

Bacheloroppgave:

SMARTHUS

Forfattere:

Edvard Brekke

Jørn-Are Flaten

Henrik Haukaas

Dato:

18/05-16

SAMMENDRAG

Tittel:	Smarthus	Dato :	18/05-16
Deltakere	Edvard Brekke		
	Jørn-Are Flaten		
	Henrik Haukaas		
Veileder:	Frode Haug		
Oppdragsgiver:	Vanja Ohna		
Stikkord/nøkkelord	Java, Android, styringssystem, systemutvikling, database, smarthus		
Antall sider/ord: 133/32802	Antall vedlegg: 15	Publiseringsavtale inngått: ja	
Kort beskrivelse av bacheloroppgaven:			
<p>Denne rapporten omhandler utviklingen av et styringssystem for et smarthus. Hensikten med prosjektet er å gjøre det enkelt for beboere å styre diverse utstyr i boligen, som lys, varmpumpe, ventilasjonsanlegg, og takvindu. Gjennom dette er det også ønskelig at beboere skal bli mer oppmerksom på eget forbruk av vann og strøm. Gjennom en kombinasjon av energieffektive og miljøvennlige komponenter med et styringssystem er vårt ønske å skape et mer miljøvennlig smarthus.</p> <p>Applikasjonen er utviklet for Android-enheter, og består av en mobilapplikasjon som kalles SHK-applikasjon og en serverapplikasjon som kalles SHK-server.</p>			

ABSTRACT

Title:	Smarthus	Date :	18/05-16	
Participants:	Edvard Brekke			
	Jørn-Are Flaten			
	Henrik Haukaas			
Supervisor(s):	Frode Haug			
Employer:	Vanja Ohna			
Keywords:	Java, Android, control system, software development, smart home, database			
Number of pages/words:	133/32802	Number of appendix:	15	
Availability:				open
Short description of the bachelor thesis:				
<p>This report revolves around the project of developing a control system for a smart home. The purpose of this project is to make it easier for the residents to control certain objects in their home, for instance lighting and heating. Through this, the control system also seeks to make the residents more aware of their own consumption of both water and power. Combining energy efficient and environmentally friendly components with a control system, our hope is to create a greener and more environmentally friendly smart home.</p> <p>This control system is developed for Android devices and consists of an Android application called SHK-app and a server application called SHK-server.</p>				

Forord

Denne rapporten tar for seg prosessen ved å utvikle et styringssystem for en tomannsbolig som deler felles energibrønn og varmepumpe.

Etter nesten fire måneder med planlegging og utvikling har vi et velfungerende styringssystem bestående av en Android-applikasjon og en serverapplikasjon.

Vi vil rette stor takk til HDL Nordic og vi ønsker spesielt å takke teknisk sjef, Morten Andersen, som gjennom hele prosessen har hjulpet oss med råd. HDL Nordic har vært meget behjelpelige med å låne oss en demokoffert, som fungerer som lite et smarthus, med IP-modul, LED-spotter, reléer, dimmere, og flere sensorer.

Vi ønsker også å få takket de som har bidratt med innspill på designet av mobilapplikasjonen og de som har korrekturlest rapporten.

Innholdsfortegnelse

1.0 Innledning	1
1.1 Problemområde	1
1.2 Hva er smarthus	1
1.3 Avgrensing	2
1.4 Oppgavedefinisjon	3
1.5 Målgruppe	4
1.6 Mål	5
1.7 Faglig bakgrunn, rammer og øvrige roller	5
1.8 Arbeidsprosessen og prosjektorganisering	6
1.9 Rapportens ordbruk og rapportorganisering	7
2.0 Forarbeid	8
2.1 Oppstartsfasen	8
2.2 Risikoanalyse	9
3.0 Kravspesifikasjon	11
3.1 Kravendring	11
3.2 Endelig kravspesifikasjon	11
4.0 Design og arkitektur	20
4.1 SHK-applikasjonens brukergrensesnitt	20
4.2 Design	24
4.4 Arkitektur	33
4.5 Kommunikasjonsprotokoller	34
5.0 Koding, kvalitetskontroll, implementering, og testing	35
5.1 Implementering av SHK-server	35
5.2 Implementering av SHK-applikasjon	39
5.3 Kvalitetskontroll og testing	46
6.0 Avslutning og konklusjon	50
6.1 Diskusjon	50
6.2 Resultater	51
6.3 Gruppeevaluering	52
6.4 Kritikk av oppgaven	53
6.5 Videre arbeid	54
6.6 Konklusjon	54

Vedlegg	56
A Terminologi	57
B Kommandoformat	58
C Z-Wave-kommandoeksempler	62
D Prosjektplan – levert 28. januar 2016	64
E Prosjektavtale	75
F Grupperegler	77
G Statusrapporter	79
H Oppdatert oppgavebeskrivelse fra oppdragsgiver	84
I Dialog med HDL	85
J Prosjektering og utstyrliste	92
K Møtereferat fra møter med oppdragsgiver	97
L Møtereferat fra møter med prosessveileder	105
M Oppdatert fremdriftsplan	120
N Ukentlig logg	121
O Timeføring	125

Figuroversikt

Figur 1 Topologinot	4
Figur 2 Use case for bruker i leilighet 1	11
Figur 3 Use case for leilighet 2 og hybelleilighet	12
Figur 4 Use case for administrator	12
Figur 5 Oversikt over inneklima	17
Figur 6 Android-distribusjon pr 2. mai 2016	18
Figur 7 Innloggingsmeny og passordbehandling	20
Figur 8 Hovedskjerm for leilighet 1	21
Figur 9 Lyssoner og lysoversikt på kjøkken	21
Figur 10 Klimameny	22
Figur 11 Temperaturoversikt og ventilasjon	22
Figur 12 Modusvalg	23
Figur 13 Innstillinger for modus	23
Figur 14 Innstillinger av feriemodus	24
Figur 15 Systemsekvensdiagram for "Glemt passord"	28
Figur 16 Endring eller etterspørsel av temperatur	29
Figur 17 Endring av modus	30
Figur 18 Håndteringsmetode i ServerConnection	32
Figur 19 Arkitektur	33
<h2>Liste med kodeeksempler</h2>	
Kodeeksempel 1 StartLoginMonitor	35
Kodeeksempel 2 Itererer gjennom brukerlisten	35
Kodeeksempel 3 Loginhåndtering	36
Kodeeksempel 4 Sjekk for om en melding er en kommandomelding	36
Kodeeksempel 5 Bygger en pakke	37
Kodeeksempel 6 Henter ut Datagrampakke	37
Kodeeksempel 7 Finner korrekt bruker	37
Kodeeksempel 8 Finner bruker med smartMessage	38
Kodeeksempel 9 Når tidsuret er ferdig	38

Kodeeksempel 10 smartMessage-utseende	38
Kodeeksempel 11 CRC	39
Kodeeksempel 12 Initialiserer onClickListener og setter bakgrunnsfarge.	40
Kodeeksempel 13 En typisk goToView metode som alle onClickListener knapper kaller på	40
Kodeeksempel 14 Er ikke tilkoblet server	40
Kodeeksempel 15 Henter ut brukernavn og passord for «Husk meg»-metoden	40
Kodeeksempel 16 Lagrer brukernavn og passord hvis «Husk meg» er krysset av	41
Kodeeksempel 17 Sjekker bruker-ID, og sender bruker til riktig hovedmeny	41
Kodeeksempel 18 Skru av eller på lys	42
Kodeeksempel 19 Skru av eller på trinn tre	42
Kodeeksempel 20 Steng vindu nr 3	43
Kodeeksempel 21 Sjekker om vinduet skal åpnes eller lukkes	43
Kodeeksempel 22 Oppdaterer brukeren på vinduets status	44
Kodeeksempel 23 Lagrer vinduets status	44
Kodeeksempel 24 Øker temperaturen med én grad Celsius.	45
Kodeeksempel 25 Modusvalg for dag	46

1.0 Innledning

1.1 Problemområde

Datamaskiner har gått fra å utgjøre en liten del av livene våre til å nå være en del av det aller meste. Det finnes blant annet smarte kjøleskap som holder oversikt over hva du har i kjøleskapet, og som man via en mobilapplikasjon kan se innholdet i når man er på butikken. Det kommer flere slike smarte enheter på markedet hele tiden, og det blir etterhvert mange løsninger å forholde seg til.

Folk flest har i dag boliger der alt fra enkle til mer komplekse oppgaver blir styrt manuelt av beboere. Alt av styring og kontroll er desentralisert til enkelte objekter, eksempelvis én styring for varmepumpen og én for varmekablene på badet. Dette medfører altså at dersom man ønsker å stille temperaturen i boligen, så må man gå bort til for eksempel varmepumpen og stille inn ønsket temperatur. Disse vil man, spesielt om man sammenligner med flere av dagens løsninger, gjerne kalle «dumme hus». Heldigvis er området i rask utvikling, og flere aktører kommer stadig på banen med sine egne løsninger. Dersom man ønsker å bygge ny eller oppgradere en bolig, vil man gjerne ha en løsning som har mulighet til å styre et vidt spekter av komponenter, som ventilasjon, varmepumpe og så videre. Det finnes i dag mange smarthusløsninger som gjør det mulig å kontrollere hele boligen fra egen smarttelefon, og det blir stadig mer populært. Vi kaller boliger med sentraliserte styringssystemer for «smarthus».

1.2 Hva er smarthus

Begrepet om smarthus omfatter en rekke elektriske installasjoner som bidrar til å gi boligen bedre komfort, lavere energiforbruk, enklere betjening og et høyere sikkerhetsnivå. Dette oppnås igjennom automasjon, styring og overvåking av elektriske installasjoner. Det vanligste i dag er styring av belysning og temperatur, men alt av teknisk installasjon kan anses som smarthusteknologi. Systemet styres med fjernkontroll, kontrollpanel eller ved hjelp av smarttelefon eller nettbrett. Dette vil si at brukeren kobler smarttelefonen sin til Internett og kan dermed fjernstyre systemet.

Ideen bak hjemmeautomasjon er ikke ny. Det har vært et tema i sci-fi i mange år og historien starter helt tilbake på 1800-tallet. Fjernkontrollen ble oppfunnet av Nikola Tesla i 1898. Tidlig på 1900-tallet ble husholdningsapparater innført, dette startet med støvsugere, men senere ble blant annet kjøleskap, vaskemaskiner og brødrister oppfunnet, men disse var dyre. På 1930-tallet ble ideen om «fremtidens hus» presentert. Her var det ikke hjemmeautomasjon som var tanken, i stedet var det snakk om å utvikle stemmekontrollerte roboter til å utføre våre ønsker. Det første hjemmet med automatisering var laget av en mekaniker i Michigan, USA i 1950. Dette var styrt med knapper på veggene og kunne blant annet dra for gardiner og skru på lyset i skapet automatisk når skapdøren ble åpnet. I 1966 ble det første kommersielle hjemmeautomasjonssystemet utviklet, Echo IV. Disse ble ingen kommersiell suksess på grunn av den høye prisen. I 1971 kom mikroprosessen, det gjorde at prisen på elektronikk falt dramatisk.¹

¹ <http://betanews.com/2015/08/24/the-history-of-home-automation-from-the-beginning/>

X10 er en protokoll for kommunikasjon mellom elektroniske enheter brukt i hjemmeautomasjon. Denne ble utviklet i 1975, og har vært brukt i former for hjemmeautomasjon siden slutten av 1970-tallet. X10 er fortsatt populær på grunn av den lave prisen og høy tilgjengelighet, og er brukt i millioner av enheter verden over.² Den ekstreme teknologiske revolusjonen de siste 20 årene har ført til at teknologi gradvis blir billigere, bedre og mer tilgjengelig. I dag kan smarthus kontrollere og overvåke TV, temperatur, belysning, sikkerhetsalarm, røykvarslere, vinduer, dører, samt gi deg oversikt over hva som er igjen i kjøleskapet. Alt dette og mer bare ved hjelp av en smarttelefon eller fjernkontroll. Det er fortsatt dyrere å bygge smarthus enn et vanlig hus, men teknologien blir bedre og billigere for hvert år og bare fantasien setter begrensningen på potensialet til hva som kan automatiseres. Inntjeningen med smarthus ligger i at energiforbruket blir redusert i at systemet er bevisst på å ikke sløse med energi. Dette oppnås ved at lys-, varme- og ventilasjonsanlegg jobber sammen.

Det vanligste når en skal oppgradere en bolig til smarthus er å starte med lys og temperaturstyring. I Norge ligger startpakker for lysstyring på ca 1500-2000 kroner, men i tillegg kommer ekstra kontakter og lyspærer som vil øke kostnaden noe. Disse startpakkene er så enkle å installere at forbruker skal kunne sette de opp selv uten hjelp fra fagpersoner. Startpakkene er ganske små, og inneholder typisk bare 2-3 dimmere og kontakter. Ferdiginstallerte lyspakker for hele leiligheter og boliger installert av fagpersoner kan ligge på 30 000 - 45 000 kroner. Det er flere norske selskaper som konkurrerer i markedet om smarthuspakker, både for nye og gamle boliger. Home Control, HDL Nordic og Future Home er ledende leverandører i det norske markedet.

1.3 Avgrensning

Dette prosjektet er avgrenset til et boligprosjekt i Kragerø og styringssystemet som skal utvikles for boligen. Videre vil vi holde oss til de rammer som er satt i oppgavebeskrivelsen gitt av oppdragsgiver og rammer som er blitt satt på grunn av hva og hvordan styringssystemene vi skal ta utgangspunkt i er utviklet. På grunn av hvordan oppgaven er satt opp og for å spare tid, vil vi gjenbruke funksjoner i HDL-systemet så langt det lar seg gjøre.

Systemets server skal ha rekkevidde til alle styringssystemene i alle boenhetene. Dette sørger komponentene for ved å danne et Mesh-nettverk.

Mobilapplikasjonen vil hovedsakelig brukes for kommunikasjon med selve BusPro-systemet, og det er der hovedmengden av funksjonalitet vil ligge. Systemets server vil fungere som et bindeledd mellom mobilapplikasjonen og systemets IP-modul på den måten at den videresender meldinger fra mobilapplikasjonen, men den inneholder også brukerdatabasen. På grunnlag av kompetanse i gruppen, samt hvor mye tid vi har til rådighet vil mobilapplikasjonen kun bli utviklet til Android. Vi ser derfor bort ifra ytterlige økonomiske kostnader for beboernes anskaffelse av Android-enheter.

Oppgaven er lagt opp på den måten av vi selv kan gjøre eksperimenter på grønn teknologi, ved å avgjøre hva som skal overvåkes av styringssystemer i tillegg til de kravene som er oppgitt. Vi står derfor fritt til å selv velge ut hvordan vi vil løse oppgaven så lenge retningslinjene blir fulgt. Dette avgrenses til hva som faktisk er mulig å få til ut i fra systemene vi skal forholde oss til.

² [https://en.wikipedia.org/w/index.php?title=X10_\(industry_standard\)&oldid=695804149](https://en.wikipedia.org/w/index.php?title=X10_(industry_standard)&oldid=695804149)

Dette gjelder:

- Er det praktisk mulig å kombinere de forskjellige systemene?
- Hva slags endringer må til for å få til ett fungerende system?
- Hva slags tilleggskomponenter må installeres for å koble sammen systemene?

Vi forutsetter at alle hardware- og elektriske komponenter fungerer når systemet skal installeres. I tillegg forutsetter vi at oppdragsgiver legger til rette for at alle komponenter og sensorer som trengs blir kjøpt og installert.

1.4 Oppgavedefinisjon

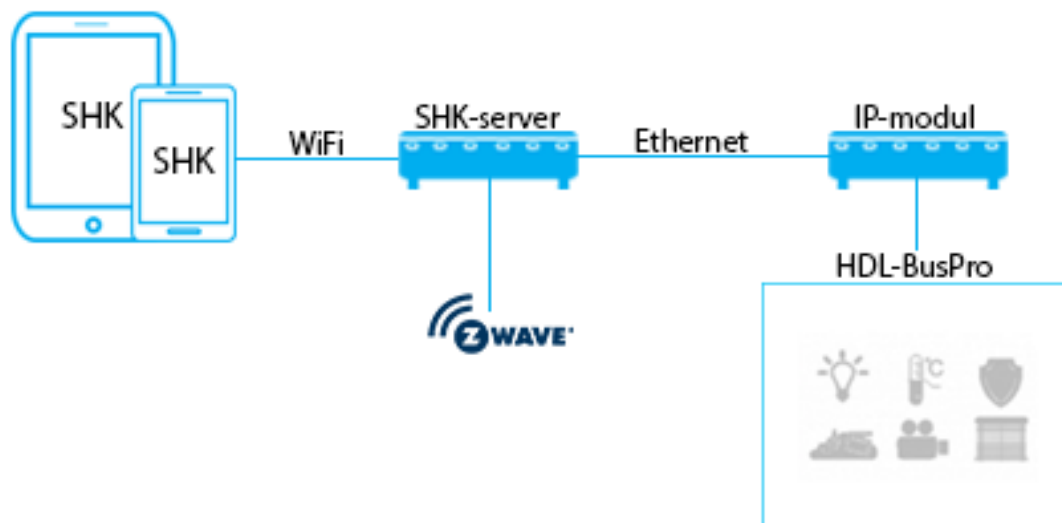
Hensikten med prosjektet er å legge til rette for smarthus med spesiell vekt på effektiv energiutnyttning og optimalt inneklima i en Siporex-bolig med to boenheter og en hybel/arbeidsrom. Energieffektivitet skal sikres ved optimal utnyttelse av lagret termisk energi og naturlig ventilasjon i kombinasjon med varmluftgjenvinning og tilførsel av varme fra bergvarmepumpe til vannbåren varme. Oppgavedefinisjonen ble ikke påvirket av kravendringen (2.1 Oppstartsfase), og er derfor lik oppgavedefinisjonen i begynnelsen av prosjektet.

Det skal utvikles et styringssystem som overvåker varmeproduksjonen fra varmepumpen og strømforbruket til ulike komponenter i bygget. Det skal kunne måles hvor mye varmtvann som går fra varmepumpen til de tre ulike boenhetene.

I tillegg til overvåkning skal styringssystemet kunne:

- Styre temperaturer i de ulike temperatursonene gjennom hele døgnet.
- Styre lys av og på, via demping og døgninnstillinger. Styling av andre elektriske komponenter kan også legges til.
- Styre åpning/lukking av tre Velux takvinduer i én av leilighetene. Disse vinduene skal registrere om det regner, og kunne lukke seg automatisk.
- Velge moduser som ferie, borte, natt og dag.
- Styre den balanserte ventilasjonen (LUNOS) som er desentralisert til de ulike rommene.
- Eventuelt:
 - Styre/regulere inneklimaet bedre - utover temperaturen, for eksempel luktfuktighet.
 - Måling av utetemperatur, vind og vindretning samt nedbør.
 - Tilpasse temperaturregulering inne til forandringer i utetemperatur
 - Utforske og eksperimentere og komme med forslag til annen teknologi som kan inngå i huset.

Med utgangspunkt i dette blir gruppens oppgave å prosjektere og velge ut komponenter som for det meste er bygd på åpne standarder (se vedlegg J Prosjektering og utstysliste), som muliggjør styring fra en tredjepart. Styringsprogramvaren skal bestå av en mobilapplikasjon, som vi kaller SHK-app. Denne applikasjonen må brukeren benytte for å sende kommandoer til en «mellomvare» på serveren, som vi kaller SHK-server. Denne serveren skal ta imot kommandoer fra mobilapplikasjonen, og utføre eller videresende disse. Serveren skal i tillegg fungere som en logikkmodul. Logikkmodulen skal kunne utføre diverse scenarioer, for eksempel ved et gitt tidspunkt skal boligen settes fra å være i «Dag»-modus til å være i «Natt»-modus. SHK-server sender kommandoer videre til IP-modulen i HDL-systemet. Topologisk visualisert ser det slik ut:



Figur 1 Topologi

Her (Figur 1) er nettbrettet og mobilen til venstre i bildet. Disse kjører SHK-applikasjonen og er koblet til SHK-server via lokalt WiFi. SHK-server er videre koblet til HDLs IP-modul via Ethernet, og til Z-Wave-nettverket via en Z-Wave-USB-avleser. IP-modulen tar imot eventuelle datapakker som inneholder kommandoer fra server og oversetter disse til HDLs eget format før de sendes videre inn på BusPro-systemet.

1.5 Målgruppe

Applikasjonen: Applikasjonen er ment å brukes av beboerne til å kontrollere og regulere lys, temperatur, ventilasjon, samt få en oversikt over strøm- og vannforbruk. Det er derfor viktig at produktet fungerer på en måte som gjør brukeropplevelsen positiv for brukerne.

Rapporten: Denne rapporten er skrevet for interessenter innenfor smarthus, oppdragsgiver, og prosjektsensor. Det tas forbehold om at leser har IT-kunnskaper tilsvarende informatikkstudenter.

1.6 Mål

1.6.1 Effektmål

- Systemet skal bidra til at oppdragsgiver oppnår et lavere og mer bevisst strømforbruk sammenlignet med norsk gjennomsnittsforbruk i tomannsboliger.
- Systemet skal legges til rette for at hver beboer skal kunne holde oversikt over sitt eget forbruk, samt at oppdragsgiver kan via en administratorkonto se det totale forbruket i hele boligen og til hver leilighet.
- Systemet skal bli laget på en måte som gjør det enkelt og bruke, slik at beboere av leilighetene enkelt skal kunne lese av sitt eget strømforbruk og bruke systemet uten særlig teknisk kompetanse.

1.6.2 Resultatmål

Målet med prosjektet er å oppnå redusert strøm- og vannforbruk ved å utvikle et system som kan overvåke og kontrollere hvordan huset kan spare energi. Prosjektet inkluderer overvåkingsverktøy som lar brukeren få mulighet til å overvåke temperaturer i forskjellige temperatursoner, samt strøm- og vannforbruk. Videre skal det inkluderes styringsverktøy som lar brukeren styre og regulere lys, ventilasjonsanlegg og temperatur. For toppleiligheten skal det også være mulig å styre tre stykk Velux Integra takvinduer. Det skal også være mulig for beboere å sette leiligheten i moduser som har forhåndsbestemte innstillinger for hver av de nevnte styrbare enhetene.

1.6.3 Læringsutbytte

Gjennom arbeidsprosessen kommer vi til å benytte forskjellige typer verktøy, som prosjektorganiseringsverktøyet for Scrum, Trello. Siden vi kommer til å utvikle programvarer og vi er tre personer i gruppen, er det viktig med versjonskontroll, så til det bruker vi versjonskontrollprogrammet, GitHub. Til utvikling bruker vi to verktøy; for å utvikle programvare til server skal vi bruke Eclipse og for mobilapplikasjonen skal vi bruke Android Studio. Siden utviklingen vil foregå i Java, må to av gruppens medlemmer bruke tid på å lære dette parallelt med utviklingsarbeidet. To av gruppens medlemmer har heller aldri jobbet med å utvikle programvare som kommuniserer over nettverk, så dette må også læres. For å lage grafikk bruker vi Adobe Photoshop. Til databasen kommer vi til å bruke Apache Derby, som er et relasjonsdatabaseverktøy(RDMS). I tillegg skal det nevnes at ingen på gruppen har jobbet med så store prosjekter før, så vi regner med at det blir en bratt, men spennende lærecurve. Andre læremål blir å benytte utviklingsmodellen, Scrum i praksis over en lengre periode.

1.7 Faglig bakgrunn, rammer og øvrige roller

1.7.1 Gruppens faglige bakgrunn

Prosjektgruppen består av tre medlemmer, Edvard Brekke, Jørn-Are Flaten og Henrik Haukaas. Vi er alle informatikkstudenter, som har fulgt studieløpet Dataingeniør ved NTNU. Edvard og Jørn-Are har fulgt Dataingeniør-løpet fra 2013, mens Henrik begynte på Programvareutvikling i 2013, og flyttet over til Dataingeniør høsten 2014. Utover de obligatoriske fagene har vi hatt relevante valgfag, Edvard har hatt fag som Introduksjon til kryptologi, Etisk hacking og penetrasjonstesting. Jørn-Are har hatt Introduksjon til kryptologi, Etisk hacking og penetrasjonstesting, og Programvaresikkerhet. Henrik har hatt Programvareutvikling og Innføring i informasjonssikkerhet.

1.7.2 Rammer

Gruppen forplikter seg til å forholde seg til rammer som er gitt av universitetet, med tanke på tidsfrister. Dette inkluderer frister for statusrapporter og prosjektrapporten 18. mai. Utvikling og testing vil i all hovedsak foregå på universitetet.

1.7.3 Øvrige roller

1.7.3.1 Oppdragsgiver

Oppdragsgiver er Vanja Ohna. Hun er universitetslektor ved NTNU i Gjøvik, og jobber til daglig ved avdeling for helse, omsorg og sykepleie.

1.7.3.2 Prosessveileder

Prosessveileder for oppgaven er Frode Haug. Han er universitetslektor ved NTNU i Gjøvik, og underviser i flere av de obligatoriske fagene i studiet Dataingeniør. Han har gjennom prosessen bidratt med tilbakemeldinger.

1.7.3.3 Kontaktperson

Morten Andersen er teknisk sjef i HDL Nordic, og har fungert som vår faglige kontaktperson. Han ble tidlig involvert i oppgaver og hjulpet oss med tekniske løsninger.

1.8 Arbeidsprosessen og prosjektorganisering

Alle gruppens medlemmer er fulltidsstudenter, og har derfor hatt gunstige forhold til å jobbe sammen daglig gjennom hele prosessen. Vi har hatt daglig dialog og vært i stand til å hjelpe hverandre når det har vært behov. Vi har hatt jevnlig kontakt med oppdragsgiver noe som har bidratt til at eventuelle innspill og ønsker har kommet regelmessig, og at hun har vært kontinuerlig oppdatert. Vi hadde regelmessig kontakt med prosessveileder, Frode Haug. Det var ønske fra begge sider om jevnlig og regelmessig kontakt, og vi hadde derfor møte hver mandag, så langt det lot seg gjøre. På møtene oppdaterte vi han på prosjektets fremgang, og drøftet eventuelle utfordringer vi stod overfor.

Det ble klart fra starten at vi hadde rimelig frie tøyler når det gjaldt hvilken teknologi og hvilke kommunikasjonsprotokoller (Z-Wave etc) som kunne brukes. Det ble også klart at vi fikk frie tøyler angående hvordan utviklingsprosessen skulle være og til en viss grad hvilket utstyr som skulle være med. Da byggeprosessen var i siste fase, var det derimot allerede bestemt hvilke takvinduer, hvilket ventilasjonsanlegg, og hvilken varmepumpe som skulle brukes. Kravet fra oppdragsgivers side til oss var at huset skulle baseres på mest mulig grønn teknologi.

Vi bestemte at Henrik skulle være prosjektleder på grunnlag av at han har hatt flest programmeringsrelaterte fag av oss. Ekstern kommunikasjon med samarbeidspartnere i HDL og andre leverandører ble håndtert av alle, men kommunikasjonen ble loggført i Google Docs. Da vi kom til å arbeide så tett sammen gjennom hele prosjektet, vurderte vi å ta en smidig utviklingsmetode. Siden prosjektbeskrivelsen ikke var tydelig, vurderte vi det som ikke utenkelig at idéer og ønsker kom til å komme underveis. Vi ønsket da å ha mulighet for at oppdragsgiver skulle tilføye idéer og ønsker underveis i prosjektet. Vi var klar over at Scrum er et populært valg for programvareutviklingsprosjekt i dag, og at erfaring med dette kun er en fordel. Vi planla at sprintene skulle vare i én uke om gangen, og at vi skulle ha totalt åtte slike.

For å holde oversikt over produktkøen og for å organisere hver sprint brukte vi Trello. Før hver sprint hadde vi sprintplanlegging der vi definerte hva som skulle lages, og delte ut arbeidsoppgaver mellom oss. Til loggføring brukte vi Google Docs. Her skrev vi daglig logg over hva som hadde blitt gjort og hva som gjenstod på hver arbeidsoppgave. Daglig logg ble

oppsummert til ukentlig logg som finnes som vedlegg. Videre førte vi arbeidstimer i Google Sheets. Her ble det loggført timer pr. dag. Dette ble videre brukt til å gi en total, og kumulativ ukentlig- og månedlig oversikt over antallet timer.

1.9 Rapportens ordbruk og rapportorganisering

1.9.1 Ordbruk

Det tas forbehold om at leser har kunnskaper tilsvarende informatikkstudenter. Her forventes det at leseren har en gjennomsnittlig forståelse av ord og uttrykk som er ofte brukt innen IT. Dersom det skulle være noen uklare fagord, vil disse finnes under vedlegg A Terminologi i rapporten. Når det gjelder terminologi så er det kjent at uttrykk innenfor IT ofte stammer fra engelsk. Vi vil etter beste evne og i samråd med Språkrådet forsøke disse når det lar seg gjøre.

1.9.2 Rapportorganisering

Rapporten vil bestå av 6 kapitler, hvert med sine egne underkapitler. Når det gjelder oppbyggingen og formateringen av rapporten har vi tatt følgende avgjørelser:

1.0 Innledning

Omhandler og gir fullt prosjektgrunnlag. Her står det litt historikk, prosjektbeskrivelse, og avgrensning.

2.0 Forarbeid

Omhandler misforståelser i oppstartsfasen og en risikoanalyse for prosjektet.

3.0 Kravspesifikasjon

Omhandler krav til ferdigstilt produkt, både operasjonelle- og kvalitetsmessige krav. Disse vises primært gjennom use cases.

4.0 Design og arkitektur

Omhandler mobil- og serverapplikasjonens nåværende design og systemets arkitektur, teknologier og kommunikasjonsprotokoller.

5.0 Koding, kvalitetskontroll, implementering, og testing

Omhandler utviklingen, testfase for å garantere operasjonelle- og kvalitetsmessige krav til ferdigstilt produkt og implementeringen av løsningen på lokasjon.

6.0 Avslutning og konklusjon

Avslutter og konkluderer prosjektarbeid.

Vedlegg

Alle vedlegg.

2.0 Forarbeid

2.1 Oppstartsfasen

Innledningsvis i prosjektet så vi på hva som fantes av løsninger for å kommunisere med de ulike komponentene i smarthuset, og hvilke alternativer vi hadde for å komme frem til et tilfredsstillende produkt. Siden ingenting av tekniske løsninger var planlagt og fastsatt når vi kom inn i bildet, ble det brukt mye tid på å finne all teknisk informasjon om komponentene som er plukket ut. Denne tekniske informasjonen gjelder blant annet hva slags styring komponentene jobber under og om det er mulig å simulere dette via andre enheter, slik at det er mulig å gå utenom leverandørens proprietære systemer. Dette har vært svært viktig for å være sikre på at komponentene kan integreres og kommunisere med SHK-systemet.

Det har vært nødvendig å undersøke hva som trengs av tilleggskomponenter fra HDL for å koble disse systemene sammen. I tillegg har vi undersøkt hvilke sensorer som skal være med i prosjektet i forhold til hva som skal måles. Dette gjelder måling av utetemperatur, innetemperatur, og luftfuktighet rundt om i leilighetene (baderom, soverom, gang, kjøkken og stue).

Vi innstilte oss tidlig i prosjektfasen å benytte HDLs løsninger når det kom til styringssystem. HDL har konfigurasjonsverktøy under utvikling som gjør det enkelt å installere og konfigurere komponentene i boligen etter behov. Dette konfigurasjonsverktøyet består av en HDL IntelliCenter-server med innebygd logikk og programvare for å «programmere» komponentene med en dra-og-slipp-løsning for kodeblokker. Denne løsningen ville gjort vårt arbeid enkelt, faktisk så enkel at vi ikke lenger hadde hatt en relevant oppgave. Vi måtte derfor endre på oppgaven, og vi bestemte oss derfor å lage vår egen løsning bestående av en trådløs HDL-gateway (IP-modul) og en Raspberry Pi 3 Model B, med nok lagringskapasitet for å lagre serverløsningen vår, som vi kaller SHK-server. Med en ny oppgave fikk vi en tilfredsstillende arbeidsbelastning gjennom hele prosjektperioden.

Det skal nevnes at det har vært stor forvirring rundt prosjektet. I oppstartsfasen ble det en del misforståelser rundt hva vårt sluttprodukt skulle inneholde, være basert på, og ikke minst hvordan vi kom frem til det endelige produktet. Det ble anbefalt produkter som allerede fantes på markedet, men som senere viste seg å ikke være det vi trengte. Det som skapte mye av misforståelsene var et samspill mellom HDLs rådgivere sin mangel på innsikt i hvor stor arbeidsbelastning en bacheloroppgave skal ha gjennom hele prosessen og derfor anbefalte løsninger som reduserte den tekniske oppgaven til noen få dagsverk. Den andre delen var oss selv. Konseptet med smarthus og løsninger rundt dette var nytt for vår del, og vi manglet innsikt i hvilke løsninger som passet for vår del. Man kan på en måte si at de to partene, HDLs rådgivere og oss selv manglet innsikt i hverandres områder.

Etterhvert som tiden gikk, og mars måned kom, og forståelsen av den anbefalte løsningen økte innså vi at vi var på fullstendig feil kurs. Dette resulterte i en meget dramatisk situasjon der vi kontaktet prosessveileder, og forklarte hva som hadde skjedd. Det ble uklart hvorvidt vi faktisk hadde en relevant oppgave, for oppgaven var nå såpass redusert. Det ble innkalt til møte med blant andre studieprogramleder, Ivar Farup. Det planlagte møtet ble satt noen dager frem i tid og vi som gruppe hadde ingen tid å miste. Vi besøkte Ivar på kontoret der det blant annet ble drøftet muligheten til å bytte til en annen bacheloroppgave. Til slutt endte vi opp med å velge å utvikle et styringssystem fra bunnen av, det inkluderer en egen serverløsning og en mobilapplikasjon. Det var over en lengre periode daglig kontakt med HDL for å komme opp med idéer rundt systemet og hvordan en tredjepartsløsning kunne knyttes til HDLs systemer. For å kunne kommunisere med HDL sitt system er det et bredt spekter av spesifikke

kommandoer som må brukes. Vi ble invitert på et 2-dagers kurs av HDL, der de underviste i programmering opp mot HDL BusPro-systemet. Se vedlegg I Dialog med HDL og statusrapport en- og to for mer beskrivelse.

2.2 Risikoanalyse

Med en ny plan begynte vi å skrive om kravspesifikasjonen (3.0 Kravspesifikasjon) og skrev en ny risikoanalyse.

I en liten eller middels stor gruppe kan selv små hendelser få store konsekvenser. Det finnes mange generelle prosjektrisikoen som frafall og sykdom, men vi vil identifisere og liste opp de som er mer spesifikt knyttet til dette prosjektet. For å rangere risikoenes sannsynlighet og konsekvens har vi valgt å gruppere de i gradene liten, middels eller stor. Vi har identifisert følgende risikoer med følgende tiltak:

		Konsekvens		
		Liten	Middels	Stor
Sannsynlighet	Stor		2	1,7
	Middels			3
	Liten		5	4,6

Nr	Beskrivelse	Sannsynlighet	Konsekvens
1	Nødvendig fysisk utstyr i hus ikke på plass innen tidsfrist går ut	Stor	Stor
2	Endring i krav fra oppdragsgiver	Stor	Middels
3	Produkt ikke ferdig i tide	Middels	Stor
4	Server slutter å fungere	Liten	Stor
5	Ustabil nettværk grunnet veggtykkelse eller avstand fra aksesspunkter	Liten	Middels
6	Produktet fungerer ikke på lokasjon	Liten	Stor
7	Innkjøp av feil/ubrukelige komponenter	Stor	Stor

For å hindre eller eliminere mulighetene for de nevnte risikoene har vi identifisert følgende tiltak. Disse vil bidra til å hindre at nødvendig infrastruktur forfaller og i verste fall gjør at SHK-systemet slutter å fungere.

Nr	Tiltak
1	Gjøre oppdragsgiver oppmerksom på at nødvendig infrastruktur er kritisk for å kunne fullføre oppgaven.
2	Nye krav legges til i produktkø og implementeres i en kommende sprint.
3	Oppdragsgiver får et uferdig produkt. Må i verste fall forbedres av oppdragsgiver.
4	Gitt lang nok tid vil server slutte å fungere, dette fører til tap av kritisk infrastruktur. Kan hindres ved å ha kildekode lagres eksternt.
5	Planlegge plassering av HDL-brytere for optimal rekkevidde av signal.
6	Forhindres gjennom testing på lokasjon.
7	Forsikre oss om at komponentene vi bestiller er korrekte i henhold til fastsatte krav.

3.0 Kravspesifikasjon

3.1 Kravendring

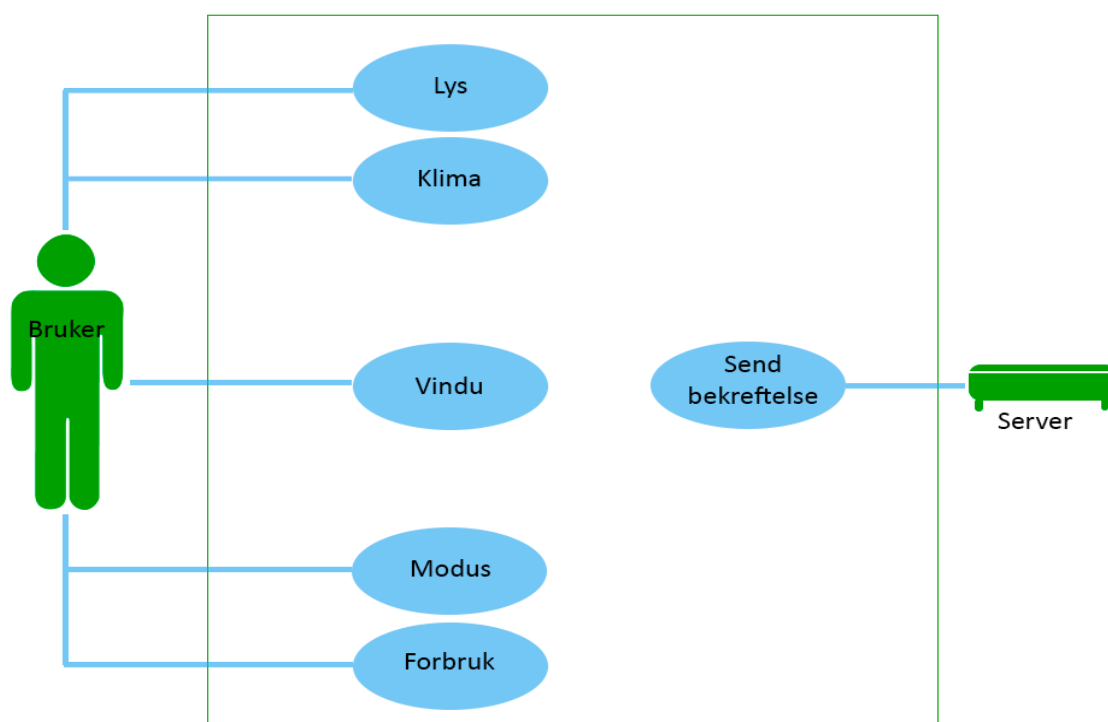
Som forklart tidligere (2.0 Forarbeid) endte vi opp med en ny oppgave. Kravspesifikasjonen som følger er en oppsummering produktkøen vi hadde i Scrum-prosessen, og er skrevet til dels retrospektivt, og underveis i utviklingen som vi og oppdragsgiver formulerte produktkøen.

3.2 Endelig kravspesifikasjon

3.2.1 Use case diagram for brukere og administrator

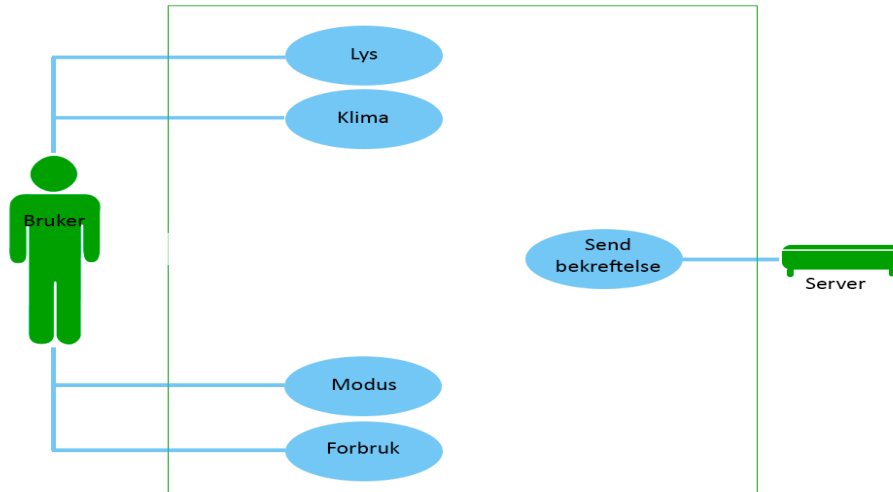
Boligen som SHK utvikles for skal ha tre brukere, i tillegg til en administratorkonto. Disse brukerne blir delt mellom de tre boenhetene, og i tillegg skal det være en administratorbruker. Funksjonaliteten skal variere avhengig av utstyr som er i boenheten, slik som takvinduene i leilighet 1.

Brukerne for hybelleilighet og leilighet 2 har derimot identisk funksjonalitet. Under vil vi beskrive brukernes funksjonalitet i mer detalj.



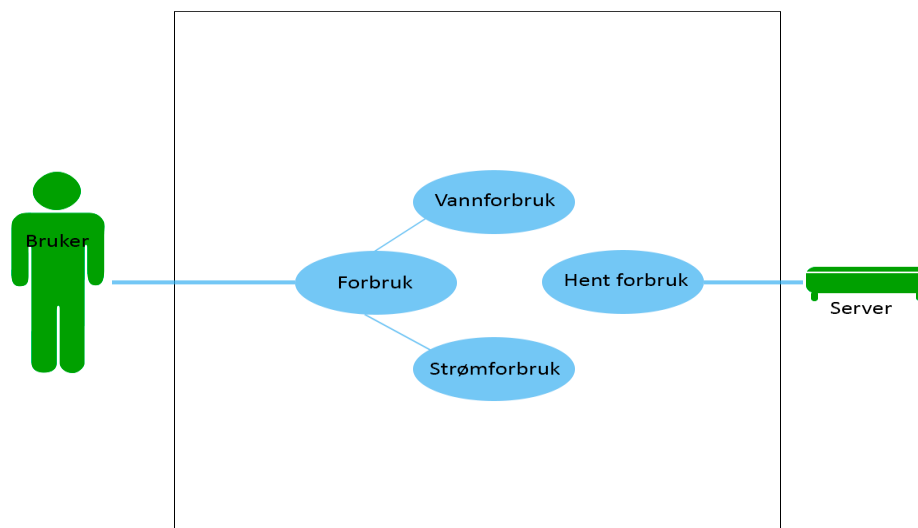
Figur 2 Use case for bruker i leilighet 1

I toppetasjen finnes det tre Velux takvinduer. Disse vinduene kan man styre via SHK-applikasjonen. Det er også denne funksjonaliteten som skiller denne brukeren fra de to andre. Av brukere finnes det tre stykker. Disse er knyttet direkte til hver enkelt boenhet. Eksempelvis kan leilighet1-bruker kun styre sitt eget utstyr som lys og styre varme i egne rom. I tillegg kan hver enkelt bruker kun få oversikt over sitt eget forbruk.



Figur 3 Use case for leilighet 2 og hybelleilighet

I Figur 3 har vi use case for leilighet 2 og hybelleilighet. Disse har samme funksjonalitet som leilighet 1, unntatt mulighet for å styre vinduer.



Figur 4 Use case for administrator

I Figur 4 ser vi at administrator bare har oversikt over boligens forbruk av strøm og vann. Under hver av disse kan administrator få full oversikt over hvor mye av enten vann eller strøm som boligen i sin helhet og hver boenhet for seg har brukt/konsumert i løpet av måneden.

3.2.2 Overordnet use case-beskrivelse

Use case	Lys
Aktør(er)	Leilighet1, Leilighet2 og hybel
Mål	Justere lyset i forskjellige soner i boligen.
Beskrivelse	Bruker kan velge rom og får da mulighet til å regulere lys enkeltvis eller samlet. Brukeren kan i tillegg bestemme dimmeverdier for «Minimum» og «Medium» dimmenivå.

Use case	Klima
Aktør(er)	Leilighet1, Leilighet2 og hybel
Mål	Regulere enten ventilasjonshastighet eller temperatur
Beskrivelse	Brukeren får her valg mellom ventilasjon eller temperatur

Use case	Ventilasjon
Aktør(er)	Leilighet1, Leilighet2 og hybel
Mål	Oppnå et godt inneklima ved hjelp av et brukerregulert ventilasjonssystem.
Beskrivelse	Fra Klima-menyen kan brukeren her styre det balanserte ventilasjonssystemet for å regulere hastigheten på anlegget.

Use case	Temperatur
Aktør(er)	Leilighet1, Leilighet2 og hybel
Mål	Overvåke og/eller regulere temperatur i temperatursoner
Beskrivelse	Fra Klima-menyen kan brukeren her overvåke temperatur i de forskjellige rommene i leiligheten og justere etter egne ønsker.

Use case	Vindu
Aktør(er)	Leilighet1
Mål	Åpning og lukking av vinduer og automatisert eller manuell utlufting
Beskrivelse	I denne menyen skal brukeren kunne velge å åpne eller lukke takvinduene. Alle tre takvinduene i boligen skal kunne styres gjennom applikasjonen. Hvert vindu har en maksimal åpningsvinkel som er valgt for at det ikke skal kunne renne vann inn. Dersom det begynner å regne mens ett eller flere av takvinduene er åpne vil de automatisk lukkes ved hjelp av regnsensorene som er integrert i vinduene.

Use case	Modus
Aktør(er)	Leilighet1, Leilighet2 og hybel
Mål	Sette leiligheten i en bestemt innstilling (modus)
Beskrivelse	Bruker kan sette leiligheten i enten «Dag»-, «Borte»-, «Ferie»- eller «Natt»-modus. Hver av disse modusene har (bruker)bestemte innstillinger for lys, temperatur, og ventilasjon.

Use case	Forbruk
Aktør(er)	Leilighet1, Leilighet2, hybel og administrator
Mål	Få oversikt over boligens forbruk av vann og strøm
Beskrivelse	Brukeren kan velge å se sitt eget strøm- og vannforbruk.

Use case	Vannmåling
Aktør(er)	Leilighet1, Leilighet2, hybel og administrator
Mål	Få oversikt over vannforbruk
Beskrivelse	Fra Forbruk-menyen. Varmtvannsforbruket til hver boenhet skal overvåkes og dette skal brukes til å regne ut strømforbruk per boenhet på oppvarmingen av vann.

Use case	Strøm
Aktør(er)	Leilighet1, Leilighet2, hybel og administrator
Mål	Få oversikt over strømforbruket for oppvarming av vann til hver boenhet.
Beskrivelse	Fra Forbruk-menyen. Strømforbruk knyttet til oppvarming av hver boenhet kan overvåkes. Her vil strømforbruket knyttes til hver enkelt måned, som gir brukeren en enkel oversikt over hvor mange kWh som er brukt i de aktuelle månedene.

Use case	Fargeinnstillinger
Aktør(er)	Leilighet1, Leilighet2, hybel
Mål	Endre bakgrunnsfarge.
Beskrivelse	Brukeren kan endre bakgrunnsfargen.

3.2.3 Detaljert use case-beskrivelse

Listet under er noen av grunnfunksjonene i SHK-applikasjonen. Det er viktig at det er god flyt og at dersom det oppstår en feil, så kan disse rettes opp i på en enkel og fornuftig måte uten for mange inngrep fra bruker.

Hver av disse use casene inneholder en aktør, som i alle tilfeller er minst én av brukerne. Vi har også med forhåndsbedingungen, som betyr at det er visse krav som er satt og som må innfris før hendelsesforløpet kan videreføres. Dette beskriver altså et ønskelig og feilfritt scenario. Feilsituasjoner er listet separat.

Use case	Lys
Aktør(er)	Leilighet1, Leilighet2 eller hybel
Forhåndsbedingungen	Bruker må være logget inn
Detaljert hendelsesforløp	<ol style="list-style-type: none"> 1. Bruker klikker på «SHK-app» 2. Bruker skriver inn sitt eget brukernavn og passord 3. Bruker trykker «Logg inn» 4. Bruker henvises til hovedskjerm 5. Bruker trykker på «lys» 6. Bruker velger ønsket rom 7. Bruker velger «Max» på ønsket lys
Feilsituasjoner	<ol style="list-style-type: none"> 1. «SHK-app» får ikke kontakt med server <ol style="list-style-type: none"> a) Restart applikasjon b) Sjekk nettverkstilkoblingen på Android-enheten. c) Gå rundt om i boligen for å se om bruker er utenfor rekkevidde. d) Restart server

	<ol style="list-style-type: none"> 2. Får ikke skrudd på ønsket lys <ol style="list-style-type: none"> a) Restart applikasjon
--	--

Use case	Klima
Aktør(er)	Leilighet1, Leilighet2 eller hybel
Forhåndsbedingungen	Bruker må være logget inn
Detaljert hendelsesforløp	<ol style="list-style-type: none"> 1. Bruker klikker på «SHK-app» 2. Bruker skriver inn sitt eget brukernavn og passord 3. Bruker trykker «Logg inn» 4. Bruker henvises til hovedskjerm 5. Bruker trykker «klima»-knapp 6. Bruker trykker «temperatur-knapp» 7. Bruker finner ønsket rom i listen 8. Bruker setter ønsket temperatur
Feilsituasjoner	<ol style="list-style-type: none"> 1. «SHK-app» får ikke kontakt med server <ol style="list-style-type: none"> a) Restart applikasjon b) Sjekk nettverkstilkoblingen på Android-enheten. c) Gå rundt om i boligen for å se om bruker er utenfor rekkevidde. d) Restart server 2. Får ikke regulert temperatur <ol style="list-style-type: none"> a) Restart applikasjon b) Se om det går an å regulere temperatur i andre soner

3.2.4 Kvalitetsmessige og operasjonelle krav

3.2.4.1 Et godt inneklima

En del av prosjektet var å kunne bidra til et bedre inneklima i boligen. Et godt inneklima er ingen triviell sak, og virkningen av et godt inneklima er dokumentert å ha positiv effekt på beboere. Virkningene av dette inkluderer redusert risiko for utvikling og plager knyttet til astma, allergier og andre overfølsomheter, og vil samtidig virke forebyggende. Personer og særlig de som da er overfølsomme kan oppleve irriterte luftveier, som kan føre til betennelse i slimhinner, og redusert eller økt slimproduksjon.

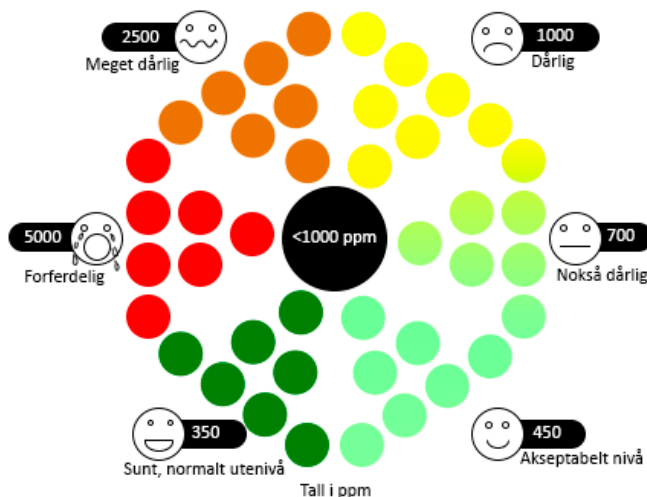
De viktigste faktorene som påvirker inneluften:

- Røyking
- Støv og mangel på tilstrekkelig renhold
- Ventilasjon og luftskifte
- Temperatur

God ventilasjon bidrar til å tilføre ren og frisk luft, og fjerner forurensinger i inneluften som for eksempel støv, fukt, avgasser. I tillegg kan høy temperatur bidra til at personer får konsentrasjonsvansker og kan bli særlig følsomme for forurensinger, fordi den høye temperaturen kan bidra til å tørke ut tårevæskene, som kan gi smerter og betennelser i øynene.³⁴

I Norge er høy luftfuktighet et større problem enn «tørr luft». Tørr luft gir som regel en følelse av irritasjon i slimhinner. Dette skyldes vanligvis luftforurensinger og da særlig svevestøv. Dersom det er høy luftfuktighet gir dette grobunn for muggvekst, som gir bedre vilkår for midd. Ideell luftfuktighet innendørs ligger på et nivå mellom 20-40 % relativ luftfuktighet.⁵

Andre viktige faktorer i et godt inneklima er nivået av gasser som CO₂ som kan føre til ubehag og redusert konsentrasjonsevne. CO₂ er en god indikator på hvor god luftkvaliteten innendørs er, siden høye nivåer indikerer innestengt luft med høy konsentrasjon av bakterier, virus, kjemiske avgasser og svevestøv. I mange land er det satt grenser for hvor høyt nivået kan være der mange mennesker oppholder seg, som eksempelvis skoler, og barnehager. Under er en illustrasjon:



Figur 5 Oversikt over inneklima

Som illustrert over (Figur 5); mellom 350-450 ppm (mg/m³) er helt normalt utendørsnivå av CO₂. Når nivået nærmer seg 700-1000 begynner plager som stivhet å forekomme. Ved nivåer fra 1000-2500 ppm begynner man å merke trøtthet, og videre opp mot 5000 ppm kan man få hodepine og få redusert konsentrasjonsevne.⁶

³ <http://www.naaf.no/no/inneklima/fakta-om-inneklima/>

⁴ <http://www.arbeidstilsynet.no/fakta.html?tid=229351#helseplager>

⁵ <http://www.inneklima.com/index.asp?key=RF>

⁶ <https://www.kane.co.uk/knowledge-centre/what-are-safe-levels-of-co-and-co2-in-rooms>

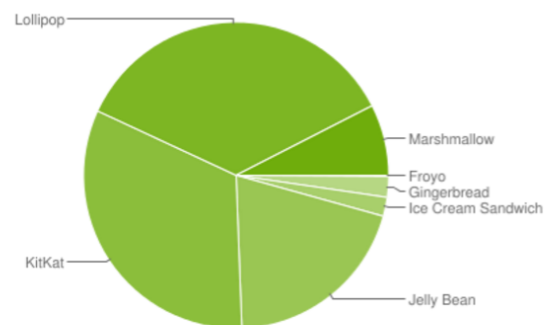
Siden det i denne boligen skal installeres ventilasjonsanlegg, temperatursensorer og luftfuktighetsmålere skal de to siste bidra til å bestemme når dette ventilasjonsanlegget slås på for å bidra til et godt inn klima.

3.2.4.2 Støtte

Vi vil utvikle applikasjonen med norsk tekstlig brukergrensesnitt. Applikasjonen vil bli utviklet for bruk på et Android-nettbrett eller smarttelefon og målet er at den skal kjøre stabilt uten nedetid. Den skal kommunisere med systemets egen Raspberry Pi, og det er på denne logikken skal utføres. Applikasjonen vil fungere som et grensesnitt for å overvåke og styre tilkoblede enheter.

Etttersom at to av boenhetene skal leies ut til enhver tid, ville vi at flest mulig skal ha mulighet til å bruke SHK-applikasjonen. Applikasjon skal være kompatibel med Android API Level 15 (Android 4.0.3), Ice Cream Sandwich MR1. Dette betyr at SHK-applikasjonen skal fungere på Android-enheter som kjører Android-operativsystem 4.0.3 eller nyere. Ved å legge API-kravet der, vil SHK-applikasjonen potensielt kunne kjøres på 97,7 % av Android-enheter⁷.

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.2%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.0%
4.1.x	Jelly Bean	16	7.2%
4.2.x		17	10.0%
4.3		18	2.9%
4.4	KitKat	19	32.5%
5.0	Lollipop	21	16.2%
5.1		22	19.4%
6.0	Marshmallow	23	7.5%



Data collected during a 7-day period ending on May 2, 2016.
Any versions with less than 0.1% distribution are not shown.

Figur 6 Android-distribusjon pr 2. mai 2016

3.2.4.3 Brukervennlighet og sikkerhet

Applikasjonen som skal brukes til å kommunisere med SHK-styringssystemet er en egenutviklet mobilapplikasjon. Det skal gjøres vurderinger og brukertester slik at det sørges for at programvaren blir så intuitiv og dermed kan brukes uten opplæring. For å sikre at applikasjonen er intuitiv og enkel å forstå har vi valgt å gå for ikoner fremfor knapper med tekst i hoved-, forbruk-, og klimameny, med unntak av «Modusvalg». Dette resulterer i en oversiktlig hovedmeny som er lett å lære og rask å bruke. Ikonene er designet for å være lett

⁷ <http://developer.android.com/about/dashboards/index.html>

forståelige og enkle å tyde. På Lys-sidene derimot har vi vurdert det som mer praktisk å ha tekst. Det skal ikke være nødvendig å ha tidligere erfaringer med styringssystem for å kunne bruke SHK-systemet.

Sikkerhet er viktig for å unngå og hindre uønsket adgang. Applikasjonen skal derfor være stengt for uvedkommende og beskyttes via passord og brukernavn. Disse verdiene lagres lokalt på serveren i en database. Passordene i databasen hashes, og i tillegg er innloggingen beskyttet mot SQL-injeksjoner.

I tilfeller der en bruker har glemt passordet sitt må brukeren finne sitt eget nødpassord. Dette nødpassordet er klistret fysisk på server, og må benyttes i tilfeller der brukeren har glemt sitt eget passord. Bruker må i slike tilfeller logge inn på vanlig vis, men med nødpassord. Når brukeren så skriver inn dette nødpassordet, blir det bedt om et nytt passord som overskriver det glemte passordet.

3.2.4.4 Pålitelighet

Applikasjonen skal kjøre stabilt så lenge kravene oppfylles. Det forutsettes at styringsenheten enten er koblet til boligens nettverk eller Internett. Hendelser som kan hindre stabil oppetid er nye utgivelser og uforutsette problemer. Hendelser som ikke kan forutsees kan endre måten vår løsning fungerer sammen med de enkelte komponentene i boligen. Et eksempel på dette kan være en programvareoppdatering for en uspesifisert komponent som påvirker kommunikasjonen mellom den gitte komponenten og styringssystemet i boligen. Dette kan medføre at bruker ikke kan styre denne komponenten, og i verste fall må programkode i SHK-applikasjonen/SHK-server skrives om.

3.2.4.5 Videre utvidelser

Siden oppdragsgiver er en privatperson og ikke en bedrift, vil det i utgangspunktet ikke være noen videre utgivelser. Dette er en ulempe, særlig med tanke på sikkerhetsoppdateringer eller dersom SHK av en eller annen grunn skulle slutte å fungere. Dersom videre utgivelser skal bli aktuelt må derfor tredjeparter bringes inn, og det er derfor svært viktig med dekkende dokumentasjon. Mer om videre utgivelser og mulige forbedringer kan leses om i kapittel 6.5 Videre arbeid.

Hvis oppdragsgiver eller senere eiere av boligen velger å installere nye komponenter vil det være mulig, men avansert. Endringer må gjøres i kildekoden til SHK-applikasjonen. Applikasjonen kompiles på nytt og eventuelle endringer på SHK-server fører til at den må kompiles og settes opp på nytt.. Avhengig av hva slags komponenter det er kan det integreres i styringssystemet. Dette forutsetter hovedsakelig at komponentene kommuniserer på lik måte som nåværende komponenter, eller via andre protokoller som støtter lik kommunikasjon. Videre utgivelser gjelder også levetiden på enkelte nåværende komponenter.

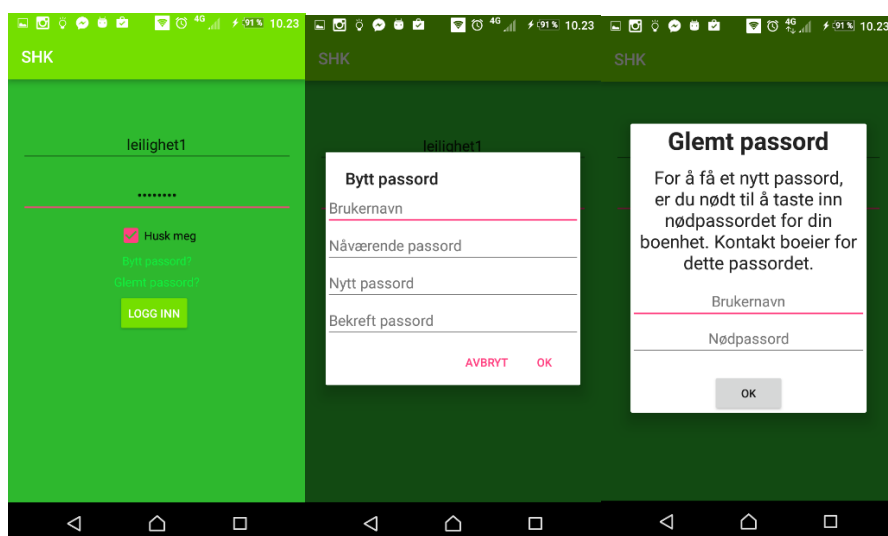
Enkle komponenter som for eksempel belysning og enkle digitale komponenter som persiener kan integreres, men krever endring/utvidelse av eksisterende kode. Mer kompliserte komponenter som flere ventilasjonssystemer eller ny varmepumpe vil kreve mer teknisk kompetanse. Hvordan komponentene kommuniserer med løsningen vår, om det gjøres på lik måte eller ikke, vil avgjøre hvordan integreringen vil være. For eksempel vil komponenter som kommuniserer på andre måter muligens trenge ekstra tilleggsenheter og sensorer for å kommunisere med eksterne systemer.

4.0 Design og arkitektur

Her vil vi først presentere hvordan brukergrensesnittet ble sendt ut, og i neste delkapittel(4.2 Design) vil vi forklare designet.

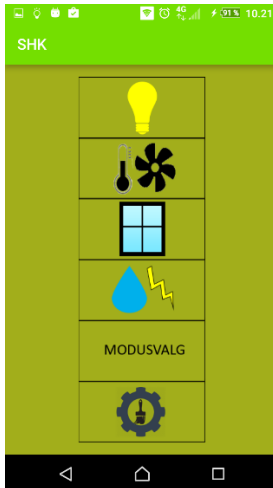
4.1 SHK-applikasjonens brukergrensesnitt

Det har vært høyt prioritert at SHK-applikasjonen skal være mest mulig brukervennlig. For vår del er det ønskelig at brukere ikke skal ha noen vanskeligheter med å bruke applikasjonen. For å skape et mest mulig intuitivt design har vi sett på lignende løsninger som finnes, og fått innspill fra venner og oppdragsgiver, samt utført brukertester (5.3.1 Brukertesting). Under har vi lagt ved noen skjermdumper fra applikasjonen som viser litt om hvordan selve applikasjonen ser ut under bruk og hvilken funksjon de forskjellige knappene har.



Figur 7 Innloggingsmeny og passordbehandling

Applikasjonen er designet for å ha et enkelt, ryddig, og intuitivt design uten for mange undermenyer som hindrer raske og effektive operasjoner. Det ble fokusert på at det skulle være maksimalt tre trykk for å utføre en operasjon. Når brukeren har åpnet applikasjonen, vil denne innloggingsmenyen vises. Fra denne menyen kan brukeren logge inn, bytte passord eller få nytt, dersom passordet er glemt. For å logge inn gjøres dette ved benytte brukernavn og passord for sin boenhet. Når innloggingen (Figur 7) er fullført blir brukeren henvist til hovedskjermen (Figur 8). Fra hovedmenyen kan brukeren med ett trykk komme inn på lys, klima, vindu, forbruk, modus, og fargeinnstillinger.

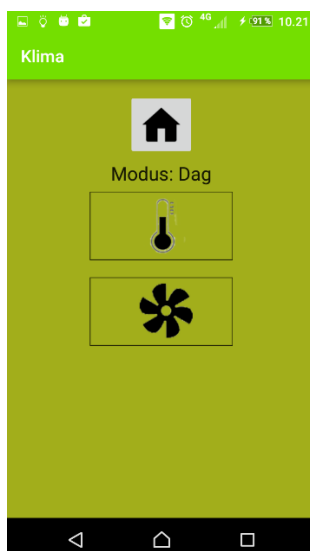


Figur 8 Hovedskjerm for leilighet 1



Figur 9 Lyssoner og lysoversikt på kjøkken

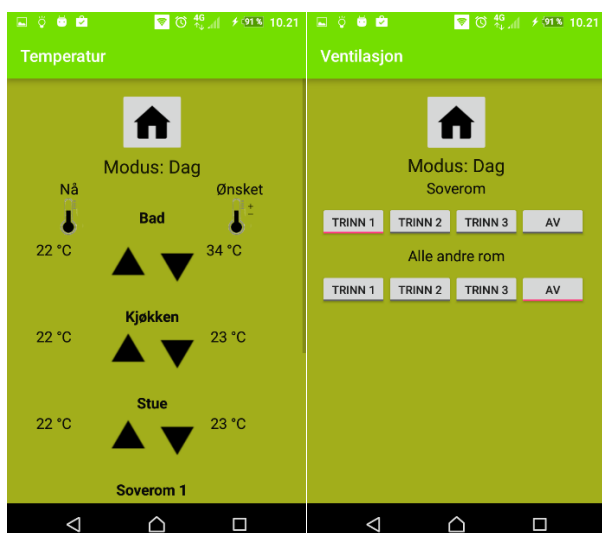
Under lys (Figur 9) vises en oversikt over alle rom som har lys som er koblet til BusPro-nettverket, og dermed er styrbare fra SHK. Det er flere rom i boligen som har rom der lysene ikke er koblet til dette. Disse rommene er bodet der det er én lyskilde, og som derfor ikke er koblet til nettverket. Velger brukeren som bor i leilighet 1 kjøkken, så vises denne listen over lys. Alle sider (med unntak av hovedmenyen) har en hurtigknapp for å returnere til hovedmenyen på bare ett klikk. Dette gjør at brukere ikke må trykke på enhetens egen tilbakeknapp for å returnere dit.



Figur 10 Klimameny

Under klima (Figur 10) har brukeren to valg. Velger brukeren ventilasjon gis det mulighet til å velge hastighet på viftene i anlegget. Velger brukeren temperatur får brukeren opp en oversikt over aktuelle varmesoner som vist i Figur 11. Nåværende temperatur for et gitt rom vises i kolonnen til venstre. Basert på tilbakemeldingene vi fikk under brukertesting valgte vi å legge til teksten «Nå» og «Ønsket» over termometerikonene Dette gjorde vi på grunn av at brukere hadde vanskeligheter med å vite hvilken temperatur som representerte «nåværende romtemperatur» og hvilken temperatur de kan stille.

Brukeren kan regulere temperatur ved å trykke på knappene. Settpunktet er plassert i høyre kolonne. Eksempelvis, er temperaturen i én gitt varmesone 28 grader Celsius vil dette stå i venstre kolonne, mens dersom bruker ønsker å endre temperatur til 30 grader Celsius vil bruker trykke to ganger på «oppknappen». Settpunktet vil da vise 30 grader Celsius.



Figur 11 Temperaturoversikt og ventilasjon



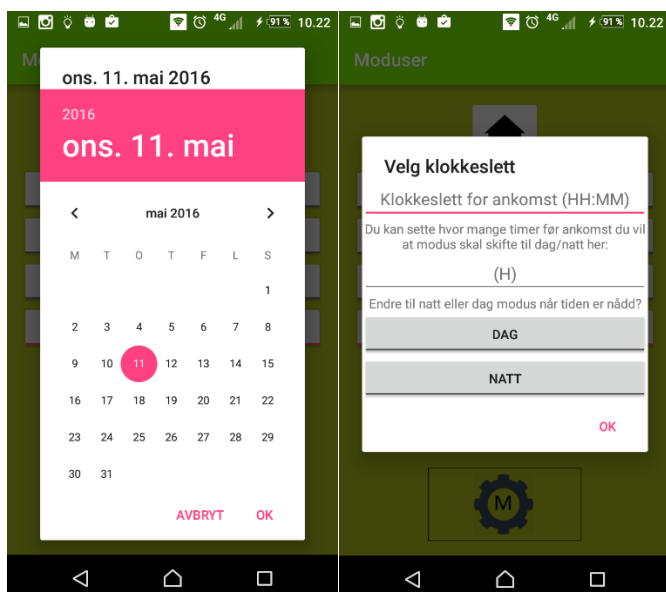
Figur 12 Modusvalg

Under «Modusvalg» (Figur 12) har brukeren fire valg. Her er det mulig for brukere å sette boenheten sin i moduser, avhengig av hva brukeren ønsker. Skal for eksempel brukeren i leilighet 1 på ferie, kan brukeren via SHK-applikasjonen velge «Ferie»-modus. Når denne velges vil flere forhåndsbestemte hendelser skje. Når denne modusen er valgt kan temperaturen senkes, og alt lys slås av. Bruker kan sette innstillinger selv, men SHK leveres med «fabrikkinnstillinger». Brukere kan gjøre at leiligheten endrer modus automatisk, avhengig av tidspunkt på døgnet. Dette er praktisk i forbindelse med «natt»- og «dag»-modus.



Figur 13 Innstillinger for modus

Velger for eksempel bruker at «natt»-modus skal starte kl 23, vil leiligheten settes i «natt» hver dag kl 23, og da kan eksempelvis lys slås av og temperaturen reduseres. I tillegg til disse to modusene, kan bruker sette innstillinger for «ferie»-modus. Velges denne modusen gis det mulighet til å fortelle systemet når brukeren forventes å være tilbake, dato og klokkeslett. Ønsker brukeren at systemet skal begynne oppvarming av leiligheten, kan dette også gjøres.



Figur 14 Innstillinger av feriemodus

4.2 Design

Til nå har vi sett på hvordan brukergrensesnittet for SHK-applikasjonen er. Nå vil vi forklare kommunikasjonskommandoene som brukes til å kommunisere med HDL. I kapittel 4.2.1 Oppbygging av kommando skal vi forklare designet av funksjoner i mobilapplikasjonen. I kapittel 4.2.2 SHK-applikasjon vil vi forklare designet av styringsfunksjoner. For flere eksempler og mer informasjon angående kommandoforamt og oppbygging av datapakker, se vedlegg B Kommando.

4.2.1 Oppbygging av kommando

For at pakkene som sendes fra server skal kunne tolkes riktig av HDL, må de bygges på en forhåndsdefinert måte. Pakkene må sendes som en UDP-datapakke, og sendes til en spesifikk port som ser satt av HDL. Ettersom det sendes via UDP foregår kommunikasjonen som «Point to Point», og pakker som ikke kommer frem vil forkastes. Kommunikasjonen mellom server og HDL er kablet slik at sannsynligheten for at pakker blir borte blir sett på som svært liten, med forutsetning at trafikken på systemet er lav. Vi ser på kommunikasjonen mellom mobilapplikasjonen og serveren som en viktigere del av prosessen for at pakker skal leveres. Ettersom denne kommunikasjonen foregår over TCP, tilfredsstiller kommunikasjonen mellom applikasjonen og HDL-systemets våre krav. I UDP-kommunikasjon er det heller ikke et krav om at mottakeren av pakken sender et svar tilbake til senderen, men dette foregår ved at pakker som mottas av HDLs IP-modul kringkastes til alle enheter som lytter etter meldinger. Dette inkluderer også serveren som er tilkoblet IP-modulen. Enhetene som kan ta imot disse pakkene og kan kjøre funksjonene som den blir bedt om, kringkaster også et resultat tilbake til alle enhetene som lytter. Pakkene som sendes til HDL startet alltid med det samme innhold i indeks 1-15, dette inkluderer IP-adresse, «HDLMIRACLE», og en lederkode. Hvordan en hel pakke ser ut kan sees i vedlegg B Kommandob kommando. Tabellen er et utsnitt av en typisk pakke som sendes.

Indeks	Bemerkelse	Bits	Omfang
16	Størrelsen på datapakken	8	11-78
17	Subnet-ID til sender	8	0-254
18	Enhets-ID til sender	8	0-254
19	Enhetsstype til sender	16	0-65535
20	Kommando	16	0-65535
21	Subnet-ID til mottaker	8	0-255
22	Enhets-ID til mottaker	8	0-255
23	Data	0-536	
24	CRC	16	0-65535

Tabellen under forklarer hver indeks i foregående tabell.

Indeks	Forklaring
16	8 bits som indikerer størrelsen på data pakken. Denne skal kalkuleres med seg selv og CRC. Minimum størrelse er 11 (da er innholdet i indeks 23, data, lik 0). Maksimal størrelse er 78
17-18	Til sammen 16 bits som inneholder subnet ID og enhets ID til serveren vi sender fra. Disse plassene er mer tilpasset for sending av kommandoer internt i HDL systemet, slik at disse verdiene er statisk satt på serveren.
19	16 bits som inneholder enhets typen til senderen, eller i dette tilfelle server. Dette feltet er også mer tilpasset for sending av kommandoer internt i HDL systemet. Enhets type er derfor også statisk satt på serveren.
20	Kommando som skal utføres av mottakerenheten.
21	Mottakerenhetens subnet-ID
22	Mottakerenhetens enhets-ID
23	Tilleggsdata som sendes med. Denne varierer ut ifra hva slags enhet det sendes til og hva slags kommando som skal utføres av enheten.
24	CRC som kalkuleres ut ifra innholdet på pakken. Er ikke CRC-en korrekt, vil ikke pakken prosesseres.

Merk: Indekser med mer enn én byte i oversikten over, vil ha mer enn en rad med innhold i disse eksemplene, ettersom hver rad tilsvarer en byte.

Her kommer et eksempel på en kommando hvor brukeren endrer på temperatur i varmestyringskontrolleren, kanal 4 (soverom):

Indeks	Innhold	Bemerkelse
17	21	Datalengde
18	1	Subnet-ID til sender
19	254	Enhets-ID til sender
20	254	Enhetsstype til sender
21	255	Enhetsstype til sender
22	28	Kommando
23	92	Kommando
24	1	Subnet ID til mottaker
25	12	Enhets ID til mottaker Varmestyringskontrolleren
26	4	Data – kanalen på varmestyringskontrolleren
27	0	Data - Nåværende arbeidstype Satt til 0 som er «Heating». Andre verdier er: 1 = Cooling 2 = Heating Power Output 3 = Cooling Power Output
28	0	Data - Temperaturtype Satt til 0 for Celsius Andre verdier er: 1 = Fahrenheit
29	2	Data – Modus kanalen er satt til Satt til 2 som vi har definert for dag. Andre verdier: 1 = Ferie 3 = Natte 4 = Borte
30	14	Temperatur ferie
31	22	Temperatur dag Temperaturen det har blitt satt til
32	18	Temperatur natte
33	16	Temperatur borte
34	1	Status på ventilene Satt til 1 for på Andre verdier er: 0 = Av Om den skal være av eller på regnes ved om ønsket temperatur er mindre enn nåværende temperatur i varmesonen.
35	15	Hvor lenge ventilen skal stå på før ventilene stenges
36	74	CRC
37	223	CRC

For flere eksempler og mer informasjon angående kommandoformat og oppbygging datapakker se vedlegg B Kommando. I kapittel 4.2.2 SHK-applikasjon skal vi forklare designet av funksjoner i mobilapplikasjonen.

4.2.2 SHK-applikasjon

Applikasjonen har støtte for tre forskjellige brukere, som er knyttet til hver sin boenhet. Hver av disse har forskjellige hovedsider og rom. Alle brukere får opp den samme innloggingsskjermen. Det er først når brukeren logger inn med brukernavnet sitt det blir valgt hvilken hovedside brukeren skal få opp. Brukeren vil bare ha tilgang til å styre rommene i sin leilighet. Hver klasse/side utenom hovedmenyene har en hjemknapp som kan brukes for å komme til hovedmenyen med ett trykk. Toppleiligheten var den mest omfattende og den vi startet å arbeide med først. Toppleiligheten har fem ulike knapper på sin hovedmeny. Disse er lys, klima, vindu, forbruk og moduser. På grunn av lite tid til selve utviklingsprosessen ble det noe kodeduplisering. Dette måtte vi gjøre for å få tid til å bli ferdig innen fristen, men det er noen metoder som støtter inndata fra flere knapper uten å bruke duplisering.

4.2.2.1 Innlogging

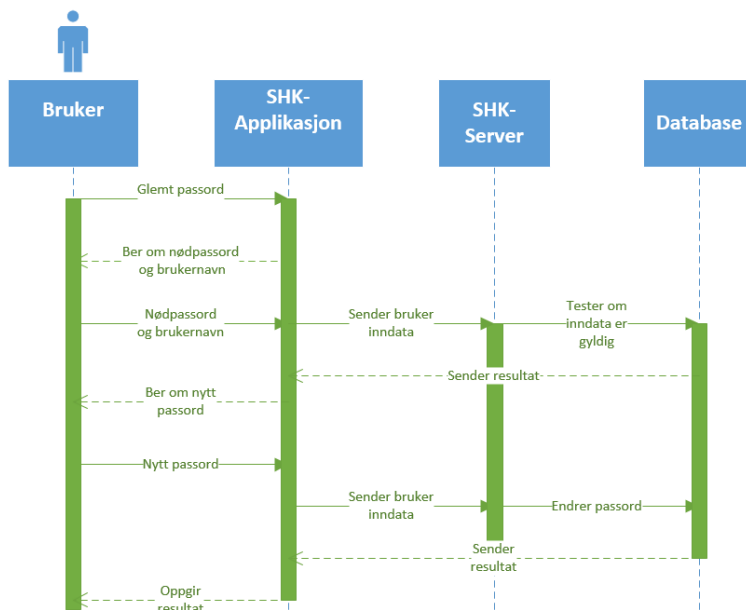
Fra innloggingsskjermen kan brukeren logge inn, endre passord eller resette et glemt passord. Brukeren kan også huke av for «Husk meg» som gjør at brukernavn og passord blir lagret i applikasjonen. Alle metodene på innloggingsskjermen har robusthet ved at den sjekker om det går for lang tid under pålogging, endring av passord og glemt passord. Det er denne klassen som tar seg av all tilkobling til server og bestemmer hvilke klasser som er tilgjengelig for hvilke brukere ved å sende de til riktig hovedmeny.

«Endre passord» sjekker om kombinasjon av brukernavn og gammelt passord er korrekt og gir tilbakemelding til bruker på dette. Metoden ber om at det nye passordet skrives inn to ganger, og har sjekk for at disse er like. I tillegg finnes det en feilmelding om det er problemer med tilkobling til server.

I starten av onCreate prøver applikasjonen å opprette tilkobling til server. Om dette ikke er mulig kommer det opp en feilmelding. Hvis brukeren prøver å logge inn på nytt, vil applikasjonen også prøve å koble til server på nytt. Om applikasjonen ikke har oppnådd eller ikke lenger har kontakt med server når brukeren trykker «Logg inn», vil det komme en feilmelding som ber brukeren prøve igjen. I det «Logg inn»-knappen blir trykt igjen prøver applikasjonen å koble til server på nytt. Det neste innloggingsforsøket kommer til å fungere om tilkobling ble oppnådd. Dersom kontakten enda ikke etableres, vil feilmelding fortsette å vises.

«Husk meg» er en avkrysningsrute som lagrer innloggingsinformasjonen, dersom brukeren krysset av denne. Hvis brukeren har krysset av «Husk meg» ved tidligere innlogging, vil brukernavn og passord være lagret i applikasjonen. Neste gang brukeren åpner applikasjonen, slipper brukeren å skrive inn dette på nytt. Dette gjøres ved å lagre brukernavn og passord i SharedPreferences-variable, samtidig som avkrysningsruten settes til true. Informasjonen hentet fra brukerinndata eller fra «Husk meg» vil så bli sendt til server og dersom brukernavn og passord samsvarer med det som står i databasen vil server respondere med et ID-nummer som applikasjonen bruker til å bestemme hvilken hovedmenyside som skal lastes inn.

Om bruker velger «Glemt passord» (se Figur 15) henvises det til en meny som gir informasjon og lar bruker logge inn med brukernavn og nødpassord. Dette ble videre forklart i 3.2.4.3 Brukervennlighet og sikkerhet. Server kommer så til å prøve å logge inn med disse verdiene. Først vil den automatisk prøve å sjekke om brukernavn og passord samsvarer med brukerens vanlige passord, om dette feiler vil den automatisk prøve nødpassordet. Om nødpassordet er korrekt blir brukeren henvist til innloggingsskjermen i applikasjonen, og får mulighet til å taste inn nytt passord og bekrefte dette. Er dette passordet gyldig sendes det til serveren og lagres i databasen. Bruker kan så logge inn med nytt passord.



Figur 15 Systemsekvensdiagram for "Glemt passord"

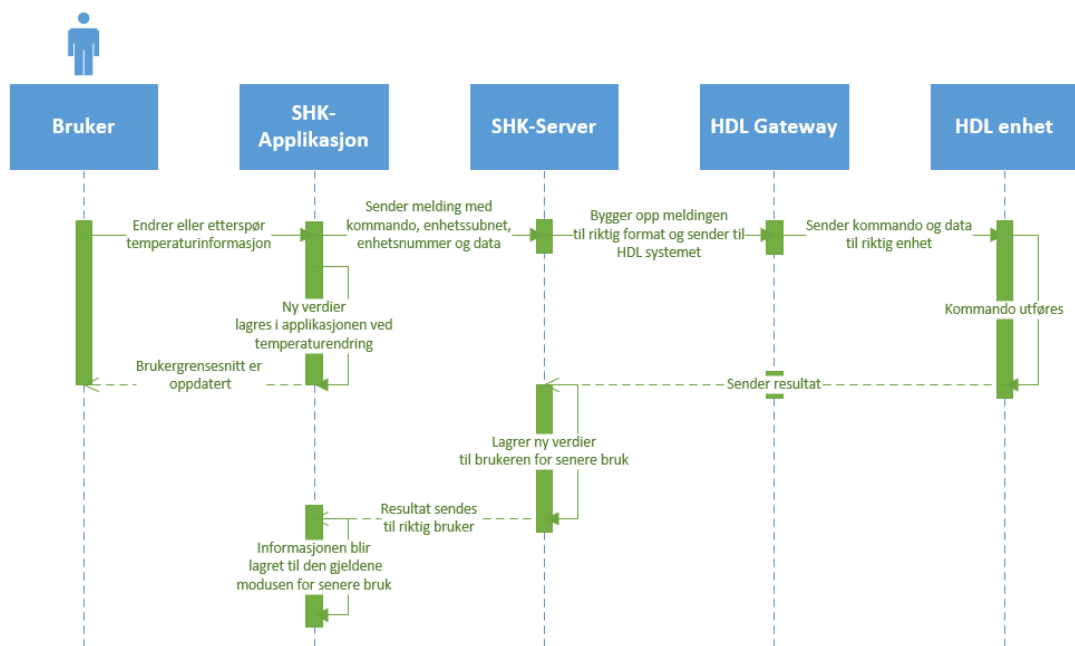
4.2.2.2 Lys

Under «Lys» er hvert rom eller sone delt opp i klasser. Disse klassene har egne Java- og XML-filer. I XML-filene er det ToggleButtons som fungerer som knappene for å justere lysnivå. Disse fungerer som en av/på bryter og markeres om den er aktivert. Her var det viktig at vi la inn robusthet slik at om for eksempel «Minimum»-bryteren er aktivert og brukeren bytter til «Maksimum», så skal «Minimum»-bryteren skru seg automatisk av ved å ikke lenger være markert. Dette løste vi ved hjelp av Boolean og ved hjelp av den innebygde Java-metoden for å sette ToggleButton til av. En metode, sendText(...), er den som sender strengen til server om hva som faktisk skal gjøres med lyset. Alle verdier brukeren setter lysene til å være, vil lagres via SharedPreferences til den gjeldende modusen huset er satt til. Neste gang brukeren skifter tilbake til denne modusen, eller kommer tilbake til denne lysinnstillingssiden, vil de lagrede verdiene avgjøre hva lysene er satt til. Brukeren kan i tillegg bestemme dimmeverdier for «Minimum» og «Medium» dimmenivå.

4.2.2.3 Temperatur

Hver leilighet er delt inn i temperatursoner, det vil si for eksempel én sone for stue, én for soverom og én kjøkken. Verdiene for alle temperatursoner som er hentet fra server er lagret via SharedPreferences slik at selv om brukeren forlater temperatursiden, vil applikasjonen alltid ha de sist mottatte verdiene lagret og klare til å vises på skjerm. Når temperaturen skal endres opp eller ned må variablene for nåværende temperatur og for hver modus sendes tilbake til server. I et slik tilfelle blir temperaturen som er satt for husets nåværende modus endret opp eller ned med én grad Celsius. Her er det mye kodeduplisering for hver enkelt knapp og alt som må skje avhenger av hvilken modus leiligheten er i.

For hvert rom må nåværende temperatur og valgt temperatur for alle fire moduser hentes fra server samtidig. Det er slik kommandoen fungerer, og dette medfører at kommunikasjon er mer kompleks og det øker mengden kode betraktelig.



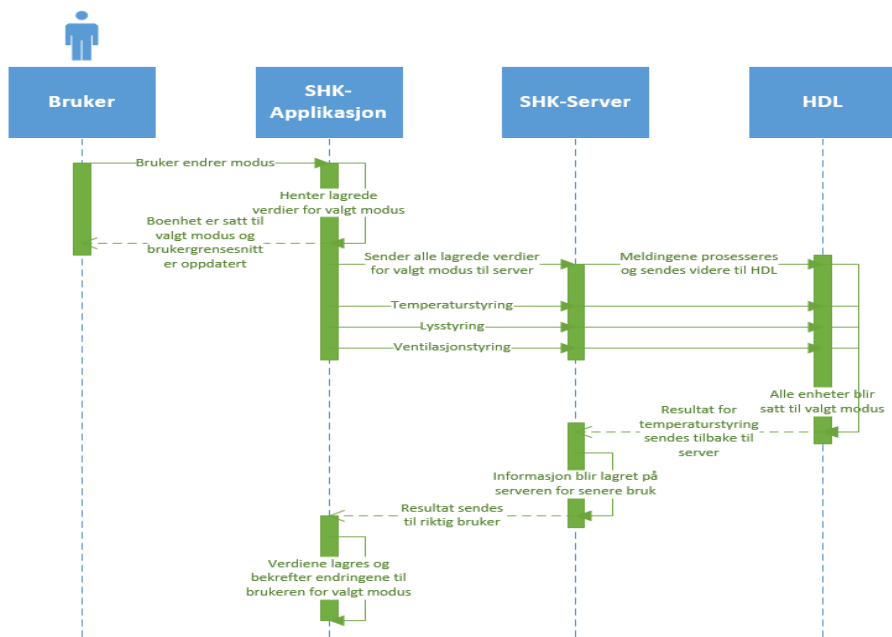
Figur 16 Endring eller etterspørsel av temperatur

I Figur 16 illustreres en handling der en bruker ønsker å endre eller etterspørre temperatur i en varmesone.

4.2.2.4 Moduser

Huset har fire forskjellige moduser. Når en bruker skifter modus, vil alle de lagrede innstillinger for temperatur, ventilasjon og lys for den valgte modusen settes. Alle disse verdiene sendes til server slik at huset settes til denne modusens verdier. Applikasjonen kan stilles inn til å automatisk skifte huset mellom dag- og nattmodus ved et gitt klokkeslett.

Feriemodus er litt forskjellig fra det andre modusene. Når du aktiverer denne modusen får du mulighet til å velge dato og klokkeslett for retur fra ferie. Slik at huset kan gå i valgt modus noen timer før du kommer hjem og dermed for eksempel øke innetemperaturen til ønsket nivå.



Figur 17 Endring av modus

I Figur 17 endrer bruker modus og verdier for denne modusen blir lagret på server og applikasjon. Meldingene prosesseres og sendes videre til HDL-systemet og leiligheten blir satt til valgt modus. Resultat blir returnert til server og applikasjon og dermed lagret på nytt.

4.2.2.5 Ventilasjon

I ventilasjonsstyringen har brukeren fire knapper per sone. Trinn en, to, tre eller av. Disse fungerer slik at trinn en og to er separate, men trinn tre aktiverer både trinn en og to. Dette er basert på hvordan reléene som skal styres fungerer. Her var det også viktig at knappene hadde relasjoner mellom hverandre. Om en knapp er aktivert og du bytter til et annet trinn, må den første skrus av. Dette må selvsagt gjøres både på applikasjonen og på det faktiske ventilasjonssystemet. Tankegangen bak dette er nesten lik den som ble brukt for lysstyringen. Den er fortsatt annerledes siden trinn tre må ta hensyn til om noen av det andre trinnene er aktivert, og så aktivere den som ikke er det. Eventuelt skru av alt om trinn tre allerede er aktivert.

4.2.2.6 Vindu

I vindusstyringen klarte vi dessverre ikke oppnå den robustheten vi selv ønsket. Nesten alt fungerer som ønsket. Metodene for åpning, lukking, og stopping av vinduet som er i bevegelse er nesten like. Hvis vinduet allerede er lukket skjer det ingenting.

Vi regnet ut hvor lang tid vinduet bruker å åpne fra 0% (lukket) til 100% (åpnet). Metoden som styrer alt dette oppdaterer applikasjonen kontinuerlig med hvor mye vinduet er åpnet. Når lukkingen starter blir tråden kjørt, og en metode som kontinuerlig oppdaterer bruker med status (åpningsprosent) på lukkingen blir kjørt. Om brukeren trykker «Stopp», stopper vinduet og åpningsprosent for vinduet vises på skjermen og lagres i applikasjonen. Problemet vi hadde med robusthet var at vi ikke klarte få støtte for at brukeren trykker «Lukke» når vinduet åpnes. Dermed kommer vinduet til å lukkes, men det som vises på skjermen blir ikke oppdatert riktig ettersom tråden fortsetter å gå til den er ferdig eller «Stopp»-knappen blir trykket. Problemet

her var derfor å få stoppet riktig tråd og fortsatt ha alle andre tråder kjørende. Applikasjonen fungerer derimot som planlagt om bruker først trykker «Stopp», og så velger om vinduet skal åpnes eller lukkes. Vi kunne løst dette mye enklere om vi hadde brukt en metode for hvert vindu, dermed hadde det vært mindre å ta hensyn til i koden og enklere å løse dette problemet. Vi valgte derimot å heller bruke denne metoden som håndterer inndata fra alle tre vinduene.

4.2.3 SHK-server

SHK-server er leddet mellom SHK-applikasjonen, databasen, Z-Wave-nettverket og HDL-systemet. Serveren er utviklet for å fjerne mye av trafikken på serveren og at brukerne skal kunne forholde seg til ett knutepunkt for sending og innhenting av informasjon. En server ga oss også mulighet til å lagre midlertidig informasjon for alle brukere på ett sted. Den kan hente informasjon fra målere, og sensorer i gitte tidsintervaller. Dette gjorde det mulig for oss å lage ulik «smart» tidsurfunksjonalitet uten at en bruker aktivt må bruke mobilapplikasjonen.

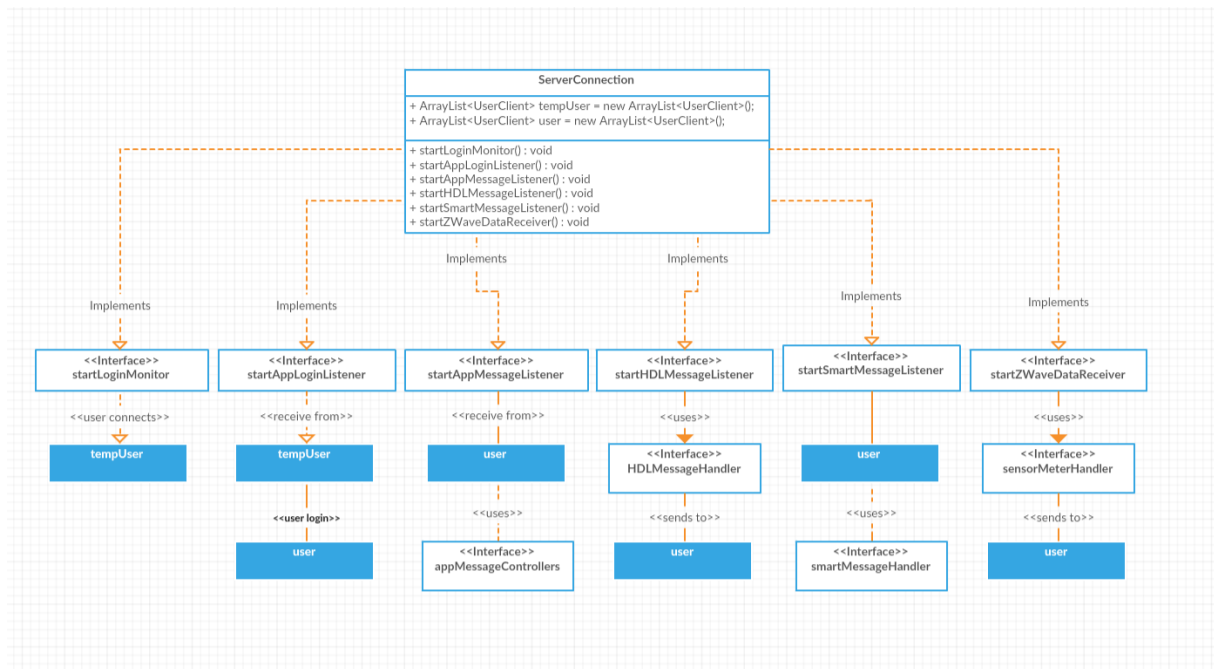
Hovedsakelig tar serveren av seg følgende funksjonalitet:

1. Håndtering av forskjellige brukere. Dette innebærer å akseptere brukere som tilkobles server og gi de tilgang til riktig bruker når de logger på.
2. Bygger opp meldingene mottatt fra applikasjonen slik at de følger riktig format og kan bli prosessert av en HDL-komponent.
3. Mottar meldinger fra HDL-komponenter, filtrerer de, og sender svar tilbake til riktig bruker som har etterspurt informasjon fra HDL-komponenten.
4. Starter forskjellige tidsur som har følgende funksjonalitet:
 - Brukeren kan bestemme når man kommer hjem fra ferie, slik at boenheten til brukeren kan ha et temperert og godt inn klima til brukeren kommer tilbake. I denne funksjonaliteten kan det spesifisere i hvor god tid før ankomst huset skal endre modus om det skal settes til dag- eller nattmodus. Når tiden er nådd vil en melding sendes til HDL-systemet som setter huset i riktig modus.
 - Brukeren kan selv bestemme når på dagen leiligheten skal settes til dag eller natt. Tiden spesifisert av brukeren vil gjelde for hver dag i ubegrenset tid fremover. Den tar også høyde for sommertid og vintertid. Når disse tidene nåes, vil huset endre modus til riktig modus. Dette gjelder så lenge modusen ikke er satt til ferie eller borte, da vil den forbli i disse modusene.
5. Henter ut data fra sensorer og målere fra Domoticz sitt grensesnitt.

4.3.2.1 Serverfunksjonalitet, melding- og påloggingshåndtering

I dette delkapittel skal funksjonalitetene som nevnt over, og hvilke metoder som gir serveren disse funksjonalitetene, forklares mer.

Når serveren startes, oppretter den håndteringsmetodene *startLoginMonitor()*, *startAppLoginListener()*, *startHDLMessageListener()*, *startSmartMessageListener()*, og *startZWaveDataReceiver()* i *ServerConnection*-klassen. Hver av disse metodene starter hver sin tråd og arbeider uavhengig av hverandre. Ut ifra figuren nedenfor (Figur 18) kan vi se hvordan de ulike håndteringsmetodene i klassen *ServerConnection* fungerer og hvordan disse forholder seg til brukeren.



Figur 18 Håndteringsmetode i ServerConnection

Metoden *startLoginMonitor()* tar alle brukere som kobler seg til serveren og plasserer de i en midlertidig brukerliste ved navn «tempUser».

Metoden *startAppLoginListener()* itererer igjennom hver bruker som ligger i «tempUser», og ser om en bruker har sendt meldinger til serveren. Hvis det finnes en melding, prosesseres den. Denne metoden gir brukeren funksjonalitet selv om man ikke er pålogget. Når en bruker kun er tilkoblet, vil brukeren ha tilgang til funksjonalitetene «Bytt passord», «Glemt passord», og pålogging. Disse funksjonalitetene er beskrevet mer i kapittelet om SHK-applikasjonen(4.2.2 SHK-applikasjon). En bruker blir værende i denne listen til man har logget på eller mistet tilkoblingen til server. Hvis brukeren har blitt frakoblet serveren vil brukeren bli fjernet fra listen. Dette gjøres ved at serveren prøver å sende meldinger til påkoblede brukere i gitte tidsintervaller. Videre, når en bruker logger seg på, vil brukeren bli overført til brukerlisten «user» og slettet fra «tempUser»-listen.

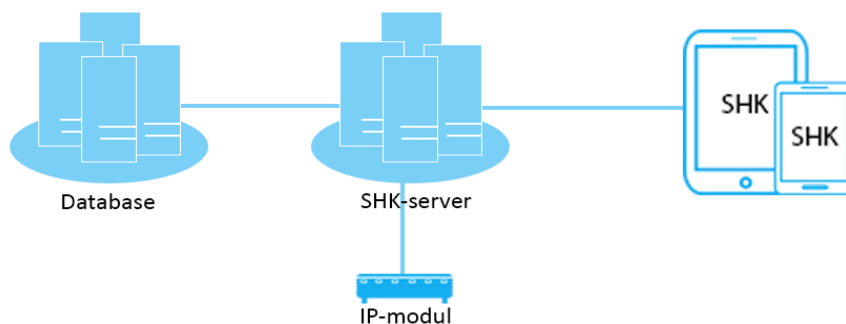
Det er metoden *startAppMessageListener()* som itererer igjennom hver bruker fra «user»-listen og prosesserer meldingene til brukere som er logget inn. Denne metoden tar imot meldinger fra brukere som innebærer kommandoer til smarthussystemet, henting av sensor- og målerdata, og iverksettelse av tidsur. Hvis en bruker sender en melding med en spesifikk kommando hvor brukeren forventer å få svar tilbake, vil disse kommandoene i tillegg til den brukerens id, lagres i en meldingsliste. Denne metoden fjerner også brukere som er utilgjengelige etter en viss tid ved å prøve å sende en melding. En bruker har tilgang til denne metoden helt til brukeren frakobles fra serveren.

Metoden *startHDLMessageListener()* håndterer meldinger fra HDL-systemet. Ettersom HDL sender mange meldinger som ikke inneholder noen spesifikk informasjon, må disse filtreres. Meldinger som inneholder informasjon som en bruker har etterspurt, prosesseres, og sendes til riktig bruker. Dette gjøres ved at metoden går igjennom hver bruker sin meldingsliste og ser om det finnes en melding med den kommandoen som metoden har mottatt fra HDL, for deretter å sende dataen til den brukeren.

Metoden `startSmartMessageListener()` itererer igjennom hver bruker sin «smartMessage»-liste. Det blir lagt meldinger i denne listen når én brukers tidsur har nådd en bestemt tid, for eksempel at tiden for når dagmodus skal starte, er nådd. Disse meldingene inneholder verdier som skal sendes til HDL-systemet og inneholder verdier som endrer modus boenheten til den brukeren skal settes i. På grunn av hvordan meldingene for gjøre dette er bygget opp av HDL, må disse meldingene også inneholde verdier om gjeldene temperaturer i boenheten til brukeren. Derfor må temperaturinformasjon lagres til hver enkelt bruker på serveren når en bruker etterspør temperatur eller endrer på temperatur, slik at disse verdiene er oppdaterte når verdiene legges i «smartMessage» og blir sendt til HDL. Når meldingene har blitt sendt, fjernes de fra listen.

Metoden `startZWaveDataReceiver()` henter ut data fra husets lokale Domoticz database og sender verdiene til hver enkelt bruker. Dette gjøres hvert tiende minutt slik at hver enkelt bruker til enhver tid har oppdatert informasjon. For å hente ut denne informasjonen, kjøres det et http request med JSON på Domoticz sitt lokale Web API. Ettersom dataen er hentet med JSON må denne dataen filtreres for informasjonen som skal sendes til brukerne. Det gjøres ved hjelp av en «Scanner» og forskjellige String håndteringer. Når dataen er filtrert vil informasjon til hver enkelt boenhets sensorer og målere, lagres til sine rettmessige brukere, slik at en bruker kan hente denne informasjonen når den blir etterspurt.

4.4 Arkitektur



Figur 19 Arkitektur

Hele systemet er basert på klient/tjener-arkitektur som er illustrert over (Figur 19). Dette medfører at modellen blir som illustrert over med en klient som kjører SHK-applikasjon, som kommuniserer med SHK-server som videre kommuniserer med IP-modul. Denne IP-modulen mottar UDP-pakker fra SHK-server som inneholder kommandoer og videresender disse ut på BusPro-nettverket. Når brukeren logger inn på SHK-applikasjonen på Android-enheten vil kommunikasjonen være mellom klient, tjener og database.

Vi har kodet det på en slik måte at SHK-applikasjonen sender en del av kommandoen til SHK-server. SHK-server tar seg av meldingshåndteringen og en bit-vis CRC-sjekk av pakken som skal sendes til IP-modul. Denne CRC-sjekken(Kodeeksempel 11) er kritisk og bit-sum som CRC returnerer legges til i pakken.

4.5 Kommunikasjonsprotokoller

SHK skal kommunisere med flere forskjellige typer komponenter. Flere av disse komponentene kommuniserer over forskjellige protokoller. Strømmålerne, fuktighetsmålerne, og vannmålerne kommuniserer via Z-Wave. Z-Wave er en trådløs kommunikasjonsprotokoll som ofte brukes i hjemautomatisering. Denne protokollen er designet for å være pålitelig og med liten forsinkelse.⁸ Vi bestemte oss for å bruke Z-Wave fordi det er en godt utviklet platform med et godt nettsamfunn. Det gode nettsamfunnet gjorde at det var enkelt å komme i kontakt med personer som har erfaring og kunnskaper vedrørende Z-Wave.

Enheter som er bygget på Z-Wave må bli inkludert i Z-Wave-nettverket før den kan bli kontrollert. For luftfuktighetsmålerne og vannmålerne sin del, kreves det en USB-kontroller som er basert på Z-Wave. Før disse komponentene kan kommuniseres med, må de «pares» med USB-kontrolleren. Denne USB-enheten kobles så til SHK-server.

⁸ <https://en.wikipedia.org/w/index.php?title=Z-Wave&oldid=714037058>

5.0 Koding, kvalitetskontroll, implementering, og testing

Til å utvikle programvarene brukte vi Android Studio til SHK-applikasjonen og Eclipse til SHK-server. I dette kapitlet presenteres den viktigste koden.

5.1 Implementering av SHK-server

I dette kapitlet vil vi presentere noen sentrale kodeeksempler fra SHK-server. Alle disse eksemplene stammer fra metodene beskrevet i 4.3.2.1 Serverfunksjonalitet, melding- og påloggingshåndtering.

Kodeeksempel fra startLoginMonitor()

```
Socket s = serverSocket.accept();
if(!(tempUser.isEmpty())) {
    for(int i=0; i < tempUser.size(); i++) {
        if((s == tempUser.get(i).returnSocket()) && uniqueUser == false){
            uniqueUser = true;
            tempUser.remove(i);
            UserClient client = new UserClient(s);
            tempUser.add(client);
            System.out.println("Replaced user...");
        }
    }
    if(uniqueUser == false){
        UserClient client = new UserClient(s);
        tempUser.add(client);
    }
}
else {
    UserClient client = new UserClient(s);
    tempUser.add(client);
    System.out.println("User connected...");
}
uniqueUser = false;
```

Kodeeksempel 1 StartLoginMonitor

Her ser vi hvordan metoden håndterer tilkobling av brukere. Nye brukere legges til i «tempUser»-listen. Hvis en bruker som var tilkoblet tidligere prøver å koble til, vil forrige bruker med denne tilkoblingen fjernes og en ny bruker vil bli laget. Dette er for å forhindre duplisering av samme bruker.

Kodeeksempler fra startAPPLLoginMonitor()

```
synchronized(tempUser) {
    Iterator<UserClient> i = tempUser.iterator();
    while (i.hasNext()) {
        UserClient u = i.next();
        try {
            String msg = u.read();

```

Kodeeksempel 2 Itererer gjennom brukerlisten

Metoden itererer over alle objekter i «tempUser»-listen som *startLoginMonitor()* har lagt til. Videre prøver den å lese inn en tekststreng sendt fra denne brukeren.

```

else if(msg.equals("Login")){
    if(u.loginChecker()){
        int id = u.returnUserIDInt();
        Socket connection = u.returnSocket();
        i.remove()
        for(int ii=0; ii< user.size(); ii++) {
            if(user.get(ii).userId == id) {
                user.get(ii).setSocket(connection);
                user.get(ii).sendText(Integer.toString(id));
            }
        }
    }
}
}
}
}

```

Kodeeksempel 3 Loginhåndtering

En av håndteringene i *startAPPLoginMonitor()*. Denne håndteringen kjøres når en bruker som er tilkoblet prøver å logge inn med brukernavn og passord. Hvis påloggingen er vellykket, blir brukeren flyttet over til riktig bruker i «user»-listen. Med riktig bruker menes med hvilken boenhet brukeren vil styre og overvåke, bestemt ut ifra oppgitt brukernavn og passord.

Kodeeksempler fra startAPPMessageListener()

```

else if (msg.startsWith("Command:")) {
    CommandMessageController(u, msg.substring(8,msg.length()));
}

```

Kodeeksempel 4 Sjekk for om en melding er en kommandomelding

Dette er en av håndteringene i *startAppMessageListener()*. En melding inneholder «Command:» når en bruker vil sende en melding til HDL-systemet. *CommandMessageController(UserClient u, String msg)* prosesserer meldingen fra brukeren og bygger opp en datapakke med gyldig format, slik at den kan bli sendt til HDL.

```

public byte[] getBytes(byte[] data, int cmd, int subnet, int devicenr) {
    byte[] p = new byte[27 + (data != null ? data.length : 0)];
    try {
        replyAddress = InetAddress.getByName("192.168.10.141");
    } catch (UnknownHostException e) {
        e.printStackTrace();
        return null;
    }
    if(replyAddress != null) {
        System.arraycopy(replyAddress.getAddress(), 0, p, 0, 4);
    }

    byte[] magic = "HDLMIRACLE".getBytes();
    System.arraycopy(magic, 0, p, 4, magic.length);

    //Setting the array values from leading code (0xAA)
    int i = 14;
    p[i++] = (byte) 0xaa; //Leading code: set at 170
    p[i++] = (byte) 0xaa; //Leading code
    p[i++] = (byte) (p.length -16); //Data package length
    p[i++] = (byte) (sourceAddress >> 8); //Original subnet ID
    p[i++] = (byte) sourceAddress; //Original device ID
    p[i++] = (byte) (sourceDevice >> 8); //Original device type: higher than 8
    p[i++] = (byte) sourceDevice; //Original device type: lower than 8
    p[i++] = (byte) (cmd >> 8); //Operation code: higher than 8
}

```

```

p[i++] = (byte) cmd; //Operating code: lower than 8
p[i++] = (byte) subnet; //Subnet ID of target device
p[i++] = (byte) devicenr; //Device ID of target device

if (data != null) {
    System.arraycopy(data, 0, p, i, data.length);
}
//Computes the correct CRC for the data package
int crc = computeCRC16(p, 16, p.length - 18);
i = p.length - 2; // -2 for the CRC
p[i++] = (byte)(crc >> 8);
p[i++] = (byte)crc;

//returns data to be sent
return p;
}

```

Kodeeksempel 5 Bygger en pakke

Dette er en metode som blir kalt etter at “Command:” meldingen har blitt prosessert. Metoden bygger opp pakken til riktig format og returnerer en byte arrayen med innholdet når dette er gjennomført. Pakken kan deretter bli sendt til HDL.

Kodeeksempler fra startHDLMessageListener()

```

byte[] data = new byte[MAX_PACKET_SIZE];
DatagramPacket receivePacket = new DatagramPacket(data,
    data.length);
socket.receive(receivePacket);

HdlPacket p = HdlPacket.parse(receivePacket.getData(),
    receivePacket.getLength());

```

Kodeeksempel 6 Henter ut Datagrampakke

Denne metoden prøver å hente ut en Datagrampakke sent fra HDL. Tidligere har “socket” blitt bundet til adressen og porten som HDL sender fra. *HdlPacket.Parse(byte[] data, int length)* prosesserer pakken fra HDL og lagrer innholdet i en byte array. Disse verdiene lagres midlertidig som et *HdlPacket*-klasseobjekt. Etter at verdiene er lagret, sjekkes det om pakken inneholder en spesifikk kommando. Dette er for å sjekke om at pakken inneholder data som en bruker har etterspurt.

```

for(int i = 0; i < user.size(); i++) {
    UserClient u = user.get(i);
    if(u.checkForMessage(messageCommand)) {
        u.setTempInfo(channel, sCurrentMode, sCurrentNormalTemp,
sCurrentDayTemp, sCurrentNightTemp, sCurrentAwayTemp, sCurrentTemp);
        u.sendText("TempInfo:" + sChannel + sCurrentMode + sCurrentNormalTemp +
sCurrentDayTemp + sCurrentNightTemp + sCurrentAwayTemp + sCurrentTemp);
    }
}
}

```

Kodeeksempel 7 Finner korrekt bruker

Inneholder pakken kommandoen for temperaturendring eller temperaturinformasjon, vil koden over finne brukeren som har etterspurt denne informasjonen. Denne går igjennom hver bruker og ser om de har etterspurt en lignende melding og om deres bruker-id er den samme som for brukeren som har etterspurt informasjonen.

Kodeeksempler fra startSmartMessageListener()

```
if(u.returnSmartMessageNotNull()){
    int n = u.SmartMessageSize();
    for(int ii = 0; ii < n; ii++) {
        String msg = u.getSmartMessageNr(ii);
        CommandMessageController(u, msg);
        u.removeSmartMessageNr(ii);
        --n;
        System.out.println("Sent message from timer to HDL");
    }
}
```

Kodeeksempel 8 Finner bruker med smartMessage

Denne metoden itererer igjennom alle brukere og ser om de har en melding i sin “*smartMessage*”-liste. Hvis brukeren har en slik melding, prosesseres den og sendes videre til HDL. Når meldingen er sendt, fjernes den fra listen. I de neste kodeeksempelene skal det forklares når meldinger blir lagt til i “*smartMessage*”-listen og hvordan en slik melding ser ut.

```
public void run() {
    if(!modeIsHoliday && !modeIsAway) sendingMessage();
}
```

Kodeeksempel 9 Når tidsuret er ferdig

Dette er metoden som kalles når et tidsur er ferdig. Dette tidsuret håndterer automatisk endring av dag og natt modus. Hvis modusen i boenheten til denne brukeren er satt til ferie eller borte, vil det ikke bli sendt en melding. Hvis den er en av de andre modusene, legges en melding i “*smartMessage*”-listen avhengig av om det er natt eller dagmodus det skal skiftes til. Etter at denne metoden har blitt kjørt, iverksettes et nytt tidsur for neste døgn.

```
int currentTemp1 = Integer.parseInt(CurrentTemp1);
int day1 = Integer.parseInt(CurrentDayTemp1);
String wateringFlag1;
if (currentTemp1 < day1) wateringFlag1 = "1"; else wateringFlag1 = "0";

smartMessage.add("007260112," + Channell + ",0,0," + ModeDay + ","
    + CurrentNormalTemp1 + "," + CurrentDayTemp1 + "," + CurrentNightTemp1
    + "," + CurrentAwayTemp1 + "," + wateringFlag1 + ","
    + "15");
```

Kodeeksempel 10 smartMessage-utseende

Kodeeksemplet () viser hvordan en “*smartMessage*” ser ut. Eksemplet over viser hvordan det hentes ut verdier fra de lagrede temperaturverdiene for denne brukeren. Dette er kun én av meldingene som blir lagt til når tidsuret er ferdig, men det sendes flere tilsvarende meldinger for å sette hele boenheten i riktig modus.

Kodeeksempler for Z-Wave håndtering

For å hente ut verdier fra Z-Wave nettverket vises det til metodene `retrieveZWaveData()` og `startZWaveDataReceiver()` i vedlegg (C Z-Wave-kommandoeksempler), samt deres tilhørende `JavaDoc`.

CRC – Cyclic redundancy check

For at HDL-systemet skal godta pakkene som blir sendt fra SHK-systemet, må pakkens integritet kontrolleres. Denne kontrollen gjøres i `computeCRC16`, som går gjennom hele pakken byte for byte. Hensikten med å utføre en slik CRC er å sjekke pakken for utilsiktede endringer i rådataene som kan forekomme.

`CRCTable` som er en tabell med 256 heksadesimale verdier.

```
protected static int computeCRC16(byte[] data, int offset, int count) {
    int crc = 0;
    int dat;

    for (int i = offset; i < offset + count; ++i) {
        dat = (crc >>> 8) & 0xff;
        crc = (crc << 8) & 0xffff;
        crc ^= CRCTable[(dat ^ (int)data[i]) & 0xff];
    }
    return crc & 0xffff;
}
```

Kodeeksempel 11 CRC

5.2 Implementering av SHK-applikasjon

Her kommer det noen sentrale kodeeksempler og en tilhørende kommentar. Detaljert beskrivelse av metodene og klassene finnes i kapittel 4.2 Design. Hver klasse har sin egen metode som heter `setupGUI()`. Denne blir kjørt på `onCreate`. Noe av koden i `setupGUI()` for toppleiligheten sin `MainActivity` følger under.

```
public void setupGUI() {
    // light control button
    lightBtn = (ImageButton) findViewById(R.id.lightButton);
    lightBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            goToRoomView();
        }
    });

    settings = (ImageButton) findViewById(R.id.settings);
    settings.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            settingsView();
        }
    });

    sharedPreferences = getSharedPreferences(savedColor, Context.MODE_PRIVATE);

    int value1 = sharedPreferences.getInt("value1", 0);
    int value2 = sharedPreferences.getInt("value2", 0);
    int value3 = sharedPreferences.getInt("value3", 0);
    int value4 = sharedPreferences.getInt("set", 0);
    if(value4 != 0){
```

```

        View v = findViewById(R.id.main_id);
        v.setBackgroundColor(Color.rgb(value1, value3, value2));
        setContentView(v);
    }
}

```

Kodeeksempel 12 Initialiserer onClickListener og setter bakgrunnsfarge.

Her settes det opp onClickListeners for noen av knappene i denne klassen. I tillegg blir informasjon om hvilken bakgrunnsfarge brukeren har valgt for sin profil lagret her.

```

public void goToRoomView() {
    Intent intent = new Intent(this, RoomView.class);
    startActivity(intent);
}

```

Kodeeksempel 13 En typisk goToView metode som alle onClickListener knapper kaller på

5.2.1 Innlogging

Fra innloggingsskjermen kan brukeren logge inn, endre passord eller resette et glemt passord. Under kommer relevante kodeeksempler for innloggingen: Kodeeksempel 14 Er ikke tilkoblet server, Kodeeksempel 15 Henter ut brukernavn og passord for «Husk meg»-metode, Kodeeksempel 16 Lagrer brukernavn og passord hvis «Husk meg» er krysset av og Kodeeksempel 17 Sjekker bruker-ID, og sender bruker til riktig hovedmeny.

```

if(!connected) {
    AlertDialog.Builder add = new AlertDialog.Builder(LoginClient.this);
    add.setTitle("Ikke tilkoblet");
    add.setMessage("Prøv å logg inn igjen");
    add.setCancelable(false);
    add.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }
    });
    add.create();
    add.show();
    //Tries to connect again
    connect();
}

```

Kodeeksempel 14 Er ikke tilkoblet server

Over (Kodeeksempel 14) får bruker melding om applikasjon ikke har tilkobling til server når «Logg inn» blir trykket. Bruker bes da trykke «Logg inn» på nytt, og får dermed logget inn.

```

rememberMeBool = loginSettings.getBoolean("saveLogin", false);
if (rememberMeBool) {
    tempUser.setText(loginSettings.getString("username", ""));
    tempPw.setText(loginSettings.getString("password", ""));
    rememberMe.setChecked(true);
}

```

Kodeeksempel 15 Henter ut brukernavn og passord for «Husk meg»-metoden


```

if (rememberMe.isChecked()) {
    loginEditor.putBoolean("saveLogin", true);
    loginEditor.putString("username", username);
    loginEditor.putString("password", password);
    loginEditor.commit();
} else {
    loginEditor.clear();
    loginEditor.commit();
}

```

Kodeeksempel 16 Lagrer brukernavn og passord hvis «Husk meg» er krysset av

Brukernavn og passord blir lagret i SharedPreferences for senere bruk. I Kodeeksempel 16 blir brukernavn og passord hentet ut fra SharedPreferences hvis «Husk meg» er krysset av.

```

switch (ID) {
    case 1:
        loggedIn = true;
        goToAdmin();
        break;
    case 2:
        loggedIn = true;
        goToHome();
        break;
    case 3:
        loggedIn = true;
        goToHome2();
        break;
    case 4:
        loggedIn = true;
        goToHome3();
        break;
    default:
        failLogin();
        break;
}

```

Kodeeksempel 17 Sjekker bruker-ID, og sender bruker til riktig hovedmeny

Hver bruker er oppført med en ID i databasen. Denne ID-en blir returnert under innloggingen, og brukes til å bestemme hvilken hovedmeny brukeren sendes til.

5.2.2 Lys

Når det gjelder lys er leilighetene delt inn i lyssoner. Disse lyssonene består av flere lamper som kan styres enkeltvis eller samlet.

```

protected View.OnClickListener light_all_listener = new View.OnClickListener()
{
    @Override
    public void onClick(View v) {
        if (!lmax) {
            MainActivity.sendText("Command:000002117,1,3");
            lmin = false;
            lmed = false;
            lmax = true;

            lightMinBtn.setChecked(false);
            lightMedBtn.setChecked(false);
            lightOffBtn.setChecked(false);
        }
    }
}

```

```

        setLightSettingsForRow(row1, lightMax);
    } else if (lmax) {
        MainActivity.sendText("Command:000002117,1,0");
        allBoolFalse();
        allButtonsOff();
        lightOffBtn.setChecked(true);
        setLightSettingsForRow(row1, lightOff);
    }
}
};

```

Kodeeksempel 18 Skru av eller på lys

Metoden viser hvordan lys blir slått av eller på, og hvordan relasjonen er mellom knappene. Denne relasjonen sier at når for eksempel knapp «Minimum» er aktivert, og bruker trykker «Maksimum», så slår «Minimum» seg av, og lysintensitet blir økt til maksimum. Alle verdier brukeren setter lysene til å være, vil lagres via SharedPreferences til den gjeldende modusen leiligheten er satt til.

5.2.3 Ventilasjon

I ventilasjonsstyringen har brukeren fire knapper per ventilasjonssone. Trinn en og to er forskjellige hastigheter. På trinn tre er trinn en og to aktivert samtidig.

```

protected View.OnClickListener toggle_level3 = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        connected = true;
        if (!ch1 && !ch2) {
            MainActivity.sendText("Command:000002114,1,1");
            // turns on level 1 and 2
            ch1 = true;
            ch2 = true;
            ch3 = true;
            lvlOneTwoOff();
            setVentlBasedOnMode(level3);
        }
        else if(ch2 && !ch1) {
            MainActivity.sendText("Command:000049114,1,100,0,1");
            // turns on level 1
            ch1 = true;
            ch3 = true;
            lvlOneTwoOff();
            setVentlBasedOnMode(level3);
        }
        else if(!ch2 && ch1) {
            MainActivity.sendText("Command:000049114,2,100,0,1"); // turns
on level 2
            ch2 = true;
            ch3 = true;
            lvlOneTwoOff();
            setVentlBasedOnMode(level3);
        }
        else if (ch2 && ch1){
            MainActivity.sendText("Command:000002114,1,0"); //
turns off 1 and 2
            allBoolFalse();
            setVentlBasedOnMode(level0);
        }
    }
};

```

Kodeeksempel 19 Skru av eller på trinn tre

I Kodeeksempel 19 blir trinn en og to skrudd av eller på. Hvis trinn en er aktivert, blir bare trinn to aktivert, og vice versa for å oppnå trinn tre. Hvis trinn en og to er aktivert, blir begge skrudd av dersom trinn tre aktiveres. Disse innstillingene blir lagret i SharedPreferences i gjeldende modus for leiligheten.

5.2.4 Vinduer

Toppleiligheten er utstyrt med tre takvinduer. Under presenteres kodeeksempler som er relevante for vinduene: Kodeeksempel 20 Steng vindu nr 3, Kodeeksempel 21 Sjekker om vinduet skal åpnes eller lukkes, Kodeeksempel 22 Oppdaterer brukeren på vinduets status, og Kodeeksempel 23 Lagrer vinduets status.

```
protected View.OnClickListener win3_close = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (!w3close) {
            MainActivity.sendText("Command:058336113,1,2");
            w3open = false;
            w3close = true;
            w3stop = false;
            stopping3 = true;
            chk3 = false;

            Thread thread = new Thread(new Runnable() {
                @Override
                public void run() {
                    try {
                        windowOpening(winOpeningTime3, 3, chk3);
                    }
                    catch(Exception e) {
                        e.printStackTrace();
                    }
                }
            });

            thread.start();

            win3Open.setChecked(false);
            win3Stop.setChecked(false);

        }else if (w3close) {
            // do nothing
        }
    }
};
```

Kodeeksempel 20 Steng vindu nr 3

Hvert av de tre vinduene har kode for åpne, lukke eller stoppe. Eksemplet over sjekker først om vinduet er åpent, og hvis det er, sendes det en melding til server om å lukke det. Det startes en metode i en tråd som kontinuerlig oppdaterer brukeren på vinduets lukkestatus. Hvis vinduet allerede er lukket skjer ingen ting.

```
public void windowOpening(int window, final int id, boolean check) {
    for(int i = 0; i < 20; i++) {
        if((stopping1 && (id == 1)) || (stopping2 && (id == 2)) || (stopping3
        && (id == 3)))
```

Kodeeksempel 21 Sjekker om vinduet skal åpnes eller lukkes

Dette er starten på «åpne/lukke eller stopp»-metoden. Denne tar imot parameter på vinduets nåværende status, vinduets ID, og om vinduet skal åpnes eller lukkes. «for»-løkken går 20 ganger, siden det er tiden det tar for motoren å lukke eller åpne vinduet. Den følgende «if»-setningen sjekker om det er «Stopp»-knappen som ble trykket og hvilket vindu som bes stoppe.

```
switch(id) {
  case 1:
    String temp1 = String.valueOf(tempwin + "%");
    if(tempwin > max) {
      degreeOpen1.setText(max + "%");
    }
    else degreeOpen1.setText(temp1);
    break;
  case 2:
    String temp2 = String.valueOf(tempwin + "%");
    if(tempwin > max) {
      degreeOpen2.setText(max + "%");
    }
    else degreeOpen2.setText(temp2);
    break;
  case 3:
    String temp3 = String.valueOf(tempwin + "%");
    if(tempwin > max) {
      degreeOpen3.setText(max + "%");
    }
    else degreeOpen3.setText(temp3);
    break;
}
```

Kodeeksempel 22 Oppdaterer brukeren på vinduets status

Kodeeksempel 22 er en del av «åpne/lukke eller stopp»-metoden (Kodeeksempel 21). Status oppdateres kontinuerlig på skjermen ved hjelp av en midlertidig variabel. Parameter som switchen mottar er ID-en til vinduet som ble sendt til Kodeeksempel 21.

```
switch(id) {
  case 1:
    winOpeningTime1 = window;
    if(window > max) {
      winOpeningTime1 = max;
    }
    break;
  case 2:
    winOpeningTime2 = window;
    if(window > max) {
      winOpeningTime1 = max;
    }
    break;
  case 3:
    winOpeningTime3 = window;
    if(window > max) {
      winOpeningTime1 = max;
    }
    break;
}
```

Kodeeksempel 23 Lagrer vinduets status

Kodeeksempel 23 er en del av «åpne/lukke eller stopp»-metoden (Kodeeksempel 21). Dette eksemplet oppdatere globale variabler, uten å skrive til skjerm. Dette er viktig for at den globale eller midlertidige variabelen som vises på skjerm er lik neste gang en «åpne eller lukke»-knapp

trykkes. Parameter som switchen mottar er ID-en til vinduet som ble sendt til Kodeeksempel 21.

5.2.5 Temperatur

I likhet med lys er leilighetene delt inn i temperatursoner. Under presenteres en metode som er relevant for temperaturreguleringen.

```
int m = Integer.parseInt(mode);
switch (m) {
    case 1:
        int h1 = Integer.parseInt(holiday);
        int c1 = Integer.parseInt(current);
        String wateringFlag1;
        h1++;
        holiday = Integer.toString(h1);

        editor.putString("holiday", holiday);
        editor.commit();
        if (h1 < c1) {
            wateringFlag1 = "1";
        }
        else wateringFlag1 = "0";

        MainActivity.sendText("Command:007260112," + channel + ",0,0," + mode +
            "," +
            + holiday + "," + day + "," + night + "," + away + "," + wateringFlag1
            + "," +
            + "15");
        holiday = holiday + " °C";
        stemp1.setText(holiday);
        break;
}
```

Kodeeksempel 24 Øker temperaturen med én grad Celsius.

I eksempelet over har brukeren trykt på knappen for å øke temperaturen i én av sonene. «Integer.parse(mode)» sjekker hvilken modus leiligheten er i og sender dette med til switchen. Dermed blir temperaturverdien i denne sonen endret med én grad Celsius. Temperaturmetodene for økning eller reduksjon er tilsvarende. Den eneste forskjellen er at den ene adderer og den andre subtraherer. WateringFlag avgjør om ventilene til varmesonen skal være åpne eller lukket. Dette blir bestemt ut ifra om den gjeldene temperaturen for varmesonen er mindre enn den temperaturen det blir satt til.

5.2.6 Moduser

Leilighetene kan settes i fire forskjellige moduser. Mer detaljert om hvordan modusene fungerer ble beskrevet i kapittel 4.2 Design.

```
sharedpreferences = getSharedPreferences(savedTemp1, Context.MODE_PRIVATE);
SharedPreferences.Editor editor1 = sharedpreferences.edit();
editor1.putString("mode", dayMode);
editor1.commit();

if (sharedpreferences.contains("channel")) {
    String channel1 = sharedpreferences.getString("channel", "1");
    String holiday1 = sharedpreferences.getString("holiday", "18");
    String day1 = sharedpreferences.getString("day", "23");
    String night1 = sharedpreferences.getString("night", "21");
    String away1 = sharedpreferences.getString("away", "19");
    String current1 = sharedpreferences.getString("current", "22");
}
```

```

int currentTemp = Integer.parseInt(current1);
int day = Integer.parseInt(day1);
String wateringFlag1;
if (currentTemp < day) wateringFlag1 = "1"; else wateringFlag1 = "0";

MainActivity.sendText("Command:007260112," + channel1 + ",0,0," + dayMode +
", "
    + holiday1 + ", " + day1 + ", " + night1 + ", " + away1 + ", " +
wateringFlag1 + ", "
    + "15");
}

```

Kodeeksempel 25 Modusvalg for dag

I eksempelet over blir gjeldende modus satt til dag. Metoden finner temperaturen til dagmodus for en bestemt temperatursone (channel) og sender verdiene til server. Videre går metoden igjennom alle resterende varmesoner, lys- og ventilasjonsinnstillingene for den valgte modusen, slik gjeldene leilighet blir satt til dette.

5.3 Kvalitetskontroll og testing

Underveis i utviklingen har vi testet og sørget for at alt fungerer etter hensikt. Vi kjørte testene slik at ønsket resultat oppstod både med gyldig og ugyldig input. Underveis i utviklingen jobbet vi alle på hver vår gren i GitHub, og kjørte tester før vi slo sammen med Master.

Det har også vært vanskelig å få testet SHK på et skikkelig praktisk vis. Boligen dette skal brukes i er ikke ferdig, men vi har kontinuerlig testet applikasjonene opp mot demokofferten. Siden boligen SHK skal benyttes i inneholder vesentlig flere komponenter, er det særlig viktig for produktets forventede ytelse og stabilitet at disse komponentene adresseres (på BusPro-nettverket) i relasjon til planen vår for adressering eller vice versa.

Fremgangsmåten vi brukte for å teste applikasjonen var at vi testet kontinuerlig etter å ha produsert kode. Dette gjorde vi for å alltid ha en fungerende versjon av SHK-applikasjonen for å bygge videre på. Ofte ble små kodesnutter testet mot demokofferten eller mot server for å fremprovosere kræsjer og finne andre feil. Feilsøking ble gjort ved å legge inn *Log.d* og skrive ut relevant tekst i konsollvinduet for å overvåke hvor i koden vi var, og for å kunne forstå hvor feil oppstod. Konsollvinduet ble også brukt for å undersøke hvilken kodelinje programmet eventuelt krasjet på. Vi brukte ingen form for formell testing slik som JUnit. Dette fordi vi hadde lite erfaring med det og ville ført ytterligere forsinkelser. Det var heller ingen logging av avvik og feil i egen fil. I etterkant har vi vurdert arbeidet og kommet frem til at vi burde ha brukt formelle metoder for testing.

Demokofferten inneholder, som nevnt tidligere, LED-spotter, reléer, dimmere, varmestyringskontroll, DLP-brytere, og flere sensorer. Vi utførte testene ved å koble en PC til IP-modulen i kofferten via en Ethernet-kabel, og sendte kommandoer til denne. Disse testene ble utført kontinuerlig under utviklingen. Hver enkelt enhet i kofferten har en egen unik adresse. Det samme prinsippet med unike adresser på hver enhet vil også bli brukt i praksis.

5.3.1 Brukertesting

En brukervennlig applikasjon er viktig, ellers kommer den på sikt ikke til å bli brukt. For å vurdere brukervennligheten utførte vi brukertester for å registrere hvordan en person som ikke har vært med i selve utviklingen synes det var å benytte applikasjonen. Siden applikasjonen skal kunne brukes av hvem som helst er det viktig at vi får testet det på personer med forskjellig bakgrunn.

For å utføre testingen opprettet vi et undersøkelsesskjema, og mens testen ble utført noterte vi testpersonens synspunkter. Før vi foretok brukerundersøkelsen gav vi en forklaring til hva applikasjonen kan brukes til. Vi stilte følgende spørsmål:

1. Hvordan synes du hovedmenyen er?
2. Er det enkelt å forstå ikonene?
3. Opplever du applikasjonen som kjapp?
4. Hvordan synes du inndelingen av funksjoner er?
5. Hvordan opplever du temperatursiden?
6. Hvordan opplever du lysstyringen?
7. Hvordan synes du det var å sette tidsinnstillinger for natt/dagmodus?
8. Hva mener du om modusene?
9. Er det noe du savner?

Svarene vi fikk var:

Testperson: Kvinne, 55 år

Test utført av: Jørn-Are, Henrik og Edvard – 04. mai 2016

1. Oversiktlig og enkel. Modusknappen kunne vært enklere.
2. Lys, vindu, ventilasjon og klima er bra. Modus er litt forvirrende.
3. Den var bra, likte innlastingen.
4. Modusmenyen, sett ferie-modus nederst.
5. Temperaturvisningen er litt forvirrende. Tydeligere ikoner. to ting er ikke greie. Er ting lagret? Hva er det til venstre? Nåværende temperatur?
6. Visuell endring. Tydeliggjøre at den øverste raden er hovedstyring. Gjøre det tydeligere å se hvilket rom man er i. Innstillingen av medium dimmeverdi på lys er forvirrende. Burde være mer informativ. Mulighet for å sette minimumsverdi for dimmere.
7. Veldig klønete, ikke tydelig hvor jeg skal trykke for å sette tid for natt/dag.
8. Savner flere moduser, som kveldmodus eller et arbeidsmodus. Enklere måte å bytte mellom lys- og modussiden.
9. Det bør være mulig å endre modus fra hovedmenyen. Dialogbokser. Lagring av settinger.

Testperson: Kvinne, 24 år

Test utført av: Edvard– 04. mai 2016

1. Modusknappen er ikke bra. Ellers veldig bra.
2. Bra, bortsett fra Modus.
3. Applikasjonen er veldig rask.
4. Likte det godt.
5. Skjønner ikke hvilken temperatur som stilles. Oversiktlig ellers. Likte pilene.
6. Litt mange knapper. Likte at man kan styre alle lys samlet.
7. Litt uoversiktlig.
8. Grei inndeling.
9. Savner OK-knapper i dialog. Forslag til modusknappen: Bytt ut med tekst?

Testperson: Kvinne, 19 år

Test utført av: Edvard– 06. mai 2016

1. Oversiktlig hovedmeny. Skjønner hva som er hva.
2. Fine ikoner.
3. Veldig kjapp.
4. Enkel inndeling.
5. Temperatursiden er litt forvirrende. Hvilken temperatur er det som er hva? Nåværende/ønsket i stedet for ikonene som er over temperaturverdiene.
6. Likte lys. Grei inndeling av rom.
7. Modus, greit at du da har forhåndssatte verdier.
8. Enkel å skjønne, bortsett fra temperatur.

Testperson: Mann, 23 år

Test utført av: Jørn-Are, Henrik og Edvard– 06. mai 2016

1. Oversiktlig. Kunne kanskje hatt tekst vedsiden av ikoner.
2. Forstår ikonene. Veldig enkelt.
3. Er ganske kjapp.
4. Klima kunne vært splittet i to, temperatur og ventilasjon. Spart litt trykking.
5. Temperatursiden er litt rar. Tekst i stedet for ikoner.
6. Grei inndeling av rom. Kunne hatt faner og heller swipet gjennom rommene.
7. Er praktisk. Drop-down meny i stedet for å skrive inn tid.
8. Likte at modus var markert og tydeliggjort.
9. Er fargeblind. Kunne kanskje hatt sterkere kontrast som markerer aktiv knapp. Noe tungvint navigasjon.

Testperson: Mann, 23 år

Test utført av: Jørn-Are – 07. mai 2016

1. Rent og oversiktlig. Lett å skjønne hvor jeg skal trykke.
2. Modus-knappen er litt uklar.
3. Den er kjapp
4. Synes det virker greit. Finner enkelt det jeg vil finne.
5. Lett forståelig. Gjerne finne en bedre måte å vise nåværende romtemperatur og ønsket.
6. Lett forståelig.
7. Lett.
8. Veldig greie.
9. Brukertilpasning. Velge mellom tekst og bilder på knapper.

Testperson: Kvinne, 57 år

Test utført av: Jørn-Are – 07. mai 2016

1. Grei nok, men forstår ikke alle ikonene.
2. Forstår de tre øverste, ikke de tre nederste.
3. Veldig kjapp.
4. Går greit å finne frem når hun skjønner ikonene.
5. Forstod ikke helt det med alle temperaturreguleringspilene på temperatursiden.
6. For så vidt greit. Ønsker å kunne styre modus fra lys-siden.
7. ---
8. ---
9. Savner:
 1. Savner mulighet til å sette et tidspunkt for når lys automatisk skal slås på.
 2. Skulle gjerne hatt mulighet til å bestemme når et bestemt rom skal bytte modus med tanke på lys.
 3. Savner også å kunne sette lyssekvenser for når hun er på ferie (for å få det til å se ut som hun er hjemme).

Oppsummert mener testpersonene at applikasjonen er enkel å navigere gjennom og generelt enkel å forstå, men at «Modus»-ikonet skulle vært tydeligere. På grunnlag av tilbakemeldingene, har vi derfor vurdert det som best å erstatte denne med tekst. Det er også ønskelig at det er mer veiledning, særlig da i dialogbokser. Her skal vi legge til en «OK»-knapp, slik at brukerne vet hvor de skal trykke for å etter å ha fått opp en dialogboks. Applikasjonen oppleves som rask og med lav responstid, og testpersonene likte at de har muligheten til å endre bakgrunnsfarge etter ønske. Da enkelte har vanskeligheter med å tyde ikonene, fikk vi tips fra en testperson å ha tekst i tillegg til ikoner. Vi skulle gjerne implementert dette ved å ha ikoner på venstre side, sidestilt med tekst på høyresiden. Dette ble ikke utbedret da vi foretok testene i begynnelsen av mai, og vi ikke hadde rukket å fått den kvaliteten vi ønsket.

Vi merket en tendens til at yngre personer hadde lettere for å prøve på egenhånd, og hadde større forståelse for hvordan applikasjonens funksjoner var inndelt og hvordan de kunne navigere seg gjennom applikasjonen. Eldre personer hadde større vanskelighet med å navigere, og det måtte mer forklaring til for at de skulle forstå applikasjonens funksjoner.

Før vi foretok brukertester hadde vi vurdert hvilke resultater vi kom til å få. Vi antok at eldre personer skulle ha større vanskeligheter med å bruke applikasjonen enn yngre. Denne antagelsen var basert på personlige erfaringer.

6.0 Avslutning og konklusjon

6.1 Diskusjon

Vi begynte prosjektfasen med å vurdere og tenke gjennom hva som skulle kombineres av allerede eksisterende systemer, slik at de til slutt dannet ett styringssystem for hele hjemmet. Et av de tidligste valgene vi tok var hvilket programmeringsspråk vi skulle benytte oss av. Vi bestemte oss for at vi skulle utvikle systemet i C, siden det er ett språket vi alle har brukt før og har erfaring med fra tidligere fag. utfordringer rundt nettverksdelen (tråder og datapakker) i C gjorde derimot at vi valgte Java for både SHK-applikasjon og SHK-server.

For at styringskomponentene skulle kunne kobles opp sammen har vi vært avhengig av finne tilleggskomponenter som gjør det mulig for at disse skal kobles til et eksternt system. Problemet med disse styringskomponentene er at de er laget av forskjellige leverandører og kan derfor ikke integreres på samme måte. I tillegg, kommuniserer ikke alle disse systemene på lik måte. Tidlig i prosjekteringsfasen, fant vi ut at det ville bli vanskelig å få integrert enkelte systemer. Dette er fordi det er proprietære systemer og for Velux sin del forgår kommunikasjonen på enheten gjennom en kryptert tilkobling. Ettersom vinduene allerede var installert i huset måtte vi finne en løsning på dette. Her fant vi da ut at det måtte installeres et sensorgrensesnitt som er direkte tilkoblet kontrollpad til Velux Integra, og deretter måtte denne sensoren koblet opp mot gardinkontroller, som igjen var koblet til BusPro-nettverket. Sensorgrensesnittet ville dermed ta imot meldinger fra styringssystemet for deretter sende det videre til kontrollpad, som igjen krypterte meldingen og videreførte det til vinduet.

Når det skal lages en programvare som skal kommunisere med komponenter som varmepumpe, ventilasjon, dimmere og så videre, er det mye som skal på plass. Komponentene kommuniserer over forskjellige kanaler og protokoller, og hver av disse krever forskjellig utstyr som takler dette. Det desidert vanskeligste har vært vannmålerne og avlesere for disse. Vannmålerne som er på markedet i dag kommer sjeldent med trådløst avlesningsutstyr basert på åpen standard, og det har vært en utfordrende oppgave i seg selv å finne slikt utstyr som vi kan bruke i prosjektet.

Vi fant også ut at varmepumpen som var bestilt fra Vaillant ikke kunne kommunisere med eksterne systemer, og vi måtte derfor prøve å finne en annen løsning. Videre ble det besluttet at den beste løsningen var å gå over til en annen varmepumpe som kunne kommunisere via KNX, og samtidig kunne plasseres på det eksisterende oppsettet som er i huset. Problemet var at denne varmepumpen foreløpig kun ble levert som 400V, slik at det derfor måtte en transformator til. Vi fant derimot ut at det ikke er varmepumpen i seg selv som skulle styres, men heller ventilene som er plassert på rørene som transporterer varmt vann rundt omkring.

Utstyret vi anbefalte ble ikke bestilt før i slutten av april og begynnelsen av mai, og samtidig var ikke HDL-komponenter som dimmere, varmestyringskontrollere montert. Dette gjorde det umulig for oss å teste SHK i praksis på lokasjon, noe som hadde vært svært ønskelig før vi leverte ifra oss SHK.

Vi trodde i begynnelsen at vi måtte ha tilgang til HDLs kildekode for å kunne utvikle styringssystemet. Dette viste seg derimot å ikke stemme, for det vi trengte var kommandoformatet på pakkene som sendes til og fra HDL. Grunnen til at vi trodde det var at vi trodde vi skulle videreutvikle deres system, og at vi ikke visste hvordan vi skulle løse oppgaven. Videre ble det nevnt at vi måtte benytte oss av HDLs egne drivere for å kommunisere med HDL. Vi fikk tilsendt driverne og tilhørende dokumentasjon, men dokumentasjonen var mangelfull og ikke forklarende. På grunn av kravendringen (2.1

Oppstartsfase) ble også hele den opprinnelige fremdriftsplanen (Gantt-diagram i vedlegg B) endret. I henhold til fremdriftsplanen skulle vi ha totalt åtte Scrum-sprint, men endte tilslutt opp med syv. Vi hadde planlagt en testperiode etter at utviklingen var ferdig, men i realiteten testet vi kontinuerlig opp mot demokoffert. Sprintene ble utsatt med flere uker, da vi planla å begynne 15. februar, men begynte i realiteten 14. mars. Vi lå altså én måned bak skjema før vi begynte med utvikling. Formulering av endelig kravspesifikasjon og fremdriftsplan (Oppdatert fremdriftsplan) foregikk underveis og retrospektivt i utviklingen.

Litteraturstudier av programvare som Iridium Mobile foregikk frem til kravendringen og før første Scrum-sprint. Når det gjelder testing av styringssystemet på lokasjon ble ikke dette utført som følge av at HDL-komponenter ikke var montert i boligen når fristen for ferdigstillelse gikk ut. Testing av SHK ble altså ikke som planlagt, men ble heller utført opp mot HDL demokoffert.

Milepælene og planene som ble formulert tidlig i prosjektfasen har ikke blitt fulgt i henhold til tidsskjema. Den første milepælen var å sette opp HDL-styringssystem. Denne milepælen utgikk som følge av kravendring. De resterende milepælene ble derimot nådd etter planlagt tid. Vi hadde en milepæl for når vi fikk etablert kontakt mellom applikasjonen og serveren. Denne milepælen skulle etter planen nås i sprint 6, men ble i realiteten nådd i sprint 1.

Statusrapportene ble ikke levert i henhold til planen. Disse ble levert henholdsvis i uke 7, uke 13 og uke 18. Dessuten ble rapportskrivningen ikke som planlagt. Hovedfokuset til gruppen var å få en fungerende applikasjon og rapportskrivning ble derfor forsinket. Vi delte derimot gruppen i to etter 11. april, der en person jobbet med rapport. Vi omorganiserte oss igjen 2. mai, nå med to personer på rapport og en på ferdigstillelse og feilretting av SHK. Fra 9. mai og frem til innlevering jobbet alle med rapport.

Når det gjelder loggføring, så har vi ført logg for hver dag, men i tilhørende vedlegg () er disse oppsummert til ukentlige logger, og når det kommer til timeføring ble dette dårlig fulgt opp til å begynne med, men ble bedre midtveis i februar (N Ukentlig logg).

6.2 Resultater

6.2.1 Mål

Før prosjektet startet formulerte vi effektmål, resultatmål, og læringsmål. Effektmålene blir vanskelig å måle ettersom SHK-systemet ikke blir tatt i bruk før boligen er ferdig.

Resultatmålene anser vi derimot som i stor grad oppnådd. Vi anser at styring av komponenter som blant andre lys, vindu, ventilasjon, og temperatur er gjennomført på en god måte, og overvåkningsverktøyene som lar beboerne overvåke temperaturer i hjemmet, hvilke lys som er på eller av, om vinduene er åpne eller lukkede.

Det mangler å ferdigstille og teste oversikt over strøm- og vannforbruk. Vi mottok i begynnelsen av mai utstyr for å måle strømforbruk ved hjelp av utstyr som måler magnetisme som strømmer gjennom strømkabler. Disse viste seg derimot å være amerikansk utgave, og derfor ikke kompatible med Z-Wave-mottageren vi hadde. Vannmålere med Z-Wave-modul er ikke mottatt og montert enda. Nødvendig utstyr for å utvikle og teste funksjonaliteten mangler, og det er derfor ikke sikkert at det som er blitt gjort kommer til å fungere i praksis. Se uke 19 i vedlegg, N Ukentlig logg.

Applikasjonen ser kanskje ikke ut som særlig mye arbeid basert på mengden funksjonalitet som finnes, men det ligger mye arbeid i å få en slik applikasjon til å fungere med så mange

komponenter. Den underliggende funksjonaliteten er omfattende og særlig funksjonalitet knyttet til modus er godt planlagt og implementert.

Hele denne prosessen har vært utrolig lærerik for alle oss som har deltatt i prosjektet, vi har lært mye om hvordan større prosjekter fungerer og ikke minst hvordan de burde organiseres fra starten. Vi har lært mye om det å ha mange personer å forholde seg til og gjøre vurderinger ut ifra de, og ikke minst hvordan det er å forholde seg til en oppdragsgiver når man bruker Scrum. Vi har virkelig lært viktigheten av versjonskontrollverktøy når man jobber i så store prosjekter med mange forandringer over tid, og ikke minst viktigheten av å loggføre hva som blir gjort og hva som skal gjøres og delegering av arbeidsoppgaver.

6.2.2 Produktet

I begynnelsen var det vanskelig å planlegge hvordan et endelig produkt skulle bli når så mange forskjellige komponenter skulle kombineres. Mange råd – gode og mange ikke fullt så gode – gjorde at perioden frem til et ferdig produkt ble utfordrende og med unødvendige vanskeligheter. Etter kravendringen var det viktigste for oss å få produsert en fungerende prototype så fort som mulig. I ettertid ser vi at flere av valgene vi tok ikke var helt gjennomtenkte, og ideelt sett skulle vi brukt mer tid på planlegging. Når en bruker skal logge inn, så blir passordet til brukeren sendt i klartekst over nettverket. Dette er fra et sikkerhetsperspektiv ikke ønskelig, da datatrafikk kan sniffes og passord kan hentes ut derfra.

Totalt sett mener at vi at produktet som er utviklet i stod grad samsvarer med oppdragsgivers forventninger og krav. Underveis i prosjektet har det kommet flere ønsker om hva som skal inkluderes i systemet, og senest i slutten av april fikk vi ønsker om integrering av CO₂-måler. Med hensyn på tid og at vi ikke fant en passende måler umiddelbart, valgte vi å ikke forsøke å inkludere dette. En slik måler hadde vært god å ha med tanke på måling og vurdering av innklimaets kvalitet, men det lot seg ikke gjøre.

Produktet i sin helhet fungerer i dag utmerket til å gjøre de mest dagligdagse tingene, som å regulere temperaturer, måle fuktighet, regulere lys, åpne og stenge vinduer, og styre ventilasjon. I tillegg har produktet en fullt fungerende styring av moduser som er mer kompliserte og omfatter de dagligdagse operasjonene. Når det kommer til oversikt over strøm- og vannforbruk er ikke denne funksjonaliteten ferdig utviklet og testet(6.2.1 Mål).

Designmessig er ikke applikasjonen optimal. Vi ser i ettertid at vi gjerne skulle fulgt Android-standard når det kommer til menysystem og ikoner. Grunnen til at vi ikke gjorde det fra begynnelsen var at vi manglet kunnskap og ferdigheter angående dette. Vi ser også at vi skulle gjennomført brukertester tidligere, slik at designet kunne blitt bedre.

6.3 Gruppeevaluering

Vi startet arbeidet så tidlig som mulig med å drøfte arbeidsmetoder, utviklingsmodell, og hvordan oppgaven burde angripes. Gruppen startet og har vært veldig motivert, og det var særlig motiverende å endelig jobbe med en virkelig problemstilling. Prosjektet har vært utfordrende grunnet oppstarten vi hadde (2.0 Forarbeid). Vi føler at arbeidsmoralen har blitt redusert mot slutten, men vi har alle bidratt til å holde samholdet og motivasjonen for arbeidet oppe.

Vi ble alle enige om at vi skulle dele ansvaret mellom oss og skrive relevante gruppregler og konsekvenser i tilfellet noe skulle skje. Vi har jobbet tett sammen gjennom hele prosjektet, og nesten uten unntak jobbet sammen på grupperom. Resultatet av et så tett samarbeid har vært at det gjorde det enklere for oss å hjelpe hverandre og har hatt mulighet til å diskutere saker fortløpende.

Siden ingen av oss før hadde jobbet med så store prosjekter før var det mye nytt, og det var utfordrende å organisere arbeidet med tanke på langsiktighet og fordi ønsker fra oppdragsgiver kom fortløpende.

Vi hadde arbeidsoppgaver i en produktkø noe som gjorde det enkelt å se hvilke oppgaver som gjenstod til enhver tid. Enhver sprint har begynt med et planleggingsmøte der vi drøftet hvilke oppgaver vi skulle gjøre i kommende sprint. Dette ble ikke så godt gjennomført i starten, da vi så at vi tok på oss for mange «tasks» til å få unnagjort på én uke. Som følge av dette reduserte vi antallet i hver sprint, slik at vi kom oss gjennom sprintene uten å ha en stor mengde med ikke-gjennomførte «tasks». Vi har hatt god dialog med både veileder og oppdragsgiver, både gjennom møter og e-post. Vi har hele tiden forsøkt å holde begge parter oppdatert på hva vi har holdt på med og hva vi har fått til.

Alt vi har gjort har blitt loggført i Google Docs og Google Sheets. Vi har vært avhengige av å enkelt kunne dele og samarbeide på dokumenter, og til dette har Google Drive fungert bra.

Prosjektet har stort sett vært uten særlige konflikter og komplikasjoner rundt hva vi ville jobbe med, men det har vært mange diskusjoner rundt implementering, selv om vi har kommet frem til enighet. Vi har vært opptatt av at bloggen skulle holdes oppdatert med det siste nye og vi har alle tatt del i å skrive innlegg.

6.4 Kritikk av oppgaven

Som beskrevet tidligere var startfasen av denne bacheloroppgaven preget av misforståelser og vi var usikre hvilken rolle vi hadde i dette prosjektet. Dette er beskrevet mer detaljert i vedleggene (G Statusrapporter, L Møtereferat fra møter med prosessveileder, N Ukentlig logg og tidligere i rapporten (2.1 Oppstartsfase)). Mye av skylden for dette ligger i at vi ikke gjorde en grundig nok undersøkelse av produktene som ble anbefalt og kravene som foreligger i en bacheloroppgave. Perioden fra mars og ut gikk bedre, og vi fikk lagt en plan for hvordan vi skulle få gjort mest mulig på den korte tiden vi hadde til rådighet.

Scrum er en smidig utviklingsmodell, og har fungert bra for oss. Vi har stadig fått nye ønsker fra oppdragsgiver, og vi har hatt muligheten til å legge ønskene rett inn i produktkøen. Dette er første gang vi har praktisert Scrum, og det har gjort at vi har glemt noen Scrum-møter, så det er et forbedringspotensial.

Det har også vært vanskelig å få testet SHK på et skikkelig praktisk vis. Boligen dette skal brukes i er ikke ferdig, og på grunn av dette har vi ikke fått laget en adresseringsplan med komponentenes adresser.

Vi har også sett begrensingene av å ha en privatperson som oppdragsgiver for et så stort prosjekt. En privatperson har sjeldent samme ressurser og faglig ekspertise som en bedrift gjerne har. Dette har gjort det vanskeligere å løse oppgaven siden vi da ikke har hatt en direkte fagressurs for utviklingen.

Vi mener at dette er en oppgave som ikke burde sluppet gjennom NTNUs filter for godkjente bacheloroppgaver, noe universitetet også har innsett og skal foreta korrigerende tiltak. Oppgaven har hatt flere mangler når det kommer til krav, mangelfull planlegging og at det er en privatperson som er oppdragsgiver leder til flere utfordringer. Samtidig mener vi at oppgavebeskrivelsen var lite dekkende og vag i sin formulering. Det ble ikke gitt retningslinjer på hva produktet skulle gjøre annet enn det prosjektskissen sier «Systemet skal også sentralt kunne regulere temperatur (vannbåren gulvvarme) og ventilasjon (minivarmegjennvinnere - Lunos) og lys i de ulike temperatursonene i boligene». Måten dette skulle gjøres på var opp til

oss å vurdere, og var grunnen til at vi lenge trodde at varmpumpen i seg selv skulle styres, når det faktisk er ventiler som styres. Vi sendte forespørsel til oppdragsgiver om ny oppgavebeskrivelse, og fikk tilsendt en mer grundig beskrivelse noe senere. De allerede valgte komponentene var en utfordring. Disse komponentene er basert på proprietære standarder, og gjør eventuell styring fra en tredjepart vanskeligere og dyrere.

6.5 Videre arbeid

Når det gjelder veien videre, da med tanke på videre utgivelser og forbedringer, blir ansvaret lagt over på oppdragsgiver. Det blir opp til oppdragsgiver å videreutvikle SHK dersom det viser seg å være ønskelig. Mer konkret om hva som kan gjøres for at SHK blir enda bedre blir opp til oppdragsgiver å vurdere, men det inkluderer gjerne å ferdigstille funksjonalitet knyttet til forbruk og videre energieffektivisering.

Strøm- og vannmåling er viktig for å gjøre beboerne mer bevisste på eget forbruk. Her gjenstår det arbeid for å få fullført funksjonaliteten knyttet til dette. Siden vi ikke fikk mottatt nødvendig utstyr, fikk vi ikke utviklet dette ferdig. Vi ønsket å få en oversikt over energiforbruk i hvert rom, slik at beboerne kunne fått en oversikt over de rommene som bruker mye energi i relasjon til andre tilsvarende rom, og identifisert mulige energilekkasjer. En slik oversikt kunne blitt visualisert med totalt forbruk per dag, måned, og kanskje til og med årlig.

Videre kan det være mulig å få boligen til å bli mer «selvstendig» med tanke på automatisk ventilasjon. Det hadde kanskje vært interessant å få en vindmåler på utsiden av boligen, som måler vindretning. Her kunne det vært mulig å få til at SHK passer på at når vindretningen er optimal, så handler ventilasjonsanlegget i forhold til dette, og på denne måten fått til en bedre gjennomstrømming av frisk luft. CO₂-målere kan også inkluderes, slik inneklimate til enhver tid er optimalt.

Lysstyring er også et område som kan bli mer automatisk. Det kunne vært aktuelt å inkludere lysmålere som måler belysningsstyrke (lux), og som regulerer lyset ut ifra dette. Det kunne vært aktuelt å få lyset å dimme ned når det er mye sollys, og vice versa når det ikke er så mye naturlig lys. Det kunne også vært aktuelt å ha en funksjon i feriemodus som gjør at huset automatisk simulerer at noen er hjemme ved å skru på lysene til bestemte tider, selv om en er bortreist på ferie.

Ved å implementere de nevnte forslagene kan energieffektiviteten og forbruket reduseres ytterligere.

Det er problematisk at SHK-applikasjon bare er tilgjengelig for de med Android-enheter, da det er ikke utenkelig at personer som skal flytte inn i boligen har enheter som kjører andre operativsystem enn Android. Produktet er heller ikke implementert i boligen. Dette er på grunn av at nødvendig infrastruktur (elektrisk anlegg og HDL-komponenter) i boligen ikke er ferdigstilt, og det blir derfor ikke mulig for oss å implementere SHK.

6.6 Konklusjon

Det har vært en interessant og utfordrende prosjektperiode. Vanja har vært en interessant og utfordrende oppdragsgiver. Oppgaven har vært veldig spennende, meget utfordrende, og samtidig lærerik. Vi har alle hatt stort utbytte i form av økt kunnskap og ferdigheter, og vi har virkelig fått satt det vi har lært i løpet av et treårig studium hos NTNU på prøve.

Det har vært spennende å være en del av et virkelig prosjekt som handler vel så mye om bærekraft som å lage et alminnelig styringssystem. Ved å bruke en smidig utviklingsmodell har vi vært i stand til å utvide funksjonaliteten kontinuerlig. Selv om det mangler viktig

funksjonalitet, håper vi at produktet ellers står til forventningene og at det blir brukt i lang tid fremover.

Vi skulle veldig gjerne ønsket at vi hadde en tydeligere oppgave fra begynnelsen og at vi hadde noe å ta utgangspunkt i når vi begynte. Oppgaven ble ikke bare å utvikle et styringssystem. Det ble i tillegg vår oppgave å finne ut av hvilke komponenter som skulle kjøpes inn, og samtidig forholde oss til komponenter som var kjøpt. Vi er veldig fornøyde med produktet vi leverer, og føler at vi gjorde det beste vi kunne med tiden vi hadde til rådighet.

Vedlegg

- A** Terminologi
- B** Kommandoformat
- C** Z-Wave-kodeeksempler
- D** Prosjektplanen
- E** Prosjektavtalen
- F** Grupperegler
- G** Statusrapporter
- H** Oppdatert oppgavebeskrivelse fra oppdragsgiver
- I** Dialog med HDL Nordic
- J** Prosjektering, og utstyrliste
- K** Referat fra møter med oppdragsgiver
- L** Referat fra møter med prosessveileder
- M** Ukentlig logg gjennom hele prosjektet
- N** Oppdatert fremdriftsplan
- O** Timeføring

A Terminologi

Android: Operativsystem utviklet av Google.

Android Studio: Utviklingsprogram.

Apache Derby: Et relasjonsdatabaseverktøy (RDMS).

BusPro: Nettverket som knytter alle komponentene i huset sammen.

C: Programmeringsspråk.

DLL: (Dynamic Link Library) bibliotekfiler som inkluderes i SHK.

DLP-bryter: Multifunksjonsbryter med skjerm.

Domoticz: Hjemmeautomasjonssystem som mottar Z-Wave-signaler og lagrer disse.

Eclipse: Utviklingsprogram.

Grønn teknologi: Teknologi som direkte eller indirekte forbedrer miljøet.⁹

Hash(ing): En teknikk for å gjøre en klartekst om til kode som representerer klarteksten.

HDL IntelliCenter: Server utviklet av HDL for styring av HDL BusPro-system.

Java: Programmeringsspråk.

JUnit: Testverktøy for programvarekomponenter.

Klient/tjener-arkitektur: Metode for å fordele arbeid mellom klient og tjener(server).

KNX: Standard for styring og regulering av blant annet lys

mBus: Kommunikasjonsprotokoll.

Mesh-nettverk: et nettverk der pakker blir sendt videre via komponentene i nettverket.

OnCreate: En aktivitet i Android, når en Activity blir initialisert.

SHK: Smarthus Kragerø - vår egenutviklede smarthusløsning (styringssystemet).

SQL: Structured Query Language.

SQL-injeksjoner: Angrepsmetode der en SQL-setning blir manipulert til å gjøre noe den ikke i utgangspunktet var ment til.

Zigbee: Kommunikasjonsprotokoll.

Z-Wave: Kommunikasjonsprotokoll.

⁹<http://www.innovasjon Norge.no/no/Innsats/nr-1-2015/aktuelt/vekstmuligheter-i-gronn-teknologi/>

B Kommandoformat

For at pakkene som sendes fra server skal kunne tolkes riktig av HDL, må de bygges på en forhåndsdefinert måte. Pakkene må sendes som en UDP-datapakke, og sendes til en spesifikk port som ser satt av HDL. Ettersom det sendes via UDP foregår kommunikasjonen som «Point to Point», og pakker som ikke kommer frem vil forkastes. Kommunikasjonen mellom server og HDL er kablet slik at sannsynligheten for at pakker blir borte blir sett på som svært liten, med forutsetning at trafikken på systemet er lav. Vi ser på kommunikasjonen mellom mobilapplikasjonen og serveren som en viktigere del av prosessen for at pakker skal leveres. Ettersom denne kommunikasjonen foregår over TCP, tilfredsstilles kommunikasjonen mellom applikasjonen og HDL-systemets våre krav. I UDP-kommunikasjon er det heller ikke et krav om at mottakeren av pakken sender et svar tilbake til senderen, men dette foregår ved at pakker som mottas av HDLs IP-modul kringkastes til alle enheter som lytter etter meldinger. Dette inkluderer også serveren som er tilkoblet IP-modulen. Enhetene som kan ta imot disse pakkene og kan kjøre funksjonene som den blir bedt om, kringkaster også et resultat tilbake til alle enhetene som lytter. Pakkene som sendes til HDL har følgende oppbygging:

Indeks	Bemerkelse	Bits	Omfang
1	IP-adressedel 1	8	0-255
2	IP-adressedel 2	8	0-255
3	IP-adressedel 3	8	0-255
4	IP-adressedel 4	8	0-255
5	Konstant tegn 1	8	0x48
6	Konstant tegn 2	8	0x44
7	Konstant tegn 3	8	0x4C
8	Konstant tegn 4	8	0x4D
9	Konstant tegn 5	8	0x49
10	Konstant tegn 6	8	0x52
11	Konstant tegn 7	8	0x41
12	Konstant tegn 8	8	0x43
13	Konstant tegn 9	8	0x4C
14	Konstant tegn 10	8	0x45
15	Lederkode	16	0xAAAA
16	Størrelsen på datapakken	8	11-78
17	Subnet-ID til sender	8	0-254
18	Enhets-ID til sender	8	0-254
19	Enhetsstype til sender	16	0-65535
20	Kommando	16	0-65535
21	Subnet-ID til mottaker	8	0-255
22	Enhets-ID til mottaker	8	0-255
23	Data	0-536	
24	CRC	16	0-65535

Tabellen under forklarer hver indeks i foregående tabell.

Indeks	Forklaring
1-4	32 bits som er satt til IP adressen til senderen. Her bruker vi en standard IP adresse som er 192.168.10.141, hvor hver av de fire feltene plasseres i hvert sitt felt.
5-14	80 bits som inneholder en forhåndsbestemt ASCII streng. Denne strengen er «HDMIRACLE». Inneholder ikke pakken denne strengen, vil den ikke bli prosessert.

15	16 bits lederkode som marker starten på datapakken. Denne er satt til 0xAAAA for alle datapakker.
16	8 bits som indikerer størrelsen på data pakken. Denne skal kalkuleres med seg selv og CRC. Minimum størrelse er 11 (da er innholdet i indeks 23, data, lik 0). Maksimal størrelse er 78
17-18	Til sammen 16 bits som inneholder subnet ID og enhets ID til serveren vi sender fra. Disse plassene er mer tilpasset for sending av kommandoer internt i HDL systemet, slik at disse verdiene er statisk satt på serveren.
19	16 bits som inneholder enhets typen til senderen, eller i dette tilfelle server. Dette feltet er også mer tilpasset for sending av kommandoer internt i HDL systemet. Enhets type er derfor også statisk satt på serveren.
20	Kommando som skal utføres av mottakerenheten.
21	Mottakerenhetens subnet-ID
22	Mottakerenhetens enhets-ID
23	Tilleggsdata som sendes med. Denne varierer ut ifra hva slags enhet det sendes til og hva slags kommando som skal utføres av enheten.
24	CRC som kalkuleres ut ifra innholdet på pakken. Er ikke CRC'en korrekt, vil ikke pakken prosesseres.

Eksempel på hvordan forskjellige pakker vil se ut:

Merk: Indekser med mer enn én byte i oversikten over, vil ha mer enn en rad med innhold i disse eksemplene, ettersom hver rad tilsvarer en byte.

Verdiene er lest ut med følgende kodeeksempel:

```
for (byte b : HDLData) {
    System.out.println(b & 0xFF);
}
```

HDLData er bytearrayen som sendes til HDL. Løkka går igjennom hver byte i bytearrayen.

Under er et eksempel på en kommando hvor brukeren etterspør temperatur i varmestyringskontrolleren, kanal nummer seks (kontor):

Byte nummer	Innhold	Bemerkelse
1	192	IP del 1
2	168	IP del 2
3	10	IP del 3
4	141	IP del 4
5	72	H
6	68	D
7	76	L
8	77	M
9	73	I
10	82	R
11	65	A
12	67	C
13	76	L
14	69	E
15	170	Leder kode

16	170	Leder kode
17	12	Data lengde
18	1	Subnet ID til sender
19	254	Enhets ID til sender
20	254	Enhets type til sender
21	255	Enhets type til sender
22	28	Kommando
23	94	Kommando
24	1	Subnet ID til mottaker
25	12	Enhets ID til mottaker Varmestyringskontrolleren
26	6	Data – kanalen på varmestyringskontrolleren
27	220	CRC
28	229	CRC

Eksempel på kommando hvor brukeren endrer på temperatur i varmestyringskontrolleren, kanal 4 (soverom):

byte 1-16 ekskluderes her ettersom verdiene er tilsvarende lik for alle pakker som sendes.

17	21	Datalengde
18	1	Subnet-ID til sender
19	254	Enhets-ID til sender
20	254	Enhets type til sender
21	255	Enhets type til sender
22	28	Kommando
23	92	Kommando
24	1	Subnet ID til mottaker
25	12	Enhets ID til mottaker Varmestyringskontrolleren
26	4	Data – kanalen på varmestyringskontrolleren
27	0	Data - Nåværende arbeidstype Satt til 0 som er «Heating». Andre verdier er: 1 = Cooling 2 = Heating Power Output 3 = Cooling Power Output
28	0	Data - Temperaturtype Satt til 0 for Celsius Andre verdier er: 1 = Fahrenheit
29	2	Data – Modus kanalen er satt til Satt til 2 som vi har definert for dag. Andre verdier: 1 = Ferie 3 = Natte 4 = Borte
30	14	Temperatur ferie
31	22	Temperatur dag

		Temperaturen det har blitt satt til
32	18	Temperatur natt
33	16	Temperatur borte
34	1	Status på ventilene Satt til 1 for på Andre verdier er: 0 = Av Om den skal være av eller på regnes ved om ønsket temperatur er mindre enn nåværende temperatur i varmesonen.
35	15	Hvor lenge ventilen skal stå på før ventilene stenges
36	74	CRC
37	223	CRC

C Z-Wave-kommandoeksempler

```
/**
 * This Method retrieves data from Domoticz local Web API. Firstly, the method does a
 * http request with JSON on the Web API. After this the result is processed.
 * Scanners filter out unnecessary data and looks for humidity information from
 * humidity sensor devices, it also retrieves the id of the device.
 * After the request has been fully processed. The result is then processed and the
 * values of each device is saved to the correct user for when the user requests this
 * information.
 * @throws MalformedURLException throws exceptions that occurs when doing a http request
 * @throws IOException throws IOExceptions
 */
public void retrieveZWaveData() throws MalformedURLException, IOException {
    String zWave;
    InputStream response = new
URL("http://127.0.0.1:8080/json.htm?type=devices&filter=temp&used=true&order=Name").openStream();
    try (Scanner scanner = new Scanner(response)) {
        String responseBody = scanner.useDelimiter("\\A").next();
        String result = responseBody.substring(responseBody.indexOf("\"result\" : ["),
responseBody.indexOf("\"status\" : \"OK\""));
        String result2 = "Humidity ";
        try(Scanner scanner1 = new Scanner(result)){
            while(scanner1.hasNextLine()){
                String humidityLine = scanner1.nextLine();
                if(humidityLine.contains("\"Humidity\"")) {
                    String hum = humidityLine.substring(humidityLine.indexOf(": "));
                    hum = hum.replaceAll("[^0-9]", "");
                    result2 = result2 + " " + hum;
                    while(scanner1.hasNextLine()) {
                        String idLine = scanner1.nextLine();
                        if(idLine.contains("\"idx\"")){
                            String id = idLine.substring(idLine.indexOf(": "));
                            id = id.replaceAll("[^0-9]", "");
                            result2 = result2 + " " + id + "\n" + "Humidity ";
                            break;
                        }
                    }
                }
            }
        }
        zWave = result2;
    }

    String[] zWaveHum = zWave.substring(0, zWave.length()).split("Humidity");
    String[] humData = new String[zWaveHum.length];
    for (int i=1, len=zWaveHum.length; i<len; i++) {
        humData[i] = String.valueOf(zWaveHum[i].trim());
    }
    for(int i=1; i < humData.length; i++) {
        if(!humData[i].isEmpty()){
            String sensorId = humData[i].substring(humData[i].indexOf(" ") + 1,
humData[i].length());
            String sensorHum = humData[i].substring(0, humData[i].indexOf(" "));
            System.out.println(sensorId + " " + sensorHum);
            int n = 1;
            if(n <= 6) {
                user.get(1).setHumidityInfo(i, sensorId, sensorHum);
                user.get(0).setHumidityInfo(i, sensorId, sensorHum);
                n++;
            }
            else if(n >= 7 && n < 12 ) {
                user.get(2).setHumidityInfo(i, sensorId, sensorHum);
                user.get(0).setHumidityInfo(i, sensorId, sensorHum);
                n++;
            }
            else if( n >= 12 && n <= 13) {
                user.get(3).setHumidityInfo(i, sensorId, sensorHum);
                user.get(0).setHumidityInfo(i, sensorId, sensorHum);
                n++;
            }
        }
    }
}
```

```
/**
 * Starts a thread which updates each user's Z-Wave data every
 * 10th minute. The information is saved for later use.
 */
private void startZWaveDataReceiver() {
    executorService.execute(() -> {
        while (!shutdown) {
            try {
                TimeUnit.SECONDS.sleep(600);
                retrieveZWaveData();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

D Prosjektplan – levert 28. januar 2016

Prosjektplan

1 MÅL OG RAMMER

1.1 Bakgrunn

1.2 Prosjekt mål

1.2.1 Effektmål

1.2.2 Resultatmål

2 OMFANG

2.1 Problemområde

2.2 Avgrensning

2.3 Oppgavebeskrivelse

3 PROSJEKTORGANISERING

3.1 Ansvarsforhold og roller i gruppen

3.2 Rutiner og regler for gruppen

4 PLANLEGGING, OPPFØLGING OG RAPPORTERING

4.1 Karakteristika ved prosjektet

4.1.1 Valg av utviklingsmodell

4.1.2 Anvendelse av systemutviklingsmodell

4.2 Metode

4.3 Liste over aktiviteter

4.4 Plan for statusmøter og beslutningspunkter

5 ORGANISERING AV KVALITETSSIKRING

5.1 Dokumentasjon

5.2 Risikoanalyse

5.2.1 Håndtering av projektrisiko

6 PLAN FOR GJENNOMFØRING

[6.1 Detaljer om Gantt-fremdriftsplan\(se 6.3\)](#)

[6.2 Milepæler](#)

[6.3 Gantt-fremdriftsplan](#)

1 MÅL OG RAMMER

1.1 Bakgrunn

Utvikling av smarthus er i stor vekst. Flere og flere ser mulighetene i dette markedet ettersom teknologien har kommet langt på vei. Kinesiske HDL mener at smarthus-teknologien har et enormt potensiale bare den blir enkel og billig nok. Selv om det å bygge smarthus er forholdsvis dyrt i Norge, er økning av smarthuskomponenter på verdensbasis og økt konkurranse med på å få prisene ned. Flere og flere begynner å se gevinsten med å bruke smarthusløsninger. Et smarthus som er bygget med ivaretagelse av energi i fokus vil ikke bare være med på å redusere strømforbruk til de som bruker det, det vil gi et bedre inneklima og være med på å forenkle hverdagen til beboerne i huset. Avhengig av hva slags systemer brukeren velger ut kan alt fra lys, varme og ventilasjon enkelt overvåkes og styres automatisk. Beboerne kan få økt sikkerhet ved at systemet styrer lås og alarm i huset og kan raskt gi beskjed hvis noe er feil. Det vil også være med på overta mer praktisk arbeid som styring av vinduer, dører, kaffemaskin, musikkanlegg og garasjeporter for å nevne noen muligheter.

For at disse systemene skal bli populære er det kritisk at de stiller opp til forbrukernes krav til robusthet, fleksibilitet og skalerbarhet. Leverandørene er avhengig av å ha systemer som alltid fungerer. Dette er svært viktig for mer kritiske funksjoner. Selv om systemene begynner å bli velutviklet, er prisene fortsatt for høye til at de mest avanserte og komplette systemene kan brukes i normale hjem. I tillegg kan mange systemer virke overveldende.

Oppdragsgiver vil at vi skal ta utgangspunkt i et fungerende åpent system som er bygget med brukervennlighet og fleksibilitet i fokus, og bygge videre på dette slik at systemet passer hennes behov.

1.2 Prosjektmål

1.2.1 Effektmål

- Systemet skal bidra til at oppdragsgiver oppnår et lavere og mer bevisst strømforbruk sammenlignet med norsk gjennomsnittsforbruk i tomannsboliger.
- Systemet skal legges til rette for at oppdragsgiver enkelt skal kunne holde oversikt over sitt eget forbruk, samt det totale forbruket i hele tomannsboligen og hver boenhet separat.
- Systemet skal bli laget på en måte som gjør det enkelt og bruke, slik at beboere av boenhetene enkelt skal kunne lese av sitt eget strømforbruk og bruke systemet uten teknisk kompetanse.
- Systemet skal legges til rette for at oppdragsgiver skal kunne legge til nye sensorer og nye komponenter.

1.2.2 Resultatmål

Målet med prosjektet er å oppnå redusert strøm- og vannforbruk ved å utvikle et system som kan overvåke og kontrollere hvordan huset kan spare energi og distribuere den energien som brukes på en mer effektiv måte. Prosjektet inkluderer overvåkningsverktøy som lar brukeren få mulighet til å overvåke og kontrollere varmepumpe og ventilasjonsanlegg, lys og vinduer. Styringssystemet skal fungere som en sentral hvor man kan styre varme, lys og takvinduer.

2 OMFANG

2.1 Problemområde

Datamaskiner har gått fra å utgjøre en liten del av livene våre til å nå være en del av det aller meste. Det finnes blant annet nå smarte kjøleskap som holder oversikt over hva du har i kjøleskapet, og som man via en mobilapplikasjon kan se innholdet når man er på butikken. Det kommer flere slike smarte enheter på markedet hele tiden, og det blir etterhvert mange løsninger å forholde seg til.

Folk flest har i dag boliger der alt fra enkle til mer komplekse oppgaver blir styrt manuelt av beboere. Alt av styring og kontroll er desentralisert til enkelte objekter, eksempelvis én for varmpumpen og én for varmekablene på badet. Dette medfører altså at dersom man ønsker å stille temperaturen i boligen, så må man gå bort til for eksempel varmpumpen og stille inn ønsket temperatur. Disse vil man, spesielt om man sammenligner med flere av dagens løsninger, gjerne kalle dumme hus. Heldigvis er området i rask utvikling, og flere aktører kommer stadig på banen med sine egne løsninger. Dette medfører samtidig at dersom man ønsker å bygge ny eller oppgradere sin egen bolig må man enten nøye seg med flere styringssystemer slik at man får én for belysning og én for ventilasjon og temperaturregulering. Eventuelt kan man velge én løsning som styrer noe mer, for eksempel belysning, temperatur og ventilasjon, men ikke vinduene. Styringssystemer brukes mer og mer, og flere hus bygges nå med slike løsningen inkludert. Vi kaller boliger med (sentraliserte) styringssystemer for smarthus.

2.2 Avgrensning

Ut ifra hvordan oppgaven er satt opp er det nødvendig å sette noen avgrensninger for oppgaven. I forhold til tidsfrister og formelle krav av bacheloroppgave er disse satt av NTNU.

Dette prosjektet er avgrenset til ett boligprosjekt i Kragerø og styringssystemet som skal utvikles for boligen. Videre vil vi holde oss til de rammer som er satt i oppgavebeskrivelsen gitt av oppdragsgiver og rammer som er blitt satt på grunn av hva og hvordan styringssystemene vi skal ta utgangspunkt i er utviklet. På grunn av hvordan oppgaven er satt opp og for å spare tid, vil vi gjenbruke funksjoner i HDL-systemet så langt det lar seg gjøre.

Systemets server skal ha rekkevidde til alle styringssystemene i alle boenhetene. Hvis signalet er for ustabil eller svakt må det plasseres signalrepeatere for å øke rekkevidden.

Mobilapplikasjonen vil hovedsaklig kun brukes for kommunikasjon til styringssystemet, slik at det er det stasjonære styringssystemet som er i boenhetene som vil ha det meste av funksjonaliteten og foretar endringer. På grunnlag av kompetanse i gruppen, samt hvor mye tid vi har til rådighet vil mobilapplikasjonen kun bli utviklet til Android. Vi ser derfor bort ifra ytterlige økonomiske kostnader for beboerens bekjøpelse av Android-enheter.

Oppgaven er lagt opp på den måten av vi selv kan gjøre eksperimenter på grønn teknologi, ved å avgjøre hva som skal overvåkes av styringssystemer i tillegg til de kravene som er oppgitt. Vi står derfor fritt til å selv velge ut hvordan vi vil løse oppgaven så lenge retningslinjene blir fulgt. Dette avgrenses til hva som faktisk er mulig å få til ut ifra systemene vi skal forholde oss.

Dette gjelder:

- Er det praktisk mulig å kombinere systemene?

- Hva slags endringer må til for å få til et fungerende system?
- Hva slags tilleggskomponenter må installeres for å koble sammen systemene?

Vi forutsetter at alle hardware- og elektriske komponenter fungerer når systemet skal installeres. I tillegg forutsetter vi at oppdragsgiver legger til rette for at alle komponenter og sensorer som trengs installeres.

2.3 Oppgavebeskrivelse

Hensikten med prosjektet er å legge til rette for smarthus med spesiell vekt på effektiv energiutnytting og optimalt inn klima i et siporexhus med to boenheter og en hybel/arbeidsrom. Energieffektivitet skal sikres ved optimal utnyttelse av lagret termisk energi og naturlig ventilasjon i kombinasjon med varmluftgjenvinning og tilførsel av varme fra bergvarmepumpe til vannbåren varme.

Det skal utvikles et styringssystem som overvåker varmeproduksjonen fra varmepumpen og strømforbruket til ulike komponenter i bygget. Det skal kunne måles hvor mye varmtvann som går fra varmepumpen til de tre ulike boenheter.

I tillegg til overvåkning skal styringssystemet kunne gjøre:

- Styre temperaturer i de ulike temperatursonene gjennom hele døgnet.
- Styre lys av og på, via demping og døgninnstillinger. Styring av andre elektriske komponenter kan også legges til.
- Styre åpning/lukking av tre Velux takvinduer i én av leilighetene. Disse vinduene skal registrere om det regner, generelle temperaturregulering og eventuelt vindretning.
- Styre den balanserte ventilasjonen (LUNOS) som er desentralisert til de ulike rommene.
- Eventuelt:
 - Styre/regulere inn klimaet bedre - utover temperaturen, for eksempel luktfuktighet.
 - Måling av utetemperatur, vind og vindretning samt nedbør.
 - Tilpasse temperaturregulering inne til forandringer i utetemperatur
 - Utforske og eksperimentere og komme med forslag til annen teknologi som kan inngå i huset.

Med utgangspunkt i dette blir gruppens oppgave å prosjektere og velge ut komponenter som for det meste er bygd på åpne standarder. Styringsprogramvaren vil basere seg på åpen kildekode fra forskjellige produsenter (Velux, Vaillant, HDL Nordic og LUNOS). Disse programmene/systemene skal dersom det er mulig kombineres og danne en så kompakt applikasjon som mulig i motsetning til separerte systemer. En del av oppgaven vil derfor bli å kartlegge hva HDL Nordic sitt styringssystem kan gjøre, sette inn alle elementene som skal eller kan brukes og deretter bygge broene som skal til for å få alt fungerende og samlet. Her blir det opp til gruppen selv å vurdere hva som vil være den beste måten å få til dette.

Systemet skal også kunne styres manuelt fra et sentralt nettbrett eller Android-enhet (applikasjon).

3 PROSJEKTORGANISERING

3.1 Ansvarsforhold og roller i gruppen

Prosjektleder er Henrik Haukaas. Resterende gruppemedlemmer er Jørn-Are Flaten og Edvard Brekke. Oppdragsgiver er Vanja Ohna fra NTNU i Gjøvik, avdeling HOS. Prosessveileder er Frode Haug fra NTNU i Gjøvik, avdeling AIMT. Kontaktperson i HDL er Haavard Haaland, og teknisk sjef i HDL Morten Andersen. Kontaktperson i LUNOS er Morten Storegjerde. Rørlegger er Roger Nilsen.

3.2 Rutiner og regler for gruppen

Se vedlegg A (grupperegler)

4 PLANLEGGING, OPPFØLGING OG RAPPORTERING

4.1 Karakteristika ved prosjektet

Prosjektet er å utvikle et styringssystem for å automatisk styre og overvåke forskjellige komponenter i et smarthus eller manuelt via en mobilapplikasjon på Android. Prosjektet blir utført av tre gruppemedlemmer med tilgang til faglige veiledere. I løp av testfasen kommer vi til å besøke smarthuset igjen, for å teste systemet i praksis.

4.1.1 Valg av utviklingsmodell

Ved valg av utviklingsmodell vil vi velge modellen som passer oss best og som vi mener vi vil oppnå høyest effektivitet med. To av gruppens medlemmer har tidligere erfaring med Scrum. Scrum er en metode vi føler vil passe dette prosjektet og gruppen godt. Den er mottakelig for endringer underveis, og passer godt for små utviklerteam. Dette er svært viktig i vårt prosjekt. Scrum gjør det også mulig å jobbe på flere moduler av applikasjonen samtidig.

4.1.2 Anvendelse av systemutviklingsmodell

Vi velger å gå for 1 ukers sprinter, der vi holder Sprint Review og sprint retrospective-møter etter 1 ukes arbeid. Vi vil bruke ca 15 minutter daglig på daily scrum. Sprintene vil bli holdt på en ukentlig basis, på grunnlag av oversiktighet. Vi mener at 2 uker blir for langt i forhold til at vi jobber tett sammen hver eneste dag.

Det kommer til å bli en delvis fri sprint produktkø, der utviklerene kan velge hvilke oppgaver som de vil gjøre. Mens de oppgaver vi trenger ekspertise utenfra, vil vi rådføre oss med faglig veileder. Etter utviklingsperioden og en liten påskeferie vil vi gå inn i testfasen. Her vil vi teste moduler på campus, simulasjonstester i HDL-programvaren og en test på lokasjon i Kragerø.

4.2 Metode

I løpet av prosjektet kommer vi til å bruke Google Docs til loggføring og forfating av dokument frem til og med første delinnlevering er levert. Etter denne innleveringen vil vi gå over til Microsoft OneDrive og Microsoft Office 365. Dette fordi at Google Docs har mangler vi oppdaget etter vi hadde startet. Prosjektstyring vil hovedsaklig foregå via Trello, Smartsheet og Basecamp. Til eventuell bilderedigering vil vi benytte gratisprogrammet Gimp. Til utvikling av kode kommer gruppen til å nytte Eclipse for Java, Microsoft Visual Studio for C/C++ og Android Studio for Android. Vi vil også bruke GitHub for enkelt samarbeide om kode. Vi vil muligens bruke Iridium Mobile og dets moduler, som for eksempel værmoduler for applikasjonen vår.

4.3 Liste over aktiviteter

- Prosjektplanlegging
- Tur til Kragerø den 25.01.16
- Utvikling av styringssystem
- Teste styringssystemet på komponenter på NTNU.
- Tur til kragerø senere i vår når vi kan teste programvaren på lokasjon.

4.4 Plan for statusmøter og beslutningspunkter

Gruppen vil møtes fem ganger i uken (mandag - fredag). Hovedsakelig vil gruppen jobbe sammen på grupperom. Her vil gruppen arbeide tett sammen selv om mye av tiden vil være individuelle oppgaver. Valget om å jobbe i samme rom er tatt for å oppnå best mulig effektivitet, kommunikasjon og samhold i gruppen. Derfor vil vi kunne ta beslutninger fortløpende siden vi alltid har tett kontakt.

5 ORGANISERING AV KVALITETSSIKRING

5.1 Dokumentasjon

For å sikre dokumentasjon av arbeidet gruppen utfører skal det føres logg over gjøremålene gruppen har foretatt seg hver dag. Samtidig skal det etter beste evne føres dokumentasjon i koden som gruppen vil generere. Loggen som føres etter hver endte dag vil beskrive oppgaver, utfordringer og tanker fra hver deltaker.

5.2 Risikoanalyse

I et prosjekt med så få medlemmer som dette vil enkelte hendelser kunne ha store konsekvenser. Konsekvenser som for eksempel frafall vil medføre ekstra arbeidstimer for de gjenværende. For å holde oss til et enkelt graderingssystem bruker vi gradene: liten, moderat og stor.

Konsekvens / risiko	Sannsynlighet	Konsekvenser
Frafall	Liten	Problematisk
Kompetansemangel	Stor	Problematisk
Teknologiske begrensninger	Liten	Problematisk
Tap av arbeid	Liten	Katastrofal
Ukombinerlige systemer	Stor	Problematisk
Styringssystemet fungerer enten ikke etter hensikt eller i det hele tatt	Moderat	Katastrofal
Stans eller kansellering av byggeprosjekt	Liten	Katastrofal

5.2.1 Håndtering av prosjektrisiko

Siden vi er en så liten gruppe som det vi er, vil selv de minste risikoene kunne få store følger.

Derfor er det desto viktigere for hvert medlem av gruppen å kunne tette et hull som oppstår som

følge av en hendelse.

Frafall	For å redusere skaden ved frafall må alle medlemmer kunne være i stand til å fylle hullet. Vi har alle et felles mål og vi er klar over hva vi skal ha med i applikasjonen. Problemet blir en tidsøke på prosjektet.
Kompetansemangel	Ved kompetansemangel vil gruppen måtte anskaffe den kompetansen den mangler ved selvstudier og ved hjelp av veileder og/eller respektiv kontaktperson.
Teknologiske begrensninger	Se etter andre teknologiske muligheter for å løseproblemet. Eventuelt revurdere målene.
Tap av arbeid	Nesten neglisjerbar risiko. Vi fører tekst rett inn i Google Docs, slik at det blir tatt back-up kontinuerlig.
Ukombinerlige systemer	Ser etter andre muligheter, evt nøye oss med flere styringsapplikasjoner/styringssystemer.
Styringssystemet fungerer enten ikke etter hensikt eller i det hele tatt	Omprogrammering. For å forhindre dette vil vi gjennomføre simuleringer ved bruk av simuleringsprogramvare.
Stans eller kansellering av byggeprosjekt	Kontakte skolen og høre om vi får fullføre oppgaven.

6 PLAN FOR GJENNOMFØRING

6.1 Detaljer om Gantt-fremdriftsplan(se 6.3)

Planen ut januar går stort sett med til å få i orden grupperegler, prosjektavtale og prosjektplan. Finne ut hvor store rammene av prosjektet er, hva som er mulig å få til, hvilke alternativer vi har der det er hinder på grunn av lukkede systemer for eksempel er en av de større utfordringene i utarbeiding av prosjektplanen. Arbeidet med kravspesifikasjonen vil starte i slutten av januar og foregå nesten to uker ut i februar. I februar vil vi starte utviklingen og bruke

utviklingmodellen Scrum. Sprintene vil vare i 1 arbeidsuke og hver sprint inneholder planlegging, oppgaver(tasks), sprint review, sprint retrospective og sprint produktøk. I tillegg vil vi ha 15 minutter daily scrums. Etter å ha gjennomført 5 av 8 sprinter vil vi ta en ukens påskeferie før vi fullfører de siste 3 sprintene. Etter hver sprint vil vi bruke en halv dag til testing av ukens arbeid. Vi vil også ha møte med oppdragsgiver etter hver sprint.

Det regnes med at litteraturstudier vil foregå under hele utviklingsperioden. Litteraturstudier vil i tillegg til selvstudier inkludere arbeid med å gjøre oss kjente med alle systemene vi skal bruke. Dette inkluderer LUNOS, HDL sitt styringssystem, Velux Integra, Iridium Mobile, Vaillant, kommunikasjonsprotokoller (som Real-time Transport Protocol) og kommunikasjonsteknologi (for eksempel Z-Wave), samt hvordan disse fungerer. Helt i starten av utviklingsperioden vil vi også bruke noe tid på å sette oss inn i koden til HDL, samtidig som vi skriver kravspesifikasjonen, for å passe på å få kravspesifikasjonen så detaljert og god som mulig.

Milepæler er listet opp og markert for når vi forventer å ha oppnådd disse. Dette inkluderer milepæler under kategoriene styringssystemer, mobilapplikasjon, testing mm. og delinnleveringer av statusrapporter. Delinnleveringsfrister er 20. februar, 18. mars og 21. april. Bloggen vil bli regelmessig oppdatert fra og med slutten av januar etter første innlevering er gjennomført. Samtidig som vi starter første uken med Scrum vil vi arbeide med å få nettsiden vår opp så vi kan laste opp oppdateringer og blogginnlegg. Selve rapporten for hele oppgaven skal være ferdig og levert innen 18.05.16. Etter denne er levert vil vi bruke den resterende tiden til å lage plakat og muntlig presentasjon.

6.2 Milepæler

Under prosjektet har vi valgt å ha flere milepæler for å oppnå kort avstand mellom hver milepæl. Dette vil ha en motiverende effekt på gruppens deltagere.

Rapportrelaterte milepæler

1. Levere forprosjektrapport
2. Fullføre kravspesifikasjon
3. Innlevering av statusrapport 1
4. Innlevering av statusrapport 2
5. Innlevering av statusrapport 3
6. Innlevering av fullført rapport

Vi har valgt følgende milepæler for det stasjonære styringssystemet:

7. Sette oss inn i HDL-styringssystemet
8. Fungerende funksjon for lyskontroll.
9. Fungerende funksjon for åpning- og lukking av vinduer.
10. Fungerende funksjon for ventilasjon.
11. Fungerende funksjon for temperaturregulering.

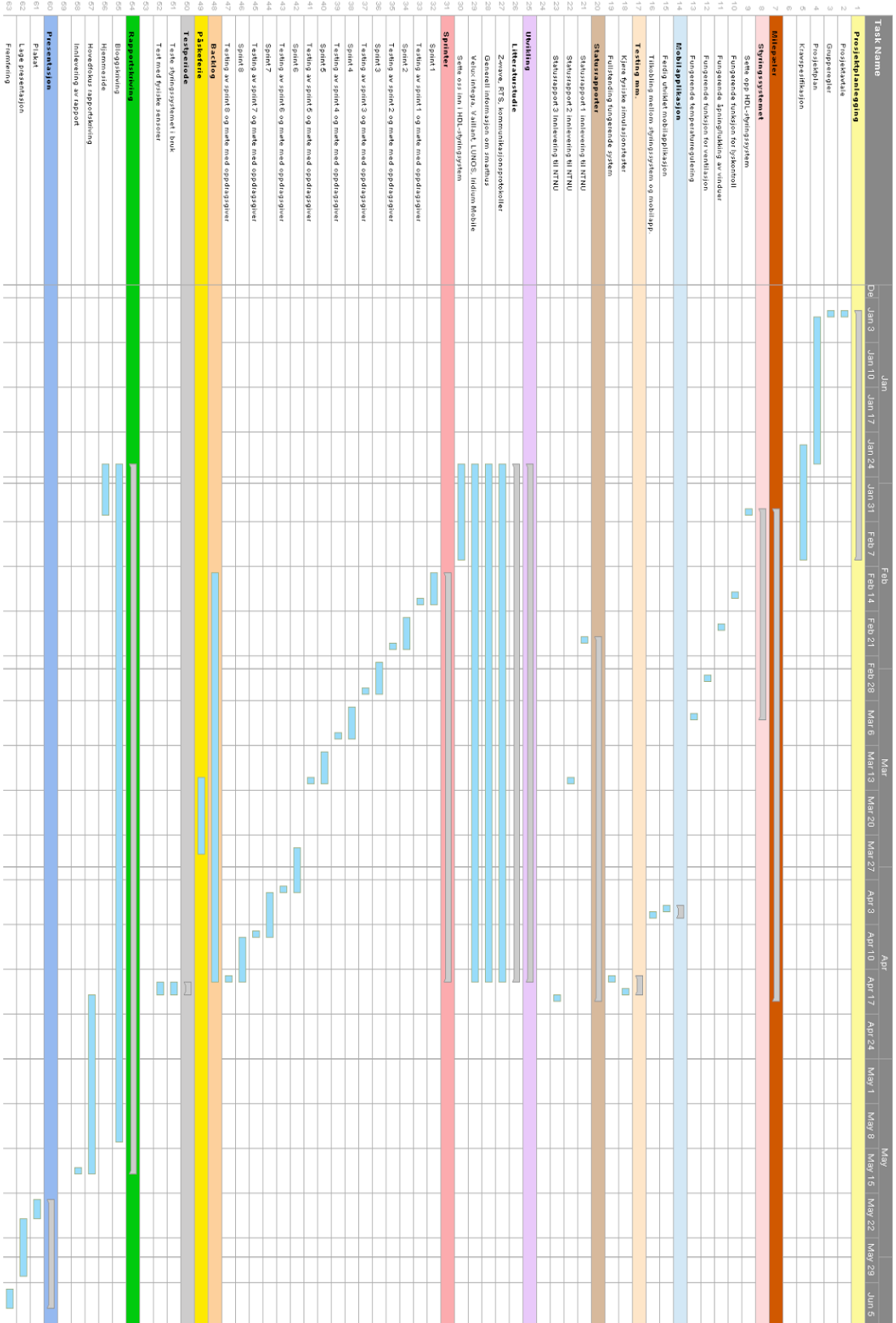
Vi har valgt følgende milepæler for mobilapplikasjonen:

12. Utviklet mobilapplikasjon.
13. Satt opp fungerende tilkobling mellom mobilapplikasjon og styringssystemet.

Når mobilapplikasjonen og styringssystemet er ferdig vil det gjenstå:

14. Fått kjørt fysiske simulasjonstester
15. Fullstendig fungerende system

6.3 Gantt-fremdriftsplan



PROSJEKTAVTALE

mellom Høgskolen i Gjøvik og oppdragsgiver Vanja Ohna

for hovedprosjektet Smarthus

utført av studentene Edvard Brakke, Henrik Hauhaas, FØrn - Are Flaten

Avtalepartenes plikter vedrørende gjennomføringen av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer;

1. Studenten(e) skal gjennomføre prosjektet i perioden fra 05/01-16 til 18/05-16
Studentene skal i denne perioden følge en oppsatt framdriftsplan der HiG 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 HiG å stille med sensor vederlagsfritt. Vedkommende person må tilfredsstillende vanlige krav til sensorer ved universiteter og høyskoler. Om oppdragsgiver ikke disponerer slike fagpersoner kan høgskolen skaffe og bekoste ekstern sensor.
2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:
 - Oppdragsgiver dekker selv gjennomføringen av prosjektet når det gjelder f.eks. materiell, telefon/fax, reiser, nødvendig overnatting på steder langt fra HiG. Studentene dekker utgifter for trykking og ferdigstilling av den skriftlige besvarelsen vedrørende prosjektet.
 - Eiendomsretten til eventuell prototyp tilfaller den som har betalt de komponenter og det materiell osv. 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. HiG 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. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.
4. Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode, disketter, taper etc. som inngår som del av eller vedlegg til besvarelsen, blir HiG's eiendom og kan vederlagsfritt benyttes av HiG til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av HiG 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 HiG og/eller studenter har interesser.
Besvarelser med karakter C eller bedre registreres og plasseres i skolens bibliotek. Det legges også ut en elektronisk prosjektbesvarelse uten vedlegg på bibliotekets del av skolens Internett-sider. Dette avhenger av at studentene skriver under på en egen avtale hvor de gir biblioteket tillatelse til at deres hovedprosjekt blir gjort tilgjengelig i papir og nettgave (jfr. Lov om opphavsrett). Oppdragsgiver og veileder godtar slik offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og instituttleder om de i løpet av prosjektet endrer syn på slik offentliggjøring.

5. Besvarelsens spesifikasjoner og resultat er oppdragsgivers eiendom, og kan anvendes i oppdragsgivers egen virksomhet. Gjør studenten i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom oppdragsgiver og student bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.
6. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten 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 og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.
7. Studenten leverer 3 - tre - eksemplarer av oppgavebesvarelsen med vedlegg til bedømmelse, hvorav oppdragsgiver skal ha en. HiG kan stille til disposisjon ytterligere eksemplar(er) for oppdragsgiver mot at denne godtgjør produksjonskostnader.
8. Denne avtalen utferdiges med et eksemplar til hver av partene. På vegne av HiG er det instituttlederen som godkjenner avtalen.
9. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og HiG som nærmere regulerer forhold vedrørende bl.a. konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene.

Dersom oppdragsgiver og student(er) ønsker en videre eller ny avtale, skjer dette uten HiG som partner.
10. 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, kap 32.

Gjøvik den 05/01-16

Studenter: Edward Brekke
John-Mat Fjellén
Henrik Hamkants

Faglærer / veileder: Arne Røis

Wanji Oppdragsgiver Instituttleder

Grupperegler

Del 1 - Generelt

1. Beslutninger fattes i flertall. Ingen har veto.
2. Prosjektleder er Henrik Haukaas. Stillingen går ikke på rundgang.
3. Dersom det påløper kostnader som ikke dekkes av oppdragsgiver, vil disse bli delt likt mellom gruppens medlemmer.
4. Alle har rett til å signere på vegne av gruppen. Dette gjelder ikke dersom det skal besluttes en avgjørelse.
5. Arbeidstid er mandag-fredag, kl 0900-1600, med unntak av forelesninger.
6. Gruppen skal så langt det er mulig jobbe samlet hver arbeidsdag. Ved dårlig vær (regn og snø) aksepteres hjemmekontor.
7. Backup foregår i Google Drive (overgang til Microsofts' One Drive etter 28. januar).
8. Rapport føres i Google Docs (overgang til Microsofts' One Drive etter 28. januar).
9. Logg føres i Google Docs (overgang til Microsofts' One Drive etter 28. januar) hver endte dag og med en ukentlig oppsummering. Dette inkluderer hva som er blitt gjort, antall arbeidstimer og andre vesentlige punkter.
10. I tilfeller der medlemmer ikke kan møte opp, det være seg ferie eller andre grunner, må medlemmet ta igjen utsatt arbeid på egen fritid.
 - a. Ved planlagt fravær skal det meldes i fra om i rimelig tid, dvs minst 24 timer før den aktuelle fraværperioden begynner.
 - b. Ved sykdom skal det meldes i fra om før den aktuelle arbeidsdagen begynner.
11. Ferier er i utgangspunktet ikke tillatt, men tillates ved felles enighet.
 - a. Avtalt ferieperiode er påske, dvs fom 21/03 tom 28/03.

Del 2 - Prosjektlederens fullmakter

1. Prosjektleder kan fatte avgjørelser som gjelder prosjektdetaljer dersom resterende medlemmer ikke er tilstede.
2. Prosjektleder har hovedansvar for delegering av arbeidsoppgaver.
3. I tilfeller der ett gruppe medlem er borte og utilgjengelig vil prosjektlederens stemme telle for to.

Del 3 - Advarsler og ekskludering

1. I tilfeller der medlem ikke utfører avtalt arbeid vil man få en muntlig advarsel. Hvis situasjonen ikke bedres vil medlemmet motta en skriftlig advarsel. I slike (og lignende) tilfeller vil gruppen møtes og diskutere hvordan manglende eller utsatt arbeid skal utføres.
2. Ved flere enn to advarsler samles gruppen til drøftingsmøte. Det skal drøftes hva som kan gjøres for at situasjonen skal korrigeres, dvs at medlemmet begynner å utføre avtalt arbeid til avtalt tid.
3. Dersom situasjonen ikke forbedres skal gruppen samles til møte med prosessveileder. Det vil bli opp til prosessveileder hvorvidt det er grunn for avskjed, eventuelt hvilke tiltak som kan tas for å bedre disiplin.

4. Logg kan brukes som bevis ved drøfting om avskjedigelse.

Kontaktinformasjon

Jørn-Are Flaten, studentnr 120459:

Klasse: 13HBIDATA

Tlf: 46 81 14 70

e-post: *nosfera.gnewm@hotmail.com* og *jorn-are.flaten@hig.no*

Edvard Brekke, studentnr 130682:

Klasse: 13HBIDATA

Tlf: 41 76 41 24

e-post: *edvard_brekke@me.com* og *edvard.brekke@hig.no*

Henrik Haukaas, studentnr 131205:

Klasse: 13HBIDATA

Tlf: 47 87 47 25

e-post: *henrik.hauk.92@gmail.com* og *henrik.haukaas@hig.no*

Edvard Brekke

Henrik Haukaas

Jørn-Are Flaten

Edvard Brekke

Henrik Haukaas

Jørn-Are Flaten

G Statusrapporter

Statusrapport 1

1 Oppsummering av arbeidet hittil og videre arbeid

Hittil i prosessen har vi jobbet med å få ferdig innlevering 1, som er prosjektplanen. Vi har også fått opp bloggen. Bloggen finnes under www.smarthusgruppen.weebly.com. Vi ønsket å bruke Weebly i stedet for Wordpress grunnet at vi ikke ville begrense oss til Wordpress-layoutene som universitetet har valgt.

Ellers i februar så har vi jobbet med å finne ut av hva vi skal ha med i styringssystemet både av utstyr i hus og funksjonalitet i applikasjonen. Vi har hatt kontakt med rørlegger, Vaillant (leverandør av varmepumpe) og Velux (leverandør av takvinduer) og HDL Nordic (leverandør av smarthusløsninger). Vi har også vært på prosjektbefaring i Kragerø. Boligen som er en horisontaldelt tomannsbolig har vært under konstruksjon i lang tid nå og er nesten ferdigstilt. Vi kjørte ned sammen med oppdragsgiver og fikk en guidet tur gjennom boligen. Det ble snakket om hvilke løsninger som kunne passet i boligen, spesielt med tanke på ventilasjon. Ventilasjonen blir av typen balansert. Dette betyr i hovedsak av ventilasjonsutstyret fungerer som lunger, som synkronisert «puster» inn og ut.

Ellers har vi hatt mye kontakt med HDL Nordic, og fått kontaktperson der, Morten Andersen. Vi har valgt å benytte oss av HDL sine løsninger på utviklingsfronten. Dette gjelder HDL IntelliCenter som vi kan programmere på og Iridium Mobile for å utvikle mobilapplikasjonen.

2 Fremgangen i prosjektet

Vi har brukt en del tid på litteraturstudier og lære oss programmer vi kommer til å bruke fremover. Fremover jobber vi med å sette opp utstyrsliste for huset og generelt litt videre på kravspesifikasjonen og litt på rapporten.

Vi har jevnlig dialog med oppdragsgiver om ønsker og leverandører for å få tak i koden til utstyr de leverer.

Statusrapport 2

1. Oppsummering av arbeidet hittil og videre arbeid
 - 1.1 Utstyr
 - 1.2 Utvikling
2. Fremgangen i prosjektet
3. Organisering av arbeidsoppgaver fremover
4. Kontakt med veileder, oppdragsgiver og andre

1 Oppsummering av arbeidet hittil og videre arbeid

1.1 Utstyr

Utstyslisten for smarthus er for det meste komplett. Vi er klare på hva som trengs av utstyr for å få alt til å fungere. Det eneste som kan være litt uklart er vannmålingen og fuktighetsmåling. Komponentene som skal ta seg av dette kommuniserer på en annerledes måte enn HDL (det er mulig å kombinere vannmåling med HDL, men da er vi avhengig av et produkt som de ikke har tilgjengelig før sommeren), og det er vanskelig å få gjort tester. Det viktigste har vært å forsikre seg om at det er fysisk mulig å få opp denne kommunikasjonen. Dette har blitt gjort klart igjennom produktmanualer og produktbeskrivelser, samt dialog med leverandører. Allikevel får vi ikke gjort tester på disse før det er installert i huset.

Oppdragsgiver vil ha fuktighetsmåling i alle områdene hvor det er satt opp varmesoner (totalt 13 i huset). Ettersom bygging av huset er for det meste ferdig, vil det være ugunstig/dyrt å legge opp flere kablede enheter. Dette betyr at alle enhetene serveren skal kommunisere med må være trådløse. HDL har ikke trådløse fuktighetsmålere, derfor må vi gå igjennom en annen leverandør. Vi har funnet en som passer prosjektet og denne kommuniserer via Z-Wave. Da må vi i tillegg ha en Z-Wave-mottaker på SHK-server. Vi får ikke testet dette før delene har kommet, noe som må skje kjapt. Dette foregår parallelt ved siden av at vi ser på andre måter å kunne øke energieffektivitet og et forbedret inn klima.

Utover dette må komponentene og tilleggskomponentene som vi har valgt ut bestilles (noen er allerede bestilt) slik at vi kan få gjort en virkelig test på smarthuset i seg selv. Frem til nå har vi fått låne en «democase» av HDL (som fungerer som et minismarthus) til å gjøre tester på. Testingen har foregått gjennom en kablet tilkobling, men en trådløs tilkobling er ikke noe mer komplisert bortsett fra at rekkevidden spiller en større faktor. Man kan sette HDL-gatewayen (som kobles til SHK-server og resten av smarthus anlegget) i ulike moduser. Det samme gjelder smarthus komponentene. Man kan stille inn disse på en måte som gjør at alle enhetene fungerer som reléer og videresender signalet slik at det kommer frem der det skal.

SHK-server, som skal være en Raspberry Pi (vi har snakket om at Raspberry Pi 3 model B er den som er mest aktuell), må også bestilles. Denne må også ha tilleggskomponenter som gjør at den kan kjøre vår Java serverapplikasjon og kunne sende og motta de signalene som trengs. Dette gjelder spesielt Z-Wave, ettersom den har innebygd trådløst nettverkskort og Ethernet-port.

1.2 Utvikling

Vi har kommet godt i gang med utviklingen og vi har nådd flere milepæler. Vi har fått satt opp et fungerende styringssystem (via SHK-server til HDL sin blackbox) som har fungerende funksjoner for lyskontroll, åpning og lukking av vinduer, styring av ventilasjonssystemet (med

av funksjon og endring av hastighet). Alt dette kan også nå styres via SHK-applikasjon. Det gjenstår arbeid for å få til en fungerende temperaturstyring, men dette er noe vi håper å ha i orden i løpet av uken (sprint 3). Man kan endre på temperaturen, men det er fortsatt mer arbeid som gjenstår for at dette gjøres på en optimal måte. Håndteringen av meldingene som kommer fra HDL blackboxen må også forbedres.

Det må også gjøres en del arbeid på databasen vår slik at den inneholder alt av brukerinnstillinger, brukeropplysninger og påloggingsinformasjon.

Det som kommer til å bli hovedfokus når varmestyringen og meldingshåndteringen fungerer på serversiden, vil være å legge til smart-logikk, slik at huset kan styre seg selv. Dette kan for eksempel være justering av temperatur i forhold til årstider og ulike moduser for hva slags tid det er på dagen.

Det jobbes også med å forbedre layouten på mobilapplikasjonen. Vi har dialog med oppdragsgiver (kontinuerlig) slik at sluttresultatet blir et produkt hun føler er brukervennlig og oversiktlig.

2 Fremgangen i prosjektet

Tidskjemaet vårt har blitt en del endret på grunn av at prosjektløpet har utartet seg annerledes enn vi hadde tenkt. Selv om utviklingsarbeidet vi skal utføre er lik som den er oppført i prosjektskissen/beskrivelsen har vi vært helt avhengig av å hatt en del forkunnskapen. Denne forkunnskapen gjelder generelt å sette seg inn i smarthusteknologien generelt, hvordan alt fungerer sammen og hva som er mulig for oss å utføre i forhold til tidsfristen. I tillegg har det vært helt avhengig av å finne ut hva slags komponenter og tilleggskomponenter som skal installeres i huset, og i det hele tatt finne ut hvordan vi kan kommunisere med disse.

Når man skal forholde seg til flere kontaktpersoner er det vanskelig å forholde seg til hva som er mulig og ikke, ettersom man kan ikke forvente at alle har like tanker om dette. Som følger av dette har det vært mye frem og tilbake om hva vår rolle i prosjektet er. Det har også vært misforståelser i bruk av ordet «programmering». Vi har forholdt oss til at dette ordet brukes i forhold til å skrive kode og generell programvareutvikling, mens andre kontaktpersoner har brukt dette ordet som et annet ord for konfigurering. Vi mente og forstod det slik at i utgangspunktet, kunne vi tilføre kode til eksisterende løsninger ved å «programmere» dette inn. Dette gjelder hovedsakelig HDL sin egen server og smarthusløsning. Vi hadde blitt fortalt at dette hadde blitt gjort tidligere. Vi føler at dette skjedde selv om vi var tydelige på hva som faktisk skulle gjøres. Derfor valgte vi å gå for det vi hadde blitt fortalt på grunn av at dette virket som den desidert beste løsningen.

Senere hvor vi finner ut at det faktisk ikke er slik vi har trodd, blir det en fullstendig helomvending. Igjen kontakter vi HDL og forklarer situasjonen, og er enda mer tydelig på hva vi prøver å få til. Vi ville lage serveren og styringen helt selv, så langt det var mulig. Vi ble fortalt at dette var mulig og dermed utviklet prosjektet seg videre til å bli det det er nå.

Hovedpoenget her er misforståelser. Når man må sette seg inn i noe stort og nytt har vi lært at man må være svært tydelig på hva man faktisk prøver å få til. I tillegg kan det ha noe med at HDL sin tekniske avdeling ikke var knyttet direkte til denne oppgaven på forhånd, selv om det er de vi har jobbet mest med. De har kun stilt seg svært behjelpelige selv om de ikke har måttet gjøre det.

Det å forholde seg til mange kontaktpersoner, få svar på alt som har vært nødvendig og forsikre seg om at informasjonen vi har fått er riktig og kan brukes har tatt svært lang tid og vært

krevenne og til dels stressende arbeid. Dette har vært veldig viktig å ha fått på plass ettersom det er snakk om forholdsvis store kostnader for oppdragsgiver, i tillegg er oppdragsgiver avhengig av et system funker ettersom det skal brukes daglig.

På grunn av forskyvingen, har utviklingsarbeidet og dets sprinter blitt utsatt. Vi er nå i sprint 3, men ifølge vårt Gantt-skjema fra forprosjektet skulle vi vært i sprint 7. Selv om tid har blitt en knapp ressurs, er vi alle innstilt på få dette til og levere et godt sluttprodukt som oppdragsgiver har nytte av. Noe som er svært positivt er at vi har fått testet alt arbeidet underveis og dermed forsikret at det funker.

Utenom utviklingsarbeidet og administrativt arbeid, har vi ikke fått startet noe særlig på rapporten, men planen er å begynne med den når det meste av det som skal utvikles er ferdiggjort. Eventuelt, skal en av oss begynne å jobbe med rapporten mens de andre jobber med utviklingen.

3 Organisering av arbeidsoppgaver fremover

Jørn-Are jobber foreløpig med utvikling av mobilapplikasjonen og applikasjonen sin layout. Edvard jobber med databaser. Dette gjelder pålogging og brukerinnstillingsdatabase.

Henrik jobber med administrative oppgaver, utstysrliste, styring av varmestyringskontroller og analysing av pakker mottatt fra HDL blackbox til SHK-server. Jobber også med meldingshåndtering av meldinger mottatt fra mobilapplikasjonen sendt til serveren.

4 Kontakt med veileder, oppdragsgiver og andre

Vi har god kontakt med oppdragsgiver. Vi har kontinuerlig dialog om layout, utstysrliste og løsnings funksjonalitet. Vi har også hatt mye kontakt med en rekke kontaktpersoner for å forsikre om oss at vi velger riktige komponenter og tilleggskomponenter til huset, slik at det kan integreres i løsningen. Vi har også hatt mye kontakt med HDL, men ettersom vi klare på hva som skal gjøres fremover og gode tanker om hvordan vi skal få det til, er denne kontakten foreløpig redusert. De er veldig klare på at det er bare å spørre de om det er noe vi lurer på om HDL systemet. Vi har vanligvis hatt møter med veileder annenhver uke, men i de siste ukene har vi ikke hatt spesielle saker å ta opp ettersom det ikke har vært noe særlig jobbing med rapporten og det meste har vært utviklingsarbeid.

Statusrapport 3

1. Oppsummering av arbeid hittil

1.1 Status på utstyr

1.2 Status på prosjektet

2. Rapport

1 Oppsummering av arbeidet hittil

1.1 Status på utstyr

Vi fikk sendt siste reviderte utgave av utstyrlisten til Vanja 15. april. Denne utstyrlisten dekker det vi anser som kritisk for å oppnå ønsket funksjonalitet og dekker: HDL-komponenter, Z-Wave- og mBus-kompatible USB-lesere til henholdsvis luftfuktighetsmålere og vannmålere, Raspberry Pi m/ Ethernet-kabel, nettverksrouter og annet nødvendig utstyr. Etter møte med oppdragsgiver ble det sagt at noe utstyr skulle bestilles i løpet av dagen (25/04). Av utstyr i huset (LED-spotter osv) er ikke montert enda (03/05). Det begynner virkelig å haste nå, og vi er usikre på om vi rekker å implementere og kjøre reelle tester i boligen før 18/05.

1.2 Status på prosjektet

Når det gjelder funksjonalitet knyttet til selve applikasjonene anser vi nå denne som tilfredsstillende. Det gjenstår derimot litt opprydding i kode, samt å knytte funksjonalitet opp mot korrekt bruker.

2 Rapport

Prosjektets endelige rapport er på god på vei. Vi holder nå på med de avsluttende kapitlene, 5 og 6 som henholdsvis omhandler prosjektets gang med tanke på koding, testing og implementasjon, og avslutning.

I kapittel 5 ønsker vi å inkludere en god del kodeeksempler, slik at vi får forklart litt om hensikt og tankegang.

Prosjektskisse – Maistanga 2, 3770 Kragerø – grønt smart hus

Hensikten med prosjektet er å legge til rette for smart hus med spesiell vekt på effektiv energiutnytting av og optimalt inneklima i et siporexhus med to boenheter og en hybel/arbeidsrom.

Energieffektiviteten skal sikres ved optimal utnyttelse av lagret termisk energi og naturlig ventilasjon i kombinasjon med varmluftgjennvinning (LUNOS) og tilførsel av varme fra bergvarmepumpe til vannbåren varme (http://www.vaillant.no/privatkunder/produkter/geotherm-exklusiv-6-10-kw-769.no_no.html).

Det skal lages et styringssystem som overvåker varmeproduksjonen fra varmepumpen og strømforbruket til ulike komponenter i bygget (varmepumpen, varmtvannsbereeder, mekanisk ventilasjon, balansert mekanisk ventilasjon, komfyr, vaskemaskin, kjøleskap, fryser samt lys og øvrige elektriske komponenter).

Det skal spesifikt kunne måles hvor mye varmtvann fra varmepumpa som går til hver av de tre ulike boenhetene (to leiligheter og en hybel/arbeidsrom). Dette gjøres ved hjelp av vannmåler og sensor som er basert på Z-wave eller lignende signaloverføring.

I tillegg kan det være måling av utetemperatur, vind og vindretning samt nedbør. Data fra uteklimaet kan brukes til å tidlig justere styringen av varme og ventilasjon inne for å sikre utnyttelsen av tregheten i lagret termisk energi.

Det kan legges opp til måling av inneklima data – utover temperatur – som eventuelt luftfuktighet.

Generelt:

Smart-hus-anlegget skal kunne:

- Styre temperaturer i de ulike temperatursonene gjennom døgnet = styre varmepumpe og ventiler for tilførsel av varme til de ulike rommene
- Styre lys av og på – demping og døgninnstillinger (HDL Nordic sitt system)
- Tilpasse temperaturreguleringen inne til forandringer i utetemperatur (valgfri)
- Styre åpning/lukking av tre Velux takvinduer i hovedleiligheten (registrer om det regner, generell temperaturregulering, evn vindretning i forhold til hvilke vinduer som skal åpnes/lukkes) (<http://www.velux.no/produkter/takvinduer/velux-integra>)
- Overvåke forbruk av strøm til varmepumpe samt vannmengder og temperatur avgitt til de tre boenhetene
- Styre den balanserte ventilasjonen (LUNOS) som er desentralisert til de ulike rommene (hvert rom har sitt egen mini-anlegg).
- Eventuelt styre/regulere inneklimaet bedre – utover temperatur (luftfuktighet mm)

I Dialog med HDL

Samtale med HDL

dato: 22.01.2016

kl 1300-1315

Henrik,Jørn-Are, Edvard

Agenda:

- Varmestyring

Referat:

Varmestyringskontroller, 240V ventiler, rørlegger de på, legger kabel fra de til varmestyringskontrollerer. Rørleggereren skal kun innstallere ventiler.

Slår de sammen ned til 13 stykker, 2 varme styringskontrollerene. DLP-bryterer, infoskjerm, DLP-touch og DLP. Kan legge til disse bryterne for manuell styring.

Slår en kabel på en ventil. De er ikke alltid like nøye på hva slags ventiler som går til hvilken rom. Rørleggerer må være sikker på hva slags ventil går til hvert rom.

Vannmålere, Ring senere på dagen.

Agenda:

- Teknisk informasjon
 - Lunos
 - Velux
 - Vannmåler
 - mBus
 - Vaillant
- Koding
 - Utvidelser på HDL IntelliCenter-server.
 - Tilgang til IntelliCenter.
 - Iridium Mobile
 - Automatisk styring
 - Funksjonalitet på vinduene
 - Lese av vannmålingene og regne ut strømforbruk

Referat:

Lunos:

Litt tilsvarende som Vaillant. De regulerer seg selv. Tar inn varmen den trenger, også måler den varmen og styrer deretter. Stilles inn en gang, også står dette bare og går. Det som normalt styres er egentlig bare hastighet og hjemmemodus. På de aller fleste styres dette via relé. Relé utgang 1 er hjemme/borte ut 2 er vifte. Trenger ikke KNX der heller. Eneste som reguleres er ønsket temperatur. På de fleste er det en liten kontrollpanel som sitter på aggregat og de fleste lar dette stå på 23 hele året. Kobles til relé.

Kan ringe og høre med Lunos.

Hvordan kobler seg 5/uno må vite mer om de øvrige funksjoner. Hvordan det kan kobles opp. Koblingsskjema, hvordan man kan koble seg opp. Går an å koble på en enkel styring. Kan styre de fire viftehastigheten via relé. 2 HDL relé-kanaler.

Velux:

VELUX sensor interface kan konvertere signaler og "snakke" trådløst mot vår plattform -> io-homecontrol. Man må først, og alltid aktivere/tilslutte takvinduet VELUX INTEGRA til nettspenning og deretter pare dette opp imot vår kontroll-Pad, slik at vi er oppe å går i bunnen. Deretter tilsluttes sensor interface nettspenning slik at det eksterne signalet kan hentes til det digitale interfacet. Sensor interface er da denne:

KLF 100 EU sensor interface

<http://www.velux.no/~media/marketing/no/dokumenter/monteringsveiledninger/elektrisk/klf%20100%200451797-0109.pdf?la=nn-no>

Her gjelder digital styring:

Et signal for åpne, et signal for å stenge. Vinduet vil stoppe hvis begge signalene sendes samtidig.

Her må en annen enhet kobles til denne KLF 100 sensoren. Har funnet at enkelte har brukt ETH008 for å styre vinduet gjennom denne.

RS-fjernkontroll, kan ikke bruke til å simulere. Hopper over kontrolleren som følger med vinduet, og går rett på relét. Kable ned til sikringssskap (alle vinduene) slik at releene kan stå i sikringssskapet da er det ingen problem. Evt bruke det vi allerede har. HDL-relé. Så sant KLF 100.

Vannmåler:

Vi har kontaktet Rørlegger som skal installere disse om mer informasjon. Han mener de ikke kan kommunisere med andre enheter, men dette er da feil. Vi har sendt han mer informasjon og fått han til å undersøke mer rundt vannmålere.

Utenom dette vet vi ikke så mye om disse, annet enn at de kommuniserer via mBUS. Lagt ved informasjon rundt disse som vedlegg. Det er de med pulsutgang som er relevante.

Vaillant:

Største problemet har vi rundt varmpumpen. Vaillant har vært lite hjelpelige og det har ikke vært enkelt å få tak i informasjon rundt den gjeldene varmpumpen og om den kan kommunisere. De vi har vært i kontakt med mener det ikke er mulig å kommunisere med den via et eksternt system. Dog, de har begynt levering av nye varmpumper som skal kunne gjøre det. Vi har vært i dialog med oppdragsgiver som mener det skal være mulig. Vi mener at det er mest logisk at det skal finnes en måte, ettersom den kommuniserer med sin egen applikasjon. Allikevel, er det vanskelig å finne noe mer informasjon rundt dette. Vi har kontakt engelsk og tyske kontorer, men ikke noe svar enda. Kan komme mer tilbake om dette.

Ny varmpumpe kommuniserer trådløst over WiFi. Hvordan kan vi få koblet denne til?

KNX-systemet er et eget styringssystem likt HDL. To forskjellige busser. KNX-IP-modul som kom med HDL, KNX kobles til LUNOS og Vaillant. Det må settes opp som et separat system med eget programvare. KNX-programvaren koster 9000 kr å integrere med HDL. Kom via IntelliCenter-server evt via KNX-overgang. KNX-programmet MÅ være konfigurert på forhånd. Simulere samme melding via datapoint. Kan neppe gjøre det uten KNX-programvare. Mulig det finnes gratisvare. KNX er egentlig noe eget. Trenger egentlig bare å regulere ventilene, ikke selve varmpumpen siden den regulerer seg selv. Kunden skal ikke behøve å regulere. Parametere settes av rørlegger én gang. Vi tar oss av styring av varmtvann. Treng bare vanlige 230V-ventiler og HDL-anlegg. Trenger ikke KNX-styring. Varmekontroller fås fra HDL. 6 reléutganger som kobles til ventiler. Ventil med to ledninger (av/på). Styres av relér. Termostatfunksjonalitet. Trekker led fra 6 ventil 7 leder i praksis inn i sikringssskap og kobler til varmestyringssystemet. 1 varmestyringskontroller pr 6 ventiler.

Kan bare bruke den vi har. La den bare gå sin gang. HDL tar seg av regulering inn til hvert enkelt rom. Kunder har feil forestilling. De ønsker å styre alt, men det trengs ikke. Kjølenskapet styres f.eks seg selv. Setter det på 3 grader, så fikser det seg selv. Tell Vanja.

HDL IntelliCenter:

Vi har også noen spørsmål rundt dette med koding. Hvordan blir å legge til kode og utvidelser til deres HDL IntelliCenter server (<http://nettbutikk.hdlnordic.no/products/hdl-intellicenter-server>)? Følger det med utstyr for å gjøre dette når serveren kjøpes? Programmeringsspråk? Det vi kunne tenkte å legge til er jo da automatisk regulerings funksjonalitet, slik at huset selv reguleres for eksempel ved endring av utetemperatur. Vi vil også legge til mer funksjonalitet for styring av vinduene. Slik at de oppgis hva slags vinkelposisjon vinduet befinner seg i. Det

samme gjelder for ventilasjonssystemet og automatisk regulering av inneklimate. Utregning av strømforbruk til oppvarming av vann, slik at brukeren kan få informasjon om dette. Mer informasjon rundt dette hadde vært svært hjelpelig.

Lisenspris blir samme uavhengig. Hvis vi bruker serveren, så snakker alle enhetene på 1 side og device på 1. Utføres av server. En helt annen måte å lage logikk. Iridium er egentlig bare en visualisering. Server har intelligens, kan sette på smart funksjoner, hvis det er behov på kjøkkenet og er på ferie skal det og det skje. Mange smarte løsninger. Mye lettere å få til på server kontra Iridium. Renger med å ha ferdig mars, og da kan vi få en slik (HDL IntelliCenter). Vi kan fint bruke den. Serverens maskinvare er klar, det som står igjen er drivere. Er basert på serverløsningen som heter Zipato. Kan lese på server, kan gjøre alt som Zipato-server kan gjøre. Han innebygd GUI. drivere for hele HDL-systemet, rule creator er logikkfunksjoner i server. Applikasjon er ferdig, Server kommer med applikasjon for iOS og Android. Konfigurasjon via nettleter, styre via nettleter. Styres vanligvis gjennom applikasjon. Den vil gjenspeile det vi har lagt inn på server. Er ingen koding. det er snakk om konfigurasjon via rule creator. Server har overgang til Z-Wave og ZigBee. Gateway.Hva er det vi ønsker skal skje? Konfigurasjon via interface. Vi må gå gjennom hele systemet, antall dimmere, type bryter etc.

Hvordan de skal kobles opp, må høre med Vanja. Led striper. akkord med ledstriper i ulike farger. Varmesoner, lyssoner, husløsning. Passende bryter.

Prosjektselegere, ta kontakt med disse. Må være dialog, om lys, relé, sensorer, på ventilene.

Kan legge til funksjonalitet på serveren, vises på appen. Zipato.com, sjekk ut dette. kan sende en beta server.

Iridium Mobile:

Lurer på om det er mulig å legge til det meste av funksjonaliteten (utvidelsene) på iRidium mobile (mobilapplikasjonen) i stedet for å legge dette til på IntelliCenter serveren? Ved å bruke iRidium sitt script API.

Device lisens for hver iPad, alternativ site lisens koster 6-7000. Fordel er kjempefleksibelt. Vanlig å sette opp på brukervennlig måte. Kommer med en enkel serverløsning. Ferdig til høst. Ikke like smart som HDL-server.

Agenda:

- Varmepumpe
- Varmestyringskontroller
- HDL-brytere
- Kabling

Referat:

Varmepumpe og ventilene i rørene:

- Ventiler, normalt lukket eller åpnet. Rørlegger spør om dette. Her er vel normalt åpnet mest normalt.
 - Vi kan styre begge deler. De trenger strøm for å være åpne, og det mest normalt med at de er lukket. Anbefaler 230V.
- Gulvfølere til varmestyringskontrolleren.
- Digital temperatursensor som kobles til varmestyringskontroller. **Kan ikke bryte opp gulvet.**
- Er den avhengig av å få informasjon om temperatur i de ulike rommene.
- Hva slags sensorer? Kan ikke ha et DLP panel i alle rom.
- Temperaturfølere i hvert rom, enten trådløs eller kablet.

DLP-brytere:

I hvert rom har HDL-bryter. De har varmeføler. Hvert rom må ha en bryter. Har de ikke det kan man ikke skru på lys. Smartbryter i hvert rom. Trådbundet bryter er best. Kan bruke en enklere bryter. Rom med fliser er det vanlig å legge inn gulvføler som kobles til varmestyringskontroller, men da er det en egen bus. Annen løsning, bad egen DLP-bryter, bus bak bryter, blå pille/tings, analog gulvføler, går inn i denne blå dingsen. Inntil fire varmesensorer.

Blå dings = temperaturinndatamodul - kobler inntil 4 analoge temperaturfølere. På denne måte vet vi gulvtemperatur i rommet. Bruker ikke føler i gulvet i vanlige rom. Kan legge inn gulvfølere i rom der parketten ikke tåler høyere enn gitt temperatur, slik at temperaturen i rommet ikke overstiger denne.

Koblinger og kabling:

- Hvordan er det med kobling imellom server og alle reléene og styringene?
- Hvor mange innganger er det på serveren?
- Plassering av serveren: Er det god rekkevidde? I forhold til plassering.
- Trengs det en HDL BUS-kobler, Må det eventuelt være en switch.
- Skal dette være fysisk kablet? Alle skal være kablet via HDL bus kabel?

Alle bryterne må ha en bus-kabel. Server har ingen innganger. Kobles til busen. Kobler reléene til en buskobler. Alle komponenter har en bus kobles i en lang skjøte.

1 relé for ventilasjon i hver boenhet: 3

1 relé for vinduer i toppleiligheten.

3 Varmestyringskontrollere, HDL-MFH06

Eventuelt flere dimmere for belysning, uvisst. Haavard skulle reise ned å se.

Vannmåler, ny type:

- mBus med temperaturføler. Sende signal.
- Designet for føle varmt og kald vann.
- Energimåler. Må installeres på rørene fra varmtvannet.
- Hvor kan det være mest gunstig å plassere disse?
- Vi ser for oss tre stykker, en til hver boenhet.
- Hun vil også se etter energitap, vil se tap av varme. Vil ha god nøyaktighet på strømforbruket.

Komponentene:

- Komponentene må i sikringsskapet, og dette må være såpass stort at det har plass til alt.
- Det er egentlig ganske likegyldig hvor alt er plassert.
- Elektrikerjobb.
- Sette opp et tilleggsskap.

Vet ikke hvordan det kan gjøres. Energibus, skal kunne gjøre det.

HDL buskabel til sikringsskapet.

Agenda:

- Nytt system.
- Trådløs gateway
- Egen server
- Programmering

Referat:

Forklarte at vi ikke kan bruke IntelliCenter, siden da har vi rett og slett ingen oppgave. Vi er nødt å ha en utviklingsoppgave med en god arbeidsmengde gjennom hele perioden. Vi kan derimot bruke HDLs trådløse gateway i stedet for IntelliCenter-serverer. Dette medfører at vi selv programmerer alt. Tanken er å bruke en Raspberry Pi som server med lagringskapasitet, koble denne til gateway via Ethernet.

Drivernivå - må programvare kommunisere med rutiner som gjør lavnivå kom på Ethernet som igjen er koblet til IP-modul. Bruker DLL til å pakke inn informasjon i en melding som produktene forstår. Men på nivå over er det rutiner. Single channel lighting, den meldingen har noen parametre. Hva er dimmernivået, hva er runningtime (dimme opp og ned), og DLL sender denne meldingen ut på anlegget. DLL får tilbake en melding som vi må tolke. Det er da en typisk svarmelding, jeg har gjort jobben og satt alt - alt er ok. Det vi må forstå er hva slags meldinger skal sendes med parametre og sende det, og tolke det. På toppen av det må logikken lages og GUI. La oss si en knapp som sender gå til 50 prosent, må vi lage et GUI som sender det til DLL. Single channel lighting commando. Det er en jobb å sette seg inn i mulighetene med anlegget. Har dokumentasjon på meldinger (HDL).

Dokumentasjonen er diger. Hvilke kommandoer er det vi trenger å forholde oss til. Anbefaler å bli med på HDL-kurs. Det fungerer slik at det tas en gjennomgang av alle HDL-komponenter, også begynner vi å programmere.

Programmering:

- Driveren
- Må gjøre lavkommunikasjonen, kobles til på Ethernet,
- DLL, kan bruke. Pakke inn informasjon i en melding.
- Nivået over, hva er innholdet i meldingen. Signal channel lightning, den har parametre, hvilke enheter, hva er dimmernivået, hva er running time, alle disse parameterne. Må få tilbake meldingen, den må tolkes. Ja, jeg har mottatt, alt er ok. Hva slags meldinger skal sendes, er det mottatt? Logikken og brukergrensesnitt må lages.
- En haug av kommandoer, hva mulighetene, sequence kontroll,
- Er dokumentert, haug med sider, må vite hvem som passer best,
- Hvilke kommandoer trenger vi, bør være et fåtall av kommandoer,

Kurs:

- Være med på kurs førstkommande mandag og tirsdag. HDL-programmering..
- Gjennomgang på kontoret, så begynner vi å programmere.

J Prosjektering og utstyrliste

Prosjektering av oppgaven

I dette prosjektet har en av våre oppgaver vært å prosjektere hva slags utstyr som skulle installeres i boligen, samt antallet komponenter. Dette har vært en viktig del, ettersom hvordan SHK-applikasjonen og SHK-serveren skulle bli utviklet har vært avhengig av dette. I prosjektskissen var det oppført en rekke ønsker og krav i forhold hva som skulle styres og hva som skulle måles. Det har vært vår oppgave å finne produkter som tilfredstilte disse kravene. For å få til dette har vi vært avhengig av å forstå ulike teknologier, som trådløse kommunikasjonsprotokoller, men også hvordan komponentene fungerer på maskinvarnivå. Dette inkluderer Z-Wave, ZigBee og kommunikasjonsprotokollene m-BUS, og HDL BusPro, effekt og spenning til de ulike komponentene og eventuelt hvilken frekvens de bruker for å sende data.

Deler av utstyret som skulle bli installert i huset var bestemt på forhånd. Dette gjelder ventilasjonsanlegget, takvindue og enkelte HDL-komponenter. Selv om dette utstyret skulle installeres var det ikke klart hvordan disse skulle kommunisere sammen. Det var derfor vår oppgave å finne tilleggskomponenter til disse for å forsikre oss om at de kunne integreres i SHK. Ettersom HDL leverer de resterende komponentene som skulle installeres, var det nødvendig å utvikle SHK med utgangspunkt i HDL-komponentene.

Ettersom vi skulle ha en aktiv rolle i bestillingen av utstyret, har det vært viktig for oss å finne produkter som tilfredstilte oppdragsgivers krav. Dette innebar en prosess med å studere hvilke produkter som fantes, teknologien, lese anmeldelser, få oversikt over den tekniske informasjon om produktene.

Under bestillingsprosessen av utstyret har vi vært klare på at vi ikke kan ta ansvar for at det som bestilles ikke fungerer som forventet. Dette gjelder spesielt på grunn de store kostnadene dette prosjektet har og at vi ikke har erfaring med produktene.

Server

Serverprogramvaren vår skulle kjøres på en sentralt plassert maskin i huset. Her stod vi fritt til å velge hva slags type maskin som skulle kjøre serveren. Vi ble enige om at dette skulle være en Raspberry Pi. Her var det nødvendig for oss å finne en variant som hadde alle nødvendige porter og innganger. Raspberryen måtte ha minst én USB-inngang for data-avleseren til Z-Wave-nettverket. Den måtte ha en Ethernet-port, i tillegg til trådløs nettverkstilkobling. Den måtte også ha mulighet til å kjøre et operativsystem som støttet vår Java-baserte server. Her var det den trådløse nettverkstilkoblingen som var avgjørende for at hva slags modell som ble valgt. Vi bestilte en Raspberry Pi 3 model B med starter kit, som inkluderte strømkabel, minnekort og et deksel til selve Raspberryen.

For å få integrert servermaskinvaren med HDL har det også vært nødvendig å bestille en HDL gateway som fungerer som bindeleddet mellom serveren og HDL BusPro-nettverket. Her har det da blitt bestilt to trådløse gatewayer fra HDL. Vi ble anbefalt å bruke to gatewayer, for å forhindre at det ble for mye trafikk på en gateway, som kunne forårsake forsinkelser eller tap av datapakker.

Lunos

Etter en kontinuerlig dialog med en ingeniør i Lavenergisystemer AS, hvor det var viktig å finne riktig ventilasjonssystem med tanke på husets planløsning, ble det enighet om at det kun skulle bestilles Lunos av typen E² Standard. For å få integrert ventilasjonssystemet Lunos i

HDL, fant vi ut at vi var avhengige av å ha en sentralstyringsenhet. Dette er et nytt produkt og gjør det mulig for Lunos å kommunisere med eksterne produkter. For å forsyne sentralstyringen med strøm, måtte vi også bestille en 5/NT18. For å få dette integrert i HDL systemet, forholdt vi oss til vår kontaktperson i HDL om hvordan dette kunne gjøres. Her ble vi enige om at det skulle brukes en av deres relemoduler.

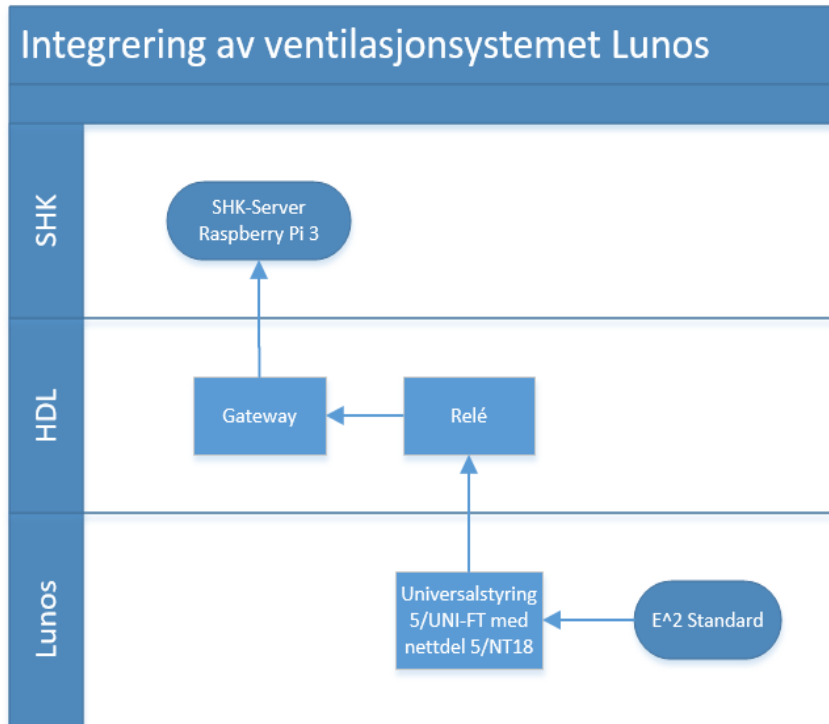


Fig. Viser komponenter som bygger opp integreringen av Ventilasjonssystemet Lunos

Velux

Integrering av Velux Integra takvinduer måtte vi også inkludere. Dette viste seg å være en mer komplisert integrering enn ventilasjonssystemet, ettersom det er et helt proprietært system og kommunikasjonen internt i systemet er kryptert. Den norske leverandør hadde lite erfaring med integrering av vinduene i eksterne systemer, og mente det ville bli vanskelig å få til en integrering utenom det proprietære systemet. Vi oppdaget derimot at de hadde en løsning for akkurat dette. Vi bestilte derfor en KLF 100 Interface sensor. Denne leveres også av Velux, og er produsert for at Velux-systemet skal kunne kommunisere med eksterne systemer. For å få de integrert i HDL-systemet, ble vi enige om at det skulle brukes deres gardinkontrollere, som igjen kobles på KLF 100.

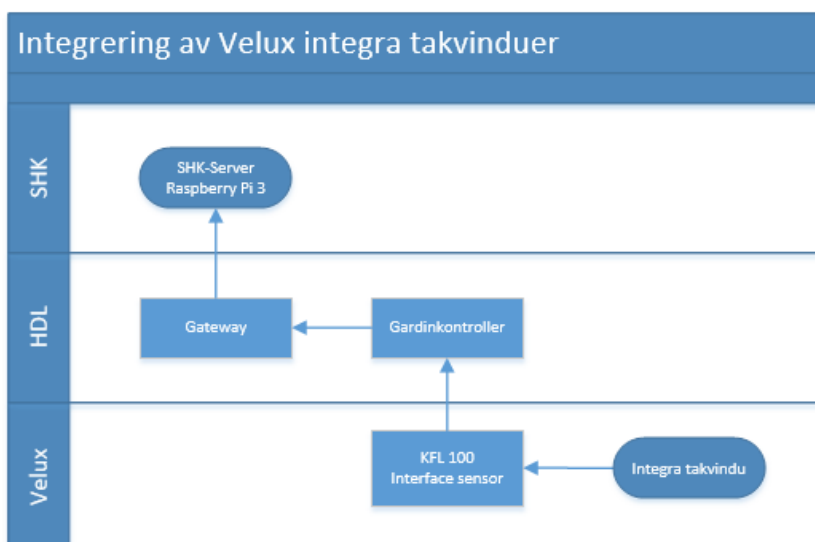


Fig. Viser hva slags komponenter som bygger opp integreringen av Velux takvinduer.

Vaillant

Når det gjelder integreringen av varmepumpen i systemet, var det misforståelser om hvordan dette skulle gjøres. Vi var overbevist om at det var varmepumpen som skulle styres, selv med en lang dialog med leverandøren Vaillant. De var lite hjelpelige i forståelsen rundt dette. Tilslutt fant vi ut at det var ventilene fra varmepumpen som skulle styres. Da forholdt vi oss til HDL og hadde dialog om hvordan de vanligvis håndterte integrering av varmepumpe. Da ble vi enige om at HDLs egne varmestyringskontrollere skulle brukes, og disse skulle kobles direkte på ventilene.

Lysstyring og paneler

For lysstyringen og manuell styring via paneler, krevdes det ulike dimmere og paneler, tilpasset rommet de skulle plasseres i. Utplukkingen av disse komponentene er hovedsakelig utført av HDL. Dimmere monteres sammen med panelene, derfor lot vi en med mer innsikt i HDL sin produktkatalog skulle ta seg av dette.

Z-Wave, sensorer og målere

Smarthuset skulle også ha sensorer og målere. Dette innebar at vi måtte velge ut komponenter som gjorde dette. Etersom HDL ikke leverte slike komponenter, måtte vi se på andre alternativer. Her måtte vi finne komponenter som kunne brukes, hva slags protokoll de kommuniserte over og hvordan disse eventuelt kunne integreres i vårt system. Her var det klart at Z-Wave var et godt alternativ, ettersom det er en åpen standard og har et rikt og svært hjelpelig samfunn. Det finnes også mange ulike og åpne hus-automatiseringssystemer for Z-Wave, og ved å bruke et slikt system i tillegg til vårt eget, kunne vi istedenfor å lage et helt eget system for Z-Wave, hente dataen vi trengte ut ifra disse.

Som en Z-Wave gateway bestemte vi at det skulle bestilles en Aeon Labs Z-Stick Gen 5. Dette er en USB-Gateway som setter opp Z-Wave nettverket og fungerer som mottaker for alle nodene i nettverket.

Oppdragsgiver ville ha fuktighetsmåling. For å dekke dette behovet anbefalte vi Aeon Labs MultiSensor 6. Dette er en sensor med fem andre funksjonaliteter i tillegg til fuktighetsmåling, disse kan videre implementeres i systemet. Produktet hadde gode anbefalinger og prisen var i ikke mye ulik priser på sensorer med kun fuktighetsmåling.

Det har vært komplikasjoner med å finne riktige målere. Oppdragsgiver ha måling av strøm til komfyr, begge varmtvannstankene i smarthuset og en for varmepumpen. Det ble først bestemt at HDL sine egne stikk-kontakt reléer skulle benyttes, men ettersom disse ikke har blitt testet på måling av kabler som fører stor strømstyrke, måtte det finnes et alternativ. Etter videre søk av produkter som kunne brukes, ble det besluttet at det skulle bestilles Z-Wave Clamp Power Meter, fra Aeon Labs. Dette er et produkt som måler magnetismen i kablene den festes på og måler med god nøyaktighet.

Målere og sensorer for vannmåling har visst seg å være det vanskeligste området for oss å finne tilfredsstillende produkter som dekker oppdragsgivers krav. Her har vi også vært avhengig å forholde oss til rørleggeren som skal installere produktene i huset og prosessen har vært preget av mange forsinkelser. Det har også vært misforståelser i forhold til hva oppdragsgiver egentlig ville med målingen, hvor energimåling, vannmåling og vannmålere har blitt brukt for å beskrive samme oppgave, men oppnår vidt forskjellige resultater.

Etter dialog med HDL kom vi frem til at deres Energimodul som er under utvikling kunne brukes og leveres i løpet av sommeren. På grunn av hvordan utvikling av prosjektet har utartet seg bestemte vi oss for å finne en annen løsning. For å finne en alternativ løsning har vi vært igjennom mange ulike produkter og teknologier, men mest spesielt M-Bus og Z-Wave produkter. Det ble først bestemt at et M-Bus produkt skulle brukes, men etter å ha erfart at det kunne bli vanskelig og ikke minst dyrt å sette opp, har det blitt sett på alternativer. Ved sluttfasen av prosjektarbeidet, har det blitt besluttet at det skulle bestilles Z-Wave Secure SES302 Temperature & Humidity Sensor Gen5, sammen med Z-Wave Secure Pipe Tank Digital Temperature Sensor for måling av temperatur på vannet som går igjennom rørene som skal måles. For måling av hvor mye vann som går igjennom rørene, har det blitt besluttet at det skal bestilles Sensus Residia vannmålere. Som en tilleggsmodul til vannmålerne, for å sende dataen videre til Z-Wave Gateway, har det blitt bestilt Z-Wave Secure Water Meter.

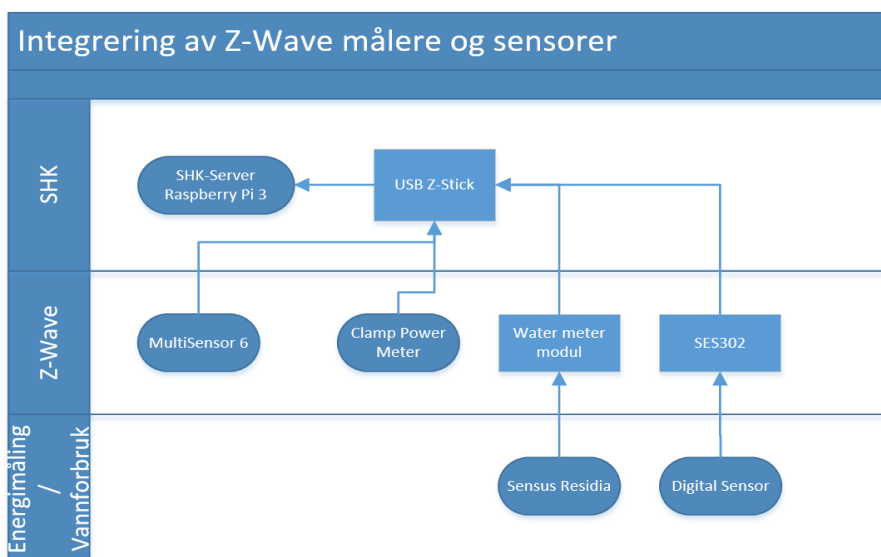


Fig. Viser hva slags komponenter som bygger opp integrering av Z-Wave målere og sensorer.

Ustyrliste

SHK-server:

1 stk Raspberry Pi model B starter kit

1 stk 1 meter Ethernet-kabel

HDL:

2 stk HDL Trådløs Gateway

3 stk HDL Varmestyringskontroller

2 stk HDL Gardinkontrollere

20 stk HDL Tre-kanals dimmere

8 stk HDL En-kanals dimmere

Annet:

4 stk Lunos Universalstyringer m/8 innganger

12 stk Lunos e² Standard

4 stk Lunos Nettdel 5/NT8

2 stk KLF 100 Interface-sensor

13 stk Aeotech Z-Wave-sensor (Fuktighetssensor)

10 stk Z-Wave Aeon Labs Clamp Power Meter(Strømmålere)

1 stk USB Aeon Labs Aeotech Z-Wave Z-stick

3 stk Sensus Residia (jet-C) (Vannmålere)

3 stk Z-Weave Secure water meter

K Møtereferat fra møter med oppdragsgiver

Møte m/oppd.giver 1 dato: 05/01-16

kl 0900-1000

Henrik, Jørn-Are, Edvard

Agenda:

- Signere prosjektavtalen
- Planlegge tur til Kragerø

Referat:

Tur til Kragerø:

- Blir tur ned 25 januar kl 0900.

Agenda:

- Prat med Morten Andersen:
- Sier vi må vite mer om kommunikasjon mellom systemene før han kan hjelpe oss mer.
- Rørlegger
- Velux-vinduene
- Vaillant varmepumpe
- HDL

Referat:

Prat med rørlegger (Varmepumpe/vannmåler):

- Han sier Vaillant-pumpa skal ha en egen nettbasert styringsapp. Men han har prøvd å ta kontakt med de angående denne, men har ikke hørt fra de.
- En varmtvannstank hver leilighet - er dette riktig ?
- Kan prøve å finne en annen type vannmåler med en åpen deling. disse går ikke an å kommunisere med.
- Rørlegger påstår at det er umulig å lese av strømforbruk til hver boenhet.
- Purrer på rørlegger, men han er vanskelig å få tak i.

Velux Integra:

- Om strømmen/sikringer går vil ikke VELUX INTEGRA kunne driftes selv om regnsensor blir våt ved regnvær, om ikke det er installert vår batteri back-up KLB 100 EU.
 - Kort fortalt vil ikke regnsensoren fungere uten strøm og vinduene vil ikke lukke seg automatisk. Kan installere batteri-backup.

Vaillant:

- Teknisk support lite behjelpelig.
- Forklarer at det ikke er mulig å kommunisere med E-Bus og WIFI.
- Det som vi har fått beskjed om er at det ikke er åpen bus slik at vi ikke kan få signaler ut av dette, foreløpig. Dvs at dere ikke får kommet inn med noe annet foreløpig.
- Kan høre med fabrikk i Tyskland.
- OpenHAB - er dette en mulig (tungvindt) løsning?

Openhab:

Skal støtte Vaillant, men er ikke 100 % dekkende. Bør fungere å kommunisere med Velux via HDL. De kommuniserer på samme måte, men er avhengig av å vite hvordan Velux applikasjonen «prater med» vinduene.

Vaillant: Det skal være mulig å kommunisere med. Må sjekke mer rundt dette.

Vannmåler - regne ut strømforbruket igjennom se hvor mye vann som går ut og temperaturen på vannet. Vannmålerne skal kunne kommuniserer, rørlegger har manglende kompetanse.

Lunos: De bør kobles sammen på en smart måte. Muligheter for å effektivisere luftingen mellom flere Lunos komponenter. Hvor vanskelig er det å lage en fase med 5 enheter?

Koordinere disse sammen. Mulig de bare kobler i par. Lage et optimalt system. For å få god gjennomstrømming. Høre om de har god løsninger for dette. Hva som finnes. Gjøre slik at de handler i koordinasjon med vinden?

Luftfuktighetssensorer.

- Finne annen teknologi som kan kommunisere med systemene vi lager.
- Legge opp til at det kan videreutvikles.
- Legge til rette for et godt inn klimaet.

Ringe teknisk hovedkontor til Velux, Lunos og Vaillant. Se etter blogger, høre med Openhub om de kan hjelpe.

HDL:

HDL mener at vi ikke trenger koden, men vi må finne ut hvordan komponentene kommuniserer, for eksempel Z-Wave kan brukes til vinduene og HDL. Lunos ventilasjon trengs ingen slik styring, siden det bare er av og på.

Kan skaffe serveren fysisk, hvis det er den vi trenger. også eventuelt koble sammen med komponenter. Ganske sikker vi trenger den.

Er det mulig å gå utenom de systemene og allikevel komme frem til mål? Det vil si plassere ut målere/sensorer for å få tak i informasjonen fra varmpumpen.

Agenda:

- Snakke om informasjonen fra møtet med HDL19.02.16:
- Se mail fra Haavard.
- Server
- Komponentene
- HDL-server

Referat:

HDL IntelliCenter:

- Avhengig av server. Kan få betaversjon i slutten av mars. Kan jobbe med Zipato-server, som er den serveren HDL har tatt utgangspunkt i, og sette seg inn i den foreløpig.

Systemet i huset:

- Vi må gå gjennom hele systemet. Antall dimmere, type bryter etc.
- Det får den informasjonen av Vanja.
- Hvordan de skal kobles opp, må høre med Vanja. LED-striper. akkord med LED-striper i ulike farger. Varmesoner, lyssoner, plantegning. Passende bryter.
- Vanja skal snakke om det.
- Prosjektselgere, ta kontakt med disse. Må være dialog, om lys, relé, sensorer, på ventilene.

Varmepumpe:

- Styre ventilene. Vet ikke hvem som leverer disse. Styres via varmestyringskontroller. Hvor mange ventiler trengs?
- En slik tar seg av 6 soner.
- <http://nettbutikk.hdlnordic.no/products/relé-varmestyringskontroller>
- 6 soner opp, 1 til badet, 3 til hovedrommene,
- 5 soner i leiligheten nede.
- 2 soner i arbeidsrommet

Vannmålere:

- Energimodul, kan installeres på senere. Kommuniserer via mBus.
- Morten hadde ikke mye erfaring med styring av Lunos og Velux.
- Hvor skal disse plasseres for optimal måling?
- Vannmåler, en til hver boenhet. Hva er den beste løsningen?
- Hun er opptatt av tap. Vil helst ha en god nøyaktighet på strømforbruket.
- Er ikke sikre på hvordan det med rørløsningen er valgt. Forbruk av vann til hver av boenhetene.
- Vannmålere, se på disse. Plassere temperaturfølere.

Velux:

- Oppkobling, hvordan har dette blitt gjort foreløpig Skal la seg gjøre via relé, som igjen er koblet opp mot KLF 100 interface sensor. Trenger to stykker. Vinduene kables ned mot sikringsskap.

Lunos:

- HDL relé kobles til 5/uni, som kan styre hastighet på viftene.

Inneklima:

- Optimalt inneklima: Hva er et optimalt inneklima?
- Finne ut mer om akkurat dette. Temperatur, luftfuktighet, CO₂.
- Høy luftfuktighet -> mer ventilasjon

Fuktighetsmålere:

- 13 luftfuktighetssensorer

SHK-app:

- Teste appen på forskjellige folk
- Innstillinger for modus: vil kunne endre selv
- År tidsmodus, ventilasjon og varme

Agenda:

- Design av SHK-applikasjon
- Ekstra funksjoner
- Logikk
- Utstysrliste
- Konfigurering av HDL-enheter
- Fuktighetssensor

Referat:

SHK-applikasjon:

- Fokus på brukervennlighet
- Lite fokus på utseende

Logikk:

- Kunne sette huset i moduser
 - Ferie
 - Dag
 - Borte
 - Natte
 - Mulighet til å endre disse
- Optimalt inneklime
 - Ventilasjon
 - Temperatur
 - Luftfuktighet
 - 13 luftfuktighetssoner
- Test av SHK-applikasjon på forskjellige folk

Utsstysrliste:

- Må begynne å tenke på å bestille nå snart
- Vi har sendt foreløpig liste

Fuktighetssensor:

- Ser etter gode alternativer.

Agenda:

- Utstysrliste
 - Vannmålere
- Test av applikasjon
- Refundering av reisekostnader
- Tur til Kragerø
- Produktkø

Referat:

Utsstysrlisten:

- Oppdragsgiver har enda ikke fått bestilt komponentene!
- Skal få bestilt i løpet av dagen.
- Vannmålere er vanskelig å få gjort noe med.
- CO₂-målere blir ikke noe av.
- Bestilt strømmålere.

Test av SHK-applikasjon:

- Ser for det meste bra ut, noen uklare ikoner.
- Ønsker å vite hvilket modus som er aktivt (ønsker at det står på hver side)
- Snakket om hardkoding av adresser til enhetene. Dette blir veldig vanskelig å få gjort noe med.

Refundering:

- Skal vurdere om det blir refundering. Hittil har vi brukt vel over 1000 kr pr/pers på reise til og fra Kragerø og til og fra Oslo på kurs. Vi har fått 200 kr pr/pers...

Tur til Kragerø:

- Tur ned til Kragerø for å implementere SHK. Må få avklart finansiering først og fremst.

Produktkø:

- Vi legger ikke til mer i produktkøen nå.

Annet:

Det står i oppgavebeskrivelsen at vi skal måle strømforbruk til komfyr. Det blir det ikke tid til. Måling av utetemperatur, vindhastighet og vindretning. Vi har ikke tid til å legge til flere ønsker nå. Vi prioriterer rapport fra nå av.

Agenda:

- Brukertest
- Om applikasjonen generelt
- Utstyr
 - Strømmålere
 - To målere til varmtvannstankene
 - Én til varmepumpe
 - Tre til komfyrene
 - Tre ekstra til diverse

Referat:

Applikasjonen:

- Innstillingsknappene kan være mer informative (på lyssiden vil hun ha et tannhjul med en lyspære i midten).
- Bedre måter å orientere seg i applikasjonen på (tilbakeknapper).
- Gjøre flere brukertester.

L Møtereferat fra møter med prosessveileder

Møte m/veileder 1

dato: 22/01-16

kl 1300-1315

Henrik, Jørn-Are, Edvard

Agenda:

- Hvordan skal åpne systemer som HDL kommunisere med et lukket system som Velux?
- Hørt om Real Time Transport protocol?
- Fremtidige møter? Onsdag/fredag passer dårlig. Annenhver uke?
- Frode vært med i slike oppgaver før?
- Angrepsmetode, vi vil gjerne ha alt til å fungere i lag (lukket og åpent)

Spørsmål rundt prosjektplanen:

- hva synes han om innholdet?
- virker oppgaveformuleringen grei?
- eksempel på flere projektrisikoeer?

Referat:

- Frode er prosessveileder
- Planlagt å ha møte hver uke

Forbedringspunkter:

- Formulering av oppgaven.
- Endre på avgrensninger og rammer.
- Effektmål skal være med.
- Bør fokusere på å se lenger frem i tid. På denne måten kan arbeidet vi gjør er føres rett over til sluttrapporten.
- Passe på å bruke riktig betegnelse på programvare og kodespråk.
- Må endre på grupperegler. Mer detaljert.
- GANTT- diagrammet må forbedres.

Agenda:

- Høre med Frode om hva han foretrekker å ha i starten av rapporten.
- Høre om forbedringspunkter til Gant-diagrammet.
- Hva mener han vi bør ha med i kravspesifikasjonen?

Referat:

- Bakgrunn:
- Bakgrunn for oppgaven. Ikke hvor vi kommer fra.

Omfang:

- Problemområde
- Styringssystemer brukes til mye mer, men er ikke et eksisterende som passer for dette huset.
- Datamaskiner er med på å styre mer og mer.
- Omfanget skal selge oppgaven. Må få leserens interesse.
- Starte bredt, og så spisse ned i detaljer.

Gantt, kravspesifikasjon og Scrum:

- Delinnlevering må skiftes til statusrapport.
- Etter hver sprint, kan vi ta et møte med oppdragsgiver.
- Kutte ned på delinnleveringer.
- Testing etter hver sprint.
- januar
- 4-8 sider.
- Use case.
- Produktkø, hva skal gjøres.
- Detaljere store produktkøer. Spesifisere.
- Generelle krav som inngår i systemet.
- Må være strenge oss selv og kontinuerlig følge med. Må holde koken oppe. Kan lett bli slapt.
- Hvordan trodde vi det ble, hvordan ble det, hva har vi lært? Konkludere.

Agenda:

- Kravspesifikasjon

Referat:

Vise kravspesifikasjon. Har brukt mye av siste tiden til å opparbeide informasjon for å i det hele tatt kunne startet med kravspesifikasjonen. Kan sende den på mail, etterhvert om flere biter kommer på plass.

Spørre hva han mener om hvordan vi kan plassere inn litteratur. Dvs. kompetanse vi må sette oss inn i som ikke har inngått i studieløpet. Elektro, tekniske informasjon. Analysearbeid i starten av kravspesifikasjonen? `

Mener vi kan legge til klassediagrammet senere.

Kan ha med forarbeid/analyse i starten av kravspesifikasjonen. Kommer helt an på hvordan det formuleres.

Agenda:

- Tilbakemelding på kravspesifikasjon
- Oppdatere om hva vi gjør og planen fremover.

Referat:

- Essensielle use case. Skriv om. For mange use cases. Dårlig flyt og vanskelig å fatte poenget.
- Lettkjørt, batterisluk, hva er det egentlig? Vi må ordlegge oss på en ikke-muntlig måte.
- Være mer spesifikke.
- Legge til et kapittel hvor det forklares hva er smarthus teknologi. Hva kjennetegner det i dag? Prisen i dag. Hva vil framtiden vise. Ukjent for mange.

Agenda:

Telefonsamtale med Morten 19.02

- Teknisk informasjon: Lunos, Velux, vannmåler, varmepumpe
- Er i gang med bytta av varmepumpe. Ny varmepumpe kommuniserer via KNX. Hvordan kan vi få koblet denne til? Samme modul som Lunos. KNX HVAC-modul
- KNX. Hvordan motta og sende mer avanserte innstillinger som varmepumpen utfører. Varme i ulike rom osv.
- Koding
 - Utvidelser på HDL IntelliCenter server. Tilgang på serveren
 - iRidium mobile
 - Automatisk styring
 - Funksjonalitet på vinduene
 - Lese av vannmålingen og regne ut strømforbruket
- Vannmålere og mBUS
 - Kjenner du til intelligente vannmålere som du kan anbefale. Som kan kommunisere digitalt.
- Belysning og media
 - Hvordan koble dette opp og koble til Iridium mobile.
- Tilbakemelding på det som er skrevet tidligere.

Referat:

Lisenspris.

Agenda:

Snakke om punktene som ble sendt på fredag.

- Misforståelser
- Oppgaven er blitt en helt annen!

Mye av kravspesifikasjonen vi har laget er irrelevant, dette gjelder også planleggingen i forprosjektet (f.eks Gantt-skjemaet). Prosjektet har gått fra å være et utviklings- og prosjekterings prosjekt/oppgave, til å være et prosjekt som gjelder å kartlegge, planlegge, prosjektere og konfigurere.

1. Kartlegging, vil si å få oversikt over prosjektet og finne ut hva som trengs av komponenter, tilleggskomponenter og hvordan disse kan kommunisere med styringssystemet.
 - Planlegging vil si å finne den beste løsningen å få dette til i praksis.
 - Prosjekteringsdelen vil bestå av hvordan vi skal gå frem for å få dette til. Hva må vi gjøre? Hvem trenger vi å kontakte/forholde oss til?
 - Konfigurering vil være å lage logikken på serveren, gjennom HDL sitt eget grensesnitt. Denne konfigurering vil foregå på serveren og vi bruker deres løsning for å enkelt lage smart logikk. Serveren får vi ikke før i starten av April.

For vår del, ser det ut til at det blir ingen eller svært lite koding/programmering. Hvis man da ikke vil kalle konfigurering for programmering. Hvordan kan vi tilrettelegge oss dette i forhold til rapporten. Hvordan kan vi gå frem her?

Utviklingsmodellen vi hadde valgt (Scrum) er dermed vanskelig å gjennomføre, ettersom arbeidet er avhengig av mange parter, slik at det er mange tannhjul som påvirker hvordan vi kan arbeide videre. Det er ikke slik at vi like enkelt kan dele opp oppgavene i tasks og sitte en hel dag med programmering.

Hvordan skal vi formulere oss i forhold til misforståelser og oppretting av disse?

Dette gjelder oppgaven generelt (hvordan oppgaven/prosjektet har endt opp), men også for eksempel varmpumpen. Her var vi sikre på at det var varmpumpen som skulle styres. Vi gikk til verks med å bytte denne, siden den ikke kunne styres via et eksternt system. Senere fant vi ut av det var ventilene som skulle styres, og byttet var derfor ikke gjeldene lenger. Når nå kommet frem til en løsning for hvordan dette skal gjøres.

Vi er veldig usikre på oppgaven når det kommer til karakter for bacheloroppgaven. Selv om dette er mye arbeid, føles det som om dette er på kanten av hva en dataingeniør kan få godkjent som oppgave.

Referat:

Triste nyheter, men må nok bli slik.

Må begynne å se på andre oppgaver og starte på nytt. Veilederen er enig i at oppgaven er litt tynn og ikke egnet for dataingeniører. Oppgaven har blitt til noe helt annet enn vi hadde tenkt. Den har gått helt bort fra å være et utviklingsprosjekt.

Agenda:

Har fått nytt forslag på hvordan vi kan løse oppgaven.

Forhøre oss rundt en egen serverløsning med bruk av lokal PC og HDL IP-modul.

Lage logikk til denne.

Referat:

Fått mer informasjon rundt serverløsning. Skal sende filer og operations codes som inneholder alle kommandoene de vanligvis sender. Invitasjon til kurs neste uke.

Agenda:

- Design av mobilapplikasjon, navn på diverse innstillinger
- Funksjoner hun vil ha med
- Hva slags logikk hun kan tenke seg
- Gå igjennom enheter som trengs og som må bli testet
- Konfigurerer av HDL-enheter

Referat:

App:

- Fokus på brukervennlighet og oversiktighet
- Er ikke så opptatt av at det skal se fint ut

Logikk:

- Kunne sette huset inn i ulike modus
- Kveldsbelysning, ulike dimmenivå

Unviersalstyringen, hvordan skal denne kunne kobles opp mot et relé? hva slags relé trengs for denne.

Database for innstillinger?

Agenda:

- Oppdatering
- Jobber med server og mobil-applikasjon.
- Forhøre oss rundt ting som har blitt endret. Hvordan skal vi forholde oss til dette?
- Kravspesifikasjon osv.

Referat:

Fortsett med jobbingen...

Agenda:

- Oppdatering (lite arbeid forrige uke på grunn av sykdom og eksamen)
- Rapportskrivning, utviklingsarbeid og bestillingsliste.
- Arbeidsfordeling fremover
 - Fordeling av arbeid. Edvard kommer til å begynne med rapportskrivning, mens Henrik og Jørn-Are fortsetter med utviklingsbiten. Vi kommer også til å jobbe med bestilling av komponenter, og forhåpentligvis bli ferdig med denne listen denne uken.
- Hva som gjenstår
 - Installasjon av utstyr i smarthuset
 - Tilrettelegge SHK-applikasjonen for smarthuset i Kragerø. Hver enhet i må ha sitt eget device number og subnet number. Dette settes manuelt.
 - Fuktighetsmåling. Installasjon av disse. Z-Wave mottaker og kode for å motta dette.
 - Bestilling av Raspberry Pi-en og konfigurering av den.
 - Bestilling av fuktighetsmåler, brukergrensesnitt til Raspberry Pi som gjør den til en mottaker for Z-Wave

Referat:

- Innholdsfortegnelse, hvordan det strukturert
- Innledningskapittel; hvordan oppgaven ble, hva endret seg,

Agenda:

- Tilbakemelding på nåværende rapport
 - Innholdsfortegnelse
 - Manglende punkter?

Referat:

- Tydeligere ordbruk
- Skrive litt om forvirringen i innledningsvis i rapporten, og referere til denne i kapittel 6.

Agenda:

- Tilbakemelding på kravendringen i rapporten
- Oppdragsgiver kommer med nye ønsker nå

Referat:

- Kan skrive kortere og egentlig bare henvise til kapittel 1.
- Vi må bare si at det er for sent å komme med flere ønsker nå.
- Siste tiden burde gå på finpuss av applikasjonen og skrive rapport.

Agenda:

- Tilbakemelding på rapport
- Endelig status på SHK

Referat:

Tilbakemelding på rapport:

- Kapittel 6 er for muntlig.
- Terminologi flyttes til vedlegg.
- Reorganiser vedlegg.
- Vedlegg er ikke eget kapittel.
- Figur- og kodeliste på egen side.
- Dropp databasebeskrivelse i kapittel 5.

Agenda:

- Tilbakemelding på rapport
 - Dersom det er mangler, hva mener han vi burde ha med i rapporten?
- Kodeeksempler
 - Klassediagram
 - Systemsekvensdiagram
- Fremføring på engelsk
- Lynkurs om presentasjon

Referat:

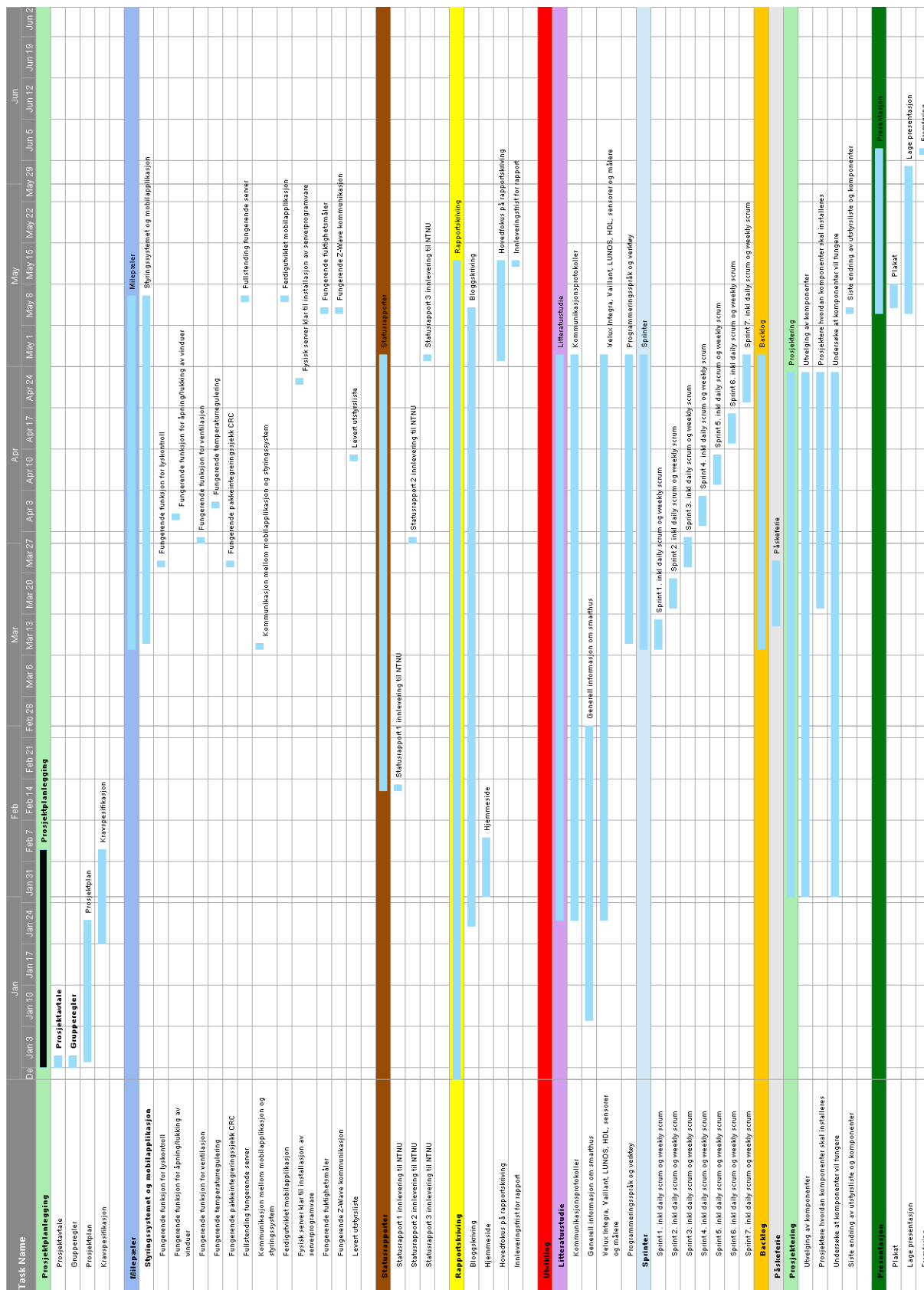
Tilbakemelding på rapport

Kodeeksempler

Fremføring på engelsk

Lynkurs om presentasjon

M Oppdatert fremdriftsplan



N Ukentlig logg

Uke 1

Lest gjennom informasjonen rundt bacheloroppgaven og har begynt å utforme grupperegler. Vi har også vært i kontakt med oppdragsgiver, Vanja Ohna. Avtalt tur til Kragerø. Jobbet med prosjektplanen.

Uke 2

Arbeid med prosjektplanen. Vært i kontakt med oppdragsgiver angående en mer detaljert oppgavebeskrivelse.

Uke 3

Kontaktet leverandører av utstyr til huset. Prøver å finne ut av hvordan en tredjepartsløsning skal kunne kommunisere med de forskjellige enhetene. Jobber forsåvidt med prosjektplanen enda.

Uke 4

Prosjektplan levert. Jobber fortsatt med å finne ut av kommunikasjonsbiten. Ser på tekniske verktøy vi kan bruke. Arbeider med oppgavebeskrivelsen. Klargjør hvilke oppgaver som skal gjøres, hva vi ikke har mulighet til å endre på (lukkede systemer), hva vi må ta hensyn til og hva som allerede finnes og kan brukes. Vi er i tillegg på prosjektbefaring i Kragerø. Startet på kravspesifikasjon.

Uke 5

Vi har en del utfordringer knyttet til lukkede systemer. Varmepumpen kan ikke kommuniseres med annet enn med Vaillants eget system. Forsøker å finne metoder å komme oss utenom Vaillants proprietære system. Ellers drøfter vi med de forskjellige leverandørene hva som skjer med enhetene under visse forhold. Bloggen er oppe og går. Kontakt med HDL Nordic. Jobber med kravspesifikasjon. Jobber med å finne passende utstyr for diverse målinger, som vannmålere. Vi har hatt kontakt med HDL Nordic. De mener at vi ikke behøver koden deres, men vi måtte finne ut hvordan de andre komponentene kommuniserer, eksempelvis vannmåler over M-bus. Rørlegger har valgt vannmåler, men denne kan ikke kommuniseres med, antakeligvis er den analog. Møter oppdragsgiver.

Uke 6

Jobber med kravspesifikasjon. Leverandør av varmpumpe er vanskelig å kommunisere med og lite imøtekommende. Drøftet med oppdragsgiver mulighet for å bytte varmpumpe. Litteraturstudie.

Uke 7

Vi har vært i kontakt med teknisk sjef i HDL Nordic, Morten Andersen. Det er mulig for oss å bruke HDL IntelliCenter, og vi skal få (betaversjon) denne i slutten av april eller begynnelsen av mai. Denne serveren er dog fortsatt under utvikling og er basert på Zipato. Vi kommer kommer til å lese oss opp på dette. Denne kan vi programmere på og lage vår løsning akkurat som oppdragsgiver vil ha det. Vi kan bruke Iridium Mobile på mobilfronten. Vi bruker uken på å lære oss Iridium Mobile og arbeider med smarthushistorikk og generelt om smarthus. Vi skal i utgangspunktet begynne med utvikling og sprinter neste uke, det ligger vi ikke an til i det hele tatt. Vi vurderer å begynne på mobilutviklingen og heller gjøre ting vi kan gjøre uavhengig av HDL IntelliCenter.

Uke 8

Vi har igjen vært i kontakt med Morten Andersen. Snakket om teknisk informasjon for alle styringskomponentene. Vi har gått bort fra KNX og bruker heller HDL sitt system og reléer. Vi har også rettet opp i en misforståelse angående varmpumpen - det er ventilene i rørene som skal styres, ikke selve varmpumpen. Varmepumpen skal ikke byttes. Er også på prosjektbefaring i Kragerø igjen. Vi møter salgs- og markedssjef i HDL Nordic, Haavard Haaland. Vi snakket om hva som er mulig å få til med tanke på kabling, i og med at gulv og slikt allerede er lagt. Snakket litt med prosessveileder om oppgaven og hva den har blitt.

Uke 9

Møte med prosessveileder. Oppgaven har nå blitt så liten at den umulig møter krav til arbeidsmengde. Vi diskuterer hva han mener om oppgaven og dens relevans. Tok også opp noen punkter om hvordan vi skal/bør forholde oss til misforståelser og endringer i planen. Prosjektet har endret seg veldig, særlig ut ifra hva vi trodde det skulle bli. Det har vært en god del misforståelser rundt hvilke komponenter som skal styres og hva som er den beste løsningen her. Vi trodde eksempelvis at det var varmpumpen som skulle styres, men det er faktisk ventilene som skal styres.

Vi drøfter muligheten for å bytte oppgave, men i og med at det er såpass seint i prosessen, vil vi naturligvis unngå det på og heller løse oppgaven vi har på en annen møte. Vi var også i kontakt med Ivar Farup som er studieprogramansvarlig for Dataingeniør. Han nevnte muligheten for å endre oppgaven til slik at vi heller lager et skyggesystem med åpne komponenter.

Uke 10

En krise avverget, men vi ligger et godt stykke bak! Vi har kommet frem, etter å ha forklart situasjonen overfor HDL Nordic, at det er mulig for oss å utvikle et styringssystem fra bunnen av. HDL Nordic sender oss dokumentasjon og DLL-filer som skal ta hånd om datapakker som skal sendes til BusPro-systemet. Morten Andersen var veldig positiv til at vi skal lage alt selv. Det involverer at vi selv utvikler drivere som skal kommunisere med det som skal styres. Vi er invitert til å delta på HDLs programmeringskurs. Oppdragsgiver nekter å finansiere det vi anser som nødvendig aktivitet. HDL er villig til å finansiere kurset og opphold i Oslo for oss. Kurset, som er et 2-dagers programmeringskurs, er nødvendig for oss om vi skal ha mulighet til å utvikle opp mot deres system. Kurset, som varte i 2 dager, var delt inn i to deler - grunnleggende kurs og et avansert kurs. Vi fikk pratet med Morten, og fikk en del tips til hvordan vi burde takle perioden som kommer. Siden DLL-filene vi fikk tilsendt er på C, bestemmer vi oss for å utvikle i C på serveren og Java på mobilen. Vi fikk låne med oss en democase fra HDL! Tilbake på NTNU onsdag. Ut uken jobber vi med å sette oss inn i programvaren vi skal benytte. Brukte også tid på å finne ut av kommunikasjonen og meldingstolkning med IP-modul som sender videre til BusPro-systemet.

Uke 11

DLL-filene gir oss egentlig ingenting. Det er null kommentering i koden, men vi leser dokumentasjonen vi fikk tilsendt. Vi får opp kommunikasjon mellom server og applikasjon og jobber videre med å få konstruert pakken som skal sendes til IP-modul og videre inn på BusPro-systemet. Problemer med kommunikasjonen mellom server og applikasjon gjør at vi velger Java på serveren også, kontra C. Kommunikasjon mellom de to er oppe og går relativt fort. Begynner med å legge til funksjonalitet på server og applikasjonen.

Uke 12

Jobber med en pakkeintegritetssjekk, CRC. Denne legges på server og er nødvendig for at å kunne sende pakker til BusPro-systemer. Det jobbes også med meldingshåndtering.

Uke 13

Utvikling av mobilapplikasjonen. Vi får nå til å kommunisere og sende meldinger fra mobilapplikasjon til server og fra server til BusPro-systemet. Vi får sendt en reell pakke som slår på lys! Vi ser videre på hvordan meldinger skal tolkes og hva som skal gjøres når forskjellige pakker mottas. Democasen sender ut mye søppeltrafikk, så det må filtreres bort. Jobber med layout på mobilapplikasjonen som nå heter SHK-applikasjon (SHK-app på kortform). Server kalles for SHK-server, og systemet i sin helhet kalles for SHK - Smarthus Kragerø. Drøfter hva slags luftfuktighetsmåler vi skal bruke. HDL har ikke egne målere for fukt, så vi ser på alternativer som kommuniserer over Z-Wave eller ZigBee.

Uke 14

Utvikling av SHK-applikasjon og SHK-server. Applikasjonene har feil, og krasjer i blant under testing av nye funksjoner. Vi har nå lysstyring og kontroll av reléer . Utfordringer når det kommer til varmestyringskontroller og det finnes lite dokumentasjon angående denne. Brukerdatabase og innlogging er oppe og går. Sykdom og eksamen gjør at lite blir gjort denne uken.

Uke 15

Brukerdatabasefunksjonalitet er ferdig. Jobber fortsatt med temperaturregulering og varmestyringskontrolleren. Lager funksjoner for å lese av temperaturer. Jobber med layout på mobilapplikasjonen. Vi ønsker at applikasjonen ikke skal kreve opplæring i for å brukes av fremtidige brukere. Vi har forhørt oss med oppdragsgiver. Nå som tiden begynner å bli knapp, begynner vi å jobbe med rapport. Vi deler oss i to - Henrik og Jørn-Are jobber videre med utvikling og Edvard jobber nå med rapportskrivning. Design av ikoner til mobilapplikasjonen er i gang.

Jobber med å finne ut av vannmålerproblematikken. Vannmålerne er ikke montert enda, og vi mangler et mottaker for Z-Wave-signaler.

Uke 16

Edvard jobber med rapport. Sendte foreløpig rapport til Frode forrige uke. Jørn-Are og Henrik jobber med mobilapplikasjon. Legger til funksjonalitet for de andre seksjonene i boligen nå (lysstyring og layout på mobilapplikasjon). Samtidig jobbes det fortsatt med å få tak i mottaker for Z-Wave-signaler til vannmåler. Vi har sendt mail til produsent/leverandør, som skulle komme tilbake med svar etter helgen.

Uke 17

Edvard og Jørn-Are jobber nå med rapport. Henrik jobber med å legge til de siste funksjonene og med å få applikasjonen til å kjøre stabilt og litt små designendringer, som synliggjøring av boligens modus (dag, natt, osv). Raspberry-en er mottatt og klargjort for SHK. Utstyr er endelig bestilt (Statusrapport 3), men HDL-komponenter er enda ikke installert i boligen!

Uke 18

Venter på Z-stick som skal kommunisere med Z-Wave-baserte komponenter, som strømmålere. Henrik jobber med finpuss av applikasjon. Jørn-Are og Edvard jobber med rapport.

Uke 19

Mottok Z-stick. Det viste seg at Vanja har kjøpt inn amerikansk, og ikke EU-versjon. Disse sender over en annen frekvens enn de som selges for EU-markedet. Disse må returneres, og samtidig er vannmålere med Z-Wave-moduler bestilt.

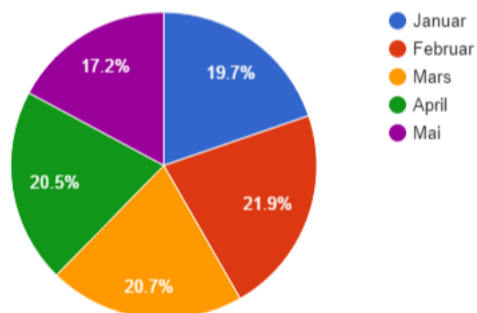
Uke 20

Siste finpuss av rapport.

O Timeføring

	Januar	Februar	Mars	April	Mai
Totalt/pers/mnd	130	144.5	136	135	113
Kumulativt per pers	130	274.5	410.5	545.5	658.5
Gruppen totalt/mnd	390	433.5	408	405	339
Gruppen kumulativt	390	823.5	1231.5	1636.5	1975.5

Totalt



Over er overordnet oversikt. Logging pr dag pr måned ligger i separerte sider, og summer av hver måned er referert til i Totalt/pers/mnd. Diagrammet over viser månedlig fordeling av arbeidstimer i prosent, og som man kan se ble det jobbet mest i februar.

Under er et diagram som viser antallet timer pr person.

