



Norwegian University of Science and Technology

Remote NDT – Secure Data

Forfatter(e):

Kristoffer Klingenberg

Rolf Arne Myraunet

Veileder(e):

Ivar Farup

Oppdragsgiver:

DolphiTech AS

Bachelor of Science in Engineering - Computer Science,
Department of Computer Science and Media Technology,
Norwegian University of Science and Technology

16.05.2017

Sammendrag av Bacheloroppgaven

Tittel:	Remote NDT - Secure Data
Dato:	16.05.2017
Deltakere:	Kristoffer Klingenberg Rolf Arne Myraunet
Veileder:	Ivar Farup
Oppdragsgiver:	DolphiTech AS
Kontaktperson:	Yngve Raudberget
Nøkkelord:	PHP, JavaScript, C++, WebRTC, NDT, IMT
Antall sider:	139
Antall vedlegg:	11
Tilgjengelighet:	Åpen

Sammendrag:	Denne oppgaven omhandler utvikling av et system som muliggjør utførelse av fjernstyrt NDT. Dette innebærer å la en eller flere NDT-eksperter inspisere et prøveobjekt vha. ultralydteknologi uten å være fysisk tilstede. Systemet kan brukes for å effektivisere utførelse av NDT-tester ved at en operatør utfører den fysiske inspeksjonen, der sertifiserte eksperter kan kobles opp via Internett for å se på data fra inspeksjonen og konkludere tilstanden til prøveobjektet.
-------------	--

Summary of Graduate Project

Title: Remote NDT - Secure Data
Date: 16.05.2017
Authors: Kristoffer Klingenberg
Rolf Arne Myraunet
Supervisor: Ivar Farup
Employer: DolphiTech AS
Contact person: Yngve Raudberget
Keywords: PHP, JavaScript, C++, WebRTC, NDT, IMT
Pages: [139](#)
Attachments: 11
Availability: Open

Abstract: This thesis deals with the development of a system that enables execution of remote NDT. This involves allowing one or more NDT experts to inspect the properties of a material using ultrasound technology without being physically present. The system can be used to more efficiently perform NDT testing by allowing a non-expert operator to execute physical inspection, while a certified expert can be connected over the Internet to view data from the inspection and conclude the state of the material being tested.

FORORD

Vi vil innlede rapporten med å takke DolphiTech for muligheten til å jobbe på dette prosjektet. I tillegg vil vi takke Yngve Raudberget som har vært vår kontaktperson i DolphiTech og alle andre i DolphiTech som har bidratt med tilbakemeldinger og gitt oss forståelse for oppgaven.

Vil vil også takke vår veileder Ivar Farup som har vært svært imøtekommende når vi har hatt behov for innspill og tilbakemeldinger på rapporten. Selv om vi ikke hadde noe spesielt å diskutere var det fortsatt trivelig å komme innom for å vise fremdriften i prosjektet.

Innhold

FORORD	iii
1 INNLEDNING	1
1.1 Bakgrunn	1
1.2 Anvendelse	2
1.3 Oppgavebeskrivelse	3
1.4 Prosjekt mål	5
1.4.1 Resultatmål	6
1.4.2 Effektmål	6
1.4.3 Læringsutbytte	6
1.5 Avgrensning	7
1.6 Rapportstruktur	7
2 ARBEIDSPROSESS	8
2.1 Valg av utviklingsmodell	8
2.2 Prosjektroller	8
2.3 Sprinter	9
2.4 Oppsummering sprintresultat	12
2.5 Dokumentasjon	16
2.6 Statusmøter	17
2.7 Workshops	17
2.8 Utviklingsverktøy	18
2.9 Rapportskriving	19
3 SPRINT 1	21
3.1 Innledning	21
3.2 User Stories	21
3.3 Kravspesifikasjon	21
3.3.1 Innlogging	21
3.3.2 Streaming av ultralyddata	22
3.4 Implementasjon	23
3.4.1 Server	24
3.4.2 Blackbox	25
3.4.3 Klienter	26
3.4.4 Innlogging	26
3.4.5 Streaming av ultralyddata	27
3.5 Beslutninger	33
3.6 Konklusjon	35
4 SPRINT 2	37
4.1 Innledning	37

4.2	User Stories	37
4.3	Kravspesifikasjon	37
4.3.1	Endre innstillinger i DolphiCam	37
4.3.2	Invitere deltaker	38
4.3.3	Endre fargepalett	38
4.3.4	Tilganger	39
4.4	Implementasjon	39
4.4.1	Endring av innstilling	39
4.4.2	Invitere deltagere	42
4.4.3	Fargepalett	42
4.4.4	Tilganger	43
4.5	Konklusjon	44
5	SPRINT 3	46
5.1	Innledning	46
5.2	User Stories	46
5.3	Kravspesifikasjon	46
5.3.1	Direktemeldinger	46
5.3.2	Kommunikasjon over video og lyd	46
5.3.3	Forbedret ytelse i portalen	47
5.4	Implementasjon	47
5.4.1	Direktemeldinger	47
5.4.2	Sikker tilkobling	48
5.4.3	Kommunikasjon over video og lyd	48
5.4.4	Ytelse klient	51
5.5	Konklusjon	52
6	SPRINT 4	54
6.1	Innledning	54
6.2	User Stories	54
6.3	Kravspesifikasjon	54
6.3.1	Tilganger	54
6.3.2	GUI	55
6.3.3	Tidslinje	55
6.4	Implementasjon	55
6.4.1	Tilganger	55
6.4.2	Fleksibelt layout	57
6.4.3	Tidslinje	58
6.5	Konklusjon	60
7	SPRINT 5	61
7.1	Innledning	61

7.2	User Stories	61
7.3	Kravspesifikasjon	62
7.3.1	Tidslinje	62
7.3.2	Målestokk	62
7.3.3	Fjernstyrt start/stopp av skanning	62
7.3.4	Fargepalett i tidslinjevisning	62
7.3.5	Varsler	63
7.4	Implementasjon	63
7.4.1	Tidslinje	63
7.4.2	Fargepalett i tidslinjevisning	64
7.4.3	Målestokk	64
7.4.4	Fjernstyrt start/stopp av skanning	65
7.4.5	Varsler	65
7.5	Konklusjon	66
8	SPRINT 6	68
8.1	Innledning	68
8.2	User Stories	68
8.3	Kravspesifikasjon	68
8.3.1	Innstillinger i DolphiCam	68
8.3.2	B-scan vertikal	69
8.3.3	Lagring av innstillinger	69
8.3.4	Snapshot	69
8.4	Implementasjon	69
8.4.1	B-scan vertikal	70
8.4.2	Lagring av innstillinger	71
8.4.3	Snapshot	71
8.5	Konklusjon	71
9	SPRINT 7	73
9.1	Innledning	73
9.2	User Stories	73
9.3	Kravspesifikasjon	73
9.3.1	Innstillinger i DolphiCam	73
9.3.2	Hilbert/Envelope funksjon	73
9.4	Implementasjon	74
9.4.1	Innstillinger i DolphiCam	74
9.4.2	Hilbert/envelope funksjon	74
9.5	Konklusjon	75
10	DISKUSJON	77
10.1	Sprint oppsummering	77

10.2 Resultat	77
10.3 Hva kunne blitt gjort annerledes?	78
10.4 Videre utvikling	78
10.5 Gruppeevaluering	79
11 KONKLUSJON	80
BIBLIOGRAFI	81
VEDLEGG	82
A Terminologi	83
B Prosjektavtale	85
C Grupperegler	88
D Oppgave	90
E Prosjektplan	93
F Statusmøter	110
G Møtereferat	114
H Timelogg	123
I Databasediagram	134
J Filstruktur på server	136
K Klassestruktur på blackbox	138

Figurer

1	Skjerm bilde av eksisterende programvare som viser ultralydbilder fra DolphiCam . . .	3
2	Overordnet use-case diagram	5
3	Prosjektroller	9
4	Skjerm bilde av sluttprodukt	13
5	Skjerm bilde av sluttprodukt med beskrivelse av funksjoner	14
6	Fullstendig use-case diagram	15
7	Trello Board	19
8	Overordnet distribusjonsdiagram	24
9	Skjerm bilde av logginn-skjerm i portalen	26
10	Sending og visning av ultralyd data	28
11	Skjerm bilde av ultralydbilder i portalen	33
12	Skjerm bilde av implementerte innstillinger	40
13	Pushing av innstilling fra portal til server	41
14	Oppdater innstilling fra server i blackbox	42
15	Skjerm bilde som viser palettvalg	43
16	Skjerm bilde som viser forskjellige fargepaletter	43
17	Databasediagram for tilganger	44
18	Burndown diagram for sprint 2	45
19	Skjerm bilde av chat	48
20	Gjennomsnittlig tidsforbrukt i nettleser før og etter optimalisering	52
21	Burndown diagram for sprint 3	53
22	Skjerm bilde som viser overføring av ekspertrollen	57
23	Skjerm bilde som viser åpnet og lukket vindu	58
24	Skjerm bilde av tidslinje	59
25	Burndown diagram for sprint 4	60
26	Skjerm bilde av hvordan start av tidslinje blir gjort	64
27	Skjerm bilde av tidslinjeopptak	64
28	Skjerm bilde av målestokk	65
29	Skjerm bilde av ny notifikasjon	66
30	Burndown diagram for sprint 5	67
31	Skjerm bilde av linjer i b-scan vertikal	70
32	Burndown diagram for sprint 6	72
33	Eksempel på kurve uten og med envelope	74
34	Eksempel på ultralydbilder uten og med envelope	75
35	Burndown diagram for sprint 7	76
36	Fullstendig UML skjema av databasen	135

Tabeller

1	Sprintoversikt	9
2	Kodebase	16
3	User stories sprint 1	21
4	Eksempel matrise for rådata	22
5	User stories sprint 2	37
6	Tilgangsmatrise for ulike deltaker-roller i portalen	39
7	User stories sprint 3	46
8	User stories sprint 4	54
9	Tilgangsmatrise for ulike deltaker-roller i portalen	56
10	User stories sprint 5	61
11	User stories sprint 6	68
12	User stories sprint 7	73

1 INNLEDNING

DolphiTech er et selskap som holder til på Raufoss og som siden oppstart i 2009 har jobbet med utvikling av ultralydteknologi. Selskapet består av en gruppe eksperter innen ultralyd og digital og analog elektronikk, og har siden oppstart utviklet et design for en ultralyd transduser som produserer bilder med høy oppløsning. Produktet som DolphiTech leverer blir brukt av store selskap innen luftfart og bilindustrien og har et stort markedspotensiale på verdensbasis.

1.1 Bakgrunn

DolphiTech utvikler og leverer et ultralydkamera spesielt designet for 2D og 3D inspeksjon av karbonfiberplater. Ved bruk av avansert ultralydteknologi kan kameraet produsere et bilde av hvordan det inspiserte materialet ser ut på innsiden. Denne prosessen kalles for ikke-ødeleggende utprøving (none-destructive test, eller [NDT](#)) og er kjernevirksomheten til DolphiTech.

Siden oppstart i 2009 har DolphiTech hatt fokus på å levere løsninger som inkluderer programvare og verktøy som effektiviserer prosessen rundt utførelse av NDT. Dette inkluderer blant annet verktøy for å produsere rapporter og dokumentasjon på prøveobjektet som er inspisert. Målet er å tilby kundene en komplett løsning både for inspeksjon, dokumentasjon og rapportering av inspeksjonens resultat.

DolphiTech sin målgruppe er i stor grad flybransjen, der det stilles høye krav til kvalitetsikring for å bestemme om et fly er i stand til å fly etter mistanke om en potensiell skade. Inspeksjon av skader på fly utføres av fageksperter med minst tre tusen timer opplæring. Dette kompetansekravet gjør at flyselskapene ofte må transportere inn eksperter fra andre land, noe som er en tidkrevende og kostbar prosess.

For å få mer fleksibilitet ønsker DolphiTech å utvikle en løsning der en slik test kan utføres eksternt. En operatør med begrenset opplæring håndterer utstyret lokalt, mens en sertifisert NDT-ekspert kan fjernanalysere ultralydbilder fra inspeksjonen. En slik løsning vil bidra til å skape økt bærekraft ved å redusere mengden tid, penger og resusser som inngår i å transportere eksperter mellom ulike inspeksjonsområder. Denne typen løsning kalles for Remote NDT [?] og anses for å være fremtiden innen NDT-fagfeltet.

Motivert av en rapport [4] publisert av Federal Aviation Administration [16] ønsker DolphiTech å utvide sitt tilbud med en slik løsning som er skreddersydd for deres ultralydkamera ([DolphiCam](#)).

1.2 Anvendelse

Ikke-ødeleggende utprøving [1] innebærer flere metoder for å finne feil eller svakheter på materialer uten å påføre skader eller redusere materialets fysiske egenskaper. NDT utføres ofte ved bruk av ultralydteknologi, der en transduser plasseres på materialets overflate. Transduseren sender lydbølger gjennom materialet. Lydbølgene reflekteres tilbake når de møter på et materiale med en annen akustisk impedans. Ved å måle tiden som brukes på å reflektere lydbølgene kan ultralydapparatet bestemme hvor langt inn i materialet denne endringen forekommer. DolphiTech har utviklet programvare som behandler signalene fra transduseren og omformer disse til data som kan visualiseres. Denne metoden gjør det mulig å skape et bilde av hvordan materialet ser ut på innsiden, og dermed inspisere potensielle feil eller skader.

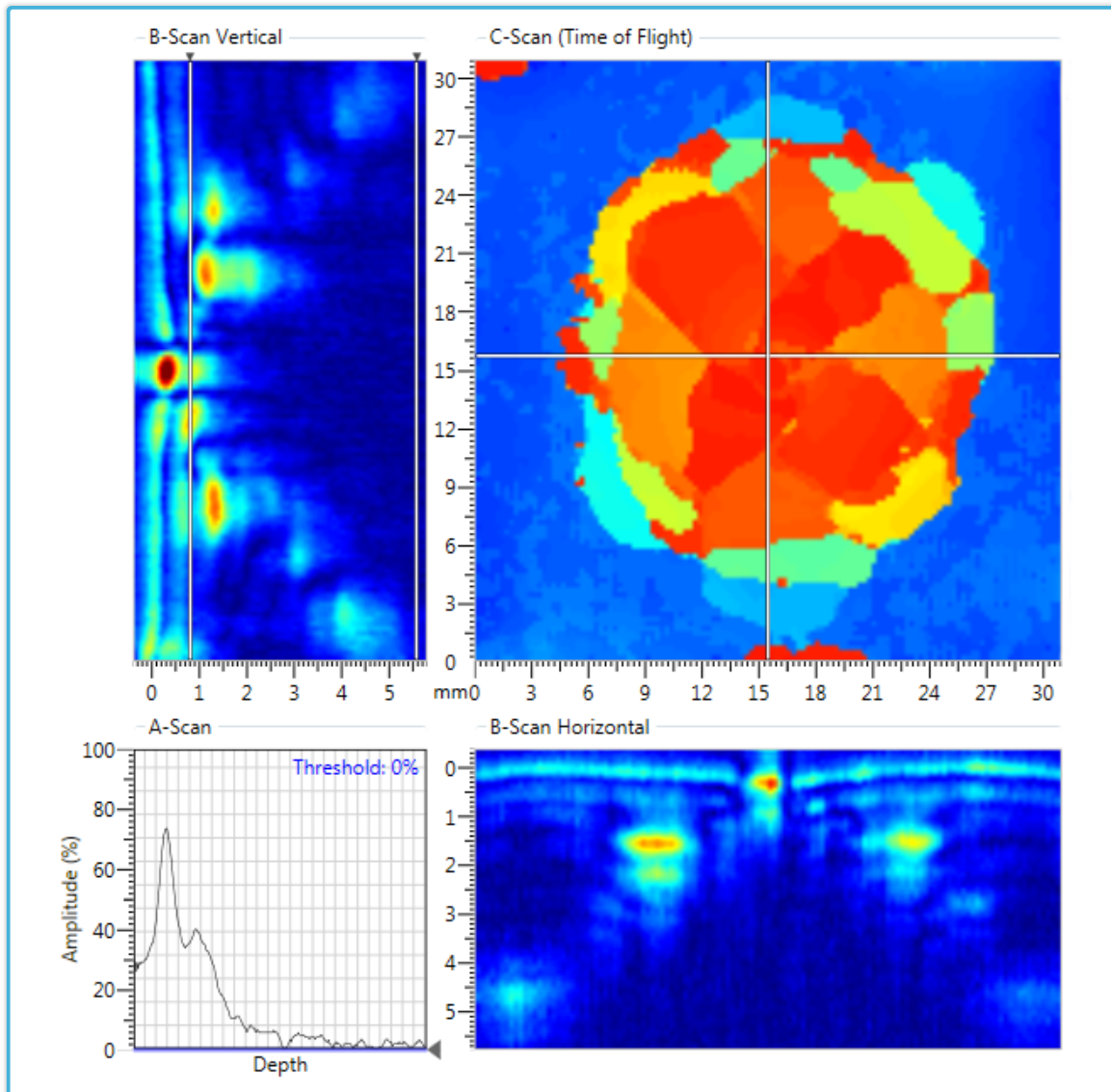
Figur 1 viser bilder produsert av DolphiCam under inspeksjon av et prøveobjekt med dagens løsning. Disse bildene gir en representasjon av materialet som blir inspisert. De ulike fargene representerer forskjellig dybde i materialet. Figuren viser det ulike ultralydbilder. Under gis en kort forklaring av hver av disse:

A-scan Viser styrken på et definert punkt i C-scan inn i materialet. Dybde vises i horisontal og styrke i vertikal retning på grafen.

B-scan Vertikal og Horisontal Viser et tverrsnitt av materialet i vertikal og horisontal retning.

C-scan Amplitude og Time of Flight Viser materialet sett ovenfra og inn.

Flybransjen er hyppig bruker av denne typen teknologi, da kvalitetsikring av flyets tilstand er viktig. Det stilles høye krav til kompetanse og opplæring for å bestemme om prøveobjektets nåværende tilstand er godkjent til flyving. På grunn av dette kravet bruker flybransjen mye resusser på å transportere eksperter innen fagfeltet mellom ulike flyplasser der en slik test skal gjennomføres.



Figur 1: Skjermbilde av eksisterende programvare som viser ultralydbilder fra DolphiCam

1.3 Oppgavebeskrivelse

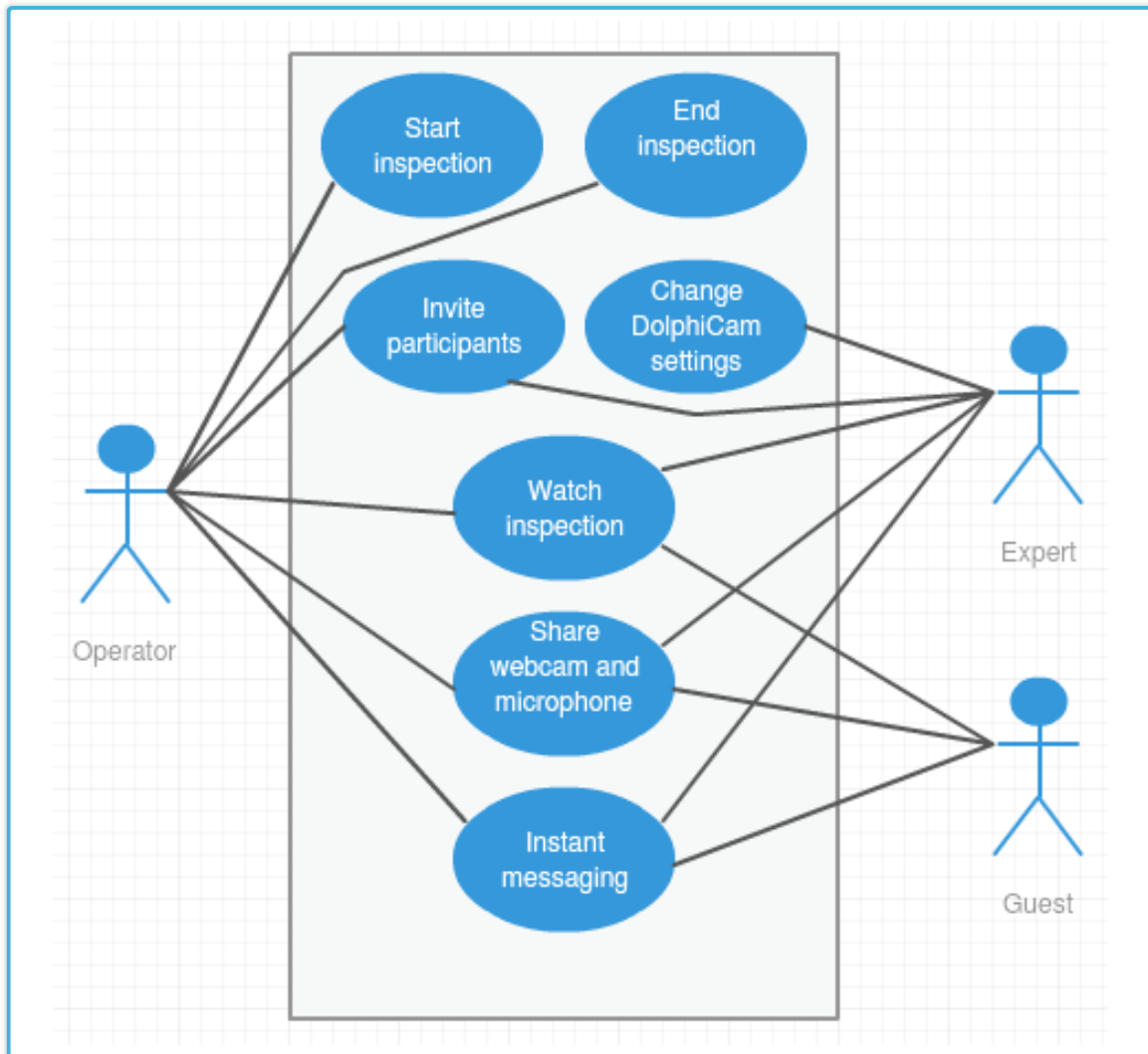
Oppgaven går ut på å skape en løsning som gjør det mulig for eksperter å analysere ultralydbilder fra en NDT-inspeksjon uten å være fysisk til stede ved prøveobjektet. En operatør som er ansvarlig for den fysiske utførelsen av en inspeksjon skal kunne starte opp en fjernstyrt økt vha. lokalt utstyr.

Operatøren skal få tilgang til å invitere andre deltakere til denne økten.

De inviterte deltakerene deltar ved å logge inn på en portal i sin nettleser som er tilgjengelig via Internet. Fra portalen skal operatøren tildele en av deltakerene rollen som ekspert, mens andre deltakere forblir såkalte gjester. En gjest har tilgang til å se på inspeksjonen, mens den som er tildelt rollen som ekspert skal ha mulighet til å konfigurere ultralydkameraet sine innstillinger direkte via portalen. Konfigurasjonene som eksperten gjør skal synkroniseres til kameraet som operatøren holder fysisk. Disse innstillingene inkluderer blant annet pulstypen som ultralydkameraet bruker når det produserer ultralydbildene.

I tillegg skal det være mulig for deltakerene å kommunisere med hverandre via portalen. Dette skal kunne gjøres via direktemeldinger, deling av webkamera og lyd. Operatøren skal ha muligheten til å koble inn flere webkameraer som typisk viser oversiktsbilder av prøveobjektet. Disse gir de andre deltakerene en oversikt over hvor på prøveobjektet det inspiseres. Figur 2 viser et sammendrag av hvilke operasjoner som skal være mulig i systemet.

Systemet som skal utvikles skal være tilsvarende eksisterende løsning som vist i figur 1.



Figur 2: Overordnet use-case diagram

1.4 Prosjektmål

Dette prosjektet har som hovedmål å lage første versjon av DolphiTech sitt system for å utføre fjernstyrte NDT-inspeksjoner (**Remote NDT**). Gjennom prosjektets varighet skal det undersøkes hva som er gode teknologier til å bygge et slikt system. Under utviklingsperioden vil målet være å konkretisere DolphiTech sitt konsept for Remote NDT vha. disse teknologiene. Resultatmålene beskrevet i kapittel 1.4.1 gir en overordnet plan for hva som ønskes implementert. Løsningen skal

også fungere som utgangspunkt for videre utvikling.

DolphiTech ønsker også underveis å finne ut av hva slags funksjonalitet som kan lønne seg å implementere i et slikt system, for så å tilpasse denne slik at produktet blir så funksjonelt som mulig.

1.4.1 Resultatmål

- Finne ut av hva som er nødvendig for å bygge et system for utførelse av remote NDT.
- Integrere DolphiCam inn i løsningen.
- La en [operatør](#) starte en fjernstyrt NDT-inspeksjon.
- Muliggjøre invitasjon av flere deltakere til en [inspeksjon](#).
- Gi operatøren mulighet til å dele webkamera som viser oversikt over [prøveobjektet](#).
- [Eksperten](#) skal kunne eksternt endre ultralydkameraet sine innstillinger.
- Gi [deltakere](#) muligheten til å kommunisere vha direktemeldinger, video og lyd.
- [Gjester](#) skal kunne delta på inspeksjon kun med tilgang til å se på inspeksjonen.

1.4.2 Effektmål

- Bidra til å skape bærekraft i NDT-prosessen.
- Gjøre NDT-prosessen mer effektiv.
- Redusere transportkostnader og tidsbruk.
- Forbedre opplæringsmuligheter innen NDT.
- Skape positiv markedsføringseffekt for DolphiTech.
- Gjøre DolphiTech mer konkurransedyktig ovenfor andre lignende leverandører.

1.4.3 Læringsutbytte

Gjennom prosjektets varighet vil vi få ferdigheter i å utarbeide konkrete problemstillinger og identifisere relevant litteratur for denne problemstillingen. Vi vil også få erfaring med prosjektarbeid i gruppe der det er flere parter.

I tillegg til emnets læringsmål kommer vi til å få erfaring innen flere tekniske fagfelt. Disse inkluderer ultralydteknologi og programvareutvikling ved bruk av flere teknologier, der de viktigste av disse er:

- C++
- PHP
- JavaScript
- MySQL databaser
- HTML5/CSS3
- WebRTC

Gjennom arbeidet skal vi bruke systemutviklingsmodellen Scrum. Det vil gi oss erfaring med smidig utviklingsmetodikk. Vi vil også få erfaring med å jobbe med en reell oppdragsgiver. Dette blir nytt fra tidligere emner. Samtidig vil avgrensning av et potensielt stort prosjekt og valg av tekniske løsninger bli et viktig læringsutbytte.

1.5 Avgrensning

Prosjektet avgrenses til implementasjon mot første versjon av DolphiTech sitt eksisterende produkt, DolphiCam. Samtidig avgrenses det til programvare og standarder som DolphiTech har utviklet i forbindelse med dette produktet. Videre vil vi holde oss til de rammer som er satt i oppgavebeskrivelsen (se vedlegg D) gitt av DolphiTech og rammer som er blitt satt på grunn av hvordan DolphiTech sin programvare og SDK er utviklet.

1.6 Rapportstruktur

Introduksjon Introduserer prosjektet og hva som er målet med oppgaven.

Arbeidsprosess Gir et overordnet innblikk i hvordan arbeidsprosessen har vært i prosjektet sammen med en generell oversikt over alle sprinter.

Sprint 1-7 Trekker frem problemstillinger, kravspesifikasjoner og implementasjoner for hver enkelt sprint.

Diskusjon Gir en kort oppsummering av sprintene og diskuterer resultatene.

Konklusjon Oppsummerer rapporten og prosjektet

2 ARBEIDSPROSESS

2.1 Valg av utviklingsmodell

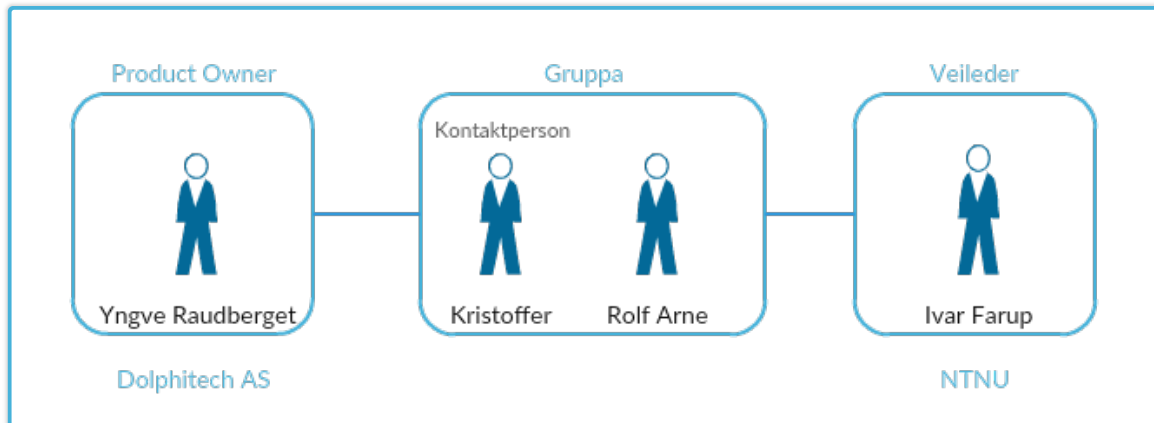
Under planlegging av prosjektet ble det valgt å bruke Scrum som utviklingsmetode. Metoden passer bra for prosjektet av ulike grunner. I første omgang har gruppa tidligere erfaringer med Scrum, noe som resulterte i en mer effektiv start på utviklingsprosessen. Samtidig har oppdragsgiver gitt oss en ganske fleksibelt kravspesifikasjon, noe som gjør at en inkrementell utviklingsmetode passer svært bra i prosjektet. Det initielle valget om å bruke Scrum ble beskrevet i prosjektplanen, se vedlegg [E](#).

Som diskutert i kapittel [2.2](#) er rollene product owner og Scrum master blitt fordelt, der Scrum master ble fordelt på begge gruppens medlemmer. Sprint planning, review og retrospective meeting blir inkludert som en del av vår implementasjon av Scrum i prosjektet. Disse blir nærmere diskutert i kapittel [2.3.2](#).

Siden gruppa for det meste har jobbet sammen har behovet for daily Scrum møter falt delvis bort. Vi har gjennom alle sprintene diskutert problemstillinger fortløpende, samtidig som vi har jobbet med hver vår del av prosjektet og oppdatert den andre på dette etter behov. Denne arbeidsformen har gjort daily scrum møter overflødige.

2.2 Prosjektroller

Under planlegging av prosjektet bestemte vi roller og ansvar. Siden gruppa består av kun to personer ble det bestemt at begge medlemmer deler rollen som Scrum master, mens Kristoffer fikk rollen som kontaktperson for gruppa. Videre ble det naturlig å sette Yngve i DolphiTech som product owner, da det er han vi hovedsakelig forholder oss til som oppdragsgiver. [Figur 3](#) viser en grafisk representasjon av rollene.



Figur 3: Prosjektroller

Utenfor de tradisjonelle Scrum rollene er Ivar Farup veileder for gruppa. Han har bistått gruppa med oppfølging av prosjektets fremdrift gjennom hver sprint. Ivar har også bistått med tilbakemeldinger på rapportens innhold og struktur.

2.3 Sprinter

En sprintplan ble satt opp i planleggingsfasen. Denne vises i tabell 1. Tabellen viser start og slutt dato for hvert enkelt sprint, samtidig som totalt antall timer estimert og brukt er summert opp. Tidsbruk gjelder for begge utviklerene sammenlagt.

Tabell 1: Sprintoversikt

Sprint	Start	Slutt	Estimert (timer)	Brukt (timer)
1	30.01.2017	12.02.2017	-	94
2	13.02.2017	26.02.2017	79	63
3	27.02.2017	12.03.2017	82	72
4	13.03.2017	26.03.2017	79	66
5	27.03.2017	09.04.2017	75	69
6	10.04.2017	23.04.2017	49	32
7	24.04.2017	07.05.2017	20	11

2.3.1 Struktur på sprinter

DolphiTech har i stor grad gitt oss frihet til å finne ut hvordan ønsket funksjonalitet skal implementeres. En overordnet beskrivelse av hvordan det ferdige produktet bør se ut har blitt presentert for gruppen, samtidig som det ble gitt et større spillerom for hvordan dette oppnås og implementeres. Sprintene er derfor i stor grad være strukturert slik at funksjonalitet implementeres i prioritert rekkefølge, der de mest sentrale elementene i produktet implementeres i de første sprintene. Det viktigste av kjernefunksjonalitet ble implementert allerede i de tre første sprintene.

Alle sprinter har hatt en varighet på to uker og de startet på en mandag og endte på søndag to uker etter. Alle sprinter utenom sprint 6 og 7 har hatt samme utviklingstid. Når sprintoppsettet ble bestemt ble det ikke tatt hensyn til påskeferie, noe som vi ser i ettertid av planleggingsfasen at burde blitt gjort. Som løsning på dette valgte vi å la påskeferie inngå som en del av sprint 6. Dette resulterte i at vi måtte avtale en redusert arbeidsmengde for denne sprinten. DolphiTech var enig i at dette var en grei måte å løse det på.

Begge gruppens medlemmer har fungert som utviklere og har jobbet på systemet. Det var satt opp 20 timer pr. utvikler pr. uke, som tilsvarer en total utviklingstid på 80 timer per sprint. Resterende 10 timer pr. utvikler pr. uke var reservert til til møter, rapportskrivning og andre oppgaver som ikke inngikk direkte i en sprint.

Under utviklingsperioden har gruppen jobbet sammen, med noen få unntak der oppgaver ble fordelt på forhånd. Av praktiske årsaker ble utviklingsprosessen fordelt på to ulike deler av systemet, klient/server og programvaren som kommuniserer med DolphiCam. Hvert gruppemedlem hadde fokus på hver sin del. Samtidig har begge bidratt til utvikling av begge delene.

Vi har valgt å kun legge inn user stories som har fokus på brukeren av systemet. Oppgaver som å sette opp server, installere programvare og test av ytelse har ikke blitt satt opp som user stories. Disse oppgavene ble gjennomført som en del av user storyen og tid til dette arbeidet er dermed estimert inn i den aktuelle storyen.

I kapittel 3 til 9 beskrives user stories, kravspesifikasjoner og implementasjoner for hver sprint. I tillegg inneholder hver sprint en introduksjon som kort forklarer hovedfokuset for sprinten. En egen konklusjon summerer opp arbeidsprosessen i sprinten. Det vises også en tabell med oversikt over alle stories for hver sprint. Disse tabellene består av fire kolonner:

Er et unikt nummer som brukes som referanse til den gitte user storyen

User story Beskriver overordnet hva som er målet med storyen.

Estimert Antall timer som har blitt estimert for gitt story. Estimeringen gjelder for sammenlagt tid for begge utviklerene.

Brukt Viser faktiske antall timer brukte på gitt story.

2.3.2 Sprint møter

I dette kapittelet diskuteres de tre møtene som inngår i hver av sprintene.

Sprint planning meeting ble holdt ved oppstart av ny sprint mens sprint review- og retrospective meeting som avsluttende møter. Av praktiske årsaker holdt vi alle disse møtene samtidig, der vi begynte med de avsluttende møtene for innværende sprint, for så å gå videre til planlegging av neste sprint. På denne måten fikk vi redusert reisetid og sparte tid rundt møteprosessen.

Naturlig nok ble det i disse møtene noe vanskelig å skille mellom de tre ulike møtene siden de gjerne overlappet hverandre. Vi valgte å ikke sette et fast skille på de ulike møtene, men heller sørge for å trekke frem alle nødvendige faktorer fra hvert møte i løpet av møtetiden.

Sprint Planning Meeting Sprint planning meeting ble gjennomført dagen sprinten startet. Til stede på møtene var primært vår kontaktperson i DolphiTech, samtidig som andre i DolphiTech ble med etter behov og ønske. Det ble bestemt et overordnet mål med sprinten og hvilken funksjonalitet som var ønskelig å implementere. User stories ble så flyttet til product backlog fra sprint backlog. Antall stories som ble inkludert i sprinten ble hovedsakelig basert på fortløpende estimering under møtet og det ble avtalt med oppdragsgiver hva som ble en passelig arbeidsmengde for gruppa.

Sprint Review Meeting Før hvert sprint planning meeting, med ett unntak av aller første, ble det gjennomført et sprint review meeting. Her gikk vi igjennom all ny funksjonalitet som har blitt utviklet i sprinten. Demo ble vist på storskjerm, slik at alle hadde mulighet til å se og komme med innspill på det som vi hadde gjort. DolphiTech gav tilbakemelding på produktet og på om funksjonaliteten var slik de ønsket. Eventuelle avvik eller endringer ble lagt til som nye user stories for neste sprint.

Sprint Retrospective Meeting Rett etter hver sprint review ble det gjennomført et sprint retrospective meeting. Her gikk vi igjennom det som gikk bra, ikke så bra og hva som kunne være gjort bedre. Om det ble noen user stories som ikke ble gjennomført i sprinten ble de tatt opp her og lagt tilbake til product backlog.

2.3.3 Estimering

Estimeringen ble gjort av gruppa og ble i stor grad bestemt basert på medlemmenes erfaring fra tidligere prosjekter. Scrum poker ble ikke brukt under estimering. Gruppen diskuterte arbeidsomfanget og ble enige om estimerer for hver user story.

Fra og med sprint to har det blitt gjort estimeringer for alle user stories som ble definert. I første sprint ble det ikke gjort estimeringer. Dette fordi vi ikke hadde fått satt oss skikkelig inn i hvordan

oppdragsgivers eksisterende teknologier fungerte og dermed hadde for lite innsikt til å vite omfanget av oppgavene.

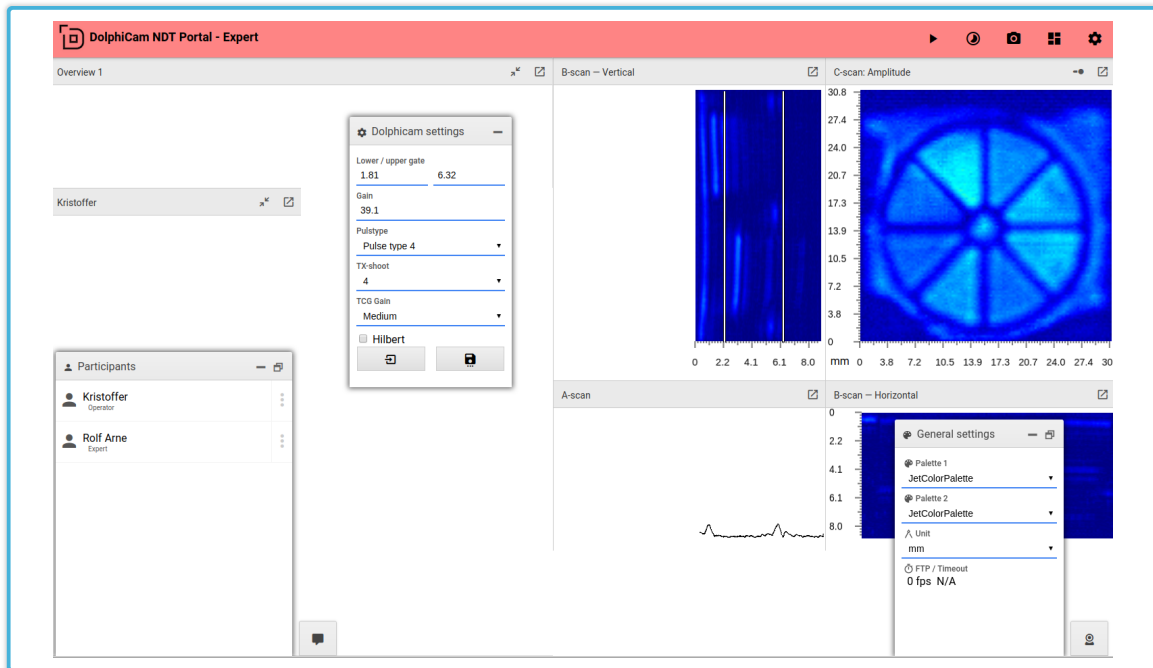
Etter første sprint hadde gruppa fått en bedre innsikt i oppgaven, noe som gjorde fremtidige estimeringer lettere. Derfor ble det fra og med sprint to gjort estimeringer for hver enkelt user story. Disse estimeringene har blitt brukt som grunnlag for burndown diagram som det vises til i hver av sprintene.

2.3.4 Verifisering og validering

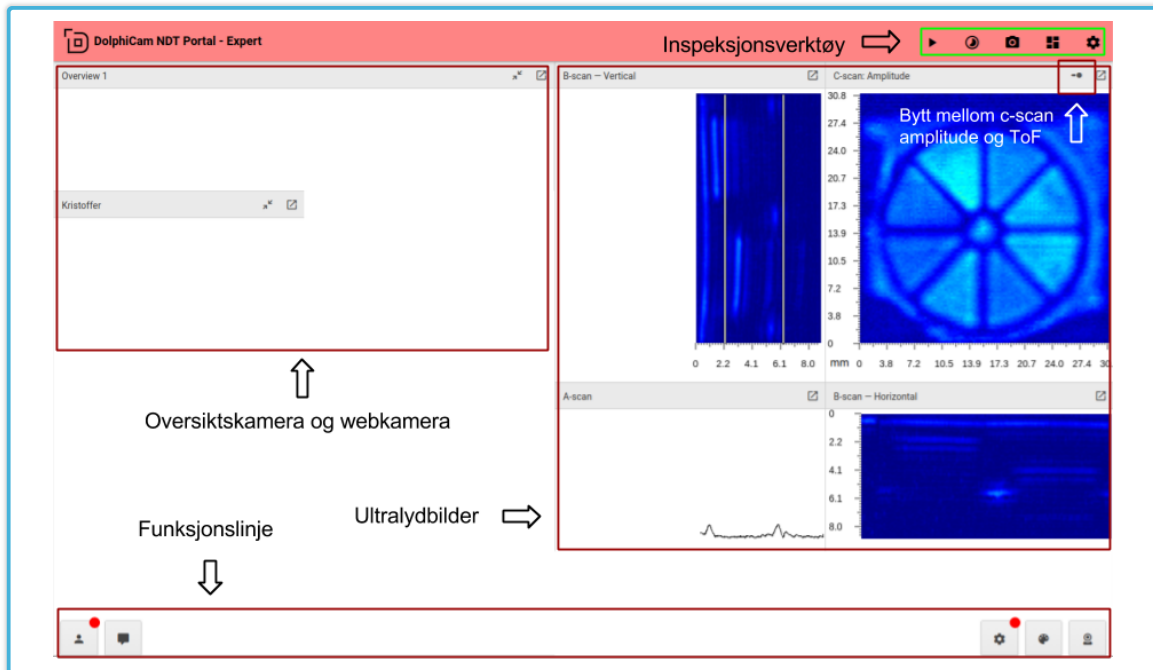
Når en user story ble ferdig implementert har storyen gått gjennom en verifisering- og validerings prosess. I denne prosessen sjekket utviklerne at funksjonaliteten tilfredstilte kravspesifikasjonene som ble definert av user storyen. Det ble også sjekket om funksjonen i tilstrekkelig grad var brukervennlig og ble integrert i GUI på en god måte. Under Scrum review meeting gav oppdragsgiver bekreftelse på at storyen tilfredstilte ønskene som ble presentert under Scrum planning meeting.

2.4 Oppsummering sprintresultat

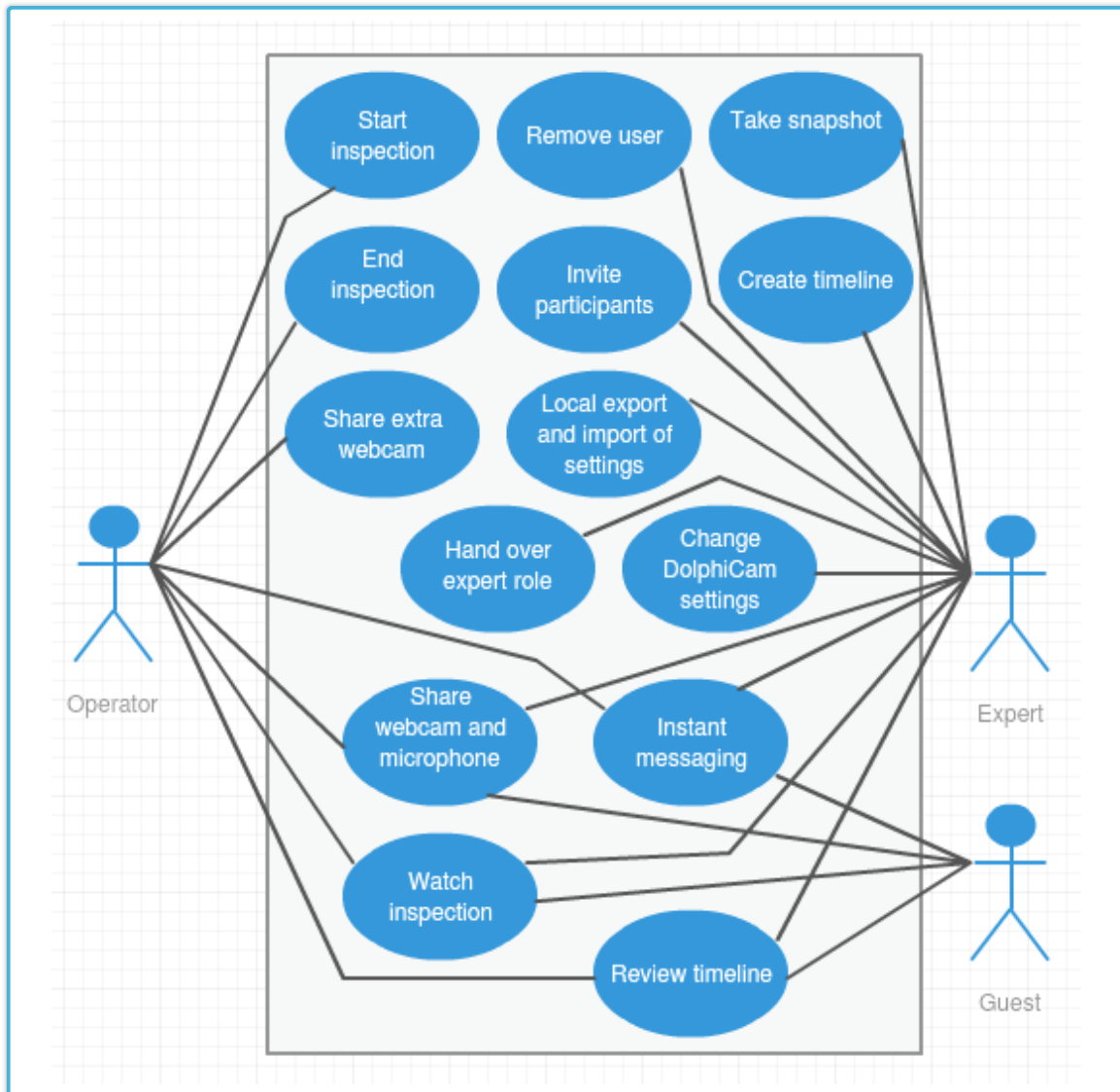
I dette kapitlet gis en oversikt over det ferdige produktet som resultat av gjennomførte sprints. Figur 4 viser skjermbilde av portalen slik den ser ut etter at sprint 7 ble ferdig. En nærmere beskrivelse av hvordan grensesnittet er bygd opp vises i figur 5.



Figur 4: Skjerm bilde av sluttprodukt



Figur 5: Skjermbilde av sluttprodukt med beskrivelse av funksjoner



Figur 6: Fullstendig use-case diagram

Figur 6 viser den fullstendige kravspesifikasjonen i form av et use-case diagram. Under gis en oppsummering av hovedinnholdet i hver av sprintene:

Sprint 1 Streaming av ultralyddata mellom DolphiCam og klienter.

Sprint 2 Overføring av oppdaterte innstillinger for DolphiCam fra portalen.

Sprint 3 Kommunikasjon mellom klientene vha. video, lyd og direktemeldinger.

Sprint 4 GUI, tildeling av tilganger mellom deltakere og permanent lagring av ultralyddata fra stream.

Sprint 5 Utbedret funksjonalitet for lagring av stream og mer funksjonell presentasjon av ultralyddata.

Sprint 6 Opprydding GUI, eksportering og importering av innstillinger, samt lagring av enkeltbilder fra stream.

Sprint 7 Tilpassinger av noen innstillinger mot DolphiCam og ferdigstilling av produktet.

Sprintene resulterte i en kodebase som domineres av Javascript. Siden Javascript ble brukt både på klient og server er mesteparten av den mest sentrale logikken implementert i denne delen av kodebasen. Tabell 2 viser fordelingen av kode mellom de ulike delene av systemet og programmeringsspråkene som har blitt brukt.

Tabell 2: Kodebase

	Språk	Kommentarer (linjer kode)	Kode (linjer kode)
Blackbox	C++	473	1 178
Server	PHP	143	1 388
	Javascript	105	910
Klient	Javascript	519	4 251
	CSS	24	1 324
	HTML	0	170

Filstrukturen til koden for klient og server beskrives i vedlegg J. Vedlegg K viser et klasse diagram som beskriver hvordan den ferdige programvaren som snakker med DolphiCam er bygd opp.

2.5 Dokumentasjon

Under utviklingsprosessen i prosjektet har kode blitt delvis dokumentert. Det har blitt brukt docblock-standardene for å dokumentere alle globale funksjoner og klasser til den grad det er funksjonelt. Samtidig er datastrukturer som er brukt dokumentert i form av egne diagram.

Gjennom sprintkapitlene er det ikke satt fokus på å dokumentere hvordan datastrukturen har utviklet seg underveis med unntak av i noen implementasjoner. Vedlegg I inneholder et komplett database-diagram for det ferdige produktet. Basert på ønske fra oppdragsgiver har all dokumentasjon og kode blitt produsert på engelsk.

For hver av kodebasene skal det også utvikles en egen readmefil. Denne filen skal beskrive hvordan oppdragsgiver kan ta i bruk koden.

2.6 Statusmøter

I planleggingsfasen av prosjektet ble det satt opp tre statusmøter. Møtene ble brukt for å ta opp saker som ikke inngikk direkte i en sprint. Alle møtene forgikk i første planlagt møtedag for månedene mars, april og mai. Møtereferat finnes i vedlegg F.

2.6.1 Milepæler

I planleggingsfasen ble det satt opp tre milepæler. Hver sprint hadde som mål å implementere funksjoner som gjorde at vi kunne nå disse milepælene. Det endte opp med at vi var veldig usikre i planleggingsfasen hva som måtte til for å nå milepælene. Det gjort at det var satt opp god tid for å nå de. De tre milepælene var følgende:

Følg med på en stream Her skulle vi data hentet ut fra DolphiCam og vist i en nettleser. Denne skulle bli nådd i enden av sprint 2. Målet ble allerede oppnådd i sprint 1.

Endre innstillinger i DolphiCam Målet var å endre innstillinger i DolphiCam over nett. Milepælen skulle være nådd i løpet av sprint 3. Målet ble oppnådd i sprint 2.

Webkamera -og lyd kommunikasjon Målet var å sett opp webkamera, slik at deltagere kunne kommunisere over video og lyd. Målet skulle bli nådd i løpet av sprint 4. Målet ble oppnådd i sprint 3.

Resterende sprinter var tilrettelagt for annen funksjonalitet som ikke var avklart på forhånd, men som vi visste ville komme.

2.7 Workshops

Det ble holdt workshops 26 og 27 april der målet var å bistå oppdragsgiver med å sette programvare i deres eget miljø.

2.7.1 Workshop 26. april

Morgenen 26. April fikk vi en epost fra DolphiTech der de spurte om vi hadde mulighet til å komme den dagen, da ikke alle var tilgjengelig til workshoppen dagen etter. Som de fleksible studentene vi er, reiste vi til DolphiTech på Raufoss. Vi har hatt problemer med å få til hilbert funksjon (user story #11). Denne funksjonen skal bli kjørt på bildedata mottatt fra DolphiCam slik at bildene blir mer jevne/glattere. En feil i DolphiLib gjorde at denne funksjonen ikke fungerte som den skulle. Derfor ble det kjørt en oppdatering av DolphiLib på datamaskinen vi har brukt for å utvikle programvaren. Etter oppdateringen kunne vi raskt bekrefte att funksjonen fungerte som den skulle. Det ble også gjort noen endringer i portalen for å fremheve hvilken rolle som er tildelt deltakeren.

2.7.2 Workshop 27. april

Workshop'en var planlagt i sprint planning meeting og det ble satt opp en varighet på 7 timer. Først hjalp vi DolphiTech med å lage en presentasjon av systemet som de skal vise på en messe 8 og 9 mai. Vi bidro så med oppsettet av vår programvare på en av DolphiTech sine datamaskiner. Vi kom ikke helt i mål, på grunn av vi ikke fikk kontakt med DolphiCam i en virtuell maskin. Når vi endelig fikk kontakt, var dagen allerede over. DolphiTech skulle ta kontakt hvis de fikk videre problemer med oppsettet.

2.8 Utviklingsverktøy

2.8.1 Trello

Trello er et utviklingsverktøy bygd for smidig utviklingsmetodikk. Vi har valgt å bruke dette verktøyet for å planlegge og holde kontroll på sprinter. Som vist i figur 7 har vi laget seks lister:

Product backlog Dette gjelder user stories som skal gjennomføres, men inngår ikke i gjeldende sprint.

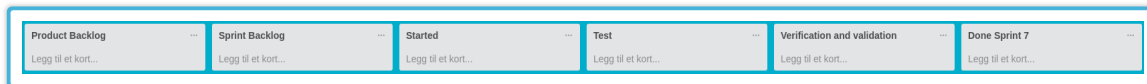
Sprint backlog Her ligger alle user stories som er satt opp for denne sprinten

Started Når en av utviklerne har begynt på en user story, blir den flyttet hit.

Test Når utvikleren tror user storien er ferdig blir den satt hit. Samme utvikler sjekker så om implementasjonen ikke påvirker noen andre deler av systemet.

Verification and validation Her sjekker begge utviklerene om implementasjonen møter kravene til user storien og den fungerer som den skal.

Done sprint X For hver sprint finnes det en egen liste for fullførte stories, der X er nummeret på sprinten. Når en user story har vært gjennom alle tidligere punkt, blir den flyttet hit for å indikere at den er ferdig implementert.



Figur 7: Trello Board

Vår kontaktperson til DolphiTech, Yngve Raudberget, har fått tilgang til Trello brettet. Det har gitt han muligheten til å følge med på fremdriften underveis i sprintene og legge til nye user stories i product backlog.

2.8.2 Bitbucket

Som versjonskontroll har det blitt brukt GIT sammen med Bitbucket. Det har blitt brukt to repositories, et for koden brukt til å snakke med DolphiCam og et for serverside koden.

2.8.3 Sharelatex

Sharelatex [14] har blitt brukt for å utforme rapporten. Sharelatex er en kollaborativ skytjeneste som lar flere samarbeide om å skrive Latex-kode [17].

2.8.4 Toggle

For å registrere tid brukt på forskjellige aktiviteter i løpet av prosjektet har skytjenesten Toggl [15] blitt brukt. Tjenesten har også blitt brukt til å produsere timeloggen i vedlegg H.

2.9 Rapportskrivning

I utviklingsperioden har det blitt tatt notat for hver av sprintene. Disse ble brukt som grunnlag til ferdigstilling av rapporten i etterkant av utviklingsperioden. Det ble satt av tid etter siste sprint til å fokusere på utarbeiding av rapporten. Mesteparten av rapportens innhold ble produsert i denne

perioden. Unntaksvis har beskrivelser av resultat fra tester utført under utvikling blitt utformet i utviklingsprosessen.

3 SPRINT 1

3.1 Innledning

Hovedmålet med denne sprinten var å muliggjøre visning av ultralydbilder produsert av DolphiCam, fra en ekstern lokasjon via en nettleser. Samtidig skal denne visningen kun gjøres tilgjengelig for de deltakere som blir invitert av operatøren som utfører inspeksjonen. Mange viktige beslutninger som valg av programvare og overordnet datastruktur ble blitt tatt i denne sprinten. Disse har vært grunnlaget for videre utvikling gjennom hele utviklingsprosessen.

3.2 User Stories

User stories i denne sprinten vart i stor grad overordna grunna gruppens begrenset oversikt over hva som skal til for å produsere produktet som var målet i sprinten. Ut av de totalt fire user stories som ble satt opp kan disse deles opp i to sammenhengende krav. Kravspesifikasjonen beskriver disse.

Tabell 3: User stories sprint 1

#	User story	Estimert (timer)	Brukt (timer)
1	Som deltaker vil jeg ha mulighet til å logge inn.	-	-
2	Som operatør vil jeg starte en inspeksjon	-	-
3	Som operatør vil jeg streame DolphiCam data	-	-
4	Som ekspert/gjest vil jeg ha mulighet til å se på en stream	-	-
		-	-

3.3 Kravspesifikasjon

3.3.1 Innlogging

Dette avsnittet beskrives kravspesifikasjonene for user story 1 som ble definert under møte med oppdragsgiver.

Oppdragsgiver ønsker en løsning som tillater at hvem som helst kan bli invitert til å delta på en inspeksjon. Dette kravet gjør tradisjonell innlogging vha. brukernavn og passord som opprettes på forhånd ikke er et alternativ. Istedenfor skal innlogging skje med en såkalt engangskode (kalles her for

token). Denne engangskoden skal kun kunne brukes en gang og skal gi tilgang til en gitt inspeksjon. En nærmere beskrivelse av engangskode finnes i kapittel 3.5.5.

Bare personer som har tilgang på utstyr levert fra DolphiTech skal kunne starte en inspeksjon. Derfor skal en ny inspeksjonsøkt startes automatisk når en operatør starter sitt ultralydutstyr. Operatørens utstyr skal da signalisere mot portalen at en ny økt skal startes. En ny inspeksjonsøkt skal opprettes og en ny bruker blir automatisk opprettet i denne økten. Denne brukeren dedikeres til operatøren som får spesielle rettigheter i portalen. Etter innlogging skal det være mulig for operatøren å invitere flere deltakere ved å generere flere engangskoder. Sistnevnte skal ikke implementeres som en del av denne sprinten. Foreløpig skal operatøren presenteres med et grensesnitt der ultralydbilder vises.

3.3.2 Streaming av ultralyddata

Ultralyddata som blir generert av DolphiCam skal overføres til alle klienter knyttet til tilhørende inspeksjonsøkt. Med ultralyddata menes det rådata i form av en gruppe matriser der hver matrise representerer et bilde. Se beskrivelse av de ulike typene bilder i kapittel 1.2. Hvert punkt i matrisene representerer et punkt på transduserens flate og består av tall fra 0 til 255. Disse verdiene forteller hvor dypt materialet er på det gitte punktet. I tabell 4 vises en forenklet versjon av hvordan en slik matrise ser ut. Eksempelet viser hvordan matrisen kan se ut dersom prøveobjektet som blir inspisert er tynt rundt kanten, og tykkere i midten.

Tabell 4: Eksempel matrise for rådata

23	22	41	33	41
25	201	237	200	37
30	205	241	207	35
21	255	251	249	38
45	21	37	32	45

For å presentere matrisene sine verdier som et lesbart bilde kobles verdiene opp mot et fargespekter (kalt fargepalett). I tabell 4 er det satt inn farger for å illustrere dette. Hvilken som helst farge kan brukes, men vanligvis brukes fargespekteret fra blå til rød. Blå for lave verdier og rød for høyere verdier. Matrisene som DolphiCam genererer er i praksis en større versjon av det som vises i tabell 4.

For å produsere et resultat som lar deltakere se på disse ultralyddataene i sanntid må en rekke forskjellige moduler implementeres. Under utarbeiding av kravspesifikasjonen ble følgende forenklet versjon av figur 10 produsert:

- Lese ut matriser for alle fem typen bilder fra DolphiCam

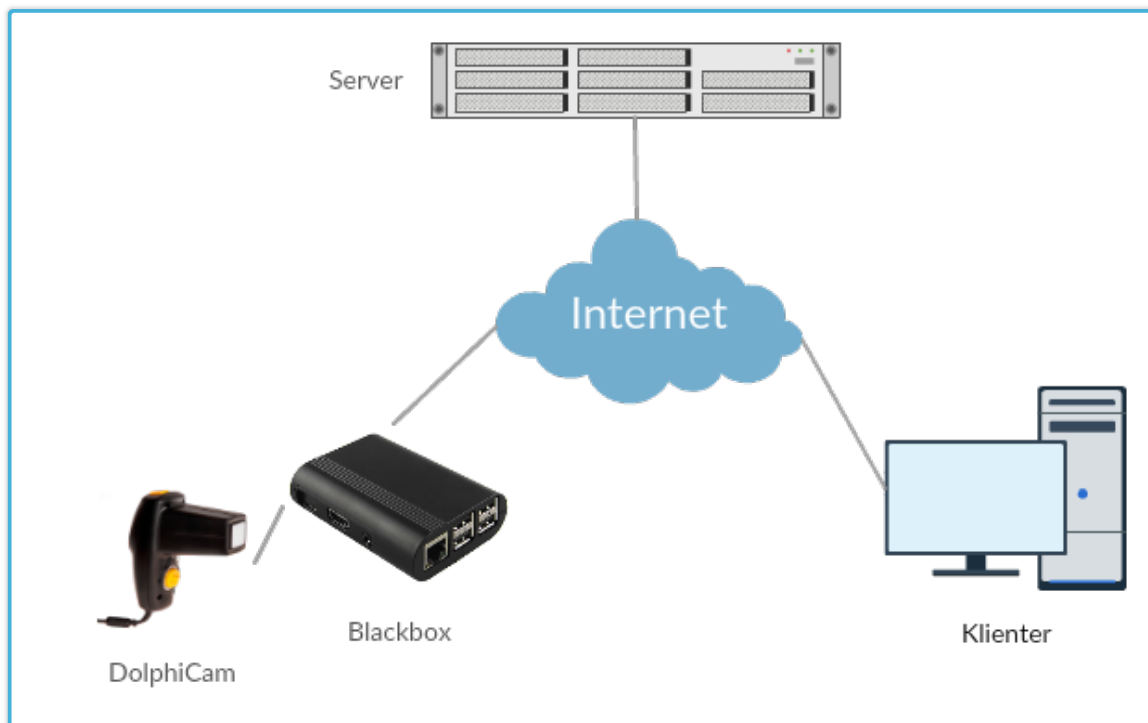
- Distribuere matrisene til tilkoblede klienter
- Omforme (rendere) matrisene til bilder som kan tolkes

DolphiCam har en oppdateringsfrekvens på cirka 3.6 bilder per sekund. Siden systemet skal operere i sanntid må implementasjonen kunne håndtere en tilsvarende oppdateringsfrekvens for å gi et komplett sluttprodukt. For å tilfredstille dette kravet settes skal systemet takle en frekvens på minst fire utlesinger per sekund.

DolphiTech har også utviklet et SDK kalt [DolphiLib](#). Dette biblioteket skal brukes for å snakke med DolphiCam og vil være det som deler av systemet bygges på toppen av.

3.4 Implementasjon

Tidlig i sprinten ble det brukt en del tid på å diskutere hvordan kravspesifikasjonene kunne implementeres. Det ble klart at implementasjonen ville kreve en klient-server løsning. Portalen som skal implementeres skal være en webklient som aksesseres via nettleser. Samtidig vil det være behov for en mini-datamaskin som DolphiCam kan kobles til fysisk over USB. Mini-datamaskinen (kalt blackbox) vil lese ut data fra DolphiCam over USB og skal så formidle denne informasjonen ut til deltakere logget på portalen. For å strømlinjeforme informasjonsflyten ble det bestemt av server skal fungere som et felles mellomledd mellom klientene og blackbox. Figur 8 illustrerer hvordan de ulike komponentene er distribuert.



Figur 8: Overordnet distribusjonsdiagram

3.4.1 Server

Serveren skal fungere som et knyttetpunkt mellom blackbox og webklientene. En kan derfor dele serverens rolle i to deler; den som brukes til å tjene klientene med selve portalens grensesnitt- og det som formidler data mellom blackbox og webklientene i sanntid. Tradisjonelle webservere er ikke spesielt godt egnet til å formidle informasjon i sanntid, noe som er nødvendig for å kunne strøme ultralyddata mellom blackbox og klient. Av denne grunn ble det valgt å sette opp en server med to ulike serverprogramvarer som brukes til hvert av disse formålene.

Valg av programvare på server

Basert på gruppens tidligere erfaringer ble det valgt å bruke en rekke kjente programvarepakker på server:

Lighttpd Webserver for å tjene både statisk og dynamisk innhold til klientene over HTTP.

Node.js Webserver som baseres på Javascript-kode på serverside. Brukes sammen med Node-modulene Express og Socket.io som sammen muliggjør toveis kommunikasjon mellom klient og server. Node.js brukes til å streame ultralyddata fra server og ut til klientene.

MySQL Databasemotoren MySQL brukes for å lagre data om ulike inspeksjonsøker, deltakere og fremtidig også til å lagre annen metadata om inspeksjonen. Vedlegg I viser den ferdige datastrukturen etter fullført utviklingsarbeid.

PHP Som serverside språk brukes brukes PHP til å generere dynamisk innhold.

3.4.2 Blackbox

Blackbox er en liten datamaskin som fungerer som mellomledd mellom DolphiCam og server. Siden DolphiCam har et USB grensesnitt er det nødvendig med en datamaskin som kan snakke med DolphiCam over USB og kommunisere data fra DolphiCam opp til server. Blackbox brukes også til å kjøre DolphiTech sitt SDK, dolphilib. Dette inneholder funksjonalitet for å abstrahere lesing og skriving til DolphiCam sitt interne register ned til funksjonskall som kan gjøres fra C++.

Valg av blackbox programvare

Basert på ønsker fra oppdragsiver og noen valg ble det valgt en programvare til bruk på blackbox.

Ubuntu DopphiTech har basert seg på GNU/Linux. Ubuntu har blitt brukt i tidligere prosjekter, samtidig som det har en stor brukerbase i forhold til mange andre distribusjoner. Derfor ble det valgt å bruke Ubuntu på blackbox.

C++ Programvaren som kjører på blackbox skal snakke med DolphiLib. Selve biblioteket, eksempelkode og dokumentasjonen som medfølger er utviklet i C++. Det ble derfor naturlig å utvikle programvaren i samme språk.

Libcurl Libcurl [2] er et multiprotokoll overføringsbibliotek. Det ble valgt å bruke dette grensesnittet for å utføre http-forespørsler fra programvaren på blackbox. Dette fordi libcurl er et mye brukt og godt dokumentert bibliotek. Samtidig er det fritt for bruk i kommersielle prosjekt så lenge curl-lisensen er med programvaren.

Zlib Zlib [3] er e bibliotek som inneholder en rekke komprimeringsalgoritmer. Biblioteket brukes for å komprimere data som skal sendes fra blackbox til server.

3.4.3 Klienter

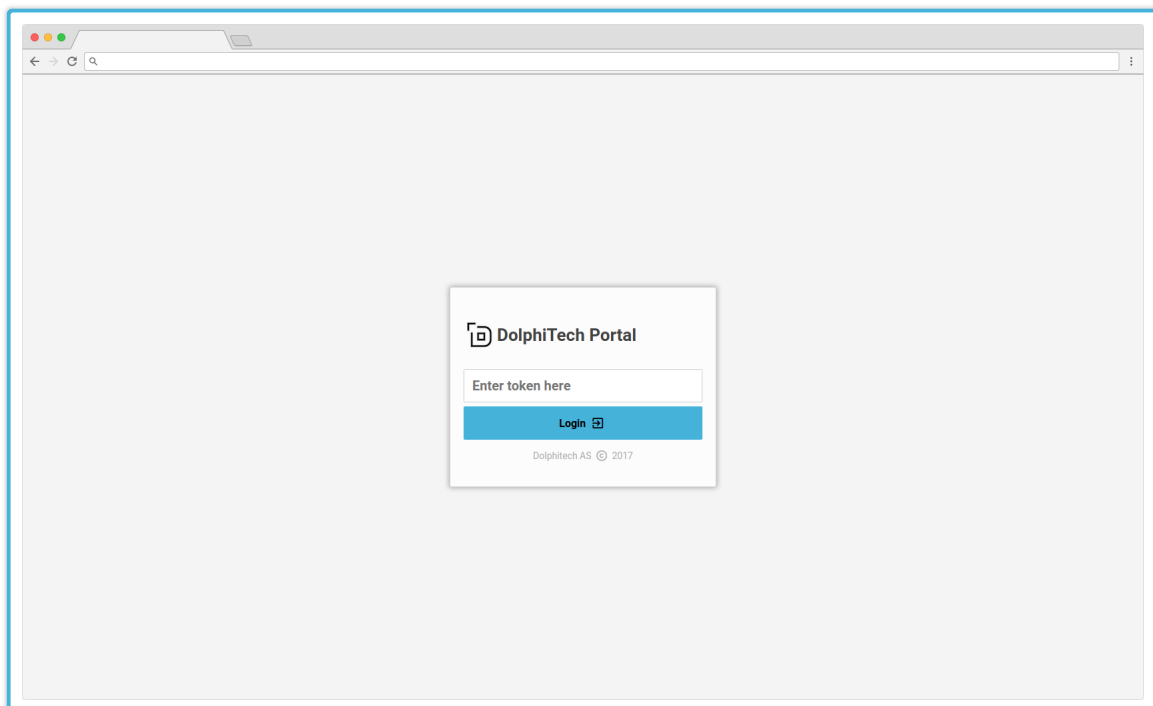
Klientene er der deltakere av en inspeksjonsøkt kobler seg opp mot server via portalens webgrensesnitt. Deltakeren bruker en nettleser for all interaksjon med portalen og baserer seg på bruk av følgende webteknologier:

Javascript All klientside-logikk kjøres ved bruk av Javascript. Nettleser må ha støtte for søtte for ECMAScript6 [9].

HTML5/CSS3 Grensesnittets grafiske layout blir produsert med HTML5/CSS3.

3.4.4 Innlogging

Som diskutert i kapittel 3.3.1 skal det brukes engangskoder for å logge inn. Figur 9 viser skjermbilde av logginn-skjermen som brukeren presenteres med.



Figur 9: Skjermbilde av logginn-skjerm i portalen

Sesjonsnøkkel

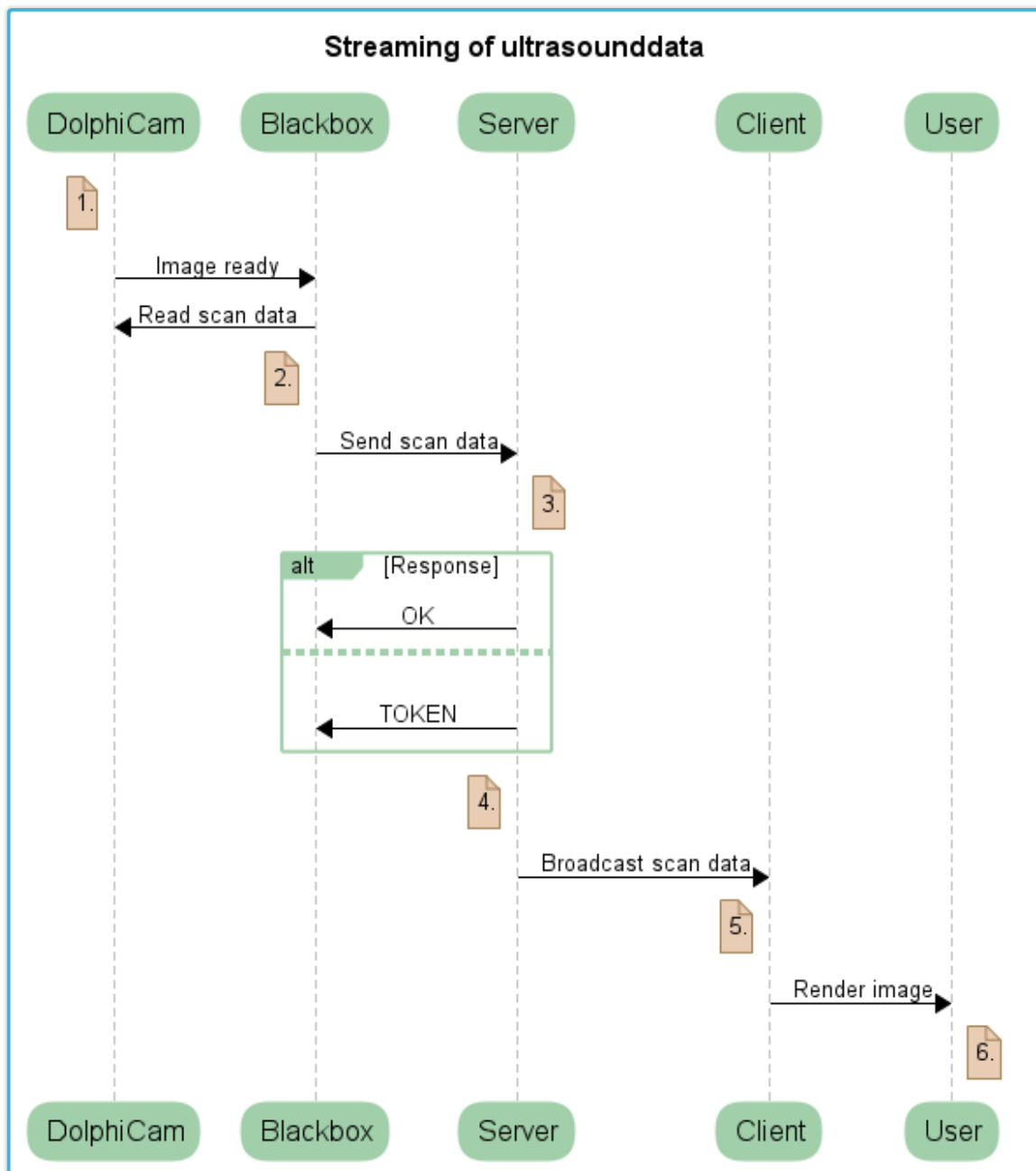
Etter at bruker har oppgitt sin engangskode genereres en sesjonsnøkkel. Denne lagres i nettleseren og brukes som autentisering frem til brukeren logger ut eller sesjonen blir avsluttet. Nøkkelen genereres ved bruk av følgende PHP-kode:

```
1 | $salt    = 'IEktPA6tEp4W3OPREozU1GZpusurVCIC'; // Randomly generated
   |     salt
2 | $userID = $this->id(); // Unique ID of the user
3 | do {
4 |     $random = rand(1111111111, 9999999999);
5 |         // Concentrate userID, salt and random number
6 |     $string = $userID.$salt.$random;
7 |         // Hashing the string using SHA256
8 |     $key = hash('sha256', $string);
9 |         // Check if key already exists in database
10 |     $exists = self::$dbh->notEmpty($sql, array($key));
11 | } while($exists); // Loop untill we get a new unique key
```

Koden genererer en sesjonsnøkkel som er unik på tvers av alle brukere i systemet. Nøkkelen knyttes opp mot selve brukeren slik at hvilken bruker som er innlogget kan identifiseres basert på denne nøkkelen. Vedlegg I beskriver databasestrukturen.

3.4.5 Streaming av ultralyddata

Sekvensdiagrammet i figur 10 viser hvordan ultralydbilder sendes fra DolphiCam og ut til klientene. Figuren inneholder tall fra 1 til 6, disse refererer til punktlisten nedenfor der de ulike sekvensene beskrives.



Figur 10: Sending og visning av ultralyd data

1. **Les ut data fra DolphiCam**

Når programvare på blackbox starter vil først programvaren initialisere tilkoblingen til Dolphi-

Cam. Dette gjøres ved et kall mot SDK'et. Det settes så opp en callback-funksjon som SDK'et vil kalle hver gang DolphiCam har skannet inn ny ultralyddata og disse er klare for utlesing. Callback-funksjonen leser ut rådata fra DolphiCam og gjør nødvendige behandlinger. Rådatasettet som leses ut av DolphiCam blir gjennom en serie behandlinger omformet til en matrise representert i strengformat. Eksempel 1 viser en forenklet versjon av hvordan denne strengen ser ut.

```
1 | [ 1, 2, 3;  
2 |   4, 5, 6;  
3 |   7, 8, 9]
```

Kodesnutt 1: Eksempel på matrise for ultralyddata i strengformat

Hver av disse matrisene konsentreres sammen til en lang streng som inneholder matriser for hver av de fem ulike ultralydbildene. Som vist i eksempel 2 skiller de ulike matrisene med et &-tegn.

```
1 | [1,2,3;4,5,6;7,8,9]& ... &[1,2,3;4,5,6;7,8,9]
```

Kodesnutt 2: Eksempel på matrise for ultralyddata i strengformat

På denne måten sørges det for at alle matrisene som hører sammen blir samlet i et sett med rådata. Dette er viktig fordi de fem ultralydbildene som skal produseres vha. disse matrisene hører sammen og er produsert i en og samme scan-syklus med DolphiCam.

2. Komprimering og opplasting

Strengen som blir produsert etter utlesing fra DolphiCam komprimeres før den sendes opp til server. Et eksperiment utført under utvikling viste at gjennomsnittlig størrelse på data før komprimering var 430kB. Opplastningshastigheten var da i gjennomsnitt 230ms.

Siden DolphiCam produserer i underkant av fire bilder per sekund er det essensielt at opplastingsprosessen bruker minst mulig tid slik at streaminga oppleves som i sanntid. Derfor ble det valgt å implementere en komprimeringsalgoritme som reduserer mengden data som lastes opp, og dermed reduserer opplastingstida.

Ved hjelp av biblioteket zlib [3] ble komprimeringsalgoritmen deflate implementert. Den komprimerte versjonen av strengen som skal lastes opp hadde en gjennomsnittlig størrelse på 71kB, ned fra 430kB. Dette resulterte i en gjennomsnittlig opplastingstid på 95ms, ned fra 230ms.

Deflate-algoritmen returnerer komprimert data i binærform. For å håndtere opplasting av binærdata over http må data gjøres om til ASCII-format. Dette gjøres ved å base64 kode [6] data etter at det har blitt komprimert. I kapittel 3.5.4 beskrives beslutningen om å overføre data over http.

Til slutt produseres en streng som inneholder den komprimerte versjonen av matrisene og DolphiCam sitt serienummer. Strengen som sendes opp til server over http vises i eksempel 3.

```
1 | s=123456&data=[base64 encoded string]
```

Kodesnutt 3: Eksempel på strengifisert data som lastes opp til server

For å laste opp strengen til server brukes Libcurl [2].

3. Motta på server

Når server mottar en pakke med ultralyddata verifiseres først serienummeret til DolphiCam. Dersom serienummeret ikke er registrert i database, droppes pakken. Når DolphiCam er verifisert sjekkes det så om det finnes en eksisterende aktiv inspeksjonsøkt med gitt serienummer. Basert på utfallet, utføres en av følgende:

Aktiv inspeksjonsøkt eksisterer Ultralydbilder distribueres til alle tilkoblede klienter i inspeksjonsøkten. Neste seksjon beskriver denne prosesses nærmere. Statusmelding 'OK' returneres så som svar på forespørselen.

Aktiv inspeksjonsøkt eksisterer ikke Ny inspeksjonsøkt opprettes sammen med en tilhørende initiell operatørdeltaker. En engangskode genereres for operatøren og denne returneres som svar på forespørselen.

4. Broadcast

Funksjonen `emitMessageBySessionID()` viser hvordan pakker blir sendt fra node.js-serveren til klienter. Funksjonen tar inspeksjonsøktens unike ID som første parameter. Andre parameter er et navn som brukes for å identifisere pakken og tredje er dataen som skal sendes. Funksjonen finner alle tilkoblede klienter på gitt inspeksjon og sender identiske pakker til hver av disse. Funksjonen bruker `socket.io` modulen sin `emit`-funksjon som håndterer pakkekommunikasjonen mellom node og klienten.

```
1 | /**
2 |  * Emit message to all participants in given session sessionID
3 |  * int sessionID Id of session
4 |  * String name Name of message
5 |  * Object data Data in message
6 |  */
7 | this.emitMessageBySessionID = function(sessionID , name, data){
8 |     var peers = sessionGetPeers(sessionID);
9 |
10 |     for(var i = 0; i < peers.length; i++){
11 |         peers[i].socket.emit(name, data);
12 |     }
13 | }
```


Funksjonen `emitMessageBySessionID()` er en generell funksjon som brukes for å sende pakker til klienter.

5. Motta på klient og dekomprimering

I nettleseren er klientbiblioteket for `socket.io` brukt for å motta pakker med ultralyddata. Data forberedes her på følgende måte.

- (a) Data konverteres fra ASCII til binærform ved bruk av `atob()`-funksjonen, en innebygd javascript funksjon for base64 decoding.
- (b) Den binære data'en dekomprimeres ved bruk av `inflate`-funksjonen i `zlib`.
- (c) Den originale tekst strengen som ble generert på `blackbox` er nå i klartekst og separeres ut til en egen streng, som inneholder rådata for hver type ultralydscan.
- (d) Hvert rådatasett konverteres til JSON [7] og er nå klargjort for rendring.

I sprint 3 ble det gjort optimaliseringer av dekomprimering på klient. Dette beskrives i kapittel 5.4.4.

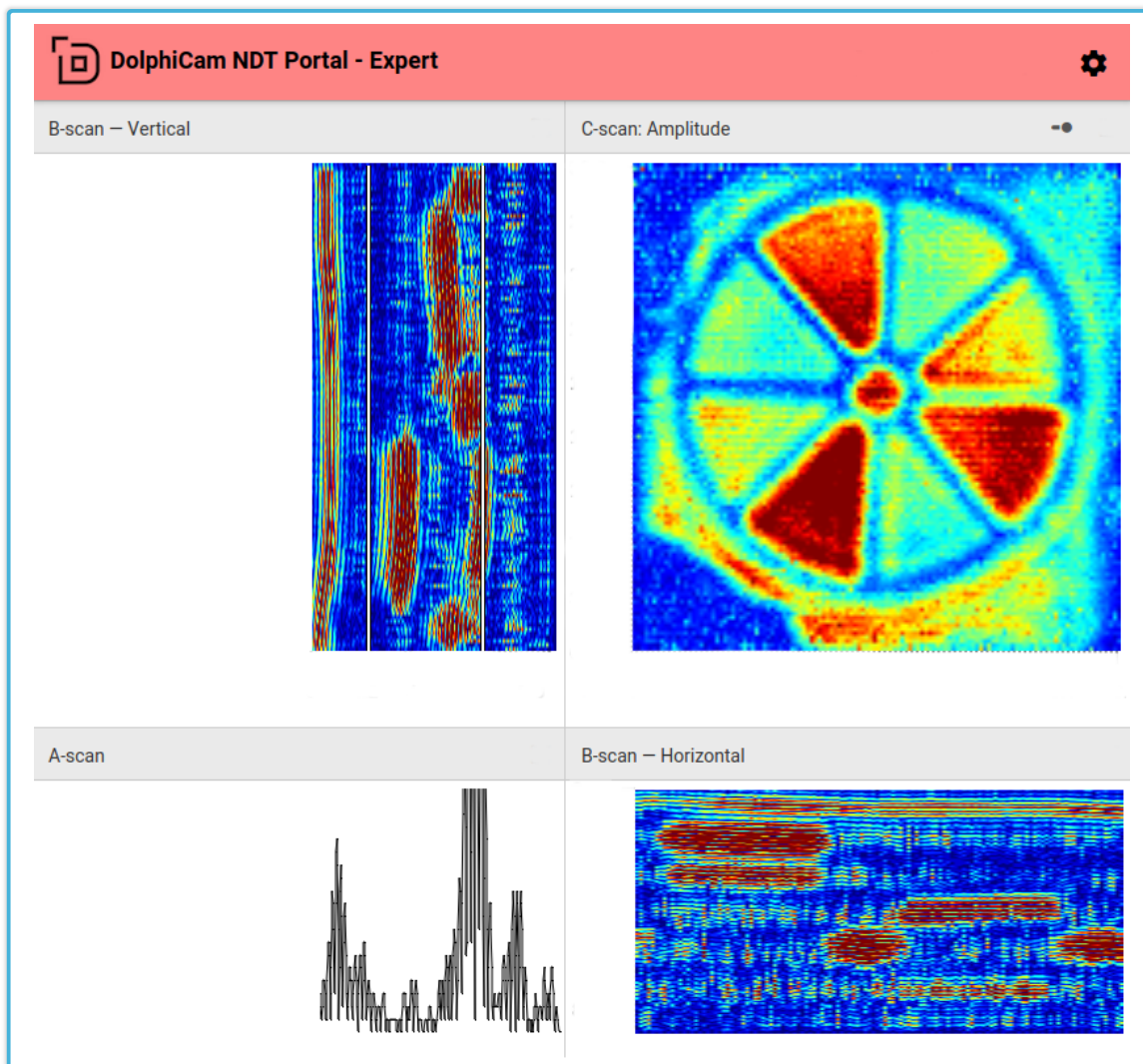
6. Render

Funksjonen `drawScanCTimeOfFlight()` brukes for å generere ultralydbildet for `c-scan TOF`. Dette er en av de fem ulike typene ultralydbilder som `DolphiCam` produserer. Rådata som brukes for å generere dette bilde ligger i den globale variabelen `imageData5`. Funksjonen henter ut en referanse til et `canvas`-element som ligger i `GUI` og genererer et blankt sett med bildedata. Den doble `for-løkke` looper over hver av pikslene i bildet og leser ut tilhørende verdi fra matrisen som inneholder rådata fra `DolphiCam`. Hver av verdiene i matrisene representerer en piksel på bilde, og linkes opp mot en fargekode. Funksjonen `getPaletteRGB2()` returnerer denne fargekoden i `RGB` format. Fargen blir så plassert inn i bildet.

```
1 function drawScanCTimeOfFlight ()
2 {
3     var width  = 124,
4         height = 124,
5         rgb ,
6         pos ;
7
8     var canvas = document.getElementById(" dcImage_2" ) ,
9         ctx    = canvas.getContext("2d" ) ,
10        imgData = ctx.createImageData( width , height ) ;
11
12    for( var y = 0; y < height; y++){
13        for( var x = 0; x < width; x++){
14            rgb = getPaletteRGB2( imageData5 [y] [x] ) ;
15            pos = ((y * width) + x) * 4;
```

```
16         imgData.data[pos] = rgb[0];
17         imgData.data[pos+1] = rgb[1];
18         imgData.data[pos+2] = rgb[2];
19         imgData.data[pos+3] = 255;
20     }
21 }
22
23     ctx.putImageData(imgData, 0, 0);
24 }
```

Figur 11 viser hvordan ultralydbildene representeres i portalen når de har blitt rendra.



Figur 11: Skjermbilde av ultralydbilder i portalen

3.5 Beslutninger

Under oppstart av sprinten ble det gjort flere beslutninger i forbindelse med overordnet infrastruktur og datastrukturer. De viktigste av disse blir diskutert her.

3.5.1 Sentralisert server

Systemets arkitektur krever en form for server som brukere kan koble seg opp mot. Denne serveren kan plasseres lokalt hos operatør eller offentlig i skyen. Oppdragsgiver ønsker en solid tjeneste, og det ble derfor valgt at server sentraliseres. På denne måten kan brukeren forbli tilkoblet selv om operatør skulle miste sin forbindelse til Internett. Dette vil gi bruker en bedre helhetsopplevelse som blir vanskelig å oppnå med en lokalt plassert server som kjører på operatørens maskinvare.

I løpet av utviklingsfasen ble det først forsøkt å plassere server hos en leverandør i London. Målinger gjort viste at en server plassert så langt borte lett kunne føre til ytelsesproblemer. Derfor ble det valgt å flytte server til en leverandør i Bergen, noe som resulterte i at responstid mot server ble redusert fra 45ms til 4ms.

3.5.2 Komprimeringsalgoritme

Vi hadde behov for en algoritme som kunne brukes for komprimering på blackbox og dekomprimering i nettleseren. Siden brukerne av systemet vil analysere bilder, er det helt avgjørende at algoritmen ikke mister data under komprimering. Vi valgte derfor zlib. Zlib er et komprimeringsbibliotek skrevet i C. Det er et mye brukt, godt dokumentert og er tillatt å bruke i kommersielle produkter uten lisens.

3.5.3 Rendering

Bildedata som leses ut fra DolphiCam blir presentert i form av en matrise med verdier. Disse verdiene linkes opp mot en gitt fargekode som blir brukt til å visualisere pixelene i selve bilde. Prosessen med å omgjøre disse dataene til et visuelt bilde, også kalt rendering, kan alternativt skje hvor som helst i systemet. Dersom vi velger å rendere bildet før det blir sendt til klientene vil dette gjøre det vanskelig å muliggjøre forskjellig rendering av bilder for de forskjellige brukerne. Derfor ble det besluttet å sende billedata direkte til webklientene for så å utføre all rendering der. Dette for å muliggjøre bruk av forskjellige fargepaletter for hver deltaker. Fargepaletter ble implementert i sprint 2.

3.5.4 Overføring av billedata

For å overføre bildepakker fra blackbox og opp til server hadde vi forholdsvis tre alternativer:

Alternativ 1 Sette opp en videostream, der bilder sendes som en konstant videostrøm ferdig kodet i feks. MPEG-4 format.

Alternativ 2 Sette opp en UDP/TCP stream, der bildepakker sendes over etter hvert som de er klargjort. Stømmen holdes åpen fra BlackBox kobles opp helt til sesjonen avsluttes.

Alternativ 3 Overføre hver enkelt bildepakke som en separat http(s) forespørsel mot server.

Alternativ 1 vil gjøre det vanskelig å la brukere velge hvordan de ønsker å rendre bildet separat. Siden bildedata kommer inn som en ferdig videostrøm må da denne stømmen manipuleres i sanntid for å kunne f.eks. la brukere velge egne fargepaletter som skal brukes på bilde. Både alternativ 2 og 3 er gode løsninger. Alternativ 3 vil gi mer overhead enn alternativ 2 siden nødvendige headers og metadata må inkluderes i hver forespørsel i alternativ 3. Det ble besluttet å gå for alternativ 3 siden dette gjør overføringen mindre følsom for potensielle nettverksbrudd eller dårlig nettverksforbindelse. Dersom operatør mister sin nettverksforbindelse vil bilder ikke nå frem til server, og dermed trenger ikke server å håndtere disse på noen måte. Klientene har muligheten til å se hvor lang tid det er siden siste bildeoppdatert, og de kan dermed ta hensyn til et eventuelt brudd i forbindelsen.

3.5.5 Engangskode

Hvordan engangskodene som deltakere bruker for å logge seg inn produseres ble diskutert som en del av denne sprinten. Det ble valgt å implementere disse ved å generere tilfeldige tall med seks siffer. Siden en engangskode bare kan brukes en gang er det lite trolig at dette vil skape et problem i løsningen selv om den skulle settes i produksjon. Lengden på engangskoden kan ved behov økes uten større endringer i systemet.

3.5.6 Eksperiment: Lagring av inspeksjon

En funksjonalitet som DolphiTech kan se for seg er å lagre en komplett inspeksjon til senere vurdering eller historik. Vi gjorde derfor et eksperiment der vi lagra alle bildesett som kommer fra DolpихCam i database, slik at strømmen med bilder hentes frem i etterkant av inspeksjonen. Resultatet viste at en inspeksjon med varighet på et minutt produserte 230 bildesett som brukte 13MB lagringsplass i database. Det ble konkludert at løsningene valgt i denne sprinten ville muliggjøre en slik implementasjon i en fremtidig sprint. Om nødvendig kan eventuelt antallet bildesett som lagres i database tidsbegrenses.

3.6 Konklusjon

Sprint 1 har vært mest krevende sprinten. Dette fordi mangel på initiale viktige valg ble tatt og mye nytt å sette seg inn i ble en del av sprinten. Det ble ikke gjort estimeringer under oppstart av

sprinten og user stories som ble satt opp var åpenbart alt for generelle til at de ble funksjonelle under utviklingsprosessen. Dette ble ikke et problem siden gruppen viste hva som var målet med sprinten og kom raskt i gang med å få det implementert.

Det ble brukt noe lengre tid på å forstå hvordan b-scan matrisene kunne traverseres for å produsere riktig representasjon av ultralydbildene. Dette ble løst etterhvert i samarbeid med DolphiTech. Oppsummert fikk gruppen implementert planlagt funksjonalitet til den grad at det fungerte bra nok til å tilfredstille selve user storiene. DolphiTech er fornøyd med gjennomførelsen og det som har blitt gjort denne sprinten.

4 SPRINT 2

4.1 Innledning

I denne sprinten ble det lagt størst fokus på å implementere muligheten for å endre på DolphiCam sine innstillinger via klienten sitt grensesnitt. Samtidig skal muligheten for å invitere flere deltakere til en inspeksjonsøkt implementeres. Disse skal få tildelt roller som gjest eller ekspert.

4.2 User Stories

Tabell 5: User stories sprint 2

#	User story	Estimert (timer)	Brukt (timer)
5	Som operatør vil jeg invitere eksperter/gjester til inspeksjonen	8	12
9	Som operatør/ekspert vil jeg sette DC-innstillingen B-scan horisontal og vertical linje i C-scan	10	16
12	Som operatør/ekspert vil jeg sette DC-innstillingen upper og lower gate	10	6
14	Som operatør/ekspert vil jeg sette DC-innstillingen gain/forsterkning	10	6
15	Som operatør/ekspert vil jeg sette DC-innstillingen pulstype	10	5
16	Som operatør/ekspert vil jeg sette DC-innstillingen TX-shoot	10	4
17	Som deltager vil jeg endre fargepallett	6	6
18	Som deltaker skal jeg ha forhåndsbestemte tilganger	15	8
		79	63

4.3 Kravspesifikasjon

4.3.1 Endre innstillinger i DolphiCam

Når DolphiCam scanner er det flere forskjellige parametre som bestemmer hvordan apparatet skal oppføre seg eller behandle signalene som leses inn. Disse parametrene refereres til som innstillinger. Som en del av denne sprinten skal det muligjgjøres å fjernstyre disse innstillingene via portalen slik at en ekspert som følger inspeksjonen kan tilpasse disse etter behov uten at operatøren trenger å ha kjennskap til disse.

Parametrene som skal settes i DolphiCam er implementert i DolphiLib og kan dermed settes direkte vha. et funksjonskall mot biblioteket. Det er derfor ikke nødvendig å beskrive i detalj hva disse parameterene gjør i praksis da det ikke inngår i oppgaven.

For å muliggjøre endring av innstillinger skal det etableres en løsning der en ekspert via portalen kan åpne et grensesnitt der en oversikt over gjeldende parametre vises. Eksperten skal så kunne endre verdien til hver av innstillingene. Disse verdien er som oftest et tall eller en av/på funksjon. Når en verdi endres skal den kommuniseres opp til server, der en komplett liste over alle innstillinger lagres. Programvaren på blackbox skal sørge for å hente ned innstillinger fra server og sette disse i DolphiCam. Samtidig må endring av innstillinger gjort av en deltaker reflekteres i grensesnittet til alle andre deltakere.

4.3.2 Invitere deltaker

I sprint 1 ble det beskrevet hvordan operatøren kan logge seg inn i portalen etter å ha opprettet en ny inspeksjonsøkt. Videre i denne sprinten skal det muliggjøres ovenfor operatøren å invitere flere deltakere til å bli med på inspeksjonen. Operatøren skal kunne velge et navn som gis til denne inviterte deltakeren for så å bli presentert med en engangskode som han kan formidle videre til den som blir invitert. Hvordan operatøren formidler denne engangskoden skal det er ikke bli tatt hensyn til. Den inviterte deltakeren skal kunne gå til portalens URL og skrive inn sin engangskode og dermed få tilgang til inspeksjonsøkten.

4.3.3 Endre fargepalett

Det ble i sprint 1 beskrevet hvordan forskjellige fargepaletter kan brukes til å presentere rådata fra DolphiCam som et bilde. I portalen skal det være mulig for de ulike deltakerene å velge hvordan fargepalett de ønsker å bruke for presentasjon av bildene. I noen tilfeller er enkelte paletter mer beskrivende en andre, samtidig som personlige preferanser blant deltakerene gjerne er en faktor. Det skal også skilles mellom pallett på C scan time of flight og de andre bildene.

Det skal implementeres en rekke fargepaletter som deltakerene kan velge mellom. Disse palettene har blitt utlevert fra DolphiTech og skal implementeres på en slik måte at hver deltaker kan velge fargepalett individuelt.

4.3.4 Tilganger

I kapittel 4.3.2 er muligheten til å invitere deltakere beskrevet. Deltakerene skal i utgangspunktet ha tilgang til all funksjonalitet som inkluderes i portalen, dette med unntak av enkeltfunksjoner som krever en spesiell rolle. Tabell 6 viser en oversikt over de ulike rollene og tilgangsnivået de får for de ulike funksjonene.

Tabell 6: Tilgangsmatrise for ulike deltaker-roller i portalen

	Operatør	Ekspert	Gjest
Inviter deltagere	Ja	Nei	Nei
Endre innstillinger	Ja	Nei	Nei

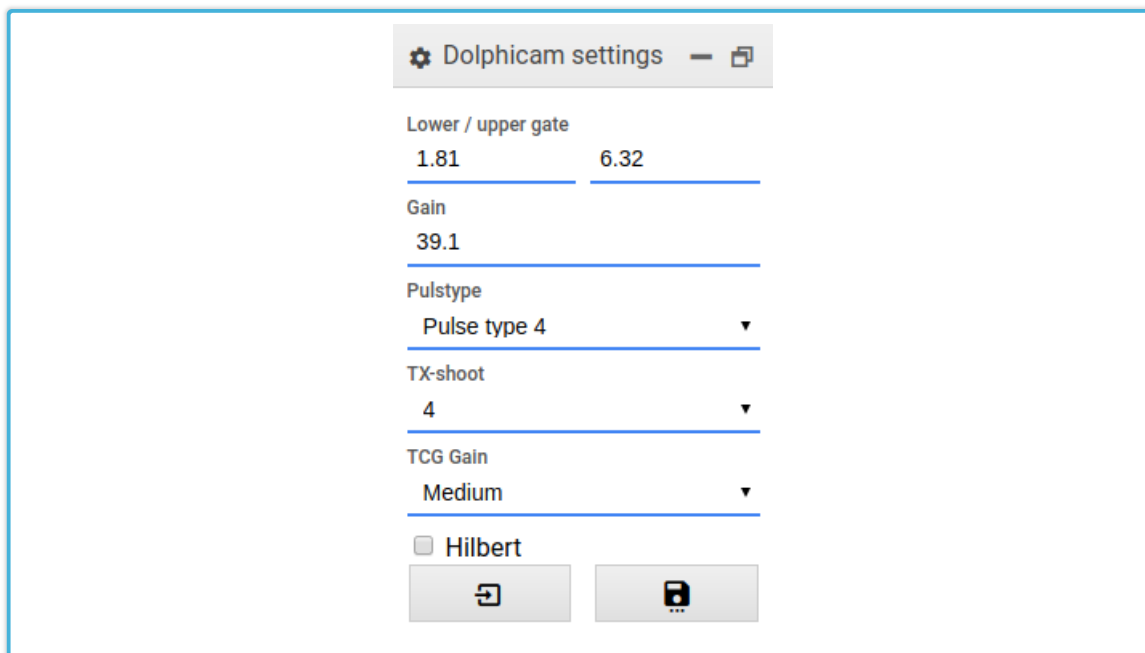
4.4 Implementasjon

4.4.1 Endring av innstilling

Endring av innstilling fra portalen er delt opp i to deler; mellom klient-server og server-blackbox.

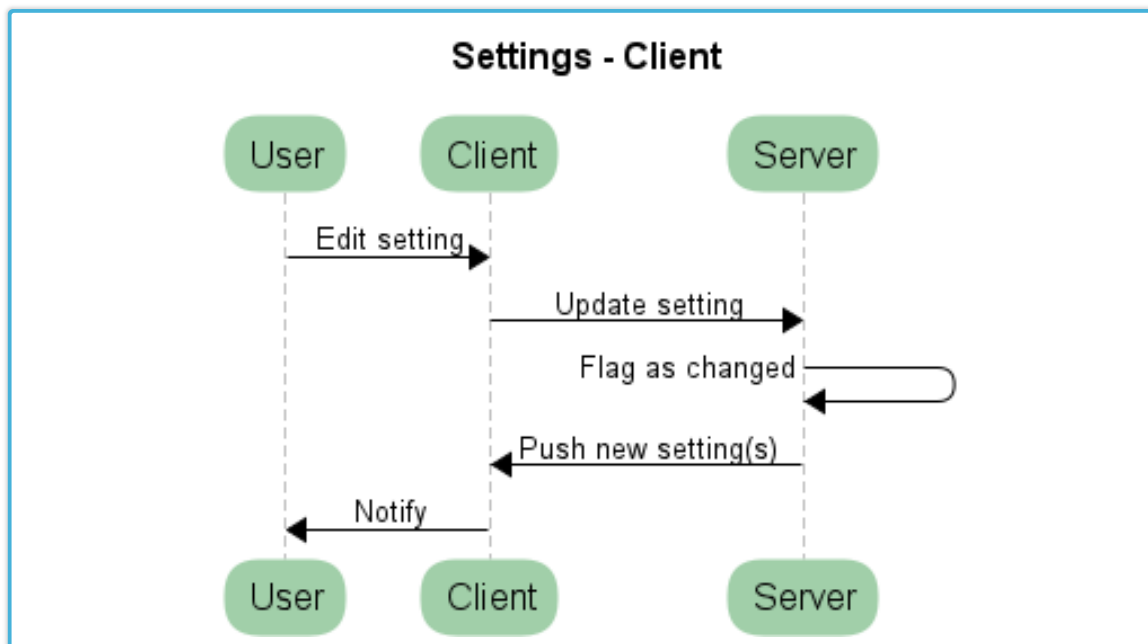
Klient til server

Alle deltager kan se hvilke innstillinger som er satt, men kun operatør og ekspert har mulighet til å endre innstillinger. Skjerm bilde av implementerte innstillinger vises i figur 12. Legg merke til at figuren også viser innstillinger som ble implementert i senere sprinter.



Figur 12: Skjerm bilde av implementerte innstillinger

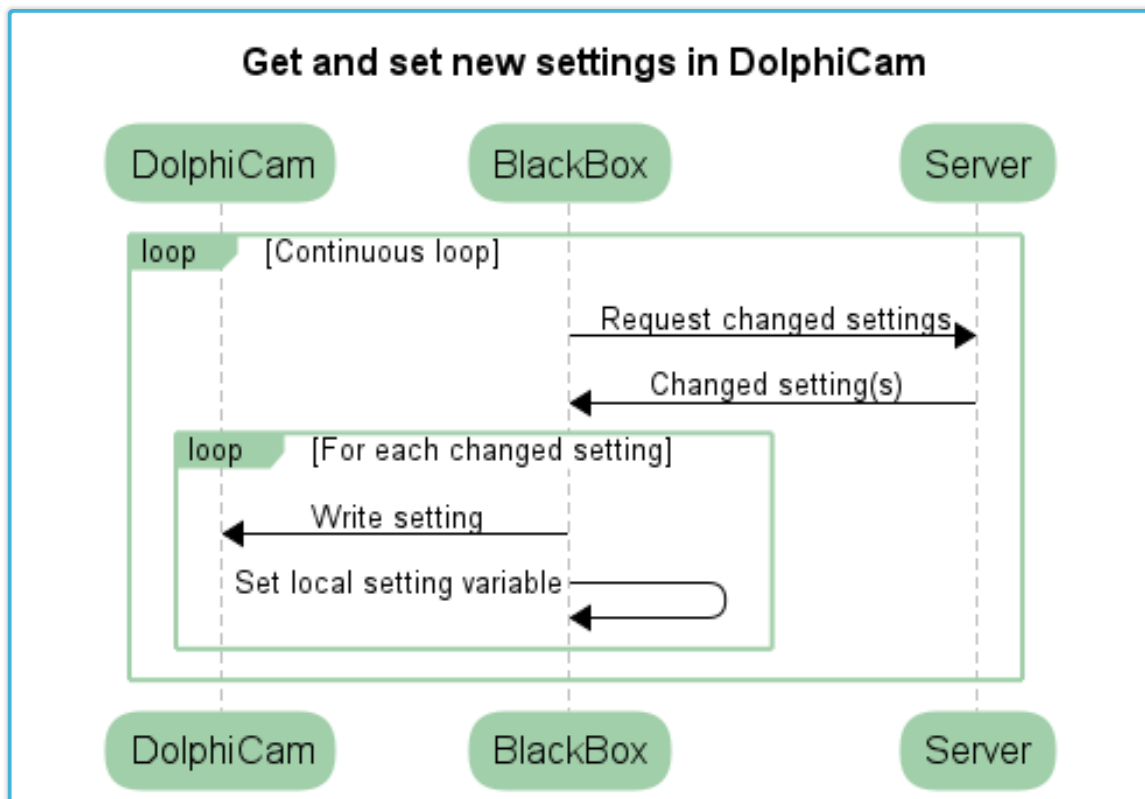
Når en operatør eller ekspert endrer en innstilling i portalen verifiseres det først at brukeren har tilstrekkelig tilgang. Deretter sendes det en melding til server med informasjon om hvilken innstilling som har blitt endret, og hva slags verdi den er endret til. Server sender så ut en pakke til alle andre tilkoblede klienter med en komplett liste over alle innstillinger. Dette er for å holde alle deltakere oppdatert med gjeldende innstillinger. På server registreres endringen på inspeksjonsøkten den gjelder for. Sammen med innstillingens verdi settes det også et flagg som indikerer at innstillingen har blitt endret på. Figur 13 illustrerer hvordan denne prosessen foregår.



Figur 13: Pushing av innstilling fra portal til server

Server til blackbox

I programvaren på blackbox startes en ny tråd som har som oppgave å hente oppdaterte innstillinger fra server. Denne tråden sender forespørsler om oppdaterte innstillinger til server med en frekvens på et sekund. Server henter alle innstillinger som har blitt flagget som endret og returnerer disse. Dette gjør at blackbox mottar kun innstillinger som faktisk har blitt endret siden siste forespørsel. Dersom ingen innstillinger har blitt endret vil blackbox motta en tom liste med innstillinger. Server vil også tilbake stille flagget som sier at innstillingen har blitt endret etter at listen har blitt returnert. Blackbox vil så skrive mottatte innstillinger til DolphiCam. Figur 14 viser sekvensene som inngår i denne prosessen.



Figur 14: Oppdater innstilling fra server i blackbox

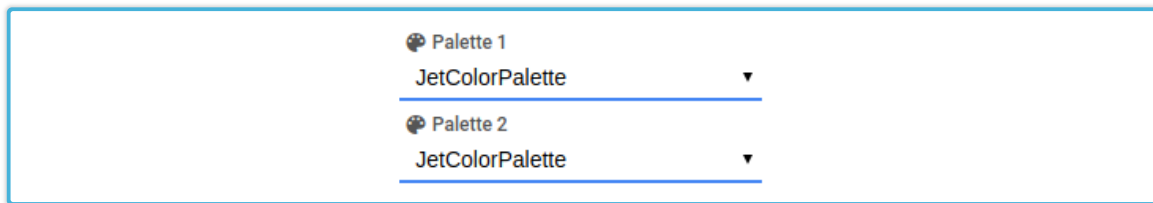
4.4.2 Invitere deltagere

I portalen er det lagt inn et valg for å invitere deltagere. Deltakere med rollen operatør og ekspert har mulighet til å invitere gjester og/eller eksperter. Invitasjonen skjer ved at deltakeren velger et visningsnavn for den inviterte brukeren. Det blir så generert en engangskode på server og laget en bruker. Den genererte engangskoden kan så formidles til inviterte deltakeren. Den inviterte kan da logge inn i portalen med denne koden.

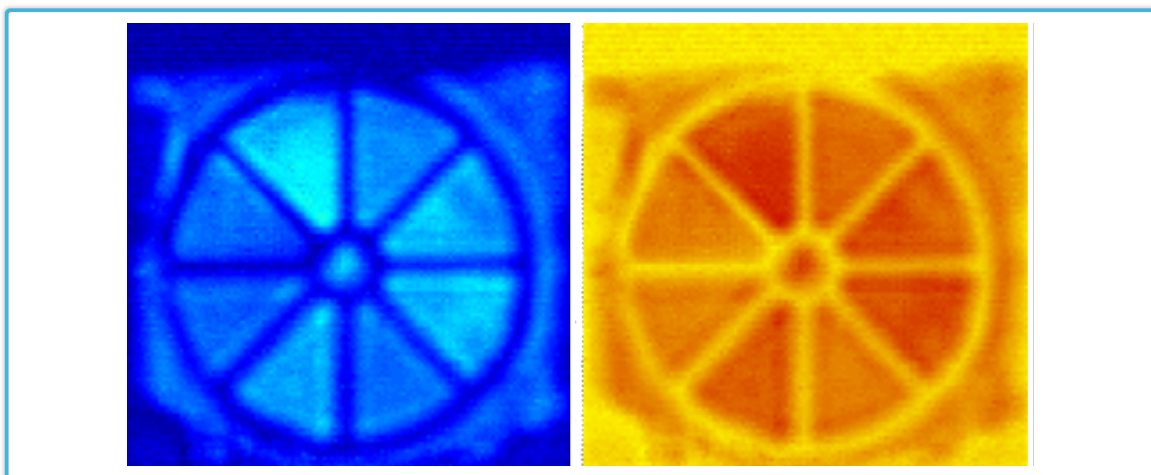
4.4.3 Fargepalett

Det er to forskjellige palettinnstillinger i portalen. Hver av disse påvirker forskjellige ultralydbilder. Begge har en forhåndsdefinert liste med paletter som kan velges, dette vises i figur 15. Når en fargepalett endres vil de bildene som påvirkes oppdateres med det nye fargevalget. Figur 16 viser

hvordan et identisk c-scan ser ut med to forskjellige fargepaletter valgt.



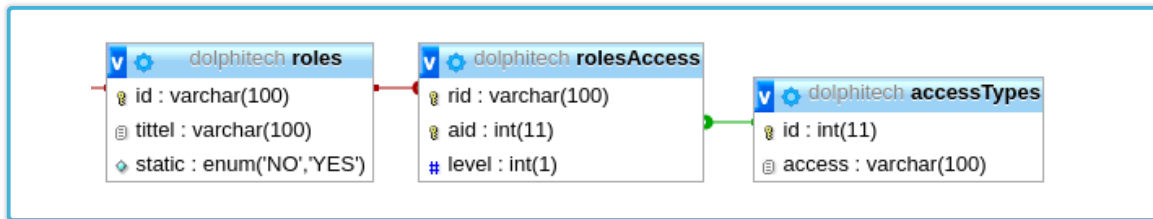
Figur 15: Skjerm bilde som viser palettvalg



Figur 16: Skjerm bilde som viser forskjellige fargepaletter

4.4.4 Tilganger

Funksjonalitet som krever forhåndsbestemte tilganger er registrert inn i database i en egen tabell som holder på informasjon om hvilken roller som har de ulike tilgangene. Figur 17 viser hvordan tabellene i database er satt opp for å håndtere tilganger.

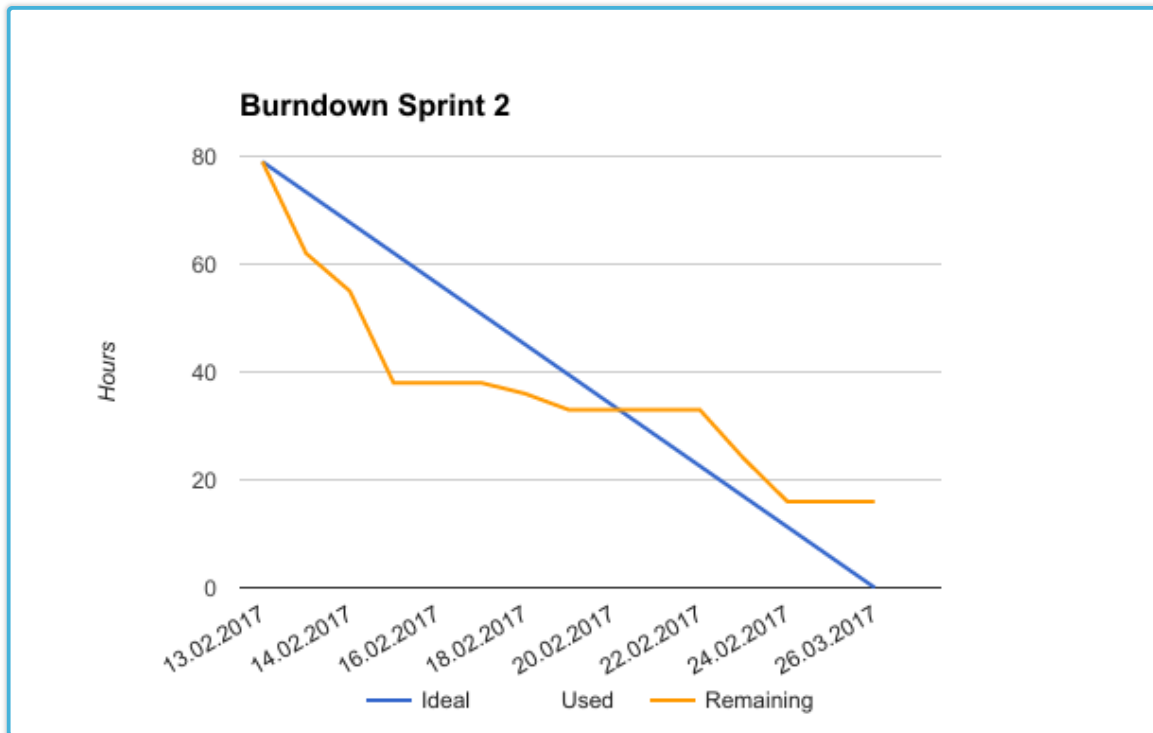


Figur 17: Databasediagram for tilganger

I roles tabellen ligger alle de ulike rollene, mens accesstypes tabellen beskriver hvilken funksjonalitet som krever godkjent tilgang. rolesaccess tabellen er et mellomledd som gjør det mulig for en rolle å ha flere tilganger. Denne type tabelloppsett skalerer bra for nye tilganger som kan legges inn i fremtiden. Når en bruker forsøker å utføre en operasjon som krever en spesiell tilgang, blir det sjekket mot disse tabellene om brukerens rolle har tilstrekkelige rettigheter.

4.5 Konklusjon

Implementasjon av DolphiCam-innstillinger fra portalen ble løst på en bra måte. Innstillinger blir oppdatert i sanntid mellom klientene og endringer på ultralydbildene forekommer få sekunder etter endring. Oppdragsgiver synes dette fungerte bra. Estimeringene gjort for sprinten viste seg og stemme bra med brukt tid. Sprintens hovedmål ble nådd uten problematikk under utvikling.



Figur 18: Burndown diagram for sprint 2

5 SPRINT 3

5.1 Innledning

Her ble det lagt størst fokus på deling av webkamera og mikrofon, både portrettbilde og oversiktsbilde. Vi vil også se litt nærmere på optimalisering av ytelse i klienten og blackbox.

5.2 User Stories

Tabell 7: User stories sprint 3

#	User story	Estimert (timer)	Brukt (timer)
6	Som deltaker vil jeg sende/motta direktemeldinger	4	4
7	Som deltaker ønsker vil jeg dele mitt webkamera	45	43
19	Som operatør vil jeg dele flere webkameraer	10	9
20	Som deltaker vil jeg snakke med andre deltakere med mikrofon	3	6
21	Som deltaker ønsker jeg bedre ytelse i portalen	20	10
		82	72

5.3 Kravspesifikasjon

5.3.1 Direktemeldinger

Portalen skal være en kallaborativ plattform der deltakere samarbeider om inspeksjonen. En av funksjonene som er nødvendige for å tilfredstille dette kravet er muligheten til å sende direktemeldinger mellom deltakerene. Direktemeldingene skal være i form av tekst og skal sendes til alle deltakere i inspeksjonen.

5.3.2 Kommunikasjon over video og lyd

Deltakerene av en inspeksjon skal ha muligheten til å kommunisere vha. webkamera og mikrofon. Derfor skal det implementeres en slags videokonferanseløsning der disse funksjonene bygges inn portalen. Alle deltakere skal ha muligheten til å dele sitt webkamera og mikrofon. Videostrømmen

fra disse webkameraene skal vises i portalens grensesnitt. Oppdragsgiver ønsker også en fullintegret løsning som baseres på moderne webteknologier.

I tillegg skal operatøren ha muligheten til å dele et såkalt oversiktskamera. Dette er et kamera som kun operatøren har muligheten til å koble opp, og skal vises i en sentral plassering i portalens grensesnitt. Oversiktskameraet er ment for å gi deltakerene en oversikt over prøveobjektet som inspiseres.

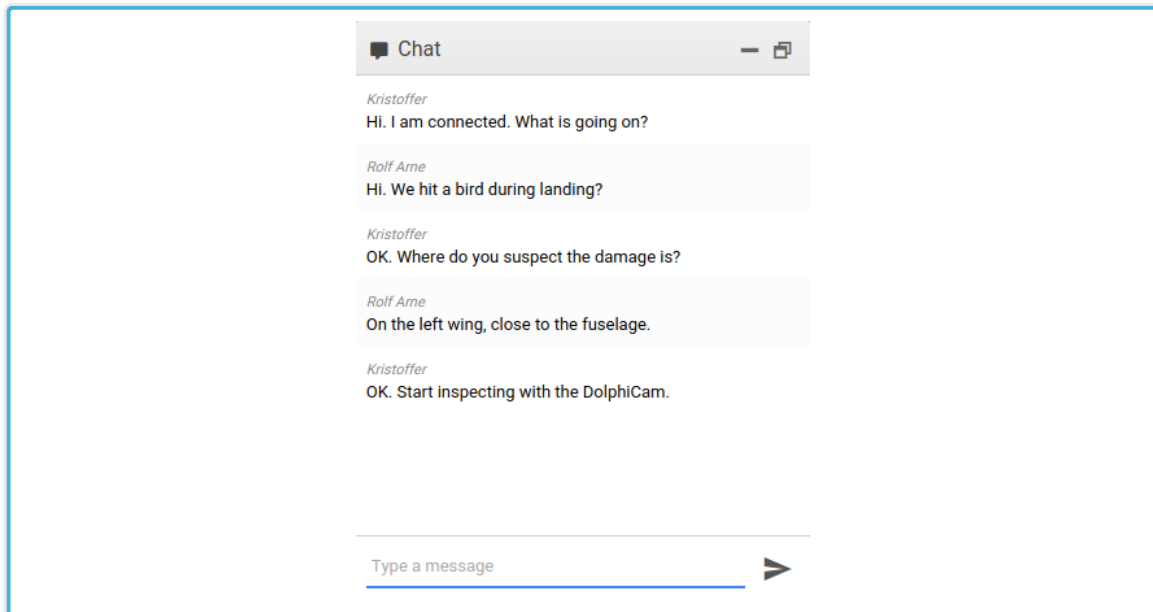
5.3.3 Forbedret ytelse i portalen

Etter de to første sprintene ble det observert at rendering av bilder i portalen krever i overkant mye resusser fra deltakerens datamaskin. Av denne grunn oppstod det et krav om å optimalisere hvordan rådata fra DolphiCam blir behandlet i webklientene.

5.4 Implementasjon

5.4.1 Direktemeldinger

Muligheten for to-veis kommunikasjon mellom klient og server ble implementert som en del av sprint 1. Derfor er funksjonalitet som er nødvendig for å bygge en slik funksjon allerede etablert. Et grensesnitt for å sende og vise direktemeldinger har blitt implementert som vist i figur 19.



Figur 19: Skjerm bilde av chat

Når en deltaker sender en melding blir meldingens innhold sendt til server. Server vil så distribuere meldingen til alle klienter tilkoblet den aktuelle inspeksjonsøkten sammen med avsenders visningsnavn. Klientene oppdaterer sitt GUI med den nye meldingen.

5.4.2 Sikker tilkobling

Moderne nettlesere krever at nettsiden bruker en sikker tilkobling for å be om tilgang til webkamera og mikrofon. Derfor er det konfigurert bruk av https mellom alle klientene og server. Sertifikatautoriteten Let's Encrypt [8] blitt brukt som leverandør av SSL-sertifikat. Let's Encrypt utsender gratis sertifikat, men krever at sertifikat fornyes hver tredje måned. Siden systemet er under utvikling skaper dette ikke noe problemer. Samtidig kan sertifikatet lett erstattes med et annet ved behov. Etter oppsett av SSL-sertifikat ble også kommunikasjon mellom blackbox og server endra fra HTTP til HTTPS.

5.4.3 Kommunikasjon over video og lyd

Som oppdragsgiver beskriver i den initielle oppgaveteksten skal WebRTC brukes for å implementere sanntidskommunikasjon. WebRTC [10] er en protokoll som muliggjør såkalt peer-to-peer [12] kom-

munikasjon mellom klienter direkte i nettleseren uten noen form for tillegg. Teknologien egner seg derfor bra til å strøkke video og lyd mellom klienter, da den typen bruk er i stor del innebygd i protokollen. Samtidig vil bruk av WebRTC gjøre at det ikke blir nødvendig å strøkke videodata fra webkamera via serveren, noe som reduserer ressursbruket på server.

For å implementere muligheten til å dele webkamera og lyd må først nettleseren få tilgang til deltakerens webkamera og mikrofon. Dette gjøres ved bruk av Javascript via `getUserMedia()`-funksjonen. Funksjonen lar oss spesifisere en såkalt constraint der det defineres hva slags media som er ønskelig fra deltakerens datamaskin, og ber brukeren om å godta tilgang til disse. I kodesnutt 4 vises eksempel på en slik constraint. Koden i eksempel 5 viser hvordan brukeren blir bedt om tilgang til webkamera og mikrofon.

```
1 /**
2  * Constraints asking for both audio and video
3  * Ideal, minimum and max values for video are specified
4  */
5 var constraints = {
6   audio: true,
7   video: {
8     width: { min: 100, ideal: 480, max: 480 },
9     height: { min: 100, ideal: 360, max: 360 },
10    frameRate: { ideal: 15, max: 20 }
11  }
12 }
```

Kodesnutt 4: Eksempel på constraint

```
1 // Check if browser supports getUserMedia
2 if(navigator.mediaDevices.getUserMedia)
3 {
4   navigator.mediaDevices.getUserMedia(constraints).then(function(
5     stream){
6     localStream = stream;
7     // localStream now contains a stream
8     // connected to the local webcam
9     webCamReady();
10  }).catch(function(err){
11    // An error occurred
12  });
```

```
12 | }
```

Kodesnutt 5: Eksempel på bruk av getUserMedia()

Når en stream bestående av deltakerens webkamera og mikrofon er klargjort på klienten kan denne kommuniseres til de andre deltakerene i portalen. Siden støtte for WebRTC er forholdsvis begrenset og implementert på forskjellige måter mellom de ulike nettleserene ble det valgt å bruke et bibliotek for å håndtere dette. PeerJS [11] er et bibliotek som abstraherer WebRTC-laget og lar en ringe opp en annen klient med sin lokale stream. Den andre klienten kan da svare med sin lokale stream, og dermed etableres ei toveis utveksling av webkamera og lyd mellom klientene. Når den lokale streamen er klar ringer hver klient opp alle andre klienter. I eksempel 6 vises det hvordan PeerJS brukes for å ringe opp en annen klient med sin lokale stream. Eksempel 7 vises det hvordan en svarer på oppringinger fra en annen klient.

```
1 | var call = peer.call(remoteID, localStream);
2 | call.on('stream', function(remoteStream){
3 |     // Create new video-container in GUI and
4 |     // connect remoteStream to it
5 | });
6 | call.on('close', function(){
7 |     // Remove the video-container from GUI
8 | });
```

Kodesnutt 6: Eksempel på bruk av PeerJS

```
1 | function webCamReady(){
2 |     // Connect and enable beeing a peer:
3 |     peer = new Peer(userID, peerProperties);
4 |
5 |     // Answer incomming calls from other peers:
6 |     peer.on('call', function(call){
7 |         call.answer(localStream);
8 |         call.on('stream', function(remoteStream){
9 |             // Create new video-container in GUI and
10 |             // connect remoteStream to it
11 |         });
12 |     call.on('close', function(){
13 |         // Remove the video-container from GUI
14 |     });
15 | });
```

Kodesnutt 7: Eksempel på bruk av PeerJS

Videre settes streamen inn i video-elementer i GUI slik som kommentert i kodeeksempel 6 og 7. Siden HTML5 video-elementer har innebygd støtte for dette kan strømmen knyttes direkte til video-elementer som vist i eksempel 8.

```
1 // variable stream is either a local or remote stream
2 var webcamElement = document.getElementById( 'webcamElement' );
3 webcamElement.srcObject = stream;
```

Kodesnutt 8: Eksempel på tilknytning av mediastream til video-element

5.4.4 Ytelse klient

Som en del av user story #21 er det ønskelig å forbedre ytelsen i webklienten. Flere krevende prosesser som rendring av bilder, dekomprimering og visning av webkamera skjer i nettleseren kontinuerlig. For en best mulig brukeropplevelse bør funksjonaliteten optimaliseres. Visning av webkamera skjer for det meste under panseret i nettleseren. Derfor ble det satt fokus på optimalisering av hvordan ultralyddata behandles i nettleseren.

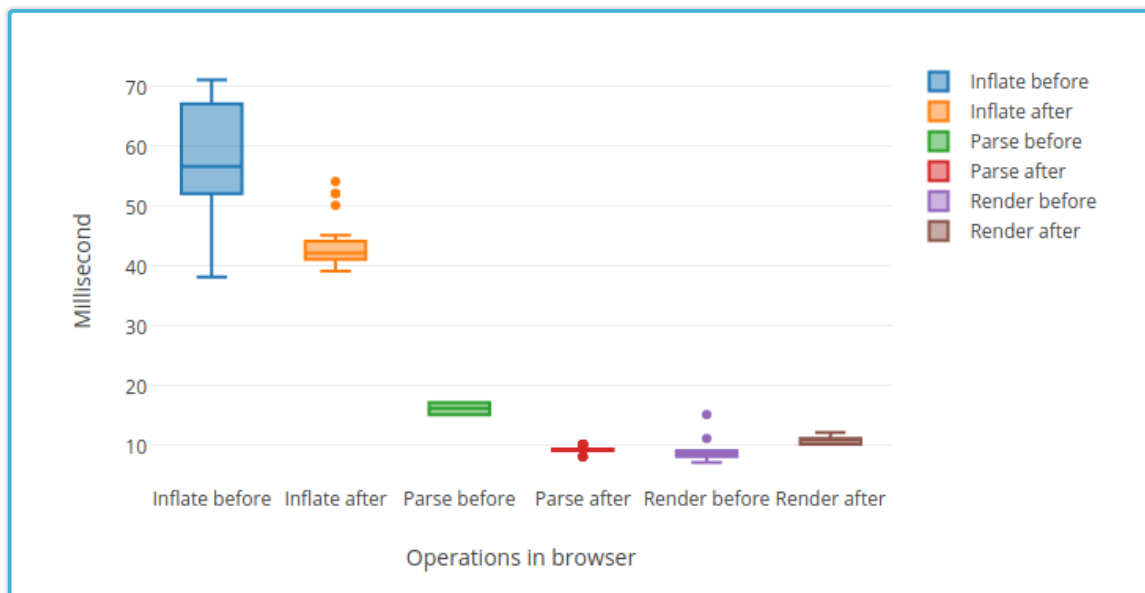
For å måle ytelsen har funksjonen med å renderere bildene på klient blitt delt opp i forskjellige logiske operasjoner. Tiden disse bruker har blitt registrert hver for seg. Prosessen med å behandle ny ultralyddata er delt opp i følgende operasjoner:

Inflate Dekomprimering

Parse Segmentering og klargjøring

Render Generering av ultralydbilder

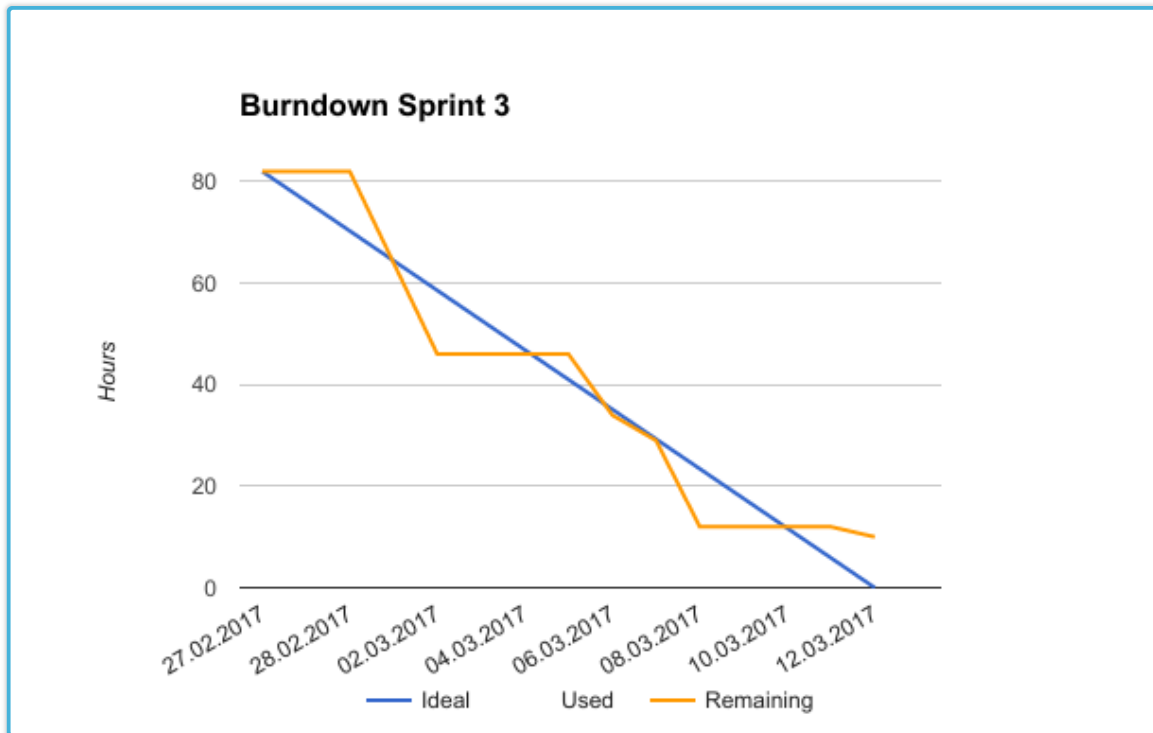
Basert på tidsbruk i de ulike prosessene kom det frem at dekomprimeringsprosessen tok lengst tid. Ved nærmere undersøkelse av inflate-funksjonen ble det observert at en spesifikk del utgjorde rundt 80 prosent av tidsbruken. Funksjonen som utgjorde denne tidmengden ble brukt for å omforme dekomprimert data til en streng. Denne funksjonen ble erstattet med en mer optimalisert tilsvarende funksjon noe som resulterte i en reduksjon på 25%. Tiden bruk på operasjonen parse ble halvert etter refaktorering av datastrukturen brukt for å holde ultralyddata i nettleserens minne under behandling. Figur 20 viser resultatet av disse optimaliseringene.



Figur 20: Gjennomsnittlig tidsforbrukt i nettleser før og etter optimalisering

5.5 Konklusjon

Under sprinten fikk gruppen implementert all funksjonalitet som var planlagt. Dette resulterte i et produkt som tilfredstilte kravspesifikasjonene for sprinten. Samtidig ble det klart at det fortsatt var mye som kunne gjøres for å forbedre brukeropplevelsen. Under innlogging i portalen burde det sjekkes om nettleseren har støtte for nødvendige teknologier for å unngå at portalen henger seg opp for de som logger inn med en gammel nettleser. Oppdragsgiver var fornøyd med fremdriften som ble gjort. Estimeringene for denne sprinten viste seg å stemme bra med brukt tid. Noe av grunnen til dette var fordi utviklerene valgte å bruke ekstra tid på user story #7. Optimalisering av implementert funksjonalitet samt refaktoring av tidligere implementert kode ble gjort som en del av denne historien. Figur 21 viser burndown diagram for sprinten.



Figur 21: Burndown diagram for sprint 3

6 SPRINT 4

6.1 Innledning

Hovedfokuset i denne sprinten ble lagt på å tilpasse eksisterende funksjonalitet. Justering av hvordan tildeling av tilganger i portalen fungerer og et mer fleksibelt grafisk grensensitt ble satt i fokus. Samtidig ønskes det funksjonalitet for å se på ultralydbilder tilbake i tid.

6.2 User Stories

Tabell 8: User stories sprint 4

#	User story	Estimert (timer)	Brukt (timer)
22	Som ekspert/"operatør-ekspert" vil jeg være eneste som kan endre på DolphiCam innstillinger.	2	6
23	Som operatør vil jeg starte ny sesjon med midlertidige ekspertretigheter.	4	4
24	Som ekspert vil jeg ha mulighet til å bytte rolle med gjest.	4	4
25	Som deltaker vil jeg ha mulighet til å kollapse webcams.	2	3
26	Som deltaker vil jeg lagre "current timeline" av DolphiCam bilder.	10	12
27	Som deltaker vil jeg se på en saved timeline".	20	17
28	Som deltaker ønsker jeg å vite hvem som lager lyd eller prater.	10	2
29	Som "operatør-ekspert" vil jeg velge en initiell ekspert.	4	2
30	Som deltaker vil jeg ha eget panel for scans / webcams.	5	4
31	Som deltaker vil jeg poppe"ut webcams og scans til justerbart vindu.	18	12
		79	66

6.3 Kravspesifikasjon

6.3.1 Tilganger

I sprint 2 ble det satt opp en tilgangsmatrise som beskrevet i kapittel 4.3.4. Oppdragsgiver ønsker nå å omstrukturere hvordan de ulike rollene fordeles og tilgangene de har.

Ekspertrollen skal gjøres unik, slik at bare en deltaker kan fungere som ekspert til etthvert tidspunkt.

Eksperten skal ha mulighet til å gi fra seg ekspertrollen til en annen deltaker, og i prosessen bli satt som gjest. På denne måten kan flere deltakere ta tur på å kontrollere inspeksjonen. Disse henger tett sammen med kravet om at kun eksperten skal kunne endre på DolphiCam sine innstillinger.

Når en ny inspeksjon blir satt opp er det operatøren som blir den første deltakeren. Dette gjør at det blir nødvendig å gi operatøren en midlertidig ekspertrolle frem til den første eksperten er valgt. På denne måten kan operatøren styre inspeksjonen før deltakerene er invitert og koblet opp.

6.3.2 GUI

Portalens grafiske grensesnitt har frem til denne sprinten ikke blitt satt i fokus. Utviklerene har fått stor frihet til å implementere denne slik som de følte passet. Etter tilbakemeldinger fra oppdragsgiver har noen krav om mer fleksibilitet i grensesnittet oppstått. Det er ønskelig å muliggjøre frigjøring av ultralydbilder og webkamera fra layout slik at de kan flyttes rundt på skjermen. Samtidig skal visse visningene kunne minimeres.

6.3.3 Tidslinje

Ultralydbildene som vises i portalen oppdateres i sanntid der tidligere bilder forsvinner etter hvert som nye kommer inn. Det skal implementeres en løsning som gjør det mulig for eksperter å produsere en såkalt tidslinje. Når eksperten velger å lage en tidslinje skal et bestemt antall av de siste bildene produsert av DolphiCam permanent lagres samlet i en tidslinje. Sammen med bildene skal også metadata for hvert bilde lagres. Metadata inkluderer innstillingene som var satt i DolphiCam på tidspunktet bildet ble generert sammen med tidsstempel. Etter at en tidslinje er produsert skal deltakerene ha muligheten til å navigere seg gjennom bildene som ligger i denne.

6.4 Implementasjon

6.4.1 Tilganger

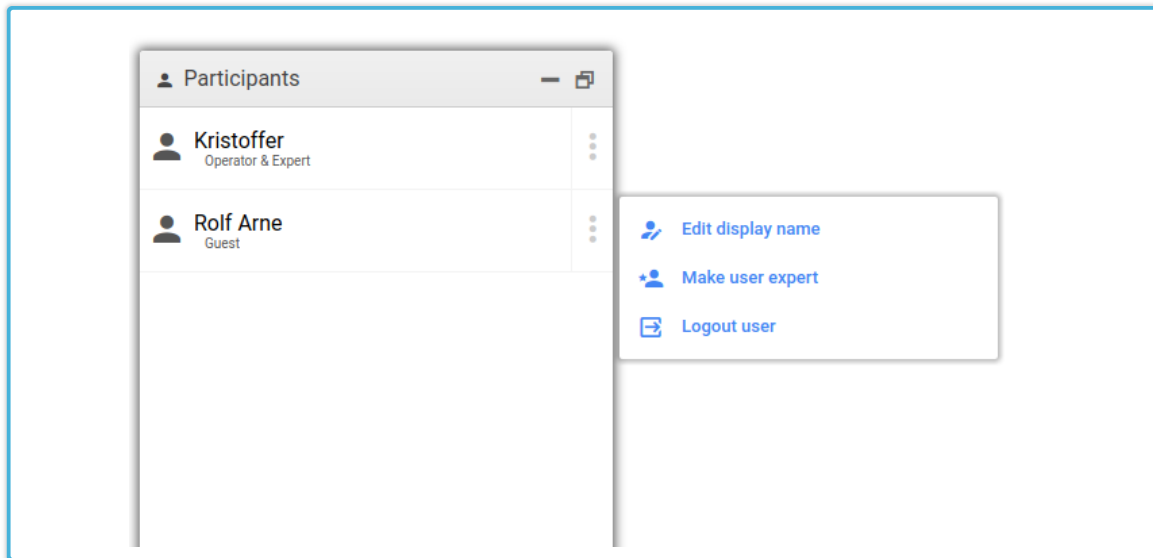
Operatøren er alltid første bruker i en inspeksjonsøkt og vil automatisk få tildelt rollen som operatør. Dette har blitt endret til at operatøren nå får rollen som operatør-ekspert. Denne rollen gir alle tilganger som operatør- og ekspertrollene har til sammen. Siden det nå skal kun være en ekspert i systemet om gangen har operatørrollen også mistet noen av sine tilganger. I tabell 6 ble første versjon av tilgangsmatrisen vist. Matrisen er nå endret til slik som vist i tabell 9.

Tabell 9: Tilgangsmatrise for ulike deltaker-roller i portalen

	Operatør-ekspert	Operatør	Ekspert	Gjest
Inviter deltagere	Ja	Nei	Ja	Nei
Endre innstillinger	Ja	Nei	Ja	Nei
Endre visningsnavn	Ja	Nei	Ja	Nei
Fjern bruker	Ja	Nei	Ja	Nei
Vise oversiktskamera	Ja	Ja	Nei	Nei
Gi fra eksperttilgang	Ja	Nei	Ja	Nei
Gi seg selv eksperttilgang	Ja	Ja	Nei	Nei
Ta snapshot	Ja	Nei	Ja	Nei
Start timeline	Ja	Nei	Ja	Nei
Vise timeline	Ja	Ja	Ja	Ja
Avslutt inspeksjon	Ja	Ja	Nei	Nei

Deltakeren med operatør-ekspert eller ekspert rollen kan nå inviterte nye deltagere til inspeksjonen. Alle inviterte deltakere får rollen som gjest. Operatør-eksperten har så mulighet til å velge en initiell ekspert. Når dette gjøres blir operatør-eksperten satt som operatør. Videre kan den som er ekspert gi fra seg ekspertrollen til en annen gjest eller til operatøren. Eksperten blir da satt tilbake som gjest. På denne måten vil ekspertrollen kunne byttes på fortløpende gjennom en inspeksjonsøkt.

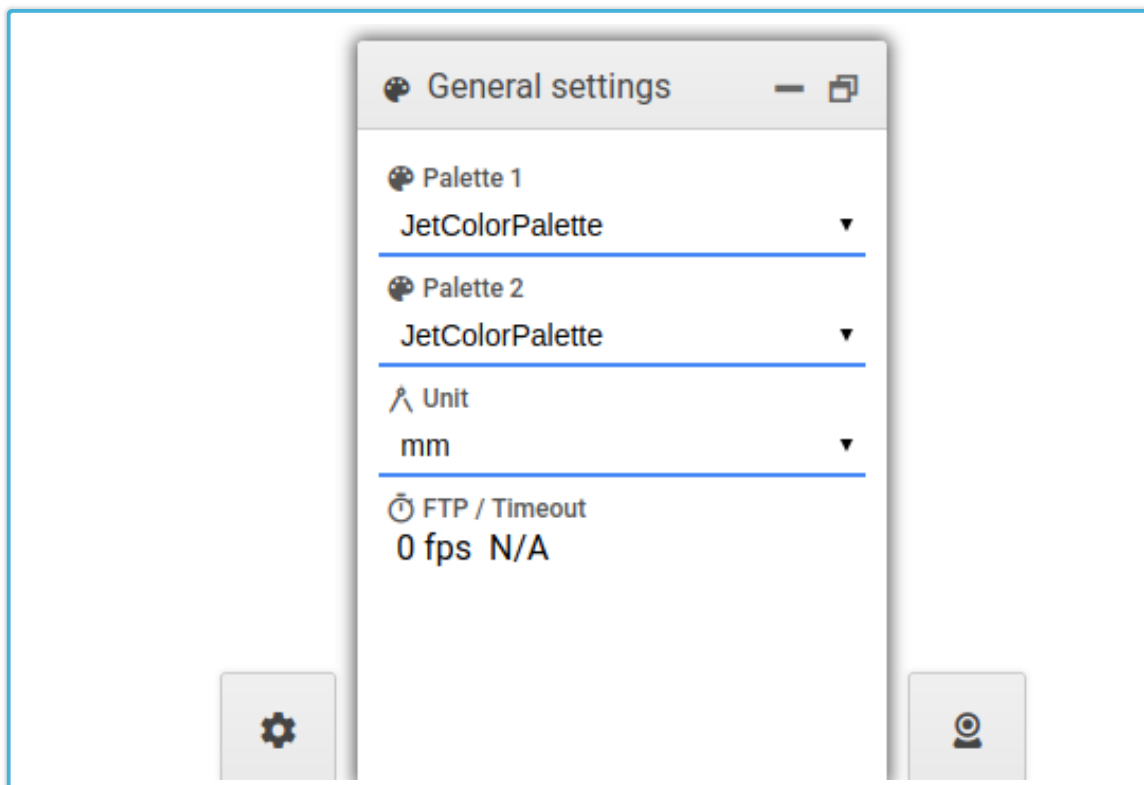
Siden operatøren er ansvarlig for selve inspeksjonen er det lagt inn muligheten for operatør å ta tilbake ekspertrollen. Dette for å ta hensyn til tilfeller der f.eks. eksperten mister tilkoblingen til portalen eller blir utilgjengelig under en inspeksjonsøkt. Figur 22 viser hvordan en operatør-ekspert gir fra seg ekspertrollen til en invitert gjest.



Figur 22: Skjerm bilde som viser overføring av ekspertrollen

6.4.2 Fleksibelt layout

Målet var å rydde i layout på en slik måte sentral funksjonalitet som visning av ultralydbilder og webkamera blir satt i fokus. Samtidig skal sekundærfunksjonalitet som direkte meldinger, innstillinger og deltakerliste ryddes bort for å frigjøre plass i grensesnittet. Funksjoner som ikke trenger å vises kontinuerlig på skjermen flyttes til egne vinduer som kan åpnes og lukkes. I figur 23 vises eksempel der to vindu er skjult og et vindu er synlig. Samtidig er det mulig å frigjøre vinduer fra sin standard plassering og flytte de fritt rundt. Ved bruk av denne metoden kan hver deltaker tilpasse grensesnittet etter egen preferanse.



Figur 23: Skjerm bilde som viser åpnet og lukket vindu

6.4.3 Tidslinje

Når deltakere av portalen skal se på en tidslinje er det viktig at også innstillinger som var satt under generering av bildene er tilgjengelig. Det er derfor nødvendig å lagre innstillinger som ble brukt for hvert bilde. Grunnet dette kravet skal nå også DolphiCam sine innstillinger inkluderes som metadata til alle ultralydbilder. Som beskrevet i sprint 1 i kapittel 3.4.5, sendes en pakke med ultralyddata og serienummer til server. Innstillinger skal nå inkluderes i denne pakken.

For å registrere hvilke innstillinger som er satt i DolphiCam for hvert generert ultralyddatasett må disse leses ut av DolphiCam. Dette blir gjort ved at innstillingene blir lagret lokalt i blackbox samtidig som de skrives til DolphiCam. På denne måten kan innstillingene inkluderes som en del av ultralyddata som sendes til server. Innstillingene sendes samme med ultralyddata fra DolphiCam og er en utvidelse av strukture vist i kodesnutt 2 fra sprint 1. Eksempel 9 viser hvordan innstillingene nå inkluderes sammen med ultralydmatrisene.

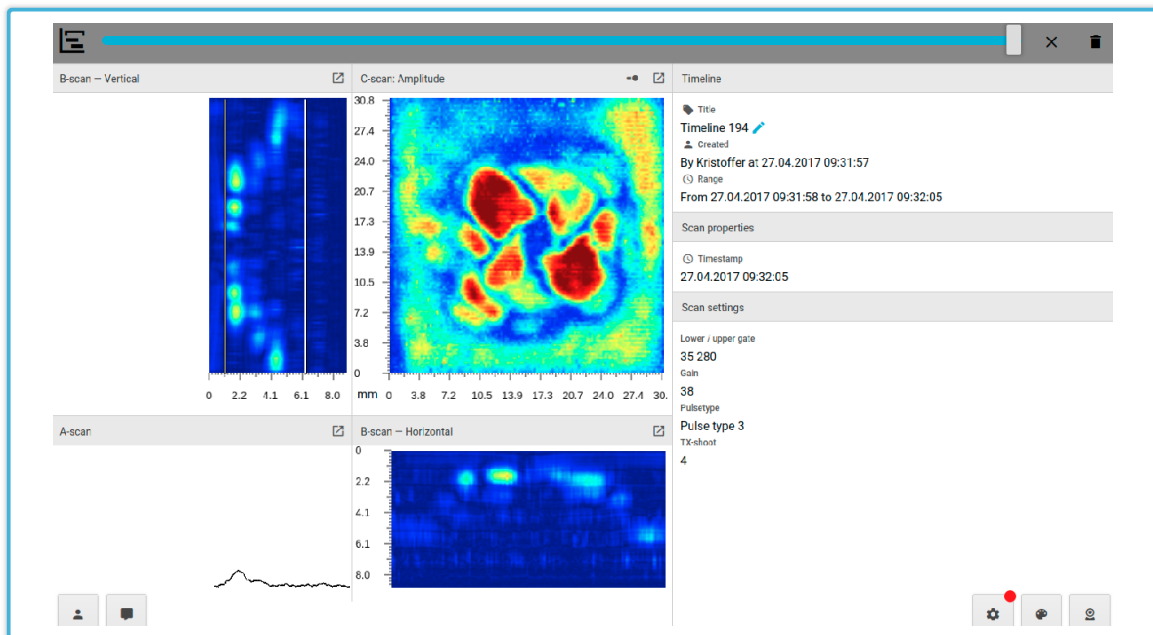
```
1 || [1,2,3;4,5,6;7,8,9]& ... &[1,2,3;4,5,6;7,8,9]& setting1=X, setting2=Y,
```

setting3=A=B

Kodesnutt 9: Eksempel på innstillinger inkludert sammen med matriser

For å kunne gi deltakere mulighet til å opprette tidslinjer må først alle ultralyddatapakker som kommer inn på server lagres i en buffer. Bufferen holder på de X siste pakkene som server har mottatt. Når en ekspert velger å lagre en tidslinje vil alle ultralyddatapakker som ligger i denne bufferen kopieres inn til en ny tidslinje. Tidslinjen inneholder nå en serie med ultralydpakker. Disse pakkene inneholder alle de fem ulike ultralydmatrixene og innstillinger satt ved generering av disse.

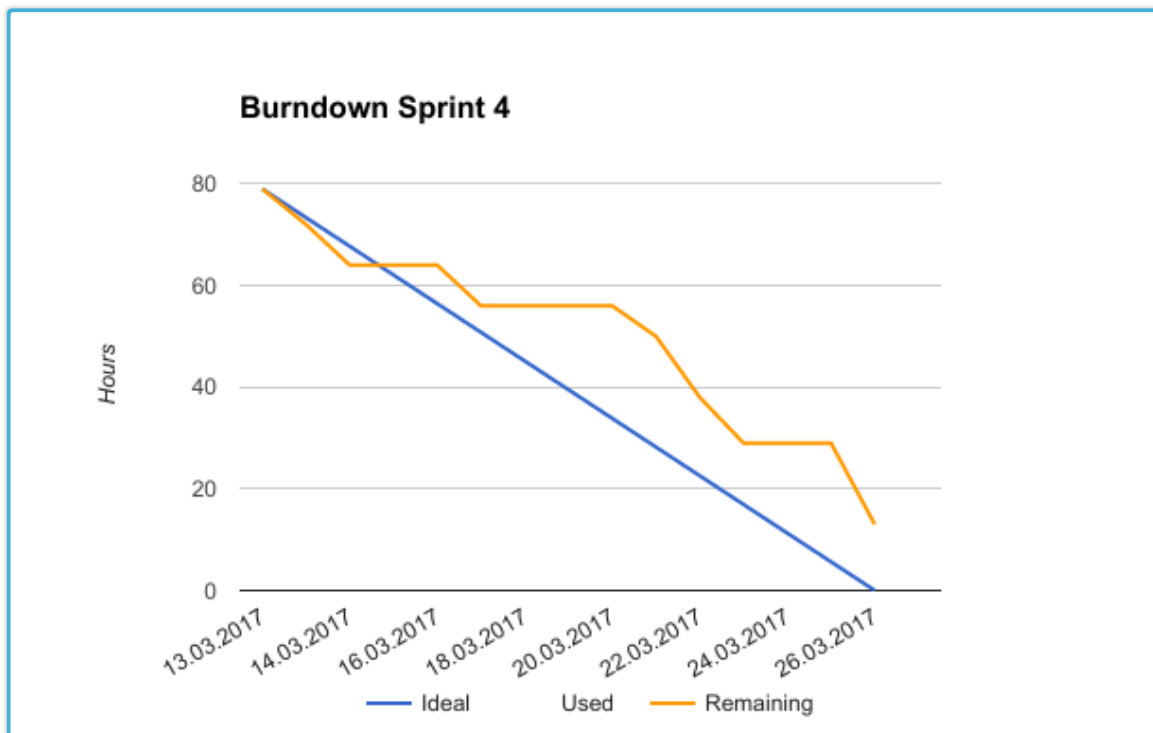
Deltakere kan nå se på en tidslinje ved å velge den fra en meny. Når en deltaker åpner en tidslinje går portalen inn i en egen modus for visning av tidslinjen. Samntidsvisningen er nå skjult og erstattes med et grensesnitt der deltakeren kan navigere frem og tilbake i serien. Visning av tidslinjen baseres på gjenbruk av eksisterende funksjonalitet som også brukes for sanntidsvisning. I tillegg inkluderes et eget panel som viser tilhørende metadata som tidsstempel og innstillinger. Figur 24 viser skjermbilde av en tidslinje.



Figur 24: Skjermbilde av tidslinje

6.5 Konklusjon

I sprinten ble det fokusert mest på å tilpasse det som ble implementert i de tre første sprintene. Mye funksjonalitet og kode ble produsert i tidligere sprinter, og det ble derfor brukt tid på refaktorering av kode og organisering av GUI. Oppdragsgiver var spesielt fornøyd arbeidet rundt GUI. Alle kravspesifikasjonene definert for denne sprinten ble implementert. Estimeringer som vist i figur 25 stemte bra med brukt tid noe som kan tyde på at utviklerene har blitt bedre til å estimere. Grunnen til dette kan også være bedre oppdeling av user stories en i tidligere sprinter.



Figur 25: Burndown diagram for sprint 4

7 SPRINT 5

7.1 Innledning

I denne sprinten ble det satt fokus på å finpusse på portalens helhet. Det ble brukt mest tid på endringer av tidslinjefunksjonen, samtidig som målestokk for ultralydbilder og varsling ble implementert. Det ble også gjort finpuss av GUI.

7.2 User Stories

Tabell 10: User stories sprint 5

#	User story	Estimert (timer)	Brukt (timer)
10	Som deltaker vil jeg endre enhet.	1	1
11	Som ekspert vil jeg kunne slå av og på Hilbert-funksjon.	8	7
13	Som deltager vil jeg se en beskrivende representasjon av A-scan.	6	0
32	Som ekspert ønsker jeg å stoppe/starte scanning i portalen.	4	6
33	Som deltaker ønsker jeg dynamisk målestokk på B/C scans.	8	12
34	Som deltaker vil jeg kunne endre fargepallett i timeline modus.	2	2
35	Som ekspert vil jeg kunne endre navn på en timeline.	2	2
36	Som ekspert vil jeg starte opptak av en timeline.	6	7
37	Som ekspert vil jeg ha mulighet til å pause opptak av timeline.	4	3
38	Som ekspert ønsker jeg at jeg kan stoppe opptak av timeline manuelt eller at den stoppes automatisk når den er full.	6	6
39	Som deltaker ønsker jeg å se opptaksstatus på en timeline.	6	4
40	Som ekspert vil jeg velge oppløsning når nytt opptak av timeline startes.	6	4
41	Som deltaker ønsker jeg et mer fleksibelt GUI.	16	15
		75	69

7.3 Kravspesifikasjon

7.3.1 Tidslinje

I sprint 4 ble tidslinje-funksjonen implementert. Etter demo av funksjonen ønsker oppdragsgiver å endre på hvordan denne funksjonen fungerer for å gjøre den mer fleksibel. Tidslinjen fra sprint 4 fungerte ved at bilder fra en buffer som holder på de X siste ultralyddatasettene fra DolphiCam kopieres over til en permanent lagret tidslinje. Kravspesifikasjonen endres nå til at ekspertene skal ha mulighet til å starte opptak av en tidslinje. Når opptak startes vil ultralydbiler bilder legges inn i tidslinjen fortløpende, frem til opptak stoppes eller til tidslinjens maksgrense for antall bilder er nådd. Eksperten skal også ha mulighet til å pause opptaket, for så å starte det igjen. Samtidig skal ekspertene ha mulighet til å velge oppløsning på opptaket. Oppløsningen skal kontrollere hvor mange bilder det skal gå mellom hvert bildet som lagres. Eksempel på oppløsning kan være et bilde per tiende bilde som produseres.

Fra tidligere sprinter har tidslinjer som blir opprettet blitt navngitt med et tilfeldig tall. Oppdragsgiver ønsker at det skal være mulig å endre navnet på enhver tidslinje til det som ekspertene ønsker.

7.3.2 Målestokk

Ultralydbildene som vises i portalen tilsvarer en bestemt dimensjon på prøveobjektet. For å visualisere hvor stor et område på ultralydbildet tilsvarer på prøveobjektet ønsker oppdragsgiver en målestokk som langs aksene på bildene. Disse skal være tilsvarende oppsettet som er brukt i eksisterende programvare vist i figur 1. Målestokken skal også ha støtte for forskjellige typer enheter som millimeter (mm), tommer (inches) og tusendels tommer (mils). Det skal være mulig å endre enhet som en ønsker å bruke i portalen.

7.3.3 Fjernstyrt start/stopp av skanning

Frem til nå har start og stopp av skanning blitt kontrollert vha en fysisk knapp på DolphiCam. Det er nå ønskelig å gjøre det mulig for ekspertene å utføre denne operasjonen via portalen.

7.3.4 Fargepalett i tidslinjevisning

Når en deltaker ser på ultralydbilder fra en tidslinje skal det være mulig å endre paletten som brukes til å rendere de historiske bildene. Denne user storien har blitt delvis implementert som en del av

sprint 4. Derfor kreves det kun små endringer i GUI for å tilfredstille kravet.

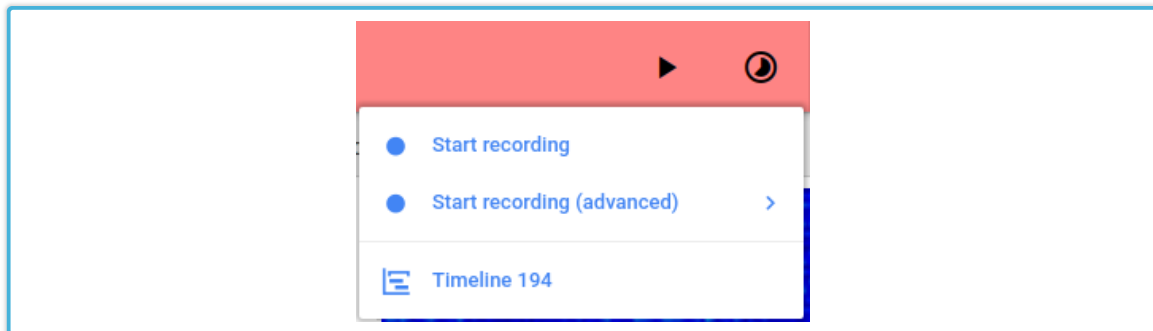
7.3.5 Varsler

Fra sprint 4 ble det implementert et mer fleksibelt GUI der grensesnittet ble delt inn i små vindu som kan minimeres, maksimeres og flyttes rundt på skjermen. For å synliggjøre at det har skjedd noe nytt i en av disse vinduene mens de er minimert ønsker oppdragsgiver at det skal gis en indikasjon på dette i GUI. Eksempel på hendelser som kan skje når et vindu er minimert er en ny direkte melding eller en endring av DolphiCam sine innstillinger.

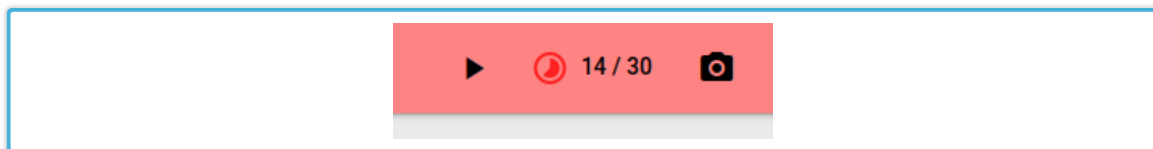
7.4 Implementasjon

7.4.1 Tidslinje

Når en ekspert velger å starte opptak av en tidslinje i portalen vil det først opprettes en blank tidslinje. Tidslinjen har en status som sier om den er under opptak, pauset eller ferdig. Dersom en tidslinje er under opptak vil ultralydpakker som kommer inn på server lagres direkte i denne tidslinjen. Eksperten kan velge å pause opptak og dermed vil ikke ultralydpakker registreres inn i tidslinjen frem til den er startet igjen. Hver gang et ny ultralydpakke legges inn i tidslinjen sjekkes det om maksimumsgrensen for antall ultralydpakker er nådd. Dersom dette er tilfellet endres status på tidslinjen automatisk til ferdig. Eksperten kan også velge å stoppe opptaket før grensen er nådd dersom ønskelig. Når en ekspert starter opptak av en tidslinje kan eksperten også velge oppløsning på opptaket. Oppløsninga bestemmer hvor mange ultralydpakker som skal ignoreres mellom hver registrert pakke. I det opptaket er ferdig kan alle deltakere se på tidslinjen slik som beskrevet i kapittel 6.4.3. Figur 26 viser hvordan en tidslinje kan startes. Figur 27 viser opptaksstatus av en tidslinje.



Figur 26: Skjermbilde av hvordan start av tidslinje blir gjort



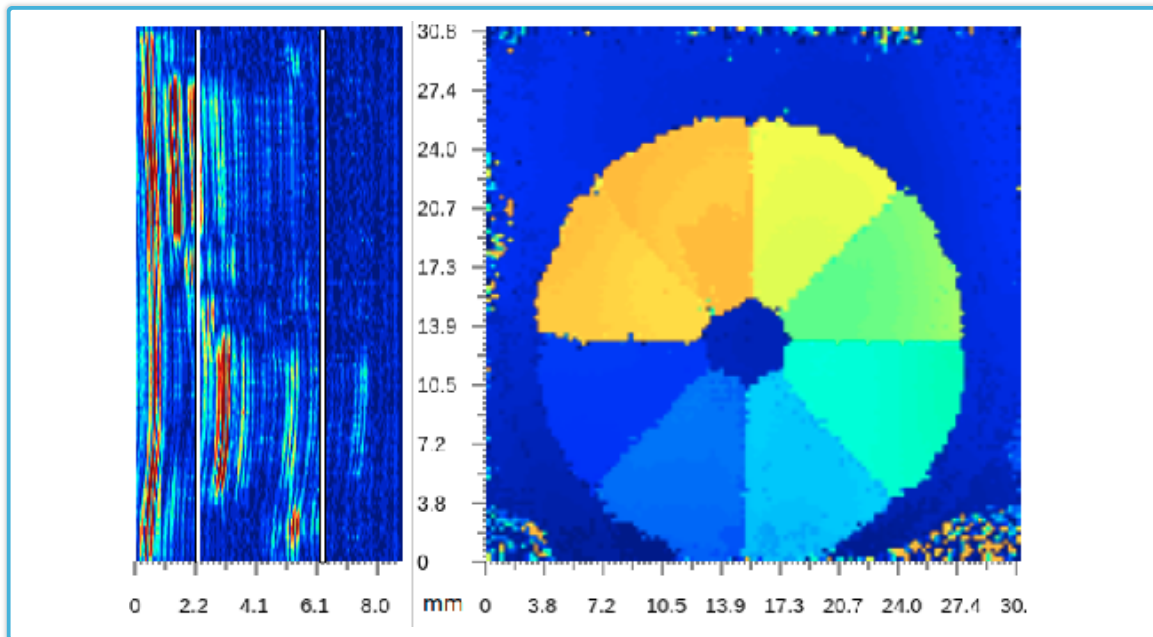
Figur 27: Skjermbilde av tidslinjeopptak

7.4.2 Fargepalett i tidslinjevisning

I sprint 4 ble gjort en del endringer i GUI som fikser dette problemet. Muligheten for å endre fargepalett er fortsatt tilgjengelig på samme sted som om du ikke var i tidslinjevisning. Siden ultralydbildene som vises i tidslinjevisning renderes med samme funksjon som i sanntidsvisning vil fargepalettvalget også påvirke tidslinjevisning. Derfor kan paletten endres i tidslinjevisning på samme måte som i sanntidsvisning og ønsket resultat blir oppnådd.

7.4.3 Målestokk

En dynamisk målestokk for alle bilder er implementert for bruk i både horisontal og vertikal retning. Figur 28 viser et eksempel på dette. Målestokkene blir generert i det portalen lastes inn, og vil automatisk oppdateres dersom deltakeren endrer enhet. Målestokken fungerer ved at en fra og til verdi settes. Dimensjonene på målestokken bestemmer hvor mange skritt som skal vises. Fra og til verdiene vil så fordeles på disse. Selve visningen av målestokken produseres ved å plote inn linjer på et canvas-element som er plassert langs aksene på tilhørende ultralydbilder. Figur 28 viser hvordan målestokken ser ut med milimeter satt som enhet.



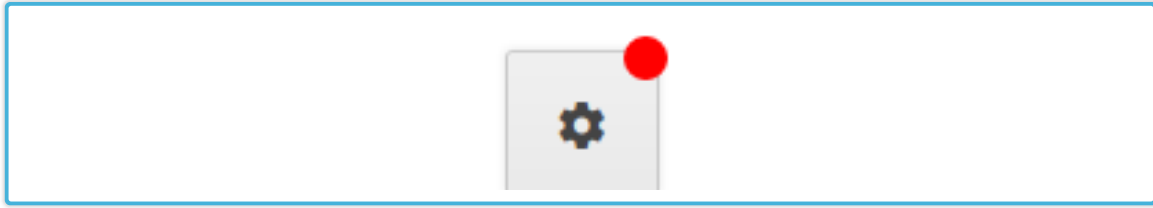
Figur 28: Skjerm bilde av målestokk

7.4.4 Fjernstyrt start/stopp av skanning

Det er to måter start/stopp av skanning kan skje; ved å trykke på den fysiske knappen på DolphiCam eller i software. Ved implementasjon av fjernstyrt start/stopp av skanning ble funksjonalitet for å sette innstillinger i DolphiCam gjenbrukt. Skanning av/på settes som en egen innstilling i DolphiCam, og dermed kan denne kommuniseres til blackbox på samme måte som andre innstillinger.

7.4.5 Varsler

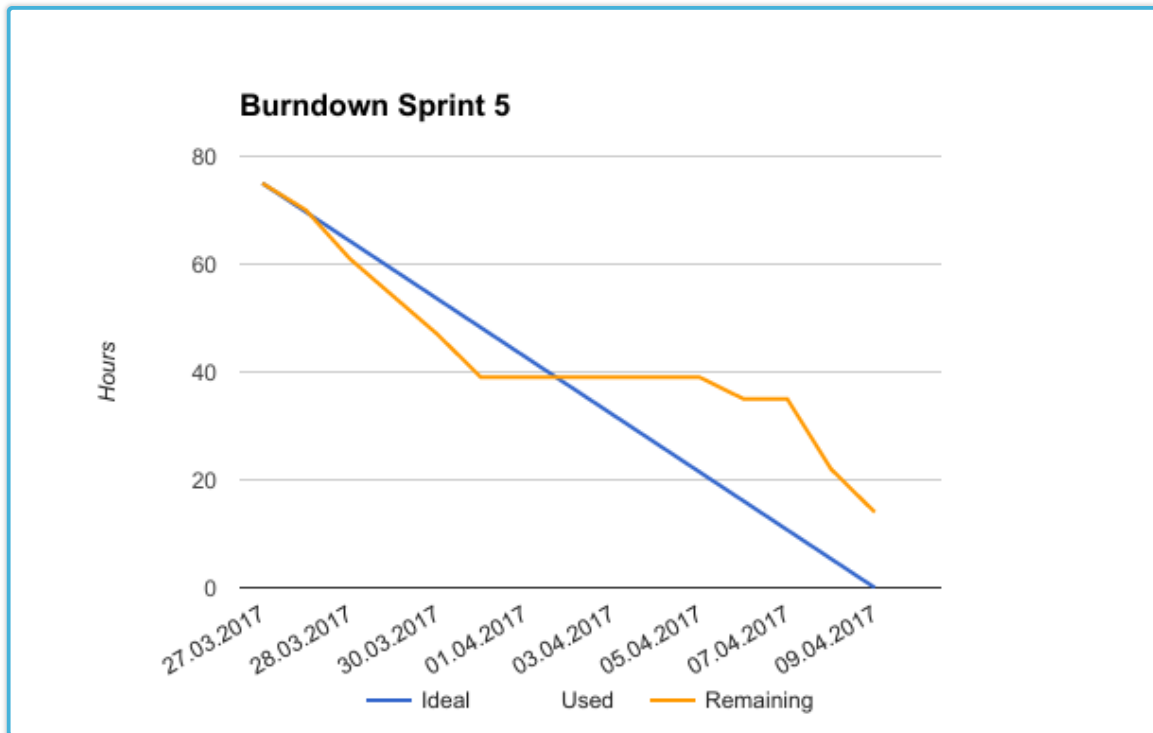
Når klienten mottar en melding som påvirker innholdet i en av vinduene som kan minimeres, sjekkes det først om vinduet er minimert. Dersom dette er tilfellet vil klienten markere det minimerte vinduet med en rød prikk. Slik blir deltakere informert om nye direktemeldinger, endrede innstillinger eller andre endringer selv om vinduer er minimert. Figur 29 viser eksempel på denne indikasjonen.



Figur 29: Skjerm bilde av ny notifikasjon

7.5 Konklusjon

Kravspesifikasjonene i sprinten ble ikke fullstendig implementert. User story #11 klarte ikke gruppen å ferdigstille. Story #13 ble ikke ferdigstilt fordi denne delvis avhenger av #11. I sprint 7 ble det klart at grunnen til at gruppen ikke klarte å implementere storyene var en feil i DolpibLib. Etter sprinten ble disse user storyene tatt med videre til sprint 6. Resterende user stories ble implementert kravspesifikasjonene. Brukt tid viste seg å stemme bra med estimeringene. Det ble også tatt en pause fra utviklingsarbeidet midt i denne sprinten, noe som visualiseres på burndown diagrammet i figur 30.



Figur 30: Burndown diagram for sprint 5

8 SPRINT 6

8.1 Innledning

I denne sprinten var det satt fokus på å skape et mer komplett produkt. En ny innstilling ble implementert, samtidig som tilpassing av en eksisterende innstilling ble gjort. Mulighet for eksperter å eksportere og importere innstillinger mellom forskjellige inspeksjonsøkter samt en mer visuell representasjon av innstillingene lower og upper gate ble også implementert i denne sprinten.

8.2 User Stories

Tabell 11: User stories sprint 6

#	User story	Estimert (timer)	Brukt (timer)
11	Som ekspert vil jeg kunne slå av og på Hilbert-funksjon	6	4
13	Som deltager vil jeg se en beskrivende representasjon av A-scan	6	0
42	Skjule B-scan vertical/horizontal settings	2	2
43	Som ekspert vil jeg lagre dolphicam-settings som en template i localStorage	4	2
44	Som ekspert vil jeg laste inn dolphicam-settings lagret lokalt i min nettleser	4	3
45	Som ekspert vil jeg endre TCG-setting	10	10
46	Som ekspert ønsker jeg å ta et snapshot	8	7
47	Lese ut gain når pulstype endrast	6	0
48	Som deltaker vil jeg se linjer for gating på b-scan vertical	3	4
		49	32

8.3 Kravspesifikasjon

8.3.1 Innstillinger i DolphiCam

Det skal implementeres muligheten til å sette TCG (Time Corrected Gain) i portalen. TCG er en innstilling som amplifiserer signalet DolphiCam tar desto lengre inn i materialet signalet kommer fra. Dette vil si at signaler som kommer fra 0.5mm inn i materialet blir gjort kraftigere enn signalene fra 0.3mm. Dette er uavhengig av hva som faktisk blir returnert av signalet. Eksperten skal ha mulighet

til å velge mellom light, medium og heavy.

8.3.2 B-scan vertikal

I figur 1 vises det hvordan DolphiTech sin eksisterende programvare viser to vertikale linjer på b-scan vertikal. Linjene er plassert på 0.8mm og 5.6mm, noe som er en refleksjon av verdiene til lower gate og upper gate innstillingen som er satt i DolphiCam. Oppdragsgiver ønsker nå å implementere denne funksjonaliteten også i portalen. Når innstillingene lower gate og upper gate endres i portalen skal linjene justeres automatisk.

8.3.3 Lagring av innstillinger

Noen eksperter har et bestemt sett med innstillinger som de ønsker å ta vare på for bruk i en fremtidig inspeksjon. Derfor skal eksperter ha muligheten for å lagre gjeldende innstillinger som er satt i DolphiCam. Innstillingene skal lagres lokalt i nettleseren til eksperten, og eksperten skal ha mulighet til å gi innstillingssettet et eget navn. Samtidig skal det være mulig for en ekspert å importere inn disse innstillingene i en senere inspeksjon.

8.3.4 Snapshot

En viktig del utførelse av en NDT-inspeksjon er å dokumentere resultatene av inspeksjonen. Dette gjøres ofte ved at et sett med ultralydbilder som viser en skade må lagres slik at det kan inkluderes som en del av en rapport. Derfor skal muligheten til å ta en såkalt snapshot implementeres. Når en ekspert tar en snapshot skal de fem ulike typene ultralydbilder, samt metadata som innstillinger i DolphiCam hentes ut i det eksperten trykker på snapshot-knappen. Bildene og metadata skal lagres sammen og gis et navn. Eksperten skal kunne åpne opp et snapshot som er tatt og ha muligheten til å lagre denne direkte fra nettleseren.

8.4 Implementasjon

Innstillinger i DolphiCam

TCG-innstillingen er implementert med light, medium og heavy. For implementasjon ble det gjenbrukt funksjonalitet fra den initielle implementasjonen av innstillinger fra sprint 2. Disse er beskrevet

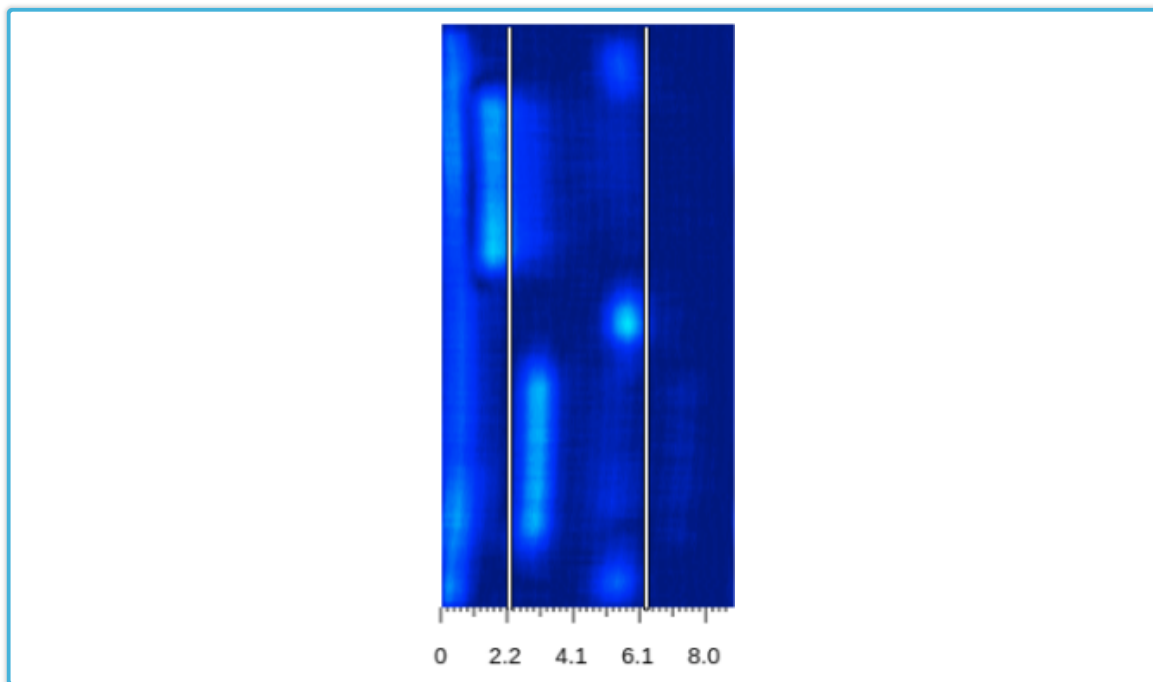
i kapittel 4.4.1.

8.4.1 B-scan vertikal

Lower og upper gate innstillingene sier noe om hvor dypt inn i materialet det skal skannes. Disse innstillingene reflekteres i c-scan, men er visualisert vha vertikale linjer vertikal b-scan. Når upper eller lower gate blir endret flyttes linjene automatisk på bildet. Linjenes posisjon kalkuleres vha. følgende ligninger:

$$\begin{aligned} lowerGateOffsetLeft &= \frac{lower}{390} * 186 \\ upperGateOffsetRight &= 186 - \left(\frac{upper}{390} * 186 \right) \end{aligned}$$

Variablene i ligningen bestemmer hvor mange piksler i horisontal retning fra y-aksen linjene for lower og upper gate skal plasseres. Figur 31 viser hvordan disse linjene vises i portalen.



Figur 31: Skjermbilde av linjer i b-scan vertikal

8.4.2 Lagring av innstillinger

Innstillinger mottatt fra server blir lagret som et JSON objekt. Dette objektet vil da inneholde alle innstillinger satt i DolphiCam. Når en ekspert vil lagre gjeldene innstillinger blir det tatt en kopi av objektet. Sammen med et egendefinert navn blir JSON objektet lagret i localStorage [13] i nettleseren. Denne metoden muliggjør vedvarende lagring i nettleseren og ivaretas frem til en fremtidig inspeksjon. En ekspert kan lagre opp til flere sett med innstillinger samtidig.

Ved å sette et brukerdefinert navn til innstillingen er det mulig for brukeren å importere et tidligere lagret sett med innstillinger. Importerte innstillinger fra localStorage blir satt på samme måte som innstillinger mottatt fra server. Dette gjør at samme funksjonskall som når innstillinger settes manuelt blir brukt. Dette sørger for gjenbruk av eksisterende kode og forenkler implementasjon av nye innstillinger.

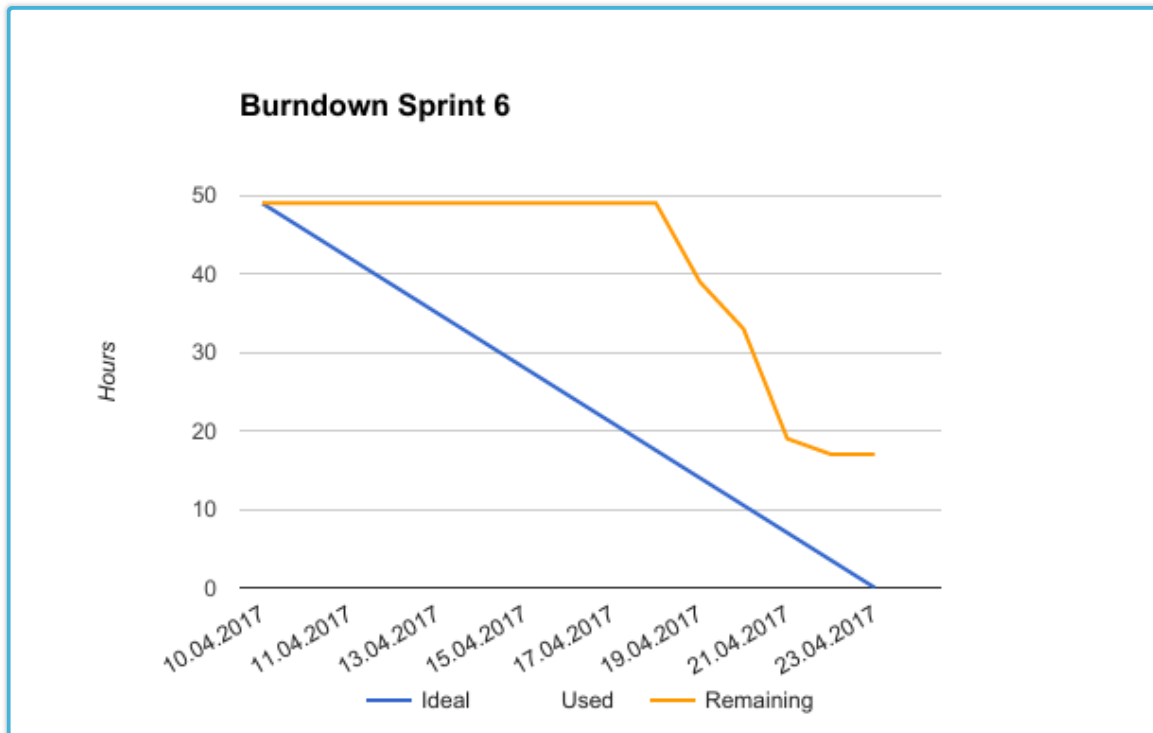
8.4.3 Snapshot

En egen knapp er lagt til i GUI der ekspert kan ta et snapshot. I det ekspert tar snapshot, vil alle fem ultralydbilder blir lest ut som et separat bilde fra nettleseren. Samtidig vil oversiktskamera bli lest ut som et bilde. Eksperten vil så kunne gi snapshot et navn. Alle bilder, med navn og innstillinger satt blir sendt til serveren og lagret i databasen.

Alle brukere kan nå velge ett snapshot som er tatt og all tilhørende data vil bli vist på ei egen html side. Dette gjør at brukeren kan lagre HTML siden og bruke den i en rapport eller sende til andre uavhengig av portalen.

8.5 Konklusjon

Kravspesifikasjonene i denne sprinte ble noe mindre en tidligere sprinter. I stedetfor ble det lagt mer fokus på refaktorering og små optimaliseringer. Som figur 32 viser ble det estimert mer en brukt tid. Dette er fordi tre user stories ble flyttet videre til sprint 7. Story #11 og #13 ble flyttet av samme grunn som beskrevet i sprint 5. Story #47 ble også tatt med videre til sprint 7. Etter kontakt med DolphiTech kom det frem at DolphiLib måtte oppdateres. Disse user storiene blir ferdig implementert i sprint 7.



Figur 32: Burndown diagram for sprint 6

9 SPRINT 7

9.1 Innledning

Det ble planlagt en workshop som skjedde 27. april, denne er beskrevet i kapittel 2.7.2. User story 11 og 13 som ble tatt med helt fra sprint 5 ble implementert. Noen små endringer når innstilling ble satt.

9.2 User Stories

Tabell 12: User stories sprint 7

#	User story	Estimert (timer)	Brukt (timer)
11	Som ekspert vil jeg kunne slå av og på Hilbert-funksjon	5	3
13	Som deltager vil jeg se en beskrivende representasjon av A-scan	4	1
47	Lese ut gain når pulstypen endres	4	2
48	Som ekspert ønsker jeg muligheten til å skru av TCG.	2	1
49	Som deltager ønsker jeg at TCG skal oppdateres når gain endres.	4	3
52	Oppdater enhetsvisning når enhet blir endret	1	1
		20	11

9.3 Kravspesifikasjon

9.3.1 Innstillinger i DolphiCam

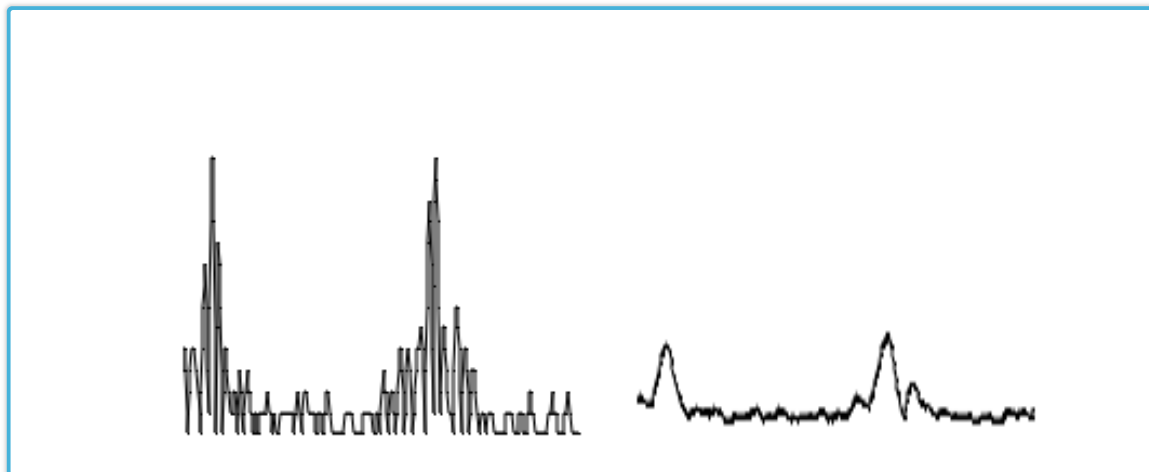
Når innstillingen 'pulstypen' endres vil DolphiCam automatisk endre innstillingen 'gain' til standardverdien for den gitte pulstypen. Derfor må programvaren i blackbox sørge for å lese ut den nylig satte gain-verdien hver gang pulstypen endres og sørge for at denne blir oppdatert i portalen.

9.3.2 Hilbert/Envelope funksjon

En viktig funksjon som brukes for å gjøre ultralydbilder mer lesbare er envelope-funksjonen. Denne funksjonen tar i praksis absoluttverdien av verdier i et datasett og jevner så ut punktene. Dette er

nyttig for å produsere en kurve som er lettere å lese. Figur 34 viser eksempel på hvordan en kurve ser ut før og etter at datasettet er kjørt gjennom envelope-funksjonen.

Oppdragsgiver ønsker å få implementert bruk av funksjonen på ultralyddata som blir generert av DolphiCam før det blir sendt til server. Det finnes en innebygd funksjon i DolphiTech's SDK for å gjøre dette. Denne kravspesifikasjonen inngår også som en del av å skape en mer beskrivende representasjon av A-scan.



Figur 33: Eksempel på kurve uten og med envelope

9.4 Implementasjon

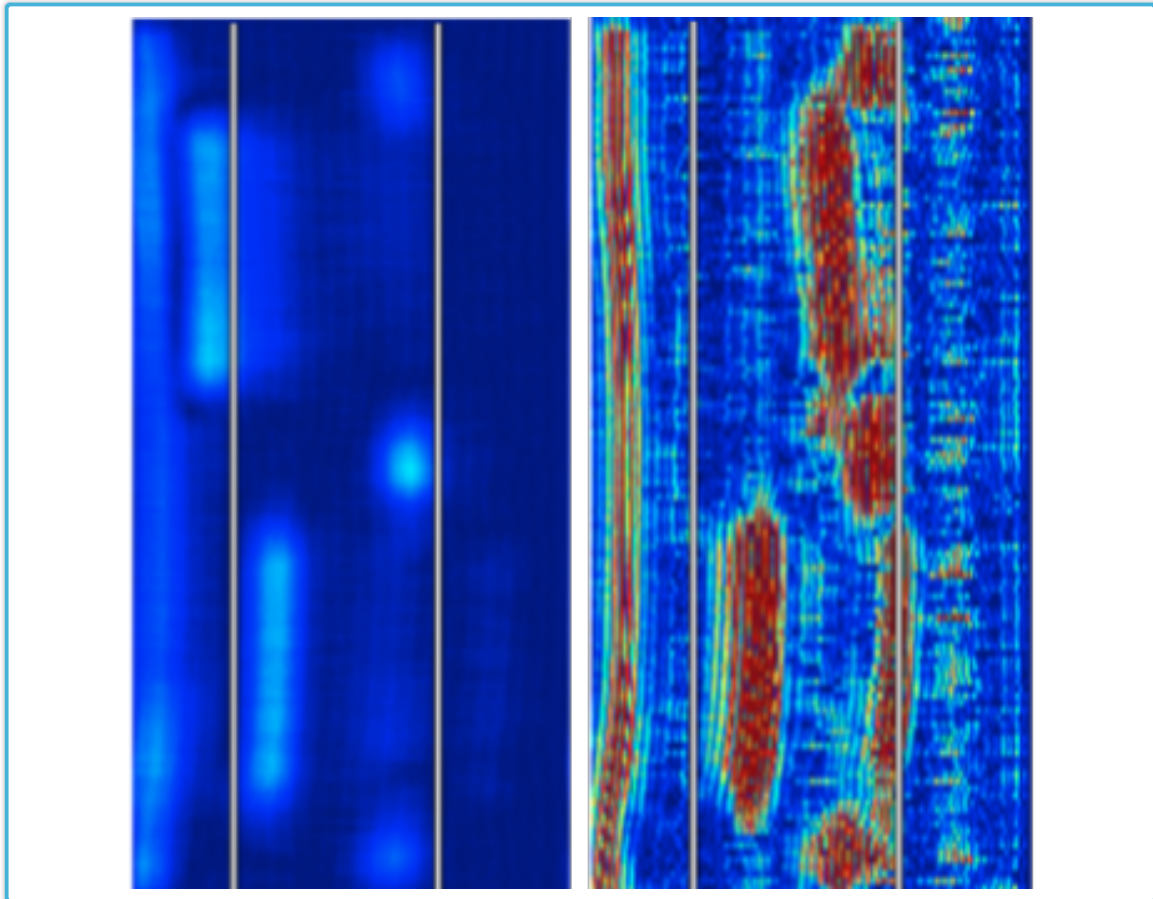
9.4.1 Innstillinger i DolphiCam

Når pulstype blir satt vil gain bli forespurt fra DolphiCam og i det svar blir mottatt vil den lokale gain variabelen i blackbox oppdateres. Alle innstillinger blir allerede sendt med ultralyddata pakken, som beskrevet i kapittel 6.4.3. Når klienten mottar denne pakken vil den lese ut innstillingene. Hvis gain innstillingen har blitt endret vil GUI bli oppdatert.

9.4.2 Hilbert/envelope funksjon

En sjekk etter at utlesning av skann blir gjort. Hvis hilbert er aktivert, vil utlest data sendt gjennom hilbert-funksjonen. Dette blir gjort på A-scan, B-scan vertikal og horisontal. Hilbert kan aktiveres ved at en ekspert setter innstillingen i portalen.

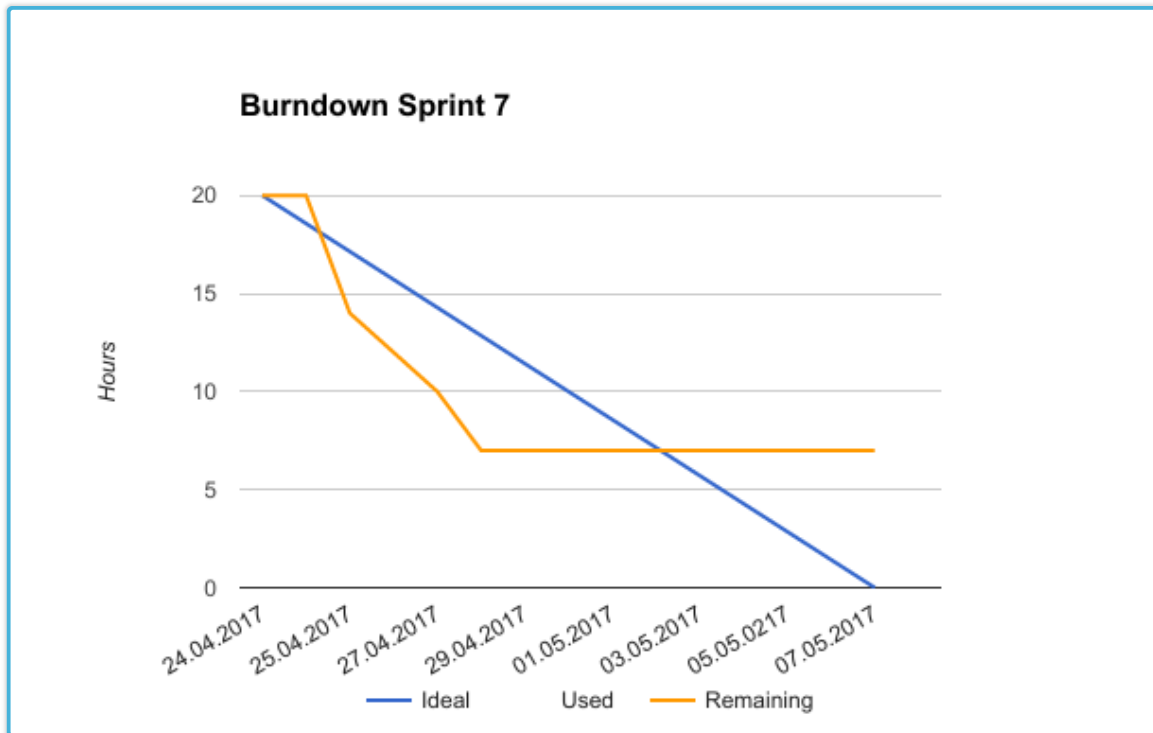
For en bedre representasjon av A-scan ble det laget en linje mellom pikslene. Dette ble gjort ved å tegne en strek fra nåværende piksel til forrige. Figur ?? viser med og uten bruk av envelopefunksjon.



Figur 34: Eksempel på ultralydbilder uten og med envelope

9.5 Konklusjon

Det ble fokusert på å ferdigstille produktet i denne sprinten. Mindre implementasjoner i tillegg til user stories #11 og #13 ble implementert. Det ble også gjennomført to workshops som en del av denne sprinten. Disse er beskrevet i kapittel 2.7.



Figur 35: Burndown diagram for sprint 7

10 DISKUSJON

10.1 Sprint oppsummering

I de første tre sprintene hadde gruppen forholdsvis god oversikt over hva slags funksjonalitet som skulle implementeres. Fra sprint 4 ble det fokusert på implementasjon av tilleggsfunksjonalitet til det som ble implementert i de første sprintene, sammen med noe forbedring av eksisterende funksjonalitet. I kapittel 2.4 gis en mer detaljert oversikt over hva som inngikk i hver av sprintene.

Fra sprint 2 og utover ble det gjort estimeringer for hver av user storyene som inngikk i sprintene. Det ble prioritert å bruke ekstra tid på storyene der vi hadde overestimert hvor mye tid som var nødvendig for å implementere funksjonaliteten. Denne ekstra tiden ble brukt til optimalisering og refaktoring av koden som ble produsert som en del av implementasjonen. Alle helg- og helligdager har vært en del av en sprint. Dette har ført til at noen burndowndiagrammene viser en rett horisontal linje over deler av sprinten, noe som er et resultat av ferie og friperioder innad i sprinten.

10.2 Resultat

Basert på den samlede kravspesifikasjonene klarte vi å implementere mesteparten av kravene som ble satt opp. Totalt sett er vi fornøyde med produktet vi har utviklet. Selv om det fortsatt gjenstår noe finpuss som bør implementeres for at produktet skal virke komplett, kan systemet tas i bruk til å utføre en Remote NDT inspeksjon, noe som var hovedmålet med oppgaven. Resultatet av prosjektet er et fungerende system som baseres på moderne web-standarder og er tilrettelagt for videre utvikling.

Grunnet tidsbegrensingene til prosjektet fikk vi ikke muligheten til å implementere alt som vi ønsket. Tilpassinger av systemet for mobil og nettbrett, muligheten til å generere rapporter og en rekke flere innstillinger som kan settes i DolphiCam er eksempler på dette. Samtidig har valget om å bruke Scrum ført til at noen kravspesifikasjoner ikke ble kjent for gruppen før sent i utviklingsprosessen. Dette førte til at disse kravspesifikasjonene ikke ble like godt implementert som de som ble presentert i starten av prosjektet.

Bruk av Scrum som utviklingsmodell sørget for at vi har hatt regelmessige møter med oppdragsgiver. Dette har vist seg å være svært nyttig både for oss og oppdragsgiver. DolphiTech har vært hjelpsomme gjennom hele utviklingsprosessen, samtidig som de har gitt oss stor frihet til å implementere løsningen slik vi selv ønsket, både med tanke på valg av teknologier og utforming av det grafiske grensesnittet. Dette resulterte i et både lærerikt og interessant prosjekt.

10.3 Hva kunne blitt gjort annerledes?

Det ble brukt for lite tid i planleggingsfasen. En bedre plan for sprintene skulle blitt satt opp. Vi forsøkte å ha et delt ansvar for alle prosjektroller der det var mulig. Selv om dette ikke har skapt noen store problemer ville det vært enklere om roller og ansvar var ble dedikert til et enkelt gruppemedlem. Vi burde også ha tatt hensyn til påskeferie under planlegging slik at sprint 6 ikke ble halvvert i arbeidsmengde. Samtidig burde det ikke ha blitt satt opp mer en seks sprinter. På grunn av at en syvende sprint ble inkludert ble det for lite tid igjen til rapportskriving. Vi ser i etterkant at det ble vanskelig å utforme en god rapport på så liten tid.

10.4 Videre utvikling

Systemet som er utviklet har et stort potensiale i form av ny funksjonalitet som kan bygges rundt det som har blitt implementert i dette prosjektet. Under beskrives noen av funksjonene som kunne blitt implementert dersom systemet skal videreutvikles:

- Inspeksjoner
 - Mulighet til å dele opp en inspeksjonsøkt i flere inspeksjoner.
- Inspeksjonsattributter
 - Muligheten til å legge inn beskrivelser om hva som har blitt inspisert og tilhørende resultater.
- Brukere
 - Muligheten til å opprette faste brukere som kan gjenbrukes for flere inspeksjonsøkter.
 - Se tidligere inspeksjoner som brukeren har vært med på.
- Rapport
 - Generering av en full rapport av inspeksjonsøkten.
- Cad modell
 - Mappe inspeksjonsområdet opp mot en cad modell, med sanntidsoppdatering.

DolphiTech holder på med utvikling av en ny versjon av DolphiCam. Denne leverer bilder med en betydelig høyere frekvens enn den versjonen som ble brukt under dette prosjektet. Systemet som ble utviklet i dette prosjektet er ikke optimalisert for å tilpasses en så høy bildefrekvens. Dermed vil utbedring av hvordan systemet overfører og håndterer ultralydbilder være en naturlig del av videreutviklingen av systemet. I tillegg vil det være naturlig å fokusere på generelle optimaliseringer av blackbox-programvaren og bedre brukeropplevelse i portalen.

10.5 Gruppeevaluering

Vi har jobbet sammen i flere prosjekter tidligere og viste derfor godt hva som var forventet av begge. Vi visste også hva slags erfaringer og kunnskaper den andre hadde. Dette gjorde at det ble lett å fordele arbeidsoppgaver.

Samarbeidet i gruppen har vært veldig bra, begge har vært innom begge deler av systemet, selv om vi har hatt fokus på våre egne oppgaver. Totalt sett har vi begge brukt en nesten identisk mengde tid på prosjektet. Vi hadde ordnet med eget kontor, slik at begge to kunne jobbe sammen under hele utviklingsperioden. Derfor kunne vi diskutere problemstillinger fortløpende noe som gav en god arbeidsflyt.

Siden vi kjente hverandre godt før vi begynte, var det ikke noe problem å kritisere eller kommentere på det den andre hadde gjort. Siden begge hadde samme målsetning for prosjektet skapte dette et godt utviklingsmiljø. Alle implementasjoner har blitt diskutert på en konstruktiv måte og ingen betydelige konflikter eller uenigheter oppstod under prosjektet.

11 KONKLUSJON

Systemet som er utviklet gjør det mulig å utføre en Remote NDT inspeksjon på en enkel og brukervennlig måte. Basert på moderne web-teknologier gir systemet eksperter muligheten til å delta på en inspeksjon fra hvor som helst i verden. Portalen leverer ultralydbilder av tilsvarende kvalitet som eksisterende løsninger der eksperten må være fysisk til stedet. Selv om vi skulle ønske at vi hadde mer tid til utvikling kan vi si oss fornøyde med sluttresultatet.

Alle resultatmålene som initialt ble satt opp ble oppnådd i løpet av prosjektet. Samtidig har vi fått erfaringer med Scrum og med å jobbe for en reell oppdragsgiver. Implementasjon av WebRTC og C++ programmering av også vært viktige læringsutbytte fra prosjektet.

Vi håper at systemet vil videreutvikles og kan tas i bruk for å bidra til å løse dagens utfordringer innen utførelse av NDT.

BIBLIOGRAFI

- [1] Wikipedia, 2017. Ikke-ødeleggende utprøving, https://no.wikipedia.org/wiki/Ikke-%C3%B8deleggende_utpr%C3%B8ving, (Besøkt april 2017)
- [2] Libcurl, 2017. oversikt, libcurl. <https://curl.haxx.se/libcurl/>, (Besøkt mai 2017)
- [3] Zlib, 2017. Zlib. <http://www.zlib.net/>, (Besøkt mai 2017)
- [4] Federal Aviation Administration, 2016. Sensory Prognostics and Management System II. <http://www.tc.faa.gov/its/worldpac/techrpt/tc15-41.pdf> (Besøkt mai 2017)
- [5] PHP, 2017. Documentation, php. <http://php.net/docs.php> (Besøkt april 2017)
- [6] Wikipedia, 2017. Base64. <https://en.wikipedia.org/wiki/Base64> (Besøkt mai 2017)
- [7] JSON, 2017. Introducing JSON. <http://www.json.org/> (Besøkt mai 2017)
- [8] Let's Encrypt, 2017. Let's Encrypt. <https://letsencrypt.org/> (Besøkt mai 2017)
- [9] Wikipedia, 2017. ECMAScript. https://en.wikipedia.org/wiki/ECMAScript#7th_Edition_-_ECMAScript_2016 (Besøkt mai 2017)
- [10] Wikipedia, 2017. WebRTC. <https://en.wikipedia.org/wiki/WebRTC> (Besøkt mai 2017)
- [11] PeerJS, 2017. The PeerJS library. <http://peerjs.com/> (Besøkt mai 2017)
- [12] Wikipedia, 2017. Peer-to-peer. <https://en.wikipedia.org/wiki/Peer-to-peer> (Besøkt mai 2017)
- [13] Mozilla, 2017. localStorage, <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage> (Besøkt mai 2017)
- [14] ShareLatex, 2017. Project, ShareLatex. <https://www.sharelatex.com> (Besøkt mai 2017)
- [15] Toggle, 2017. Toggle. <https://toggl.com/> (Besøkt mai 2017)
- [16] Federal Aviation Administration, 2017. <https://www.faa.gov/> (Besøkt mai 2017)
- [17] LaTeX – A document preparation system, 2017. <https://www.latex-project.org/> (Besøkt mai 2017)

Remote NDT – Secure Data

Vedlegg

Vedlegg A

Terminologi

NDT Ikke-ødeleggende utprøving^[1] (engelsk: Non-Destructive testing) er en metode brukt for å påvise feil i materialer, uten å redusere prøveobjektets nåværende fysiske egenskaper.

Remote NDT Beskriver en metode for å utføre en NDT uten å fysisk være tilstede ved prøveobjektet.

DolphiCam Ultralydkameraet som DolphiTech leverer for å utføre en NDT.

Operatør / Operator Personen som fysisk holder DolphiCam og setter opp en inspeksjon. Operatør er rollen personen automatisk får tildelt i portalen.

Ekspert / Expert En rolle en person kan få tildelt i portalen. Rollen gir ekstra tilganger i portalen.

Gjest / Guest En rolle alle nye deltagere har i portalen.

Deltaker En fellesnevning for alle rollene (operatør, ekspert, gjest).

BlackBox Egendefinert PC som kjører programvaren som kommuniserer med DolphiCam og sender data til Web server.

DolphiLib Et C bibliotek som DolphiTech har utviklet. Håndterer kommunikasjonen mellom DolphiCam og vår programvare.

Systemet Beskriver alle deler av systemet. DolphiCam, BlackBox, server og klient.

Server Serveren som håndterer kommunikasjonen mellom BlackBox og klienter.

Klient / client Dette er en enhet med nettilgang som er brukt for å logge inn i portalen. Alle deltagere må bruke en klient for å bli med i en inspeksjon.

Innstillinger / Settings Refererer til innstillinger i DolphiCam.

Inspeksjon En inspeksjon beskriver når DolphiCam streamer data og deltager(e) kan se det i en nettside.

Prøveobjektet Refererer til materialet DolphiCam inspiserer. Dette kan f.eks være et flyskrog eller en bildel.

Token Token eller engangskode er en kode brukt for deltagere til å logge inn på nettsiden.

Vedlegg B

Prosjektavtale

PROSJEKTAVTALE

mellom NTNU v/Avd. Informatikk og Medieteknikk (NTNU/AIMT) (utdanningsinstitusjon), og

YNGVE RAUDBERGET DOLPHITECH

(oppdragsgiver), og

Kristoffer Klingenberg

Rolf Arne Myraumet

(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 9/1-17 til 16/5-17.

Studentene skal i denne perioden følge en oppsatt fremdriftsplan der AIMT 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 AIMT 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/fax, reiser og nødvendig overnatting på steder langt fra Gjøvik/AIMT. 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. AIMT 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 faglærer/veileder og sensor (intern og ekstern sensor). Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.
4. Alle bacheloroppgaver som ikke er klausulert og hvor forfatteren(e) har gitt sitt samtykke til publisering, kan gjøres tilgjengelig via NTNUs institusjonelle arkiv hvis de har skriftlig karakter A, B eller C.

Tilgjengeliggjøring i det åpen 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

Norges teknisk-naturvitenskapelige universitet

prosjektavtalen, og må evt. gi skriftlig melding til studenter og dekan 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 AIMT 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/AIMT og/eller studenter har interesser.

6. 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.
7. 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.
8. Studenten(e) leverer oppgavebesvarelsen med vedlegg (pdf) i Fronter. I tillegg leveres et eksemplar til oppdragsgiver.
9. Denne avtalen utferdiges med et eksemplar til hver av partene. På vegne av AIMT er det dekan/prodekan som godkjenner avtalen.
10. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og AIMT 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 AIMT som partner.
11. Når NTNU/AIMT også opptre som oppdragsgiver, trer NTNU/AIMT inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.
12. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene i mellom. 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.
13. Deltakende personer ved prosjektgjennomføringen:

NTNU/AIMTs veileder (navn): Ivar Farup

Oppdragsgivers kontaktperson (navn): Yngve Raudberget

Student(er) (signatur): Kristoffer Klingenberg dato 15/1-17

Rolf Arne Myravnset dato 15/1-17

_____ dato _____

_____ dato _____

Vedlegg C

Grupperegler

- §1. Alle gruppens medlemmer skal møte til faste avtaler der tidspunkt er avtalt på forhånd.
- §2. Eventuelle kostnader i følge med prosjektarbeidet skal deles likt mellom gruppens medlemmer.
- §3. Dokumentasjon av kode skal følge retningslinjer som er bestemt internt i gruppa og/eller av oppdragsgiver.
- §4. Dersom et medlem ikke utfører tildelt arbeidsoppgaver innen avtalt tid, skal dette meldes ifra om snarest mulig.
- §5. Gruppa har i utgangspunktet ingen fast arbeidstid, med mindre annet er avtalt på forhånd. Enkeltmedlemmer har selv ansvar for å utføre oppgaver innen gitt frist.

Signaturer

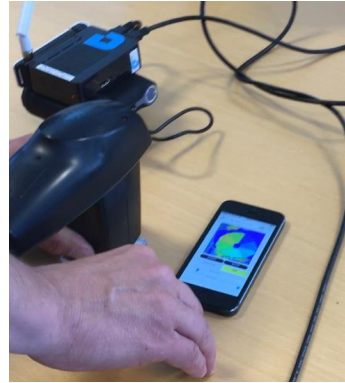
Rolf Arne Myraunet

Kristoffer Klingenberg

Vedlegg D

Oppgave

Oppdragsgiver: DolphiTech AS
Kontaktperson: Yngve Raudberget
Adresse: Enggata 40, 2830 Raufoss
Telefon: +47 92654040
E-post: yngve@dolphitech.com
Web: www.dolphitech.com



Last years project: DolphiCam image on a phone

DolphiTech

DolphiTech provides an advanced ultrasound camera, DolphiCam, for non-destructive-testing (NDT) of materials like CFRP (Carbon Fiber Reinforced Plastic). DolphiTech has developed a novel ultrasound transducer design that produces high-resolution images. We utilize an advanced technology developed by a team of experts in ultrasound, analog and digital electronics, firmware and software. We also have our own lab facilities to do research and development on electronics and transducer technology. Our main application areas are within aerospace, automotive, marine, and energy.

Project: Remote NDT – Secure Data

The future of NDT is pointing towards Remote NDT. That means that an NDT Expert can control NDT equipment and analyze live NDT data from an NDT non-expert (operator) over the internet.

We want to expand upon last year's bachelor project "DolphiCam Web Client". The outcome of that project was an html page showing images from DolphiCam Ultrasound Camera. The Web page was hosted on an Apache server running on an Odroid mini PC under linux OS. The DolphiCam was connected to the Odroid through USB.

Further development on this project will have emphasis on video, data and sound communication between web clients. We have suggested WebRTC as the technology for this, but if the students have alternatives that fill our requirements, we are open for that. Not all tasks needs to be completed for successful completion of the project and we can adapt the content of the project to the students wishes and skills. But our current prioritized list is as follows:

Minimum requirements for the project:

- Use WebRTC to set up a secure client-to-client connection between two or more web clients.
- Use WebRTC to send live video data (from Web Cam/Phone Camera), sound (speech), streaming DolphiCam images/video and Text (Chat) between clients.



- Sound, video and text should be two way for all client roles, DolphiCam images is sent only from the operator holding the physical DolphiCam camera.
- Find the best way to send DolphiCam data stream: as video, a series of bitmaps, make it appear as a web cam(may require driver development/implementation)
- Design the web page in an appropriate way – how to display/organize all information on a small phone screen?
- Separate between different Roles: Expert, operator and visitor
 - Operator and Expert can control the camera and view Live data
 - Visitor can only view DolphiCam data but can still communicate through video, sound and text.
- Only the user/operator that holds the physical DolphiCam should be able to start a session and create a logon token.
- Create a simple Portal for logon – use the token generated by the operator to log on to the operators session. Should find a way to create both expert and visitor tokens, or separate the roles otherwise.
- Control as many settings on the DolphiCam as possible through the Web page (a subset of settings is already implemented).

Additional tasks if time permits:

- Add an additional web cam to the operator role, so that the expert/visitors can see both a close up of the operator and an overview camera showing the part under inspection in a larger context.
- DolphiCam web server must be reachable on public internet.
- Possible to save session data, including image stream and chat/comments to a database for later lookup or reporting.

Technical tasks in the project (will or may) include

- Web programming
 - HTML5
 - JavaScript
 - Client-to-client audio, video and data communication (WebRTC)
- Web Security
- Integration with a backend C/C++ SDK from web for DolphiCam communication
- C++ development
- Linux and Apache configuration on Odroid

The students will have access to the code base of last year's project, A DolphiCam, an Odroid configured with Linux and Apache and an the DolphiCam SDK, a library written in C++ for communication with DolphiCam.

Vedlegg E

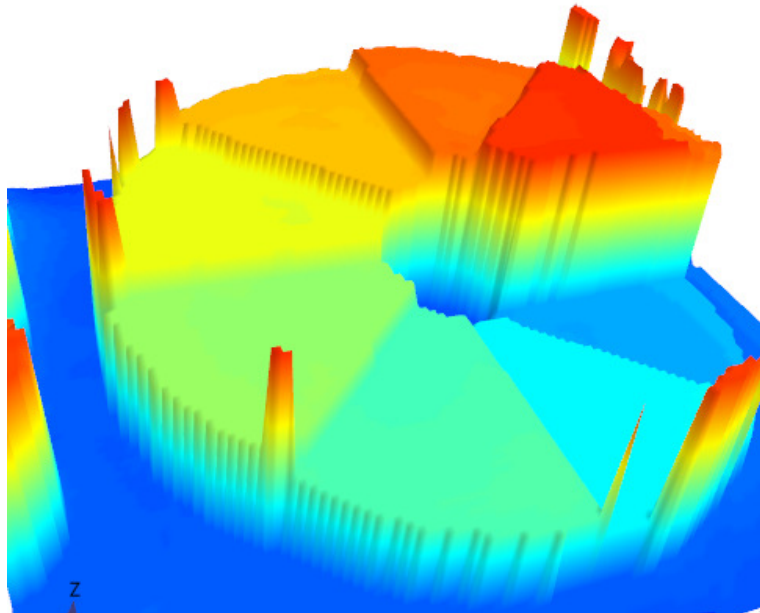
Prosjektplan

Secure NDT – Remote Data

Kristoffer Klingenberg

Rolf Arne Myraunet

28.01.2017



Innhold

1	INNLEDNING	4
1.1	Bakgrunn	4
1.2	Prosjekt mål	4
1.2.1	Resultat mål	4
1.2.2	Effekt mål	5
1.2.3	Læringsutbytte	5
1.3	Avgrensning	5
2	OMFANG	6
2.1	Fagområde	6
2.2	Oppgavebeskrivelse	6
3	PROSJEKTORGANISERING	7
3.1	Ansvar og roller	7
3.2	Gruppregler og rutiner	8
4	PLANLEGGING, OPPFØLGING OG RAPPORTERING	9
4.1	Valg av utviklingsmodell	9
4.2	Anvendelse av systemutviklingsmodell	9
4.3	Plan for statusmøter	10
5	ORGANISERING AV KVALITETSIKRING	11
5.1	Dokumentasjon, standardbruk og kildekode	11
5.2	Konfigurasjonsstyring	12
5.3	Risikoanalyse	12
6	PLAN FOR GJENNOMFØRING	14
6.1	Milepæler	14

1 INNLEDNING

1.1 Bakgrunn

DolphiTech er et selskap som utvikler og leverer et ultralydssystem (DolphiCam) for å detektere skader i ulike materialer som feks. karbonfiberplater. Systemet nyttes i fly- og automobilbransjen for å finne skader samt kvalitetsikre materialer. En slik testprosess kalles for ikke-ødeleggende utprøving (none-destructive test, eller NDT).

Spesielt i flybransjen stilles det høye krav til kvalitetssikring. Det kreves en ekspert med minst tre tusen timer opplæring for å utføre en NDT. Løsningen som DolphiTech bruker i dag er at kun kvalifiserte brukere av DolphiCam kan utføre en inspeksjon. Denne løsningen er ikke optimal da eksperter må selv reise til inspeksjonsområdet, utføre inspeksjonen, for å så dra til neste området.

Grunnet dette kravet ønsker DolphiTech og utvikle en løsning der en slik test kan utføres eksternt med en operatør som håndterer utstyret lokalt mens eksperten kan fjernanalysere data fra inspeksjonen. En slik løsning er bærekraftig da det kan spare store mengder tid, penger og resusser som inngår i å transportere eksperter mellom ulike inspeksjonsområder.

1.2 Prosjektmål

En NDT inspeksjon består av minst to deltakere. Personen som opererer DolphiCam kalles for en operatør. En faglig ekspert vil analysere ultralydbildene og omgivelsene rundt prøveobjektet for å komme til en konklusjon om prøveobjektets tilstand. I noen tilfeller ønsker en å involvere flere eksperter, men også andre deltakere, f.eks personer under opplæring. Sistnevnte blir gjerne kalt for gjester.

Målet med prosjektet er å muliggjøre utførelse av en slik inspeksjon uten at eksperter eller gjester trenger å være fysisk til stedet. Dette skal oppnås ved hjelp av en portal som lar brukere koble seg opp via Internett til å få tilgang til ultralydbilder og webkamera som viser omgivelsene rundt inspeksjonen. Deltakere skal også kunne kommunisere vha. lyd, bilde og direkte meldinger.

1.2.1 Resultatmål

- Kun operatøren kan invitere deltakere til inspeksjonen.
- Kun en operatør skal kunne starte inspeksjon og dele inspeksjonskamera og DolphiCam bilde.
- Eksperter og gjester skal kunne følge med på en inspeksjon uansett hvor de befinner seg.
- Ekspert skal kunne se og styre webkamera og DolphiCam innstillinger tilknyttet operatøren.

- Gjest skal kunne delta med rettighet til å se webkamera og DolphiCam bilde.
- Alle deltakere skal kunne velge å dele eget webkamera og/eller lyd samt kommunisere med direktemeldinger.

1.2.2 Effektmål

- Bidra til å skape bærekraft i NDT prosessen
- Effektivisere prosessen
- Redusere transportkostnader og tidsbruk
- Forbedre opplæringsmuligheter innen NDT
- Skape positiv markedsføringseffekt for DolphiTech
- Gjøre DolphiTech mer konkurransedyktig ovenfor andre lignende leverandører

1.2.3 Læringsutbytte

Gjennom prosjektets varighet vil vi få ferdigheter i å utarbeide konkrete problemstillinger og identifisere relevant litteratur for denne problemstillingen. Vi vil også få erfaring med prosjektarbeid i gruppe der det er flere parter.

I tillegg til emnets læringsmål kommer vi til å få erfaring innen flere tekniske fagfelt som omfatter blant annet ultralydteknologi, programvareutvikling og database for å nevne noen. Gjennom arbeidet skal det brukes systemutviklingsmodellen Scrum, som vil gi oss erfaring med arbeidsflyten spesifikt for denne utviklingsmodellen. Vi vil også få erfaring med å jobbe med en reell oppdragsgiver. Dette blir nytt fra tidligere emner.

1.3 Avgrensning

Dette prosjektet er avgrenset til implementasjon mot DolphiTech sitt eksisterende produkt DolphiCam, programvare og standarder som DolphiTech har utviklet i forbindelse med dette produktet. Videre vil vi holde oss til de rammer som er satt i oppgavebeskrivelsen gitt av DolphiTech og rammer som er blitt satt på grunn av hva og hvordan DolphiTech sin programvare, API og SDK er utviklet.

2 OMFANG

2.1 Fagområde

En NDT innebærer flere metoder for å finne feil eller svakheter på materialer uten å påføre skader eller redusere materialets fysiske egenskaper. Ikke-ødeleggende utprøving utføres ofte ved bruk av ultralydteknologi, der en transduser plasseres på materialets overflate. Transduseren sender lydbølger gjennom materialet. Lydbølgene reflekteres tilbake når de møter på et materialet med en annen akustisk impedans. Ved å måle tiden som brukes på å reflektere lydbølgene kan ultralydapparatet bestemme hvor langt inn i materialet denne endringen forekommer. DolphiTech har utviklet programvare som behandler signalene fra transduseren og omformer disse til data som kan visualiseres. Denne metoden gjør det mulig å skape et bilde av hvordan materialet ser ut på innsiden, og dermed inspisere potensielle feil eller skader.

Flybransjen er hyppig bruker av denne typen teknologi, da kvalitetsikring av flyets tilstand er viktig. Det stilles også høyt krav til kompetanse og opplæring for å bestemme om prøveobjektets nåværende tilstand er godkjent til flyving. På grunn av dette bruker flybransjen mye resusser på å transportere eksperter innen fagfeltet mellom ulike flyplasser der en slik test skal gjennomføres.

2.2 Oppgavebeskrivelse

Oppgaven går ut på å bygge en portal som lar en operatør dele en NDT økt utført med DolphiCam med andre over Internett. Portalen skal fungere som et slags lobby der operatøren setter opp en økt og deler ultralydbilder fra DolphiCam samt webkamera som viser videostrøm av området som skal inspiseres. Andre brukere skal kunne bli invitert til økta av operatøren. Disse brukerne er typisk ekspert eller gjest, og vil få tilgang til å utføre ulike funksjoner i portalen. En ekspert skal ha tilgang til å endre innstillingene på DolphiCam som operatøren holder fysisk. Muligheten til å styre rotasjon og zoom på web-kameraene skal også være mulig. Gjester skal kunne delta uten å få tilgang til å gjøre endringer.

3 PROSJEKTORGANISERING

3.1 Ansvar og roller

Oppdragsgiver og veileder

Vår oppdragsgiver er DolphiTech AS, der Yngve Raudberget er kontaktpersonen. Yngve vil være den vi forholder oss til når vi har teoretiske spørsmål angående applikasjonen. DolphiTech bidrar med alt utstyr som trengs for å utvikle applikasjonen. Vår veileder er Ivar Farup (professor, NTNU). Ivar vil bidra med veiledning gjennom hele prosjektets varighet både med rapportskrivning og ev. tekniske spørsmål.

Prosjektleder

Siden gruppa består av kun to medlemmer, ser vi liten fordel med å ha et gruppemedlem med overordnet ansvar. Prosjektlederrollen vil derfor bli delt mellom medlemmene i gruppa. Dersom det oppstår uenigheter vil det forsøkes å veie fordeler opp mot ulempene, for så å velge det som passer best for oppdragsgiver. Mest sannsynlig vil vi klare å komme til enighet.

Kontaktperson

Vår kontaktperson er Kristoffer Klingenberg. All formell kontakt mellom gruppa og eksterne parter vil skje gjennom han. Kristoffer vil holde kontakt med oppdragsgiver og veileder for å avtale møter eller diskutere spørsmål som oppstår underveis.

Delt ansvar

De områdene som ikke er spesifisert ovenfor, vil være et felles ansvar for begge gruppas medlemmer. Kodekvalitet, dokumentering og rapportskrivning, samt andre oppgaver som for eksempel skriving av møtereferat og timelogg vil begge ha felles ansvar for. Arbeidsoppgaver vil fordeles slik at de blir løst på en måte som tilfredsstillende oppgavens kravspesifikasjoner.

3.2 Grupperegler og rutiner

Et sett med grupperegler (vedlegg A) som gruppa har blitt enige om har blitt signert av gruppas medlemmer. I tillegg til grupperegler har gruppa definert følgende rutiner:

- All kode som er skrevet i en sprint skal være ferdig dokumentert i det sprinten avsluttes.
- Referat av møter skrives underveis og sendes til alle parter om de ønsker det.
- Rapporten blir skrevet på underveis i prosjektet, alle beslutninger skal noteres så fort som mulig.
- Timebruk logges fortløpende etter hvert som oppgaver utføres.

4 PLANLEGGING, OPPFØLGING OG RAPPORTERING

4.1 Valg av utviklingsmodell

Valg av utviklingsmodell baseres på modellen som passer best for gruppa, oppgaven og oppdragsgiver. Prosjektet har flere sentrale milepæler, som hver går et skritt nærmere mot et ferdig produkt. Enkelte deler av prosjektet er nødvendige for å resultere i et fungerende produkt, samtidig regnes noe som ekstra funksjonalitet. Systemet skal deles inn i flere moduler, der hver modul skal utvikles parallelt med mulighet for implementasjon av fremtidige moduler.

Begge av gruppas medlemmer har erfaring med Scrum fra tidligere prosjekter. Scrum-modellen passer godt i prosjekt der endringer kan forekomme underveis, noe det her er stor sjanse for. Prosjektet har et fleksibelt omfang og derfor er muligheten til å utvide produktets funksjonalitet gjennom en inkrementell utviklingsform optimalt. Derfor er det besluttet å bruke Scrum i prosjektet. DolphiTech har stilt seg positive til dette valget. De vil også i stor grad være tilgjengelig i løpet av prosjektet. Dette er en viktig faktor siden Scrum krever en aktiv oppdragsgiver.

4.2 Anvendelse av systemutviklingsmodell

Sprinter på kun en uke blir for korte og for mye å gjøre med kun to utviklere, i sammenligning med alle møter vi må ha i gruppa og sammen med DolphiTech. Samtidig vil sprinter på over to uker bli for lang tid siden oppgaven i stor grad er inkrementell. Derfor har vi valg at sprinter skal ha en varighet for på to uker. Vi forholder oss til ca 15 minutter daglig med daily scrum, men vil vanligvis jobbe sammen i størsteparten av utviklingsprosessen. Arbeidsoppgaver som det kreves ekspertise fra DolphiTech vil vi rådføre oss med de før vi begynner på disse oppgavene. Det vil bli tatt utgangspunkt i Scrum gjennom hele prosjektet, men vi vil gjøre egne tilpasninger ved behov. Under utvikling av kritiske deler vil enkelte aspekt fra XP (eXtreme programming) bli brukt. Det vil også blir gjort følgende avvik fra tradisjonell Scrum:

Scrum master-rollen vil bli delt av begge på gruppa. Begge vil aktivt fylle opp product backlog forløpende gjennom prosjektet. Denne metoden passer bra da en dedikert Scrum master blir et for stort ansvar for en person, med tanke på gruppas størrelse.

Verifisering og validering

Etter at hver user story er gjennomført vil den bli flytta til testfase. Der vil utvikler teste at alle kravspesifikasjoner er møtt og at ikke andre deler av programvare blir påvirket at implementasjonen. Etter at den har gått gjennom testfasen går story'en videre til valideringsfasen. Her vil en annen

utvikler kvalitetsikre koden. Til slutt implementeres funksjonaliteten som tilhører den aktuelle user story inn i produktet.

4.3 Plan for statusmøter

Hver sprint vil starte med et sprint planning event, her vil det velges hva som skal bli gjort. En sprint backlog blir produsert.

I slutten av hver sprint skal det holdes et sprint review meeting. Her gjennomgås alle planlagte arbeidsoppgaver, både utførte og ev. oppgaver det ikke var tilstrekkelig tid til. Det vises også en demo for oppdragsgiver over utført arbeid. I tillegg skal det holdes et sprint retrospective meeting der det skal reflekteres over hva som gikk bra og ikke. Dette vil bidra til å forbedre fremtidige sprinter. Både sprint- review og retrospective meeting er i praksis planlagt å foregå i samme møte som neste sprint sin planning event. Vi samler disse til et felles møte for å redusere antall møter gjennom prosjektet.

Med unntak av de møtene som inngår i Scrum vil gruppa holde overordnede statusmøter med oppdragsgiver for å evaluere utviklingsprosessen i sin helhet. Disse møtene skal holdes i det første planlagte møte i månedene mars, april og mai. Målet er å planlegge utviklingsprosessen på overordna nivå for å forsikre at det blir tilstrekkelig tid til alle prosjektets deler.

5 ORGANISERING AV KVALITETSIKRING

5.1 Dokumentasjon, standardbruk og kildekode

Dokumentasjon

Datastrukturer og kommunikasjonsflyt skal dokumenteres ved hjelp av relevant type skjema og tabeller. Det skal lages en skisse av disse før hver implementasjon for så å ferdigstille skissen dersom vi mener løsningen gir et godt grunnlag til å bygge videre på. Programkode skal dokumenteres på overordnet nivå som klasser og offentlige funksjoner. Det skal brukes docblocks standarden der dette er relevant.

Standardbruk

Gruppas medlemmer vil forsøke å følge standarder for syntakse og dokumentasjon av kode så langt det lar seg gjøre. I de tilfeller der DolphiTech har egne standarder skal vi følge disse.

Programmeringspråk

Utover prosjektet vil følgende språk brukes:

- PHP
- JavaScript
- HTML5/CSS
- C/C++

Serverens webløsning vil basere seg på bruk av PHP på serverside. Python og .NET er også alternative språk. Basert på gruppemedlemmene sine tidligere erfaringer har vi besluttet å basere oss på PHP. JavaScript vil bli brukt både for klient- og serverside logikk. Utforming av portalens grafiske layout blir gjort vha. HTML5 mens CSS3 brukes for å legge på stilsett. Det vil ikke legges stor vekt på å dokumentere kode brukt i forbindelse med layout eller design, da design ikke er oppgavens primærfokus. C++ skal brukes til utvikling av programvare som kjører lokalt på maskinvare hos operatør. Denne programvaren skal kobles opp mot DolphiCam over USB for å lese ut relevante data som så skal overføres til server.

5.2 Konfigurasjonsstyring

Prosjektets kode skal kontrolleres ved hjelp av versjonskontrollprogramvaren GIT. Vi skal sette opp repository i BitBucket og la all kode organiseres der. På den måten holder vi oversikt over alle endringer som er gjort i koden, og hvilke oppgave hver av endringene hører til. De ulike kodebasene vil fordeles i ulike repositories.

5.3 Risikoanalyse

I et slikt prosjekt finnes det mange generelle risikoer som sykdom og frafall. Vi vil forholde oss til å analysere risikoer som er spesifikt knyttet til dette prosjektet. Både sannsynligheten og konsekvensen for hver risiko ble funnet ut ved en diskusjon der begge la fram sine argumenter. For så å estimere hver for seg, i form av et tall og hvis tallene var ulike begynte diskusjonen på nytt.

Tabell 1 viser risikoer i forbindelse med prosjektet og kan leses på følgende måte: Sannsynlighet -og konsekvenskolonnen er rangert fra 1 til 10, der 10 gir størst sannsynlighet/konsekvens. Produktkolonnen viser produktet av de to foregående kolonnene og viser hvor stor virkning det har på prosjektet. Tabell 2 viser tiltak for risikoer med produktet i tabell 1 er mer eller lik 20.

Tabell 1: Risikoer

Nr	Beskrivelse	Sannsynlighet	Konsekvens	Produkt
1	Produkt ikke ferdig i tide	4	9	36
2	Server eller utstyr slutter å fungere	2	5	10
3	Server eller nettverk har ikke tilstrekkelig kapasitet	3	4	12
4	Har ikke nødvendig programvare/teknologi	1	7	7
5	Får ikke tilgang til nødvendig fysisk utstyr	2	10	20
6	Mangler tilstrekkelig tid til å oppnå prosjektkrav	3	8	24
7	Implementasjon av kjernefunksjonalitet tar for lang tid	3	7	21
8	Nødvendig funksjon lar seg ikke implementere	3	6	18

Tabell 2: Tiltak

Nr	Tiltak
1	Halvveis ut i prosjektet vil det gjøres en gjennomgang av prosjektets fremdrift. Dersom det viser seg at det mangler for mye funksjonalitet til at produktet kan ferdigstilles innen tidsfristen, vil vi diskutere med oppdragsgiver for å prioritere de viktigste delene av oppgaven.
5	Tre uker før vi trenger utstyret vil vi høre med DolphiTech om tilgang til utstyret.
6	Seks uker før prosjektfristen skal funksjonalitet gjennomgås. Dersom det er for store mangler til at kravene kan nås innen fristen vil vi spørre DolphiTech om de ønsker å bidra med teknisk hjelp.
7	Om vi en uke før milepæl en skal være nådd ser vi en uke er alt for kort tid til å implementere funksjonaliteten, vil vi spørre DolphiTech om assistanse.

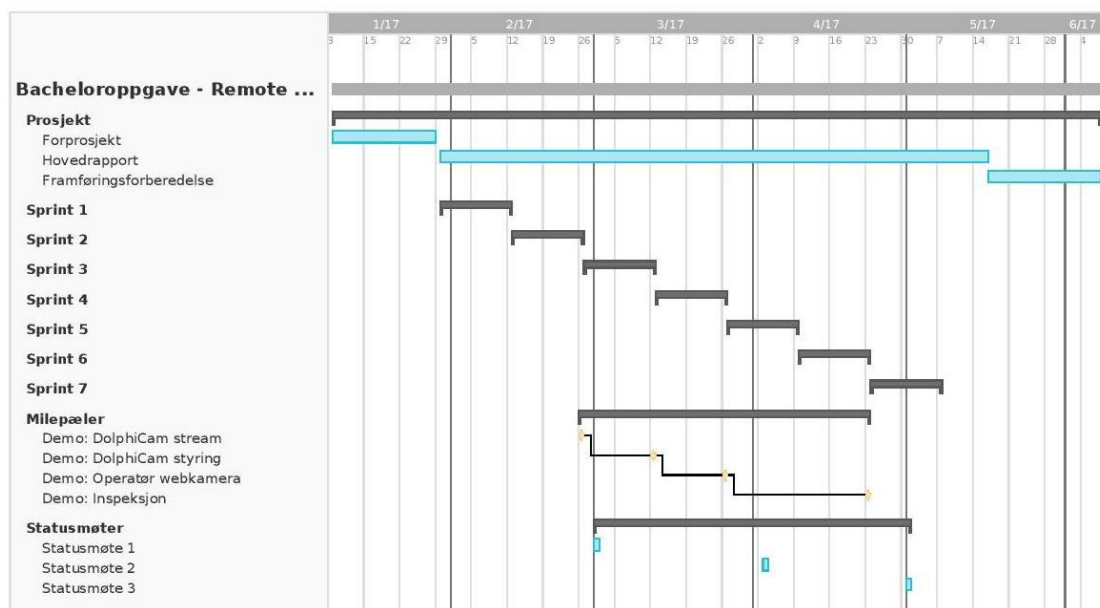
6 PLAN FOR GJENNOMFØRING

6.1 Milepæler

Milepæler bestemmes fra et punkt i utviklingsprosessen da en fungerende demo av en spesifikk sentral funksjonalitet kan presenteres for oppdragsgiver. Etter hvert som vi når de ulike milepælene vil det i førstkommende møte med oppdragsgiver bli presentert en demo. Nedenfor er en beskrivelse på hva som inngår i de ulike milepælene.

- **DolphiCam stream:** Operatør kobler opp et DolphiCam og starter en inspeksjon. Eksperter og gjester har så mulighet til å se inspeksjonen.
- **DolphiCam styring:** Eksperter skal kunne endre på DolphiCam sine innstillinger.
- **Operatør webkamera:** Deltakere skal kunne se på operatørens webkamera.
- **Inspeksjon:** Deltakere skal kunne kommunisere via direktemeldinger samt dele egne personlige webkamera med lyd.

Som nevnt i gruppereglene (vedlegg A) er det i utgangspunktet ikke satt faste arbeidstider. Vi vil også forholde oss til helgedager som vanlige arbeidsdager. Derfor går sprintene fra mandag til søndag. Figur 1 viser fremdriftsplanen i prosjektet i form av et Gantt-skjema



Figur 1: Fremdriftsplan

A Grupperegler

§1. Alle gruppens medlemmer skal møte til faste avtaler der tidspunkt er avtalt på forhånd.

§2. Eventuelle kostnader i følge med prosjektarbeidet skal deles likt mellom gruppens medlemmer.

§3. Dokumentasjon av kode skal følge retningslinjer som er bestemt internt i gruppa og/eller av oppdragsgiver.

§4. Dersom et medlem ikke utfører tildelt arbeidsoppgaver innen avtalt tid, skal dette meldes ifra om snarest mulig.

§5. Gruppa har i utgangspunktet ingen fast arbeidstid, med mindre annet er avtalt på forhånd. Enkeltmedlemmer har selv ansvar for å utføre oppgaver innen gitt frist.

Signaturer

Rolf Arne Myraunet

Kristoffer Klingenberg

Vedlegg F

Statusmøter

Møte sammendrag

Møte #6, del 2
Tema Statusmøte 1

Dato 13 Mars 2017
Lokasjon DolphiTech, Raufoss
Varighet 30 minutt

Deltakere Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Raudberget

- Sprint 1 til 3 er gjennomført.
- Hovedfunksjonalitet til systemet er implementert.
 - Vise ultralydbilder i en nettportal.
 - Endre DolphiCam innstillinger direkte fra portalen
 - Kommunikasjon over webkamera og lyd.
- Vi har kommet lengre enn det som var planlagt.
- Bytte av serverplass i London til Bergen da lokasjon av server en stor betydning for et funksjonelt system.
- DolphiTech er positiv til hvor langt vi har kommet i forhold til tiden vi har brukt.
 - Selve oppgavebeskrivelsen levert av DolphiTech er nesten gjennomført.
- Nå er neste fokus med å supplere hovedfunksjonaliteten.
-

Møte sammendrag

Møte #8, del 2
Tema Statusmøte 2

Dato 7 April 2017
Lokasjon DolphiTech, Raufoss
Varighet 30 minutt

Deltakere Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Raudberget

- Sprint 1 til 5 er gjennomført.
- Har ikke vært noe problemer i utviklingen.
- Har vært kontinuerlig fremgang i prosjektet og så langt ønsket funksjonalitet har blitt implementert.
- Milepæl 2, endre innstillinger i DolphiCam og milepæl 3, kommunikasjon med WebRTC over har blitt nådd og demo vist.
- DolphiTech er fornøyd med det som er gjort så langt.

Møte sammendrag

Møte	#10, del 2
Tema	Statusmøte 3
Dato	8 Mai 2017
Lokasjon	DolphiTech, Raufoss
Varighet	30 minutt
Deltakere	Kristoffer Klingenberg, Rolf Arne Myraunet

-
- Alle sprinter er gjennomført.
 - Mangler fortsatt en del dokumentasjon av kode.
 - Workshop 26 og 27 april ble suksessfullt gjennomført. Både gruppa og DolphiTech fikk godt utbytte av de.
 - Readme filer for innstallasjon av blackbox programvaren og server koden har blitt laget.
 - Siste oppsatt milepæl for demo av en full inspeksjon med all funksjonalitet gjennomført suksessfullt.
 - DolphiTech er generelt veldig fornøyd med systemet som har blitt utviklet.
 - Rapportskriving starter for fullt nå.

Vedlegg G

Møtereferat

Møte sammendrag

Møte	#4
Tema	Sprint 1 - Sprint planning meeting
Dato	2 Februar 2017
Lokasjon	DolphiTech, raufoss
Varighet	2 timer
Deltagere	Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Rundberget, Fredrik Lingvall

Sprint 1 - Sprint planning meeting

Gikk over de oppgaver vi skal gjennomføre i sprint 1.

Følgende oppgaver ble satt opp og/eller hentet fra produkt backlog:

- #1 Innlogging
- #2 Som operatør vil jeg starte en inspeksjon
- #3 Som operatør vil jeg streame DolphiCam data
- #4 Som ekspert/gjest vil jeg ha mulighet til å se på en stream

Målet med sprint 1 er å sende data fra DC til webserver og så opp til brukere som er tilkoblet den sesjonen.

Annet

Vi viste en demo av hvordan opplasting av bilde fra DC til visning på klient.

DC sender bilder på en FPS på ca 2.9 med CF8

DC_V1_B_SCAN_Intersection ← A-scan

Cast array om til char for å endre størrelsen

Send kun denne?

Generer bilde på klient?

DC fungerer best med carbonfiber

Hvordan lydbølger oppfører seg i forskjellige materialer

Plana de neste møtene framover, satt til annenhver mandag kl 12:00-14:00

Product backlog

Inches eller millimeter i dybde etc. på GUI

Møte sammendrag

Møte #	4
Tema	Sprint review, retrospective og planning meeting
Dato	13 Februar 2017
Lokasjon	DolphiTech, Raufoss
Varighet	1 time 30 minutt
Deltagere	Kristoffer Klingenberg, Rolf Arne Myraunet, Øystein Knauserud, Fredrik Lingvall

Sprint 1 review meeting

Det ble vist en demo av det vi har jobbet med i sprint 1.

Koblet DC til en PC og startet å les ut data fra DC. Mottok en token fra server som Øystein skrev inn på sin PC og fikk fulgt med på bildedata.

DolphiTech er fornøyd med det vi har fått til og er klar til å gå videre.

Sprint 1 retrospective meeting

Siden Yngve ikke var tilgjengelig på møte var det vanskelig å vite hvordan sprint 1 gikk fra DolphiTech sin side.

På vår side har vi brukt mest tid på å forstå A og- bscan data for å rendere det rett i portalen. I senere sprinter skal vi være litt tidligere ute med å ta kontakt med Yngve, slik at vi slipper å stå fast på et problem som vi vet de kan svare på.

Ellers har den kontakten vi har hatt med Yngve vært veldig bra, har fått utfyllende og raske svar.

Sprint 2 planning meeting

Hovedmålet for sprint 2 er å få satt innstillingene i DolphiCam som er viktigst, fra portalen.

Tilgangskontroll på brukerne, f.eks kun eksperter skal ha mulighet til å endre DolphiCam innstillinger.

Følgende user stories ble valgt ut:

#17 Som deltager vil jeg endre fargepallett

#5 Som operatør vil jeg invitere eksperter/gjester til inspeksjonen

#9 Som operatør/ekspert vil jeg sette DC-innstillingen B-scan horizontal og vertical linje i C-scan

#12 Som operatør/ekspert vil jeg sette DC-innstillingen upper og lower gate

#14 Som operatør/ekspert vil jeg sette DC-innstillingen gain/forsterkning

#15 Som operatør/ekspert vil jeg sette DC-innstillingen pulstype

#16 Som operatør/ekspert vil jeg sette DC-innstillingen TX-shoot

Møte sammendrag

Møte	#5
Tema	Sprint 2 review og retrospective meeting, sprint 3 planning meeting
Dato	27 Februar 2017
Lokasjon	DolphiTech, Raufoss
Varighet	2 timer
Deltakere	Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Raudberget, Øystein Knauserud, Fredrik Lingvall, Eskil Skoglund

Sprint 2 Review Meeting

Demo av det vi har gjort i sprint 2.

Operatør kan invitere eksperter og gjester direkte i portalen ved å generere en token.

Sette innstillinger direkte i portalen, se at innstillinger blir endret i visning og i webklienten til alle deltakere.

Sprint 2 Retrospective Meeting

Det har vært lite kontakt mellom oss og DolphiTech denne sprinten, men oppsatte user case's og krav har blitt utført slik som planlagt, DolphiTech var fornøyd med det som har blitt gjort.

Sprint 3 Planning Meeting

Fokuset i denne sprinten går på å forbedre ytelsen i nettleseren slik at det ikke oppstår problemer i klient når webkamera og oversiktsbilde implementeres. Etter at tilstrekkelige optimaliseringer er på plass, vil fokus skiftes til implementasjon av kommunikasjon via chat og lyd, samtidig som webkamera fra deltakere implementeres. Operatør skal ha mulighet til å bruke flere webkamera samtidig, der et av disse skal velges som "oversiktsbilde". Bilder fra dette kameraet skal settes i fokus sammen med scanvisningen.

Følgende user stories er planlagt:

#7 Som deltaker vil jeg dele mitt webkamera.

#19 Som operatør vil jeg dele flere webkameraer

#20 Som deltaker vil jeg snakke med andre deltakere med mikrofon.

#21 Som deltaker ønsker jeg bedre ytelse i portalen.

En rask estimering ble gjort, vi fant ut user storiene har stor nok arbeidsmengde.

Møte sammendrag

Møte	#6
Tema	Sprint 3 review og retrospective meeting, sprint 4 planning meeting
Dato	13 Mars 2017
Lokasjon	DolphiTech, Raufoss
Varighet	3 timer 30 minutt
Deltakere	Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Raudberget, Fredrik Lingvall, Eskil Skoglund, Jan Olav Endrerud

Sprint 2 Review Meeting

Demo av live webkamera og audio kommunikasjon mellom klienter vist. Noen oppstartsproblemer med at streamingen fungerte bare halvveis, litt usikker på hva grunnen kan ha vært. Etter en stund fikk alle opp sine egne og andres webkamera. Det var trolig noe med rettigheter i nettleser og evt restart av nettleser som var nødvendig.

Sprint 2 Retrospective Meeting

Har være lite kommunikasjon mellom oss og DolphiTech, har egentlig ikke vært behov for det. DolphiTech synes vi har utviklet i denne sprinten stemmer med det vi hadde planlagt og er fornøyd med det.

Sprint 3 Planning Meeting

Her skal vi ha fokus på tilganger til rette deltagere, DolphiTech ville at kun en person kan endre på DC innstillinger om gangen. Lagring av en inspeksjon er også i fokus, der bilder, chat og satte DC innstillinger blir lagret. Skal også være mulig å se en replay av gitt lagret inspeksjon.

Følgende user stories er planlagt:

- #22 Som ekspert/"operatør-ekspert" vil jeg være eneste som kan endre på DolphiCam innstillinger.
- #23 Som operatør vil jeg starte ny sesjon med midlertidige ekspertrettigheter.
- #24 Som ekspert vil jeg ha mulighet til å bytte rolle med gjest.
- #25 Som deltager vil jeg ha mulighet til å kollapse webcams
- #26 Som deltaker vil jeg lagre "current timeline" av DolphiCam bilder.
- #27 Som deltaker vil jeg se på en "saved timeline".
- #28 Som deltaker ønsker jeg å vite hvem som lager lyd eller prater.
- #29 Som "operatør-ekspert" vil jeg velge en initiell ekspert.
- #30 Som deltaker vil jeg ha eget panel for scans / webcams
- #31 Som deltaker vil jeg "poppe" ut webcams og scans til justerbart vindu

Møte sammendrag

Møte	#7
Tema	Sprint 4 review og retrospective meeting, sprint 5 planning meeting
Dato	27 Mars 2017
Lokasjon	DolphiTech, Raufoss
Varighet	3 timer
Deltakere	Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Raudberget, Jan Olav Endrerud

Sprint 4 review meeting

Hadde en liten bug når vi skulle starte demoen, på grunn av noe restrukturering av koden tidligere, men fant raskt en løsning slik at vi kom i gang med demoen.

Tilgangsendringen som vi gjorde ble vist, der kun en ekspert har mulighet til å endre på innstillinger som påvirker alle deltagere. Fungerte slik som DolphiTech hadde tenkt.

Lagring og visning av tidslinje fungerte, men DolphiTech ville gjøre noen endringer. I stedet for at x antall bilder tilbake i tid blir lagret, vil de at det skal kunne starte og så stoppe et opptak av bilder. En user story har blitt laget for neste sprint.

Sprint 4 retrospective meeting

Utviklinga gikk bra, ikke noe problemer.

Sprint 5 planning meeting

Fokus på flere deler denne sprinten.

Endring av tidslinje. Alle skal se når en tidslinje blir laget, men kun ekspert har mulighet til å lage/stoppe. Flere valg til å velge oppløsning på tidslinjen, der middels f.eks kan være at hver 4 bilde skal lagres.

Gjøre visningen av bildene mer funksjonsrik. Velge hvilken enhet bildene skal vises i, (millimeter, inches eller mils). Gjør de dynamisk ut ifra innstillinger satt i DolphiCam.

Møte sammendrag

Møte	#8
Tema	Sprint 5 review og retrospective meeting, sprint 6 planning meeting
Dato	7 April 2017
Lokasjon	DolphiTech, Raufoss
Varighet	2 timer
Deltakere	Kristoffer Klingenberg, Yngve Raudberget, Eskil Skoglund

Sprint 5 review meeting

Det ble vist en demo av det oppdaterte GUI'et som plasserer ulike UI komponenter i justerbare vindu. Videre ble det gitt en gjennomgang av diverse endringer i timeline funksjonalitet. Det ble ikke lagt mye vekt på demo siden det var lite ny funksjonalitet å vise frem. Dolphitech var positive til endringene som var gjort.

Det ble bestemt at B-scan horizontal og vertical settings skal fjernes fra GUI siden dette ikke er aktuelt for Dolphitech i første omgang.

Sprint 5 retrospective meeting

Sprint 6 planning meeting

Fokus på flere deler denne sprinten.

Muligheten for eksperter å lagre DC-settings lokalt i sin nettleser slik at de kan brukes senere.

Noen finpuss rundt setting av innstillinger, samt skal det kunne velges mellom low, medium eller high på TCG (time corrected gain).

Gate på B-scan vertical skal også visualiseres på bilde.

Møte sammendrag

Møte	#9
Tema	Sprint 6 review og retrospective meeting, sprint 7 planning meeting
Dato	24 April 2017
Lokasjon	DolphiTech, Raufoss
Varighet	2 timer 30 min
Deltakere	Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Raudberget, Eskil Skoglund, Jan Olav Endrerud, Fredrik Lingvall, Thomas Berg de Jose

Sprint 5 review meeting

Endringer på TCG gain innstilling, DolphiCam bilde blir oppdatert som det skal, men det må legges til funksjonalitet for å skru av TCG gain. Dette blir tatt med i neste sprint.

Snapshot, eksperten kan lagre gjeldende bilder, samt bilde av oversikt kamera i databasen. Alle deltakere har så mulighet til å se på alle snapshot. Fungerer slik som DolphiTech ønsket.

Lagre og importere innstillinger lokalt i nettleseren. Vi lagrer gjeldende innstillinger, endrer på et par innstillinger, laster inn siden på nytt og importerer de innstillingene igjen, ok. DolphiTech ønsker videre utvikling av funksjonen ved å også lagre i databasen, men det får vi ikke tid til og blir derfor ikke tatt med i neste sprint.

Vertikale linjer i B-scan vertical som blir oppdatert når gate innstilling blir endret, ser bra ut.

Sprint 5 retrospective meeting

Sprint 6 planning meeting

Ingen ny funksjonalitet skal inn i denne sprinten.

Nå blir det fokus på finpuss og fjerne bugs fra programvaren.

Vi skal også bidra med å lage en demonstrasjonsvideo slik at DolphiTech kan vise den fram på ei messe de skal på i Mai.

Readme filer til BlackBox og web-server skal lages og dokumentasjon av programvaren.

Problem med at når gain innstillingen blir endret, skjer det ingenting på bildene. Laget user story på å oppdatere TCG gain når gain innstilling blir endret.

Har vært litt problemer med hilbert funksjonen. Vi så litt mer på problemet under møte slik at de på DolphiTech kunne bidra til å komme fram til en løsning. Endte opp med at user storyen blir tatt med videre i neste sprint og vi venter med å gjøre noe på den til vi har hørt tilbake fra DolphiTech.

Møte sammendrag

Møte #10
Tema Sprint 7 review og retrospective meeting

Dato 8 Mail 2017
Lokasjon DolphiTech, Raufoss
Varighet 1 time

Deltakere Kristoffer Klingenberg, Rolf Arne Myraunet, Yngve Raudberget

Sprint 7 review meeting

På grunn av de 2 webshops vi hadde denne sprinten og det ikke skulle implementeres noe ny funksjonalitet, ble det ikke vist noe demo. Det var ikke noe nytt å vise demo av.

Sprint 7 retrospective meeting

Alle user stories ble ferdig.

Litt lite user stories, men ble jo enig på forhånd at det ikke skulle gjøres så mye denne sprinten.

Vedlegg H

Timelogg

Detailed report



2017-01-01 - 2017-12-31

Total 841 h 04 min

Kristoffer, Rolfarne selected as users

Remote NDT - Secure Data selected as projects

Blackbox, Møte DolphiTech, Møte Veileder and 4 more selected as tags

Date	Description	Duration	User
01-09	Oppstartsmøte med DolphiTech og Veileder DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech, Møte Veileder]	2:00:00 11:00-13:00	Kristoffer
01-09	Møte med DolphiTech, med veileder DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech, Møte Veileder]	2:00:00 11:00-13:00	Rolfarne
01-10	Oppsett av server DolphiTech AS - Remote NDT - Secure Data - [Server]	10:00:00 12:00-22:00	Rolfarne
01-10	Oppsett server DolphiTech AS - Remote NDT - Secure Data - [Server]	10:00:00 12:00-22:00	Kristoffer
01-11	Begynt koding av portal DolphiTech AS - Remote NDT - Secure Data - [Portal]	7:30:00 14:00-21:30	Kristoffer
01-11	Programmering portal DolphiTech AS - Remote NDT - Secure Data - [Portal]	7:30:00 14:00-21:30	Rolfarne
01-16	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:00:00 12:00-14:00	Kristoffer
01-16	DolphiTech Møte DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:00:00 12:00-14:00	Rolfarne
01-16	Forprosjekt skriving DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:00:00 14:00-21:00	Rolfarne
01-16	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:00:00 14:00-21:00	Kristoffer
01-18	Møte med ivar DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Rolfarne
01-18	Generell gjennomgang DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 13:00-14:00	Kristoffer
01-24	Gjennomgang av prosjektplan DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Kristoffer
01-24	Møte med ivar DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Rolfarne
01-24	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	6:00:00 13:00-19:00	Kristoffer

01-24	Forprosjekt skrivning DolphiTech AS - Remote NDT - Secure Data - [Rapport]	6:00:00 13:00-19:00	Rolfarne
01-25	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	8:00:00 14:00-22:00	Kristoffer
01-25	Forprosjekt skrivning DolphiTech AS - Remote NDT - Secure Data - [Rapport]	1:01:34 14:57-15:58	Rolfarne
01-25	Forprosjekt skrivning DolphiTech AS - Remote NDT - Secure Data - [Rapport]	0:00:06 16:06-16:06	Rolfarne
01-25	Forprosjekt skrivning DolphiTech AS - Remote NDT - Secure Data - [Rapport]	0:00:56 16:06-16:07	Rolfarne
01-25	Forprosjekt skrivning DolphiTech AS - Remote NDT - Secure Data - [Rapport]	5:57:34 16:08-22:06	Rolfarne
01-26	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	2:00:00 18:00-20:00	Kristoffer
01-27	Forprosjekt skrivning DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:21:34 14:58-19:20	Rolfarne
01-27	Rettskriving prosjektplan DolphiTech AS - Remote NDT - Secure Data - [Rapport]	2:00:00 20:00-22:00	Kristoffer
01-28	Forprosjekt skrivning DolphiTech AS - Remote NDT - Secure Data - [Rapport]	3:00:00 18:00-21:00	Rolfarne
01-28	Siste finpuss prosjektplan DolphiTech AS - Remote NDT - Secure Data - [Rapport]	2:50:00 21:00-23:50	Kristoffer
01-29	Oppsett hovedrapport DolphiTech AS - Remote NDT - Secure Data - [Rapport]	0:30:00 20:00-20:30	Kristoffer
01-30	DolphiCam SDK DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	8:10:00 12:20-20:30	Kristoffer
01-30	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	8:08:00 12:22-20:30	Rolfarne
02-02	Planlegging av utvikling DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:00:00 12:00-14:00	Kristoffer
02-02	Planlegge utvikling DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:00:00 12:00-14:00	Rolfarne
02-02	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	5:00:00 15:00-20:00	Kristoffer
02-02	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	5:00:00 15:00-20:00	Rolfarne
02-03	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	7:45:00 12:00-19:45	Rolfarne

02-03	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	7:45:00 12:00-19:45	Kristoffer
02-06	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	7:00:00 12:00-19:00	Rolfarne
02-06	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	7:00:00 12:00-19:00	Kristoffer
02-07	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	5:00:00 14:00-19:00	Kristoffer
02-07	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	3:00:00 14:00-17:00	Rolfarne
02-08	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	8:00:00 12:00-20:00	Rolfarne
02-08	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	8:00:00 12:00-20:00	Kristoffer
02-09	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	6:30:00 12:30-19:00	Kristoffer
02-09	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	6:30:00 12:37-19:07	Rolfarne
02-11	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	5:00:00 17:00-22:00	Rolfarne
02-11	Sprint 1 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	5:00:00 17:00-22:00	Kristoffer
02-13	Sprint review, retrospective and planning meeting DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	1:30:00 12:00-13:30	Rolfarne
02-13	Sprint review, retrospective og sprint planning meeting, sprint 2 DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	1:30:00 12:00-13:30	Kristoffer
02-13	Sprint 2 DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	7:00:00 14:30-21:30	Kristoffer
02-13	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	7:00:00 14:30-21:30	Rolfarne
02-14	Møte veileder DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Rolfarne
02-14	Demo DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Kristoffer
02-14	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	2:30:00 13:30-16:00	Kristoffer
02-14	Rapportskriving DolphiTech AS - Remote NDT - Secure Data - [Rapport]	2:30:00 13:30-16:00	Rolfarne

02-15	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	8:44:00 13:30-22:14	Rolfarne
02-15	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	8:30:00 13:30-22:00	Kristoffer
02-18	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	2:31:00 14:29-17:00	Rolfarne
02-19	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	2:50:19 16:33-19:23	Rolfarne
02-20	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	1:01:00 17:29-18:30	Rolfarne
02-21	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	1:35:05 14:13-15:48	Rolfarne
02-23	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	9:00:00 13:00-22:00	Kristoffer
02-23	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	8:01:00 14:01-22:02	Rolfarne
02-24	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	7:30:00 13:00-20:30	Kristoffer
02-27	Sprint 3 oppstart DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:00:00 12:00-14:00	Kristoffer
02-27	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:00:00 12:00-14:00	Rolfarne
02-28	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Kristoffer
02-28	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Rolfarne
03-01	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal, Rapport, Server]	9:00:00 13:00-22:00	Kristoffer
03-01	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal, Rapport, Server]	9:00:00 13:00-22:00	Rolfarne
03-02	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	9:00:00 12:30-21:30	Rolfarne
03-02	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	8:30:00 12:30-21:00	Kristoffer
03-06	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	6:00:00 19:00-01:00	Kristoffer
03-06	#7 DolphiTech AS - Remote NDT - Secure Data - [Portal]	6:00:00 19:00-01:00	Rolfarne

03-07	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	0:04:00 17:00-17:04	Rolfarne
03-07	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal, Server]	2:30:00 17:00-19:30	Kristoffer
03-07	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	2:26:00 17:04-19:30	Rolfarne
03-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	2:04:00 12:45-14:49	Rolfarne
03-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	8:15:00 12:45-21:00	Kristoffer
03-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	6:10:00 14:50-21:00	Rolfarne
03-10	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	6:00:00 17:00-23:00	Rolfarne
03-10	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	6:00:00 17:00-23:00	Kristoffer
03-12	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	2:20:15 14:40-17:00	Rolfarne
03-12	Gjennomgang demo DolphiTech AS - Remote NDT - Secure Data - [Portal]	1:00:00 18:45-19:45	Kristoffer
03-12	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	1:00:00 18:45-19:45	Rolfarne
03-13	Oppstart Sprint 4 DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	3:30:00 12:00-15:30	Kristoffer
03-13	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	3:30:00 12:00-15:30	Rolfarne
03-13	#22, #23, #24, #29 DolphiTech AS - Remote NDT - Secure Data - [Portal]	3:30:00 18:30-22:00	Kristoffer
03-13	#22, #23, #24, #29 DolphiTech AS - Remote NDT - Secure Data - [Portal]	3:24:35 18:36-22:00	Rolfarne
03-14	Møte med Ivar DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:15:00 12:00-13:15	Kristoffer
03-14	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:15:00 12:00-13:15	Rolfarne
03-14	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	4:00:00 14:00-18:00	Rolfarne
03-14	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:00:00 14:00-18:00	Kristoffer

03-17	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:00:00 16:00-20:00	Kristoffer
03-17	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:00:00 16:00-20:00	Rolfarne
03-21	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	3:00:00 16:15-19:15	Rolfarne
03-21	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	3:00:00 16:15-19:15	Kristoffer
03-22	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	5:45:00 15:30-21:15	Kristoffer
03-22	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	6:00:00 15:30-21:30	Rolfarne
03-23	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal, Server]	4:30:00 13:45-18:15	Kristoffer
03-23	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:30:00 13:45-18:15	Rolfarne
03-26	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	8:00:00 14:00-22:00	Rolfarne
03-26	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	8:00:00 14:00-22:00	Kristoffer
03-27	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	3:00:00 12:00-15:00	Kristoffer
03-27	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	3:00:00 12:00-15:00	Rolfarne
03-27	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:30:00 17:00-21:30	Kristoffer
03-27	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:30:00 17:00-21:30	Rolfarne
03-28	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Kristoffer
03-28	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:10:00 14:30-18:40	Kristoffer
03-28	#32 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	0:48:47 14:31-15:20	Rolfarne
03-28	#11 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	2:12:15 15:20-17:32	Rolfarne
03-28	#33 DolphiTech AS - Remote NDT - Secure Data - [Portal]	1:10:08 17:32-18:42	Rolfarne

03-29	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	3:30:00 18:30-22:00	Kristoffer
03-29	#11 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	3:23:12 18:31-21:55	Rolfarne
03-30	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	7:30:00 15:30-23:00	Kristoffer
03-30	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	7:30:00 15:30-23:00	Rolfarne
03-31	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:00:00 16:00-20:00	Kristoffer
03-31	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:00:00 16:00-20:00	Rolfarne
04-06	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:00:00 18:00-22:00	Kristoffer
04-07	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:00:00 12:00-14:00	Kristoffer
04-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	6:30:00 12:30-19:00	Rolfarne
04-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	6:30:00 12:30-19:00	Kristoffer
04-09	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	4:00:00 16:00-20:00	Kristoffer
04-09	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox, Portal]	4:00:00 16:00-20:00	Rolfarne
04-19	#45 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	5:00:00 14:00-19:00	Rolfarne
04-19	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	5:00:00 14:00-19:00	Kristoffer
04-20	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	3:00:00 13:00-16:00	Kristoffer
04-20	#45 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	3:00:00 13:00-16:00	Rolfarne
04-21	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	1:57:00 14:20-16:17	Rolfarne
04-21	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	6:40:00 14:20-21:00	Kristoffer
04-21	#11 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	4:43:00 16:17-21:00	Rolfarne

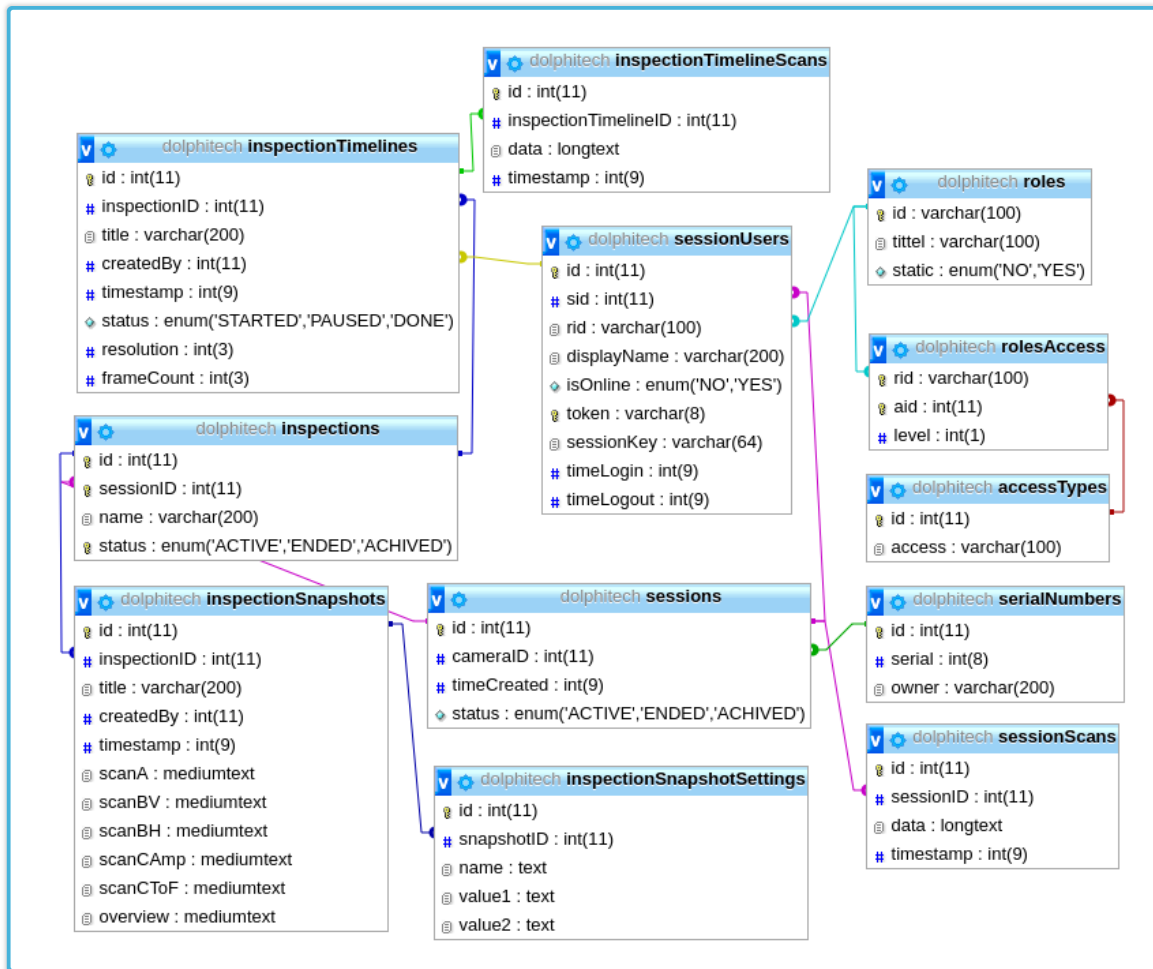
04-22	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	2:00:00 16:00-18:00	Rolfarne
04-22	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	5:30:00 16:00-21:30	Kristoffer
04-22	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	3:30:00 18:00-21:30	Rolfarne
04-24	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:30:00 12:00-14:30	Rolfarne
04-24	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	2:30:00 12:00-14:30	Kristoffer
04-24	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	2:00:00 16:00-18:00	Kristoffer
04-24	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	2:00:00 16:00-18:00	Rolfarne
04-25	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Kristoffer
04-25	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Rolfarne
04-25	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	6:30:00 13:30-20:00	Kristoffer
04-25	#48 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	0:49:49 14:27-15:17	Rolfarne
04-25	#48 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	0:03:22 15:17-15:21	Rolfarne
04-25	#49, #47 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	4:37:47 15:22-19:59	Rolfarne
04-26	Workshop DolphiTech AS - Remote NDT - Secure Data - [Workshop]	4:00:00 12:30-16:30	Kristoffer
04-26	Workshop DolphiTech AS - Remote NDT - Secure Data - [Workshop]	4:00:00 12:30-16:30	Rolfarne
04-26	#11 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	1:18:12 18:23-19:41	Rolfarne
04-27	Readme dt-client DolphiTech AS - Remote NDT - Secure Data - [Rapport]	1:00:00 00:43-01:43	Rolfarne
04-27	Workshop DolphiTech AS - Remote NDT - Secure Data - [Workshop]	7:30:00 09:00-16:30	Kristoffer
04-27	Workshop DolphiTech AS - Remote NDT - Secure Data - [Workshop]	7:30:00 09:00-16:30	Rolfarne
04-28	#11 DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	1:30:00 00:30-02:00	Rolfarne

04-28	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:00:00 15:00-22:00	Rolfarne
04-28	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:00:00 15:00-22:00	Kristoffer
04-28	(no description) DolphiTech AS - Remote NDT - Secure Data - [Portal]	1:30:00 21:30-23:00	Kristoffer
05-01	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	1:30:00 17:00-18:30	Kristoffer
05-01	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	1:00:00 17:00-18:00	Rolfarne
05-02	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Kristoffer
05-02	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Rolfarne
05-02	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:30:00 13:30-18:00	Kristoffer
05-02	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:30:00 13:30-18:00	Rolfarne
05-03	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:30:00 12:30-17:00	Rolfarne
05-03	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:30:00 12:30-17:00	Kristoffer
05-04	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:30:00 12:30-17:00	Kristoffer
05-04	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	4:30:00 12:30-17:00	Rolfarne
05-05	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	9:30:00 12:30-22:00	Kristoffer
05-05	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	9:30:00 12:30-22:00	Rolfarne
05-06	(no description) DolphiTech AS - Remote NDT - Secure Data - [Blackbox]	1:00:00 13:45-14:45	Rolfarne
05-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	1:15:00 12:00-13:15	Kristoffer
05-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte DolphiTech]	1:15:00 12:00-13:15	Rolfarne
05-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	6:15:00 13:45-20:00	Kristoffer

05-08	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	6:15:00 13:45-20:00	Rolfarne
05-09	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	0:00:00 12:00-12:00	Kristoffer
05-09	(no description) DolphiTech AS - Remote NDT - Secure Data - [Møte Veileder]	1:00:00 12:00-13:00	Rolfarne
05-09	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	9:00:00 13:00-22:00	Kristoffer
05-09	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	9:00:00 13:00-22:00	Rolfarne
05-10	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:30:00 12:00-19:30	Rolfarne
05-10	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:30:00 12:00-19:30	Kristoffer
05-11	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	10:00:00 13:00-23:00	Kristoffer
05-11	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	10:00:00 13:00-23:00	Rolfarne
05-12	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	9:45:00 14:00-23:45	Rolfarne
05-12	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	9:45:00 14:00-23:45	Kristoffer
05-13	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:45:00 18:00-01:45	Rolfarne
05-13	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	7:45:00 18:00-01:45	Kristoffer
05-14	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	11:30:00 15:30-03:00	Kristoffer
05-14	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	11:30:00 15:30-03:00	Rolfarne
05-15	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	13:00:00 17:00-06:00	Rolfarne
05-15	(no description) DolphiTech AS - Remote NDT - Secure Data - [Rapport]	13:00:00 17:00-06:00	Kristoffer

Vedlegg I

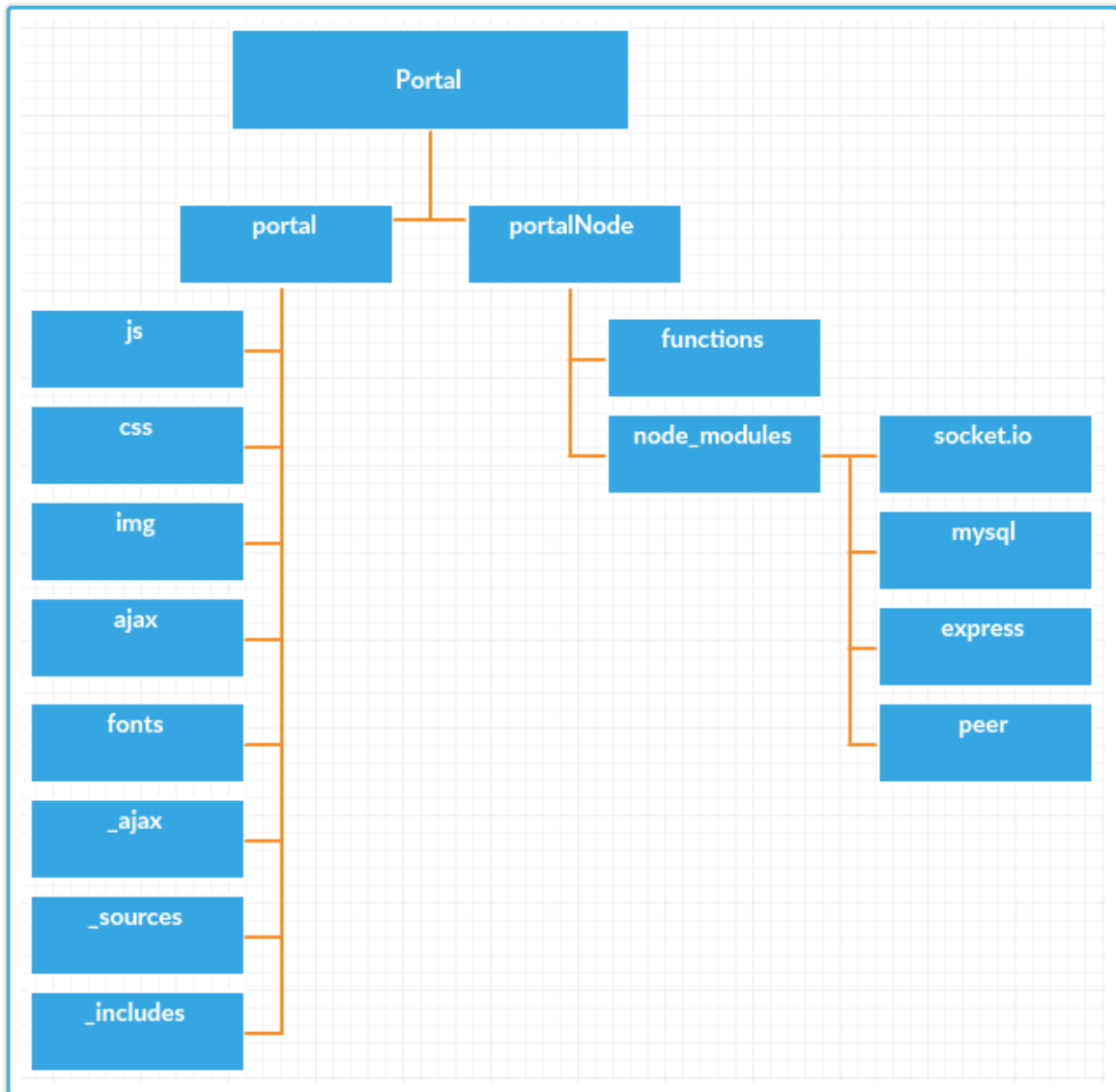
Databasediagram



Figur 36: Fullstendig UML skjema av databasen

Vedlegg J

Filstruktur på server



Vedlegg K

Klassestruktur på blackbox

Hver blå box representerer en klasse og strekene viser hvilken vei en klasse kan kommunisere med de andre.

Main Programmet starter å kjøre her. Lager tre objekt av klassene DolphiCamStreamer, DolphiCamCommunicator og SettingsUpdater henholdsvis.

DolphiCamCommunicator Håndterer all kommunikasjon med DolphiCam.

DolphiCamStreamer Les ut ultralyddata og sender til server.

SettingsUpdater Sender forespørsel til server om nye innstillinger og oppdaterer de.

StringCompressor Komprimerer en tekst streng.

Uploader Brukt for å sende data til server.

