss-referanser for å henvise fra en tekst til en annen. En kryss-referanse til første seksjon i kapittel 2 vil for eksempel se slik ut: 2.1. Når kildekode diskuteres i tekst skrives den ofte i *kursiv*.

1.10 Introduksjon til SOI-moduler

Følgende er en oversikt over og introduksjon til sentrale moduler i en SOI. For nærmere beskrivelse av modulene se 2.1.2. Eksempler på SOI-er finnes i vedlegg J.

1.10.1 Autentiseringstavle

Autentiseringstavler brukes for å forsikre seg om at motparten har lik SOI og dermed mest sannsynlig er alliert. Tavlen inneholder rekker av bokstaver knyttet til en enkelt bokstav og for noen rekker også et siffer. Ved forespørsel om autentisering sender man en bokstav eller siffer og et nummer. Motparten må da bevise sin validitet ved å returnere den bokstaven som er på nummerets plass telt fra venstre i rekken tilhørende bokstaven eller sifferet. Slik autentisering kan repeteres for å øke sikkerheten.

1.10.2 Forhåndsavtalte koder

Ved mistanke eller bekreftelse på at sambandet avlyttes av uvedkommende kan man kryptere meldinger som sendes. Forhåndsavtalte koder er en samling sentrale ord og fraser knyttet til koder. Med dette kan man konstruere beskjer for å få ut viktig informasjon. For enkel og rask bruk er ordene og frasene kategorisk sortert.

1.10.3 Frekvenstabell

Heimevernet har forskjellige radioer og hver av disse har forskjellig bruk. En frekvenstabell er en tabell som inneholder den informasjonen som trengs for å etablere og holde samband mellom de forskjellige radioene. Eksempler på data i en frekvenstabell kan være frekvenser, nett-id, kanalnummer eller kodeord.

	MRR			HF	
Nett	Frekvenser	Nett-ID	NIS	Frekvenser	Kodeord

Tabell 1.1: Eksempel på tom frekvenstabell med to radiotyper

1.10.4 Kodefraser

En SOI kan inneholde en liten tabell med ord eller fraser tilknyttet et kodeord. Dette er ord eller fraser for spesielle situasjoner eller handlinger som er kritiske. Eksempelvis "Start radiotaushet" eller "Mistet SOI". Fraser som har logisk sammenheng skal ha kodeord med kategorisk sammenheng. Eksempelvis "Start radiotaushet" - "Hamburger" og "Slutt radiotaushet" - "Pizza". Tabellen skrives ofte på et eget ark slik at soldater kan pugge den og destruere den så snart innholdet er lært.

1.10.5 Sambandsdiagram

Sambandsdiagram er en graf som viser strukturen i et nettverk. Hver node i grafen representerer en militær enhet og inneholder dens kallesignal. Hver kant representerer en forbindelse mellom enheter. Dette gir et oversiktsbilde av sambandet.

1.10.6 SOP

SOP, "Standard Operating Procedure", inneholder beskrivelse av militære feltprosedyrer. For eksempel prosedyre ved tap av samband.

1.10.7 Subtraktorkoder

Subtraktorkoder er bokstaver knyttet til en kartreferanse. En kartreferanse består av breddegrad og lengdegrad. Ved å sende summen av subtraktorkoden og kartreferansen du ønsker å dele sammen med valgt subtraktorkodes bokstav kan mottakeren trekke fra subtraktorkoden tilhørende bokstaven og få den samme kartreferansen. Denne teknikken forhindrer at de som ikke har lik SOI får tilgang til kartreferansen.

1.10.8 Telefonliste

Telefonliste er en samling sentrale telefonnumre, e-poster, satellittelefon-numre og FDN-nummer som ikke nødvendigvis brukes militært taktisk.

1.11 Introduksjon til kodebok

En kodebok har to deler, en for koding og en for dekoding. Delene kan enten komme som en samlet bok eller som to separate bøker. Kodeboken er en alfabetisk ordliste hvor hvert ord er knyttet til en kode. Dekodeboken har likt innhold som kodeboken, men er sortert alfabetisk på koder. Kodebøker brukes for å konstruere og dekode krypterte beskjeder.

Kapittel 2

Kravspesifikasjon

Dette kapittelet presenterer spesifikke krav til SOI og kodebok, i tillegg til øvrige funksjonelle og ikke-funksjonelle krav. Krav til SOI og kodebok uttrykkes ved presise tekstlige beskrivelser, funksjonelle krav uttrykkes med use cases, og ikke-funksjonelle krav uttrykkes med kategoriserte punktlister.

2.1 Krav til SOI

2.1.1 Krav til toppsegment

Hver side i SOI skal ha et toppsegment som inneholder tittel, beskrivelse, sidetall, eksemplarnummer, gyldighetsperiode, opprettelsesdato, graderingsmerke og emblem. Sidetall suppleres med totalt antall sider. Eksemplarnummer suppleres med totalt antall eksemplarer av SOI-en. Gyldighetsperiode uttrykkes med start- og sluttdato. Graderingsmerke kan for eksempel være "GRADERT" eller "UGRADERT", og skal være rødt. Emblem viser våpenskjold eller logo relevant til operasjonen SOI brukes i. Sider i SOI som er vedlegg skal tydelig markeres som dette.

2.1.2 Krav til SOI-moduler

En SOI består av flere "moduler". I denne seksjonen beskrives alle moduler programvaren kreves å støtte, samt moduler det er ønskelig at programvaren skal støtte. Reelle eksempler på modulene finnes i SOI-er lagt til som vedlegg J. Forklaring på bruk av moduler i praksis finnes i seksjon 1.10.

Autentiseringstavle

Autentiseringstavle inneholder bokstaver A til Z og tall 0 til 9, med en tilfeldig kode på 25 bokstaver knyttet til hver av disse. Bokstaver og tall kan knyttes til samme kode. Antall oppføringer av koder med tilhørende bokstav og tall, eller bare bokstav, skal være brukervalgt. Modulen inkluderer også to "testelement", nummer som brukes for å kommunisere hvilke del av koden som skal brukes til

autentisering. Testelement skal være et tilfeldig nummer i intervallet 1 til 25. Testelementene skal være unike innad i modulen.

Kodefraser

Kodefraser består av vilkårlige brukervalgte fraser med tilknyttede kodeord, hvor alle kodeord er av samme kategori. En frase kan for eksempel være "Kompromitert SOI" eller "Start radiotaushet". En kategori kan for eksempel være "Tresort", og eksempler på ord i samme kategori er "Eik", "Bjørk" eller "Lønn". Dersom flere moduler bruker "kategori-ord" på denne måten skal modulen velge en kategori som er unik på tvers av modulene i SOI.

Forhåndsavtalte koder

Forhåndsavtalte koder består av kategoriserte ord med tilknyttede koder. Ordene er i praksis alle ord i den "lille kodeboken" 2.2. Modulen kan struktureres på to måter:

1. Hvert ord gis kode som er unik på tvers av kategorier. Kodene er bokstaver A til Z i stigende leksikografisk rekkefølge. Antall bokstaver i hver kode er høyt nok til at alle ord får unik kode, men ikke høyere.
2. Hver kategori gis kode som er unik blant alle kategorier i modulen, og ord gis koder som bare er unike innad i sin kategori. Koder er bokstaver A til Z i stigende leksikografisk rekkefølge. Antall bokstaver i hver kode er høyt nok til at henholdsvis ord og kategorier får unik kode, men ikke høyere. Eksempel på koder som deles ut til 28 ord i en kategori er: "AA", "AB", ..., "AZ", "BA", "BB". Siden koder til ord bare er unike innad i en kategori vil 28 kategorier også ha de samme kodene i dette eksempelet. Et annet eksempel er at om en kategori bare består av ett ord, og kategorien er den eneste i modulen, så vil både kategori og ord få koden "A".

Støtte for én av de nevnte struktureringene er et krav, og støtte for begge er ønskelig. I begge tilfeller skal modulen sorteres først på kategori, deretter på kode i stigende leksikografisk rekkefølge.

Modulen inneholder også et varslingsord: et tilfeldig utvalgt ord som bruker kan velge å overskrive om brukeren ønsker.

Frekvenstabell

Frekvenstabell inneholder et utvalg attributter nødvendig for å kommunisere over et utvalg radioer. Eksempler på radioer er "MRR", "HF", "Hytera" og "LFR". Eksempler på attributter er "Frekvenser", "Nett-ID", "Område" og "Kanal nummer". Hver radiotype kan ha ulike attributter. Radiotyper og attributter kan endre over tid. Hvilke attributter som knyttes til hvilke radiotyper kan også endre over tid. Det skal derfor være mulig å konfigurere hvilke attributter som inngår i ulike radioer, samt hvilke radioer som støttes.

Fritekst

Fritekst består av en vilkårlig brukervalgt overskrift og et tekstfelt som kan inneholde vilkårlig brukervalgt fritekst. Med fritekst menes tekst som kan inneholde "tvungne" nye linjer. Det er dermed brukers ansvar å formattere tekst, og bruker får mulighet til å uttrykke seg fritt. Eksempel på innhold er overskriften "Prosedyre: Bytte SOI" og tekstfelt som inneholder punktene "1. Makuler SOI." og "2. Skriv ut ny SOI."

Sambandsdiagram

Sambandsdiagram viser en grafisk framstilling av forbindelsene mellom enheter i et nettverk. Mulighet for å laste inn et bilde av et sambandsdiagram i en modul vil tilfredstille kravet.

Subtraktorkoder

Subtraktorkoder inneholder bokstaver A til Z med to tilfeldige rekker heltall på 4 siffer knyttet til hver av disse. Tallene utgjør breddegrad og lengdegrad. Antall oppføringer av koder med tilhørende breddegrad og lengdegrad skal være brukervalgt.

Telefonliste

Telefonliste består av en eller flere "funksjoner" og en eller flere av attributtene "mobilnummer", "iridiumnummer", "FDN-nummer" og "e-post" tilknyttet hver funksjon. "Funksjon" brukes her som samlebetegnelse for navn på rolle, enhet eller person, for eksempel "Tropp A" eller "KO"¹. Hvilke attributter som er nødvendige kan endre over tid, og dermed er det ønskelig at utvalg av attributter kan konfigureres.

2.2 Krav til kodebok

Kodebok skal inneholde en samling ord med tilhørende kategori og kode. En kode består av tilfeldige bokstaver A til Z. Antall bokstaver i en kode velges slik at alle ordene kan gis unik kode. Det skilles mellom "stor" og "liten" kodebok. Stor kodebok inneholder alle ord i kodeboken, og liten kodebok inneholder de ordene som inngår i modulen "Forhåndsavtalte koder" beskrevet i seksjon 2.1.2.

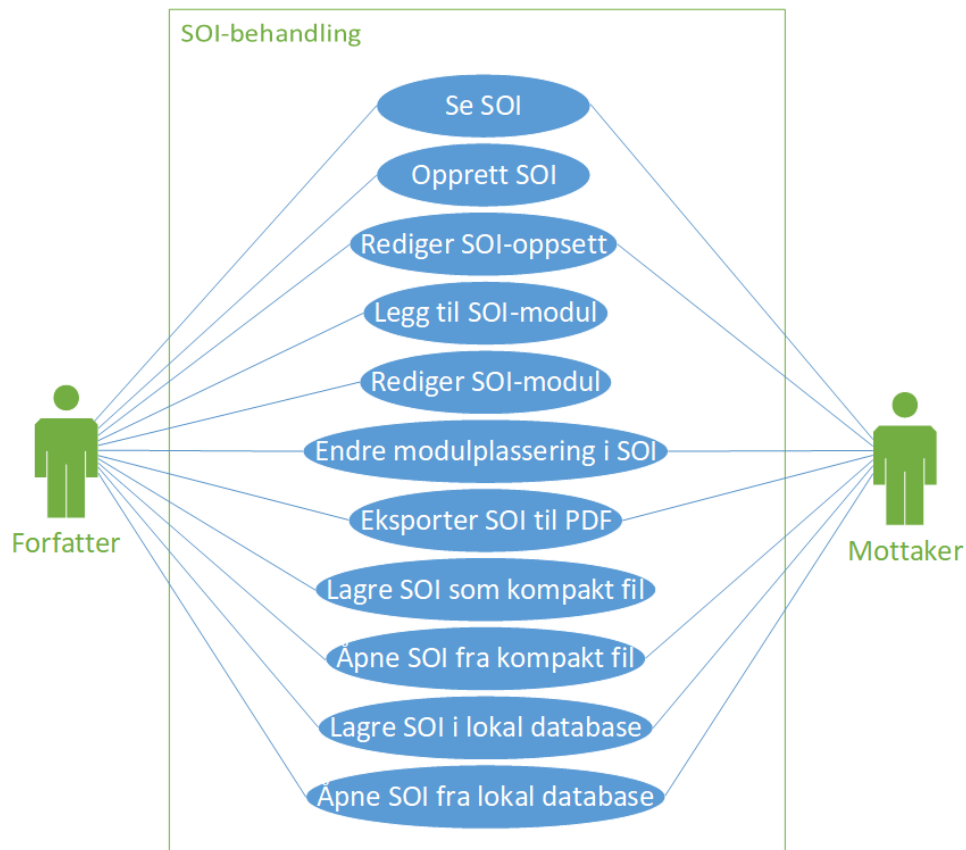
2.3 Funksjonelle krav

Funksjonelle krav beskrives ved bruk av use cases. Use cases har to aktører: forfatter og mottaker. Forfatter er skaper av SOI, og vil typisk være i ledende posisjoner

¹Kommandoplass, stabs plass eller sambandssentral. Der staben utfører ledelse.

i Heimevernet. Mottakere skaper ikke SOI-er, men trenger programvaren for å håndtere SOI-er som mottas. I praksis vil mange brukere ha begge disse rollene siden en bruker både kan motta og forfatte SOI.

Use cases vises i to diagrammer: ett for SOI-behandling og ett for kodebokbehandling. Disse vises henholdsvis i figur 2.1 og 2.2. Use cases utdypes i seksjon 2.3.1. Merk at diagrammene bruker en enkel tilnærming til UML da kravene tydelig kan beskrives uten mer avansert notasjon.



Figur 2.1: Use case-diagram: SOI-behandling



Figur 2.2: Use case-diagram: Kodebokbehandling

2.3.1 Use cases

Use case:	Se SOI
Aktører:	Forfatter, Mottaker
Beskrivelse:	Bruker kan visualisere hvordan utskrevet SOI vil se ut gjennom programvarens grensesnitt.

Tabell 2.1: Use case: Se SOI

Use case:	Opprett SOI
Aktører:	Forfatter
Beskrivelse:	Bruker kan opprette en ny SOI. SOI-en vil bestå av en enslig, tom side med toppsegment, og uten moduler.

Tabell 2.2: Use case: Opprett SOI

Use case:	Rediger SOI-oppsett
Aktører:	Forfatter, Mottaker
Beskrivelse:	Bruker kan endre SOI-ens tittel, beskrivelse, versjon, opprettelsesdato, gyldighetsperiode og klassifisering, i tillegg til plasseringsinnstillinger for moduler.

Tabell 2.3: Use case: Rediger SOI-oppsett

Use case:	Legg til SOI-modul
Aktører:	Forfatter
Beskrivelse:	<p>Bruker kan legge til flere typer moduler. Valgt modul vil vises i SOI, og plasseringen av denne baseres på valgte plasseringsinnstillinger, se use case "Endre modulplassering i SOI". Modulen vil kun inneholde standard utgangsverdier.</p> <p>Bruker kan alternativt legge til modul som vedlegg. Da plasseres modulen på en egen side som er ekskludert fra totalt sidetall i SOI, og som er tydelig merket som vedlegg.</p>

Tabell 2.4: Use case: Legg til SOI-modul

Use case:	Rediger SOI-modul
Aktører:	Forfatter
Beskrivelse:	<p>Bruker kan klikke og bruke tastatur for å redigere de fleste modulers innhold. Kompliserte moduler som ikke lar seg redigere i editoren kan redigeres i separat dialog som åpnes med en knapp eller hurtigtast. I modulene autentiseringstavle, subtraktorkoder og forhåndsavtalte koder bestemmer brukeren innholdsmengden, men innholdet genereres automatisk av programvaren.</p>

Tabell 2.5: Use case: Rediger SOI-modul

Use case:	Endre modulplassering i SOI
Aktører:	Forfatter, Mottaker
Beskrivelse:	<p>Bruker kan plassere moduler manuelt, la programvaren ta seg av dette, eller benytte en hybrid av disse:</p> <ul style="list-style-type: none"> • Ved manuell plassering har bruker kontroll over, og ansvar for, modulplassering. • Ved automatisk plassering lar brukeren programvaren bestemme modulplassering.

Tabell 2.6: Use case: Endre modulplassering i SOI

Use case:	Eksporter SOI til PDF
Aktører:	Forfatter, Mottaker
Beskrivelse:	<p>Bruker kan eksportere SOI til PDF-format med et valgt antall SOI-eksemplarer. PDF-en kan videre håndteres utenfor programvaren, slik brukeren ønsker.</p>

Tabell 2.7: Use case: Eksporter SOI til PDF

Use case:	Lagre SOI som kompakt fil
Aktører:	Forfatter
Beskrivelse:	Bruker kan lagre SOI på fil i kompakt filformat som egner seg for nettverks-overføring. Filen skal inneholde all informasjon som er nødvendig for å reprodusere SOI gjennom programvaren.

Tabell 2.8: Use case: Lagre SOI som kompakt fil

Use case:	Åpne SOI fra kompakt fil
Aktører:	Forfatter, Mottaker
Beskrivelse:	Bruker kan åpne SOI mottatt som kompakt fil. SOI vil da gjen-skapes og åpnes i programvaren klar for redigering og PDF-eksport.

Tabell 2.9: Use case: Åpne SOI fra kompakt fil

Use case:	Lagre SOI i lokal database
Aktører:	Forfatter, Mottaker
Beskrivelse:	Bruker kan lagre SOI i lokal database.

Tabell 2.10: Use case: Lagre SOI i lokal database

Use case:	Åpne SOI fra lokal database
Aktører:	Forfatter, Mottaker
Beskrivelse:	Bruker kan åpne SOI fra lokal database. Bruker vil bli vist en oversikt over alle tilgjengelige SOI-er i databasen, og valgt SOI vil åpnes i programvaren klar for redigering og PDF-eksport.

Tabell 2.11: Use case: Åpne SOI fra lokal database

Use case:	Legg til ord i kodebok
Aktører:	Forfatter
Beskrivelse:	Bruker kan legge til ord i kodeboken. Bruker angir da et ord, ordets kategori og om ordet skal legges til i stor eller liten kodebok. Tilhørende kode genereres automatisk og tilføyes når ordet legges til. Dersom antall ord overskrider kapasiteten til nå-værende kodelengde, vil kodelengden økes og alle koder opp-dateres automatisk.

Tabell 2.12: Use case: Legg til ord i kodebok

Use case:	Fjern ord fra kodebok
Aktører:	Forfatter
Beskrivelse:	Bruker kan fjerne ord fra både stor og liten kodebok.

Tabell 2.13: Use case: Fjern ord fra kodebok

Use case:	Rediger oppføring i kodebok
Aktører:	Forfatter
Beskrivelse:	Bruker kan redigere kodebok-oppføringer. Ord, kategori og om ordet tilhører stor eller liten kodebok, kan redigeres.

Tabell 2.14: Use case: Rediger oppføring i kodebok

Use case:	Oppdater kodebok-koder
Aktører:	Forfatter
Beskrivelse:	Bruker kan når som helst utløse en oppdatering av kodebok-koder manuelt, selv om programvaren sørger for å oppdatere disse hver 24. time. Ved manuell oppdatering vil automatisk oppdatering utføres 24 timer senere.

Tabell 2.15: Use case: Oppdater kodebok-koder

Use case:	Eksporter kodebok til PDF
Aktører:	Forfatter
Beskrivelse:	Bruker kan eksportere både stor og liten kodebok til PDF. PDF-en er klar for utskrift og bruk med sin kode- og dekodebok.

Tabell 2.16: Use case: Eksporter kodebok til PDF

2.4 Ikke-funksjonelle krav

Denne seksjonen beskriver krav som ikke direkte sikter til funksjonalitet i programvaren.

2.4.1 Brukervennlighet

- Brukergrensesnittet skal være på norsk bokmål.
- Brukergrensesnittet skal ha visualisering av SOI i fokus.
- Programvaren skal gi tilbakemelding til bruker ved uventede feil.
- Utskrevet SOI skal være lettleseleg, selv i svart/hvitt. Med lettleseleg menes i praksis at valg av skrifttype og skriftstørrelse på papir er godt begrunnet og dokumentert.
- Det er ønskelig at programvaren skal kunne advare om kodeord som er fonetisk like.

2.4.2 Operasjonelle krav

- Eksportert SOI i filformat skal lagres i US ASCII².
- Filstørrelsen på eksportert SOI i kompakt filformat skal være slik at overføring med hastighet på 2.4 kbps fullføres innen 5 minutter, men det er ønskelig at overføringstiden begrenses til ett minutt. Dette betyr at filstørrelsen skal være lavere enn 90 kB, og helst ikke overstige 18 kB.
- Programvaren skal bruke database av "embedded" type slik at den vil fungere i isolasjon.
- Databasen skal kunne flyttes via minnepenn. Dette vil i praksis si at valgt database må være lagret på filsystem på en slik måte at den enkelt kan hentes ut og erstattes.
- Programvaren skal være kompatibel med 64-bit maskiner med nyere versjon av operativsystemet Windows 10. Med nyere versjon menes en versjon ikke eldre enn ved starten av prosjektperioden 08.01.2020.
- SOI skal ikke overstige to A4-ark når alle moduler er lagt til, vedleggssider medregnes ikke, se vedlegg J for eksempler.

2.4.3 Leveranse

- Programvaren skal leveres med instruks for installasjon og enkel bruk. Installasjonsinstruksene skal ikke overstige en halv A4-side med skriftstørrelse 11.
- Et skriftlig gjennomførbarhetsbevis på synkronisering av kodebok og SOI mellom flere brukere skal leveres sammen med programvaren.

²<http://www.columbia.edu/kermit/ascii.html>, akseptert 16.05.2020.

2.4.4 Lisensiering

Heimevernet bruker åpen kildekode, så programvaren kan også gjøre det. Kontaktperson informerer om at LGPL³ er uproblematisk, at GPL⁴ kan være utfordrende, men ofte løsbart, og at BSD 3-clause⁵ er den enkleste å forholde seg til. Det er dermed et krav at gruppen holder seg til denne prioriteringen så mye som mulig.

2.4.5 Kodekvalitet

- Valgt programmeringsspråk skal være av lett leselig karakter siden arbeidet skal overtas og muligens jobbes videre med etter endt prosjekt. Med "lett leselig karakter" menes et språk hvor allmenn konsensus i datamiljøet er at det er leselig.
- Programvaren skal være modulbasert, det vil si at man skal kunne utvide programvaren med nye SOI-moduler.
- Programvaren skal testes og dokumenteres for å øke kodekvalitet, og for å gjøre eventuell videre utvikling på kodebasen enklere etter endt prosjekt.

2.5 Domenemodell

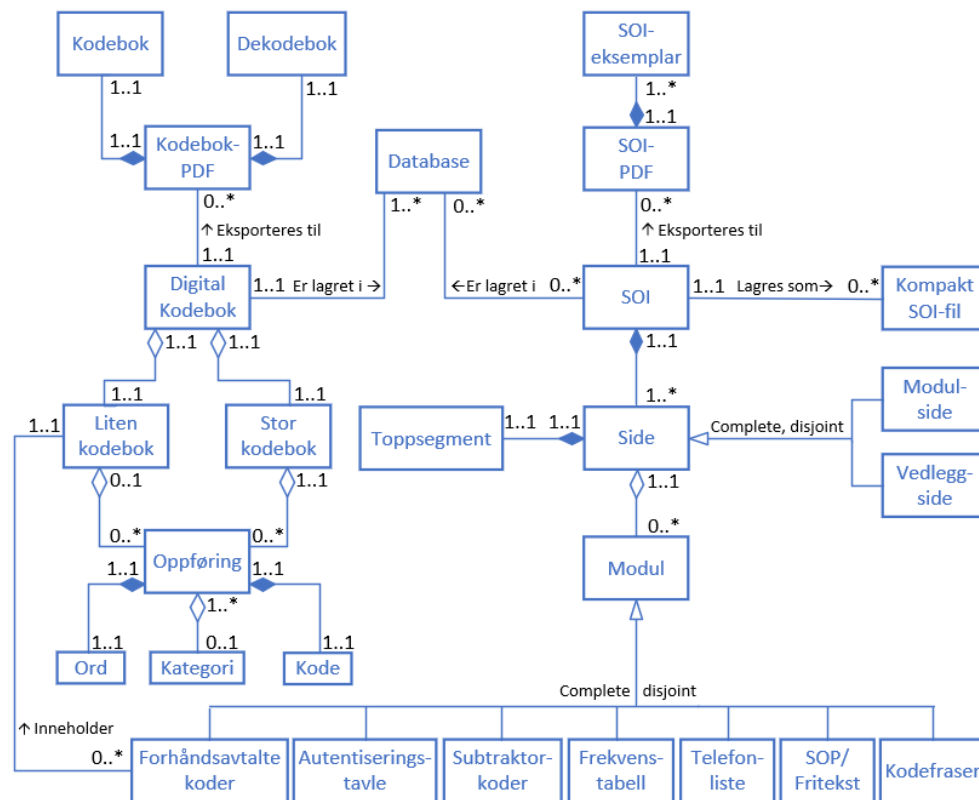
Domenemodellen i figur 2.3 modellerer de relevante "tingene" som programvaren utvikles med bakgrunn i, og relasjonene mellom dem. Det modelleres for å konkretisere domeneforståelsen som programvaren blir utviklet med. Modellen bruker klassesdiagram med "UML 2.5"⁶-notasjon. Det er nevneverdig at ved generalisering brukes begrepene "complete" og "incomplete", som henholdsvis betyr "må være en av" og "må ikke være en av", og begrepene "disjoint" og "overlapping" som henholdsvis betyr "kan bare være en av" og "kan være flere av".

³LGPL: Lesser General Public License. En mindre restriktiv utgave av GPL som tillater inkludering av programvare uten å tvinge gjenbruk av samme lisens, betinget at inkludert programvare ikke endres. <https://www.gnu.org/licenses/lgpl-3.0.txt>, aksessert 16.05.2020.

⁴GPL: General Public License. "Copyleft" lisens som tvinger gjenbruk av samme lisens på all kode som tar i bruk programvare. <https://www.gnu.org/licenses/gpl-3.0.txt>, aksessert 16.05.2020.

⁵BSD: Berkeley Software Distribution. En lisens med få restriksjoner som tillater fri gjenbruk av programvare, betinget at en kopi av BSD-lisensen og et "lisensvarsel" inkluderes. <https://opensource.org/licenses/BSD-3-Clause>, aksessert 16.05.2020.

⁶UML - Universal Modelling Language. Oppsummert notasjon i <https://www.omg.org/spec/UML/2.5> og <https://www.uml-diagrams.org/generalization.html?context=class-diagrams>, aksessert 05.05.2020, brukes i denne rapporten.



Figur 2.3: Domenemodell

Kapittel 3

Teknisk design

Dette kapitlet beskriver den tekniske strukturen. Både hvordan enkeltkomponenter og programvaren som helhet er bygget opp presenteres. Først beskrives programvarens modulbaserte arkitektur, deretter hvordan SOI er representert på fil og avslutningsvis design av databasen. Grunnlaget for innholdet i dette kapitlet er designprosessen utdypet i seksjon 5.4.

3.1 Modul-arkitektur

Programvaren skal støtte utvikling av nye SOI-moduler. Det er derfor ønskelig med en tilnærming til implementasjon av moduler som gir ”løs kobling” mellom programvaren og individuelle moduler. Ved å bestemme fellestrekk moduler må ha i programvaren kan vi generalisere hvordan programvaren behandler dem.

Fellestrekk for moduler i programvaren er at de må kunne:

- Serialisere seg selv til en streng på eget format.
- Initialiser seg selv fra serialisert streng på eget format.
- Oppgi sin egen størrelse på papir.
- Tilby et brukervennlig navn på seg selv.
- Tilby et ikon til bruk i grafisk brukergrensesnitt.
- Redigeres av bruker.
- Presentere seg selv på skjerm.
- Presentere seg selv på PDF.

Ved å bare interagere med moduler fra programvaren via disse fellestrekkene slipper programvaren å bry seg om modulspekifikke implementasjonsdetaljer. Som eksempel kan én modul bestå av tabeller, og en annen av tekstfelt. Utregning av størrelse på disse vil foregå ulikt, men dette kan holdes skjult fra resten av programvaren.

På høynivå løses dette med bruk av ”interfaces”, en programmeringsmetode som tilbyr forutsigbare grensesnitt til programvarekomponenter. Så lenge nye

SOI-moduler respekterer overordnet interface vil programvaren kunne støtte dem. Konkret implementasjon av løsningen gis i seksjon 6.5.1.

3.2 SOI på fil

En svært viktig del av programvaren er at den støtter eksportering og importering av SOI til og fra fil. En av årsakene til dette er at en SOI gjerne benyttes gjentatte ganger, men med oppdatert kryptografi. En bruker kan dermed lagre en SOI-mal og bygge på denne hver gang, eller gjenbruke forrige SOI etter å ha oppdatert dens kryptografi. Den andre årsaken er at en SOI skal kunne distribueres over lav båndbredde, noe som ikke er ønskelig å gjøre med store filstørrelser slik som PDF-er har. Vårt design av filbasert SOI, se 3.2.1, muliggjør denne distribusjonen, samt rekonstruksjon av SOI på mottakersiden slik at den kan skrives ut og brukes.

3.2.1 Serialisering

En SOI serialiseres til en JSON-streng før den skrives til fil. Denne inneholder all data programvaren trenger for å rekonstruere en SOI med all funksjonalitet tilgjengelig. JSON-strengen følger "schema" vist i kodeliste 3.1, og listen under beskriver JSON-nøkklene i formatet.

- **title:** Tittel.
- **description:** Kort beskrivelse.
- **version:** Versjon.
- **date:** Opprettelsesdato.
- **valid:** Gyldighetsperiode.
 - **from_date:** Startdato for gyldighet.
 - **to_date:** Sluttdato for gyldighet.
- **icon:** Relativ filbane til emblem.
- **classification:** Sikkerhetsgradering.
- **orientation:** Papirretning.
- **placement_strategy:** Manuell eller automatisk modulplassering.
- **algorithm_bin:** Algoritme som velger hvilken side moduler skal plasseres i. Les mer om dette i seksjon 6.6.1.
- **algorithm_pack:** Algoritme som plasserer moduler innad i en side. Les mer om dette i seksjon 6.6.1.
- **algorithm_sort:** Størrelses-egenskap som modulene sorteres etter, eller ingen sortering. Les mer om dette i seksjon 6.6.1.
- **modules og attachments:** Inneholder data om alle moduler.
 - **type:** Navnet på klassen modulen tilhører.
 - **data:** Modulens innhold.
 - **meta:** Data som trengs for å plassere og navngi modulen.
 - **x og y:** Hvor på siden modulen er plassert.

- **page**: Hvilken side modulen tilhører.
- **name**: Automatisk eller brukerbestemt modulnavn.

Kodeliste 3.1: Serialisert SOI-schema

```
{
  "title": And(str, len),
  "description": str,
  "version": And(str, len),
  "date": Or(str, None),
  "valid": {"from_date": Or(str, None), "to_date": Or(str, None)},
  "icon": Or(str, None),
  "classification": And(str, len, Or("UGRADERT", "BEGRENSET", "KONFIDENSIELT",
    "HEMMELIG", "STRENGT HEMMELIG")),
  ),
  "orientation": And(str, len, Or("portrait", "landscape")),
  "placement_strategy": And(str, len, Or("manual", "auto")),
  "algorithm_bin": And(str, len, Or("BFF", "BBF")),
  "algorithm_pack": And(
    str, len, Or("MaxRectsBl", "SkylineBl", "GuillotineBssfSas")),
  ),
  "algorithm_sort": And(str, len, Or("none", "area", "width", "height")),
  "modules": [
    {
      "type": And(str, len),
      "data": object,
      "meta": {
        "x": And(Or(int, float), lambda x: x >= 0),
        "y": And(Or(int, float), lambda y: y >= 0),
        "page": And(int, lambda page: page >= 0),
        "name": And(str, len),
      },
    },
  ],
  "attachments": [
    {
      "type": And(str, len),
      "data": object,
      "meta": {
        "x": And(Or(int, float), lambda x: x >= 0),
        "y": And(Or(int, float), lambda y: y >= 0),
        "page": And(int, lambda page: page >= 0),
        "name": And(str, len),
      },
    },
  ],
}
```

3.2.2 Komprimering

For å oppnå tidskravet ved filoverføring beskrevet i seksjon 2.4.2, komprimeres den serialiserte JSON-strengen tapsfritt. Resultatet av komprimeringen er bytes, disse blir typekonvertert til streng før de skrives til fil for å møte kravet om US ASCII beskrevet i seksjon 2.4.2.

3.3 Database

Programvaren bruker en relasjonsdatabase som inneholder tabellene vist i figur 3.1. Det finnes ingen relasjoner mellom tabellene siden dette ikke var nødvendig.

SOI	Codebook
Title: varchar (PK) Version: varchar (PK) SOI: text Date: datetime	Word: varchar (PK) Category: varchar Type: varchar Code: varchar unique
LastUpdated	CategoryWords
Timestamp: datetime (PK)	Word: varchar (PK) Category: varchar

Figur 3.1: Database-skjema

SOI

SOI-tabellen inneholder SOI-er som skrives til databasen, og brukes til gjenåpning av disse. Med "Title" og "Version" som sammensatt hovednøkkel er tanken at databasen kan inneholde flere versjoner av samme SOI, og at SOI-er med lik tittel og versjon overskrives ved lagring, slik som lagring av filer med samme navn i et filsystem. "SOI"-kolonnen inneholder en serialisert og komprimert SOI-streng, denne beskrives i avsnitt 3.2.1.

Kodeliste 3.2: SQL-statement for SOI

```
CREATE TABLE SOI(
  Title VARCHAR NOT NULL CHECK(length(Title) > 0),
  Version VARCHAR NOT NULL CHECK(length(Version) > 0),
  SOI TEXT NOT NULL CHECK(length(SOI) > 0),
  Date DATETIME,
  PRIMARY KEY(Title, Version)
);
```

Codebook

Tabellen Codebook representerer en digital kodebok som er tenkt å inneholde ord og uttrykk som brukes av Heimevernet 1.11. Ethvert ord har en tilhørende, unik kode som oppdateres hvert døgn, i tillegg til en valgfri kategori. En del av kodeboken omtales som liten kodebok. Denne inneholder ord som brukes ofte,

eksempelvis innenfor kategorier som etterforsyning, materiell, himmelretninger, tiltak og militære avdelinger. Kolonnen ”Type” avgjør om ordene tilhører ”Stor” eller ”Liten” kodebok, som er de eneste lovlige verdiene. Når en bruker lager modulen kalt forhåndsavtalte koder, beskrevet i 2.1.2, er det ord fra liten kodebok som brukes.

Kodeliste 3.3: SQL-statement for Codebook

```
CREATE TABLE Codebook(  
  Word VARCHAR PRIMARY KEY NOT NULL CHECK(length(Word) > 0),  
  Category VARCHAR,  
  Type VARCHAR NOT NULL DEFAULT 'Stor' CHECK(Type='Stor' OR Type='Liten'),  
  Code VARCHAR UNIQUE  
);
```

LastUpdated

Tabellen LastUpdated inneholder forrige tidspunkt kodene i kodeboken ble oppdatert, det vil si ”Code”-kolonnen i tabellen Codebook. Dette tidspunktet sjekkes ved oppstart av programmet. Dersom tidspunktet er mer enn 24 timer gammelt, vil kodeboken oppdateres umiddelbart. Dersom tidspunktet er mindre enn 24 timer gammelt, vil det startes en nedtelling på resterende tid, og kodeboken vil automatisk oppdateres når tiden går ut.

Kodeliste 3.4: SQL-statement for LastUpdated

```
CREATE TABLE LastUpdated(  
  Timestamp DATETIME PRIMARY KEY  
);
```

CategoryWords

Tabellen CategoryWords inneholder ord som er innenfor tydelige kategorier. Eksempler på kategorier er ”presidenter”, ”hunderaser” og ”land”, og ord i førstnevnte kategori kan være ”Trump”, ”Obama”, ”Bush” og ”Nixon”. Tilfeldige ord innenfor samme kategori brukes som kodeord for uttrykkene i SOI-modulen ”Kodefraser” 2.1.2. Tanken er at brukere av SOI vil gjenkjenne kategorien ”hunderaser” når noen sier ”puddel” over samband, og raskt forstå hvilken del av SOI-en meldingen gjelder.

Kodeliste 3.5: SQL-statement for CategoryWords

```
CREATE TABLE CategoryWords(  
  Word VARCHAR PRIMARY KEY,  
  Category VARCHAR NOT NULL CHECK(length(Category) > 0)  
);
```


Kapittel 4

Design av brukergrensesnitt

Dette kapittelet presenterer designet av brukergrensesnittet til programvaren. Figurer i kapittelet er hentet fra det ferdige produktet, selv om designvalg ble gjort før utviklingen startet som utdypet i seksjon 5.4. Dette er gjort konsekvent for å gjøre presentasjonen i dette kapittelet presis. Papirprototypen som var grunnlaget for designet finnes i vedlegg C.

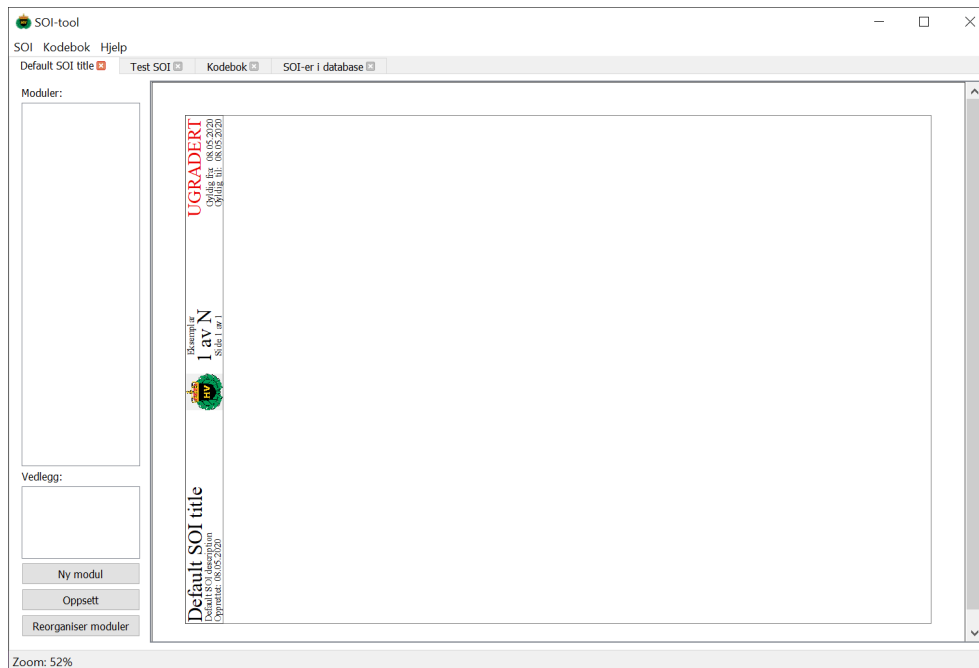
4.1 Hovedvinduet

Hovedvinduet for applikasjonen sto sentralt i utviklingfasen. Designet av det grafiske brukergrensesnittet baserer seg på design-tankemåten: ”Design for noen, ikke for alle”. Siden brukere av denne applikasjonen vil være personell i Heimevernet, som det stilles særskilte helsekrav til¹, ble universell utforming ingen stor prioritet. Dette betyr i praksis at det ikke finnes mange måter å utføre samme handling på for å tilfredstille flere brukergrupper. Applikasjonens grensesnitt gir derfor to fremgangsmåter. Hurtigtaster er lagt inn for å øke effektiviteten hos erfarne brukere av denne applikasjonen, da samme handling ofte forekommer ved bruk av programvaren.

Ulike grafiske elementer som handlingsmessig hører sammen har blitt gruppert i samme område. Det er spesielt to grupperinger som foregår her; de elementene som er globale og endrer på applikasjonen og de lokale som endrer på en bestemt SOI. De globale elementene er toppmenyen og dens valg, mens de lokale er valg av nye moduler, endring av oppsett og listene over moduler i SOI-en, se figur 4.1.

Hovedvinduet har fane-funksjonalitet som muliggjør arbeid på flere SOI-er ”samtidig”. En fane kan enten inneholde en SOI, en oversikt over SOI-er i database eller kodebok.

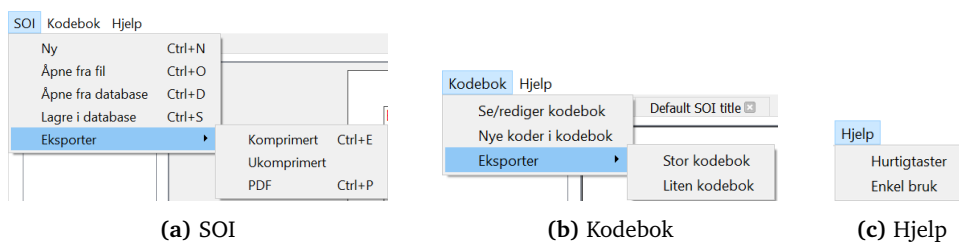
¹<https://forsvaret.no/karriere/krav/medisinske-krav>, aksessert 16.05.2020.



Figur 4.1: Hovedvindu

4.1.1 Toppmeny

Toppmenyen har blitt utformet og plassert på gjenkjennelig vis fra andre klassiske skrivebordsapplikasjoner. Dette gjør det enklere for brukeren å finne menyen og vite hva de ulike valgene gjør. Toppmenyen består av de tre dropdown-menyene "SOI", "Kodebok" og "Hjelp". En oversikt over hvilke undermenyer toppmenyen har kan man se på figur 4.2. Toppmenyen er designet slik at den inneholder få menyvalg, har korte navn og hvert menyvalg er kategorisert.



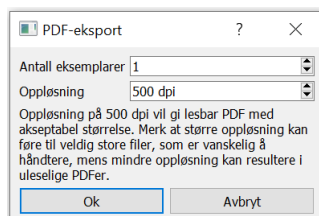
Figur 4.2: Valgene i toppmenyen

SOI

Første meny i toppmenyen er ”SOI”, som i mange andre programvarers toppmeny kalles ”Fil”. I denne menyen kan man:

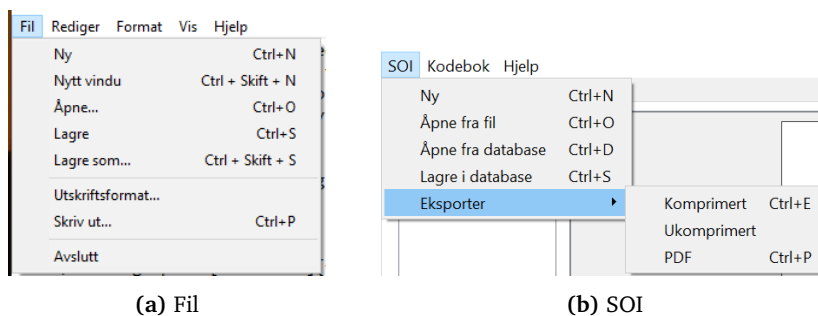
1. Lage ny SOI.
2. Åpne eksisterende SOI fra datamaskinens filsystem.
3. Åpne eksisterende SOI i databasen.
4. Lagre SOI til databasen.
5. Eksportere komprimert eller ukomprimert SOI til fil.
6. Eksportere SOI som PDF.

Om man velger ”Ny”, ”Åpne fra fil” eller ”Åpne fra database”, vil aktuell SOI åpnes i en ny fane. Valgene ”Lagre i database” eller ”Eksporter” lagrer enten SOI-en i den aktive fanen i henholdsvis databasen eller på maskinens filsystem. Om



Figur 4.3: Eksport av PDF

man velger å eksportere en SOI som PDF, åpnes dialogvinduet vist i figur 4.3 opp. Her kan man velge antall eksemplarer og oppløsning for PDF.

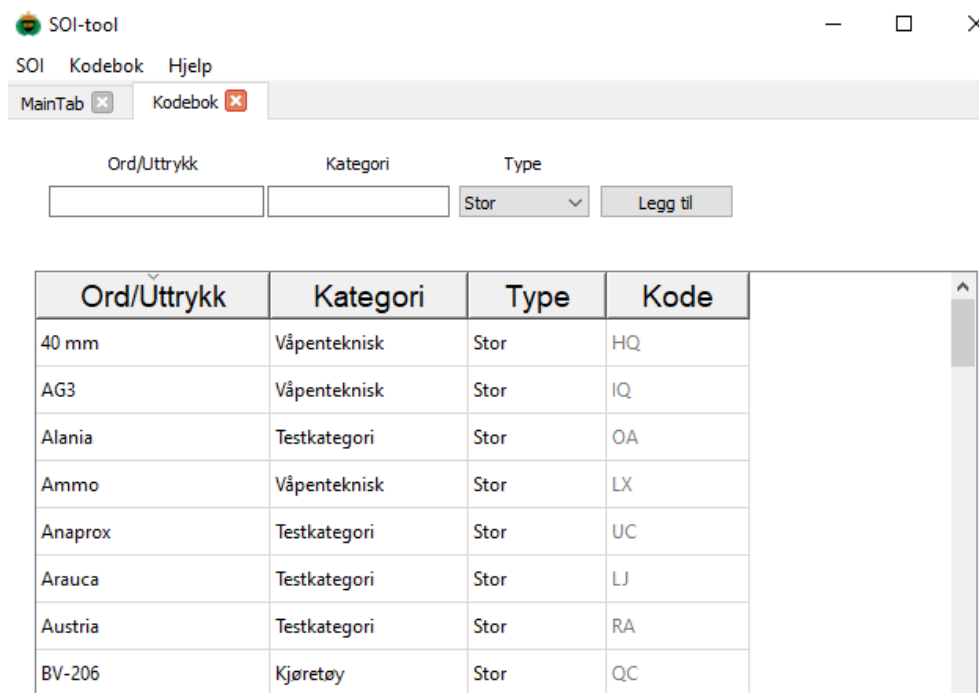


Figur 4.4: Likhet mellom Fil og SOI

Som nevnt innledningsvis i seksjonen er denne menyen veldig lik menyen ”Fil” i andre applikasjoner. Her brukes notisblokk på Windows som et eksempel. På figur 4.4 kan man se at ”Fil” og ”SOI” inneholder lik type funksjonalitet. Dette gjør at brukerne enklere vil kjenne seg igjen i bruken av denne menyen.

Kodebok

Gjennom menyen kodebok i toppmenyen kan man generere nye koder i databasen, eksportere kodeboken eller åpne/redigere den. Om man velger å åpne kodeboken, åpnes en ny fane som vist i figur 4.5. Her ser man kodeboken som en stor tabell. Kolonnene kan klikkes på for alfabetisk sortering. Det er også mulig å legge inn nye ord eller uttrykk, og velge om de skal tilhøre stor eller liten kodebok. Dette gjøres via inntastingseksjonene på toppen av kodeboken.



Figur 4.5: Utklipp av grensesnitt mot kodebok

Om man velger "Eksporter" under "Kodebok" i toppmenyen, får man valget mellom stor og liten kodebok. Begge de eksporterte kodebøkene følger samme tankegang som ordbøker for språkoversettelse. Første halvdel kalles kodebok og er tilpasset koding ved at den sorteres på ord/uttrykk, se figur 4.6a, og andre halvdel kalles dekodebok og er tilpasset dekoding ved at den sorteres på kode, se figur 4.6c. Mellom de to halvdelene finnes en side som tydelig markerer skillet på en slik måte at siden raskt oppdages når utskrevet dokument blas gjennom, se figur 4.6b. Hver del har individuelle "Side X av Y" for å markere skillet enda tydeligere. Forskjellen mellom eksportert utgave av stor og liten kodebok er overskriftene i tillegg til at "Type"-kolonnen ikke vises for liten kodebok fordi alle rader vil ha verdien "Liten".

Kodebok				Dekodebok			
Ord/Uttrykk	Kode	Kategori	Type	Kode	Ord/Uttrykk	Kategori	Type
0	CN	Tall	Liten	AA	expropriated	Testkategori	Stor
1	DD	Tall	Liten	AB	perenniality	Testkategori	Stor
2	RN	Tall	Liten	AC	juddered	Testkategori	Stor
3	KU	Tall	Liten	AD	omers	Testkategori	Stor
4	CD	Tall	Liten	AE	streaked	Testkategori	Stor
40 mm	ND	Våpenteknisk	Stor	AF	ephebiphobia	Testkategori	Stor
5	ES	Tall	Liten	AG	nonrecombinant	Testkategori	Stor
6	PH	Tall	Liten	AH	ebullient	Testkategori	Stor
7	VK	Tall	Liten	AI	invasion	Testkategori	Stor
8	IK	Tall	Liten	AJ	Y	Bokstaver	Liten
9	SU	Tall	Liten	AK	lathery	Testkategori	Stor
A	SY	Bokstaver	Liten	AL	irrigationist	Testkategori	Stor
AG3	UO	Våpenteknisk	Stor	AM	M04	Bekledning	Stor
Alania	RI	Testkategori	Stor	AN	unblooded	Testkategori	Stor

(a) Kodebok

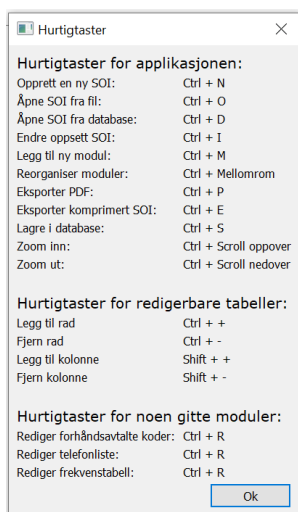
(b) Skilleside

(c) Dekodebok

Figur 4.6: Utklipp fra eksportert utgave av stor kodebok

Hjelp

Menyen "Hjelp" inneholder to menyvalg: "Hurtigtaster" og "Enkel bruk". Velger brukeren "Hurtigtaster" vises dialogen i figur 4.7a. Her får brukeren en oversikt over alle hurtigtastene som støttes i programvaren, både hurtigtaster for hele programmet og hurtigtaster for utvalgte moduler. Mange av hurtigtastene vises også ved siden av de tilhørende menyvalgene i toppmenyen, noe man kan se på figur 4.2. Trykker brukeren derimot på "Enkel bruk" vil dialogen i figur 4.7b vises. Her vil brukeren få hjelp til hvordan programvaren kan brukes. Om brukeren for eksempel trenger hjelp til å redigere en SOI, blar brukeren ned til "Rediger SOI" og leser gjennom det avsnittet.



(a) Hurtigtaster



(b) Enkel bruk

Figur 4.7: Hurtigtaster og enkel bruk

4.1.2 Modullister

På venstre side av applikasjonen finner man to lister som inneholder alle modulene til SOI-en i aktiv fane. Den øverste listen inneholder de modulene som tilhører SOI-en og den andre listen er for moduler som er vedlegg. Plasseringen av modulene i listene korrelerer til plasseringen inne i SOI-en. Dette betyr at modulen som ligger først i listen havner øverst til venstre i SOI-en. I figur 4.8 ser man hvordan modulene vises i listen. Om man tar utgangspunkt i denne, vil ”Forhåndsavtalte koder 8” komme før ”Frekvenstabell 5” siden den ligger over i listen.

Moduler:

Forhåndsavtalte koder 8
Frekvenstabell 5
Telefonliste 6
Autentiseringstavle 2
Fritekst 4
Subtraktorkoder 3
Kodefraser 7
Generisk tabell 1

Figur 4.8: Modulliste

4.1.3 Ny modul

For å legge til moduler i SOI-en må man trykke på ”Ny modul”. Denne knappen ligger under modullistene og åpner en dialog der brukeren kan gjøre ulike valg. Dialogen ser man i figur 4.9. Denne dialogen gir brukeren en oversikt over hvilke moduler som kan velges, hvilket navn hver modul skal ha og om de skal legges inn i vedlegglisten. Moduler blir som standard lagt inn i modullisten. I oversikten ser man også små ikoner av hver modul og hvordan de ser ut. Å legge til en ny modul er også lagt inn som en hurtigtast, da dette er en oppgave som kommer til å bli gjort flere ganger.

The screenshot shows a window titled "Ny modul" with a close button (X). The window contains several sections:

- Generisk tabell:** A table with columns A, B, C and rows 4, 6, tekst; 8, 12, mer tekst; 16, 24, tekst.
- Autentiseringstavle:** A table with columns A, B, C and rows 4, 6, tekst; 8, 12, mer tekst; 16, 24, tekst.
- Subtraktorkoder:** A table with columns A, B, C and rows 4, 6, tekst; 8, 12, mer tekst; 16, 24, tekst.
- OVERSKRIFT:** A section with a title "OVERSKRIFT" and a list of items: "Innhold innhold innhold", "Liste", "Punkt 1", "Punkt 2", "Punkt 3".
- Fritekst:** A section with a title "Fritekst" and a list of items: "Kodefraser", "Frekvenstabell", "Telefonliste".
- Kodefraser:** A table with columns A, B, C and rows 4, 6, tekst; 8, 12, mer tekst; 16, 24, tekst.
- Frekvenstabell:** A table with columns A, B, C and rows 4, 6, tekst; 8, 12, mer tekst; 16, 24, tekst.
- Telefonliste:** A table with columns A, B, C and rows 4, 6, tekst; 8, 12, mer tekst; 16, 24, tekst.
- Forhåndsavtalte koder:** A table with columns A, B, C and rows 4, 6, tekst; 8, 12, mer tekst; 16, 24, tekst.

At the bottom of the window, there is a "Navn:" field with the text "La stå tom for automatisk navn", a "Vedlegg:" checkbox, and "Ok" and "Avbryt" buttons.

Figur 4.9: Oversikt modulvalg

4.1.4 Oppsett

For å endre innstillingene eller oppsettet for en SOI trykker man på "Oppsett". Når denne knappen har blitt trykket på, eller hurtigtast har blitt brukt, åpnes dialogen i figur 4.10. Oppsettdialogen inneholder innstillinger for å endre utseende på SOI. Her kan man endre på:

- Toppsegment ("Headerdata")
- Papirretning
- Modulplassering
- Plasseringsalgoritmer

Endring av toppsegmentet til en SOI skjer når man endrer innholdet i innskrivingsfeltene i dialogen og trykker "Lagre". Med det kan man endre tittel, beskrivelse, versjon, dato og klassifisering hos den valgte SOI-en. Når man lagrer vil man se at toppsegment hos SOI-en er endret. Figur 4.11 viser to ulike toppsegment, den første er standard toppsegment som genereres på en ny SOI og den andre er et toppsegment som har blitt endret.

Papirretningen bestemmer om SOI-en skal være horisontal eller vertikal. Modulplasseringen bestemmer om brukeren kan flytte modulene i SOI-en, eller om programvaren skal gjøre det automatisk. Om man endrer til "Manuell", kan brukeren flytte modulene i modullisten og bestemme hvilke moduler som skal plasseres først. Plasseringsalgoritmene bestemmer hvordan modulene skal pakkes og plasseres på SOI-en. Når man er ferdig med å endre innstillingene, kan man enten lagre eller forkaste endringene. Her kan man lagre med "Enter" og avbryte med "Esc", eller bruke knappene i dialogen.

Figur 4.10: Innstillinger til SOI

(a) Standard toppsegment

(b) Endret toppsegment

Figur 4.11: SOI-toppsegment

4.2 SOI

SOI-ens design står sentralt i applikasjonen og i utskrevet produkt. Følgende seksjoner beskriver designet til ulike aspekter ved SOI, fra brukeropplevelse ved redigering til utseende på papir.

4.2.1 Redigering av SOI

Siden SOI-en består av et toppsegment og et område for moduler, vil redigering av SOI foregå på to måter. Redigering av toppsegmentet gjøres, som nevnt i 4.1.4, gjennom oppsettdialogen. Skal man derimot redigere modulene, kan dette gjøres direkte i dem. Redigeringen av SOI-moduler er bygget på prinsippet ”WYSIWYG”. WYSIWYG står for ”What You See Is What You Get” og betyr at SOI-ens utseende i applikasjonen tilsvarer hvordan SOI-en vil se ut som papirutgave. Siden programmet er basert på WYSIWYG vil SOI-en i figur 4.12 se lik ut under redigering. Området for redigering støtter WCAG 2.0 suksesskriterie 1.4.4 på nivå AA, som forteller at tekst skal kunne endres til 200% størrelse uten tap av innhold eller funksjon². Tap av innhold løses med scrollbar.

Test SOI

Default SOI description

Opprettet: 05.05.2020

Eksempel

I av 1

Side 1 av 1

UGRADERT

Gjeldig fra: 05.05.2020

Gjeldig til: 05.05.2020

Forhåndsavtalte koder

Varslingsord: TOYOTA

A Bokstaver	B Landemerker	E Tall
A P	A Elv	A 0
B V	B Sti	B 6
C D	C Bø	C 7
D M		D 2
E X	C Retninger	E 5
F L	A Vest	F 8
G Q	B Sør	G 9
H B	C Fram	H 3
I K	D Tilbake	I 1
J C	E Nord	J 4
K E	F Øst	
L T		
M U	D Støtte	F Våpenteknikk
N H	A Sanitet	A Bombe
O G	B Helikopter	B Håndgranat
P I	C MP	C MP-5
Q A	D K9	D HK-416
R Y		
S J		
T W		
U S		
V F		
W N		
X Z		
Y R		
Z O		

FREKVENSTABELL			
Nett	MRR		
	Nett-ID	Type	Område
Hæren Brig-N INGBN	2000	CNR	TD Norge
Hæren Brig-N INGBN	2001	CNR	TD Norge
Hæren Brig-N INGBN	2002	CNR	TD Norge
Hæren Brig-N INGBN	2003	CNR	TD Norge
Hæren Brig-N INGBN KP Samvirke	2090	CNR	TD Norge
Hæren Brig-N INGBN KP Samvirke	2091	CNR	TD Norge
Hæren Brig-N INGBN	2092	CNR	TD Norge
Hæren Brig-N INGBN	2093	CNR	TD Norge
Hæren Brig-N INGBN	2095	CNR	TD Norge

Prosedyre ved manglende VHF-forbindelse

1. Kontroller NET ID
2. Kontroller fysisk kabling (Strøm, antenne, mikrofon)
3. Kontroller tid/synk status og sendeeffekt
4. Kontroller kryptonøkkel
5. Kontroller at du er innenfor dekningshele fra reie
6. Kontroller endringer i området (store antenner, aggregat, jammere)
7. Forflytt deg til siste posisjon hvor du hadde samband

SUBTRAKTORKODER	
A	9446 7304
B	3039 5314
C	6062 0151
D	8795 0395
E	7502 7404
F	8926 7340
G	1833 1811

KODEFRASER	
Kode	Fraser
Husky	Start radiotaushet
Terrier	Stopp radiotaushet
Puddel	Kompromittert radio
Labrador	Kompromittert pc

TELEFONLISTE			
Funksjon	Telefon	FDN	Iridium
INGBN OPS Devdis	77 89 21 16	0580 2116	
INGBN OPS felt	488 84 500		
BRIGKO	400 29 999	(0)580 8500	
BrigKp-1	488 88 511		+88 165 1437 527
IngKp-2	488 88 521		+88 165 1437 528
BrigKp-3	488 88 531		
IngKp-4	488 88 541		+88 165 1437 525
IngKp-5	488 88 551		
IngKp-6	488 88 561		+88 165 1437 524
Skredgruppe			+88 165 1437 311

AUTENTISERINGSTAVLE (21 & 25)			
A	0	CVWQ	QODIE
B	1	KBYAA	DRTNB
C	2	PFQEX	XGONN
D	3	VQYRH	SAGOX
E	4	IRLMT	DTQPR
F	5	OLRFI	INWVB
G	6	XJRVU	QZTTU
H	7	UDQDE	QGVRF
I	8	VVAYU	PYEHT
J	9	UXKZQ	KCZOV

AREUG	UQPA	HLDJH
HVPK	FRUDT	SYDKP
HZNZC	DYRDX	SBTCD
NEVKI	BOFE	ORYGA
MGNBP	SHYVO	DXTQO
DAQHG	CHBZG	ITZNZ
SOMUL	BLSQB	PNKLW
OBWSH	JHCXA	GVZFS
FZNPC	MYRRR	DYHGX
FVYFF	VWQJY	REHGO

A	B	C
4	6	tekst
8	12	mer tekst
16	24	enda mer tekst

Figur 4.12: SOI med moduler

²<https://uu.difi.no/krav-og-regelverk/wcag-20-standard/144-endring-av-tekststorrelse-niva-aa>, aksessert 17.05.2020.

4.2.2 Toppsegment til SOI

Toppsegmentet til SOI-en er plassert vertikalt for å være plassbesparende. Ved å plassere segmentet i denne retningen slipper man at det strekkes og at man får mye rom mellom hvert element i toppsegmentet. Fargen på graderingsmerket er rød for å oppfylle krav 2.1.1. I midten av toppsegmentet står eksemplarnummer og totalt antall eksemplarer av SOI-en med stor skrift. Det er plassert tydelig for å gjøre håndtering av SOI enklere under innsamling og utdeling.

4.2.3 SOI-Moduler

Generisk tabell og fritekst

I programvaren kan man velge to lavnivås moduler, hvorav den ene er en tabell og den andre er en tekstboks. Disse kan fylles med vilkårlig innhold. I figur 4.13 kan man se modulene fylt med innhold.

A	B	C
4	6	tekst
8	12	mer tekst
16	24	enda mer tekst

(a) Generisk tabell

Prosedyre ved manglende VHF-forbindelse
1. Kontroller NET ID
2. Kontroller fysisk kabling (Strøm, antenne, mikrofon)
3. Kontroller tid/synk status og sendeeffekt
4. Kontroller kryptonøkkel
5. Kontroller at du er innenfor dekningsboble fra relé
6. Kontroller endringer i området (store antenner, aggregat, jammere)
7. Forflytt deg til siste posisjon hvor du hadde samband

(b) Fritekst

Figur 4.13: Generisk tabell og fritekst

Autentiseringstavle og subtraktorkoder

Velger man autentiseringstavle eller subtraktorkode vil man få opp et dialogvindu som vist i figur 4.14. Begge modulene, som man kan se i figur 4.15, har samme type dialogvindu. I dette vinduet kan man gjøre endringer på modulen som for eksempel å endre overskrift eller lengde på kodene.

python

Overskrift SUBTRAKTORKODER

Antall koder 7

Kodelengde 8

Intervall for kodemellomrom 4

Mellomrom per intervall 3

Opprett

Figur 4.14: Opprettelsesdialog

AUTENTISERINGSTAVLE (21 & 25)						
A	0	CVHWQ	QQDIE	AREUG	UTQPA	HLDJW
B	1	KBYAA	DRTWB	HVPMK	FRUDT	SYDKP
C	2	PFQRX	XWXNN	HMZNC	DYRDX	SBTCD
D	3	VOYRH	SKODX	NEVKI	BDFZE	OXYGA
E	4	IRLMT	DTQPR	MGNBP	SHYVD	DXTQQ
F	5	OLRFT	INVVB	DAQHG	CHBZG	ITZNZ
G	6	XJRVU	QZTTU	SOMUL	BLSQB	PNKLW
H	7	UDJQE	QGVRF	OBWSH	JHCXA	GVZFS
I	8	VVAYU	PYEHT	FZNPC	MYYRR	DYMWX
J	9	UXRZQ	MCZOV	FVYYF	VWOJY	REHGO

(a) Autentiseringstavle

SUBTRAKTORKODER		
A	9446	7304
B	3039	5314
C	6062	0151
D	8795	0395
E	7502	7404
F	8926	7340
G	1833	1811

(b) Subtraktorkoder

Figur 4.15: Autentiseringstavle og subtraktorkoder

Frekvenstabell

I frekvenstabell vist i figur 4.16a kan brukeren selv velge hvilke kolonner som skal være med i modulen. For å gjøre dette må brukeren bevege musepekeren over modulen og velge ”Kolonner”-knappen som dukker opp under modulen. Trykker man på knappen eller bruker hurtigtast åpnes dialogen vist i figur 4.16b. Her har man flere valg for forskjellige radiotyper og deres kolonner.

FREKVENSTABELL			
	MRR		
Nett	Nett-ID	Type	Område
Hæren Brig-N INGBN	2000	CNR_TD	Norge
Hæren Brig-N INGBN	2001	CNR_TD	Norge
Hæren Brig-N INGBN	2002	CNR_TD	Norge
Hæren Brig-N INGBN	2003	CNR_TD	Norge
Hæren Brig-N INGBN KP Samvirke	2090	CNR_TD	Norge
Hæren Brig-N INGBN KP Samvirke	2091	CNR_TD	Norge
Hæren Brig-N INGBN	2092	CNR_TD	Norge
Hæren Brig-N INGBN	2093	CNR_TD	Norge
Hæren Brig-N INGBN	2095	CNR_TD	Norge
Kolonner	+	-	

(a) Frekvenstabell

Velg kolonner til frekvenstabell

MRR	HF	Hytera	LFR
<input checked="" type="checkbox"/> Frekvenser <input checked="" type="checkbox"/> Nett-ID <input checked="" type="checkbox"/> NIS <input type="checkbox"/> Type <input type="checkbox"/> ID/CH <input type="checkbox"/> Område <input type="checkbox"/> Z <input type="checkbox"/> Key ID <input type="checkbox"/> Radio <input type="checkbox"/> SEC <input type="checkbox"/> Sync Master <input type="checkbox"/> Access <input type="checkbox"/> SMTF Host Name	<input checked="" type="checkbox"/> Frekvenser <input checked="" type="checkbox"/> Kodeord <input checked="" type="checkbox"/> NIS <input type="checkbox"/> Type <input type="checkbox"/> ID/CH <input type="checkbox"/> Z <input type="checkbox"/> Key ID <input type="checkbox"/> Radio <input type="checkbox"/> SEC <input type="checkbox"/> Sync Master <input type="checkbox"/> Access <input type="checkbox"/> SMTF Host Name	<input checked="" type="checkbox"/> Kodeord <input checked="" type="checkbox"/> Kodeord	<input checked="" type="checkbox"/> Kanal nr. <input checked="" type="checkbox"/> Kanal nr.

Veksle alle Veksle alle Veksle alle Veksle alle Veksle alle

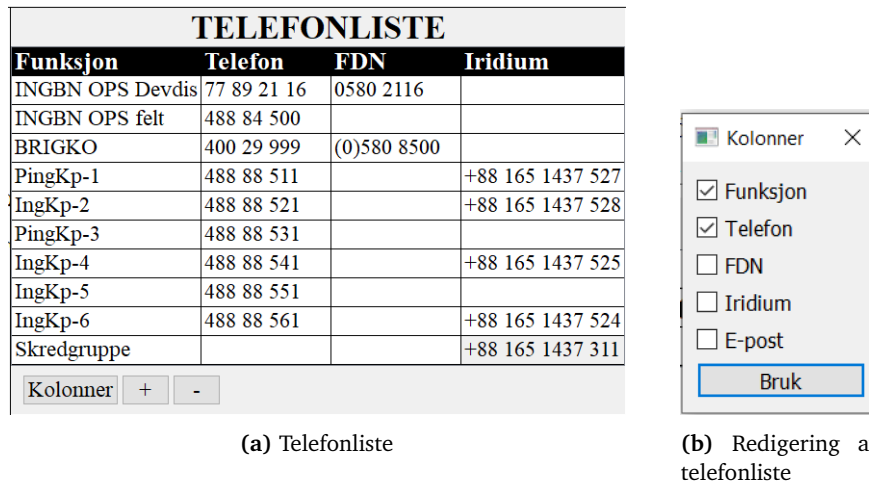
Bruk

(b) Redigering av frekvenstabell

Figur 4.16: Frekvenstabell

Telefonliste

Telefonliste vist i figur 4.17 har samme grensesnitt som frekvenstabell beskrevet i avsnitt over.



Figur 4.17: Telefonliste

Kodefraser

I kodefraser-modulen vist i figur 4.18 kan man kun skrive i andre kolonne, da første kolonne automatisk utfylles med kodeord. Brukeren kan legge til rader ved hjelp av hurtigtaster. Da vil "Kode"-kolonnen i den nye raden automatisk fylles ut med ord i samme kategori som i resten modulen. Brukeren kan også fjerne rader ved hjelp av hurtigtaster.

KODEFRASER	
Kode	Frase
Husky	Start radiotaushet
Terrier	Stopp radiotaushet
Puddel	Kompromittert radio
Labrador	Kompromittert pc

Figur 4.18: Kodefraser

Forhåndsavtalte koder

Velger man modulen ”Forhåndsavtalte koder”, vist i figur 4.19a, vil man få opp en dialog før modulen legges til i SOI-en. I dialogen, vist i figur 4.19b, kan man endre overskriften, varslingsordet, den maksimale kolonnehøyden og kategori-rekkefølgen til modulen, mer om disse i seksjon 6.5.4. Som nevnt i 4.1.1 inneholder oppføringer i kodeboken en ”Type” som er ”Stor” eller ”Liten”, forhåndsavtalte koder viser ord og uttrykk som har ”Type” ”Liten”.

FORHÅNDSAVTALTE KODER			
Varslingsord: TOYOTA			
A Bokstaver	B Landemerker	E Tall	
A N	A Bro	A	0
B P	B Elv	B	7
C L	C Sti	C	5
D G	C Retninger	D	3
E E		E	4
F B		F	1
G F		G	2
H V		H	9
I I		I	8
J H		J	6
K U	D Støtte	F Våpenteknisk	
L T		A	MP-5
M K		B	HK-416
N S		C	Bombe
O W	D K9	D	Håndgranat
P X			
Q J			
R Q			
S M			
T Y			
U A			
V R			
W O			
X Z			
Y D			
Z C			

(a) Forhåndsavtalte koder

(b) Redigering av modulen

Figur 4.19: Forhåndsavtalte koder

4.2.4 Font og kontrast

Tidligere i rapporten, under brukervennlighet 2.4.1, beskrives et krav om at SOI skal være lettleselig i utskrevet form. Det er derfor viktig å velge riktig fonttype og skriftstørrelse, i tillegg til tilstrekkelig kontrast mellom tekst og bakgrunn. Da det er viktig å ha god leselighet på papir, velges en "serif"-font. Slike fonter brukes mye for skrift som skal skrives ut, for eksempel i aviser eller magasiner³. "Times New Roman" velges fordi denne opprinnelig ble designet til bruk på utskrevet media [1]. Det blir også sagt at Times New Roman er en av de beste fontene å lese på papir [2].

For å øke lesbarheten ytterligere, ble kontrasten på utskrevet SOI sjekket opp mot WCAG 2.0 suksesskriterie 1.4.3 på nivå AA⁴⁵. Kriteriet forteller at kontrastforholdet mellom normal tekst og bakgrunn bør være minst 4.5:1, og 3:1 for stor tekst, og SOI-en ble justert for å oppfylle dette.

³<https://en.wikipedia.org/wiki/Typography>, aksessert 16.05.2020.

⁴<https://uu.difi.no/krav-og-regelverk/wcag-20-standard/143-kontrast-minimum-niva-aa>, aksessert 17.05.20

⁵<https://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html>, aksessert 17.05.2020.

Kapittel 5

Utviklingsprosess

Dette kapittelet beskriver ulike aspekter ved utviklingsprosessen som ble fulgt under prosjektperioden. Seksjonene følger løst rekkefølgen de ble utført. Grunnlaget for utviklingsprosessen er prosjektplanen lagt til som vedlegg E.

5.1 Utviklingsmodell

Valgt utviklingsmodell har elementer av både planbasert og smidig metodikk. Siden oppgaven har en fastsatt leveringsfrist og hovedtrekkene til leveransen er klargjort, har utviklingsmodellen ytre karakteristika av fossefall. Gruppen ønsket derimot å angripe utviklingsarbeidet i oppgaven i den rekkefølgen som falt seg naturlig under utvikling. Dermed kombineres planbasert og smidig metodikk ved at oppgaven planlegges med fossefall-tilnærming på høynivå, mens utviklingsdetaljer utføres smidig. Gantt-diagrammet laget i seksjon 5.2 speiler dette, da den viser selve utviklingen som én sammenhengende aktivitet. Rekkefølgen av ”MVP”-er, ”Minimum Viable Product” eller ”Minste Gyldige Produkt”, i Gantt er ment å være løs.

Underveis i prosjektet ble detaljer ved oppgaven stadig oppdatert gjennom møter med oppdragsgiver. Smidig metodikk gjorde at gruppen var forberedt på dette, mens den fossefallinspirerte høynivå-planleggingen holdt gruppen klar over tidsfrister og mål.

5.2 Planlegging og estimering

Under prosjektplanleggingsfasen ble prosjektet delt opp i underoppgaver, basert på daværende forståelse av hva som måtte gjøres. Oppgavene ble estimert ved bruk av ”Planning Poker”¹, og resultatet av vurderingen ble et Gantt-diagram vist

¹”Planning Poker” er en aktivitet for estimering. Hvert gruppemedlem får utdelt kort med ulike tallverdier, og aktiviteten går over flere runder. I hver runde estimerer gruppemedlemmer aktuell oppgave med kort hver for seg, før alle kort vises og estimeringer diskuteres. Dette fortsetter til gruppen kommer til enighet.

som vedlegg B.1. For å planlegge utviklingsarbeidet ble kjernefunksjonaliteter i programvaren delt opp i MVP-er. En MVP beskrev et hendelsesforløp som utnyttet en funksjonalitet, for eksempel ”Velge modul, redigere, se forhåndsvisning, skrive ut PDF”. Estimering ble gjort på dagsverk-nivå, slik at en oppgave estimert til to dagsverk enten kunne utføres av ett gruppemedlem på to dager eller av to gruppemedlemmer på én dag. Prosjektplanleggingsfasen er dokumentert i eget prosjektplanleggingsdokument lagt til som vedlegg E.

Før utviklingsperioden begynte ble det gjennomført en ny estimeringsrunde, da med en mer konkret forståelse for oppgaven. Resultatet av denne estimeringen ble prioritering av oppgaver i arbeidstavlen beskrevet i seksjon 5.5. Estimering ble også gjort fortløpende under utviklingsperioden for å oppdatere arbeidstavlen.

5.3 Research

Som planlagt i Gantt-diagrammet, vedlegg B.1, ble to uker brukt til research. En liste med forutsette teknologiske problemstillinger ble satt opp og punktene ble fordelt ut til gruppemedlemmer slik at gruppen som helhet ville ha kunnskaper for å dekke alle områdene. Siden brukergrensesnitt er en sentral del av oppgaven brukte alle gruppemedlemmer tid på å sette seg inn i bibliotek for dette. På slutten av research-perioden presenterte hvert av gruppemedlemene sitt arbeid for gruppen slik at alle fikk et innblikk i research på de ulike områdene. Alt arbeidet som kom ut av perioden ble lastet opp til et ”Git-repository” slik at gruppen hadde tilgang til hverandres arbeid.

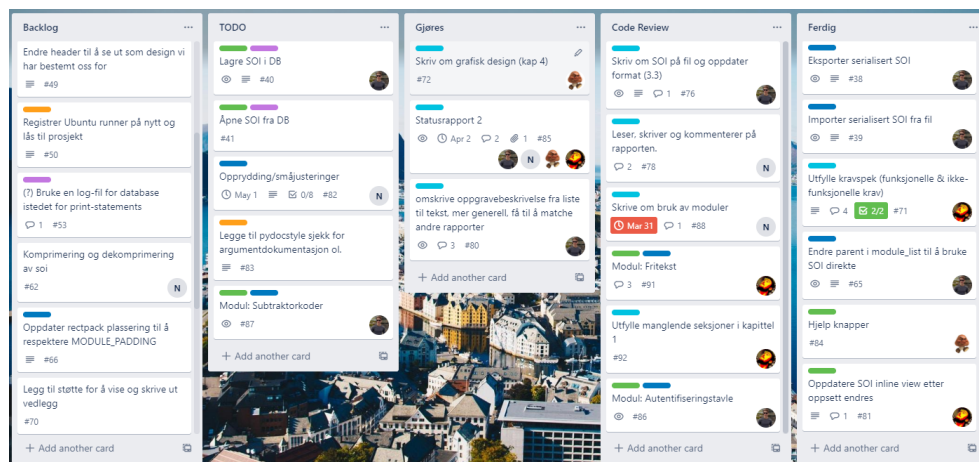
Resultatet av research ble kunnskap om flere alternative løsninger for hvert forutsette problem. Denne kunnskapen ble brukt til å gjøre endelige valg på hvordan problem skulle løses. Kapittel 8 inneholder diskusjon rundt alternative løsninger funnet under research-perioden.

5.4 Designprosess

Det ble gjennomført en iterativ designprosess med prototyping av det grafiske brukergrensesnittet. For at de tidlige prototypene ikke skulle være begrensende, ble det besluttet å tegne ”lo-fi” prototyper på ark. Det ble satt en tidsramme på 20 minutter der gruppemedlemmene fikk egne design-idéer ned på papir. Prototypene ble gjennomgått og diskutert i fellesskap, og prosessen gjentok seg flere ganger. Da vi ble enige om hoved-design bevegde vi oss videre mot en mer detaljert prototype som var en blanding av horisontal² og vertikal³, og tok for seg kjernefunksjonalitet. Det var denne prototypen som skulle brukertestes med en representant fra Heimevernets sambandsseksjon. Derfor ble den videre raffinert og det ble gjort utklipp slik at vi kunne utføre ”Wizard of oz” under testingen, som beskrives i seksjon 7.3.

²En horisontal prototype gir oversikt uten å gå i detalj.

³En vertikal prototype går i dybden på én spesifikk ”interaksjons-sti”.



Figur 5.1: Trello-tavle

5.5 Arbeidsmetodikk

Verktøyet "Git" ble valgt for versjonskontroll, med skolens "GitLab"-tjeneste som git-hosting. All utvikling skjedde via "feature-branches", en tilnærming til Git hvor hver ny funksjonalitet utvikles på en egen "branch", og ikke "merges" inn på hoved-branch før funksjonaliteten anses som ferdig.

For å gi gruppe-medlemmer innsyn i hverandres arbeid, og for å sikre kodekvalitet, ble "Merge Requests" anvendt i GitLab. Dette er en funksjonalitet i GitLab som presenterer alle endringer en feature-branch har gjort i forhold til hoved-branch, slik at det er tydelig hva som endres. I tillegg presenteres resultat av automatiske tester som beskrives i kapittel 7. Med denne informasjonen som bakgrunn diskuterer gruppen endringen, og forbedrer den til det blir enighet. Når gruppen er fornøyd merges endringen, og Merge Request løses. En stor fordel er at historikk for Merge Requests ligger lagret, så det er mulig å se bakover i tid på alle endringer som har blitt gjort på hoved-branch, med tilhørende diskusjoner.

Utviklingen ble gjort med utgangspunkt i et typisk smidig virkemiddel - "user stories", som vi anvendte som "lapper". Trello ble brukt som verktøy for å organisere disse i en arbeidstavle, som vist i figur 5.1. Hver lapp representerer et stykke arbeid som burde utføres, og merkes med kategori og ansvarlig gruppe-medlem. Arbeidstavlen inneholder flere kolonner, og en lapp begynner i venstre kolonne og flytter seg mot høyre. Kolonnene som ble brukt var: "Backlog", "TODO", "Gjøres", "Code Review" og "Ferdig". For å flytte en lapp fra "Code Review" til "Ferdig" kreves det at et annet gruppe-medlem enn det som har utført lappen skal kvalitetsjette arbeidet. For utviklingsarbeid betyr dette i praksis gjennomgang av Merge Request som beskrevet tidligere. På denne måten er det garantert at en "Ferdig" lapp er godkjent og forstått av minst to gruppe-medlemmer. Lapper ble også automatisk nummerert av Trello, og dette nummeret ble nevnt i relevante Git commits og Merge Requests.

Kapittel 6

Implementasjon

I dette kapittelet beskrives implementasjonen av programvaren. Implementasjonen ble gjort med grunnlag i krav beskrevet i kapittel 2, teknisk design beskrevet i kapittel 3 og design av brukergrensesnitt beskrevet i kapittel 4. Kapittelet unngår konsekvent å presentere alle detaljer ved implementasjonen, og går heller i dybden på noen utvalgte aspekter som gruppen regner som viktige. En oppsummering av effektive kodelinjer¹ vises i tabell 6.1.

Kodelister som vises i dette kapittelet er ikke nødvendigvis linje for linje eksakt lik den faktiske koden, og docstrings er fjernet for å spare plass.

	Kodelinjer	Kommentarer og dokumentasjon
SOI-tool	3982	3213
Testkode	1522	621

Tabell 6.1: Effektive kodelinjer. Hentet via verktøyet "pygount".

6.1 Valgte teknologier og verktøy

Nøkkelteknologier i løsningen er Python sammen med Python-bibliotekene PySide2 [3] og SQLite3 [4]. PySide2 tilbyr tilgang til kryssplattform-biblioteket Qt [5], som originalt er skrevet i C++. Videre i rapporten vil det refereres direkte til Qt, da PySide2 i praksis bare fungerer som et mellomledd. SQLite3 tilbyr en lett diskbasert SQL-database. Begrunnelse for valg av disse utdypes i seksjon 8.1.

Gruppen brukte i hovedsak teksteditoren "Visual Studio Code", med unntak av ett medlem som foretrakk "Vim". Begge disse er å regne som lettvektede, og gjorde at utviklingen ikke ble låst til ett verktøy. De fleste på gruppen utviklet fra Windows 10, med unntak av ett medlem som valgte Ubuntu 18.04. På grunn av at valgt GUI-bibliotek er kryss-plattform var ikke dette et problem, men heller en fordel, da gruppen fikk se hvordan programvaren oppførte seg på to ulike

¹Med effektive kodelinjer menes alle kodelinjer ekskludert luft.

systemer. Sluttproduktet leveres til Windows 10, men under utvikling har altså programvaren også blitt brukt på Ubuntu 18.04.

Øvrige verktøy og teknologier som er relatert til testing beskrives i kapittel 7.

6.2 Oversikt over brukergrensesnitt

I Qt bygges brukergrensesnitt opp av ”widgets”. En widget kan vise informasjon til bruker, håndtere brukerhendelser som tasting på tastatur og klikking med mus, samt inneholde andre widgets. En widget som ikke ligger inne i en annen blir et vindu. Qt kommer med en rekke innebygde widgets som tekstfelt, tabeller, lister og knapper. Nye moduler kan utvikles ved å lage subklasser av superklassen til alle widgets, ”QWidget”, eller ved å subklasse spesifikke moduler en vil utvide. Ved å velge én widget til å være et vindu, for så å legge nye widgets inne i denne, kan man bygge det brukergrensesnittet man måtte ønske.

Brukergrensesnittet presentert i kapittel 4 implementertes ved å dele grensesnittet opp i følgende widgets. Listen under er bygd opp slik at widget på øverste nivå i praksis er et vindu, mens dypere widgets eksisterer inne i omfavnende widget. Merk at mindre widgets som brukes i en støttende rolle ikke er inkludert. Dialoger inkluderes heller ikke, da disse skapes når de trengs av widgets i listen, og alltid blir til egne vinduer. Der det er hensiktsmessig refereres det til senere seksjoner for implementasjonsdetaljer.

- **MainWindow:** Alle programvarens widgets, med unntak av dialoger inkluderes i denne. Den er basert på den innebygde widgeten ”QMainWindow”, som tilbyr toppmeny og statusbar.
 - **QTabWidget (Qt):** Innebygd widget som kan legge widgets i lukkbare faner med fanetittel.
 - **SOIWorkspaceWidget:** Presenterer et komplett brukergrensesnitt for SOI-behandling.
 - **ModuleList:** Muliggjør redigering og visualisering av alle moduler i SOI, samt prioriteringen av disse. Se seksjon 6.6.3 for hva prioritering brukes til.
 - **InlineEditableSOIView:** Muliggjør WYSIWYG-redigering av SOI, samt visualisering med støtte for zooming. Implementasjon av dette utdypes i seksjon 6.4.
 - **CodebookWidget:** Presenterer et komplett grensesnitt til bruker for behandling av kodebok.
 - **CodebookRowAdder:** Muliggjør å legge til oppføringer i kodebok.
 - **CodebookTableView:** Viser sorterte oppføringer i kodebok og muliggjør redigering og fjerning av disse.
 - **SOIDbWidget:** Presenterer et komplett grensesnitt mellom bruker

og SOI-er i databasen.

- **SOITableView**: Viser oversikt over SOI-er i databasen og muliggjør åpning av disse.

6.3 SOI-klassen

I programvaren er SOI representert som en egen klasse. Klassen har attributter for alle innstillinger knyttet til SOI som eksempelvis overskrift og opprettelsesdato, og lagrer moduler og vedleggsmoduler i lister. SOI kan vises på skjerm gjennom ulike klasser, og når SOI endrer seg er det nødvendig at disse klassene får vite om endringen. Eksempler på klasser som viser SOI på skjerm er modullisten, figur 4.8, og redigeringsvinduet, seksjon 6.4. Dette implementeres via "listeners", en programmeringsteknikk for å gjøre vilkårlige oppdateringer som reaksjon på en hendelse. SOI-klassen lagrer lister av listener-funksjoner som skal kjøres når ulike hendelser skjer, og andre klasser kan registrere listeners hos SOI-klassen. Som eksempel kan klassen "ModuleList", seksjon 6.2, som viser en liste over SOI-moduler i SOI registrere en funksjon *self.update_list* hos SOI-klassen som listener på hendelsen at en ny modul blir lagt til. SOI-klassen vil da kjøre denne funksjonen når en ny modul blir lagt til, som gir ModuleList en mulighet til å oppdatere seg selv til å inneholde den nye modulen. Listeners kan ligne på Qts konsept om "Signals and slots"². For enkelhets skyld har ikke dette blitt tatt i bruk, da implementasjon av listeners var trivielt og effektivt.

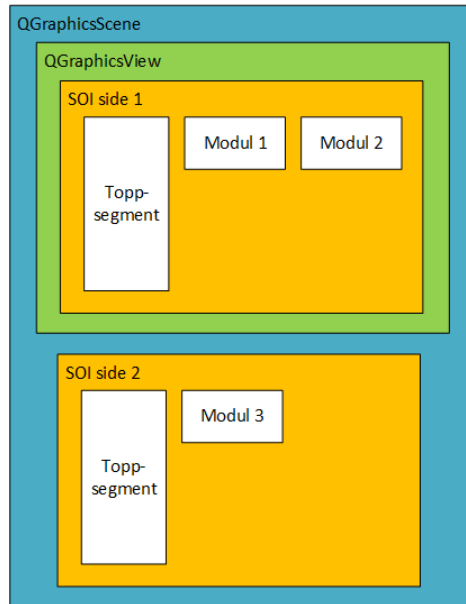
6.4 SOI-redigering

Som beskrevet i seksjon 4.2.1 skal redigering av SOI-moduler foregå med en WYSIWYG-tilnærming, altså skal dokumentet under redigering vise hvordan dokumentet blir seende ut ferdigstilt. For å oppnå dette har to widgets fra Qt kalt "QGraphicsScene" og "QGraphicsView" blitt tatt i bruk, ofte refert til heretter som henholdsvis "scene" og "view". QGraphicsScene er en beholder for to-dimensjonelle grafikk-elementer, og muliggjør brukerhendelseshåndtering for disse. QGraphicsView har i oppgave å presentere innholdet i en QGraphicsScene, og støtter skalering, fremvisning av utvalgte områder i scene, og scrollbarer for navigasjon. I praksis fungerer disse to i kombinasjon som en "tegneflate" bruker kan navigere seg rundt i, mye likt Microsoft-programmene "Paint" og "Word".

En scene består av elementer av typen "QGraphicsItem", som i utgangspunktet er adskilt fra vanlige QWidgets som knapper og tekstfelt. Qt tillater likevel at vanlige widgets blir plassert i scene via bruk av "QGraphicsProxyWidget". Dette er en QGraphicsItem som kan presentere widgets basert på QWidget i scene. QGraphicsProxyWidget åpner for at hele det omfattende biblioteket av widgets fra Qt kan brukes, noe som i stor grad effektiviserer utvikling.

²https://wiki.qt.io/Qt_for_Python_Signals_and_Slots, aksessert 16.05.2020.

Redigeringsvindu for SOI er implementert som én sammenhengende scene som strekker seg over flere SOI-sider. Elementer i SOI er i praksis QGraphicsItems plassert med x og y koordinater. Disse plasseres relativt til sin SOI-side, slik at to elementer i lik posisjon på side 1 og 2 skilles på y-aksen med høyden til én SOI-side pluss eventuell mengde luft mellom SOI-sidene.



Figur 6.1: Sammenheng mellom QGraphicsScene, QGraphicsView og elementer i SOI

Figur 6.1 viser sammenhengen mellom scene og view, og hvordan elementer plasseres for å bygge opp SOI. QGraphicsScene er farget i blått, QGraphicsView i grønt, SOI-sider i oransje og QGraphicsItems i hvitt. Scene har bredde lik bredden på SOI-ark, og høyde lik høyden til alle sider i SOI pluss luft mellom disse. I figuren er view satt til å vise frem side 1 i SOI, og det er bare dette bruker i realiteten ser. Som eksempel på bruk av view kunne den her blitt skalert ned til å dekke begge SOI-sidene i figuren, skalert opp til å fokusere bare på én enkelt modul, eller flyttet nedover til å bare vise side 2. At alle sidene av SOI tegnes på én sammenhengende scene betyr her at selv om "Modul 3" er plassert først på sin side vil den i realiteten flyttes nedover på y-aksen med én side-høyde pluss luften mellom sidene. Dette er en implementasjonsdetalj som bruker ikke vil merke noe til.

Følgende seksjoner beskriver hvordan redigering av ulike aspekter ved SOI er implementert.

6.4.1 Redigering av SOI-moduler

Alle SOI-moduler arver fra QWidget, og kan dermed brukes direkte i Qt sidestilt med innebygde widgets. En SOI-modul har full kontroll over hvordan den selv blir redigert, og kan for eksempel implementere dette gjennom tekstfelter og tabeller,

som bruker er kjent med fra andre programmer. Siden modul-widgets plasseres i scene via `QGraphicsProxyWidget`, vil bruker kunne skalere og inspisere dem, i tillegg til å kunne redigere widget som vanlig.

Noen SOI-moduler implementerer redigering via ekstra menyer som dukker opp når bruker holder musepeker over widget, via dialoger eller via hurtigtaster. Detaljer for hver modul beskrives i seksjon 6.5.

6.4.2 Redigering av toppsegment

Toppsegment fra figur 6.1 består av flere sidestilte tekster og én logo, som vist i figur 4.12. Disse bruker standard widgets fra Qt, og er til forskjell fra SOI-moduler ikke direkte redigerbare. Bruker forventes å redigere innhold i toppsegment via oppsett-dialogen beskrevet i seksjon 4.1.4.

6.5 SOI-moduler

I denne seksjonen detaljeres implementasjon av hver SOI-modul, samt superklassen "ModuleBase".

6.5.1 Modul-base

Alle SOI-moduler er spesialiseringer av klassen "ModuleBase" som vises i kodeliste 6.1. Klassen brukes som et interface for å sørge for at spesialiseringer av denne implementerer dens funksjoner. ModuleBase er en spesialisering av ABC³ (Abstract Base Classes) slik at modulene kan være spesialiseringer av ulike klasser samtidig. Eksempelvis kan en tabell-modul arve fra `QTableWidget` og ModuleBase samtidig. Interface-løsningen diskuteres videre i seksjon 8.2 og 8.11.12.

I tillegg til fellestrekkene beskrevet i seksjon 3.1, har alle moduler en *type*-variabel og to fonter. *type*-variabelen brukes ved serialisering og deserialisering for å bestemme hvilken klasse modulen tilhører. Som nevnt i seksjon 4.2.4 skal modulene bruke en spesifikk font-familie, dette løses i linje 8 og 9 i kodeliste 6.1.

Kodeliste 6.1: Utdrag fra ModuleBase-klassen

```

1 class ModuleBase(ABC):
2
3     type = None
4
5     def __init__(self):
6         if self.type is None:
7             raise NotImplementedError
8         self.headline_font = HEADLINE_FONT
9         self.setFont(DEFAULT_FONT)
10
11     def get_size(self):
12         raise NotImplementedError

```

³ABC er en Python-modul som muliggjør definisjon av abstrakte klasser <https://docs.python.org/3/library/abc.html>, aksessert 16.05.2020.

```

13
14     def get_data(self):
15         raise NotImplementedError
16
17     def prepare_for_pdf_export(self):
18         raise NotImplementedError
19
20     @staticmethod
21     def get_user_friendly_name():
22         raise NotImplementedError
23
24     @staticmethod
25     def get_icon():
26         raise NotImplementedError

```

Listen under beskriver hva hver ModuleBase-funksjon brukes til. Dersom en modul mangler én eller flere funksjoner, vil funksjonen(e) i ModuleBase brukes, og kaste `NotImplementedError`.

- **init:** Sjekker at en spesialisering har definert *type*-variabelen og setter font.
- **get_size:** Returnerer modulens størrelse, brukes for modulplassering.
- **get_data:** Returnerer modulens innhold i et passende objekt, brukes ved serialisering.
- **prepare_for_pdf_export:** Brukes til å fjerne markert tekst og musemarkør fra tekstfelt, samt til annullering av aktive og tidligere valgte tabellceller før SOI-en eksporteres.
- **get_user_friendly_name:** Returnerer brukervennlig modul-navn som vises ved valg av moduler.
- **get_icon:** Returnerer modul-ikon som vises ved valg av moduler.

For å unngå kodeduplisering, inneholder *module_base.py*-filen noen frittstående funksjoner som brukes av flere moduler. Funksjonene utfører resizing og størrelsesberegning av tabeller, sjekk av hurtigtaster og klargjøring av spesifikke widgets for PDF-eksport.

En egenskap som er felles for alle moduler, men som gjøres på modulnivå, er at de ikke bruker mer plass enn de har behov for. Dette betyr eksempelvis at tabellkolonner er like brede som sitt bredeste innhold.

6.5.2 Autentiseringstavle og subtraktorkoder

Autentiseringstavle og subtraktorkoder har flere fellestrekk: de er tabeller med rad-identifikatorer A til Z i første kolonne og en kode i siste kolonne. I vår implementasjon av de to modulene, vist i figur 4.15, er det ytterligere fellestrekk: De har en redigerbar overskrift, innholdet i alle celler genereres automatisk og kan ikke redigeres, og bruker kan legge til og fjerne rader med hurtigtaster. For å unngå kodeduplisering, arver disse av en felles "CodeTableBase"-klasse.

Det som skiller modulene fra hverandre er at autentiseringstavle bruker bokstavkoder mens subtraktorkoder bruker tallkoder. Autentiseringstavlen har også to autogeneratede tall *x* og *y* i overskriften, der *x* er mindre enn *y*, som angir test-element, se 2.1.2, i tillegg til en ekstra kolonne med radidentifikatorer 0 til 9.

Autentiseringskodene skrives i en monospaced font⁴ for å øke lesbarheten, dette er ikke nødvendig for subtraktorkoder siden alle siffer er like brede med valgt font.

Når bruker skal legge til en av disse modulene, vises en dialog med modulinnstillinger vist i figur 4.14. Modulene har individuelle standard-verdier som er ferdig utfylt i dialogens felter. De resterende inputfeltene har det gruppen antar er meningsfulle verdispenn. Modulene bør inneholde minimum to koder av sikkerhetsmessige årsaker. Maksimalt antall koder bestemmes av antall radidentifikatorer, A til Z muliggjør 26. Kodene må bestå av minimum to tegn fordi det kreves ved autentisering, og fordi subtraktorkoder brukes på kartreferanser bestående av to tall. Bruker kan selv velge om det skal legges til mellomrom i koden, antall mellomrom, og mellomromsintervall.

6.5.3 Frekvenstabell og telefonliste

Frekvenstabell og telefonliste, henholdsvis figur 4.16 og 4.17, er implementert med lik tankegang. Modulene har en fast overskrift, en tabell, en seksjon med knapper for redigering av tabellen og en dialog for valg av kolonner i tabellen.

Tabellen inneholder et satt antall kolonner med gitte overskrifter. Hvilke overskrifter tabellen inneholder er avhengig av en Python-dictionary som inneholder par av strenger, eller ”nøkler”, og sannhetsverdier. Strengene er overskriftene til kolonnene og sannhetsverdiene indikerer om kolonnen skal vises eller ikke. Ved initialisering av tabellen blir det laget en kolonne for hvert par uavhengig av sannhetsverdien, deretter blir de gjort usynlige eller synlige. Etter initialisering kan dictionaryen endres via et modulært dialogvindu. Dialogvinduet er dynamisk generert etter dictionaryen, og har en avkrysningsboks med tilhørende tekst som representerer hvert par i dictionaryen.

Frekvenstabell inneholder to nivåer med overskrifter, der første rad i tabellen inneholder overskrifter som spenner over flere kolonner. Teksten for overskriften er lagt i første kolonne for spennet og deretter er cellen slått sammen med antall celler lik antall underoverskrifter for den gjeldende overskriften. Dette vil danne en virtuell kolonne på første rad som i realiteten vil bestå av en rekke logiske kolonner der overskriftsteksten ligger i den første av dem. I dictionaryen løses dette med et ytterlige nivå nøkler.

Modulene bruker en konfigurasjonsfil av typen JSON som leses inn ved initialisering av modulen. Formatet på filen er slik at det kan og blir direkte lagret i dictionaryen som styrer strukturen på tabellen og dialogvinduet. Endringer i konfigurasjonsfilen vil altså endre grensesnittet for modulen. Eksempel på en slik konfigurasjonsfil finnes i kodeliste 6.2.

Modulen anvender flere ”events”. ”KeyPress”-events brukes til tastaturnarveier arvet fra modulbasen 6.5.1. Seksjonen med knapper for å redigere modulen er pakket i en egen widget og denne vises kun når musepekeren er over modulens område. Dette gjøres med ”enterEvent” og ”leaveEvent”.

⁴I en monospaced font er hvert tegn like bredt.

Kodeliste 6.2: Eksempel på konfigurasjonsfil for frekvenstabell som gir strukturen vist i tabell 1.1

```

1 {
2   "": { "Nett": true, "ADR": false },
3   "MRR": { "Frekvenser": true, "Nett-ID": true, "NIS": true},
4   "HF": { "Frekvenser": true, "Nett-ID": false, "Kodeord": true }
5   "LFR": { "Kodeord": false, "Kanal nr.": false }
6 }

```

6.5.4 Forhåndsavtalte koder

Forhåndsavtalte koder, vist i figur 6.2, er en modul med overskrift, varslingsord og en layout med tabeller. Tabell-layoutet består av en "QHBoxLayout" med et visst antall "QVBoxLayouts". Hver QVBoxLayout inneholder et visst antall tabeller, og omtales videre som en kolonne.

Modulen følger strukturen gitt i alternativ 2 i seksjon 2.1.2: Det finnes én tabell per kategori fra liten kodebok 2.2. Tabellene har to kolonner og en overskrift bestående av én bokstav etterfulgt av kategorien. Første kolonne inneholder bokstavene A til Z, og danner en unik kode sammen med bokstaven i overskriften. Andre kolonne inneholder et ord/uttrykk som tilhører kategorien.

Modulen initialiseres gjennom modifierings-dialogen vist i figur 4.19b. Denne er ferdig utfylt med standard overskrift og et tilfeldig valgt varslingsord fra databasetabellen "CategoryWords" beskrevet i seksjon 3.3. Brukeren kan påvirke modulens størrelse gjennom "Maksimal kolonnehøyde"-feltet, men det er satt en nedre grense lik høyden til den høyeste tabellen for å sikre at alle tabeller får plass. Brukeren kan også bestemme kategori-rekkefølgen gjennom dra-og-slipp-funksjonalitet på elementene i listen.

Når dialogen er akseptert, lages modulens tabell-layout med følgende fremgangsmåte: Lag en ny kolonne og fyll den med så mange tabeller som mulig uten å overstige maksimal kolonnehøyde. Legg til kolonnen i layoutet og gjenta prosessen med de resterende tabellene helt til alle er lagt til. For å gi et strukturert utseende, settes alle tabeller til å være like brede som den bredeste tabellen i sin kolonne. Denne algoritmen vises i kodeliste 6.3.

Ved bruk av hurtigtast åpnes dialogen igjen slik at modulen kan modifieres også etter at den har blitt laget. Da vil alle koder oppdateres ved at ord i tabellene tilfeldig bytter rekkefølge, og ord som eventuelt har blitt lagt til i liten kodebok i mellomtiden vil føyes til i aktuell tabell. Hensikten med denne oppdateringen er at modulen vil kunne gjenbrukes på flere SOI-er siden dens kryptografi er oppdatert.

Figur 6.2 gir et inntrykk av hvordan modulen reagerer på endringer gjort gjennom dialogen. Figur 6.2a viser modulen med en brukerbestemt maksimal kolonnehøyde på 726 piksler. Figur 6.2b viser samme modul med maksimal kolonnehøyde endret til 526 piksler, endret kategorirekkefølge og oppdatert kryptografi.

FORHÅNDSAVTALTE KODER	
Varslingsord: LABRADOR	
A Landemerker	D Uttrykk/tiltak
A Bro	A Bekreft, -er
B Elv	B Trenger forsyninger
C Sti	C Går av nett
B Støtte	D Skyting hørt
A K9	E Oppgi posisjon
B MP	F Sett i marsj
C Sanitet	G Anmoder om
D Helikopter	H Gjennomøk, -e, -er
C Tall	E Retninger
A 6	A Vest
B 2	B Fram
C 8	C Sør
D 5	D Nord
E 4	E Øst
F 3	F Tilbake
G 9	F Våpenteknisk
H 1	A HK-416
I 7	B Håndgranat
J 0	C MP-5
	D Bombe

FORHÅNDSAVTALTE KODER		
Varslingsord: ARGENTINA		
A Tall	C Støtte	E Våpenteknisk
A 7	A K9	A HK-416
B 1	B Helikopter	B Håndgranat
C 4	C Sanitet	C MP-5
D 9	D MP	D Bombe
E 0	D Uttrykk/tiltak	F Retninger
F 3	A Oppgi posisjon	A Øst
G 2	B Anmoder om	B Tilbake
H 6	C Bekreft, -er	C Fram
I 5	D Sett i marsj	D Sør
J 8	E Gjennomøk, -e, -er	E Vest
B Landemerker	F Trenger forsyninger	F Nord
A Sti	G Går av nett	
B Bro	H Skyting hørt	
C Elv		

(a) Maksimal kolonnehøyde: 726px

(b) Maksimal kolonnehøyde: 526px

Figur 6.2: Forhåndsavtalte koder: Høy og bred utgave

Kodeliste 6.3: Hoveddelen av funksjonen `create_table_layout`

```

table_layout = QHBoxLayout()
# Algorithm-explanation: Create a "column" and fill it with tables until adding the
# next table will overshoot maximum_column_height, then add a new column and repeat
# the process until all tables are added.
i = 0
column_height = 0
# While all tables are not added
while i < len(self.tables):
    # Create a "column"
    vbox_layout = QVBoxLayout()
    max_table_width = 0
    # While there are more tables and there is room for the next table,
    # add table to column and save it's width if it is the widest table
    while (
        i < len(self.tables)
        and column_height + self.tables[i].height() <= self.maximum_column_height
    ):
        table = self.tables[i]
        if table.width() > max_table_width:
            max_table_width = table.width()
        vbox_layout.addWidget(table)
        # Increase column-height
        column_height += table.height()
        i += 1

    # Column does not have room to fit the next table.
    # Make all tables in column have equal width.
    for j in range(vbox_layout.count()):
        widget = vbox_layout.itemAt(j).widget()
        column_one_width = widget.columnWidth(0)

```

```

        widget.setFixedWidth(max_table_width)
        widget.setColumnWidth(1, max_table_width - column_one_width)

# Add a QSpacerItem so that tables in column are pushed to the top
vbox_layout.addStretch(1)

# Add column to the table_layout and reset column-height.
table_layout.addLayout(vbox_layout)
column_height = 0

```

6.5.5 Kodefraser

Kodefraser, figur 4.18, inneholder en overskrift og en tabell med rader bestående av kodeord og tilhørende fraser. Når modulen lages blir en kategori fra databasetabellen "CategoryWords", seksjon 3.3, valgt. Alle kodeord i modulen vil komme fra oppføringer i databasetabellen med den valgte kategorien. Kategorien velges slik at den ikke deles med en annen instans av modulen. Dette gjøres via en klassevariabel som holdes oppdatert med alle opptatte kategorier. Om det er tomt for kategorier når en modul lages, vil modulen gi en exception og avslutte umiddelbart, da håndtering for dette ikke er implementert. For hver rad innad i en modul velges kodeord slik at de er unike i modulen. Om det er tomt for kodeord når ny rad blir forsøkt lagt til vil ikke raden bli lagt til.

Overskriften i modulen bruker "size hints" på lik linje med fritekstmodulen, 6.5.6. Modulens tabell, i motsetning til andre tabeller i programvaren, bruker "QHeaderView.ResizeToContents"⁵ for dynamisk størrelse. Dette gjør at tabellen passer inn i programvaren uten at bearbeidingen som gjøres på de andre tabellene er nødvendig.

6.5.6 Fritekst

Fritekst, figur 4.13b, inneholder tekst med tilhørende overskrift. Modulen kan brukes til prosedyrer, eller SOP 1.10.6, og til å holde vilkårlig informasjon. Modulen er implementert som to tekstfelt. Det ene tekstfeltet brukes til overskrift og vises med fet skrift. Det andre tekstfeltet brukes til hovedinnhold av tekst, og støtter "tvungne" brukerplasserte nye linjer, altså vil nye linjer i tekst bare forekomme om bruker selv velger å trykke på "Enter"-tasten. Det er bemerkelsesverdig at modulen er implementert til å utnytte "size hints" i Qt, et konsept som gir widgets en måte å rapportere hvor store de er. Fritekstmodulen bruker size hints til å fortelle programvaren at den er akkurat stor nok til å vise fram alt sitt innhold.

6.5.7 Generisk tabell

En generisk tabell ble implementert fordi den vil støtte mange bruksområder. Tabellen, vist i figur 4.13a, initialiseres som en 2 x 2 tabell der cellene i øverste rad

⁵<https://doc.qt.io/qt-5/qheaderview.html#ResizeMode-enum>, aksessert 11.05.2020.

brukes som kolonne-overskrifter. Gjennom bruk av hurtigtaster kan brukeren legge til og fjerne både rader og kolonner for å tilpasse tabellen til sitt bruk. Modulen implementeres ved å spesialisere den innebygde widgeten "QTableWidget".

6.6 Automatisk modulplassering i SOI

Da kjernen i oppgaven er å demonstrere hvordan programvare kan gjøre produksjon av SOI-er enklere, ble automatisk plassering av moduler tidlig et mål for prototypen. SOI-moduler kan ses på som en rekke rektangler i ulike størrelser som skal plasseres optimalt over flere SOI-sider. SOI-moduler må plasseres slik at de ikke kolliderer med toppsegmentet, og slik at de ikke overstiger de ytre kantene av brukbar plass i SOI. Brukbart område i en SOI-side kan også ses på som et rektangel. Problemet reduseres dermed til at en rekke rektangler (moduler) i ulike størrelser skal plasseres optimalt inne i en annen rekke rektangler (sider). Dette problemet er kjent som "Two-dimensional rectangle bin packing" [6] og flere algoritmer eksisterer for å løse det. Merk at for SOI-tool er begrepet "bin", engelsk for "boks" eller "beholder", det samme som brukbart område i SOI. For å øke lesbarheten til påfølgende seksjoner blir "side" ofte brukt istedenfor "bin".

Python-biblioteket "rectpack" [7] tilbyr en samling algoritmer for løsning av "Two-dimensional rectangle bin packing"-problemet. SOI er strukturert i programvaren til å hypotetisk kunne støtte flere modulplasseringsstrategier, men automatisk plassering med rectpack er det eneste som er implementert.

6.6.1 Introduksjon til rectpack

Plassering, eller pakking, med rectpack gjøres basert på tre innstillinger: plasseringsalgoritme, sorteringsalgoritme og sidevalgsalgoritme. Disse beskrives i denne seksjonen. Rectpack er basert på forskningsartikkelen "A Thousand Ways to Pack the Bin - A Practical Approach to Two-Dimensional Rectangle Bin Packing" [6], hvor detaljer om hver enkelt av innstillingene utdypes.

Plasseringsalgoritme: Angir hvilken algoritme som skal brukes til å plassere rektangler innad i en gitt side. Eksempler på algoritmer for dette er "Guillotine"-algoritmen og "MaxRects"-algoritmen, som begge er beskrevet i forskningsartikkelen nevnt over.

Sorteringsalgoritme: Angir hvilken rekkefølge rektangler skal plasseres i. Rektangler kan for eksempel sorteres på areal, bredde eller høyde. Innstillingen kan settes til "ingenting", som vil få rectpack til å respektere rekkefølgen til gitte rektangler direkte.

Sidevalgsalgoritme: Angir hvordan rectpack skal velge side å plassere rektangel i. Eksempler er "Bin First Fit" som velger første side hvor rektangel passer, og "Bin Best Fit" som legger rektangel der det "passer best" i rectpacks øyne.

6.6.2 Anvendelse av rectpack

Konseptuelt skjer modulplassering i programvaren ved følgende steg, hvor X er ønsket mengde luft mellom moduler. Figur 6.3 visualiserer stegene. Toppsegment og ubrukbart område er tegnet inn i blått, brukbart område er tegnet inn i oransje og moduler i hvitt. Stiplede linjer markerer "luft" rundt moduler, som beskrevet i stegene under.

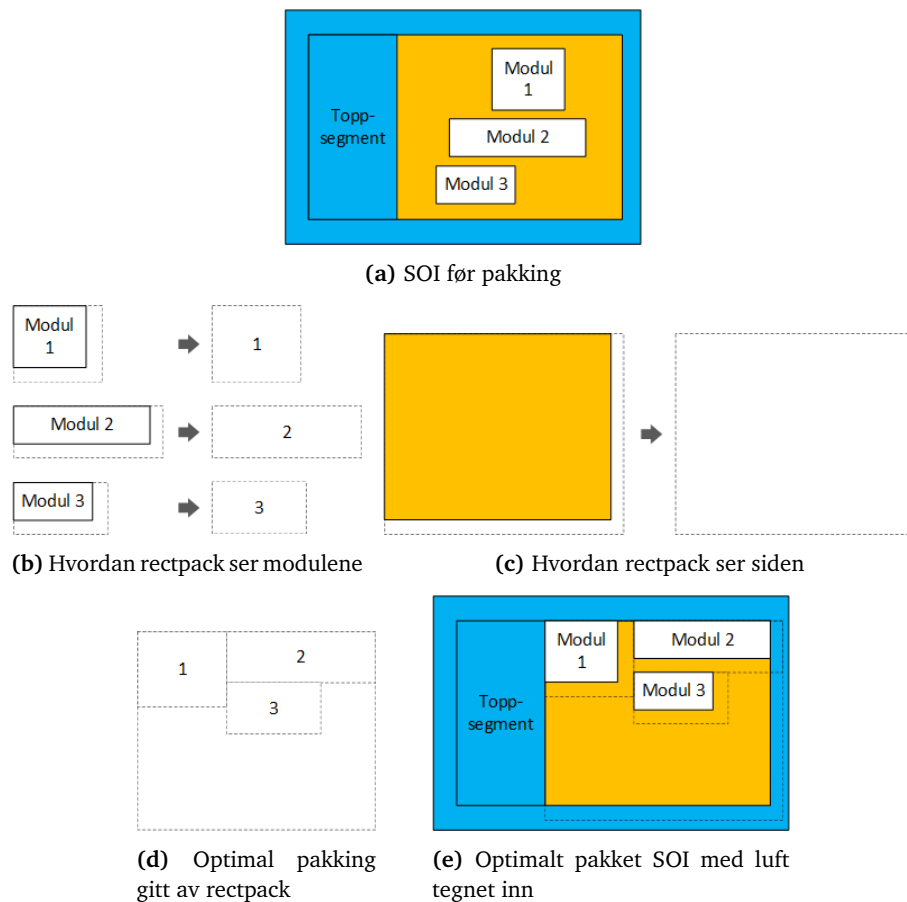
1. Rectpack forberedes, og de tre innstillingene beskrevet i forrige seksjon settes til brukervalgte verdier.
2. For hver SOI-modul lagres et rektangel med modulens bredde og høyde. Denne størrelsen økes med X i hver dimensjon, slik at modulene ser større ut for rectpack. Dette vil resultere i luft mellom modulene. Rektanglene lagres med en "id" slik at de kan assosieres med riktige moduler etter pakking. Steget visualiseres i figur 6.3b.
3. Rectpack blir fortalt dimensjoner på "bins", altså størrelse på brukbart område i hver SOI-side. Størrelser blir oppgitt til å være X større i hver dimensjon for å kompensere for ekstra størrelse gitt til rektangler. Steget visualiseres i figur 6.3c.
4. Rectpack kjører og oppgir optimal plassering for rektangler, som visualisert i figur 6.3d.
5. SOI-moduler oppdateres med posisjoner foreslått av rectpack. For å knytte rektangler til riktige moduler brukes "id" gitt tidligere. Steget visualiseres i figur 6.3e.

6.6.3 Brukerinnflytelse på plassering

Bruker har innflytelse på automatisk modulplassering gjennom de tre innstillingene presentert i seksjon 6.6.1. Programvaren er i utgangspunktet innstilt etter rectpacks anbefalinger, men om bruker ønsker kan dette overstyres for å oppnå andre plasseringer. De alternative plasseringene er potensielt dårligere enn plassering gjort med utgangsinstillinger, men kan likevel være mer ønskelige for bruker.

Figur 6.4 demonstrerer hvordan endring av rectpacks innstillinger gir bruker innflytelse over plassering. Figuren er hentet fra den utviklede løsningen, og viser ulike plasseringer av tre "generisk tabell"-moduler 6.5.7, med overskrift "Stort areal" med størst areal, "Stor høyde" med størst høyde og "Stor bredde" med størst bredde. Med utgangsinstillinger for plassering vil moduler bli plassert som i figur 6.4a. Om bruker endrer sorteringsalgoritme til "bredde" vil modulene plasseres som i figur 6.4b. Om bruker velger plasseringsalgoritme "Guillotine" vil modulene plasseres som i figur 6.4c.

Rectpack kommer med en rekke valg for hver av plasseringsinnstillingene, men programvaren utnytter bare et utvalg av disse. Dette er fordi bare et utvalg er nødvendig for å demonstrere at brukerinnflytelse er mulig. Implementasjonen er løst koblet til valgmulighetene, så flere valg kan gjøres tilgjengelig for bruker med litt programmeringsarbeid.



Figur 6.3: Anvendning av rectpack. Fra kaotisk plassering av moduler til optimalisert plassering av moduler

Bruker har også innflytelse på automatisk plassering gjennom manuell prioritering av moduler. Om bruker velger "Egendefinert rekkefølge" som sorteringsalgoritme vil modul-listen til venstre i brukergrensesnittet tillate omprioritering av moduler. Som eksempel kan en bruker som ønsker at en spesiell modul alltid skal bli plassert først legge denne øverst i modul-listen.

Stort areal			Stor høyde	Stor bredde				

(a) Utgangsinstillinger

Stor høyde	Stort areal		Stor bredde					

Stort areal			Stor bredde					

Stor høyde								

(b) Sorteringsalgoritme: bredde

(c) Plasseringsalgoritme: Guillotine

Figur 6.4: Ulike plasseringer av tre moduler basert på plasseringsinnstillinger

6.7 SOI på fil

Detaljer rundt implementasjonen som følger 3.2 trekkes frem i følgende seksjoner. Programvaren støtter i praksis lagring og innlasting av både komprimert og ukomprimert fil, noe som gir mulighet for innsyn i serialiserte SOI-er.

6.7.1 Serialisering og deserialisering

Serialisering følger formatet beskrevet i seksjon 3.2.1 og resulterer i en streng. Strengen kan enten skrives direkte til fil, eller komprimeres som beskrevet i påfølgende seksjon. Konseptuelt skjer serialisering av SOI i følgende steg:

1. Forbered nytt JSON-objekt.
2. Hent alle attributter til SOI unntatt moduler og vedlegg og lagre disse i tilsvarende attributter i JSON-objekt.
3. For hver SOI-modul i SOI:
 - a. Hent serialisert streng av SOI-modul via grensesnitt beskrevet i seksjon 6.5.1.
 - b. Hent SOI-modulens klassenavn samt x og y posisjon og sidetall.
 - c. Bruk hentet informasjon til å lage ny oppføring i JSON-objektet under enten **modules** eller **attachments**, avhengig av om SOI-modulen brukes som vedlegg eller ikke.

Deserialiseringsprosessen tilsvarer serialiseringsprosessen, men i revers. Igjen utnyttes felles modul-grensesnitt som beskrevet i seksjon 6.5.1. Før deserialisering sjekkes innlest JSON-objekt opp mot schema i kodeliste 3.1 for å sikre at objektet ikke er misformet.

6.7.2 Komprimering og dekomprimering

Koden har funksjoner for komprimering og dekomprimering. Begge baserer seg på LZMA, og mottar en streng som parameter. Strenger i Python3 lagres som Unicode⁶ og LZMA komprimerer bytes til bytes. Funksjonens parameterstreng må derfor kodes til bytes før den kan komprimeres. Nevnte bytes konverteres så tilbake til streng før de skrives til fil for å møte US ASCII-kravet beskrevet i kapittel 2.4.2. I dette steget vil Python escape⁷ backslash som alt er i de komprimerte dataene for at strengen skal inneholde faktisk backslash, dette resulterer i dobbel backslash i datane. Når data skal dekomprimeres tilfører dette noen steg til prosessen. Først må *b'*-notasjon, som er resultat av konvertering fra bytes til streng, fjernes. Deretter må strengen kodes til bytes igjen. Til slutt må dobbel backslash forenkles før datane kan dekomprimeres til bytes som igjen må dekodes tilbake til den strengen man hadde før komprimering. Denne prosessen demonstreres i kodeliste 6.4.

Kodeliste 6.4: Formatering ved dekomprimering

```
# Remove extra b' notation
lzma_compressed_json_string = lzma_compressed_json_string[2:-1].encode(
    encoding="ascii"
)
# Remove double \
lzma_compressed_json_string = codecs.escape_decode(
    lzma_compressed_json_string, "hex"
)
# Decompress
lzma_compressed_json_string = lzma.decompress(
    lzma_compressed_json_string[0]
)
# Cast to ascii string
lzma_compressed_json_string = lzma_compressed_json_string.decode(
    encoding="ascii"
)
```

6.8 Eksport av SOI til PDF

For å eksportere SOI til PDF utnyttes Qts tegnesystem. Qt tegner ved bruk av klassene "QPainter" og "QPaintDevice". QPainter utfører tegneoperasjoner, og QPaintDevice er en abstraksjon for to-dimensjonelle tegneflater som kan bli tegnet på ved bruk av QPainter.⁸ Klassen QWidget er selv en QPaintDevice, og det er slik widgets tegner seg selv til skjerm. Det er mulig å tegne widgets på andre tegneflater ved å bruke instanser av QPaintDevice. "QPrinter" er en innebygd QPaintDevice som representerer en printer og støtter PDF-format. Dette kan brukes til å skrive ut widgets til PDF.

⁶<https://www.programiz.com/python-programming/methods/string/encode>, aksessert 16.05.2020.

⁷Å "escape" betyr å legge backslash foran et symbol for å indikere at symbolet skal spesialbehandles. Å escape en backslash, \, vil spesialbehandle backslash selv, og resulterer i én enkelt backslash.

⁸Forklaring hentet fra <https://doc.qt.io/qt-5/paintsystem.html>, aksessert 11.05.2020.

Eksport av SOI til PDF skjer direkte fra redigeringsvinduet beskrevet i seksjon 6.4 da SOI representeres der som en widget - `QGraphicsScene`. Før eksport fjernes alle detaljer som ikke inkluderes på PDF, som ”markert celle” i tabeller og ”markert tekst” i tekstfelt. Deretter blir hvert område i `QGraphicsScene` som inneholder en SOI-side skrevet til en PDF-side, som resulterer i en PDF bestående av hele SOI-en, én SOI-side per PDF-side.

En eksportert SOI kan bestå av flere eksemplarer, hvor hvert eksemplar merkes med eksemplarnummer og totalt antall eksemplarer. Dette implementeres ved å gjøre prosessen i avsnittet helt til ønsket antall eksemplarer oppnås. En viktig detalj ved dette er at eksemplarnummerering må vises ulikt under redigering og på eksportert SOI. Under redigering vet ikke programvaren hvor mange eksemplarer som ønskes, og det er bare nødvendig å vise ett eksemplar, mens på eksportert SOI skal reell eksemplarnummerering vises. Med andre ord er det ønskelig at redigeringsvindu for SOI skal kunne vise seg ulikt til skjerm og på PDF. Dette implementeres ved å lage en underklasse av `QGraphicsProxyWidget` beskrevet i seksjon 6.4 som tegner seg ulikt basert på hvilken `QPaintDevice` den tegnes til. Underklassen har navnet `ProxyLabelWithCustomQPrintText` og brukes til å tegne teksten ”1 av N” under redigering og ”X av Y” ved eksport, hvor X og Y er reelle verdier for henholdsvis eksemplarnummer og totalt antall eksemplarer.

Eksportert PDF har brukervalgt oppløsning på mellom 50 dpi og 1200 dpi, hvor 500 dpi er standard. Lav dpi vil produsere PDF av lav kvalitet og lav filstørrelse. Høy dpi vil produsere PDF av høy kvalitet og høy filstørrelse. Standardoppløsning på 500 dpi gir fullt leselig PDF med ”akseptabel” filstørrelse. Et eksempel på en SOI eksportert til PDF finnes i vedlegg K.1.

6.9 Dialoger

Applikasjonen har to typer dialoger som brukes ofte. Den ene typen er en `QDialog`-klasse kalt `AcceptRejectDialog` med knappene ”Ok” og ”Avbryt”. Kodeduplisering unngås ved å spesialisere denne generelle klassen hver gang det er behov for en dialog med disse knappene. Dialogen for oppsett av en SOI vist i figur 4.10 og dialogen for å legge til en modul vist i figur 4.9, er eksempler på slike spesialiseringer.

Den andre typen dialog er `QMessageBox` som brukes til å gi bruker tilbakemeldinger med et ikon som indikerer meldingens alvorlighetsgrad; ”info”, ”advarsel” eller ”kritisk feil”. Det ble implementert tre funksjoner `exec_info_dialog`, `exec_warning_dialog` og `exec_critical_dialog`, som tar en kort tekst og en beskrivelse som parameter. Disse åpner en `QMessageBox` som viser et dialog-vindu med teksten og beskrivelsen som blir sendt med, i tillegg til riktig alvorlighets-ikon.

6.10 Kodegenerering

Flere steder i programvaren er det behov for ulike strenger bestående av tilfeldige tegn til kryptografisk bruk, disse omtales videre som ”koder”. Funksjonen i kodeliste 6.5 genererer en tilfeldig kode med en gitt lengde, bestående av en gitt type tegn: store bokstaver, siffer eller en kombinasjon av disse. For å øke leseligheten er det implementert funksjonalitet som legger til et gitt antall mellomrom bak hvert N-te tegn i koden.

Kodeliste 6.5: Funksjon som genererer en ”kode”

```
def get_code(code_length, mode="ascii", space_interval=0, space_amount=1):

    if mode == "ascii":
        characters = string.ascii_uppercase
    elif mode == "digits":
        characters = string.digits
    elif mode == "combo":
        characters = string.ascii_uppercase + string.digits
    else:
        raise ValueError(
            "Invalid value for argument 'mode': " '{}{}'.format(mode)
        )

    code = ""
    i = 0
    while i < code_length:
        letter = secrets.choice(characters)
        code += letter
        i += 1

    if space_interval > 0:
        code = insert_spaces(code, space_interval, space_amount)

    return code
```

Enkelte SOI-moduler trenger et gitt antall unike koder. Dette løses ved å legge til resultatet fra kodeliste 6.5 i et ”Python set”⁹ til det inneholder ønsket antall koder, se kodeliste 6.6.

Kodeliste 6.6: Funksjon som genererer flere koder

```
def get_code_set(count, code_length, mode="ascii", space_interval=0, space_amount=1):
    codes = set()
    while len(codes) < count:
        codes.add(get_code(code_length, mode))
    return codes
```

6.11 Kodebok

Kodeboken representeres av database-tabellen ”Codebook” beskrevet i seksjon 3.3. Beskrivelse av brukergrensesnitt og PDF-eksport av denne presenteres i følgende seksjoner.

⁹Python set er en usortert samling av unike elementer.

6.11.1 Grensesnitt mot kodebok

Grensesnittet mot kodeboken, vist i figur 4.5, består av en modifisert QWidget-klasse kalt "CodebookWidget". Som kodeliste 6.7 viser, initialiserer klassen en "CodebookTableView" og en "CodebookRowAdder", og legger dem i en QVBoxLayout.

Kodeliste 6.7: Utdrag fra *CodebookWidget*

```
class CodebookWidget(QWidget):
    """Widget for viewing and editing codebook."""
    def __init__(self, database):
        super().__init__()
        self.view = CodeBookTableView(database)
        self.row_adder = CodebookRowAdder(database, self.view)
        self.create_and_set_layouts()
```

CodeBookTableView og CodeBookTableModel

Kodebokvisningen gjøres med et "model/view"-samarbeid, der view er en modifisert QTableView kalt "CodeBookTableView", og model er en modifisert "QSqlTableModel" kalt "CodeBookTableModel". CodeBookTableView sin oppgave er å presentere data som ligger i sin CodeBookTableModel. I tillegg gir den navn til kolonnene, muliggjør sortering ved å klikke på dem og ber sin model om å slette den valgte raden når brukeren trykker delete.

CodeBookTableModel, vist i kodeliste 6.8, leser inn data fra database-tabellen "Codebook" og skriver endringer brukeren gjør til databasen. Funksjonen *flags* i linje 9 er overstyrt og gjør alle kolonner redigerbare med unntak av "Kode"-kolonnen. Funksjonen *setData* i linje 21 er også overstyrt, og sørger for gyldig verdi i *type*-kolonnen. Dersom brukeren skriver ugyldig verdi, vil tabellcellen tømmes, og verdien vil ikke bli skrevet til databasen.

Kodeliste 6.8: Utdrag fra *CodebookTableModel*

```
1 class CodeBookTableModel(QSqlTableModel):
2     def __init__(self):
3         super().__init__(None, QSqlDatabase.database(CONNAME))
4         self.setEditStrategy(QSqlTableModel.OnFieldChange)
5         self.setTable("CodeBook")
6         self.setSort(0, Qt.AscendingOrder)
7         self.select()
8
9     def flags(self, index):
10         flags = super().flags(index)
11
12         # Disable items in column 'Code'
13         if index.column() == 3:
14             flags ^= Qt.ItemIsEnabled
15         # Enable editing on items in columns 'Word', 'Category' and 'Type'
16         else:
17             flags |= Qt.ItemIsEditable
18
19         return flags
```

```

20
21     def setData(self, index, value, role):
22         # If a change is made to column 'Type'
23         if index.column() == 2:
24             # If correct string, replace with correctly cased str
25             if value.lower() == "liten":
26                 value = "Liten"
27             elif value.lower() == "stor":
28                 value = "Stor"
29             else:
30                 return False
31
32         return super().setData(index, value, role)

```

CodebookRowAdder

CodebookRowAdder-klassen er en modifisert QWidget som inneholder to "QLineEdit"-felter, en "QComboBox" og en "QPushButton", med tilhørende "QLabels". Klassen legger til ord og uttrykk i database-tabellen "Codebook" og oppdaterer kodebok-visningen når en ny rad blir lagt til. Brukeren får tilbakemeldinger når ord/uttrykk ikke er fylt inn, allerede eksisterer eller har blitt lagt til i databasen.

Følgende avsnitt refererer til kodeliste 6.9. CodebookRowAdder kan brukes alene, men har også funksjonalitet for å brukes sammen med en CodebookTableView. Viewet sin model har en eksklusiv lås som hindrer andre å skrive til database-tabellen, og det er derfor nødvendig at CodebookRowAdder vet at den brukes i kombinasjon med et view, slik at den kan oppheve låsen før skriving, som vist i kodelinje 4. Etter at et ord/uttrykk har blitt skrevet til databasen, tegnes det en kode i linje 11. Deretter tilbakestilles og tømmes alle input-widgets og det gis tilbakemelding til brukeren. Til slutt, i linje 24, gis viewet en ny model som sørger for at oppdatert kodebok vises.

Kodeliste 6.9: Utdrag fra funksjonen *insert_row* i *CodebookRowAdder*

```

1  try:
2      # If a view is used, remove it's model temporarily
3      if self.codebook_view is not None:
4          self.codebook_view.setModel(None)
5
6      # Try to add row to database
7      stmt = "INSERT INTO CodeBook(Word, Category, Type) VALUES(?, ?, ?)"
8      self.database.conn.execute(stmt, (word_input, category_input, type_input,))
9
10     # Add unique code to row and commit changes
11     self.database.add_code_to(word_input)
12     self.database.conn.commit()
13
14     # Give feedback and reset input
15     self.label_feedback.setText("Ord/Utrykk lagt til.")
16     self.text_field_word.setText("")
17     self.text_field_category.setText("")
18     self.combo_type.setCurrentIndex(0)
19 except IntegrityError:
20     self.label_feedback.setText("Ord/uttrykk eksisterer allerede.")
21

```

```

22 # If a view is used, create and set a new, updated model
23 if self.codebook_view is not None:
24     self.codebook_view.setModel(CodeBookTableModel())

```

6.11.2 Eksport av kodebok

Eksport av kodeboken gjøres med et PDF-bibliotek kalt ReportLab [8]. Tabellene følger stilen til et *TableStyle*-objekt som vises i kodeliste 6.10. Linjene 4-8 er kommandoer for formatering av celler. Første element i hver kommando angir hvilket attributt kommandoen gjelder, andre og tredje element angir henholdsvis start- og slutt-celle (kolonne, rad) og siste element angir attributtets verdi.

Kodeliste 6.10: Tabellstil

```

1 HEADER_BG_COLOR = colors.HexColor("#a6a6a6")
2 TABLE_STYLE = TableStyle(
3     [
4         ("FONTSIZE", (0, 0), (-1, 0), 16),           # Header-fontsize
5         ("BOTTOMPADDING", (0, 0), (-1, 0), 10),      # Header-padding bottom
6         ("BACKGROUND", (0, 0), (-1, 0), HEADER_BG_COLOR), # Header background-color
7         ("ALIGN", (0, 0), (-1, 0), "CENTER"),         # Header-text centered
8         ("GRID", (0, 0), (-1, -1), 1, colors.black),  # Cell borders
9     ]
10 )

```

Kodeliste 6.11 viser fremgangsmåten som brukes for å lage PDF-dokumentet. Første steg er å lage tabeller for kode- og dekodebok i linje 2 og 3, *repeatRows* settes til "1" slik at kolonneoverskriftene gjentas på hver side som tabellen strekker seg over. Linjene 5 til 16 viser steg to, som er å lage en liste og legge til alle elementene som PDF-en skal inneholde. Til slutt, i linjene 19 til 22, opprettes dokumentet med en tilpasset "SimpleDocTemplate", og innholdet legges til med en tilpasset *Canvas*. Tilpasningene er nødvendig for at kode- og dekodebok skal ha individuelle sidetall "x av y" og en skilleside mellom seg, som beskrevet i seksjon 4.1.1.

Kodeliste 6.11: Utdrag fra funksjonen *generate_codebook_pdf* i *codebook_to_pdf.py*

```

1 # Create code- and decodebook-tables with predefined style
2 table_code = Table(data_code, style=TABLE_STYLE, repeatRows=1)
3 table_decode = Table(data_decode, style=TABLE_STYLE, repeatRows=1)
4
5 elements = []
6 # Add title, vertical spacer and table for codebook
7 elements.append(title_code)
8 elements.append(Spacer(0, 1 * cm))
9 elements.append(table_code)
10 # Double pagebreak to add separating page between code- and decodebook
11 elements.append(PageBreak())
12 elements.append(PageBreak())
13 # Add title, vertical spacer and table for decodebook
14 elements.append(title_decode)
15 elements.append(Spacer(0, 1 * cm))
16 elements.append(table_decode)

```

```
17 |  
18 | # Create document, add elements and save as PDF  
19 | doc = CodeAndDecodebookDocTemplate(  
20 |     file_name, page_size=orientation(page_size), topMargin=30  
21 | )  
22 | doc.build(elements, canvasmaker=CodeAndDecodebookCanvas)
```

6.12 Leveranse

Utviklet programvare har to brukergrupper: sluttbrukere og utviklere. Med sluttbruker menes brukere som vil ta i bruk verktøyet direkte. Med utviklere menes brukere som vil inspisere kildekoden og videreutvikle programvaren. Leveranse av programvare er laget slik at den tilfredstiller begge disse brukergruppene. Programvaren leveres som ZIP-fil med følgende komponenter:

Installasjonsinstrukser: Dokument med enkle instruksjoner på hvordan programvaren kan tas i bruk. Siden leveransen inkluderer eksekverbar fil klar for bruk sier instruksene i praksis bare hvor denne ligger, og hvordan Windows-snarvei kan lages til denne. Dokumentet inneholder også grunnleggende detaljer om programvaren som hvilke skjermskaleringer som støttes og hvor produserte filer vil ende opp, samt en beskrivelse av resten av komponentene i ZIP-filen. Dokumentet finnes som vedlegg D.5.

Eksekverbar fil: Denne filen inneholder alt programvaren trenger for å kjøre, og kan startes ved ”dobbeltklikk”. Filen er produsert av verktøyet ”pyinstaller”¹⁰, som sørger for at sluttbruker ikke trenger å installere programmeringsspråk eller biblioteker for å kjøre programvaren.

Dokumentasjon: En mappe med dokumentasjon fra kildekoden omformet til HTML-filer. Dokumentasjonen kan dermed åpnes i nettleser, noe som gir økt brukervennlighet sammenlignet med å søke seg gjennom programfiler.

Kildekode: Hele kildekoden inkluderes, sammen med instruksjoner til utviklere på hvordan denne skal installeres og kjøres. Instruksjoner på hvordan eksekverbar fil kan lages for sluttbrukere inngår også.

¹⁰pyinstaller er et verktøy for å omforme Python kode til selvstående eksekverbare filer.

Kapittel 7

Testing

I dette kapitlet blir testing av programvaren beskrevet. Automatisert testing ble gjort med unit testing og linting. Manuell testing ble utført av gruppemedlemmer under utvikling. I tillegg ble en brukertest gjennomført.

7.1 Automatisert testing

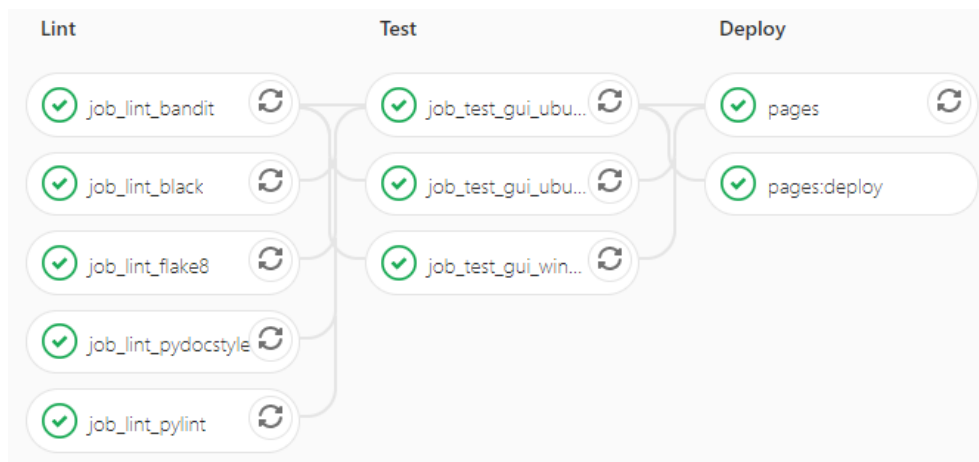
Fordi programvaren utvikles som prototype til et fremtidig ferdig produkt er det sannsynlig at den vil brukes som referanse. Kodekvalitet er derfor viktig. Programvaren testes kontinuerlig via "GitLab CI" på to nivå: statisk analyse, også kalt "linting", og "Unit testing". Figur 7.1 viser en kjøring av testene i GitLab CI. De fleste "testjobbene" kjøres via "Docker"¹, som gjør testene reproducerbare lokalt. Docker image som brukes er utviklet av gruppen, og inneholder alle nødvendige avhengigheter. Docker image er lagt til som vedlegg D.4.

Fordi "Docker-image" er basert på Ubuntu, Linux er tester i denne ikke nok til å bevise at programvaren fungerer på Windows. I tillegg er gruppens erfaring at Qt fungerer dårlig kjørt i "Docker-containere" på grunn av manglende skjerm. Unit testing gjøres derfor via GitLab CIs "shell executor"² koblet til arbeidsmaskiner kjørende i skolens OpenStack-installasjon, "SkyHigh". To arbeidsmaskiner er konfigurert, begge med 16 GB RAM og 8 VCPUs³. Én maskin kjører Ubuntu 18.04 og én kjører Windows 10. Konfigurasjon av maskinene er lagt til som vedlegg D.2 og D.3.

¹<https://docs.gitlab.com/runner/executors/docker.html>, aksessert 15.05.2020.

²<https://docs.gitlab.com/runner/executors/shell.html>, aksessert 16.05.2020.

³Virtuelle kjerner i OpenStack.



Figur 7.1: Pipeline i GitLab CI

7.1.1 Linting

Verktøyene "Flake8"⁴, "Pylint"⁵, "Black"⁶, "Pydocstyle"⁷ og "Bandit"⁸ brukes til statisk analyse av koden. Flake8 og Pylint tilbyr "linting" og overlapper på noen områder, men begge brukes for å sikre kodekvalitet til en så stor grad som mulig. Flake8 følger stilguiden "PEP8"⁹.

Black er en kodeformaterer som omformer Python kode til et standardisert format. Black kjøres i Python for å sjekke om koden følger standard format, og gir tilbakemelding om noe må endres. I praksis brukes dette bare til å fange opp situasjoner hvor utvikler glemmer å kjøre kode gjennom Black lokalt før det pushes opp. Black og PEP8 sørger til sammen for en grad av konsistens i formatering av kode. Pydocstyle sikrer at Python "docstrings" følger valgt konvensjon. Dette prosjektet bruker konvensjonen "numpydoc". Bandit fanger opp kjente sikkerhetsproblemer i Python-kode.

7.1.2 Unit testing

Det innebygde Python-biblioteket "unittest" brukes, sammen med Qt-klassen "QTest" [9] for GUI-testing. QTest inneholder funksjonalitet for å simulere at en bruker navigerer seg rundt i applikasjonen, og kan derfor brukes til "whitebox"¹⁰ GUI-testing. Museklikk, musebevegelse og tastetrykk er mulig. Ved å kjøre GUI-testene på maskiner med sluttbrukerens operativsystem som en del av pipeline sikres det at programvaren fungerer som forventet ved leveranse. GUI-testingen omfatter

⁴<https://flake8.pycqa.org/en/latest/>, aksessert 19.05.2020.

⁵<https://www.pylint.org/>, aksessert 19.05.2020.

⁶<https://github.com/psf/black>, aksessert 19.05.2020.

⁷<https://pypi.org/project/pydocstyle/>, aksessert 19.05.2020.

⁸<https://pypi.org/project/bandit/>, aksessert 19.05.2020.

⁹<https://www.python.org/dev/peps/pep-0008/>, aksessert 19.05.2020.

¹⁰Testing av programvare hvor testverktøy har innblikk i programvarens indre.

ikke "blackbox"-testing¹¹. Det ble avgjort å ikke implementere dette da whitebox-testingen QTest tilbyr er tilstrekkelig til vår prototype. For en utdyping av hvordan QTest brukes se seksjon 8.5.

7.2 Manuell testing

I tillegg til automatisert testing gjennom GitLab CI, ble koden også manuelt testet lokalt. Det ble laget script for dette i Bash og Powershell for å støtte gruppemedlemmenes operativsystemer, Bash-utgaven er vist i kodeliste 7.1 og Powershell-utgaven finnes som vedlegg D.1. Scriptet mottar en eller flere mapper eller Python-filer og kjører testene beskrevet i seksjon 7.1.1 og 7.1.2 rekursivt på disse. Bruk av scriptet var svært tidsbesparende fordi man ikke trengte å commite, pushe og vente på pipeline hver gang testresultat var ønskelig. I tillegg stopper pipeline ved første feil den finner, og gjengir bare denne. Siden scriptet gjør seg ferdig og viser alle feil den finner, kunne disse enkelt og raskt løses lokalt.

Kodeliste 7.1: Bash-script for sjekk av kodekvalitet

```
if [[ $# -eq 0 ]]; then
    files=$(find -type f -name '*.py' )
    echo "\nChecking all .py-files"
else
    args=$@

    for file in "${args[@]}; do

        if [[ $file =~ ^[A-Za-z0-9_]+\.[py] ]] && ! [[ $file =~ .pyc ]]; then
            files+=( "$file" )
        else
            files+=$(find $file -type f -name '*.py')
        fi
    done
fi

for file in $files; do
    printf "=====$(basename -- $file)=====\n"
    printf "\n===PYLINT===\n"
    pylint --rcfile=./scripts/pylintrc $file
    printf "===FLAKE8===\n"
    flake8 --config ./scripts/flake8 $file
    printf "\n===BANDIT===\n"
    bandit $file
    printf "\n===PYDOCSTYLE===\n"
    pydocstyle --convention=numpy $file
    printf "\n===BLACK===\n"
    black -l 79 --check --diff $file
done
```

¹¹Testing av programvare hvor testverktøy ikke har innblikk i programvarens indre.

7.3 Brukertestning

Det var viktig at programvaren ble brukertestet på målgruppen. Denne brukertesten ble gjennomført med en mockup av applikasjonen før utviklingsfasen startet, den 17.02.2020. Mockupen som ble brukt finnes som vedlegg C, og var en "low-fidelity" papirprototype. Lo-fi prototype gjorde at vi kunne prøve ut de underliggende design-idéene, strukturen på applikasjonen, innholdet og grunnfunksjonaliteten. Å ha prototypen på papir gjorde det også enklere å gjennomføre brukertesten med "Wizard of Oz"-metoden. Gjennom hele brukertesten hadde vi en testleder som skulle lede testingen. Testlederen fortalte brukeren hva som skulle skje og at brukeren måtte snakke høyt da han prøvde prototypen. Ved siden av testleder hadde vi også to observatører som noterte ned hva som skjedde gjennom testen. Flere varianter av prototypen ble testet i løpet av brukertesten slik at gruppen kunne finne de beste elementene av hver prototype. Notater fra møtet finnes som vedlegg H.

Kapittel 8

Diskusjon

Dette kapittelet inneholder en samling diskusjoner rundt ulike aspekter ved prosjektet. Kapittelet bygger videre på innhold presentert tidligere i rapporten. Teknologimessige valg drøftes, problemer ved gjennomføringen utdypes, og arbeidsprosess blir reflektert over. Til slutt presenteres mangler i implementasjonen, mulige utbedringer og kjente programvarefeil.

8.1 Valg av teknologier

I denne seksjonen følger utdyping av teknologivalg. Teknologier vi ikke fant alternativer til, eller som tidlig viste seg å være et klart valg, er utelatt.

8.1.1 Programmeringsspråk

På bakgrunn av at prosjektresultatet skal være en prototype ønsket oppdragsgiver at det ble brukt et letteselig programmeringsspråk. Oppdragsgiver poengterte at C++ og Assembly ikke er letteselige språk, og at Java var utelukket. Både Go og JavaScript ble tatt opp som alternativer. Begge språkene har støtte for brukergrensesnitt, men på grunn av manglende erfaring med språkene i deler av eller hele gruppen ble disse utkonkurrert. I oppgaveteksten ble Python nevnt som eksempel på letteselig språk. Da alle medlemmene hadde erfaring med Python og språket er anerkjent for å ha et stort og bredt utvalg biblioteker ble det valgt, nærmere bestemt Python versjon 3.7.

8.1.2 GUI-bibliotek

Programvaren skal være en brukervennlig desktop-applikasjon og må kunne kjøre på Windows. Siden halve gruppen foretrekker å utvikle på Linux var det ønskelig med et kryss-plattform-bibliotek som ville fungere på begge operativsystemer. Python har flere gode alternativer, og vi så spesielt på "TkInter"¹, "PyQt5"² og

¹<https://wiki.python.org/moin/TkInter>, aksessert 16.05.2020.

²<https://www.riverbankcomputing.com/software/pyqt/intro>, aksessert 16.05.2020.

”PySide2” [3]. Som utdypet i seksjon 6.1 ble PySide2 valgt til slutt. Dette er fordi vi ville ta i bruk Qt, og i motsetning til PyQt5 tilbys PySide2 under LGPL, som foretrekkes av oppdragsgiver, krav 2.4.4. PySide2 er også det offisielle Python-biblioteket for Qt, og støttes av Qt-utviklerne. Resten av denne seksjonen detaljerer beslutningsgrunnlaget vårt.

Qt er et Open Source kryss-plattform-rammeverk skrevet i C++ [5]. Qt støtter mange plattformer og har et solid fotfeste i markedet. Ifølge ”Qt Group Plc” sely, ble Qt brukt av over en million brukere verden over i 2017. ”Qt Group Plc” påstår også at Qt brukes av ledende aktører i mer enn 70 industrier³. Verdt å nevne er at Rotem Schnoer i boken ”Handbook of Research on Born Globals”⁴ estimerte antall brukere til 350 000 i 2014. Qt kan anvendes fra Python ved bruk av PyQt5 eller PySide2.

PyQt5 muliggjør bruk av Qt fra Python, og er etter vårt inntrykk et populært valg blant GUI-biblioteker til språket. I utgangspunktet er PyQt5 lisensiert med GPL, men mot betaling kan programvaren brukes under LGPL⁵. Heimevernet er for Open Source-lisenser, men LGPL er å foretrekke over GPL. Dette er spesifisert i kravspesifikasjonen under seksjon 2.4.4.

PySide2 er et Python-bibliotek med ”bindings” til Qt i likhet med PyQt5, men regnes av ”The Qt Company” som den offisielle ”wrapperen” av Qt til Python⁶. PySide2 er lisensiert under LGPL, slik som den betalte versjonen av PyQt5. Siden PySide2 er offisielt støttet, finnes dokumentasjonssider lignende de man finner for C++, noe PyQt5 mangler. Vi velger PySide2 i hovedsak på grunn av bedre dokumentasjon og offisiell støtte fra utviklerne av Qt. PySide2 sin største svakhet er at ikke alle funksjonaliteter fra Qt er tilgjengelige⁷, men gruppen forutser bare ”enkel” bruk av Qt.

Tkinter er et annet GUI-bibliotek som brukes mye i Python og regnes som de-facto standard i Python⁸. Tkinter ble vurdert tidlig i prosjektfasen, da den har en lett læringskurve. Dessverre er den best for små og enkle applikasjoner, og vil være begrensende for komplekse applikasjoner.

8.1.3 Database

Et av kravene til programvaren er at den skal kunne kjøres isolert. Den skal kommunisere med en database, og det er ønskelig at denne enkelt skal kunne flyttes

³<https://investors.qt.io/releases/?release=A84F7B0247145FCF>, aksessert 14.05.2020.

⁴*Handbook of Research on Born Globals*, 2014. https://books.google.no/books?id=t6uQNbVYoJoC&pg=PA172&redir_esc=y#v=onepage&q&f=false, aksessert 14.05.2020.

⁵<https://www.riverbankcomputing.com/commercial/license-faq>, aksessert 14.05.2020.

⁶<https://www.qt.io/qt-for-python>, aksessert 14.05.2020.

⁷https://wiki.qt.io/Qt_for_Python_Missing_Bindings, aksessert 14.05.2020.

⁸<https://wiki.python.org/moin/TkInter>, aksessert 15.05.2020.

over på minnepinne. Disse to kravene fører til at databasen er nødt til å være av "embedded" type, og at den for enkelhets skyld bør bestå av en enkelt fil. Valget stod mellom "TinyDB"⁹ og "SQLite3" [4]. TinyDB er en liten "NoSQL"-database som er enkel i bruk, men siden synkronisering senere vil være ønskelig for oppdragsgiver, ble det besluttet å bruke SQLite3. Sistnevnte er i tillegg støttet av Qt [5] sin "QSqlDatabase" som har et "model/view"-grensesnitt for kommunikasjon med databasen gjennom "QSqlTableModel" og "QTableView". En modifisert utgave av disse passer ypperlig for visning og redigering av kodeboken som ligger i databasen.

8.1.4 Serialisering

SOI må skrives til fil for lagring og distribusjon. Vi så her på flere alternativer.

Pickle er Pythons standardbibliotek for serialisering. Fordelen med Pickle er at det er svært lett å bruke da man kan skrive objekter direkte til fil og dermed har støtte for alle Pythons typer. Problemet med Pickle i denne oppgaven er at oppdragsgiver har et krav 2.4.2 om at data skal sendes som US ASCII, mens Pickle lager binære filer. I tillegg har Pickle ulemper som at det er Python-spesifikt, det ikke er menneskeleselig, og at data som leses inn direkte kjøres på maskinen, som sikkerhetsmessig er ugunstig.

XML har fordelen med støtte for kommentarer som kan øke leselighet. På den andre siden er XML tungvint å skrive og krever mange tegn, på grunn av fremgangsmåten `<tag>innhold</tag>`. Dette gjør filer store som er ugunstig når de skal sendes over lav båndbredde.

YML har støtte for kommentarer og bruker få tegn sammenlignet med XML. Dette hadde også vært et godt valg for oss, men er hakket mer vrient å ta i bruk enn valget vi endte opp med: JSON. JSON er et subsett av YML, og i realiteten ville vi nok ikke tatt i bruk mer enn hva JSON kan tilby uansett.

JSON har ikke kommentarer, men er lettvindt å ta i bruk da Python-kode enkelt kan konverteres til JSON. JSON enkodes i UTF-8 hvor de 128 første tegnene har lik encoding som ASCII. Da Python har god støtte for JSON, og vi ikke strengt tatt trenger støtte for kommentarer, ble valget JSON.

8.1.5 Komprimering

Det ble i researchperioden 5.3 skrevet et script, vedlegg D.7, for å sammenligne forskjellige komprimeringsbiblioteker. Resultatet fra dette scriptet dannet grunnlaget for valg av bibliotek. Det ble senere oppdaget en systematisk feil i scriptet

⁹<https://tinydb.readthedocs.io/en/latest/index.html>, aksessert 14.05.2020.

som beskrives i seksjon 8.11.13. Under researchperioden ble scriptet brukt på vilkårlige JSON-filer fra nettet. Det viste seg i etterkant at disse var betydelig større en JSON-filene produsert av programvaren, se kodeliste i vedlegg K.1. Videre følger bibliotekene som ble sammenlignet.

zlib¹⁰ som bruker DEFLATE som igjen er basert på LZ-77 og kom best ut på små filer siden den ikke bruker mye headerdata.

LZMA (Lempel–Ziv–Markov chain algorithm)¹¹ som bruker mer headerdata og gjorde det dårligst på små filer, men komprimerte mest på større filer.

bzip2¹² bruker "Burrows–Wheeler"-algoritmen og plasserte seg mellom LZMA og zlib på alle filstørrelsene.

LZMA kom seirende ut av første runde og programvaren ble implementert med denne da det kreves lite arbeid å bytte mellom de nevnte bibliotekene grunnet lik bruk. Da programvaren var i et stadie der den kunne produsere realistiske SOI-er og eksportere dem som JSON-filer ble det laget et sett med SOI-er i økende størrelser. Disse filene ble kjørt gjennom det samme scriptet som tidligere og det ble bekreftet at valget av bibliotek var korrekt. Da scriptet ble korrigert for feilen som diskuteres i seksjon 8.11.13, viste det seg at LZMA fortsatt var det beste alternativet blant bibliotekene diskutert i denne seksjonen, selv om utbytte av komprimeringen er svært mye lavere enn tidligere antatt.

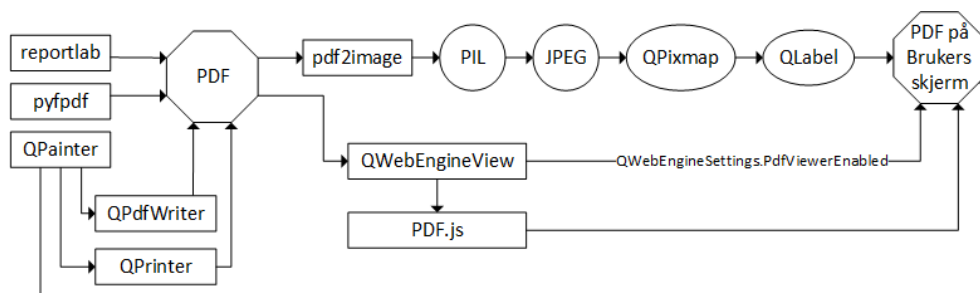
8.1.6 PDF-behandling

Programvaren må produsere PDF-filer til både SOI og kodebok, og forhåndsviser hvordan SOI vil se ut på PDF. Så fort vi slo fast Python og Qt som kjerneteknologier, begynte letingen etter en løsning som tillater både produksjon og visning av PDF. Av alle teknologimessige valg vi måtte ta var nok dette det vanskeligste, fordi vi kom over flere løsninger, og ingen av dem var tydelig bedre enn de andre. For å demonstrere valgene vi sto ovenfor har vi laget figur 8.1. Figuren er en rettet asyklisk graf hvor runde noder representerer data, firkantede noder representerer teknologier og åttekantede noder viser de ønskede resultatene vi ville ha, nemlig en PDF-fil og visning av PDF på brukerens skjerm. Kanter representerer overføring eller skaping av data. Med denne grafen som referanse kan vi presentere de ulike løsningene vi vurderte.

¹⁰<https://en.wikipedia.org/w/index.php?title=Zlib&oldid=940510830>, aksessert 14.05.2020.

¹¹https://en.wikipedia.org/w/index.php?title=Lempel%E2%80%93Ziv%E2%80%93Markov_chain_algorithm&oldid=931340969, aksessert 14.05.2020.

¹²<https://en.wikipedia.org/w/index.php?title=Bzip2&oldid=938289875>, aksessert 16.05.2020.



Figur 8.1: Visualisering av mulige valg for PDF-behandling

Produksjon av PDF

Som figur 8.1 viser kan produksjon av PDF gjøres på fire måter. Med "ReportLab" [8] eller "PyFPDF"¹³, to Python-biblioteker for produksjon av PDF, eller ved å bruke "QPainter" på enten en "QPdfWriter" eller en "QPrinter". PyFPDF var biblioteket vi så på først, og virket lenge ledende. PyFPDF er en Python-"port" av PHP-biblioteket FPDF¹⁴, og tilbyr et enkelt, men lavnivå grensesnitt. I vårt tilfelle ble grensesnittet desverre for lavnivå, da spesielt tabeller var vrient å produsere. Da mye av innholdet vi planla å fylle PDF med var nettopp tabeller, ble dette grunnlag for å lete etter noe nytt. ReportLab ble da undersøkt, og det viste seg å være akkurat det vi trengte. ReportLab brukes tilsynelatende i hovedsak til å generere rapporter og lignende med automatisk formatering, ikke ulikt \LaTeX . Vi ønsket å gjøre formateringen selv, så dette bruksområdet var vi ikke interessert i, men biblioteket tillater heldigvis også bruker å gjøre lavnivå endringer. Til vårt bruk kunne vi utnytte ReportLabs gode høynivå støtte for til eksempel tabeller, med lavnivå operasjoner som manuell formatering.

Bruk av QPainter i kombinasjon med en av QPdfWriter og QPrinter tar en helt annen tilnærming, da tegning først skjer gjennom Qts tegnesystem, og deretter oversettes til PDF. Se seksjon 6.8 for en introduksjon til Qts tegnesystem. Dette kan utnyttes enten ved å tegne alt fra bunnen av med QPainters grensesnitt, eller ved å tegne widgets i Qt direkte. Å basere seg på widgets gjør at man automatisk får samme resultat på skjerm som på PDF. Når det kommer til forskjellen mellom QPdfWriter og QPrinter er vi usikre, da vi aldri fikk QPdfWriter til å fungere, mens QPrinter fungerte med en gang. Merk at fordi QPdfWriter er spesiallaget til nettopp å lage PDF-filer vil nok denne fungere best av de to.

Visning av PDF

Som figur 8.1 viser kan dette gjøres på 3 måter. Ved å gjøre PDF til en JPEG, og vise denne på skjerm via "QLabel", med "QWebEngineView" direkte, eller i JavaScript-biblioteket "PDF.js"¹⁵. Verdt å merke er at om vi går ut ifra at produsering av PDF

¹³<https://pyfpdf.readthedocs.io/en/latest/>, aksessert 19.05.2020.

¹⁴<http://www.fpdf.org/>, aksessert 14.05.2020.

¹⁵<https://mozilla.github.io/pdf.js/>, aksessert 19.05.2020.

vil skje via QPainter er en annen gyldig løsning å tegne QPainter direkte til skjerm, som beskrevet i forrige seksjon.

Den lange veien innom JPEG var metoden vi forsøkte først. Prosessen starter ved at PDF blir omgjort til et bilde via biblioteket "pdf2image"¹⁶. Bildet oppgis via "Python Imaging Library", derav navnet "PIL". PIL omgjøres til JPEG, og denne lastes som QPixmap. QLabel brukes til å vise frem QPixmap på skjerm til brukeren. Denne prosessen tok opp til flere sekunder med SOI-lignende PDF-er under testingen vår. Dette ville vært akseptabelt om prosessen ville blitt brukt til bare å vise frem PDF i ny og ne, men til vårt bruk ble den uakseptabel, da PDF vil vises til bruker ofte i programvaren.

QWebEngineView tilbyr en nettleser inne i Qt-applikasjoner. Den enkleste måten å utnytte dette på for å vise PDF-er er å skru på innstillingen *QWebEngineSettings.PdfViewerEnabled*, som vil gi nettleseren mulighet for fremvisning av PDF med maskinens interne PDF-viser. Alt som behøves for å vise bruker en PDF er da å sikte nettleseren til ønsket PDF. Nettleserenvinduet kan til og med gjøres så diskret at bruker ikke vil være klar over at operasjonen skjer i en nettleser. En PDF-viser basert på dette er implementert i vår innlevering via klassen "PDFPreviewWidget" i modulen *soitool.pdf_preview_widget.py*, og fungerer flott. Den ble i praksis aldri brukt, men gjenlates i kildekoden for å demonstrere bruk av *QWebEngineSettings.PdfViewerEnabled*.

En annen måte å utnytte QWebEngineView på som ikke krever at bruker har en PDF-viser på sin datamaskin er ved å løse problemet i ren JavaScript med biblioteket PDF.js. Dette ble ikke utforsket, da det andre nettleserbaserte alternativet fungerte uten trøbbel. Til en seriøs ferdigstilling av SOI-tool kan dette være aktuelt om det er ønskelig å ikke knytte seg til innebygde PDF-visere. Et viktig poeng å ta stilling til om PDF.js skal brukes er dog om man vil åpne for en hel ny verden av problemer ved å inkludere JavaScript i en programvare ellers basert på Python og C++.

Valgt fremgangsmåte for SOI

For SOI ble QPainter valgt, da dette forenklet utviklingen i stor grad. Istedenfor å måtte programmere representasjoner av SOI til både PDF og skjerm, tillot dette oss å bruke skjerm-utgaven til PDF. Dette sparte oss for mye tid. ReportLab ville nok produsert bedre SOI-er både utseendemessig og med tanke på filstørrelse på PDF, men til en for stor bekostning av tid i vårt prosjekt.

Valgt fremgangsmåte for kodebok

Til forskjell fra SOI trengte vi ikke å vise kodeboken på skjerm, så ReportLab ble et godt valg. Bruken av ReportLab til kodeboken kan brukes som referansepunkt for å se for seg hvordan SOI kunne blitt skrevet ut på samme måte.

¹⁶<https://github.com/Belval/pdf2image>, aksessert 19.05.2020.

8.1.7 Kodegenerering

Pythons modul "random" er pseudo-tilfeldig¹⁷ og egner seg best til modellering og simulering¹⁸, den regnes ikke som kryptografisk sikker. Da denne oppgaven er en prototype ville det nok vært tilstrekkelig å bruke denne modulen, men siden tiltaket med å endre til modulen "secrets" var minimalt valgte vi å gjøre dette. "secrets" bruker entropien til operativsystemet og regnes som kryptografisk sikker¹⁹.

Ved å bruke "Python set" for generering av mengder med "koder" vil man unngå duplikater. Python-settet tillater ikke duplikater, derfor kan man kontinuerlig legge til "koder" til ønsket størrelse på settet er nådd, vel vitende om at settet kun inneholder unike koder. En annen fordel med å bruke sett er når en enkel kode skal legges til et eksisterende sett med koder, da vil det være raskere å sjekke om denne alt ligger i settet enn å sjekke om den alt ligger i en liste. Fordelen med lister er at man har indeksering, men dette er ikke nødvendig for mengdene med koder da de tilfeldig deles ut.

Ved bruk av tilfeldig kodegenerering når man lager sett av koder, vil det for hver kode som blir lagt til i settet øke sannsynligheten for at den neste koden som genereres alt er i settet og "går til spille". Denne sannsynligheten øker altså når antall ubrukte permutasjoner for nåværende kodelengde minker. Faktumet at svært mange genereringer "går til spille" regnes ikke som et problem da det krever svært lite ressurser av maskinen, og slike genereringer ikke vil forekomme ofte.

8.1.8 System for versjonskontroll

GitLab velges fordi NTNU Gjøvik har egen installasjon vi kan bruke med støtte for GitLab CI og GitLab Pages. GitHub og BitBucket ble også vurdert, men fort avslått da vi konkluderte med at vi raskere kunne bli produktive på GitLab, siden ett gruppemedlem var godt kjent med mulighetene der. Det finnes også en sentralisert tjeneste for GitLab, gitlab.com, men skolens tjeneste ble valgt da vi også fikk ressurser til arbeidsmaskiner på skolens infrastruktur, og disse var enklere å nå fra skolens GitLab enn den sentraliserte utgaven.

8.1.9 Dokumentasjon

Docstring-konvensjon

Vi så hovedsakelig på "Numpydoc" og "Sphinx". Numpydoc velges fordi syntaxen virker mest oversiktlig, som demonstrert i kodeliste 8.1.

¹⁷Pseudo-tilfeldige tall genereres ved algoritmer uten ekstern entropi.

¹⁸<https://docs.python.org/3/library/secrets.html>, akssert 16.05.2020.

¹⁹<https://realpython.com/lessons/cryptographically-secure-random-data-python/>, akssert 19.05.2020

Kodeliste 8.1: Numpydoc sammenlignet med Sphinx

```

1 # Numpydoc
2 def divide(a, b):
3     """Divide a by b.
4
5     Parameters
6     -----
7     a : float
8         Dividend.
9     b : float
10        Divisor.
11
12    Returns
13    -----
14    float
15        Quotient
16
17    Raises
18    -----
19    ZeroDivisionError
20        Indicate division by zero.
21    """
22    return a / b
23
24 # Sphinx
25 def divide(a, b):
26     """Divide a by b.
27
28     :param a: Dividend
29     :type a: float
30     :param b: Divisor
31     :type b: float
32     :return: Quotient
33     :rtype: float
34     :raises ZeroDivisionError: Indicate division by zero.
35     """
36     return a / b

```

Generering av dokumentasjon

”pdoc3”²⁰ velges for å generere kode-dokumentasjon. Dette verktøyet støtter NumPy sitt docstring-format og kan generere både Markdown, PDF og HTML-filer. Vi velger å bruke HTML-filer fordi disse ser best ut, og fordi generering av PDF krever ytterligere programvare og legger til ekstra steg i prosessen. Generert dokumentasjon ble forøvrig også publisert til en nettside som en del av pipeline på adressen <http://bachelor-paa-bittet.pages.stud.idi.ntnu.no/soitool/> under utviklingsperioden.

²⁰<https://pypi.org/project/pdoc3/>, aksessert 16.05.2020.

8.1.10 Produksjon av eksekverbar fil

”PyInstaller”²¹ og ”fbs”²². ble vurdert. Begge lager eksekverbar fil til operativsystemet de kjøres på. Siden vi skal bruke Python 3.7, og fbs bare støtter versjon 3.5 og 3.6, valgte vi PyInstaller.

8.2 Modul-arkitektur

Som nevnt i seksjon 3.1 har alle moduler i programvaren fellestrekk. Den objekt-orienterte måten å modellere dette på er arv og overstyring av funksjoner, men det kan også gjøres med interfaces.

Python har ikke interfaces innebygd, og biblioteket ”python-interfaces”²³, som ville gitt denne funksjonaliteten, støtter ikke Python 3.7. Vi ville uansett ha støtt på et problem fordi arving av et interface og en QWidget samtidig vil føre til en metaklasse-konflikt²⁴.

Følgende avsnitt beskriver det gruppen mener er den optimale løsningen, vist i kodeliste 8.2. For å unngå metaklasse-konflikt når det arves fra flere klasser samtidig, ble vi nødt til å bruke ”ABC”²⁵, ”Abstract Base Classes”. Dette biblioteket har en ”decorator” kalt *abstractmethod* (linje 6), som forteller at spesialiseringer ikke kan bli initialisert hvis de mangler implementasjon av metodene. Når vi initialiserer *DerivedClass* (linje 13) med PyQt5, kastes en error fordi *DerivedClass* ikke har implementert metoden *method_to_override*. Dette er den ønskede oppførselen, men når vi bytter fra PyQt5 til PySide2 (linje 2 og 3) kastes ikke erroren, og hele poenget er borte. Valget mellom PyQt5 og PySide2 diskuteres i seksjon 8.1.2.

Kodeliste 8.2: Optimal løsning

```

1 from abc import ABC, abstractmethod
2 #from PySide2 import QtWidgets
3 from PyQt5 import QtWidgets
4
5 class Interface(ABC):
6     @abstractmethod
7     def method_to_override(self):
8         pass
9
10 class Meta(type(Interface), type(QtWidgets.QTableWidget)):
11     pass
12
13 class DerivedClass(Interface, QtWidgets.QTableWidget, metaclass=Meta):
14
15     def __init__(self):
16         QtWidgets.QTableWidget.__init__(self)
17         super(QtWidgets.QTableWidget)
```

²¹<https://pypi.org/project/PyInstaller/>, aksessert 14.05.2020.

²²<https://pypi.org/project/fbs/>, aksessert 14.05.2020.

²³<https://pypi.org/project/python-interface/>, aksessert 16.05.2020.

²⁴<https://docs.python.org/3/reference/datamodel.html> 3.3.3.1, aksessert 18.05.2020.

²⁵<https://docs.python.org/3/library/abc.html>, aksessert 16.05.2020.

Dette avsnittet beskriver den valgte løsningen vist i kodeliste 8.3. Løsningen var å bruke "NotImplementedError". Ulempen er at spesialiseringer kan initialiseres uten å ha overstyrt funksjoner, og at det først kommer error når funksjonene kalles. Vi ser likevel ikke på dette som et stort problem fordi testene vil oppdage dette ved at generaliseringens metode kaster NotImplementedError når spesialiseringen ikke har metoden. I tillegg avdekkes dette av kodekvalitet-scriptene vist i kapittel 7. Derfor kunne vi trygt velge denne løsningen selv om bruk av *abstract-method* hadde vært noe bedre.

Kodeliste 8.3: Valgt løsning

```
class Interface(ABC):
    def method_to_override(self):
        raise NotImplementedError
```

8.3 Redigering av moduler

Under designfasen hadde gruppen to versjoner av hvordan man kunne redigere modulene i en SOI. Den ene var en versjon hvor man redigerte direkte i modulene og den andre hadde en ekstra dialog som man kunne redigere modulene med. Brukte man den første versjonen kunne man skrive direkte inn i modulene og endre størrelsene på dem i SOI-en. Med den andre versjonen fikk man opp en ekstra dialog etter å ha valgt en modul. I denne dialogen kunne man skrive inn og endre størrelse. Når man var ferdig med modulen ble den lagt inn i SOI-en. Skulle man dermed redigere en modul, ville man få opp samme dialog slik at man kunne redigere modulen igjen. Versjonen med direkte redigering endte opp å bli den gruppen brukte videre i utviklingsfasen, fordi den virker mer naturlig for brukere.

8.4 Drøfting av implementasjon av SOI-redigering

Vi valgte å implementere SOI-redigering via "QWidgets" i "QGraphicsScene" via "QGraphicsProxyWidget" som beskrevet i seksjon 6.4. Dette gjorde utvikling enklere, men har noen ulemper. Som presisert i QGraphicsProxyWidgets dokumentasjon²⁶, er klassen ikke ment for reelt bruk med ytelseskrav, men heller som en enkel måte å fylle QGraphicsScene med innhold. Ved store SOI-er observerer vi at programvaren blir noe treg, avhengig av hvilke maskin den kjører på. Etter vår forståelse kan en oppnå det samme grensesnittet vi har laget ved bruk av QGraphicsProxyWidgets ved å lage egne subklasser av QGraphicsItem²⁷. En annen ulempe ved QGraphicsScene er at den virker som nærmest erstattet av "Qt

²⁶<https://doc.qt.io/qt-5/qgraphicsproxywidget.html#details>, aksessert 15.05.2020.

²⁷Slides fra presentasjon "The Graphics View Canvas" <https://kogs-www.informatik.uni-hamburg.de/~seppke/content/teaching/sose16/bvproj/QGraphicsView.pdf>, aksessert 15.05.2020.

Quick Scenegraph”²⁸, en teknologi vi ikke undersøkte særlig grundig.

Da oppgavens formål 1.4 var å lage en prototype for å demonstrere muligheter ved egen programvare for SOI tillot vi bruk av QGraphicsProxyWidget på tross av de ytelsesmessige ulempene.

8.5 GUI-testing i Qt via QTest

Som beskrevet i seksjon 7.1 brukes ”QTest” [9] til testing av brukergrensesnittet til programvaren. Vi klarte å bruke QTest til testing av knapper, tekstfelt og lister, samt selvlagde widgets basert på disse. Innskriving i tabellceller og utløsning av hendelser rundt flytting av mus fikk vi derimot ikke til å fungere. For å teste en hel widget ble det derfor av og til nødvendig å ”jukse” med deler av interaksjonen ved å interagere med dem programmatisk.

8.5.1 Testing av modulære dialoger

Et problem vi støtte på var testing av ”modulære”, eller på engelsk: ”modal”, dialoger. Dialoger som er modulære tar over ”event-tråden” til de fullføres, for å hindre at noe annet arbeid blir utført mens dialogen kjører. For bruker betyr dette at hovedvindu låses til dialog er fullført, en effekt som er ønskelig. For vår programvare betyr det at koden som starter dialogen ikke vil fortsette før dialogen fullfører. Under testing er dette problematisk, siden vi ofte vil interagere med dialog som en del av tester. Kodeliste 8.4 viser et minimalt eksempel på problemet, hvor vi trykker på en knapp som åpner en modulær dialog i linje 6, og deretter forsøker å lukke dialogen i linje 14. Koden vil stoppe ved linje 6, fordi den modulære dialogen tar over ”event-tråden”, og stopper linje 14 fra å kjøre.

Kodeliste 8.4: Ikke-fungerende test av modulær dialog

```

1 # Widget vi vil teste som har en knapp som property 'button'. Knappen åpner
2 # en modulær dialog som kan lukkes ved å trykke på Enter
3 widget = WidgetWithButtonActivatedDialogClosableByEnterButton()
4
5 # Å trykke på knappen åpner en modulær dialog
6 QTest.QTest.mouseClick(widget.button, QtCore.Qt.LeftButton)
7
8 # Her forsøker vi å trykke Enter på dialogen ved å nå tak i den via
9 # 'app.activeModalWidget()'
10 # Se https://doc.qt.io/qt-5/qapplication.html#activeModalWidget
11 #
12 # Denne linjen kjører ikke før dialogen over lukkes, dermed vil den aldri klare
13 # å lukke dialogen
14 QTest.QTest.keyClick(app.activeModalWidget(), QtCore.Qt.Key_Enter)
15
16 # Vi når aldri hit...
17 print("Suksess!")

```

²⁸<https://www.qt.io/blog/2017/01/19/should-you-be-using-qgraphicsview> og <https://www.qt.io/blog/2017/01/19/should-you-be-using-qgraphicsview>, aksessert 15.05.2020.

Løsningen vår på dette problemet er funksjonen `QtCore.QTimer.singleShot`²⁹, som kan brukes for å utføre oppgitt handling etter oppgitt mengde tid. Funksjonen gjør i praksis at oppgitt handling blir utført på "event-tråd" på et senere tidspunkt, noe vi kan utnytte for å teste modulære dialoger. Kodeliste 8.5 viser et minimalt eksempel basert på kodeliste 8.4, hvor lukking av åpnet dialog skjer via `QtCore.QTimer.singleShot`. I dette eksempelet vil linje 19 nås, og "Suksess" skrives til skjerm.

Merk at i kodeliste 8.4 brukes 10 millisekunder som ventetid gitt til `QtCore.QTimer.singleShot`. Dette ble gjort for demonstrasjon, da det er lettere å se for seg scenarioet slik. I våre tester har vi konsekvent brukt ventetider på 0 millisekunder for å unngå å sette en kunstig ventetid. Dette har fungert godt i alle våre tester, men det er ikke utenkelig at Qt kan finne på å utføre `press_enter_on_dialog` mellom linje 14 og linje 17, noe som vil ødelegge testen. Vår tilnærming har vært å heller reagere på dette om det skjer i praksis, og vi har aldri observert at det har skjedd.

Kodeliste 8.5: Fungerende test av modulær dialog

```

1 # Widget vi vil teste som har en knapp som property 'button'. Knappen åpner
2 # en modulær dialog som kan lukkes ved å trykke på Enter
3 widget = WidgetWithButtonActivatedDialogClosableByEnterButton()
4
5 def press_enter_on_dialog():
6     # Her forsøker vi å trykke Enter på dialogen ved å nå tak i den via
7     # 'app.activeModalWidget()'
8     # Se https://doc.qt.io/qt-5/qapplication.html#activeModalWidget
9     QTest.QTest.keyClick(app.activeModalWidget(), QtCore.Qt.Key_Enter)
10
11 # Forteller Qt at funksjonen 'press_enter_on_dialog' skal utføres om 10
12 # millisekunder. I praksis gjør dette at funksjonen får kjøre **etter** den
13 # modulære dialogen har blitt åpnet
14 QtCore.QTimer.singleShot(10, press_enter_on_dialog)
15
16 # Å trykke på knappen åpner en modulær dialog
17 QTest.QTest.mouseClick(widget.button, QtCore.Qt.LeftButton)
18
19 print("Suksess!")

```

En løsning som unngår problemet er å bruke modulære dialoger som ikke tar over "event-tråden". Dette er mulig ved `setModal(True)`³⁰ i kombinasjon med at dialogen starter med `show()` istedenfor `exec_()`. Vi gikk ikke for denne løsningen da vårt inntrykk er at dette ikke er vanlig i Qt.

8.5.2 Tilgang til omfavnede widgets

For å interagere med widgets under testing må vi nå tak i en referanse til dem. Når en widget er omfavnet av en annen er dette i utgangspunktet et problem. Vi ser to måter å gjøre dette på. En er å lagre omfavnede widgets som attributter til omfavnende widget. Kodeliste 8.5 viser et eksempel på dette i linje 17, hvor det

²⁹<https://github.com/pytest-dev/pytest-qt/issues/256>, aksessert 15.05.2020.

³⁰<https://doc.qt.io/qt-5/qdialog.html#modal-prop>, aksessert 15.05.2020.

klikkes på en knapp inne i ”widget” ved å sikte på dens ”button”-attributt. Det er denne løsningen som blir brukt i vår implementasjon.

En annen måte å løse dette på er å ta i bruk *findChild*³¹, en medlemsfunksjon alle widgets har som tillater søking etter omfavnede widgets. Søking skjer på type, så om man vil nå tak i en knapp inne i en widget ”A” kan dette gjøres med *A.findChild(QPushButton)*. Denne metoden er fleksibel, men gruppen vurderer den til å være unødvendig komplisert å skrive og lese.

8.6 Skjermoppløsningsproblemer

Under utvikling merket vi at innhold i SOI-er fikk ulik størrelse på forskjellige datamaskiner. Dette resulterte i at en SOI laget med én datamaskin ikke ble lik en SOI med samme informasjon laget på en annen datamaskin, og førte i noen tilfeller til at åpning av SOI laget med en annen datamaskin kræsjet programvaren. Ved undersøkelse viste det seg at fonter oppgitt med samme punktstørrelse fikk ulik pikselstørrelse basert på brukerens skjermoppløsning. Dette var et problem, da hele vår løsning baserer seg på pikselstørrelser. Som et eksempel er SOI 1700 piksler bred, så om en tekst blir 200 piksler bred på én skjerm og 300 piksler bred på en annen vil resulterende PDF bli ulik. Dette ble løst ved å bruke pikselstørrelser istedenfor punktstørrelser på fonter brukt i SOI.

En test ble skrevet for å oppdage videre skjermoppløsningsproblemer, og denne ligger i modulen *test.test_resolution_smoke_test*. Ved å kjøre denne testen på ulike oppløsninger har vi avdekket at programvaren vår ikke fungerer riktig på skjermoppløsningsskaleringer over 125%.

Skjermoppløsningsproblemene ble oppdaget sent i prosjektperioden, så det eksisterer usikkerhet rundt løsningen vår. Vi har derfor lagt til tabell 8.1 som viser hvilke skjermoppløsninger og skaleringer våre datamaskiner har hatt under utvikling og vi dermed vet at programvaren fungerer med.

Oppløsning	Skalering
1600 x 900	100%
1920 x 1080	100%
1920 x 1080	125%
2560 x 1440	100%

Tabell 8.1: Skjermoppløsninger og skaleringer gruppen vet med stor sikkerhet programvaren fungerer godt på

8.7 Heftelse ved komprimeringsløsning

Vår implementasjon av komprimering 6.7.2 typekonverterer de resulterende bytes til streng før de skrives til fil. Dette for å imøtekomme kravet om US ASCII 2.4.2.

³¹<https://doc.qt.io/qt-5/qobject.html#findChild>, aksessert 15.05.2020.

Konverteringen har en stor ulempe ved at størrelsen på de komprimerte dataene øker betydelig for filer under en viss størrelse. I vedleggsfigur D.1 kan man se en grov oversikt over dette. Den implementerte komprimeringen av SOI er altså ikke lønnsom før filstørrelsen er opp mot det som maksimalt er ønsket for å holde sendetid under ett minutt. I praksis bør man som bruker ikke komprimere SOI før den inneholder minst to fylte sider.

8.8 Databasesynkronisering

Vi vil her grovt skissere to muligheter for synkronisering av data mellom to eller flere databaser. Den første muligheten er å ta i bruk eksisterende verktøy som kan ta seg av dette, eksempelvis "LiteSync"³² som fungerer med valgt database, SQLite3.

Den andre muligheten er å utvikle en egen klient/server-struktur, eksempelvis med sockets, der serveren har "hoved-databasen" og hver klient har sin egen lokale database. Data kan synkroniseres mellom dem gjennom nettverksmeldinger som følger en forhåndsdefinert protokoll, eksempelvis i JSON-format. Klienter vil dermed sende egne endringer til serveren, og motta endringer utført av andre klienter fra serveren. Denne løsningen med lokal database gjør det mulig for klienter å arbeide i frakoblet tilstand, og data kan synkroniseres ved neste tilkobling. Serveren må belage seg på samtidige meldinger og enhver databasetransaksjon må følge ACID-prinsippene³³ for å opprettholde databasens integritet. Denne beskrivelsen går ikke dypt inn i selve synkroniseringen da det er et svært krevende tema ved mange samtidige og/eller frakoblede klienter med hver sin lokale database, og gruppen ikke har kompetanse på området.

8.9 Refleksjon over fremdrift

Figur 8.2 viser et Gantt-diagram laget i prosjektets slutfase. Sammenlignet med det opprinnelige diagrammet laget under prosjektplanleggingsfasen B.1 er flere oppgaver fjernet. Dette er gjort konsekvent for å vise de grove trekkene i fremdriften. Datoer i diagrammet er oppdatert ved å inspisere referat og kildekode, og oppgaven "Fysisk møte med bruker. Gjennomgang av mockup" er lagt til.

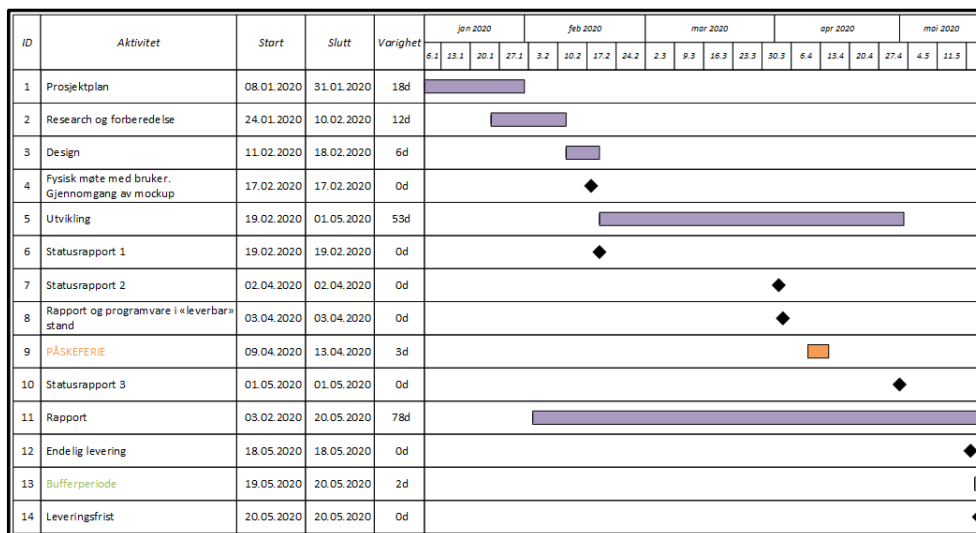
Ved å sammenligne planlagt Gantt i vedlegg B.1 og oppdatert Gantt i figur 8.2 kan man gjøre flere observasjoner om fremdriften. Mest tydelig er at utviklingsfasen i realiteten varte til 01.05.2020, mens den var planlagt å vare til 17.04.2020. Dette gjorde den planlagte innleveringsdatoen 11.05.2020 urealistisk, og datoen ble flyttet til 18.05.2020, med bare 2 dager bufferperiode fra innlevering til endelig frist. En annen observasjon rundt utviklingsfasen er at alle underoppgavene er fjernet i det nye diagrammet. Dette er for å vise at hele utviklingsarbeidet i realiteten ble utført "smidig", og ikke strengt fulgte "MVP"-datoene som planlagt.

³²<http://litesync.io/en/sqlite-synchronization.html>, aksessert 17.05.2020.

³³ACID: Atomicity, Consistency, Isolation & Durability.

Det nye diagrammet viser også at det bare ble ett fysisk møte med brukere, og at dette møtet ble mye tidligere enn forventet. Møtet ble brukt til intervju og brukertesting av mockup som beskrevet i seksjon 7.3.

Milepælen for å ha rapport og programvare i ”leverbar” stand 03.04.2020 er uendret i det nye diagrammet, for å vise at dette målet ble nådd som planlagt.



Figur 8.2: Gantt med fremdrift sett i ettertid

8.9.1 Covid-19s effekt på fremdriften

12.03.2020 kom det omfattende tiltak fra regjeringen på grunn av koronavirus-sykdommen covid-19, og NTNU stengte sine campuser for fysisk oppmøte samme dag. For bacheloroppgaven medførte dette to store konsekvenser: Gruppemedlemmene måtte jobbe fysisk separert fra hverandre, og å holde flere fysiske brukertester ble uaktuelt. Frem til 12.03.2020 var fremdriften i arbeidet god, men etter 12.03.2020 fikk fremdriften et støt, trolig på grunn av omstillingen til å måtte samarbeide digitalt. Det tekniske ved arbeidet ble ikke påvirket, da dette allerede i stor grad skjedde over internett. Koordinasjon ble dog en utfordring. Gruppen tok i større grad i bruk kommentarverktøy på GitLab, og holdt regelmessige digitale møter. Digital kommunikasjon viste seg desverre ikke å være en fullverdig erstatning for fysisk samarbeid, da man ble fysisk begrenset. I tillegg var det for noen vanskelig å konsentrere seg alene.

8.10 Mangler

Mangler er fravær av funksjonalitet som gjør at programvaren ikke imøtekommer kravspesifikasjonen gitt av kapittel 2. Mangler til programvaren beskrives i følgende seksjoner.

8.10.1 Sambandsdiagram

Sambandsdiagram 1.10.5 er den eneste av de nødvendige modulene 1.3 som ikke er implementert. Hovedårsaken til dette er prioritering av tid som en konsekvens av misforståelser. Oppgaven hadde en mulig utvidelse hvor gruppen skulle utvikle funksjonalitet for å la brukeren designe og generere sambandsdiagram. Dette var altså ikke en nødvendighet for oppgaven og i lys av dette har gruppen delvis misforstått og trodd at sambandsdiagram som helhet ikke var en nødvendig del av oppgaven. Dette ble nøstet opp i mot slutten av prosjektperioden og burde vært løst tidligere.

Teknisk kan modulen implementeres ved å la brukeren laste inn et bilde og bruke dette som en modul. I dag bruker Heimevernet Powerpoint eller annet program med mulighet for tegning kombinert med utklippsverktøy. Komplikasjoner som kan forekomme ved å la brukeren laste inn et bilde er at bilder har en størrelse som gjør at sending av fil vil ta lengre tid og muligens overstige krav om overføringstid 2.4.2. Enkleste løsning her er ved bruk av et dialogvindu å varsle brukeren om at handlingen vil øke sendetiden kraftig. En mer komplisert, men bedre løsning ville vært å gi brukeren valg for justering av bildet før det legges til. Da kan brukeren redusere bildet så mye som ønskelig, samtidig som budskapet i bildet bevares. Muligheter for å redusere filstørrelsen på bilder er:

- Redusere farger til svart/hvitt.
- Redusere oppløsningen på bildet.
- Redusere dimensjonene på bildet.

Modulen i sin helhet med muligheter for reduserende operasjoner ser gruppen som fullt mulig å implementere om det hadde vært tilstrekkelig tid. Flere av gruppemedlemmene har hatt emnet "IMT3881 - Vitenskapelig programmering"³⁴ hvor dette var blant temaene.

Modulen ville i tillegg til støtte for sambandsdiagram i praksis gjort det mulig å legge til det man måtte ønske av innhold så lenge det er på bildeform. Dette ville gjort det mulig å replikere en hvilken som helst SOI og programvaren ville vært svært mye mer komplett, selv om dette ikke hadde vært en god nok løsning for alle potensielle ønskelige moduler.

Bevis på implementasjon av bildemodul

Kodeliste 8.6 viser hvordan en enkel bildemodul kan implementeres. Modulen gjør det mulig å legge til bilder på SOI, og støtter dermed også sambandsdiagram i bildeform. Hovedtrekk ved implementasjonen er at bildefil lastes ved standard filutforsker, at bilde vises i widget via QPixmap, og at modulen serialiseres ved å omforme bildet til tekst. Modulen fungerer som den skal, men er ikke lagt til i kodebasen som leveres fordi den ble laget mot slutten av prosjektet og er dermed ikke testet fullt.

³⁴<https://www.ntnu.no/studier/emner/IMT3881#tab=omEmnet>, akseptert 11.05.2020.

Kodeliste 8.6: Fungerende modul for bilder på SOI

```

import os
from PIL import Image
from PySide2.QtWidgets import QDialog, QLabel
from PySide2.QtGui import QIcon, QPixmap
from soitool.modules.module_base import ModuleBase
from PIL.ImageQt import ImageQt
from soitool.soi import SOI

class Meta(type(ModuleBase), type(QLabel)):
    pass

class ModuleImage(ModuleBase, QLabel, metaclass=Meta):
    def __init__(self, data=None):
        self.type = ModuleImage.__name__
        QLabel.__init__(self)
        ModuleBase.__init__(self)

        if data is None:
            file_path = QDialog().getOpenFileName(
                self, "Åpne bilde", os.getcwd(), "Bilde (*.png)"
            )[0]
            pixmap = QPixmap(file_path)
            self.image = Image.open(file_path)
        else:
            self.image = Image.frombytes(
                data[0], data[1], data[2].encode("latin1")
            )
            pixmap = QPixmap.fromImage(ImageQt(self.image))
        if (
            pixmap.width() > SOI.CONTENT_WIDTH
            or pixmap.height() > SOI.CONTENT_HEIGHT
        ):
            raise Exception(
                "Bildet er høyere og/eller bredere enn SOI-en og kan ikke "
                "legges til."
            )
        self.setPixmap(pixmap)

    def get_size(self):
        size = self.sizeHint()
        return (size.width(), size.height())

    def get_data(self):
        # The line below is from https://stackoverflow.com/a/56428453/12864311
        string = self.image.tobytes().decode("latin1")
        return [self.image.mode, self.image.size, string]

    def prepare_for_pdf_export(self):
        pass

    @staticmethod
    def get_user_friendly_name():
        return "Bilde"

    @staticmethod
    def get_icon():
        return QIcon()

```

8.10.2 Feil ved innlesing av konfigurasjonsfiler

Brukervennlighetskravene 2.4.1 spesifiserer at programvaren skal gi bruker tilbakemeldinger ved uventede feil. For moduler med konfigurasjonsfiler 6.5.3 er dette ikke tilfelle om innlesning av fil feiler. Om filen ikke er tilstede eller har feil syntaks vil ikke modulens initialiseringsfunksjon fullføres, ingenting vil bli lagt til i SOI og en feilmelding vil bli skrevet til konsoll.

Problemet kan løses ved å legge innlesingen av filen i en "try", og i tilhørende "except" må det åpnes en dialog 6.9 med beskjed om hva feilen er. En ytterlig mulighet er at programmet bruker en enkel standard-struktur som er hardkodet i programmet for at brukeren fortsatt skal ha litt "spillerom" om fil-innlesing feiler.

8.11 Utbedringer

Mulige utbedringer er funksjonalitet eller forbedringer av funksjonalitet som vil gjøre programvaren mer fullverdig, men ikke er et krav i henhold til kravspesifikasjonen, kapittel 2. I denne seksjonen er beskrivelse av utbedringer begrenset til de utbedringene som er svært nyttige og tiltrengte, og de utbedringene som krever lite innsats og er klare til implementasjon, såkalt "lavthengene frukt".

8.11.1 Fjerning av moduler i SOI

I applikasjonen er det ikke mulig å fjerne moduler som er lagt til i en SOI. Sletting av moduler er formentlig en grunnleggende handling i redigeringsapplikasjoner og man kan påstå at det forventes i møte med dem. Årsaken til at det imidlertid ikke er implementert er mangel på tid og dermed ble det nødvendig å prioritere det som er viktigst. I lys av at oppgavens formål var å lage en prototype og å demonstrere muligheter i egen programvare for SOI 1.4 så vi det som viktigere å implementere moduler og funksjonalitet som var unikt for dette. Den store ulempen er at om en modul blir lagt til ved en feil er det ikke reverserbart og SOI-en må potensielt lages på nytt for å oppnå ønsket resultat.

Fjerning av moduler kan implementeres ved å først fjerne modulen fra SOI, deretter bruke tilsvarende kode som brukes til å la resten av programvaren vite om ny modul 6.3 til å la resten av programvaren vite om fjerningen, og til slutt kjøre automatisk modulplassering for å "fylle hullet" som den fjernede modulen etterlater seg. Fjerning kunne blitt koblet til en hurtigtast, for eksempel "CTRL + DELETE".

8.11.2 Manuell plassering av moduler

Manuell plassering av moduler støttes ikke i programvaren. Dette var ikke en nødvendighet og det ble prioritert å fokusere på automatisk plassering av moduler siden dette var den største nyvinningen og ville demonstrere muligheter og potensielle fordeler med egen programvare for SOI-behandling. Manuell plassering

kunne blitt implementert ved "drag-and-drop", som vist i Qts eksempel "Drag and Drop Robot Example"³⁵

8.11.3 Automatisk modulplassering med økt brukerinnflytelse

Som utdypet i seksjon 6.6 var det ønske om automatisk plassering i programvaren, og dette ble løst ved å se på problemet som "Two-dimensional bin packing" [6] og ta i bruk biblioteket "rectpack" [7]. Vi vurderte også å implementere vår egen algoritme, men nedprioriterte dette da rectpack allerede tilbyr en stor mengde algoritmer. En annen strategi kunne vært å modifisere en eksisterende algoritme fra rectpack, og dette ble i noen grad utforsket. Dette utdypes i seksjon 8.11.3.

Resultatet av automatisk plassering via rectpack gir optimale resultater målt i plassutnyttelse. Om man derimot også ønsker plasseringer som er optimale i brukerens subjektive øyne, blir problemet vanskeligere da rectpacks algoritmer ikke støtter dette. I implementasjonen forsøker vi å forbedre situasjonen over ved å la bruker justere innstillinger til ønsket plassering er oppnådd, og ved å la bruker selv bestemme rekkefølgen moduler plasseres i. Innstillingsmuligheter er beskrevet i seksjon 6.6.1. Sistnevnte gjør at bruker kan garantere at en valgt modul alltid kommer først, og vil gjøre det meget sannsynlig at påfølgende moduler vil plasseres nært denne første modulen.

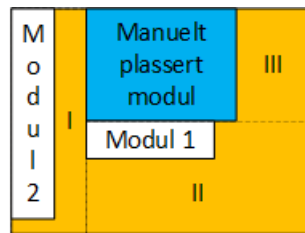
Vi kom på en rekke hypotetiske løsninger for å gi bruker optimal innflytelse på automatisk plassering.

Forbedring: Modifisere rectpack-algoritme

En løsning kunne vært å la bruker plassere et utvalg SOI-moduler fritt i SOI, og modifisere en rectpack-algoritme til å respektere disse manuelt plasserte modulene. Det er tenkelig at dette kunne blitt implementert ved å modifisere "Guillotine"-algoritmen. En full utdyping av Guillotine-algoritmen passer ikke her, så vi velger heller å referere til forklaringen i "A Thousand Ways to Pack the Bin - A Practical Approach to Two-Dimensional Rectangle Bin Packing" [6]. Med bakgrunn i forklaringen gitt der vil vi legge frem et forslag til en modifisert utgave.

Algoritmen går allerede ut på å dele sider, "bins", opp i mindre sider, så om vi kjører vår egen oppdeling for å fjerne manuelt plasserte moduler før algoritmen starter kan vi ekskludere plass brukt av disse modulene fra automatisk plassering. Deretter kan moduler som ikke manuelt plasseres gis til algoritmen som vanlig, og vi vil få optimal pakking av disse som respekterer de manuelt plasserte modulene. Figur 8.3 viser dette. Den manuelt plasserte modulen er tegnet i blått, SOI-siden i oransje og automatisk plasserte moduler i hvitt. De stiplede linjene viser oppdelingene som blir gjort for å lage nye sider "I", "II" og "III". I, II og III gis til Guillotine-algoritmen sammen med modul 1 og 2, og resulterer i plasseringene som vises. Poenget er at I, II og III håndteres av algoritmen som vanlige SOI-sider.

³⁵<https://doc.qt.io/qt-5/qtwidgets-graphicsview-dragdroprobot-example.html>, aksisert 15.05.2020.



Figur 8.3: Illustrasjon av modifisering av Guillotine-algoritmen for å låse posisjon til en gitt modul

Guillotine-algoritmen er en av de dårligere algoritmene ifølge "A Thousand Ways to Pack the Bin - A Practical Approach to Two-Dimensional Rectangle Bin Packing" [6], så en bedre løsning hadde vært en implementasjon som baserte seg på en annen algoritme.

Forbedring: Delvis sortering av SOI-moduler

I implementert programvare kan brukeren enten velge å sortere alle modulene selv, eller å velge en sorteringsalgoritme for å gjøre det. Ved å velge sorteringen selv kan bruker oppnå plasseringer som ellers ikke er mulig, men til gjengeld vil ikke programvaren hjelpe til med sortering i det hele tatt. Om bruker bare vil manuelt sortere et utvalg moduler, ville det vært en fordel om programvaren kunne håndtere automatisk sortering av resten. Med vår bruk av rectpack er dette trivielt å implementere, men vi tok oss ikke tid til det.

Forbedring: Sammenslåing av moduler

Det er tenkelig at bruker vil at to moduler skal bli plassert nært hverandre i SOI, for eksempel autentiseringstavle 1.10.1 og subtraktorkoder 1.10.7. Dette kan implementeres ved å "lime" modulene sammen til ett rektangel, og oppgi dette rektangelet til rectpack for automatisk plassering sammen med resten av modulene. Når rectpack oppgir optimal plassering kan de sammensveisede modulene plasseres med dette rektangelet som referansepunkt. En slik funksjonalitet kunne også vært nyttig for å konstruere "nye" moduler bygd opp av mindre moduler, for eksempel ved å knytte en fritekstmodul 1.10.6 til en kodefrasemodul 1.10.4 for å supplere modulen ved å oppgi eksempel på bruk.

8.11.4 Hindre bruk av like kodeord

Like kodeord på samme SOI er svært uheldig fordi innholdet i en kodet melding blir tvetydig, og konsekvensen i verste fall fatal. Det hadde derfor vært en stor forbedring dersom programvaren ikke tillater dette. Like kodeord kan forekomme når modulene forhåndsavtalte koder 1.10.2 og kodefraser 1.10.4 brukes på samme SOI. Nærmere bestemt kan varslingsordet til den førstnevnte bli lik et av

kodeordene til den sistnevnte, fordi begge bruker et tilfeldig ord fra databasetabellen "CategoryWords" 3.3.

En løsning, som også vil fungere med fremtidige moduler, er at kodeord som tas i bruk legges til i en liste på SOI-nivå. Hver enkelt modul må selv sørge for at kodeord den skal bruke ikke finnes i listen. Listen må også være med i SOI på fil.

8.11.5 Evne til å oppdage fonetisk like kodeord

Misforståelser kan oppstå ved kommunikasjon over samband. Dersom programvaren får evnen til å oppdage fonetisk like³⁶ kodeord, kan forekomsten av kodeordmisforståelser reduseres.

For å kunne lete etter fonetisk like kodeord, må SOI-klassen ha en oversikt over alle kodeord som er i bruk. Det må også utvikles en algoritme som foretar sjekk av fonetisk likhet på tvers av ordene. Hvordan algoritmen kan løse dette utdypes ikke her da gruppen ikke har kompetanse innen fonetikk og fordi vi antar at dette vil være omfattende.

8.11.6 Søkefunksjonalitet i kodebok

Kodebok-grensesnittet, vist i figur 4.5, har ingen søkefelt. Søkefunksjonalitet vil forbedre brukeropplevelsen ved at det blir enklere å finne frem til konkrete ord, uttrykk, kategorier og koder. Det er verdt å nevne at brukeren kan klikke på en kolonne og trykke på en tast for å hoppe til neste ord som har tastens bokstav som forbokstav, men dette er ikke særlig brukervennlig.

8.11.7 Endre papirretning

Gruppen tok et valg om å først implementere landskaps-papirretning. Fra eksemplene vi fikk utdelt, vedlegg J, kan man se at portrett og landskap er brukt halvt om halvt. I oppsettmenyen, figur 4.10, kan man se at det er klargjort og tiltenkt et valg for portrett papirretning.

For å støtte begge papirretningene må en del kode rundt plassering av SOI og organisering av toppsegment endres, da dette nå er hardkodet til landskap som papirretning. Arbeidet som kreves er ikke teknisk vanskelig, men tidkrevende.

Om programvaren hadde hatt støtte for begge papirretningene, er én relatert utbedring at papirretning kunne blitt en faktor ved optimal modulplassering. Som eksempel kan det tenkes at en SOI blir mer optimalt pakket om første side er i portrett og andre side er i landskap.

8.11.8 Valg av emblem i toppsegmentet

Det kan være seg at Heimevernets emblem ikke er passende for en gitt situasjon eller at det finnes et annet emblem som er mer passende. Mulighet for å endre

³⁶Ord, eller deler av ord, som er fonetisk like høres like ut når de uttales.

emblem i toppsegmentet støttes ikke, nå er Heimevernets emblem fastsatt. Med en tilsvarende løsning som ved de andre attributtene i toppsegmentet ville det vært fullt mulig å å via oppsettdialogen, figur 4.10, gi brukeren mulighet til å laste inn eget bilde. Dette ville for eksempel vært gunstig om programvaren skal distribueres og brukes i Forsvaret utover Heimevernet.

8.11.9 Legge til kjente snarveier for redigering

Programmet støtter nå ikke kjente snarveier for redigering av tekst, eksempelvis "CTRL + C" for kopiering av markert tekst. Da dette ikke er en nyvinning er det ikke prioritet. Støtte for disse snarveien vil derimot øke brukervennligheten til programvaren.

8.11.10 Farger på moduler

For å øke leselighet av SOI kan det brukes farger i moduler for å eksempelvis markere viktige seksjoner. Dette er ikke prioritert da det har vært et fokus på at SOI skal være leselig i svart/hvitt på papir, vedlegg H.

8.11.11 Benytte posisjon til QGraphicsWidgetProxy direkte

I nåværende implementasjon posisjoneres SOI-moduler i "QGraphicsScene" ved å sette posisjon på selve modulen. En forbedring ville vært å sette posisjon på tilhørende "QGraphicsProxyWidget" istedenfor. Fra et programvarearkitektur-perspektiv er dette bedre, da SOI-klassen ikke ville behøvd å vite om QGraphicsScene i det hele tatt, noe den gjør med nåværende implementasjon. Se seksjon 6.4 for en forklaring på bruk av QGraphicsScene, og funksjonen *SOI.update_module_widget_position* i modulen *soitool.soi*.

8.11.12 Undersøke nødvendighet av ABC og metaklasse i modul-arkitektur

Mot slutten av prosjektperioden undersøkte gruppen metakonflikt-problemet som oppstod ved arving fra flere klasser samtidig, diskutert i seksjon 8.2. Dette ble undersøkt nærmere fordi vi mente at det ikke bør være noe problem med arv av denne typen. Vi forsøkte å droppe ModuleBase-arving fra ABC og fjernet metaklasse-arving fra modulene, og det ser ut til at dette fungerer likevel. Programvaren leveres ikke med endringene fordi det ikke var nok tid til nøye gjennomgang og kvalitetssikring.

8.11.13 Gjøre komprimering lønnsom

Grunnet en systematisk feil der lengden av komprimerte og ukomprimerte data ble sammenlignet før de komprimerte datane ble typekonvertert til streng, ble ikke heftelsen i seksjon 8.7 oppdaget før ved slutten av prosjektet. Vi trodde lenge

at komprimeringen, inkludert typekonvertering, var langt bedre enn den faktisk var. I lys av den systematiske feilen som ble gjort, kan løsningen eksempelvis utbedres ved å advare brukeren med en dialog i tilfeller hvor det ikke er lønnsomt å komprimere SOI. Det kan også tenkes at bruk av en algoritme som komprimerer til streng er et godt alternativ slik at typekonvertering unngås. Dette ble det dessverre ikke tid til da feilen diskutert i denne seksjonen ble oppdaget avslutningsvis i prosjektperioden.

8.12 Kjente programvarefeil

I denne seksjonen blir programvarefeil, eller ”bugs”, diskutert. Dette vil spare oppdragsgiver som skal ta over prosjektet og muligens videreutvikle programvaren for tid og samtidig gi et innsyn i programvaren og mulige ”feller” å gå i.

8.12.1 Formatering av PDF produsert av QPrinter

For å produsere PDF av SOI brukes ”QPrinter”, som beskrevet i seksjon 6.8. Dette var lettvtint å implementere, men ser ut til å ha to ulemper. I begge tilfeller er overgang fra QPrinter til ”QPdfWriter” én mulig løsning, og bruk av ”ReportLab” [8] som gjort til kodebok-PDF en annen løsning. Alternativer for PDF-behandling drøftes i seksjon 8.1.6.

PDF-filer blir større enn forventet

For å observere at PDF-filer blir større enn forventet, kan man åpne programvaren med utgangsverdier i kodebok og eksportere tom SOI til PDF, i tillegg til å eksportere stor kodebok. SOI-PDF 6.8 vil få en filstørrelse på over 4 MB, mens kodebok-PDF 6.11.2 bare vil være på rundt 80 kB. Dette er spesielt merkelig fordi SOI-PDF bare er på én side, mens kodebok-PDF er på 18 sider. Om filstørrelse på SOI PDF blir et problem kan man stille ned oppløsningen fra standard 500 dpi under eksport, men det vil påvirke lesbarheten.

PDF-filer ser ikke ut til å være formattert riktig

For å demonstrere at PDF-filer ikke ser ut til å være formattert riktig, kan man eksportere SOI med én autentiseringstavle 2.1.2 til PDF, og åpne denne i et program for PDF-visning. Tekst vil ikke kunne markeres i PDF, og om man dobbeltklikker på SOI-elementer vil PDF-visningsprogrammet oppføre seg som om PDF-en er bygget opp av bilder. Dette indikerer at QPrinter i realiteten bare tegner inn bilder av widgets, som er skuffende. Det var ønskelig, og forventet, at QPrinter skulle tegne tekst skikkelig. QPdfWriter, diskutert i seksjon 8.1.6, er et alternativ til QPrinter som muligens håndterer PDF bedre. Til sammenligning kan man åpne kodebok-PDF og observere at tekst kan markeres og kopieres, slik man forventer av en PDF.

8.12.2 Mulig å trykke på SOI-side gjennom modul

Én måte å utløse automatisk plassering av moduler på er å trykke på selve SOI-siden. Dette kjører "reorganisering" på lik linje som om bruker skulle trykket på reorganiseringsknappen i nedre venstre hjørne. Noe vi oppdaget sent er at det i noen tilfeller er mulig å trykke på SOI-siden *gjennom* en SOI-modul, noe som ikke er meningen. Området brukeren kan trykke gjennom er det grå området rundt en QLabel, for eksempel i modulene frekvenstabell eller telefonliste. Som beskrevet i seksjon 6.5.3 har disse to modulene menyer som vises når bruker holder musepekeren over dem, og dette fører til økt størrelse på modulene. Den økte størrelsen skaper problemer om bruker trykker *gjennom* modulene, da reorganisering vil skje med størrelsen på menyen medregnet, noe som produserer en feilaktig plassering. Menyen vil forbli synlig helt til musen beveges igjen. Når modulen oppfatter at musepekeren ikke lenger er i dens område vil menyen skjules og området der menyen var vil stå åpent til neste reorganisering utføres. En mulig løsning på halve problemet kan være å endre modulenes `get_size()`-funksjon slik at om menyen er synlig vil det trekkes fra høyden.

8.12.3 Frekvenstabellens overskrifter forsvinner

Se 6.5.3 for utdypelse av implementasjon av frekvenstabell. Om en overskrift på første rad har én eller flere underoverskrifter som til sammen er smalere enn overskriften på første rad, vil denne delvis eller helt forvinne og komplementeres med tre punktum. Unntaket er om en av underoverskriftene tilhører samme logiske kolonne som overskriften ligger i, da kan underoverskriften være kortere enn overskriften uten å påvirke den. Feilen kommer av at om ikke underoverskriften som ligger i samme logiske kolonne som overskriften ligger i er synlig, vil hele den logiske kolonnen være usynlig og overskriften vises i den virtuelle kolonnen. Når modulen skal endre størrelse regner den bare med synlige logiske kolonner og tar dermed bare høyde for underoverskriftene, den virtuelle kolonnen vil altså få lik bredde som summen av bredden til underoverskriftene. Innholdet i kolonner er førende for bredden og kan bidra til å utligne feilen om den har oppstått.

Modulen bruker funksjonalitet fra modulbasen 6.5.1 for å endre størrelse på tabellen. For å løse feilen må det lages en egen funksjon for størrelsesendring av tabellen i frekvenstabellmodulen som tar høyde for bredden av overskriftene i første rad. Dette har hatt lav prioritet da gruppen ser det som lite sannsynlig at feilen vil bli reproduert i tiltenkt faktisk bruk. Radionavn er ofte korte og trenger flere attributter samt at underoverskriften med lik logisk kolonne som overskriften vil typisk stå sentralt.

8.12.4 Menyer i Frekvenstabell og Telefonliste kan havne bak andre moduler

Modulene frekvenstabell og telefonliste har som beskrevet i seksjon 6.5.3 menyer som vises når bruker flytter musepekeren over dem. Et problem med nåværende

implementasjon er at denne menyen potensielt kan plasseres bak andre moduler, avhengig av hvor på z-aksen modulene befinner seg. z-aksen bestemmer rekkefølgen QGraphicsScene tegner sine QGraphicsItems. Om to widgets overlapper, vil widget med høyest z-verdi tegnes sist, og dermed plasseres foran. Om frekvenstabell eller telefonliste har lavere posisjon på z-aksen enn en annen modul som er plassert høyere på y-aksen, og begge ligger likt på x-aksen, vil menyen ikke vises skikkelig fordi plasseres *bak* den andre modulen.

8.12.5 Last inn SOI fra fil på Linux

Denne feilen har svært lav betydning da programvaren er laget for Windows, 2.4.2. Ved bruk av plattformspesifikk fildialog på Linux brukes "GTK-vinduer"³⁷. GTK-vinduer ønsker GTK-vinduer som sine foreldre, dette skjer ikke da "QFileDialog" tar en QWidget som forelder, dette fører til koden returnerer NULL/0 og programmet fryser³⁸. En enkel mulig løsning på dette er for Linux å bruke fildialogsfunksjonaliteten til Tkinter³⁹ som ikke bruker plattformspesifikt fildialog-vindu og dermed unngår GTK-vinduer. Denne programvarefeilen er ikke reparert grunnet svært lav prioritet.

³⁷<https://www.linux.org/threads/understanding-gtk.10429/>, aksessert 14.05.2020.

³⁸<https://forum.qt.io/topic/85997/gtk-warning-for-native-qfiledialog-under-linux/2>, aksessert 14.05.2020.

³⁹<https://wiki.python.org/moin/TkInter>, aksessert 14.05.2020.

Kapittel 9

Konklusjon

Dette kapittelet avslutter rapporten med konkluderende tekster om arbeidet som er utført. Oppgaven blir evaluert sammen med gruppens evne til å løse den, og mulig videre arbeid presenteres.

9.1 Kritikk av oppgaven

Oppgaven kunne hatt en klarere oppgavebeskrivelse. Oppdragsgiver har nevnt at oppgaven krever så mye innføring at det er vanskelig å uttrykke alt på papir. En løsning på dette kunne vært å lage en ytterligere oppgavebeskrivelse som tar utgangspunkt i at leseren har innsyn i problemstillingen og fagfeltet. Oppgavebeskrivelsen har vært avhengig av møter med kontaktperson hvor gruppen har måttet stille spørsmål og skrive gode referater. Samtidig skal det nevnes at kontaktperson har prioritert møter med gruppen og delt mye av sin kunnskap.

Spesielt rundt modulen sambandsdiagram 1.10.5 savner gruppen klarhet. Skillet mellom utvikling av verktøy for å lage sambandsdiagram og utvikling av modul for å vise diagrammet, der førstnevnte var ønsket og sistnevnte var et krav, 2.1.2, er et eksempel på dette. Manglende klarhet bidro til at sambandsdiagram ikke ble implementert.

Gruppen har ikke sett utviklingen av kodebok 1.11 og tilhørende funksjonalitet som svært relevant til oppgavens formål 1.4. Den eneste sammenhengen gruppen ser kodeboken har med selve SOI-dokumentet er at liten kodebok 2.2 benyttes i forhåndsavtalte koder 1.10. Om kodebok ikke hadde vært en del av oppgaven kunne gruppen hatt fullt fokus på SOI-dokumentet og sannsynligvis utført denne delen av oppgaven på en bedre måte ved å blant annet implementere et utvalg av mangler 8.10 og utbedringer 8.11.

Tidlig i prosjektet ble det sagt at kontaktperson skulle instruere personell fra Heimevernet til å lage enkle use cases på hva de ønsket av applikasjonen. Dette ble aldri gjort og kan ha påvirket oppgaven i form av at vi som gruppe ikke har klart å innfri oppdragsgivers visjon. Gjennom kontaktperson har vi forsøkt så godt som mulig å imøtekomme det oppdragsgiver har ønsket seg.

9.2 Videre arbeid

Utover mindre stykker arbeid som utvikling av mangler og utbedringer som beskrevet i henholdsvis seksjon 8.10 og 8.11, samt reparasjon av feil som beskrevet i seksjon 8.12 er det særlig to oppgaver som kan ha potensiale til å bli egne bacheloroppgaver. Disse er beskrevet i seksjonene under.

9.2.1 Synkronisering av data

Programvaren utviklet i denne oppgaven er et lokalt verktøy. For å utnytte potensialet til en egen programvare for SOI kan det utvikles funksjonalitet for synkronisering av data i et nettverk. Nettverk som blir brukt er hierarkiske og har en ”tre-struktur”. Det kan være nyttig å dele data til underliggende, sideliggende og overliggende enheter i nettverket. Felles er at to enheter som skal dele data nødvendigvis ikke er direkte koblet og må bruke reléer og mellomliggende enheter. Samtidig har ikke alle enheter bruk for hverandres data. Data kan også bli delt fra to grener uten direkte kobling samtidig, det må da sørges for at data sys sammen på en hensiktsmessig måte. Synkronisering av data som bygger på den utviklede programvaren drøftes i seksjon 8.8.

9.2.2 Verktøy for å lage sambandsdiagram

I dette verktøyet skal det være mulig å skissere opp hindringer og enheters plasseringer. Ut i fra dette vil applikasjonen generere et sambandsdiagram, 1.10.5.

I tillegg kan verktøyet utvides med følgende funksjonalitet

- Tegne enheter direkte inn på kart og genere sambandsdiagram med mulige forbindelser der det tas høyde for hindringer i terrenget.
- Mulighet for å gi en data-input i form av en tabell eller lignende som det direkte kan genereres et diagram ut av.
- Et tegneverktøy for å enkelt tegne sambandsdiagrammer.

Ved endringer skal det være enkelt og raskt å generere et oppdatert diagram. Verktøyet kan integreres i programvaren utviklet i denne oppgaven.

9.3 Evaluering av gruppens arbeid

I følgende seksjoner evaluerer vi gruppens arbeid. Både positive og negative sider drøftes.

9.3.1 Hva vi er fornøyde med

På grunn av at gruppen som helhet ikke hadde samarbeidet på prosjekter tidligere var det i begynnelsen noe usikkerhet rundt hvordan gruppedynamikken ville bli,

men vi ble alle positivt overrasket. Tekniske diskusjoner har vært seriøse, uenigheter har blitt løst på en profesjonell måte, og møter har vært effektive. Gruppen har samarbeidet godt, og opplevelsen har vært behagelig.

Gruppemedlemmene har forskjellige interesseområder, kunnskaper og kvaliteter. Dette var viktig å utnytte for at gruppen skulle yte best mulig. Eksempelvis er noen spesielt interessert i utforming av brukergrensesnitt mens andre har gode kunnskaper om "DevOps" og automatisering av tester. Det faller da naturlig at de tar ansvar for disse områdene. Noen liker å diskutere og jobbe sammen mens andre konsentrerer seg best alene. Gruppen har funnet en god balanse der medlemmene både får utnyttet sine kvaliteter og har blitt utfordret. Samtidig har kunnskap blitt delt i gruppen på en slik måte at hver enkelt har økt sin kompetanse og bredde i fagfeltet.

Vi synes vi har vært særdeles flinke med "Merge Request"-funksjonaliteten i GitLab. Dette har hjulpet under "Code Review" ved at hver endring har fått saklig og detaljert gjennomgang, og ved at hele gruppen har fått innsyn i endringer på kildekoden. Denne bruken av Merge Requests gjør at endringer krever mer tid, men øker kvaliteten på resultatet og unngår fremtidig refaktoring, slik at vi sparer tid totalt sett. Dette viste seg å være ekstra nyttig siden all kommunikasjon ble digital grunnet covid-19.

9.3.2 Hva vi ville gjort annerledes

Det er noen punkter gruppen ville ha gjort annerledes dersom prosjektet skulle gjentas. Vi burde bedt kontaktperson om å få alle kravspesifikasjons-detaller ved prosjektstart, istedenfor å "fiske" ut deler av denne gjennom spørsmål som dukket opp underveis i utviklingsperioden. Vi ville valgt en mer egnet arbeidschat separert fra sosiale medier, fordi det til tider var vanskelig å oppdatere seg på prosjektstatus gjennom den brukte chatten. Gode kandidater er "Mattermost" og "Slack". For å spare tid ved gjennomgang av rapporten burde vi ha definert stilguider og standard formatering før vi begynte å skrive på rapporten.

Et annet aspekt av vår tilnærming til oppgaven vi kunne gjort bedre er å være mindre "medgjørige" under møter med kontaktperson. Det har desverre hendt at gruppen har unngått å stille viktige utdypende spørsmål under møter, med begrunnelser som at "kontaktperson utdyper sikkert senere" eller at "resten av gruppen forstår det sikkert".

En mer konkret utdypelse av fremdriften i prosjektet finnes i seksjon 8.9

9.4 Drøfting av resultater

Vi mener vi har løst oppgaven på en tilstrekkelig måte der vi har oppfylt oppgavens formål 1.4 ved å lage en prototype på programvare for SOI-behandling.

9.4.1 Effektmål

Selv om løsningen på oppgaven har forbedringspotensiale som diskutert i 8.10 og 8.11, når den effektmålene 1.4.1 godt. Ved autogenerering av innhold i moduler og en rekke tastaturnarveier er det tilrettelagt for produksjon av SOI-er i et raskere tempo. Koder i moduler blir tilfeldig generert 6.10, noe som hindrer menneskelig svikt der koder blir gjenbrukt. Oppgaven demonstrerer også nye muligheter egen programvare gir, som eksempelvis automatisk plassering av moduler 6.6, kompakt filformat 6.7 og automatisk generering av modulinnhold.

9.4.2 Læringsmål

Gjennom oppgaven har gruppemedlemmene tilegnet seg erfaring og kunnskaper innenfor både programmeringsspråket Python og brukergrensesnittbiblioteket Qt, samt ytterligere erfaring med SQL-databaser. Gjennom kontaktperson har gruppe-medlemmene lært seg hvordan moduler som blant annet subtraktorkoder 1.10.7 og forhåndsavtalte koder 1.10.2 er bygget opp og brukt for kryptografi. Gruppe-medlemmene har i varierende grad benyttet kunnskaper fra de fleste datatekniske emner de har gjennomført, blant disse utmerker eksempelvis "IMT2243 - Systemutvikling"¹, "IMT3281 - Applikasjonsutvikling"² og "IMT2571 - Datamodellering og databasesystemer"³ seg. Oppgavens størrelse har gjort det nødvendig med systematisk planlegging og metodisk arbeid, samt god dialog med kontaktperson. Å jobbe med en oppdragsgiver er nytt for gruppen og spesielt her er det høstet nye erfaringer. Gruppemedlemmene har med andre ord nådd læringsmålene 1.4.2.

9.4.3 Resultatmål

Oppgaven oppfyller kravspesifikasjonen i kapittel 2 med unntak av to mangler beskrevet i seksjon 8.10. Gruppen ønsker å trekke frem at SOI-en i vedlegg K produsert av vår programvare er 4,97 kB i ukomprimert serialisert utgave. Dette er godt innenfor kravet 2.4.2 på maksimalt 90 kB for overføring under fem minutter, og i tillegg innenfor ønsket om maksimalt 18 kB for overføring under ett minutt.

9.5 Konklusjon

Gruppen har på oppdrag fra Opplandske Heimevernsdistrikt 05, G6-seksjonen utviklet en prototype på programvare for SOI, "Samband- og operasjonsinstruks". Programvaren har et fokus på å effektivisere produksjon og behandling av SOI og setter brukervennlighet sentralt. Den er modulær og støtter utvikling av nye moduler. Smidig utviklingsmetodikk er anvendt og programvaren er testet med automatiserte tester. Resultatet er en tilnærmet komplett prototype i henhold til kravspesifikasjon med bare få mangler.

¹<https://www.ntnu.no/studier/emner/IMT2243/2016#tab=omEmnet>, aksessert 19.05.2020.

²<https://www.ntnu.no/studier/emner/IMT3281#tab=omEmnet>, aksessert 16.05.2020.

³<https://www.ntnu.no/studier/emner/IMT2571#tab=omEmnet>, aksessert 16.05.2020.

Bibliografi

- [1] A. Z. M. Ali, R. Wahid, K. Samsudin og M. Z. Idris, *Reading on the Computer Screen: Does Font Type has Effects on Web Text Readability?*, [På internett; besøkt 18-05-2020], jan. 2013. adresse: <https://files.eric.ed.gov/fulltext/EJ1067757.pdf>.
- [2] G. K. Connolly, *Legibility and readability of small print: effects of font, observer age and spatial vision*, [På internett; besøkt 18-05-2020], feb. 1998. adresse: <https://prism.ucalgary.ca/bitstream/handle/1880/26040/31338Connolly.pdf?sequence=1>.
- [3] Qt, *PySide2*, [På internett; besøkt 13-02-2020]. adresse: https://wiki.qt.io/Qt_for_Python.
- [4] *SQLite3*, [På internett; besøkt 18-05-2020]. adresse: <https://docs.python.org/3/library/sqlite3.html>.
- [5] Qt, [På internett; besøkt 24-02-2020]. adresse: <https://doc.qt.io/qt-5/index.html>.
- [6] J. Jylang, *A Thousand Ways to Pack the Bin - A Practical Approach to Two-Dimensional Rectangle Bin Packing*, [På internett; besøkt 18-05-2020], feb. 2010. adresse: <http://pds25.egloos.com/pds/201504/21/98/RectangleBinPack.pdf>.
- [7] *rectpack*, [På internett; besøkt 14-05-2020]. adresse: <https://github.com/secnot/rectpack>.
- [8] *ReportLab*, [På internett; besøkt 17-05-2020]. adresse: <https://www.reportlab.com/>.
- [9] *QTest*, [På internett; besøkt 13-02-2020]. adresse: <https://doc.qt.io/qtforpython/PySide2/QtTest/QtTest.html>.
- [10] B. Smith, *A Quick Guide to GPLv3*, [På internett; besøkt 13-02-2020]. adresse: <https://www.gnu.org/licenses/quick-guide-gplv3.html>.
- [11] F. S. Foundation, *A Quick Guide to LGPLv3*, versjon 3, [På internett; besøkt 13-02-2020], jun. 2007. adresse: <https://www.gnu.org/licenses/lgpl-3.0.html>.
- [12] *Unittest*, [På internett; besøkt 13-02-2020]. adresse: <https://docs.python.org/3/library/unittest.html>.

Vedlegg A

Ordliste

Autentiseringskoder: Se 1.10.1.

Binding: En "binding" til et bibliotek er et grensesnitt mot biblioteket fra et annet språk, uten at biblioteket omskrives i språket.

Code Review: Gjennomgang av kode i plenum.

Dekoding: Omforme data fra ett format til et annet. Se "enkoding".

Decorator: En Python "decorator" beskriver en egenskap ved en funksjon. Decorators kan brukes for å tildele spesielle egenskaper til funksjoner.

Deserialisering: En programmeringsteknikk for å laste inn en datastruktur fra en tekst bestående av serialisert datastruktur.

Dialog: I denne rapporten brukt til å mene "pop-up"-vindu - Vinduer som legger seg fremfor programvaren og må lukkes av bruker før normal bruk kan gjenopptas. Dialoger som forhindrer bruker å fortsette normal bruk av programvaren før de lukkes kalles "modulære".

Docker: Virtualiseringsteknologi hvor underliggende operativsystem og ressurser deles. En "Docker container" kan ses på som et virtuelt miljø hvor programvare kan kjøres med alle sine avhengigheter, uavhengig av hvem som kjører containeren. En Docker container skapes fra et "Docker image", som kan ses på som en oppskrift på en container.

Docstring: En "docstring" er i Python en tekststreng som skrives som første "statement" i en modul, funksjon, klasse eller definisjon av klassefunksjon. En docstring blir til `__doc__`-attributtet på det objektet.¹

DPI: "Dots Per Inch". Beskriver hvor mange punkter som forekommer per tomme. Brukes som måleenhet ved utskrift.

¹<https://www.python.org/dev/peps/pep-0257/#id15>, aksessert 12.05.2020.

Embedded database: En "embedded" database er en database innebygget i programvare. Med andre ord sameksisterer programvaren og databasen, istedenfor at programvaren behandler databasen som et eksternt system.

Enkoding: Omforme data fra ett format til et annet. Se "dekoding".

Exception: En "exception" er en feilrapport som "kastes" av programvare ved uventede feil. Exceptions inneholder informasjon som kontekstualiserer feilen.

Feature-branches: Gren dedikert til et stykke arbeid i Git.

Forhåndsavtalte koder : Se 1.10.2.

Frekvenstabell: Se 1.10.3.

Interface: Klasser som implementerer et interface må selv implementere interfaces funksjoner.

Kbps: Kilobit per sekund.

Kodebok: En kodebok er innen kryptografi en bok som inneholder oppslagstabell for koding og dekoding. Se 1.11.

Kodefraser: Se 1.10.4.

Leksikografisk rekkefølge: En rekke tekster sortert i leksikografisk rekkefølge vil være sortert alfabetisk, bokstav for bokstav. For eksempel vil rekken "BA", "AB", "AZ" bli "AB", "AZ", "BA" i stigende leksikografisk rekkefølge.

Linting: Statisk anayse av kildekode for å avdekke feil.

Lo-fi prototype: En udetaljert prototype som ofte er på papir.

Merge Request: Forespørsel om å integrere en gren til hovedkildekoden.

MVP: "Minimal Viable Product" angir minste gyldige løsning på en ønsket funksjonalitet.

Model/View-grensesnitt: Grensesnitt der et "view" presenterer data i en "model".²

Pipeline: Ment her som verktøy for å utføre en handling ved en hendelse. Et eksempel er å teste kode ved push, eller sende melding ved ny "Merge Request".

Port: En "port" av et bibliotek er en omskriving av biblioteket i et annet språk.

Prototype: En prototype er et implementasjonsbevis.

²<https://doc.qt.io/qt-5/model-view-programming.html>, aksessert 19.05.2020.

Samband: Samband er kommunikasjon, og beskriver kommunikasjonsnettverket i en organisasjon.

Schema: Et schema er et dokument som beskriver struktur i data. Data som følger et schema oppfyller kravene i dokumentet. Programvaren bruker schema-pakken <https://pypi.org/project/schema/>.

Serialisering: En programmeringsteknikk for å representere en datastruktur som tekst.

Subtraktorkoder: Se 1.10.7.

Telefonliste: Se 1.10.8.

Toppsegment: Toppsegment er området i SOI hvor overskrift, underoverskrift, opprettelsesdato og lignende står. Se 2.1.1.

Unit test: Unit test brukes til å mene en automatisert programvaretest som tester én funksjonalitet. En samling Unit tester kan brukes for å teste en hel programvare.

US ASCII: US ASCII er et sett bokstaver med 128 tegn.³

Widget: I denne rapporten referer ”widget” til klasser, eller instanser av klasser, som arver fra ”QWidget”-klassen i Qt.

Wizard of Oz brukertest: En papirprototype gjøres ”klikkbar” ved at ark og papirutklipp byttes ut manuelt ved brukerinteraksjon.

WYSIWYG: ”What You See Is What You Get”: Betegnelse på redigering hvor en til en hver tid ser hvordan det ferdige dokumentet vil se ut. Microsoft Word er et godt eksempel på en WYSIWYG editor, hvor bruker slipper å redigere .docx filer direkte, som i praksis ville vært å redigere XML. \LaTeX er typisk et eksempel hvor redigering ikke nødvendigvis skjer gjennom WYSIWYG-editor.

³<http://www.columbia.edu/kermit/ascii.html>, aksessert 19.05.2020.

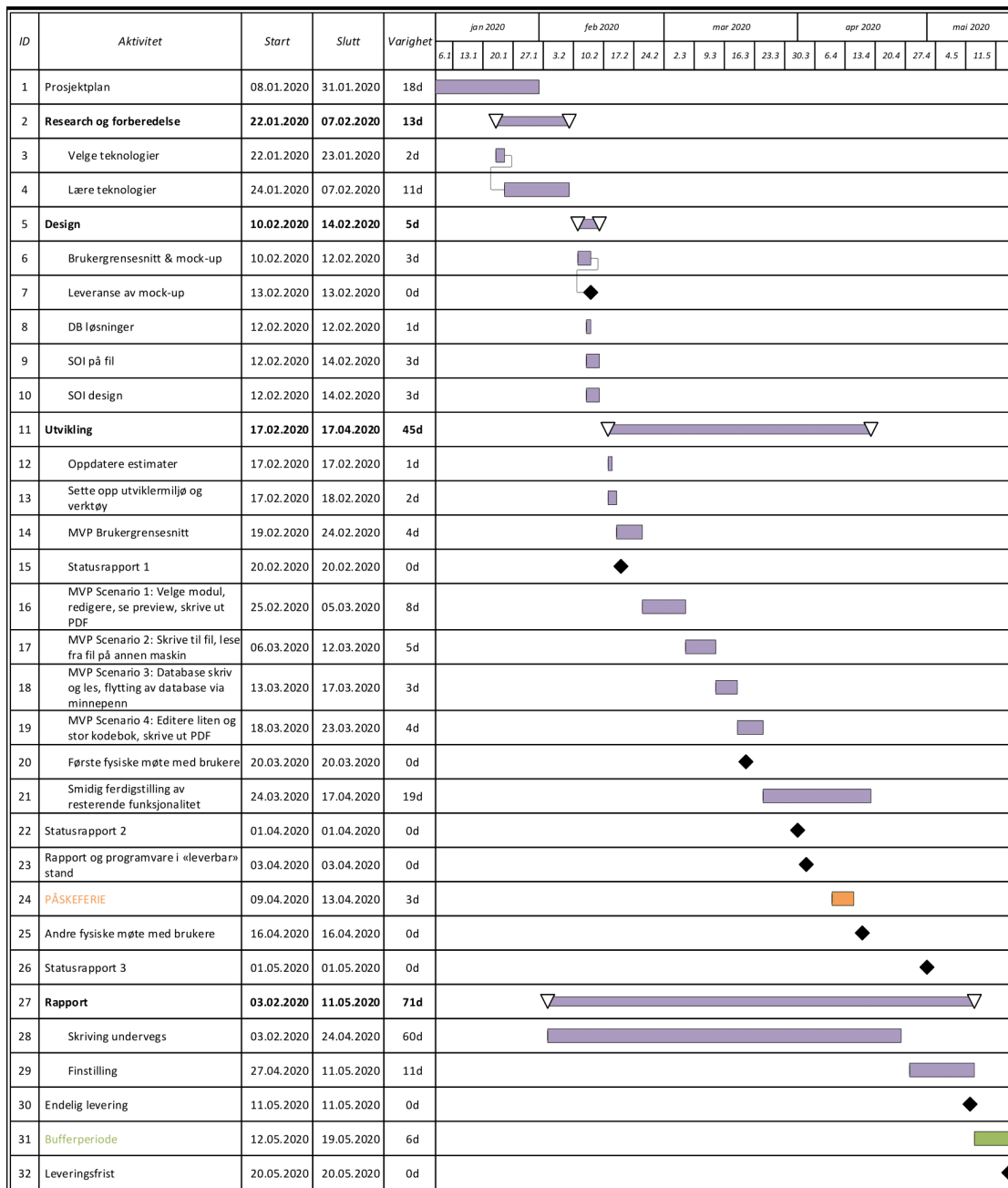
Vedlegg B

Gantt-diagram

Vedlegget inneholder Gantt-diagram for planlagt gjennomføring av bachelorprosjekt. Diagrammet kommer fra prosjektplanen i vedlegg E.

Kommentarer til Gantt-skjema fra prosjektplanen

- Aktivitetene er fargelagt etter følgende konvensjon:
 - **Lilla:** Arbeid.
 - **Oransje:** Ferie.
 - **Grønn:** Potensielt arbeid.
- Utviklingsfasen (id:11) vil i praksis bli utført i ”sykluser” på én uke.
- Første punkt i utviklingsfasen (id:11) er å utføre en ny estimering før vi begynner å skrive kode. Dette er for å få mer nøyaktige estimater på bakgrunn av utført research og design.
- Det er innlagt en bufferperiode i tilfelle noe uforutsett inntreffer (id:31).
- MVP står for Minimal Viable Product, altså den enklest mulige versjonen av applikasjonen som kan vises fram.
- Statusrapport er korte rapporter som skal leveres til veileder innen de gitte datoene. Disse skal vise veileder hvor langt prosjektet har kommet.



Figur B.1: Gantt-skjema fra prosjektplan

Vedlegg C

Papirprototype

Vedlegget inneholder papirprototypen brukt under brukertesting av programvaren. Beskrivelse av prototypen ligger også vedlagt.

Fi | kode | opsett | Htch

- [] X

SOI-num

NØD

+

SOI

A

B

F

D

G

E

I

H

Vedlegg

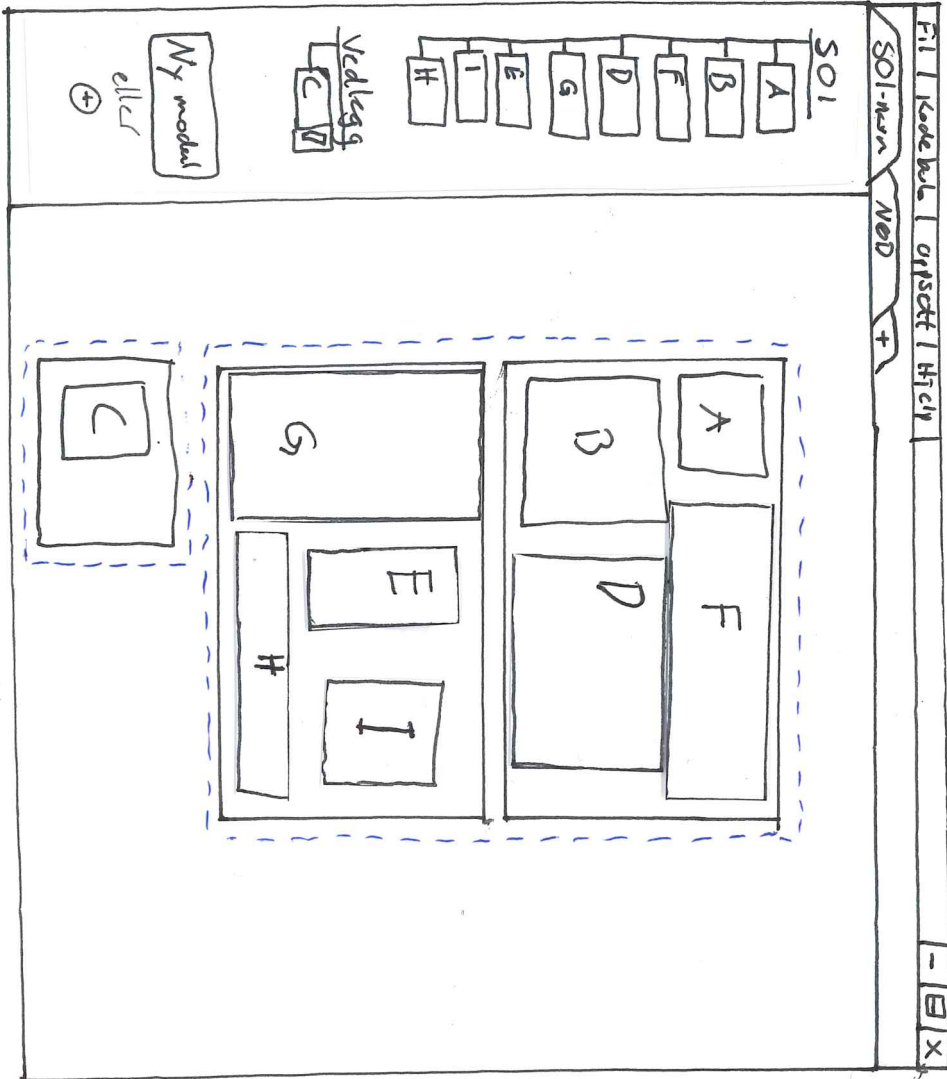
C

D

Ny model

eller

+





Åpne
Åpne fra DB
Forkindsvis
Lagre i DB
Eksporter

Stor kodebok
Liten kodebok

[illegible]

Innstillinger Oppsett

Headerdata

- ~~~~~
- ~~~~~
- ~~~~~
- ~~~~~

Papirretning

- □
- ▤

Plasseringsalgoritme

- ~~~~
- ~~~~
- ~~~~
- ~~~~

Modulplassering

- Automatisk
- Manuell

Ambryt Lagre



Hartigster
Enkel broek

Hurtigtaster

Her får man en
oversikt over hvilke
hurtigtaster man kan
bruke i applikasjonen

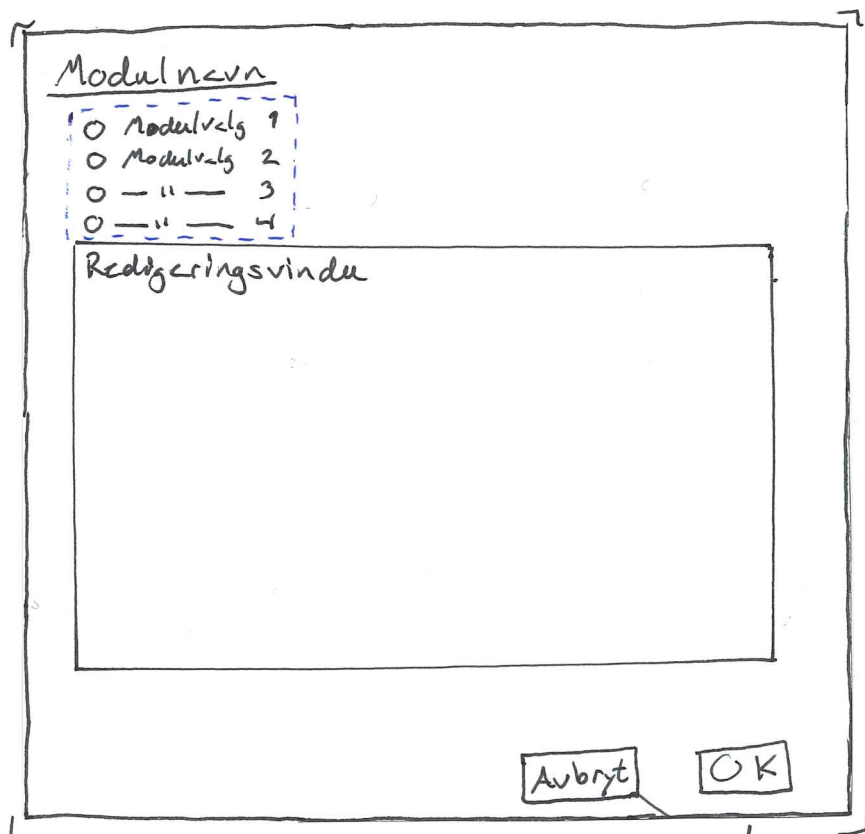
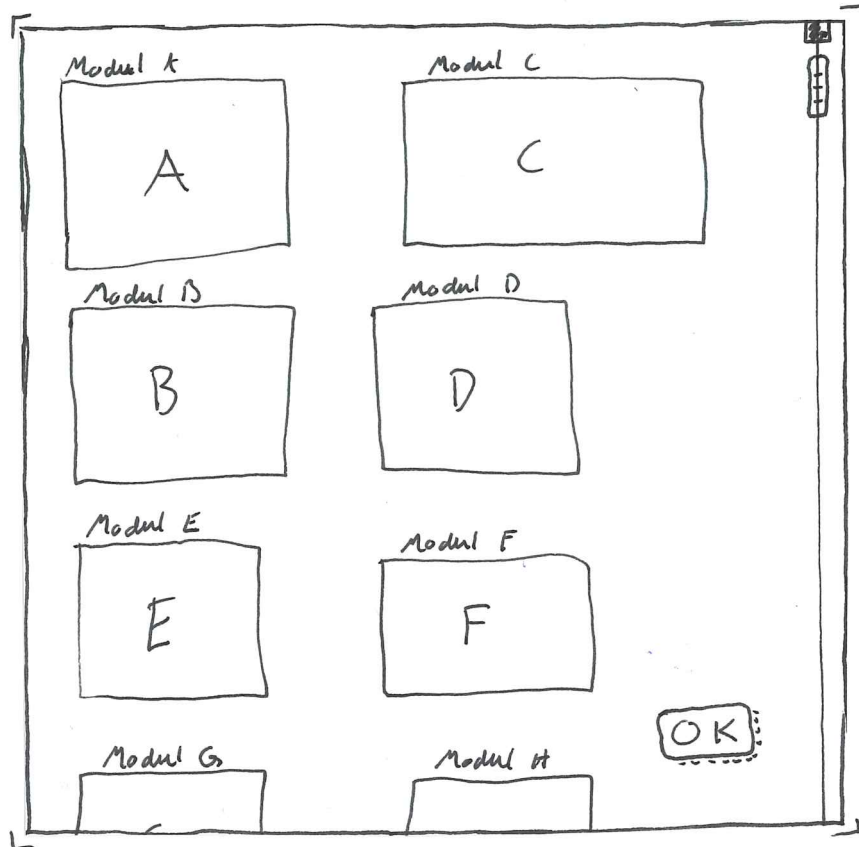
F.eks

$\boxed{\text{Ctrl}} + \boxed{M} \rightarrow \boxed{\text{My modul}}$

$\boxed{\text{Ctrl}} + \boxed{I} \rightarrow \boxed{\text{Oppsett}}$

Ny modul

eller



Det første bildet som ligger i pdf'en, er hovedskjermen for applikasjonen. Her har man en toppmeny med ulike valg, to liste for moduler til SOI, en knapp for å legge til nye moduler og et område som viser SOI mens man jobber med den. Videre i dette dokumentet skal disse forklares slik at det blir enklere å forstå hva vi tenker.

Vi starter først med toppmenyen. Her finner man 4 ulike knapper; fil, kodebok, oppsett og hjelp. Disse knappene vises på side 2 av pdf'en. Oppsett er den som må forklares mest grundig av menyvalgene. Trykker man på oppsett vil man få en popup som inneholder en del valg. Her kan man endre på headerdata, papirretning på SOI, hvilken algoritme som skal brukes til å plassere modulene mest effektivt og om man vil gjøre dette automatisk eller manuelt. I oppsett-popupen endrer man da på en del innstillinger for hvordan SOI'en vil se ut.

På venstre side av hovedvinduet har man to lister. En som heter SOI og en annen som heter vedlegg. I disse listene vil modulene som legges til havne i. Dette skjer slik at brukeren kan velge å flytte moduler opp/ned i listen og endre hvilke moduler som skal komme først eller sist. F.eks flytter man modul "B" over modul "A", vil "B" komme før "A" i SOI'en. Man kan se hvordan listene henger sammen ved å se på høyre side på hovedvinduet, på området hvor SOI'en vises.

Skal man legge til nye moduler i SOI'en, trykker man på knappen ny modul. Oversikten for denne knappen ligger på side 3 av pdf'en. Etter å ha trykket på knappen, kommer det opp en liste med alle modulene som kan brukes. Når man velger en modul kan det skje 2 forskjellige ting. De "enkle" modulene, som f.eks. tabeller, havner direkte inn i SOI'en i området til høyre i hovedvinduet. Deretter kan man redigere direkte i modulen. Velger man en litt mer "avansert" modul, f.eks. en modul som må gjøres litt mer med, havner man inn i et vindu hvor man kan redigere å endre på modulen før den settes inn i SOI'en.

Om man har en SOI allerede åpnet og man har lyst til å åpne en til, vil de havne som faner i applikasjonen. Om man tar og endrer oppsettet på en av SOI'ene, vil dette bare endre på den SOI'en man er aktiv på.

Vedlegg D

Script

Vedlegget inneholder script brukt under prosjektperioden som ikke er vist ellers i rapporten. Dette er script for undersøkelse av kodekvalitet, testing og konfigurering. Vedlegget inneholder også tekstfiler som ikke er hensiktsmessige å formatere til L^AT_EX for denne rapporten.

Kodeliste D.1: Powershell-script for sjekk av kodekvalitet

```
if ($args.Count -eq 0){
    $files=Get-ChildItem -recurse |
        Where-Object {$_ .name -match "[A-Za-z0-9_]+\ .py" -And $_.Name -notmatch
            ↪ ".pyc"} |
        % { $_.FullName }
    Write-Output "`nChecking all .py-files"
}
else{
    $files = @()

    for($i=0; $i -lt $args.Length; $i++){

        if($args[$i] -match "[A-Za-z0-9_]+\ .py"){
            $files += $args[$i]
        }
        else{
            $files += Get-ChildItem -recurse $args[$i] |
                Where-Object {$_ .Name -match "[A-Za-z0-9_]+\ .py" -And $_.Name -notmatch
                    ↪ ".pyc"} |
                % { $_.FullName }
        }
    }
}

for ($i=0; $i -lt $files.Length; $i++){
    Write-Output "`n=====$(Split-Path $files[$i] -leaf)
        ↪ ====="
    Write-Output "===PYLINT===`n"
    pylint --rcfile=./scripts/pylintrc $files[$i]
    Write-Output "`n===FLAKE8===`n"
    flake8 --config ./scripts/.flake8 $files[$i]
    Write-Output "`n===BANDIT===`n"
    bandit $files[$i]
```

```

Write-Output "'n===PYDOCSTYLE==='n"
pydocstyle.exe --convention=numpy $files[$i]
Write-Output "'n===BLACK==='n"
black -l 79 --check --diff $files[$i]
}

```

Kodeliste D.2: Steg for å konfigurere Ubuntu GitLab Runner

```

# Hva som ble gjort for å kunne teste GUI i Ubuntu VM

* Lagde VM via https://www.ntnu.no/wiki/display/skyhigh
* Ga floating IP
* SSH til VM
  * Så på <https://www.makeuseof.com/tag/ubuntu-remote-desktop-builtin-vnc-compatible-dead-easy/> og <https://community.spiceworks.com/how_to/155718-install-xrdp-on-ubuntu-18-04> og <https://github.com/naturalis/openstack-docs/wiki/Howto:-Adding-a-graphical-interface-to-your-Ubuntu-Openstack-instance>
  '''
  sudo apt update
  sudo apt dist-upgrade
  sudo apt-get install --no-install-recommends ubuntu-desktop
  sudo apt-get install firefox xrdp
  sudo passwd ubuntu # endre til noe vi husker
  sudo systemctl enable xrdp # kanskje unødvendig
  # installer alle dependencies... pside python pip osv osv
  # helst direkte fra requirements.txt filen i repo
  sudo reboot
  '''

* Koble til med Windows RDP klient (må åpne security group for RDP ingress)
* Åpne terminal og 'echo DISPLAY'
* SSH til VM
  * 'export DISPLAY' til verdien fått fra RDP
  * Kan nå kjøre Python Qt kode
  * installer 'gitlab-runner'
  * Følg samme steg for gitlab-runner som for Windows.

```

Kodeliste D.3: Steg for å konfigurere Windows GitLab Runner

```

# Hva som ble gjort for å få GitLab CI til å prate med Windows OpenStack VM for
  ↳ GUI-tester

* Fikk quota i SkyHigh
* Gjorde alt oppsett påkrevd for å få på plass en VM. Se
  https://www.ntnu.no/wiki/display/skyhigh
* Lagde 'Windows 10 Enterprise [Evaluation]' VM
* Fulgte https://www.ntnu.no/wiki/display/skyhigh/Windows for å nå tak i passord
  * MEN dette passordet fungerte ikke.. Endte heller opp med å logge inn som
    "Administrator", for så å havne i en prompt om at adminpassord må endres.
    Endret passordet, uten å taste inn det gamle - Voila!
* Logget inn som Administrator med passord satt tidligere via "Remote Desktop
  Connection" i Windows (Tilkobling til et eksternt skrivebord" på norsk).
  Prøvde også med Remmina på Ubuntu. Domain kan forbli tom ved innlogging.
  Funker flott!
* I Admin Powershell:
  * Installerte chocolatey via https://chocolatey.org/docs/installation
  * 'choco install python --version=3.7.2'
  * 'choco install git'
  * 'choco install gitlab-runner'

```

```

* Installer pakker fra
<https://gitlab.stud.idi.ntnu.no/bachelor-paa-bittet/soitool/-/blob/
master/requirements.txt>
  * 'python.exe -m pip install setuptools --upgrade' for å få installasjon
    av pdoc3 til å funke
  * 'python.exe -m pip install -r requirements.txt'
* For å sette opp gitlab-runner:
  * 'gitlab-runner install'
  * 'gitlab-runner register' <- fyll inn info fra "Runners" biten av
    GitLab prosjektet i innstillinger. Fulgte denne guiden:
    <https://docs.gitlab.com/runner/register/index.html>
    * Satte tag som kan huske senere
    * Shell executor
  * I '.gitlab-ci.yml' kan nå bruke denne runneren med 'tags:'

**NOTE**: Når 90 dager gikk ut ble det gjort "rebuild" på instance, deretter
ble guiden over fulgt (fra punkt 3)

```

Kodeliste D.4: Docker-image for testing

```

FROM ubuntu:18.04

# This Dockerfile describes the container used in .gitlab-ci.yml

# Need docker build arg, such as --build-arg FLAKE8_VERSION=3.0.4
ARG FLAKE8_VERSION=3.7

# Installing python3.7
RUN apt-get update
RUN apt-get install software-properties-common -y
RUN add-apt-repository ppa:deadsnakes/ppa -y
RUN apt-get update
RUN apt-get install python3.7 -y
RUN python3.7 -V
RUN apt-get install python3-pip -y
RUN pip3 install --upgrade pip

# To make -platform offscreen,minimal,vnc work
RUN apt-get install -y libegl-mesa-dev \
    libfontconfig-dev \
    libssl-dev \
    libwayland-dev \
    libxkbcommon-dev

WORKDIR /code
COPY requirements.txt requirements.txt
RUN python3.7 -m pip install setuptools
RUN python3.7 -m pip install -r requirements.txt

# Set the locale
# Necessary for container to properly handle æøå
# See issue #10
RUN apt-get install -y locales
RUN locale-gen en_US.UTF-8
ENV LANG en_US.UTF-8
ENV LANGUAGE en_US:en
ENV LC_ALL en_US.UTF-8

```

```
# From https://it.i88.ca/2016/03/how-to-fix-importerror-libgssapikrb5so2.html
# python3.7 -m unittest fails without this
RUN apt-get -y install libgssapi-krb5-2

# Necessary for import of PySide2.QtWebEngineWidgets to work
RUN apt-get -y install libnss3 \
    libxcomposite-dev \
    libxcursor-dev \
    libxi-dev \
    libxtst-dev \
    libxrandr-dev
```

Kodeliste D.5: Installasjonsinstrukser fra leverbar ZIP

Installasjonsinstrukser for SOI-tool

SOI-tool er et verktøy for å lage SOI - "Samband- og operasjonsinstruks". Programvaren ble utviklet som en bacheloroppgave av fire studenter ved NTNU våren 2020 med Opplandske Heimevernsdistrikt 05 som oppdragsgiver. Dette dokumentet inneholder all informasjon nødvendig for å ta programvaren i bruk.

Programvaren leveres som en ZIP-fil som inneholder følgende:

- * Installasjonsinstrukser.txt
Dette dokumentet.
- * SOI-tool
Mappe som inneholder en eksekverbar utgave av programvaren til Windows 10.
- * Kildekode
Mappe med all kildekode til programvaren.
- * Dokumentasjon
Mappe med dokumentasjon til programvaren fremstilt som HTML.

Mappen "SOI-tool" inneholder mange filer, men de fleste kan ignoreres av bruker. I mappen finnes en fil "SOI-tool.exe", og ved å kjøre denne starter programvaren. Merk at programvaren kan dukke opp sammen med et "konsollvindu" med løpende tekst. Dette konsollvinduet er bare for å gi mer innsyn i programvaren, og kan ignoreres av bruker. For å gjøre det enklere å starte verktøyet i fremtiden kan det være lurt å lage en "snarvei". Høyreklikk på "SOI-tool.exe" og trykk "Lag snarvei". Denne kan plasseres hvor som helst, for eksempel på skrivebordet. Merk at filer produsert fra programvaren vil dukke opp i "SOI-tool" mappen hvor "SOI-tool.exe" ligger selv om snarvei brukes.

En fullstending innføring i bruk av programvaren er utenfor omfanget til dette dokumentet. Ved å trykke på "Hjelp > Enkel bruk" i toppmenyen vil en kort innføring i bruk av verktøyet presenteres for bruker.

Resten av dette dokumentet kan sannsynligvis ignoreres om du bare er interessert i å bruke selve verktøyet.

Støttede skjermskaleringer

Programvaren støtter i utgangspunktet ikke skjermskaleringer over 125%. Programvaren kan kjøres på høyere skaleringer, men vi kan desverre ikke garantere at SOI-filer laget med en skjermskalering på over 125% kan åpnes av mottaker. Se bachelorrappport for utdyping.

Dokumentasjon

Mappen "Dokumentasjon" inneholder en HTML versjon av dokumentasjonen til programvaren. Informasjonen en finner her er den samme en kan finne ved å lese dokumentasjon i kildekoden, men å lese som en lokal nettside kan være mer behagelig. Denne kan åpnes ved å trykke på filen "index.html".

Kildekode

Mappen "Kildekode" inneholder kildekoden i sin helhet. Kildekoden eksisterer også på NTNU GitLab [1], hvor en også kan finne testhistorikk og tekniske diskusjoner.

Konfigurasjon av moduler

Noen moduler kan konfigureres via konfigurasjonsfiler. Dette gjør det enklere å tilpasse og vedlikeholde programvaren. Konfigurasjonsfilene er av typen json og formatet er beskrevet i tilhørende modul sin dokumentasjon. Konfigurasjonsfilene ligger i SOI-tool/soitool/modules/config.

[1] <https://gitlab.stud.iie.ntnu.no/bachelor-paa-bittet/soitool>

Kodeliste D.6: Avhengigheter til programvaren

```
altgraph==0.17
astroid==2.3.3
bandit==1.6.2
colorama==0.4.3
entrypoints==0.3
flake8==3.7.9
future==0.18.2
gitdb==4.0.1
gitdb2==4.0.2
GitPython==3.0.8
isort==4.3.21
lazy-object-proxy==1.4.3
linecache2==1.0.0
Mako==1.1.1
Markdown==3.2.1
MarkupSafe==1.1.1
mccabe==0.6.1
nose==1.3.7
pbr==5.4.4
pdoc3==0.7.4
pefile==2019.4.18
Pillow==7.0.0
pycodestyle==2.5.0
pydocstyle==5.0.2
pyflakes==2.1.1
PyInstaller==3.6
pylint==2.4.4
pyparsing==2.4.6
PySide2==5.14.1
pywin32-ctypes==0.2.0
```

```

PyYAML==5.3
rectpack==0.2.1
reportlab==3.5.34
shiboken2==5.14.1
six==1.14.0
smmap==3.0.1
snowballstemmer==2.0.0
stevedore==1.32.0
svgwrite==1.3.1
tinydb==3.15.2
traceback2==1.4.0
typed-ast==1.4.1
unittest2==1.1.0
wrapt==1.11.2
schedule==0.6.0
black==19.10b0
schema==0.7.1

```

Kodeliste D.7: Script brukt for å sammenligne kompresjonsbiblioteker

```

import zlib
import bz2
import lzma
import json

# https://stackoverflow.com/questions/196345/how-to-check-if-a-string-in-python-is-in-ascii
def is_ascii(s):
    return all(ord(c) < 128 for c in s)

def compare_algorithms(s, file_name):
    str_len = len(s)

    zlib_compression = str(zlib.compress(s.encode()))
    zlib_data = {
        "name": "zlib",
        "length": len(zlib_compression),
        "ascii": is_ascii(zlib_compression),
        "diff": len(zlib_compression) - str_len,
    }

    bz2_compression = str(bz2.compress(s.encode()))
    bz2_data = {
        "name": "bz2",
        "length": len(bz2_compression),
        "ascii": is_ascii(bz2_compression),
        "diff": len(bz2_compression) - str_len,
    }

    lzma_compression = str(lzma.compress(s.encode()))
    lzma_data = {
        "name": "lzma",
        "length": len(lzma_compression),
        "ascii": is_ascii(lzma_compression),
        "diff": len(lzma_compression) - str_len,
    }

    algorithms = [zlib_data, bz2_data, lzma_data]

    winner_value = min(

```

```
        zlib_data["length"], bz2_data["length"], lzma_data["length"]
    )

    print("\n\n" + file_name + "_( " + str(len(s)) + " )\n")

    for a in algorithms:
        print(a)

    print("")

    for a in algorithms:
        if winner_value == a["length"]:
            print("\U0001F37E" + " WINNER IS: " + a["name"] + " \U0001F37E")

files = {
    "SOI_tom": "/home/nikolai/Desktop/compression_research/tom_soi.json",
    "SOI_mini": "/home/nikolai/Desktop/compression_research/mini_soi.json",
    "SOI_liten": "/home/nikolai/Desktop/compression_research/liten_soi.json",
    "SOI_medium": "/home/nikolai/Desktop/compression_research/medium_soi.json",
    "SOI_stor": "/home/nikolai/Desktop/compression_research/stor_soi.json",
    "SOI_mega": "/home/nikolai/Desktop/compression_research/mega_soi.json",
}

for file in files:
    with open(files[file], "r") as f:
        x = json.load(f)
        compare_algorithms(str(x), file)
```

```
SOI tom (413)
{'name': 'zlib', 'length': 720, 'ascii': True, 'diff': 307}
{'name': 'bz2', 'length': 839, 'ascii': True, 'diff': 426}
{'name': 'lzma', 'length': 919, 'ascii': True, 'diff': 506}
🔑 WINNER IS: zlib 🔑

SOI mini (1694)
{'name': 'zlib', 'length': 2765, 'ascii': True, 'diff': 1071}
{'name': 'bz2', 'length': 2820, 'ascii': True, 'diff': 1126}
{'name': 'lzma', 'length': 2821, 'ascii': True, 'diff': 1127}
🔑 WINNER IS: zlib 🔑

SOI liten (3415)
{'name': 'zlib', 'length': 4697, 'ascii': True, 'diff': 1282}
{'name': 'bz2', 'length': 4891, 'ascii': True, 'diff': 1476}
{'name': 'lzma', 'length': 4709, 'ascii': True, 'diff': 1294}
🔑 WINNER IS: zlib 🔑

SOI medium (8366)
{'name': 'zlib', 'length': 10712, 'ascii': True, 'diff': 2346}
{'name': 'bz2', 'length': 9688, 'ascii': True, 'diff': 1322}
{'name': 'lzma', 'length': 9345, 'ascii': True, 'diff': 979}
🔑 WINNER IS: lzma 🔑

SOI stor (16130)
{'name': 'zlib', 'length': 17674, 'ascii': True, 'diff': 1544}
{'name': 'bz2', 'length': 15904, 'ascii': True, 'diff': -226}
{'name': 'lzma', 'length': 14822, 'ascii': True, 'diff': -1308}
🔑 WINNER IS: lzma 🔑

SOI mega (26887)
{'name': 'zlib', 'length': 27295, 'ascii': True, 'diff': 408}
{'name': 'bz2', 'length': 23689, 'ascii': True, 'diff': -3198}
{'name': 'lzma', 'length': 22443, 'ascii': True, 'diff': -4444}
🔑 WINNER IS: lzma 🔑
```

Figur D.1: Output fra scriptet D.7

Vedlegg E

Prosjektplan

Prosjektplan skrevet som forberedelse til oppgaven. Hele dokumentet er inkludert, også vedlegg som overlapper med vedlegg fra denne rapporten.

Prosjektplan Bacheloroppgave

Nikolai Mork
BIDAT

Thomas Løkkeborg
BITSEC

Anders H. Rebner
BIDAT

Petter Sagvold
BIDAT

Januar 2020

Innhold

1	Mål og rammer	3
1.1	Bakgrunn	3
1.2	Prosjektmål	3
1.2.1	Resultatmål	3
1.2.2	Effektmål	3
1.2.3	Læringsmål	4
1.3	Rammer	4
2	Omfang	5
2.1	Fagområde	5
2.2	Avgrensning	5
2.3	Oppgavebeskrivelse	6
3	Prosjektorganisering	8
3.1	Ansvarsforhold og roller	8
3.2	Rutiner og regler i gruppa	8
4	Planlegging, oppfølging og rapportering	10
4.1	Hovedinndeling av prosjektet	10
4.2	Plan for statusmøter	10
5	Organisering av kvalitetsikring	11
5.1	Dokumentasjon, standardbruk og kildekode	11
5.2	Konfigurasjonsstyring	11
5.3	Risikoanalyse	11
5.3.1	Prosjektrisiko	12
5.3.2	Produktrisiko	13
6	Plan for gjennomføring	14
6.1	Liste over aktiviteter	14
6.2	Gantt-skjema	17
6.2.1	Kommentarer til Gantt-skjema	18

1 Mål og rammer

1.1 Bakgrunn

Studenter ved IDI¹ på NTNU i Gjøvik skal som et avsluttende arbeid for utdanningen utføre et større prosjektarbeid der de skal sette sammen kunnskapene de har tilegnet seg iløpet av utdanningen.

Oppgaven i dette prosjektet er gitt av Opplandske Heimevernsdistrikt 05 som har et reelt problem i sin sambandsseksjon. Prosjektet skal utføres over en periode på 18 uker og blir vurdert utifra en rapport av arbeidet.

1.2 Prosjektmål

1.2.1 Resultatmål

Ved prosjektets slutt skal det leveres en prototype som støtter:

- modulene radiodiagram, autentiseringstavle, subtraktorkoder, forhåndsavtalte koder, frekvenslister, kallesignaler, kodebøker og telefonlister. Eksempler på disse finnes i vedlegg A og B.
- et grafisk brukergrensesnitt der brukeren skal kunne velge hvilke moduler som skal være med i dokumentet, samt mulighet for å redigere enkelte moduler.
- autogenerering av enkelte moduler.
- serialisering og komprimering av SOI-fil.
- innlesing av dokument på fil, som deretter kan redigeres og/eller omgjøres til leselig PDF.

Prototypen skal vise at SOI-generering kan gjøres raskere og bedre med programvare, og at genererte SOI-er kan sendes mer kompakt.

1.2.2 Effektmål

På lang sikt er målet at det blir allokert ressurser for å utvikle en full versjon basert på prototypen. Dette vil avlaste sambandsseksjonen og gi dem mer tid til planlegging, vedlikehold og drift av sambandssystemene.

¹IDI: Institutt for datateknologi og informatikk

1.2.3 Læringsmål

Gruppemedlemmene skal tilegne seg:

- kunnskap innenfor valgt programmeringsspråk, samt rammeverk for grafisk brukergrensesnitt og database.
- grunnleggende forståelse av kryptografi innenfor militær sambandskommunikasjon.
- erfaring vedrørende systematisk planlegging og metodisk arbeid med et større prosjekt.
- erfaring med å samle og anvende kunnskap fra utdanningen vår.
- erfaring i samarbeid med oppdragsgiver og kontaktperson om en reell oppgave.

1.3 Rammer

- Gruppa har ikke fast arbeidsrom, og er avhengige av møterom ved NTNU i Gjøvik og bruk av egne hybler/leiligheter.
- Programvaren skal kunne kjøres på 64-bit maskiner som kjører nyere versjon av operativsystemet Windows 10.
- Tidsrammen for prosjektet er satt fra 08.01.20 til 20.05.20.
- Vi er fire på gruppa, og er gjennom gruppereglene(3.2) pliktet å jobbe minst 25 timer i uka. Med 18 arbeidsuker utgjør dette minst 1800 timer totalt.

2 Omfang

2.1 Fagområde

”Kommunikasjon er det å formidle og dele ideer og informasjon.”[1]. Med tiden har elektronisk kommunikasjon blitt avgjørende i samfunnet. Elektronisk kommunikasjon kommer i en rekke former, eksempelvis internett, radio, telefoni og satellitt. Ved nettverksutbygging av nevnte teknologier har trådløs langdistansekommunikasjon blitt et faktum. [2]. Telefonsamtaler, sosiale medier og e-post dekker de fleste behov i sivile situasjoner. I militære situasjoner² stilles det derimot andre krav til kommunikasjon, da ofte omtalt som samband. Heimevernet utgjør hoveddelen av Norges førstelinjeforsvar og reagerer raskt lokalt i hele landet[3]. Som alle grener i Forsvaret er Heimevernet også avhengig av samband for å gjennomføre sine operasjoner.

I militær sammenheng er radio mye brukt for kommunikasjon mellom enheter i felten. VHF (Very High Frequency) kan sende data raskt, men har begrenset rekkevidde. HF (High Frequency) har tregere dataoverføring, men har til gjengjeld lengre rekkevidde. Hvilke kommunikasjonsutstyr en enhet er utstyrt med avhenger av hvilke distanser de kommuniserer på, og om de er oppsatt på kjøretøy. Utstyret kan variere fra små håndholdte enheter til tunge radioer som krever store antenner. Mobiltelefoni, satellitt og fysisk filoverføring brukes også til dels. Både klassisk radiokommunikasjon med lyd-samtale og meldingsutveksling med skjerm benyttes.

2.2 Avgrensning

For at kommunikasjonen via sambandet skal foregå korrekt og sikkert brukes et dokument kalt en SOI (Signal Operating Instructions). En SOI inneholder alle nødvendige opplysninger en avdeling og underliggende avdelinger trenger for å etablere kommunikasjon over ulike kommunikasjonsmidler, som f.eks. HF-radio og nødnett. SOI består av flere ulike moduler slik som subtraktorkoder, kode-tavler, frekvenstabeller og lignende. I praksis inneholder altså en SOI informasjon om hvordan etablere samband og kommunisere uten at det skal være forståelig for uvedkommende.

Heimevernet bruker i dag svært mye tid på å produsere dette dokumentet fordi det i stor grad gjøres manuelt. Dette er tid som gjerne skulle blitt brukt på andre viktige oppgaver som sambandsplanlegging, og vedlikehold og drift av systemene.

Oppgaven er avgrenset til å håndtere generering, produksjon og administrasjon av SOI'er og omfatter ikke distribusjon av dem. Resultatet av oppgaven skal være en så godt som mulig fungerende prototype, ikke et produkt klart for daglig drift.

²Øvelser og oppdrag

2.3 Oppgavebeskrivelse

Følgende krav skal oppfylles av den ferdigstilte prototypen:

- Funksjonalitet for modulene radiodiagram, autentiseringstavle, subtraktorkoder, forhåndsavtalte koder, frekvenslister, kallesignaler, kodebøker og telefonlister.
- Modulene skal være redigerbare.
- Moduler som for eksempel subtraktorkoder, autentiseringstavler og kodeordlister har innhold som skal generes automatisk og tilfeldig.
- Brukeren skal kunne plassere modulene i forhold til hverandre på SOI'en, men det er også ønskelig at programvaren skal gi et optimalisert forslag til modulplasseringen.
- Hver SOI må som et absolutt minimum inneholde graderingsmerke (rødt), dato og versjonsnummer.
- Når en SOI er ferdigstilt må den serialiseres og komprimeres (US ASCII) til en svært kompakt versjon slik at det er mulig å sende den over lav båndbredde på eksempelvis 2.4Kilobit per sekund (kbps). Det vil være rimelig at overføring skjer innen ett minutt, mer enn fem minutter er uakseptabelt. Mottaker av denne kompakte filen må kunne generere SOI tilbake til leselig PDF-format. Mottaker skal også kunne tilpasse utseende av mottatt SOI og fjerne eller legge til innhold.
- SOI-er må produseres som godt leselige PDF-er for enkel utskrift.
- Utskrevet skal SOI være på maksimalt to A4-ark.
- Ved døgnskifte eller kompromittering skal SOI skiftes ut, og det skal derfor være mulig å produsere SOI'er som ligger i kø.
- Brukergrensesnittet skal ha et design som er effektivt for militært personell slik de kan prioritere tid til annet arbeid.
- Selve SOI'en skal være organisert og utformet på en måte som gjør at den er enkel og rask i bruk, selv i svart/hvitt. Valgt font skal være vitenskapelig vektet og gi optimal lesbarhet på papir.
- Alle militære enheter trenger ikke lik mengde data på en SOI. Utifra en hoved-SOI skal det derfor være mulig å generere mindre versjoner av denne.
- Applikasjonen skal være modulbasert slik at det etter fullført prosjekt er enkelt å legge til ytterligere moduler.
- Programmeringsspråket skal være av lett leselig karakter³ siden arbeidet skal overtas og muligens jobbes videre med etter endt prosjekt.

³Med dette menes et språk hvor allmenn konsensus i datamiljøet er at det er leselig

- Hver SOI skal lagres i en database slik at alle tidligere SOI'er er tilgjengelig.
- Databasen som skal brukes i prosjektet må være innebygget i programvaren slik at denne kan fungere isolert.
- Innholdet i databasen skal kunne fraktes på minnepinne. Dette er for å muliggjøre fysisk flytting av data i situasjoner hvor elektronisk kommunikasjon ikke egner seg.
- Applikasjonen skal inneholde en kode- og dekodebok:
 - Det skal være mulig å oppdatere innholdet i denne.
 - Det skal være en kort og en lang versjon av kodeboka. Den korte skal inneholde de mest brukte kodene, mens den lange skal inneholde alle.
 - Kodeboka og dekodeboka kommer enten som separate bøker eller som et samlet verk.
- Installasjonsinstruksjoner for programvaren skal inkluderes og utgjøre maksimalt en halv side. Korte instruksjoner for bruk av programvaren skal også inkluderes.
- SOI skal kunne leveres med "vedlegg" i form av ekstra kodeord som må pugges. Det må være tydelig at dette vedlegget må destrueres etter at kodeord er pugget.

Om det blir tid omfatter også oppgaven følgende:

- Evne til å fange opp kodeord som er fonetisk like.
- Verktøy for å lage sambandsdiagrammer. I dette verktøyet skal det være mulig å skissere opp hindringer og enheters plasseringer og ut i fra dette vil applikasjonen generere et sambandsdiagram. Ved endringer skal det være enkelt og raskt å generere et oppdatert diagram.
- Mulighet for å synkronisere data oppover og nedover i hierarkiet. Eller i det minste "proof of concept" på dette.

3 Prosjektorganisering

3.1 Ansvarsforhold og roller

Gruppemedlemmene sikter på å være sidestilt, som innebærer at alle skal jobbe med research, rapport, utvikling og andre prosjektrelaterte aktiviteter. Flertallet skal bestemme ved faglige avgjørelser i henhold til avsnitt 3.2. Vi har som mål at gruppa samarbeider tett slik at alle medlemmene får utbytte og innsikt gjennom hele prosjektet, noe som også vil gagne oppgavens resultat.

Anders H. Rebner er utnevnt til prosjektleder. Prosessveileder er Frode Haug, universitetslektor ved NTNU i Gjøvik. Oppdragsgiver er Opplandske Heimevernsdistrikt 05, G6-seksjonen. Kontaktpersoner er Major Leif Tanum og Espen Torseth, seniorrådgiver ved NTNU i Gjøvik.

3.2 Rutiner og regler i gruppa

Gruppemedlemmene plikter å:

- Jobbe (minst) 25 timer i uken med oppgaven
- Føre individuell timelogg som skal inneholde navn, dato og timer
 - Navn
 - Dato
 - Antall timer
 - Stikkord om arbeidsoppgaven(e)
- Delta aktivt i oppgaven:
 - Alle deltar i prosjektplanlegging
 - Alle skriver kode og rapport
- Møte opp i alle møter om ikke annet er avtalt. Om oppmøte ikke er mulig, eller ved forsinkelse, pliktes det å si ifra “i god tid”.
- Møte fysisk etter de planlagte mandagsmøtene.
- Dele på ansvaret om å skrive referat etter aktuelle møter.
- Si ifra om en henger etter med arbeidsoppgave(r).
- Følge flertallet ved faglige avgjørelser.
- Melde fra tidlig om en er misfornøyd med innsatsen til andre gruppemedlemmer.
- Skrive dokumentasjon der det er hensiktsmessig

- Følge følgende språkkonvensjon:
 - Kode skrives på engelsk
 - Alt annet skrives på norsk bokmål

Rutine ved ("alvorlige nok") brudd på reglene:

1. Samtale(r) mellom alle gruppedeltakerne om problemet.
2. Deretter en skriftlig advarsel fra de andre (evt. bare lederen) der:
 - regelbruddet blir påpekt
 - konkret (tidsfrister, forventet arbeid(sinnsats), m.m.) om hva som kreves for å "bøte på skaden"
 - konkret om konsekvensene (samtale m/veileder, ekskludering o.l.) av manglende kravoppfyllelse
3. Så: samtale mellom hele gruppen og veileder der passende beslutning nås

4 Planlegging, oppfølging og rapportering

4.1 Hovedinndeling av prosjektet

Gruppen har resonnert seg frem til å bruke en slags smidig fossefall som utviklingsmodell. Valget er blant annet begrunnet i at det har blitt gitt et relativt tydelig bilde av hva som skal utvikles. Vi ser derfor for oss at det ikke vil bli nødvendig med store endringer underveis. I tillegg har prosjektet en bestemt leveringsdato. Kombinasjonen av disse argumentene viser at det er hensiktsmessig å sette klare datoer for når deler av prosjektet skal være fullført.

Innad i hver av fossefall-bolkene skal arbeidsmetodikken være mer smidig. Vi vil bruke en typisk smidig arbeidstavle for oversikt over arbeidsflyten på et mer detaljert nivå. Det skal holdes ukentlige møter med kontaktperson for å gjøre opp status. Applikasjonen skal være brukervennlig og intuitiv grunnet mulig tidspress under bruk. Det er derfor planlagt å brukerteste, noe som betyr at små endringer kan måtte gjøres basert på tilbakemeldinger fra disse. Tidlig i prosjektet skal en enkel mock-up brukertestes slik at vi unngår omveltende endringer ved et senere tidspunkt i prosjektet.

4.2 Plan for statusmøter

Gruppen skal ukentlig møte:

- kontaktperson og oppdatere på ukas framskritt, kommende ukes mål, klargjøre aspekter ved prosjektet og eventuelt finstille retningen til oppgaven.
- prosessveileder for å få forslag og anbefalinger til prosessen i arbeidet og skriving av rapport.
- for å ta avgjørelser knyttet til oppgaven, codereview og diskutere aktualiteter fra de to overnevnte møtene.

5 Organisering av kvalitetsikring

5.1 Dokumentasjon, standardbruk og kildekode

Oppgaven utvikles som en prototype på et potensielt fremtidig system, derfor stiller oppdragsgiver krav til god kvalitetssikring. Dokumentasjon vil skrives der det er hensiktsmessig, på norsk. Kode skrives på engelsk. Et lesbart programmeringsspråk må velges, og kildekode skal skrives slik at den blir lettest mulig forståelig for de som eventuelt skal overta arbeidet. Standarder fra Forsvaret og Nato vil følges der det passer seg.

Automatisert testing av kildekode vil foregå der vi finner det hensiktsmessig. Verktøy for kvalitetsikring av kildekode og pipelines vil bli brukt. Gruppa skal utføre code review for å sikre både kvaliteten på selve koden og for å dele forståelsen for koden.

Vi skal følge WCAG 2.0⁴ suksesskriterie 1.4.3 kontrast på nivå AA. Kriteriet sier at det skal være tilstrekkelig kontrast(4,5:1) mellom bakgrunn og tekst. Dette skal gjelde i applikasjonen, men også på utskrevet dokument da lesbarhet er svært viktig. Det vil ikke være nødvendig med ytterligere fokus på universell utforming fordi militært personell oppfyller Forsvarets helsekrav[4], og har dermed ikke spesielle utfordringer med syn og hørsel.

Til SOI på fil skal standard "US ASCII" brukes. Dette for å maksimere mulig bruk av filen.

5.2 Konfigurasjonsstyring

All kode, samt selve rapporten, vil gå under konfigurasjonsstyring. Git skal brukes, med privat repository som hosting. Featurebranches skal brukes for å sikre orden i prosjektet. Branchene vil ha pipelines og kreve merge requests for å sikre kvalitet i koden.

5.3 Risikoanalyse

Vi har identifisert potensielle risikoer, vektlagt dem på sannsynlighet og konsekvens, og bestemt nødvendige tiltak. Prosjektrisiko og produktrisiko har blitt vurdert separat.

⁴WCAG: Web Content Accessibility Guidelines

5.3.1 Prosjektrisiko

Prosjektrisikoer omhandler risikoer rundt selve bachelorprosjektet. Konsekvenser er vurdert med dette som kontekst.

Risiko	Sannsynlighet	Konsekvens	Kommentar og Tiltak
Gruppemedlem kan ikke møte (sykdom e.l.) over kortere tid	Høy	Middels	Gruppemedlemmer deler på arbeidsoppgaver slik at tapet av et gruppemedlem ikke "blokkerer"
Gruppemedlem kan ikke møte (sykdom e.l.) over lengre tid	Lav	Høy	Gruppemedlemmer deler på arbeidsoppgaver slik at tapet av et gruppemedlem ikke "blokkerer"
Prosjektet har flere dype tema en kan "grave seg inn i". En risiko er at vi bruker for mye tid på en oppgave, slik at det går ut over andre oppgaver	Middels	Høy	Vi må estimere tidsbruk før vi begynner på oppgaver, og huske på at prosjektet bare er en prototype
Fundamental forandring i "kravspesifikasjon" / omfang til prosjektet	Lav	Høy	Ha god dialog med oppdragsgiver om dette, og kommunisere vår forståelse av oppgaven klart undervegs
Tap av data rundt bacheloroppgaven	Lav	Høy	Lagre viktige dokumenter i Google Docs, ha gode vaner rundt versjonskontroll
Ikke rekke å bli ferdig med viktig funksjonalitet i prototypen	Middels	Middels	Definere "minste brukbare produkt" og ferdigstille det først
Samarbeid i gruppa kollapser	Lav	Høy	Klart definerte grupperegler og faste statusmøter

5.3.2 Produktrisiko

Produktrisikoer omhandler risikoer rundt SOI-verktøyet vi skal utvikle. Konsekvenser er vurdert med dette som kontekst.

Risiko	Sannsynlighet	Konsekvens	Kommentar og Tiltak
Valgte teknologier fungerer ikke som forventet	Middels	Middels	Utføre grundig undersøkelse av teknologier før de tas i bruk
Teknologi vi forventer å kunne ta i bruk finnes ikke (eks. oppdager det ikke eksisterer noen gode PDF-bibliotek til Python)	Lav	Høy	Utføre grundig undersøkelse av teknologier før de tas i bruk
SOI-verktøy utvikles av andre	Lav	Lav	Vårt prosjekt er bare en prototype, så ingen tiltak er nødvendig
Prototype kan ikke kjøres på brukers maskiner	Lav	Høy	Prototype må testes på en maskin tilsvarende brukers maskin. Installasjonsinstrukser må være skrevet med tanke på brukers maskiner
Ferdig prototype representerer ikke ønskene til brukerne våre, og kan ikke tas med videre opp i systemet	Middels	Middels	Brukertesting blir kritisk for å sikre at prototypen vår møter ønskene til brukerne

6 Plan for gjennomføring

6.1 Liste over aktiviteter

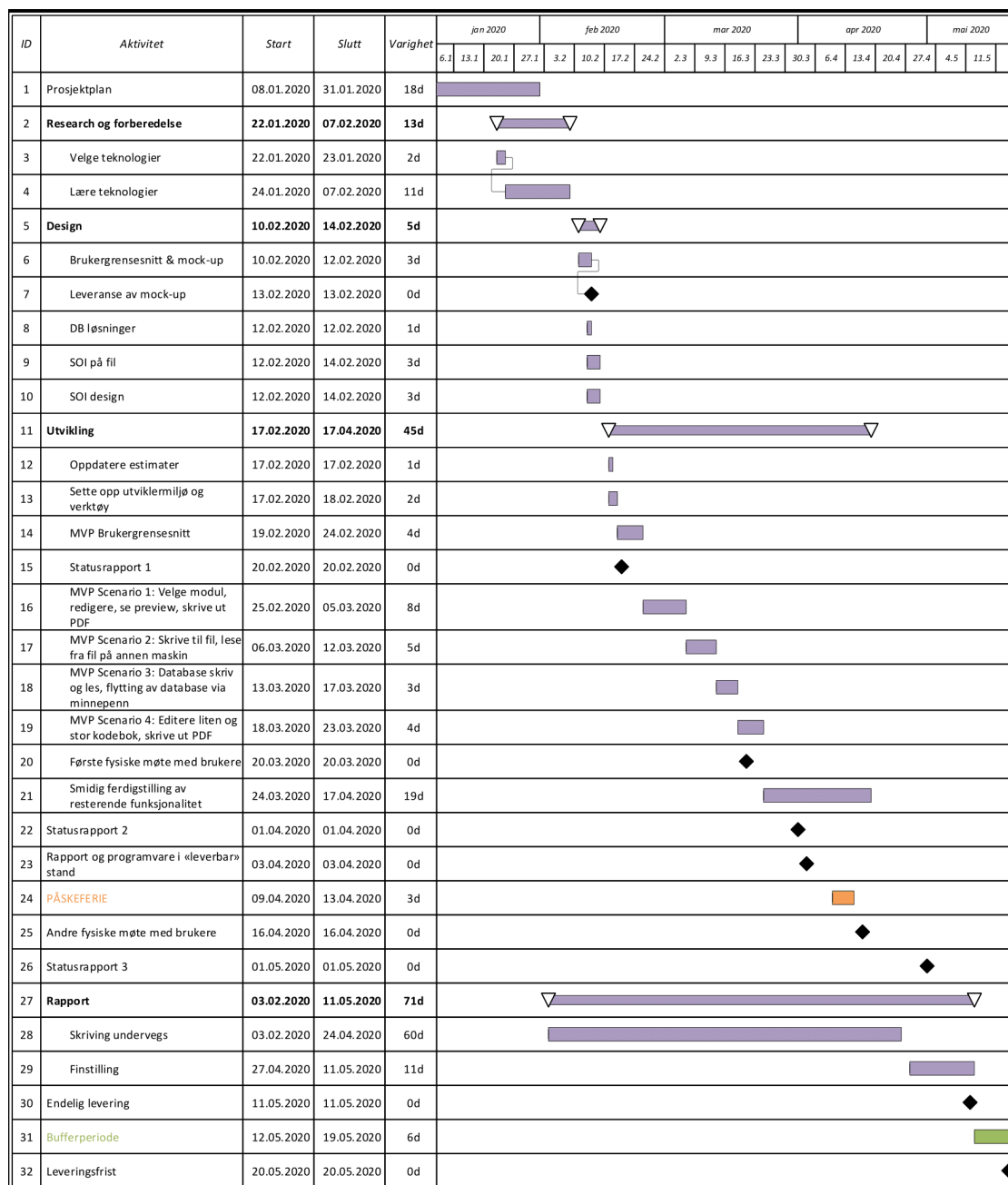
Følgende liste inneholder aktiviteter som må utføres for å fullføre prosjektet. Lista er ikke sortert, og flere punkter avhenger av tidligere aktiviteter.

1. Prosjektstyring
 - 1.1. Grupperegler
 - 1.2. Prosjektplan
 - 1.2.1. Mål og rammer
 - 1.2.2. Omfang / kravspesifikasjon
 - 1.2.3. Prosjektorganisering
 - 1.2.4. Planlegging, oppfølging og rapportering
 - 1.2.5. Organisering av kvalitetssikring
 - 1.2.6. Plan for gjennomføring
 - 1.2.7. Tilbakemelding fra veileder
 - 1.3. Prosjektavtale
2. Research og forberedelse
 - 2.1. Velge teknologier
 - 2.1.1. Database
 - 2.1.2. Programmeringsspråk
 - 2.1.3. GUI-bibliotek
 - 2.1.4. PDF-bibliotek
 - 2.1.5. Databaseklient-bibliotek
 - 2.2. Lære seg valgte teknologier
 - 2.3. Sette i verk valgt kvalitetssikring
 - 2.4. Utforske lignende eksisterende programvare
 - 2.5. Konkretisere ønskede moduler
 - 2.6. Samle use-case fra brukere
3. Design
 - 3.1. SOI
 - 3.1.1. Overordnet design

- 3.1.2. Individuelle moduler
 - 3.1.3. Layout av moduler
- 3.2. Brukergrensesnitt
 - 3.2.1. Editering av SOI
 - 3.2.2. Editering av individuelle moduler
 - 3.2.3. Editering av kodebok
 - 3.2.4. Import av SOI på fil
 - 3.2.5. Eksport av SOI på fil
 - 3.2.6. Manuell redigering av modul-layout i SOI
- 3.3. Database
 - 3.3.1. Konkretisere data som må lagres i database
 - 3.3.2. Tenke ut hvordan data skal lagres i praksis (schema, ol.)
 - 3.3.3. Tenke ut hvordan data kan synkroniseres mellom brukere
- 3.4. SOI på fil
 - 3.4.1. Lagringseffektivt filformat
 - 3.4.2. Komprimering
- 4. Utvikling
 - 4.1. Brukergrensesnitt
 - 4.1.1. Editering av individuelle moduler
 - 4.1.2. Editering av kodebok
 - 4.1.3. Import av SOI på fil
 - 4.1.4. Eksport av SOI på fil
 - 4.1.5. Editering av SOI
 - 4.1.6. Automatisk layout av moduler i SOI
 - 4.1.7. Manuell redigering av modul-layout i SOI
 - 4.2. Utskrift av SOI til PDF
 - 4.3. SOI på fil
 - 4.3.1. Eksport av SOI på fil
 - 4.3.2. Import av SOI på fil
 - 4.3.3. Komprimering
 - 4.4. Database

- 4.4.1. Implementere schema ol.
 - 4.4.2. Hente data fra database
 - 4.4.3. Skrive data til database
 - 4.4.4. Synkronisering av database med andre brukere
- 5. Brukertestning
 - 5.1. Planlegge tester
 - 5.2. Utføre tester
 - 5.3. Ta i bruk tilbakemelding
- 6. Rapport
 - 6.1. Skrive rapport
 - 6.2. Tilbakemelding fra veileder
 - 6.3. Finlesing og forbedring

6.2 Gantt-skjema



6.2.1 Kommentarer til Gantt-skjema

- Skjema ble laget på grunnlag av 6.1, med Planning Poker som estimeringshjelpemiddel.
- Aktivitene er fargelagt etter følgende konvensjon:
 - **Lilla**: Arbeid
 - **Oransje**: Ferie
 - **Grønn**: Potensielt arbeid
- Utviklingsfasen (id:11) vil i praksis bli utført i ”sykluser” på én uke.
- Første punkt i utviklingsfasen (id:11) er å utføre en ny estimering før vi begynner å skrive kode. Dette er for å få mer nøyaktige estimater på bakgrunn av utført research og design.
- Det er innlagt en bufferperiode i tilfelle noe uforutsett inntreffer (id:31).
- MVP står for Minimal Viable Product, altså den enklest mulige versjonen av applikasjonen som kan vises fram.
- Statusrapport er korte rapporter som skal leveres til veileder innen de gitte datoene. Disse skal vise veileder hvor langt prosjektet har kommet.

Ordliste

codereview Gjennomgang av kode i plenum.

featurebranches Gren dedikert til en funksjonalitet i Git.

forhåndsavtalte koder Se kodetavle.

frekvenstabell Tabell i SOI med oppdeling av frekvenser.

kallesignaler Modul i SOI som viser kallesignaler på ulike nivåer basert på et hierarkisk organisasjonstre.

kbps kilobit per sekund.

kodebok En kodebok er innen kryptografi ei bok som inneholder en oppslags-tabell for koding og dekodning[8].

kodetavle Modul i SOI som viser relevante koder fra kodebok.

merge requests Forespørsel om å integrere en gren til hovedkildekoden.

pipelines Ment her som verktøy for å utføre en handling ved en hendelse. Eks. teste kode ved push, eller sende melding ved ny "merge request".

radiodiagram Tabell i SOI med informasjon om radio.

samband Samband er kommunikasjon, og beskriver kommunikasjonsnettverket i en organisasjon.[7].

subtraktorkode Modul i SOI nødvendig for å tolke lokasjonsdata.

Referanser

- [1] Nicholas Allott. Kommunikasjon. i store norske leksikon., Mai 2019. [På internett; besøkt 14-01-2020].
- [2] Gunnar Bothner-By, Halvor & Stette. Kommunikasjonsteknologi. i store norske leksikon, Mai 2019. [På internett; besøkt 14-01-2020].
- [3] Forsvaret. Heimevernet. [På internett; besøkt 20-01-2020].
- [4] Forsvaret. Helsekrav, Januar 2020. [På internett; besøkt 17-01-2020].
- [5] Wikipedia. Serialisering — wikipedia,, 2018. [På internett; besøkt 13-01-2020].
- [6] Wbs, estimation and scheduling. <https://cs.uwaterloo.ca/~apidduck/CS846/Lectures/WBS>. [På internett; besøkt 10-01-2020].
- [7] Wikipedia. Samband, August 2019. [På internett; besøkt 16-01-2020].
- [8] Wikipedia contributors. Codebook, Desember 2019. [På internett; besøkt 16-01-2020].

RESTRICTED\REL NOR
UPDATED 030602_5EPT16

SOI 2BN

PAGE 1 of 2

RESTRICTED\REL NOR

2BN VHF NET PRIMARY/ALTERNATE COMS									
NET NAME	NET ID	TYPE	FREQ/MHz	SEC	SYNCH MASTER	ACCESS	SWTCH HOST NAME	NET NAME	NET ID
2BN PRIMARY CMD VOICE/BMS									
ZBN CMD1	1401	CNR FF	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN CMD2	1402	CNR FF	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN CMD3	1403	CNR FF	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN CMD4	1404	CNR FF	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN CMD5	1405	CNR FF	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN ALTERNATE VOICE/ADMS									
ZBN ADM/BMS1	1411	CNR TD	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN ADM/BMS2	1412	CNR TD	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN ADM/BMS3	1413	CNR TD	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN ADM/BMS4	1414	CNR TD	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
ZBN ADM/BMS5	1415	CNR TD	R	9/93A/93B/93C		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY PRIMARY CMD VOICE/BMS									
BPRI CMD/BMS	1420	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
BALT VOICE	1421	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
B1 PRI	1422	CNR FF	R	B-11		2BN		2BN PLATOON NET	
B1 ALT	1423	CNR FF	R	O/O		2BN		2BN PLATOON NET	
B2 PRI	1424	CNR FF	R	B-21		2BN		2BN PLATOON NET	
B2 ALT	1425	CNR FF	R	O/O		2BN		2BN PLATOON NET	
B3 PRI	1426	CNR FF	R	B-31		2BN		2BN PLATOON NET	
B3 ALT	1427	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
CPRI CMD/BMS	1430	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
CAALT VOICE	1431	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
C1 PRI	1432	CNR FF	R	C-11		2BN		2BN PLATOON NET	
C1 ALT	1433	CNR FF	R	O/O		2BN		2BN PLATOON NET	
C2 PRI	1434	CNR FF	R	C-21		2BN		2BN PLATOON NET	
C2 ALT	1435	CNR FF	R	O/O		2BN		2BN PLATOON NET	
C3 PRI	1436	CNR FF	R	C-31		2BN		2BN PLATOON NET	
C3 ALT	1437	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
DPRI CMD/BMS	1440	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
DALT VOICE	1441	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
D1 PRI	1442	CNR FF	R	D-11		2BN		2BN PLATOON NET	
D1 ALT	1443	CNR FF	R	O/O		2BN		2BN PLATOON NET	
D2 PRI	1444	CNR FF	R	D-21		2BN		2BN PLATOON NET	
D2 ALT	1445	CNR FF	R	O/O		2BN		2BN PLATOON NET	
D3 PRI	1446	CNR FF	R	D-31		2BN		2BN PLATOON NET	
D3 ALT	1447	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
EPRI CMD/BMS	1450	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
EALT VOICE	1451	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
E1 PRI	1452	CNR FF	R	E-11		2BN		2BN PLATOON NET	
E1 ALT	1453	CNR FF	R	O/O		2BN		2BN PLATOON NET	
E2 PRI	1454	CNR FF	R	E-21		2BN		2BN PLATOON NET	
E2 ALT	1455	CNR FF	R	O/O		2BN		2BN PLATOON NET	
E3 PRI	1456	CNR FF	R	E-31		2BN		2BN PLATOON NET	
E3 ALT	1457	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
FPRI CMD/BMS	1460	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
FALT VOICE	1461	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
F1 PRI	1462	CNR FF	R	F-11		2BN		2BN PLATOON NET	
F1 ALT	1463	CNR FF	R	O/O		2BN		2BN PLATOON NET	
F2 PRI	1464	CNR FF	R	F-21		2BN		2BN PLATOON NET	
F2 ALT	1465	CNR FF	R	O/O		2BN		2BN PLATOON NET	
F3 PRI	1466	CNR FF	R	F-31		2BN		2BN PLATOON NET	
F3 ALT	1467	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
GPRI CMD/BMS	1470	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
GALT VOICE	1471	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
G1 PRI	1472	CNR FF	R	G-11		2BN		2BN PLATOON NET	
G1 ALT	1473	CNR FF	R	O/O		2BN		2BN PLATOON NET	
G2 PRI	1474	CNR FF	R	G-21		2BN		2BN PLATOON NET	
G2 ALT	1475	CNR FF	R	O/O		2BN		2BN PLATOON NET	
G3 PRI	1476	CNR FF	R	G-31		2BN		2BN PLATOON NET	
G3 ALT	1477	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
HPRI CMD/BMS	1480	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
HALT VOICE	1481	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
H1 PRI	1482	CNR FF	R	H-11		2BN		2BN PLATOON NET	
H1 ALT	1483	CNR FF	R	O/O		2BN		2BN PLATOON NET	
H2 PRI	1484	CNR FF	R	H-21		2BN		2BN PLATOON NET	
H2 ALT	1485	CNR FF	R	O/O		2BN		2BN PLATOON NET	
H3 PRI	1486	CNR FF	R	H-31		2BN		2BN PLATOON NET	
H3 ALT	1487	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
IPRI CMD/BMS	1490	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
IALT VOICE	1491	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
I1 PRI	1492	CNR FF	R	I-11		2BN		2BN PLATOON NET	
I1 ALT	1493	CNR FF	R	O/O		2BN		2BN PLATOON NET	
I2 PRI	1494	CNR FF	R	I-21		2BN		2BN PLATOON NET	
I2 ALT	1495	CNR FF	R	O/O		2BN		2BN PLATOON NET	
I3 PRI	1496	CNR FF	R	I-31		2BN		2BN PLATOON NET	
I3 ALT	1497	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
JPRI CMD/BMS	1500	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
JALT VOICE	1501	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
J1 PRI	1502	CNR FF	R	J-11		2BN		2BN PLATOON NET	
J1 ALT	1503	CNR FF	R	O/O		2BN		2BN PLATOON NET	
J2 PRI	1504	CNR FF	R	J-21		2BN		2BN PLATOON NET	
J2 ALT	1505	CNR FF	R	O/O		2BN		2BN PLATOON NET	
J3 PRI	1506	CNR FF	R	J-31		2BN		2BN PLATOON NET	
J3 ALT	1507	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
KPRI CMD/BMS	1510	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
KALT VOICE	1511	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
K1 PRI	1512	CNR FF	R	K-11		2BN		2BN PLATOON NET	
K1 ALT	1513	CNR FF	R	O/O		2BN		2BN PLATOON NET	
K2 PRI	1514	CNR FF	R	K-21		2BN		2BN PLATOON NET	
K2 ALT	1515	CNR FF	R	O/O		2BN		2BN PLATOON NET	
K3 PRI	1516	CNR FF	R	K-31		2BN		2BN PLATOON NET	
K3 ALT	1517	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
LPRI CMD/BMS	1520	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
LALT VOICE	1521	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
L1 PRI	1522	CNR FF	R	L-11		2BN		2BN PLATOON NET	
L1 ALT	1523	CNR FF	R	O/O		2BN		2BN PLATOON NET	
L2 PRI	1524	CNR FF	R	L-21		2BN		2BN PLATOON NET	
L2 ALT	1525	CNR FF	R	O/O		2BN		2BN PLATOON NET	
L3 PRI	1526	CNR FF	R	L-31		2BN		2BN PLATOON NET	
L3 ALT	1527	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
MPRI CMD/BMS	1530	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
MALT VOICE	1531	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
M1 PRI	1532	CNR FF	R	M-11		2BN		2BN PLATOON NET	
M1 ALT	1533	CNR FF	R	O/O		2BN		2BN PLATOON NET	
M2 PRI	1534	CNR FF	R	M-21		2BN		2BN PLATOON NET	
M2 ALT	1535	CNR FF	R	O/O		2BN		2BN PLATOON NET	
M3 PRI	1536	CNR FF	R	M-31		2BN		2BN PLATOON NET	
M3 ALT	1537	CNR FF	R	O/O		2BN		2BN PLATOON NET	
2BN COY PRIMARY CMD VOICE/BMS									
NPRI CMD/BMS	1540	CNR TD	R	B-9		MAULKE BN		2BN COY PRIMARY CMD VOICE/BMS	
2BN COY ALTERNATE VOICE									
NALT VOICE	1541	CNR FF	R	O/O		MAULKE BN		2BN COY ALTERNATE VOICE	
2BN PLATOON NET									
N1 PRI	1542	CNR FF	R	N-					

22

Vedlegg B

SOI ENGRBN				2015 trenings-SOI		Nr _____		BEGRENSET	
Til bruk ved kurs/undervisning				Sikkerhetsloven § 11 og § 12 J/ Offentlighetsloven § 13.					
NET	NET ID	NET NAME	TYPE	Område	FRQ	ALTERNATIVE COMMS			
Hæren Brig N INGEN	2000	INGBN1	CHL_TD	Norge					
Hæren Brig N INGEN	2001	INGBN2	CHL_TD	Norge					
Hæren Brig N INGEN	2002	INGBN3	CHL_TD	Norge					
Hæren Brig N INGEN	2003	INGBN4	CHL_TD	Norge					
Hæren Brig N INGEN KP Samvirke	2090	INGBN_5_1	CHL_TD	Norge					
Hæren Brig N INGEN KP Samvirke	2091	INGBN_5_2	CHL_TD	Norge					
Hæren Brig N INGEN	2092	INGBN_R_1	CHL_TD	Norge					
Hæren Brig N INGEN	2093	INGBN_R_2	CHL_TD	Norge					
Hæren Brig N INGEN	2094	INGBN_R_3	CHL_TD	Norge					
Hæren Brig N INGEN	2095	INGBN_R_4	CHL_TD	Norge					
Hæren Brig N INGEN Pinglo A	2070	PINGPA1	CHL_TD	Norge					
Hæren Brig N INGEN Pinglo A	2071	PINGPA2	CHL_FF	Troms					
Hæren Brig N INGEN Pinglo A	2072	PINGPA3	CHL_FF	Troms					
Hæren Brig N INGEN Pinglo A	2073	PINGPA4	CHL_FF	Troms					
Hæren Brig N INGEN Pinglo A	2074	PINGPA5	CHL_FF	Troms					
Hæren Brig N INGEN Pinglo A	2075	PINGPA6	CHL_FF	Troms					
Hæren Brig N INGEN Inglo B	2080	INGPB1	CHL_TD	Norge					
Hæren Brig N INGEN Inglo B	2081	INGPB2	CHL_FF	Troms					
Hæren Brig N INGEN Inglo B	2082	INGPB3	CHL_FF	Troms					
Hæren Brig N INGEN Inglo B	2083	INGPB4	CHL_FF	Troms					
Hæren Brig N INGEN Inglo B	2084	INGPB5	CHL_FF	Troms					
Hæren Brig N INGEN Inglo B	2085	INGPB6	CHL_FF	Troms					
Hæren Brig N INGEN Pinglo C	2030	PINGPC1	CHL_TD	Norge					
Hæren Brig N INGEN Pinglo C	2031	PINGPC2	CHL_FF	Norge					
Hæren Brig N INGEN Pinglo C	2032	PINGPC3	CHL_FF	Norge					
Hæren Brig N INGEN Pinglo C	2033	PINGPC4	CHL_FF	Norge					
Hæren Brig N INGEN Pinglo C	2034	PINGPC5	CHL_FF	Norge					
Hæren Brig N INGEN INGP-4	2050	INGP4-1	CHL_TD	Norge					
Hæren Brig N INGEN INGP-4	2051	INGP4-2	CHL_FF	Troms					
Hæren Brig N INGEN INGP-4	2052	INGP4-3	CHL_FF	Troms					
Hæren Brig N INGEN INGP-4	2053	INGP4-4	CHL_FF	Troms					
Hæren Brig N INGEN INGP-4	2054	INGP4-5	CHL_FF	Troms					
Hæren Brig N INGEN M&KOP	2010	M&KOP1	CHL_TD	Norge					
Hæren Brig N INGEN M&KOP	2011	M&KOP2	CHL_FF	Norge					
Hæren Brig N INGEN M&KOP	2012	M&KOP3	CHL_FF	Norge					
Hæren Brig N INGEN INGP-6	2040	INGP6-1	CHL_TD	Norge					
Hæren Brig N INGEN INGP-6	2041	INGP6-2	CHL_FF	Troms					
Hæren Brig N INGEN INGP-6	2042	INGP6-3	CHL_FF	Troms					
Hæren Brig N INGEN INGP-6	2043	INGP6-4	CHL_FF	Troms					
Hæren Brig N INGEN INGP-6	2044	INGP6-5	REF_ONE	Troms					

Crypto information			
Key Type	Group-ID	Key-ID	Key-ID / SEPT / ORT
Traffic	2094	1234	

BRIG-N NET (interaction)				NET ID		NET NAME		TYPE		FRQ	
INTERACTION											
1. TILBEN SAMV											
2. TILBEN SAMV 1											
3. TILBEN SAMV 2											
4. TILBEN SAMV 3											
5. TILBEN SAMV 4											
6. TILBEN SAMV 5											
7. TILBEN SAMV 6											
8. TILBEN SAMV 7											
9. TILBEN SAMV 8											
10. TILBEN SAMV 9											
11. TILBEN SAMV 10											
12. TILBEN SAMV 11											
13. TILBEN SAMV 12											
14. TILBEN SAMV 13											
15. TILBEN SAMV 14											
16. TILBEN SAMV 15											
17. TILBEN SAMV 16											
18. TILBEN SAMV 17											
19. TILBEN SAMV 18											
20. TILBEN SAMV 19											
21. TILBEN SAMV 20											
22. TILBEN SAMV 21											
23. TILBEN SAMV 22											
24. TILBEN SAMV 23											
25. TILBEN SAMV 24											
26. TILBEN SAMV 25											
27. TILBEN SAMV 26											
28. TILBEN SAMV 27											
29. TILBEN SAMV 28											
30. TILBEN SAMV 29											
31. TILBEN SAMV 30											
32. TILBEN SAMV 31											
33. TILBEN SAMV 32											
34. TILBEN SAMV 33											
35. TILBEN SAMV 34											
36. TILBEN SAMV 35											
37. TILBEN SAMV 36											
38. TILBEN SAMV 37											
39. TILBEN SAMV 38											
40. TILBEN SAMV 39											
41. TILBEN SAMV 40											
42. TILBEN SAMV 41											
43. TILBEN SAMV 42											
44. TILBEN SAMV 43											
45. TILBEN SAMV 44											
46. TILBEN SAMV 45											
47. TILBEN SAMV 46											
48. TILBEN SAMV 47											
49. TILBEN SAMV 48											
50. TILBEN SAMV 49											
51. TILBEN SAMV 50											
52. TILBEN SAMV 51											
53. TILBEN SAMV 52											
54. TILBEN SAMV 53											
55. TILBEN SAMV 54											
56. TILBEN SAMV 55											
57. TILBEN SAMV 56											
58. TILBEN SAMV 57											
59. TILBEN SAMV 58											
60. TILBEN SAMV 59											
61. TILBEN SAMV 60											
62. TILBEN SAMV 61											
63. TILBEN SAMV 62											
64. TILBEN SAMV 63											
65. TILBEN SAMV 64											
66. TILBEN SAMV 65											
67. TILBEN SAMV 66											
68. TILBEN SAMV 67											
69. TILBEN SAMV 68											
70. TILBEN SAMV 69											
71. TILBEN SAMV 70											
72. TILBEN SAMV 71											
73. TILBEN SAMV 72											
74. TILBEN SAMV 73											
75. TILBEN SAMV 74											
76. TILBEN SAMV 75											
77. TILBEN SAMV 76											
78. TILBEN SAMV 77											
79. TILBEN SAMV 78											
80. TILBEN SAMV 79											
81. TILBEN SAMV 80											
82. TILBEN SAMV 81											
83. TILBEN SAMV 82											
84. TILBEN SAMV 83											
85. TILBEN SAMV 84											
86. TILBEN SAMV 85											
87. TILBEN SAMV 86											
88. TILBEN SAMV 87											
89. TILBEN SAMV 88											
90. TILBEN SAMV 89											
91. TILBEN SAMV 90											
92. TILBEN SAMV 91											
93. TILBEN SAMV 92											
94. TILBEN SAMV 93											
95. TILBEN SAMV 94											
96. TILBEN SAMV 95											
97. TILBEN SAMV 96											
98. TILBEN SAMV 97											
99. TILBEN SAMV 98											
100. TILBEN SAMV 99											
101. TILBEN SAMV 100											
102. TILBEN SAMV 101											
103. TILBEN SAMV 102											
104. TILBEN SAMV 103											
105. TILBEN SAMV 104											
106. TILBEN SAMV 105											
107. TILBEN SAMV 106											
108. TILBEN SAMV 107											
109. TILBEN SAMV 108											
110. TILBEN SAMV 109											
111. TILBEN SAMV 110											
112. TILBEN SAMV 111											
113. TILBEN SAMV 112											
114. TILBEN SAMV 113											
115. TILBEN SAMV 114											
116. TILBEN SAMV 115											
117. TILBEN SAMV 116											
118. TILBEN SAMV 117											
119. TILBEN SAMV 118											
120. TILBEN SAMV 119											
121. TILBEN SAMV 120											
122. TILBEN SAMV 121											
123. TILBEN SAMV 122											
124. TILBEN SAMV 123											
125. TILBEN SAMV 124											
126. TILBEN SAMV 125											
127. TILBEN SAMV 126											
128. TILBEN SAMV 127											
129. TILBEN SAMV 128											
130. TILBEN SAMV 129											
131. TILBEN SAMV 130											
132. TILBEN SAMV 131											
133. TILBEN SAMV 132											
134. TILBEN SAMV 133											
135. TILBEN SAMV 134											
136. TILBEN SAMV 135											
137. TILBEN SAMV 136											
138. TILBEN SAMV 137											
139. TILBEN SAMV 138											
140. TILBEN SAMV 139											
141. TILBEN SAMV 140											
142. TILBEN SAMV 141											
143. TILBEN SAMV 142											
144. TILBEN SAMV 143											
145. TILBEN SAMV 144											
146. TILBEN SAMV 145											
147. TILBEN SAMV 146											
148. TILBEN SAMV 147											
149. TILBEN SAMV 148											
150. TILBEN SAMV 149											
151. TILBEN SAMV 150											
152. TILBEN SAMV 151											
153. TILBEN SAMV 152											
154. TILBEN SAMV 153											
155. TILBEN SAMV 154											
156. TILBEN SAMV 155											
157. TILBEN SAMV 156											
158. TILBEN SAMV 157											
159. TILBEN SAMV 158											
160. TILBEN SAMV 159											
161. TILBEN SAMV 160											
162. TILBEN SAMV 161											
163. TILBEN SAMV 162											
164. TILBEN SAMV 163											
165. TILBEN SAMV 164											
166. TILBEN SAMV 165											
167. TILBEN SAMV 166											
168. TILBEN SAMV 167											
169. TILBEN SAMV 168											
170. TILBEN SAMV 169											
171. TILBEN SAMV 170											
172. TILBEN SAMV 171											
173. TILBEN SAMV 172											
174. TILBEN SAMV 173											
175. TILBEN SAMV 174											
176. TILBEN SAMV 175											
177. TILBEN SAMV 176											
178. TILBEN SAMV 177											
179. TILBEN SAMV 178											
180. TILBEN SAMV 179											
181. TILBEN SAMV 180											
182. TILBEN SAMV 181											
183. TILBEN SAMV 182											
184. TILBEN SAMV 183											
185. TILBEN SAMV 184											
186. TILBEN SAMV 185											
187. TILBEN SAMV 186											
188. TILBEN SAMV 187											
189. TILBEN SAMV 188											
190. TILBEN SAMV 189											
191. TILBEN SAMV 190											
192. TILBEN SAMV 191											
193. TILBEN SAMV 192											
194. TILBEN SAMV 193											
195. TILBEN SAMV 194											
196. TILBEN SAMV 195											
197. TILBEN SAMV 196											
198. TILBEN SAMV 197											
199. TILBEN SAMV 198											
200. TILBEN SAMV 199											
201. TILBEN SAMV 200											
202. TILBEN SAMV 201											
203. TILBEN SAMV 202											
204. TILBEN SAMV 203											
205. TILBEN SAMV 204											
206. TILBEN SAMV 205											
207. TILBEN SAMV 206											
208. TILBEN SAMV 207											
209. TILBEN SAMV 208											
210. TILBEN SAMV 209											
211. TILBEN SAMV 210											
212. TILBEN SAMV 211											
213. TILBEN SAMV 212											
214. TILBEN SAMV 213											
215. TILBEN SAMV 214											
216. TILBEN SAMV 215											
217. TILBEN SAMV 216											
218. TILBEN SAMV 217											
219. TILBEN SAMV 218											
220. TILBEN SAMV 219											
221. TILBEN SAMV 220											
222. TILBEN SAMV 221											
223. TILBEN SAMV 222											
224. TILBEN SAMV 223											
225. TILBEN SAMV 224											
226. TILBEN SAMV 225											
227. TILBEN SAMV 226											
228. TILBEN SAMV 227											
229. TILBEN SAMV 228											
230. TILBEN SAMV 229											
231. TILBEN SAMV 230											
232. TILBEN SAMV 231											
233. TILBEN SAMV 232											
234. TILBEN SAMV 233											
235. TILBEN SAMV 234											
236. TILBEN SAMV 235											
237. TILBEN SAMV 236											
238. TILBEN SAMV 237											
239. TILBEN SAMV 238											
240. TILBEN SAMV 239											
241. TILBEN SAMV 240											
242. TILBEN SAMV 241											
243. TILBEN SAMV 242											
244. TILBEN SAMV 243											
245. TILBEN SAMV 244											
246. TILBEN SAMV 245											
247. TILBEN SAMV 246											
248. TILBEN SAMV 247											
249. TILBEN SAMV 248											
250. TILBEN SAMV 249											
251. TILBEN SAMV 250											
252. TILBEN SAMV 251											
253. TILBEN SAMV 252											
254. TILBEN SAMV 253											
255. TILBEN SAMV 254											
256. TILBEN SAMV 255											
257. TILBEN SAMV 256											
258. TILBEN SAMV 257											
259. TILBEN SAMV 258											
260. TILBEN SAMV 259											
261. TILBEN SAMV 260											
262. TILBEN SAMV 261											
263. TILBEN SAMV 262											
264. TILBEN SAMV 263											
265. TILBEN SAMV 264											
266. TILBEN SAMV 265											
267. TILBEN SAMV 266											
268. TILBEN SAMV 267											
269. TILBEN SAMV 268											
270. TILBEN SAMV 269											
271. TILBEN SAMV 270											
272. TILBEN SAMV 271											
273. TILBEN SAMV 272											
274. TILBEN SAMV 273											
275. TILBEN SAMV 274											
276. TILBEN SAMV 275											
277. TILBEN SAMV 276											
278. TILBEN SAMV 277											
279. TILBEN SAMV 278											
280. TILBEN SAMV 279											
281. TILBEN SAMV 280											
282. TILBEN SAMV 281											



CALLSIGN HF INFRING		
Infraet	Call	Callign
NR VO	NR000	NR000
QVS DEVIDS	NR000DEVIDS	
BN SEF	NR01A	
NR VO	NR01A	
Hillingen	NR05A2	
PING 1 CP	NR10CP	
PING 1 KP SEF	NR11A	
RETR_KO_Lag 1	NR12A	
RETR_Lag 1	NR161	
PING 2 CP	NR162	
PING 2 CP	NR20CP	
PING 2 KP SEF	NR21A	
RETR_KO	NR22A	
RETR_KO_Lag 1	NR221	
RETR_KO_Lag 2	NR222	
BRIG REK TR_KO	NR24A	
BRIG REK_Lag 1	NR241	
BRIG REK_Lag 2	NR242	
NR VO	NR25A	
Dyckerdag	NR231	
PING 3 CP	NR30CP	
PING 3 KP SEF	NR31A	
RETR_KO	NR32A	
RETR_KO_Lag 1	NR361	
RETR_KO_Lag 2	NR362	
PING 4 CP	NR40A	
PING 4 KP SEF	NR41A	
MAKIN TR_KO	NR41A	
DVS TR_KO	NR42A	
FLUCH	NR43A	
FLUCH	NR44A	
PING 5 CP	NR50CP	
PING 5 KP SEF	NR51A	
MAKIN TR_Lag 1	NR52A	
MAKIN TR_Lag 2	NR53A	
DVS Tropp	NR54A	

WINGMAN CODEWORDLIST									
Wingmanword for type of codewords:									
AOSAM									
Supplies	Actions	Nationality	Letters	Comms	Weapon	Equipment	Miscellaneous		
111A Battery (C2C)	313 Establish	PRN CAN	67C A	24E RA 5500	028 12.7mm mach gun	812 Medicine			
111B Battery for LFR	313B Paculate	PRN CAN	67C B	24E RA 5500	028 12.7mm mach gun	812 Medicine			
111C Battery for MMR	313C Fire-ward	PRN ESP	68B C	20Y Antenna HF, Portable	060 AG-3	881 Supply commo			
111D FIM (filter for DMR)	313D Fire support	PRN FIN	68B D	12Y Antenna LFR	100 Hand grenade	851 Unloading post			
111E FIM (filter for DMR)	313E FIM/Stop	PRN GER	68B E	20Y Antenna MMR stationary	071 F-6	851 Unloading post			
111F Food (DOS)	313F Have observed	PRN HUN	67C F	23E Antenna MMR, portable	242 IED				
111G Gasoline (LFR)	313G IED	PRN ITA	67C G	313F RASBAT Tactical	010 M100				
111H Lightsticks (es)	313H Meet in grid	PRN JDR	68B H	313C NORMS (Pacvar)	072 MG-3				
111I Lubricant BA 5500 (es)	313I Need recovery	PRN LAT	69H I	24E RA 5500	028 12.7mm mach gun	812 Medicine			
112 Toilet paper	313J Move	PRN NED	67C J	21E HF	072 MG-3				
112L Water (Star)	313J Need repair support	PRN NOR	68B K	313C IED	072 MG-3				
112M Penlight/LFR BA 3058 (es)	313M Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112N Penlight/LFR BA 3058 (es)	313N Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112O Penlight/LFR BA 3058 (es)	313O Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112P Penlight/LFR BA 3058 (es)	313P Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112Q Penlight/LFR BA 3058 (es)	313Q Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112R Penlight/LFR BA 3058 (es)	313R Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112S Penlight/LFR BA 3058 (es)	313S Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112T Penlight/LFR BA 3058 (es)	313T Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112U Penlight/LFR BA 3058 (es)	313U Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112V Penlight/LFR BA 3058 (es)	313V Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112W Penlight/LFR BA 3058 (es)	313W Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112X Penlight/LFR BA 3058 (es)	313X Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112Y Penlight/LFR BA 3058 (es)	313Y Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
112Z Penlight/LFR BA 3058 (es)	313Z Need supplies	PRN NOR	68B L	313C IED	072 MG-3				
000 9 mm cartridge	313R Recon arrow	PRN UKR	68B O	27C PC	060 Recon arrow	851 Unloading post			
001 30mm MP	313R Recon arrow	PRN UKR	68B O	27C PC	060 Recon arrow	851 Unloading post			
002 5.56 cartridge	313R Recon arrow	PRN UKR	68B O	27C PC	060 Recon arrow	851 Unloading post			
003 7.62 cartridge	313R Recon arrow	PRN UKR	68B O	27C PC	060 Recon arrow	851 Unloading post			
004 2.62 cartridge for MG	313R SH RP	010 C, 2	313B	Unit	060	851			
005 9 mm cartridge	313R Station pistol	010 C, 2	313B	Unit	060	851			
006 Handbang	313R Switching of comms	010 C, 2	313B	Unit	060	851			
007 Hand Grenade, fragment	313R Switching of comms	010 C, 2	313B	Unit	060	851			
008 Hand Grenade, stun	313R Troops in contact	020 C, 2	313B	Unit	060	851			
009 High-explosive shell	313R VCP	020 C, 2	313B	Unit	060	851			
010 Illuminated flare	313R You must	020 C, 2	313B	Unit	060	851			
020 M72	313R Station pistol	010 C, 2	313B	Unit	060	851			
030 Mortar illumination	313R Station pistol	010 C, 2	313B	Unit	060	851			
040 Mortar illumination	313R Station pistol	010 C, 2	313B	Unit	060	851			
050 Signal Alarm (Coloured)	313R Station pistol	010 C, 2	313B	Unit	060	851			
060 Smoke grenade	313R Station pistol	010 C, 2	313B	Unit	060	851			
Numbers				A1E ATV					
352 0				A1E AVLB					
353 1				A1E B					
354 2				A1E B					
355 3				A1E B					
356 4				A1E B					
357 5				A1E B					
358 6				A1E B					
359 7				A1E B					
360 8				A1E B					
361 9				A1E B					
362 10				A1E B					
363 11				A1E B					
364 12				A1E B					
365 13				A1E B					
366 14				A1E B					
367 15				A1E B					
368 16				A1E B					
369 17				A1E B					
370 18				A1E B					
371 19				A1E B					
372 20				A1E B					
373 21				A1E B					
374 22				A1E B					
375 23				A1E B					
376 24				A1E B					
377 25				A1E B					
378 26				A1E B					
379 27				A1E B					
380 28				A1E B					
381 29				A1E B					
382 30				A1E B					
383 31				A1E B					
384 32				A1E B					
385 33				A1E B					
386 34				A1E B					
387 35				A1E B					
388 36				A1E B					
389 37				A1E B					
390 38				A1E B					
391 39				A1E B					
392 40				A1E B					
393 41				A1E B					
394 42				A1E B					
395 43				A1E B					
396 44				A1E B					
397 45				A1E B					
398 46				A1E B					
399 47				A1E B					
400 48				A1E B					
401 49				A1E B					
402 50				A1E B					
403 51				A1E B					
404 52				A1E B					
405 53				A1E B					
406 54				A1E B					
407 55				A1E B					
408 56				A1E B					
409 57				A1E B					
410 58				A1E B					
411 59				A1E B					
412 60				A1E B					
413 61				A1E B					
414 62				A1E B					
415 63				A1E B					
416 64				A1E B					
417 65				A1E B					
418 66				A1E B					
419 67				A1E B					
420 68				A1E B					
421 69				A1E B					
422 70				A1E B					
423 71				A1E B					
424 72				A1E B					
425 73				A1E B					
426 74				A1E B					
427 75				A1E B					
428 76				A1E B					
429 77				A1E B					
430 78				A1E B					
431 79				A1E B					
432 80				A1E B					
433 81				A1E B					
434 82				A1E B					
435 83				A1E B					
436 84				A1E B					
437 85				A1E B					
438 86				A1E B					
439 87				A1E B					
440 88				A1E B					
441 89				A1E B					
442 90				A1E B					
443 91				A1E B					
444 92				A1E B					
445 93				A1E B					
446 94				A1E B					
447 95				A1E B					
448 96				A1E B					
449 97				A1E B					
450 98				A1E B					
451 99				A1E B					
452 100				A1E B					
453 101				A1E B					
454 102				A1E B					
455 103				A1E B					
456 104				A1E B					
457 105				A1E B					
458 106				A1E B					
459 107				A1E B					
460 108				A1E B					
461 109				A1E B					
462 110				A1E B					
463 111				A1E B					
464 112				A1E B					
465 113				A1E B					
466 114				A1E B					
467 115				A1E B					
468 116				A1E B					
469 117				A1E B					
470 118				A1E B					
471 119				A1E B					
472 120				A1E B					
473 121				A1E B					
474 122				A1E B					
475 123				A1E B					
476 124				A1E B					
477 125				A1E B					
478 126				A1E B					
479 127				A1E B					
480 128				A1E B					
481 129				A1E B					
482 130				A1E B					
483 131				A1E B					
484 132				A1E B					
485 133				A1E B					
486 134				A1E B					
487 135				A1E B					
488 136				A1E B					
489 137				A1E B					
490 138				A1E B					
491 139				A1E B					
492 140				A1E B					
493 141				A1E B					
494 142				A1E B					
495 143				A1E B					
496 144				A1E B					
497 145				A1E B					
498 146				A1E B					
499 147				A1E B					
500 148				A1E B					
501 149				A1E B					
502 150				A1E B					
503 151				A1E B					
504 152				A1E B					
505 153				A1E B					
506 154				A1E B					
507 155				A1E B					
508 156				A1E B					
509 157				A1E B					
510 158				A1E B					
511 159				A1E B					
512 160				A1E B					
513 161				A1E B					
514 162				A1E B					
515 163				A1E B					
516 164				A1E B					
517 165				A1E B					
518 166				A1E B					
519 167				A1E B					
520 168				A1E B					
521 169				A1E B					
522 170				A1E B					
523 171				A1E B					
524 172				A1E B					
525 173				A1E B					
526 174				A1E B					
527 175				A1E B					
528 176				A1E B					
529 177				A1E B					
530 178				A1E B					
531 179				A1E B					
532 180				A1E B					
533 181				A1E B					
534 182				A1E B					
535 183				A1E B					
536 184				A1E B					
537 185				A1E B					
538 186				A1E B					
539 187				A1E B					
540 188				A1E B					
541 189				A1E B					
542 190				A1E B					
543 191				A1E B					
544 192				A1E B					
545 193				A1E B					
546 194				A1E B					
547 195				A1E B					
548 196				A1E B					
549 197				A1E B					
550 198				A1E B					
551 199				A1E B					
552 200				A1E B					
553 201				A1E B					
554 202				A1E B					
555 203				A1E B					
556 204				A1E B					
557 205				A1E B					
558 206				A1E B					
559 207				A1E B					
560 208				A1E B					
561 209				A1E B					
562 210				A1E B					
563 211				A1E B					
564 212				A1E B					
565 213				A1E B					
566 214				A1E B					
567 215				A1E B					
568 216				A1E B					
569 217				A1E B					
570 218				A1E B					
571 219				A1E B					
572 220				A1E B					
573 221				A1E B					
574 222				A1E B					
575 223				A1E B					
576 224				A1E B					
577 225				A1E B					
578 226				A1E B					
579 227				A1E B					
580 228				A1E B					
581 229				A1E B					
582 230				A1E B					
583 231				A1E B					
584 232				A1E B					
585 233				A1E B					
586 234				A1E B					
587 235				A1E B					
588 236				A1E B					
589 237				A1E B					
590 238				A1E B					
591 239				A1E B					
592 240				A1E B					
593 241				A1E B					
594 242				A1E B					
595 243				A1E B					
596 244				A1E B					
597 245				A1E B					
598 246				A1E B					
599 247				A1E B					
600 248				A1E B					
601 249				A1E B					
602 250				A1E B					
603 251				A1E B					
604 252				A1E B					
605 253				A1E B					
606 254				A1E B					
607 255				A1E B					
608 256				A1E B					
609 257				A1E B					
610 258				A1E B					
611 259				A1E B					
612 260				A1E B					
613 261				A1E B					
614 262				A1E B					
615 263									

Vedlegg F

Prosjektavtale

Prosjektavtale inngått mellom NTNU Fakultet for informasjonsteknologi og elektroteknikk (IE) på Gjøvik, Opplandske Heimvernsdistrikt 05, G6-seksjonen, og gruppe-medlemmer. Merk at avtalen mangler signatur fra instituttleder/faggruppeleder, da dette ble gjort digitalt.

Prosjektavtale

mellom NTNU Fakultet for informasjonsteknologi og elektroteknikk (IE) på Gjøvik (utdanningsinstitusjon), og

OPPLANDSKE HEIMEVERNSDISTRIKT 05

GB - SEKSJONEN

(oppdragsgiver), og

PETTER SAGVOLD, ANDERS H. REBNER,

NIKOLAI MØRK, THOMAS LØKKEBORG

(student(er))

Avtalen angir avtalepartenes plikter vedrørende gjennomføring av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer:

1. Studenten(e) skal gjennomføre prosjektet i perioden fra 08.09.20 til 20.05.20.

Studentene skal i denne perioden følge en oppsatt fremdriftsplan der NTNU IE på Gjøvik yter veiledning. Oppdragsgiver yter avtalt prosjektbistand til fastsatte tider. Oppdragsgiver stiller til rådighet kunnskap og materiale som er nødvendig for å få gjennomført prosjektet. Det forutsettes at de gitte problemstillinger det arbeides med er aktuelle og på et nivå tilpasset studentenes faglige kunnskaper. Oppdragsgiver plikter på forespørsel fra NTNU å gi en vurdering av prosjektet vederlagsfritt.

2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:

- Oppdragsgiver dekker selv gjennomføring av prosjektet når det gjelder f.eks. materiell, telefon, reiser og nødvendig overnatting på steder langt fra NTNU i Gjøvik. Studentene dekker utgifter for ferdigstillelse av prosjektmateriell.
- Eiendomsretten til eventuell prototyp tilfaller den som har betalt komponenter og materiell mv. som er brukt til prototypen. Dersom det er nødvendig med større og/eller spesielle investeringer for å få gjennomført prosjektet, må det gjøres en egen avtale mellom partene om eventuell kostnadsfordeling og eiendomsrett.

3. NTNU IE på Gjøvik står ikke som garantist for at det oppdragsgiver har bestilt fungerer etter hensikten, ei heller at prosjektet blir fullført. Prosjektet må anses som en eksamensrelatert oppgave som blir bedømt av intern og ekstern sensor. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.

4. Alle beståtte bacheloroppgaver som ikke er klausulert og hvor forfatteren(e) har gitt sitt samtykke til publisering, kan gjøres tilgjengelig via NTNUs institusjonelle arkiv NTNU Open.

Tilgjengeliggjøring i det åpne arkivet forutsetter avtale om delvis overdragelse av opphavsrett, se «avtale om publisering» (jfr Lov om opphavsrett). Oppdragsgiver og veileder godtar slik offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og instituttleder/fagenhetsleder om de i løpet av prosjektet endrer syn på slik offentliggjøring.

Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode mv. som inngår som del av eller vedlegg til besvarelsen, kan vederlagsfritt benyttes til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av NTNU til andre formål, og ikke overlates til utenforstående uten etter avtale med de øvrige parter i denne avtalen. Dette gjelder også firmaer hvor ansatte ved NTNU og/eller studenter har interesser.

5. Besvarelsens spesifikasjoner og resultat kan anvendes i oppdragsgivers egen virksomhet. Gjør studenten(e) i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom oppdragsgiver og student(er) bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.
6. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten(e) ikke rett til å publisere sin besvarelse, det være seg helt eller delvis eller som del i annet arbeide, uten samtykke fra oppdragsgiver. Tilsvarende samtykke må foreligge i forholdet mellom student(er) og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.
7. Studenten(e) leverer oppgavebesvarelsen med vedlegg (pdf) i NTNUs elektroniske eksamenssystem. I tillegg leveres ett eksemplar til oppdragsgiver.
8. Denne avtalen utferdiges med ett eksemplar til hver av partene. På vegne av NTNU, IE er det instituttleder/faggruppeleder som godkjenner avtalen.
9. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og NTNU som regulerer nærmere forhold vedrørende bl.a. eiendomsrett, videre bruk, konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene. Dersom oppdragsgiver og student(er) ønsker en videre eller ny avtale med oppdragsgiver, skjer dette uten NTNU som partner.
10. Når NTNU også opptrer som oppdragsgiver, trer NTNU inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.
11. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene imellom. Dersom det ikke oppnås enighet, er partene enige om at tvisten løses av voldgift, etter bestemmelsene i tvistemålsloven av 13.8.1915 nr. 6, kapittel 32.

12. Deltakende personer ved prosjektgjennomføringen:

NTNUs veileder (navn): Frode Haug

Oppdragsgivers kontaktperson (navn): Espen Torseth

Student(er) (signatur): Petter Segvold dato 20/1-20

Nikolai Mark dato 20/1-20

Thomas Tellebooy dato 20/1-20

Anders H. Rebner dato 20/1-20

Oppdragsgiver (signatur): Leif E. Vanum dato 17/1-20

Signert avtale leveres digitalt i Blackboard, rom for bacheloroppgaven.

Godkjennes digitalt av instituttleder/faggrupeleder.

Om papirversjon med signatur er ønskelig, må papirversjon leveres til instituttet i tillegg.

Plass for evt sign:

Instituttleder/faggrupeleder (signatur): _____ dato _____

Vedlegg G

Statusrapporter

Vedlegget består av alle statusrapportene som ble skrevet gjennom hele prosjektperioden. Statusrapporter er korte rapporter levert til veileder for å gi innsyn i hvor langt prosjektet har kommet.

Bachelor - Statusrapport 1

Nikolai Mork
BIDAT

Thomas Løkkeborg
BITSEC

Anders H. Rebner
BIDAT

Petter Sagvold
BIDAT

Februar 2020

Status

Prosessmessig går oppgaven etter planen. De fleste på gruppen møter klokken 9 hver dag bortsett fra tirsdag, da hele gruppen har andre emner. Vi har fulgt Gantt-diagrammet vårt så langt. Vi jobber som regel ilag på skolen. Omfattende referater blir skrevet for alle møter, og vi prøver å notere ned valg vi gjør så de kan utdypes i rapporten. På rapporten har vi begynt å skissere ned de tankene vi har, da særlig rundt designprosessen og research-prosessen som nå er fullført. Vi har til dømes argumentert for valg av teknologier. Diverse argumentasjon for valg som er tatt står også i egne dokumenter som ennå ikke er flyttet over til rapporten.

Vi har gjennomført en research-periode på to uker hvor vi først i plenum listet opp alle områdene vi visste ville kreve nærmere undersøkelse. Deretter plukket vi problemområder fortløpende fra listen. Mye tid gikk her på å lete gjennom diverse teknologier og vurdere dem opp mot hverandre. Områder som vi vet vil være svært omfattende og sentrale i oppgaven slik som brukergrensesnitt brukte alle tid på å sette seg inn i. På ett aspekt av programvaren har vi flere gode veier å velge mellom teknologimessig, så her har vi ikke bestemt oss helt. Det gjelder generering og forhåndsvisning av PDF i Python. Kort fortalt kan vi enten bruke funksjonalitet innebygd i GUI-biblioteket vi har valgt, Qt, for å håndtere dette, eller vi kan ta i bruk et helt eget bibliotek, Reportlab. Begge veier har oppsider og nedsider.

Vi hadde et møte med Jon Tjøstel Ree-Lindstad som er S6 i Grebe på hans arbeidsplass Norsk Tipping på Hamar. Han har ansvaret for å motta og hente data ut av SOIer fra høyere enheter for så å produsere nye SOIer basert på disse som han distribuerer videre nedover i hierarkiet. Jon ga oss et innblikk i sin arbeidssituasjon og sine vaner når han lager SOIer. Han fortalte om en rekke ønsker og visjoner for programmet. Siden han ikke viste omfanget av oppgaven

spesielt rundt tid vil det diskuteres med Espen hvor mye som kan taes med. Vi hadde med en papir-mockup for å vise Jon tankene våre rundt brukergrensesnittet. Han likte godt det han så, men hadde noen innvendinger på hvordan han foretrakk å gjøre ting. Nok en gang er dette noe som vurderes. Siden det ikke er noe verktøy for å lage SOI idag har "alle" sine måter å gjøre ting på.

I design-perioden jobbet vi sammen for å designe hvert designområde. I design av brukergrensesnittet tegnet vi skisser hver for oss som vi diskurtete i etterkant. Deretter kjørte vi en ny iterasjon hvor vi tegnet og diskuterte før vi samlet det til en skisse. Her har vi kommet fram til to design på brukergrensesnitt, hvor det ene er enklere for oss å programmere, men mindre brukervennlig. Vi velger å ha denne i "backup", og planlegger å dokumentere begge i rapporten. Vi har laget mockup på begge alternativene. På de andre designområdene diskuterte vi i plenum og kom fram til løsninger.

Kravspesifikasjonen, programvarearkitekturen og oppsett av utviklingsmiljøet er elementer som er under arbeid. Programvarearkitekturen skulle egentlig vært ferdig rundt denne tiden, men henger etter 1-2 dager. Begrunnelsen for dette kommer under.

Når det gjelder tidsfristene som er satt i Gantt, henger gruppen etter med 2-3 dager. Dette fordi det var noen elementer innenfor design vi måtte vente til etter møtet med Jon. Derfor ble designperiodens tidsfrister litt overskredet. Ellers så har alle tidsfrister blitt overholdt.

Vi avslutter nå forarbeidet og vil begynne å programmere selve applikasjonen.

Totalstatus

Gruppen er fornøyd med innsatsen så langt. Vi har gjennomført research og design samt hatt et møte med Jon fra HV hvor vi viste mockup. Vi er tilnærmet i rute og føler at vi har kommet godt i gang.

Muligheter? Trusler/Problemer?

Gjennom perioden har det dukket opp noen få problemer for gruppen. Det første problemet er at oppdragsgiveren er veldig opptatt, noe som gjør det vanskelig å kommunisere med personen med tanke på spørsmål og tanker rundt oppgaven. Det andre problemet er at vi skulle få use caser for å bruke opp mot applikasjonen, noe som blir vanskelig å få pga. det som blir nevnt over. Grunnbildet av applikasjonen er klart, men det er detaljer som blir avduket etter hvert som oppgaven utvikler seg.

Motivasjon

Gruppen samarbeider godt og effektivt sammen. Vi jobber for det meste sammen på skolen, der vi kan diskutere og argumentere for ulike valg for applikasjonen. Under teknologifasen jobbet de ulike gruppemedlemmene med valgene og lærte seg disse. Dette for å gjøre utviklingsfasen enklere, da man kan mye av teknologien man har valgt. På noen av valgene er det noen gruppemedlemmer som har litt mer spisset kompetanse enn de andre på gruppen, siden de brukte litt mer tid på noen av valgene.

Forhold som oppmuntrer gruppen er når våre teknologivalg støtter funksjonaliteten applikasjonen vår trenger. Ellers er det ingen forhold som frustrerer gruppen enda.

Veilederkontakt

Når det gjelder kontakten mellom gruppen vår og veileder, fungerer den veldig godt. Vi hadde møter hver uke med veileder i starten av bachelorperioden, noe som var til stor hjelp da vi ikke var sikre på hvordan man skulle angripe oppgaven. Veileder kommer med gode og hjelpsomme tilbakemeldinger som er mye til hjelp for gruppen. Alt i alt er gruppen veldig fornøyd med veilederkontakten.

Bachelor - Statusrapport 2

Nikolai Mork
BIDAT

Thomas Løkkeborg
BITSEC

Anders H. Rebner
BIDAT

Petter Sagvold
BIDAT

Mars 2020

Status

Per nå følger vi Gantt-diagrammet rimelig godt, men vi begynner å lure på om vi ikke ga oss selv nok tid til ferdigstilling. Dette skyldes nok manglende erfaring. Både produktet og rapporten begynner å ta form. Produktet er nå brukbart men mangler mengde i innhold, da spesifikt variasjonen i SOI-moduler man kan legge til. Rapporten bærer preg av usikkerhet da vi ikke vet helt hvor vi skal plassere forskjellige bolker med tekst, samt at det er svært tynt i både kapittelet om implementasjon og ”deployment”. Vi har jobbet en del med å få kravspesifikasjonen ned på papir. Det er en god stund siden vi kommuniserte med kontaktperson (Espen), men vi har fått ordnet et møte med han denne uken. Vi frykter at vi ikke har samme tankegang rundt kravspesifikasjonen og at kontaktpersonen er uenig i det vi har produsert. Vi har også sendt mockups til arbeidsgiver, som vi har ikke fått noen tilbakemelding på.

Programvaren har nå støtte for:

- Visning, utskrift og redigering av kodebok
- Automatisk oppdatering av kodebok både på tid og etter brukers ønske
- Visning av SOI
- Legge til modul til SOI
- Automatisk plassering av moduler
- Serialisering og komprimering av SOI
- Endre oppsett til SOI
- Eksport til PDF
- 3 av 10 SOI-moduler, hvor noen av de gjenstående modulene er betydelig mer kompliserte enn de vi har gjort så langt

Muligheter? Trusler/Problemer?

Vi følger FHIs råd og jobber derfor hjemmefra. Dette går relativt greit, men har sine ulemper. Diskusjoner med gruppa kan ta lengre tid da det krever at gruppedlemmer er logget på og leser meldingene på de forskjellige kanalene. Det som kunne ha blitt gjort når skolen stengte, var at gruppen kunne ha satt opp et samarbeidsområde slik at all diskusjon kunne ha vært på samme kanal. I Git-Lab bruker vi diskusjonsmuligheten til "Merge Requests" og i Overleaf hvor vi skriver rapporten bruker vi kommenteringsverktøyet. Mandagsmøtene blir fortsatt gjennomført, men nå blir de gjennomført over Discord. På disse møtene går vi gjennom arbeidstavlen på Trello og gjør opp status og planlegger nærmeste fremtid. I tillegg har vi discordmøter der det er nyttig som for eksempel ved problemer som må diskuteres så mye at det ikke er hensiktsmessig å bruke tekst. I praksis snakkes vi ca. annenhver dag over Discord. Denne løsningen gikk litt tregt i starten, men har blitt mer innarbeidet på dette tidspunktet. Før kunne vi bare dulte borti hverandre og ta diskusjoner fortløpende.

Om den nasjonale situasjonen ikke forbedres ser vi det som et utfordring å få gjennomført brukertester. Dette er ikke heldig situasjon da brukergrensesnittet har en sentral rolle i vår applikasjon. Situasjonen gjør det også litt mer tungvindt å hjelpe hverandre da vi ikke jobber sammen.

Motivasjon

Motivasjonen til gruppen er høy, men uklarthet på rapportskriving påvirker fremdriften.

Veilederkontakt

I den siste perioden har vi brukt mesteparten av tiden til programmering, følgelig har det da blitt mindre tid til å skrive rapport. Dermed har vi ikke sett det som nødvendig å kontakte veileder. Vi ser for oss hyppigere kontakt med både veileder og kontaktperson nå fram mot innlevering. Nå jobber vi med å legge til moduler til programvaren samt skrive det på rapporten som krever tilbakemelding fra både veileder og kontaktperson.

Totalstatus

Det gjenstår fortsatt noen solide arbeidstimer, men gruppen er komfortabel med situasjonen både arbeidsmessig og tidsmessig. Ved rapporten er det noe usikkerhet som vil bli avklart gjennom samtale med veileder. Gruppen er med godt mot.

Bachelor - Statusrapport 3

Nikolai Mork
BIDAT

Thomas Løkkeborg
BITSEC

Anders H. Rebner
BIDAT

Petter Sagvold
BIDAT

April 2020

Status

Vi følger fortsatt Gantt-diagrammet rimelig godt, men på noen steder har vi glidd litt ut. Dette henger sammen med at vi fant en del ting i programvaren som var viktig å få fikset. Produktet og rapporten har tatt mer form siden siste statusrapport. Produktet har fått mer funksjonalitet og flere moduler, noe vi manglet ved forrige statusrapport. Medlemmene i gruppa begynner å bytte fra å programmere til å skrive på rapporten. Dette skyldes at medlemmene blir ferdige med sine oppgaver når det gjelder programmeringen. Så hvor mye som skrives på rapporten vil øke fram til det bare er rapporten det jobbes på. Programvaren har nå støtte for:

- Visning, utskrift og redigering av kodebok
- Automatisk oppdatering av kodebok både på tid og etter brukers ønske
- Visning av SOI
- Legge til modul til SOI
- Automatisk plassering av moduler
- Serialisering og komprimering av SOI
- Endre oppsett til SOI
- Eksport til PDF
- 8 av 10 SOI-moduler

Muligheter? Trusler/Problemer?

Vi følger fortsatt myndighetenes råd og fortsetter å jobbe hjemmefra. Vi har fortsatt våre ukentlige mandagsmøter over Discord hvor vi diskuterer ulike temaer rundt prosjektet. Der bruker vi Trello og gjør opp status og planlegger videre jobbing.

Da den nasjonale situasjonen mest sannsynlig ikke forbedres før innlevering, får vi ikke gjennomført brukertester av målgruppen. Dette må tas i betraktning og diskuteres om i rapporten. Det er fortsatt litt vanskelig å hjelpe hverandre når vi ikke jobber sammen, men med litt tålmodighet går det greit.

Gruppen fikk nettopp tips av oppdragsgiver om en stor forbedring vi kan gjøre på programvarearkitekturen. Dette kan ses på som både en trussel og en mulighet. Trussel hvis vi synker mye tid i å implementere det, og mulighet hvis vi får implementert det relativt fort. Vi vil avslutte programmering så fort vi kan, så mest sannsynlig vil vi velge å ikke implementere forbedringen.

Motivasjon

Når det gjelder motivasjonen på gruppa ligger den mellom høy og middels.

Veilederkontakt

Da koden nærmer seg fullføring, vil mye av tiden framover brukes til rapport-skriving. Det vil derfor bli avtalt flere møter med veileder slik at vi kan avklare ulike elementer vi er usikre med rapporten. Små ting med rapporten som vi trenger hjelp med vil bli sendt på mail. Hadde nylig et lite møte med kontaktperson, hvor vi fikk mange gode tilbakemeldinger.

Totalstatus

Selv om mye av koden så og si er ferdig, gjenstår det fortsatt noen solide arbeidstimer. Tilbakemeldingen vi fikk fra kontaktpersonen ga gruppa en liten boost som vil være til hjelp i innspurten nå. Gruppa er ved godt mot.

Vedlegg H

Møtereferater

Vedlegget inneholder referater fra alle møtene gjort gjennom prosjektperioden. Interne møter med gruppemedlemmer, møter med veileder og møter med kontaktperson står alle i dette vedlegget.

Konvensjonen for referater har vært at hvert møte får egen overskrift i fet skrift som kort beskrivever hva møtet omdler, deretter følger dato, og til slutt nevnes i parentes hvem som har skrevet referatet. Selve referatet gjøres i punktlistor for effektivitet, og komplett innhold prioriteres over rettskriving.

Møte med Frode 08.01.2020 (Thomas)

- Første møte
- Oppklaring i hvordan samarbeid med veileder vil foregå:
 - Ukentlig møte. Bestemt til å være mandager klokken 08:30, med unntak av uke 3 hvor møtet holdes på tirsdag 10:00. Disse møtene fortsetter vi med så lenge det er hensiktsmessig
 - Veileder vil fungere som "prosessveileder". Kan ikke nødvendigvis hjelpe på det tekniske
 - Det skal leveres tre "statusrapporter" til veileder ila. semesteret. Foreløpige innleveringsfrister satt til: 20.02.2020, 01.04.2020, 01.05.2020
 - Dokumenter vi vil veileder skal ta en kikk på må sendes 24 timer før
- Tips fra veileder:
 - Del to av prosjektplanen ("omfang") burde skrives så bra at den kan brukes direkte som første del i selve rapporten
 - Burde diskutere hvilken arbeidsmetode som skal brukes med oppdragsgiver. Scrum passer ikke om hele kravspesifikasjonen er avgjort på forhånd
 - Viktig å få på plass grupperegler
 - Oppklar med oppdragsgiver hvordan han vil gå fram med kravspesifikasjon
 - Veien videre: Skrive prosjektplan (10-30) sider. Følg lynkurset til Rune

Møte med Espen 09.01.2020 (Thomas)

- Bacheloroppgaven vår er ikke hemmelig, og vi kan ta opptak av møtene (opptak for dette møtet gikk skeis)
- Framgangsmåte:
 - Samarbeid med oppdragsgiver: Vi møtes en gang i uken, klokken 12:00 på mandager. Vi holder disse møtene så lenge det er hensiktsmessig
 - Vi burde begynne med en mockup (på papir?), som kan vises til HV05. (Terningmoen?)
 - Vi skal besøke Elverum minst to ganger for å prate med endebbrukerene
 - Oppdragsgiver vil fungere som proxy mellom oss og sluttbrukere. Vil skaffe use-case diagrammer
- Oppgaven vår er gitt i forbindelse med HV05, som er et vanskelig område geografisk
- Introduksjon til SOI
 - Dokument med informasjon nødvendig for at styrker skal kunne kommunisere sikkert med hverandre
 - 10-15 kommunikasjonsmetoder er ikke unormalt, og SOI-er må støtte alle sammen (ofte på samme tid)
 - SOI vi skal lage burde sikte på 2 ark
 - SOI inneholder alltid
 - unikt nummer
 - periode
 - gradering
 - SOI inneholder ofte
 - Forhåndsavtalte koder.
 - Denne modulen kan representeres på to måter. Programmet burde ideelt sett støtte begge. En bruker en kode for hver

kategori, den andre bruker lengre koder for hver enkelt punkt.

Eksempel:

- Kategorien “retning” kan være R, underpunkt “sør” kan være S. Koden blir da RS
- Kategorien “retning” har ingen kode, og underpunkt “sør” har kode “HA”. Koden blir da “HA”
- Forhåndsavtalte koder kommer fra kodebok (?)
- Radio-tabell med info nødvendig for å kommunisere
- “Authentication table”
 - A-J og 0-9 bruker ofte samme koder
- “Subtractor codes”
- Sambandsdiagram - Veldig nyttig!
- Prosedyrer
- Bacheloroppgaven vår går ut på:
 - Kjerne:
 - Lage verktøy for produksjon og administrering av SOI
 - Skal kunne fungere helt isolert
 - Skal fungere godt i svart/hvitt
 - Graderingsmerket må være rødt
 - Sikkerhetsinstruks må være blått
 - Forsvaret er på bokmål!
 - Må være modulbasert. Skal være mulig å utvide programmet vårt
 - Lesbar kode. Det er et krav at koden vi leverer skal være lesbar, slik at det kan brukes som en prototyp på et potensielt fremtidig system
 - Burde bruke en slags “embedded” database. Skal kunne flyttes rundt på en minnepinne
 - Burde kunne automatisk foreslå layout av valgte moduler, må da også støtte manuell redigering av layout
 - Må kunne automatisk oppdatere koder daglig ol.
 - Kodebok:
 - Må kunne oppdatere begrepene en bruker
 - Kort / lang kodebok
 - Enten kode/dekode -bøker eller én bok til begge. Støtte begge disse? Brukere vil foretrekke ulike representasjoner
 - Beste kilder på kodebøker kommer fra telegrammenes tid. Boken “The Code Breakers” er veldig aktuell.
 - Må levere korte instruksjoner på hvordan en bruker programvaren
 - Installasjonsprosess burde kunne forklares på max. en halv side
 - Det må være tydelig fra programmet hva som skrives ut som sluttprodukt
 - Vi må velge gode fonter, god lesbarhet
 - Funksjonalitet over utseende
 - Må kunne produsere en “serialisert” utgave av SOI som senere kan leses for å produsere pdf. Dette er for å kunne sende SOI-er over dårlige koblinger
 - Må kunne kjøre på sluttbrukerens maskiner:

- En rimelig oppdatert Windows
 - Vi kan anta at moderne Python vil fungere
 - Vi kan anta 64-bit maskiner
- En må kunne vite hvem som har fått hvilke versjon av SOI
- Ekstra:
 - Verktøy for å lage og vedlikeholde sambandsdiagram. Bruker taster inn info om geografi og hindringer, og verktøy spytter ut diagram som viser kobling mellom styrker. Ved endringer burde kjapt kunne kalkulere nytt sambandsdiagram
 - Synkronisering av data (kodebøker) mellom brukere. Dette er vanskelig. Vi burde i det minste beskrive i rapporten hvordan vi ville angrepet dette
 - Evne til å fange opp kodeord som er fonetisk like, eller har lik ending

Møte med Espen 13.01.2020 (Thomas)

- Prototype skal kunne produsere...
 - X antall SOI i papirform i valgt format, der format er kort, lengre eller lengst
 - Notat om format: Kanskje format til og med har noe å si innad i en modul? Foreks. Én kolonne i en tabell i en modul er overflødig på kort versjon, men skal inkluderes i lang versjon
 - Kompakt SOI i "standard US ASCII"
 - Ikke binær, for det er "ofte vrient å håndtere"
 - Helst US ASCII, for å maksimere støtte
- SOI på fil skal ikke overføres av vår programvare. Vi skal bare produsere en fil som senere kan overføres ved Forsvarets standardmetoder. Filen må være liten og US ASCII for å kunne flyttes over så mange medium som mulig
- SOI kan sendes opp, ned, eller sidelengs i hierarkiet. Selv om ikke vår programvare skal utføre selve sendingen må vi støtte denne typen bruk. (Foreks. en SOI skrevet på bunnen skal kunne brukes på topp)
- Format på Mockup er opp til oss. Poenget er at brukere skal kunne visuelt se hvordan bruk av applikasjonen vil bli
- Espen fikk kontrakt av oss, skal videreføre til Leif. Burde signeres av både Espen og Leif
- Espen planlegger fysisk møte med Leif
- Tips fra Espen rundt brukervennlighet: Vi burde tenke på at arket som til slutt produseres er fokus. Applikasjonen burde designes rundt dette

Møte med Frode 14.01.2020 (Thomas)

- Feedback fra Frode skader ikke "selvstendighet" på karakteren
 - MEN det har skjedd at for mye hjelp har skadet en karakter. Dette var en "special case"
 - Frode kan kommunisere med sensor rundt karakter. Foreks. ved vippekarakter. Frode møter også sensor morgenen før presentasjonen
- Liste av aktiviteter ser bra ut. Frode hadde heller ikke vært borti "Work Breakdown Structure" før

- Den skal mate Gantt diagrammet, men vi burde ikke ha fullt så detaljerte punkter som vi har i WBS
- Vi kan stjele fra oppgaveteksten uten å henvise til kilde. Klipp-og-lim er helt lov
- Risikovurdering ser bra ut
 - Oppdeling av prosjekt/produkt -mål OK
 - Mengde av risikoer OK
- Vi fikk utskrift av prosjektplanen vår så langt med tilbakemelding fra Frode skrevet på. Petter tok imot denne
- Feedback på grupperegler
 - Burde vurdere å skrive konkret rundt møter og felles arbeidstid. Det er vanlig
 - Reglene ser OK ut
 - Istedenfor å starte alle punkter med “pliktes å” burde dette flyttes opp slik at punktene blir “Gruppemedlemmer pliktes å... 1) møte opp 2) være greie og snille” osv.
- Feedback på prosjektplan
 - Omfang skal være begynnelse av rapport. Bakgrunn burde derfor være mindre konkret, og mye info vi nå har i Bakgrunn burde flyttes til Omfang. Se feedback på papirer vi fikk tilbakelevert av Frode
 - Husk blank rundt alle paranteser
 - Prosjektmål: Legg til læringsmål
 - Resultatmål skal vise oppdragsgivers fortjeneste. “Hva tjener oppdragsgiver”
 - Mål og rammer skal “stå litt for seg selv”
 - Effektmål trenger ikke nødvendigvis være så konkret at den bruker målbare mål (tall, prosent, osv.) Om vi har muligheten til det så “er det fint det”
 - Rammer skal for eksempel være:
 - Tidsrammer: Vi har fra Januar til Mai å bli ferdig på. Vi kan bare møte brukere da og da
 - Teknologi: Hvilke teknologier er relevante
 - Fagområde burde inkludere kommunikasjonsformer
 - SOI burde først introduseres i punkt 2.2 (Omfang - Avgrensning)
 - I oppgavebeskrivelsen må vi unngå “tåkete prat”. Foreks: “effektivt og intuitivt brukergrensesnitt” og “leslige fonter” er for tåkete.
 - Rapport skal skrives til sensorer og medstudenter, så vi kan gå ut ifra at leseren har datakunnskaper, men ikke leseren kan mye om Heimvernet og SOI osv.
 - Kravspesifikasjon skal være konkret og målbar, til en så stor grad at den kunne brukes i rettsak. Se kommentar over om “tåkete prat”
 - Ansvarsforhold: Vi har skrevet at gruppemedlemmer er sidestilt, men vi må også skrive hvilke konkrete oppgaver vi deler på
 - Vi må gjerne levere før 1. februar å begynne å jobbe
 - “Tids- og ressursplan” fra malen til Rune blir “smør på flesk”. Kan la Gantt diagram vise dette
- Arbeidsmetodikk: Det er greit med “plukk & mix” arbeidsmetodikk, og vi kan også endre kurs undervegs

Møte med Frode 20.01.2020 (Thomas)

- Overgang til prat om Heimvernet i Fagområde veldig brå. Burde flyte bedre, for eksempel “En av de som bruker kommunikasjoner / samband er...”
- Mål
 - Resultatmål: Hva skal leveres
 - Effektmål: Hva er gevinst
 - Frode er enige med vår forståelse av forskjell på resultatmål og effektmål
 - Bytte rekkefølge?
- Gantt:
 - Dele opp utviklingsbit
 - Opp til 10 biter pluss minus
 - Burde legge til milepæler for statusrapporter
- Rakk ikke lese, levert for sent. Vi kan komme innom onsdag

Møte innad i gruppa for å estimere oppgaver i utviklingsperioden 20.01.2020 (Thomas)

- Estimeringer:
 - Grunnleggende oppsett av utviklermiljø:
 - Dager: 2
 - Personer: 4
 - MVP Brukergrensesnitt
 - Dager: 4
 - Personer: 2
 - Notat:
 - Implementere det vi har bestemt oss for i design-fasen
 - Helt basic
 - Knapper, tekstfelt osv uten tilhørende funksjonalitet
 - MVP Scenario: Kunne velge modul, redigere litt, se preview, og skrive ut PDF
 - Dager: 8
 - Personer: 4
 - MVP Scenario: Skrive laget SOI til fil, lese fra annen maskin
 - Dager: 5
 - Personer: 4
 - MVP Scenario: Flytte data fra én maskin til en annen maskin via minnepenn
 - Dager: 5
 - Personer: 4
 - MVP Scenario: Scenario: Legge til og fjerne kodeord i liten / stor kodebok, for så å skrive ut
 - Dager: 4
 - Personer: 2
- Fant fram til estimat ved bruk av Planning Poker
- Resten av utviklingsoppgavene ble lagt inn i en egen bolk. Tanken er at vi da “smidig” velger hvordan vi vil angripe resten av nødvendig funksjonalitet
- Vi kom frem til at vi vil gjøre en ny runde med estimering ved starten av utviklingsfasen
- Brukte estimat til å skrive en ny version av Gantt. Link til JPEG i GitHub: <https://github.com/tholok97/tholok97-bitsec3-bacheloroppgave-prosjektplan/blob/ef7bcd38aa00a1aff7205f29686782852109070c/gantt.jpg>

Møte med Espen 20.01.2020 (Thomas, Nikolai)

- Espen pratet med Leif fysisk på fredag 17.01.2020
 - Mål er å gi prototype til seksjon (?) i HV som ser på forbedringspotensiale
 - Prosjektavtale er fin. Den er signert. Vi skal levere signert kopi tilbake på mail
 - Han har ingen datoer for møte mellom oss og HV klart ennå
- Espen har ikke hatt tid til å kikke på prosjektplan. Gjør det mest sannsynlig innen onsdag. Skal si ifra på mail når vi kan ta en prat
- Ny info:
 - Radioer (Frekvenstabell)
 - Ulike radiotyper krever ulike typer kolonner
 - "MRR trenger foreks de og de kolonnene"
 - Bruker velger hvilke kolonner de vil ha
 - Ulike tropper/avdelinger trenger ulik mengde data om hver radio
 - Enkelte kodeord skal bare pugges, disse skal komme på eget ark
 - Er fortsatt del av SOI, men leveres som separat ark som må pugges og deretter destrueres
 - Det er maks 5 slike kodeord (f.eks radiotaushet, tapt SOI)
 - Alle distrikter har "OOB" (Order of Battle)
 - Verktøy skal ha funksjonalitet for kallesignaler
 - Basert på hierarkisk organisasjonstre. Denne inneholder kallesignaler på ulike nivåer
 - Gjøvik er 205
 - Styrke er foreks 7, en tropp er 1, sanitet er 8 osv.
 - Trenger å kunne "importere" kallesignaler i en trestruktur (dette blir en modul?)
 - Vises som på midten av det fargerike SOI-eksempelet (ENGRBN) som bare er en stor tabell
 - Info kan også vises som en del av sambandsdiagram slik som SOI-eksempelet med sambandsdiagram. Dette gir mest mening for radionett
 - Mulighet for å legge til og fjerne callsigns
 - SOI på fil skal også muliggjøre tilpasning av mottatt SOI ved utskrift. Det er altså to måter å bruke mottatt SOI:
 - Reprodusere det sender ga deg
 - Generere i annet format
 - Et eksempel på bruk av dette er at en tropp kanskje alltid vil ha moduler i en viss rekkefølge, mens SOI-ene de mottar ikke nødvendigvis følger denne rekkefølgen. De kan da endre rekkefølge og utseende (og kanskje til og med legge til / fjerne litt info?) for å få SOI til å være mest mulig leselig for akkurat den troppen
- "Burde være fornuftige defaults på antall skrevet ut osv."
 - Det skal være en fornuftig default innstilling på utskriftsbestilling med tanke på antall av hver versjon av SOI
- Notat om BMS
 - Er en "stygg touch PC"

- Brukes til å sende “tekstmeldinger” frem og tilbake
- Jobber på et høyere nivå av nettverksstacken enn f.eks. radio. Kan kobles i ulike radioer, men likevel beholde samme adressering. (Altså adressering av BMS skjer på et høyere nivå enn radio)

Møte med Frode 27.01.2020 (Thomas)

- Prosjektplan er leverbar
- Kravspek i rapporten
 - Burde fylles ut ifra backloggen
 - “Hva vil kunden ha”
 - “Hva, Hva, Hva, Hva”
- Ordliste: Skru av “forekomster” ELLER bli helt sikker på at alle forekomster er listet opp
- Gantt
 - Frys versjon levert i prosjektplan
 - Lag nye versjoner, og diskuter endringer i rapporten (halv side)
 - Stort pluss
 - Merke hva som endret seg med farge
- Design
 - Diskuter på flere nivåer
 - Rød tråd
 - Zoom inn. “Her er hele bildet, nå skal vi se nærmere på denne biten” osv

Møte med Espen 27.01.2020 (Thomas og Nikolai)

- Synkronisering, flere på samme SOI
 - Isolert lokalnettverk
 - Brukere jobber sjeldent sammen på en SOI, så vi skal slippe å kikke på dette
 - Om vårt design tilrettelegger for samarbeid (database som støtter synkronisering foreks.) så er det er pluss
 - Ikke toppprioritet
- SOI skal inneholde “X av Y” informasjon. En kan da vite at en har utskrift nr. 3 av 20 utskrevne for eksempel.
- SOI skal inneholde “gyldig fra til”
- SOI skal inneholde “avdelingen SOI tilhører”
- SOI skal inneholde sidetall og totalt antall sider for den versjonen slik at man kan skille størrelsesutgaven innad i en SOI.
- Det skal til enhver tid eksistere:
 - Nåværende SOI (vanlig)
 - Neste SOI (vanlig)
 - Reserve SOI (vanlig)
 - Nød SOI. Denne inneholder mindre informasjon
- Lisensiering: LGPL er best, men GPL er greit (mellom de to)
- Dato på første og andre møte i Gantt’en vår virker rimelig. Espen skal avklare datoer for møtene
- Vårt forslag til dato for leveranse av mock-up er greit. 13.02.2020

- Auth tavle burde ha “størrelse” som en parameter. Altså en kan velge mindre tabell og mindre sikkerhet, eller maksimal tabell for maksimal sikkerhet
 - Dette gjelder også subtraktorkoder
- Ny funksjonalitet:
 - Kult om programmet kan spytte ut kodeord til annet bruk
 - Dette er for å forhindre duplisering innad i SOL'en
 - Kodeord kan genereres basert på tema? (byer, dyr)
- Radiotabellmodul må parameterisere kolonner
 - “RADIOMODELL” - Hvilke kolonner er viktige for hver radio
 - Finnes flere måter å vise tabell på, støtte flere? Minst den som vises i eksemplene vi fikk

Møte med Frode 03.02.2020 (Nikolai)

- Vi hadde ikke noe å ta opp og Forde hadde ikke noe å ta opp. Han ønsket oss en god uke :)

Møte med Espen 03.02.2020 (Thomas)

- Fysisk møte
 - “Midten av februar til (?) i mars skjer alt”
 - Uke 10 skal 4 HV-områder skal øve
 - 13-16 februar skal noen ha øvelse i “Soløya” (“Gokk”)
 - Espen sjekker om vi kan komme på onsdag
- Use-case diagrammer er ikke påbegynt
- Espen vil finne elektronisk utgave av utdanningsdirektiv (husker ikke nummer)
 - Gir forståelse av totalsystemet, bakgrunnssystemet
 - Vi kan søke opp på HV-skolen
 - US-Marines sine ressurser er også relevante, siden Norsk doktrine ligner

Møte Frode 10.02.2020 (Thomas)

- Trengte ikke møte, ble enige om å slutte med regelmessige møter. Vi tar kontakt når vi vil ta en prat eller få feedback på rapport

Møte med gruppen 10.02.2020 (Thomas)

- Diskutere teknologivalg. Skrev konklusjoner på tavle. Bilde av tavlen er lastet opp til Google Drive:
<https://drive.google.com/file/d/1n8UNOdn6L8U-2-X8f7X6y8RC1h6JQqje/view>

Møte med Espen (Thomas og Nikolai)

- Fikk kontaktinfo til en som “skal på øvelse” fra “S6 Grebe” på en post-it lapp. Vi skal kontakte han. (Kanskje han har testdata?)
 - Navn: Jon Tjøstel Ree-Lindstad
 - Tlf: 41559710
- i18n trenger vi ikke å tenke på, men om det “tar en halvtime” så må vi gjerne gjøre det
 - SOLer er enten ren engelsk eller norsk
- DB

- Hvis DB som støtter synkronisering er vanskelig -> velg en enkel en
- "TinyDB kan jo også synkroniseres", men da må dette gjøres manuelt
- Synkronisering i database er ikke et krav, men det er et pluss
- I praksis kommer mest sannsynlig et eksportert format sendt
- Utskrift:
 - Skal skrive ut alle versjoner på en gang (1 av 5, 2 av 5, ...)
 - Generere én komplett PDF med alle SOIer.
 - Dette er faktisk best. Oppgaven å skrive ut flyttes ut av vårt verktøy. "Alle printere kan skrive ut PDF"
 - Det er også enklest
- Om en SOI skal skrives ut alene (1 av 1) burde dette gjøres ved å lage en ny SOI basert på den gamle. "Problemet løses med prossess"

Telefonsamtale med Jon Tjøstel 11.02.2020 (Anders)

Jon forteller at det ikke er noen sambands-øvelse og at det derfor ikke er nødvendig å møtes der. Vi avtalte å møte med ham på jobben hans (Norsk tipping?) mandag 17/2 kl. 9. Han spurte litt om hva vi vil få ut av møtet og ville være litt forsiktig mtp. at noe data er begrenset.

Møte med Jon Tjøstel Ree-Linstad 17.02.2020 (Thomas)

- Kontekst:
 - Jobber i Grebe, en avdeling under HV-05. Leif er leder for G6, sambandsavdelingen. (? disse faktaene burde dobbeltsjekkes)
 - Møtetid: 1 time og 10 minutter
 - Sted: Norsk Tipping
- Frekvensliste må alltid med
- Subtraktor og autentifiseringskode skal også alltid med
- Forhåndsavtalte koder kan gjøres på flere måter
 - kort kode, og stokke om på bokstaver
 - ren kodeliste
- Kodeord på frekvensliste er vanlig, rart det ikke dukker opp på noen av våre eksempel-SOI
 - Gir en enklere måte å indikere hvilken Net ID som skal brukes
 - Net ID gjør det mulig å kommunisere på flere ulike måter på samme frekvens. Net ID skal likevel være unik på tvers av frekvenser
- Kategoriserte kodenavn er fint. Hundenavn for Net ID og blomster for kjøretøy foreks
- Frekvenser utdeles fra oven. Om programmet vårt kunne ha automatisk lastet dette ville det være fint (mindre mulighet for menneskelig feil)
- Typer av kommunikasjon
 - FF for bare tale
 - TD for tale og data
 - CNR for ?
 - FF har lengre rekkevidde enn TD
- Tjøstel vil helst ta utgangspunkt i en "full" SOI, markere hvilke tropper som er aktuelle, også få resten generert
 - Fritekst burde lagres så den kan gjenbrukes

- Forhåndsavtalte koder burde lagres, til og med flere utgaver hadde vært fint
- Forhåndsavtalte koder er det samme som kodeliste <- notat fra Thomas
- “Sambandsdiagram er det samme som frekvenstabell”
- Frekvenstabell og sambandiagram viktigst. Autentifiseringstavle og subtraktorkoder er alltid med
- “Legger inn NetID jeg kan bruke, også velge fra NetID hva jeg vil bruke til hver”
- Frekvenstabell
 - Mindre info i HV områder, derfor ofte enklere SOI. De er mer interesserte i kodeliste
 - Få med MRR / HF / ? type radio
 - “alt som er MRR i én boks”
 - “alt som er HF i én boks”
 - Begrunnelse: “MRR og HF har ikke nødvendigvis samme NetID”
- Fint med “ORBAT” (order of battle) struktur på høyere nivå. Callsign til forskjellige tropper og lag
 - Tjøstel har i tabelform
- “Hvis tropp 1 er med, må reflekteres i alle moduler”
- “Nå klipper ikke ut irrelevant info, dette kan et automatisk system fikse!” Da enklere å opprettholde “need-to-know”
- “I min drømmeverden huker jeg av alt jeg vil ha, så generes SOI automatisk”
- Kodeliste type 2: egne ord for dedikerte ting
- radiotaushet på egen side
 - “standard-forslag” -> mulighet for utvide / fjerne
- Bruker nå excel, blir mye gjennbruk, vanskelig å endre
- Kodeord
 - Enkel kodeliste stokker bare om på ord. Koder holdes alfabetisk
 - Excel har “bibliotek”, da har koder kommet fra et sted. Enkel kodeliste stokker bare på bokstaver
 - Felles kodebok har ikke blitt brukt før
 - “Sende grunnfil med brukte koder, lage SOI basert på det. De og de brukt opp”
 - “Så det ikke blir kræsje og krøll”
 - Felles kodebok hadde vært fint
- Ønsket scenario for Tjøstel virker som å være: Få SOI fra oven + grunnfil, lage SOI basert på denne
- Type må være med i frekvenstabell
- “Få fil fra høyere oppe som gir begrensninger” (Når SOI inngår i større operasjon)
- Vi spurte om nøkkeldata, han var ikke klar på hva det kunne være i vår kontekst. Vi skal spørre Espen / Leif
 - Har kanskje med “rød” og “svart” nøkkel å gjøre, hvor kombinasjonen av disse skaper en sikker kanal
- Callsigns: order of battle som er fixed
 - “Gitt hvem som skal være med, enkelt å generere”
- SOP
 - Hvordan skal snakke på nett
 - “Ved Hot Mic gjør det og det” <- eksempel

- Lagre fritext en alltid vil ha med er bra
- Mockup
 - Vi presenterer mockup. Følger ikke foreslått kjøreplan. Går igjennom to scenarier med mye diskusjon
 - Når skal lage ny SOI vil basere seg på gammel SOI
 - Ønske sortere etter frekvenser og velge ut NetID
 - "Få opp alle tropper som skal være med, fjern irrelevante fra SOI"
 - NetID burde være unik på tvers av frekvenser
 - Evt. få varsel ved overlapp, med mulighet for overlapp om nødvendig
 - "endre størrelse på moduler hadde vært snedig"
 - Plassering
 - Automatisk plassering i kombinasjon med prioritering virker smart. Prioriterer de viktigste
 - "Manuell redigering er noen ganger ønskelig, men jeg ville alltid brukt auto"
 - Ønskelig at skulle kunne spesifisere at en modul alltid skal på en viss side
 - Redigeringsmoduser (pop-up eller inline)
 - Beste kommer ann på hva om redigeres. Frekvenstabell best i pop-up, mens enkel fritext / tabell kan gjøres direkte
 - Tilby begge basert på kompliseringsnivå på modul som redigeres
 - Ønskelig å kunne legge til vilkårlige vedlegg
 - Kart fra annet program, NordBMS. Limes inn i SOI som vedlegg
 - Ønskelig fordi må sendes med SOI uansett
 - Kodelistekoder er random
 - "Hurtigtaster bra"
 - "Modul er fint konsept"
 - Kodebok
 - Ønskelig med egendefinerbar kodebok
 - Flere egne kodebøker basert på hva en driver med er praktisk
 - "Kodebok henger sammen med kodeliste"
 - "Når baserer seg på gammel SOI vil koder måtte genereres på nytt"
 - "Selve produksjonen må automatiseres"
 - X eksemplarer
 - Signeringsliste
- "Nå velger logo selv, men finnes sikkert en standard på dette"
 - Logo kommer av patch
- Kvitteringsliste
 - Noe som må skrives under på ved mottatt SOI for å holde styr på hvem som har fått hva
 - Ønskelig at denne skrives ut når en bestemmer hvor mange eksemplarer man vil ha. Eksempler:
 - 7 SOI skal til tropp A. Kvitteringsliste sendes med
 - 3 SOI skal til tropp B. Kvitteringsliste sendes med
 - "Distribuerer er kanskje en egen oppgave"

- Trolig ikke noe vi vil fokusere på, da dette er mer direkte relevant til distribusjon
- “Alle begynner med liste frekvenser”

Møte med Espen 17.02.2020 (Thomas og Nikolai)

- Distrikter eier frekvenser til de under
- Frekvenslister får Leif “en gang i blant”
 - “skal vi lese dette fra et sted?” Anders. Espen svarer med diagram, og svaret er vel egentlig at ja det ville vært fint. Idag gjøres det manuelt, men det gjør feil lett
 - Diagrammet er i Thomas sin notisblokk
- Flott om info (om begrensninger, frekvenser osv) sendes som fil som leses av program
- Regionale / Nasjonale / Lokale frekvenser
- Norge delt inn i X 4 / 5 regioner
- “Behovet til Tjøstel er lett oversikt over tilgjengelige frekvenser”
- Kodebok
 - Hvilke kolonner trengs? Kategorisering? “Vi må ta en prat med Leif”
 - Foreslåtte kolonner ord / uttrykk, beskrivelse, kategori (valgfri?)
 - Kan sorteres alfabetisk eller på kategori
- Forhåndsavtalte koder: Begrep fra denne må være i kodebok
 - Kodene i kodeboka brukes til forh. koder
 - Stor / liten kodebok: Forh. koder ER liten kodebok
 - Kode i forh. koder BØR og ER separat fra kode i kodeboka
- Mockup: Vil Espen og Leif ha den? Ja, men Espen må sjekke med Leif
 - Scannet PDF på E-post
 - Skype

Gruppemøte etter møte med Tjøstel og Espen 17.02.2020 (Thomas)

- I kodebok:
 - Kryss for liten kodebok og ord som må pugges
 - Forhåndsavtalte koder trenger ikke bruke samme kode som i kodeboka -> bra for oss
- Database:
 - Er “SELECT kodeord WHERE den er i liten kodebok” tregt ved veldig mange kodeord?
 - Hva må lagres for hver SOI? Bare dato + versjon (skrives størrelse her?) + (tittel?) + dump av SOI JSON + avdeling?
- Vi må bestemme oss for design på forhåndsavtalte koder
- Tjøstel vil ha én tabell per type radio -> bra for oss
- Vi må få løst relasjon mellom frekvenser og NetID
 - Hvilken får en fra oven?
 - Hva er egentlig NetID?

Møte med Frode 24.02.2020 (Thomas)

- Research under teknisk design er naturlig, men se senere punkt om dokumentasjon av teknologier vi ikke valgte
- Research burde komme så tidlig som mulig, i teknisk design (forslag han hadde)
- Hvordan dokumentere valgene våre og teknologivalg vi ikke gjorde?
 - Enten “Sånn gjore vi det, kommer tilbake til det vi ikke gjorde senere”
 - Eller begge i én
 - Kjedelig å lese 3 sider om hva vi faktisk gjorde, også lese 7 sider om det vi ikke gjorde
 - Valg av teknologier og diskusjon rundt dette er en noe sensor ser nøye på og er derfor veldig viktig.
- “Avslutning” kapitlet er viktig, “innledning” er også viktig. Sensor leser disse to nøye, ikke nødvendigvis resten særlig nøye
- Ja, vi burde lage use-cases på vegne av Espen og Leif
 - Kommer i kravspek
- Analysekapitlet passer om kravspek må graves fram av oss (fra Frodes mal)
 - Om vi må analysere oss fram til kravspek
- Vi kan hoppe over “grovt design” i Frodes mal
- Kravspek er hva, ikke hvordan
- Kravspek skal holde i retten, oppgavebeskrivelse trenger ikke det
- Skrive kravspek i ettertid? Naturlig om vi jobber i sprint. Vi burde ikke skrive kravpek sprintvis
- Frode sent oss en rapport, vi skal ikke spre den til andre grupper
 - Den var ikke Frode veileder på
- Kravspek er ofte 8 til 25 sider
- Vi burde lete etter Toms anbefalinger på kravspek. “De som bruker Frodes mal burde vi se på”
- Ordliste, skal det komme tidlig i rapporten eller som vedlegg
 - “malen er en mal, ikke en kvass føring”
 - “Jeg syns det er greit å samle opp ekstra opplysninger på slutten”
 - “Jo viktigere jo tidligere”, med tanke på rekkefølgen av vedlegg. Prosjektplan kommer for eksempel sent
 - A ordliste
 - B Gantt
- Når vi refererer til bøker burde vi henvise til spesifikke sider
- “Småressurser kan komme som footnote”
 - Vår litteraturliste er veldig lang til nå, mye kan flyttes til footnote. For eksempel link til hjemmevernet
 - Litteraturliste / referanser er som regel ett A4 ark
- Grafisk design kan komme som eget kapittel etter Teknisk Design
 - Disse kan også komme sammen under ett kapittel “Design”, men anbefalingen til oss er å gjøre dem separat
 - Dette kapitlet kan inneholde en del figurer av hvordan resultatet ble, skisser og slikt er ikke så spennendes her

Møte med Espen 02.03.2020 (Nikolai og Thomas)

- Koder i kodebok skal oppdateres jevnlig, “ikke lagres statisk”
 - Kodene kan være enten 4-5 siffer, 2-3 bokstaver eller sammensetning av ord
 - Siffer kan misforståes som posisjoner
 - 2-3 bokstaver A-Z (ikke æ,ø,å) foretrekkes
- Send mockup til Leif og Espen
- Use-case & testdata: “Bare å begynne å fylle inn”
- Møtetid flyttes til klokka 11:30 (fortsatt på mandag)

Internt statusmøte i gruppa 10.03.2020 (Thomas)

- Møteleder: Nikolai Mork
- Hva gjør folk?
 - Thomas
 - *litt tynt skrevet her siden Thomas skriver referat også*
 - Implementert inline edit view
 - Implementert pdf produksjon basert på QGraphicsScene
 - Petter:
 - Oppsettdialog fungerer
 - Kan ikke hente ut informasjon fra dialogen enda
 - Kan ikke fortsette med oppsettdialog før SOI klasse er klar til å fylle informasjon
 - Mock-up mail er under arbeid
 - Anders:
 - Jobber med modulliste: omprioritering og endring av navn fungerer
 - Begynt på tabs, fungerer ikke å lukke tab
 - Stor og liten kodebok: fungerer i egne tabs, funker med database.
 - Dumt: En kan bruke vilkårlige typer, og det er vanskelig å hindre. Bruker ikke tid på det.
 - Lite problem at rad skrives til db før primary key har fått verdi.
 - Kan implementere sortering på andre ting enn ord.
 - “Kodebok view er ferdig”
 - Liten kodebok har litt problemer, kan for eksempel ikke legge til en rad. Dette er pga. måten liten kodebok er implementert på. Dette er det eneste som stopper oss fra å være ferdig med MVP rundt kodebok i Gantt.
 - Bonus: Enkel søking funker! Kan for eksempel trykke på “r” for å hoppe gjennom ord som begynner på “r”.
 - Autogenerering av kode mangler, det kan bli litt vanskelig.
 - Forslag fra Nikolai: Modifisere set_data til å inserte ord og kode, bruke get_code
 - Når dette er på plass er stor kodebok perfekt
 - Nikolai: Søk ikke prioritet, men kan komme tilbake til det hvis tid
 - Problem: Tomme primary keys ser ut til å lov, dette er et problem

- MVP 4: må også kunne skrive ut til PDF, dette er ikke implementert
 - Lagd sammendragmalene i LaTeX som Frode krever, fant ut etterpå at malene ligger ute. Mangler antall ord.
- Nikolai:
 - Jobbet med database
 - Generering av koder (også automatisk justering av lengde basert på antall ord i database)
 - Testing. For eksempel teste at koder oppdateres til å være lengre når antall ord overstiger grense
 - Diskusjon fra Anders: Skal vi automatisk oppdatere koder basert på tid eller oppdatere manuelt? Nikolai: Foreslår å støtte begge
 - QAction på kodebok for å oppdatere koder
 - Lagre sist endret tidspunkt i database
- Gjennomgang av Trello:
 - Ble enige om å splitte opp main_window så snart som mulig. Prioritet i Trello brukes
 - #25 flyttet til Code Review
 - #29 Under arbeid. Leter etter bibliotek for å generere koder, siden vår implementasjon muligens ikke er optimal.
 - Anders: Kodefelt, hadde vært flott å ha det til "not null". Nå brukes null-verdier for å fjerne alle koder før nye koder legges til.
 - Implementasjon som genererer nye koder er sub-optimal fordi den bare gjetter på nye koder. Når vi nærmer oss maks antall koder blir det veldig mye gjetting. Foreslår å øke kodelengde før det strengt tatt er nødvendig, for å forhindre at implementasjonen vår sliter.
 - Det som gjenstår: testing, implementere laveste kode, ... Testingen vil ta tid.
 - #42, #43 Hvis vi dropper robusthet mot brukeren, så er stor ferdig. Liten mangler funksjonalitet for å legge til rader. Diskutert tidligere i møtet.
 - Inline edit av liten database fungerer ikke.
 - Blocker: Må få unik kode fra arbeidet til Nikolai.
 - Trenger vi egentlig en egen tab med liten kodebok, siden det er forhåndsavtalte koder? Konklusjon: Nei!
 - Skrive ut kodebok: Bare skrive ut stor kodebok.
 - Bruke reportlab så vi kan gjøre skikkelig layout med sidetall osv. Alternativ løsning er å kjøre .render på table til QPrinter
 - #31 dette gjøres kjapt av Thomas
- Møtetid: 1.5 time

Internt statusmøte over Discord 16.03.2020 (Thomas)

- Møtet ble holdt over Discord pga. Corona
- Det ble gått gjennom hva enkelt gjør, og klargjort hva vi vil gjøre fremover
- #58 ble laget etter diskusjon i møtet

Internt statusmøte over Discord 19.03.2020 (Thomas)

- Fra Gantt skal vi være ferdig med alle “MVP”s, det tror gruppa vi skal få til. MVP 1, 3 og 4 er nesten ferdige, og MVP 2 påbegynnes nå.
- 04.03.2020 skal rapport og programvare være i “leverbar” stand. Det tror gruppen skal gå fint, men vi trenger mye fokus på rapporten fremover.

Internt statusmøte over Discord 23.03.2020 (Thomas)

- GitLab fungerer ikke. Nikolai har problemer med å gjøre “apply” på suggestions, og Thomas får ikke tilgang i det hele tatt
 - Ticket har blitt laget gjennom “NTNU Hjelp” om dette
- Vi spør om veiledningsmøte når vi leverer statusrapport. Møtet vil skje digitalt
- Fremover må vi jobbe på rapporten

Internt mini-møte om kravspesifikasjon over Discord 27.03.2020 (Thomas)

- Finner ikke noe god bruk av “lavnivå” use-case, utelater denne
- Finner ikke noe god bruk av “misuse-case”, utelater denne
- Må inkludere mer generelle beskrivelser av hver modul et sted rapporten før kapittel 3.
 - Sammen med harde fakta om hver modul i kravspesifikasjon?
 - I kapittel 1?
- Burde flytte fakta om hver modul i kravspesifikasjon til start, slik at dette er forstått før funksjonelle og ikke-funksjonelle krav.

Internt møte over Discord 30.03.2020 (Thomas)

- Vi må fokusere på statusrapport 2. Alle jobber på denne i dag, også diskuterer vi imorgen.
- Kontakter veileder for møte seint denne uka.

Statusmøte med Frode over Zoom 02.04.2020 (Thomas og Nikolai)

- Rapport:
 - Ser bra ut så langt
 - Mangler innhold, men dette kommer vel
 - “Ta bort de blanke sidene”
 - Men siden de kommer fra malen, så vi kan fortsette med det
 - Publiseringsavtale i sammendrag: Er det noe vi skal ha?
 - Vi kan skrive at den er åpen. Det er mer relevant når prosjektet er lukket.
 - Når vi skal levere til biblioteket må vi krysse av for at vi vil publisere
 - Kapittel 9 og 10 er vanlig å slå sammen, men siden vi har mye diskusjon kan det godt stå som eget kapittel
 - Vi må passe på å få med resten av punktene i lynkurs om konklusjonen
 - Kapittel 1 ser bra ut, inneholder alt som forventes
 - Hvorfor kommer det ny side før 1.3?
 - Rollebeskrivelse: Frode vil komme til slutt
 - Gjerne kulepunkter til oppgavebeskrivelse

- Forklaring på moduler (krav, bruk): kunne flyttes til eget kapittel, men vi kan gjerne fortsette å ha dem separert. Virker som veileder liker nåværende løsning (separert)
 - Vi må tenke hva som er best for leseren, det er medstudenter og forelesere som skal lese rapporten
 - delkapittel om oppgavebeskrivelse kan henvise til modulbeskrivelse
- Kravspesifikasjon
 - Ser bra ut, alt som skal være der er der
 - den skal være på 8-15 sider
- Kapittelinnledningene ser fine ut
- Hvert kapittel skal ha en innledning på 2-10 linjer om hva som kommer
- Leveranse: "jeg ville kalt det installasjon"
 - Vi kan stryke hvis installasjon er veldig enkelt
 - Heller bare skrive noen linjer om det under implementasjon
- "Kapittel 9 blir et innmari bra kapittel"
- I refleksjonskapittelet skal vi skrive litt om hvordan korona påvirket situasjonen vår
- Konklusjonskapittelet: Kall det "Avslutning" ?
- Inkluder i diskusjon Gantt diagrammet vi lagde i prosjektplanen kontra hvordan fremgangen ble i praksis
- Rapportstruktur: Denne er nyttig å ha med, men kanskje vi burde legge til mer informasjon?
 - Også viktig at alle kapitler inneholder "10 linjer" på hva det inneholder. Viktig
- Vedlegg: Hvordan inkludere vedlegg som er i prosjektplanen?
 - Rapporten må stå alene, å lese vedlegg skal ikke være obligatorisk.
- Fremover vil det være vårt ansvar å ta kontakt med veileder. Han er tilgjengelig til kontortid på lik linje med tilgjengelighet på kontor.

Møte med Espen over Zoom 03.04.2020 (Thomas og Nikolai)

- Espen har lest kapittel 2, og det er "ingenting jeg reagerer sterkt på, ingenting jeg stusser på", strukturen på rapporten er fin.
- Angående telefonliste:
 - En telefonliste kan inneholde rolle, sivilt nr., FDN-nr. og satelittf.nr.
 - Sivilt og FDN-nr har de fire siste siffer likt, og sivilt har da fire unike siffer før disse og FDN har tre unike siffer før disse.
 - FDN = Forsvarets Digitale Nettverk (- rolle, sivilt, number (nummer))
 - TADKOM = Taktisk Digital Kommunikasjon - Ikke egen adresse knyttet til denne, det er bare et virkemiddel for FDN kommunikasjon (utvider nettverket)
 - Fire siste siffer på FDN og sivilt nummer er det samme (tre siffer først på FDN) - "Blånummer" regnes som sikkert opp til begrenset (?)
 - FDN-nummer er internt mappet mot sivile nummer
- Testelement:
 - Stasjonsautentisering: Velge to forskjellige tall eller bokstaver

- Meldingsautentisering: Skrive ut melding, testelement er to bokstaver, må telle opp
 - testelement skal varieres
- Angående kallesignaler og ADM/BMS, CMD/BMS, GW
 - Order of battle, alle avdelinger som er knyttet til systemet
 - Order og battle lages ofte på et overordnet nivå og som konsekvens av dette kan det forekomme differanse i kallesigner over de forskjellige kommunikasjonsmidlene

Møte med gruppe over Discord 15.04.20 (Nikolai)

- Vi går gjennom lapper på trello og gjør opp status
- Vi skal enten skrive på trello-lapper eller sende folk en melding når vi kommenterer i Overleaf
- Petter er ferdig med matte og skal gi gass på grafisk design
- Nikolai jobber med telefonliste som en oppvarming og sandkasse før han begynner med frekvenstabell
- I vedleggsmodulen vil fraser legges inn manuelt (typ "SOI komprimert") og kodeord legges inn automatisk
- Vi diskuterte overskrifter til moduler
- Kallesignal skal fylles manuelt og løses med tabell
- Thomas fikser problemet med at testene som baserer seg på tid og sjekker differanse feiler på bl.a. windows fordi runneren er for treg, han begynner med å øke tillatt tidsdifferanse
- Finne en "monospaced" font som gjør at alle bokstaver har lik bredde, dette er ekstremt gunstig i moduler som subtraktorkoder som baserer seg på telling av tegn feks

Internt møte over Discord 20.04.2020 (Thomas)

- Referatskrivers PC kræsjet flere ganger under møtet, så referat kan desverre ha mangler
- Petter blir ferdig med grafisk design lapp idag #72
- Arbeidsprosess spørsmål Nikko: Skal lapp som er påbegynt, men pauset, flyttes tilbake til TODO? Konklusjon: Ja, men må kommentere på lapp at litt arbeid allerede er påbegynt
- Thomas forklarer "hack" nødvendig for #119
 - Gruppen er nå alle klar over løsningen
- Angående oppløsningskrise i kort #125
 - SOI kan ha ulikt antall moduler i bredden avhengig av skjermopløsning
 - Thomas ser på denne
 - Nødløsning er å lage programmet til å se fint ut på én valgt skjermopløsning, og så diskutere hvor vi gikk feil i rapporten
- Vi har nevnt kontrastkrav i forprosjektet, og dette støtter vi ikke enda. Hvis tid burde vi se på det
- Det er nå én måned til innlevering
- Vi sikter på å avslutte programmering søndag denne uken
- Forhåndsavtalte koder:

- fra referat: "liten kodebok ER forh. koder"
- Vi konkluderer at innhold i forhåndsavtalte koder kommer direkte fra "liten kodebok". Dermed ligger arbeidet stort sett i å implementere layout
- Motivasjonssjekk: Gruppen er motivert, og innser at mange gode arbeidstimer står igjen før levering av oppgaven

Internt statusmøte over Discord 27.04.2020 (Thomas)

- #125 oppløsningstrøbbel:
 - Kjører vi samme font på alle moduler? Det ligger kommentar på MR Thomas må se på. Konklusjon: Ja, men dette gjøres på ny lapp, så MR kan merges.
 - Times New Roman burde brukes til SOI font, siden denne er mest leselig på papir - Petter
- Forhåndsavtalte koder
 - Angående varslingsord: Dette kan kræsje med koder i Kodefrasemodulen, men vi tillater dette i vår prototype, da dette vil ta mye tid å fikse. Vi diskuterer det i rapporten vår.
 - Angående diskusjon på Messenger om å gi én kategori flere kolonner: Vi prioriterer ikke dette, da vår testdata ikke vil lage en modul høyere enn SOI høyde
- Telefonlister:
 - Modulen er ferdig, men mangler tester
 - Oppdaget bug under møte, hvor funksjonalitet som vises når bruker drar mus over ikke fungerer inne i SOI (bare når widget åpnet isolert)
- Referatskrivers PC døde her..
- Lesbarhetspolish:
 - Petter skal prøve å fjerne fokus på moduler før utskrift, slik at modulene ikke dukker opp med celler markert som "siste redigerte" i utskrift.
- Kodefrasemodul:
 - Thomas ser om det er mulig å fikse "tomrommet" mellom tabell og høyre side av modulen, som kommer når tabellen er kortere enn overskriften.
- Oppsett dialog må fikses til å bruke forståelige norske beskrivelser på alternativer (foreks. "sort -> none, area, osv.")
- Vi vil prioritere å endre fanenavn når SOI tittel endres. Dette kan løses ganske rent med "listeners"
- Vi prioriterer ikke sletting av moduler i første omgang, men vi kommer kanskje tilbake til det under rapportskriving om vi vil allokere en person til å jobbe på det istedenfor rapport

Internt møte over Discord 30.04.2020 (Thomas)

- Angående diskusjon om gyldighet fra/til i header: flertallet stemmer "Gyldig fra", "Gyldig til", med ekstra mellomrom foran "til", og med to til tre mellomrom mellom teksten og selve dato.
- Fanenavn løses med listeners. Dette ble tatt opp igjen, og listeners ble valgt.
- Nikolai har møte med kontaktperson rundt frekvenstabell etter dette møtet.

Møte mellom Espen og Nikolai over Teams 30.04.20 (Nikolai)

Møtet skulle originalt kun handle om detaljer rundt frekvenstabell, men utvidet seg etter hvert som praten gikk.

- Frekvenstabell baserer hele sin struktur på en dict, denne kan med forhold leses inn fra json. Da vil det både være enkelt å vedlikeholde programvaren når det kommer nye radioer og brukerne kan selv velge innhold med å endre denne json-filen.
- Kun sifrene 0-9 skal vises på autentiseringstavler.
- Kun en kolonne med bokstaver (A-Z), ikke to (slik som det er nå) før kodene.
- Moduler kan registreres på en dynamisk måte. Det var en del prat rundt dette. Espen klarte ikke helt å komme på hvor han hadde sett dette. "Det var så python, det var så vakkert." Uansett så han at vi hadde tenkt rett. Her er noen nøkkelord jeg noterte under samtalen:
 - registrere moduler dynamisk
 - oppstart plugin katalog
 - laste dynamisk kode
 - tenk på demoeffekt (lege til ny modul, starte programmet på nytt)
 - triks med import og underkatalog
- Han spurte om fontstørrelser og slikt. Jeg svarte at vi hadde fast fontsørrelse pga variasjoner av modulstørrelser på forskjellig skjermopløsninger. Han svarte igjen at det alltid var noe sånt med font og at det ikke var så nøye.
- Rundt komprimering av soi var det viktigst at det ble US ASCII.
 - Han snakket også om fordeler med base 64 vs base 85
- Om ting finnes i standardbibliotek er det veldig bra, for da er det mindre å sjekke for ansatte i forsvaret, det var noe politikkgreier...
- Generelt var det som ble vist fram veldig nærme det Leif har tenkt på!

Internt statusmøte over Discord 04.05.2020 (Thomas)

- Anders og Thomas har to programmeringsoppgaver til de vil gjøre, ellers annser vi programvaren som "ferdig"
 - Legge til reorganiserings knapp. Veldig lite arbeid
 - Litt oppdrydding i utseende på moduler: Gjør under møtet
 - Border til kodefrasemodul
 - Øke størrelse på ny modul dialog
- Hvordan gjennomføre siste innspurt?
 - Konsekvent bruk av lapper
 - "Code review" mulighetene i Overleaf er dårlige..
 - Forslag: Etter reviewer har lagt inn kommentarer, kan ringe forfatter og gå igjennom tilbakemeldingene
 - Burde lage alle lapper på forhånd så vi ser hvor vi er i verden
 - Dette ble gjort i fellesskap
- Spørre prosessveileder om
 - Kjente bugs er vedlegg. Hvordan diskutere disse i rapporten.
 - Hvordan få fram mangler ift. kravspek. Fjerne fra kravspek? Diskutere mangler i diskusjon?

- Skal vi inkludere referat som vedlegg?
- Hvor detaljer skal beskrivelser av moduler være i kravspesifikasjon? Greit at vi diskuterer helt konkret slik det "ble"?
- Burde våre seksjoner 2.1 "Krav til SOI-moduler" og 2.2 "Krav til kodebok" gå under funksjonelle krav, eller kan de stå "fritt" slik de gjør nå?
- Vil vi gjennomføre brukertest over Zoom? Mulig nå som vi har leverbar ZIP.
 - Gir mer å skrive på brukertesting
 - Kan få bekreftelse på mål
 - Må ikke bruke mye tid på det, da brukertesting kapittelet er gyldig allerede
 - Estimerer 40 minutt til en time på selve brukertest
 - Avventer dette valget til mandag 11.05.2020.
- Gruppen vil ha møte neste torsdag 07.05.2020 med prosessveileder. Vi sender rapport 06.05.2020
 - Petter tar kontakt
- Innlesing av config fil gir ikke melding til bruker om feil, noe som bryter med kravspek.
- Må bli flinkere til å bruke Trello lapper til å diskusjon og oppdateringer. Før kunne vi bruker GitLab MR til dette, men Overleaf sin støtte for diskusjoner er dårlig.
- Thomas må få timene sine inn i Excel

Møte mellom Espen og Nikolai over Teams 06.05.20 (Nikolai)

Møte omhandlet i hovedsak oppklaringer rundt sambandsdiagram og radiodiagram og nødvendigheten av disse.

- Sambandsdiagram er en visuell framstilling av strukturen i et nettverk. Det viser direkte kobling mellom stasjoner og reléer. Det begynner alltid med 9 nederst.
 - En komplett SOI skal da også ha en dekningskalk som viser hvor de forskjellige har dekning på de forskjellige frekvenser. I praksis er dette mest for sambandskorporaler.
 - En av grunnen til at dette er visuelt er at alt kan ikke beskrives i teks.
- I praksis er sambandsdiagram og radiodiagram det samme, det er noen nyanser i forskjell, men det trenger ikke vi tenke på.
- Enkleste løsning på å gi støtte for dette er å gi mulighet for å laste inn et bilde. Da har man samtidig gitt støtte for alt, man kan da altså replikere en total SOI.
 - Om dette blir implementert er det "tipp topp tommel opp", men det må vurderes opp mot hvor mye tid gruppen har.
 - Om det å legge til et bilde krasjer med det generelle oppsettet vi har, layoutalgoritmer eller ikke passer med de bibliotekene vi har skal vi ikke prioritere det.
 - Om det er lettvindt for oss er en løsning å gi en enkel og skranglete løsning for det slik at vi har støtte for det og heller utdype utviklingsarbeid i rapporten. Da blir det et "proof of concept"
 - Bildet trenger ikke være i høy oppløsning.
 - Bildet kan komplisere kompakt filformat eller størrelsen på denne slik at det ødelegger sendetid.

- Det er mulig å bare gi en advarsel om at bildet vil forsinke sendetid og være ferdig med det. Eller en advarsel om at bildet overskrider max størrelse. Vi kan anta at folk i denne stillingen "kan bruke hue".
- Bilder kan også skaleres med python image library.
- Et verktøy for å lage radiodiagrammer og sambandsdiagrammer er utenfor rekkevidden til denne oppgaven.
- Om dette ikke blir laget er det lett å tegne for hånd å legge til som papirvedlegg.
- Man kan regne med at de som produserer SOI har tilgang til minimum powerpoint.
- Man kan tenge en graf i powerpoint og gjøre den svart/hvitt for å sjekke størrelser angående grense på størrelse.
- Man kan gjøre bilder svart/hvitt for å spare størrelse.
- Konklusjon: Om det er tid og det ikke gir tekniske problemer vil det være veldig positivt å bare ha en enkel løsning for å legge til bilde. Denne modulen kan gi en enkel beskjed om komplikasjoner rundt tillegging av bildet. Resten kan diskuteres i rapporten for å spare oss for utviklingsarbeid.

Møte med Frode over Zoom 07.05.2020 (Thomas og Nikolai)

- Svar på spørsmål vi har stilt:
 - Linker i fotnoter burde også få "assesert <dato>"
 - Vi skal dokumentere kjente bugs i diskusjonskapittelet, på slutten. Svaret forutsetter at vi har ca. 4-5 bugs å dokumentere
 - Angående mangler ift. krav: Krav som vi har funnet på selv og ikke har møtt går det an å fjerne der det er hensiktsmessig, mens krav direkte fra oppdragsgiver må beholdes. Vi kan diskutere mangler i diskusjonskapittelet.
 - Portrett / landskap er et eksempel hvor den som ikke er implementert kan fjernes fra krav, da dette var et krav vi fant på.
 - Vedlegg burde følge rekkefølgen i Froles lynkus-PDF
- Rapport er i rute. Vi får vite at vi ligger greit an.
- Rapport er fin på høynivå. Antall referanser er bra.
- Muntlig framføring skal i hovedsak være kap.1 og det er viktig av vi framfører stoffet på en populærvitenskapelig måte. Vi skal selge og få dem til å ha lyst til å lese rapporten.
- Det har ikke så mye å si hvordan vi skriver nye uttrykk i rapporten. ("x" etc.)
- Vi må ha en klarere innledning til funksjonelle krav som linker det opp mot det som nettopp var skrevet, slik at overgangen blir mykere.
- Innledninger må være på 2-10 setninger avhengig av innholdet som kommer og hvor mye introduksjon det trenger.
- Kap. 3 hadde litt lite volum, men om det ikke er mer å skrive om skal vi ikke skrive mer. Legg til innledning.
- Kapittel om Leveranse skal føres inn under slutten av kapittel om implementasjon

Internt statusmøte over Discord 11.05.2020 (Thomas)

- Neste mål for TODO-lapper: fredag 15.05.2020

- Fram til neste møte prioriteres gjennomføring av lapper, ikke code review, med begrunnelsen at det er bedre med bacheloroppgave med komplett og potensielt ikke-reviewet tekst, enn bacheloroppgave med ukomplett og reviewet tekst.
- Nikolai kontakter oppdragsgiver med spørsmål om Heimevernet stiller samme helsekrav som Forsvaret generelt.

Internt statusmøte over Discord 15.05.2020 (Petter)

- Dagene fram til 18. mai prioriteres til å få skrevet ferdig alt i rapporten
- 18. mai skal gruppen bruke hele dagen på å lese gjennom rapporten og rette den slik at den er grei til innlevering
- Petter sender mail til Frode om spørsmål angående NTNU grafisk senter og forside rapport

Vedlegg I

Logg

Vedlegget inneholder loggføring av arbeidet til hvert gruppemedlem fra hele prosjektperioden. Arbeid loggføres for hver dag med dato, antall timer, og en kort beskrivelse av hva som ble utført. Merk at ett gruppemedlem, Thomas, førte sine timer i et automatisk verktøy, som resulterer i annet format på antall timer.

Petter			Nikolai	
Dato	Timer	Beskrivelse	Timer	Beskrivelse
07.01.20	1	Oppstartsmøte	1	Oppstartsmøte
08.01.20	2.5	Møte med veileder, grupperegler	2.5	Veiledermøte, grupperegler
09.01.20	6	Møte oppdragsgivers kontaktperson	6	Møte m/ kontaktperson, prosjektplan
10.01.20	3.5	Prosjektplan, skriving	5	Prosjektplan
11.01.20				
12.01.20				
13.01.20	5	Møte kontaktperson, prosjektplan	5.5	Møte m/ kontaktperson, prosjektplan
14.01.20	4.5	Møte veileder, prosjektplan	4.5	Møte m/ veileder, prosjektplan
15.01.20				
16.01.20	5.5	Prosjektplan	6	Prosjektplan
17.01.20	6	Prosjektplan	6	Prosjektplan
18.01.20	1.5	Codebreakers		
19.01.20	2.5	Codebreakers, litt skissering		
20.01.20	7.5	Møte med veileder og kontaktperson	7.5	Møte m/ veileder, kontaktperson og prosjektplan
21.01.20				
22.01.20	7	Research: gui + kryptografi	7	Resarch: kodekvalitetsverktøy og linting
23.01.20	5	Mer research: PyQt5 vs. PySide2	6.5	Resarch: linting + 2d bin packing problem
24.01.20	5	Research + læring av valgt teknologi	7	Research: docker image og rectpack
25.01.20				
26.01.20				
27.01.20	8	Møter, PyQt5 widgets, "Python & Qt"	7	Møter, rectpack demo og CI/CD-kødd
28.01.20	3	PyQt5	2	Research
29.01.20	5.5	Widgets for PyQt5	5	PyQt5 tutorial
30.01.20	5.5	Custom widgets + layout	7	PyQt5 tutorial
31.01.20	5	Custom widgets PyQt5	6.5	PyQt5 tutorial
01.02.20				
02.02.20				
03.02.20	6	Qt Designer	7.5	Møter og testing av bilder
04.02.20	5	Fortsettelse Qt designer	3	Fikset ferdig gårsdagens greier
05.02.20	3.5	Qt Designer	6	serialisering
06.02.20	5.5	GUI	7	komprimering
07.02.20	5	GUI	5.5	komprimering og comboboxtesting
08.02.20				
09.02.20				
10.02.20	7.5	Design av GUI + møter	8	Møter og GUI-skissing
11.02.20	3	Start på mockup	1	Diskutert GUI navigasjon
12.02.20	7	"Ferdigstille" mockup	6.5	SOI moduler design diskusjon
13.02.20	6	Mockup + statusrapport	6	Research på rapport
14.02.20	5	Rapport + GUI	4.5	rapport + json-design
15.02.20				
16.02.20				
17.02.20	7	Møte Hamar, møte kontaktperson, DB-design	7	møter + desig
18.02.20	3.5	Statusrapport	4	design og statusrapport
19.02.20	5	Filstruktur, programvarearkitektur, gui-oppsett	6	JSON serialisert design og mer
20.02.20	6	Planlegging MainWindow og MainWidget		
21.02.20	5.5	Start MainWindow/-Widget, ny estimering	1	Estimering
22.02.20				
23.02.20				
24.02.20	10	Møte veileder, endringer på rapport, mainwindow	7.5	Møter og rapport
25.02.20	7	Videre utvikling av mainwindow	6.5	db kodebok
26.02.20	4	Finpuss mainwindow	2	db kodebok
27.02.20	5	Popup for oppsett, scan mockup	8	db kodebok + statusbar + keybindings
28.02.20	5.5	Videre jobb med oppsett, stylesheet	5	db kodebok stor og liten
29.02.20				
01.03.20				
02.03.20	7	Oppsett-knapp	6.5	smoke test, møte og kodegenerering
03.03.20	3	Stylesheet setup	3	kodegenerering og testing
04.03.20			2.5	Testing av kodegenerering
05.03.20	6	Lagre og avbryt på oppsett	7	Regenerering av koder
06.03.20	3	Oppsett ferdig?	3	Test kode-regenerering og utvidelse av kodegenereringsfunksjon
07.03.20				

08.02.20			
09.03.20	6	Oppsett ferdig + småfiks kode + mockupbeskr.	5 Funksjon for å legge til 1 kode + testing
10.03.20	5	Mockupbeskrivelse, finpuss oppsett, statusmøte	6.5 Testing av db-funksjonalitet og generering av testdata
11.03.20	4	Småttteri rundt oppsett + litt rapport	6.5 Testing og refactoring av db-funksjonalitet
12.03.20	5	Liten start på endring av SOI gjennom oppsett	5 Merge request og begynte på tidsoppdatering av db
13.03.20	5	Forts oppsettendring	5.5 Automatisk oppdatering av db (tid)
14.03.20			
15.03.20			
16.03.20	6.5	Møte gruppe + oppsettjobb	6 Møte + automatisk oppdatering på tid
17.03.20	7	Alle filer gjennom Black + oppsett	8.5 Ferdigstilling/MR av automatisk kodeoppdatering på tid
18.03.20	8.5	Clear editfelt, radiobuttons i groupbox	5 MR
19.03.20	5	Hente data fra soi	5 db singleton + minimøte + kompresjon
20.03.20	4	Hente data fra soi	4.5 Fikset MR
21.03.20			
22.03.20			
23.03.20	7	Hente data fra soi og lagre til soi + gruppemøte	5 gruppemøte + db gui + MR
24.03.20	5	Fullføre oppsett-dialog	2.5 gransket koderegenerering
25.03.20	5	Fikse treads fra MR for oppsett	5.5 generell jobbing med rapport
26.03.20	3	Rapportjobb	5.5 generell jobbing med rapport
27.03.20	7	Hjelp-dialoger	6.5 MR + gruppemøte + generell jobbing med rapport
28.03.20			
29.03.20	2	Rettet på MR + generell skrivning GUI	
30.03.20	6	Gruppemøte + rapport gui	4 møte + rapport: modulerbeskrivelser
31.03.20	4	Gruppemøte + statusrapport/rapport	5 møte + statusrapport + MR + rapport: utbedring modulbeskrivelser
01.04.20	5	Statusrapport	6.5 MR + statusrapport + rapportkorrigering + kodecleanup
02.04.20	4	Møte veileder og internt + rapport	5 MR + veiledermøte + raferat + setching på telefonlistemodul
03.04.20	3	Møte kontaktperson, litt rapport	2 kontaktpersonmøte + headerdesign og litt telefonlistemodul
04.04.20			
05.04.20			
06.04.20			
07.04.20	1	Lett skissering til gui-kapittel	
08.04.20	2	MVC? for GUI	5 Telefonlistemodul
09.04.20	1	Samme som over, gui-modell	2 Telefonlistemodul
10.04.20	2	Rapport + MR	4 Telefonlistemodul
11.04.20			4.25 Telefonlistemodul
12.04.20			
13.04.20			
14.04.20			5.5 MR + Telefonlistemodul
15.04.20	4.5	Gruppemøte + rapportskrivning	4.5 MR + gruppemøte + Telefonlistemodul
16.04.20	4.5	Rapport, teste applikasjon, rote i rapport	5.5 MR + Telefonlistemodul
17.04.20			
18.04.20			5 Frekvenstabellmodul
19.04.20			2 Frekvenstabellmodul
20.04.20	7	Gruppemøte, rapport, font til SOI	7 Frekvenstabellmodul
21.04.20	7.5	Rapport, lesbarhetpolish	6.5 Frekvenstabellmodul
22.04.20	7	Lesbarhet, hurtigtaster	2.5 Frekvenstabellmodul
23.04.20	4.5	Lesbarhet, skaleringtest	7 MR + Frekvenstabellmodul + Telefonlistemodul
24.04.20	2	Fonter skjerm/papir, skaleringtest	3 Telefonlistemodul
25.04.20			4 Telefonlistemodul
26.04.20			
27.04.20	7.5	Gruppemøte, oversett oppsett, hurtigtastfiks	5 Gruppemøte + Telefonlistemodul + smått begynt på testing
28.04.20	4.5	MR-fiksing, statusrapport 3	6 Testing telefonlistemodul
29.04.20	5.5	MR, statusrapport, brukertesting	9 Ferdigstilling telefonlistemodul + frekvenstabellmodul med testing
30.04.20	5	MR, fullføre statusrapport, WCAG 2.0, møte	6 Møte med Espen, gruppemøte, frekvenstabell med testing
01.05.20	4	MR, litt brukertest	9 Testing telefonlistemodul + MR
02.05.20			2 MR
03.05.20	3.5	MR, sjekke kontrast	8.5 MR
04.05.20	10	Gruppemøte, rapport, bughunting	6.5 Gruppemøte + Rapport
05.05.20	8	Rapport, grafisk design	7.5 Rapport
06.05.20	4	Grafisk design rapport	10 Rapport + møte med espen
07.05.20	5.5	Rapport, møte veileder	7 Rapport + møte med espen
08.05.20	5	Grafisk design SOI-moduler	5.5 Rapport
09.05.20			
10.05.20	4.5	Font og kontrast, MR	

11.05.20	6	Gruppemøte, rapport brukertesting	5	Rapport + gruppemøte
12.05.20	5	Brukertesting, inline edit	3.5	Rapport
13.05.20			2	Rapport
14.05.20			2	Rapport
15.05.20	6	Gruppemøte, inline vs popup	12.5	Rapport + gruppemøte
16.05.20	6	Vedlegg, vedleggstekster, vår SOI	9.5	Rapport
17.05.20	7	Review av rapport	11	Rapport
18.05.20	14	Lese og fikse rapporten	15	Rapport
19.05.20	16	Lese rapporten	16	Rapport
20.05.20				

	Anders		Thomas	
Dato	Timer	Beskrivelse	Timer	Beskrivelse
07.01.20	1,00	Oppstartsmøte	1:0:0	møte
08.01.20	2,50	Veiledning, grupperegler	3:15:15	frodemøte + gruppearbeid,lynkurs 1
09.01.20	6,00	Møte m/ kontaktperson, prosjektplan	5:59:59	møte med Espen + gruppearbeid etterpå
10.01.20	5,00	Prosjektplan	4:30:30	arbid med prosjektplan,arbid med prosjektplan
11.01.20				
12.01.20				
13.01.20	5,50	Møte m/ kontaktperson, prosjektplan	5:25:25	arbid med prosjektplan + møte med espen + referat
14.01.20	4,50	Møte m/ veileder, prosjektplan	3:44:44	møte frode + gruppearbeid,grupppearbeid
15.01.20			1:43:43	arbeid med prosjektplan
16.01.20	6,00	Prosjektplan	2:59:59	arbid med prosjektplan,arbeid med prosjektplan
17.01.20	6,00	Prosjektplan	5:42:42	arbeid med prosjektplan, gantt,arbeid med prosjektplan
18.01.20				
19.01.20				
20.01.20	7,50	Møte m/ veileder, kontaktperson og prosjektplan	7:30:30	frodemøte, planning poker
21.01.20			0:12:12	arbid med prosjektplan
22.01.20	7,50	Research	6:50:50	arbeid med prosjektplan, research på teknologier
23.01.20	7,00	Research, test sqllite3	5:6:6	research,research
24.01.20	5,50	Research	7:39:39	læring,læring,læring,læring,research
25.01.20				
26.01.20			0:33:33	læring
27.01.20	7,00	Møte m/veileder, kontaktperson og research	7:3:3	møte med frode, læring, møte med espen
28.01.20	2,00	Research, tabeller i PyQt5	1:29:29	læring
29.01.20	5,50	Research, tabeller m funksjonalitet	5:38:38	læring, pyqt testing,læring,læring
30.01.20	8,00	Research, implementert tabell, startet tabellutskrift på PDF	5:3:3	læring,læring,læring
31.01.20	2,00	Research	5:46:46	læring,læring,læring
01.02.20				
02.02.20			0:1:1	læring
03.02.20	7,00	Møte m/ kontaktperson og research på gui-testing QtTest	7:0:0	læring,læring
04.02.20	3,00	PyInstaller og internasjonalisering		
05.02.20	5,00	Pytest-qt	6:0:0	læring
06.02.20	6,00	Model view, tinydb	5:52:52	læring,læring
07.02.20	6,00	Pipeline-script PS, tinydb, QtTest	7:21:21	læring
08.02.20				
09.02.20	0,50	Gått gjennom referat, klargjøring mandagsmøte		
10.02.20	8,00	Møte m/ kontaktperson og GUI-design	6:56:56	design,design
11.02.20	3,50	Tif Tjølset og GUI-design	2:55:55	research,research,research
12.02.20	6,00	Modul-design	6:0:0	design
13.02.20	7,00	Rapport, QAbstractTableModel	6:31:31	rapportskriving
14.02.20	5,50	Toppmeny, design SOI på fil	5:54:54	design + hv møte planlegging,design + hv møte planlegging
15.02.20				
16.02.20	1,00	Rapport: teknologivalg og utviklingsprosess		
17.02.20	7,00	Hamar- og kontaktperson-møter, DB- og frekvenstabelldesign	7:0:0	reise til hamar + møte med tjølset + møte med espen + gruppearbeid
18.02.20	5,00	Rapport++		
19.02.20	6,00	Programvarearkitektur, filstruktur	7:0:0	statusrapport 1 + docstring research
20.02.20	6,00	Script i Bash, README-instruksjoner	3:50:50	oppsett av utviklermiljø,oppsett av utviklermiljø
21.02.20	6,00	Database-setup, ny estimering av utvikling	2:42:42	estimering + miljø,rart ubuntu runner problem med gui testing
22.02.20			0:15:15	lesing
23.02.20				
24.02.20	7,00	Justeringer på rapport, database.py	7:12:12	frodemøte + miljøoppsett + rapportskriving + espenmøte
25.02.20	7,00	Database-tester	2:57:57	fiks pipeline på master, miljø og oppsett
26.02.20	6,50	Multi-arv abstractmethod, QSqlDatabase, lynkurs skrijving	3:0:0	miljø og oppsett,lynkurs 2
27.02.20	9,00	Multi-arv interface, tabellmodul, diskusjon	8:26:26	miljø og oppsett + MVP brukergrensesnitt
28.02.20	6,00	Rapport, table_module	6:19:19	MVP brukergrensesnitt
29.02.20	1,00	Oppdaterte kodevalitet-scriptene	1:29:29	MVP brukergrensesnitt
01.03.20			2:17:17	MVP brukergrensesnitt + gitlab ci runner config
02.03.20	7,00	Rapport (db-arkitektur, sammendrag), møte m/kontaktperson	6:24:24	MVP brukergrensesnitt + espen møte
03.03.20	7,00	Use case-diagram-sketch, rapport: modul-arkitektur, kravspek	1:47:47	MVP brukergrensesnitt + espen møte
04.03.20	2,00	Code review, prøve å fikse shortcuts med flere instanser	3:14:14	pdf preview,utvikling,utvikling
05.03.20	1,50	Samme som over	3:9:9	utvikling,utvikling,utvikling
06.03.20	4,00	Laget ModuleList-klassen	4:50:50	utvikling
07.03.20	1,00	Splittet main_window.py inn i moduler		
08.02.20	5,50	Kodebok-GUI		
09.03.20	8,00	Kodebok-GUI, fiksing i ModuleList	6:33:33	utvikling, review av MR, møte med espen
10.03.20	3,00	Statusmøte	2:10:10	internt statusmøte,utvikling
11.03.20	7,00	Kodebok-GUI robust redigering og pyntet	7:0:0	utvikling
12.03.20	6,00	Eksport av kodebok til PDF	4:0:0	utvikling rectpack og code review
13.03.20	7,00	Justering kodebok til PDF, legge til rad i CodeBook	6:53:53	utvikling rectpack og code review
14.03.20	4,00	Ferdigstilte CodebookRowAdder og sydde den inn med view	2:12:12	utvikling rectpack og code review
15.03.20			0:16:16	code review
16.03.20	7,00	Møte, kodebok-pdf har kode- og dekodebok	6:22:22	utvikling rectpack og code review,utvikling rectpack og code review
17.03.20	2,00	Code review	4:22:22	code review
18.03.20	5,00	Code review, ferdigstilte kodebok-pdf	5:47:47	implementere black pipelinetest,code review & rectpack
19.03.20	4,00	Minimøte, SOI-eksportering	1:5:5	statusmøte,code review
20.03.20	9,00	SOI-importering	6:38:38	utvikling legg til ny modul
21.03.20	5,00	Testing serialisering/eksport/import av SOI		
22.03.20	2,50	Gjennomgått MR, kodebok-gui slettet rad, testmodus db	1:8:8	code review & script for å få timer ut av toggl
23.03.20	9,00	Møte, rettet på MR's, laget CodebookWidget	6:52:52	internt statusmøte & rapport kravspek & code review
24.03.20	7,00	Rapport: SOI på fil, kodekvalitetscript. Gjennomgått MR.	4:45:45	rapport kravspek & code review
25.03.20	6,00	Rapport: figur/listing-oppdatering og db-tabell-beskrivelse	4:46:46	code review,rapport kravspek
26.03.20	4,00	Rapport, MR-fiksing og errorfiksing	1:22:22	rapport kravspek & code review
27.03.20	5,00	Gruppemøte, rapport	5:38:38	rapport kravspek & code review
28.03.20	2,00	Endring ModuleList, fikset SOI-import og utvidet tester, MR	1:36:36	utvikling
29.03.20			0:12:12	utvikling
30.03.20	6,00	Møte + autentiseringstavle	5:11:11	utvikling oppsett & møte & codereview & statusrapport & fritekst modul
31.03.20	4,00	Møte + fikset kodeduplisering	5:25:25	fritekst modul & statusrapport & utfylle kapittel 1
01.04.20	6,00	Oppgavebeskrivelse, MR-fiksing autentiseringstavle		
02.04.20	7,00	Møte veileder, subtraktorkoder	6:57:57	code review & frodemøte,code review,code review
03.04.20	7,00	Møte kontaktperson, ferdigstilling subtraktorkoder	5:52:52	utvikling header nytt design,utvikling header nytt design
04.04.20				
05.04.20	4,00	Oppsett timelogg, tab-lukking, Ny SOI-tab, oppgbeskrivelse		

06.04.20	4,00	Rapport-review, skrev om kodebok-GUI		
07.04.20				
08.04.20	0,50	Kodebok-GUI ferdigskrevet		
09.04.20	3,00	Lagre SOI i database		
10.04.20	6,00	Se/åpne SOI fra db, ferdigskrev database i teknisk design		
11.04.20				
12.04.20				
13.04.20	3,00	Autentiseringstavle 'element x og y' og småjusteringer		
14.04.20	5,00	SOI-lagring i db endringer og MR-fiksing	3:22:22	code review & nytt header design utvikling
15.04.20	6,50	Møte, SOI-db-åpning, autentiseringstavle-overskrift, rapport	6:10:10	nytt header design utvikling & møte, (støtte for vedlegg foreks.),utvikling
16.04.20	4,50	Modul-kode-font, MR-er	5:41:41	utvikling, støtte for vedlegg, støtte for "utskrift" av flere eksemplarer
17.04.20	6,50	Kodetabell-dialog, skrev aut-/subkoder-implementasjon, MR	5:9:9	utvikling, støtte for "utskrift" av flere eksemplarer
18.04.20				
19.04.20				
20.04.20	5,50	Møte, kodetabell-popup-MR, import-modules/attach-funksjon	6:7:7	utvikling, støtte for "utskrift" av flere eksemplarer + code review + fikse skjermopløsningsproblem
21.04.20	6,50	MR, forhåndsavtalte koder	4:23:23	utvikling, fikse skjermopløsningsproblem + code review
22.04.20	7,50	Forhåndsavtalte koder	6:42:42	fikse skjermopløsningsproblem,code review + kodefrasemodul
23.04.20	7,50	Forhåndsavtalte koder: popup med eksisterende modul	6:25:25	utvikling, kodefrasemodul
24.04.20	6,00	Forhåndsavtalte koder: dokumentasjon og testing	6:37:37	utvikling, kodefrasemodul
25.04.20	2,00	MR-fiksing forhåndsavtalte koder	1:48:48	code review
26.04.20	3,00	Forhåndsavtalte koder lik tabellbredde, la til kodebok-ord		
27.04.20	9,00	Gruppemøte, bugfikser, klargjør moduler for PDF-eksport	6:38:38	møte & codereview & forsøk på å fikse kodefrasemodul luft på høyre side, rectpack
28.04.20	7,50	Bra SOI på ark, MR		
29.04.20	6,00	Font-bug, klargjort kodefrasemodul for PDF-eksport, MR-er	7:12:12	tester til kodefrasemodul & fikse oppløsnings bug,code review
30.04.20	5,00	Enkel bruk, kodetabell-modifisering, MR-er	5:19:19	modulplassering,møte,forklaring algoritmer, statusrapport, modul padding, installasjonsinstrukser,fikse windows runner
01.05.20	6,50	SOI-fane-tittel-endring, MR-fiksing, MR-er	9:37:37	fikse windows runner, fikse telefonliste oppløsningstest, merge, fikse høyde tabeller..., enkel bruk scroll area
02.05.20	2,50	MR-er, frekvenstabell-fiksing	2:58:58	code review & installasjonsinstrukser
03.05.20	7	Klargjør for PDF-eksport, oppdater modul-bilder, MR	8:51:51	installasjonsinstrukser & code review & fikse moduler overlapper rammer & utvikling av zoom
04.05.20	6,5	Møte, finpussing, bugfixs	5:51:51	code review & reorganize knapp & møte,kravspesifikasjon 2.1 moduler & sjekke bugfixs av forh. koder
05.05.20	10	Funksjonelle og ikke-funksjonelle krav, klassediagram	9:11:11	intro, kravspek., beskrivelse av moduler, usecase, diskusjon klassediagram, installasjonsinstrukser
06.05.20	6,5	Klassediagram, reviews	11:1:1	rapportskriving & eksperimentering med eksportert trello-json
07.05.20	7	Teknisk design db og SOI-fil, møte m/ veileder, designprosess	8:42:42	eksperimentering med trello eksport,rapportskriving,møte med frode
08.05.20	4,25	Kodegenerering, implementasjon aut- og subtraktorkoder		
09.05.20				
10.05.20	7	Reviews, kodebok og kodebok-PDF-implementasjon		
11.05.20	7,5	Møte, modul-base, generisk tabell, forh. koder og diskusjon	7:54:54	møte & rapportskriving,rapportskriving
12.05.20			1:28:28	rapportskriving
13.05.20	0,5	Forhåndsavtalte koder-implementasjon	5:37:37	rapportskriving
14.05.20	1	Forhåndsavtalte koder	7:47:47	rapportskriving
15.05.20	6	Møte, utbedringer	10:48:48	rapportskriving
16.05.20	10,5	Dialog-implementasjon, rapportstruktur, ModuleImage	9:42:42	rapportskriving
17.05.20	7,5	Databasesynkronisering, reviews	9:36:36	rapportskriving
18.05.20	13,5	Rapportreviews og rapportgjennomgang	15:00:00	rapportreview
19.05.20	15	Rapportgjennomgang	15:30:00	rapportreview
20.05.20				

Vedlegg J

Eksempler på eksisterende SOI-er

Vedlegget inneholder eksempler på eksisterende SOI-er som ble gitt til gruppen av oppdragsgiver i starten av prosjektperioden.

J.1 ”SOI 2BN”



SOI 2BN

2BN VHF NET PRIMARY/ALTERNATE COMS

NET NAME	NET ID	TYPE	FRQ.mhz	SEC	SYNC MASTER	ACCESS	SMTF HOST NAME
2BN PRIMARY CMD VOICE/BMS:							
2BN CMD1	1401	CNR_FF	R	9/93A/93B/93C	MAUKE BN		
2BN CMD2	1402	CNR_FF	R	9/93A/93B/93C	MAUKE BN		
2BN CMD3	1403	CNR_FF	R	9/93A/93B/93C	MAUKE BN		
2BN CMD4	1404	CNR_FF	R	9/93A/93B/93C	MAUKE BN		
2BN CMD5	1405	CNR_FF	R	9/93A/93B/93C	MAUKE BN		
2BN ALTERNATE VOICE/ADM:							
2BN ADM/BMS1	1411	CNR_TD	R	9/93A/93B/93C	MAUKE BN		
2BN ADM/BMS2	1412	CNR_TD	R	9/93A/93B/93C	MAUKE BN		
2BN ADM/BMS3	1413	CNR_TD	R	9/93A/93B/93C	MAUKE BN		
2BN ADM/BMS4	1414	CNR_TD	R	9/93A/93B/93C	MAUKE BN		
2BN ADM/BMS5	1415	CNR_TD	R	9/93A/93B/93C	MAUKE BN		
BRAVO COY PRIMARY CMD VOICE/BMS:							
BPRI CMD/BMS	1420	CNR_TD	R	B-9	MAUKE BN		
BRAVO COY ALTERNATE VOICE							
B ALT VOICE	1421	CNR_FF	R	O/O	MAUKE BN		
BRAVO PLATOON NET:							
B1 PRI	1422	CNR_FF	R	B-11	2BN		
B1 ALT	1423	CNR_FF	R	O/O	2BN		
B2 PRI	1424	CNR_FF	R	B-21	2BN		
B2 ALT	1425	CNR_FF	R	O/O	2BN		
B3 PRI	1426	CNR_FF	R	B-31	2BN		
B3 ALT	1427	CNR_FF	R	O/O	2BN		
CHARLIE COY PRIMARY CMD VOICE/BMS:							
CPRI CMD/BMS	1430	CNR_TD	R	C-9	MAUKE BN		
CHARLIE COY ALTERNATE VOICE							
C ALT VOICE	1431	CNR_FF	R	O/O	MAUKE BN		
CHARLIE PLATOON NET:							
C1 PRI	1432	CNR_FF	R	C-11	2BN		
C1 ALT	1433	CNR_FF	R	O/O	2BN		
C2 PRI	1434	CNR_FF	R	C-21	2BN		
C2 ALT	1435	CNR_FF	R	O/O	2BN		
C3 PRI	1436	CNR_FF	R	C-31	2BN		
C3 ALT	1437	CNR_FF	R	O/O	2BN		
DELTA COY PRIMARY CMD VOICE/BMS:							
DPRI CMD/BMS	1440	CNR_TD	R	D-9	MAUKE BN		
DELTA COY ALTERNATE VOICE							
D ALT VOICE	1441	CNR_FF	R	O/O	MAUKE BN		
DELTA PLATOON NET:							
PATROL PRI	1442	CNR_FF	R	D-11	2BN		
PATROL ALT	1443	CNR_FF	R	O/O	2BN		
SNIPER PRI	1444	CNR_FF	R	D-21	2BN		
SNIPER ALT	1445	CNR_FF	R	O/O	2BN		
MUAS PRI	1446	CNR_FF	R	D-31	2BN		
MUAS ALT	1447	CNR_FF	R	O/O	2BN		
SPECIAL NOTES:							
1. GPS ZULU TIME ON ALL COMS - ALWAYS							
2. ESTABLISH COMS ACCORDING TO COMPLAN and COVERAGE MAP							
3. CODELIST TO BE USED ON UNCLASSIFIED COMMS							
4. NOPLAY EMERGENCY MESSAGES SHOULD BE STARTED WITH: "NO PLAY-NO PLAY"							
5. CHANGES IN SOI WILL BE MARKED WITH YELLOW MARKINGS							
6. ALL BMS COMPUTERS LOADED WITH Fisb-Taktisk V2.2.1							

NET NAME	NET ID	TYPE	FRQ.mhz	SEC	SYNC MASTER	ACCESS	SMTF HOST NAME
ECHO COY PRIMARY CMD VOICE/BMS:							
EPRI CMD/BMS	1500	CNR_TD	R	E-9	MAUKE BN		
ECHO COY ALTERNATE VOICE							
E ALT VOICE	1501	CNR_FF	R	O/O	MAUKE BN		
COMBAT SUPPORT PLATOON NET:							
SIGNAL PRI	1510	CNR_TD	R	E-19	2BN		
MED PRI	1520	CNR_FF	R	E-29	2BN + SANBN		
IMED ALT	1521	CNR_FF	R	O/O	2BN + SANBN		
SUPPLY PRI	1530	CNR_FF	R	E-39	2BN + STRBN		
SUPPLY ALT	1531	CNR_FF	R	O/O	2BN + STRBN		
MORTAR NET A	1560	CNR_TD	R	E-69A	2BN + ARTBN		
MORTAR NET B	1561	CNR_TD	R	E-69B	E-1/E-6/6-S		
2BN FS 1	1562	CNR_TD	R	O/O	E-1/E-6/6-S		
2BN FS 2	1563	CNR_TD	R	O/O	E-1/E-6/6-S		
2BN FS 3	1564	CNR_TD	R	O/O	E-1/E-6/6-S		
2BN FS 4	1565	CNR_TD	R	O/O	E-1/E-6/6-S		
JAVELIN PRI	1570	CNR_FF	R	E-79	2BN		
JAVELIN ALT	1571	CNR_FF	R	O/O	2BN		
JOINT ACTION LINK-UP FRQ.							
CGG FM	1070	CNR_FM		62.300 U	ALL		
CGG FF	1071	CNR_FF	R	O/O	NOR DEF		
CAG FM	1072	CNR_FM		64.550 U	ALL		
CAG FF	1073	CNR_FF	R	O/O	NOR DEF		
CGS FM	1065	CNR_FM		74.825 U	ALL		
CGS FF	1066	CNR_FF	R	O/O	NOR DEF		
2BN SAMV 1	1057	CNR_FF	R	O/O	BRIG-N		
2BN SAMV 2	1058	CNR_FF	R	O/O	BRIG-N		
2BN SAMV 3	1059	CNR_TD	R	O/O	BRIG-N		
2BN SPECIAL FREQUENCIES							
2BN - 339	1586	CNR_FM		73.650 U	ALL		
2BN RESERVE 1	1057	CNR_FF	R	O/O	MAUKE BN		
2BN RESERVE 2	1058	CNR_FF	R	O/O	MAUKE BN		
2BN RESERVE 3	1059	CNR_TD	R	O/O	MAUKE BN		
2BN FM 1	1598	CNR_FM		38.475 U	ALL		
2BN HF CONTINGENCY COMS							
NAME	TIME	MODE	CRYPTO	VALID	TEK/TYPE	TOD SERVER	
2BN HF DAY	0700-1859Z	3G	NMST 261 - G - 003 SEPTEMBER 16	1/ANDVT	SKOLD HQ		
2BN HF NIGHT	1900-0659Z	3G	NMST 261 - G - 003 SEPTEMBER 16	1/ANDVT	SKOLD HQ		

2BN EMERGENCY COMS

CONNECTED TO SIV INFRASTRUCTURE

FUNCTION

NUMBER

SEC

FUNCTION

NUMBER

2BN CP

40442600

U

TVS PHONE NUMBER

317-1000

2BN COMMANDER

40442601

U

NOR BMS

N2.HQ

2nd IN COMMAND

40442602

U

XO MAIL ADR

BDEN 2BN CP A

B COY COMMANDER

40442720

U

C COY COMMANDER

40442730

U

FUNCTION

2BN 91A/TACCP

D COY COMMANDER

40442740

U

TVS PHONE NUMBER

317-1009

E COY COMMANDER

40442750

U

NOR BMS

N2.91A

IRIDIUM PHONE LIST

FILE SHARING TO HQ

93

DESKTOP FOLDER

FUNCTION

XO MAIL ADR

SEC

BDEN 2BN CMDGRP

2BN CP

00881692757423

U

CONTNAP SYSTEMS AND NUMBERS#

BRAVO COY ALTERNA

00881692757425

U

FUNCTION

2BN CPR/SKIOLD HQ

B COY

00881692757426

U

TVS PHONE NUMBER

317-0000

BRAVO PLATOON NET

00881651437221

U

NOR BMS

D COY

00881651437224

U

XO MAIL ADR

BDEN 2BN CP R

E COY

00881651437219

U

CALL FROM TVS TO SIV:

CALL FROM IRIDIUM TO SIV; Dial 0047 and number

2 BN UNCLASSIFIED MAIL ADDRESS

2.BN.UGRADERT@GMAIL.COM

CODES & SPINNS

CALL TO TVS FROM SIV/FDN:

Dial (00) 03003 ask for transfer to TVS NR#

AUTHENTICATING BOARD

6

2

8

3

C

Z

O

D

A

N

H

K

P

X

M

R

G

J

U

B

V

Q

Y

W

S

F

L

I

T

Example:

"AUTHENTICATE CHARLIE - 8"

"AUTHENTICATING DELTA"

"AUTHENTICATE BRAVO - 3"

"AUTHENTICATING WHISKEY"

RAMROD

0

1

2

3

4

5

6

7

8

9

H

U

N

T

E

R

B

O

Y

S

LIBRARY

START RADIO SILENCE

CIRCUS

END RADIO SILENCE

MOUTHPIECE

COMPROMIZED RADIO

Apple

GO TO BRIG-N CAG PT LINK UPP FRQ

EAGLE

GO TO BRIG-N CAG PT LINK UPP FRQ

MUD

GO TO 2BN FM 1

DIRT

TKN SYSTEMS AND NUMBERS#

UNIT

BMS CALLSIGN

2BN ADM/BMS

X PRI CVID/BMS

GW

UNIT

BMS CALL SIGN

2BN CVID/BMS

X PRI CVID/ GW

2BN SKIOLD

N2.SKIOLD HQ

2BN ADM/BMS

2BN ADM/BMS

E79

N2.E79

2BN ADM/BMS

2BN HQ

N2.HQ

2BN ADM/BMS

2BN ADM/BMS

X

91A

N2.91A

2BN ADM/BMS

2BN ADM/BMS

92A

N2.92A

2BN ADM/BMS

2BN ADM/BMS

9

N2.9

2BN ADM/BMS

2BN ADM/BMS

9A

N2.9A

2BN ADM/BMS

2BN ADM/BMS

9E

N2.9E

2BN ADM/BMS

2BN ADM/BMS

93

N2.93

2BN ADM/BMS

2BN ADM/BMS

B9

N2.B9

2BN ADM/BMS

BPRI CVID/BMS

X

B91A

N2.B91A

2BN ADM/BMS

2BN ADM/BMS

B11V

N2.B11V

BPRI CVID/BMS

B31V

N2.B31V

BPRI CVID/BMS

C9

N2.C9

2BN ADM/BMS

2BN ADM/BMS

X

CONNECTED TO 2BN BMS ON HF CHANNEL

C91V

N2.C91A

2BN ADM/BMS

2BN ADM/BMS

SKIOLD

N2.SKIOLD HQ

C9

N2.C9

C11V

N2.C11V

2BN ADM/BMS

2BN ADM/BMS

HQ

N2.HQ

C91

N2.C91A

C21V

N2.C21V

2BN ADM/BMS

2BN ADM/BMS

9

N2.9

E-29

N2.E29

C31V

N2.C31V

2BN ADM/BMS

2BN ADM/BMS

91A

N2.91A

E-39

N2.E39

D9

N2.D9

2BN ADM/BMS

2BN ADM/BMS

X

D92A

N2.D92A

2BN ADM/BMS

2BN ADM/BMS

D92A

N2.D92A

E-69A

N2.E69A

D11V

N2.D11V

2BN ADM/BMS

2BN ADM/BMS

D11

N2.D11

E-69B

N2.E69B

D12A

N2.D12A

2BN ADM/BMS

2BN ADM/BMS

D12

N2.D12

E-651

N2.E651

D21B

N2.D12B

2BN ADM/BMS

2BN ADM/BMS

D12

N2.D12

E-652

N2.E652

D21V

N2.D13V

2BN ADM/BMS

2BN ADM/BMS

D21

N2.D21

E-653

N2.E653

D1A1

N2.D14V

2BN ADM/BMS

2BN ADM/BMS

D21P

N2.D21P

E-79

N2.E79

B31P

N2.D31P

2BN ADM/BMS

2BN ADM/BMS

D22P

N2.D22P

E-7A

N2.E7A

D32P

N2.D32P

2BN ADM/BMS

2BN ADM/BMS

D23P

N2.D23P

E-7B

N2.E7B

E9

N2.E9

2BN ADM/BMS

2BN ADM/BMS

D24P

N2.D24P

E-7C

N2.E7C

E91A

N2.E91A

2BN ADM/BMS

2BN ADM/BMS

D31P

N2.D31P

E-7D

N2.E7D

E19

N2.E19

2BN ADM/BMS

2BN ADM/BMS

B9

N2.B9

73A

N2.93A

2BN ADM/BMS

2BN ADM/BMS

B91

N2.B91A

73B

N2.93B

2BN ADM/BMS

2BN ADM/BMS

73C

N2.93C

2BN ADM/BMS

2BN ADM/BMS

E29

N2.E29

2BN ADM/BMS

2BN ADM/BMS

E2A

N2.E2A

2BN ADM/BMS

2BN ADM/BMS

E2B

N2.E2B

2BN ADM/BMS

2BN ADM/BMS

E21

N2.E21

2BN ADM/BMS

2BN ADM/BMS

E39

N2.E39

2BN ADM/BMS

2BN ADM/BMS

E3R

N2.E3R

2BN ADM/BMS

2BN ADM/BMS

E3J

N2.E3J

2BN ADM/BMS

2BN ADM/BMS

E3M

N2.E3M

2BN ADM/BMS

2BN ADM/BMS

E3N

N2.E3N

2BN ADM/BMS

2BN ADM/BMS

E41

N2.E41

2BN ADM/BMS

2BN ADM/BMS

E42

N2.E42

2BN ADM/BMS

2BN ADM/BMS

E69A

N2.E69A

2BN ADM/BMS

2BN ADM/BMS

E69B

N2.E69B

2BN ADM/BMS

2BN ADM/BMS

1. Kontroller NET ID

2. Kontroller fysisk kabling (Strøm, antenne, microtelefon)

3. Kontroller tid/synk status og sendeefekt

4. Kontroller at du er indenfor dækningsboble fra rele

5. Restart Sender-Mottaker

6. Kontroller endringer i området (Hørspennt, aggregat, lammere)

7. Forflytt deg til siste posisjon hvor du hadde samband

8. Kontroller kryptonektel og TMR ID

Page 2 of 2

RESTRICTED\\REL NOR

J.2 "SOI ENGRBN"

SOI ENGRBN

2015 trenings-SOI

Til bruk ved kurs/undervisning

Nr _____

BEGRENSET

Sikkerhetsloven § 11 og § 12 Jf Offentlighetsloven § 13.

NET	NET ID	NET NAME	TYPE	Område	FRQ
Hæren Brig-N INGBN	2000	INGBN1	CNR_TD	Norge	
Hæren Brig-N INGBN	2001	INGBN2	CNR_TD	Norge	
Hæren Brig-N INGBN	2002	INGBN3	CNR_TD	Norge	
Hæren Brig-N INGBN	2003	INGBN4	CNR_TD	Norge	
Hæren Brig-N INGBN KP Samvirke	2090	INGBN_S_1	CNR_TD	Norge	
Hæren Brig-N INGBN KP Samvirke	2091	INGBN_S_2	CNR_TD	Norge	
Hæren Brig-N INGBN	2092	INGBN_R_1	CNR_TD	Norge	
Hæren Brig-N INGBN	2093	INGBN_R_2	CNR_TD	Norge	
Hæren Brig-N INGBN	2095	INGBN_R_4	CNR_TD	Norge	
Hæren Brig-N INGBN Pingkp A	2070	PINGKPA1	CNR_TD	Norge	
Hæren Brig-N INGBN Pingkp A	2071	PINGKPA2	CNR_FF	Troms	
Hæren Brig-N INGBN Pingkp A	2072	PINGKPA3	CNR_FF	Troms	
Hæren Brig-N INGBN Pingkp A	2073	PINGKPA4	CNR_FF	Troms	
Hæren Brig-N INGBN Pingkp A	2074	PINGKPA5	CNR_FF	Troms	
Hæren Brig-N INGBN Pingkp A	2075	PINGKPA6	CNR_FF	Troms	
Hæren Brig-N INGBN Ingkp B	2080	INGKP1	CNR_TD	Norge	
Hæren Brig-N INGBN Ingkp B	2081	INGKP2	CNR_FF	Troms	
Hæren Brig-N INGBN Ingkp B	2082	INGKP3	CNR_FF	Troms	
Hæren Brig-N INGBN Ingkp B	2083	INGKP4	CNR_FF	Troms	
Hæren Brig-N INGBN Ingkp B	2084	INGKP5	CNR_FF	Troms	
Hæren Brig-N INGBN Ingkp B	2085	INGKP6	CNR_FF	Troms	
Hæren Brig-N INGBN Pingkp C	2030	PINGKPC1	CNR_TD	Norge	
Hæren Brig-N INGBN Pingkp C	2031	PINGKPC2	CNR_FF	Norge	
Hæren Brig-N INGBN Pingkp C	2032	PINGKPC3	CNR_FF	Norge	
Hæren Brig-N INGBN Pingkp C	2033	PINGKPC4	CNR_FF	Norge	
Hæren Brig-N INGBN Pingkp C	2034	PINGKPC5	CNR_FF	Norge	
Hæren Brig-N INGBN INGKP-4	2050	INGKP4-1	CNR_TD	Norge	
Hæren Brig-N INGBN INGKP-4	2051	INGKP4-2	CNR_FF	Troms	
Hæren Brig-N INGBN INGKP-4	2052	INGKP4-3	CNR_FF	Troms	
Hæren Brig-N INGBN INGKP-4	2053	INGKP4-4	CNR_FF	Troms	
Hæren Brig-N INGBN INGKP-4	2054	INGKP4-5	CNR_FF	Troms	
Hæren Brig-N INGBN M&KKP	2010	MoKKP1	CNR_TD	Norge	
Hæren Brig-N INGBN M&KKP	2011	MoKKp2	CNR_FF	Norge	
Hæren Brig-N INGBN M&KKP	2012	MoKKP3	CNR_FF	Norge	
Hæren Brig-N INGBN INGKP-6	2040	INGKP6-1	CNR_TD	Norge	
Hæren Brig-N INGBN INGKP-6	2041	INGKP6-2	CNR_FF	Troms	
Hæren Brig-N INGBN INGKP-6	2042	INGKP6-3	CNR_FF	Troms	
Hæren Brig-N INGBN INGKP-6	2043	INGKP6-4	CNR_FF	Troms	
Hæren Brig-N INGBN INGKP-6	2044	INGKP6-5	REP_ONE	Troms	

Crypto information

Key Type	Group-ID	Key-ID; AUGUST / SEPT / OKT
Traffic	2094	1234

ALTERNATIVE COMMS

	Mobil/ugradert	(FDN)/Tadkom	Iridium (ugrad.)
INGBN OPS Devdis	77 89 21 16	0580 2116	
INGBN OPS felt	488 84 500		
BRIGKO	400 29 999	(0)580 8500	
PingKp-1	488 88 511		+88 165 1437 527
IngKp-2	488 88 521		+88 165 1437 528
PingKp-3	488 88 531		
IngKp-4	488 88 541		+88 165 1437 525
IngKp-5	488 88 551		
IngKp-6	488 88 561		+88 165 1437 524
Skredgruppe			+88 165 1437 311

ANTI JAM MEASUREMENTS

NET ID	NET NAME	Minutes after full hour
2092	INGBN_R_1	0
2093	INGBN_R_2	15
2094	INGBN_R_3	30
2091	INGBN_S_2	45

SUBTRACTOR CODES

A H O V	70272	68330
B I P W	81214	00146
C J Q X	40538	45874
D K R Y	94351	46398
E L S Z	49498	41732
F M T	15900	01052
G N U	82737	55393

RADIO SILENCE

Codeword start
Apple
Codeword stop
Android

AUTHENTICATION BOARD

A N 0	SKPAEMUOINVDVCLWZBFHRTJGQ
B O 1	ANXBRYOHPCLKIZESDTUFQMJGWW
C P 2	MHPQAIFKEUGBSYWDZLORVXNTC
D Q 3	KOLEVHQSBMIUXCJNDFWZAGTRYP
E R 4	TRIWXSZBAQDGNVUCHLPOMEFYJ
F S 5	QDKHYGOVIRNXSPCZBJMWAEUTLF
G T 6	VRYLKCOQEBMIUXSNJHPDFWZAGT
H U 7	JVXFUKLDWGAPMZOCYCHRQBSNT
I V 8	AGTBSKPEDCFUNOVYIHJXZLRWM
J W 9	HXIBYDULRMPJSWOJECQNVZGTFAK
K X	GHFAWDSQERJKOYULMVICXNZTPB
L Y	FGDHPBQZLIJSACOEXWUMTKNRY
M Z	PCQYEMJWOGUFKHVZNXSDITRBAL

Testelement messageaut 4 and 7 letter/digit of the text.

BRIG-N NET (interaction)	NET ID	NET NAME	TYPE	FRQ
INTERACTION				
Stridstren Samvirke Net	1050	STRBN SAMV	CNR_FF	
PBn Samvirke Net 1	1051	PBN SAMV 1	CNR_FF	
PBn Samvirke Net 2	1052	PBN SAMV 2	CNR_FF	
PBn Samvirke Net 3	1053	PBN SAMV 3	CNR_TD	
TMBN Samvirke Net 1	1054	TMBN SAMV 1	CNR_FF	
TMBN Samvirke Net 2	1055	TMBN SAMV 2	CNR_FF	
TMBN Samvirke Net 3	1056	TMBN SAMV 3	CNR_TD	
2.Bn Samvirke Net 1	1057	2BN SAMV 1	CNR_FF	
2.Bn Samvirke Net 2	1058	2BN SAMV 2	CNR_FF	
2.Bn Samvirke Net 3	1059	2BN SAMV 3	CNR_TD	
(U) Common Ground-to-Ground FM	1060	CGG FM	CNR_FM (U)	
Common Ground-to-Ground FF	1061	CGG FF	CNR_FF	
(U) Common Air-to-Ground FM	1062	CAG FM	CNR_FM (U)	
Common Air-to-Ground FF	1063	CAG FF	CNR_FF	
(U) Common Ground-to-Sea FM	1065	CGS FM	CNR_FM (U)	
Common Ground-to-Sea FF	1066	CGS FF	CNR_FF	
Brig-N Samvirke Net FF 1	1067	BN SAMV FF 1	CNR_FF	
Brig-N Samvirke Net FF 2	1068	BN SAMV FF 2	CNR_FF	
Brig-N Samvirke Net FF 3	1069	BN SAMV FF 3	CNR_FF	
Brig-N Samvirke Net TD 1	1070	BN SAMV TD 1	CNR_TD	
Brig-N Samvirke Net TD 2	1071	BN SAMV TD 2	CNR_TD	
Brig-N Samvirke Net TD 3	1072	BN SAMV TD 3	CNR_TD	

Prosedyre ved manglende VHF-forbindelse

1. Kontroller NET ID

2. Kontroller fysisk kabling (Strøm, antenne, microtelefon)

3. Kontroller tid/synk status og sendeeffekt

4. Kontroller kryptonøkkel

5. Kontroller at du er innenfor dekningsboble fra rele

6. Kontroller endringer i området (store antenner, aggregat, jammere)

7. Forflytt deg til siste posisjon hvor du hadde samband

Time slots HF reporting

All timeslots are in GMT ZULU time.

Pingkp-1	00:00 - 00:15	06:00 - 06:15	12:00 - 12:15	18:00 - 18:15
Ingkp-2	00:15 - 00:30	06:15 - 06:30	12:15 - 12:30	18:15 - 18:30
Pingkp-3	00:30 - 00:45	06:30 - 06:45	12:30 - 12:45	18:30 - 18:45
Ingkp-4	00:45 - 01:00	06:45 - 07:00	12:45 - 13:00	18:45 - 19:00
Ingkp-5	01:00 - 01:15	07:00 - 07:15	13:00 - 13:15	19:00 - 19:15
Ingkp-6	01:15 - 01:30	07:15 - 07:30	13:15 - 13:30	19:15 - 19:30

Other units use the remaining slots and stay of net in listed time slots.

Reporting iat. Engrbn sync matrix.

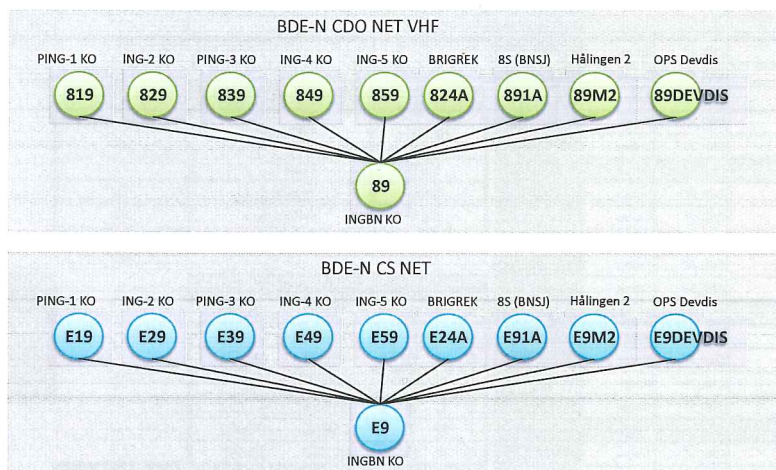
Current RPA Ver:

INGBN HF NET: INGBN HF NET v 1501.RPD

There will be no day/night channel. The same channel will be used 24h.

Crypto information BRIG-N NET (interaction)

Key Type	Group-ID	Key-ID; AUG / SEPT / OKT
Traffic	1051 / 1060	1234



CALLSIGN HF INGBN	
Enhet	Callsign
BN KO	N89
OPS DEVDIS	N89DEVDIS
BN SJEF	N891A
Hålingen	N89M1
Hålingen	N89M2
PING 1 CP	N819CP
PING 1 KP SJEF	N8191A
REK TR. KO	N816A
REK TR. lag 1	N8161
REK TR. Lag 2	N8162
ING 2 CP	N829CP
ING 2 KP SJEF	N8291A
REK TR. KO	N822A
REK TR. Lag 1	N8221
REK TR. Lag 2	N8222
BRIG REK TR.KO	N824A
BRIG REK Lag 1	N8241
BRIG REK lag 2	N8242
BÅT Tropp	N823A
Dykkerlag	N8231
PING 3 CP	N839CP
PING 3 KP SJEF	N8391A
REK TR. KO	N836A
REK TR. Lag 1	N8361
REK TR. Lag 2	N8362
ING 4 CP	N849CP
ING 4 KP SJEF	N8491A
MASKIN Tr. KO	N841A
OVS TR. KO	N842A
FUCHS	N8431
FUCHS	N8441
ING 5 CP	N859CP
ING 5 KP SJEF	N8591A
MASKIN tr. 1	N852A
MASKIN tr. 2	N853A
OVS Tropp	N854A
FUCHS	N8551

ENGBN CODEWORDLIST													
Warningorder for use of codewords: ADSAM													
Supplies		Actions		Nationality		Letters		Comms		Weapon equipment		Miscellaneous	
I7C	Battery CR123	S3Y	Establish	P0B	CAN	G7C	A	Y4E	RA 5800	D8N	12.7mm machin gun	K1Y	Medevac
I7J	Battery for LFR	S8D	Evacuate	P5F	DAN	G4K	B	Y0L	Antenna HF, Dipol	D2A	40mm	K4Z	Rotary wing
I5Q	Battery for MRR	S8N	Fire heard	P4A	ESP	G8S	C	Y5U	Antenna HF, Portable	D6H	AG-3	K8L	Supply convoy
I5T	F34 (liter) (Diesel)	S3O	Fire support	P9A	FIN	G5W	D	Y2Y	Antenna LFR	D0Q	Hand grenade	K5T	Unloading point
I8D	Flashlightbatt / BA-3030 (Stk)	S3U	Halt / Stop	P8H	GER	G0B	E	Y9X	Antenna MRR stationary	D7T	HK-416		
I1M	Food (DOS)	S1C	Have observed	P7O	HUN	G9L	F	Y5X	Antenna MRR, portable	D4J	IED		
I9M	Gasoline (liter)	S3P	IED	P9B	ITA	G7X	G	Y3I	FISBasis Tactical	D1J	M100		
I2R	Lightsticks (ea)	S2S	Meet in grid	P8F	JOR	G9A	H	Y3P	NorBMS (Facnav)	D9U	M19		
I4X	Lithiumbatt BA 5590 (ea)	S6Z	Need recovery	P1O	LAT	G4W	I	Y1W	HCOM	D9A	M72		
I0J	Toilet paper	S5Y	Move	P9N	NED	G7K	J	Y1Z	HF	D7I	MG-3		
I2L	Water (liter)	S5Q	Need repair support	P7A	NOR	G5N	K	Y9C	Iridium	D7K	Mine		
I8T	Penlightbatt / BA 3058 (ea)	S6V	Need supplies	P8G	POR	G8W	L	Y2L	LFR	D0V	Mortar		
		S8X	OP	P2H	RUM	G0G	M	Y5S	Mobilphone	D1D	P80 pistol/Pistol		
	Ammo/explosives	S4F	Patrol	P6T	SWE	G3L	N	Y6E	MRR	D5I	MP5		Engineer specific
Q4K	12.7MM (single shot)	S9O	Recon area	P0R	UK	G4R	O	Y7E	PC	D6O	RCIED	U6I	Compound
Q3E	12.7mm cartridges	S5S	Recon designated area	P5Z	US	G7B	P	Y4Z	PFR	D7E	RPG	U9O	Construction
Q1P	30mm arrow	S3H	Rocenoitro axis			G8X	Q	Y6X	PRR	D0K	SBIED	U0D	Crossing
Q1W	30mm MP	S9Q	Relieve	Unit		G0V	R	Y6L	GPS	D5T	Signal pistol	U6P	Heights
Q6T	5.56 cartridge	S7C	Request	C6S	Brig N	G9J	S			D2X	Smoke grenade	U8R	Lake
Q9G	7.62 cartridge	S7I	Search	C3E	Ingbn	G4O	T	Unit size		D6L	Sniper rifle	U4Z	Moske
Q7J	7.62 cartridge for MG	S7M	Sit rep	C3K	2. bn	G1O	U	F5B	Battalion			U9I	Open area
Q0S	9 mm cartridge	S3V	State position	C1B	Pbn	G9Z	V	F5I	Company	Expression		U6M	Path
Q2C	Flashbang	S6F	Stronghold	C1E	TMBN	G9G	W	F3P	Platoon	H0D	South-West	U1Q	River bed
Q7E	Hand Grenade, fragmental	S7P	Switching of comms	C9H	Kesk	G7N	X	F3B	Squad	H9K	West	U0B	Road
Q8K	Hand Grenade, stunn	S1U	Troops in contact	C2Q	Signbn	G7P	Y	F4P	Task force	H9N	North-West	U7Y	Route
Q2L	High-explosive shell	S2A	VCP	C3A	Artbn	G0A	Z			H0A	North - South	U7F	Valley
Q6Y	illuminated flare	S7Y	You must	C6G	Medbn	G0Z	Æ	Vehicles		H2K	East-West	U5M	Village
Q2G	M72			C8L	StrTrbn	G1J	ø	A8D	Armoured MB (Vehicle)	H1J	Up	U8Z	Bridge
Q4E	Mortar illumination			C1N	HMKG	G7V	Å	A0J	BV 206	H4O	Down		
Q4S	Mortar smoke			C6Z	SOF			A6V	M113 F3	H6U	Right		
Q9X	Signal ammo (Coloured)							A8X	M113 205 (APC)	H4D	Left		
Q5X	Smoke grenade					Numbers		A4E	ATV				
						J5Q	0	A9N	AVLB				
						J0V	1	A7O	CV 9030				
						J0G	2	A2S	Iveco				
						J5G	3	A2D	Leo 1				
						J5N	4	A7D	Leo 2				
			El optic equipment			J3U	5	A7K	Man (vehicle)				
						T9H	Binoculars	J3X	6	A5R	PE-2200, generator		
						T3N	Laser range finder	J4L	7	A7E	PE-230, generator		
						T4T	Night vision binoculars	J6D	8				
						T0E	Night vision goggles	J5D	9				
						T2G	Sniper sight						

BEGRENSET

Sikkerhetsloven § 11 og § 12 J-Offendighetsloven § 13.

J.3 ”SOI HV-05 Generell”



SOI HV-05 Generell

MRR				HF			Hyttera		Autentiseringstavle		Subtraktorkode
Nett	ADR	Frekvenser	Nett-ID	NIS	Frekvenser	Kodeord	NIS	Kanal nr	0	A	A
DISTR	QHQB								1	B	B
	QVY								2	C	C
	QHZ								3	D	D
GREBE	HPT								4	E	E
	HQM								5	F	F
	AMW								6	G	G
VALDRES	DXK								7	H	H
	CXX								8	I	I
	PZZ								9	J	J
	PZB								K		K
LILLEHAMMER	KKU								L		L
	KNA								M		M
	ETN								N		N
	EQX								O		O
HADELAND	KFQ								P		P
	FZG								Q		Q
	CJS								R		R
	DKU								S		S
RESERVE	AJR								T		T
	RGJ								U		U
	CES								V		V
RADIOTAUSHET :									W		W
START :	POKER								X		X
STOPP :	MONOPOL								Y		Y
L E Tanum, Major/G-6									Z		Z
Testelemt ved meldingsautentisering : 2 og 8 bokstav/tall i meldingen du skal sende.											
SOI mistet/kompromittert : kodeord : ARMAGEDDON										E UGRADERT	

FORHÅNDSAVTALTE KODER

Varslingsord: OTER

KODEGRUPPE: A

Etterforsyning

A	Proviant (DOS)	F enheter
B	Drivstoff (l)	F enheter
C	Meldingsblankett (blokker)	F enheter
D	Parafin	F enheter
E	1077-batteri / NBA-5598 (stk)	F enheter
F	Vann (l)	F enheter
G	Toalettpapir (ruller)	F enheter
H	Strømper petromax (stk)	F enheter
I	Rødsprit (l)	F enheter
J	Blyfri bensin (l)	F enheter
K	Bensin (l)	F enheter
L	SEM-batteri / BA-3052 (stk)	F enheter
M	Lommelyktbatteri / BA-3030 (stk)	F enheter
N	Stridsjournal (blokker)	F enheter
O	F-34 (l)	F enheter
P	Lommelykt pærer (stk)	F enheter
Q	PACE-batteri /BA-3090 (stk)	F enheter

KODEGRUPPE: B

Retninger

A	SØR
B	VEST
C	NORD
D	Frem
E	Tilbake
F	ØST

KODEGRUPPE: K

Uttrykk/tiltak/opdrag

A	Går av nett
B	Trenger forsyninger
C	Sett i marsj
D	Bekreft, -er
E	Har observert
F	Sitrap
G	Gjennomsøkt, -e, -er
H	Evakuer, -er
I	Møt meg i rute
J	Er i rute
K	Iverksett, -er
L	Oppgi posisjon
M	Avløse, -er
N	Skyting hørt
O	Ankommet
P	Trenger repstøtte/bergning
Q	Anmoder om
R	Du/dere må
S	Fortsett, -e, -er
T	Flytt, -e, -er (til)
U	Stans, -e, -et
V	Etabler, -er

KODEGRUPPE: C

Materiell

A	PE-2200, aggregat
B	MK - 456/GRC - relekabel
C	Telt 14 m2
D	MB-290 (Feltvogn)
E	BV-206 (beltevogn)
F	Scania, P93 (lastevogn)
G	GPS
H	HF - 3
I	PE-230, aggregat
J	MB-240 (Feltvogn)
K	Multifuel
L	Primus
M	Ved

KODEGRUPPE: D

Defekt SB-materiell

A	NORTaC
B	GP 900 Motorola
C	NO/PRC 1077(VHF)
D	AN/PRC-77 (VHF)
E	Repeater GP 900
F	GPS (navigasjonssett)
G	PACE (krypto)
H	NO/PRC-111 (HF)
I	SEM-52N (VHF)
J	Veksler 10 linjer
K	AS-102 (jordplanantenne)

KODEGRUPPE: J

Defekt el.optisk materiell

A	Nattbriller
B	Nattkikkert
C	Kikkert
D	Laser avstandsmåler

KODEGRUPPE: I

Våpenteknisk

A	Lysrakett
B	40 mm
C	Håndgranat
D	P-80
E	Bombe
F	Ammo
G	Signalpistolet
H	ERYX
I	MP-5
J	RFK
K	MG-3
L	Mine
M	UXO
N	AG-3

KODEGRUPPE: E

Mil avd

Posisjoner

A	Kompani	G	BN HQ
B	Tropp	H	KP HQ
C	Lag	I	CAMP
D	Bataljon	J	VKP
E	Kompanistridsgr	K	OP
F	Brigade	L	KHK

KODEGRUPPE: F

Tall

A	3	F	7
B	1	G	2
C	9	H	5
D	0	I	8
E	4	J	6

KODEGRUPPE: G

Bokstaver

A	- M	P	- Z
B	- Ø	Q	- E
C	- N	R	- V
D	- L	S	- Æ
E	- J	T	- Q
F	- F	U	- Å
G	- Y	V	- O
H	- P	W	- C
I	- W	X	- S
J	- K	Y	- H
K	- B	Z	- G
L	- U	Æ	- A
M	- X	Ø	- I
N	- T	Å	- R
O	- D		

KODEGRUPPE: H

Støtte

A	Bergning
B	BV transport
C	Egne styrker
D	Helikopter
E	Skarpskytter
F	EOD
G	MP
H	Sting
I	Dog
J	San

KODEGRUPPE: L

Landemerker

A	Kryss	E	Rute
B	Sti	F	Elv
C	Veg	G	Dal
D	Høyde	H	Bro

EKSEMPEL

(Alle kodemeldingene innledes med varslingsordet)

Anmoder om EOD og DOG: KQ HF GV GZ HI

Trenger forsyninger bensin 60 l: KB AK FJ FD

J.4 "SOI for HVS SBSK"



SOI FOR HVS SBSK

MRR NETS	ID/CH	NAME	TYPE	Z	KEY ID	FRQ	RADIO
CMD Net							
CMD 1 TD / FF	1000 / 1001	KL13_CDO / FF	TD / FF	NR	1000	30,175	1
13_CDO_REP1	1002	13_CDO_REP1	REP_FF	NR	1000	30,175/72,000	1
13_CDO_REP2	1003	13_CDO_REP2	REP_FF	NR	1000	30,175/79,250	1
KL13_CDO_ONE	1004	KL13_CDO_ONE	REP_ONE	NR	1000	30,175	1
CMD 2 TD / FF / REPONE	1011/1021/1031	CDO2/FF2/ONE2	TD/FF/RO	NR	1000	37,125	1
CMD 3 TD / FF / REPONE	1012/1022/1032	CDO3/FF3/ONE3	TD/FF/RO	NR	1000	74,575	1
CMD 4 TD / FF / REPONE	1013/1023/1033	CDO4/FF4/ONE4	TD/FF/RO	NR	1000	78,050	1
CMD 5 TD / FF / REPONE	1014/1024/1034	CDO5/FF5/ONE5	TD/FF/RO	NR	1000	34,250	1
CMD 6 TD / FF / REPONE	1015/1025/1035	CDO6/FF6/ONE6	TD/FF/RO	NR	1000	45,500	1
CMD 7 TD / FF / REPONE	1016/1026/1036	CDO7/FF7/ONE7	TD/FF/RO	NR	1000	47,125	1
CMD 8 TD / FF / REPONE	1017/1027/1037	CDO8/FF8/ONE8	TD/FF/RO	NR	1000	72,000	1
CDO COMMON		HEIMDAL	FM	NS	H2151	148,325	3,4
KPA							
CMD TD	1050	KPA_CDO	TD	NR	1000	78,975	1
CMD FF	1051	KPA_CDO_FF	FF	NR	1000	78,975	1
CMD REP	1052	KPA_CDO_REP	REP_FF	NR	1000	78,975/47,125	1
CMD REPONE	1053	KPA_CDO_ONE	REP_ONE	NR	1000	78,975	1
KPB							
CMD TD	1060	KPB_CDO	TD	NR	1000	32,375	1
CMD FF	1061	KPB_CDO_FF	FF	NR	1000	32,375	1
CMD REP	1062	KPB_CDO_REP	REP_FF	NR	1000	32,375/78,050	1
CMD REPONE	1063	KPB_CDO_ONE	REP_ONE	NR	1000	32,375	1
KPC							
CMD TD	1070	KPC_CDO	TD	NR	1000	33,900	1
CMD FF	1071	KPC_CDO_FF	FF	NR	1000	33,900	1
CMD REP	1072	KPC_CDO_REP	REP_FF	NR	1000	33,900/74,575	1
CMD REPONE	1073	KPC_CDO_ONE	REP_ONE	NR	1000	33,900	1
COMMON							
PLAIN 1	1090	PANIC	FM	U	-	37,725	1,2,3,4
PLAIN 2	1091	PANDA	FM	U	-	68,725	1,3,4
PLAIN 3	1092	PARTY	FM	U	-	81,450	1,3,4
LINK UP SECURE	1093	LU CT	TD	NR	1000	44,500	1
LINK UP PLAIN	1094	LU PT	FM	U	-	44,500	1,2,3,4
CAG PLAIN	1095	CAG PT	FM	U	-	41,650	1,2,3,4
CAG SECURE		CAG CT	FM	NS	H2151	41,650	2,3,4
EMERGENCY Net							
SAR A/112 (PRI)		BLUEFOX	AM	U	-	270,300	3,4
SAR B/112 (ALT)		PINKDOG	AM	U	-	290,225	3,4
HF 3G NET INFO							
HF CMD	DAY	Active 0800-1800Z	3G	NS	3636	-	2
	NIGHT	Active 1800-0800Z	3G	NS	3636	-	2
SPECIAL NOTES							
-GPS ZULU TIME ON ALL COMMS							
-HF DAY: 08:00Z, NIGHT 18:00Z							
-TOD-SERVER: BASE							
-CODE LIST TO BE USED ON UNCLASSIFIED COMMS- SEE BACK OF DOCUMENT							
TACSAT CHANNELS							
ISAF JARN	JARN	CH 999 WB 173 deg, 49 elev U-316.175/D 249.450				H2151	
RC N DAMA	RC N DAMA	CH 128 DAMA 225 deg, 29 elev U-298.000/D-245.925				3636	
CRYPTO NAME/TYPE							
H2151 VINSON	TEK 1	TEK 1	TEK 1	WEEK	Thursday	0001Z	
3636 ANDVT	TEK 1	TEK 1		WEEK	Thursday	0001Z	
DAMA TSK-SATELITE	TSK 1			Mnd	1/Mnd	1200Z	
MRR 2000				Mnd	Last day	2358Z	
CRYPTO INFORMATION MRR/LFR							
Key id:	1000						
December Key:	114						
January key:	115						

Procedures for TIC

1. FACNAV

- 1.1 Plot own position in FACNAV
- 1.2 Push "TIC" button in FACNAV

2. HF

- 2.1 Push "CALL"-button, type "Best", Station, choose BASE in address
- 2.2 PTT

Procedures for HOT MIC on CMD net

1. Check own mic
2. Switch to CG LU PT

RADIO CODE

MRR/LFR IP	1	PFR	5		
150C	2				
MBITR	3				
117F	4				

IP ADDRESS LIST

CALLSIGN	UNIT/FUNCTION	HF Wireless IP	MRR-IP
9BASE	OPS	192.168.100.9	
KPA	KP A LED	192.168.100.10	
A1	TR 1 / KP A	192.168.100.11	
A2	TR 2 / KP A	192.168.100.12	
A3	TR 3 / KP A	192.168.100.13	
KPB	KP B LED	192.168.100.20	
B1	TR 1 / KP B	192.168.100.21	
B2	TR 2 / KP B	192.168.100.22	
B3	TR 3 / KP B	192.168.100.23	
KPC	KP C LED	192.168.100.30	
C1	TR 1 / KP C	192.168.100.31	
C2	TR 2 / KP C	192.168.100.32	
C3	TR 3 / KP C	192.168.100.33	

DAMA ADDRESS LIST

DAMA	NAME	TBA
GUARD	RC N	40001
BASE		16309
9FOB		16398
9FWD		16397
KPA		16310
KPB		16320
KPC		16330
KPD		16340

PFR KANAL OPPSETT

KANAL	NET	FRQ	COMSEC	TYPE
1	FELLES	352,900	AES1	FIX FSK NB
2	GJENG1	368,225	AES2	TDMA 111XXX
3	GJENG2	368,225	AES2	TDMA XXX222
4	LFNETT	368,225	AES2	TDMA 111222
5	REBROAD	368,225	AES2	TDMA 111222
6	LAG1	385,925	AES3	TDMA 11XXXX
7	LAG2	385,925	AES3	TDMA XX11XX
8	LAG3	385,925	AES3	TDMA XXXX11
9	LF	385,925	AES3	TDMA 112233
10	LAG1HD	387,675	AES4	TDMA 1XXDDD
11	LAG2HD	387,675	AES4	TDMA X1XDDD
12	LAG3HD	387,675	AES4	TDMA XX1DDD
13				
14				

CODEWORD of the MONTH

BLOCKHEADS

0 1 2 3 4 5 6 7 8 9

Subtraktorkoder

A	46173	12073
B	32951	53899
C	07720	67787
D	33760	74055
E	07139	96216
F	98997	07565
G	05407	31612
H	63085	48040
I	35314	77167
J	58813	98674
K	96861	32882
L	86180	59469

Autentiseringstavle

0	A	MFCOIHVQASLDWNPJKRYGUETZB
1	B	YUHQNBOSVKLCMXEWFRTGZPJDA
2	C	EZUDFVHJWRBTKLXMINOQAGCSPY
3	D	TMFCOIHVQASLDWNPJKRYGBUZ
4	E	HWTLFQCVMRJBODSUKVIGENZAP
5	F	MLNPSEBYOKAGUIZRXCTDQHVFW
6	G	NLXYADMIKBUOVZGERWSJFPHTCQ
7	H	TFQYRVGBIXUJSAKOLCDWEHPMN
8	I	LNPOKGUZFHBXQYTDRAIMESWV
9	J	NXYDIKBUOVZGEWSJFPHTCQMLAR
	K	MBKTLQOAZUWSDPNVEFXRYCJHI
	L	HGKIZUJFBPDYACQWSERLTVOMXN

Supplies	Groups	Milit unit	Own forces	Weapon equipment	Green Forces	Names and areas I	Names and areas II	Names and areas III
3G	12.7mm cartridges	3H	Squad	3E	3Q	3R	3U	3L
14G	5.56 cartridge	14H	Platoon	14E	14Q	14R	14U	14L
4G	7.62 cartridge	4H	Company	4E	4Q	4R	4U	4L
11G	7.62 cartridges for MG	11H	Battalion	11E	11Q	11R	11U	11L
28G	9 mm cartridge	28H	Brigade	28E	28Q	28R	28U	28L
24G	Battery CR123	24H	Task force	24E	24Q	24R	24U	24L
22G	A/A/AAA Battery (specify)	22H	GRF	22E	22Q	22R	22U	22L
2G	F34 (litter) (diesel)	2H	NSE	2E	2Q	2R	2U	2L
33G	Flashlightbatt / BA-3030 (stk)	Numbers	OMLT/1 KDK	33E	33Q	33R	33U	33L
16G	Food (DOS)	G	0	16E	16Q	16R	16U	16L
7G	Gasoline (litter)	V	1	7E	7Q	7R	7U	7L
25G	Lightsticks (ea)	A	3	25E	25Q	25R	25U	25L
3P	Lithiumbatt BA 5590 (ea)	P	4	3P	3Q	3R	3U	3L
14P	PAGE-batt BA-3090 (stk)	A	3	14E	14Q	14R	14U	14L
11P	Penlightbatt / BA-3059 (ea)	El optic equipment	11E	11Q	11R	11U	11L	11L
28P	Toilet paper	30	Laser range finder	28E	28Q	28R	28U	28L
12G	Water (litter) -	140	Night vision goggles	12E	12Q	12R	12U	12L
22P	-	40	Night vision binoculars	22E	22Q	22R	22U	22L
3V	- A	110	Binoculars	3V	3Q	3R	3U	3L
14V	- B	280	IR Illuminator (any type)	14E	14Q	14R	14U	14L
4V	- C	240	Weapon sights (any type)	4E	4Q	4R	4U	4L
11V	- D	Landmarks	11E	11Q	11R	11U	11L	11L
28V	- E	3D	Valley	28E	28Q	28R	28U	28L
22V	- F	140	Route	22E	22Q	22R	22U	22L
22V	- G	4D	Crossing	22E	22Q	22R	22U	22L
2V	- H	11D	Heights	2V	2Q	2R	2U	2L
33V	- I	28D	Village	33E	33Q	33R	33U	33L
16V	- J	24D	Antenna Mast	16E	16Q	16R	16U	16L
6V	- K	22D	Power Line	6E	6Q	6R	6U	6L
25V	- L	Nationality	25E	25Q	25R	25U	25L	25L
3T	Armoured MB (vehicle)	31	CANADA	3T	3Q	3R	3U	3L
14T	Ready to dispose UXO/IED	141	ROMANIA	14E	14Q	14R	14U	14L
11T	Back in camp	21	GERMANY	11E	11Q	11R	11U	11L
24T	Leaving Camp	221	SWEDEN	24E	24Q	24R	24U	24L
2T	Broken	21	DENMARK	2E	2Q	2R	2U	2L
17P	MB-290 (vehicle)	331	PORTUGAL	17E	17Q	17R	17U	17L
33T	MB-290 ML (vehicle)	161	UK	33E	33Q	33R	33U	33L
16T	Recovery vehicle	61	US	16E	16Q	16R	16U	16L
6T	Scania P113 (truck)	3Y	Camp Griffin	6E	6Q	6R	6U	6L
7T	SISU / PASI	14Y	Camp Nidaros	7E	7Q	7R	7U	7L
35T	Message on BKM	4Y	1st ANP Substation	35E	35Q	35R	35U	35L
1T	Return to camp	11Y	2nd ANP Substation	1E	1Q	1R	1U	1L
1J	Toyota LC 95	24Y	ANP OP	1J	1Q	1R	1U	1L
13T	Secured	24Y	Airfield	13E	13Q	13R	13U	13L
38T	Shooting	22Y	Airport	38E	38Q	38R	38U	38L
12T	We are heading to PRT CAMP	2Y	ANP HQ/CCP	12E	12Q	12R	12U	12L
10T	Technical Stop	33Y	ANP QRF CAMP	10E	10Q	10R	10U	10L
19T	Civil Accident	16Y	Camp	19E	19Q	19R	19U	19L
31T	Crowd Of People	6Y	Chinese Camp	31E	31Q	31R	31U	31L
39T	Stone Throwing	25Y	FOB Ghornmach	39E	39Q	39R	39U	39L
39T	Tent	35Y	FOB Oyasar	39E	39Q	39R	39U	39L
3N	Large antenna	1Y	FOB	3N	3Q	3R	3U	3L
14N	ROSHAN	2N	ROSHAN	14E	14Q	14R	14U	14L
4N	IRidium	33N	PC	4E	4Q	4R	4U	4L
11N	PRR	16N	MRR	11E	11Q	11R	11U	11L
28N	IFTS	6N	HF	28E	28Q	28R	28U	28L
7N	BMS	18Y	Ring route North	7E	7Q	7R	7U	7L
31Y	Border Police HQ	31Y	Border Police HQ	31E	31Q	31R	31U	31L

Vedlegg K

SOI fra programvaren

Dette vedlegget inneholder en SOI laget av programvaren. SOI-en har to eksemplarer for å demonstrere dynamisk nummerering i toppsegmentet. Samme SOI ligger også vedlagt i komprimert og ukomprimert form.

K.1 Eksportert SOI-PDF med to eksemplarer



Forhåndsavtalte koder				
Varslingsord: TOYOTA				
A Bokstaver	B Landemerker	E Tall		
A P	A Elv	A 0		
B V	B St	B 6		
C D	C Bro	C 7		
D M		D 2		
E X	C Reininger			
F L	A Vest	E 5		
G Q	B Sør	F 8		
H B	C Fram	G 9		
I K	D Tilbake	H 3		
J C	E Nord	I 1		
K E	F Øst	J 4		
L T				
M U	D Støtte			
N H	A Sanitet	F Våpenteknisk		
O G	B Helikopter	A Bombe		
P I	C MP	B Håndgranat		
Q A	D K9	C MP-5		
R Y		D HK-416		
S J				
T W				
U S				
V F				
W N				
X Z				
Y R				
Z O				

FREKVENSTABELL

MRR

Nett	Nett-ID	Type	Område
Hæren Brig-N INGBN	2000	CNR_TD	Norge
Hæren Brig-N INGBN	2001	CNR_TD	Norge
Hæren Brig-N INGBN	2002	CNR_TD	Norge
Hæren Brig-N INGBN	2003	CNR_TD	Norge
Hæren Brig-N INGBN KP Samvirke	2090	CNR_TD	Norge
Hæren Brig-N INGBN KP Samvirke	2091	CNR_TD	Norge
Hæren Brig-N INGBN	2092	CNR_TD	Norge
Hæren Brig-N INGBN	2093	CNR_TD	Norge
Hæren Brig-N INGBN	2095	CNR_TD	Norge

Prosedyre ved manglende VHF-forbindelse

1. Kontroller NET ID

2. Kontroller fysisk kabling (Strøm, antenne, mikrofon)

3. Kontroller tid/synk status og sendeefekt

4. Kontroller kryptonøkkel

5. Kontroller at du er innenfor dekningsbøle fra rele

6. Kontroller endringer i området (store antenner, aggregat, jammere)

7. Forflytt deg til siste posisjon hvor du hadde samband

SUBTRAKTORKODER

A	9446	7304
B	3039	5314
C	6062	0151
D	8795	0395
E	7502	7404
F	8926	7340
G	1833	1811

KODEFRASEL

Kode	Frase
Husky	Start radiotau
Terrier	Sløpp radiotau
Puddel	Kompromitter

- Prosedyre ved manglende VHF-forbindelse**
1. Kontroller NET ID
 2. Kontroller fysisk kabling (Strøm, antenne, mikrofon)
 3. Kontroller tid/synk status og sendeeffekt
 4. Kontroller kryptonøkkel
 5. Kontroller at du er innenfor dekningsboble fra rele
 6. Kontroller endringer i området (store antenner, aggregat, jannere)
 7. Fortlytt deg til siste posisjon hvor du hadde samband

SUBTRAKTORKODER	
A	9446 7304
B	3039 5314
C	6062 0151
D	8795 0395
E	7502 7404
F	8926 7340
G	1833 1811

KODEFRASER	
Kode	Fraser
Husky	Start radiotauhet
Terrier	Stopp radiotauhet
Puddel	Kompromittert radio
Labrador	Kompromittert pc

TELEFONLISTE			
Funksjon	Telefon	FDN	Iridium
INGBN OPS Devdis	77 89 21 16	0580 2116	
INGBN OPS felt	488 84 500		
BRIGKO	400 29 999	(0)580 8500	
PingKp-1	488 88 511		+88 165 1437 527
IngKp-2	488 88 521		+88 165 1437 528
PingKp-3	488 88 531		
IngKp-4	488 88 541		+88 165 1437 525
IngKp-5	488 88 551		
IngKp-6	488 88 561		+88 165 1437 524
Skredgruppe			+88 165 1437 311

AUTENTISERINGSTAVLE (21 & 25)	
A 0	CVHMQ QQDIE AREUG UTQPA HLDJW
B 1	KBYAA DRTWB HVPBK FRUDT SYDKP
C 2	PFQRX XMXNN HMZNC DYRDX SBTCD
D 3	VOYRH SKODX NEVKI BDFZE OXYGA
E 4	IRLMT DTQPR MGNBP SHYVD DXTQO
F 5	OLRFT INWVB DAQHG CHBZG ITZNZ
G 6	XJRVU QZTTU SOWUL BLSQB PNKLW
H 7	UDJQE QGVRF OBMSH JHCXA GVZFS
I 8	VVAVU PYEHT FZNPC MYRRR DYMWX
J 9	UXRZQ MCZOV FVYVF VMOJY REHGO

A	B	C
4	6	tekst
8	12	mer tekst
16	24	enda mer tekst

Test SOI

Test SOI med alle moduler
Opprettet: 15.05.2020



Eksempelr

1 av 2
Vedlegg A

UGRADERT

Gyldig fra: 15.05.2020
Gyldig til: 16.05.2020

Modul i vedlegg

Denne modulen ligger i
et vedlegg til SOI-en



Forhåndsavtalte koder				
Varslingsord: TOYOTA				
A Bokstaver	B Landemerker	E Tall		
A P	A Elv	A 0		
B V	B Sti	B 6		
C D	C Bro	C 7		
D M		D 2		
E X	C Reininger	D 2		
F L	A Vest	E 5		
G Q	B Sør	F 8		
H B	C Fram	G 9		
I K	D Tilbake	H 3		
J C	E Nord	I 1		
K E	F Øst	J 4		
L T				
M U	D Støtte			
N H	A Sanitet	F Våpenteknisk		
O G	B Helikopter	A Bombe		
P I	C MP	B Håndgranat		
Q A	D K9	C MP-5		
R Y		D HK-416		
S J				
T W				
U S				
V F				
W N				
X Z				
Y R				
Z O				

FREKVENSTABELL

MRR

Nett	Nett-ID	Type	Område
Hæren Brig-N INGBN	2000	CNR_TD	Norge
Hæren Brig-N INGBN	2001	CNR_TD	Norge
Hæren Brig-N INGBN	2002	CNR_TD	Norge
Hæren Brig-N INGBN	2003	CNR_TD	Norge
Hæren Brig-N INGBN KP Samvirke	2090	CNR_TD	Norge
Hæren Brig-N INGBN KP Samvirke	2091	CNR_TD	Norge
Hæren Brig-N INGBN	2092	CNR_TD	Norge
Hæren Brig-N INGBN	2093	CNR_TD	Norge
Hæren Brig-N INGBN	2095	CNR_TD	Norge

Prosedyre ved manglende VHF-forbindelse

1. Kontroller NET ID

2. Kontroller fysisk kabling (Strøm, antenne, mikrofon)

3. Kontroller tid/synk status og sendeeffekt

4. Kontroller kryptonøkkel

5. Kontroller at du er innenfor dekningsbøle fra rele

6. Kontroller endringer i området (store antenner, aggregat, jammere)

7. Forflytt deg til siste posisjon hvor du hadde samband

SUBTRAKTORCODER

A	9446	7304
B	3039	5314
C	6062	0151
D	8795	0395
E	7502	7404
F	8926	7340
G	1833	1811

KODEFRASEL

Kode	Frase
Husky	Start radiotau
Terrier	Sløpp radiotau
Puddel	Kompromitter

- Prosedyre ved manglende VHF-forbindelse**
1. Kontroller NET ID
 2. Kontroller fysisk kabling (Strøm, antenne, mikrofon)
 3. Kontroller tid/synk status og sendeeffekt
 4. Kontroller kryptonøkkel
 5. Kontroller at du er innenfor dekningsboble fra rele
 6. Kontroller endringer i området (store antenner, aggregat, jannmere)
 7. Fortlytt deg til siste posisjon hvor du hadde samband

SUBTRAKTORKODER	
A	9446 7304
B	3039 5314
C	6062 0151
D	8795 0395
E	7502 7404
F	8926 7340
G	1833 1811

TELEFONLISTE			
Funksjon	Telefon	FDN	Iridium
INGBN OPS Devdis	77 89 21 16	0580 2116	
INGBN OPS felt	488 84 500		
BRIGKO	400 29 999	(0)580 8500	
PingKp-1	488 88 511		+88 165 1437 527
IngKp-2	488 88 521		+88 165 1437 528
PingKp-3	488 88 531		
IngKp-4	488 88 541		+88 165 1437 525
IngKp-5	488 88 551		
IngKp-6	488 88 561		+88 165 1437 524
Skredgruppe			+88 165 1437 311

AUTENTISERINGSTAVLE (21 & 25)			
A	0	CVHMQ	QQDIE AREUG UTQPA HLDJW
B	1	KBYAA	DRTWB HVPBK FRUDT SYDKP
C	2	PFQRX	XMXNN HMZNC DYRDX SBTCD
D	3	VOYRH	SKODX NEVKI BDFZE OXYGA
E	4	IRLMT	DTQPR MGNBP SHYVD DXTQO
F	5	OLRFT	INWVB DAQHG CHBZG ITZNZ
G	6	XJRVU	QZTTU SOWUL BLSQB PNKLW
H	7	UDJQE	QGVRF OBMSH JHCXA GVZFS
I	8	VVAVU	PVEHT FZNPC MYRRR DYMWX
J	9	UXRZQ	MCZOV FVYVF VMOJY REHGO

KODEFRASER		
Kode	Fraser	
Husky	Start radiotaushet	
Terrier	Stopp radiotaushet	
Puddel	Kompromittert radio	
Labrador	Kompromittert pc	

A	B	C
4	6	tekst
8	12	mer tekst
16	24	enda mer tekst

Test SOI

Test SOI med alle moduler
Opprettet: 15.05.2020



Eksempel
2 av 2
Vedlegg A

UGRADERT

Gyldig fra: 15.05.2020
Gyldig til: 16.05.2020

Modul i vedlegg

Denne modulen ligger i
et vedlegg til SOI-en

```
{
  "title": "Test S0I", "description": "Test S0I med alle moduler", "version": "1",
  "date": "2020-05-15", "valid": {
    "from_date": "2020-05-15", "to_date": "2020-05-16"
  }, "icon": "soitool/media/HVlogo.png", "classification": "UGRADERT",
  "orientation": "landscape", "placement_strategy": "auto", "algorithm_bin": "BFF",
  "algorithm_pack": "MaxRectsBL", "algorithm_sort": "area", "modules": [
    {
      "type": "PredefinedCodesModule", "data": {
        "headline": "Forh\u00e5ndsavtalte koder",
        "warning_word": "TOYOTA", "maximum_column_height": 920,
        "categories": [
          "Bokstaver", "Landemerker", "Retninger", "St\u00f8ttet", "Tall", "V\u00e5penteknisk"
        ], "tables": [
          {
            "table_headline": "A Bokstaver", "expressions": [
              "P", "V", "D", "M", "X", "L", "Q", "B", "K", "C", "E", "T", "U", "H", "G", "I", "A", "Y", "J", "W", "S", "F", "N", "Z", "R", "O"
            ]
          }, {
            "table_headline": "B Landemerker", "expressions": [
              "Elv", "Sti", "Bro"
            ]
          }, {
            "table_headline": "C Retninger", "expressions": [
              "Vest", "S\u00f8rt", "Fram", "Tilbake", "Nord", "S\u00d8st"
            ]
          }, {
            "table_headline": "D St\u00f8ttet", "expressions": [
              "Sanitet", "Helikopter", "MP", "K9"
            ]
          }, {
            "table_headline": "E Tall", "expressions": [
              "0", "6", "7", "2", "5", "8", "9", "3", "1", "4"
            ]
          }, {
            "table_headline": "F V\u00e5penteknisk", "expressions": [
              "Bombe", "H\u00e5ndgranat", "MP-5", "HK-416"
            ]
          }
        ], "meta": {
          "x": 0, "y": 0, "page": 1, "name": "Forh\u00e5ndsavtalte koder 8"
        }
      }, {
        "type": "FrequencyTableModule", "data": {
          "": {
            "Nett": [
              "H\u00e6ren Brig-N INGBN", "H\u00e6ren Brig-N INGBN", "H\u00e6ren Brig-N INGBN", "H\u00e6ren Brig-N INGBN", "H\u00e6ren Brig-N INGBN KP Samvirke", "H\u00e6ren Brig-N INGBN KP Samvirke", "H\u00e6ren Brig-N INGBN", "H\u00e6ren Brig-N INGBN", "H\u00e6ren Brig-N INGBN"
            ], "MRR": [
              "2000", "2001", "2002", "", "2090", "2091", "2092", "2093", "2095"
            ], "Type": [
              "CNR_TD", "CNR_TD", "CNR_TD", "CNR_TD", "", "CNR_TD", "CNR_TD", "CNR_TD", "CNR_TD", "CNR_TD", "CNR_TD"
            ], "Omru\u00e5de": [
              "Norge", "Norge", "Norge", "Norge", "", "Norge", "Norge", "Norge", "Norge", "Norge"
            ]
          }, "meta": {
            "x": 736, "y": 0, "page": 1, "name": "Frekvenstabell 5"
          }
        }, {
          "type": "PhonebookModule", "data": {
            "Funksjon": [
              "INGBN OPS Devdis", "INGBN OPS felt", "BRIGKO", "PingKp-1", "IngKp-2", "PingKp-3", "IngKp-4", "IngKp-5", "IngKp-6", "Skredgruppe"
            ], "Telefon": [
              "77 89 21 16", "488 84 500", "400 29 999", "488 88 511", "488 88 521", "488 88 531", "488 88 541", "488 88 551", "488 88 561", "", "FDN": [
                  "0580 2116", "", "(0)580 8500", "", "", "", "", "", "", ""
                ], "Iridium": [
                  "", "", "", "+88 165 1437 527", "+88 165 1437 528", "", "+88 165 1437 525", "", "+88 165 1437 524", "+88 165 1437 311"
                ]
            ], "meta": {
              "x": 1395, "y": 0, "page": 1, "name": "Telefonliste 6"
            }
          }, {
            "type": "AuthenticationBoardModule", "data": {
              "cells": [
                "AUTENTISERINGSTAVLE (21 & 25)", ["A", "0"], ["CVHWQ QODIE AREUG UTQPA HLDJW"], ["B", "1"], ["KBAYAA DRTWB HVPKM FRUDT SYDKP"], ["C", "2"], ["PFQRX XWXNN HMZNC DYRDY SBTC"], ["D", "3"], ["VOYRH SKODX NEVKI BDFZE OXYGA"], ["E", "4"], ["IRLMT DTQPR MGNBP SHYVD DXTQQ"], ["F", "5"], ["OLRFT INVVB DAQH GHBZG ITZN"], ["G", "6"], ["XJRUV QZTTU SOMUL BLSQB PNKLW"], ["H", "7"], ["UDJQE QGVRF OBWSH JHCXA GVZFS"], ["I", "8"], ["VVAYU PYEHT FZNP MYRR DYMWX"], ["J", "9"], ["UXRZQ MCZO VYYF VWJOY REHG"]
              ], "code_length": 25, "space_interval": 5, "space_amount": 2, "code_character_type": "ascii", "meta": {
                "x": 1395, "y": 409, "page": 1, "name": "Autentiseringstavle 2"
              }, {"type": "FreeTextModule", "data": [
                  "Prosedyre ved manglende VHF-forbindelse", "1. Kontroller NET ID\n2. Kontroller fysisk kabling (Str\u00f8m, antenne, mikrofon)\n3. Kontroller tid/synk status og sendeeffekt\n4. Kontroller krypton\u00f8kkel\n5. Kontroller at du er innenfor dekningsboble fra rel\u00e9\n6. Kontroller endringer i omr\u00e5det (store antenner, aggregat, jammere)\n7. Forflytt deg til siste posisjon hvor du hadde samband"
                ], "meta": {
                  "x": 736, "y": 448, "page": 1, "name": "Fritekst 4"
                }, {"type": "SubtractorcodesModule", "data": {
                    "cells": [
                      "SUBTRAKTORKODER", ["A", "", "9446 7304"], ["B", "", "3039 5314"], ["C", "", "6062 0151"], ["D", "", "8795 0395"], ["E", "", "7502 7404"], ["F", "", "8926 7340"], ["G", "", "1833 1811"]
                    ], "code_length": 8, "space_interval": 4, "space_amount": 3, "code_character_type": "digits", "meta": {
                      "x": 736, "y": 699, "page": 1, "name": "Subtraktorkoder 3"
                    }
                  }
                ]
              }
            ]
          }
        }
      ]
    }
  ]
}
```

```

"CodePhraseModule", "data": [{"KODEFRASER", "Hunderase", ["Dalmantiner",
"Sch\u00e6fer", "Chihuahua", "Collie"], [{"Husky", "Start radiotaushet"},
["Terrier", "Stopp radiotaushet"], ["Puddel", "Kompromittert radio"], ["Labrador",
"Kompromittert pc"]]}, "meta": {"x": 1119, "y": 751, "page": 1, "name": "Kodefraser
7"}}, {"type": "TableModule", "data": [{"A", "B", "C"}, {"4", "6", "tekst"}, {"8",
"12", "mer tekst"}, {"16", "24", "enda mer tekst"}], "meta": {"x": 1455, "y": 751,
"page": 1, "name": "Generisk tabell 1"}}, {"attachments": [{"type":
"FreeTextModule", "data": ["Modul i vedlegg", "Denne modulen ligger i\ net vedlegg
til SOI-en"], "meta": {"x": 0, "y": 0, "page": 2, "name": "Vedleggsmodul"}]}]}

```

K.3 SOI i komprimert form

Kodeliste K.2: Komprimert SOI

```

b'\xfd7zXZ\x00\x00\x04\xe6\xd6\xb4F\x02\x00!\x01\x16\x00\x00\x00t/\xe5\xa3\xe0\x16\
xfc\x07\xdf]\x00\x11\x1e\xc7\x82#\xd1\xd6E0\xa8\x91=6\x13(\xb5+\xb3\xd2\xd5\x00\xf1\
\xe0hC5\xb9\xbb\xf8\xa0\x8fr;\xf5\xa6\x01\x9d\xa4#\xe0\xa62\x9c\x97 (\xb4\x91ik\x98\
Z\x8f<\xd4mx\x80\xa2\x1b!\xe9a\x9e\xbb\xdb\x7u\xc3\xc9\xabt9f\xca\x89\xfc\x94\pT\
xef\xbb\xe2\x06{\xd8V\xd2{\n\xf5\xb1\xe8kj\x08\xccz|WV0JG\xc1(\x98\x99/\x9e\xc2\xf9\
v\xd8W\xd3\xfd3\xa0$a6\xc6\x9c\x84L\xc1\xec\x85\xdd.6\x1e\x18B\xa2>\x06\x12=6\x89\
\xf6R\x00\xd9(\xb6@\xfb\xbe\xdc7\xecZ\xce\x9d\xf8\xa0!\xe6\x8fFaBL\x91\x05\x0b\xe9\
x9ad@\x07\x94)\xe3\x0e)#\xf1p\xffc\xff=\x048*\xbcx4\xae3\xa7\xe4\xe6p\xb2\x1f\xaa\
xc8\xa8\xeb[-P#a0\x83\x92\x06R\xa1\xc7\x82\xabj\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\
x6\xedI\xfd\xfd\xfd\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\xfb\
x84\x07\x7f\xfd6\xfd6J\xfd4h\xbb\xfd5X\xce\xbb\xfb70&\xA\x088\x19'\x11\xeb1j1\xfd5\
e0\x80\xfb9B\xd7\xfb3\x00"\x00\xc6\xef\x07\xbeRV\x01\x80\x0eB0\x98\xfb9\x1f\xa9\xfb4\
xe5\x1b\xfb2&\xc2\x90\xfb2\xafY\xe9F\x00\xae\x1c\x05\xac\xe7H\x05\xac\xfd7\x1c\
xc4\xea\x92\x01\xcc2}\xe4\x83\xfd7K\x80\x07\x04\x9d\x8b\xed\xc7\x89\xfd8s\xc8\x11\
x9dm\xfb\xdb3\xfbH\x03\xfc\xcf\xfd8\x89\x18\xef\x9b\x0E\x90\xfb\x0f\xfb\xfd\xfd\xfd\
xc2I\x02h\x13\xfd0\x99\xeeN\xdeqV7\x82z)#p\leep\xe3\xe6\x1a\x00\xa1\x0f\x8b\x0c\x04\
xdb\x88G\xfb1\x10<\x8d\x94N\x8a<\xfcn\xdf\xcd\x04 \xca\x0c\xf7\xde\x8c\xfb9\xfb9\
x0e\x13\xcd\xbb1\x0f\xca\xfb6&\x3\xfd7\xea\x90\xfb1\xfb8c\x97\x01\xbb1C t\x9c\
x8ax\xdaI\x9b\x8d\x85\x08\x8aQ\xb9W-=VMK\xe1\x06\xfb1\xfb0\xa7C\xbaJ\xa5\
x7f~\x05W\xfd5R\xe0l\xa1\xed\x00\x02\xbbj\x99\xa6\xfb8'\x07\xe2-\x06\
x87\x1e]_Xp\x1b\x00\x9d1\xcb&\x8d*\xcaW\x0e$3\x02oh\xef\xfb1\x1d\x0b\
x0b\x1b\xac=XKm!\x9d\x08#g:}\x81\xe5JVV\x7fP\xfb8\x11\xfb1\x9f1\x06\
xd3(\xc1\x8c\xbc\x03:\x15\x04\x02\x87\xbb5\xda10AYvC\x0b\x0c2u=W"\x9d\
xe1\x9e\xef\xfd1\xdb\x1a2\x11\xfb\x029\x8f\x04)(\xf7\x9c\xbc\xdb\
x81aT\x8d\x01\x1e\x02\xa5=6\xa5\xb7\xab5\x92\xa2\xcd{\xfbb\xba\
xf6\xfb9e\xbb9\xeb7\xeb\x16'\x8f\x1e\x05U:\x0d\x96n*\x14\x7fYAt7\
x0eS5\x08\xdab\x01'\x8d\x01\x1b\x02\x8f\xed\x1b\xfb\x8e\xfb?\xc2\
xa1v\x0f2\xe8\xbb7\x8d\xdfU\x00/\xed\xed\x02\x08\x0c1cs\
xea\xfb7\x96\xddo\x0c4\x99\x9c\x7fe\x8f\xa5\x17\x81?\
xe1l\xae6\x01\xbe@\x8f\xfb0\x08\xfb\x05\xab\x9bWk\xddj\
\xda\x18-\xcbb\x064"x\xfb2\x92\xafI\xef\xa2t\xcb\x92b\
xe9\xed\xa2\xa9L\x14\x83\x01\x87\x02"\xff\xa2\x03a\
xe6\xbb8\x04\xa2\x03E\xa2\x0c\x07\xe7S[\x8bkx\x03\
x7f\x02\xde\x00\xfe\xebJ\xfb{\#V\xb7\x02\x0dah\
x8c,\x106\xae\x89hJR\xfb9\xbb4\x880\xe9\x9cd\
xa6\xcf/F\xbb[\x92\xa6Y\x14\x1cT~.2\x9f\
xf4\x88\xfe1C\x81f\x81nY\xe5\x92\xfb;\x06\
x9bY\xa2\x00\xdfj\x06]\x84\x03\x0c3,VtL'\
x95>\x7fI\x0c9>\x0c\x08bs\xe7j\x01G\x04\
xb2d2\x19'3\xda\xfb1\x98\x85P\x00k\
xe4\xe3\xfc\x91V\x0e\xcf\xff\xfb\x02R.\xc6\
x90V\x0c1WG\xab&q\xa9>\x12\x05:I\xba\
x13\x87)\xc8\x0c:\xf8\x0c7\x06\xe4\x1e+\
xdf\x07\x1a\x98.\x84\x0cfe;\x1d\x89\
xb5\xa6\xa9X\xa97\x03x\xde\xa2t\
xe3S\xbb\x04\xae\x85\xdcg\xaa\
xaa\x9ay8\x07\x16$|\x14\xdb"\
xa\xc6s\xcd\xad\xe7\x06c"\x9b\
xa2\x88;\xef-X\x06\xa2,\
x08\x8c\xfb8\x1a\xef\x96\x02\
0\xb3I\xe9\x13qu\x06G\x02\
xc6\x1bQ\x94\xbc\xfb4\x08y\
x1b)\x15s(@\xacs\
xf6\x142\x00"\x01\
x96L\x04-\x03\xafkl\xdb\
xe3t\x1a\xfeP\x17\x9a\
xca\xe6\xfb9\x96[\x04\
x10d\xca\xfe\xa9\
_x87CZs\x9e3!\x1d\
So0J\x84\x01\
\x83\x86\x02r\x0c\
xa5J\raL\x05\
x0c\x0b"WF\xec?!\$x\
a2d\xcc9\x03\
xb3\x09Dk\
xf6\x1b\x00;\xa0Z_}\
x1b\x80gp\
x07\xba\
x18\x0c0?<\x8a\
xea\x95\x8328,\
xeb\
xf4\xec\x96\
x0c\xe4\
xc4q]\x96qp\
x03\x14\
xabp\
xe0\
x9b\
xd6T\
xfeK(w\
xc3\
xa8)\
x\
fd\xfb2\
x0e\
x00\
x07\
xf7Sn\
x86\
x94;\x02U\
x88\
xe7n\
n\
rr\x01T\
v\
xc\
fc\
xc5\
x01\
|\
xc9\
xdb\
x00\
\xbb9\
xeb\
xb0\
xad\
xa3\
xc\
fd\
x06~\
x86$@T]]\
xf2^"\
xe6e#\
xb3\
x9b@'\
x92T2CC\
xc74\
xf5\
xab\

```

```

\# \xe1\xc3Aa\x88\xf1\xa8e\xceW\xc2\xd2\xc9U\xa6\x1a\xa1\xbb~\x98\x85\x84\x83\xdac\xe9\x
b7k\xe8=\x08\xb0\xda.\xcc\xe6\x18L\xd0\xee\x8b\xb0!\xd9'p\x90p\x1d\xd3\x91B\x81
\aaabr\x1ec}_\x9d\xe8*7.\xa0\x92\x88\xbd\x023\x97c5\x1b\xe5\xad\x90\xb11\x9fc\x85@
\xb9\x90X\xfb\x8e\x12\x1a\x8fX6\xd3\xb73o\x08\xe9\xb0\x85\xc4\xec\x83\xf0\x86\xae\x8e
\x8e\x8e\x01\xf2\x1e\xef\xb0\xea\xa2\x89\x15a\xa5\xc4+|\xaaGq\x91\x15\xd6\xb0\xd7R\x
c4L\xe6g\xd0\x13\xe8&ii\da\xe5\xdb\xbc\xab\xff\xab\x8d\x91\x03$-T\x87f6\xec\x0
e\x15\xf7\xb9F\x04\x9e$I\x86\\ \x1e\x89\xb5\xb6\xbfC\xb6\xd8G\x894\xcaAF\xcc\x93\x9
c\x82\x07\xbc;u>v)\x9e\x06\xc94u\x19\xbdb'\x1cF\xff\xcaq\xea\x9c!7\xb8u\xa4\xed\x9
3\xd9\xed|\xa0\xb9\x8e\t\xcd\xbf\x9cy\x04\xe7\xa3\x07\xb2\xa6\xb2\xf8\x1cx\x10t\xfe
\r\x83>\x85\x8f\x92\xd8o\xaeCG\xcam\xac\xefn$B\x0cd%\x11#|:\x84\xfa+\xd0\xb3\xc4@
\x93\x1b\xe7\xbc\xa6\xa1\xe9\x04\xca\x12\x05Hc\xfd\xceDbj\xab\x853C\\\x16% \x85\x96
\xa7sZ\x12\x1e\x81\x0ba\xbc\x9a\xce\r\r\xa7\x87\x0by\x84t\xb5\xd4\x8c-I\x8f\x8c\xcb
\x9a|0u\x17g\x03\x99:M*H\xcfSj\x12\xcc\xb5\xf4\x08\x00\xf1\xfe\x17\xc1\x8d"3\x86\x
da\xfb\xde\x8a\x8b8\xa00M\xd0)lQu\x8f8p\x99\x959\x8P\x00\xb5\x87\ae0!2=5\x9c\x9a\x9
b\x03\x99\x89\xe8'\xa8A\x17\xaa\xd9[\xfdx\xb0+\xfa3\xa2t\x00\x9d\xce\rh\x0fPfg\x8f
\xeb\xfa#9:\x97\xad\xcb\x8b\x8c6P\xd0\xb80\x98\x00T\x00c=CT~\x18\xd6\xecGJ\xce\x
97\x82\xe3\xe5\\\xd3\xca?\x0c\t\x00\xdbu\xa4\xb1\xf6\xa6+Y*\xfe\xb2\xa2e(\xfdx\xaa\x
9e\xdd\x14\xb4\x13\x96\x16N\xde\xfb\x0f \xaa;\xc2o!\x8e\x87\xb1\x8e\x17\xd2\x85\x90
\x13~\xa2J\x81\x06\xda\xda9\x1aV\x98z\xc5\xfa\xd4\x11\xe9\xfa\xe7\x00\xcb\xe9\xfc
\xdeM\xe9\x8e\xc2\xb1\xa3\xce\xb8'\x00\xf2t\xa4\x98\x07C\xf3\xaa\xa5\xd8\x8d0\x01.
\xabs\xf7b\x8f\x89tR\xcf\xcc\x8b8|N{!\x1f5:\x83\xbb\xe9\x9c\x19e/\xba\x90u\x86\xef
@_r\x19\xd0A\x15t\xe9\xdb\xb4\x95\xdeu\x9d\x19_\xf3kA}\xfc\x1a>ra\xe8P\xf8_\xcd\xa
61\xcaWTn\x85Y\x9eeY\x00(T\xd8\x0f\x95\x7f\x0e7i\x1f\xab\xe2\x96\xdd\xd8,Dzh\xce\x
f2\xdb\x0c>\x80Qn$\xce\xe8\x80\x07\xa5\x80f}\xbfxdf\x8a\xa8\x04\xc5\x06170\x01\x16
\xca\xae*\xb0\x0cnKe0\xacC\x8f\x8f\xc8{ "3xv\x0f\x13~\xd5!\xecbz\x9d?\x8f\x8fB\xe5H2
\xbbH\xf7\xe4\xa0d1b\xd6\xa4\x8a[, \xf8\xba\xb0\xbc\x9f\xeeftK\xac\xa8^\x943\xc7aD
%\x03ka\xd4\xb0y\x198\xbe\xa2\xcfT\xaf\xf7\xc6\xdf\xaez\xff\xaf\xc7\xb0\niV\xd1\x94
A\xbb\x17\x0b\x81\xdc\xdb\xffb\xc7\x82\to\x9f\x95\x0e\x11Y\xe0\x14\x84\xc6\xb5\x02
xcb{\x05\xe4\x00\x00\x00|\xa9\xf5\xf1&\x06\x8c\x85\x00\x01\xfb\x0f\xfd-\x00\x00\x1f
\x81\x8c9z\xb1xc4g\xfb\x02\x00\x00\x00\x00\x00\x04YZ'

```