

BACHELOROPPGAVE:

**HØYHASTIGHETSBREDBÅND TIL
ALLE I HEDMARK**

FORFATTERE:

JULIE KRISTIN THOWSEN PETTERSEN
LASSE SVILAND
INGVILD NÆSS

Dato: 16.05.2017

SAMMENDRAG

Tittel:	<u>Høyhastighetsbredbånd til alle i Hedmark</u>	Dato : 16.05.17
<hr/> <hr/>		
Deltaker(e)/	<u>Julie Kristin Thowsen Pettersen</u>	
	<u>Lasse Sviland</u>	
	<u>Ingvild Næss</u>	
Veileder(e):	<u>Frode Haug</u>	
Evt. oppdragsgiver:	<u>Hedmark fylkeskommune v/ Egil Langseth</u>	
Stikkord/nøkkelord (3-5 stk)	<u>bredbånd, app, Xamarin, Rails, Angular</u>	
Antall sider/ord:	<u>213/24845</u>	Antall vedlegg: <u>11</u>
		Publiseringsavtale inngått: <u>ja</u>
Kort beskrivelse av bacheloroppgaven:		
<p>Internett er en viktig ressurs i det norske samfunnet. Å bygge ut bredbåndstilbudet er en krevende prosess der flere aktører må samarbeide for å gjøre det så kostnadseffektivt som mulig. Norsk kommunikasjonsmyndighet(Nkom) ønsker at alle skal ha tilgang på høyhastighetsbredbånd og stiller opp med midler for utbyggingen. Kommunene kan få innvilget støtte fra Nkom dersom de kan dokumentere at de har et prosjektområde med behov for forbedring.</p> <p>Formålet med oppgaven er å lage et system som kan hjelpe Hedmark fylkeskommune og kommunene i Hedmark å kartlegge gode prosjektområder slik at de enklere kan få støtte fra Nkom for å bygge ut bredbåndstilbudet. Systemet skal samle inn informasjon om det nåværende bredbåndstilbudet og gi en oversikt over dagens situasjon slik at det blir enkelt å finne nye områder å bygge ut i.</p>		

ABSTRACT

Title:	<u>Høyhastighetsbredbånd til alle i Hedmark</u>	Date :	16.05.17
Participants/	<u>Julie Kristin Thowsen Pettersen</u>		
	<u>Lasse Sviland</u>		
	<u>Ingvild Næss</u>		
Supervisor(s)	<u>Frode Haug</u>		
Employer:	<u>Hedmark fylkeskommune v/ Egil Langseth</u>		
Keywords (3-5)	<u>broadband, app, Xamarin, Rails, Angular</u>		
Number of pages/words:	213/24845	Number of appendix:	11
		Availability:	open
Short description of the bachelor thesis:			
<p>Internet is an important resource in the norwegian society. Expanding the current broadband service is a challenging process where several contributors must cooperate to make it as cost-effective as possible. Norwegian Communications Authority(Nkom) wants high-speed broadband for everyone and they provide funding for the development of improved broadband capacity. The municipalities may be granted support from Nkom if they can prove that they have a project area in need of improvement.</p>			
<p>The purpose of the bachelor thesis is to create a system that can help Hedmark county and the municipalities in Hedmark to select good project areas so that they can more easily receive support from Nkom to expand the broadband capacity there. The system will collect information about the existing broadband capacity, give an overview of the current situation and make it easy to find new areas to expand.</p>			

Forord

Bacheloroppgaven «Høyhastighetsbreddband til alle i Hedmark» har blitt gjennomført av tre studenter ved Bachelor i Programvareutvikling ved NTNU i Gjøvik. Dette prosjektet har vært en positiv læringsopplevelse. Vi har fått testet kunnskap vi har fra tidligere i studiet, men også lært mye nytt som vi helt sikkert kommer til å ha nytte av etter studiet. Vi føler at dette prosjektet har vært en fin avslutning på studiet, og en spennende prøvelse før vi alle tre fortsetter videre ut i arbeidslivet.

Vår veileder har vært Frode Haug. Vi vil gjerne takke han for et godt samarbeid. Han har gitt oss mange nyttige råd og gode tilbakemeldinger gjennom hele prosjektperioden.

Vi vil også takke vår oppdragsgiver fra Hedmark fylkeskommune, Egil Langseth. Egil har vært en veldig engasjert oppdragsgiver med stor tro på oss og oppgaven. Gjennom prosjektet har han vært en god ressurs da prosessen rundt breddbandutbygging var ukjent for oss ved starten av prosjektet. Egil har også bidratt med gode tilbakemeldinger og ideer. Vi har gjennom hele prosjektet følt at denne oppgaven har vært en viktig prioritet for fylkeskommunen, og Egil har ofte vært tilgjengelig for møter og spørsmål.

Innhold

1 Innledning	1
1.1 Omfang	1
1.1.1 Fagområde	1
1.1.2 Avgrensning	2
1.1.3 Oppgavebeskrivelse	2
1.2 Formål	7
1.3 Målgruppe	8
1.3.1 Målgruppe for systemet	8
1.3.2 Målgruppe for rapporten	8
1.4 Egen bakgrunn og kompetanse	9
1.5 Rammer	9
1.5.1 Arbeidsmetoder	9
1.5.2 Prosjektorganisering	10
1.5.3 Plan for gjennomføring	12
1.5.4 Ytre føringer	13
1.5.5 Systemutviklingsmodell	14
1.5.6 Arbeidsverktøy	15
1.6 Organisering av rapporten	16
1.6.1 Avklaringer om rapporten	16
1.6.2 Kapitteloppsummering	16
2 Teoretisk grunnlag	18
2.1 Bredbånd	18

2.2	National Kommunikasjonsmyndighet(Nkom)	19
2.2.1	Hvitt område	20
2.2.2	NGA område	20
2.2.3	Søknadsprosessen	20
2.3	Hastighetsmåling	21
2.3.1	Ping — hva er det?	21
2.3.2	Ping — hvordan gjør vi det?	22
2.3.3	Nedlastingshastighet — hva er det?	22
2.3.4	Nedlastingshastighet — hvordan gjør vi det?	23
2.3.5	Opplastningshastighet — hva er det?	25
2.3.6	Opplastningshastighet — hvordan gjør vi det?	25
3	Kravspesifikasjon	26
3.1	Brukergrupper	26
3.1.1	Pact-analyse	26
3.1.2	Personas	28
3.2	Use Case	34
3.2.1	Aktører	34
3.2.2	Use Cases	35
3.2.3	Extended Use Case	39
3.3	Sikkerhet	42
4	Design og arkitektur	44
4.1	Teknologivalg	44
4.1.1	App	44
4.1.2	Frontend på nettsiden	46
4.1.3	Backend	46
4.1.4	Database	47
4.2	Systemarkitektur	48
4.3	Prototype	49
4.3.1	Lo-fi prototype	49

4.3.2	Hi-fi prototype	49
4.4	Fargeprofil	52
4.5	Universell utforming	54
4.5.1	Bilder	55
4.5.2	Skjema	56
4.5.3	Kontraster	60
5	Implementering	62
5.1	Frontend	62
5.1.1	App	62
5.1.2	Nettside	82
5.2	Backend	88
5.2.1	Database	88
5.2.2	API	96
6	Testing og kvalitetssikring	101
6.1	Testing av Backend	101
6.2	Style guide Backend	102
6.3	Testing av frontend-koden	103
6.4	Versjonskontroll	106
7	Avslutning	108
7.1	Resultater	108
7.1.1	Læringsutbytte	108
7.1.2	Resultatmål	109
7.1.3	Effektmål	109
7.1.4	Gantt diagram	110
7.2	Drøftinger og valg	111
7.3	Videre arbeid	112
7.4	Evaluering av gruppas arbeid	113
7.4.1	Innledning	113
7.4.2	Organisering	113

7.4.3	Fordeling av arbeidet	113
7.4.4	Prosjekt som arbeidsform	114
7.5	Konklusjon	114
Bibliografi		116
Ordliste		120
Vedlegg		
A	Skjermdump fra "http://speedtest1.telenor.net/ mini/speedtest/", 10.04.17	
B	Lo-fi prototype skisser	
C	Utdrag av HFK Profilmanual 2017	
D	Database Diagram, med modell koblinger	
E	Database Diagram, med uten modell koblinger	
F	Vurdering av å bytte til PHP som backend-språk	
G	Prosjektavtale	
H	Forprosjektrapport	
I	Statusrapporter	
J	Møtereferater	
K	Timelister	

Figurer og tabeller

Figurer

1.1 Organisasjonskart	10
2.1 Illustrasjon av ping	22
2.2 Aktivitetsdiagram ved testing av nedlastningshastighet	24
3.1 Personas: Elenor Martinsen	29
3.2 Personas: Johan Fredriksen	30
3.3 Personas: Aksel Overvik	31
3.4 Personas: Oskar Alstad	32
3.5 Personas: Katrine Olsen	33
3.6 Use Case Diagram	35
3.7 System sekvensdiagram for søknad	40
3.8 System sekvensdiagram for forespørre bedre bredbånd	43
4.1 Logisk inndeling av vårt system	48
4.2 Sammenligning av lo-fi og hi-fi prototype	50
4.3 Prototype: Forside	50
4.4 Prototype: Hastighetsmåling	50
4.5 Prototype: Resultater	51
4.6 Prototype: Forespørsel	51
4.7 Prototype: Forsiden	51
4.8 Prototype: Varmekart	51

4.9	Prototype: Pålogging fra forsiden	52
4.10	Prototype: Kommune-forside	52
4.11	Prototype: Prosjektet "Gaupen"	52
4.12	Prototype: Nytt prosjekt	52
4.13	Forside med hovedfargene	53
4.14	Bruk av fargevariasjoner	53
4.15	Varselfarger	54
4.16	Fargen "Vampyr" i bruk	54
4.17	Knapper med tekst	55
4.18	Alternativ tekst på knapper uten tekst	55
4.19	Forsiden, med CSS-kode	56
4.20	Tekstboks med ledetekst. HTML-kode som viser implementeringen	57
4.21	Skjema med veiledningstekst	58
4.22	Skjema med feilmelding til bruker	59
4.23	Innloggings-skjema med feilmelding til bruker	59
4.24	Måling av kontrastkravet på grønn knapp med stor tekst	60
4.25	Måling av kontrastkravet på rød knapp med liten tekst	61
4.26	Måling av kontrastkravet på meny med liten tekst og stort ikon	61
5.1	Hovedstruktur i app	62
5.2	Strukturen i fellesmappen	63
5.3	iOS filstruktur	64
5.4	Android filstruktur	64
5.5	Illustrasjon av Dependency Injection	66
5.6	Original progressbar på iOS, Android og Windows Phone [1]	68
5.7	iOS med tykk grønn bar	68
5.8	Android med tykk grønn bar	68
5.9	App: Informasjonsside	74
5.10	App: Godkjenning av rettigheter	74
5.11	App: Forside	76

5.12 App: Alert når man prøver å logge inn	76
5.13 Sammenheng mellom speedtest filer i app	77
5.14 App: Resultat-siden	79
5.15 App: Resultat-siden når man ikke får koblet seg til databasen	79
5.16 App: Forespørsel side 1	80
5.17 App: Forespørsel med navigasjonsbar	80
5.18 En komponent på nettsiden er markert	82
5.19 Filstruktur og metadata til en komponent	83
5.20 API-forespørsel til egen webserver	84
5.21 API-forespørsel til ekstern webserver	85
5.22 Registrerings skjema	85
5.23 Side for å godkjenne brukere	86
5.24 Prosjekt liste	86
5.25 Nytt prosjekt side	86
5.26 Prosjektside Statuser	87
5.27 Prosjektside Meldinger og Forespørsler	87
5.28 Prosjektside Kommuner og Brukere	87
5.29 Prosjektside Varslinger og Søknader	87
5.30 Prosjektside Kart og Områder	87
5.31 Prosjektside Søknadvurdering	87
5.32 Markering av adresser på google maps	91
5.33 Nærmere visning av hus i markering	91
5.34 Oversikt over modeller	92
5.35 Oversikt over modeller for brukere	93
5.36 Oversikt over modeller for hastighetsmålinger og forspørsler	94
5.37 Oversikt over for modeller Nkom søknad	94
5.38 Oversikt over modeller for prosjekter	95
5.39 Oversikt over modeller for adresser og områder	96
5.40 Oversikt over kontroller klasser	98

6.1 Dekning av rspec tester	102
6.2 Rubocop redundant slef	103
6.3 Rubocop too many lines	103
6.4 Filer som bli opprettet når man oppretter en komponent med Angular CLI	104
6.5 Skjema for å forespørre bedre nett	104
6.6 Resultatet av testene i konsollen	106
6.7 Resultatet av testene i nettleseren	106
6.8 Eksempel på to brancher med issue nummer HYH-146 og HYH-142	106

Tabeller

4.1 Sammenligning av Xamarin og React Native	45
--	----

1 *Innledning*

1.1 **Omfang**

1.1.1 **Fagområde**

Internett har gjennom de siste 20 årene hatt en eksplosiv utvikling og erstatter stadig flere og flere oppgaver i det daglige liv. Bruken av internett øker samtidig som bruken av papiraviser, ukeblader og fjernsyn reduseres[2]. Internett har blitt hovedkilden for informasjon, tjenester og kommunikasjon.

Internett krever utbygging av bredbånd. Tall fra SSB[3] viser at 84% av norske husstander har et fast grunnleggende bredbåndstilbud. Dette betyr at hele 16% av husstandene i Norge mangler eller har mindre enn 4Mbit nedlastningshastighet.

Det finnes mange forskjellige typer internett-tilkoblinger. De vanligste internett-tilkoblingene er analog, ISDN, ADSL, VDSL, parabol, coax, radio, fiber og 4G. Tilkoblingene gir forskjellig hastighet og tilbys ikke overalt. Dette gjør at enkelte deler av landet har et generelt bra tilbud, mens andre deler av landet ikke har det.

Å bedre bredbåndstilbudet er kostbart og krever at flere interessenter samarbeider. Da vil fylkeskommunen, kommunene og internettleverandører være involvert. Samtidig kreves det bidrag fra innbyggerne i kommunen. Bredbåndsløse leverandørene deler ikke kartdata om sin egen infrastruktur med andre, noe som gjør det vanskelig for de andre interessentene å få et reelt bilde av dagens situasjon. På den andre siden har ikke leverandørene kontroll på hvor det er ønskelig å bygge ut bedre bredbånd. Denne kommunikasjonsmuren gjør det vanskelig å kartelegge utbygningen.

1.1.2 Avgrensning

Vår oppgave omhandler å utvikle et verktøy for kommunene i Hedmark som skal brukes av kommunene. Dette skal gjøre det enklere for kommunene å bygge ut bredbånd i sitt område. Dette skal gjøres ved å samle inn mer detaljert og korrekt informasjon enn det som allerede finnes i dag angående nettverkshastighet. Dette vil kunne hjelpe kommunene å få en bedre oversikt over hvilke områder i kommunen det burde søkes om støtte til utbygging.

For å hjelpe kommunene med å få mer støtte fra [Nkom](#), skal vi utvikle et system som vil hjelpe kommunen med å lage bedre søknader. Dette skal vi gjøre ved å bidra med ytterligere informasjon angående dagens situasjon, og å kunne gi bedre tilbakemeldinger på hvor optimalt et prosjekt innfrir kravene Nkom stiller for å innvilge støtte for utbygging. Det er ikke forventet at hele systemet skal bli ferdiglagd i løpet av bacheloren, men at store deler av det viktigste skal være på plass.

1.1.3 Oppgavebeskrivelse

Oppgaven er delt i flere deler. En åpen del for alle innbyggere, som kan brukes til å måle brukers hastighet. En del der brukere kan logge inn for å få mer informasjon. Samt en del hvor kommuner, fylket og leverandører skal kunne legge inn informasjon og hente ut informasjon som trengs for å generere søknader om støtte for utbygning av bredbånd.

1.1.3.1 Roller

Leverandør

Leverandørene skal kunne se hvilke prosjekter kommunene har opprettet og ha mulighet til å legge inn deres [sjablongvurdering](#) (hva de regner med at det vil koste per husstand for å bygge ut i området). De skal kunne oppgi informasjon om hvilke planer de har for utbygging i dette området de neste tre årene, og hvor mye de kommer til å bidra med dersom de blir å bygge ut i et område.

Leverandørene skal også ha informasjon om alle målinger og forespørsler om bedre bredbånd. Dette kan hjelpe leverandørene med å finne ut hva kostnaden ved å bygge ut vil være i et bestemt område, og hvor mye de ønsker å bidra.

Kommune

Kommuner skal kunne opprette et prosjekt i et område hvor det ønskes å søke om stønad for utbygging. Dette gjøres ved hjelp av et kart hvor de skal kunne markere områder hvor det ønskes å bygge ut bredbånd. Markeringen skal kunne ha deler som ikke er en del av markeringen, slik at områder som allerede har bra nettverkstilkobling kan fjernes fra området.

Kommunen kan også legge inn hvor mye de ønsker å støtte prosjektet med per husstand, og se hva leverandør og fylkeskommune har sagt at de skal bidra med. Kommunen skal kunne legge inn informasjon på vegne av en leverandør, dette vil være informasjon om en leverandørs sjablong verdi, og dokumentasjon på om en leverandør kommer til å bygge ut i området.

Fylkeskommune

Fylkeskommunen kan legge inn informasjon på et prosjekt om hvor mye de ønsker å støtte prosjektet med.

Fylkeskommunen vil også måtte godkjenne de kommunene som registrerer seg for å benytte seg av verktøyet. De skal også kunne legge inn eller oppdatere informasjon på vegne av en kommune eller en leverandør.

Ildsjel

En ildsjel er en person som har bidratt spesielt til et prosjekt i et område og derfor skal ha tilgang til ytterligere informasjon. Ildsjelen kan ofte ha mer detaljert informasjon om området og de som bor der slik at deres bidrag ofte kan være til stor hjelp i et prosjekt.

De skal ha mulighet til å legge til meldinger på et prosjekt slik at de på en bedre måte kan informere og kommunisere med kommunen. De vil ha mulighet til å legge inn hvor mye de vil spare per husstand eller totalt ved dugnad, og legge inn bedrifter i området som ønsker bedre bredbånd. Bedrifter skal legges inn med navn, organisasjonsnummer, telefonnummer og adresse. De skal kunne se status og informasjon om prosjektet, men skal ikke kunne legge inn eller endre informasjon på vegne av de andre aktørene. De vil heller ikke ha mulighet til å se sjablong verdiene fra leverandørene.

Innbygger

Prosjektene må utlyses offentlig en måned før det sendes søknad til [Nkom](#). Fra dette tidspunktet, eller om kommune/fylke gjør dette tidligere, vil alle innbyggere kunne se all informasjon som kan offentliggjøres rundt prosjektet. Innbyggeren kan også se alle meldinger som er sendt til de som har registrert at de ønsker varsler fra området.

1.1.3.2 Hastighetsmåling

En innbygger skal kunne bruke en app eller nettside til å melde interesse om bedre bredbånd. Når hastigheten er målt skal de få mulighet til å forespørre bedre bredbånd. Etter de har gjort dette må man oppgi telefonnummer og epost for å knyttes til forespørselen. De må også oppgi hvilken måte de ønsker å bli kontaktet for statusmeldinger, dersom det blir utbygging av bredbånd i området. De vil ha mulighet til å bli kontaktet på sms, epost eller varslinger i appen.

Brukerens GPS posisjon vil også bli lagret, og brukeren vil bli spurt om å angi riktig adresse. Det skal foreslåes en adresse i nærheten av brukerens posisjon som brukeren eventuelt kan korrigere. Brukeren vil også ha mulighet til å oppgi at dette er en arbeidsplass.

Dersom en bruker forespør nettverk i et område der det allerede er utbygget bedre bredbånd fra andre leverandører skal brukeren få informasjon om hvilke leverandører som tilbyr dette.

1.1.3.3 Pålogging og registrering

Vanlige innbyggere skal kunne registrere seg på nettsiden eller appen med epost og passord. Her skal de få tilgang til informasjon om prosjekter i området der de har gjort en måling, og informere om at det ikke lenger ønskes bedre bredbånd.

Kommuner og leverandører skal kunne logge inn og registre ytterligere informasjon om deres stilling og kontaktinformasjon. En representant fra fylkeskommunen eller en annen godkjent bruker i samme kommune vil måtte godkjenne brukeren før de får tilgang til kommunens informasjon. Leverandøren må godkjennes av fylkeskommunen før de får tilgang til å se og legge inn tilbud.

1.1.3.4 Varmekart

På nettsiden og appen skal det være et varmekart som viser hvordan bredbåndstilknytningen er i fylket. Dette skal kunne brukes til å finne andre leverandører i et område som tilbyr bedre nett, eller se hvilke områder det vil være viktigere å bygge ut. Det vil være muligheter for å sortere frem mer spesifikke målinger, slik at man bare viser målinger basert på enkelte teknologier eller leverandører.

1.1.3.5 Interesse-varmekart

Dette vil være et varmekart basert på hvem som har sagt at de ønsker bedre nett. Dette vil være en enkel måte å kunne se hvilke områder det er mange som ønsker bedre nett og det er dårlig nett i dag. Dette kartet skal være tilgjengelig for fylkeskommune, kommune, leverandør og ildsjel.

1.1.3.6 Husstandskart

Kommunen skal kunne generere en markering på et kart som skal kunne brukes i et prosjekt. Dette området skal kunne inneholde mindre områder der det ikke ønskes utbygging, dermed vil ikke områdene være fastsatte former. På denne måten vil kommunen kunne kartlegge spesifikke områder som skal gjennomgå en utbyggelse.

1.1.3.7 Prosjekt

Kommunen eller fylkeskommunen skal kunne opprette et prosjekt for utbygging av bredbånd. Dette området skal spesifiseres ved hjelp av et husstandskart. Systemet skal kunne hente ut informasjon om husstander i området, og eventuelt målinger som er utført. Det skal fortløpende oppdateres med mer informasjon dersom innbyggere bruker appen eller nettsiden til å gjøre målinger eller forespørre bedre nett i området.

Et slikt prosjekt skal kunne offentliggjøres av kommunen eller fylkeskommunen på et hvilket som helst tidspunkt. Det skal også være støtte for å offentliggjøre et prosjekt på et bestemt tidspunkt slik at man er sikker på å overholde eventuelle frister fra Nkom.

1.1.3.8 Søknad

Kommunene og fylkeskommunen skal kunne generere et utkast til en søknad til Nkom for et av områdene. Søknaden skal inneholde informasjon om hvor mange husstander det er i området. Hva slags hastigheter appen og nettsiden har fått med hastighetsmålinger og hvor mange i området som allerede har fremmet et ønske om bedre bredbånd.

Alle prosjekter skal til enhver tid ha en status. Tilgangen til hvem som kan se hvilken informasjon om en søknad skal i noen grad kunne oppdatere seg ut ifra statusen. Under planleggings fasen vil i utgangspunktet bare kommunen som har opprettet prosjektet og Hedmark fylkeskommune ha tilgang. Når prosjektet lyses ut til innbyggerene vil all offentlig informasjon angående prosjektet være offentlig tilgjengelig på nettsiden. Når søknaden er sendt til Nkom skal det tas vare på søknaden slik den var da den ble sendt, slik at denne alltid vil være tilgjengelig dersom den endres på et senere tidspunkt. Hedmark fylkeskommune skal kunne støtte et prosjekt dersom det ikke har fått godkjent søknad fra Nkom. Kommunene vil også kunne bruke verktøyet fullt ut uten å sende inn noen søknad for situasjoner der de ikke ønsker støtte fra Nkom eller Hedmark fylkeskommune.

Det skal være støtte for at kostnaden skal kunne reduseres dersom innbyggerene ønsker å hjelpe til selv i noen grad, dette kan for eksempel være at alle utfører graving på egen tomt som kan legges inn av kommune, fylkeskommune eller en ildsjel.

På et hvilket som helst tidspunkt vil kommunen har mulighet til å gjøre prosjektet tilgjengelig for alle andre kommuner i fylket, dersom de ønsker innspill eller tilbakemeldinger fra andre kommuner. Denne tilgangen kan på et hvilket som helst tidspunkt trekkes tilbake.

Det skal finnes diverse informasjon om kostnader, det vil si hva det vil koste for leverandøren å bygge ut, om det vil være noen bidrag fra innbyggerene og hva slags støtte kommunen og fylkeskommunen ønsker å bidra med. Dersom prosjektet har fått tilstrekkelig finansiering skal det sendes anbud til de aktuelle leverandørene. På det tidspunktet vil leverandøren få full tilgang til å se prosjektet og bruke systemet til å sende meldinger til innbyggerene.

1.1.3.9 Kommunikasjon

Kommunen, fylkeskommunen eller en leverandør skal kunne informere brukere, som har oppgitt at de ønsker varslinger dersom det er en endring på et prosjekt.

Det skal kunne være kommunikasjon helt fra det opprettes et prosjekt og frem til det er ferdig utbygget. Det skal sendes ut informasjon om søknaden og dens status, men det skal også kunne sendes informasjon fra leverandører under utbyggingsfasen.

Informasjonen sendes på sms, epost eller som varslinger i appen, men alle skal også kunne se dette på nettsiden.

1.2 Formål

Hedmark fylkeskommune har fått tilbakemeldinger fra både næringslivet og private om at bredbånd og bredbåndsdekning i fylket er viktig for innbyggerne, bedrifter og fremtiden. I dag ligger Hedmark fylke et stykke etter landsgjennomsnittet når det gjelder høyhastighetsbredbånd, og de ligger i bunnsjiktet i en oversikt over bredbåndsutbyggingen i Norge. Selv om det er bredbåndsdekning de aller fleste steder i fylket, så er denne dekningen altfor lav og ustabil. Dette gjør det vanskelig for fylket og kommunene å reklamere for at Hedmark er et attraktivt sted å bo og jobbe. Det vil være vanskelig å friste folk til å flytte til eller opprette bedrifter i Hedmark hvis den digitale infrastrukturen ikke kan sammenlignes med den de er vant med i større byer. Derfor har Hedmark fylkeskommune satt i gang et prosjekt for å bedre dekningen og kunne gi høyere bredbåndshastighet til befolkningen i hele fylket [4, 5].

Nasjonal kommunikasjonsmyndighet (Nkom) har blant annet tilsyn med operatører og internettleverandører i Norge. De vil forplikte leverandører til å levere et grunnleggende bredbånd til hele befolkningen i Norge, uansett hvor du bor. Nkom definerer [grunnleggende bredbånd](#) som minst 4 Mbit/s i nedlastingshastighet. Og de ønsker at flest mulig skal kunne få tilbud om [høyhastighetsbredbånd](#), som er minst 30 Mbit/s i nedlastingshastighet. I dag opplever fylkeskommunen at kartleggingen av den nåværende situasjonen av bredbåndstilbudet er utfordrende, siden fylkeskommunen og kommunene ikke har tilgang til kartverk som viser hvor det er fiber eller andre typer infrastrukturer. De kan kun basere seg på de teoretiske dekningskartene fra Nkom. Ut fra disse dekningskartene kan kommunene enkeltvis søke om utbyggingsmidler

fra Nkom. Disse gir støtte til utbyggingsprosjekter der det er behov for grunnleggende bredbånd, øke kapasiteten på eksisterende bredbånd, eller hvis området ikke kommer til å bli bygd ut kommersielt de neste årene. Hvis kommunene og fylket har et bedre dekningskart, vil de kunne lage mer nøyaktige søknader til Nkom, og med det få mer tilskudd for utbygging i sine områder [6].

Hedmark fylkeskommune ønsker derfor å få en mer nøyaktig oversikt over hvordan dagens situasjon er. Dette vil gjøre at det kan jobbes for en mer målrettet og systematisk utbygging av bredbåndsnettet, slik at den treffer der det gir størst effekt. For å oppnå dette ønsker de et tettere samarbeid mellom kommunene, lokale krefter og involvering fra innbyggerne. De ønsker å få førstehåndsinformasjon fra innbyggerne om hvilke nettverkshastigheter som finnes, slik at det kan opprettes et nøyaktig dekningskart og oversikt over standarden på bredbåndstilbudet.

1.3 Målgruppe

1.3.1 Målgruppe for systemet

Systemet har ikke en enkelt målgruppe, men opptil flere. Systemet er ment til å bli brukt i arbeidssammenheng av ansatte i fylket, i kommunen og hos internettleverandører. Flere av disse menneskene er dedikerte ressurser som jobber med bredbåndsutbygning og har kjennskap til hvordan utbyggingsprosessen foregår fra før av. [Ildsjeler](#) skal jobbe sammen med disse gruppene på prosjektstyringsdelen av systemet. Siden ildsjelene er vanlige innbyggere er de en målgruppe med varierende kjennskap til prosessen. Dette betyr at prosjektstyringsdelen må tilrettelegges for dem.

Den åpne delen som er tilgjengelig for alle er i hovedsak ment for innbyggere. Hedmark fylkeskommune ønsker å nå ut til flest mulig uavhengig av deres alder og/eller bakgrunn. Derfor er det viktig å ha et stort fokus på brukervennlighet på den åpne delen.

1.3.2 Målgruppe for rapporten

Denne rapporten er skrevet av og ment for individer med kunnskap innenfor IT og utvikling av programvare. Det forventes lite/ingen kunnskap om bredbånd og prosessen rundt utbygningen. De teoretiske aspektene rundt dette vil vi dekke i [kapittel 2](#).

1.4 Egen bakgrunn og kompetanse

Alle gruppe medlemmene er studenter ved Bachelor i programvareutvikling, så dermed har vi den samme akademiske bakgrunnen. Fra studiet har vi erfaring med programmeringsspråk og verktøy som C++, Java, MySQL og webutvikling i PHP, HTML, CSS, Bootstrap og JavaScript med JQuery. Vi har også erfaring med native mobilutvikling. Innenfor mobilutvikling har vi litt ulik erfaring. Ett av gruppe medlemmene har bare erfaring med Android-utvikling, mens de to andre har prøvd å utvikle til både Android og iOS. Vi har også 20 studiepoeng i systemutvikling inkludert i vårt studie. Gjennom disse studiene har vi fått kunnskap om ulike systemutviklingsmodeller, ulike verktøy og planleggingsprosessen. Vår kunnskap i planlegging, design og ulike utviklingsmetoder er mest teoretisk. Dette er det største prosjektet gjennom studiet og gir en god mulighet til å sette teorien ut i praksis. I studiet har vi også hatt faget Ergonomi i digitale medier, som har gitt oss kunnskap i universell utforming og brukervennlighet. Av annen kunnskap utenfor studiet har Lasse erfaring med webutvikling og Ruby. Det var en del som var nytt for oss gjennom dette prosjektet og vi hadde litt ulik erfaring med webutvikling. Ingen av oss hadde erfaring med frontend-rammeverket Angular og språket typescript som blir brukt der. Ingen hadde erfaring med [cross-platform](#) utvikling til mobil, så innenfor dette ble C# og Xamarin nytt for oss. Ingvild og Kristin hadde heller ingen erfaring med programmeringsspråket Ruby.

1.5 Rammer

1.5.1 Arbeidsmetoder

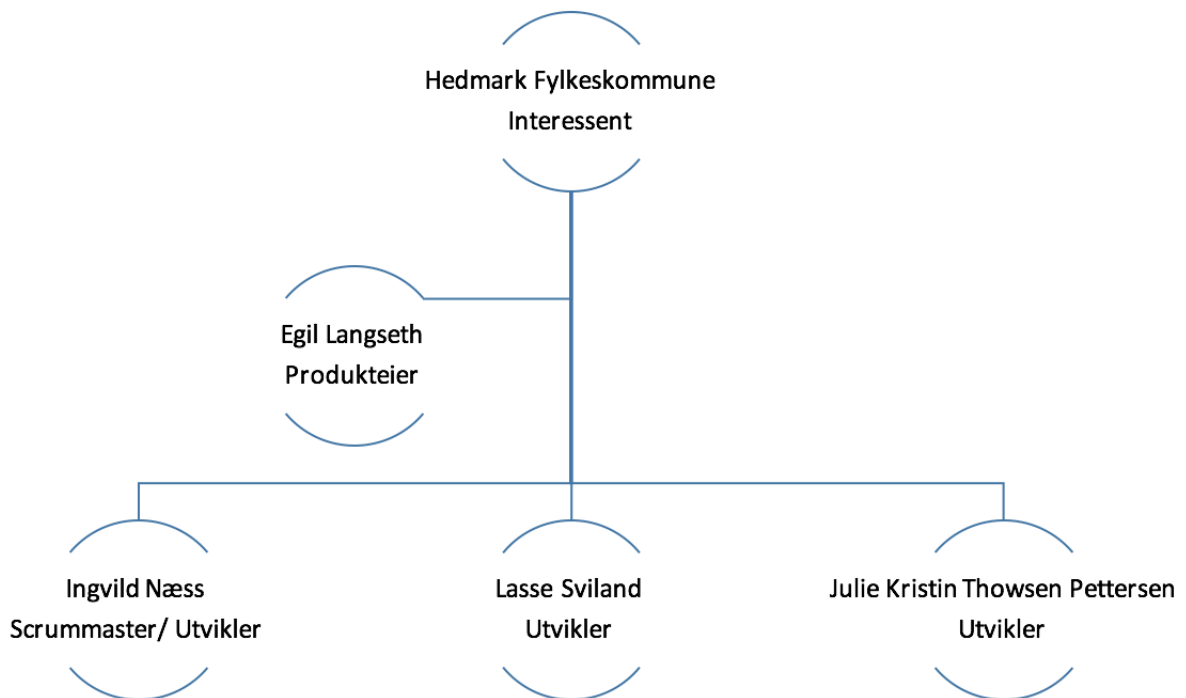
En del av prosjektet har vært å sette oss grundig inn i hvordan systemet skal fungere, og hva oppdragsgiveren ønsket. Vi har blant annet måttet gjøre oss kjent med søknadsprosessen og utbyggingsprosessen til Nkom. For å sikre en god forståelse av prosjektet, har vi hatt mange møter med oppdragsgiver. Vi har i tillegg vært på et seminar om bredbåndsutbygging i Hedmark, der vi presenterte vårt prosjekt, og vi fikk mye verdifull informasjon fra de forskjellige kommunene. Vi har også brukt tid på å sette oss inn i teori om bredbåndsteknologier og hastighetsmåling. Noe av denne nye informasjonen har vi fått fra oppdragsgiver, men det meste har vi funnet ut

på egenhånd gjennom søking på internett.

1.5.2 Prosjektorganisering

Prosjektet ble gjennomført av studentene på gruppa, i tett samarbeid med oppdragsgiver. I tillegg har vi hatt jevnlig møter med veileder. I starten av prosjektet var vi fire medlemmer på gruppa, men etter hvert ble vi redusert til tre medlemmer.

1.5.2.1 Organisasjonskart



Figur 1.1: Organisasjonskart

1.5.2.2 Rollebeskrivelser

Scrum-teamet er organisert etter vanlige scrumroller: [produkteier](#), [Scrum Master](#) og [utviklings-teamet](#).

Egil Langseth, leder for bredbåndsprosjektet i Hedmark fylkeskommune, er valgt til å være produkteier for dette prosjektet. Han har vært ansatt som leder av bredbåndsprosjektet siden mai 2016, og har derfor god kjennskap til hvilke utfordringer Hedmark fylke står ovenfor og hva slags produkt de ønsker fra oss. Egils hovedrolle er å være bindeleddet mellom oss og de andre interessentene hos Hedmark fylkeskommune, representere deres visjon og hjelpe oss å prioritere våre arbeidsoppgaver i løpet av prosjektet.

Ingvild Næss er Scrum Master for dette prosjektet. Hennes rolle er å ha et overordnet blikk over hele prosjektet slik at tidsfrister, fremdrift og møter blir holdt. Hun vil også sørge for at utviklingsteamet jobber effektivt slik at alle arbeidsoppgaver under hver sprint blir gjennomført. Ingvild vil ha en delt rolle i dette prosjektet. Mens hun er scrum master, vil hun også være en del av utviklingsteamet.

Utviklingsteamet består av Julie Kristin Thowsen Pettersen, Lasse Sviland og Ingvild Næss. Utviklerne har ansvaret for å levere ferdige oppgaver når hver sprint er over. Det blir fordelt oppgaver mellom utviklerne på begynnelsen av hver sprint. Alle medlemmene av utviklingsteamet hjelper hverandre for å sikre at en sprint blir vellykket og ferdigstilt.

1.5.2.3 Øvrige roller

Veileder

Vi har hatt Frode Haug som veileder. Frode Haug er ansatt som universitetslektor ved NTNU i Gjøvik. Han har vært ansatt på skolen siden 1993 hvor han underviser i ulike datafag. Han har også vært veileder for ulike bacheloroppgaver i flere år. Frode har vært en god støtte gjennom prosjektet ettersom han sitter med mye kunnskap om selve prosessen og rapportskrivningen[7].

1.5.2.4 Endringer i gruppa

I starten av prosjektet var vi fire gruppemedlemmer. Vi hadde derfor planlagt, både antall arbeidsoppgaver og tidsskjema, ut fra at det skulle være fire stykker som la ned en arbeidsinnsats i dette prosjektet. Et av gruppemedlemmene la dessverre ikke inn så mye innsats som gruppa og skolen kunne forvente. Da dette ikke lot seg løse, ble det bestemt av veileder at gruppa skulle splittes i to. Begge gruppene har tilgang til alt som ble produsert av dokumentasjon og kode

frem til 6. april, men hver av gruppene har jobbet selvstendig etter dette. Det fjerde gruppe-medlemmet har til denne oppgaven bidratt med å være med på diskusjoner og planlegging i planleggingsfasen. Det finnes ingen original kode fra han.

1.5.3 Plan for gjennomføring

Vi har valgt å dele hele prosjektperioden inn i fire hoveddeler: planlegging, utvikling, fullføring av hovedrapport og forberede muntlig presentasjon. Se en mer detaljert oversikt over vår fremdriftplan i Gantt-diagrammet vårt fra 28.01.17 i forprosjektrapporten i [vedlegg H](#).

Planleggingsperioden består av fire sprints og varer fra 16. januar til 12. februar. Under denne perioden er første sprint og deler av andre sprint satt av til å jobbe med forprosjektrapporten som skal leveres 28.januar. Første sprint vil gå med til å skrive rapporten, mens vi vil rette opp og forbedre dette arbeidet i andre sprint etter tilbakemeldinger fra oppdragsgiver og veileder. Mens vi forbedrer og klargjør forprosjektrapporten til innlevering, vil vi også starte på kravspesifikasjonen i sprint 2. Arbeidet med denne vil fortsette ut i sprint 3. Kravspesifikasjonen er et viktig arbeid for å få definert all funksjonalitet, så det er viktig at denne er på plass før vi kan starte med utviklingen. I sprint 3 har vi også satt opp at vi skal forberede oss til seminar på Elverum, som vi skal delta på 7. og 8. februar. Dette er et seminar der representanter fra Hedmark Fylkeskommune, kommunene i Hedmark, Nkom og ulike bredbåndsleverandører vil være tilstede for å snakke om bredbåndsutbyggingen i Hedmark Fylke. På seminaret skal vi presentere vår bacheloroppgave. I sprint 4 vil vi i tillegg til å delta på seminaret ferdigstille backloggen med alle arbeidsoppgaver som skal gjennomføres under utviklingsperioden. Vi vil også sette opp de ulike utviklingsmiljøene og databasene vi trenger.

Utviklingsperioden vil vare fra 13. februar til 9.april, og vil bestå av 8 sprints. I denne perioden vil vi utvikle all funksjonalitet i systemet, og vi har satt noen datoer for når de viktigste delene skal være ferdig:

- 27. februar: Det skal nå være mulig å gjøre hastighetsmåling på nettsiden og appen.
- 6. mars: Varmekart over hastighetsmålinger og innlogging for ulike roller skal være ferdig. Dette innebærer også registrering av nye brukere.
- 20. mars: Brukere som er registrert som innbyggere kan nå motta meldinger.

- 27. mars: Det skal nå være mulig å generere søknader og rapporter, og kommunene skal kunne opprette områder de ønsker utbygging i, ut fra et kart.
- 3. april: Leverandører og kommunene kan legge inn informasjon.
- 10. april: Nettsiden og appen skal være ferdig utviklet.

Perioden fra 10.april til 16.mai er satt av til å skrive ferdig rapporten vi skal levere inn til vurdering 16.mai. Dette har vi satt av 5 sprinter til. Etter at hovedrapporten er levert inn, har vi fra 17. mai til 5. juni til å forberede den muntlige presentasjonen vi skal ha 6. eller 7. juni. Vi har valgt å ikke dele denne perioden opp i sprinter, siden vi i denne perioden også må prioritere eksamen i andre emner vi har.

1.5.4 Ytre føringer

- Applikasjonsstøtte for Android og iOS
- Støtte for Internet Explorer, Firefox, Safari og Chrome
- Kommunikasjonsenheten ved Hedmark fylkeskommune har gitt oss noen retningslinjer når det gjelder det grafiske uttrykket.
- Kommunikasjonsenheten mener også at universell utforming og brukervennlighet er avgjørende for en løsning av denne typen.
- Gruppen har ikke tilgang til noen fast reservert grupperom, men bruker universitetets system for reservering av grupperom for å ha en ganske sikker tilgang på arbeidsplass.
- Tidsperioden for dette prosjektet er fra 16.januar til 16.mai. Vi har som mål å bruke til sammen 78 timer pr. uke fordelt på 3 studenter. Vi har ikke mulighet til å jobbe fulltid med dette prosjektet ettersom vi tar andre emner ved siden av.
- Møtedagene våre er mandag til torsdag fra kl. 09.00.

1.5.5 Systemutviklingsmodell

Vi har valgt å jobbe etter systemutviklingsmodellen Scrum. Oppgavebeskrivelsen fra oppdragsgiver er ganske omfattende. Vi har gjort en avgrensning på hva vi har mulighet til å rekke på den tiden vi har til rådighet. Scrum er en fleksibel utviklingsmodell, som åpner opp muligheten for å legge til mer arbeid etter hvert i prosessen hvis vi ser at den opprinnelige arbeidsmengden blir for liten.

[Sprint Planning Meeting](#) og [Sprint Review Meeting](#) er med på å gi oss nær kontakt med oppdragsgiveren. Dermed får vi stadig tilbakemelding på om vi arbeider i riktig retning, og kan raskt ordne opp i eventuelle misforståelser. På denne måten sikrer vi oss å gi oppdragsgiver det produktet han ønsker. Dette er både en fordel for oss og oppdragsgiver.

Aspektet med å ha sprinter kan være med på å holde motivasjonen og fokuset i gruppen oppe. Sprintene gjør at vi alltid har et klart mål å jobbe opp mot. Vi vil alltid vite hva å gjøre i løpet av en uke slik at vi slipper å bli distraheret av fremtidige oppgaver. Dette er en fordel da oppgaven er omfattende og det er lett å ta tak i for mye på en gang.

Scrum tilrettelegger for at utviklerteamet skal ha mye kommunikasjon seg imellom. Dette gjør det lettere for oss å oppdage tidlig hvis arbeidet vi utfører skaper konflikt med andre sitt arbeid. På denne måten sparer vi tid.

1.5.5.1 Hvordan anvende modellen

I prosjektet kommer vi til å ta i bruk følgende elementer fra Scrum: [Scrum Master](#), [Product Backlog](#), [Sprint](#), [Sprint Backlog](#), [Sprint Planning Meeting](#), [Sprint Review Meeting](#) og [Daily Scrum Meeting](#). Product Backloggen kommer til å bli lagd i starten av utviklingsfasen og vil vokse seg til etter hvert som flere krav avdekkes.

Vi har fastslått at vi skal ha sprinter som varer i syv dager. Sprinten starter på mandag og avsluttes på søndag. I starten av hver sprint har vi møte med produkteieren. Dette møtet blir av praktiske årsaker en blanding av [Sprint Review Meeting](#) og [Sprint Planning Meeting](#). På møtet får vi tilbakemeldinger på forrige sprint og vi diskuterer hvilke oppgaver i product backloggen som skal prioriteres i den nye sprinten. Oppgavene blir så estimert og fordelt.

Sprinten varer i syv dager, men i praksis er arbeidsdagene våre på prosjektet fra mandag

til torsdag. Hver arbeidsdag møtes utviklerteamet for å ha et Daily Scrum Meeting. På møtet diskuterer vi hva vi gjorde dagen før, eventuelle problemer som har oppstått underveis og hva vi planlegger å gjøre denne dagen.

1.5.6 Arbeidsverktøy

1.5.6.1 Latex

Alt av skriftlige dokumenter som vi skal utarbeide skal skrives i Latex. For å forhindre en stor jobb med å sette sammen dokumenter på slutten skal vi benytte ShareLatex som gir oss muligheten til å skrive sammen i sanntid. Dette gir oss fleksible dokumenter som følger lik stil fra første stund.

1.5.6.2 Jira

Som nevnt tidligere skal vi bruke utviklingsmodellen Scrum. Dette innebærer ukentlige sprinter hvor vi fordeler oppgaver på starten av hver sprint. For å samle inn oppgaver og fordele oppgavene skal vi bruke Jira sitt [Scrum-board](#). Dette gir oss kontroll på hva som gjøres og av hvem til enhver tid.

Internt i gruppa vil Jira i tillegg bli benyttet som et verktøy for [issue tracking](#). Medlemmene i gruppa legger selv inn oppgaver eller feil som oppdages fortløpende i Jira.

1.5.6.3 Git og Bitbucket

For å ivareta integriteten til kildekoden skal Git benyttes under utvikling som et versjonskontrollverktøy. Siden vi skal utvikle to forskjellige utgaver av produktet (mobil og web) har vi besluttet å opprette et repository per utgave. Dette skal være med på å forhindre store mappestrukturer og gjøre det lettere for gruppemedlemmene å jobbe på hver sin oppgave.

Repositoryene opprettes på Bitbucket som er det verktøyet flertallet i utviklerteamet bruker fra før.

1.5.6.4 Google Drive

I tillegg til kildekode og rapporter som vi selv produserer vil det oppstå andre filer som vi trenger å ha tilgjengelig oss i mellom. Dette kan være notater skrevet fortløpende, filer delt med oss av oppdragsgiver og lignende. Disse filene kommer til å ligge i en felles Google Drive mappe.

1.5.6.5 Toggl

Vi bestemte i januar at vi skulle registrere timene vi brukte på prosjektet, gjennom hele prosjektperioden. Vi bestemte at alle gruppe medlemmene skulle bruke samme timeregistreringsprogrammet slik at vi lettere kunne følge med på total tidsbruk. Vi har valgt å bruke timeregistreringsprogrammet Toggl ettersom dette lot oss enkelt opprette team som skal jobbe sammen, det er gratis og brukervennlig. Se liste over tidsbruk i [vedlegg K](#).

1.6 Organisering av rapporten

1.6.1 Avklaringer om rapporten

Vi bruker en font gjennom hele rapporten. Ord som det finnes forklaring på i ordlista er markert med blå skrift. Disse ordene fungerer som lenker til ordlista. Når du klikker på et blått ord, går du til siden med ordlista i dokumentet.

I rapporten vil vi vise utsnitt fra vår kode. Vi har valgt å ikke ta med kommentarene fra koden ettersom vi forklarer koden i teksten.

1.6.2 Kapitteloppsummering

Kapittel 1: Innledning

I dette kapitlet avklarer vi formålet med oppgaven, hva selve oppgaven går ut på og litt om prosessen rundt prosjekt jobbingen.

Kapittel 2: Teoretisk grunnlag

Kapitlet tar for seg det teoretiske grunnlaget som leseren trenger for å forstå vår oppgave bedre. Vi har skrevet teori om bredbånd, hvordan utbyggingsprosessen til Nkom fungerer og hvordan en hastighetsmåling fungerer.

Kapittel 3: Kravspesifikasjon

I dette kapitlet avklarer vi brukergruppene og kravene til systemet. Brukergruppene blir fremstilt ved hjelp av personas og pact-analyse. Kravene fremstiller vi ved hjelp av use case og system sekvensdiagrammer.

Kapittel 4: Design og arkitektur

Oversikt over den totale sammenhengen i systemet, teknologivalg, prototype, universell utforming og fargeprofil blir omtalt her.

Kapittel 5: Implementering

I dette kapitlet viser vi hvordan vi implementerte kravene vi definerte i kapittel 3.

Kapittel 6: Testing og kvalitetssikring

Kapitlet tar for seg hvordan vi arbeidet med kvalitetssikring og hvordan vi testet koden vår gjennom prosjektet.

Kapittel 7: Avslutning

Evaluering av arbeidet og prosessen, muligheten for videre arbeid på prosjektet og eventuell kritikk blir omtalt her.

2 *Teoretisk grunnlag*

I dette kapitlet skal vi se på det teoretiske grunnlaget som vi har opparbeidet oss. I kapitlet vil vi dekke temaer som vi selv har satt oss inn i før vi startet med utviklingen. Her tar vi for oss enkelte definisjoner som dukker opp i oppgaven og ser dypere på disse. Alle temaene brukes ikke like hyppig i oppgaven, men er med for å bygge opp en forståelse rundt fagområdet. Temaene som vi vil ta for oss her er bredbånd generelt, hvem [Nkom](#) er og deres egne begreper og metoder, og til slutt hva de forskjellige begrepene innenfor hastighetsmåling er og hvordan man kan utføre en hastighetsmåling.

2.1 **Bredbånd**

Internett er et globalt datanettverk som gjør det mulig for ulike enheter å kommunisere sammen på tvers av forskjellige nettverk og nettverkstilkoblinger som for eksempel fiber, satellitt, trådløst nett osv. De ulike nettverkene og tilkoblingstypene bygges og driftes av teleselskaper og interettleverandører. Alle de ulike nettverkene blir koblet sammen til ett stort nettverk ved hjelp av rutere. En ruter er en datamaskin som er programmert til å formidle overføringen videre. Alle nettverkene mellom ruterne kan være av ulike tilkoblingstyper, men felles for alle tilkoblingstypene er at de kan overføre digitale signaler.[8].

Bredbånd kan defineres som tilknytting til internett med høy overføringskapasitet, eller høy båndbredde. Båndbredde måles i bit pr sekund og gir informasjon om hvor mye data som kan overføres på nettverket pr sekund[8]. Det finnes flere måter å få bredbånd på. Du kan få bredbånd gjennom telefonlinja, i kabel inn i huset ditt, via radiokommunikasjon eller satellitt.

Bredbånd via kabel kan foregå på tre ulike måter. Du kan koble deg til via telenettet, via fiberkabel eller via kabel-tv-nettet. Å koble seg på via telenettet forutsetter at det lokale telenettet

tilbyr denne tjenesten og du trenger et modem som overfører analoge telefonsignaler til digitale signaler. Oppkobling via telenettet er også kjent som DSL og ADSL, der ADSL er den vanligste typen ettersom den gir bedre nedlastningshastighet enn opplastningshastighet. Fiberkabel er den tjenesten som gir størst kapasitet og hastigheter. Dette er en relativt ny tilkoblingstype og den er fortsatt under utbygging og det er derfor mange som fortsatt ikke har tilbud om denne tilkoblingstypen der de bor. Fiberkabler kan kobles til huset ditt via kabler i bakken eller kabler i luft og det gir deg muligheten til å overføre flere tv-kanaler, internett og telefoni samtidig med en høy overføringskapasitet. Internett via kabel-TV-nett er tilgjengelig for boliger i områder der det er bygd ut med kabel-TV, og dette er gjerne i byer eller tettsteder. Før ble denne tilkoblingstypen bare benyttet til å overføre TV-signaler, men kan i dag i tillegg overføre internett og bredbåndtelefoni gjennom samme kabel[9].

Oppkobling via radiokommunikasjon kan foregå på tre ulike måter, 3G, 4G og WiMax. 3G og 4G betyr at du kobler deg på via mobilnettet og har mulighet for oppkobling nesten uansett hvor du er i verden. WiMax er et system som gir radiotilgang med rekkevidde opp til 50 km. Dette er en tilkoblingstype som kan brukes i områder der bredbånd ikke er tilgjengelig via kabel. WiMax tar i mot signalene via en ekstern antenne, og det gir best resultater hvis antennen har god sikt til senderen, altså en radioantenne[9].

Å få bredbånd gjennom satellitt krever at du har en egen parabolantenne med fri sikt til en geostrasjonær bane. Det er ofte store avstander mellom senderen og mottakeren, dette vil gjøre at det blir store forsinkelser på signalene selv om hastighetene er nokså gode[9].

2.2 National Kommunikasjonsmyndighet(Nkom)

National Kommunikasjonsmyndighet (Nkom) er underlagt samferdselsdepartementet, og er ansvarlig for de som driver post- og teletjenester i Norge.

Nkom driver blant annet tilsyn etter ekomloven. Denne lovens formål er blandt annet å sikre brukerne i hele landet gode, rimelige og fremtidsrettede elektroniske kommunikasjonstjenester.

[10] [11]

2.2.1 Hvitt område

Et hvitt område er et område som regnes for å ikke ha et grunnleggende tilbud på bredbånd. For å ha et grunnleggende tilbud, må det være minimum 4Mbit/s nedlastnings hastighet. [6]

2.2.2 NGA område

I 2010 kom EU-kommisjonen med en anbefaling om å oppgradere bredbåndstilbudet slik at alle innbyggere i EU skal ha tilgang til veldig raskt bredbånd innen 2020. Dette ble kalt Next generation access(NGA) networks.[12]

For at et område skal defineres som et NGA område må det oppfylle Nkom sitt krav om å ha en nedlastningshastighet på minst 30 Mbit/s.[6]

2.2.3 Søknadsprosessen

Stortinget bevilget i 2017 138,7 millioner kroner for utbygging av bredbånd.[6]

Det gis støtte for utbygging av NGA nettverk i områder der det ikke lønner seg kommersielt å bygge ut. Søknadene blir vurdert ut i fra en rekke kriterier.

- Antall huser under 4Mbit/s.

Huser som i dag ikke regnes som å ha et grunnleggende tilbud.

- Antall huser mellom 4Mbit/s og 30Mbit/s.

Huser som ikke fyller kravet til å være et NGA nettverk.

- Kostands effektiv nettutbygging.

Hvor mye det koster å bygge ut per husstand.

- Lokal medfinansiering.

Hvor stor andel av prosjektet som finansieres av kommunen og innbyggerne i området.

- Plan for bærekraftig drift.

Hvilke krav som settes til utbygger angående vedlikehold og videreutvikling av nettverket.

- Betydning for lokal samfunnsutvikling og verdiskapning.

Hvordan lokalt næringsliv påvirkes og hvilken gevinst samfunnet vil få.

Av disse kriteriene vektlegges husstander uten et grunnleggende tilbud, kostnad per husstand og hvor mye som blir finansiert lokalt, høyere enn de andre kriteriene. Dersom husstander i området har tilbud om NGA nettverk, vil det ikke gis støtte til prosjektet.[6]

2.3 Hastighetsmåling

Hele prosjektet baserer seg rundt målinger av bredbåndshastigheter. Dermed er nøyaktige hastighetsmålinger et viktig kriterium for suksess. Vi har i vår oppgave valgt å ta utgangspunkt i Ookla sin artikkel "How does the test itself work? How is the result calculated?"[13] når vi skulle lage hastighetsmålingen. Ookla er velkjente innenfor fagområdet og driver nettstedet speedtest.net hvor man kan sjekke sin egen hastighet. Vårt produkt tester mot de samme serverne som speedtest.net. Alle serverne inneholder de samme filene. Disse filene brukes for å gjennomføre målingen. En skjermdump av filoppsettet ligger med som vedlegg. Se [vedlegg A](#).

Ved å sammenligne våre resultater med Ookla sine resultater kan vi få en god pekepinn på hvor nøyaktige målinger vi klarer å produsere.

Måling av bredbånd består av tre viktige deler: ping, nedlastning og opplastning. I følgende avsnitt skal vi se nærmere på hva de forskjellige delene er og hvordan de gjennomføres.

2.3.1 Ping — hva er det?

Ookla definerer [ping](#) som følgende[14]:

The ping is the reaction time of your connection—how fast you get a response after you've sent out a request. A fast ping means a more responsive connection, especially in applications where timing is everything (like video games). Ping is measured in milliseconds (ms).

Ping måles fordi responstiden mellom klienter og servere varierer. Dette vil slå ut på målingene av nedlastning og opplastning. Jo lengre tid det tar klienten å nå serveren, jo lengre tid

vil også målingene ta. Dette gir dårlige hastigheter. Ping er round-trip delay time. Dette vil si at pingtiden er tiden det tar å sende en forespørsel fra klienten til serveren pluss tiden det tar å få svar fra serveren.



Figur 2.1: Illustrasjon av ping

2.3.2 Ping — hvordan gjør vi det?

Når man skal måle hastigheter burde man lete etter en server der man får lav [ping](#). Med en slik server kan vi måle hastigheter som gir et reelt bilde av hva bredbåndsforbindelsen kan yte. Ookla har en offentlig liste over sine speedtest-servere. Ved å pinge forskjellige servere kan man avgjøre hvilken server å teste mot. Dette har vi ikke hatt tid til å implementere, og vi bruker derfor Telenor sin speedtest-server som befinner seg i Oslo. Dette er serveren vi har benyttet oss av under utviklingsperioden. Serveren har høy oppetid og gir gode resultater mot indre Østlandet.

For å teste ping gjør klientene våre en forespørsel om å hente ned filen `latency.txt` fra serveren. Samtidig starter klienten en stoppeklokke. Serveren sender filen til klienten. Når klienten har mottatt filen stoppes stoppeklokken. Filen som hentes er svært liten og gjør ikke noe betydelig utslag på målingen. Pingtiden blir da tiden det tok å motta filen målt i millisekunder.

Denne prosedyren gjentar vi frem til en forhåndsbestemt tid har gått. Den laveste målingen blir gjeldende.

2.3.3 Nedlastingshastighet — hva er det?

Ookla definerer nedlastingshastighet som følgende^[14]:

The download speed is how fast you can pull data from the server to you. Most connections are designed to download much faster than they upload, since the majority

of online activity, like loading web pages or streaming videos, consists of downloads.

Download speed is measured in megabits per second (Mbps).

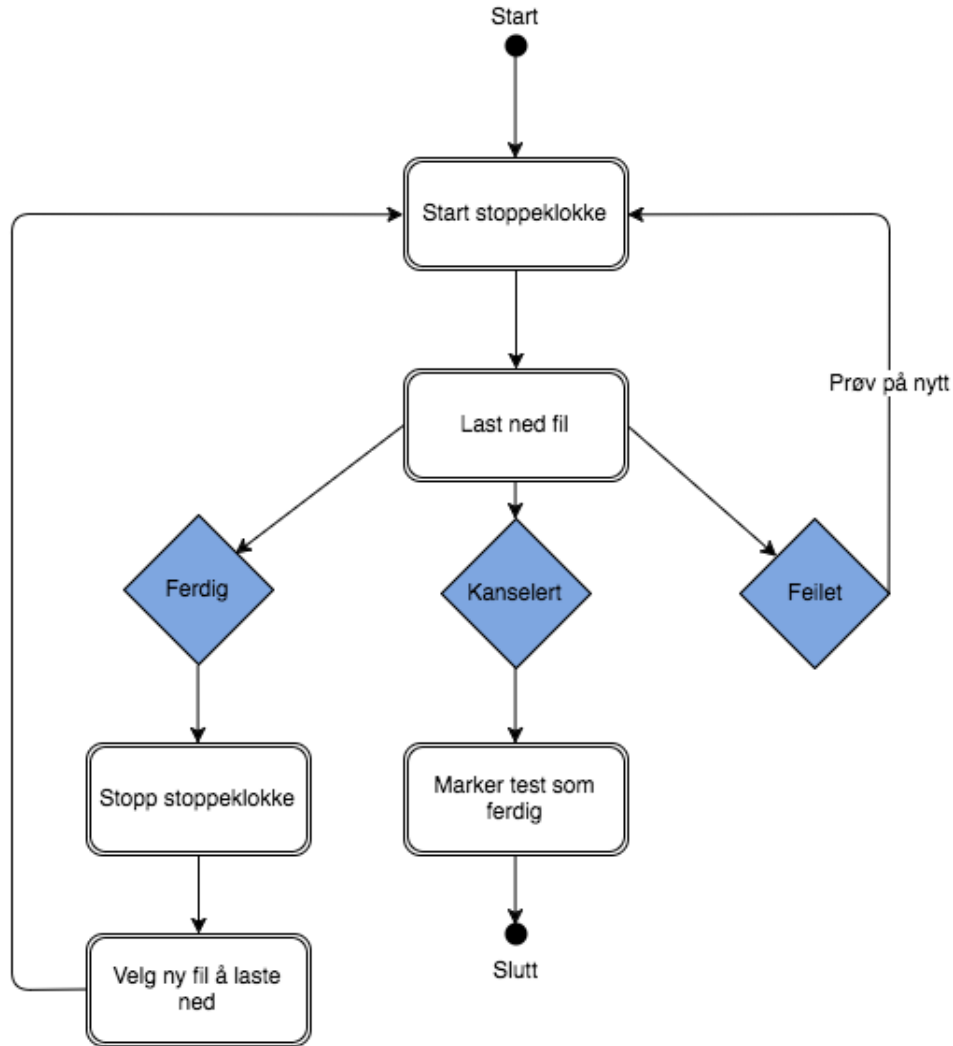
2.3.4 Nedlastingshastighet — hvordan gjør vi det?

Testing av nedlastningshastighet er svært lik testing av ping. Slik som med ping, gjøres det en rekke målinger i løpet av en bestemt tidsperiode. Klienten starter en stoppeklokke og laster ned en fil fra serveren [asynkront](#). Når filen er lastet ned stoppes stoppeklokken. Deretter blir det regnet ut hvor mange megabits den klarte å laste ned per sekund.

Filen vi får fra serveren har en størrelse oppgitt i bytes, og stoppeklokken gir oss et resultat oppgitt i millisekunder. Dermed ender vi opp med regnestykket:

$$\frac{FileSizeInBytes \cdot 8}{TimeInMs \cdot 1000 \cdot 1000000} = x \text{ Mbps}$$

Når en enkelt fil har blitt lastet ned ser vi på hvor lang tid nedlastingen tok. Dersom det gikk raskt endrer vi hvilken fil vi laster ned slik at vi laster ned en større fil. Ved å gjøre dette belaster vi bredbåndet mer og får mer nøyaktige målinger.



Figur 2.2: Aktivitetsdiagram ved testing av nedlastningshastighet

Siden vi bruker [asynkrone](#) metoder vil metodene kjøre i bakgrunnen samtidig som programmet fortsetter å arbeide. Hadde vi kjørt flere nedlastninger etter hverandre ville disse blitt gjennomført samtidig og gitt utslag på hverandres hastigheter. Derfor må vi alltid vente på en enkelt nedlastning skal markeres som ferdig før vi starter på neste. Hele testen av hastigheten skal fullføres etter en bestemt tid, på dette tidspunktet blir vi nødt til å kansellere pågående test.

Filene vi benytter oss av under testen heter på serveren `random350x350.jpg`, `random500x500.jpg`, `random750x750.jpg` o.l. Oversikt finnes i [vedlegg A](#)

2.3.5 Opplastningshastighet — hva er det?

Ookla definerer opplastningshastighet som følgende[14]:

The upload speed is how fast you send data from you to others. Uploading is necessary for sending big files via email, or in using video-chat to talk to someone else online (since you have to send your video feed to them). Upload speed is measured in megabits per second (Mbps).

2.3.6 Opplastningshastighet — hvordan gjør vi det?

For å sjekke opplastningshastigheten laster vi opp de samme filene som vi hentet ned når vi sjekke nedlastningshastigheten.

Først prøver vi å laste opp den minste filen. Filen gjøres om til bytes før den sendes. Vi starter en stoppeklokke og bruker en [asynkron](#) metode for å laste opp filen til serveren. Når filen er lastet opp stoppes stoppeklokken. Vi velger så en større fil som gjøres om til byte. Denne prosedyren gjentas frem til en forhåndsbestemt tid har gått. Når hver enkel opplastning er gjennomført regner vi ut hastigheten på samme måte som vi gjorde med nedlastningshastighet. Hele prosedyren ligner på prosedyren som gjøres ved test av nedlastningshastighet, og anvendes slik som vist i [figur 2.2](#)

Filen vi laster opp lastes opp til serveren mot filen "upload.php". Se [vedlegg A](#).

3 *Kravspesifikasjon*

I dette kapittelet bruker vi metoder innenfor systemutvikling for å kartlegge hvilke krav som settes til systemet. Vi tar først for oss brukergruppene våre. Dette gjør vi gjennom en PACT-analyse og ved bruk av personas. Deretter kartlegger vi de funksjonelle kravene i systemet ved bruk av Use Case.

3.1 Brukergrupper

3.1.1 Pact-analyse

Vi har valgt å inkludere en PACT-analyse inn i vårt prosjekt, slik at vi kan ta bedre hensyn til ergonomien i systemet. Et ergonomisk system er viktig for å gi brukeren en god brukeropplevelse. PACT-analysen ser på systemet i fire kategorier: Mennesker, aktiviteter, sammenhenger og teknologi. Ved å gjøre en analyse av disse fire kategoriene får vi en forståelse av de viktigste tingene i de kategoriene, og kan finne en løsning som kombinerer alle de områdene på en god måte. Vi som utvikler dette systemet må forstå vår brukergruppe, hvilke aktiviteter de brukerne vil gjøre og i hvilken sammenheng de aktivitetene kan finne sted. Vi trenger også å forstå hvilke teknologier som kan brukes, og hvordan disse teknologiene kan brukes for å støtte aktivitetene som vil finne sted i forskjellige sammenhenger.

3.1.1.1 Mennesker

Mennesker i alle aldre som bor i Hedmark fylke er de potensielle brukerne av systemet. Dette inkluderer både de menneskene som er opprinnelig fra Hedmark fylke, og de som har flyttet til fylket på grunn av jobb eller skole. Systemet skal være tilgjengelig og lett å bruke for folk med

ulik bakgrunn og forutsetninger. Vi må derfor ta hensyn til mennesker med orienteringsvansker, som dårlig syn eller blindhet. Vi må også ta hensyn til mennesker med motoriske funksjonshemninger. Klikkbare objekter må derfor være lett å treffe for brukere med redusert presisjonsevne. Vi må også ta høyde for at alle brukere ikke forstår norsk, derfor må systemet også støtte engelsk språk. Systemet skal også ta hensyn til mennesker med ulik teknisk forståelse og med ulik erfaring med mobile apper og nettsider. Det skal være enkelt å navigere seg rundt og med hjelpende tekster der dette er nødvendig. Vi skal også ta høyde for at brukere skal kunne få hjelp til å gjøre en hastighetsmåling og forespørsel om bedre nett. Det skal derfor være mulig å legge til flere kontaktpersoner når en forespørsel blir foretatt.

3.1.1.2 Aktiviteter

Systemet vårt består av flere deler og disse delene har ulike funksjonaliteter. For en innbygger i Hedmark fylke vil det være mulig å gjøre en hastighetsmåling, registrere denne og eventuelt registrere sin interesse for bedre internetthastighet. Alle disse funksjonalitetene vil det være mulig å gjøre via vår app, som finnes for Android og iOS, og via vår nettside. Den andre delen av vårt system vil rette seg mot ansatte i fylket, kommuner, internettleverandører og ildsjeler. Alle disse ulike brukertypene vil ha litt forskjellig funksjonalitet, se avsnittet *úopp-gavebeskrivelsež* for en mer utfyllende tekst om dette. Hovedfunksjonaliteten for denne delen av systemet vil være at en kommune skal kunne opprette et prosjekt med ett eller flere under-områder. Dette skal gjøres ved hjelp av å markere et område på et kart. Ut fra dette kartet får de også oversikt over hvor mange husstander det er innenfor et område. Systemet skal også tilby mer eksakt statistikk i forhold til konkrete hastighetsmålinger og liste over hvem som har registrert interesse for bedre nett. Kommunene kan også skrive søknaden til Nkom i vårt system, for så å generere den til et word-dokument. Systemet skal også tilby at ansatte, ildskjeler og andre mennesker som har tilgang til et prosjekt kan kommunisere med hverandre. Det skal også tilby at en kommune skal kunne sende ut felles beskjeder på mail eller sms til alle som har registrert sin interesse for bedre nett innenfor et område.

3.1.1.3 Sammenhenger

Hovedbruken av løsningen vil mest sannsynlig være innendørs, både for appen og nettsiden. Men vi må også ta hensyn til at brukere vil gjøre en hastighetsmåling og forespørsel av bedre nett utendørs, spesielt da på appen. Vi må derfor ta hensyn til at løsningen kan brukes i situasjoner med mye lys og lite lys. Det krever derfor at det er gode kontraster i løsningen.

3.1.1.4 Teknologi

Både nettsiden og appen trenger tilgang til internett for å fungere. Når en bruker gjennomfører en hastighetsmåling og forespørsel av bedre nett på nettsiden, så må brukeren også godkjenne at vi henter inn brukeren posisjon. Ved første gangs bruk av appen må brukeren godkjenne at vi bruker posisjonen dems senere. Hvis brukeren velger å hoppe over dette steget her, vil meldingen dukke opp igjen når appen prøver å bruke posisjonen senere.

3.1.2 Personas

Vi benytter oss av personas for å skape oss et bilde av hvilke mennesker som skal bruke plattformen vår. Våre personas er basert på informasjon vi har fått fra oppdragsgiver og våre egne erfaringer fra turen vår til Elverum hvor vi fikk møte mennesker fra de forskjellige brukergruppene. Personasene våre skal gi et ansikt til våre målgrupper og være med på å vise hvordan deler av vår funksjonalitet kan hjelpe de enkelte gruppene. Alle menneskene i personasene er fiksjo-nelle karakterer.



Elenor Martinsen

Alder: 73 år
 Bor: Koppang, Hedmark
 Yrke: Pensjonist
 Sivilstatus: Enke

PERSONLIGHET

- Omtenksom
- Pratsom
- Redd for nye ting

INTERESSER

- Kunst
- Strikking
- Barnebarn

Innbygger

“Jeg skulle ønske jeg kunne tilbringe mer tid sammen med barnebarnene mine.”



Teknologi

FERDIGHETSnivå

IT og internett

Hvordan bredbånd fungerer

Mobilapper

ENHETER OG PLATFORMER



Problembeskrivelse

Elenor Martinsen bor alene i Koppang. Hun har to døtre, en i Ålesund og en i Elverum. Hvis det er en ting Elenor virkelig bryr seg om så er det barnebarna sine, de setter alltid et smil på ansiktet hennes. På grunn av avstanden til Ålesund ser hun svært sjeldent alle barnebarna sine.

Moren til barnebarna har informert henne om at de kan ha en videosamtale over internett. Elenor har brukt mobilt bredbånd til dette, men det ble dyrt. Elenor ønsker nå å skaffe seg bredbånd, men vet ikke helt hvor å starte.

Figur 3.1: Personas: Elenor Martinsen

Bildet i figuren ovenfor er hentet fra Flickr[15] under Creative Commons 2.0.



Johan Fredriksen

Alder: 41 år

Bor: Brumunddal, Hedmark

Yrke: Dataelektroniker

Sivilstatus: Gift

PERSONLIGHET

- Ansvarsfull
- "Arbeidsnarkoman"
- Utmattet

INTERESSER

- Teknologi
- Fotball
- Jakt

Innbygger

"Jobb krever mye av energien min, men ikke i nærheten av så mye som å være tennåringsfar"



Teknologi

FERDIGHETSNIVÅ

IT og internett


Hvordan bredbånd fungerer


Mobilapper


ENHETER OG PLATTFORMER



Problembeskrivelse

Johan bor på utsiden av Brumunddal med sin kone Henriette, hans to tenårings sønner på 14 og 17 år og en datter på 16. Sønnene hans tilbringer mesteparten av tiden på rommene sine hvor de spiller dataspill med venner. Dette sluker opp store deler av båndbredden og Johan hører mye klaging fra datteren som også vil bruke nettet. Familien har rett og slett blitt for stor for deres nåværende plan.

Johan har prøvd å sjekke ut hvilke tilbud som finnes, men har ikke funnet noen leverandører som gir noe bedre tilbud. Johan skulle gjerne blitt kvitt maset fra familien, og han kunne tenkt seg fiber. Johan har meldt interesse på nettleverandøren sin hjemmeside, men har ikke hørt noe mer etter det.

Figur 3.2: Personas: Johan Fredriksen

Bildet i figuren ovenfor er hentet fra Flickr[16] under Creative Commons 2.0.



Aksel Overvik

Alder: 33 år
 Bor: Ridabu, Hedmark
 Yrke: Fotograf
 Sivilstatus: I et forhold

PERSONLIGHET

- Engasjert
- Arbeidsom
- Eventyrlysten

INTERESSER

- Fotografi
- Lokal politikk
- Reise

Ildsjel

“Alle problemer kan løses”



Teknologi

FERDIGHETSnivå



ENHETER OG PLATTFORMER



Problembeskrivelse

Aksel Overvik er et engasjert menneske som liker å prøve seg ut på nye ting. Aksel liker å reise, men hjemme vil alltid være best.

For Aksel er lokal politikk veldig viktig siden han har bodd i området rundt Mjøsa i alle år. Han trives her og har ingen intensjoner om å flytte bort. Aksel har gjennom tidene hatt flere verv hvor han har jobbet aktivt for å bedre folks trivsel i nærmiljøet.

Aksel vet egentlig ikke så veldig mye om bredbånd, men som vanlig velger han å engasjere seg.

Figur 3.3: Personas: Aksel Overvik

Bildet i figuren ovenfor er hentet fra Flickr[17] under Creative Commons 2.0.



Oskar Alstad

Alder: 39 år
 Bor: Lillehammer, Oppland
 Yrke: Moderniseringsdirektør
 Sivilstatus: Gift

PERSONLIGHET

- Utadvendt
- Glad
- Nytenkende

INTERESSER

- Premier League
- Sykling
- Skøyter

Leverandør

“Tilgangen på internett burde ikke forhindre deg fra å bo der du ønsker å bo”



Teknologi

FERDIGHETSnivå

IT og internett

Hvordan bredbånd fungerer

Mobilapper

ENHETER OG PLATTFORMER



Problembeskrivelse

Oskar bor i Lillehammer sammen med dama. Oskar er en aktiv mann som er glad i friluftslivet. Egentlig foretrekker Oskar å bo på bygda, men på grunn av jobb har han slått seg ned i byen.

Oskar mener at tilgangen på internett ikke skal forhindre deg fra å bo der du selv ønsker å bo. Han kjenner selv til fra egne erfaringer viktigheten av å være tilgjengelig. Mange yrker gir deg nå muligheten til å arbeide hjemefra ved hjelp av internett. Oskar mener at ved å bedre tilgangen på nett kan flere bo der de ønsker.

Figur 3.4: Personas: Oskar Alstad

Bildet i figuren ovenfor er hentet fra Flickr[18] under Creative Commons 2.0.



Katrine Olsen

Alder: 36 år
 Bor: Elverum, Hedmark
 Yrke: Planlegger
 Sivilstatus: Gift

PERSONLIGHET

- Reflektert
- Moden
- Positiv

INTERESSER

- Litteratur
- Maling
- Familie

Kommunerepresentant

“Vi trenger mer informasjon for å komme i mål”



Teknologi

FERDIGHETSnivå



ENHETER OG PLATTFORMER



Problembeskrivelse

Katrine bor på Elverum sammen med sin mann og barnet deres. Katrine har i flere år arbeidet for kommunen. Nå om dagen jobber hun aktivt med kartlegging av utbygging av bredbånd i kommunen.

Katrine synes dette er krevende og vanskelig fordi det er mye usikkerhet rundt hvor det allerede er tilbud og hvor det ikke er det. Mangelen på åpen dokumentasjon er til stadig til frustrasjon. Hun skal også prøve å gjøre utbygningen så kostnadseffektiv som mulig, men dette er også vanskelig uten nok informasjon. Dette gjør at ting tar lang tid.

Figur 3.5: Personas: Katrine Olsen

Bildet i figuren ovenfor er hentet fra Flickr[19] under Creative Commons 2.0.

3.2 Use Case

3.2.1 Aktører

3.2.1.1 Kommune

Kommune skal kunne opprette et prosjekt i et område hvor det ønskes å søke om stønad for utbygging. Dette gjøres ved hjelp av et kart der hvor de kan markere områder hvor det ønskes å bygge ut bredbånd. Markeringen skal kunne ha deler som ikke er en del av markeringen, slik at områder som allerede har bra nettverkstilkobling kan fjernes fra området.

Kommunen kan også legge inn hvor mye de ønsker å støtte prosjektet med per husstand, og se hva leverandør og fylkeskommune har sagt at de skal bidra med. Kommunen skal kunne legge inn informasjon på vegne av en leverandør, dette vil være informasjon om en leverandørs sjablongverdi, og dokumentasjon på om en leverandør kommer til å bygge ut i området.

3.2.1.2 Hedmark Fylkeskommune

Fylkeskommunen kan legge inn informasjon på et prosjekt om hvor mye de ønsker å støtte prosjektet med, eller oppdatere informasjon på vegne av en kommune eller en leverandør.

Fylkeskommunen vil også ha ansvar for å godkjenne nye kommune eller leverandør brukere.

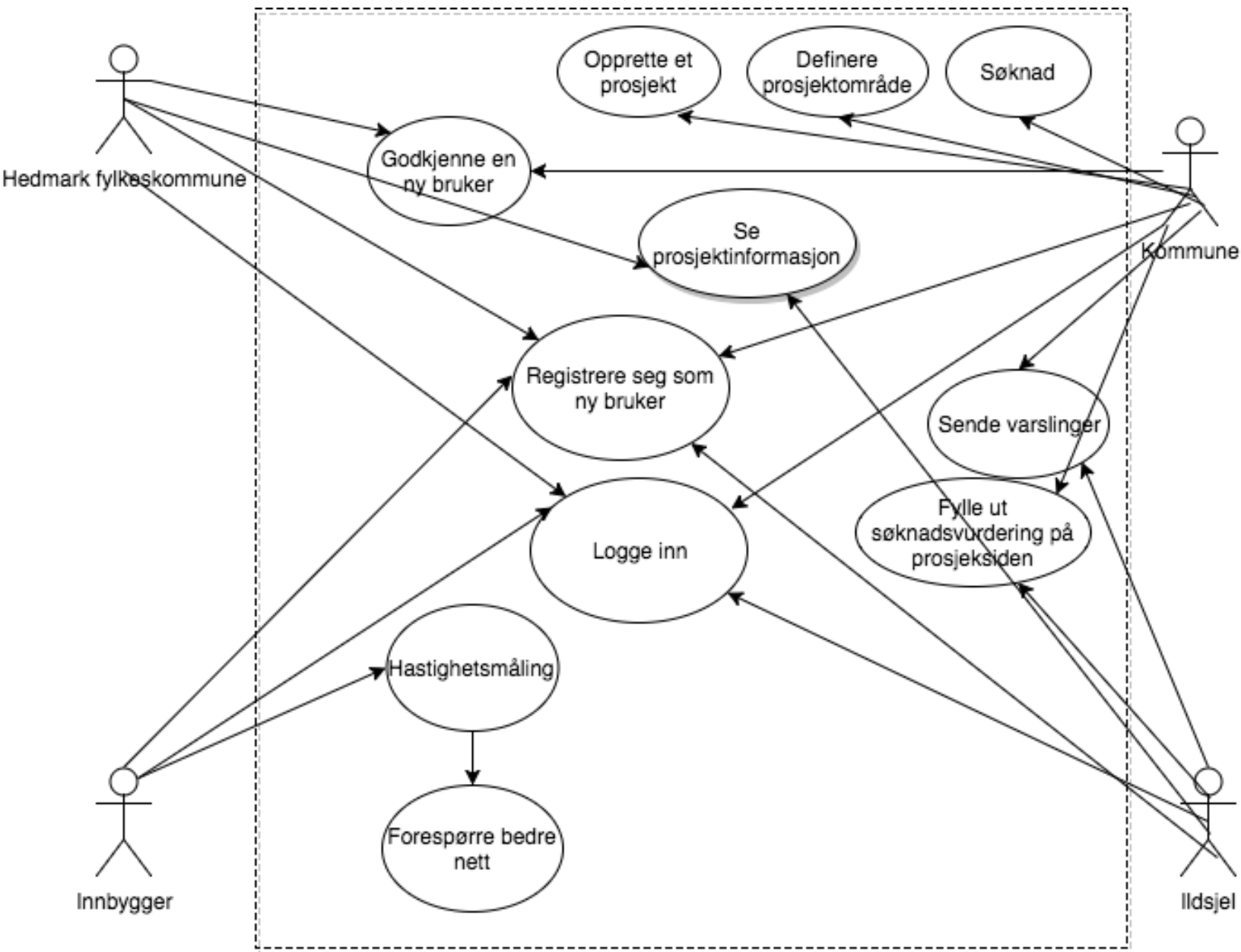
3.2.1.3 Ildsjel

Ildsjeler skal ha tilgang til et meldingssystem på et prosjekt som de kan bruke til å kommunisere med kommunen. De vil også ha mulighet til å legge inn hvor mye de vil spare per husstand eller totalt ved dugnad, og legge inn bedrifter i området som ønsker bedre bredbånd.

3.2.1.4 Innbygger

En innbygger skal kunne utføre hastighetsmålinger, forespørre bedre nett, innhente informasjon angående pågående prosjekter, motta varslinger for prosjekter og få kart med oversikt over hastigheter på bredbånd i et område.

3.2.2 Use Cases



Figur 3.6: Use Case Diagram

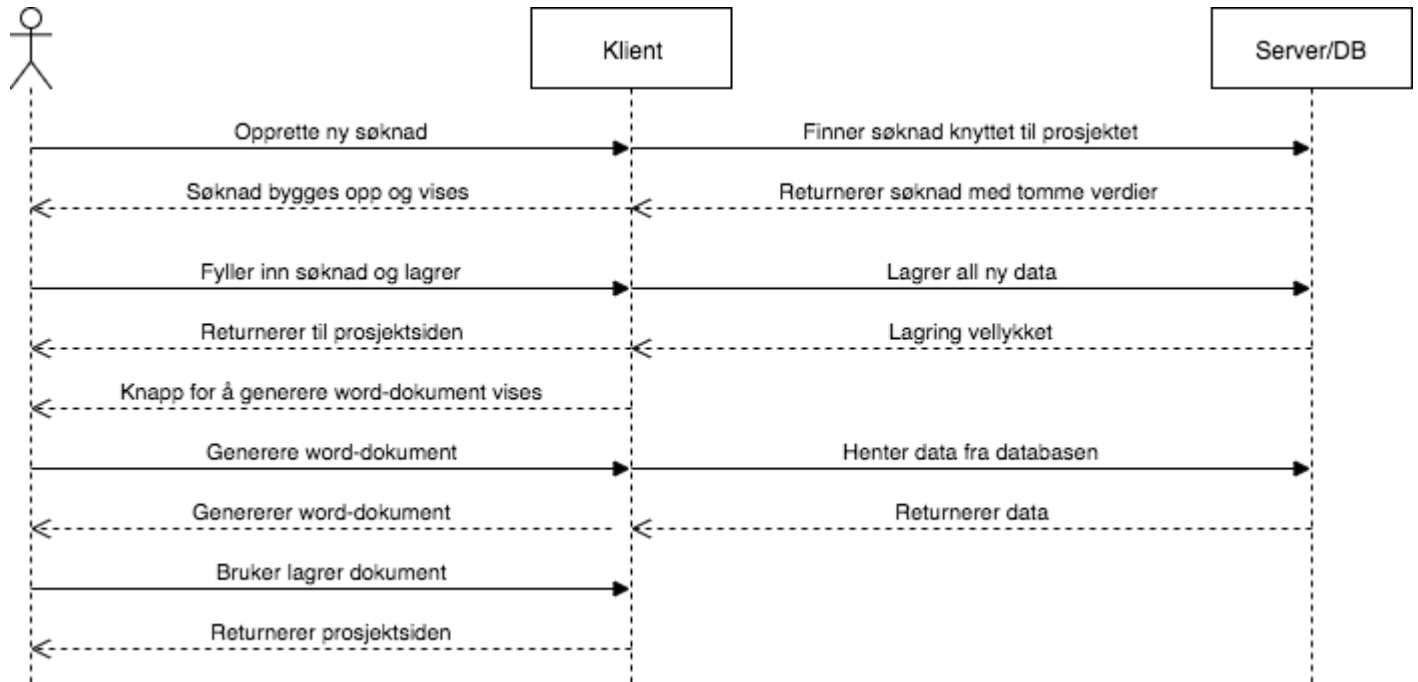
Use Case	Hastighetsmåling
Aktør	Innbygger
Mål	Se hastighet
Beskrivelse	Brukeren kan bruke appen eller nettsiden til å utføre en hastighetsmåling. Det vil være informasjon om at hva brukeren kan gjøre for å få best mulig målingsresultater. De vil få opp en progress bar som viser hvor lenge som er igjen av målingen og siden vil oppdatere hastigheten fortløpende. Det vil bli målt responstid, nedlastningshastighet og opplastingshastighet. Vi vil også finne brukerens IP-adresse og leverandør.
Use Case	Logge inn
Aktører	Hedmark fylkeskommune, kommune, ildsjel, innbygger
Mål	Man ønsker tilgang til å utføre handlinger som ikke er tilgjengelig for alle, eller er forskjellig for hver enkelt rolle.
Beskrivelse	Alle skal ha mulighet til å logge inn på nettsiden. Innbyggere skal kunne logge inn med epost og passord, andre brukere logger inn med minID.
Use Case	Registrere bruker
Aktører	Hedmark fylkeskommune, kommune, ildsjel, innbygger
Mål	En person som ikke allerede har en bruker ønsker tilgang til å utføre handlinger som ikke er tilgjengelig for alle, eller er forskjellig for hver enkelt rolle.
Beskrivelse	På nettsiden skal det være mulighet for at ansatte hos Hedmark fylkeskommune, kommunene i Hedmark, innbyggere eller ildsjelene å registrere seg.

Use Case	Godkjenne ny bruker
Aktør	Hedmark fylkeskommune, kommunebrukere
Mål	Ønsker å gi tilgang til en bruker som har registrert seg.
Beskrivelse	Hedmark fylkeskommune og kommunebrukere vil ha en side der de kan se alle som har registrert seg. De vil kunne se kontakt informasjonen til brukeren, og godkjenne eller avvise brukeren. Først når brukerne er godkjent vil de kunne få tilgang til ytterligere informasjon.
Use Case	Opprette prosjekt
Aktør	Kommune
Mål	Man ønsker å kunne ha informasjon om et prosjekt samlet, mens det planlegges, søkes og bygges ut.
Beskrivelse	En kommune skal kunne opprette et prosjekt for et område der det ønskes bedre bredbånd. Det skal kunne spesifiseres et navn på prosjektet, men for å gjøre det mer effektivt fylles det ut et automatisk prosjektnavn som standard. De blir deretter sendt videre til å definere området.
Use Case	Definere prosjektområde
Aktør	Kommune
Mål	Man ønsker å definere hvilket område prosjektet gjelder, slik at denne informasjonen kan brukes i en søknad og som informasjon til brukere.
Beskrivelse	Kommunen skal kunne definere et prosjektområde ved hjelp av et kart. Når en kommune har markert et område på kartet, vises husstandene som ligger innenfor dette området.

Use Case	Fylle ut prosjektinformasjon
Aktører	Kommune
Mål	Man ønsker å ha felles informasjon om et prosjekt slik at denne kan være tilgjengelig for innbyggere eller brukes i søknad til Nkom.
Beskrivelse	Leverandørene skal kunne legge sin sin sjablongverdi for utbygging i området og gi informasjon om de planlegger å bygge ut i området de neste årene.
Use Case	Se prosjektinformasjon
Aktører	Hedmark fylkeskommune, ildsjeler
Mål	Gjøre det mulig å se hva kommunen legger inn som informasjon
Beskrivelse	Hedmark fylkeskommune og ildsjeler skal kunne se informasjon om et prosjekt som kommunene har lagt inn. Hedmark fylkeskommune kan se alle prosjektinformasjoner, mens ildsjeler kan bare se prosjekter de har fått tilgang til.
Use Case	Sende varslinger
Aktører	Kommune, ildsjeler
Mål	Man ønsker å informere innbyggere i et prosjektområde om status på et prosjekt, eller annen informasjon angående prosjektet.
Beskrivelse	En kommune og ildsjeler kan sende informasjon om et prosjekt til de brukerne som har sagt at de ønsker varslinger.

3.2.3 Extended Use Case

Use Case	Søknad
Aktører	Kommune
Mål	Sende en søknad til Nkom som skal gi midler til utbygging av bredbånd i området.
Beskrivelse	Ansatte i en kommune skal kunne samarbeide om å opprette en søknad. De skal kunne fylle ut tekst felt som blir satt sammen i en endelig søknad.
Pre-betingelse	Brukeren må være logget inn som kommunne og ha opprettet et prosjekt som det ønskes å generere søknad til.
Post-betingelse	Brukeren vil ikke lenger kunne endre søknaden
Hendelsesforløp	<ol style="list-style-type: none"> 1. Opprette søknaden fra prosjekt siden. 2. Ansatte i kommunen kan fylle ut feltene i søknaden på nettsiden. 3. Brukerne kan lagre søknaden, slik at de eventuelt kan jobbe videre med denne senere. 4. Det er en knapp som lar dem laste ned søknaden som et word dokument



Figur 3.7: System sekvensdiagram for søknad

Use Case	Forespørre bedre bredbånd
Aktør	Innbygger
Mål	En innbygger ønsker å informere om at de ikke er fornøyd med deres nåværende bredbånd og ønsker bedre bredbånd.
Beskrivelse	En innbygger skal kunne utføre en hastighetsmåling som vil gi dem informasjon om hastigheten på nettverket deres. Deretter vil de få mulighet til å forespørre bedre nett.
Pre-betingelse	Brukeren må laste ned appen, eller åpne nettsiden.
Post-betingelse	Brukeren har mulighet til å logge inn.
Hendelsesforløp	<ol style="list-style-type: none"> 1. Når en bruker trykker på knappen for å starte hastighetmålingen vil de få informasjon om at de burde slå av ting som bruker nettet og lignende før de går videre for å utføre målingen. 2. Målingen vil vise en fremdriftsindikator som viser hvor mye som er igjen av målingen. 3. Når målingen er ferdig vil brukeren ha mulighet til å forespørre bedre bredbånd eller eventuelt avslutte målingen. Data fra hastighetsmålingen lagres automatisk. 4. Dersom de ønsker å forespørre bedre bredbånd, blir de bedt om å fylle inn informasjon i skjemaet. 5. De får foreslått en adresse basert på deres posisjon, dersom vi har tilgang til denne, men brukeren kan endre dette. Det vil også være en mulighet for å velge at adressen tilhører en bedrift eller fritidsbolig. 6. Brukeren blir bedt om å fylle ut telefon nummer og epost adresse. 7. Deretter vil brukeren få mulighet til å velge om de skal motta varslinger på sms, epost eller begge deler.

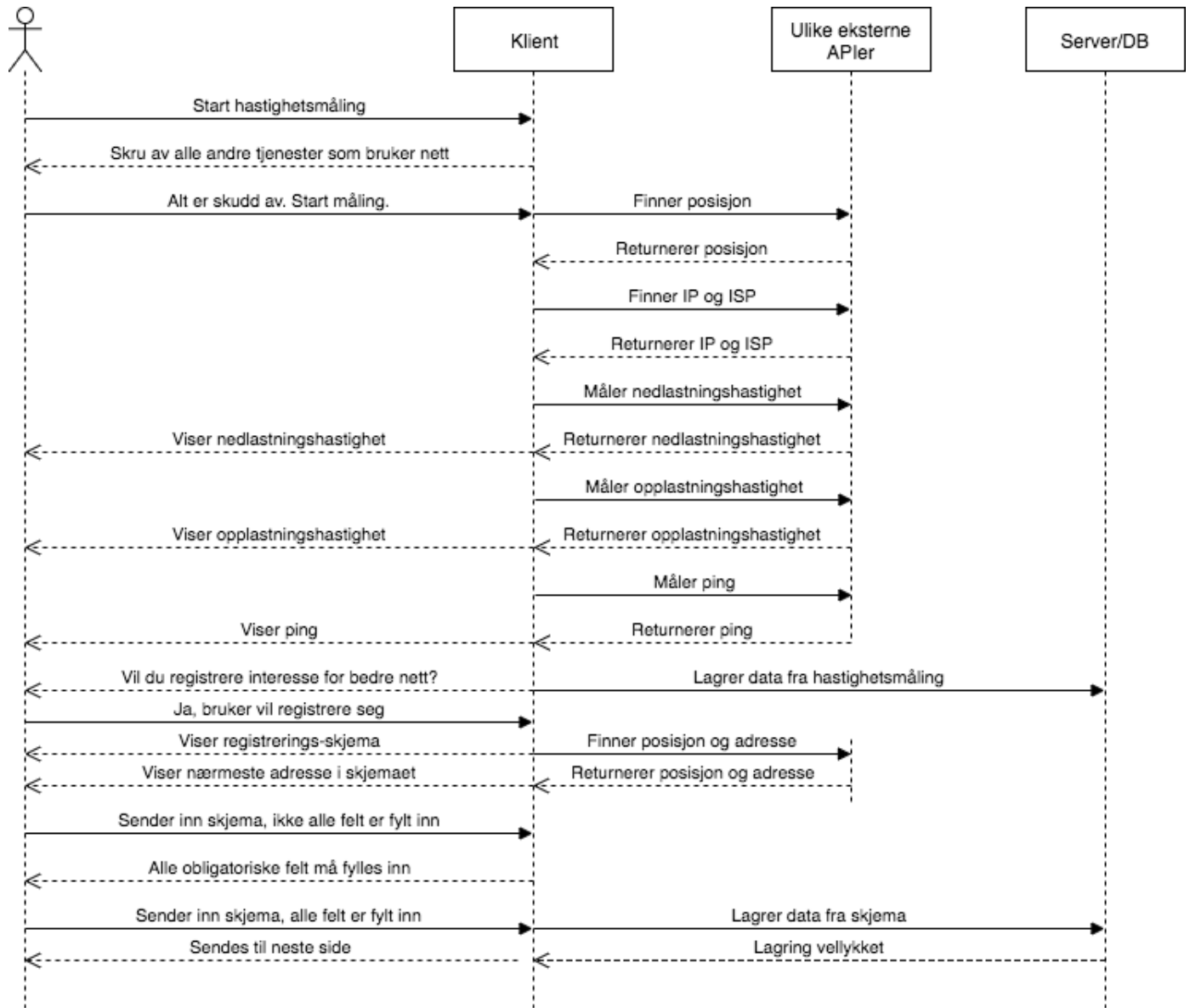
3.3 Sikkerhet

Hedmark Fylkeskommune stiller en rekke sikkerhetskrav. For å kunne se informasjon om forespørsler, andres målinger, bruker eller prosjektinformasjon kreves det at personen er logget inn med ID-porten. Dette er for å sikre at enkeltpersoner ikke kan få informasjon de ikke skal ha krav på.

ID-porten identifiserer en person ved hjelp av fødsels- og personnummer. Vi skal lagre en kryptert versjon av denne informasjonen, slik at man ikke skal kunne finne tilbake til fødsels- og personnummer dersom man får tilgang til systemet vårt.

De stiller det samme kravet til kryptering av passord, for brukere som ikke logger inn ved hjelp av ID-porten.

I tillegg til dette krever fylkeskommunen at alle brukere skal bli godkjent av enten en annen ansatt i samme kommune, eller av fylkeskommunen før de får tilgang til informasjon.



Figur 3.8: System sekvensdiagram for forespørre bedre bredbånd

4 *Design og arkitektur*

I dette kapitlet tar vi for oss designet og arkitekturen for systemet vårt. Vi har gjort en rekke teknologivalg som har vært avgjørende for arkitekturen vår. I kapitlet tar vi for oss teknologivalgene og systemarkitekturen. Til slutt ser vi på design gjennom en tidlig prototype, universell utforming og vår egen fargeprofil.

4.1 **Teknologivalg**

4.1.1 **App**

Det første valget som måtte gjøres før vi kunne starte å utvikle appen var om vi skulle utvikle **native**, gjennom Google og Apple, eller om vi skulle utvikle appen gjennom et tredjeparts framework. Å utvikle nettsiden med et responsivt design som var tilpasset mobilstørrelse var ikke aktuelt da oppgaven spesifikt gikk ut på å lage app til iPhone og Android.

Ved å utvikle native har man en stor fordel med å kunne utnytte telefonen til det fulle. Det er ingen begrensninger som man ikke ville ha møtt i et tredjeparts frameworks, og de nyeste funksjonalitetene er alltid tilgjengelige. I tillegg hadde alle gruppe medlemmene på et vært borti utvikling til minst en av platformene. En ulempe som ble avgjørende i vårt valg var at **native** utvikling krever at appene blir utviklet separat, noe som tar mer tid.

Vi anså på dette tidspunktet prosjektet som så stort at det kunne bli lønnsomt å heller utvikle appene i et tredjeparts framework med støtte for cross-platform utvikling. På dette tidspunktet hadde vi allerede avgjort at prosjektet ikke hadde noen særskilte behov som krevde å bli skreddersydd gjennom **native** utvikling.

Det er mange frameworks som støtter cross-platform utvikling, men det var særlig to fram-

eworks vi så på: Xamarin og React Native. Dette skyldes at vi ønsket å utvikle en app som hadde den samme utførelsen og designet over seg som man finner i en app utviklet native.

	Xamarin	React Native
Cross-platform	Ja	Ja
Språk	C#	JavaScript
År	2011	2015
Delt kode på tvers av platform	Ja	Ja
Støtte for Java	Gjennom automatisk binding	Gjennom moduler
Støtte for Objective-C og/eller Swift	Gjennom automatisk binding	Gjennom moduler
IDE	Xamarin Studio eller Visual Studio	Ingen

Tabell 4.1: Sammenligning av Xamarin og React Native

Xamarin og React Native tilbyr mye av de samme funksjonalitetene. Vi bestemte oss for å bruke Xamarin da det har eksistert lengre og er mer modent. I tillegg skrives Xamarin i C# og bruker .NET bibliotekene, som vi selv ønsket å få erfaring med.

Det neste steget var så å bestemme oss for om vi skulle lage et Xamarin Forms Shared prosjekt eller et Xamarin Forms PLC prosjekt. I et Xamarin Forms Shared[20] prosjekt har platformene en delt kodebase. Ved å bruke [conditional compilation](#), kompilering der man sender med et parameter for å differensiere platformen, kan man lage platform spesifikk funksjonalitet. I et Xamarin Forms PCL[21] prosjekt benytter man seg av PCL biblioteker. Bibliotekene er allerede programert opp mot de forskjellige platformene slik at man ikke selv blir nødt til å kode det hver for seg på platformene. For å programmere inn platform spesifikk funksjonalitet blir man nødt til å bruke [dependency injection](#) gjennom et interface.

Shared prosjekter brukes gjerne dersom man har et behov for å utføre en del platform spesifikke handlinger. Vi skulle i vårt prosjekt benytte oss av GPS, nedlastning, opplastning og ping. Vi mente på at dette ikke krevde å bli håndtert forskjellige på iOS og Android. Dermed valgte vi å lage et Xamarin Forms PCL prosjekt.

4.1.2 Frontend på nettsiden

Da vi skulle velge frontend rammeverk sto det mellom to stykker, Angular 2 og React. Vi bestemte tidlig at vi ville bruke en av disse ettersom vi viste at dette var to av de mest populære rammeverkene på markedet og erfaring i disse er ønskede kvalifikasjoner hos fremtidige arbeidsgivere. Etter å ha lest litt om begge rammeverkene og om andres erfaring med dem fant vi ut at de tilbyr mye av den samme funksjonaliteten. Vi så ingen spesielle grunner til at den ene skulle passe bedre til vårt prosjekt enn den andre. Valget falt litt tilfeldig på Angular 2 ettersom dette rammeverket var det nyeste og det virket spennende å prøve. Vi har valgt å bruke Angular CLI som er et kommandolinjegrensesnitt for Angular. Angular CLI gjorde arbeidet enklere ettersom vi kunne generere komponenter, stier og testfiler med bare noen enkle kommandoer. Angular CLI bruker Karma for enhetstesting, derfor var det et naturlig valg. Vi viste også at vi ville bruke et rammeverk for å hjelpe oss med utviklingen av brukergrensesnittet. Her valgte vi å bruke Bootstrap. Bootstrap er et verktøy som inneholder CSS, HTML og Javascript komponenter for å gjøre utviklingen av nettsider raskere og mer gjennomført. Vi valgte å bruke Bootstrap ettersom alle utviklerne på dette prosjektet hadde erfaring med det tidligere, det har støtte for de fleste nettleserne og det er et populært rammeverk med god dokumentasjon og hjelp. Les mer om hvordan vi brukte Angular 2, Angular CLI, Karma og Bootstrap i [kapittel 5](#) og [kapittel 6](#).

4.1.3 Backend

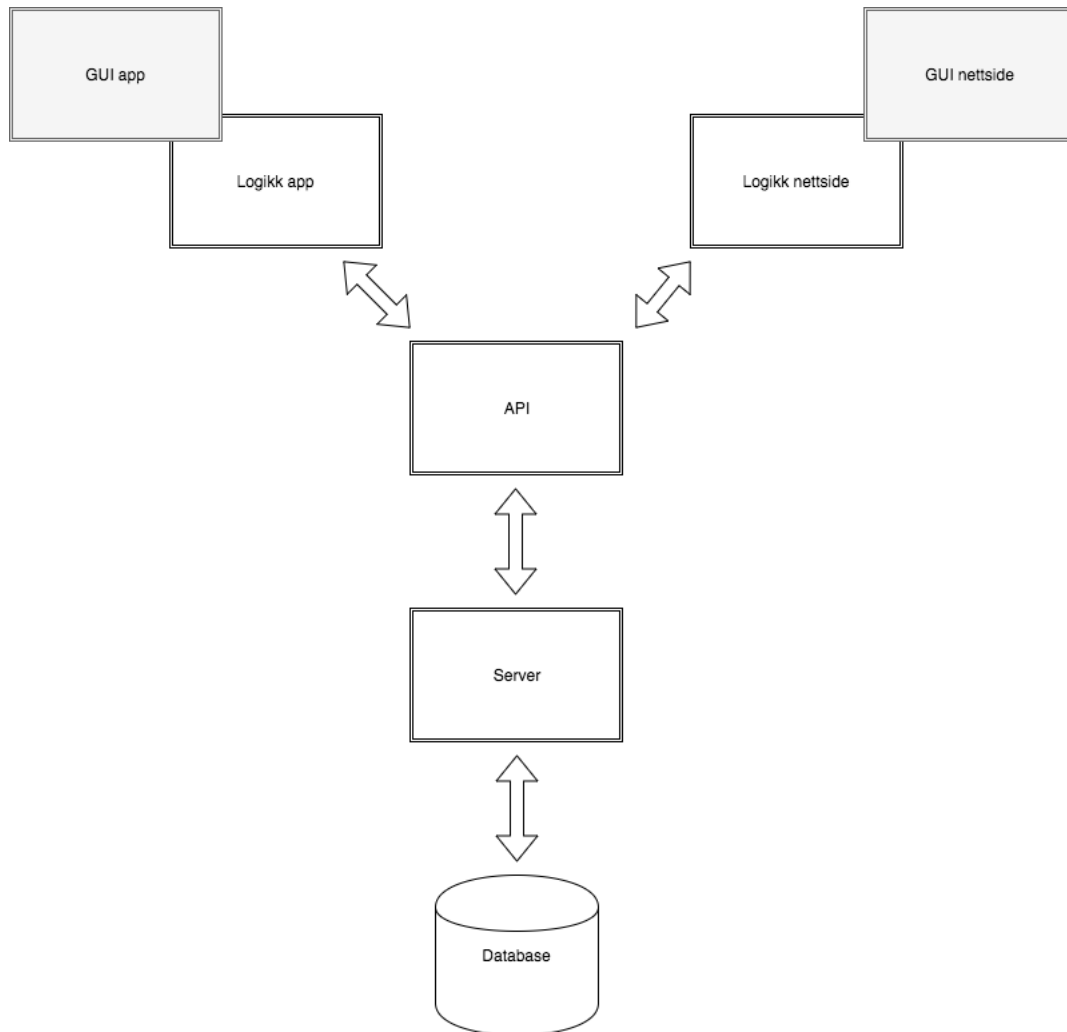
Da vi skulle velge backend rammeverk sto vi mellom fire ulike, .NET, Node.js, Django og Ruby on Rails. Lasse hadde erfaring med å jobbe med Ruby on Rails, men resten av gruppa hadde ingen erfaring med backend rammeverk. Vi snakket om fordeler og ulemper knyttet til alle fire rammeverkene og vi vil her nevne noen av de tingene vi diskuterte rundt de rammeverkene vi ikke valgte. .NET så vi kunne være veldig aktuelt til senere arbeidsliv, men ingen hadde noen erfaring med C# eller .NET fra tidligere. Alle hadde tidligere brukt javascript som blir brukt i Node.js, men ulempen var at det var en litt annerledes måte å tenke på på grunn av callback funksjoner. Fordelen til Django var at det er mye fokus på dokumentasjon og det er mange store sider som for eksempel Pinterest som bruker det. Ulempen var at ingen hadde noe erfaring med språket Python. Vi valgte til slutt Ruby on Rails ettersom vi så at det var mange positive fordeler

ved å velge det rammeverket. Fordelen med Ruby on Rails var at det er enkelt å lære, raskt å programmere i og Lasse hadde erfaring fra tidligere slik at han kunne være til hjelp for gruppa når andre sto fast i kodingen.

4.1.4 Database

Da vi skulle velge database sto vi mellom to ulike typer, MySQL og PostgreSQL. Vi hadde alle tre tidligere kjennskap til MySQL ettersom vi hadde brukt dette i tidligere emner på skolen, så i utgangspunktet var dette første valget. Men vi fant ut at vi trengte en database som hadde gode funksjonaliteter for geografi ettersom vi skulle ha mange kart-funksjonalitet blant annet knyttet til polygoner. Vi fant etterhvert ut at MySQL hadde litt begrenset funksjonalitet på dette i forhold til PostgreSQL. Derfor valgte vi PostgreSQL i stedet for. PostgreSQL har en stabil og rask database-motor som er mye brukt. PostgreSQL har et tilleggs program, PostGIS, som tilbyr mange og bra funksjonaliteter for geografi.

4.2 Systemarkitektur



Figur 4.1: Logisk inndeling av vårt system

Det øverste nivået av vårt system er delt inn i to deler, en del for appen og en del for nettsiden. På dette nivået finnes GUI, det grafiske brukergrensesnittet for brukeren. I dette laget finnes også logikken bak det grafiske brukergrensesnittet som inneholder all funksjonalitet og som håndterer alle forespørsler til og fra API-et. På neste nivå i vår modell finner vi API-et. API-et vil motta spørringer fra frontend-logikken og vil formatere og sende forespørslene riktig videre. API-et vil så bruke serveren til å hente og lagre data fra databasen. All kommunikasjon til og fra databasen håndteres av serveren. De ulike delene av vårt system blir nærmere forklart i [kapittel 5](#).

4.3 Prototype

Tidlig i planleggingsfasen dro vi på seminar i Elverum for å få et innblikk i hvordan fylket jobbet sammen med kommunene for å bygge ut bredbåndstilbudet. Til denne anledningen lagde vi en rask prototype for å presentere systemet som vi skulle være med på å utvikle. Dette gjorde vi i verktøyet Pingendo[22], som hjelper deg med å lage bootstrap prototyper.

Vi bestemte oss for at vi ville ha med forsiden, hastighetsmålingen, varmekartet og deler av det en kommune skulle kunne se inne på en prosjektside i prototypen vår. Dette var fordi flertallet av de vil skulle presentere ideen for var kommuneansatte. Det ville da være viktig å ha med hastighetsmålingen og kartet for å vise hva brukerne kunne gjøre, og hvordan denne informasjonen ble relevant inne på prosjektsidene hvor kommuneansatte skal arbeide.

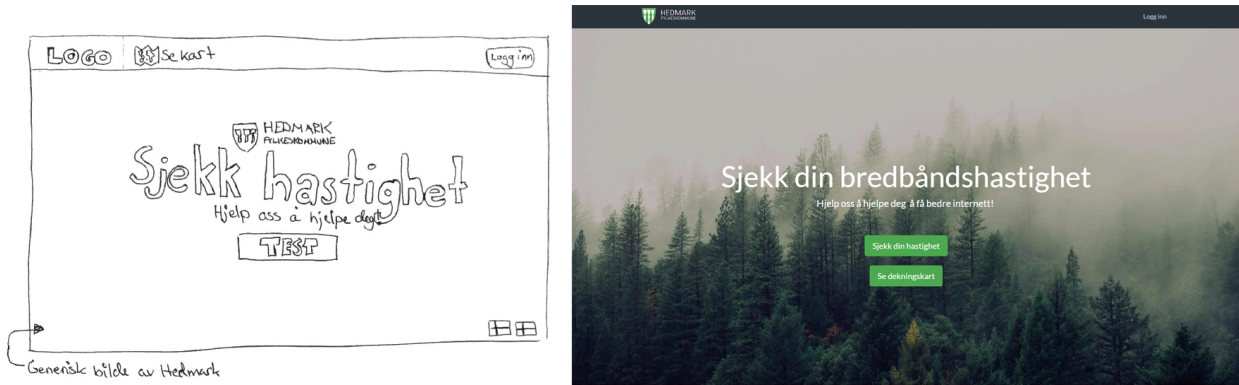
Prototypen har vært med på å gi oss tilbakemelding fra produkteier at vi har forstått oppgaven og jobber i riktig retning. Den har også vært med på å gi oss et utgangspunkt å arbeide ut fra under utviklingsfasen. Prototypen ble lagd når vi fremdeles var i planleggingsfasen og gjenspeiler derfor ikke alle valg som er tatt i senere tid. Den gjenspeiler ikke fargeprofilen i dette kapitlet.

4.3.1 Lo-fi prototype

Det første vi gjorde var å å lage grove skisser av prototypen på ark, så kalt lo-fi prototype. Se [vedlegg B](#). Under lagingen av skissene brainstormet vi hvilke funksjonaliteter hver enkelt side måtte inneholde. Deretter prøvde vi å hver for oss å sette inn funksjonaliteten på siden på en ryddig måte. Noen av skissene er frittstående, andre peker direkte videre til andre skisser.

4.3.2 Hi-fi prototype

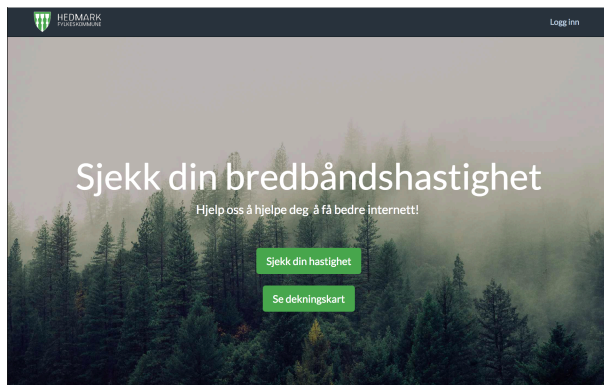
Når vi begynte å lage hi-fi prototypen tok vi utgangspunkt i skissene i lo-fi prototypen vår. Prototypen kan finnes som vedlegg i zip-filen.



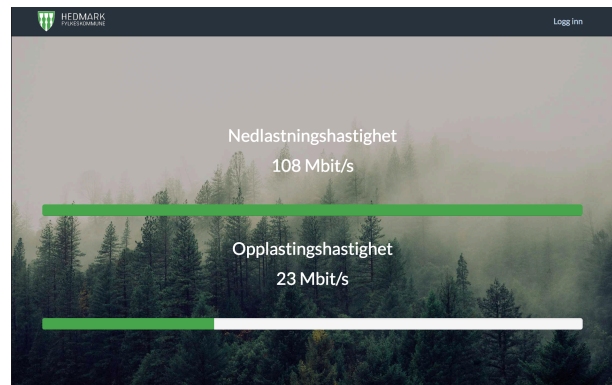
Figur 4.2: Sammenligning av lo-fi og hi-fi prototype

Prototypen er statisk og imiterer at den har data. Enkelte steder utfører den simpel JavaScript for å gi en mer ekte følelse. Prototypen er klikkbar og fører brukeren videre til tilsvarende sider.

Fra forsiden kan man trykke seg inn på "Sjekk din hastighet". Da kommer man inn på hastighetsmålingen.



Figur 4.3: Prototype: Forside



Figur 4.4: Prototype: Hastighetsmåling

Når JavaScripten har fullført en falsk måling blir man automatisk navigert til en resultatside. Her kan man trykke på "Jeg ønsker bedre bredbånd" og bli navigert til en side hvor man skal fylle ut en forespørsel.

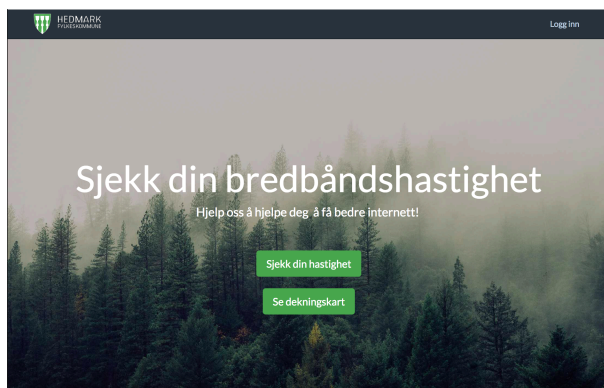


Figur 4.5: Prototype: Resultater

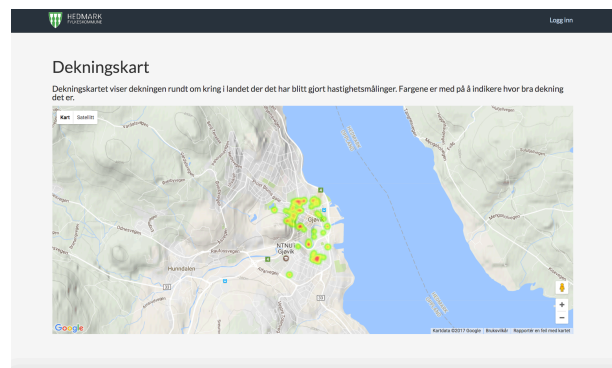


Figur 4.6: Prototype: Forespørsel

Fra forsiden kan man også navigere til varmekartet ved å trykke på "Se dekningskart". Kartet er et Google Maps kart hvor fargene er med på å indikere om det er høy eller lav hastighet i området.

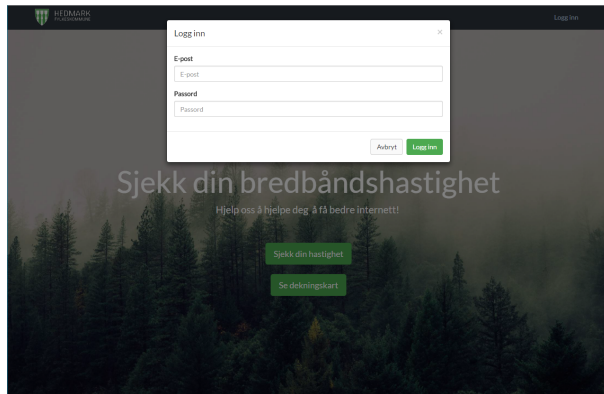


Figur 4.7: Prototype: Forsiden

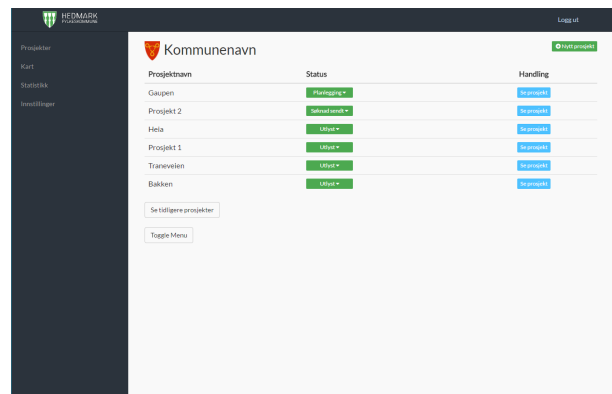


Figur 4.8: Prototype: Varmekart

Når man er på forsiden kan man trykke på knappen "Logg inn" i høyre hjørne. Da vil en tekstboks dukke opp og man kan logge på. I prototypen vår later vi som om man logger på med en kommune-bruker. Da får man opp en kommune-forside hvor man har oversikt over pågående prosjekter i kommunen.

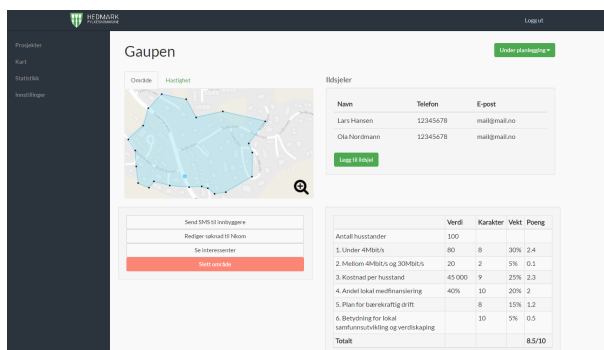


Figur 4.9: Prototype: Pålogging fra forsiden

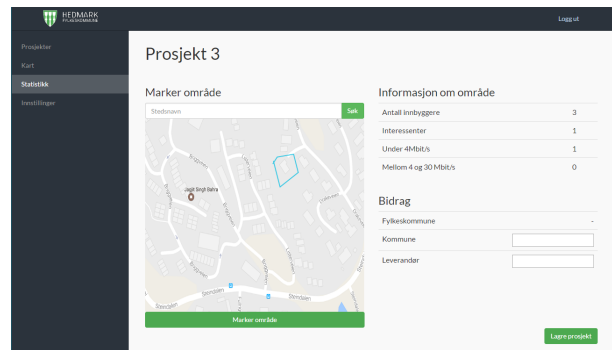


Figur 4.10: Prototype: Kommune-forside

Fra kommune-forsiden har man to valg. Man kan enten gå inn og se på det øverste prosjektet kalt "Gaupen" ved å trykke på "Se prosjekt", eller man kan opprette et nytt prosjekt ved å trykke på "Nytt prosjekt" knappen øverst i høyre hjørne. Inne på det nye prosjektet kan man velge å markere et område på kartet som dukker opp. Dette gjøres gjennom JavaScript. Etter et område er markert dukker det opp falske tallverdier som skal representere det valgte området.



Figur 4.11: Prototype: Prosjektet "Gaupen"



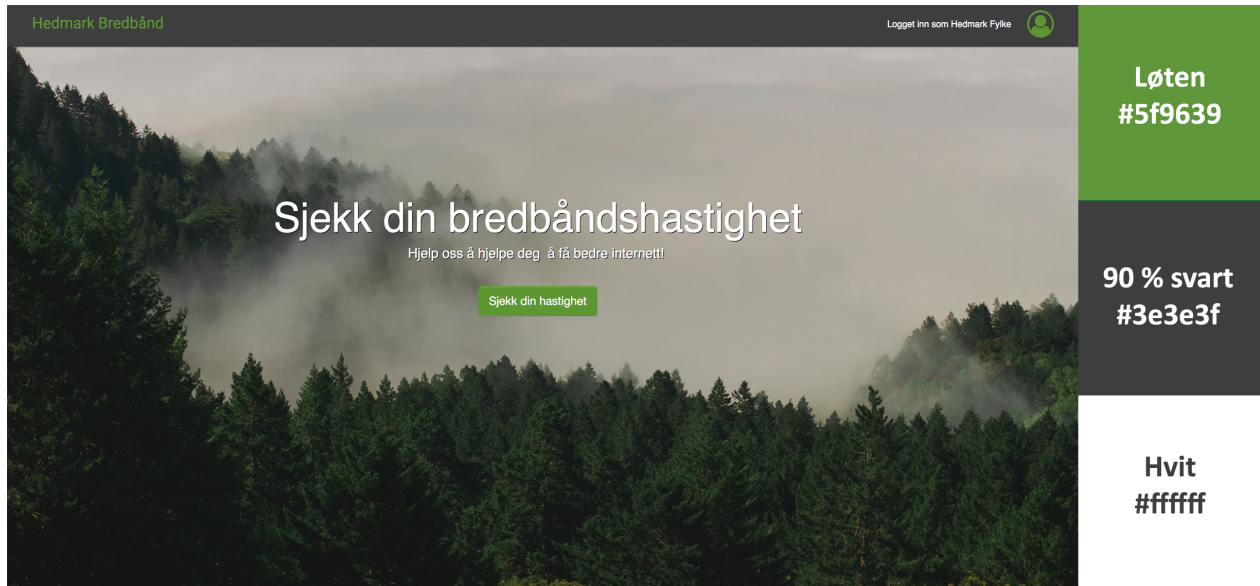
Figur 4.12: Prototype: Nytt prosjekt

4.4 Fargeprofil

Fra Hedmark fylkeskommune mottok vi dokumentet "Profilmanual for Hedmark fylkeskommune". I dette dokumentet tar de for seg fargene, profil-elementene og malene som de selv bruker ved utforming av alt fra nettsider til offisielle klær. Ved å ta utgangspunkt i denne har vi lagd vår egen forenklete fargeprofil. I vedleggene finnes et utdrag fra denne manualen (se [vedlegg C](#)). Dette utdraget inneholder de sidene som handler om farger. Vi bruker samme navn på fargene

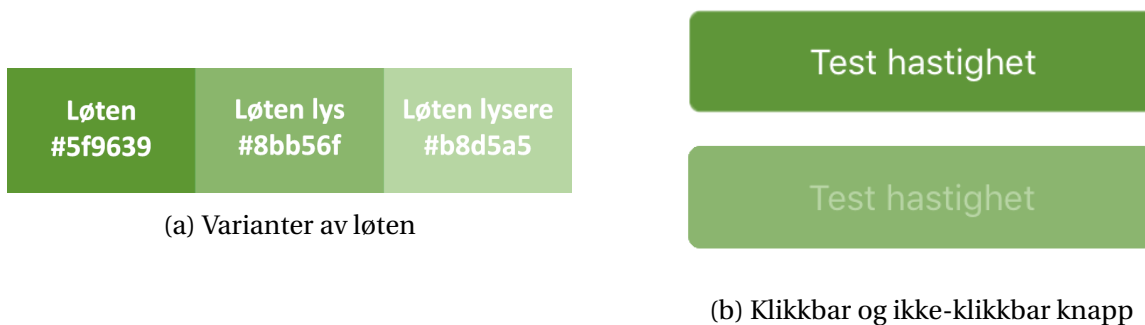
våre som profilmanualen.

Fargeprofilen vår består av tre hovedfarger: Løten, en lys svart farge og hvit. Løten brukes på knapper og andre grafikk-elementer, hvit brukes på tekst og den lyse svarte fargen brukes på blant annet navigasjonsbaren.



Figur 4.13: Forside med hovedfargene

Varianter av disse fargene brukes på enkelte elementer. Varianter av Løten brukes blant annet for å markere om en knapp er klikkbar eller ikke klikkbar.



(a) Varianter av løten

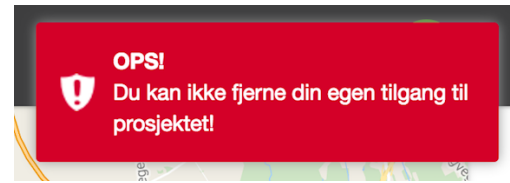
(b) Klikkbar og ikke-klikkbar knapp

Figur 4.14: Bruk av fargevariasjoner

Vi har også farger som vi bruker for å markere varsler. Begge fargene er hentet fra profilmanualen.



Figur 4.15: Varselfarger



Figur 4.16: Fargen "Vampyr" i bruk

4.5 Universell utforming

Et av kravene vi fikk fra kommunikasjonsenheten hos fylkeskommunen var at løsningen skulle være brukervennlig og ha universell utforming. Universell utforming vil si at løsninger blir utviklet slik at den kan brukes av alle og at spesielløsninger for enkeltgrupper ikke blir nødvendig [23]. Universell utforming er regulert av Forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKT)- løsninger fra 2013. Forskriften sier at alle nye IKT-løsninger som blir utviklet etter 1. Juli 2014 må være universelt utformet og skal minst være utformet i samsvar med standarden Web Content Accessibility Guidelines 2.0 [24]. Vi har brukt nettsiden til Direktoratet for forvaltning og IKT (Difi) som hjelp og veiledning. Difi har ansvaret for å følge opp reglene i forskriften.

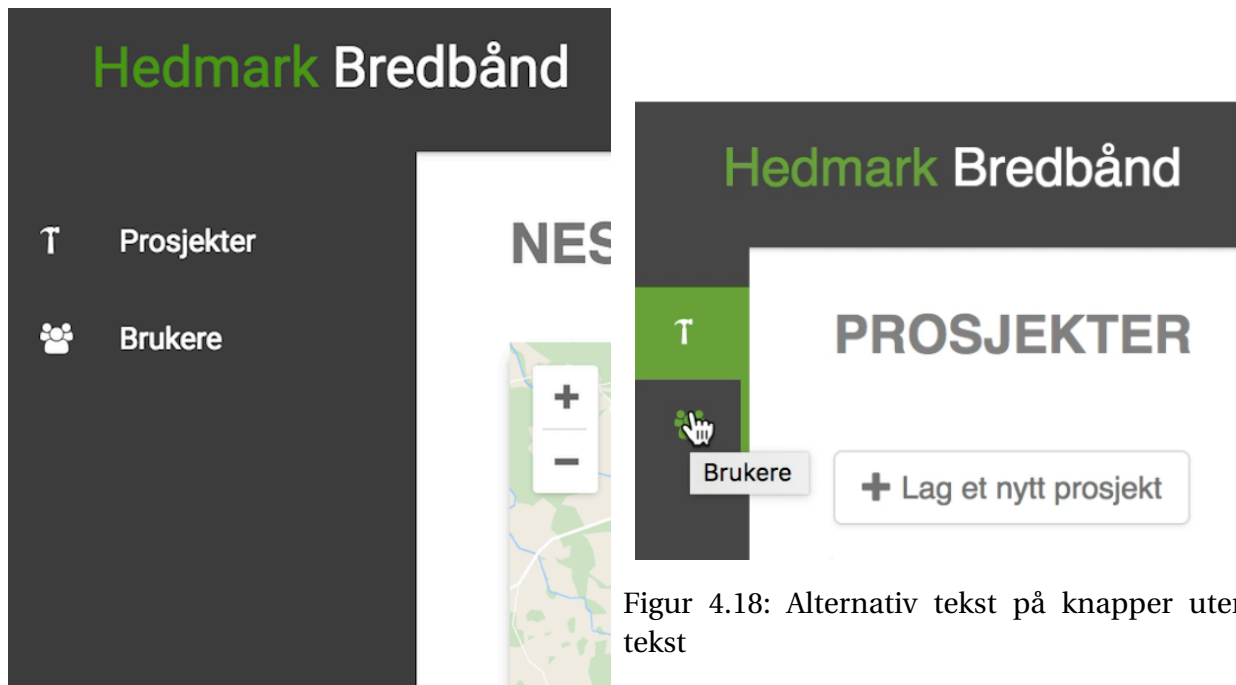
For at reglene i denne forskriften skal gjelde for en løsning må løsningen være rettet mot kunder eller innbyggere i Norge, være en del av virksomhetens hovedløsning og være vesentlig oppgradert eller ny etter 1. Juli 2014. Hvis ikke disse tre punktene gjelder for en løsning, så er det heller ikke lovpålagt at løsningen skal være universelt utformet [25]. I forskriften er hovedløsning definert slik: «IKT-løsninger som er en integrert del av den måten virksomheten informerer og tilbyr sine tjenester til allmennheten på og som er knyttet til virksomhetens alminnelige funksjon» [24]. Ut fra denne definisjonen tolker vi det som at løsningen vi lager ikke er en del av hovedløsningen til Hedmark fylkeskommune. Dette spesielt fordi løsningen vår ikke er knyttet til fylkeskommunen sin alminnelige funksjon. Selv om universell utforming ikke er lovpålagt for vår løsning, har vi allikevel valgt å ha noe fokus på universell utforming både på nettsiden. Vi vil videre kommentere noen de lovpålagte kravene og andre anbefalinger vi har valgt å inkludere i vårt prosjekt, med konkrete eksempler fra nettsiden.

4.5.1 Bilder

Alternativ bildetekst på bilder og lenkede bilder

Alle brukere må få korrekt innhold eller motiv på bilder eller illustrasjoner. Dette gjelder også bilder eller illustrasjoner som opptrer som lenker. Derfor må motivet, innholdet og eventuelt lenkens mål presenteres tydelig ved å bruke alternativ bildetekst. Alternative bildetekster blir lest opp av hjelpemidler for brukere som ikke kan se bildene. Derfor er det viktig at bildene får alt-attributtet i HTML-koden, slik at det er sikkert at den alternative teksten blir lest opp og ikke for eksempel filnavnet [26]. Alt-attributtet fungerer på bilder, hvis det er andre ting som trenger alternativ tekst kan det brukes title-attributtet. På vår nettside har vi ingen bilder eller lenkede bilder. Vi har en meny langs venstre side som viser ikoner med tekst som fører brukeren til andre sider, se figur 4.17. Når denne menyen minimeres på smalere skjermer, vil teksten forsvinne. Vi har derfor lagt til alternative tekster for å forklare hva ikonene betyr, se figur 4.18. Menyen er bygget opp av en liste med punkter. Den alternative teksten er lagt til ved hjelp av title-attributtet på hvert punkt.

```
<li class="..." title="Brukere">
```



Figur 4.17: Knapper med tekst

Figur 4.18: Alternativ tekst på knapper uten tekst

Alternativ bildetekst på dekorative bilder

Bilder som er tilstede i løsningen bare for å bygge opp det grafiske designet skal også ha alternativ tekst ved hjelp av alt-attributtet. I disse tilfellene skal alt-attributtet være tomt, for eksempel alt="" . Det aller beste er hvis de dekorative bildene er lagt inn ved hjelp av CSS-koden [26]. På nettsiden vår har vi brukt ett dekorativt bilde, og det er bakgrunnsbilde som vi blant annen bruker på forsiden. Dette bilde er lagt inn ved hjelp av CSS-koden, og derfor var det ikke behov for å legge til et tom alt-attributt. I figur 4.19 er det et bilde av forsiden, og et utsnitt av css-koden som legger inn bilde background.jpg ved hjelp av background-image attributtet.



Figur 4.19: Forsiden, med CSS-kode

4.5.2 Skjema

Ledetekster

Spesielt for brukere med nedsatt motorikk og for sterkt svaksynte brukere med skjermleser er det viktig at elementer i skjemaet har ledetekster knyttet til seg. Ledetekster kan også være nyttig for vanlige brukere ettersom det kan gjøre skjemaet mer oversiktlig og brukervennlig ettersom ledeteksten forteller hva brukeren skal taste inn. For å koble elementer i skjemaet korrekt sammen med ledetekster er det anbefalt å bruke HTML-tagen <label> [27]. Figur 4.20 viser en tekstboks med ledetekst. Og utdraget fra HTML-koden viser hvordan vi har koblet ledeteksten sammen med tekstboksen ved hjelp av <label>-taggen.

Skriv inn din nye melding

```
<label for="message" > Skriv  
  inn din nye melding </label>  
<textarea class="form-control"  
  id="message" name="message"  
  ...
```

Figur 4.20: Tekstboks med ledetekst. HTML-kode som viser implementeringen

Veiledningstekster

Hvis et skjema er veldig komplekst eller det krever informasjon som brukeren er lite kjent med, bør brukeren hjelpes med mer enn bare en ledetekst. Det bør da være en veiledningstekst tilgjengelig med en utdypende veiledning. Disse veiledningstekstene bør ha en utvid-/kollapsfunksjon slik at de brukerne som har behov kan åpne dem opp, men at de ikke ligger åpne hele tiden etter som dette også kan virke forvirrende. Veiledningstekstene bør ligge i nær tilknytning til skjema-elementet det handler om, dette gjør at brukeren kan lese veiledningen og legge inn informasjon parallelt [27]. Figur 4.21 viser hvordan vi har brukt veiledningstekster i et skjema. Hvis det er en veiledningstekst tilgjengelig, vises et spørsmålstegn-ikon ved siden av skjema-elementet. Hvis brukeren trykker på spørsmålstegnet kommer boksen med veiledningsteksten frem.

SØKNADS VURDERING

Score: 0/9.5

Dette er kun en veileder ment for å gi en vurdering på utfylt informasjon. Trykk på pilene for å fylle karakter-feltene må du selv fylle inn et tall mellom 0-10 som gjenspeiler hvor bra du mener prosje

	Verdi
▶ Antall husstander	
▶ Husstander under 4Mbit/s ?	
▶ Husstander mellom 4Mbit/s og 30Mbit/s ?	
▶ Kostnad per husstand ?	
▶ Andel lokal medfinansiering i prosjektet ?	
Hvilke krav vil stilles til utbygger i anbudsprosessen? ?	
Betydning for lokal samfunnsutvikling og verdiskaping ?	
Totalt	

Husstander mellom 4Mbit/s og 30Mbit/s

Gjelder områder med forbedring av eksisterende tilbud, sjekk mot begge dekningskart. Hvis søknaden kun gjelder områder uten grunnleggende tilbud, sett 0 og fyll ut "Husstander under 4Mbit/s".

Figur 4.21: Skjema med veiledningstekst

Feilhåndtering i skjema

Ved utfylling av skjema er det viktig å informere brukeren dersom det er feil eller mangler ved utfyllingen av skjemaet. Hvis det er noe feil i skjemaet skal det feltet som har feilen markeres visuelt. En feilmelding skal komme i tillegg til den visuelle markeringen, og feilmeldingen skal være i tekst og inneholde spesifikk informasjon om hvert enkelt felt. En feilmeldingen som inneholder teksten «feltet er obligatorisk» er ikke presist nok. En feilmelding burde heller inneholde teksten «Du må oppgi telefonnummer». Dette gjør det lettere for brukeren å rette opp eventuelle feil ettersom feilmeldingen er klar på hva som mangler [28]. Figur 4.22 viser hvordan vi varsler brukeren om feil eller mangler i et skjema. Det kommer en visuell markering i form av at tekstboksen får en rød ramme. Og det vises en konkret feilmelding til det spesifikke feltet om hva som er feil.

Skriv inn din nye melding

Melding må ha minimum 10 tegn

Figur 4.22: Skjema med feilmelding til bruker

Registrering og innlogging

Registrering og innlogging er noe mange brukere kan streve med å gjennomføre, og dette er uheldig ettersom det da hindrer brukeren i å bruke hele løsningen. Her gjelder de samme kravene som for andre skjemaer, at det skal være feilhåndtering med hjelpende tekster[29]. Figur 4.23 viser feilmeldingene som vises hvis brukeren prøver å logge inn uten å ha skrevet inn e-post og passord.

Login ×

E-post

E-post må fylles ut

Passord

Passord må fylles ut

Figur 4.23: Innloggings-skjema med feilmelding til bruker

Justert tastatur på mobileenheter

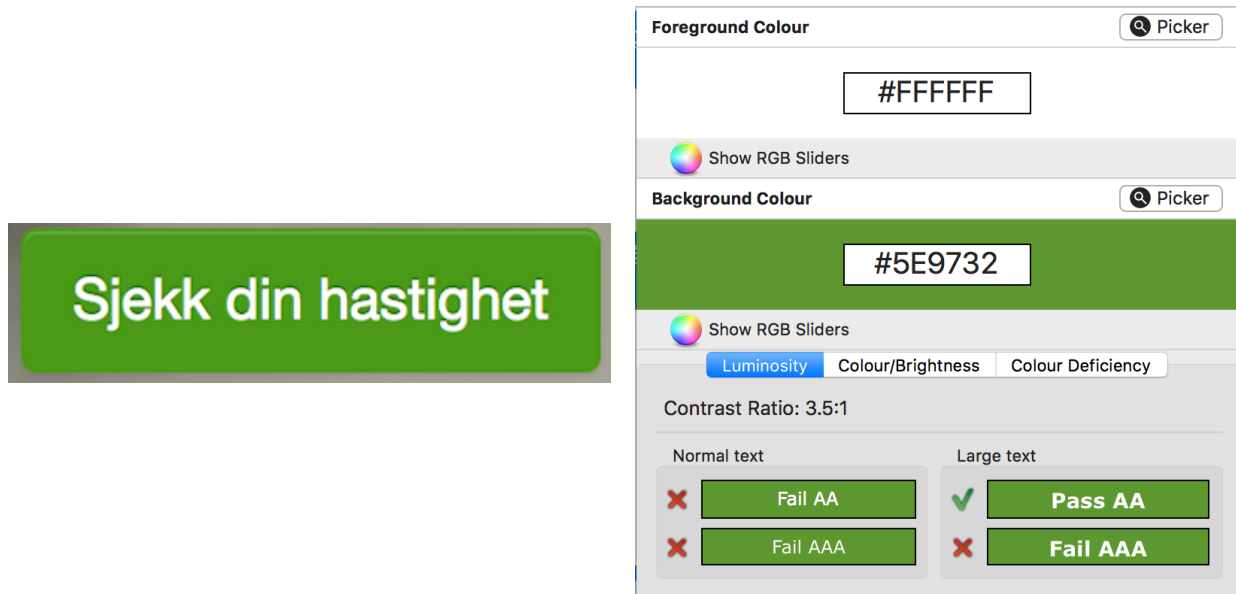
Ved å bruke input-typene som tilbys i HTML5 vil tastaturet justeres til å passe med hva som skal fylles inn. Dette vil det gjøre enklere å fylle ut skjemaet på mobile enheter [27]. Vi har brukt to av disse input-typene i skjemaer på nettsiden. Typen number gir numerisk tastatur. Typen email gir bokstavgastatur som inkluderer @.

```
<input type='number' ... >
```

```
<input type='email' ... >
```

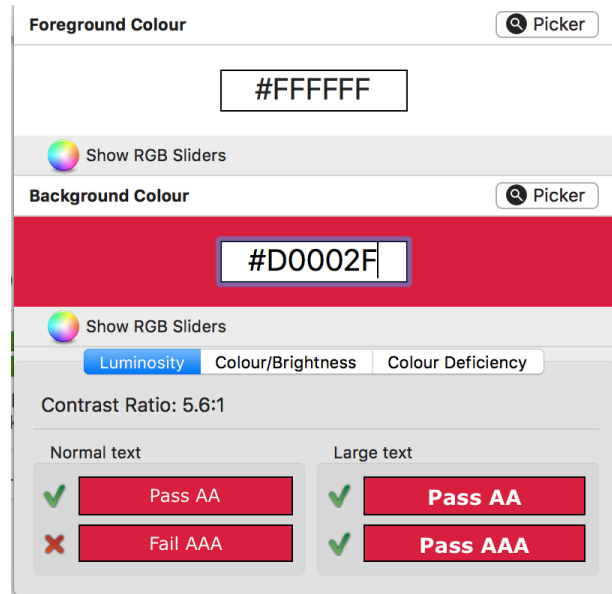
4.5.3 Kontraster

Tekst og grafikk må ha tilstrekkelig kontrast mot bakgrunnen for å sikre at alle brukere får med seg alt av innhold, særlig under krevende lysforhold. Brukere som spesielt trenger dette er svak-synte, dyslektikere og fargeblinde. For stor tekst er kontrastkravet 3.0:1. For liten tekst er kontrastkravet 4.5:1 [30]. Vi har ikke klart å innfri riktige kontrastkrav alle steder på vår nettside, men vi vil vise noen eksempler på steder vi har klart dette. Figur 4.24 viser at vi innfrir kontrastkravet til stor tekst når vi har grønn knapp med stor hvit tekst.



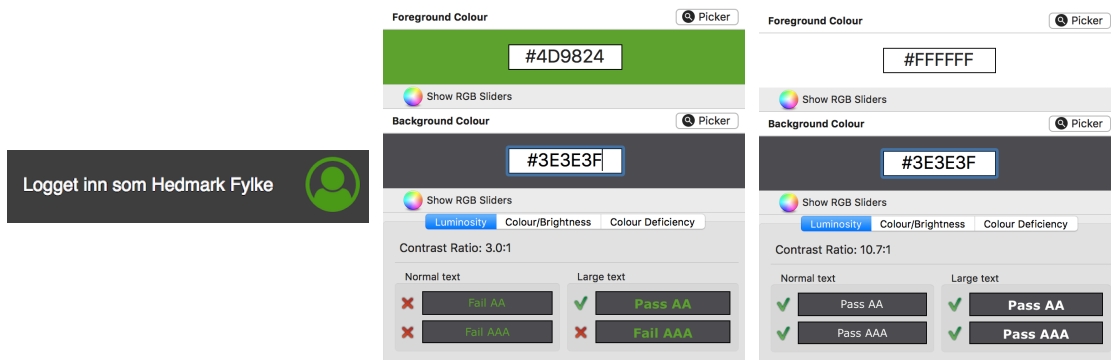
Figur 4.24: Måling av kontrastkravet på grønn knapp med stor tekst

Figur 4.25 viser at vi innfrir kontrastkravet til liten tekst når vi har rød knapp med liten hvit tekst.



Figur 4.25: Måling av kontrastkravet på rød knapp med liten tekst

Figur 4.26 viser at vi innfrir kontrastkravet til stor tekst når vi har grå bakgrunn med grønn stor ikon, og at vi innfrir kravet til liten tekst når vi har hvit liten tekst på grå bakgrunn.



Figur 4.26: Måling av kontrastkravet på meny med liten tekst og stort ikon

5 *Implementering*

I dette kapitlet tar vi for oss hvordan vi har gått frem for å implementere appen, nettsiden, api-et og databasen. Vi ser på hvordan de forskjellige delene er bygd opp og hvordan de kommuniserer med hverandre. Her ser vi både på brukergrensesnittet som vi har skapt og logikken bak. Etter vi har gått gjennom forskjellige aspekter av logikken går vi gjennom de ulike sidene på appen og nettsiden, og forklarer litt rundt de forskjellige sidene.

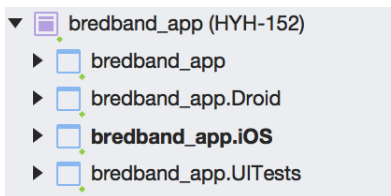
5.1 Frontend

5.1.1 App

I denne delen ser vi på hvordan vi kodet og implementerte appen. Her tar vi for oss blant annet filoppsettet, Xamarin, noen plugins og hvordan vi bruker dependency injection.

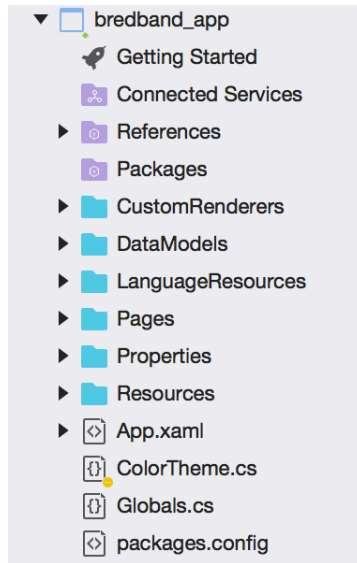
5.1.1.1 Filstruktur

Når et Xamarin Forms PCL prosjekt blir lagd blir det automatisk delt opp i fire deler: en felles mappe, en iOS mappe, en Android mappe og en mappe for tester.



Figur 5.1: Hovedstruktur i app

I fellesmappen "bredband_app" finner man koden som er felles. Her ligger koden for hver enkelt side i appen og logikken bak sidene. Mesteparten av app-koden ligger i denne mappen.



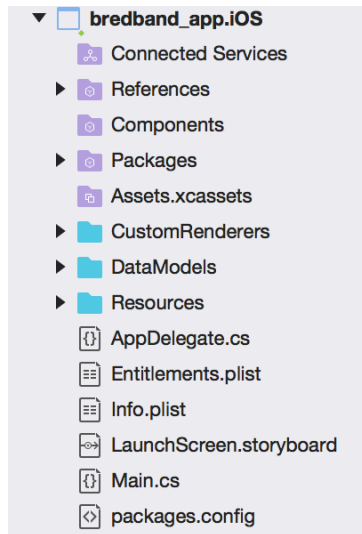
Figur 5.2: Strukturen i fellesmappen

- CustomRenderers — Mappe med kode for å overskrive platform-spesifikk brukergrensesnitt.
- DataModels — Mappe med datamodeller for objekter i systemet. Objektene arbeider uavhengig av hverandre, men enkelte modeller inneholder andre modeller.
- LanguageResources — Mappe med string-filer med oversettelser.
- Pages — Mappe som inneholder hver enkelt side i appen
- Properties — Autogenerert mappe med informasjon om hvordan appen skal bygges opp.
- Resources — Autogenerert mappe med ressurser, f.eks. filer, brukt av appen.

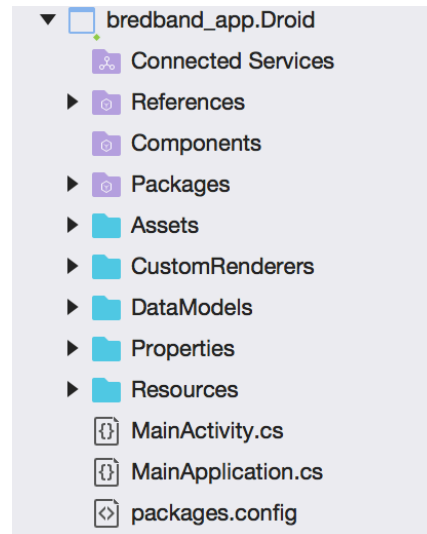
Vi har jobbet etter objektorientert programmering og har derfor endt opp med en struktur med en del datamodeller med private verdier og metoder som arbeider på disse veridene. Derfor har vi en del filer i mappen DataModels.

Det er to filer det er verdt å merke seg i den delte mappen. Den ene er filen "Globals.cs". Her har vi lagret en rekke globale verdier og noen metoder som brukes flere steder i systemet. Her finner man adressen til webserveren, konfigurasjon på hastighetsmålingen o.l. Den andre filen er "ColorTheme.cs". Denne filen brukes for å definere alle farger som brukes av systemet. Mer om dette kommer senere i rapporten.

iOS og Android mappene inneholder en del autogeneratede filer og mapper som man vanligvis ville ha hatt hvis man hadde lagd appene hver for seg nativt. Foruten dette ligner filstrukturen mye på strukturen i fellesmappen.



Figur 5.3: iOS filstruktur



Figur 5.4: Android filstruktur

5.1.1.2 Forms ContentPage XAML

Alle sidene i mappen "Pages" er filer av typen "Forms ContentPage XAML". Dette er egentlig to forskjellige filer som er linket sammen. Disse filene er bestått av en ContentPage skrevet i XAML og en CodeBehind klasse skrevet i C#. Begge filene arbeider opp mot samme side i appen og kan begge sette opp brukergrensesnittet.

For å holde orden i filene våre har vi valgt å sette opp brukergrensesnittet i XAML filen, og ta oss av logikken i C# filen.

5.1.1.3 Dependency Injection

Vi ble nødt til å bruke Dependency Injection i appen vår. Dette måtte vi gjøre fordi iOS og Android bruker forskjellige deler av .NET bibliotekene. Koden vår for Android og iOS er nesten identisk, men på grunn av begrensinger i den delte delen (Forms) måtte vi gjøre det på denne måten. Dependency Injection brukes kun i sammenheng med hastighetsmålingen. Kode-eksemplene i denne delen er hentet fra hastighetsmålingen.

Dependency Injection er et systemutviklingsmønster som går ut på klienten mottar et objekt det trenger istedenfor å måtte lage det selv/kalle på et objekt som lager ønsket objekt. Det ønskede objektet blir noen andres problem.

Måten dette gjøres i Xamarin Forms er at man lager seg et interface[31]. Dette interface-et ligger i fellesmappen og har alle metoder og variabler som skal kalles på fra utsiden.

```
public interface IConnectivity
{
    event EventHandler<ConnectivityEventArgs> UploadChanged;
    event EventHandler<ConnectivityEventArgs> UploadCompleted;

    event EventHandler<ConnectivityEventArgs> DownloadChanged;
    event EventHandler<ConnectivityEventArgs> DownloadCompleted;

    string GetIP();
    void Upload(Uri uri, string file);
    void Download(Uri uri);
    void CancelAsync();
}
```

Deretter lager man seg klasser som arver fra interface-et. Her implementerer man alt av logikk bak interface-et. Disse klassene ligger i sin tilhørende platform (iOS/Android) sin mappe.

```
[assembly: Dependency(typeof(Connectivity_Android))]
namespace bredband_app.Droid
{
    public class Connectivity_Android : IConnectivity
    {
        // Implementering av Android klassen her
    }
}
```

```
[assembly: Dependency(typeof(Connectivity_iOS))]
namespace bredband_app.iOS
{
    public class Connectivity_iOS : IConnectivity
    {
```

```
// Implementering av iOS klassen her
}
}
```

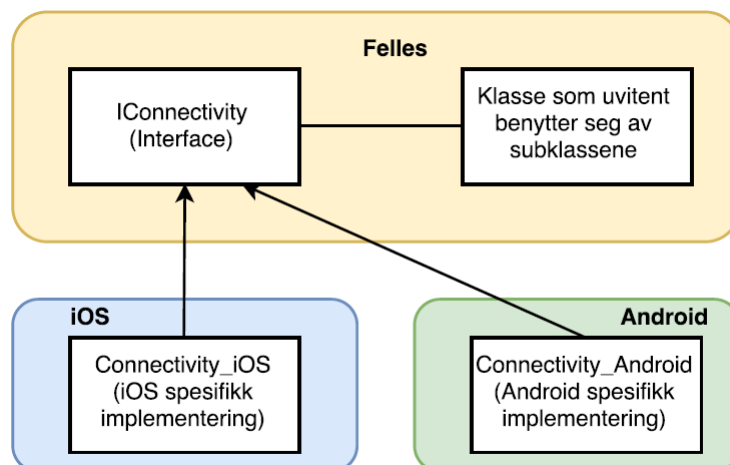
Når dette er gjort kan det nye objektet brukes ved å kalle på funksjonene og variablene i interface-et. Standarden i Xamarin Forms er at man kan kalle direkte på interface-et uten å sette opp ett nytt objekt. Da brukes det et globalt objekt som allerede er satt opp.

```
DependencyService.Get<IConnectivity>().GetIP();
```

Vi ønsket å lage et nytt objekt for hver gang en fullstendig hastighetsmåling ble gjennomført og har derfor lagd et interface objekt og brukt "NewInstance" for å få et nytt objekt. Dette gjøres i Speedtest-klassen i "Speedtest.cs".

```
private readonly IConnectivity Connection;
Connection = DependencyService.Get<IConnectivity>
    (DependencyFetchTarget.NewInstance);
```

Dette gjør at vi kan bruke IConnectivity objektet i Speedtest-klassen uten å tenke over hvordan dette håndteres på hver av platformene. På denne måten kan vi gjøre målingene på forskjellige måter på platformene, og Speedtest-klassen trenger kun å forholde seg til den informasjonen som den får fra IConnectivity-objektet.



Figur 5.5: Illustrasjon av Dependency Injection

5.1.1.4 Custom renderers

I appen har vi overskrevet noe av standard GUI-en i iOS og Android. Dette har vi gjort ved å bruke Custom Renderers. Dette kan likne litt på Dependency Injection, men her bruker vi klasser og ikke interfaces.

I Xamarin Forms finnes det ferdig definerte ”controllers”. Dette kan være alt fra progressjonsbarer til tekstbokser. Når man bruker en controller vil Xamarin oversette disse til riktig platform for oss slik at vi slipper å skrive kode for hver enkelt platform. Dersom man ønsker å overskrive disse controllerene må man først lage en klasse som arver fra controlleren. Denne plasseres i fellesmappen.

```
public class CustomProgressBar : ProgressBar
{
    public CustomProgressBar()
    {
    }
}
```

Deretter lager man en klasse i iOS og en klasse i Android som arver fra denne klassen.

```
[assembly: ExportRenderer(typeof(CustomProgressBar),
    typeof(CustomProgressBarRenderer))]
namespace bredband_app.iOS
{
    public class CustomProgressBarRenderer : ProgressBarRenderer
    {
        // Kode for skreddersydd progressjonsbar
    }
}
```

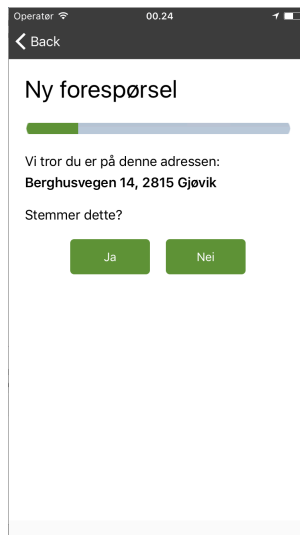
Deretter kan man bruke den direkte inn i en XAML fil eller i en C# fil. Xamarin passer på at riktig kode blir hentet for hver platform.

```
<control:CustomProgressBar x:Name="progressBar" Progress=".3" />
```

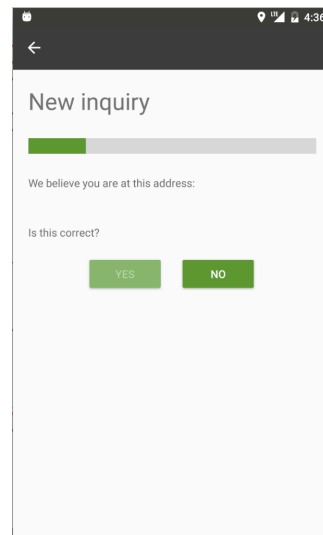
Vi har brukt Custom Renderers på en progressjonsbar og flere felter hvor brukeren skal skrive inn informasjon. Ved å implementere kodesnutten ovenfor har vi gått fra en progressjonsbar bestående av en tynn blå stripe, til en tykkere progressjonsbar med fargen ”Løten”.



Figur 5.6: Original progressbar på iOS, Android og Windows Phone [1]



Figur 5.7: iOS med tykk grønn bar



Figur 5.8: Android med tykk grønn bar

5.1.1.5 Internasjonalisering

Appen støtter to språk: norsk og engelsk. For å gjøre internasjonlisering i appen bruker vi RESX Resource Files. Dette er den innebygde metoden for å gjøre internasjonlisering i .NET applikasjoner. Vi har to forskjellige RESX Resource Files i vårt system: "LanguageResources.resx" som er standard filen med engelske oversettelser og "LanguageResources.no.resx" som er filen med norske oversettelser. Disse finnes i mappen "LanguageResources" som befinner seg i fellesmappen. Filene inneholder data med et unikt navn "name" og en string "value". Samme data plasse-

res i begge filene, men med forskjellige oversettelser av stringen.

```
<data name="FrontpageMapButton" xml:space="preserve">
  <value>Se dekningskart</value>
</data>
```

```
<data name="FrontpageMapButton" xml:space="preserve">
  <value>View coverage map</value>
</data>
```

Når en side først lastes inn med en Xaml fil inneholder den ingen stringer. Disse lastes inn i en konstruktør i CodeBehind-filen. Eks:

```
fronpageMapButton.Text =
    LanguageResources.LanguageResources.FrontpageMapButton;
```

Grunnen til at vi må gjenta samme ord to ganger er fordi Xamarin har lagd et eget namespace for språk-filene våre på grunn av mappestrukturen.

For at stringene skal vises må de forskjellige plattformene bestemme hvilket språk som skal vises. For å gjøre dette må vi gjennomføre Dependency Injection. Koden som brukes for dette er hentet direkte fra Xamarin Forms sin dokumentasjon på "Localizing Xamarin.Forms Apps with RESX Resource Files" [32]. Hele "ILocalize.cs" i fellesmappen, "Localize.cs" i Android mappen og "Localize.cs" i iOS mappen er direkte kopiert. Appen bruker denne koden ved oppstart i "App.xaml.cs".

```
if (Device.OS == TargetPlatform.iOS || Device.OS == TargetPlatform.Android)
{
    var ci = DependencyService.Get<ILocalize>().GetCurrentCultureInfo();
    LanguageResources.LanguageResources.Culture = ci;
    DependencyService.Get<ILocalize>().SetLocale(ci);
}
```

Hvis mobilen bruker norsk som systemspråk blir norsk valgt i appen, brukes noe annet blir engelsk valgt.

Vi har også tatt med koden fra dokumentasjonen som gjør det mulig å oversette direkte i XAML filene. Vi bestemte oss for å ikke bruke dette fordi det forhindret oss fra å endre språk samtidig som man var inne i appen. Man ble da låst til å bruke systemspråket.

5.1.1.6 API

Flere steder i appen gjøres det API-forespørsler. Dette gjøres til vår egen webserver for å få ut informasjon fra databasen og mot eksterne webserveren for å hente ut annen informasjon. Adressene til API-ene finnes i Globals.cs slik at vi enkelt kan endre adressene i hele appen. Når vi gjør API-forespørsler bruker vi HttpClient-klassen i .NET.

Egen webserver

Mot vår egen webserver gjøres det tre forespørsler. To av forespørslene er POST-forespørsler, den siste er en GET-forespørsel. Dette gjøres når brukeren gjennomfører en hastighetsmåling, når brukeren sender en ny forespørsel og når en bruker taster inn et postnummer.

For å gjøre POST-forespørslene lager vi først et nytt objekt av .NET Object-klassen. Dette objektet fyller vi med de nøklene og verdiene som vi ønsker å sende til databasen. Objektet blir så konvertert til JSON. Dette blir så lagt til i et nytt StringContent-objekt sammen med HTTP-header informasjon.

```
StringContent content = new StringContent(jsonObj.ToString(), Encoding.UTF8,
    "application/json");
```

Dette blir så sendt avgårde i et POST-request til webserveren vår.

Å gjøre et GET-request er simplere. Da gjør man et GET-request til webserveren hvor man sender med parameteren i url-en.

```
var response = await client.GetAsync(Globals.ServerUrl + "api/post_codes/" +
    postalCode.Text);
```

Så er det bare å hente ut svaret fra variabelen "response".

Eksterne API-forespørsler

I appen gjøres det to forespørsler mot et annet API. Begge disse forespørslene gjøres på appen sin side for å forespørre bedre nett "InquireBetterBroadbandPage".

Den første forespørselen som gjøres er å konvertere GPS-koordinater til en reel adresse. For å gjøre dette bruker vi Google sin Geocode API[33]. Da gjøres det et GET-request på samme

måte som ovenfor mot webserveren til Google. Her sender vi med breddegrad og lengdegrad i adressen.

```
var api = "https://maps.googleapis.com/maps/api/geocode/json?latlng=" +  
    latitude + "," + longitude;
```

Den andre forespørselen som gjøres er for å hente ut IP-adressen til brukeren. I dette requestet trenger man ikke å sende med noen parametere, bare hente ut resultatet. Derfor kan vi bruke en metode kalt "downloadString", som kun forventer å hente ut en enkelt string fra en adresse.

```
var response = client.DownloadString(new Uri(Globals.IPHost));
```

5.1.1.7 Biblioteker

Appen bruker følgende biblioteker foruten standard bibliotekene til Xamarin Forms:

- Connectivity Plugin av James Montemagno
- Permissions Plugin av James Montemagno
- Geolocator Plugin av James Montemagno
- Json.NET av James Newton-King

Alle bibliotekene er PCL biblioteker som kan brukes på tvers av platformene.

Connectivity Plugin

Connectivity Plugin brukes for å sjekke om brukeren er koblet til internett, hvis brukeren ikke er det kommer de kun til forsiden. De kan ikke klikke seg videre for å ta målinger eller se dekningskartet.

Permissions Plugin

Permissions Plugin brukes på informasjonssiden hvor brukeren må godta at appen kan bruke GPS. Appen gjør det mulig for oss å sjekke innstillingene til brukeren og be om at innstillingene endres.

Geolocator Plugin

Geolocator Plugin brukes for å hente ut GPS koordinatene til brukeren fra mobilen. Dette gjør vi for å foreslå en adresse til brukeren på hvor han/hun befinner seg. Disse koordinatene brukes sammen med Google Geocode API-et for å finne adressen.

Json.NET

Json.NET brukes for å konvertere objekter til JSON og konvertere JSON om til objekter i systemet. Dette gjøres i sammenheng med API-forespørsler der vi benytter oss av JSON formatet.

5.1.1.8 Farge tema

I appen bruker vi fargene fra fargeprofilen som vi skrev om i [kapittel 4](#). For å gjøre dette har vi lagd et sett av globale styles for hele appen. Dette har vi gjort gjennom XAML i appen sin ResourceDictionary. Dette gjør vi i filen "App.xaml" under "<Application.Resources>". Et eksempel på et generisk sett av styles er som følger:

```
<Application xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="Styles.App">
  <Application.Resources>
    <ResourceDictionary>
      <Style x:Key="styleName1">
        // Style informasjon
      </Style>
      <Style x:Key="styleName2">
        // Style informasjon
      </Style>
      <Style x:Key="styleName3">
        // Style informasjon
      </Style>
    </ResourceDictionary>
  </Application.Resources>
</Application>
```

Her kan man føre opp forskjellige styles som man skal bruke på forskjellige elementer. Man kan skrive fargekodene rett inn i denne i filen, men vi valgte å definere fargene i en egen fil "ColorTheme.cs" for å gjøre det lettere å finne fargekodene og endre dem. Fargene ligger globalt i systemet og kan bli kalt på fra hvor som helst.

I XAML filen definerer vi først globale farger. Her definerer vi bare at det er en farge med et unikt navn. Vi definerer ikke hvilken farge det er.

```
<Color x:Key="textColor"></Color>
```

Deretter lager vi en style som bruker denne fargen.

```
<Style x:Key="styleName1">
    <Setter Property="TextColor" Value="{DynamicResource textColor}"
</Style>
```

I filen "ColorTheme.cs" har vi en funksjon som heter "createTheme()". Denne funksjonen blir kalt på når appen starter. Denne setter de globale fargene definert i XAML-filen lik fargene som er definert i "ColorTheme.cs".

```
// Fargekodene defineres
public static string color1 = '#ffffff';

// Flere elementer skal bruke samme fargekode og peker til samme fargen.
public static string textColor = color1;
public static string tilfeldigElement = color1;

// Fargene i XAML fylles inn med fargene fra temaet.
public static void createTheme() {
    // ...
    Application.Current.Resources['textColor'] = textColor;
    // ...
}
```

Deretter kan den nye stilen brukes i hele appen. Denne kodesnutten viser hvordan man kan bruke eksmeplene ovenfor for å sette tekst-fargen på en knapp:

```
<Button Style={DynamicResource styleName1}></Button>
```

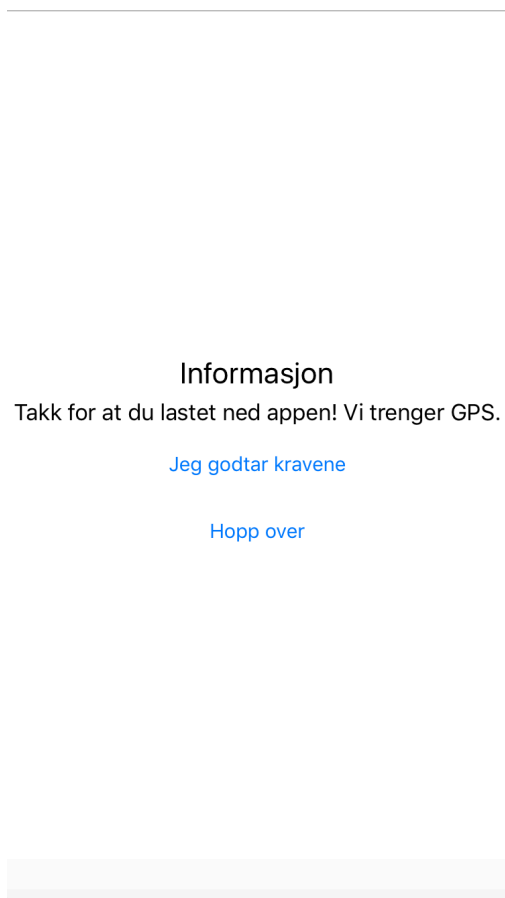
Ved å bruke DynamicResource istedenfor StaticResource kan vi endre farge temaet under run-time dersom det er ønskelig.

5.1.1.9 Sider i appen

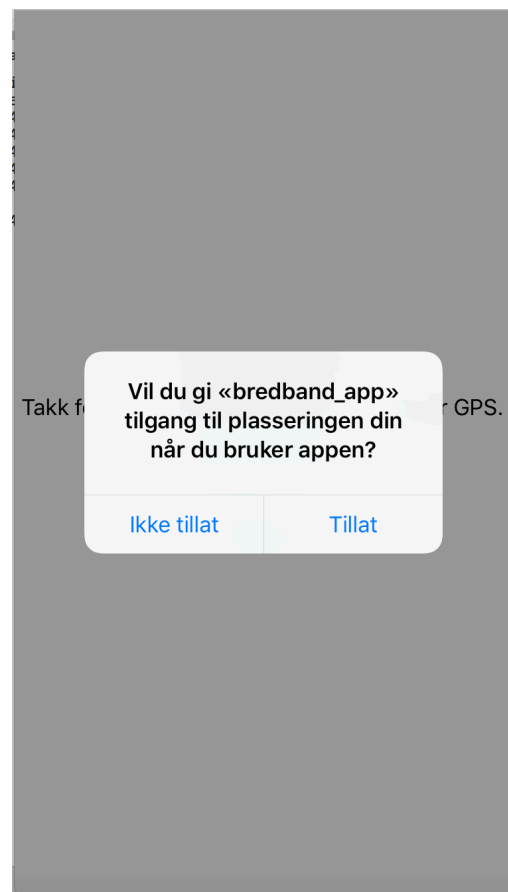
Appen består av seks forskjellige sider.

InformationPage

Den første siden man møter på er en informasjonsside "InformationPage.xaml". Denne siden skal kun vises første gang appen brukes. Informasjonssiden skal fortelle hvilken informasjon appen samler inn, og hvilke rettigheter på mobilen appen trenger. Man kan enten velge å godkjenne kravene eller hoppe over steget. Oppdragsgiver har bestemt seg for å finne ut hva som skal stå på denne siden senere, så ved nåværende tidspunkt inneholder siden kun fylltekst.



Figur 5.9: App: Informasjonsside



Figur 5.10: App: Godkjennelse av rettigheter

For å sjekke om det er første gang appen brukes benytter vi oss av objektet "Application" sin variabel kalt "Properties". Dette er en samling av nøkler og verdier som brukes når appen startes, lukkes midlertidig eller avsluttes helt.

Når appen startes opp sjekker vi i "App.xaml.cs" om en bestemt verdi eksisterer. Dersom den ikke gjør det laster vi inn "InformationPage.xaml", dersom den finnes laster vi inn "Frontpage.xaml".

```
// Decides if it is first time use or not. Loads either information page or
// front page.
if (!Current.Properties.ContainsKey("firstUse"))
{
    MainPage = new NavigationPage(new InformationPage())
    {
        BarBackgroundColor = Color.FromHex(ColorTheme.navigationBarColor),
        BarTextColor = Color.FromHex(ColorTheme.navigationBarTextColor)
    };
}
else
{
    // Loads frontpage as a navigationpage.
    MainPage = new NavigationPage(new Frontpage())
    {
        BarBackgroundColor = Color.FromHex(ColorTheme.navigationBarColor),
        BarTextColor = Color.FromHex(ColorTheme.navigationBarTextColor)
    };
}
```

Dersom vi ikke fant nøkkelen i Properties må denne legges til. Dette skjer i InformationPage når en bruker trykker å for godkjenne/avslå kravene. Dersom man velger å hoppe over informasjonssiden blir nøkkelen ikke lagret. Dette fører til at informasjonssiden vil vises på nytt neste gang appen starter.

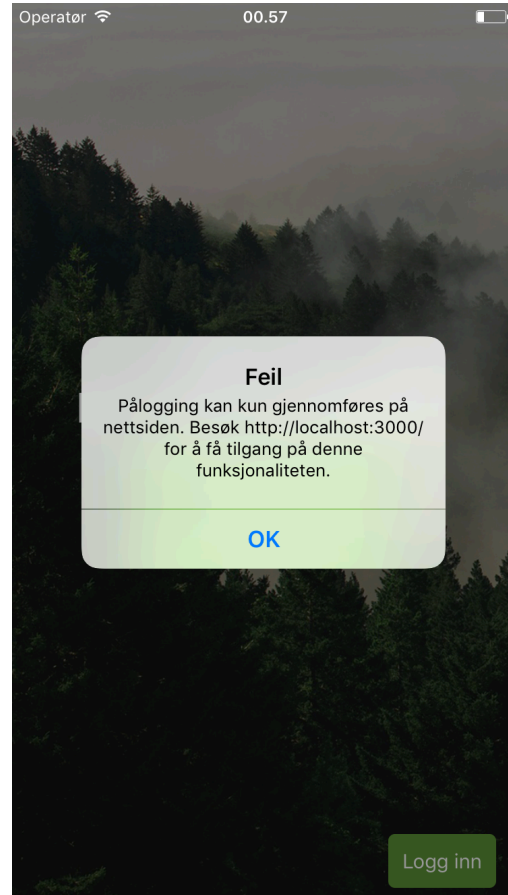
Frontpage

Den neste siden man møter er forsiden "Frontpage.xaml". På forsiden har man tre muligheter: sjekke hastighet, se dekningskart eller logge inn. Når appen ble lagd ble det besluttet at innlog-

ging kun skulle gjøres på nettsiden i første omgang. Knappen for å logge inn er her fremdeles. Trykker man på den kommer det opp en alert som informerer om at innlogging gjøres på nettsiden.



Figur 5.11: App: Forside



Figur 5.12: App: Alert når man prøver å logge inn

I feilmeldingen står det at man må besøke "http://localhost::3000" for å logge på. Dette er en global string som peker mot webserveren vår. Bildene er tatt i utviklingsfasen og er grunnen for at det ikke står en reell adresse her.

MapPage

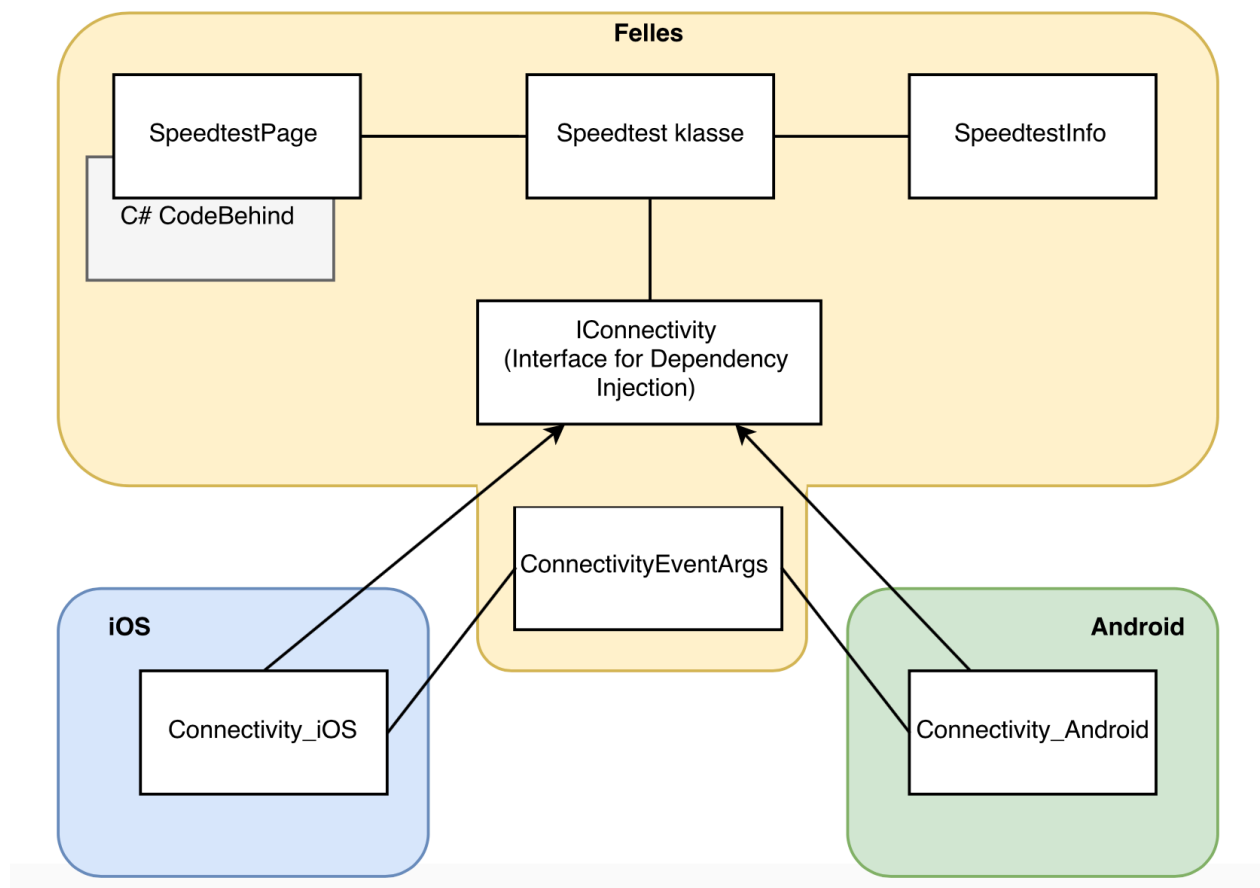
Når man trykker på "Se dekningskart" kommer man til den tredje siden "MapPage.xaml". Denne siden viser frem et WebView. I CodeBehind klassen lastes det inn en nettsideadresse i WebView-et. Denne adressen skal egentlig peke til varmekartet på nettsiden, men vi fikk ikke nok tid til å lage varmekartet.

SpeedtestPage

Den fjerde siden er hastighetsmålingen "SpeedtestPage". Dette er den mest avanserte siden på appen. Her kan brukeren trykke "Test hastighet" for å starte en hastighetsmåling. Hvordan denne målingen gjennomføres er beskrevet i [kapittel 2: Teoretisk grunnlag](#).

De tre forskjellige testene som gjennomføres har alle hver sin forhåndsbestemt tid for hvor lenge testene skal holde på. Dette kan man konfigurere i filen "Globals.cs".

Vi kunne ha implementert inn all koden for siden direkte i tilhørende C# fil, men har valgt å lage en egen klasse for et objekt som gjennomfører hastighetsmålinger. Denne klassen må bruke Dependency Injection for å fungere på både iOS og Android. Det som er felles i logikken ligger derfor i klassen "Speedtest" i "Speedtest.cs", som igjen kaller på et interface "IConnectivity" for å gjøre Dependency Injection.



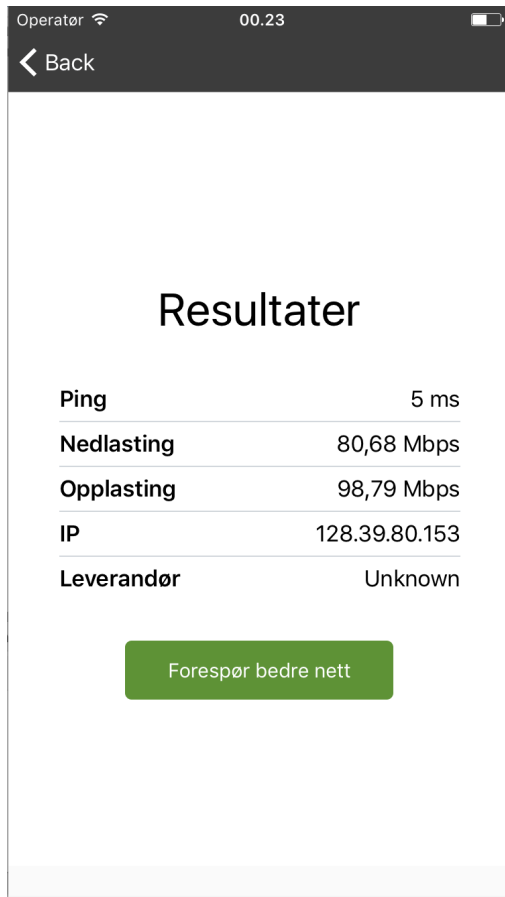
Figur 5.13: Sammenheng mellom speedtest filer i app

I Speedtest klassen og i Connectivity klassene brukes det mange EventHandlers. I Connecti-

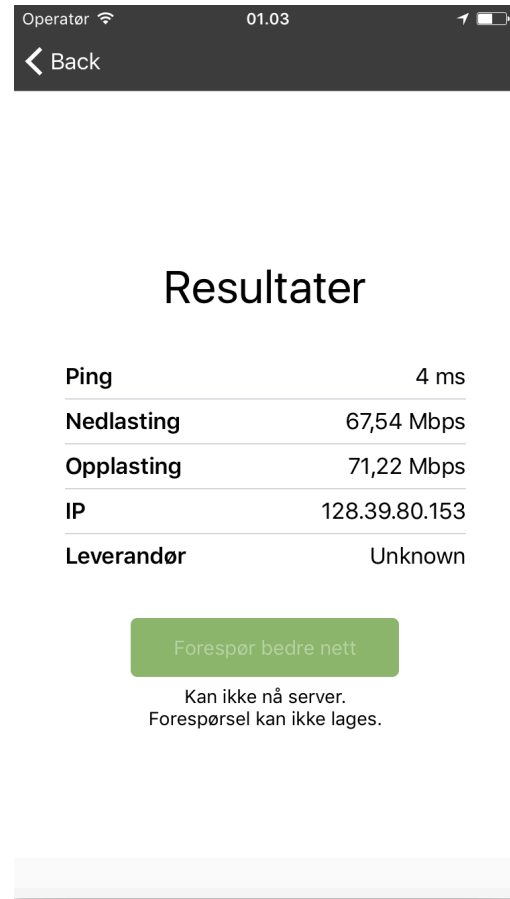
vity_iOS og Connectivity_Android gjennomføres enkelt målinger av nedlastning, opplastning og ping. Connectivity-klassene sender ut Event-er for hver gjennomførte måling. Da sender de med et objekt av typen "ConnectivityEventArgs". Dette objektet inneholder informasjon om målingen. Speedtest klassen tar så hånd om informasjonen og sender den videre til et SpeedtestInfo objekt. Dette objektet ivaretar målingene for hele testen. Deretter ber Speedtest klassen Connectivity om å kjøre flere målinger. Hver gang speedtest-objektet mottar informasjon oppdateres informasjonen på SpeedtestPage og informasjonen i SpeedtestInfo-objektet. Til slutt når testen er gjennomført blir hastighetsmålingen sendt til databasen vår og man blir automatisk sendt videre til neste side i appen. SpeedtestInfo-objektet som ble lagd under testen blir også sendt med videre til den neste side.

SpeedResultsPage

På denne siden vises frem informasjonen som ble samlet inn under hastighetsmålingen. Informasjonen ligger i et SpeedtestInfo-objekt som blir sendt til SpeedtestResultPage-klassen sin konstruktør. Dersom man fikk postet en hastighetsmåling kan man trykke seg videre for å forespørre bedre nett. Dersom man ikke fikk postet målingen blir knappen deaktivert.



Figur 5.14: App: Resultat-siden

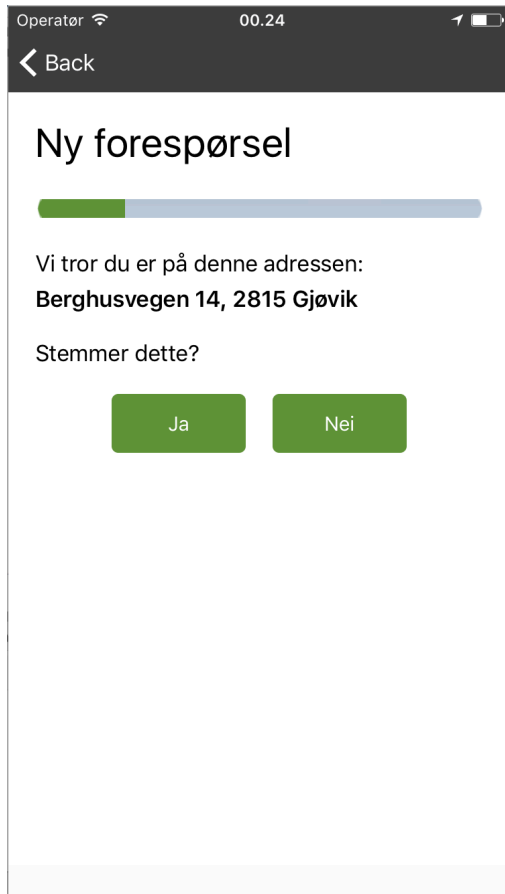


Figur 5.15: App: Resultat-siden når man ikke får koblet seg til databasen

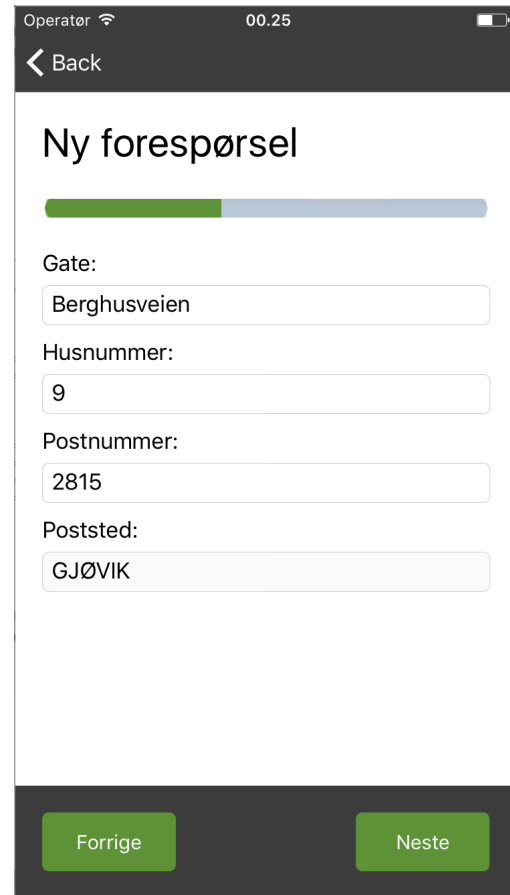
InquireBetterBroadbandPage

Den siste siden er forespørsel-siden. Denne siden består av flere sider som vises og skjules med en navigasjonsmeny på bunnen av siden.

Den første siden som kommer opp er brukerens lokasjon. Her bruker vi GPS for å hente ut adressen. Dette gjøres asynkront i bakgrunnen. Dersom vi finner en adresse kan brukeren godkjenne dette som sin adresse eller velge å skrive inn manuelt. Dersom vi ikke finner en adresse blir ikke godkjenn knappen aktivert, og brukeren må gjøre det manuelt. Når brukeren taster inn et postnummer gjøres det en GET-forespørsel til serveren vår for å hente ut riktig poststed.



Figur 5.16: App: Forespørsel side 1



Figur 5.17: App: Forespørsel med navigasjonsbar

De neste sidene i forespørselen henter informasjon fra brukeren om hvordan type hus det gjelder, hvem som skal kontaktes om eventuell utbygging og om det nåværende tilbudet. Til slutt sendes forespørselen til databasen vår. Brukeren får på dette tidspunktet tilbakemelding om at den ble sendt/ikke sendt.

Underveis når brukeren fyller ut informasjon gjøres det en rekke sjekker på teksten som skrives inn. Dersom brukeren prøver å gå videre uten å fylle inn informasjon vil det vises alerts. I Xamarin Forms finnes det ikke kontrollere som gjør dette, så valideringen måtte vi også implementere selv.

Forespørselen er bare en side. For å få den til å se ut som flere sider har vi først lagd hele brukergrensesnittet for alle stegene i samme Xaml fil. Her har vi skjult alle untatt det steget man skal se. Deretter bruker vi en egen funksjon for å bytte steg når brukeren trykker på "Neste" eller "Forrige". Når denne funksjonen kalles på sender man bare med det steget man ønsker å se.

```
public void SwitchInquireStep(int step)
{
    // Hides current step
    if (currentStep > 0)
    {
        forms[currentStep].IsVisible = false;
    }
    // Updates current step to new step
    currentStep = step;
    // Shows new step
    forms[currentStep].IsVisible = true;
    if (step == 1)
    {
        bottomNav.IsVisible = false;
    }
    else
    {
        bottomNav.IsVisible = true;
    }
    if (step == nrOfSteps)
    {
        nextButton.Text = LanguageResources.LanguageResources.InquirySendInquiry;
    }
    else
    {
        nextButton.Text = LanguageResources.LanguageResources.InquiryNextStep;
    }
    progressBar.Progress = (float)currentStep / nrOfSteps;
}
```

I denne kodesnutten blir først steget man er på skjult. Deretter oppdateres variabelen som sier hvilket steg man er på. Så gjøres riktig grensesnitt synlig. Til slutt er det en rekke sjekker som håndterer om navigasjonsbaren på bunnen av siden skal vises eller ikke, og som endrer teksten på "Neste" knappen dersom man er på siste sted. Progresjonsbaren blir også oppdatert etter hvilket steg man er på.

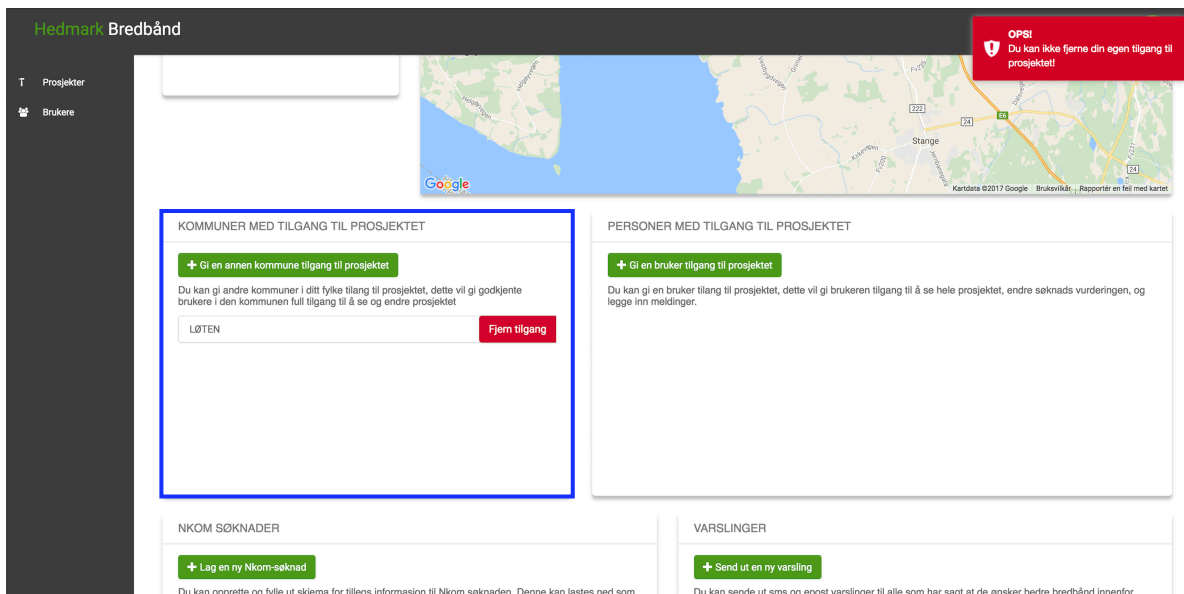
5.1.2 Nettside

5.1.2.1 Angular 2

Vi har valgt å bruke Angular 2 som frontend-rammeverk. Angular 2 er et Javascript rammeverk som brukes for å bygge opp web applikasjoner. Angular 2 bruker Typescript som primærspåk, men dette kompiles til Javascript når koden lagres. Typescript er en utvidelse av Javascript som blant annet tilbyr såkalt «static typing», noe som bidrar til mindre potensielle feil i koden.

Komponenter

En Angular applikasjon er laget av mange komponenter som til sammen utgjør en web applikasjon. En komponent har kontroll på en del av siden, også kalt et «view». I [figur 5.18](#) er en komponent markert med en blå ramme.



Figur 5.18: En komponent på nettsiden er markert

En komponent består av en komponent-klasse som håndterer logikken og HTML- og CSS-kode som håndterer visningen. Alle komponent klassene inneholder en `@Component`. Dette er en funksjon som legger til metadata til komponenten. [figur 5.19](#) viser hvordan metadata blir lagt til i koden, og hvordan filstrukturen til en komponent er.



Figur 5.19: Filstruktur og metadata til en komponent

Angular CLI

Angular CLI er et verktøy som gjør utviklingen av en Angular applikasjon raskere og lettere å vedlikeholde. Det er et kommandolinje verktøy som gjør at du som utvikler ikke trenger å gjøre alle oppgaver fra bunnen av. Vi valgte å bruke dette verktøyet for å spare så mye tid som mulig på utviklingen, slik at vi fikk tid til å lage mer funksjonalitet i systemet. Ved for eksempel å skrive inn

```
ng generate component [navn på komponenten]
```

i kommandolinja, genererer Angular CLI en ny komponent for deg. Alle nødvendige filer blir generert med nødvendig kode slik at alt er klart til å starte å utvikle direkte. Vi som utviklere trenger derfor ikke tenke på å deklare komponenten riktig eller å legge til all nødvendig metadata.

5.1.2.2 API

Mange av komponentene våre gjør API-forespørsler. Dette gjøres enten til vår egen webserver for å hente ut, legge inn eller oppdatere informasjon i databasen. Eller mot eksterne webservere for å hente ut annen informasjon. En API-forespørsel kan gjøres direkte i koden inne i komponenten, men det er vanlig å legge denne funksjonaliteten ut i en service. Dette gjør at flere komponenter kan bruke den samme servicen hvis den samme informasjonen skal hentes ut flere plasser på nettsiden. Man slipper da duplisering av kode ettersom koden som gjør spørringen bare finnes i servicen som alle komponenten har tilgang til.

Egen webserver

Når vi skal gjøre en forespørsel mot vår egen webserver, for eksempel hente ut et sub-prosjekt basert på en id, starter vi i komponent-klassen. Vi spør etter funksjonen `getSubProject` som ligger i `projectService`, og vi sender med id til sub-prosjektet vi ønsker.

```
getSubProject() {
    this.projectService.getSubProject(this.subProjectId)
        .subscribe(subProject => { this.subProject = subProject; });
}
```

I `projectService` gjøres spørringen videre til `api` som er en annen service. Denne service-en brukes alltid når vi gjør spørringer til vår egen webserver. I `api` ligger koden som legger til header-informasjon på forespørselen. Hvis det er en post eller put forespørsel så gjøres data som sendes om til en json-string. Denne koden er lagt i `api` servicen slik at vi slipper å skrive denne koden i alle forespørlene i andre service-filer. Det sendes også med hvilken route som skal brukes, og iden til sub-prosjektet.

```
getSubProject(subProjectId: number) {
    return this.api.get('subproject/' + subProjectId)
        .map(response => response as SubProject);
}
```

I `api` legges altså header-informasjonen til forespørselen ved hjelp av `getOptions` funksjonen

```
get(url: string) {
    return this.http.get(environment.apiUrl + url, this.getOptions())
        .map(res => res.json());
}
```

figur 5.20 viser en illustrasjon av hvordan en API-forespørsel til egen webserver foregår.



Figur 5.20: API-forespørsel til egen webserver

Eksterne API-forespørsler

Når vi gjør en forespørsel til en ekstern webserver fungerer det på nesten samme måte som når vi gjør en forespørsel til vår egen server, bortsett fra at vi ikke trenger å gå innom api servicen. Et eksempel på å bruke en ekstern webserver er når vi under hastighetsmålingen finner IP-adressen og internettleverandøren til brukeren. I komponent-klassen gjør vi en spørring mot speedTestService:

```
this.speedTestService.getIpAndIsp();
```

Funksjonen i speedTestService gjør så en spørring må den eksterne webserveren:

```
getIpAndIsp() {
    return this.http.get('http://ip-api.com/json').map(res => res.json());
}
```



Figur 5.21: API-forespørsel til ekstern webserver

5.1.2.3 Gjennomgang av nettsiden

Registrer

På registreringssiden har man mulighet for å velge brukertype. Dersom man velger noe annet enn vanlig bruker må man registrere seg gjennom ID-Porten. Dersom man velger vanlig bruker registrerer man seg med e-post og passord.

De brukerne som registrerer seg med ID-Porten vil komme tilbake til registreringssiden der de vil måtte fylle ut mer informasjon om seg selv.

The screenshot shows a registration form with the following fields:

- Fornavn (First Name): text input
- Efternavn (Last Name): text input
- E-post (Email): text input
- Passord (Password): text input with a strength indicator
- Gjenta Passord (Repeat Password): text input with a strength indicator
- Registrer (Register): green button

Figur 5.22: Registrerings skjema

Godkjenne Brukere

Navn	E-post	Telefon	Opprettet	Bruker Type	Kommune	Godkjenning	Avslå
Litzy Nienow	litzy.nienow@yahoo.com	1-339-104-0014	2017-05-15T19:47:35.396+02:00	Municipality	Nord-aurdal	Godkjenning	Avslå
Justyn Kris	justyn.kris@gmail.com	166.447.6481	2017-05-15T19:47:35.467+02:00	Municipality	Kongsvinger	Godkjenning	Avslå
Erica Runolfsdottir	erica.runolfsdottir@yahoo.com	174-997-2129 x8910	2017-05-15T19:47:35.540+02:00	Municipality	Åseral	Godkjenning	Avslå
Vida Hackett	hackett.vida@hotmail.com	(729) 090-8708	2017-05-15T19:47:35.616+02:00	Municipality	Trysil	Godkjenning	Avslå

Figur 5.23: Side for å godkjenne brukere

På denne siden vil ansatte i fylkeskommunen kunne godkjenne brukere i hele fylket. Kommuneansatte vil på denne siden kunne godkjenne andre som registrerer seg i samme kommune.

Prosjekter

På prosjektsiden vil brukere se de prosjektene de har tilgang på. Fylkesbrukere vil se alle prosjekter for alle kommuner i fylket. Kommunebrukere vil kunne se prosjekter i sin egen kommune. I tillegg vil alle brukere ha muligheten til å kunne se prosjekter dersom noen(kommune/fylke) har lagt dem til i prosjektet.

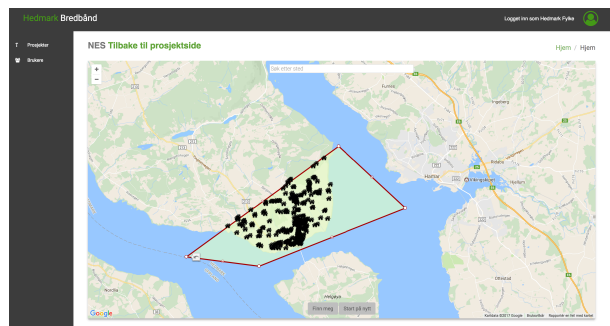
PROSJEKTER
Hjem / Prosjekter

Navn	Status	
Haudh-en-Nirmaeth	Klar til søknad	
Vaiya	Uttyst til leverandører	
Timeless Halls	Planlegging	

Figur 5.24: Prosjekt liste

Nytt Prosjekt

Når man oppretter et nytt prosjekt vil man måtte opprette et prosjekt-område. Her kan man bruke Google Maps sin Drawing Manger til å lage en polygon. Når man gjør endringer vil området hele tiden oppdateres med ikoner for hvor det finnes adresser. Figuren lagres hver gang det skjer endringer.

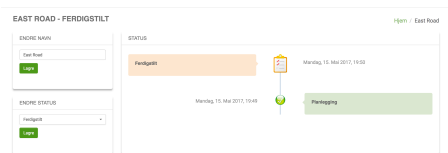


Figur 5.25: Nytt prosjekt side

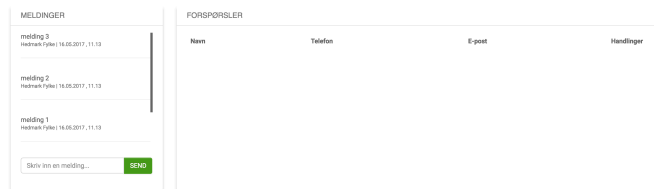
Prosjekt

Prosjektsiden inneholder mye av funksjonaliteten på nettsiden. Her kan brukeren legge til og fjerne andre brukere og kommuner.

Brukere inne på prosjektsiden vil kunne etterlate meldinger til de andre brukerne, endre navn og status på prosjektet.

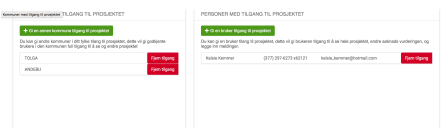


Figur 5.26: Prosjektside Stater

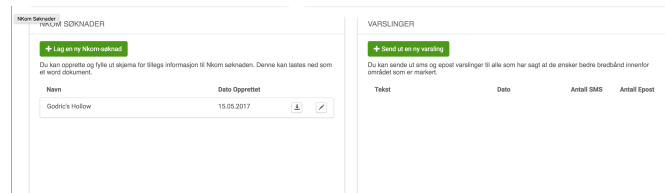


Figur 5.27: Prosjektside Meldinger og Forespørsler

De vil kunne lage nye prosjekt områder og Nkom søknader, og sende ut varslinger til de som har fårseput bedre bredbånd i området.

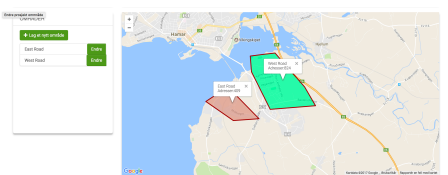


Figur 5.28: Prosjektside Kommuner og Brukere

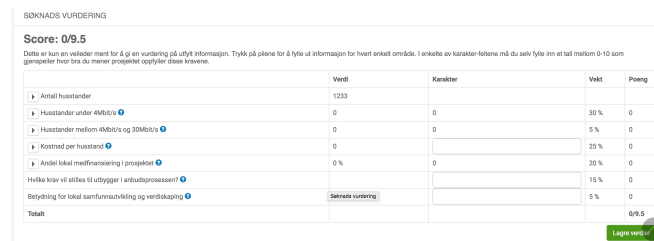


Figur 5.29: Prosjektside Varslinger og Søknader

De får diverse innformasjon om antall husstander i prosjekt områdene og de vil ha mulighet til å estimere hvor bra søknaden er i forhold til Nkom sine krav.



Figur 5.30: Prosjektside Kart og Områder



Figur 5.31: Prosjektside Søknadsvurdering

5.2 Backend

5.2.1 Database

Siden vi hadde valgt å bruke Ruby on Rails som backend, var det naturlig å sette opp databasen ved hjelp av ActiveRecord migrations. Dette gjør at alle enkelt kan opprette en ny tom database på sin lokale maskin, uten å måtte gjøre noe med databasen selv. Vi satt også opp filer for å fylle databasen med generert informasjon, slik at vi hadde informasjon i alle felter i databasen da vi utviklet.

Ved å legge til «timestamps» ved oppsettet av tabellene, vil ActiveRecord oppdatere feltene «created_at» og «updated_at» når vi endrer eller lager et objekt. Det blir også lagt til et felt «id» på alle tabeller som brukes som primær nøkkel.

For å representere tabellene i koden, bruker vi modeller fra ActiveRecord. Disse lar oss hente ut og endre objekter i databasen på en enkel måte uten å måtte direkte oppdatere databasen. Det lar oss også sette opp sammenkoblinger mellom modeller, som ikke finnes i databasen, men som kan lages ved hjelp av flere joins i SQL statementene. ActiveRecord oversetter spørringene våre til SQL slik at vi kan på en raskere og enklere måte søke i databasen, uten å måtte skrive alle SQL statements. Ved å bruke ActiveRecord vil parameterne bli sendt separat slik at vi ikke trenger å tenke på SQL-injection angrep og lignende. Vi har fortsatt mulighet for å skrive deler eller hele SQL-queryen selv, dersom vi ønsker spesiell funksjonalitet fra utvidelser som Postgis.

Dette er et eksempel på hvordan vi kan finne en bruker med en bestemt id ved hjelp av ActiveRecord.

```
user = User.find_by(id: 3)
```

Dette vil lage en SQL query som dette:

```
SELECT "users".* FROM "users" WHERE "users"."id" = $1 LIMIT $2;
```

Parametere:

id	3
LIMIT	1

Her får vi tilbake et «User» objekt som vi kan endre på, og som vil bli lagret til databasen

derom vi kaller metoden `save` på objektet. Det er bare endringene som blir oppdatert, så dersom bruker objectet allerede har fornavnet «Ola» så er det bare feltene «`last_name`» og «`email`» som blir oppdatert.

```
user = User.find_by(id: 3)
user.first_name = 'Ola'
user.last_name = 'Normann'
user.email = "ola@normann.no"
user.save
```

Når vi kaller metoden `save` her vil ActiveRecord kjøre denne forespørselen til databasen:

```
UPDATE "users" SET "last_name" = $1, "email" = $2, "updated_at" = $3 WHERE
  "users"."id" = $4;
```

Parametere:

last_name	Normann
email	ola@normann.no
updated_at	2017-05-14 15:59:54 +0200
id	3

Siden vi kan sette opp koblinger i modellene som ikke finnes i databasen, men som går gjennom, andre tabeller, så vil ActiveRecord kunne spare oss for en del kode. I databasen har vi 3 tabeller for post nummer, kommuner og fylker. Et postnummer hører til en kommune og en kommune hører til et fylke. I modellene har vi lagt til at et postnummer hører til et fylke gjennom en kommune. Vi setter opp denne koblingen ved «`has_one`» metoden i `PostCode` modellen:

```
has_one :county, through: :municipality
```

Dette gir oss metoden `county` på `PostCode` objektet. Denne koden vil derfor først hente ut `PostCode` objektet og deretter finne objektet for fylket.

```
post_code = PostCode.find_by(code: "2818")
post_code.county
```

Først henter den ut post nummeret:

```
SELECT "post_codes".* FROM "post_codes" WHERE "post_codes"."code" = $1 LIMIT
  $2;
```

Parametere:

code	2818
LIMIT	1

Og deretter finner den fylket:

```
SELECT "counties".* FROM "counties" INNER JOIN "municipalities" ON
  "counties"."id" = "municipalities"."county_id" WHERE
  "municipalities"."id" = $1 LIMIT $2;
```

Parametere:

id	78
LIMIT	1

5.2.1.1 Bruk av geografiske data

For å på en bedre måte kunne bruke PostGIS med activerecord bruker vi utvidelsen «activerecord-postgis-adapter» til ActiveRecord. Denne gir oss mulighet til å gjøre enklere spørringer for geografiske data. Men vi må bruke noe SQL dersom vi ønsker å gjøre mer komplekse spørringer.

For å lagre geografiske figurer og punkter til databasen bruker vi markeringsspråket «Well-known text». Med dette kan vi spesifisere ren rekke forskjellige former og figurer. Vi benytter oss hovedsaklig av «POINT» som brukes for å spesifisere et bestemt punkt. Alle posisjoner på forespørsler, og adresser er lagret med dette. Vi bruker også «POLYGON» og «MULTIPOLYGON» som brukes til å spesifisere figurer og grupper av figurer.

«activerecord-postgis-adapter» gir oss mulighet til å lagre figurer som dette.

```
sub_project = SubProject.find_by(id: 1)
sub_project.area = "POLYGON((10.95 60.76,10.94 60.76,10.93 60.76,10.94
  60.76,10.93 60.77))"
sub_project.save
```

Dette vil bli gjort om til en SQL squery som dette.

```
UPDATE "sub_projects" SET "area" = $1, "updated_at" = $2 WHERE
  "sub_projects"."id" = $3 ;
```

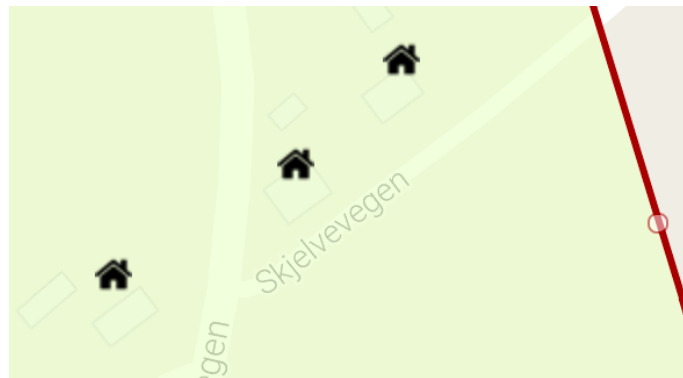
Parametere:

area	002000000300000000000000001000000084025e86380000000404e625c67...
updated_at	2017-05-15 12:03:57
id	1

På flere steder har vi behov for å finne frem til geografiske data i databasen. Blant annet så ønsker vi å på nettsiden kunne vise adresser innenfor et markert område på et kart, og vise oppsummeringer med antallet adresser.



Figur 5.32: Markering av adresser på google maps



Figur 5.33: Nærmere visning av hus i markering

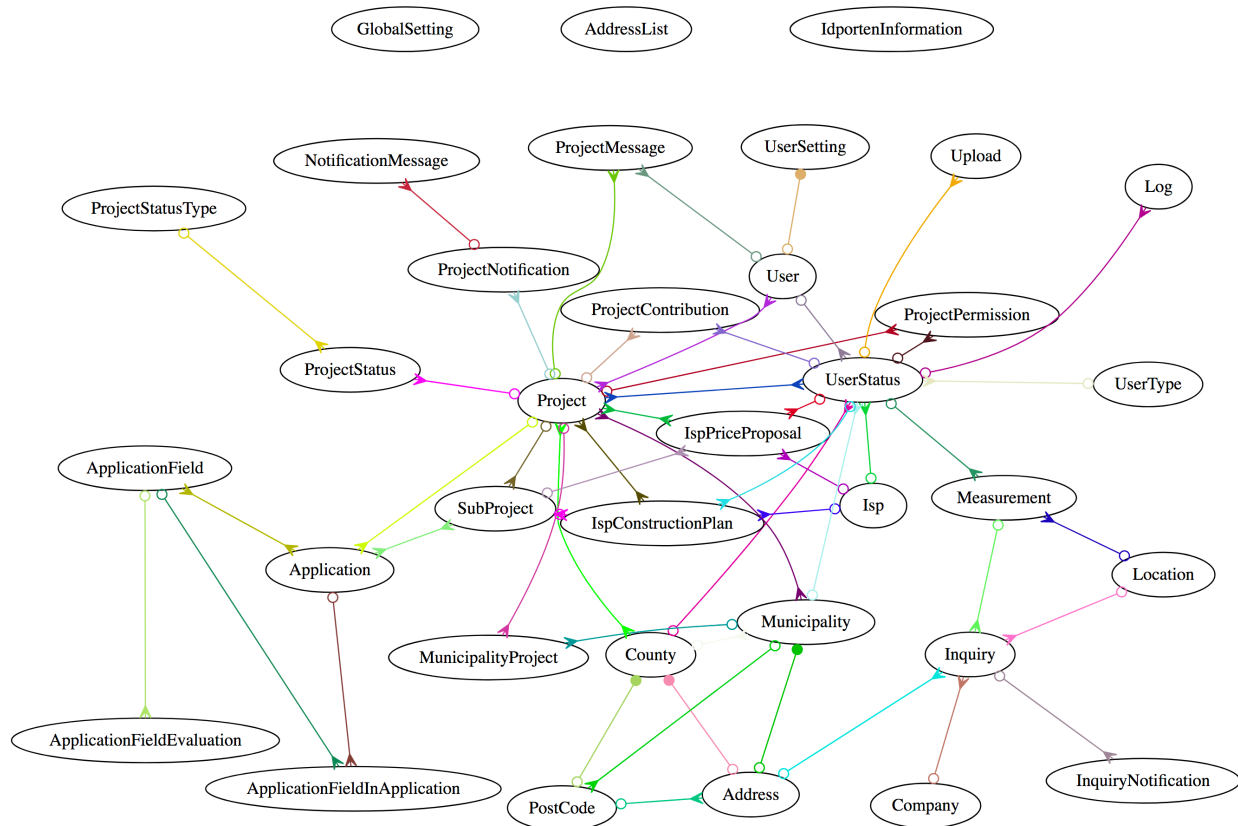
Får å gjøre forespørsler til databasen som bruker nøkkelord fra PostGIS trenger vi å skrive deler av SQL spørringen selv. Dette er en forenklet versjon av hvordan vi gjør spørringen mot AdresseList tabellen. AdresseList inneholder alle registrerte adresser i Hedmark, basert på kartdata vi har fått i fra GIS avdelingen i Hedmark Fylkeskommune. Her vil «##{id}» bli erstattet med id-en til sub prosjektet vi henter adresse informasjon fra.

```
join_query = "INNER JOIN sub_projects ON sub_projects.id=##{id} AND"
join_query += 'st_contains(sub_projects.area, address_lists.possition)'
AddressList.joins(addresses_join_query).limit(limit)
```

Dette vil gi en liste av adresse objekter som er innenfor «sub_projects.area».

5.2.1.2 Database design

Systemet skal lagre en del forskjellige typer informasjon, og vi trengte derfor en relativt stor og kompleks database. Siden vi starter på et prosjekt som skal utvikles videre etter vi er ferdig, så har vi enkelte tabeller og koblinger for funksjonalitet som vi ikke har implementert på frontend, slik at det skal være enklere å utvide med den nye funksjonaliteten som er planlagt. Totalt har vi 35 tabeller.



Figur 5.34: Oversikt over modeller

Figuren viser en oversikt over modellene, og de viktigste koblingene mellom dem. Koblingene er både koblinger fra databasen, og koblinger vi har gjort i modellene. For en fullstendig oversikt over alle tabeller og hva de inneholder se [vedlegg D](#) og [vedlegg E](#)

Her kommer vi til å gå igjennom de viktigste delene av databasen og oppsettet, men kommer ikke til å gå i detalj på alle modeller. Figurene er utsnitt fra figuren over, med de modellene som hører til.

Brukere

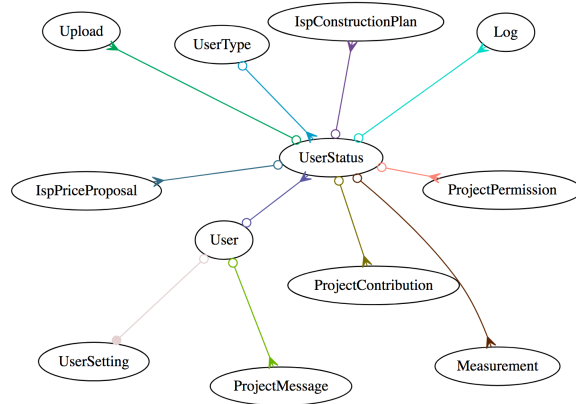
Når det opprettes en ny bruker, opprettes det et User objekt, et UserSetting object, og et UserStatus objekt. User objektet lagrer informasjon om brukeren som navn, epost telefon nummer og passord hash.

UserStatus er koblet til aktiviteter en bruker gjør med en bestemt tilgang, som IspPriceProposal og ProjectPermission. ProjectPermission er koblingen mellom brukeren og et prosjekt dersom en enkeltbruker er gitt direkte tilgang til et prosjekt.

UserStatus lagrer informasjon om hvordan type bruker det og hva brukeren skal ha tilgang til. En bruker som er ansatt i kommunen vil ha en UserStatus for User, og en for Municipality, da brukeren skal ha mulighet tilgjengelighet for normale brukere og kommune brukere.

Vi har fire forskjellige bruker typer:

- **User** Normal bruker
- **Isp** Bruker registrert som en nettverksleverandør
- **Municipality** Bruker registrert som kommune ansatt
- **County** Bruker registrert som ansatt i fylkeskommunen



Figur 5.35: Oversikt over modeller for brukere

Hastighetsmålinger og forespørsler

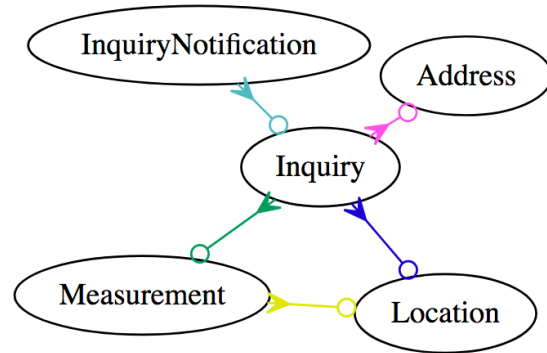
Når det utføres en hastighetsmåling på nettsiden eller appen, blir resultatet lagret til databasen. Da lagrer vi informasjon om hvilke hastigheter som ble oppnådd og hvor målingen ble foretatt. Dersom målingen var gjort gjennom appen lagres det også hvordan type enhet som ble brukt, og hvordan tilkoblings-type brukeren brukte.

Etter at en bruker har utført en hastighetsmåling vil de ha mulighet til å forespørre brede bredbånd. Da opprettes et Inquiry objekt.

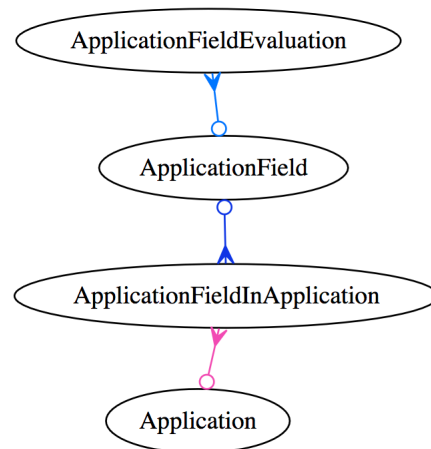
Dette inneholder informasjon om hvordan type bolig det er, hvordan tilbud de har idag, og koblinger til adresse, varslingsinformasjon og posisjon. En InquiryNotification inneholder kontaktinformasjon til personer som har forespurt informasjon om utbygging i området.

Nkom søknad

En Application representerer en søknad til Nkom på et spesifikt prosjekt. Nkom har en rekke felter som skal være med i søknaden. Disse feltene er representert i ApplicationField. Hvert felt har også en ApplicationFieldEvaluation som inneholder informasjon om hvordan feltet blir evaluert på en søknad. ApplicationFieldInApplication inneholder de verdiene som er lagt til for et felt på en spesifikk søknad.



Figur 5.36: Oversikt over modeller for hastighetsmålinger og forespørsler



Figur 5.37: Oversikt over for modeller Nkom søknad

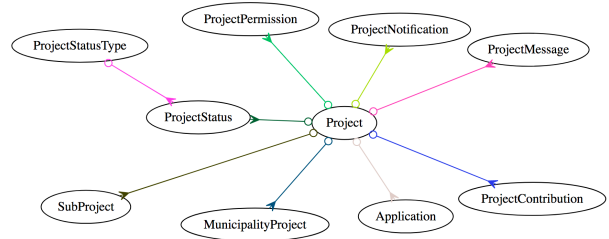
Prosjekter

Prosjekter representerer en gruppe av områder der en kommune ønsker å bygge ut bedre bredbånd. Prosjekt modellen inneholder navnet på prosjektet, og hvordan brukerne selv har rangert prosjektet. Resten av informasjonen til et prosjekt ligger i andre tabeller som referer til et prosjekt.

ProjectStatusType har alle tilgjengelige statuser for et prosjekt.

- Planlegging
- Utlyst
- Klar til søknad
- Sendt
- Utlyst til leverandører
- Under Arbeid
- Ferdigstilt
- Arkivert

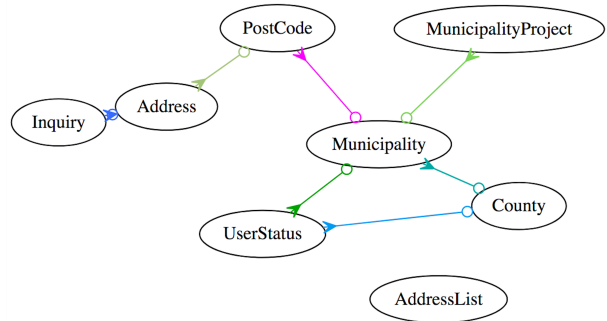
Et prosjekt har alltid en status. Brukeren kan endre statusen frem og tilbake, men det skal lagres hvilke statuser et prosjekt har hatt og når det ble endret. ProjectStatus tabellen er koblingen, som holder informasjon om hvilke statuser et prosjekt har hatt, og når det ble endret. MunicipalityProject og ProjectPermission inneholder informasjon om hvilke kommuner og enkeltbrukere som har tilgang til et prosjekt. ProjectMessage lagrer meldinger som en bruker har lagret



Figur 5.38: Oversikt over modeller for prosjekter

Adresser og områder

Databasen inneholder alle post koder, kommuner og fylker i Norge. Adresser er koblet til en postkode slik at man kan finne frem til kommunen og fylket gjennom den. Det er også en modell AddressList som inneholder alle registrerte adresser i Hedmark, med geografisk posisjon. Denne brukes for å vise hvilke adresser som er innenfor markerte prosjekt områder og å telle opp antall adresser i et område. Kommune objektene brukes for å koble kommune brukerne til et prosjekt. Og en fylkes bruker kobles gjennom fylkes objektet, og de får tilgang til prosjekter for alle fylkets kommuner.



Figur 5.39: Oversikt over modeller for adresser og områder

5.2.2 API

5.2.2.1 Autentisering og verifisering

Får å verifisere en bruker sendes det med en «Authentication» header i forespørelsen. Denne inneholder en Json Web Token som vi genererte da brukeren logget inn eller registrerte seg. Ved denne finner vi id-en til brukeren som sendte requestet, som vi igen bruker til å finne brukeren i databasen. For å sikre hvilke request som skal kreve at det er en logget inn bruker, så bruker vi «before actions» som er et sett av funksjoner som kjører før kontroller metodene. For å kreve at det er en logget inn bruker, så bruker vi «before actionen» «require_login». Denne sikrer at Json Web Tokenen er gyldig og at brukeren finnes.

```
before_action :require_login
```

For noen forespørsler ønsker vi at det bare er de brukerne som er logget inn som en kommune bruker og har tilgang til det nåværende prosjekt som skal a tilgang. For dette bruker vi «before actionen» «require_project_municipality». Dette bruker vi spesielt på endpoints som

brukes for å endre innformasjon på prosjekter som ikke skal kunne endres av enkeltbrukere som har fått tilgang.

5.2.2.2 Reserverte parametere

Enkelte parametere har vi reservert for å identifisere objekter. Vi har reservert parametere både som «snake_case» og «camelCase».

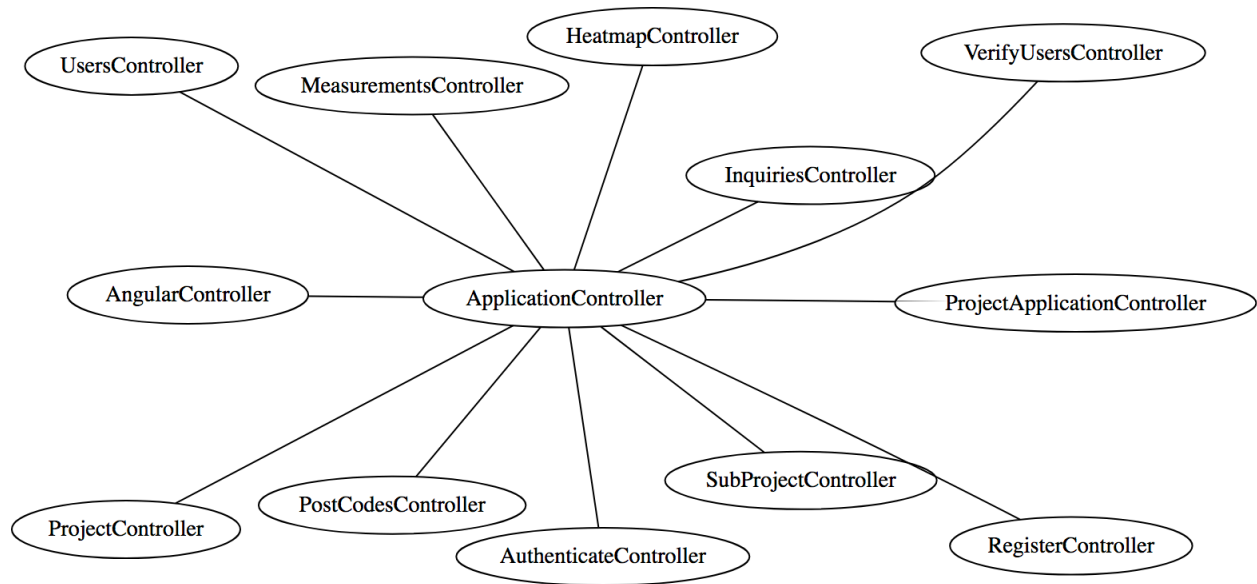
- «**project_id**» brukes for prosjekter.
- «**sub_project_id**» brukes for sub prosjekter.
- «**application_id**» brukes for Nkom søknader.
- «**municipality_id**» brukes for kommuner.

Dersom et av disse parametere sendes med, vil det gjøres lookup til databasen for å finne objektet det blir testet om brukeren har tillatelse til å endre objektet. Dersom brukeren ikke har tilgang vil det sendes en melding tilbake, og kontroller metoden vil ikke kjøre.

```
render json: { message: 'User is not authenticated for this object' },  
             status: :unauthorized
```

5.2.2.3 Kontrollere

Kontrollerene gjør størsteparten av logikken i api-et. Kontrollerene tar i mot forespørseler, behandler parametere, og henter ut informasjon fra databasen som skal brukes i json svaret.



Figur 5.40: Oversikt over kontrollere klasser

Alle kontrollere arver fra «ApplicationController» som har metoder som brukes på tvers av kontrollere. Dette er blant annet metoder for å verifisere at brukere er logget inn, sette globale objekter ut i fra reserverte parametere, og sette standard respons format til json. Vi har totalt 43 api endpoints fordelt på 12 kontrollere.

Oprette nytt prosjekt

Dette endpointet oppretter et nytt prosjekt basert på parameterene «municipalityId» og «new-ProjectName». Siden «municipalityId» er et reservert parameter, blir kommune objektet lagret i variabelen «@municipality»

```

@project = Project.new(name: params[:newProjectName])
if !current_user.municipalities.include?(@municipality)
  render_message('User is not authenticated to create projects for that
    municipality', :unauthorized)
elsif @project.save
  @project.add_municipality(@municipality)
  render :show
else
  render_errors(@project)
end
  
```

Her ser vi at dersom brukeren som er logget inn ikke har tilgang til kommunen som ble sendt med, sendes det tilbake en melding, med en «401 unauthorized» status. Dersom objektet av en eller annen grunn ikke kan lagre, så sendres feilmeldingene tilbake. Dette kan for eksempel være dersom navent som var medsent var lenger enn 255 tegn som er limiten. Dersom prosjektet lagres, legges kommunen til prosjektet slik at alle ansatte i den kommunen får tilgang, og Jbuilder filen for «:show» metoden blir kjørt.

5.2.2.4 Jbuilder

Vi bruker Jbuilder for å skrive json reponsene fra api-et. Jbuilder lar oss på en relativt enkel måte skrive json reponsene på en oversiktlig måte. Jbuilder gjør det også enkelt å sette partials, noe vi har benyttet oss mye av. Vi har partials for de fleste modellene som vi returnerer fra api-et, dette gjør at api-et virker konsistent og returnerer data på likt format på de forskjellige endpointsene. Det sparer oss også for tid ved at vi slipper å lage json objektene flere ganger.

```
json.array! @messages, partial: 'partials/project_message', as: :  
  message
```

Vil hente innholdet i «partials/project_message» og returnere denne json-en.

```
[  
  {  
    "id": 703,  
    "user_id": 143,  
    "user": "Travis Turner",  
    "message": "Melding to.",  
    "date": "15.05.2017",  
    "time": "19.50"  
  },  
  {  
    "id": 702,  
    "user_id": 190,  
    "user": "Carissa Schultz",
```

```
"message": "Melding en.",  
"date": "15.05.2017",  
"time": "19.40"  
}  
]
```

Jbuilder gir oss også muligheten til å hente ut verdier direkte fra objekter, noe vi har benyttet oss av der vi ser at det har vært naturlig.

```
json.extract! isp, :name
```

Dette vil generere dette json objektet

```
{  
  "name": "Telenor"  
}
```

5.2.2.5 Bakgrunnsjobber

Enkelte oppgaver må vi gjøre i bakgrunnen på grunn av at oppgaven tar lang tid. Disse oppgavene er å sende sms og å sende epost. Vi benytter oss i utgangspunktet av «sidekiq» for å kjøre bakgrunnsjobber, men det er satt opp slik at IT-avdelingen hos fylkeskommunen enkelt vil kunne velge ut i fra en rekke løsninger.

Når det sendes varslinger til alle som har forsepurt bedre nettverk i et område, vil det ta lang tid å skulle sende sms og epost til alle, spesielt om det er mange forespørsler.

6 *Testing og kvalitetssikring*

I dette kapitlet tar vi for oss de ulike tiltakene vi har gjort for å forsikre oss om at koden følger en bestemt standard og at kodens integritet er ivaretatt. Dette gjør vi ved hjelp av testing og ved hjelp av aktiv bruk av versjonskontrollverktøyet Git.

6.1 **Testing av Backend**

For testing av backend bruker vi RSpec, og vi har totalt 240 tester som tester backend koden. Det brukes en funksjon «expect» får å definere et resultat, deretter brukes det en rekke forskjellige nøkkelord for å teste om resultatet er det ønskede resultatet.

```
it 'can add user status' do
  measurement = create(:measurement)
  measurement.add_user(create(:user))
  expect(measurement.user_status).not_to be(nil)
end
```

Her testes det at funksjonen «add_user» på et «Measurement» objekt legger til en bruker status.

På backend har vi 100% av ruby koden dekket av tester. Dette er ikke inkludert enkelte funksjoner som er ekskludert fra testing. Dette er funksjoner som for eksempel sender sms, og koster penger per gang, og derfor ikke skal kjøres av tester.

All Files (100.0% covered at 35.23 hits/line)

33 files in total. 710 relevant lines. 710 lines covered and 0 lines missed

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
app/controllers/project_controller.rb	100.0 %	199	83	83	0	1.8
app/models/project.rb	100.0 %	135	69	69	0	31.0
app/models/application.rb	100.0 %	158	69	69	0	155.4
app/controllers/application_controller.rb	100.0 %	138	67	67	0	52.0
app/models/user.rb	100.0 %	123	64	64	0	101.2
app/models/sub_project.rb	100.0 %	83	41	41	0	26.0
app/models/inquiry.rb	100.0 %	79	41	41	0	4.7
app/models/measurement.rb	100.0 %	84	32	32	0	2.4
app/controllers/register_controller.rb	100.0 %	62	28	28	0	2.1
app/controllers/sub_project_controller.rb	100.0 %	49	22	22	0	1.9
app/controllers/project_application_controller.rb	100.0 %	46	18	18	0	1.3
app/controllers/heatmap_controller.rb	100.0 %	32	18	18	0	3.1
app/controllers/authenticate_controller.rb	100.0 %	29	15	15	0	4.1
app/models/user_status.rb	100.0 %	18	14	14	0	1.0
app/controllers/verify_users_controller.rb	100.0 %	25	14	14	0	1.6
app/models/inquiry_notification.rb	100.0 %	23	13	13	0	2.1
app/models/application_field_in_application.rb	100.0 %	23	12	12	0	17.3
app/models/location.rb	100.0 %	22	11	11	0	6.5
app/controllers/measurements_controller.rb	100.0 %	24	11	11	0	1.8
app/controllers/inquiries_controller.rb	100.0 %	25	10	10	0	4.4
app/controllers/users_controller.rb	100.0 %	21	8	8	0	1.0
app/models/municipality.rb	100.0 %	8	6	6	0	1.0
app/models/address.rb	100.0 %	9	6	6	0	1.0
app/models/upload.rb	100.0 %	9	5	5	0	1.0
app/models/project_status.rb	100.0 %	8	4	4	0	1.0
app/models/project_message.rb	100.0 %	7	4	4	0	1.0
app/models/post_code.rb	100.0 %	6	4	4	0	1.0

Figur 6.1: Dekning av rspec tester

6.2 Style guide Backend

På backend brukes rubocop for å sikre at all kode følger vår guide for utseende, kompleksitet og kvalitet. For vår style guide har vi tatt utgangspunkt i rubys «community style guide», som setter en rekke regler for koden.

Dette er eksempler på feil rubocop vil klage på.

app/models/sub_project.rb - 3 offenses

Line #75 – **Convention:** Redundant `self` detected.

```
if self.address_count == info['houses']
```

Line #76 – **Convention:** Operator `=` should be surrounded by a single space.

```
self.inserted_address_count = nil
```

Figur 6.2: Rubocop redundant slef

app/controllers/projects_controller.rb - 56 offenses

Line #2 – **Convention:** Class has too many lines. [313/250]

```
class ProjectsController < ApplicationController ...
```

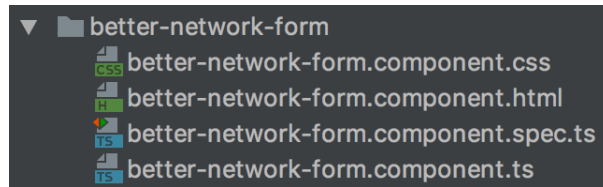
Figur 6.3: Rubocop too many lines

De fleste feilene kan fikses automatisk. Dette kan for eksempel være regler som går på indentering, mellomrom, blanke linjer og lignende. I tillegg til style guiden, har vi vår style guide i tillegg satt som krav at alle offentlige funksjoner, metoder og klasser skal kommenteres.

6.3 Testing av frontend-koden

Hvis man oppretter en ny komponent ved hjelp av Angular CLI, som beskrevet i [kapittel 5](#), vil det automatisk bli generert en fil som skal brukes til testing. Testene for frontend-koden er skrevet ved hjelp av testrammeverket Jasmine. Testene kjøres så ved hjelp av Karma som er en så kalt

test runner. Denne sørger for simulering av brukerinteraksjon i en nettleser. På denne måten kan vi for eksempel teste hva som skjer når vi klikker på en knapp.



Figur 6.4: Filer som bli opprettet når man oppretter en komponent med Angular CLI

Tester som blir skrevet blir kalt «specs». Endelsen på test-filen må være .spec.ts. Det er ved hjelp av filen `better-network-form.component.spec.ts` vi kan teste komponenten som viser skjemaet for å forespørre bedre nett, se figur [figur 6.5](#).

Figur 6.5: Skjema for å forespørre bedre nett

Ved å generere en komponent ved hjelp av Angular CLI, vil mye av innholdet i test-filen bli automatisk generert, og alt blir satt riktig opp slik at det bare er å starte og skrive egne tester. Ettersom vi ikke endrer på dette, vil vi ikke kommentere og vise den koden her.

Vi vil her vise en av testene til denne komponenten. Testen sjekker om et element med id «`hiddenAddressFields`» vises på siden hvis brukeren mener at adresseforslaget fra oss er feil.

```
it('if wrong address, hiddenAddressFields should be set to block', () => {
  fixture.debugElement.query(By.css('#radioButtonAddressNotCorrect'))
    .triggerEventHandler('click', null);
  expect(fixture.debugElement.query(By.css('#hiddenAddressFields'))
```

```
.nativeElement.style.display).toBe('block');
});
```

La oss se på hva som skjer i første del av testen:

```
fixture.debugElement.query(By.css('#radioButtonAddressNotCorrect'))
  .triggerEventHandler('click', null);
```

fixture er noe som blir definert og satt av koden som Angular CLI genererer. fixture inneholder en forekomst av komponenten som testes og et testmiljø som omgir den opprettede komponenten.

fixture.debugElement henter ut komponentens DOM-element

.query tar i mot en parameter og søker i hele DOM-treet etter det første elementet som stemmer overens med parameteren. I dette tilfellet søkes det etter et element med id «radioButtonAddressNotCorrect» og kjører en .triggerEventHandler med parameteren «click» som betyr at det simuleres at det elementet blir klikket på. Dette tilsvarer at brukeren har klikket nei i skjemaet, at adressen vi forslø ikke var riktig.

Det neste som skjer i testen er dette:

```
expect(fixture.debugElement.query(By.css('#hiddenAddressFields'))
  .nativeElement.style.display).toBe('block');
```

expect bruker fixture.debugElement.query til å søke etter et element med id «hiddenAddressFields» og hente ut det elementet sin display-verdi.

.toBe sjekker om verdien som ble hentet ut i expect er «block».

Fra kommandolinja kan vi kompilere og kjøre testene ved å skrive

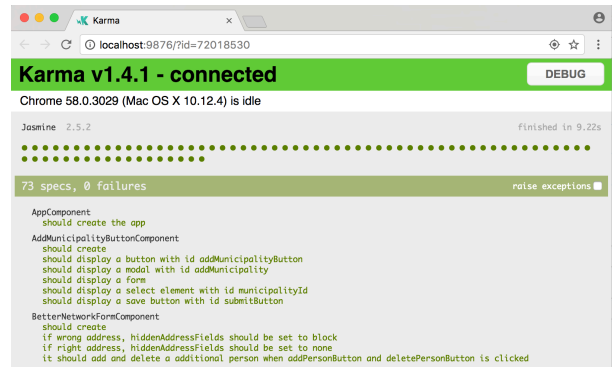
```
npm test
```

Deretter åpner Karma opp en nettleser og begynner å skrive til konsollen. Vi kan både i nettleseren og konsollen se om testene vi har skrevet passerer, i figur [figur 6.6](#) og [figur 6.7](#) ser vi at alle testene våre passerer.

```

15 05 2017 03:50:53.756:WARN [karma]: No captured browser, open http://localhost:9876/
15 05 2017 03:50:53.781:INFO [karma]: Karma v1.4.1 server started at http://0.0.0.0:9876/
15 05 2017 03:50:53.781:INFO [launcher]: Launching browser Chrome with unlimited concurrency
15 05 2017 03:50:53.917:INFO [launcher]: Starting browser Chrome
15 05 2017 03:50:56.332:INFO [Chrome 58.0.3029 (Mac OS X 10.12.4)]: Connected on socket g0NgJkxW07D0HKA4AA with id 72018530
15 05 2017 03:50:56.332:INFO [Chrome 58.0.3029 (Mac OS X 10.12.4)]: Executed 73 of 73 SUCCESS (9.253 secs / 9.174 secs)
    
```

Figur 6.6: Resultatet av testene i konsollen

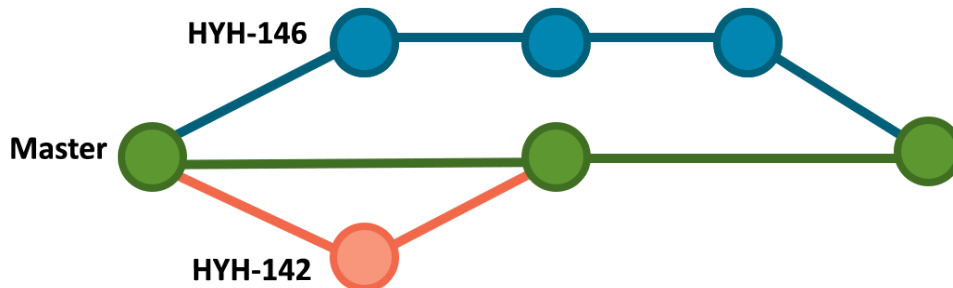


Figur 6.7: Resultatet av testene i nettleseren

6.4 Versjonskontroll

Under utviklingen har vi brukt git som versjonskontrollverktøy. På denne måten har vi alltid hatt tilgang til den nyeste koden og tidligere historikk.

For å unngå å skape konflikt mellom koden vår har vi arbeidet på branches. Master-branchen inneholder den ferdige koden og de andre branchene inneholder funksjonalitet under utvikling. Hver gang vi startet på et nytt issue i Jira lagde vi en ny branch i repositoryet vårt. Branchene har det samme navnet som issue nummeret i Jira.



Figur 6.8: Eksempel på to brancher med issue nummer HYH-146 og HYH-142

I branchen implementerte vi ny funksjonalitet. Når denne var ferdigutviklet satt vi opp et pull request. Det ble da opp til et annet gruppemedlem å undersøke og teste koden. Dersom koden passerte testene kunne branchen bli slått sammen med master-branchen. Dersom koden ikke passerte var det opp til branch eieren å fikse opp i konfliktene. På denne måten har vi forsikret oss om at koden på master alltid fungerer.

Som en ekstra sikkerhetsmekanisme har vi gjort det umulig å sende kode direkte til master branchen. Kode må alltid bli sendt til en branch og bli slått sammen med master derfra. Mesteparten av tiden har det kun vært en person som har jobbet på en branch om gangen. Dette har gjort at vi har hatt lite konflikter underveis.

Branchene har samme navn som issue nummeret i Jira. Dette gjorde vi for at det skulle være lett å navigere i Jira og oppdatere issue-kortene. I tillegg brukte vi smart commits som linket commitene våre i BitBucket direkte til Jira. På denne måten kunne vi i Jira se hvilke brancher som var koblet til kortet og hvilke commits som gjaldt kortet. Dette gjorde navigering mer effektivt samtidig som det var med på å forsikre oss om at arbeidet vårt på branchene var relevant til sprintene våre i Jira.

7 *Avslutning*

7.1 Resultater

7.1.1 Læringsutbytte

I emnebeskrivelsen til IMT3912 Bacheloroppgave[34] settes det mål for hva vi skal ha oppnådd etter endt bachelor. Her settes det mål for både kunnskap og ferdigheter. Vi har tatt utgangspunkt i disse målene for de følgende avsnittene.

Vi har i løpet av de siste månedene opparbeidet oss mye kunnskap om prosessen rundt bredbåndsutbygging. Vi har sammen med en aktiv oppdragsgiver hatt mange dager hvor vi har fokusert på hvordan utbyggingen fungerer. I tillegg har vi vært med på et seminar hvor vi har møtt mennesker som driver innen fagfeltet. Dette har vært med på å bygge opp en forståelse og kompetanse rundt en prosess som vi visste svært lite om fra før av.

Vi har også tilegnet oss kunnskap rundt programmeringsspråk og frameworks som vi kunne lite/ingenting om fra før av. Disse språkene og frameworks-ene ble valgt når vi tok en teknologivurdering på oppgaven. Under vurderingene var vi ute etter relevante teknologier, men også teknologier som kunne øke vårt eget læringsutbytte. Dette har vi oppnådd.

Vi har brukt kunnskap som vi har opparbeidet oss gjennom andre emner for å arbeide på en strukturert og metodisk måte. Vi har fått prøvd ut hvordan det er å anvende denne kunnskapen i et reelt prosjekt.

Vi har fått testet ut våre egne ferdigheter i identifisere og løse en problemstilling av større omfang. Her har vi måtte benytte oss av ferdigheter innenfor analyse, estimering og prioritering i større grad enn tidligere.

Oppgaven har inneholdt flere emner som vi har måttet sette oss inn i. Med de nye teknolo-

giene som vi skulle bruke og med det teoretiske grunnlaget som måtte dannes har vi investert mye tid i å finne relevante og pålitelige kilder og dokumentasjon.

7.1.2 Resultatmål

I forprosjektrapporten([vedlegg H](#)) skrev vi om resultatmålene til oppgaven. Vi skulle utvikle en app til iOS, en app til Android og en nettside. Alle tre skulle kunne gjennomføre hastighetsmålinger og registrere at de ønsker raskere internett. Vi har utviklet to apper og en nettside. På alle tre er det mulig å gjennomføre hastighetsmåling og sende en forespørsel om raskere internett.

Vi skulle lage et varmekart som skulle vise en grafisk fremstilling av dekningsgraden i fylket. Dette målet har vi ikke oppnådd. Denne oppgaven viste seg å ta lengre tid enn estimert, og gjorde at vi ikke rakk å gjennomføre oppgaven.

Vi skulle gjøre det mulig for kommunene og fylkeskommunen å se statistikk og dekningsgrad på kart i bestemte områder. Dette har vi gjort gjennom å lage innlogging for de forskjellige brukergruppene. Fylkeskommunen har ikke tilgang på noe ekstra informasjon per dags dato. Som kommune kan man lage seg prosjekter over bestemte områder. Disse områdene markeres opp på et kart. Dette bruker vi for å finne ut hvor mange husstander som er innenfor området. Kommunene får også opp informasjon om hvem som ønsker raskere bredbånd. Inne på prosjektet kan de fylle inn en veilder som gir de tilbakemelding på hvor bra prosjektet gjør det i forhold til Nkom sine krav.

Til slutt skrev vi i resultatmålene at kommunene skulle kunne bruke den innhentede informasjonen for å raskere skrive og generere søknader til Nkom. På nettsiden kan de fylle ut det obligatoriske vedlegget "Skjema for tilleggsinformasjon" og generere dette. Vedlegget bruker ikke informasjon fra prosjektsiden for å fylle seg selv ut automatisk.

7.1.3 Effektmål

I forprosjektrapporten([vedlegg H](#)) skrev vi at Hedmark fylkeskommune ønsket at systemet skulle være til hjelp for kommunene slik at de kunne få bedre oversikt over bredbåndstilbudet i sitt område. Dette målet har vi delvis oppnådd. Vi har lagd et system som samler inn informasjon om bredbåndshastighetene til brukerne i en omfattende database. Vi samler også inn informa-

sjon om hvem som ønsker seg bedre bredbånd. Kommunen kan se hvem som ønsker seg bedre bredbånd i et område, men har ikke muligheten til å se målingene som er gjort der. I forespørselen om bedre bredbånd får kommuene en del ekstra informasjon om området, men mangler fremdeles hva brukerne fikk på hastighetsmålingene sine. Dette skulle vises i et kart, men kartet ble ikke ferdig i tide. Slik som systemet er i dag kan det brukes til innsamling av informasjon, men ikke fullstendig fremvisning av denne informasjonen.

7.1.4 Gantt diagram

I forprosjektrapporten ([vedlegg H](#)) hadde vi lagd et Gantt diagram som viste de forskjellige sprintene vi hadde tenkt å gjennomføre. I løpet av de første fire sprintene skulle vi lage forprosjektrapporten, lage kravspesifikasjon og dra på seminar for å lære mer om bredbånd. Vi klarte å forholde oss til dette. I den fjerde sprinten skulle også utviklingsmiljøet og databasen settes opp. Dette tok lengre tid en forventet da det viste seg at databasen ble veldig stor. På dette tidspunktet begynte tidsskjema-et å bli forskjøvet.

I femte sprint fortsatte vi å sette opp utviklingsmiljøet og fikk startet på nettsiden og appen. I denne sprinten hadde vi egentlig planlagt å starte å lage hastighetsmålingen, innlogging og varmekart. Dette ble det ikke noe av da det var en del arbeid med å finne ut hvordan ting fungerte.

I den sjetten sprinten kom vi oss i gang med det som skulle vært startet på i sprint 5. Fra dette punktet var vi nøyaktig en uke bak tidsskjema. Vi klarte ikke å jobbe inn igjen den tapte tiden og ble sakte men sikkert liggende mer og mer bak skjema da vi ikke hadde tenkt på alle de små tingene som måtte fikses før vi kunne starte på hvert av punktene i Gantt diagrammet vårt.

Fra og med sprint 13 skulle vi dedikere resten av tiden på å skrive på bacheloroppgaven. Vi fikk startet å skrive litt, men det var mye som ikke var kodet ferdig så vi måtte utsette skrivingen. Til slutt bestemte vi oss for å gjøre skriving parallelt med utvikling frem til innlevering.

Gantt diagrammet vårt stemte dårlig med slik det til slutt ble. Det var mange små ting vi ikke hadde tenkt på før vi startet utviklingen og de nye teknologiene gjorde at vi fikk en treg start. Det var også første gangen vi skulle prøve å estimere, og størrelsen på enkelte av oppgavene ble undervurdert. Oppgavene i Gantt diagrammet var ment for fire utviklere, men ble i realiteten utført av tre. Dette gjorde at det var vanskelig å komme i mål hver sprint.

7.2 Drøftinger og valg

Det ble gjort en rekke drøftinger og valg underveis i oppgaven. Disse har vi skrevet om i de kapitlene der de er relevante:

- Kapittel 1: Hvorfor vi jobber etter Scrum
- Kapittel 4: Utvikle appen native eller i tredjeparts framework
- Kapittel 4: Hvilket rammeverk å bruke på frontend nettsiden
- Kapittel 4: Hvilket rammeverk å bruke på backend
- Kapittel 4: Hvilke farger å bruke (fargeprofil)

I tillegg til disse drøftingene hadde vi en periode der vi ble nødt til å vurdere å bytte fra Ruby til PHP. Dette skjedde da IT-tjenesten i Hedmark Fylkeskommune ønsket å sette opp et nytt krav om bruk av PHP etter vi hadde startet utviklingen. Til denne anledningen lagde vi et felles dokument for vår gruppe hvor vi tok for oss fordeler og ulemper ved å bytte, se [vedlegg F](#). Et av hovedargumentene for å velge PHP var at alle på gruppen hadde arbeidet med det før, problemet med dette var at ingen av oss hadde brukt det i et rammeverk før. I tillegg hadde vi kommet godt i gang med Ruby, og vi ville ha mistet mange timers arbeid ved å bytte. Vi bestemte oss for å fortsett å bruke Ruby.

I planleggingsfasen hadde vi planlagt oppgaver som skulle fordeles utover fire gruppelemmer. Vi jobbet lenge etter denne planen. Vi mistet etterhvert et gruppe medlem. I tillegg hadde vi estimert at enkelte oppgaver skulle ta en uke, men noen av oppgavene viste seg å fort ta tre uker. Med vår feilestimering og tap av ressurser ble vi etterhvert nødt til å nedprioritere enkelte deler av systemet.

Vi hadde allerede investert mye tid i å sette opp en omfattende database som skulle dekke alle aspekter av systemet. Vi hadde også kommet oss godt i gang med appen og fått lagd deler av hastighetsmålingen og forespørselen om raskere bredbånd. Dette er kjernefunksjonalitet som alt annet bygger på. Det ble smått med tid og vi måtte vurdere hva som skulle nedprioriteres og hva som skulle droppes helt. Vi bestemte oss da for å droppe alt som handlet om leverandører fordi leverandørene har lite betydning for systemet i helhet. Vi bestemte oss etterhvert

for å fullføre kjernefunksjonaliteten med innhenting av data og fokusere på å lage delen som kommunene skulle arbeide på. Dette valgte vi fordi kommune delen er den delen av systemet hvor den innhentede dataen blir brukt. Det er også kommunene som i størst grad skal bruke systemet.

7.3 Videre arbeid

Hedmark fylkeskommune ønsker at dette prosjektet skal utvikles videre, og bli et verktøy som kommunene kan ta i bruk våren 2018. De har derfor planer om å ansatte noen for å ferdigstille dette frem til da.

Funksjonalitet som vi ikke har prioritert eller ikke blitt ferdig med er:

- Motta varslinger i appen
- Logge inn via appen
- Ta ut et lite område innenfor et markert område på kartet, som har god bredbåndshastighet
- Dersom en bruker forespør nettverk i et område der det allerede er utbygget bedre bredbånd fra andre leverandører skal brukeren få informasjon om hvilke leverandører som tilbyr dette.
- Funksjonalitet for leverandører
- Videreutvikling av funksjonalitet for ildskjeler og kommuner.
- Legge til støtte for engelsk språk på nettsiden
- Grafisk fremtilling av dekningsgraden i et varmekart
- Grafisk fremtilling av interessen for bedre nett i et varmekart
- Ekstra funksjonalitet for fylkeskommune, som for eksempel å hente ut ulike statistikk
- Bruke informasjon automatisk fra prosjektsiden for å fylle inn Nkom-søknaden

- Siden vart hovedfokus har vært på utvikling av funksjonalitet, ser vi at det er muligheter for forbedring av brukergrensesnitt og universell utforming.

Vår vurdering er at det nok arbeidsoppgaver til en ny Bacheloroppgave, spesielt fordi oppdragsgiver kommer på ny funksjonalitet til systemet hele tiden. Vi tror ikke det er aktuelt for Hedmark fylkeskommune å videreføre disse punktene til en ny Bacheloroppgave, da de ønsker at systemet skal komme i bruk så fort som mulig.

7.4 Evaluering av gruppas arbeid

7.4.1 Innledning

Gruppen har gjennom bacheloroppgaven vært svært motivert og har investert mye tid i oppgaven. Det har blitt brukt mange timer i fellesskap på skolen hvor vi har arbeidet aktivt med oppgaven, men også mange timer på fritiden har gått med å arbeide med oppgaven. Vi har hatt en organisering i gruppen som har fungert bra, men ser vi kunne ha fordelt arbeidet litt anderles for å øke effektiviteten.

7.4.2 Organisering

Vi brukte flere verktøy for å organisere oss innad i gruppen. Vi har alltid hatt tilgang på hverandres kode gjennom Bitbucket og alt av dokumenter gjennom Google Drive og ShareLatex. Disse verktøyene har gjort det enkelt å holde seg oppdatert på hverandres arbeid. Ved å bruke Jira har vi hatt et ryddig og tilgjengelig Scrum-Board hele tiden. Dette har gjort det enkelt for oss å holde orden på hvem som har jobbet med hva og når. Alle gruppemedlemmene har aktivt brukt verktøyene noe som har vært med på å holde gruppen organisert til en hver tid.

7.4.3 Fordeling av arbeidet

Vi bestemte oss tidlig for å jobbe parallelt på nettsiden og appen. I starten av prosjektet fordelte vi oss slik at en skulle jobbe primært med å sette opp databasen, en skulle jobbe med å lage app og to skulle jobbe med nettsiden. Dette fungerte greit frem til vi skulle begynne å koble appen

og nettsiden opp mot databasen. Siden det kun var en som visste hvordan å gjøre dette gikk det veldig tregt når de andre skulle prøve å lære seg dette. Ganske sent i utviklingen bestemte vi å fordele oss slik at de som hovedsakelig jobbet på frontend skulle holde seg til dette, og at det ene gruppemedlemmet som kunne Ruby on Rails fra før av skulle få håndtere å sette opp koblingen mellom databasen og frontend. Dette gjorde arbeidet mer effektivt.

7.4.4 Prosjekt som arbeidsform

Gruppen har gjennom skolen jobbet mye med prosjekter før. Prosjektene har aldri vært i nærheten av så store eller krevende. Vi har i løpet av oppgaven fått anvendt mye av det vi har lært i andre emner og sett hvordan dette fungerer i en større oppgave. Dette har vært svært lærerikt og er med på å forberede oss for arbeidslivet.

7.5 Konklusjon

Vi visste fra starten av at systemet ikke skulle ferdigstilles innen endt bachelor. Selv om vi ikke rakk å gjøre alle oppgavene som vi hadde planlagt å fullføre fikk vi på plass det viktigste av kjernefunksjonaliteten. Vi sitter igjen med et system som kan samle inn verdifull informasjon om innbyggernes bredbånd og som delvis kan utnytte denne informasjonen. Systemet er lagd slik at det skal være mulig å utvikle videre. Deler av systemet, slik som databasen, bærer preg av å skulle tilhøre et større system.

Vi valgte å bruke teknologier som vi hadde varierende kunnskap til fra før av. Vi ser i ettertid at vi heller skulle ha organisert gruppen slik at de forskjellige gruppemedlemmene fikk muligheten til å spesialisere seg innenfor hvert sitt felt. På denne måten kunne vi kanskje ha spart oss for noe av tiden som gikk bort på at alle skulle prøve å lære alt fra bunnen av. På den positive siden sitter alle sammen igjen med ny kunnskap.

Til tross for at det var mye nytt som skulle læres er vi fornøyde med de teknologivalgene vi har gjort. Det var mye informasjon som skulle ivaretas og det viste seg å være lurt å bruke Ruby on Rails for å gjøre det mer effektivt å sette opp vår store database. Xamarin Forms bydde på noen utfordringer når vi skulle gjennomføre hastighetsmålinger, men samtidig sparte vi mye tid i resten av app-utviklingen ved at vi kunne bruke mye av den samme koden på Android og iOS.

Oppgaven har vist seg å være en lærerik opplevelse. Vi har fått prøvd oss ut på et ukjent fagområde som har krevd en del teoretisk forståelse. Dette har gitt oss utfordringer både i lagingen av kravspesifikasjonen og i implementeringen av systemet. Vi har også fått muligheten til å møte målgruppen for systemet, og ser at oppgaven er virkelighetsnær og at systemet er høyt ønsket.

Etter å ha arbeidet på et prosjekt av denne størrelsesordenen i en lengre periode har vi fått testet ut våre egne ferdigheter innenfor programvareutvikling og sitter igjen med verdifull erfaring.

Bibliografi

- [1] Xamarin.Forms.ProgressBar Class [Web Page]; [cited 15/05-17]. Available from: <https://developer.xamarin.com/api/type/Xamarin.Forms.ProgressBar/>.
- [2] Norsk mediebarometer, 2015 [Web Page]; 2016 [cited 24/01-17]. Available from: <https://www.ssb.no/medie/>.
- [3] Internett-målinga, 3. kvartal 2016 [Web Page]; 2016 [cited 24/01-17]. Available from: <https://www.ssb.no/teknologi-og-innovasjon/statistikker/inet>.
- [4] Bedre bredbånd til folket [Web Page]; 2014 [cited 27/01-17]. Available from: <https://www.hedmark.org/Om-fylkeskommunen/Naeringsutvikling/Bredbaandsprosjekt-i-Hedmark/Bedre-bredbaand-til-folket>.
- [5] Gjennomføringen av prosjektet [Web Page]; 2014 [cited 27/01-17]. Available from: <https://www.hedmark.org/Om-fylkeskommunen/Naeringsutvikling/Bredbaandsprosjekt-i-Hedmark/Gjennomfoeringen-av-prosjektet>.
- [6] Stønad til breibandutbygging [Web Page]; 2016 [cited 08/05-17]. Available from: <http://www.nkom.no/spørsmål-og-svar/stønad-til-breibandutbygging>.
- [7] De rene/tørre fakta om Frode Haug [Web Page]; [cited 04/05-17]. Available from: <http://frode.haugianerne.no/personal.html>.
- [8] Internett [Web Page]; 2017 [cited 08/05-17]. Available from: <https://no.m.wikipedia.org/wiki/Internett>.
- [9] Kva typar breiband finst? [Web Page]; 2014 [cited 08/05-17]. Available from: <https://www.nkom.no/teknisk/bredb%C3%A5nd/breiband/kva-typar-breiband-finst>.

-
- [10] Lov om elektronisk kommunikasjon [Web Page]; 2017 [cited 08/05-17]. Available from: <https://lovdata.no/dokument/NL/lov/2003-07-04-83>.
- [11] Om Nasjonal kommunikasjonsmyndighet (Nkom) [Web Page]; 2017 [cited 08/05-17]. Available from: <https://www.nkom.no/om-nkom>.
- [12] Next Generation Access Networks (NGA) [Web Page]; [cited 08/05-17]. Available from: <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=URISERV%3Aasi0018>.
- [13] How does the test itself work? How is the result calculated? [Web Page]; 2012 [cited 10/04-17]. Available from: <https://support.speedtest.net/hc/en-us/articles/203845400-How-does-the-test-itself-work-How-is-the-result-calculated>.
- [14] What is "ping", "download speed", and "upload speed"? [Web Page]; 2014 [cited 10/04-17]. Available from: <https://support.speedtest.net/hc/en-us/articles/203845290-What-is-ping-download-speed-and-upload-speed>.
- [15] A very beautiful old lady II [Web Page]; [cited 15/05-17]. Available from: <https://www.flickr.com/photos/pedrosimoes7/443738371/>.
- [16] March 6, 2011 [Web Page]; [cited 15/05-17]. Available from: <https://www.flickr.com/photos/jenumfamily/5502460911/>.
- [17] 297/366 [Web Page]; [cited 15/05-17]. Available from: <https://www.flickr.com/photos/andreasoverland/8116045577/>.
- [18] Business Portrait [Web Page]; [cited 15/05-17]. Available from: <https://www.flickr.com/photos/jenumfamily/10382700334/>.
- [19] Business Portrait [Web Page]; [cited 15/05-17]. Available from: <https://www.flickr.com/photos/hendrikwieduwilt/24024050690/>.
- [20] Shared Projects [Web Page]; [cited 03/05-17]. Available from: https://developer.xamarin.com/guides/cross-platform/application_fundamentals/shared_projects/.

-
- [21] Introduction to Portable Class Libraries [Web Page]; [cited 03/05-17]. Available from: https://developer.xamarin.com/guides/cross-platform/application_fundamentals/pcl/introduction_to_portable_class_libraries/.
- [22] THE DEAD-SIMPLE BOOTSTRAP 4 BUILDER [Web Page]; [cited 03/05-17]. Available from: <http://pingendo.com/>.
- [23] Universell utforming [Web Page]; 2017 [cited 13/05-17]. Available from: https://no.m.wikipedia.org/wiki/Universell_utforming.
- [24] Forskrift om universell utforming av informasjons- og kommunikasjonsteknologiske (IKT)-løsninger [Web Page]; 2013 [cited 13/05-17]. Available from: <https://lovdata.no/dokument/SF/forskrift/2013-06-21-732>.
- [25] Kva seier forskrifta? [Web Page]; 2016 [cited 13/05-17]. Available from: <https://uu.difi.no/krav-og-regelverk/kva-seier-forskrifta>.
- [26] Bilder og grafikk [Web Page]; 2017 [cited 13/05-17]. Available from: <https://uu.difi.no/krav-og-regelverk/losningsforslag-web/bilder-og-grafikk>.
- [27] Skjema [Web Page]; 2017 [cited 14/05-17]. Available from: <https://uu.difi.no/krav-og-regelverk/losningsforslag-web/skjema>.
- [28] Feilhåndtering [Web Page]; 2017 [cited 14/05-17]. Available from: <https://uu.difi.no/krav-og-regelverk/losningsforslag-web/feilhandtering>.
- [29] Registrering og innlogging [Web Page]; 2017 [cited 14/05-17]. Available from: <https://uu.difi.no/krav-og-regelverk/losningsforslag-web/registrering-og-innlogging>.
- [30] Kontrast [Web Page]; 2017 [cited 15/05-17]. Available from: <https://uu.difi.no/krav-og-regelverk/losningsforslag-web/kontrast>.
- [31] Xamarin Forms Dependency Injection [Web Page]; 2017 [cited 10/05-17]. Available from: <https://xamarinhelp.com/xamarin-forms-dependency-injection/>.

- [32] Localizing Xamarin.Forms Apps with RESX Resource Files [Web Page]; 2016 [cited 11/05-17]. Available from: <https://developer.xamarin.com/guides/xamarin-forms/advanced/localization/>.
- [33] Geocoding API [Web Page]; 2017 [cited 12/05-17]. Available from: <https://developers.google.com/maps/documentation/geocoding/intro>.
- [34] IMT3912 - Bacheloroppgave - IMT [Web Page]; [cited 15/05-17]. Available from: <https://www.ntnu.no/studier/emner/IMT3912>.

Ordliste

Asynkron En asynkron handling er en handling som kan gjennomføres samtidig som andre handlinger gjennomføres. [23–25](#)

Conditional Compilation Conditional Compilation er kompilering der man sender med et parameter for å produsere forskjellige resultater i kompileringen. [45](#)

Cross-platform Utvikling hvor man kan bruke samme kode for å utvikle til flere plattformer samtidig. [9](#)

Daily Scrum Meeting Et møte som avholdes hver arbeidsdag i en sprint. På disse møtene må hvert gruppelem fortelle hva som ble gjort i går, hva som skal gjøres i dag og om det eventuelt er noen umiddelbare problemer som må løses for at arbeidsdagen skal gå bra. [14](#)

Dependency Injection Dependency Injection vil si at man mottar nødvendige objekter annestedsfra i stedet for å lage dem selv. Dette kan gjøres ved å lage en generalisert interface som man spesialisierer. [45](#)

Grunnleggende bredbånd Bredbånd med minst 4 Mbit/s nedlastningshastighet. [7](#)

Hvitt område Et område som regnes for å ikke ha et grunnleggende tilbud på bredbånd (under 4Mbit/s). [iii, 20](#)

Høyhastighetsbredbånd Bredbånd med minst 30 Mbit/s nedlastningshastighet. [7](#)

Ildsjel En engasjert innbygger som gjør ekstra arbeid i forhold til bredbåndsutbyggingen. Har tilgang på ekstra informasjon. [3, 6, 8, 34](#)

Issue tracking Oversikt over oppgaver og feil som oppdages i løpet av prosjektet og statusen på disse. Vanlige statuser er skal gjøres, under arbeid og ferdig. [15](#)

Native utvikling Utvikling som er gjort for en bestemt platform i denne platformen sitt "mors-mål". iOS apper skrives i Objective-C eller Swift, og Android apper skrives i Java. [44](#)

NGA område Et område som har tilbud om bredbånd på minst 30 Mbit/s nedlastningshastighet. [iii](#), [20](#)

Nkom Kort for "Nasjonal kommunikasjonsmyndighet". Driver tilsyn med dem som tilbyr post- og teletjenester. Gir støtte for bredbåndsutbygning. [2](#), [4](#), [6](#), [7](#), [18](#), [20](#)

Ping Ping, også kjent som responstid, er tiden det tar for en klient å få svar fra en server. [21](#), [22](#)

Product Backlog En prioritert liste over funksjonaliteter og arbeidsoppgaver som systemet skal inneholde. Denne listen vokser og forandrer seg gjennom hele prosjektet. [14](#)

Produkteier Produkteier representerer interessentene sine visjoner. Produkteier skal ta beslutninger om hva som skal prioriteres gjennom hele prosjektet. [10](#)

Scrum Master Scrum Master har et overordnet blitt over hele prosjektet og at samarbeidet mellom produkteier og utviklingsteamet fungerer godt. [10](#), [14](#)

Scrum-board For å synliggjøre arbeidsoppgavene i sprint backloggen kan vi bruke et scrum-board. Dette har forskjellige kolonner der oppgavene kan plasseres under: skal gjøres, under arbeid og ferdig. Oppgavene blir flyttet rundt på brettet gjennom sprinten. [15](#)

Scrum-teamet De personene som gjør det nødvendige arbeidet i prosjektet. Scrum-teamet består av Produkteier, Scrum-master og Utviklingsteamet. [10](#)

Sjablongvurdering I dette tilfellet en vurdering på hvor mye en leverandør tror det vil koste per husstand å bygge ut i et bestemt område. [2](#)

Sprint En fast tidsperiode scrum-teamet jobber. Målet i løpet av sprinten er å jobbe med noen utvalgte arbeidsoppgaver, og ingen andre oppgaver utover dette. [14](#)

Sprint Backlog Oversikt over oppgaver som skal gjennomføres i løpet av en sprint. Innholdet i denne listen bestemmes på Sprint Planning Meeting. [14](#)

Sprint Planning Meeting Møtet avholdes på starten av hver sprint. Produkteier, Scrum Master og utviklingsteamet deltar på disse møtene. Produkteier beskriver de arbeidsoppgavene med høyest prioritet. Utviklingsteamet stiller spørsmål slik at arbeidsoppgavene blir ordentlig forstått. Resultatet av møtet bør være at neste sprint er planlagt og at Product Backlog er oppdatert. [14](#)

Sprint Review Meeting Et møte der Scrum-teamet viser og evaluerer hva som har blitt gjort i løpet av sprinten. Dette møtet avholdes etter hver sprint. [14](#)
















Utviklingsteamet Det er medlemmer av utviklingsteamet som gjør det som er nødvendig for å levere ferdige arbeidsoppgaver på slutten av hver sprint. [10](#)

Varmekart Et kart med et lag av farger på toppen som markerer konsentrasjonen av noe. [5](#)

VEDLEGG A

Skjermdump fra "<http://speedtest1.telenor.net/mini/speedtest/>", 10.04.17

Index of /mini/speedtest

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 expressInstall.swf	2009-08-24 14:20	773	
 latency.txt	2006-07-28 17:31	10	
 random350x350.jpg	2006-03-06 16:26	240K	
 random500x500.jpg	2008-08-07 11:31	494K	
 random750x750.jpg	2006-03-06 16:26	1.1M	
 random1000x1000.jpg	2006-03-06 16:26	1.9M	
 random1500x1500.jpg	2006-03-06 16:26	4.3M	
 random2000x2000.jpg	2006-03-06 16:26	7.5M	
 random2500x2500.jpg	2006-08-17 20:36	12M	
 random3000x3000.jpg	2006-08-17 20:36	17M	
 random3500x3500.jpg	2006-08-17 20:37	23M	
 random4000x4000.jpg	2006-08-17 20:37	30M	
 swfobject.js	2009-06-11 16:14	10K	
 upload.php	2007-08-28 23:26	272	

Apache/2.4.7 (Ubuntu) Server at speedtest1.telenor.net Port 80

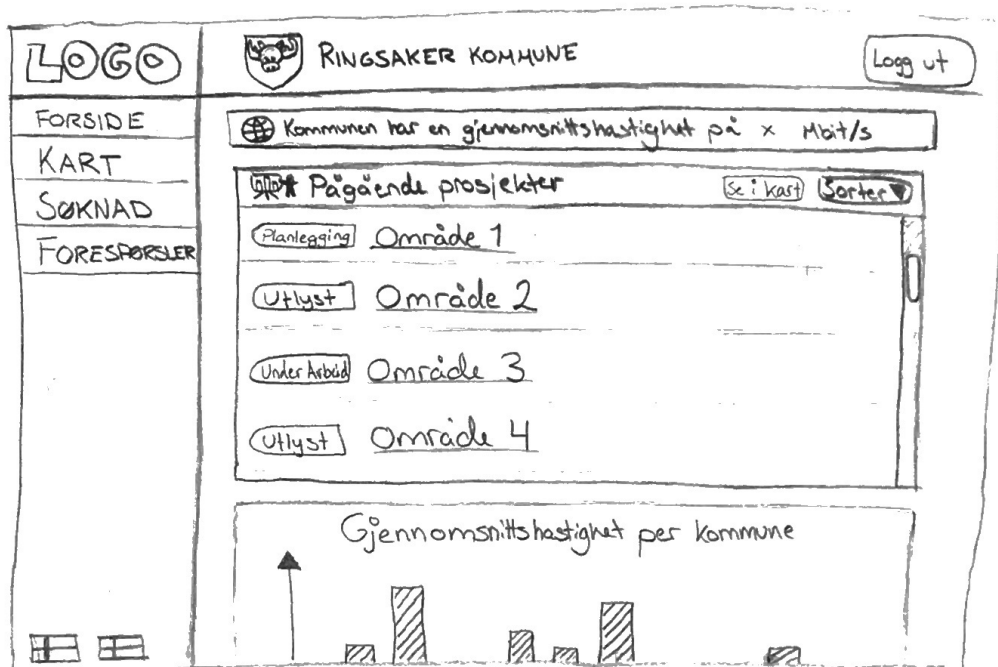
VEDLEGG B

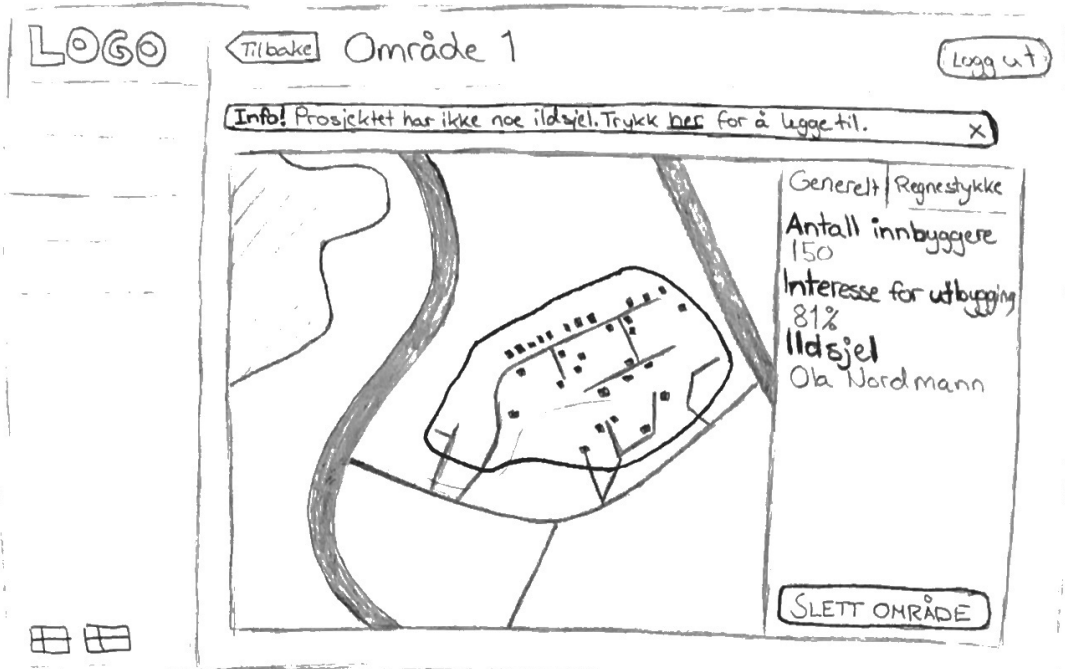
Lo-fi prototype skisser

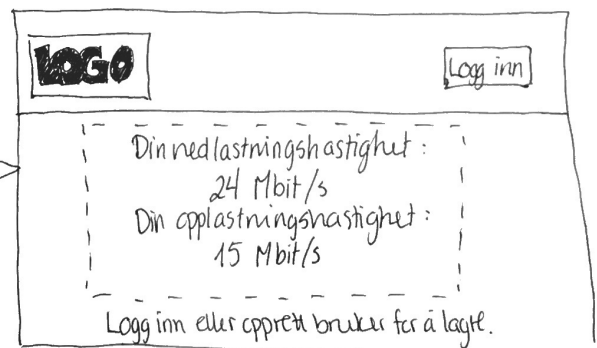
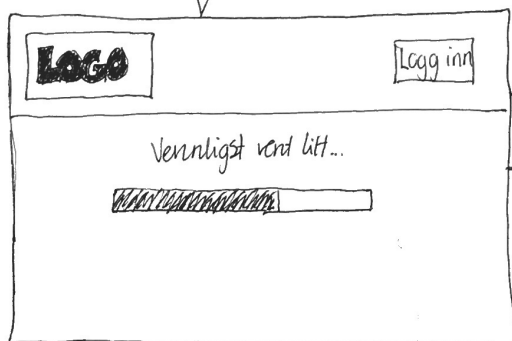
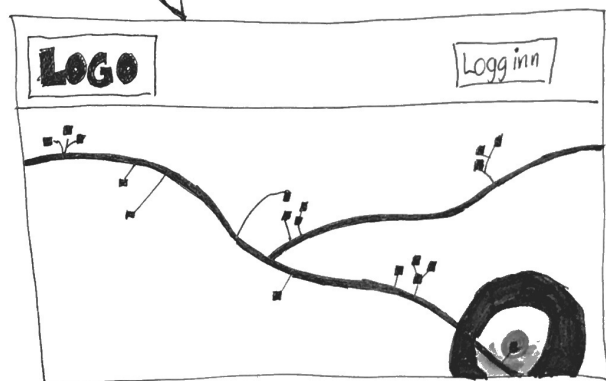
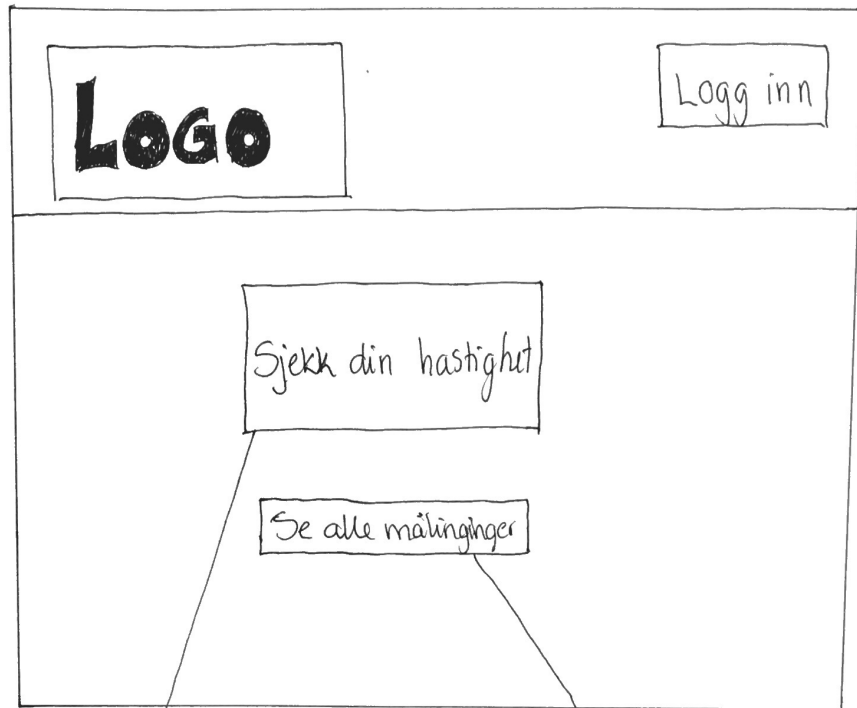
5/2



Generisk bilde av Hedmark







Logget inn som kommune-bruker:

Logo

Kommune navn

Logg ut

Prosjekter

Gaupen

Statistikk

Prosjekt 2

Innstillinger

Rudshøgda

Jessnes

Nes

Se tidligere søknader

Jessnes

Opprette nytt område

Kommer opp vår knapp blir trykket

Logo

Gaupen

Logg ut

Prosjekter

Statistikk

Innstillinger

Kart

Namnekart

Iidsjeler

Thomas Hansen

Legg til iidsjel

Antall innbyggere	15
Interessenter	7
Under 4 Mbit/s	2
Mellom 4 og 30 Mbit/s	5

Se liste over alle interessenter

Send melding til alle interessenter

Rediger søknad

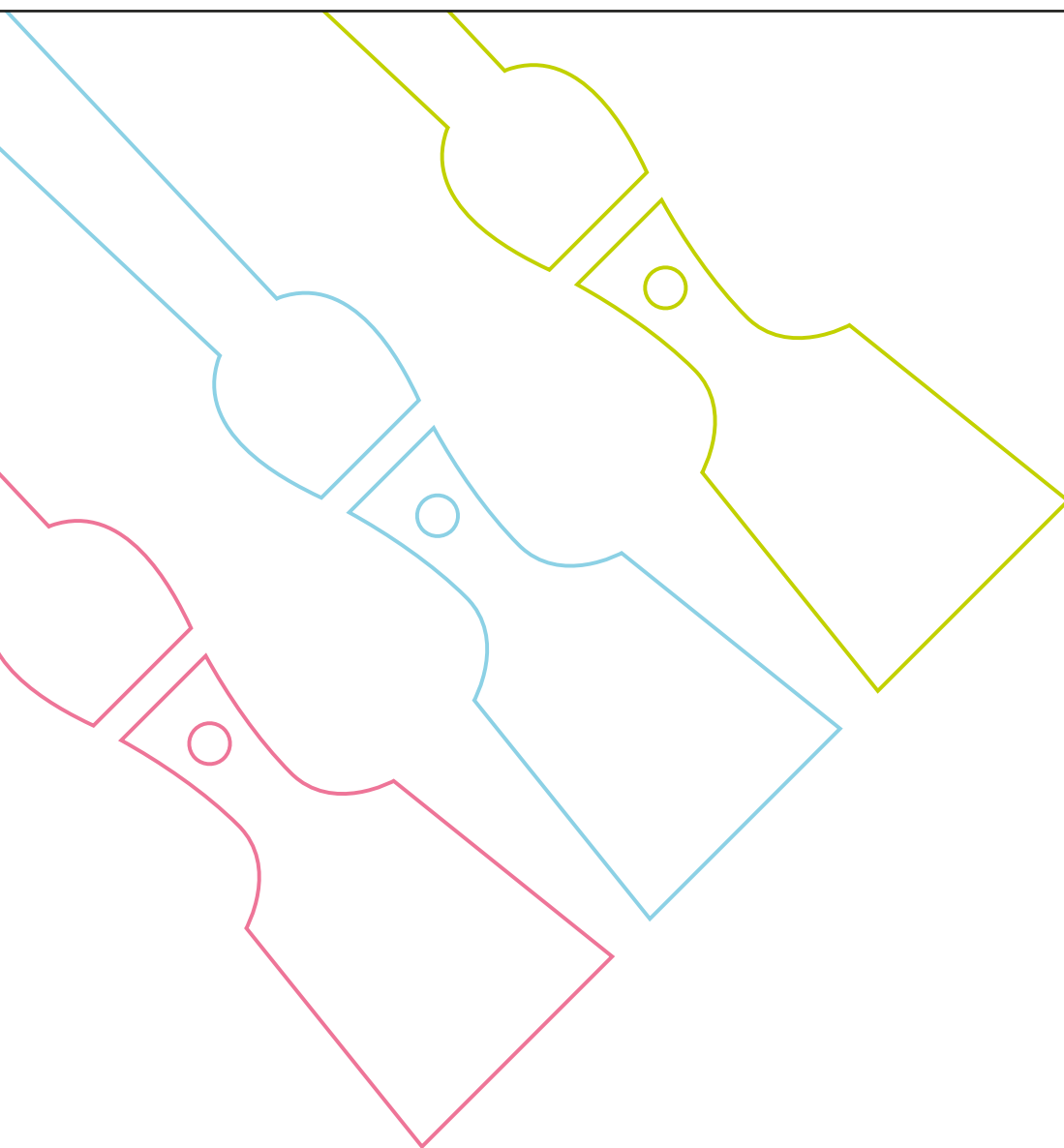
Velg prosjekt blir maskert

VEDLEGG C

Utdrag av HFK Profilmanual 2017

Profilmanual


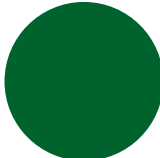
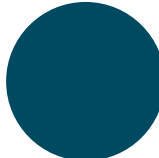

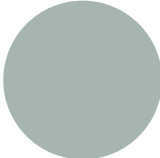

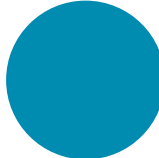


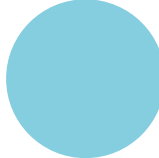
for Hedmark fylkeskommune



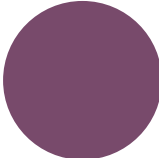
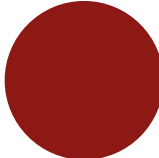
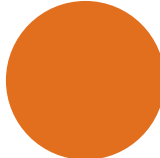
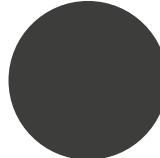
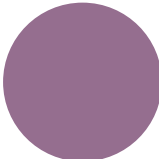


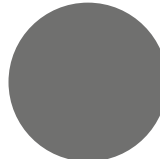
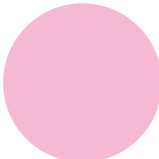
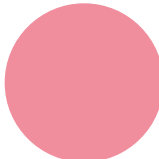


Farger

Edvard Munch er født i Løten og kom fra en skogbruksfamilie. Hans malerier med sterke farger som beskriver stemninger og følelser er utgangspunkt for Hedmark fylkeskommunes nye fargepalett. Hver fargeserie har fått et navn, og deles igjen inn i lys, klar og mørk. Det finnes i tillegg en fargeserie i grader av svart.

Alle lyse farger kan benyttes i dekk fra 10 til 100 %. Fargen Munch som benyttes i det heraldiske våpenet, samt alle klare og mørke farger kan kun benyttes i dekk 100 %.

	Munch	Grafikk	Løten	Åsgårdstrand
Mørk		 PMS: 5487 CMYK: 35-0-16-54 RGB: 89-123-124	 PMS: 357 CMYK: 80-0-100-56 RGB: 0-99-37	 PMS: 309 CMYK: 100-0-9-72 RGB: 0-74-97
Klar	 PMS: 361 CMYK: 69-0-100-0 RGB: 84-185-72 RAL: 6018	 PMS: 443 CMYK: 12-0-12-30 RGB: 166-180-172	 PMS: 370 CMYK: 56-0-100-27 RGB: 94-151-50	 PMS: 633 CMYK: 100-0-10-25 RGB: 0-139-176
Lys		 PMS: 441 CMYK: 6-0-7-9 RGB: 217-226-219	 PMS: 382 CMYK: 29-0-100-0 RGB: 193-216-47	 PMS: 636 CMYK: 45-0-9-0 RGB: 131-210-228



	Madonna	Vampyr	Skrik	Svart
Mørk	 PMS: 5125 CMYK: 65-86-49-0 RGB: 121-73-106	 PMS: 188 CMYK: 0-97-100-50 RGB: 139-14-4	 PMS: 159 CMYK: 0-66-100-7 RGB: 227-111-30	 90 %
Klar	 PMS: 5135 CMYK: 47-64-28-0 RGB: 149-110-142	 PMS: 187 CMYK: 0-100-79-20 RGB: 196-18-48	 PMS: 151 CMYK: 0-48-95-0 RGB: 248-151-40	 60 %
Lys	 PMS: 230 CMYK: 0-34-0-0 RGB: 247-185-212	 PMS: 190 CMYK: 0-55-22-0 RGB: 244-143-156	 PMS: 121 CMYK: 0-11-69-0 RGB: 255-222-108	 20 %

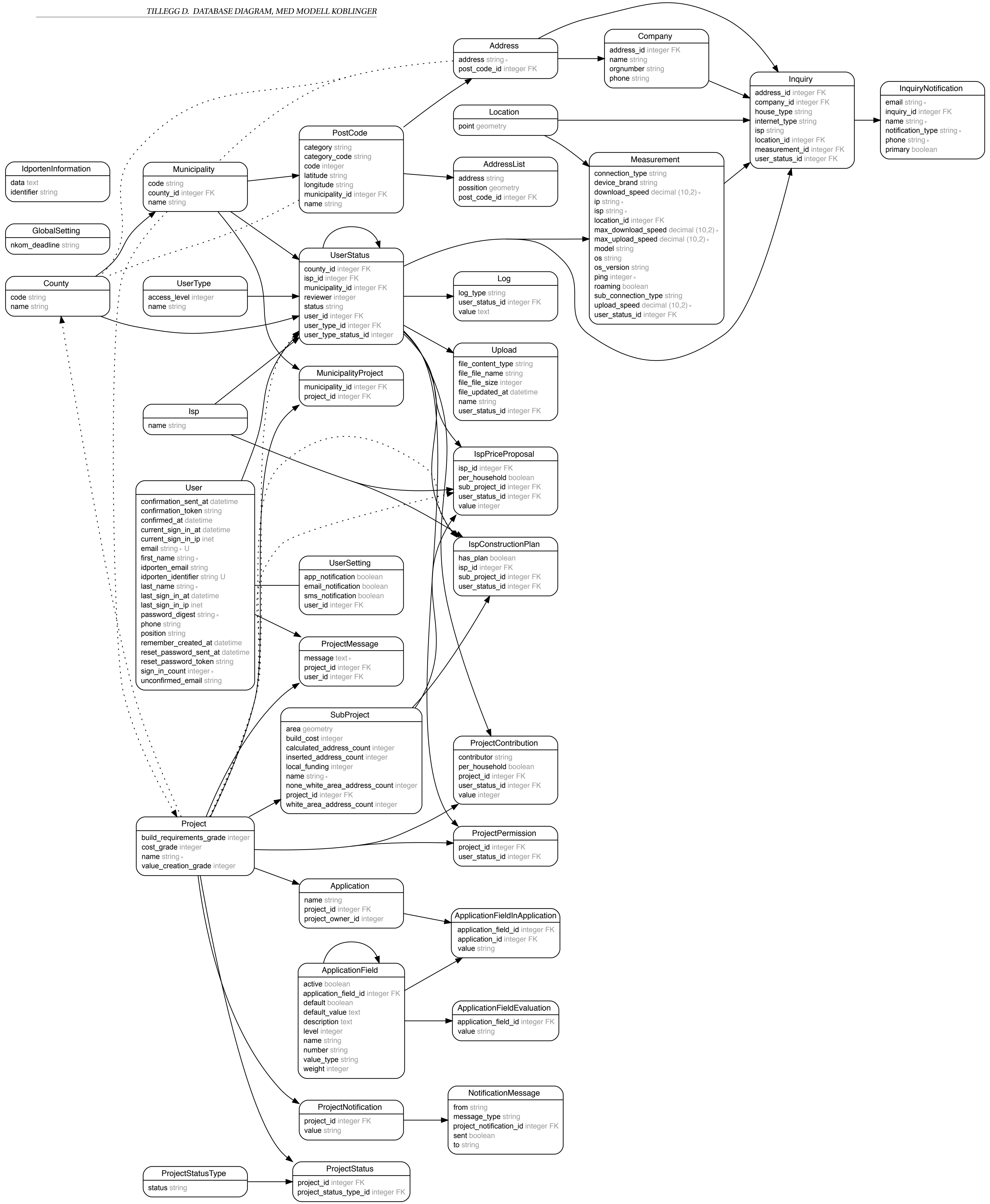
Fargekombinasjoner

De sikreste fargekombinasjoner er å velge to fra samme fargeskala (lys og klar eller mørk). For å skape større oppmerksomhet, er det lov å benytte skarpe farger mot mørk bunn, eller å kombinere fra forskjellige fargeskalaer. I sistnevnte tilfelle er det viktig å kunne fargelære, samt å huske at barkespaden skal komme tydelig fram. Bruk altså ikke to lyse farger eller to mørke farger mot hverandre.



VEDLEGG D

Database Diagram, med modell koblinger



VEDLEGG E

Database Diagram, med uten modell koblinger

VEDLEGG F

Vurdering av å bytte til PHP som backend-språk

Fordeler ved å bytte:

- IT-avdelingen har mulighet til å bidra senere, og under utviklingen.
- Ferdig kode som vi kan bruke på minID-innlogging.
- Flere av oss har erfaring med PHP-språket.
- PHP er raskere (men rammeverkene er ikke så raske).
- Det er laget flere biblioteker i PHP (ikke sikkert vi får så mye bruk for det i dette prosjektet).

Ulemper ved å bytte:

- Må lære oss et test-rammeverk til. Må ha et test-rammeverk for backend og et for frontend.
- Det finnes mer dokumentasjon på Rails enn på PHP-rammeverk.
- Ingen av oss har erfaring med PHP-rammeverk.
- IT-avdelingen har ingen erfaring med PHP-rammeverk.
- Vi kan levere mindre funksjonalitet.
- Vi har jobbet noen timer forgjeves - ting må løses på nytt.
- PHP er tyngre å jobbe med.
- Må sette opp server på nytt.
- Vanskeligere å sette opp bakgrunnsjobber i PHP enn ruby.
- Mange innebygde funksjoner i ruby - det går raskt å programmere.
- Lasse har allerede mye kode i ruby.

Andre ting:

- Byttet påvirker ikke karakteren vår.

VEDLEGG G

Prosjektavtale

PROSJEKTAVTALE

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

Hedmark fylkeskommune

_____ (oppdragsgiver), og

Ingvild Næss, Julie Kristin Thowsen Pettersen,

Petter Uppheim Grøttheim og

Lasse Sviland

_____ (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 2017-01-12 til 2017-05-16 .

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 opptrer 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): Frode Haug

Oppdragsgivers kontaktperson (navn): Birgit Aasgaard Jenssen

Student(er) (signatur):	<u>Ingvild Nass</u>	dato	<u>25.01.17</u>
	<u>Kristin Pettersen</u>	dato	<u>25.01.17</u>
	<u>Petter Grøttheim</u>	dato	<u>27.01.17</u>
	<u>Læsse Suiland</u>	dato	<u>25.01.17</u>

Oppdragsgiver (signatur): Birgit Fosgaard Jensen dato 24-1-2017

Signert avtale leveres digitalt i Fronter(IMT3912)

Godkjennes digitalt av AIMTs dekan

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

Plass for evt sign:

AIMT Dekan/prodekan (signatur): _____ dato _____

VEDLEGG H

Forprosjektrapport

Forprosjektrapport

Lasse Sviland, Ingvild Næss,

Petter Grøttheim, Julie Kristin Thowsen Pettersen

Januar 2017

Innhold

1	Mål og rammer	4
1.1	Bakgrunn	4
1.2	Prosjekt mål	5
1.2.1	Resultat mål	5
1.2.2	Effekt mål	5
1.3	Rammer	6
1.3.1	Tekniske rammer	6
1.3.2	Praktiske rammer	6
2	Omfang	7
2.1	Fagområde	7
2.2	Avgrensning	8
2.3	Oppgavebeskrivelse	8
2.3.1	Roller	8
2.3.2	Hastighetsmåling	10
2.3.3	Login og registrering	10
2.3.4	Varmekart	11
2.3.5	Interesse-varmekart	11
2.3.6	Husstandskart	11
2.3.7	Prosjekt	11
2.3.8	Søknad	12
2.3.9	Kommunikasjon	13

3	Prosjektorganisering	14
3.1	Organisasjonskart	14
3.2	Rollebeskrivelser	14
3.3	Gruppregler	15
4	Systemutviklingsmodell	18
4.1	Valg av modell	18
4.2	Hvordan anvende modellen	19
5	Organisering av kvalitetssikring	20
5.1	Dokumentasjon, standardbruk og kildekode	20
5.1.1	Latex	20
5.1.2	Kodestandard og dokumentasjon	20
5.1.3	Tester	21
5.2	Konfigurasjonsstyring	21
5.2.1	Issue Tracking og Bug Tracking	21
5.2.2	Versjonskontroll av kildekode	21
5.2.3	Samskriving	22
5.2.4	Andre filer	22
5.3	Risikoanalyse	22
5.3.1	Identifisere og analysere prosjektrisikoer	22
5.3.2	Plan for håndtering av de viktigste risikoene	24
6	Plan for gjennomføring	26
6.1	Gantt-skjema	26

1 *Mål og rammer*

1.1 **Bakgrunn**

Hedmark fylkeskommune har fått tilbakemeldinger fra både næringslivet og private om at bredbånd og bredbåndsdekning i fylket er viktig for innbyggerne, bedrifter og fremtiden. I dag ligger Hedmark fylke et stykke etter landsgjennomsnittet når det gjelder høyhastighetsbredbånd, og de ligger i bunnsjiktet i en oversikt over bredbåndsutbyggingen i Norge. Selv om det er bredbåndsdekning de aller fleste steder i fylket, så er denne dekningen altfor lav og ustabil. Dette gjør det vanskelig for fylket og kommunene å reklamere for at Hedmark er et attraktivt sted å bo og jobbe. Det vil være vanskelig å friste folk til å flytte til eller opprette bedrifter i Hedmark hvis den digitale infrastrukturen ikke kan sammenlignes med den de er vant med i større byer. Derfor har Hedmark fylkeskommune satt i gang et prosjekt for å bedre dekningen og kunne gi høyere bredbåndshastighet til befolkningen i hele fylket [1, 2].

Nasjonal kommunikasjonsmyndighet (Nkom) har blant annet tilsyn med operatører og internettleverandører i Norge. De vil forplikte leverandører til å levere et grunnleggende bredbåndstilbud til hele befolkningen i Norge, uansett hvor du bor. Nkom definerer grunnleggende bredbånd som minst 4 Mbit/s i nedlastingshastighet. Og de ønsker at flest mulig skal kunne få tilbud om høyhastighetsbredbånd, som er minst 30 Mbit/s i nedlastingshastighet. I dag opplever fylkeskommunen at kartleggingen av den nåværende situasjonen av bredbåndstilbudet er utfordrende, siden fylkeskommunen og kommunene ikke har tilgang til kartverk som viser hvor det er fiber eller andre typer infrastrukturer. De kan kun basere seg på de teoretiske dekningskartene fra Nkom. Ut fra disse dekningskartene kan kommunene enkeltvis søke om utbyggingsmidler fra Nkom. Disse gir støtte til utbyggingsprosjekter der det er behov for grunnleggende

bredbånd, øke kapasiteten på eksisterende bredbånd, eller hvis området ikke kommer til å bli bygd ut kommersielt de neste årene. Hvis kommunene og fylket har et bedre dekningskart, vil de kunne lage mer nøyaktige søknader til Nkom, og med det få mer tilskudd for utbygging i sine områder [3].

Hedmark fylkeskommune ønsker derfor å få en mer nøyaktig oversikt over hvordan dagens situasjon er. Dette vil gjøre at det kan jobbes for en mer målrettet og systematisk utbygging av bredbåndsnettet, slik at den treffer der det gir størst effekt. For å oppnå dette ønsker de et tettere samarbeid mellom kommunene, lokale krefter og involvering fra innbyggerne. De ønsker å få førstehåndsinformasjon fra innbyggerne om hvilke nettverkshastigheter som finnes, slik at det kan opprettes et nøyaktig dekningskart og oversikt over standarden på bredbåndstilbudet.

1.2 Prosjektmål

1.2.1 Resultatmål

Hedmark fylkeskommune ønsker å utvikle apper for iOS og Android, i tillegg til en nettside, som innbyggerne kan bruke for å måle sin internetthastighet. Innbyggerne skal også kunne registrere om de ønsker raskere internett. I tillegg til å registrere hastigheten sin, skal brukerne kunne se et varmekart som skal vise en grafisk fremstilling av dekningsgraden i fylket. Fylkeskommunen og kommunene skal også kunne bruke nettsiden til å se statistikk og varmekart over nettverkstilbudet og dekningsgraden i de forskjellige områdene i fylket. De skal også kunne se en oversikt over hvem som har meldt sin interesse for bedre bredbånd. På nettsiden skal kommunene også kunne bruke informasjonen som er innhentet til å generere og skrive videre på søknader til Nkom.

1.2.2 Effektmål

Hedmark fylkeskommune ønsker at dette systemet skal hjelpe kommunene i fylket til å få bedre oversikt over bredbåndstilbudet i sine områder. Dette vil la kommunene skrive søknader med mer nøyaktige tall angående tilstanden av nåværende utbyggelse og kundegrunnlag. Mer nøyaktige søknader fra kommunene vil føre til en økning av offentlige tilskudd fra Nkom, fra ca. 10 millioner i året, til 25-30 millioner. Dette vil bety en investering på 200-300 millioner fra bred-

båndleverandørens side, som vil være en økning av utbyggingen med 300%.

1.3 Rammer

1.3.1 Tekniske rammer

- Applikasjonsstøtte for Android og iOS
- Støtte for Internet Explorer, Firefox, Safari og Chrome

1.3.2 Praktiske rammer

- Gruppen har ikke tilgang til noen fast reservert grupperom, men bruker universitetets system for reservering av grupperom for å ha en ganske sikker tilgang på arbeidsplass.
- Tidsperioden for dette prosjektet er fra 16.januar til 16.mai. Vi har som mål å bruke til sammen 120 timer pr. uke fordelt på 4 studenter. Vi har ikke mulighet til å jobbe fulltid med dette prosjektet ettersom vi tar andre emner ved siden av.

2 Omfang

2.1 Fagområde

I dagens Norge er vi avhengige av å ha en internettilkobling. Internett har gjennom de drøyt siste 20 årene hatt en eksplosiv utvikling og erstatter stadig flere og flere oppgaver i det daglige liv. Bruken av internett øker samtidig som bruken av papiraviser, ukeblader og fjernsyn synker[4]. Internett har blitt hovedkilden for informasjon, service, tjenester og kommunikasjon.

Internett er ikke noe som kommer av seg selv, men krever utbygging av bredbånd. Tall fra SSB[5] viser at 84% av norske husstander har fast bredbånd. Dette betyr at hele 16% av husstandene i Norge mangler eller har under 4Mbit nedlastningshastighet.

Det finnes mange forskjellige typer internett-tilkoblinger. De vanligste internett-tilkoblingene er analog, ISDN, ADSL, VDSL, parabol, coax, radio, fiber og 4G. Tilkoblingene gir forskjellig hastighet og tilbys ikke overalt. Dette gjør at enkelte deler av landet har et generelt bra tilbud, mens andre deler av landet ikke har det.

Å bedre bredbåndstilbudet er kostbart og krever at flere interessenter samarbeider. Da vil fylkeskommunen, kommunene og internettleverandører være involvert. Samtidig kreves det bidrag fra innbyggerne i kommunen. Bredbåndsløse leverandører deler ikke sine kartdata om sin egen infrastruktur, noe som gjør det vanskelig for de andre interessentene å få et reelt bilde av dagens situasjon. Leverandørene på den andre siden har ikke kontroll på hvor det er ønskelig å bygge ut bedre bredbånd.

Det er her løsningen vår kommer inn i bildet. Vi skal være med på å utvikle en plattform som bedrer kommunikasjonen mellom interessentene og som viser hvilke steder det er behov for utbygging.

2.2 Avgrensning

Vår oppgave omhandler å utvikle et verktøy for kommunene i Hedmark som kal brukes av kommunene. Dette skal gjøre det enklere for kommunene å bygge ut bredbånd i sitt område. Dette skal gjøres ved å samle inn mer detaljert og korrekt informasjon enn det som allerede finnes i dag angående nettverkshastighet. Dette vil kunne hjelpe kommunene å få en bedre oversikt over hvilke områder i kommunen det burde søkes om støtte til utbygging.

For å hjelpe kommunene med å få mer støtte fra Nkom, skal vi utvikle en portal som vil hjelpe kommunen med å lage bedre søknader. Dette skal vi gjøre ved å bidra med ytterligere informasjon angående dagens situasjon, og å kunne gi bedre tilbakemeldinger på hvor optimalt et prosjekt innfrir kravene Nkom stiller for å innvilge støtte for utbygging.

2.3 Oppgavebeskrivelse

Oppgaven er delt i flere deler. En åpen del for alle innbyggere, som kan brukes til å måle brukerens hastighet. En del der brukere kan logge inn for å få mer informasjon. Samt en del hvor kommuner, fylket og leverandører skal kunne legge inn informasjon og hente ut informasjon som trengs for å generere søknader om støtte for utbygning av bredbånd.

2.3.1 Roller

2.3.1.1 Leverandør

Leverandørene skal kunne se hvilke prosjekter kommunene har opprettet og ha mulighet til å legge inn deres sjablongvurdering (hva de regner med at det vil koste per husstand for å bygge ut i området). De skal kunne oppgi informasjon om hvilke planer de har for utbygging i dette området de neste tre årene, og hvor mye de kommer til å bidra med dersom de blir å bygge ut i et område.

Leverandørene skal også ha informasjon om alle målinger og forespørsler om bedre bredbånd. Dette kan hjelpe leverandørene med å finne ut hva kostnaden ved å bygge ut vil være i et bestemt område, og hvor mye de ønsker å bidra.

2.3.1.2 Kommune

Kommuner skal kunne opprette et prosjekt i et område hvor det ønskes å søke om stønad for utbygging. Dette gjøres ved hjelp av et kart der hvor de skal kunne markere områder hvor det ønskes å bygge ut bredbånd. Markeringen skal kunne ha deler som ikke er en del av markeringen, slik at områder som allerede har bra nettverkstilkobling kan fjernes fra området.

Kommunen kan også legge inn hvor mye de ønsker å støtte prosjektet med per husstand, og se hva leverandør og fylkeskommune har sagt at de skal bidra med. Kommunen skal kunne legge inn informasjon på vegne av en leverandør, dette vil være informasjon om en leverandørs sjablong verdi, og dokumentasjon på om en leverandør kommer til å bygge ut i området.

2.3.1.3 Fylkeskommune

Fylkeskommunen kan legge inn informasjon på et prosjekt om hvor mye de ønsker å støtte prosjektet med.

Fylkeskommunen vil også måtte godkjenne de kommunene som registrerer seg for å benytte seg av verktøyet. De skal også kunne legge inn eller oppdatere informasjon på vegne av en kommune eller en leverandør.

2.3.1.4 Ildsjel

En ildsjel er en person som har bidratt spesielt til et prosjekt i et område og derfor skal ha tilgang til ytterligere informasjon. Ildsjelen kan ofte ha mer detaljert informasjon om området og de som bor der slik at deres bidrag ofte kan være til stor hjelp i et prosjekt.

De skal ha mulighet til å legge til meldinger på et prosjekt slik at de på en bedre måte kan informere og kommunisere med kommunen. De vil ha mulighet til å legge inn hvor mye de vil spare per husstand eller totalt ved dugnad, og legge inn bedrifter i området som ønsker bedre bredbånd. Bedrifter skal legges inn med navn, organisasjonsnummer, telefonnummer og adresse. De skal kunne se status og informasjon om prosjektet, men skal ikke kunne legge inn eller endre informasjon på vegne av de andre aktørene. De vil heller ikke ha mulighet til å se sjablong verdiene fra leverandørene.

2.3.1.5 Innbygger

Prosjektene må utlyses offentlig en måned før det sendes søknad til Nkom. Fra dette tidspunktet, eller om kommune/fylke gjør dette tidligere, vil alle innbyggere kunne se all informasjon som kan offentliggjøres rundt prosjektet. Innbyggeren kan også se alle meldinger som er sendt til de som har registrert at de ønsker varsler fra området.

2.3.2 Hastighetsmåling

En innbygger skal kunne bruke en app eller nettside til å melde interesse om bedre bredbånd. Når hastigheten er målt skal de få mulighet til å forespørre bedre bredbånd. Etter de har gjort dette må man oppgi telefonnummer og epost for å knyttes til forespørselen. De må også oppgi hvilken måte de ønsker å bli kontaktet for statusmeldinger, dersom det blir utbygging av bredbånd i området. De vil ha mulighet til å bli kontaktet på sms, epost eller varslinger i appen.

Brukerens GPS posisjon vil også bli lagret, og brukeren vil bli spurt om å angi riktig adresse. Det skal foreslåes en adresse i nærheten av brukeren posisjon som brukeren kan eventuelt korrigere. Brukeren vil også ha mulighet til å oppgi at dette er en arbeidsplass.

Dersom en bruker forespør nettverk i et område der det allerede er utbygget bedre bredbånd fra andre leverandører skal brukeren få informasjon om hvilke leverandører som tilbyr dette.

2.3.3 Login og registrering

Vanlige innbyggere skal kunne registrere seg på nettsiden eller appen med epost og passord. Her skal de få tilgang til informasjon om prosjekter i området der de har gjort en måling, og informere om at det ikke lenger ønskes bedre bredbånd.

Kommuner og leverandører skal kunne logge inn og registre ytterligere informasjon om deres stilling og kontaktinformasjon. En representant fra fylkeskommunen eller en annen godkjent bruker i samme kommune vil måtte godkjenne brukeren før de får tilgang til kommunens informasjon. Leverandøren må godkjennes av fylkeskommunen før de får tilgang til å se og legge inn tilbud.

2.3.4 Varmekart

På nettsiden og appen skal det være et varmekart som viser hvordan bredbåndstilknytningen er i fylket. Dette skal kunne brukes til å finne andre leverandører i et område som tilbyr bedre nett, eller se hvilke områder det vil være viktigere å bygge ut. Det vil være muligheter for å sortere frem mer spesifikke målinger, slik at man bare viser målinger basert på enkelte teknologier eller leverandører.

2.3.5 Interesse-varmekart

Dette vil være et varmekart basert på hvem som har sagt at de ønsker bedre nett. Dette vil være en enkel måte å kunne se hvilke områder det er mange som ønsker bedre nett og det er dårlig nett i dag. Dette kartet skal være tilgjengelig for fylkeskommune, kommune, leverandør og ildsjel.

2.3.6 Husstandskart

Kommunen skal kunne generere en markering på et kart som skal kunne brukes i et prosjekt. Dette området skal kunne inneholde mindre områder der det ikke ønskes utbygging, dermed vil ikke områdene være fastsatte former. På denne måten vil kommunen kunne kartlegge spesifikke områder som skal gjennomgå en utbyggelse.

2.3.7 Prosjekt

Kommunen eller fylkeskommunen skal kunne opprette et prosjekt for utbygging av bredbånd. Dette området skal spesifiseres ved hjelp av et husstandskart. Systemet skal kunne hente ut informasjon om husstander i området, og eventuelt målinger som er utført. Det skal fortløpende oppdateres med mer informasjon dersom innbyggere bruker appen eller nettsiden til å gjøre målinger eller forespørre bedre nett i området.

Et slikt prosjekt skal kunne offentliggjøres av kommunen eller fylkeskommunen på et hvilket som helst tidspunkt. Det skal også være støtte for å offentliggjøre et prosjekt på et bestemt tidspunkt slik at man er sikker på å overholde eventuelle frister fra Nkom.

2.3.8 Søknad

Kommunene og fylkeskommunen skal kunne generere et utkast til en søknad til Nkom for et av områdene. Søknaden skal inneholde informasjon om hvor mange husstander det er i området. Hva slags hastigheter appen og nettsiden har fått med hastighetsmålinger og hvor mange i området som allerede har fremmet et ønske om bedre bredbånd.

Alle prosjekter skal til enhver tid ha en status. Tilgangen til hvem som kan se hvilken informasjon om en søknad skal i noen grad kunne oppdatere seg ut ifra statusen. Under planleggings fasen vil i utgangspunktet bare kommunen som har opprettet prosjektet og Hedmark fylkeskommune ha tilgang. Når prosjektet lyses ut til innbyggerene vil all offentlig informasjon angående prosjektet være offentlig tilgjengelig på nettsiden. Når søknaden er sendt til Nkom skal det tas vare på søknaden slik den var da den ble sendt, slik at denne alltid vil være tilgjengelig dersom den endres på et senere tidspunkt. Hedmark fylkeskommune skal kunne støtte et prosjekt dersom det ikke har fått godkjent søknad fra Nkom. Kommunene vil også kunne bruke verktøyet fullt ut uten å sende inn noen søknad for situasjoner der de ikke ønsker støtte fra Nkom eller Hedmark fylkeskommune.

Det skal være støtte for at kostnaden skal kunne reduseres dersom innbyggerene ønsker å hjelpe til selv i noen grad, dette kan for eksempel være at alle utfører graving på egen tomt som kan legges inn av kommune, fylkeskommune eller en ildsjel.

På et hvilket som helst tidspunkt vil kommunen har mulighet til å gjøre prosjektet tilgjengelig for alle andre kommuner i fylket, dersom de ønsker innspill eller tilbakemeldinger fra andre kommuner. Denne tilgangen kan på et hvilket som helst tidspunkt trekkes tilbake.

Det skal finnes diverse informasjon om kostnader, det vil si hva det vil koste for leverandøren å bygge ut, om det vil være noen bidrag fra innbyggerene og hva slags støtte kommunen og fylkeskommunen ønsker å bidra med. Dersom prosjektet har fått tilstrekkelig finansiering skal det sendes anbud til de aktuelle leverandørene. På det tidspunktet vil leverandøren få full tilgang til å se prosjektet og bruke systemet til å sende meldinger til innbyggerene.

2.3.9 Kommunikasjon

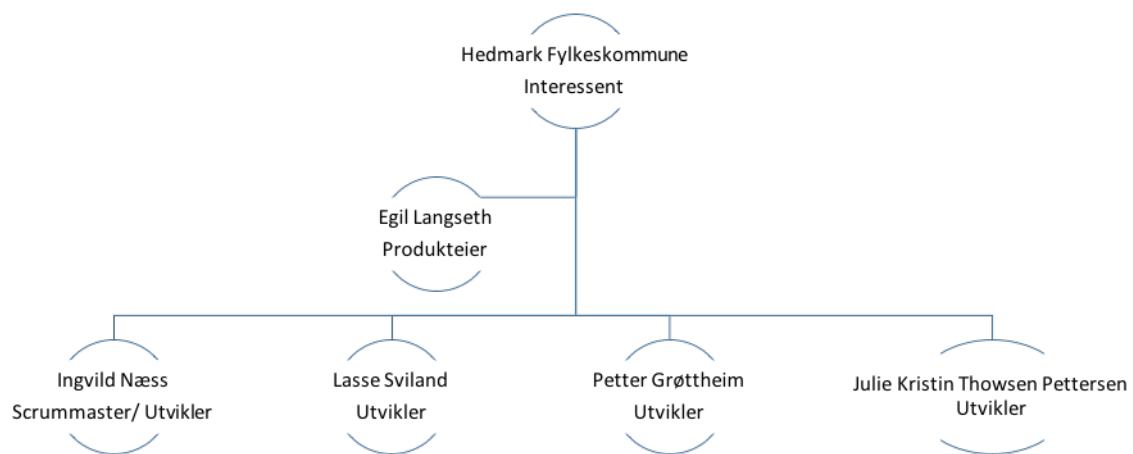
Kommunen, fylkeskommunen eller en leverandør skal kunne informere brukere, som har oppgitt at de ønsker varslinger dersom det er en endring på et prosjekt.

Det skal kunne være kommunikasjon helt fra det opprettes et prosjekt og frem til det er ferdig utbygget. Det skal sendes ut informasjon om søknaden og dens status, men det skal også kunne sendes informasjon fra leverandører under utbyggingsfasen.

Informasjonen sendes på sms, epost eller som varslinger i appen, men alle skal også kunne se dette på nettsiden.

3 *Prosjektorganisering*

3.1 Organisasjonskart



Figur 3.1: Organisasjonskart

3.2 Rollebeskrivelser

Scrumteamet er organisert etter vanlige scrumroller: Produkteier, scrum master og utviklings-teamet.

Egil Langseth, leder for bredbåndsprosjektet i Hedmark fylkeskommune, er valgt til å være produkteier for dette prosjektet. Han har vært ansatt som leder av bredbåndsprosjektet siden

mai 2016, og har derfor god kjennskap til hvilke utfordringer Hedmark fylke står ovenfor og hva slags produkt de ønsker fra oss. Egils hovedrolle er å være bindeleddet mellom oss og de andre interessentene hos Hedmark fylkeskommune, representere deres visjon og hjelpe oss å prioritere våre arbeidsoppgaver i løpet av prosjektet.

Ingvild Næss er scrum master for dette prosjektet. Hennes rolle er å ha et overordnet blikk over hele prosjektet slik at tidsfrister, fremdrift og møter blir holdt. Hun vil også sørge for at utviklingsteamet jobber effektivt slik at alle arbeidsoppgaver under hver sprint blir gjennomført. Ingvild vil ha en delt rolle i dette prosjektet. Mens hun er scrum master, vil hun også være en del av utviklingsteamet.

Utviklingsteamet består av Julie Kristin Thowsen Pettersen, Lasse Sviland, Petter Grøttheim og Ingvild Næss. Utviklerne har ansvaret for å levere ferdige oppgaver når hver sprint er over. Det blir fordelt oppgaver mellom utviklerne på begynnelsen av hver sprint. Alle medlemmene av utviklingsteamet hjelper hverandre for å sikre at en sprint blir vellykket og ferdigstilt.

3.3 Grupperegler

Generelt

- Beslutninger fattes i flertall. Gruppen består av et partall antall personer så det kan oppstå situasjoner hvor beslutninger ikke kan tas ved hjelp av avstemming. Hvis det oppstår to mot to stemmer, og gruppen ikke kommer til enighet, så er det enten veileder eller oppdragsgiver som tar den endelige beslutningen.
- Gruppemedlemmer plikter å si fra tidlig hvis en ikke er fornøyd med arbeidsinnsatsen til de andre gruppemedlemmene.
- Gruppemedlemmer plikter å si fra så tidlig som mulig om en merker at tidsfrister ikke kan overholdes.
- Ved alvorlig brudd på gruppereglene:
 - Samtale mellom alle gruppemedlemmene om problemet.

- Hvis problemet fortsetter skal det aktuelle gruppemedlemmet få en skriftlig advarsel der:
 - * Regelbrudd blir påpekt.
 - * Konkrete tiltak som kan bøte på skaden blir gitt. Dette kan være forventet arbeid, forventet arbeidsinnsats og tidsfrister for det arbeidet.
 - * Konkrete konsekvenser hvis disse tiltakene ikke blir gjennomført.
- Hvis de konkrete tiltakene fra den skriftlige advarselen ikke blir oppfylt blir det samtale med alle gruppemedlemmene og veileder.
- Til slutt blir det eventuell skriftlig ekskludering fra gruppen. Et gruppemedlem kan ikke bli ekskludert fra gruppen de siste 10 dagene før innleveringsfristen.

Møter

- Møte med oppdragsgiver avholdes hver mandag, noen uker tirsdag.
- Møte med veileder avholdes hver tirsdag.
- Prosjektmøte innad i gruppen hver mandag. På disse møtene skal det være gjennomgang av oppgaver vi fordelte på forrige møte, og fordeling av nye oppgaver for kommende uke.
- Det skal skrives et kort referat etter hvert møte med oppdragsgiver, veileder og prosjektmøter. Dette innebærer blant annet fattede avgjørelser, utført arbeid, og arbeid som skal utføres til neste møte.

Prosjektleder

- Prosjektleder er Ingvild Næss, og stillingen går ikke på rundgang.
- Prosjektleders ansvar er å kalle inn til møter, skrive møtoreferater, og ha kontroll på tidsfrister og fremdrift.

Gruppemedlemmer

- Alle må forholde seg til ulike tidsfrister og innleveringer. Dette gjelder både innleveringer innad i gruppen og mer formelle innleveringer til skolen og oppdragsgiver.

- Alle fører egne arbeidstimer kontinuerlig gjennom prosjektet. Det forventes at hvert gruppe-medlem jobber minst 26 timer i uken.
- Alle møter til avtalte tider. Det er møteplikt på møter med oppdragsgiver, veileder og pro-sjektmøter.
- Fravær skal meldes tre dager i forveien. Ved akutt fravær, sykdom og lignende, skal dette meldes til resten av gruppa som fort som mulig.
- Gruppemedlemmer forpliktes til å gjøre sitt beste for å sikre at fastsatte arbeidsoppgaver blir utført til avtalt tid, selv ved planlagt fravær.
- Det pliktes å tenke på rapporten fra dag 1. Det skal dokumenteres ulike funn, kilder og store endringer vi gjør gjennom hele prosjektet.
- Alle skal forholde seg til kodestandardene vi utarbeider.

4 *Systemutviklingsmodell*

4.1 **Valg av modell**

Vi har valgt å jobbe etter systemutviklingsmodellen Scrum. Scrum er en fleksibel utviklingsmodell som åpner opp muligheten for å legge til mer arbeid etter hvert. Som en gruppe på 4 medlemmer er det alltid en risiko at prosjektet kan bli for lite. Muligheten til å legge til mer arbeid etter hvert kan hindre denne situasjonen i å oppstå. Samtidig fastslår den skriftlige oppgavebeskrivelsen fra oppdragsgiver at det er mulighet for utvidelser utenfor det obligatoriske kravet som stilles til produktet.

Sprint Planning Meeting og Sprint Review Meeting er med på å gi oss nær kontakt med oppdragsgiveren. Dermed får vi stadig tilbakemelding på om vi arbeider i riktig retning, og kan raskt ordne opp i eventuelle misforståelser. På denne måten sikrer vi oss å gi oppdragsgiver det produktet han ønsker. Dette er både en fordel for oss og oppdragsgiver.

Aspektet med å ha sprinter kan være med på å holde motivasjonen og fokuset i gruppen oppe. Sprintene gjør at vi alltid har et klart mål å jobbe opp mot. Vi vil alltid vite hva å gjøre i løpet av en uke slik at vi slipper å bli distraheret av fremtidige oppgaver. Dette er en fordel da oppgaven er omfattende og det er lett å ta tak i for mye på en gang.

Scrum tilrettelegger for at utviklerteamet skal ha mye kommunikasjon seg imellom. Dette gjør det lettere for oss å oppdage tidlig hvis arbeidet vi utfører skaper konflikt med andre sitt arbeid. På denne måten sparer vi tid.

4.2 Hvordan anvende modellen

I prosjektet kommer vi til å ta i bruk følgende elementer fra Scrum: Scrum Master, Product Backlog, Sprinter, Sprint Planning Meeting, Sprint Review Meeting og Daily Scrum Meeting. Product Backloggen kommer til å bli lagd i starten av utviklingsfasen og vil vokse seg til etter hvert som flere krav avdekkes.

Vi har fastslått at vi skal ha sprinter som varer i 7 dager. Sprinten starter på mandag og avsluttes på søndag. I starten av hver sprint har vi møte med produkteieren. Dette møtet blir av praktiske årsaker en blanding av Sprint Review Meeting og Sprint Planning Meeting. På møtet får vi tilbakemeldinger på forrige sprint og vi diskuterer hvilke oppgaver i product backloggen som skal prioriteres i den nye sprinten. Oppgavene blir så estimert og fordelt.

Sprinten varer i 7 dager, men i praksis er arbeidsdagene våre på prosjektet fra mandag til torsdag. Hver arbeidsdag møtes utviklerteamet for å ha et Daily Scrum Meeting. På møtet diskuterer vi hva vi gjorde dagen før, eventuelle problemer som har oppstått underveis og hva vi planlegger å gjøre denne dagen.

5 *Organisering av kvalitetssikring*

5.1 **Dokumentasjon, standardbruk og kildekode**

5.1.1 **Latex**

Hver gruppe av filer skal ligge i en egen mappe, med eventuelle undermapper. Mappene skal starte med et nummer som representerer hvor i teksten mappen skal legges inn etterfulgt av et beskrivende navn på innholdet i mappen. Alle mapper og tekstfiler skal bruke små bokstaver, og ord skal skilles med understrek. Bokstavene æ, ø, å skal byttes ut med e, o, a. Mappe og filnavn skal helst være korte og beskrivende, og det unngås å bruke unødvendige symboler i navnene.

Alle mapper har sin egen fil som inkluderer alle filer og undermapper, slik at bare den øverste mappen i et hierarki må inkluderes for å inkludere alle undermapper og filer. Alle filer som inneholder tekst skal ha en overskrift for å beskrive filens innhold.

5.1.2 **Kodestandard og dokumentasjon**

Vi kommer til å følge de kodestandarder som blir brukt i de utviklingsmiljøene vi kommer til å bruke. Alle klasser og funksjoner og alle public variabler skal dokumenteres, ved hjelp av de dokumentasjonsverktøy som brukes i de rammeverk vi velger å bruke. I programmeringsspråk der det ikke er en naturlig standard, kommer vi til å bli enige om en mye brukt standard for kommentering.

5.1.3 Tester

Vi kommer til å skrive tester som kan kjøres automatisk for alle klasser og funksjoner som skrives. Det skal også skrives tester for GUI for å sikre at vi alltid er sikker på at det ikke vil være kode som bare fungerer på noen enheter.

5.2 Konfigurasjonsstyring

Under utviklingen kommer det til å være store mengder delte filer og oppgaver som skal utføres på disse filene. For å håndtere filene og endringer gjort i filene skal det brukes forskjellige konfigurasjonsstyringsverktøy. På denne måten kommer filene alltid til å være tilgjengelige for gruppen samtidig som vi får loggført endringer.

5.2.1 Issue Tracking og Bug Tracking

Som nevnt tidligere skal vi bruke utviklingsmodellen Scrum. Dette innebærer ukentlige sprinter hvor vi fordeler oppgaver(issues) på starten av hver sprint. For å samle inn oppgaver og fordele oppgavene skal vi bruke Jira sitt Scrum-board. Dette gir oss kontroll på hva som gjøres og av hvem til enhver tid.

Internt i gruppa vil Jira i tillegg bli benyttet som et verktøy for bug tracking. Medlemmene i gruppa legger selv inn bugs i Jira fortløpende.

5.2.2 Versjonskontroll av kildekode

For å ivareta integriteten til kildekoden skal Git benyttes under utvikling som et versjonskontrollverktøy. Siden vi skal utvikle tre forskjellige utgaver av produktet (iOS, Android og web) har vi besluttet å oppdrette et repository per utgave. Dette skal være med på å forhindre store mappestrukturer og gjøre det lettere for gruppemedlemmene å jobbe på hver sin oppgave.

Repositoryene oppdrettes på Bitbucket som er det verktøyet flertallet i utviklerteamet bruker fra før.

5.2.3 Samskriving

Alt av skriftlige dokumenter som vi skal utarbeide skal skrives i Latex. For å forhindre en stor jobb med å sette sammen dokumenter på slutten skal vi benytte ShareLatex som gir oss muligheten til å skrive sammen i sanntid. Dette gir oss fleksible dokumenter som følger lik stil fra første stund.

5.2.4 Andre filer

I tillegg til kildekode og rapporter som vi selv produserer vil det oppstå andre filer som vi trenger å ha tilgjengelig oss i mellom. Dette kan være notater skrevet fortløpende, filer delt med oss av oppdragsgiver og lignende. Disse filene kommer til å ligge i en felles Google Drive mappe.

5.3 Risikoanalyse

5.3.1 Identifisere og analysere prosjektrisikoe

For dette prosjektet har vi valgt å sette opp en tabell der vi tar for oss hver risiko, ser på sannsynligheten og konsekvensene for den. Vi har valgt å kategorisere sannsynligheten for at en risiko skal inntreffe etter lite sannsynlig, sannsynlig og svært sannsynlig. Konsekvensene av at risikoen inntreffer er kategorisert i uproblematisk, problematisk og kritisk.

Nummer	Risiko	Sannsynlighet	Konsekvens	Tiltak
1	Vi klarer ikke å levere produktet til avtalte tidspunkter fordi en eller flere arbeidsoppgaver tar lenger tid enn estimert.	Sannsynlig	Kritisk	Ja
2	Store endringer i krav fra oppdragsgiver underveis i prosjektet.	Sannsynlig	Problematisk	Ja

3	Løsningen er ikke brukervennlig nok for alle brukergrupper som igjen fører til at løsningen ikke blir brukt.	Sannsynlig	Kritisk	Ja
4	En eller flere av utviklerne blir syke over en lengre periode, eller slutter i løpet av prosjektet.	Lite sannsynlig	Problematisk	Nei
5	Tap av kildekode eller dokumentasjon.	Lite sannsynlig	Kritisk	Ja
6	Konkurrenter kommer med samme ideen og lager en tilsvarende løsning.	Sannsynlig	Uproblematisk	Nei
7	Hedmark Fylkeskommune kan avslutte prosjektet.	Lite sannsynlig	Kritisk	Nei
8	Vi støter på tekniske problemer. Visse funksjonaliteter vil ikke bli implementert fordi det er for vanskelig teknologisk eller ikke mulig å oppnå.	Sannsynlig	Kritisk	Ja
9	Vi klarer ikke å måle bredbåndshastigheten nøyaktig nok.	Sannsynlig	Kritisk	Ja

5.3.2 Plan for håndtering av de viktigste risikoene

Vi har valgt ut noen av risikoene som vi vil presentere noen tiltak for. Dette er tiltak som enten minsker sannsynligheten for at risikoen kan inntreffe, gjøre konsekvensen mindre hvis risikoen skjer, eller begge deler.

Risiko nr	Tiltak
1	For å redusere sannsynligheten for at dette skal skje kan vi sørge for å ikke bli forsinket underveis. Vi kan passe på at alle oppgaver i sprinten gjennomføres. Hvis vi ser at vi ligger bak tidsmessig på en sprint, kan en utvikler jobbe litt overtid den aktuelle uken slik at det blir brukt mer tid på oppgaven. Hvis noen av de andre utviklerne er ferdig med sine oppgaver tidlig, så kan vi sette inn mer arbeidskraft for å løse den aktuelle oppgaven. Vi kan eventuelt også løse oppgaven på en annen måte som tar mindre tid som ikke påvirker det endelige produktet for mye. Hvis vi blir forsinket på slutten av prosjektet så kan vi informere produkteier. Vi kan sammen bli enige om hva som bør tas ut av leveransen.
2	Det er viktig at vi har en god prosess tidlig i prosjektet slik at Hedmark Fylkeskommune kan uttrykke sine behov og ønskede resultater. Vi kan diskutere ideene deres og komme med innspill på hva som er mulig å gjøre på et teknologisk nivå. Gjennom prosjektperioden er det viktig at vi har møter med produkteier ofte slik at vi kan vise og demonstrere løsningen. Ved å ha møte ofte blir det enklere for produkteier å kunne uttrykke nye ideer og hva som er bra eller dårlig med den eksisterende løsningen. Dette gjør det mulig for oss å gjøre endringer raskt.
3	For å sikre at løsningen vår blir brukervennlig nok for alle brukergrupper kan vi gjennomføre 3 runder med brukertester. Vi kan gjennomføre brukertester når vi har implementert ny funksjonalitet i løsningen. Det er viktig at disse brukertestene blir gjennomført på personer med ulik alder og teknisk forståelse slik at eventuelle vanskeligheter blir fanget opp.
5	For å sikre at vi ikke får tap av kildekode eller dokumentasjon vil vi ta sikkerhetskopi av dette hver time. Alle disse lagres i tre dager. For backup som er 3 dager eller eldre tar vi bare vare på en sikkerhetskopi for hver dag.

8	Vi kan tidlig i prosjektet bestemme hvilken teknologi vi skal benytte slik at vi kan begynne å sette oss inn i fagstoff. Hvis vi under prosjektet ser at vi mangler kompetanse til å klare en arbeidsoppgave, så kan vi hente inn hjelp fra ulike lærere ved NTNU i Gjøvik. Mangel på kompetanse kan oppdages tidlig med god planlegging.
9	For å sikre at vår hastighetsmåling er nøyaktig nok så må vi gjennomføre omfattende testing av denne funksjonaliteten på ulike enheter, nettverk og miljøer. Hvis vi ser at målingene ikke er tilfredsstillende etter endt testing må vi eventuelt se etter andre metoder for å måle hastigheten.

6 *Plan for gjennomføring*

6.1 Gantt-skjema

Vi har valgt å dele hele prosjektperioden inn i fire hoveddeler: planlegging, utvikling, fullføring av hovedrapport og forberede muntlig presentasjon.

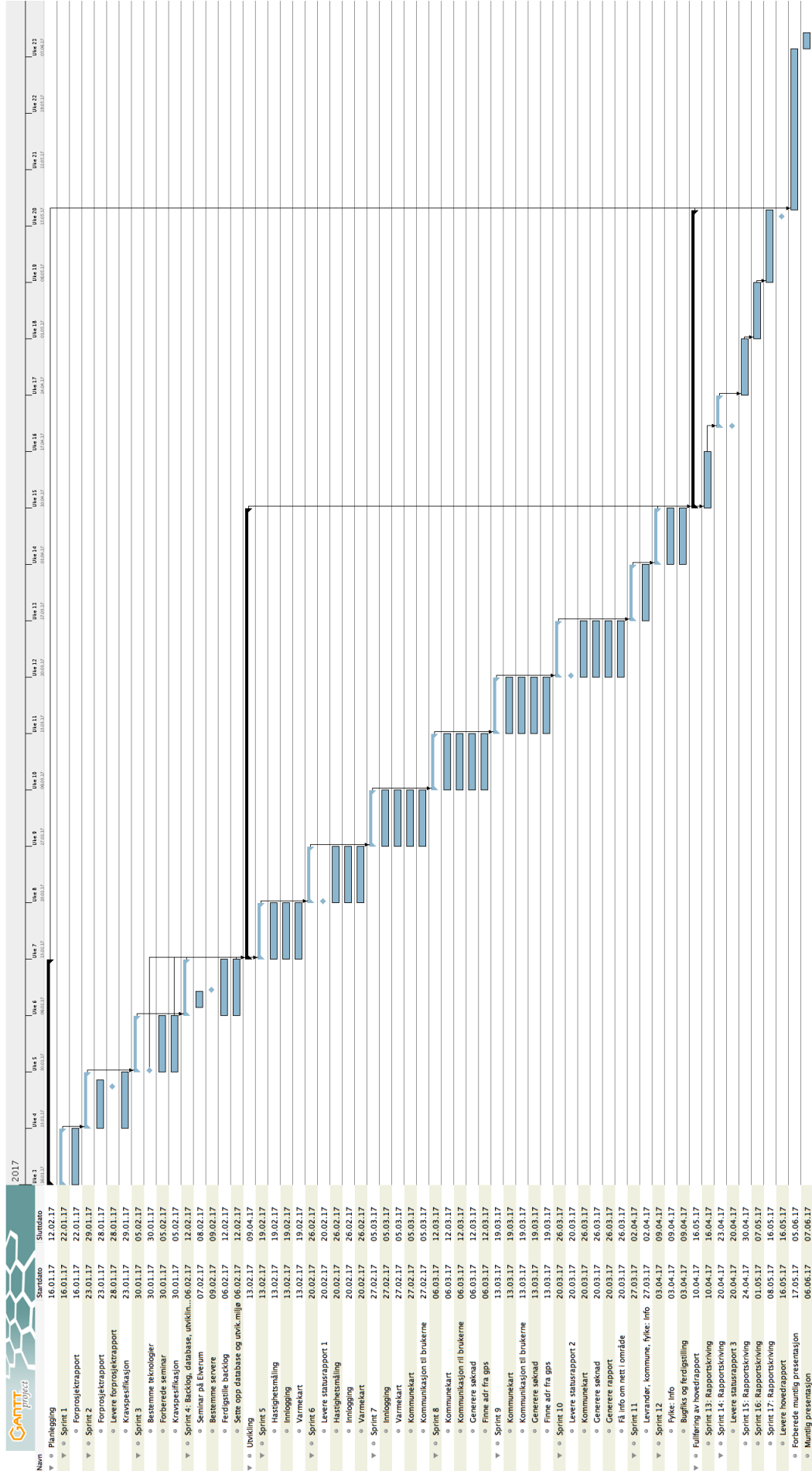
Planleggingsperioden består av fire sprinter og varer fra 16. januar til 12. februar. Under denne perioden er første sprint og deler av andre sprint satt av til å jobbe med forprosjektrapporten som skal leveres 28.januar. Første sprint vil gå med til å skrive rapporten, mens vi vil rette opp og forbedre dette arbeidet i andre sprint etter tilbakemeldinger fra oppdragsgiver og veileder. Mens vi forbedrer og klargjør forprosjektrapporten til innlevering, vil vi også starte på kravspesifikasjonen i sprint 2. Arbeidet med denne vil fortsette ut i sprint 3. Kravspesifikasjonen er et viktig arbeid for å få definert all funksjonalitet, så det er viktig at denne er på plass før vi kan starte med utviklingen. I sprint 3 har vi også satt opp at vi skal forberede oss til seminar på Elverum, som vi skal delta på 7. og 8. februar. Dette er et seminar der representanter fra Hedmark Fylkeskommune, kommunene i Hedmark, Nkom og ulike bredbåndsleverandører vil være tilstede for å snakke om bredbåndsutbyggingen i Hedmark Fylke. På seminaret skal vi presentere vår bacheloroppgave. I sprint 4 vil vi i tillegg til å delta på seminaret ferdigstille backloggen med alle arbeidsoppgaver som skal gjennomføres under utviklingsperioden. Vi vil også sette opp de ulike utviklingsmiljøene og databasene vi trenger.

Utviklingsperioden vil vare fra 13. februar til 9.april, og vil bestå av 8 sprinter. I denne perioden vil vi utvikle all funksjonalitet i systemet, og vi har satt noen datoer for når de viktigste delene skal være ferdig:

- 27. februar: Det skal nå være mulig å gjøre hastighetsmåling på nettsiden og appene.

- 6. mars: Varmekart over hastighetsmålinger og innlogging for ulike roller skal være ferdig. Dette innebærer også registrering av nye brukere.
- 20. mars: Brukere som er registrert som innbyggere kan nå motta meldinger.
- 27. mars: Det skal nå være mulig å generere søknader og rapporter, og kommunene skal kunne opprette områder de ønsker utbygging i, ut fra et kart.
- 3. april: Leverandører og kommunene kan legge inn informasjon.
- 10. april: Nettsiden og appene skal være ferdig utviklet.

Perioden fra 10.april til 16.mai er satt av til å skrive ferdig rapporten vi skal levere inn til vurdering 16.mai. Dette har vi satt av 5 sprinter til. Etter at hovedrapporten er levert inn, har vi fra 17. mai til 5. juni til å forberede den muntlige presentasjonen vi skal ha 6. eller 7. juni. Vi har valgt å ikke dele denne perioden opp i sprinter, siden vi i denne perioden også må prioritere eksamen i andre emner vi har.



Figur 6.1: Gantt-skjema

Bibliografi

- [1] Bedre bredbånd til folket [Web Page]; 2014 [cited 27/01-17]. Available from: <https://www.hedmark.org/Om-fylkeskommunen/Naeringsutvikling/Bredbaandsprosjekt-i-Hedmark/Bedre-bredbaand-til-folket>.
- [2] Gjennomføringen av prosjektet [Web Page]; 2014 [cited 27/01-17]. Available from: <https://www.hedmark.org/Om-fylkeskommunen/Naeringsutvikling/Bredbaandsprosjekt-i-Hedmark/Gjennomfoeringen-av-prosjektet>.
- [3] Stønad til breibandutbygging [Web Page]; 2016 [cited 08/05-17]. Available from: <http://www.nkom.no/spørsmål-og-svar/stønad-til-breibandutbygging>.
- [4] Norsk mediebarometer, 2015 [Web Page]; 2016 [cited 24/01-17]. Available from: <https://www.ssb.no/medie/>.
- [5] Internett-målinga, 3. kvartal 2016 [Web Page]; 2016 [cited 24/01-17]. Available from: <https://www.ssb.no/teknologi-og-innovasjon/statistikker/inet>.

VEDLEGG I

Statusrapporter

I.1 Statusrapport 1 - 20.02.17

I.1.0.1 Fremdrift og tidsfrister

Vi har holdt oss til fremdriftsplanen så langt. Eneste avviket fra fremdriftsplanen er datoen for når oppsett av databasen skulle være ferdig. Opprinnelig skulle det være ferdig 12.02, men vi har valgt å sette 26.02 som ny dato for dette. Ellers har vi overholdt de tidsfristene vi har hatt så langt i prosjektet.

I.1.0.2 Organisering av gruppens arbeid og ansvarsområder

Arbeid og ansvarsområder holdes som i forprosjektrapporten. Alle gruppemedlemmer bidrar til diskusjon og vi tar avgjørelser i fellesskap. Kristin har startet på utviklingen av appen, mens Petter, Lasse og Ingvild har startet på utviklingen av nettsiden.

I.1.0.3 Rapportskriving

Vi har skrevet og levert forprosjektrapporten. Vi har kommet godt i gang med kravspesifikasjonen, men venter med videre arbeid på hovedrapporten til vi er kommet lengre i kodingen. Av annen rapportering skrives det møtereferat etter hvert møte med veileder og oppdragsgiver.

I.1.0.4 Muligheter/Trusler/Problemer?

Det har tatt noe lenger tid enn vi hadde planlagt å sette opp Angular, vårt frontend rammeverk, med Ruby on Rails, som er vårt backend rammeverk. Databasen har også blitt mye større enn vi i utgangspunktet trodde, derfor trengte vi litt ekstra tid til å gjøre denne ferdig. Ellers opplever vi at det går mye tid på å snakke og avklare ulike ting i oppgaven, men vi vet at dette er nødvendig for å ikke gjøre unødvendige feil når vi starter å kode.

I.1.0.5 Hva er avsluttet/Hvilke oppgaver er ferdige?

Vi er ferdig med forprosjektrapporten og deler av kravspesifikasjonen. Vi har også laget en low-fi og high-fi prototype som danner et fint grunnlag for utviklingsarbeidet. Oppsett av app og

andre rammeverk er ferdig. På appen har det blitt laget en enkel fremside og vi har satt opp internasjonaliseringen.

I.1.0.6 Hva er under arbeid?

Denne uken starter vi arbeidet med hastighetsmålingen på både appen og websiden. Vi jobber også med innlogging, registrering av nye brukere, og varmekart. Vi skal jobbe med løsningen som håndterer geografiske data i databasen. Vi skal også skrive tester for alle klassene som representerer en tabell i databasen.

I.1.0.7 Gruppens motivasjon

Gruppen samarbeider godt, og vi diskuterer problemstillinger og problemer som kommer opp underveis. Til tider kan ukjente teknologier være litt frustrerende, men dette går fort over når vi begynner å forstå og se resultater. Ellers er motivasjonen på topp.

I.1.0.8 Kontakt med veileder

Kontakten med veileder Frode Haug har fungert veldig fint. I januar hadde vi ukentlige møter hvor vi oppdaterte veileder på hvordan vi lå an og vi fikk gode konstruktive tilbakemeldinger på forprosjektrapporten. I februar har vi valgt å ikke ha veiledningsmøter. Dette er fordi vi var borte en uke på grunn av seminar på Elverum og vi har akkurat startet med utviklingen. Vi følte vi ikke hadde noen konkrete spørsmål og ting vi ønsket tilbakemelding på under denne prosessen. Vi vil ta opp igjen de ukentlige møtene i slutten av februar.

I.2 Statusrapport 2 - 20.03.17

I.2.0.1 Fremdrift og tidsfrister

Vi ligger en del etter det opprinnelige tidsskjemaet. Vi har brukt lengre tid på noen arbeidsoppgaver enn det vi hadde estimert. Derfor har vi laget en plan for hvordan dette skal løses videre:

- Lage en liste over hvilke funksjonaliteter som skal prioriteres, og hvilke funksjonaliteter som skal nedprioriteres.

- Gjøre ting enklere enn vi i utgangspunktet hadde tenkt i forhold til kodingen.
- Forlenge utviklingsperioden for noen av gruppemedlemmene.
- Gjøre appen raskt ferdig, slik at alle kan jobbe med nettsiden.
- Alle gruppemedlemmer skal jobbe minst 26 timer i uka.
- Skrive færre tester enn vi opprinnelig hadde tenkt. Kutte ut end-to-end tester og tester for appen.

I.2.0.2 Organisering av gruppens arbeid og ansvarsområder

Fordelingen av arbeidet er fortsatt som ved statusrapport 1. Kristin jobber fortsatt med utviklingen av appen, men får av og til litt hjelp av Lasse. Lasse jobber i hovedsak med utviklingen av nettsiden sammen med Petter og Ingvild. Vi har valgt å dele opp arbeidet slik, med flere gruppe-medlemmer som jobber på nettsiden, fordi nettsiden har mer funksjonalitet enn appen. Alle gruppemedlemmer bidrar fortsatt i diskusjoner rundt oppgaven, og alle avgjørelser blir tatt i fellesskap i gruppen eller sammen med oppdragsgiver.

I.2.0.3 Rapportskriving

Vi har ikke skrevet noe på hovedrapporten siden forrige statusrapport. Alle gruppemedlemmer tenker på rapporten gjennom hele prosjektet ved å dokumentere ulike funn, avgjørelser og kilder slik at dette kan dokumenteres i rapporten senere. Av annen rapportering skrives det fortsatt møtereferat etter hvert møte med veileder og oppdragsgiver.

I.2.0.4 Muligheter/Trusler/Problemer?

Vi mistet litt over en halv uke med jobbing da IT-avdelingen ved Hedmark fylkeskommune kom med et ønske om at vi skulle bruke PHP som backend-språk. Dette ønsket kom de med etter at vi hadde startet kodingen, selv om vi hadde spurt spesifikt om de hadde noen ønsker om hvilke programmeringsspråk vi skulle bruke. Etter vi fikk vite dette brukte vi litt tid på å tenke på mulige konsekvenser og eventuelle fordeler ved å bytte språk så sent i prosessen. Vi tok etterhvert

en avgjørelse sammen med oppdragsgiver om at vi skulle fortsette med det programmerings-språket vi allerede hadde valgt. Det er ikke alle gruppemedlemmer som bruker så mye tid på bacheloroppgaven hver uke som vi hadde avtalt i starten av prosjektet. Dette medfører at færre arbeidsoppgaver blir utført i løpet av en uke. Vi har ikke klart å estimere riktig i starten av prosjektet, hvor lang tid hver oppgave kommer til å ta. Dette betyr at vi bruker lenger tid enn planlagt per oppgave og dermed ligger vi bak det opprinnelige tidsskjemaet.

I.2.0.5 Hva er avsluttet/Hvilke oppgaver er ferdige?

På appen er vi ferdig med siden som møter en bruker ved første gangs bruk. Dette er en informasjonsside med en sjekk på om brukeren har internett og at brukeren må godkjenne at appen bruker GPS posisjonen videre. På appen er også test av nedlastningshastighet og ping ferdig. På nettsiden har vi laget fremsiden og innlogging for vanlige brukere. Vi har også gjort ferdig databaseoppsettet og modeller for backend og oppsett av biblioteker for å lage API. Klasser for å sende data frem og tilbake mellom API-et og klientene er også ferdig.

I.2.0.6 Hva er under arbeid?

Denne uken skal vi ferdigstille hastighetsmålingen på nettsiden, og test av opplastningshastigheten på appen. Vi skal jobbe videre med innlogging slik at dette også inkluderer brukere som skal bruke minID som innloggingsmetode. Vi skal denne uken starte å arbeide med de ulike kart-funksjonalitetene. Ellers skal vi jobbe med at en bruker kan forespørre bedre nett, både via appen og nettsiden. Vi skal også fortsette arbeidet med prosjektsiden på nettsiden.

I.2.0.7 Gruppens motivasjon

Som gruppe samarbeider vi fortsatt godt. Det kan til tider være litt frustrerende når koden vår ikke fungerer helt som den skal, eller hvis ting tar lenger tid enn vi trodde. Det er da litt lett å miste motivasjonen for en periode.

I.2.0.8 Kontakt med veileder

Siden forrige statusrapport har vi hatt to korte veiledninger med Frode. På det første møtet snakket vi om eventuelle konsekvenser hvis vi byttet backend-språk. På det andre møtet oppdaterte

vi han på hvordan vi lå an generelt i prosjektet. Vi har ikke følt behov for veiledning nå når vi har så stort fokus på kodingen. Men vi regner med at vi får et større behov for veiledning når vi starter opp igjen med rapportskrivningen.

I.3 Statusrapport 3 - 20.04.17

I.3.0.1 Fremdrift og tidsfrister

Ved forrige statusrapport lå vi litt etter tidsskjemaet. Vi laget da en plan for hvordan vi skulle løse dette videre. Vi har fulgt de fleste av disse punktene. Alle på gruppa har jobbet minst 26 timer i uka, vi har skrevet litt færre tester og løst ting på en enklere måte enn vi i utgangspunktet hadde tenkt. Vi har også utvidet utviklingsperioden med to uker. Vi har rukket de aller fleste arbeidsoppgavene som vi skrev på vår prioriteringsliste ved forrige møte. Men vi har også blitt et gruppemedlem færre siden forrige statusrapport, så det forklarer også hvorfor vi har rukket mindre enn vi hadde planlagt. Vi har nå satt en siste tidsfrist for når utviklingen skal være ferdig, og det er onsdag 26. april.

I.3.0.2 Organisering av gruppens arbeid og ansvarsområder

Siden utviklingen av appen nå er ferdig, så jobber alle gruppemedlemmene på nettsiden. Lasse jobber mest med backend, samt noe frontend. Kristin og Ingvild jobber bare med frontend. Etter at vi valgte å dele opp arbeidsoppgavene på denne måten har utviklingen gått raskere.

I.3.0.3 Rapportskrivning

Forrige uke startet vi litt opp igjen med å skrive på rapporten. Fra neste uke er det bare rapportskrivning frem til innleveringen. Av annen rapportering skrives det fortsatt møtereferat etter hvert møte med veileder og oppdragsgiver.

I.3.0.4 Muligheter/Trusler/Problemer?

Vi ble et gruppemedlem mindre rett før påske. Vi brukte litt tid og energi på denne prosessen, men vi føler det var en nødvendig prosess å ta. Som nevnt tidligere i statusrapporten har utvik-

lingsoppgavene gått raskere etter at vi delte opp oppgavene til hva hvert gruppelem er best på.

I.3.0.5 Hva er avsluttet/Hvilke oppgaver er ferdige?

- Innlogging med minID
- Forespørre bedre nett på nettsiden
- Liste over alle prosjektene i en kommune
- Lage et nytt prosjekt og sub-prosjekt. Markere et område på kartet
- Gi og fjerne tilgang til en kommune til et prosjekt
- Chat-system mellom alle som har tilgang til et prosjekt
- Varslingssystem med epost og sms til personer som har vist interesse innenfor et område

I.3.0.6 Hva er under arbeid?

- Finne antall husstander innenfor et opptegnet område på kartet
- Ferdigstille varmekart
- Fylle ut nkom-søknad, og generere dette til en word-mal
- Liste over alle som har tilgang til et prosjekt
- Vise og endre informasjon om et prosjekt, som f.eks antall husstander og kostnad pr. husstand
- Ferdigstille hastighetsmålingen på nettsiden
- Sette opp serveren hos fylkeskommunen
- Sette opp riktig server-adresse, legge til ikon og bakgrunn på appen.
- Endelig layout og design

I.3.0.7 Gruppens motivasjon

Gruppen jobber veldig godt sammen og motivasjonen for å bli ferdig er på topp. Vi har av og til noen bekymringer for at vi ikke har fått tid til alt vi hadde planer om i januar, og at det kommer til å bli en intens periode med mye jobbing frem til innleveringsdatoen.

I.3.0.8 Kontakt med veileder

Vi har hatt en del møter med veileder den siste måneden på grunn av at den opprinnelige gruppen vår ble splittet i to. Det har vært godt å kunne få hjelp og råd fra Frode om hvordan vi skulle håndtere denne situasjonen. Vi føler Frode har involvert seg på en god måte, og at han virkelig bryr seg om oss som gruppe. Vi regner med at vi kommer til å få mer og mer nytte av Frode nå fremover når vi skal jobbe med rapporten.

VEDLEGG J

Møtereferater

Fredag 13.01.17

Tilstede: Petter, Lasse, Kristin, Ingvild

- Vi har avtalt møte med oppdragsgiver 16.01.17.
- Prosjektavtale ble fylt ut og underskrevet av gruppemedlemmene. Prosjektavtalen tas med til første møte med oppdragsgiver slik at Egil (kontaktpersonen vår) kan skrive under.
- Vi utarbeidet grupperegler. Ingvild skriver disse ferdig. Gruppereglene tas med på første møte med veileder for å få veiledning på disse.
- Ingvild avtaler møte med veileder i uke 3.
- Vi skal bruke google disk for å oppbevare alle felles dokumenter under prosjektet.
- Alle gruppemedlemmene skal føre egne arbeidstimer gjennom prosjektet. Hvert gruppemedlem finner seg et verktøy for å føre timer.
- Rapporten skal skrives på norsk. Vi skal bruke latex.
- Systemutviklingsmodellen vi skal jobbe etter er scrum, med sprinter på 1 uke.
- Vi skal bruke Jira som verktøy for å holde orden på arbeidsoppgaver.
- Vi ønsker møte med oppdragsgiver en gang i uka, dette avtales på første møte 16.01.
- Vi skal utarbeide kodestandarder som alle gruppemedlemmene må forholde seg til når vi begynner kodingen.

Mandag 16.01.17

Tilstede: Egil, Petter, Lasse, Kristin, Ingvild, Frode Hansen fra GIS (Deler av møte), Svein-Inge Kvalø fra IKT (Deler av møte)

- Forventninger fra Egil: At verktøyet kan brukes etter endt prosjekt og at de andre fylkeskommunene kan begynne å bruke dette etterhvert.
- Vi hadde en gjennomgang fra Egil om ulike måter en kunne få bredbånd på og hastighetene til disse: Analog, ISDN, ADSL, VDSL, Parabol, COAX, Radio, Fiber, 4G, Icc.
- Nkom har satt grenser for hva som regnes som tilfredsstillende hastigheter:
 - 4 Mbit < : Hvitt område - ikke dekning
 - 4 - 30 Mbit : Tilfredsstillende
 - <30 Mbit : NGA (next generation access). I år 2020 blir denne grensa økt til 100 Mbit.
- Prosjekteringsverktøyet:
 - App for android og ios, og webside.
 - Får vite hvem sluttbrukeren er med en GPS lokasjon. Knytte den GPS lokasjonen til en husstand(adresse). Hvis det ikke gis riktig husstand, da kan sluttbrukeren hjelpe oss med å få riktig lokasjon. Enten med å skrive inn adressen eller velge et sted på kartet. Sluttbrukeren burde også bekrefte sin egen epost adresse. Slik at det hindrer at det blir registrert feil informasjon og at kommunen/ildsjelen får ekstra arbeid i å rette opp eventuelle slike feil.
 - Rapporter

- Varmekart med hastigheter
- Varmekart over om sluttbrukeren er fornøyd med sin hastighet.
- Hvilke type teknologi som finnes i et område (fiber, radio, 4G osv).
- Kommunisere med sluttbruker via sms, epost, i appen, på nettsiden. Kan få kommunikasjon fra kommunen ut til sluttbrukeren.
- Har naboen bedre bredbånd? Info hvis husstander i område har bedre hastighet. Gi informasjon om hvilke leverandør som tilbyr bedre nett og til hvilken månedspris.
- Hente ut informasjon om hvilke nett de har? Hvilken leverandør og hvilken type teknologi det er.
- Roller i verktøyet: Fylkeskommune, kommune, ildsjel, innbygger, leverandør. De ulike rollene kan se ulike ting når de logger seg på løsningen.
- Sjablong vurdering. Hver kommune kan gå inn å velge et område. Og hver leverandør kan gå inn å registrere hvor mye utbygging av det område kommer til å koste fra leverandørene sin side. Kan registrere kostnader for ulike teknologier?
- Kommunen skal kunne godkjenne ildsjeler. Skal kommunen kunne opprette brukerkontoer?
- En leverandør skal kunne få vite hvem som ønsker bedre bredbånd?
- Trenger også å vite hva slags type bolig sluttbrukeren har? Er det hytte? Vil du fortsatt ha bedre bredbånd?
- Generere søknader og rapporter til nkom automatisk ut fra informasjon i verktøyet.
- Skal nkom kunne logge seg inn og se ulike rapporter og statuser?
- Kart:
 - Hastighet - Varmekart
 - Nærmest tilkoblings sentraler. Kutte ut disse fra kartene?
 - Hvordan skal vi synliggjøre husstandene på kartet? Hvordan skal vi skille mellom husstander som har registrert interesse og de som ikke har det.
- Møte med kommunene 7. og 8. februar på Elverum. Nkom, Telenor og Eidsiva kommer også. Vi skal presentere hva vi har tenkt til å gjøre. Vise demo av appen. Ingen teknisk gjennomgang.
- Hva bør prioriteres? Appen bør prioriteres slik at den kan vises og begynne å bli brukt på seminaret.
- Det bør utarbeides en fremdriftsplan
- Det må sjekkes hva som er lov i forhold til datatilsynet.
- Det burde utvikles slik at andre kan ta over prosjektet når bacheloren er over.
- Generisk view på appen. Skal kunne brukes av andre fylkeskommuner senere. Det kan gjerne bli ulike utseender utfra hvilke fylkeskommune du hører til når du logger inn.
- Kan kontakte Frode om spørsmål om kart. Får mailadr fra Egil.
- Egil skal gis tilgang til google Disk.
- Egil har ferie torsdag 09-17.02. Han er ikke tilgjengelig denne perioden.
- Neste møte: Sprint planning meeting mandag 23. januar

- Statusrapporter: Egil har ingen spesielle ønsker om hvor mange og hvor ofte statusrapporter det skal være. Det kan vi selv bestemme. Dette skal tas opp på møte med veileder.
- Samtale med Svein-Inge fra IKT:
 - Hva vil det koste å bruke en annen leverandør? Hva med hvis løsningen går ned på kveldstid? IKT-avdelingen på fylkeskommunen har bare kontortid på dagtid. Eller hva med de dagene linjene har mye trykk? Det kan da påvirke testene våre, og gi feil data.
 - Vi får ikke tilgang til å gjøre fysiske endringer på serveren. Hvis det trengs å gjøre endringer, må det tas kontakt med IKT-tjenester.
 - For IKT-tjenesten er MySQL kanskje det letteste.
 - Gi en skriftlig bestilling til IKT-tjenesten om hva vi trenger. Denne kan sendes direkte til Svein-Inge Kvalø. Hvilken database og hvor ofte skal det tas backup. Hvor mange sms det skal sendes i måneden?
- Nyttige lenker vi burde kikke på:
 - <https://www.eidsiva.net/privat/dugnadsfiber>
 - <https://www.jegvilhafiber.no/>
 - <http://eidsiva.net/nh>

Tirsdag 17.01.17

Tilstede: Frode, Petter, Lasse, Kristin, Ingvild

- Vi burde dokumentere hva som er laget før jul av løsningen. Og ta med dette i hovedrapporten vår for å vise hvor langt vi har kommet, og hva som faktisk ble gjort i bacheloroppgave prosessen.
- Forprosjektet skal ligge ved som vedlegg når vi leverer hovedrapporten. Vedlegget av forprosjektet skal være samme versjon som vi leverer 28. januar.
- Mye av jobben vi gjør i forprosjektet kan brukes direkte i kapittel 1 i hovedrapporten, så det lønner seg å gjøre en god jobb allerede nå.
- Hvis Frode skal lese gjennom noe av det vi skriver, burde det leveres printet ut 24 timer før veiledning.
- Kommentering av noen punkter i forprosjektet:
 - Effektmål: Hva skal oppdragsgiver sitte igjen med.
 - Resultatmål: Programvaren vi produserer. Det vi leverer i mai.
 - Rammer: Teknisk (Tilgang til software, hardware osv), praktisk (grupperom), føringer (Rammer og krav fra oppdragsgiver).
 - 2.1: Fagområde: Snakke litt generelt om hva apper kan gjøre. Sammenhengen mellom apper og websider. (ca 1/3 A4 side)
 - 2.2: Avgrensing: Skrive om hva appen og løsningen vår skal gjøre (ca 1/3 A4 side)
 - 2.3: Oppgavebeskrivelse: Konkret hva vår løsning skal gjøre (1 A4 ark)
 - Fagområde, avgrensing og oppgavebeskrivelse: Dette blir kapittel 1 i hovedrapporten. Dette burde tilsammen være 2 sider. Viktig at dette skrives bra ettersom dette er det første som møter sensor og leseren.

- Vi må begrunne hvorfor vi har valgt scrum og forklare hvorfor scrum passer for oss. Pass på at Scrum passer for oss og vårt prosjekt, og at vi ikke velger det fordi det er populært.
- Kravspek: Bare hva som skal lages, ikke hvordan det skal lages. Pass på at kravene ikke er for ullne. Kravene burde være så konkrete at de kan måles eller bevises.
- Vi burde begynne å sette oss inn i ny teknologi allerede i januar.
- Det skal leveres minst 3 statusrapporter
 - Disse burde være på ca 2 A4-sider. Det finnes en mal på fronter.
 - Forslag på innleveringsdato på disse: 20 feb, 25 mars, 25 april
- Fast veiledning hver uke: Tirsdager kl 10.15
- Frode skal høre ang. websiden.

Mandag 23.01.16

Tilstede: Lasse, Petter, Kristin, Ingvild, Egil

- Hver enkelt kommune sender inn søknad til Nkom. Fylkeskommunen v/ Egil hjelper til med prosessen. Regionalforvaltning.no brukes til å sende inn søknad.
- Søknad skal kunne genereres av kommunene og fylke.
- Kommune kan legge inn informasjon på vegne av leverandører.
- Leverandører kan ha tilgang til alle målinger.
- Innlogging av brukere: Er dette nødvendig?
 - Henvendelse som ikke er forpliktende
 - Kan være todelt?
 - Hastighetsmåling og registrering skal være lett. Verifisering med sms?
 - Varmekart skal være åpent.
 - Opprette bruker for å se tidligere målinger og få kommunikasjon.
 - Finne ut hva minId innebærer
 - Datatilsynet: Hva kan vi lagre?
 - Hva trigger innbyggeren til å delta mer? Kommune high-score? Ha en positiv vinkling på ting.
- Kommunen kan bestemme på forkant om et område skal bli bygget ut. Før de har fått inn interesse fra innbyggeren. Eller de kan bestemme i etterkant, etter interesse fra innbyggerne.
- Take rate: de innbyggerne som har signert kontrakt.
- Sjablongverdi for hvert område.
- Nkom regner med at det koster i snitt 55 000 i sjablong, de skal ha mest mulig bredbånd for pengene, de vil helst ha under 55 000
- Krav fra nkom. Områder som skal lyses ut i mai (søke i mai) skal lyses ut offentlig minimum en måned før. Dette område skal lyses ut, meld din interesse.
- Informasjon om hvilken husstand som har gjort målinger eller svart at de er interessert burde kanskje ikke vises helt åpent.
- Sette et tidspunkt der informasjon blir offentlig lyst ut.
- Egil skal holde en workshop for hvordan skjema til søknad skal fylles ut på seminaret på Elverum. Dette bør vi være med på.

- Egil sender over søknadsmalen fra nkom og en tidligere søknad til oss.

Tirsdag 24.01.16

Tilstede: Ingvild, Kristin, Petter, Lasse, Frode

Tilbakemeldinger på rapporten:

- Lese hverandres kapitler. Vær kritisk fra første stund på språket.
- Få noen utenfra til å lese rapporten mot slutten. Fint å se rapporten med friske øyne.
- Bakgrunn:
 - Hvorfor finnes dette prosjektet?
 - Hva er allerede gjort? Hos fylkeskommunen og før jul.
 - Hvorfor ønsker de seg dette prosjektet?
 - Hva er bakgrunnen for dette prosjektet?
 - Første setning - utklar. Denne kan strykes.
 - Det heter ikke utstrøk.
- Effektmål:
 - Fylkeskommunen skal ved hjelp av.. "Håper" er ikke riktig ord å bruke.
 - Dårlig norsk på deler av dette avsnittet.
 - Ikke start to setninger rett etter hverandre med "dette".
- Resultatmål
 - ios som ..
 - Ikke så mye som
- Rammer:
 - Har vi kontor på Hamar?
 - Sitter vi på skolen?
 - Skal det inn mer her som er føringer?
- Fagområde:
 - utvikling til ios foregår i xcode. er dette egne valg eller føringer? Er dette eneste muligheten til å utvikle til ios?
 - Teori om bredbånd, og historikken til bredbåndsutvikling. Hvordan ble det gjort før, hvordan gjøres det i dag? Hvordan startet internett? Det er forskjellig dekning i landet osv.
 - Dra inn at leverandører har egne dekningskart som de ikke vil dele med andre.
 - Alt det tekniske som står i denne delen må komme senere i oppgaven. Enten i slutten av oppgavebeskrivelsen. Eller etter kravspek.
- 2.3: Veldig bra! Konkrete og mye.
- Avgrensing:
 - Vi skal se på er å måle dekning i tynt på befolkede områder. De ønsker en app og nettside.
- Systemutviklingsmodell:
 - Vise at vi har vurdert andre modeller.
 - Begrunne hvorfor disse ikke er aktuelle.
- Risikoanalyse

- Bruk farger.
- Burde vi ha tiltak for punkt 4 og 5
- Fjerne punkt 5? hvis vi har rutiner på backup
- Gantt:
 - Legge inn statusrapportene i gantt-skjema
 - Milepæler. Er det noe som må være ferdig før resten av prosjektet kan gå videre
 - Beslutningspunkter - Er det noen datoer vi må ha bestemt noe spesielt?
 - Teknologier, kart, servere,
- Kravspek: 4-8 sider er det normale (bare tekst, uten figurer). For oss blir det automatisk mer ettersom vi skal lage så mye.

Torsdag 26.01.17

Tilstede: Kristin, Lasse, Ingvild, Egil

- Alle søknader skal ha en status (Se eget dokument om dette).
- Må synliggjøre i søknaden at lokal medfinansiering burde vært tett opp til 40%.
- Poengsum helt til slutt i søknaden. Poengsummen burde oppdateres ettersom flere og flere ting blir fylt inn i søknaden. Kan gi opp til 10 i score.
- Verktøyet skal støtte at det kan tas ut en smultring fra kartet (Et område der det er bedre dekning).
- Jo flere personer som blir lagt inn under hvitt område i søknaden, jo bedre ettersom dette feltet er vektet med 30%.
- Punkt 3.1: Egil misforstod dette punktet når han fylte inn søknaden i fjor.
- En kommune skal kunne dele sin søknad med andre kommuner.
- Tidligere søknader skal ligge arkivert i verktøyet. Skal kunne ta kopi av søknaden, slik at man kan jobbe videre på den et senere år.
- En ildsjel skal ikke kunne se sjablong verdien.
- En ildsjel skal kunne fylle inn dugnad per husstand (informasjon om graving på egen tomt).
- En ildsjel skal kunne legge inn informasjon og meldinger til kommunen og innbyggerne. Disse meldingene skal logges.
- Regnestykke i søknaden skal kunne bytte mellom å være per husstand eller totalt.
- En ildsjel skal kunne registrere om det finnes en bedrift på en husstand.
- Når en måling er gjort kan en innbygger krysse av på om dette er privatadresse, hytte eller arbeidsplass.
- Vi må huske på at det er mange arbeidsplasser som ikke har noen husstand og adresse (Skoghogst, turisme osv)

Mandag 30.01.17

Tilstede: Kristin, Lasse, Ingvild, Petter, Egil

- Hva skal vi gjøre denne uken her:

- Kravspesifikasjon
- Forberede oss til seminaret på Elverum
- Hva forventes på presentasjonen:
 - 30 minutter - åpent til å bruke mer tid hvis vi trenger det
 - Egil kommer til å fortelle om digital plan for kommunene rett før vi skal presentere. Promotere vårt prosjekt som et verktøy for å få status
 - 48 deltagere med oss
 - Det er i hovedsak kommuner som deltar på seminaret
 - Kommune, Regionråd (4 regionråd i Hedmark, kommunene hører til hvert sitt), Sør-Trøndelag fylkeskommune (har gjort dette i 9 år), Holtålen (kommune fra Sør-Trøndelag som er god på dette, Telenor, Eidsiva, Nkom.
- Lag en presentasjon
 - Starte med oppgaveteksten
 - Ta de tingene som vi forventer at verktøyet skal klare - Ta med skjermbilder? Prototype?
 - Vise mye med bilder : Kommunekart - et kart som viser hvordan de kan markere et område til utbygging
 - Vise frem en fremdrift?
 - Innbygger gjør en hastighetsmåling. Hva skjer så?
- Workshop på Elverum
 - Vi skal delta som en kommune.
 - Lære mer om søknadsskriving.
 - Deler oss opp slik at vi deltar på alle gruppene så vi kan høre erfaringer og utfordringer fra alle kommunene.
- Litt forskjellig ang. hvor mye de ulike kommunen har kunnskap om.
 - I 2016 fikk Trysil og Stange innvilget søknad
 - I 2015 fikk Nord Odal og Helgøya og Nes innvilget søknad
 - Hun som skriver søknad for Trysil er veldig god - andre kommuner har leid inn henne for å få hjelp til å skrive søknad
- Søknaden kan ha en liste over alle leverandører som kommunen må spørre om utbygging før de sender inn søknaden
 - Fylke kan legge inn de største leverandørene globalt
 - Kommunene kan legge inn de små lokale leverandørene selv.
- Neste møte - Tilbakemelding presentasjonen
 - Torsdag 02.02 kl. 9

Mandag 30.01.17

Tilstede: Lasse, Petter, Kristin, Ingvild

- Ruby on rails: versjon 5.0.1
 - Ruby: versjon 2.3.3
- Xamarin for iOS og Android

Tirsdag 31.01.17

Tilstede: Petter, Kristin, Ingvild, Frode

- Setningslengde
- Klammeparenteser rundt referanser i teksten.
- Skrive Nkom riktig. Nkom eller NKOM.
- Ikke start og slutt setningen med samme ord.
- I bakgrunn: Få det klart at det er kommunene som skal søke om støtte hos Nkom. At fylkeskommunen bare skal støtte dem.
- Avgrensing:
 - Få det klarere hvem som har bestilt dette og hvem brukere er.
 - Verktøy som er laget for fylkeskommunen som kommunene skal bruke.
- Gant-skjema:
 - Forklare hva fargene betyr
 - Kutte ut den ene fargen.
 - Legg den ved som den er i forprosjektrapporten - og en ny oppdatert versjon i hovedrapporten med oppdatert informasjon.
- Legg ved oppdatert versjon av forprosjektrapporten i hovedrapporten.
- Kravspek:
 - Oppetid på servere: Skal vi skrive om dette ettersom det ikke er vi som skal drifte det. Naturlig å si hva det forventes at systemet skal klare.
- Si noe om rammene rundt de serverne
- Neste uke:
 - Ikke møte tirsdag
 - Send mail hvis vi vil ha møte på mandag i stedet for.

Torsdag 02.02.17

Tilstede: Lasse, Petter, Kristin, Ingvild, Egil

- Egil er litt usikker på om søknaden til Nkom er en fastsatt mal der alle avsnittene må være med. Vi kan spørre representanten for Nkom på seminaret.
- Tall og andre ting i søknaden kan forandre seg fra år til år. Det må være enkelt å endre dette i systemet slik at det er lett å vedlikeholde videre.
- Hvis det er enkelt å redigere inne på søknaden, så holder det med at det bare blir generert en pdf. Trenger ikke å generere word-fil.
- Kolonne N i excelarket over svar fra Elverum kommune burde være med i søknaden. Grunn til hvorfor husstanden trenger nett. Informasjon om dette kommer ikke til å gi noe utslag på søknaden hos Nkom. Det blir tatt med i vurderinga når fylkeskommunen skal vurderer å gi støtte.
- Innbyggerne skal kunne velge om de registrerer hastigheten på en husstand, bedrift eller fritidsbolig.
- Vi kan tipse innbyggerne om at det lønner seg at flest mulig registrerer seg. At det gir større sannsynlighet for utbygging. Skal det være mulig at en innbygger tipser en annen innbygger ved å sende en link til nettsiden og appen?
- Vi må informere om på nettsiden og appene at vår kartlegging ikke er bindende.
- Vi må ta hensyn til at ikke alle brukere klarer å bruke appen og nettsiden.

- Hvis du har over en viss grense med hastighet, så får du ikke alle spørsmålene når du registrerer deg?
- Gjør så mye som mulig enkelt for brukerne som skal registrere seg. Komme med forslag for hva slags bredbånds tilkobling er mest sannsynlig utfra hvilken leverandør det er.
- Brukerne skal kunne verifisere sin registrerte epost med en link.
- Kommunen og ildsjeler skal kunne fylle inn interesse informasjon manuelt.
- Ildsjelen kan legge inn eposter til hyttefeltet. Sende ut mail til alle dem. Hytteeierne registrere seg selv. Eventuelt hoppe over hastighetsmålingen ettersom de mest sannsynlig ikke er på hytta når de mottar mail. Knytte hastighetsmåling som blir gjort senere til den første forespørselen. Knytte med adresse eller navn?
- Må ta høyde for at registreringen ikke nødvendigvis skjer på den aktuelle adressen.
- Mandag 06.02: kl 14.30. Svein-Inge vil ha en agenda. Lasse sender spørsmål på mail til Egil og Svein-Inge.

Mandag 06.02.17

Tilstede: Lasse, Petter, Kristin, Ingvild, Egil, Svein-Inge, Anders

- SMS tjenester:
 - 500 - 1000 SMS i 2016. Litt mer i 2017.
 - Link mobility
 - Vi får eget brukernavn og passord. Dette blir en underkonto til hele løsningen.
 - Svein-Inge sender dette på mail så fort det er klart.
- Domene.
 - De kan bestille domene.
 - Det støtter æ,ø,å.
 - norid.no for å sjekke ledige domenenavn
 - Svein-Inge setter opp drifta på dette når vi har bestemt et domenenavn.
- Servere
 - PostgreSQL er greit.
 - Send over IP-adressen til NTNU. IT-avdelinga setter opp ssh tilgang til serveren. Vi må koble oss på via VPN når vi jobber hjemmefra.
 - Hvis vi trenger å installere noe på root, så må vi sende en mail til IT-tjenesten så ordner de dette for oss.
 - Vi kan sende en liste over ting vi veit vi trenger fra de serverne så fort som mulig.
 - Servere skal være klare til bruk i løpet av uka.
 - De oppretter en disk med minimum 10 GB til lagring av pdf og bilder.
- Apple developer account
 - Svein-Inge kan ringe i morgen, for å høre om den registrerte kontoen er knyttet til Hedmark trafikk eller fylkeskommunen.
 - Svein-Inge kan gi oss tilbakemelding på dette.
- 23. Februar: Møte på Hamar 09.00 - 13.00

Mandag 20.02.17

Tilstede: Lasse, Petter, Ingvild, Kristin, Egil

- En forespørsel kan sende varsel til mange personer
- Skal leverandør gi pris på prosjekt eller subprosjekt: subprosjekt
- Skal leverandør svare på planer på hvert subprosjekt eller prosjekt: subprosjekt
- Legge på varsel på prosjektet, om at en leverandør ikke har godkjent ett eller flere subprosjekt.
- Kommune kan legge inn totalsum, som fordeles på hver husstand. Eller de kan legge inn pris per husstand, som blir summert opp til en total sum.
- Petter til torsdag: Finne ut hvilke info vi kan hente inn uten at en brukes skal verifisere seg?

Torsdag 23.02.17

Tilstede: Lasse, Petter, Kristin, Ingvild, Egil, Svein-Inge, Marit

- Vi må lage en brukermanual som brukere kan følge for å gjennomføre en hastighetsmåling og forespørsel av bedre nett. Denne kan fylkeskommunene sende ut hvis noen har problemer med å gjennomføre ting. Vise med bilder hvordan ting skal gjøres. Og eventuelt problemer som kan oppstå med løsninger.
- IT-avdelingen vil at vi skal bruke den grafiske profilen som er utarbeidet for fylkeskommunen. Denne skal brukes på alle ting som fylkeskommunene eier. Den grafiske profilen finnes her: <https://www.hedmark.org/Om-fylkeskommunen/Organisasjonen/Grafisk-profil>
- IT-avdelingen vil gjerne at vi skal bruke PHP som backend språk, slik at de har mulighet til å drifte løsningen videre. De har ingen kompetanse på Ruby on Rails.
- IT-avdelingen vil også at vi skal utvikle slik at det som er innholdsmessig nyttig kan kunne brukes videre - som med iframe.
- Sett opp en liste til mandag, med fordeler og ulemper for bytte til PHP. Vi bestemmer om vi skal eventuelt bytte under møte på mandag.

Mandag 27.02.17

Tilstede: Lasse, Petter, Kristin, Egil, Ingvild

- IT-avdelingen kom med et ønske om at vi skulle bytte til PHP som backend språk forrige uke. Vi har satt opp noen fordeler og ulemper å bytte til dette nå.
 - Vi gjennomgikk fordelene og ulempene med Egil.
 - Egil vil at vi skal fortsette med Ruby. Alle er enig i dette.
- Det er lovpålagt om universell utforming etter forskrift for universell utforming av IKT-løsninger: <https://lovdata.no/dokument/SF/forskrift/2013-06-21-732>
- Det har vært en navnekonkurranse på fylkeskommunen: Det skal være samme navn på nettsiden og appen.
 - bredbåndhedmark → bredbaandhedmark, bredbandhedmark
 - breibandhedmark
 - bbhedmark
 - bredbandinnlandet

- Link Mobility:
 - Tilbyr leveringsrapport på sms som blir sendt ut
 - 1000 sms i måneden = 650 kr. 290 i fastpris, så pris per sms.
 - Når du logger inn som admin på nettsiden, burde den kunne se hvor mye av denne kvoten er brukt opp.
 - Det skal settes opp en brukerkonto som vi får tilgang på.
- Det bør være en logg inn knapp i appen: Denne knappen skal føre til logg inn siden i nettleseren.

Mandag 06.03.17

Tilstede: Lasse, Petter, Ingvild

- Hendelsesforløp for websiden utgikk forrige uke på grunn av sykdom. Ny tid for møte er onsdag 08. mars kl 11.30.
- Appelkonto: Dette oppretter vi sammen på møte på onsdag
- Egil fortsetter prosessen ang. Link Mobility

Onsdag 08.03.17

Tilstede: Lasse, Petter, Egil, Ingvild

- På nettsiden:
 - Knapp for registrering av ny bruker inne på logg inn pop-up.
 - Registreringssiden:
 - Dropdown-meny der brukeren velger hvilken rolle de vil ha - innbygger, ildsjel, kommune, leverandør, fylkeskommune.
 - Flere valg dukker opp utfra hvilken rolle de vil registrere seg til.
- Ildsjel:
 - Har full tilgang til alle prosjekter i kommunen de er godkjent hos.
 - Kan ikke endre prosjekt informasjonen (navn, område osv).
 - Skal kunne legge til bedrifter uansett hvor de er lokalisert i kommunen.
 - Skal kunne godkjennes og legges til av fylke?
- Leverandører kan ikke legge til andre brukere.
- Verifisere e-post til brukere(kommune, leverandør, fylke), i tillegg til min-id innlogging.
- Fylkeskommunen:
 - Må godkjenne leverandører som ønsker å registrere seg.
 - Må godkjenne kommuner som ønsker å registrere seg. Skal kunne gi en kommune-bruker tilgang til flere kommuner.
 - Liste over brukere som venter på godkjenning inne på nettsiden.
 - Skal kunne legge til leverandører i leverandør-lista.
- Søknad:
 - Fryser søknaden på det tidspunktet når søknaden blir levert til Nkom. Det er ikke mulig å gjøre endringer etter dette. En kommune må eventuelt da lage en kopi for å jobbe videre.

- Prosjekter:
 - Sub-prosjekt:
 - Send varslings til innbyggerne.
 - Se statuser fra leverandører.
 - Hoved-prosjekt:
 - Utrekning.
 - Ildsjeler.
- Ønsket funksjonalitet (senere):
 - På registreringssiden: En innbygger kan registrere seg som interessert i å være ildsjel. Denne forespørselen blir så sendt til kommunen for godkjenning.
 - En fylkeskommune kan gi en kommune-bruker tilgang til flere kommuner.
 - Skal kunne slå sammen flere prosjekter, eller splitte opp prosjekter.
 - Fylkeskommunen vil ha en oversikt over alle prosjekter i fylket og statusene på disse prosjektene. Generere rapport (csv-format).
 - Søknad ut i word-mal

Mandag 20.03.17

Tilstede: Lasse, Kristin, Ingvild, Egil

- IT-tjenesten har allerede kode for minID-innlogging. Vi kommer til å bruke denne koden for å spare tid, og for at IT-tjenesten da kan vedlikeholde den delen av koden. Egil synes det er en god ide.
- Vi har laget en liste over prioriterte arbeidsoppgaver, dette diskuterer vi med Egil og lager en liste alle parter er enig om. Se eget dokument. Dette sendes også på mail til Egil.

Tirsdag 28.03.17

Tilstede: Lasse, Ingvild, Kristin, Petter, Frode

- Frode synes det er lurt å starte med rapportskrivning, mens vi utvikler.
- Neste uke er Frode borte på tirsdag. Møte blir torsdag 06. april kl. 9.00.
- Bestemme i gruppa hva Petter må gjøre fremover for å ta igjen lite arbeidsinnsats. Status på dette skal tas opp på neste møte. Sendt tiltakene inn til Frode, signert av alle fire.

Torsdag 06.04.17

Tilstede: Lasse, Ingvild, Kristin, Petter, Frode

- Oppsummering av hvordan oppgaven til Petter gikk.
- Frode tok en avgjørelse at gruppen vår blir splittet i to. Lasse, Kristin og Ingvild fortsetter sammen. Petter fortsetter alene. Petter får tilgang til all kode og dokumentasjon frem til denne datoen.

Tirsdag 25.04.17

Tilstede: Kristin, Lasse, Ingvild, Frode

- Gjennomgang av siste statusrapport.
- Vanlig med 20-30 figurer i rapporten.
- Hvordan bygge opp rapporten:
 - Kap 1: Innledning
 - Kap 2: Kravspek
 - Kap 3: Design: Hvordan kravspeken skal løses - userstories
 - kap 4: Forklaring av teknologier vi har brukt
 - implementering, kodingen, testingen
- Hvordan fremstille databasen i rapporten:
 - Kan gjerne vise utklipp fra svg-filen i rapporten som eksempel, men forklaringer. Kan legge ved hele svg-filer som vedlegg for å vise kompleksiteten av databasen. Men bare referere til hele fra hovedteksten.
 - Skal vi skrive om alle tabellene? Nei, vise en eller to typiske tabeller i rapporten.

Tirsdag 02.05.17

Tilstede: Kristin, Lasse, Ingvild, Frode

- Møte neste uke: Tirsdag kl. 10.15 og Fredag kl. 13.00
- Gjennomgang av vår disposisjon:
 - Omfang bør komme tidlig i oppgaven.
 - Bakgrunn kan bakes inn i formålet med oppgaven.
 - Vi burde slå sammen kap 1 og 2. Prosjektorganisering og rammer bør komme i slutten på kapittel 1.
 - Under rammer og gjennomføringen av prosjektet: Der kan vi skrive om at vi mistet et gruppemedlem.
- Andre kommentarer til rapporten:
 - Kravspek skal skrives ut fra at vi er 3 stykker på gruppa.
 - Vi burde skrive om de oppgavene som er halvferdige. Vise hva vi har brukt tiden vår på.
 - Ikke ha med ting i kravspek som ikke ble realisert.
 - Alt vi brukte tid og energi på burde vi skrive om.
 - Det vi ikke rakk å gjøre, skal vi skrive om i siste kapittel.
 - Product backlog: Liste over hva vi har gjort. Hvis dette kommer under andre punkter, så trenger vi ikke product backlog

Tirsdag 09.05.17

Tilstede: Frode, Ingvild, Kristin, Lasse

- Veldig bra rapport, omfattende skrevet.
- Omstrukturere på kapittel 1.

- Fagområde, avgrensning og oppgavebeskrivelse, Formål, Målgruppe, egen bakgrunn og kompetanse.
- Arbeidsmetoder: Hvordan har vi jobbet. Hvordan har vi gått frem for å lære nye ting? Hvordan kom vi inn i problemstillingen?
 - eks: Vi har vært i Hamar, vi har vært på seminar om bredbånd
- Legge til Gantt diagrammet fra forprosjektrapporten i kapittel 1. Den oppdaterte Gantt diagrammet kommer i siste kapitlet.
- Skrive til slutt i kap 1 hvordan hele rapporten er bygd opp. Hva som kommer i hvert i kapittel.
- Kap 7. Husk å følg malen
- Fagområde: For muntlig skrevet.
- Rammer kan heller kalles ytre rammer.
- Plattformen, løsningen eller systemet. Velg et ord for hele prosjektet
- Begrunn hvorfor vi har valgt scrum i kapittel 7. Var dette lurt? Eller passet det ikke helt?
- Ha innledning på alle kapitler
- Teorikapitlet er et veldig bra kapittel - gir et veldig godt grunnlag.
- Si det klarere at personas ikke er reelt og kommentere bilderettigheter.
- Veiledning mandag 15.05 kl 11. Ikke veiledning fredag.

VEDLEGG K

Timelister

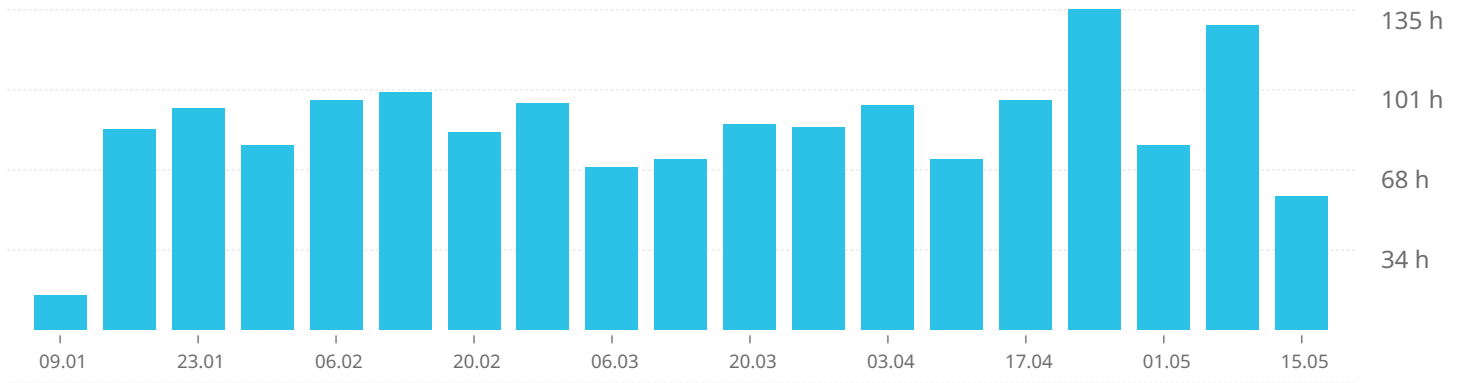
Summary report



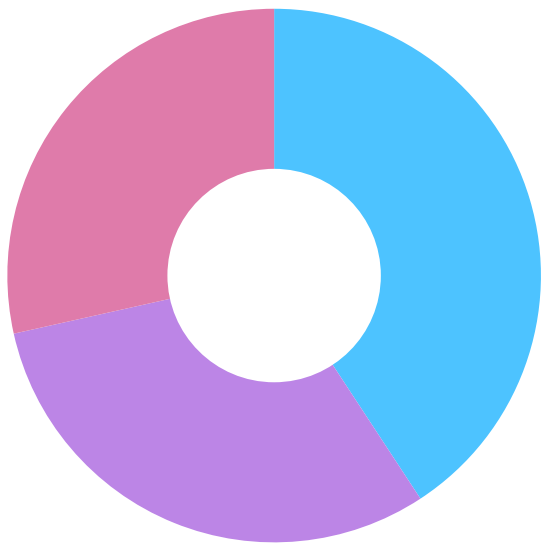
2017-01-11 - 2017-05-16

Total 1609 h 22 min

Ingvilna, Kristin Pettersen, Lassesviland selected as users

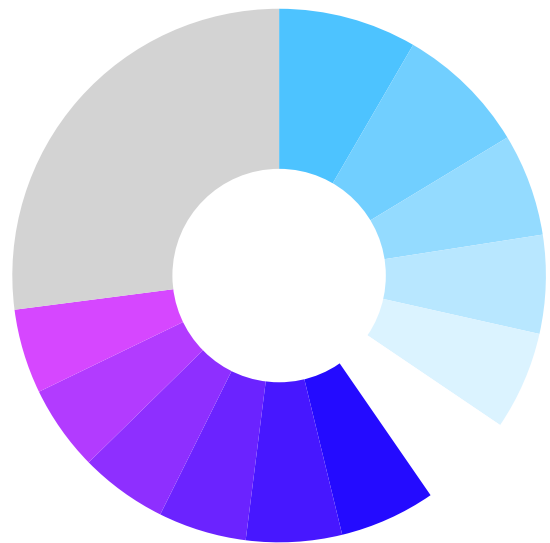


Users



- Lassesviland 655:59:54
- Ingvilna 494:47:07
- Kristin Pettersen 458:35:58

Projects



- Sprint 15 134:49:15
- Sprint 17 128:09:01
- Sprint 5 99:53:10
- Sprint 4 95:59:18
- Sprint 14 95:47:55
- Sprint 7 95:23:51
- Sprint 12 93:28:15
- Sprint 2 93:22:30
- Sprint 10 86:04:52
- Sprint 11 85:20:02
- Sprint 1 83:21:17
- Sprint 6 82:27:47

Users / Projects	Duration
Ingvilna	494:47:07
Sprint 0	5:40:00
Sprint 1	33:25:01
Sprint 10	28:58:00
Sprint 11	31:49:32
Sprint 12	29:07:15
Sprint 13	23:27:44
Sprint 14	28:03:55
Sprint 15	35:48:09
Sprint 16	18:22:32
Sprint 17	40:55:14
Sprint 18	14:48:45
Sprint 2	31:15:00
Sprint 3	25:00:00
Sprint 4	27:46:00
Sprint 5	24:05:00
Sprint 6	27:15:00
Sprint 7	20:45:00
Sprint 8	26:40:00
Sprint 9	21:35:00
Kristin Pettersen	458:35:58
Sprint 0	4:25:00
Sprint 1	20:55:14
Sprint 10	27:10:52
Sprint 11	23:59:30
Sprint 12	33:10:00
Sprint 13	17:27:00
Sprint 14	24:57:00
Sprint 15	35:57:06
Sprint 16	23:15:38
Sprint 17	33:28:47
Sprint 18	20:40:01
Sprint 2	23:26:42
Sprint 3	30:12:16
Sprint 4	24:07:18
Sprint 5	27:38:10
Sprint 6	23:44:47
Sprint 7	35:37:37

Sprint 8	12:27:00
Sprint 9	15:56:00
Lassesviland	655:59:54
Sprint 0	4:25:50
Sprint 1	29:01:02
Sprint 10	29:56:00
Sprint 11	29:31:00
Sprint 12	31:11:00
Sprint 13	30:45:00
Sprint 14	42:47:00
Sprint 15	63:04:00
Sprint 16	35:45:00
Sprint 17	53:45:00
Sprint 18	19:40:00
Sprint 2	38:40:48
Sprint 3	21:38:00
Sprint 4	44:06:00
Sprint 5	48:10:00
Sprint 6	31:28:00
Sprint 7	39:01:14
Sprint 8	28:56:00
Sprint 9	34:09:00