



OSLOMET

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITET

Prosjektrapport

Wallboard for Zisson

Bachelorprosjekt 2020 / 2021

Gruppe 32

Martine Holmberg s236808
Camilla H. Carlsen s197235
Synne Angell s331153
Nikola Sekiw s333975



Institutt for Informasjonsteknologi
Postadresse: Postboks 4 St. Olavs plass, 0130 Oslo
Besøksadresse: Holbergs plass, Oslo

PROSJEKT NR.

32

TILGJENGELIGHET

Åpen

Telefon: 22 45 32 00

BACHELORPROSJEKT

HOVEDPROSJEKTETS TITTEL	DATO
Wallboard for Zisson	26.05.2021
ANTALL SIDER / BILAG	235
PROSJEKTDELTAKERE	INTERN VEILEDER
Martine Holmberg, s236808 Camilla H. Carlsen, s197235 Synne Angell, s331153 Nikola Sekiw, s333975	Aiko Yamashita

OPPDRAUGSGIVER	KONTAKTPERSON
Zisson AS	Thomas Tretli (CTO)

SAMMENDRAG
Gruppen har utviklet et wallboard for Zisson AS. Wallboardet skal vise agenter i sanntid, der hensikten er å kunne vise agentdata på store skjermer rundt om i kontorlokaler hos kunder av Zisson. Dataene skal vises i sanntid, og wallboardet skal kommunisere med Zisson sitt allerede eksisterende brukergrensesnitt, Zisson Interact. Det er brukt teknologiene C# og Angular for back- og frontend implementasjon.

3 STIKKORD
Wallboard
Sanntid
Agenter

Forord

Denne rapporten er en fullstendig rapport som beskriver bachelorprosjektet for gruppe 32 fra OsloMet - Storbyuniversitet, våren 2021. Prosjektet er utført i samarbeid med oppdragsgiver Zisson AS, som er et innovativt IT-selskap med kontorer i Oslo og Stockholm. Målet med prosjektet var å utvikle en ny wallboardløsning for deres eksisterende kunder.

Bachelorgruppen ønsker spesielt å takke Thomas Tretli (CTO hos Zisson) for tett veiledning under hele bachelorprosessen, og Moritz Lilleby (utvikler hos Zisson) for teknisk veiledning. Gruppen ønsker også å takke Kim Josefson (product manager hos Zisson) som har bidratt med inspirasjon og gode innspill under designprosessen. Ønsker også å takke veilederen vår fra OsloMet, Aiko Yamashita, for hjelp underveis i hele prosjektet. Tilslutt ønsker gruppen å takke alle som har lest og gitt gruppen tilbakemeldinger på innholdet i rapporten, samt alle testpersoner som har tatt seg tid og bry med å gi tilbakemeldinger og innspill på prototypene.

Gruppen kom i kontakt med Zisson AS gjennom en bekjent av en av gruppens medlemmer. Det rettes derfor en stor takk til Mahyar Carlsen Harirchi for sitt engasjement og bidrag med formidling av gruppens kunnskap og interesse.

Rapporten er optimalisert for digital lesing da det i innholdsfortegnelsen, samt andre steder er linket til forskjellige deler av dokumentet. Det er også linket til vedlegg i repository på GitHub.

Innholdsfortegnelse

A. Innledning	10
1.0 Introduksjon	10
1.1 Bachelorgruppen	10
1.2 Oppdragsgiver	12
2.0 Bakgrunn for prosjektet	12
3.0 Veiledere	13
3.1 Veiledere fra Zisson	13
3.2 Veileder fra OsloMet	15
4.0 Covid-19	15
5.0 Avklaringer	15
5.1 Rapportstruktur	15
5.2 Zisson sin kontaktsenterløsning	16
B. PROSESSDOKUMENTASJON	19
1.0 Innledning	19
2.0 Planlegging og metode	19
2.1 Forarbeid	19
2.1.1 Fremdriftsplan	19
2.1.2 Arbeidsplan	22
2.1.3 Risikoanalyse	22
2.2 Arbeidsmetodikk	24
2.2.1 Scrum	24
2.2.1.1 Scrumboard (Jira)	26
2.2.1.2 Sprint Planleggingsmøte	26
2.2.1.3 Sprintreview møte	27
2.2.1.4 Standupmøter	27
2.2.2 Utarbeidelser av brukerhistorier	27
2.2.3 Workshops	28
2.2.4 Design thinking	28
2.3 Verktøy og teknologier	29
2.4 Ny kunnskap	36

3.0 Utviklingsprosessen	37
3.1 Designprosess (design thinking)	37
3.1.1 Sette seg inn i brukerens behov / utvikle empati.	38
3.1.1.1 Empathy map	38
3.1.1.2 Undersøkelser	41
3.1.1.3 Personas	42
3.1.2 Definere problemet	45
3.1.3 Utvikling av MVP (Minimum Viable Product)	45
3.1.4 Generere ideer	46
3.1.4.1 Workshop - Oversikt av wallboard	46
3.1.4.2 Skisser - Oversikt av wallboard	48
3.1.5 Prototyper og testing av prototyper	48
3.1.5.1 Workshop - utvikling av prototyper	49
3.1.5.3 Refleksjon av prototypeprosessen	50
3.2 Scrum	52
3.3 Prosjektdagbok	54
4.0 Oppsummering	54
C. KRAVSPESIFIKASJON	57
1.0 Innledning	57
2.0 Kravspesifikasjonsprosessen	57
2.1 Endringer i kravspesifikasjonen	57
2.2 Kravspesifikasjonens betydning	58
2.3 Samsvar med produkt	59
3.0 Endelig kravspesifikasjon	59
3.1 Krav til systemkonstruksjon	59
3.2 Funksjonelle krav	60
3.3 Ikke-funksjonelle krav	62
D. PRODUKTDOKUMENTASJON	64
1.0 Innledning	64
2.0 Beskrivelse av programmet	64
2.1 Siste prototype	65
3.0 4+1 Architectural view	67

3.1 Scenario View	67
3.2 Process View	70
3.2.1 Sekvensdiagram	70
3.2.2 Aktivitetsdiagram	72
3.3 Logical View	76
3.4 Implementation/development View	79
3.4.1 Wallboard	79
3.4.1.1 WallboardService	79
3.4.1.2 AgentRealTimeWallboard	80
3.4.2 Zisson Interact	82
3.4.2.1 Rewave	82
3.4.2.2 WaveWebAPI	84
3.5 Physical View	85
4.0 Sentrale datastrukturer og løsninger	86
4.1 Datastrukturer	86
4.2 Bruk av designmønstre i programmet	89
4.3 Websocket tilkobling	90
5.0 Systemets andre forhold	93
6.0 Brukergrensesnittet	93
6.1 Zisson Interact	93
6.2 Wallboard	95
7.0 Designvalg av sluttprodukt	97
7.1 Tabell	97
7.2 Ikoner	97
7.3 Fargevalg	99
7.3.1 Fargene som er brukt på status	100
7.3.2 Andre fargevalg basert på Zisson Interact	102
7.4 Overordnet design	102
7.4.1 Font	102
7.4.2 Diagram	102
7.4.3 Feilmelding	103
7.4.5 Språk	103
E. TESTDOKUMENTASJON	105

1.0 Identifikasjon	105
2.0 Innledning	105
3.0 Begreper	106
4.0 Testobjekter	107
5.0 Tentativ tidsplanlegging	109
6.0 Fremgangsmåte	109
6.1 Ikke-funksjonelle tester	109
6.2 Funksjonelle tester	112
Enhetstester	112
Integrasjonstester	112
Systemtest	112
6.3 Akseptansetest	113
6.4 Statisk analyse	114
7.0 Kriterier og miljø for testene	114
7.1 Brukertest	114
7.2 Enhetstest	115
7.3 Integrasjonstest	116
7.4 Systemtest	116
7.5 Akseptansetest	118
8.0 Resultater	119
8.1 Brukertester	119
8.1.1 Intervju/ samtale om første prototype	119
8.1.2 Første brukertest - MVP	121
8.1.3 Heuristisk evaluering av MVP prototype	124
8.1.4 Andre runde brukertest - MVP	129
8.1.5 Tredje runde brukertester - MVP	132
8.1.6 Fjerde runde brukertester - MVP	138
8.1.7 System usability scale test	139
8.2 Enhetstester	140
8.2.1 Backend	140
8.2.2 Frontend	142
8.3 Integrasjonstester	144

8.4 Systemtester	147
8.5 Akseptansetester	148
8.5.1 Akseptansetest av kravspesifikasjon	148
8.5.2 Akseptansetest av prototype	149
8.5.3 Akseptansetest av inkrementer	149
8.5.4 Akseptansetest av sluttprodukt	150
9. Oppsummering	150
F. AVSLUTNING	151
1.0 Resultatet	151
2.0 Hva kunne blitt gjort annerledes?	152
3.0 Forbedringspotensiale	153
4.0 Læringsutbytte	154
G. REFERANSER	155
H. VEDLEGG	162
1.0 Dokumentasjon av Scrum	162
1.1 Sprint 1 (1.februar - 12. februar)	162
1.2 Sprint 2 (15.februar - 26.februar)	162
1.3 Sprint 3 (1.mars - 12.mars)	165
1.4 Sprint 4 (15.mars - 26.mars)	169
1.5 Sprint 5 (6.april - 16.april)	173
1.6 Sprint 6 (19.april - 30.april)	176
2.0 Prosjektdagbok	183
3.0 Arbeidsplan	206
4.0 Dokumentasjon av tester	210
4.1 Skjermbilder av tester i Katalon	210
4.2 Akseptansetester av inkrementer	221
4.3 Spørsmål fra brukertester	225
4.3.1 Intervju/ samtale om første prototype	225
4.3.2 Første brukertest - MVP	226
4.4 Andre runde brukertest - MVP	226
4.4 SUS spørreundersøkelse	229
4.4.1 Skjermbilder av SUS spørreundersøkelse (Survey monkey)	229

4.4.2 Utregning SUS spørreundersøkelse	233
5.0 Utkast prototyper	234
6.0 Kildekode	234
I. REFERANSER FOR VEDLEGG	235

A. Innledning

1.0 Introduksjon

Etter ønske fra bachelorgruppen om et prosjekt som skulle være så relevant som mulig for arbeidslivet ønsket gruppen å jobbe med en ekstern oppdragsgiver med primærfokus på programmering.

Gruppen er den første bachelorgruppen som Zisson har samarbeidet med, og oppdragsgiver har etter beste evne prøvd å tildele gruppen en oppgave de mener var av fin størrelse både i omfang og kompleksitet, som både kunne gi gruppen utfordringer, men også mestring.

1.1 Bachelorgruppen

Bacheloroppgaven er utført av bachelorgruppe 32 fra OsloMet - Storbyuniversitet. Gruppen består av fire studenter som har jobbet tett sammen gjennom hele studiet. Tre av gruppens medlemmer (Nikola, Camilla og Martine) går på Anvendt datateknologi, mens en (Synne) går på Informasjonsteknologi.



Martine Holmberg
martine.holmberg@gmail.com

Rolle:

Prosjektleder. Teknisk ansvarlig (frontend/ backend), fullstack utvikler.



Camilla H. Carlsen
camill200592@gmail.com

Rolle:

UX-leder. Tillitsvalgt, rapportansvarlig, fullstack utvikler.



Synne Angell
synneang@gmail.com

Rolle:

Teknisk ansvarlig (frontend/ backend), fullstack utvikler.



Nikola Sekiw
nsekiw@gmail.com

Rolle:

Testleder. Koordinator, fullstack utvikler.

1.2 Oppdragsgiver

“Zisson er et innovativt IT-selskap med kontorer i Oslo og Stockholm, som utvikler og drifter kontaktsenter- og sentralbordløsninger. Zisson ble etablert i 2007 med ambisjoner om å forenkle og forbedre kommunikasjonen mellom bedrifter og sluttkunder.”

([Zisson](#), 2020).



Figur 0: Zisson-logo

Zisson Contact Center er en skybasert og omnikanal telefonsentral. De setter raskt opp fleksible kommunikasjon- og kundeserviceløsninger for henvendelser fra telefon, webchat, e-post, chatbot, sosiale medier og SMS. De er partnere med blant annet Telenor, Techstep, Friday Networks og Mobit. Kundene deres er bedrifter, og sammen med partnerne deres har de over 400 kunder i offentlig og privat sektor.

Adresse: Zisson AS, Tollbugata 32, 0157 Oslo

2.0 Bakgrunn for prosjektet

I dag opplever Zisson at deres eksisterende wallboard er utdatert, både når det gjelder design og teknologi. Designet er opprinnelig fra 2010 og var laget som et testprosjekt. Brukergrensesnittet deres, Zisson Interact, er skrevet om med nyere teknologi som angular på frontend, og nå vil de gjøre det samme med wallboardet. Zisson ønsker at gruppen gjennom dette prosjektet skal løfte wallboardet ved bruk av nyere teknologier og redesign.

Et wallboard viser informasjon om kundesenteret på TV-skjermer rundt om i kontorlokalene, og kan gi en rask oversikt over hvor godt agentene jobber og hvilke køer som trenger mer oppmerksomhet. Wallboard viser vanligvis informasjon om hvordan agenter har klart seg i løpet av dagen i form av å vise informasjon om hvor mange som har svart, gjennomsnittlig taletid, gjennomsnittlig ventetid, maks ventetid, etc, samt også vise live visning av callsenterets agenter og køer. En kunde kan selv definere wallboard med hvilke køer som skal vises og dataintervallet.

Applikasjonen skal være moderne og enkel for en administrator å sette opp, samt være oversiktlig og intuitiv for brukerne av systemet. Det skal være mulig og enkelt sette opp wallboardet med blant annet elementer som agentnavn, agentgruppe, tid i status, callstatus, profil, samt mulighet for filtrering av ønskede elementer.

3.0 Veiledere

3.1 Veiledere fra Zisson



Thomas Tretli
CTO
thomas.tretli@zisson.com

Rolle:
Hovedkontaktperson



Moritz Lilleby
Utvikler
moritz.lilleby@zisson.no

Rolle:
Teknisk veileder



Kim Josefson
Customer Success Manager

Rolle:
Veiledning innen design og prototyping



Christer Nygaard
Customer Success - Support

Rolle:
Veiledning og testperson innen prototyping

3.2 Veileder fra OsloMet



Aiko Yamashita
Førsteamanuensis

Aiko.Yamashita@dnb.no,
Aiko.Yamashita@oslomet.no

Rolle:
Intern veileder

4.0 Covid-19

Covid-19 har gitt gruppen flere utfordringer med kommunikasjon. Dette gjelder spesielt ved teknisk hjelp og brukertesting. Det har ikke vært like enkelt å brukerteste personer uten å kunne møtes fysisk, og det har heller ikke vært enkelt å rekruttere testpersoner. Det har også vært perioder der det har vært tungt å jobbe atskilt. Gruppen har likevel hatt jevnlige møter på teams gjennom hele perioden og samt skriftlig kommunikasjon over Slack og Messenger.

5.0 Avklaringer

5.1 Rapportstruktur

Det er valgt å bygge opp oppgaven fra A-l, etter disse hovedoverskriftene: A. Innledning, B. Prosessdokumentasjon, C. Kravspesifikasjon, D. Produktdokumentasjon, E. Testdokumentasjon, F. Avslutning, G. Referanser, H. Vedlegg og I. Referanser for vedlegg.

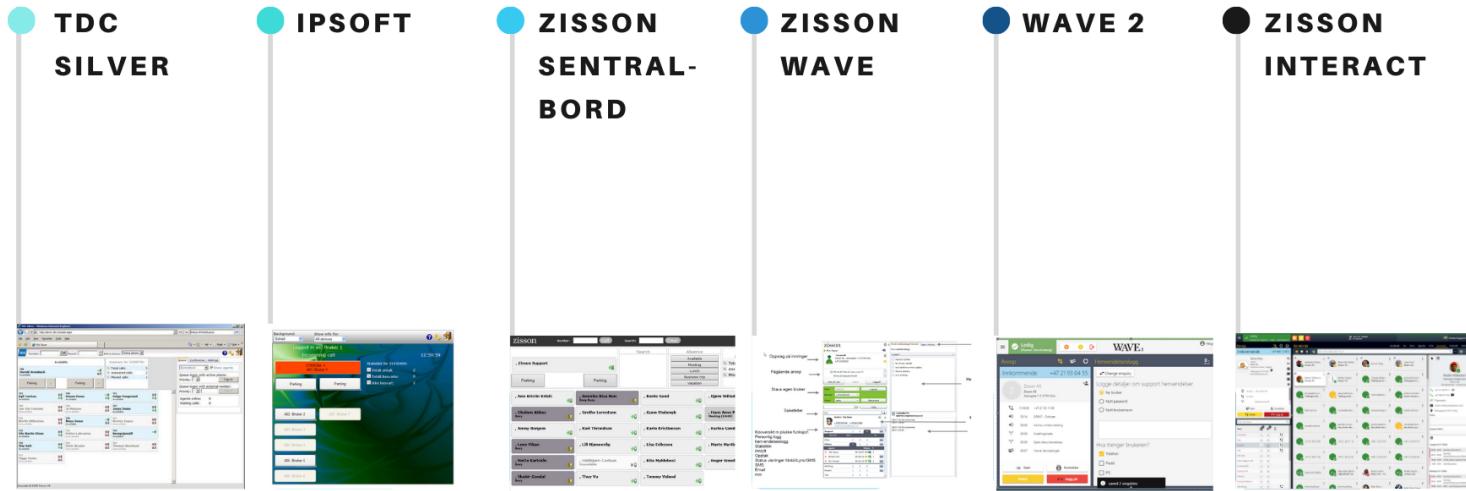
B. Prosessdokumentasjon inneholder både planleggingsprosess, designprosess og scrum metodikk. C. Kravspesifikasjon inneholder den endelige kravspesifikasjon, med logisk modell, samt brukerhistorier. D. Sluttdokumentasjon inneholder utvalgt kildekode, bilder av brukergrensesnitt, begrunnelse av designvalg, samt oppbygging av systemet, og viktige datastrukturer. E. Testdokumentasjon inneholder planlegging, kriterier og resultat av alle brukertester, enhetstester, integrasjonstester og systemtester.

5.2 Zisson sin kontaktsenterløsning

I figur 1 kan man se at navnet og brukergrensesnittet til kontaktsenterløsningen til Zisson har endret seg gjennom tidene, som i dag heter Zisson Interact. Heretter vil kontaktsenterløsningen som helhet bli kalt for *Zisson Interact*. Når det blir henvist til frontend prosjektet tilhørende Zisson Interact vil dette heretter bli kalt *ReWave*.

BRUKERGRENSESNIITT I ZISSON

2007-2020



Figur 1: Forklarer brukergrensesnittet til Zisson fra 2007 til i dag



OSLOMET

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITET

Prosessdokumentasjon

Wallboard for Zisson

Bachelorprosjekt 2020 / 2021

Gruppe 32

Martine Holmberg s236808
Camilla H. Carlsen s197235
Synne Angell s331153
Nikola Sekiw s333975

B. PROSESSDOKUMENTASJON

1.0 Innledning

I denne delen av oppgaven blyses det hva slags planlegging- og prosess-metodikk som er brukt underveis i prosjektet.

2.0 Planlegging og metode

2.1 Forarbeid

2.1.1 Fremdriftsplan

Fremdriftsplanen, figur 2, var noe av det første som ble utarbeidet under planleggingen. En fremdriftsplan er en oversikt over de ulike aktivitetene som inngår i prosjektet med tidsestimering av hver aktivitet.

Til å begynne med ble de ulike aktivitetene identifisert med avhengigheter, for så en tidsestimering av hver aktivitet. En stor del av fremdriftsplanen ble delt opp i sprinter på 2 uker, da selve utviklingsprosessen av prosjektet skulle foregå med en smidig tilnærming. Ut i fra denne informasjonen ble det mulig å lage et søylediagram som utgjør fremdriftsplanen for prosjektet. Denne måten å fremstille fremdriftsplanen på ga de involverte i prosjektet et godt visuelt bilde av de ulike aktivitetene og deres avhengigheter.

Fremdriftsplanen ga verdifull informasjon. Bde i form av å planlegge prosjektet nøye på forhånd og få en oversikt over tidsbruken på de ulike aktivitetene, men også for å kunne måle fremdriften underveis i prosjektet. På denne måten var det hele tiden en

oversikt over hvordan prosjektet lå an med tanke på estimert tid. Da det ikke er mulig å utsette sluttdateoen på et bachelorprosjekt, ble det mer et spørsmål om man måtte sette inn mer tid i enkelte uker for å holde tritt med fremdriftsplanen.

Her kommer også valget av en smidig arbeidsmetodikk inn. Da man for hver sprint som var planlagt, kun valgte de høyeste prioriterte funksjonalitetene, ga det en sikkerhet på at det ble levert et funksjonelt produkt til slutt. Spørsmålet ble heller hvor mye som rakk å bli implementert i sluttproduktet av annen ønsket funksjonalitet, såkalt “nice to have” funksjonalitet.

Fremdriftsplanen ble delt med oppdragsgiver, Zisson, for at de også skulle ha en oversikt over fremdriften og hvilke aktiviteter som ble gjennomført til enhver tid.

Aktivitet	Startuke	Varighet (uker)	4-10 Jan	11-17 Jan	18-24 Jan	25-31 Jan	1-7 Feb	8-14 Feb	15-21 Feb	22-28 Feb	1-7 Feb	8-14 Mar	15-21 Mar	22-28 Mar	29-4 Apr	5-11 Apr	12-18 Apr	19-25 Apr	26-2 Mai	3-9 Mai	10-16 Mai	17-23 Mai	24-30 Mai	31-6 Juni	7-10 Juni	
			1	2	3	4	5 - Sprint 1	6	7 - Sprint 2	8	9 - Sprint 3	10	11 - Sprint 4	12	13	14 - Sprint 5	15	16 - Sprint 6	17	18	19	20	Halvuke 21	22	23	
Planleggingsfase og skriving av prosjekt:	1	4																								
Rapportskriving	1	21																							Innlevering 26. mai	
Utarbeidelse av kravspesifikasjon	2	13																								
Dagbok	2	15																								
Oppsett av programmer	4	2																								
Prototyping	5	3																								
Utvikling	5	11																								
Testing	5	11																								
Ferdigstilling av sluttrapport	17	4,5																								
Presentasjon	12 og 21	4															Mid-term: Mandag 29.mars									

Figur 2: Fremdriftsplan

2.1.2 Arbeidsplan

Det ble utarbeidet en tentativ arbeidsplan etter fremdriftsplanen var på plass.

Arbeidsplanen er en mer detaljert oversikt over hva som må gjøres innenfor de ulike aktivitetene som er planlagt i fremdriftsplanen.

Til å begynne med var det mulig å planlegge i detalj de første ukene av prosjektet, som innebar aktivitetene: planleggingsfase, utarbeidelse av kravspesifikasjon, oppsett av programmer og prototyping. Videre utfylling av arbeidsplanen ble gjort under sprint planleggingsmøter, der det ble avgjort hvilke oppgaver som skulle prioriteres for hver sprint.

Se punkt [3.0 Arbeidsplan](#) i vedleggene for flere detaljer.

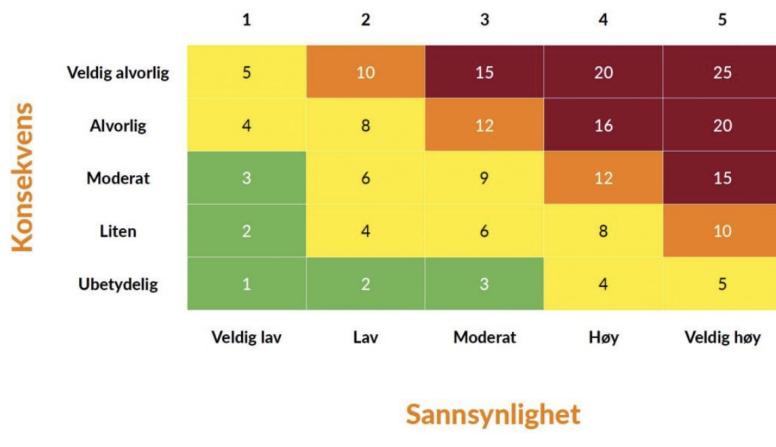
2.1.3 Risikoanalyse

Som en del av planleggingen av prosjektet ble det utarbeidet en risikoanalyse. Store Norske Leksikon definerer risikoanalyse slik:

“Risikoanalyse er en studie av risiko for å få innsikt i hva slags hendelser som kan skje, hvorfor de kan skje og hva konsekvensene vil kunne være.”

(Aven, 2020)

Figur 3 viser modellen som ble brukt for å fylle ut fargekode og nummer ut fra alvorlighetsgraden av konsekvensene.



Figur 3: Risikoanalyse-modell (Vera Tank, u.å.)

Risikoanalysen bestod av alle hendelsene som prosjektgruppen antok hadde en mulighet for å kunne oppstå under prosjektet. Videre ble sannsynlighet vurdert og konsekvensen av hendelsen, som da ga et tall på en risikorangering. Ut ifra dette diskuterte gruppen forebyggende tiltak til hver av hendelsene og ga ansvar til de ulike rollene som er definert i prosjektet.

Risikoanalysen ses i figur 4 på neste side.

Nr.	Hendelse	Sannsynlighet	Konsekvens	Risikorangering	Forebyggende tiltak	Ansvarlig
1	Lite tid	Moderat	Alvorlig	12	Godt utarbeidet arbeidsplan og fremdriftsplan.	Utviklere
2	Kortvarig sykdom	Moderat	Liten	6	Standupmøter.	Prosjektleder, utviklere
3	Langvarig sykdom	Lav	Alvorlig	8	God fremdriftsoversikt. Standup-møter.	Prosjektleder, utvikler
4	Konflikt innad i gruppa	Lav	Moderat	6	God kommunikasjon, ta opp ting tidlig, evt. møter med en utenforstående. Ta opp med tillitsvalgt.	Tillitsvalgt, utviklere
5	Teknologiske problemer	Moderat	Moderat	9	Ta kontakt med teknisk leder på Zisson og få hjelp.	Teknisk ansvarlig, utviklere
6	Tap av data	Lav	Veldig alvorlig	10	Lage backup underveis. Bruke Git.	Teknisk ansvarlig, utviklere
7	Kravendringer	Lav	Alvorlig	8	Smidig utvikling. Utarbeide krav sammen med Zisson. Sette av mer tid/bedre kommunikasjon underveis	Scrum master
8	Ekstra runder med brukertesting	Moderat	Liten	6	Sette av nok tid til designprosessen.	UX -leder.
9	Oppdragsgiver/organisasjon går konkurs	Veldig lav	Veldig alvorlig	5	Dette er noe gruppen ikke kan påvirke	Zisson
10	Omfang av løsningen er mer krevende enn planlagt	Moderat	Alvorlig	12	Ikke gape over alt på en gang. Scrum sikrer at prosessen blir delt opp i konkrete mindre oppgaver.	Zisson, utviklere, veileder
11	Ny teknologi utviklerne ikke er kjent med	Veldig høy	Liten	10	Bruke tid på å sette oss inn i verktøy og teknologi.	Utviklere
12	Uenighet om kvaliteten på det ferdige produktet	Lav	Moderat	6	Alle er involvert i prosessen hele veien.	Zisson, utviklere

Figur 4: Prosjektets risikoanalyse

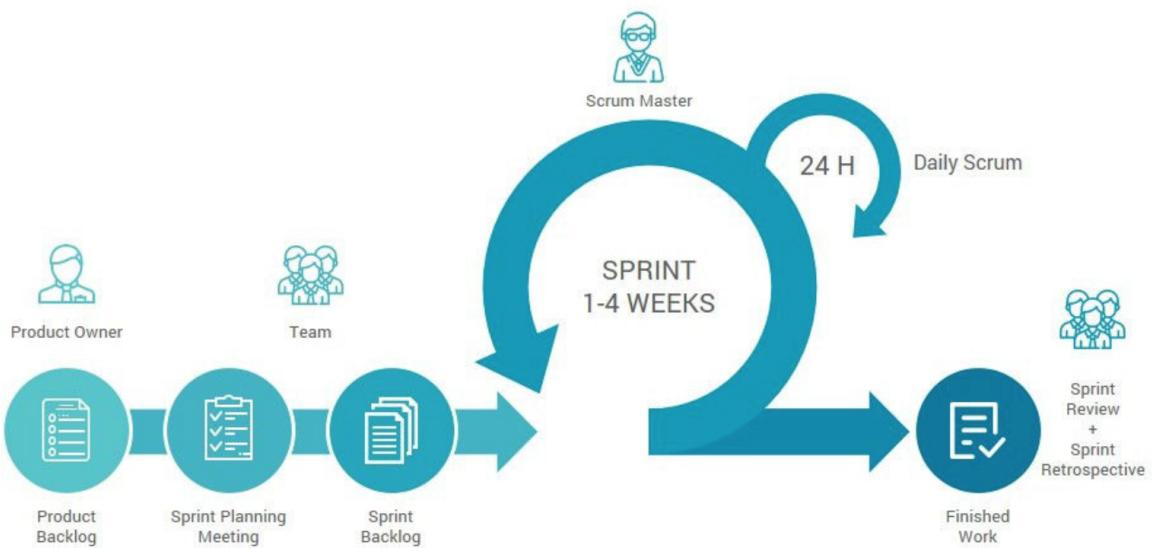
2.2 Arbeidsmetodikk

2.2.1 Scrum

Det ble planlagt å bruke Scrum som arbeidsmetodikk for utviklingen av systemet.

Scrum er en smidig arbeidsmetodikk hvor planleggingen gjøres litt etter litt og hvor det ikke er like stort behov for detaljert beskrivelse av kravene. Dette gjør det enklere å endre prosessen for å tilpasse endrede krav fra kunden underveis i prosjektet. Hovedfokuset baserer seg på å utføre de viktigste aktivitetene først (Sommerville, 2016, s. 75). Scrum ble utviklet for å utnytte tid og ressurser på en god måte (Sommerville, 2016, s. 69), og prosessen er basert på en sprint syklus som vist i figur 5.

Hver sprint baserer seg på en produkt backlog, hvor det under hver iterasjon i prosessen blir levert et inkrement av produktet (Sommerville, 2016, s. 70). Backlogen består av en kravspesifikasjon som ofte er utviklet som brukerhistorier (Sommerville, 2016, s. 70). Et inkrement er et tillegg i funksjonaliteten som utvikles gjennom en ny iterasjon (Sommerville, 2016, s. 49).



Figur 5: Scrum sprint prosess (Kukhnavets. P. 2019).

Et sprint er en serie med iterasjoner og varer vanligvis mellom 2-4 uker, hvor det i hver sprint leveres et inkrement av systemet (Sommerville, 2016, s 70). En iterasjon er derimot en fase i prosjektet og gjennom nye iterasjoner kan kvaliteten på samme funksjonalitet forbedres. I samråd med Zisson ble det valgt at sprintene skulle vare i 2 uker. Ved starten av hver sprint ble det valgt ut prioriterte krav fra backlog som det skulle arbeides med under sprinten, hvor hele teamet fikk være med å bestemme hvilke (Sommerville, 2016, s. 70). Dersom kravene ikke ble ferdige under sprinten ble kravene lagt tilbake i backlogen til senere sprinter (Sommerville, 2016, s.70). Etter hver sprint ble det også gjort en gjennomgang på hvordan det hadde gått og hvordan gruppen kunne forbedre seg i løpet av neste sprint.

En av grunnene til å velge Scrum er at dette systemet skal utvikles og leveres innen en kort tidsramme og det kan forekomme endringer i kravspesifikasjonene underveis. Sommerville skriver at det er vanlig at kravspesifikasjonen endrer seg underveis fordi det er vanskelig for kundene å forutse hvordan systemet vil påvirke interaksjon med andre systemer, samt hvilke brukeroperasjoner som skal automatiseres (Sommerville. I., 2016, s. 57).

Fordelene ved en smidig tilnærming er at endringer i kravspesifikasjonene enkelt kan implementeres ved neste sprint og tjenesten kan utvikles relativt raskt (Sommerville, 2016. s. 57).

[2.2.1.1 Scrumboard \(Jira\)](#)

For å etterse at arbeidsmetodikken ble fulgt ble programmet Jira tatt i bruk som Scrumboard. Scrumboard er et verktøy som hjelper teamet med å produsere en sprint backlog, som er en essensiell del av arbeidsmetodikken Scrum. Scrum backlog er en prioritert liste med arbeidsoppgaver og funksjonalitet, og i dette tilfellet bestod den av en samling av brukerhistorier som ble utarbeidet under kravspesifikasjonsprosessen. Se [C. Kravspesifikasjon](#) for mer detaljer.

Jira ble valgt som prosjektstyringsverktøy da dette er et verktøy oppdragsgiver er godt kjent med, samtidig som det er et velkjent verktøy brukt under smidig prosjektstyring.

[2.2.1.2 Sprint Planleggingsmøte](#)

Før hver sprint ble det avholdt et sprint planleggingsmøte med Zisson for å kartlegge hvilken funksjonalitet som var viktigst for den utvalgte sprinten. Dette sikret en tett kommunikasjon med oppdragsgiver under hele prosessen.

2.2.1.3 Sprintreview møte

Etter hver sprint ble det organisert et sprint review-møte, slik at gruppen fikk tilbakemeldinger på det arbeidet som var utført under sprinten. Gruppen gikk igjennom hver brukerhistorie, presenterte kode og brukergrensesnitt som var blitt implementert under sprinten.

2.2.1.4 Standupmøter

Under sprintene var målet at teamet skulle ha et kort daglig stand up-møte hver dag, hvor prosjektgruppen skulle gå gjennom progresjon, og om nødvendig hva som måtte opprioriteres (Sommerville, 2016, s.71). Slik visste alle på gruppen hva som ble gjort av arbeid til enhver tid.

Gruppen har hatt tett kontakt hver dag, og alle har hatt en god oversikt over hva gruppemedlemmene har arbeidet med. Ved utfordringer har gruppen hjulpet hverandre og tatt kontakt med teknisk veileder hos Zisson ved behov.

2.2.2 Utarbeidelser av brukerhistorier

I utarbeidelsen av brukerhistoriene, utførte prosjektgruppen flere møter sammen med Zisson. Det ble blant annet gjennomført flere møter hvor det ble diskutert MVP og hvilken funksjonalitet som var den viktigste for prosjektet. Med MVP menes et produkt med nok funksjoner til å validere en produktidé tidlig i produktutviklingsfasen (ProductPlan, u.å.).

Etterhvert som kravspesifikasjonen ble tydeligere lagde prosjektgruppen et utkast til brukerhistorier som ble utformet i verktøyet Jira. Her fikk gruppen retningslinjer fra Zisson om hvordan dette best kunne utføres med blant annet inndelinger etter epics, tasks, og stories. Da gruppen hadde opprettet brukerhistoriene ble det

gjennomført et nytt møte for å gå i gjennom og sjekke at kravene var forstått riktig, deretter ble det gjort endringer basert på tilbakemeldingene.

2.2.3 Workshops

En av metodene gruppen brukte for møter var workshops hvor det ble anvendt verktøyet Miro for å utvikle ideer både sammen som gruppe og sammen med oppdragsgiver. Slik ble oppdragsgiver en del av prosessen fra start til slutt.

2.2.4 Design thinking

Under prosjektet er det brukt design thinking fra Hasso Plattner institutt for design ved universitetet i Stanford (Scott Doorley. et.al., 2018). Design thinking er en prosess som blir brukt for å forstå brukerne, omdefinere problemet, og skape innovative løsninger for prototype og test. Ved å involvere fem faser - empati, definering, idé, prototype og testing - er det mest nyttig for å takle problemer som er dårlig definert eller ukjent (Friis et al., 2021).

Det ble dermed valgt å bruke en designprosess med design thinking som metode. Dette for å sette brukeren i sentrum for utviklingen av produktet. I følge Kathryn McElroy (2017, s.22) står dette i motsetning til å lage et produkt hvor brukeren må tilpasse seg produktet.

Andre fordeler ved design thinking baserer seg på at det er viktig for å forstå problemet som skal løses, kommunisere designideer til bedriften og få tilbakemeldinger på dette, samt for å teste designet på brukere, få tilbakemeldinger og gjøre endringer (McElroy, 2017, s.36-37).

Design thinking er delt inn i forskjellige faser (Dam, 2021):

1. Empatisering

2. Definere problem
3. Generere ideer
4. Prototyping
5. Testing

Alle fasene er sentrale i planleggingen og utviklingen av systemet.

2.3 Verktøy og teknologier

De fleste av verktøyene som har blitt brukt i prosjektet er valgt etter anbefalinger fra Zisson, da dette er verktøy de selv bruker. Teknologiene er valgt grunnet krav som kom frem i oppgavebeskrivelsen og etter samtaler med Zisson. Verktøyene som er brukt til design og kommunikasjon er valgt av gruppen selv, da dette var verktøy som en eller flere hadde kjennskap til og ønsket å bruke grunnet gode erfaringer med disse.

IDE

Visual Studio Code

Visual Studio Code er et open source IDE-verktøy utviklet av Microsoft. VSC kan brukes på alle operativsystemer, og tilbyr en rekke utvidelser som f.eks. Git (Microsoft, u.å). VS Code ble brukt til alle frontend prosjektene.



Visual Studio (Community)

I likhet med VS code er Visual Studio også et IDE-verktøy utviklet av Microsoft. Det er et integrert utviklingsmiljø for .NET plattformen til Microsoft, og brukes til å utvikle programmer for Windows, samt nettsider og applikasjoner som kjører på det mobile operativsystemet til Microsoft (Microsoft, u.å).

Gruppen hadde kjennskap til verktøyet fra tidligere av, og det ble brukt til alle backend prosjektene.



Frontend

Angular

Angular er en plattform som er laget for utvikling av single page webapplikasjoner (SPA), hvor det brukes HTML og TypeScript (Angular, u.å). Gruppen hadde noe kjennskap til Angular fra før. Plattformen er brukt både i AgentRealTimeWallboard-prosjektet og ReWave-prosjektet.



RxJS

RxJS er et bibliotek for blant annet utvikling av webapplikasjoner. Biblioteket tilbyr en måte å håndtere diverse hendelser på og ved hjelp av "observables" blir det lettere å lage asynkron kode(RxJS, u.å). Gruppa hadde ikke kjennskap til dette biblioteket fra tidligere, og det ble derfor brukt tid på å bli kjent med dette slik at gruppa kunne ta det i bruk. I oppgaven ble biblioteket brukt i AgentRealTimeWallboard-prosjektet for kode som hadde med websocket å gjøre.



Bootstrap

Dette er et bibliotek for frontend utvikling. Biblioteket brukes for å style web-komponenter, i likhet med CSS. Det er et enklere alternativ til CSS, men disse brukes likevel ofte sammen. Bootstrap er vedlikeholdt av twitter og fokuserer på mobile-first og responsivt design (Bootstrap, u.å). Dette biblioteket ble brukt hovedsakelig i AgentRealTimeWallboard prosjektet for å style wallboardet, og var noe gruppen hadde kjennskap til fra før.



Backend

C#

C# ble utviklet Microsoft i 2000 som en del av .NET, og er et moderne objektorientert og type-safe programmeringsspråk. Det kan brukes til å bygge applikasjoner som kjører i .NET-økosystemet (Microsoft, u.å). C# ble brukt til både WallboardService og WaveWebAPI i backend prosjektene, og var et språk gruppen hadde jobbet noe med tidligere.



.NET Core

.NET Core gitt ut i 2016 og er den nye multi plattform implementasjonen av .NET Framework. Tidligere var det kun mulig å kjøre .NET på Windows, mens med .NET Core støttes alle operativsystemene (Microsoft, u.å).



RabbitMQ

RabbitMQ er et åpent kildekodeverktøy som fungerer som en “meldingsmegler” og støtter flere typer meldingsprotokoller. Det gir applikasjoner en felles plattform for å sende og motta meldinger, hvor meldingene også bevares trygt på plattformen i mellomtiden (RabbitMQ, u.å). Gruppen hadde ikke erfaring med RabbitMQ fra tidligere, og måtte dermed sette seg inn i dette. RabbitMQ ble brukt for å få tak i JSON-strenger med agenter som avfyrte ulike eventer, og som en midlertidig erstatning for agent-eventer før Zisson Interact var på plass eller da det var feil med Zisson Interact-systemet.



Etter gruppen fikk tilgang til VPN, ble RabbitMQ erstattet med MessageQueue. MQ er en kø med meldinger sendt mellom forskjellige applikasjoner. På denne måten kunne eventer om agenstatuser mottas fra Zisson sin egen MQ.

LINQ (Language Integrated Query)

LINQ er Microsoft sitt spørrespråk, og gjør at det er enklere å blant annet få datatilgang fra objekter i databaser (Microsoft, u.å). I dette prosjektet ble LINQ brukt til å hente ut verdier fra Obelix-databasen som Zisson benytter seg av.



Database

Obelix

Det er brukt Zisson sin database Obelix som er en relasjonsdatabase. Gruppen benyttet seg av denne i form av databasekall fra backendprosjektet for uthenting av ønskede data.

Versjonskontroll og andre dokumentasjonsverktøy

Git

Git er et åpent kildekode-distribuert versjonskontrollsysteem brukt for å håndtere programmeringsprosjekter. Git gjør det mulig for utviklere å jobbe sammen med samme kode. Det er også mulig å se hvem som har gjort hva, historikken på endret kode, og alle kan laste ned lokale versjoner av koden for å gjøre endringer lokalt. Dette gjør også at man kan jobbe offline, alltid har backup av kodebasen, og utviklere kan teste hele systemet på lokale maskiner etter at endringer er gjort før man pusher det ut til de andre (Git, u.å). Git ble derfor brukt i oppgaven da dette gjorde samarbeidet enklere og gjorde at koden lå trygt på skyen med mulighet for å se historikken, og var noe gruppen var kjent med fra før.



GitHub og GitHub Desktop

GitHub og GitHub Desktop er plattformer for versjonskontroll og samarbeid, og anvender Git. For at det skal være enklere å jobbe med Git er det



utviklet GUI som gjør det enkelt å få utført oppgaver uten å måtte skrive kommandoer for det. I dette prosjektet brukes disse til henting av repository, pull/push, oppretting av brancher og pull-request (GitHub, u.å). GitHub er nettversjonen, mens GitHub Desktop er en applikasjon som lastes ned på datamaskinen. Begge disse ble brukt i oppgaven og var verktøy gruppen var kjent med fra før.



Jira

Jira er et verktøy for smidig utvikling, og som hjelper med blant annet oppretting av brukerhistorier, planlegging av sprints, tildeling av oppgaver til gruppemedlemmer, sporing av hva som har blitt gjort og holde oversikt over hva som må gjøres i hver sprint (Jira, u.å). Jira ble brukt av Zisson fra før, mens ingen i bachelorgruppen var kjent med dette verktøyet fra tidligere av.



Testverktøy

TestCafé

TestCafé er et testverktøy for ende-til-ende-tester som kjører på Node.js-plattformen. Verktøyet er ikke avhengig av andre test-programvarer (TestCafe, u.å). I dette prosjektet ble TestCafé brukt til integrasjonstesting i Angular, og var et verktøy som var nytt for gruppen.



xUnit

xUnit et åpent kildekode-enhetstestingverktøy for .NET-rammeverket som brukes til å enhetsteste C# kode i backend (xUnit, u.å). Verktøyet ble brukt i WallboardService-prosjektet til å teste de mest kritiske delene av programmet. Ingen i gruppen hadde erfaring med dette fra tidligere.

Xunit.net

Karma

Karma er et verktøy som kjører tester for JavaScript som kjøres på Node.js. Dette verktøyet er godt egnet til å teste Angular-applikasjoner og JS-prosjekter. Karma kan brukes til å kjøre tester ved hjelp av JavaScript-testingpakker som Jasmine og Mocha, og utføre testene i den nettleseren og plattformen man ønsker (Karma, u.å). Karma ble brukt til enhetstester i AgentRealTimeWallboard-prosjektet, og var nytt for gruppen.



Katalon & Katalon Recorder

Katalon er et multiplattform-verktøy for automatisert testing. Det er designet for å lage og gjenbruke automatiserte testskripter og tillater kjøring av automatiserte tester på brukergrensesnitt-elementer (Katalon, u.å). I prosjektet ble Katalon Recorder brukt, som er en nettleserutvidelse. Denne ble brukt til å gjennomføre automatiserte systemtester på systemet, og var et verktøy gruppen hadde brukt tidligere. Heretter blir dette referert til som Katalon.



Andre verktøy til dokumentasjon, design, prototyping, diagram og kommunikasjon

Miro

Miro er et digitalt verktøy hvor man kan lage digitale whiteboard-tavler. Verktøyet gjør det mulig å jobbe effektivt sammen online, med alt fra idemyldring og skisser, til planlegging og administrering av smidige arbeidsflyter (Miro, u.å.). Miro ble brukt for å gjennomføre digitale workshops med alt ifra idemyldring til prototypeutvikling. Det ble også gjennomført brukertester, laget KanBan-boards og andre visualiseringer gjennom Miro, som har vært et godt brukt verktøy i dette prosjektet.



Google Drive

Google Drive en skylagringstjeneste fra Google som gjør det lettere å samarbeid med kollegaer online, samt sikrer sikkerhetskopiering av alle dokumentene. I dette prosjektet ble Google Drive brukt til alt av rapporter, dokumentasjoner og presentasjoner (docs, regneark/ sheets, slides), og er et verktøy gruppen har god kjennskap til fra tidligere av.



Slack

Slack er en samarbeidsplattform med skybasert chatttjeneste. Det gjør det enkelt å kommunisere med gruppen ved å lage dedikerte kanaler (chatter) for et spesifikt tema. Det er også mulig å ringe, sende direktemeldinger og dele skjermen. I prosjektet ble dette brukt til skriftlig kommunikasjon og organisasjon.



Draw.io

Draw er en programvare for å lage diagrammer digitalt, og tilbyr lagring på sky. Man kan velge mellom automatisk layoutfunksjon, eller en tilpasset layout. Programvaren er rask og enkel å bruke med dra og slipp-funksjoner, samt at den tilbyr et stort utvalg av former og hundrevis av visuelle elementer (Computer Hope, 2020). Verktøyet ble brukt til å lage modeller og diagrammer som aktivitets-, sekvens- og use case-diagrammer, og er et verktøy gruppen har kjennskap til fra før.



OpenVPN

OpenVPN er programvare for opprettelse av egne virtuelle nettverk for å sikre kommunikasjonen mellom to endepunkter (OpenVPN, u.å). Det ble brukt



Zisson sitt VPN i dette prosjektet slik at det ble mulig å få kontakt med Obelix-databasen og Zisson sin MQ.

Microsoft Teams

Microsoft Teams er en samarbeidsapplikasjon og gjør det enklere for et team å holde seg organisert og ha samtaler.

Man kan opprette egne kanaler hvor man kan holde møter, ha samtaler eller dele filer. Man kan holde oversikt over dagens eller ukens agenda, og alle møter synkroniseres med Outlook-kalenderen (Microsoft, u.å.). Microsoft Teams er brukt til dette formålet, og er et verktøy gruppen har noe kjennskap til fra før.



[2.4 Ny kunnskap](#)

Prosjektet har krevd at gruppen har vært nødt til å sette seg inn i nye teknologiske områder, noe som til tider har vært både tidkrevende og utfordrende.

[Tekniske løsninger](#)

Websocket

For å motta sanntidsdata hadde Zisson et ønske om å bruke websocket for å håndtere dette. Bruk av websocket var nytt for studentene, og det ble derfor satt av tid for å sette seg inn i hva en websocket er og hvordan det fungerer.

Zisson Interact

Det ble også satt av en del tid til å sette seg inn i Zisson sitt eget brukergrensesnitt, Zisson Interact, samt Zisson sin allerede eksisterende kode. Funksjonene hvor man oppretter nye wallboards skulle implementeres i Zisson sitt brukergrensesnitt for å gjøre det så brukervennlig som mulig for kundene.

RabbitMQ

For å få tilgang til Zisson sine systemer og koder måtte det settes opp en VPN-tilkobling. Dette tok noe tid å få på plass fra oppdragsgiver, og det var noen oppstartsproblemer. I mellomtiden ble studentene bedt om å sette opp en midlertidig simulert MQ i systemet RabbitMQ for å kunne teste med å sende data. Derfor ble det brukt litt tid på å sette seg inn i hva dette var og hvordan få satt det opp.

Verktøy og rammeverk

Prototyping

For prototyping ble det planlagt å bruke Miro. Dette hadde ikke alle kjennskap til fra før og det ble derfor planlagt å bruke litt tid på å sette seg inn i det.

IDE

Visual Studio Code var et verktøy som de fleste ikke hadde tidligere kunnskap i å bruke. I starten ble det derfor planlagt at alle skulle sette seg inn i dette.

Testing

Testverktøyet TestCafé og Karma, og rammeverket Jasmine, var nytt for studentene. Siden dette krevde ekstra tid å sette seg inn i, ble det derfor satt av ekstra tid til å utføre testingen.

3.0 Utviklingsprosessen

Herunder delt inn i de forskjellige fasene.

3.1 Designprosess (design thinking)

Dette kapittelet utdyper hvordan de forskjellige stegene (empatisering, definere problemet, generere ideer, prototyping, og testing) i design thinking er gjennomført.

Kravene som var spesielt viktige å fokusere på under designprosessen var de ikke-funksjonelle kravene:

- Som bruker av systemet ønsker jeg at det er brukervennlig slik at det er effektivt og forståelig å bruke.
- Som bruker ønsker jeg at designet samsvarer med designet i Zisson sitt allerede eksisterende system Zisson Interact, slik at systemet oppleves helhetlig.

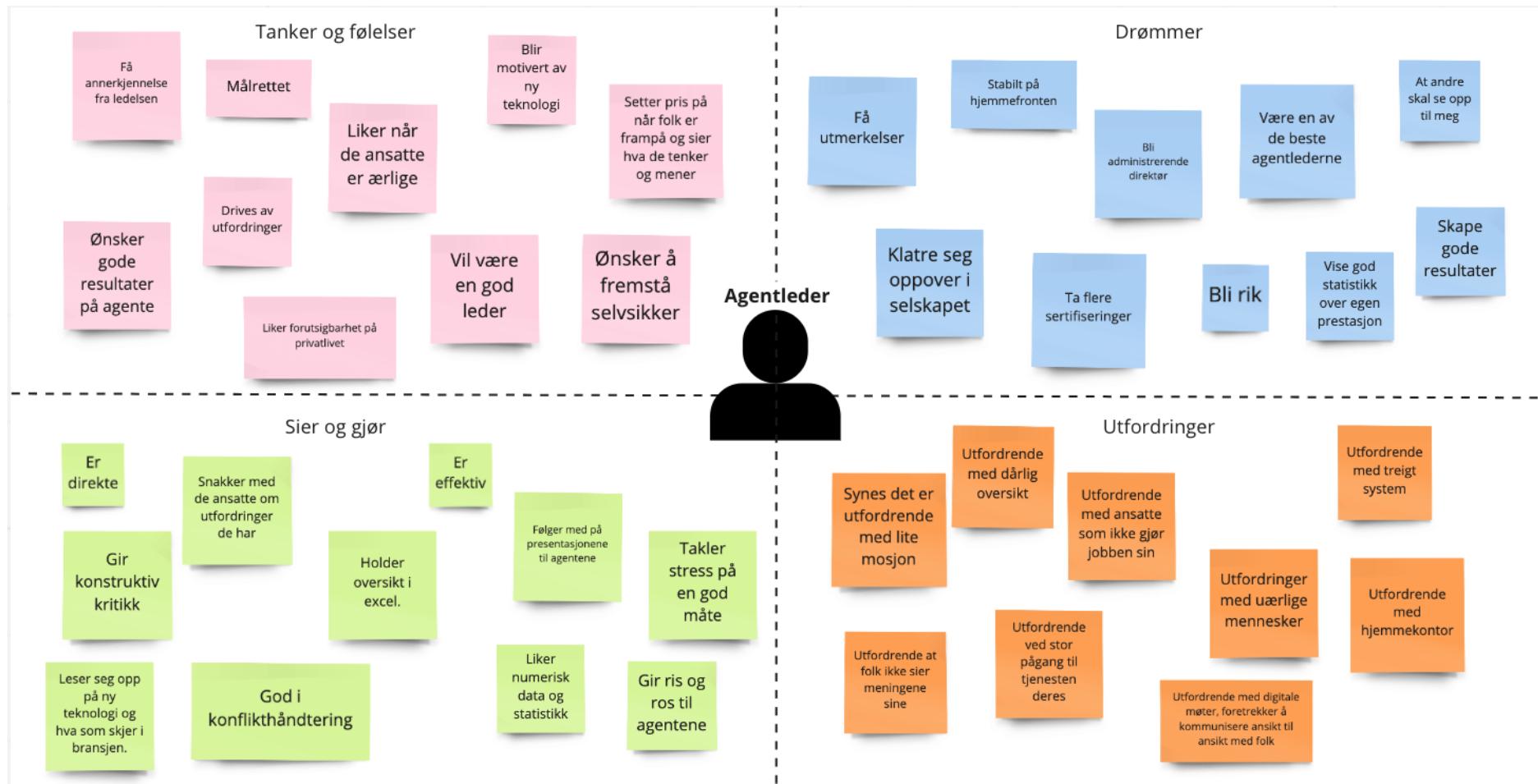
3.1.1 Sette seg inn i brukerens behov / utvikle empati.

Dam beskriver denne delen av design thinking som den delen man skal innhente en empatisk forståelse av problemet som skal løses (Dam og Siang, 2021, Overskrift: 1. Empathize). Dette kan omhandle å involvere eksperter for å finne ut mer om problemområdet til produktet, samt engasjere og empatisere med mennesker for å forstå deres erfaringer og motivasjon (Dam og Siang, 2021, Overskrift: 1. Empathize).

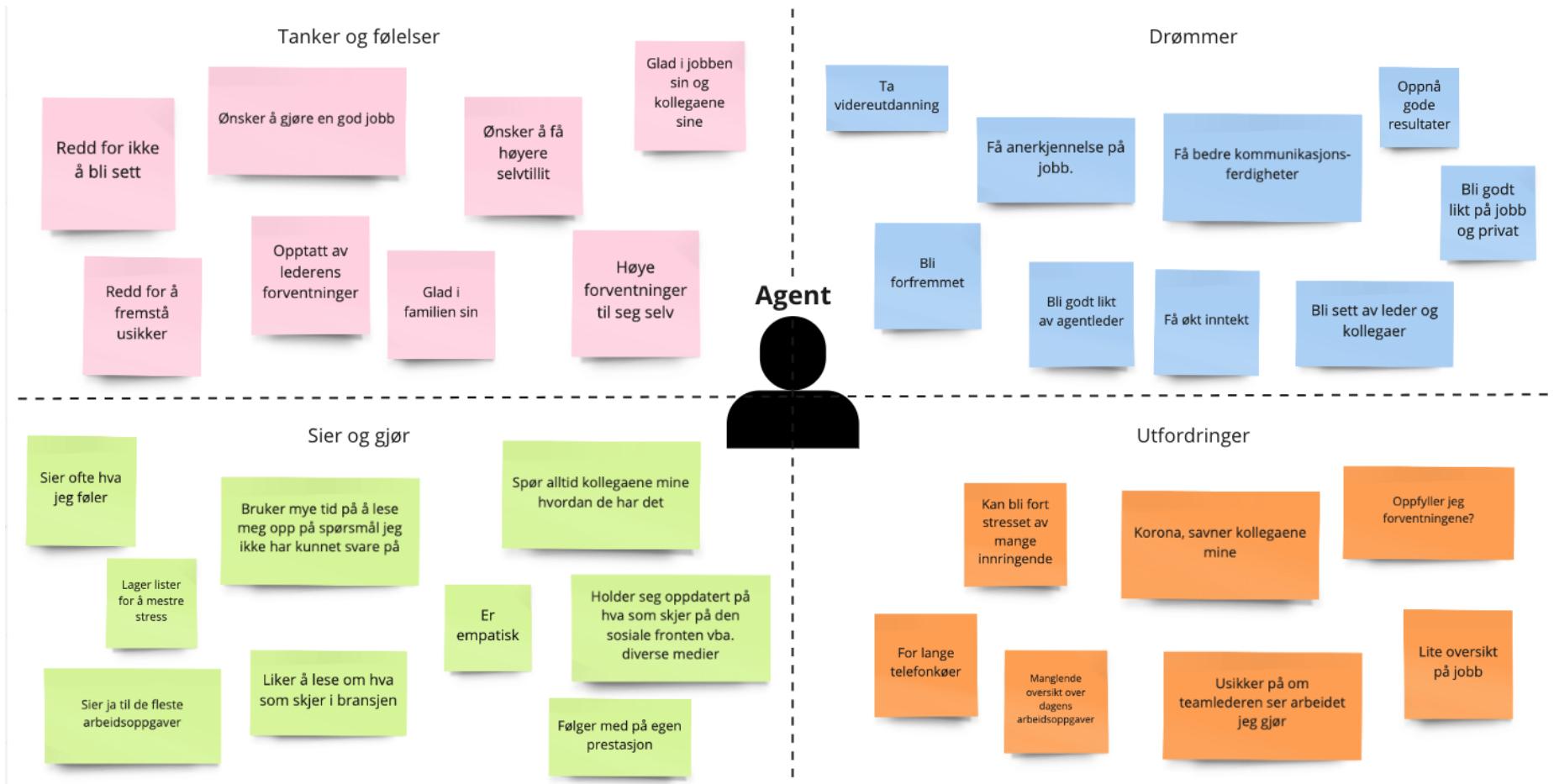
I dette steget var hovedmålet til gruppen å sette seg inn i hva brukerne tenker, hva slags utfordringer de har i sin arbeidshverdag og å se produktet som skal utvikles som en del av et større bilde.

3.1.1.1 Empathy map

Før gruppen gjorde dypere research ble det forsøkt å lage empathy map på en agent og en agentleder, figur 6 og 7. Denne delen av designprosessen gjør det mulig å tenke igjennom hva brukerne tenker, føler, sier og gjør, og samtidig bli bedre kjent med utfordringene brukerne kan ha i hverdagen (McElroy, 2017, s. 23).



Figur 6: Empathy map for agentleder



Figur 7: Empathy map for en agent

[3.1.1.2 Undersøkelser](#)

Først og fremst har prosjektgruppen vært i kontakt med forskjellige nøkkelpersoner hos Zisson som har erfaringer med salg av tjenestene, samt personer fra produktavdelingen. Til disse nøkkelpersonene ble det stilt forskjellige spørsmål om dagens system, herunder hva som er utfordrende med dagens situasjon.

Zisson har også utført undersøkelser før bachelorperioden startet slik at de hadde en god oversikt over utfordringer og ønsker ved systemet som skulle utvikles.

Gruppen tok også kontakt med personer som jobber på kontaktsenteret og spurte de om utfordringer de har i sin arbeidshverdag og informasjon om kundenes utfordringer.

Svar som gruppen fikk var blant annet at det var utfordrende med få personer på jobb og mange innkommende forespørsler. En annen utfordring som ble nevnt var at det var mangel på oversikt både for agenter og agentledere.

[Research - design av Wallboard:](#)

Yeastar skriver at noe av det viktigste å tenke på ved utviklingen av et wallboard er at dataene som skal vises er tidssensitive (Yeastar, 2020). Med dette menes at viktig data må kunne bli målt og visualisert til enhver tid, for å kunne vurdere agentenes arbeid og bedriftenes mål (Yeastar, 2020).

Videre skriver Yaester at noen av de viktigste målene til et wallboard er å skape motivasjon for agentene, gi muligheten for å gjøre endringer i arbeid underveis, se oversikt over kundetilfredshet, og øke produktivitet blant agentene (Yeastar, 2020).

Yaester poengterer også at et callsenter-wallboard ikke bør inneholde for mye tekst eller for mange elementer da dette kan virke forvirrende og uoversiktlig for brukeren (Yaestar, 2020).

3.1.1.3 Personas

Kunnskapen som var innhentet fra undersøkelsene ble brukt til å utforme to personas, figur 8 og 9. Personas er i følge McElroy (2017, s.23) karakterer som er oppfunnet, og blir brukt til å representere forskjellige brukere av et system. Disse er med på å hjelpe designerne til å holde fokus på brukerne videre i designprosessen (McElroy, 2017, 23).

Slik fikk gruppen et bedre innblikk i hvem brukerne av systemet er og hva de kan tenke og føle. De elementene gruppen gikk ut i fra var: Navn, stilling, demografi, personlighetstrekk, mål/drømmer, utfordringer/ frustrasjoner, interesser, medievaner og beskrivelse.

Personas 1



Navn: Lasse Tordenskjold
Stilling: Agentleder hos Telenor

Demografisk

Alder: 36 år
Bosted: Røa, Oslo
Inntekt: 750.000
Utdannelse: Bachelor i HR og personaledelse, MBA
Sivilstatus: Samboer, ett barn på 13 måneder

"Jeg ønsker å effektivisere hjelpesentralen vår til det ytterste, og ønsker en effektiv og oversiktlig måte å holde oversikt over agentene mine på, slik at vi kan analysere hvordan vi kontinuerlig kan forbedre oss."

Personlighetstype

- Mer utadvendt enn innadvendt
- Mer reflektert enn følsom
- Mer funksjonell enn kreativ
- Mer impulsiv enn vurderende

Mål og drømmer

- **Jobb:** Være en av de beste agentlederne, mål å få de beste tallene på sin agentgruppe, liker å utmerke seg med gode tall
- **Karriere:** Ønsker å klatre oppover og bli administrator eller leder for en avdeling/ selskapet. Ønsker en stabil lederstilling med personalansvar da han liker å jobbe med mennesker.
- **Personlige mål:** Ønsker å gå Elbrus og Kilimanjaro

Interesser:

- Friidrett, utendørsaktivitet. Liker å gå fjellturer, tur i skog og mark, stå på ski.
- Sport, friidrett. Ikke avhengig, men liker å oppdatere seg og følge med litt.
- Musikk: hovedsaklig rock. Liker å gå på konserter og festivaler.
- Reiseliv, liker å reise og oppleve forskjellig; alt i fra storbyer, strand og sol, til fjell og natur.

Frustrasjoner/ utfordringer:

- Blir frustrert av ineffektivitet
- Blir frustrert av lite intuitive og abstrakte verktøy og systemer, og liker at ting er enkelt og raskt å bruke, samt vise resultater
- Blir frustrert av folk som ikke gjør jobben sin og kaster bort tid
- Blir frustrert over folk som går sakte på gata
- Blir frustrert over folk som ikke sier hva de mener eller tenker på
- Synes det er utfordrende med abstrakte ting, og liker tydelighet og klarhet
- Synes det er utfordrende å sitte en hel dag inne på sofaen
- Synes det er utfordrende å følge en matoppskrift til punkt og prikke, og utfordrer gjerne retten med egne impulser
- Synes det er utfordrende med hjemmekontor og liker å kunne kommunisere med mennesker enten ansikt-til-ansikt eller en gruppe

Medievaner:

Liker å oppdatere seg på den nyeste teknologien, og kjøper ofte den nyeste telefonen eller datamaskinen på markedet. Når han trener liker han å følge med på applikasjoner på mobilen og smartklokka for å følge med på sin helsedata og prosjeksjon.

- **Søkmotorer:** Google
- **Sosiale medier:** LinkedIn, Twitter, Facebook, Instagram
- **Nettsider:** VG, Aftenposten, Finansavisen
- **Streaming:** NRK, Netflix, TV2 Sumo, YouTube

Beskrivelse:

Lasse er en målrettet og ivrig mann som har ambisjoner og drømmer både karrieremessig og i privatlivet. På jobb ønsker han stort personalansvar og liker å se resultater i form av nummer og statistikk. Han ønsker å være en av de beste i sitt fagfelt da han har som ambisjon å bli toppleder en dag. På fritiden liker han å være aktiv og går ofte lange turer - gjerne på fjellet eller i marken. Han liker seg bedre på en fjelltopp enn på en båt, og tar gjerne med familien eller gutta på tur. Når marken virker litt langt unna tar han seg heller en løpetur i nærområdet i stedet for. Han liker å se resultatet på pulsklokka forbedre seg, eller bestige fjell med et visst antall høydemeter. Han liker å kunne konkurrere med seg selv og kontinuerlig utvikle seg. Når han skal koble av med noe annet enn frisk luft er han glad i å ta seg en øl med kollegaer eller venner og får energi av å være sosial og aktiv, men går gjerne hjem rundt midnatt da han liker å ha energi til å ta seg en liten utfukt dagen derpå. Han er også glad i musikk og dyrker gjerne amerikansk eller klassisk rock og tar seg gjerne en tur på konsert om anledningen byr seg. Om sommeren går han gjerne på festivaler, både på grunn av kultur og musikk, men også på grunn av det sosiale aspektet rundt det.

miro

Figur 8: Personas for Lasse

Personas 2



Navn: Josefine Sørum
Stilling: Kundeservicemedarbeider hos Norwegian

Demografisk

Alder: 32 år
Bosted: Bjerke, Oslo
Inntekt: 450.000
Utdannelse: Bachelor i Service, hotell og ledelse
Sivilstatus: Singel

"Ønsker å bli anerkjent som en god kundeservice-medarbeideren i Norwegian"

Personlighetstype

- Mer utadventt enn innadventt
- Mer følsom enn reflektert
- Mer kreativ enn funksjonell
- Mer vurderende enn impulsiv

Mål og drømmer

- Jobb:** Ønsker å bli anerkjent som en god kundeservice-medarbeideren i Norwegian.
- Karriere:** Hennes 5 års mål er å ta en master innenfor strategisk kommunikasjon. Dette fordi hun elsker å snakke med mennesker og synes utfordringene som inngår i dette fagfeltet er ekstremt spennende.
- Personlige mål:** Stifte familie

Interesser:

- Er et aktivt friluftsmenneske.
- Elsker å gå turer i skog og mark, og gå skitur om vinteren.
- Vært på mange toppturer, bla. Galdhøpiggen.
- Trener crossfit.
- Liker å være sosial og drar gjerne på storbyreiser med venninnene sine.
- Over gjennomsnittet interessert i fotball.

Frustrasjoner/ utfordringer:

- Josefine har noen utfordringer i hverdagen. Hun ønsker veldig gjerne å få seg en familie og er derfor aktiv på datingfronten, men har ikke funnet den rette enda. Noe hun til tider synes kan være tungt.
- Hun synes også det til tider er vanskelig å holde motivasjonen opp på jobb, da hun føler at teamlederen ikke ser all innsatsen hun legger ned i jobben sin. Hun ser selv at hun jobber mye hardere enn og tar jobben mer seriøst enn de andre på teamet og ønsker å bli mer verdsett på dette i det daglige.
- Til tider kan det også være utfordrende at det er lange køer for innringende og få personer på jobb.
- Til tider savner hun også en god oversikt over teamets helhetlige innsats. Josefine tenker at det hadde vært fint å se hvor teamet som helhet skal legge innsatsen for å forbedre resultatet. Dette synes hun er viktig fordi hun ønsker at hele gruppen skal kunne nå bonusmålet og ikke bare at enkeltpersonene klarer det. Hun tenker at dette kunne være motiverende for de ansatte.
- Josefine synes det er utfordrende med coronasituasjonen, da hun synes det har vært vanskelig å motivere seg selv for å jobbe når hun ikke møter kollegaer. I tillegg har det vært vanskelig å vite om teamet som helhet har nådd målene sine eller ikke.

Medieværer:

Bruker mye teknologi. Har en iphone som hun bruker mye, både til å chatte med venner, snakke med familie, lese aviser, dating, samt følge med på sosiale plattformer som instagram og facebook. Har også en linkedin profil hvor hun har mange forbindelser (blant annet til de som jobber hos Norwegian). På fridaten liker Josefine å utvikle ideer i photoshop.

- Søkmotorer:** Google
- Sosiale medier:** Facebook, Instagram, LinkedIn, Tinder
- Nettsider:** Kvinneguiden, VG, Dagsbladet
- Streaming:** Netflix, Viaplay, Discovery +, HBO, NRK, tv2sport

Beskrivelse:

Josefine elsker jobben sin og det faktumet at hun kan få bidra til å hjelpe så mange mennesker i løpet av en dag. Hun ser på jobben sin som svært samfunnsnyttig. Josefine har høye ambisjoner, og ser for seg å kunne ta en master så fort hun har mulighet. Denne utdannelsen ønsker hun å bruke for spisse kommunikasjonsferdighetene sine i rollen som kundeservicemedarbeider. På sikt ønsker hun gjerne å få mer teknisk kompetanse til å kunne jobbe på helpdesk avdelinger.

Josefine er veldig glad i dyr og har både en hund og en katt som hun bruker mye tid på. I tillegg liker hun godt å dra til sydligere strek for å slappe av. Det er også viktig for Josefine å besøke sin eldre bestemor, og bidra med det praktiske der hun kan. I ny og ne drar hun på tinderdate?

miro

Figur 9: Personas for Josefine

3.1.2 Definere problemet

I dette steget var hovedmålet å definere problemet som skal løses med produktet.

Etter undersøkelsene som ble utført i forrige steg ble det utarbeidet disse hovedutfordringene:

Hovedproblemområde for MVP (minimum viable product):

1. Mangler oversikt over agenterars arbeid.
2. Mangler funksjonalitet for filtrering av agentgrupper.
3. Mangler funksjonalitet for oppsett av egendefinerte wallboard.

Disse problemområdene ble brukt som bakgrunn for utviklingen av brukerhistorier som ble utarbeidet i samarbeid med Zisson. Se punkt [C. KRAVSPESIFIKASJON](#) for utdypning.

3.1.3 Utvikling av MVP (Minimum Viable Product)

I samarbeid med Zisson har prosjektgruppen hatt flere møter, både før og under designprosessen, hvor gruppen har utarbeidet forslag til både prototyper og brukerhistorier og fått tilbakemeldinger på disse under hele prosessen. Spesielt under denne prosessen var modellering viktig for å kommunisere ulike visualiseringer og få tilbakemeldinger på om de ulike modellene kunne løse problemet som var definert. Se punkt [3.2 Process View](#) i produktdokumentasjon.

Disse valgene har vært basert på hvilken funksjonalitet som er den viktigste å få på plass først. Dette i henhold til betegnelsen MVP som står for Minimum Viable Product. Med MVP menes et produkt med nok funksjoner til å validere en produktidé tidlig i produktutviklingsfasen (ProductPlan, u.å.). MVP kan også hjelpe

teamet med å motta tilbakemeldinger fra brukere så raskt som mulig for å kunne forbedre produktet raskt (ProductPlan, u.å.).

3.1.4 Generere ideer

Hovedmålet i dette stadiet var å generere så mange ideer som mulig til prototypene som skulle utvikles (Dam og Siang, 2021, Overskrift: 3. Ideate). Videre var det viktig at teamet prøvde å tenke utenfor boksen for å finne nye innovative løsninger på problemet (Dam og Siang, 2021, Overskrift: 3. Ideate).

3.1.4.1 Workshop - Oversikt av wallboard

For idéutvikling og inspirasjon ble det utført workshops både internt og sammen med produktansvarlig i Zisson. Her ble det utført idémyldringer hvor det ble generert ideer til design av prototypene. Her var fokuset å være åpne til nye ideer og komme med innspill. Det ble opprettet en digital tavle i Miro, hvor man kunne skrive ned ideer og innspill som post-it-lapper, vist i figur 10.



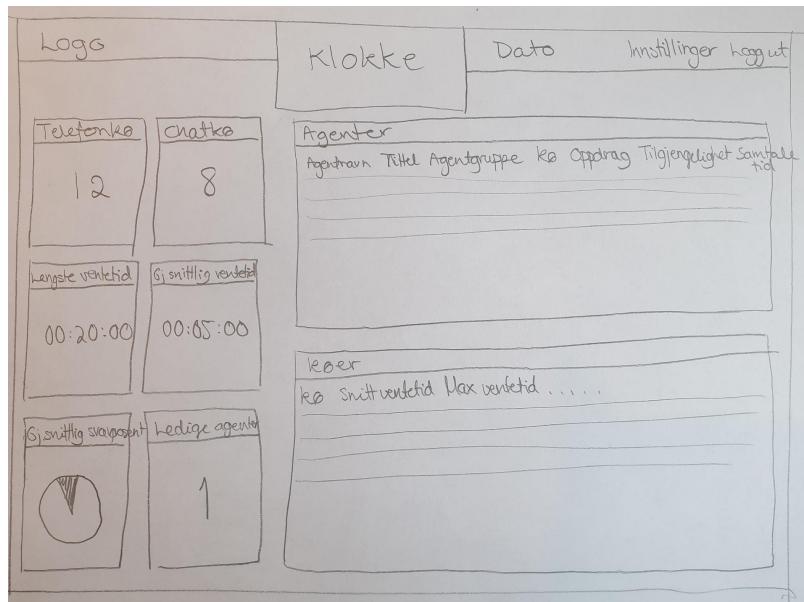
Figur 10: Idemyldrings-kart

3.1.4.2 Skisser - Oversikt av wallboard

Skissering er i følge (UX planet, 2016) viktig for:

- Å få utløp for alle ideene designeren har.
- Få frem stegene i et aktivitetsdiagram/flytdiagram.
- Å utforske forskjellige ideer for layout.
- Å vise grunnleggende struktur i et system.

Figur 11 viser en skisse for den prototypen det ble valgt å arbeide videre med.



Figur 11: Håndtegnet skisse av prototype

3.1.5 Prototyper og testing av prototyper

Målet med å lage prototyper er å forstå, kommunisere, teste, og forbedre produktet på en billig og rask måte (Mc Elroy, 2017, s.17).

Gjennom å gjøre tidlige problemoppdagelser, finner man brukerens utfordringer tidlig i utviklingen og det kan i et reelt prosjekt spare bedriften for både tid og penger (Mc Elroy, 2017, s.17). Gjennom skisseringer og tegninger kan man også finne flere forskjellige måter å løse utfordringene på (Mc Elroy, 2017, s.18). Videre

gir prototypeprosessen mulighet til å forstå målet til brukeren og selskapets visjon (Mc Elroy, 2017, s.20). Prototyper er videre viktige for å kommunisere ideene til bedriften og få tilbakemeldinger på hva som fungerer og hva som ikke fungerer (Mc Elroy, 2017, s. 26).

Ved testing av prototyper var målet å få brukere til å prøve produktet og gi tilbakemeldinger på hva som fungerte mindre bra (Mc Elroy, 2017, s.31). Dette står i kontrast til å bruke egen magefølelse til å tenke at noe fungerer eller ikke fungerer ved produktet. Det er anbefalt å teste på fire til åtte brukere for å få et så mangfoldig publikum som mulig og for å finne de mest vanlige mønstrene som oppdages hos brukerne (Mc Elroy, 2017, s.254).

Prosjektgruppen valgte å lage prototyper med low fidelity funksjonalitet da det var ønskelig å komme raskt i gang med programmeringen og det videre arbeidet med systemet. Med low fidelity menes en prototype som er billig, og rask å lage med materialer som er lett tilgjengelige (Mc Elroy, 2017, s.51). Low fidelity prototyper brukes i de sammenhengene hvor man ønsker å teste funksjonalitet raskt (Mc Elroy, 2017, s.51).

Det ble utviklet seks prototyper, hvor det ved hver av prototypene ble utført forskjellige brukerrelaterte tester, se punkt [6.1 Ikke-funksjonelle tester](#) for utdypning av hvordan disse ble utført og resultater og endringer fra disse i punkt [8.0 Resultater.](#)

[3.1.5.1 Workshop - utvikling av prototyper](#)

På workshopen startet gruppen med å plassere ulike elementer rundt på wallboardet. Slik jobbet gruppen seg videre ut i fra hva som så best ut i fra de designspesifikasjonene gruppen hadde kommet frem til i den forrige workshopen. I

dette steget var målet å lage så mange prototyper som mulig for å finne en god måte å løse utfordringene på.

Etter at gruppen hadde utviklet flere ulike forslag, ble det valgt ut en prototype man skulle ta utgangspunkt i for videre utvikling.

Med bakgrunn i MVP hvor rammene til prosjektet var å fokusere på agentoversikt først, valgte gruppen å videreutvikle prototypene med fokus på et agentvindu i wallboardet. Se punkt [8.1.1 Intervju/samtale om første prototype](#) for bilder av prototypen.

Det ble valgt å jobbe videre med den utvalgte prototypen fordi:

1. Den oppfylte ønskede kvalifikasjoner fra designønskene, med blant annet oversiktighet, tydelig definerte komponenter, og tabelloversikt.
2. Den gav også god synlighet og kontrast med fargevalget i radene på agentene.

Etter innspill og kommentarer fra Kim Josefson (product manager hos Zisson) jobbet prosjektgruppen videre med prototypen, hvor det var viktig å ha fokus på kontraster og fargevalg, samt å få på plass alle attributtene i tabellen.

[3.1.5.3 Refleksjon av prototypeprosessen](#)

Det er gjennomført brukertester på flere nøkkelpersoner hos Zisson med ulike roller, og som har forskjellige ønsker og prioriteringer. Dette resulterte i at testpersonene også hadde veldig ulike meninger og oppfatninger på hvordan systemet skulle og burde se ut. For å få et klarest mulig bilde over hvordan Zisson faktisk ønsket det ble det valgt å sette opp mange prototyper på kort tid. På denne måten kunne Zisson raskt bekrefte eller avkrefte forslagene som ble laget. Samtidig hjalp dette også

Zisson å få visualisert hvordan de ønsket at det skulle se ut, da dette ikke var noe de hadde et klart bilde av selv fra starten av.

På grunnlag av uklarheter i ønsker fra oppdragsgiver ble det også tvetydig hvordan oppsettet og innstillingene av wallboardet skulle løses. I de tidligste prototypene var dette implementert på selve wallboardet, men det kom senere frem at det var ønskelig at dette skulle implementeres i Zisson Interact. Derfor er noen av tilbakemeldingene fra testene og den heuristiske evalueringen ikke aktuelle for det endelige resultatet.

Grunnen til at innstillinger skulle implementeres i Zisson Interact, var for å bidra til at kundene til Zisson enkelt skulle kunne gjenkjenne designet på et brukergrensesnitt som de var kjent med fra tidligere. Derfor ble det valgt å bruke det samme designet på overskrifter og oppsett som Zisson har fra før.

Det ble også kartlagt at funksjonalitet for logo skulle være en del av MVP fordi det var viktig for kundene å få satt sitt eget preg på wallboardet som skulle presenteres i deres lokaler. Dette er blant annet viktig fordi wallboardet skulle være i miljøet til kundene til Zisson, og bidra til at de får en følelse av eierskap til wallboardet.

På grunn av noen utfordringer under utviklingen av diagramform, ble det valgt å gjøre endringer med dette kravet etter sprint review for sprint 5. Hvor det var ønskelig å gjøre om til bokser med oppstelling av agentene i statusene. Det ble også forandringer i kravene til MVP slik at to av kolonnene i agenttabellen ble fjernet.

Under prototypeprosessen har det vært eksperimentert med forskjellige farger og nyanser av disse, samt størrelse og tykkelse på tekst, og ulike fonter.

Innstillingermenyen har også blitt bearbeidet, før det ble valgt å gå bort i fra denne og heller implementere den i Zisson Interact.

3.2 Scrum

Å arbeide med Scrum som arbeidsmetodikk har vært en lærerik prosess.

Arbeidsmetodikken var noe studentene hadde lært om i teorien, men aldri tatt i bruk i praksis. Derfor ga det studentene et stort utbytte å få prøvd det ut i praksis.

I planleggingsfasen ble Scrum med sine sprinter planlagt ganske rigid for eksempel med tanke på når møter skulle avholdes eller hvor lenge en sprint skulle være. Det ble fort en erfaring med at dette ikke var like lett å overholde når sprintene var i gang.

Møtene ble ikke alltid holdt siste dag av sprinten da oppdragsgiver ikke alltid hadde kapasitet, og etterhvert ble det mest effektive å avholde et sprint review- og et sprint planleggingsmøte på samme dag. Dette gjorde også at noen av sprintene økte med et par dager.

Møtene var allikevel en sentral del planleggingen og å holde kontroll på arbeidet. De ble avholdt med oppdragsgiver som ga gruppen råd og ønsker om hvilke inkrementer som skulle implementeres, og for hver sprint review ble det gjort en akseptansetest for å sikre at inkrementene som ble implementert ble godkjent.

Den første sprinten ble satt av til å jobbe med design av systemet. Da sprinten var over var det planlagt at det kun skulle programmeres de neste sprintene. Men etter endringer i krav ble det ofte behov for å gå tilbake til designdelen av utviklingen og

gjøre endringer der, og derfor ble det ikke så tydelig oppstykket som det egentlig var planlagt.

Andre sprint der gruppen startet med selve programmeringen, gikk litt tregere enn ønsket. Det var mye teknisk å sette seg inn i for gruppa, og alle hadde et ønske om å henge med fra start. Derfor jobbet gruppen sammen om de grunnleggende funksjonelle kravene som ble implementert i systemet. Dette ga alle en god grunnforståelse, men var derimot lite effektivt.

Etter sprint retrospective-møte ble det en enighet om at arbeidsoppgavene fra da av måtte fordeles. Dette gjorde at neste sprint opplevdes mye mer effektiv og håndterlig, og gruppen kjente på mestring av å ha fått til mye arbeid.

For hver sprint som gikk ble gruppen mer og mer komfortabel med Scrum som en arbeidsmetodikk. Alle utfordringer ble tatt opp ved sprint retrospective-møter, og forslag til hvordan de kunne forbedres ble tatt med til neste sprint.

Daily standups sørget for at gruppen hele tiden hadde dialog. Dette var nyttig både for å sikre et godt teamarbeid der man alltid sjekket inn og visste hvordan det sto til med hverandre. Det sikret også effektivitet i prosjektarbeidet da man alltid visste hva hverandre drev med, og om noen trengte hjelp med utfordringer.

Scrumboardet som ble opprettet i Jira ble brukt hver dag for å sikre at riktige krav ble implementert og for å ha kontroll på fremgangen. Det som ofte var krevende var tidsestimeringen. Å implementere et inkrement handlet ikke bare om det ene kravet som skulle implementeres. Det kunne fort avhenge av at en rekke andre faktorer måtte på plass, og ofte dukket det opp nye utfordringer som gjorde at implementasjonen tok enda lengre tid. Det ble diskutert hvorvidt man skulle dele

opp enkelte krav i mindre brukerhistorier for å ikke måtte dra kravene utover flere sprinter. Det ble besluttet at dette ikke var nødvendig, og at det var oversiktlig nok å videreføre kravene til neste sprint.

Utdypende informasjon om hver sprint med dokumentasjon av implementasjon av inkrementer og møtesammendrag er lagt som vedlegg under [punkt 1.0, Dokumentasjon av Scrum](#).

3.3 Prosjektdagbok

Det er skrevet en prosjektdagbok for hver arbeidsdag i prosjektperioden. Denne ble brukt til å loggføre hva man gjorde, samt hva ukens eller dagens agenda var. På denne måten var det enkelt å holde oversikt over møter, hva andre hadde fått gjort, hva som var igjen å gjøre, og eventuelt hva som ikke ble rukket å gjøre ferdig den dagen. En slik loggføring gjorde det også enklere å holde oversikt om et av medlemmene ikke hadde anledning til å delta på stand up-møte eller var syk en dag. Prosjektdagboken er lagt som vedlegg under [punkt 2.0, Prosjektdagbok](#).

4.0 Oppsummering

Alt i alt har prosessen fra planlegging til ferdig produkt fungert godt.

Planleggingsfasen la et godt grunnlag for videre arbeid med prosjektet, da studentene hadde en godt utarbeidet arbeids- og fremdriftsplan, som gjorde arbeidsprosessen strukturert og målbar.

Risikovurderingen som ble gjort under planleggingen innebar at forebyggende tiltak ble satt inn. Dette sikret studentene da risikoer som teknologiske utfordringer, endringer i krav, og ny teknologi oppstod. Risikoene utgjorde derfor ingen skade på

det ferdige produktet, da de forebyggende tiltakene ga rom for å at ting kunne ta ekstra tid, eller at man måtte oppsøke hjelp ved tekniske problemer.

De planlagte arbeidsmetodikkene med design thinking, modellering, utarbeidelse av kravspesifikasjon, og Scrum ble fulgt hele veien og hjalp til med å kvalitetssikre arbeidsprosessen og produktet, skape et godt teamarbeid, la grunnlaget for god dokumentasjon og et ferdig sluttprodukt til normert tid.



OSLO METROPOLITAN UNIVERSITY

STORBYUNIVERSITET

Kravspesifikasjon

Wallboard for Zisson

Bachelorprosjekt 2020 / 2021

Gruppe 32

Martine Holmberg s236808
Camilla H. Carlsen s197235
Synne Angell s331153
Nikola Sekiw s333975

C. KRAVSPESIFIKASJON

1.0 Innledning

Hensikten med kravspesifikasjonen er å sikre at alle parter er inneforstått med kravene for prosjektet, og for å evaluere og akseptere sluttproduktet. Den har også blitt brukt som et verktøy i prosjektet ved å være utgangspunktet for arbeidet som skulle gjøres.

2.0 Kravspesifikasjonsprosessen

Kravspesifikasjonen var noe av det første arbeidet som ble gjort i prosjektarbeidet. Den ble utarbeidet i møter med Zisson slik at alle var enige om kravene for prosjektet, og det dannet et grunnlag for videre arbeid.

Alle funksjonelle krav ble delt opp i epics. En epic er en slags overordnet brukerhistorie. Dette ble gjort for å gjøre kravspesifikasjonen mer ryddig og oversiktlig, da det gir et raskt overblikk over hvilke brukerhistorier som henger sammen. For hver epic ble det knyttet opp flere brukerhistorier.

2.1 Endringer i kravspesifikasjonen

Kravspesifikasjonen har endret seg en del underveis. Dette skyldes blant annet utydelighet i kravspesifikasjonsarbeidet i starten av prosjektet, og det faktumet at nye krav eller utfordringer dukket opp underveis i prosessen.

Eksempelvis ble det endringer i hvordan konfigurasjonen av et wallboard skulle foregå. I utgangspunktet var det tenkt at det skulle være i et eget isolert system sammen med wallboardet. Senere kom det et ønsket fra oppdragsgiver om at dette

skulle implementeres i deres eksisterende brukergrensesnitt, Zisson Interact. Dette førte til endringer i kravspesifikasjonen.

Oppdragsgiver måtte underveis i prosjektarbeidet gjøre en del endringer i sine prosjekter og det ble noe frem og tilbake på hva som ble det riktige å gjøre ut i fra deres eksisterende systemer. Dette resulterte igjen i at prosjektgruppen brukte lengre tid enn planlagt på noen av kravene, da det måtte gjøres en del endringer på allerede implementerte krav.

I tillegg kom det etterhvert frem at det var behov for flere tekniske brukerhistorier enn det til å begynne med var tiltenkt.

2.2 Kravspesifikasjonens betydning

Kravspesifikasjonen har hele veien vært grunnlaget for arbeidet som har blitt gjort i prosjektet, i alt fra design til implementering. Derfor har kravspesifikasjonen hatt en meget sentral rolle i dette arbeidet.

I designarbeidet ble skisser og prototyper utarbeidet etter kravene som var stilt i kravspesifikasjonen. De ikke-funksjonelle kravene ble implementert i designprosessen. Prototypene ble akseptansetestet og på denne måten ble det klarere for Zisson hva de egentlig ønsket og hvilke krav de ville sette for oppgaven. Derfor ble det i denne fasen også mer tydelighet rundt kravspesifikasjonen og funksjonalitet.

I Scrum-prosessen var kravspesifikasjonen hele grunnlaget for hvordan arbeidet foregikk. Målet var at alle de funksjonelle kravene skulle bli implementert gjennom de sprintene man hadde til rådighet, og derfor ble alle arbeidsoppgaver tatt ut i fra kravspesifikasjonen.

2.3 Samsvar med produkt

Systemet som er utviklet samsvarer med den endelige kravspesifikasjonen.

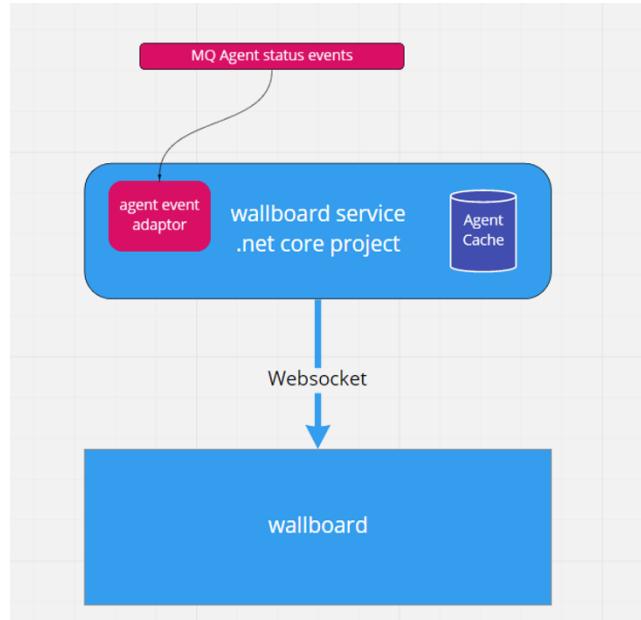
Kravspesifikasjonen har hele tiden vært grunnlaget for utviklingen av nye funksjoner og har derfor vært utgangspunktet for utviklingen av systemet. Gjennom jevnlig møter og akseptansetester av alle krav som har blitt implementert, har det blitt sikret at systemet samsvarer med kravspesifikasjonen.

3.0 Endelig kravspesifikasjon

3.1 Krav til systemkonstruksjon

Det var et krav om at Wallboard backend-tjenesten (heretter kalt WallboardService) skulle bruke programmeringsspåret C# med .NET Core versjon 3.1 eller nyere. Zisson foretrakk at wallboardet ble skrevet i TypeScript med et kjent rammeverk som Angular. Angular ble foretrukket siden Zisson hadde kompetanse innen dette feltet, men dersom gruppen ønsket et annet rammeverk var Zisson åpne for diskusjon.

Illustrasjonen i figur 12 viser en logisk modell av systemet. Figuren ble utlevert i oppgavebeskrivelsen fra oppdragsgiver og var krav for hvordan delsystemene skulle kommunisere. WallboardService har som krav å måtte lytte på MQ etter Agentstatus-eventer, og kommunisere med wallboardet sin klientdel gjennom en websocket-tilkobling.



Figur 12: Logisk modell av systemet

3.2 Funksjonelle krav

Epic: Vise agentoversikt

- Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.
- Som agentleder ønsker jeg et wallboard som viser frem agentene i en tabell for å ha en oversikt over agentene som er på jobb.
- Som agentleder ønsker jeg å se agentnavn for å skille mellom agentene.
- Som agentleder ønsker jeg å se profil for å vite hvilken profil agentene tilhører.
- Som agentleder ønsker jeg å se agentgruppe for å vite hvilken agentgruppe agentene tilhører.
- Som agentleder ønsker jeg å se agentstatus for å vite hva agentene foretar seg.
- Som agentleder ønsker jeg å se tid i agentstatus for å vite hvor lenge agenten har vært i en status.

- Som agentleder ønsker jeg å se callstatus for å vite hvor lenge hver agent er i hver telefonsamtale.
- Som agentleder ønsker jeg å se callstatus for å vite hvilket telefonnummer en agent er i samtale med slik at jeg har oversikt.
- Som agentleder ønsker jeg å se en overordnet oversikt over statusene til agentene i diagramform, slik at jeg har bedre oversikt ved mange agenter.
- Som agentleder ønsker jeg at wallboardet fremstår responsivt på store skjermer, slik at wallboardet ser bra ut når vises frem på storskjerm.

Epic: Teknisk

- Som agentleder ønsker jeg at wallboardet gjør et nytt forsøk på tilkobling til websocket dersom tilkobling mislykkes.
- Som agentleder ønsker jeg at agentene lagres til cache, slik at jeg ved tilkobling får inn de siste dataene som har vært registrert.
- Som agentleder ønsker jeg at cache håndterer ulike wallboards ut i fra en wallboard nøkkel, slik at jeg kun får dataene som tilhører et gitt wallboard.
- Som agentleder ønsker jeg å kunne oppdatere cache med nye agenter hvis det har kommet en endring på en wallboardkonfigurasjon som blir hentet fra databasen.

Epic: Konfigurasjon av wallboard

- Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.
- Som administrator ønsker jeg å distribuere wallboardet via en unik URL-lenke til den som skal overvåke den, slik at de jeg sender den til får tilgang til det spesialtilpasset wallboard jeg opprettet.
- Som administrator ønsker jeg å legge til en logo til wallboardet fordi det er en del av selskapets profil.

- Som bruker ønsker jeg å sortere på de ulike kolonnene for å kunne tilpasse sorteringen ut i fra hva som er relevant.
- Som administrator ønsker jeg å kunne filtrere på agentgruppe for å vise ønskede agenter på wallboardet.
- Som administrator ønsker jeg å se ikoner for det nye og det gamle wallboardet i oversikten over wallboards for å kunne skille mellom de ulike typene.
- Som administrator ønsker jeg å kunne klone det nye wallboardet for å raskt kunne opprette et nytt wallboard av samme type.
- Som administrator ønsker jeg å kunne endre på et wallboard og da få inn de verdiene som tidligere var skrevet inn.
- Som administrator ønsker jeg å kunne endre konfigurasjonen til et wallboard slik at jeg ikke må opprette et nytt dersom endringer er nødvendig.
- Som administrator ønsker jeg å ha en knapp som sier noe om jeg skal opprette et wallboard med den nye eller den gamle konfigurasjonen.
- Som administrator ønsker jeg under oppretting/oppdatering av wallboard å kun se felt som er relevante dersom det er valgt diagramform.

3.3 Ikke-funksjonelle krav

- Som bruker av systemet ønsker jeg at det er brukervennlig slik at det er effektivt og forståelig å bruke.
- Som bruker ønsker jeg at designet samsvarer med designet i Zisson sitt allerede eksisterende system Zisson Interact, slik at systemet oppleves helhetlig.



OSLO
OMET

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITET

Produktdokumentasjon

Wallboard for Zisson

Bachelorprosjekt 2020 / 2021

Gruppe 32

Martine Holmberg s236808
Camilla H. Carlsen s197235
Synne Angell s331153
Nikola Sekiw s333975

D. PRODUKTDOKUMENTASJON

1.0 Innledning

Produktdokumentasjonen utdyper hvordan programmet er bygget opp og gir en presentasjon av hvordan sluttproduktet ser ut og fungerer.

Produktdokumentasjonen består av en beskrivelse av programmet, en presentasjon av løsningen gitt gjennom “4+1 Architectural View”, sentrale datastrukturer og løsninger som er brukt i prosjektet og en forklaring på systemets andre forhold. Til slutt presenteres den ferdige løsningen med en gjennomgang av brukergrensesnittet.

2.0 Beskrivelse av programmet

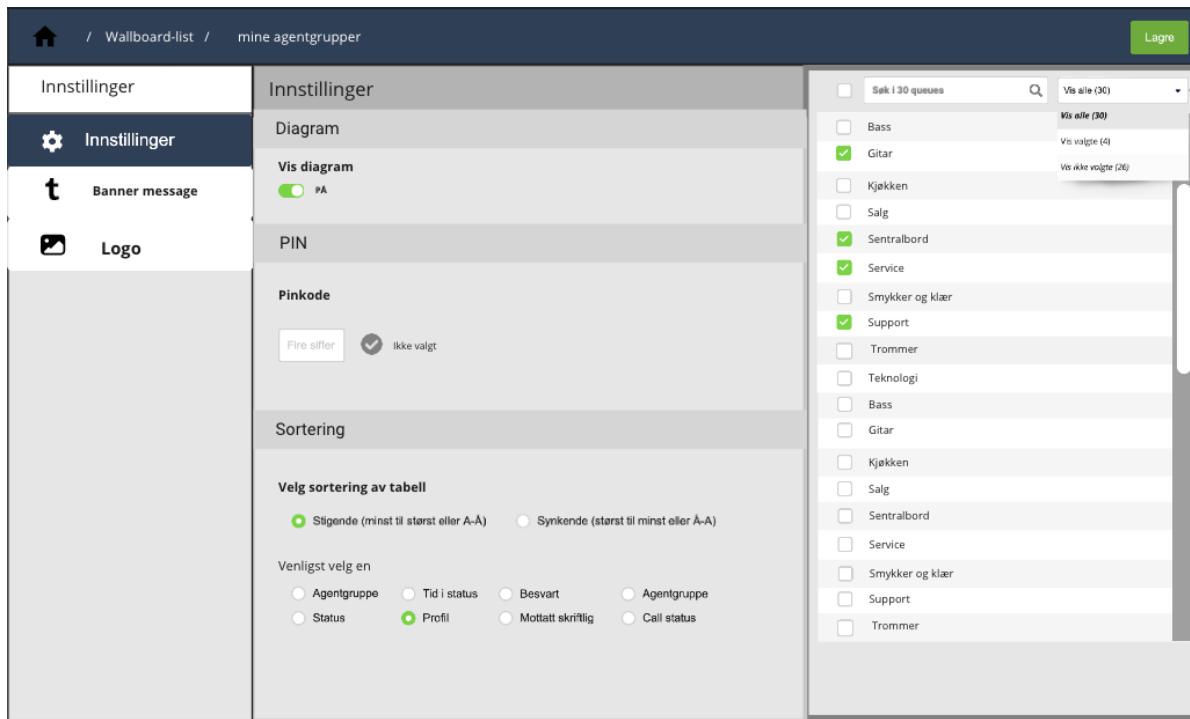
Hensikten med programmet er å vise en oversikt over agenter, samt informasjon om deres arbeid på en eller flere skjermer på bedriftens avdeling. Oversikten skal vise agentens navn, hvilken status agenten har, tid i statusen, callstatus og hvilken agentgruppe agenten tilhører. Wallboardet skal vise informasjon i sanntid, slik at hvis det eksempelvis ringer inn en kunde til en agent, så skal det vises på wallboardet hvilket telefonnummer det ringes fra, samt kontinuerlig tid i samtaLEN. Agenten kan selv endre status, som f.eks. pause eller after call work, i den eksisterende callcenter løsningen Zisson Interact. Dersom agenten endrer status skal dette vises i sanntid på wallboardet.

Det skal være mulig for en administrator å kunne sette opp et wallboard i Zisson Interact, slik at et tilpasset wallboard rettet mot den avdelingen som skal bruke det er tilgjengelig der. Oppsettet som kan velges er: valg av agentgrupper, hvilken

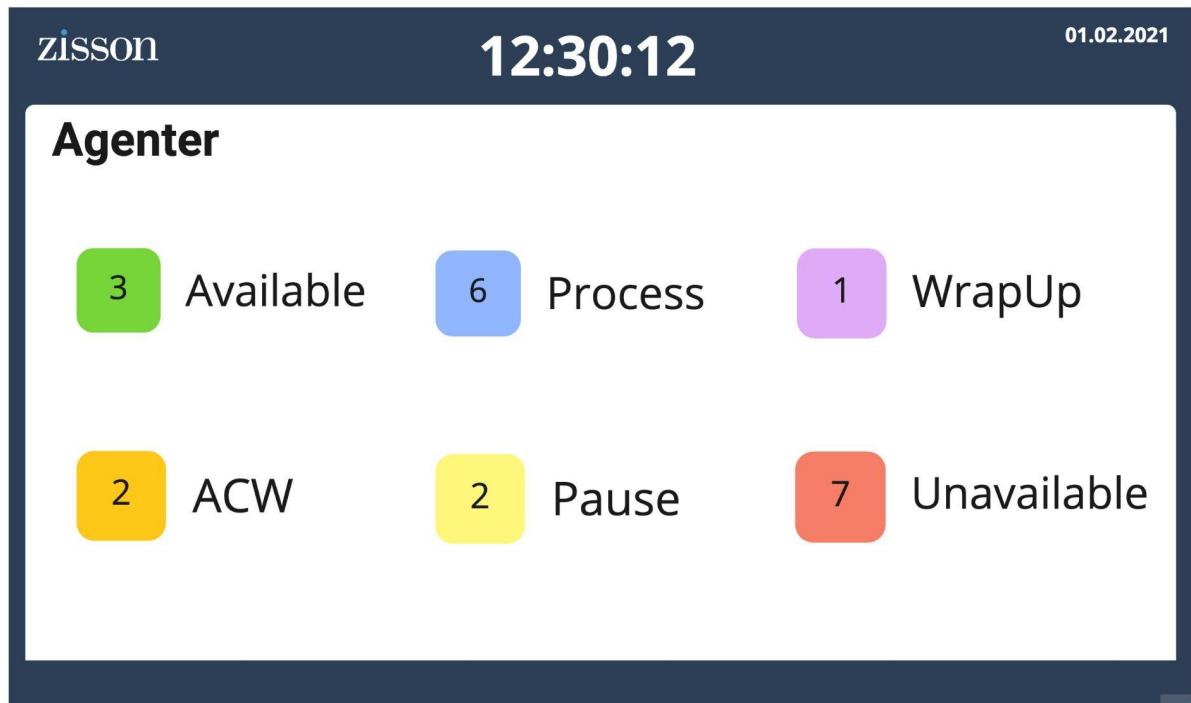
kolonne det skal sortert på, valg av sorteringsretning stigende eller synkende, og om informasjonen skal vises som diagram eller ikke. Når dette er valgt kan wallboardet lagres, og vil da dukke opp i en liste med wallboard under wallboard-liste i Zisson Interact. I denne listen skal det også være markert med et ikon om wallboardet er av den nye typen eller den gamle. Herfra skal det også være mulig å åpne wallboardet som ble opprettet med en unik URL-lenke. På wallboardet skal det deretter vises informasjon som administrator har valgt.

2.1 Siste prototype

Figur 13, 14 og 15 viser de endelige prototypene.



Figur 13: Wallboard-innstillinger i Zisson Interact



Figur 14: Agentvindu i diagram-form

Agentnavn	Status ↑	Tid i status	Callstatus	Profil	Agentgruppe
Christian Strand	LEDIG	00:04:10		Vintersport	Ski
Lise Johansen	LEDIG	00:04:10		Vintersport	Skiskyting
Bent Stiansen	LEDIG	00:04:10		Vintersport	Vintersport
Thomas Alsgaard	BEHANDLER	00:04:10		Vintersport	Ski
Jenny Skavland	BEHANDLER	00:04:10		TV	TV
Espen Lind	BEHANDLER	00:04:10		Sentralbord	Sentralbord
Kurt Nilsen	WRAPUP	00:04:10		Musikk	Gitar
Lise Johansen	ACW	00:04:10		Musikk	Gitar
Kristian Eriksen	ACW	00:04:10		Musikk	Piano
Stian Staysman	PAUSE	00:04:10		Musikk	Sanger
Tom Eriksen	OPPTATT	00:04:10	00:01:10 +4792338233	Musikk	Gitar
Solveig Kloppen	OPPTATT	00:04:10	00:01:10 +4792338233	Musikk	Gitar
Maria Mena	OPPTATT	00:04:10	00:01:10 +4792338233	Kjøkken	Kjøkken
Erlend Haaland	OPPTATT	00:04:10	00:01:10 +4792338233	Fotball	Fotball

Figur 15: Nytt agentvindu der to kolonner er fjernet fra forrige forslag

3.0 4+1 Architectural view

Det er valgt å bruke architectural view for å vise forskjellige vinklinger fra forskjellige sider ved utviklingen: Scenario view, process view, logical view, implementation view og physical view. Architectural view dreier seg ifølge Phillippe Kruchten om å samle alle beskrivelsene og valgene som er gjort av et system rundt disse fem vinklingene (Kruchten, 1995, s. 2). Hvor process view, logical view, implementation/ development view og physical view hovedsakelig er utformet ut i fra scenario view (Kruchten, 1995, s. 2). Dette vises i figur 16 Architectural view.

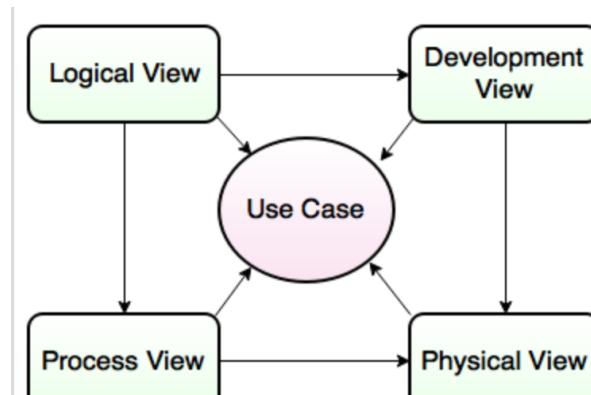


Fig. 4 + 1 View Architecture

Figur 16: Architectural view (CarreerRide, u.å.)

3.1 Scenario View

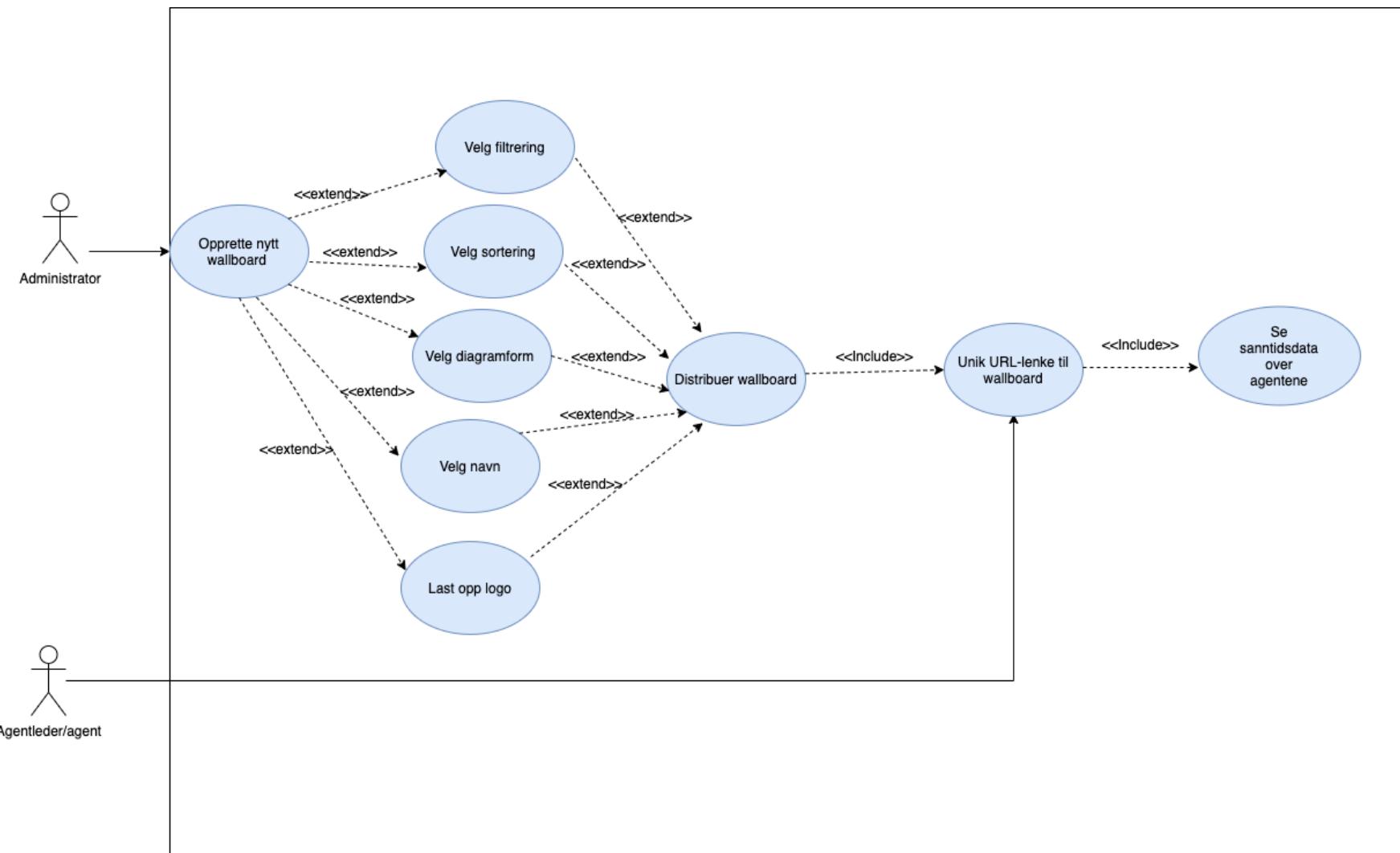
Scenario view innebærer en forenkling av brukerhistoriene og er ofte modellert ved use case diagrammer (Drolsjammer og Olasveengen, 2016). Denne vinklingen er ment å gi utfyllende informasjon til de andre vinklingene (Drolsjammer og Olasveengen, 2016).

Ut i fra brukerhistoriene som ble utarbeidet (se punkt [3.2 Funksjonelle krav](#) for utfyllende brukerhistorier) ble det modellert et use case diagram, figur 17. Et use case diagram brukes for å vise systemets funksjonalitet og samspillet mellom

systemet og omgivelsene. Man får en helhetlig oversikt over aktørene i systemet og deres mål (Sommerville, 2016, s.145).

Ikke alle brukerhistoriene ble valgt ut, da mange av brukerhistoriene kun er basert på hva som skal vises frem i wallboardet. Det ble derfor kun valgt de mest overordnede som går på de interaktive delene av systemet for å danne et bilde av hovedfunksjonaliteten i systemet.

Primæraktørene i diagrammet er agent og agentleder, som vil si at aktørene har egne mål. Uttrykkene “includes” og “extends” er også brukt i diagrammet. Includes vil si at to eller flere use cases kan ha noen like steg. Denne kan da legges ut i eget use case som disse use casene kan inkludere. Extends derimot, beskriver hvordan man kan oppnå tilleggsresultat (tilleggsoppførsel) som utføres under gitte omstendigheter, da utvides funksjonaliteten (Nishadha, 2020).



Figur 17: Use case-diagram

I diagrammet har aktøren administrator som mål å opprette et nytt wallboard. Dette starter med ”Opprett nytt wallboard” som igjen kan føre til de ulike valgene med filtrering, logo, diagramform og valg av sortering. Disse vil igjen føre til at man kan distribuere wallboardet, som vil si at man får et ”Unik URL-lenke til wallboard”. Dette resulterer da i at primæraktørene administrator, agentleder og agenter kan aksessere URL-løkken og se sanntidsdata over agentene.

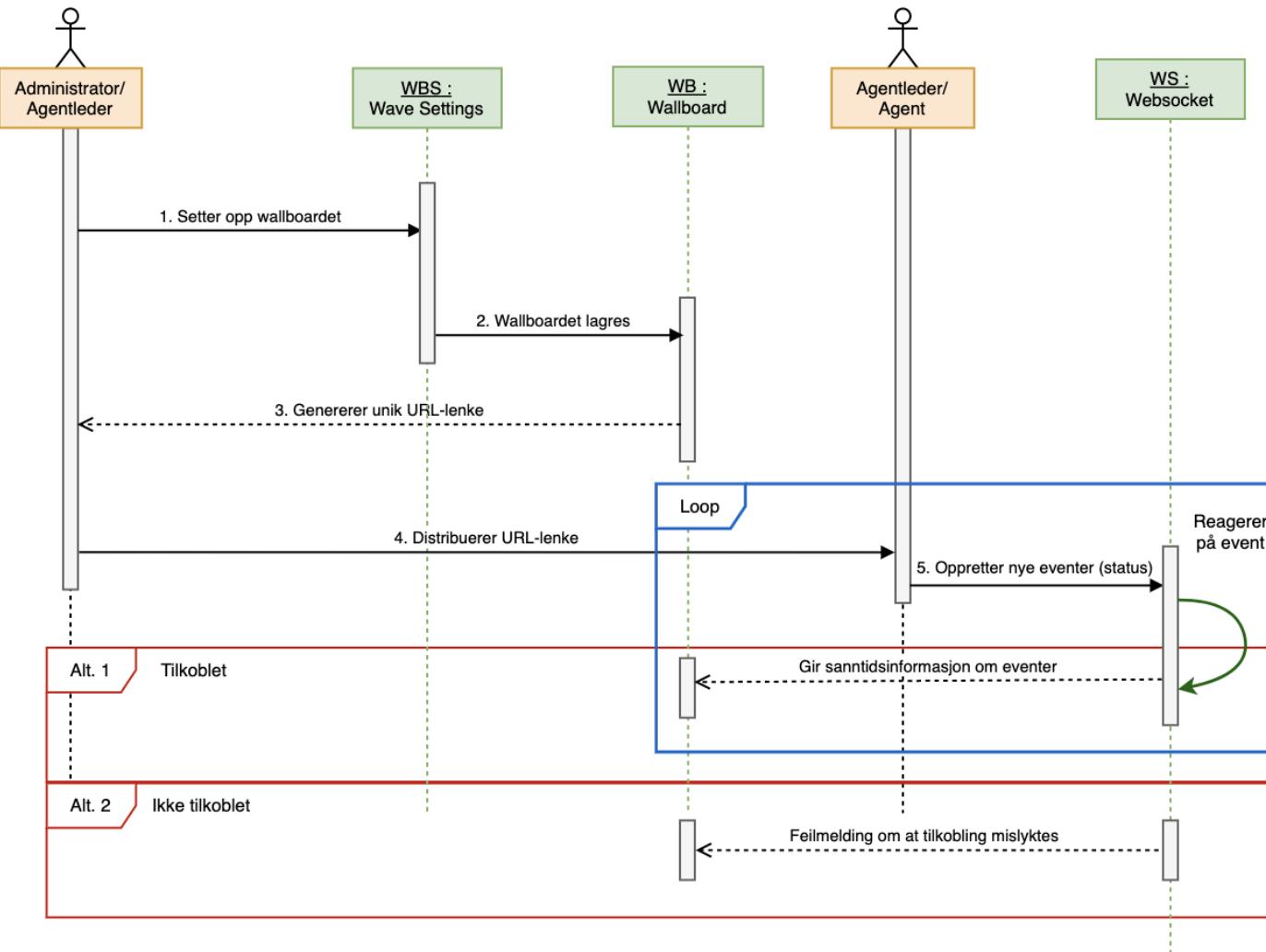
3.2 Process View

Process view blir brukt til å forklare hvordan de ulike delene i systemet kommuniserer med hverandre (Kontio, 2005, s. 4). I følge Drolsjammer og Olasveengen er det vanlig å bruke sekvensdiagram og aktivitetsdiagram ved prosess view (Drolsjammer og Olasveengen, 2016). Disse modellene er brukt og baserer seg på brukerhistoriene som ble utviklet under kravspesifikasjonen.

Ved modellering bruker man brukerhistoriene til å visualisere systemet på ulike nivåer. Modellering brukes for å danne en oversikt og en felles forståelse av systemet man skal utvikle (Sommerville, 2016, s.139).

3.2.1 Sekvensdiagram

Et sekvensdiagram, figur 18, ble utarbeidet for å modellere interaksjonen i systemet og få et overblikk over de sentrale objektene og metodene i systemet (Sommerville, 2016, s.207).



Figur 18: Sekvensdiagram

Tekstlig beskrivelse - Hovedflyt:

1. Administrator eller agentleder setter opp et wallboard.
2. Wallboardet lagres og opprettes.
3. En unik URL-lenke til wallboardet blir generert.
4. Administrator sender ut URL-lenkten til agentleder og agenter.
5. Når en agent endrer status opprettes nye eventer til websocket.

WebSocket oppdaterer sanntidsinformasjon til wallboardet.

[3.2.2 Aktivitetsdiagram](#)

Det ble laget aktivitetsdiagrammer (figur 19 og 20) for å få en visuell oversikt over interaksjonen brukeren må gjennom i systemet. I følge Sommerville (2016, s.135) brukes aktivitetsdiagram for å vise aktivitetene involvert i en prosess.

Hensikten var å bruke diagrammene som et virkemiddel for å også selv kunne sette seg inn i systemet som skulle lages, samtidig som det er et godt verktøy for kommunikasjon av forståelsen av systemet i samarbeid med oppdragsgiveren.

Det ble utarbeidet to aktivitetsdiagrammer. Et for selve interaksjonen med hele systemet som helhet og et for flyten i oppsettet av et wallboard.

Diagrammet starter med en full sirkel som indikerer starten på aktiviteten, diamantene betyr valg som brukeren står ovenfor, og strekene betyr flyten fra en aktivitet til en annen (Sommerville, 2016, s.137).

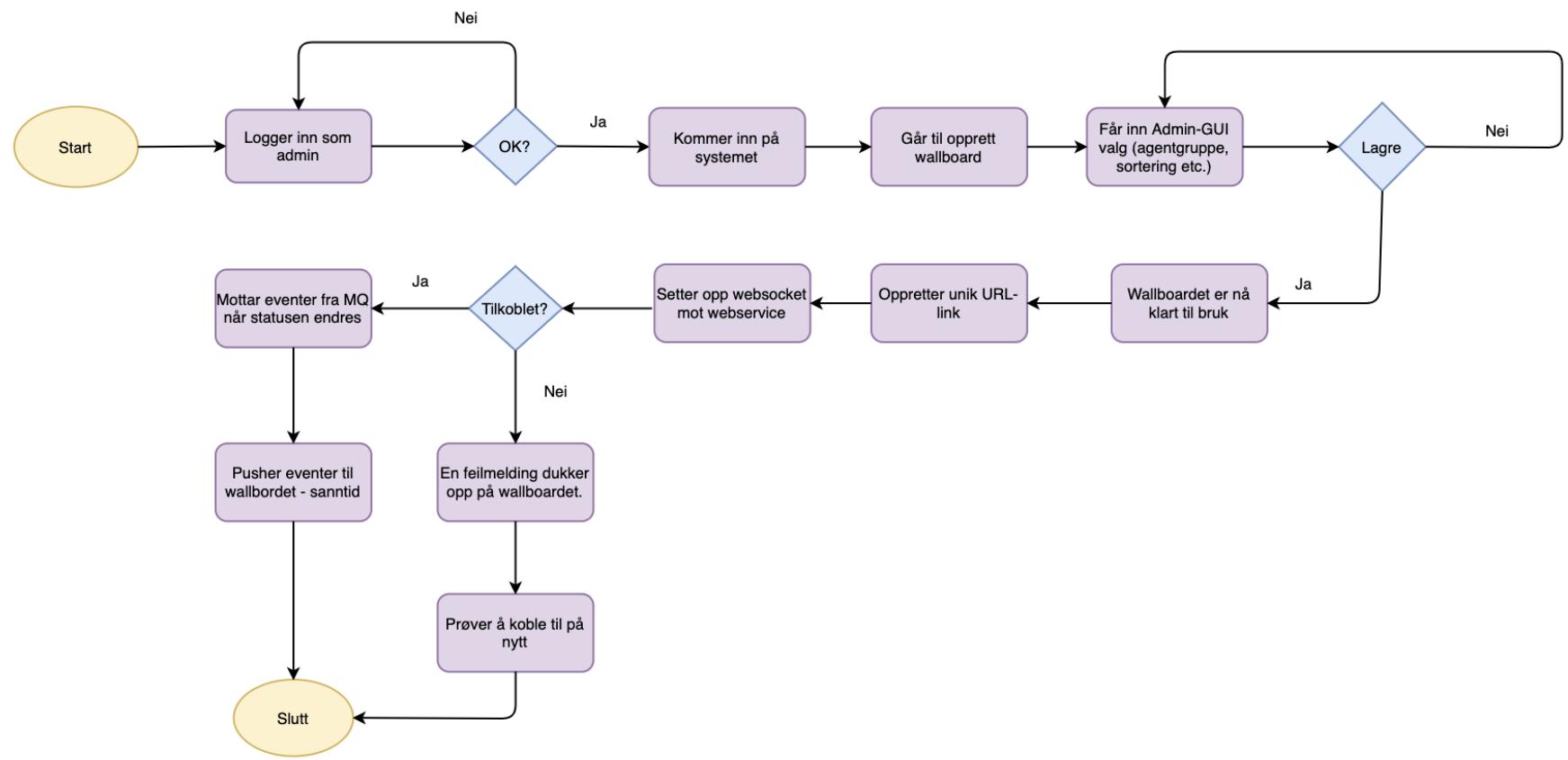
[Gangen i systemet](#)

Det første aktivitetsdiagrammet, figur 19, viser interaksjonen i systemet. Det viser hva en administrator må gjøre fra start til slutt for å få publisert og distribuert et velfungerende wallboard. Diagrammet viser samtidig den tekniske utførelsen som

skjer ved opprettelsen av et wallboard som å motta eventer, pushe eventer og reconnect.

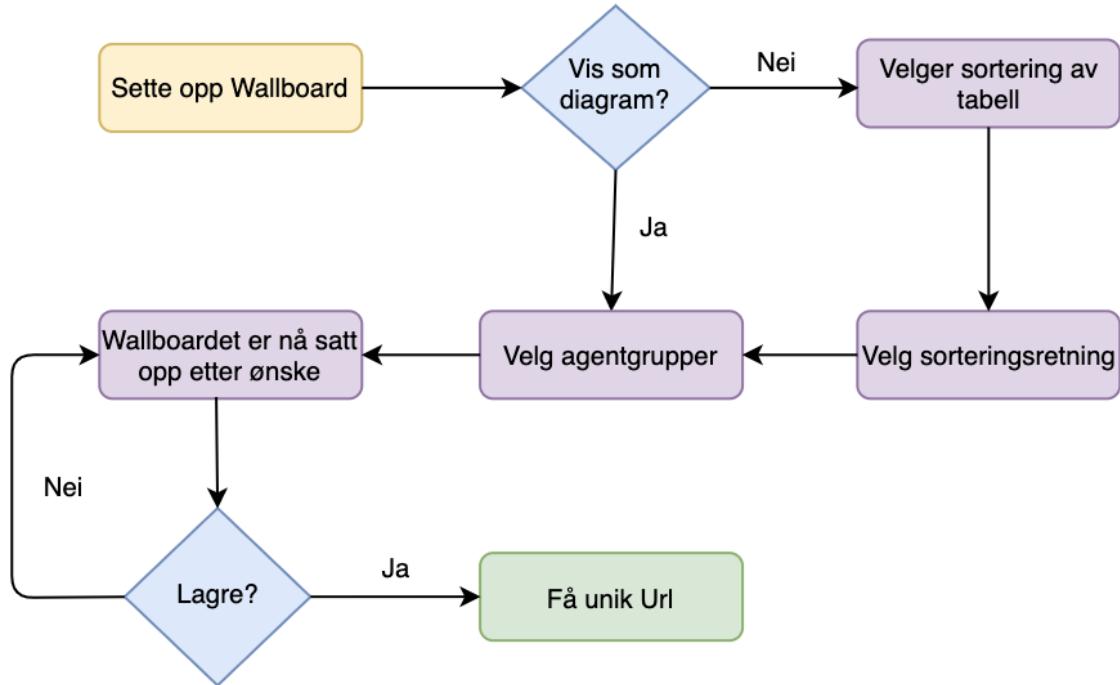
Det er brukt farger for å lettere se de forskjellige aktivitetene. Gult er for start og slutt, blått er valgene en må ta, og lilla er selve handlingen. Gjennom denne prosessen er det mulig for en å ta flere valg. Noe som vil føre til at utfallet kan bli forskjellig.

Interaksjonen starter med at en med administratorrettigheter prøver å logge seg inn på systemet. Hvis innloggingen blir vellykket, kommer administratoren seg inn på systemet. Administrator kan deretter gå til siden der opprettelse av et wallboard med ønsket funksjonalitet skjer. Dersom wallboardet blir lagret, blir det gitt en unik URL-lenke. Dersom man aksesserer URL'en vil wallboardet prøve å gjøre en tilkobling til websocket slik at det kan oppdateres med sanntidsdata. Hvis tilkoblingen mislykkes, vil det gi en feilmelding og systemet vil prøve å koble seg opp på nytt.



Figur 19: Aktivitetsdiagram som viser hele interaksjonen i systemet

Det andre aktivitetsdiagrammet, figur 20, viser oppsettet av wallboardet i wallboard-innstillinger i Zisson Interact. Diagrammet viser stegene for hvordan brukeren kan tilpasse wallboardet etter behov.



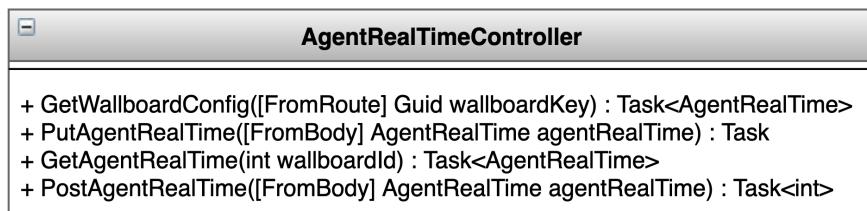
Figur 20: Aktivitetsdiagram som viser oppsett av et wallboard

Interaksjonen her starter med at det er et ønske om å sette opp wallboardet. Da antar man at administrator allerede er logget inn i systemet. Administrator får valg som går på filtrering, sortering og diagramform. Dersom det blir valgt diagramform, utgår valg om sortering på kolonne og sorteringsretning, da disse i dette tilfellet ikke vil ha noe betydning for hvordan wallboardet blir visualisert. Videre kan man lagre wallboardet ved å trykke på Lagre, og man blir da tatt til en ny side og får en unik URL-lenke slik at wallboardet kan distribueres videre. Dersom man ikke trykker på lagre-knappen forblir man på samme siden.

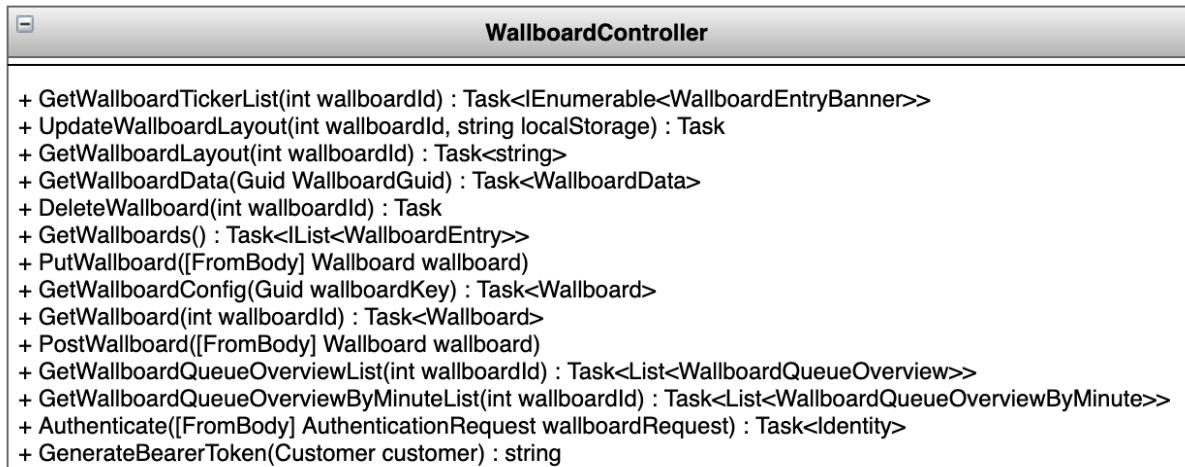
3.3 Logical View

Logical view beskriver strukturen til de logiske delene av løsningen (Drolsjammer og Olasveengen, 2016). Logical view består ofte av klassediagrammer (Drolsjammer og Olasveengen, 2016). En slik tilnærming bidrar til å vise arv, abstraksjon og innkapsling av de ulike klassene (Kruchten, 1995, s. 3).

I backend prosjektet WaveWebAPI er det controlleren AgentRealTimeController og WallboardController det har blitt jobbet med. Figur 21 viser controlleren som tilhører det nye wallboardet, mens figur 22 viser controlleren for Zisson sitt eksisterende wallboard som inneholder DeleteWallboard og GetWallboards metoden som blir brukt på både det nye og det eksisterende wallboardet.

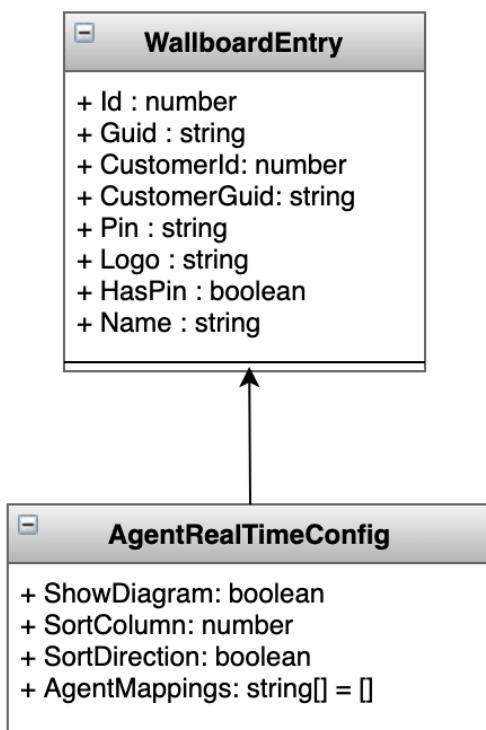


Figur 21: AgentRealTimeController i WaveWebAPI



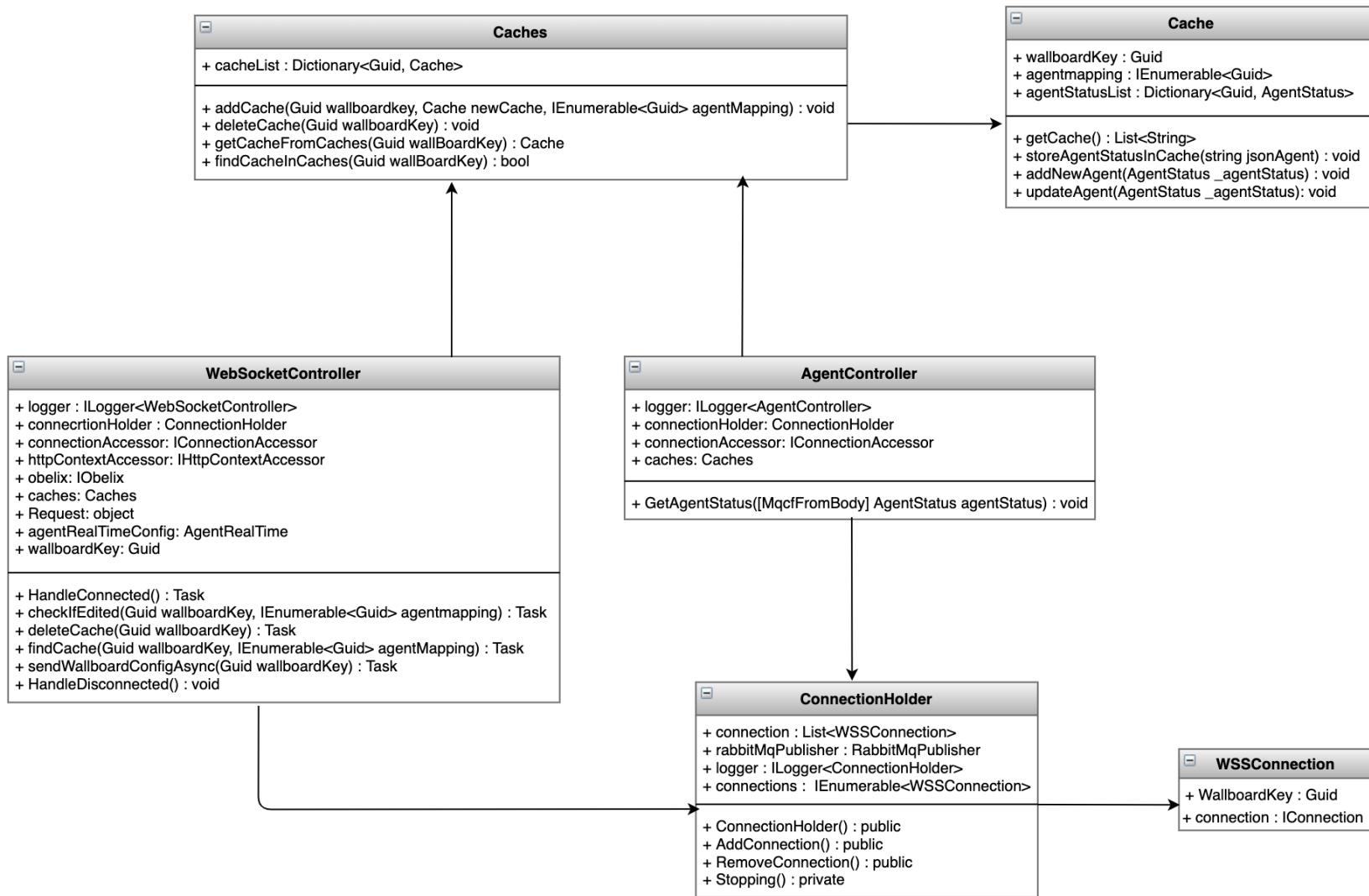
Figur 22: WallboardController i WaveWebAPI

I frontend prosjektene Rewave og AgentRealTimeWallboard er det kun jobbet med klassen AgentRealTimeConfig som arver fra klassen WallboardEntry. Dette er klassene som representerer hvert objekt av et wallboard som blir opprettet. Figur 23 illustrerer dette i gjennom et klassediagram. I kildekoden til Rewave kan man se at klassen inneholder flere attributter. Dette er attributter som tidligere ble brukt av Zisson i deres eksisterende wallboard, men som ikke har blitt brukt av studentene og er derfor tatt bort i illustrasjonen av klassen.



Figur 23: Klassediagram fra prosjektene Rewave og AgentRealTimeWallboard

Figur 24 viser et klassediagram for prosjektet WallboardService da det i dette prosjektet er benyttet en del objektorientert programmering. Klassediagrammet viser en forenklet fremstilling av klassene som er i prosjektet og deres forhold i mellom.



Figur 24: Klassediagram over prosjektet WallboardService

3.4 Implementation/development View

Implementation/development view fokuserer på den faktiske system modulens organisasjon i utviklingsmiljøet (Kruchten, 1995, s. 6). Subsystemene er ofte organisert i et hierarki med lagdelinger (Kruchten, 1995, s. 6).

Det er her valgt å beskrive viktige komponenter og controllere i de forskjellige prosjektene, samt vise mappestrukturen hvor lagdelingen i prosjektet vises.

3.4.1 Wallboard

3.4.1.1 WallboardService

WallboardService, som er serverdelen av wallboardet, har som oppgave å lytte på agenteventer og sende dette gjennom websocket til klient. WallboardService skal også hente ut informasjon om wallboardkonfigurasjon fra databasen. Dette gjøres ved at WallboardService leser url'en til websocket tilkoblingen og får tak i nøkkelen som hører til et wallboard. Så gjøres en spørring mot databasen med nøkkelen. Alle eventer om agenter som WallboardService lytter på skal lagres til cache. Ved tilkobling til et wallboard skal det hentes fra cache og sendes til klient.

WallboardService består av:

Controllers

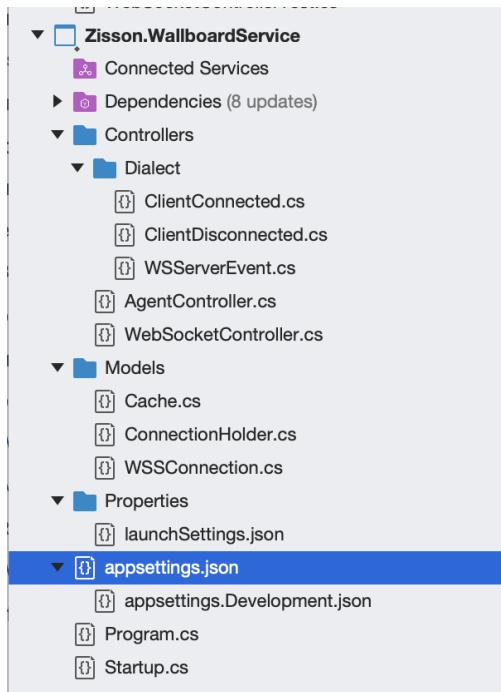
- AgentController: Controller som håndterer eventer knyttet til agentene.
- WebSocketController: Controller som håndterer kobling til websocket samt konfigurasjon av wallboard.

Modeller

- Cache

- ConnectionHolder
- WSSConnection

Bilde av mappestrukturen til WallboardService vises i figur 25.



Figur 25: Mappestruktur av Zisson.WallboardService

3.4.1.2 AgentRealTimeWallboard

AgentRealTimeWallboard er frontend-prosjektet til wallboardet. Denne delen av prosjektet har som oppgave å ta imot informasjon om agenter og wallboard konfigurasjon, og vise dette frem. Informasjonen som kommer oppdaterer seg kontinuerlig i en tabell. Her håndteres sortering av kolonnene og om diagram skal vises eller ikke.

AgentRealTimeWallboard består av:

Komponenter

- agent-overview-wallboard: Selve vinduet hvor agenttabellen vises.
- error-wallboard: Errormelding ved mislykket tilkobling til server/ websocket.

- header-wallboard: Selve bakgrunnen på wallboardet med en rad øverst for dato og tid, samt logo.
- data.service.ts: Service-komponent for å håndtere deling av data mellom komponentene agent-overview-wallboard og header-wallboard. Brukt for å kunne dele logo som kommer inn fra websocket på agent-overview-wallboard og sende denne over til header for å vise frem logo.

Modeller

- AgentRealTimeConfig
- AgentStatus
- WallboardEntry

Bilde av mappestrukturen til AgentRealTimeWallboard vises i figur 26.

```

    <pre>
        <code>
            <ul>
                <li><code>agent-overview-wallboard</code>
                    <ul>
                        <li><code># agent-overview-wallboard.component.css</code></li>
                        <li><code><agent-overview-wallboard.component.html></code></li>
                        <li><code>TS agent-overview-wallboard.component.spec.ts</code></li>
                        <li><code>TS agent-overview-wallboard.component.ts</code></li>
                    </ul>
                <li><code>error-wallboard</code>
                    <ul>
                        <li><code># error-wallboard.component.css</code></li>
                        <li><code><error-wallboard.component.html></code></li>
                        <li><code>TS error-wallboard.component.spec.ts</code></li>
                        <li><code>TS error-wallboard.component.ts</code></li>
                    </ul>
                <li><code>header-wallboard</code>
                    <ul>
                        <li><code># header-wallboard.component.css</code></li>
                        <li><code><header-wallboard.component.html></code></li>
                        <li><code>TS header-wallboard.component.spec.ts</code></li>
                        <li><code>TS header-wallboard.component.ts</code></li>
                    </ul>
                <li><code>models</code>
                    <ul>
                        <li><code>TS AgentRealTimeConfig.ts</code></li>
                        <li><code>TS AgentStatus.ts</code></li>
                        <li><code>TS WallboardEntry.ts</code></li>
                    </ul>
                <li><code>services</code>
                    <ul>
                        <li><code>TS data.service.spec.ts</code></li>
                        <li><code>TS data.service.ts</code></li>
                    </ul>
                </li>
            </ul>
        </code>
    </pre>

```

Figur 26: Mappestrukturen i AgentRealTimeWallboard prosjektet

3.4.2 Zisson Interact

3.4.2.1 Rewave

ReWave er frontend-prosjektet til Zisson Interact. ReWave har som oppgave å gi brukeren mulighet til å sette opp et wallboard, for så å sende oppsettet til WaveWebAPI gjennom HTTP protokollen ved bruk av REST arkitekturen. Gjennom ReWave kan man også aksessere det nye wallboardet, og på denne måten aksessere AgentRealTimeWallboard-prosjektet der selve wallboardet visualiseres.

Delene som gruppen har jobbet med i ReWave består av:

Komponenter

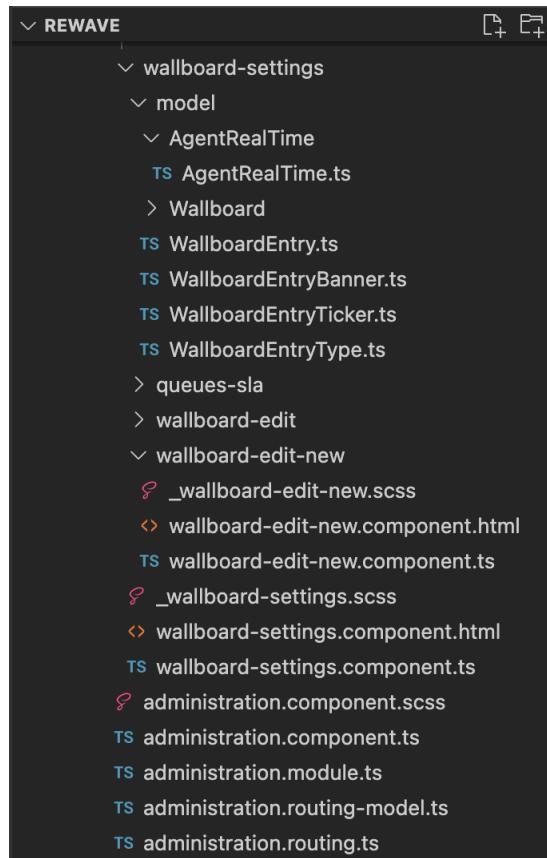
- wallboard-edit-new: Komponentet for oppretting av wallboard med ny konfigurasjonsmodell.
- wallboard-settings: Komponent for visning av alle opprettede wallboards, både med ny og gammel konfigurasjonsmodell. Her er det også mulig å opprette, klone, kopiere og slette wallboard. I tabellen vises også typen til wallboardene, om de er opprettet med ny eller gammel konfigurasjon.
- web-api: i Dette komponentet ble det lagt inn metoder som inneholder REST-kall som blir videresendt til WaveWebAPI-prosjektet.
- administrasjon.routing: Her ble det laget en "sti" som kobler sammen variablene laget i administration.routing-module og navnet på komponenten, slik at man blir navigert til riktig komponent fra wallboard-settings-komponenten.
- administration.routing-module: Her ble det laget variabler som hjalp til med å navigere seg riktig i de nye komponentene for nye konfigurasjonen av wallboardet.
- administration.module: Den nye komponenten med konfigurasjonen for wallboardet ble deklarert her.

- WavePlatformConfig.ts: Her ble porten som prosjektet AngularRealTimeWallboard kjører på lagt til. På denne måten ble det nye wallboardet åpnet, fremfor den eldre versjonen Zisson hadde fra tidligere.

Modeller

- AgentRealTime.ts
- WallboardEntry.ts
- WallboardEntryType.ts

Bilde av mappestrukturen til ReWave vises i figur 27.



Figur 27: Mappestrukturen i ReWave prosjektet

3.4.2.2 WaveWebAPI

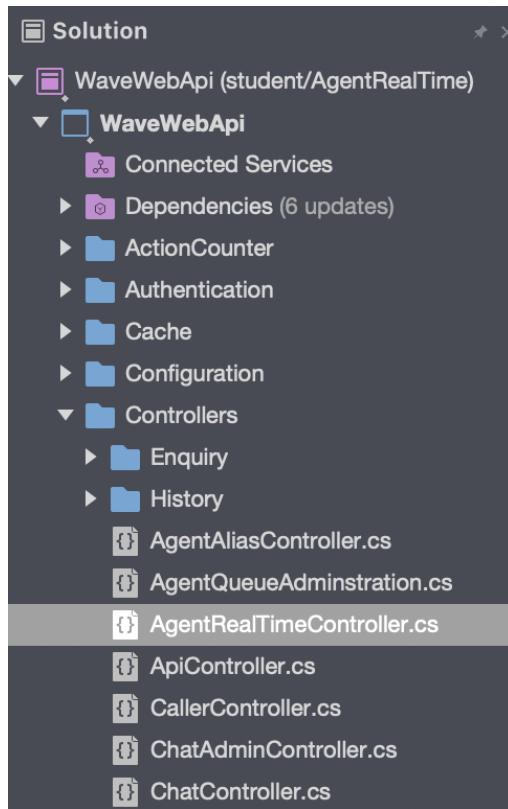
WaveWebAPI er backend prosjektet til ReWave. Det har som oppgave å sørge for at alle kallene som kommer inn fra ReWave kjøres mot databasen. Dette er REST-kallene: “POST”, “PUT”, “DELETE” og “GET”.

Deler vi har arbeidet med består av:

[Controller](#)

- AgentRealTimeController: Controller som mottar REST-kall fra frontend og gjør kall mot databasen.

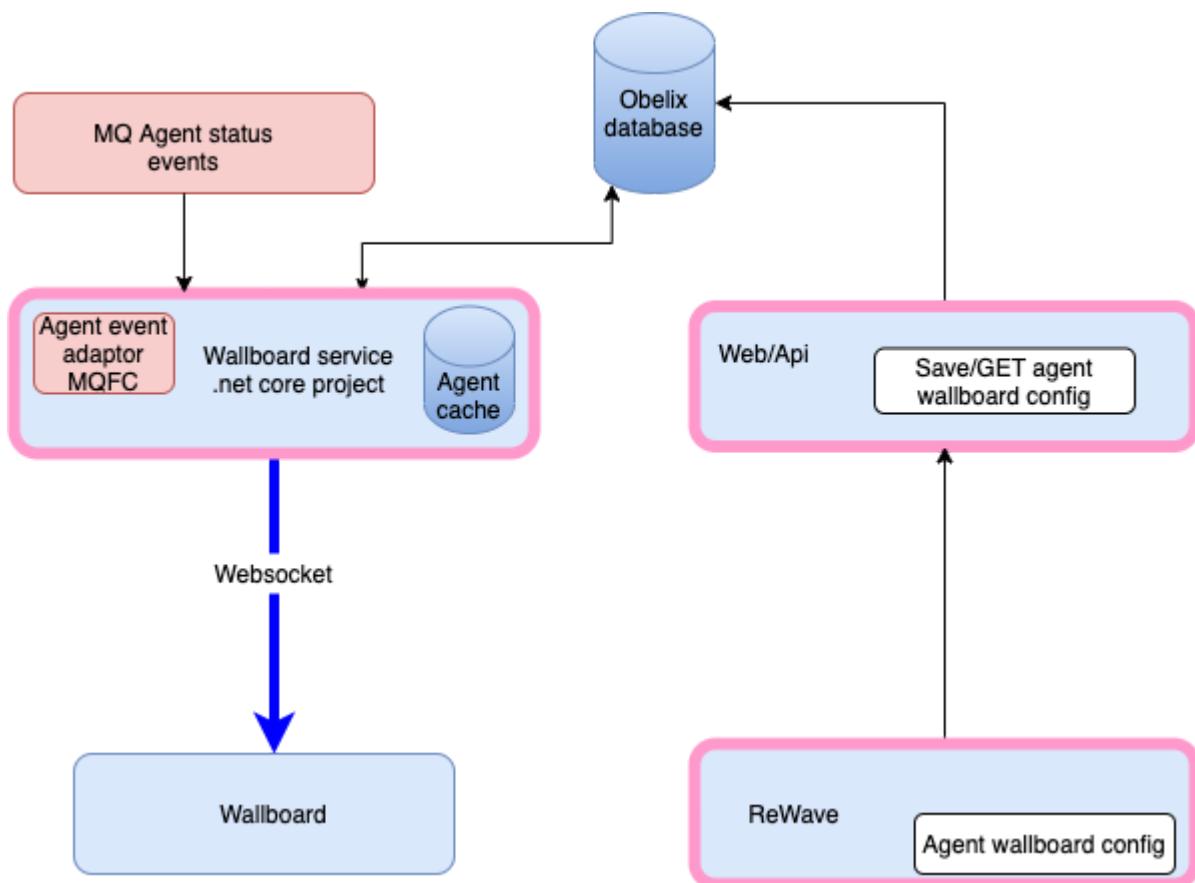
Bilde av mappestrukturen til WaveWebAPI vises i figur 28.



Figur 28: Utsnitt av mappestrukturen i WaveWebAPI prosjektet

3.5 Physical View

Physical view beskriver hvordan applikasjonen er installert og hvordan det oppfører seg i forhold til andre systemer og datamaskiner (Kontio, 2005, s. 4).



Figur 29: Modell av programmets oppbygging

Figur 29 viser hvordan oppsettet fra ReWave skal gå gjennom Web/Api, og lagre og hente wallboard fra databasen Obelix. WallboardService skal videre hente en unik wallboardnøkkel fra URL-en fra nettadressen som konfigureres ved opprettelsen av wallboardet. Denne unike nøkkelen skal fungere som en unik id mot Obelix-databasen for hvilke konfigurasjonsvalg som er gjort i ReWave.

Konfigurasjonen fra ReWave blir deretter sendt gjennom websocket til wallboardet som viser det oppsettet som er blitt utført. [Se punkt 4.3 Websocket tilkobling, for nærmere forklaring av websocket funksjonaliteten.](#)

I WallboardService skal agenter kontinuerlig lagres til minne (cache) slik at hvis det eventuelt skulle oppstå feil i websocket tilkoblingen, så vil siste eventer på agentene hentes ut fra cache ved gjenoppkobling til websocket. WebSocket skal gjøre det mulig å vise informasjonen fra agentene i sanntid og fungerer som et sugerør mot statuseventer. MQ er en server som fungerer som en meldingsprotokoll og registrerer hendelser som WallboardService lytter på ved endringer i statuser, innkommende samtaler og andre eventer. Agent event adaptor MQCF (Message Queue Controller Framework) er et mellomlag mellom MQ og WallboardService, som gjør informasjonen lett forståelig for WallboardService-prosjektet. MQCF hjelper til med å parse eventmeldingene og kjøre de riktige instansene og metodene.

4.0 Sentrale datastrukturer og løsninger

4.1 Datastrukturer

En *datastruktur* er den måten en samling dataverdier er organisert på (Uttersrud, 2021, Avsnitt: 1.1.1 Hva er en algoritme?).

Sentrale datastrukturer som er brukt i programmet er:

[Dictionary](#)

1. Dictionary <TKey, TValue> som er en rask datastruktur og er implementert som et hashmap (Microsoft, Avsnitt: Remarks, u.å.). Det vil si at en nøkkel brukes for å slå opp i en tabell for å finne verdien som hører til.

Figur 30 og 31 viser kodeeksempel fra prosjektet:

```
3 references
public Cache getCacheFromCaches(Guid wallBoardKey){
    Cache cache;      You, 2 weeks ago • Endret til diction
    if(cacheList.TryGetValue(wallBoardKey, out cache)){
        return cache;
    }
    return null;
}
```

Figur 30: Dictionary sin TryGetValue

Figur 30 bruker Dictionary sin metode TryGetValue for å hente ut en spesifikk cache fra Dictionary som holder på alle cachene.

```
1 reference
public void AddCache(Guid wallboardkey, Cache newCache, IEnumerable<Guid> agentMapping)
{
    newCache.agentmapping = agentMapping;
    cacheList.TryAdd(wallboardkey, newCache);
}
```

Figur 31: Dictionary sin TryAdd

Figur 31 bruker Dictionary sin metode TryAdd for å legge til en ny cache.

List

2. List<> som er en sterkt typet liste som kan aksesseres med index (Microsoft, Avsnitt: List<T> Class, u.å.). Med sterkt typet liste menes en liste som bare kan ha en datatype. Ifølge Microsoft er List<> raskere enn ArrayList<> i de fleste tilfeller og er samtidig “type safe” (Microsoft, Avsnitt: Performance considerations, u.å.).

Figur 32 viser kodeeksempel fra prosjektet:

```

    1 reference
public List<String> getCache(){
    List <String> agents = new List<String>();

    foreach(KeyValuePair<Guid, AgentStatus> agentstatus in agentStatusList){
        agents.Add(JsonSerializer.Serialize(agentstatus.Value));
    }
    return agents;
}

```

Figur 32: List

Figur 32 viser at det brukes List med strings for å returnere agentstatus objekter fra cache.

Mat-table og DataSource

3. Mat-table i angular og DataSource-array, vist i figur 33, 34, og 35. Mat-table er selve tabellen i wallboardet. Fordelen med Mat-table og DataSource er at man kan innkapsle hvilken som helst logikk for sortering, filtrering, paginering og datahenting i en tabell (Angular Material, Avsnitt: DataSource, u.å.).

Figur 33 - 35 viser kodeeksempel fra prosjektet:

```
datasource : MatTableDataSource<AgentStatus>;
```

Figur 33: Datasource

```

sortTable(sortAsc : boolean, sortOn : number){
    this.sort.sort({ id: null, start: 'desc', disableClear: false });
    this.sort.sort({ id: this.convertSortOn(sortOn), start: this.sortDir(sortAsc), disableClear: false });

    const toState = 'active';

    this.datasource.sort = this.sort;
    (this.sort.sortables.get(this.convertSortOn(sortOn)) as MatSortHeader)._setAnimationTransitionState({ toState });
}

```

Figur 34: SortTable

```

<!-- Hvis Toggle = false vises vanlig tabell-->
<ng-template #elseBlock>
  <table mat-table #table class="table table-striped" [dataSource]="datasource"
    matSort aria-label="Elements" matSort [matSortActive]="sortOn" [matSortDirection]="sortDirection"
    matSortDisableClear>

    <!-- Kolonne: Navn -->
    <ng-container matColumnDef="FirstName">
      <th mat-header-cell *matHeaderCellDef mat-sort-header>Agentnavn</th>
      <td mat-cell *matCellDef="let data" >
        | {{ data.FirstName }} {{ data.LastName }}
      </td>
    </ng-container>

```

Figur 35: Mat-table i HTML

4.2 Bruk av designmønstre i programmet

Det er brukt designmønsteret *dependency injection* sammen med *singleton* for å sikre at det kun opprettes en instans av klassen Caches, som med en dictionary er beholderen for alle cachene i systemet.

Fordelene ved bruk av dependency injection er (V. Khorikov, 2016):

1. Overføre avhengigheter til de objektene som bruker dem. En avhengighet er et objekt som et annet objekt er avhengig av.
2. Er enklere å teste.

I WallboardService sin Startup.cs er det en metode ConfigureServices. Metoden oppretter en instans av Caches med metoden addSingleton. På denne måten brukes dependency injection for å sikre at det kun skjer én opprettelse av Caches, og dette skjer ved kjøring av prosjektet. Det vil da si at Startup.cs er avhengig av objektet Caches.

4.3 Websocket tilkobling

For å opprette websocket-tilkobling mellom server og klient ble RxJS biblioteket (Reactive Extensions for JavaScript) brukt på frontend. Websocket-tilkoblingen ble laget i metoden “connect” i “agent-overview-wallboard”-komponentet på wallboard-prosjektet, der det ble satt en subscriber som kontinuerlig lytter på meldinger fra websocket tilkoblingen. RxJS gir tilgang til en type *Observable*, som gjør at man hele tiden kan lytte på hendelser fra server igjennom websocket-tilkoblingen (RxJS, u.å).

Dataene kan videre hentes ut i JSON-strenger og blir gjort om til et objekt som inneholder disse dataene i modellen. Modellene som benyttes er AgentStatus og AgentRealTimeConfig.

Figur 36 og 37 viser metoden Connect som lytter til websocket tilkoblingen og henter ut data om agenthendelser:

```

connect(): void {
    this.subject = webSocket(
        'ws://localhost:5000/ws?wallboardKey=' + this.wallboardKey
    );

    this.subject.subscribe(
        (msg) => {
            this.object = JSON.parse(JSON.stringify(msg));
            console.log(msg);

            /* If blokk for håndtering av config data fra wallboard config */
            if (this.object.hasOwnProperty('Guid')) {
                this.configWallboard = JSON.parse(JSON.stringify(msg));
                this.checker = this.configWallboard.ShowDiagram;
                this.setSharedLogo(this.configWallboard.Logo);

                //Sortere tabell
                if (!this.checker) [
                    this.sortOn = this.configWallboard.SortColumn;
                    this.sortAsc = this.configWallboard.SortDirection;
                    this.sortDirection = this.sortDir(this.sortAsc);
                    this.sortTable(this.sortAsc, this.sortOn);
                    this.datasource.data = this.datasource.data;
                ]
            } else {
                /* Else blokk for håndtering av agentstatususer */
                const agentStatusObject = <AgentStatus>(
                    JSON.parse(JSON.stringify(msg))
                );
                const datasource = this.datasource.data;

                datasource.forEach((element) => {
                    if (element.LoginGuid === agentStatusObject.LoginGuid) {
                        const indeks = datasource.indexOf(element);
                        this.datasource.data.fill(agentStatusObject, indeks, indeks + 1);
                        this.agentUpdated = true;
                        this.datasource.data = this.datasource.data;
                    }
                });
            }

            if (!this.agentUpdated) {
                datasource.push(agentStatusObject);
                this.datasource.data = this.datasource.data;
            }
        }
    );
}

```

Figur 36: Connect-metoden i AgentRealTimeWallboard prosjektet.

```

    /* Logikk for oppstelling av agentstatuser til diagram */
    this.countAvailable = 0;
    this.countProcess = 0;
    this.countAfterCallWork = 0;
    this.countWrapUp = 0;
    this.countPause = 0;
    this.countUnavailable = 0;

    datasource.forEach((element) => {
        if (element.AgentState === agentStatusEnum.Available) {
            this.countAvailable++;
        } else if (
            element.AgentState === agentStatusEnum.Unavailable &&
            (element.PauseGuid === agentStatusEnum.Pause |||
            element.PauseGuid === agentStatusEnum.PauseKaffe |||
            element.PauseGuid === agentStatusEnum.PausePersonlig)
        ) {
            this.countPause++;
        } else if (element.AgentState === agentStatusEnum.Process) {
            this.countProcess++;
        } else if (element.AgentState === agentStatusEnum.WrapUp) {
            this.countWrapUp++;
        } else if (element.AgentState === agentStatusEnum.AfterCallWork) {
            this.countAfterCallWork++;
        } else if (element.AgentState === agentStatusEnum.Unavailable) {
            this.countUnavailable++;
        }
    });
}

this.agentUpdated = false;
this.datasource.data = this.datasource.data;

this.datasource.data = datasource.filter(
    (element) => !this.statesToRemove.includes(element.AgentState)
);
}
},
(err) => {
    this.errorAlert = true;
    console.log('Error occurred: ', err);
    setTimeout(() => {
        this.connect();
        console.log('Reconnect');
        this.errorAlert = false;
    }, 5000);
},
() => console.log('websocket complete')
);
}
}

```

Figur 37: Fortsettelse Connect-metoden i AgentRealTimeWallboard prosjektet.

5.0 Systemets andre forhold

Systemet er avhengig av relasjonsdatabasen Obelix for å hente konfigurasjon fra ReWave, med wallboardnøkkelen for å åpne det unike wallboardet og for å lagre og slette wallboard som er satt opp i Zisson Interact. Det er også behov for VPN-tilgang for å få tilgang til både verdier i Obelix-databasen og agenteventer fra MQ agent status events. Systemet er også avhengige av Zisson sine interne API'er. API (Application Programming Interface) er et programmeringsgrensesnitt som brukes for å utveksle data mellom forskjellige applikasjoner (HubSpire, u.å.).

Det ble jobbet med NuGet-pakker som inneholdt funksjonalitet Zisson selv hadde laget, og som var nødvendige gjennom hele prosjektet. Pakkene måtte kontinuerlig oppdateres i begge backend-prosjektene, da disse stadig ble oppdatert hos Zisson. Dette var for å få inn nødvendig funksjonalitet, som f.eks. å kunne sende kall til databasen. Ettersom utviklingen av nytt wallboard var i gang, måtte Zisson legge til nye databaseoppdateringer og andre endringer i forhold til dette i såkalte pre-release NuGet-pakker. Dette fordi koden fremdeles var under utvikling og innholdet i pakkene stadig måtte endres, før det senere kunne legges til i de offisielle NuGet-pakkene som vil bli brukt av utviklerne hos Zisson.

6.0 Brukergrensesnittet

6.1 Zisson Interact

Zisson Interact er der innstillingene og oppsettet av wallboardet skjer. Figur 38 viser hvor man skal trykke for å legge til et nytt wallboard med det nye oppsettet, samt listen hvor alle wallboard ligger. Kolonnen helt til høyre i listen, bestående av ulike ikoner, var noe Zisson allerede hadde i brukergrensesnittet sitt fra før. Dette er

ikoner for å åpne URL-lenke for wallboardet, kopier lenke til utklippstavle, og rediger, klon eller slett wallboardet.

The screenshot shows a table titled 'Wallboard-liste' with three rows: 'Avdeling jul', 'TIRSDAG', and 'ONSDAG'. A context menu is open over the first row, containing options '+ Nytt wallboard' and '+ Gammelt wallboard'. To the right of the table, there are columns for 'Kør' (Run), 'Lenke' (Link), and 'Kopier Lenk' (Copy Link). The 'Lenke' column contains icons for opening in new tab, copying, and deleting. The 'Kopier Lenk' column contains icons for opening in new tab, copying, and deleting.

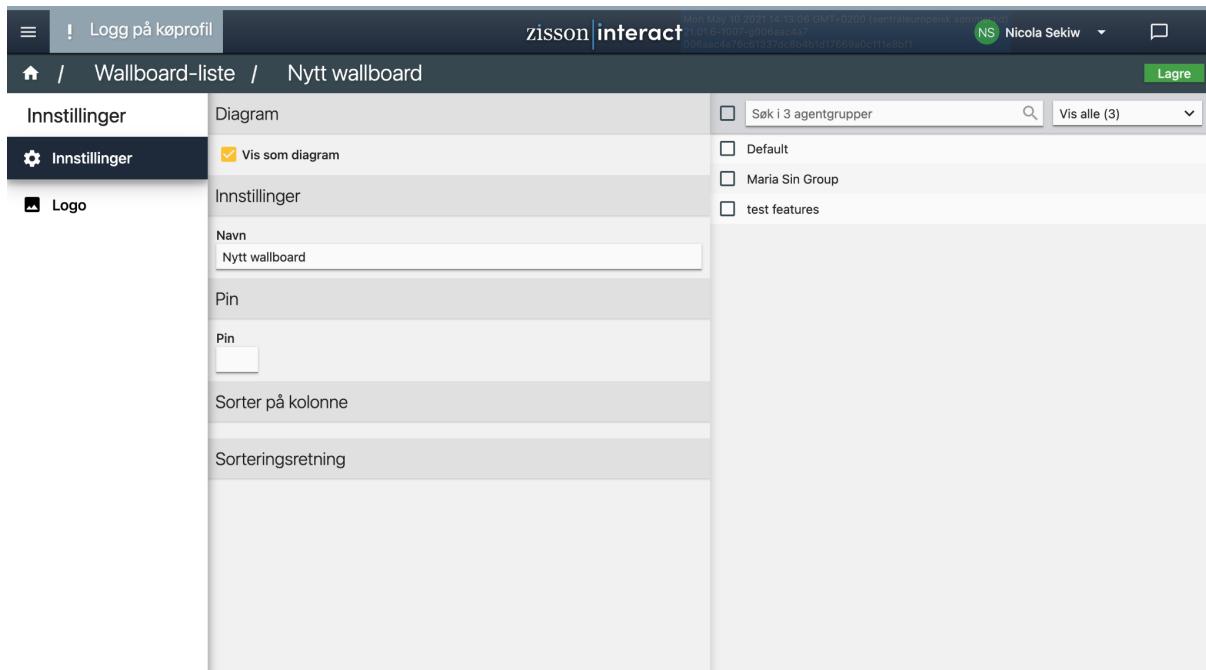
Figur 38: Wallboard-liste

Figur 39 viser innstillingene for oppsettet av nytt wallboard. Her kan man velge å vise agentvinduet som diagram, sette navn på wallboard, velge sortering på de ulike kolonnene, og sorteringsrekkefølge. Til høyre kan man velge ønskede agentgrupper som skal vises på wallboardet.

The screenshot shows the 'Nytt wallboard' configuration page. On the left, there's a sidebar with 'Innstillinger' (Settings) selected. The main area has a 'Diagram' tab active. Under 'Innstillinger', there's a checkbox 'Vis som diagram'. Below it, the 'Navn' (Name) is set to 'Nytt wallboard'. Under 'Pin', there's a placeholder 'Pin'. In the 'Sorter på kolonne' (Sort by column) section, there are two rows of radio buttons: 'Agentnavn', 'Agentgruppe', 'Status', 'Tid i status' in the first row, and 'Call status', 'Profil' in the second row. At the bottom, there's a 'Sorteringsretning' (Sorting direction) section with a checkbox 'Sorter stigende (A-Å)' (Sort ascending (A-Z)).

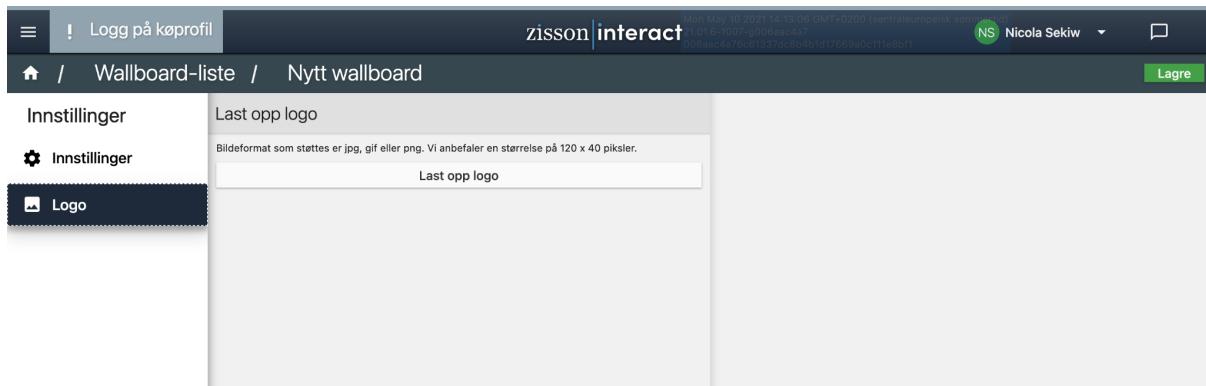
Figur 39: Nytt wallboard

Figur 40 viser at sorteringsfunksjonaliteten og sorteringsretningen blir gjort utilgjengelig når det velges diagramform, da disse valgene ikke lenger er relevante.



Figur 40: Vis som diagram

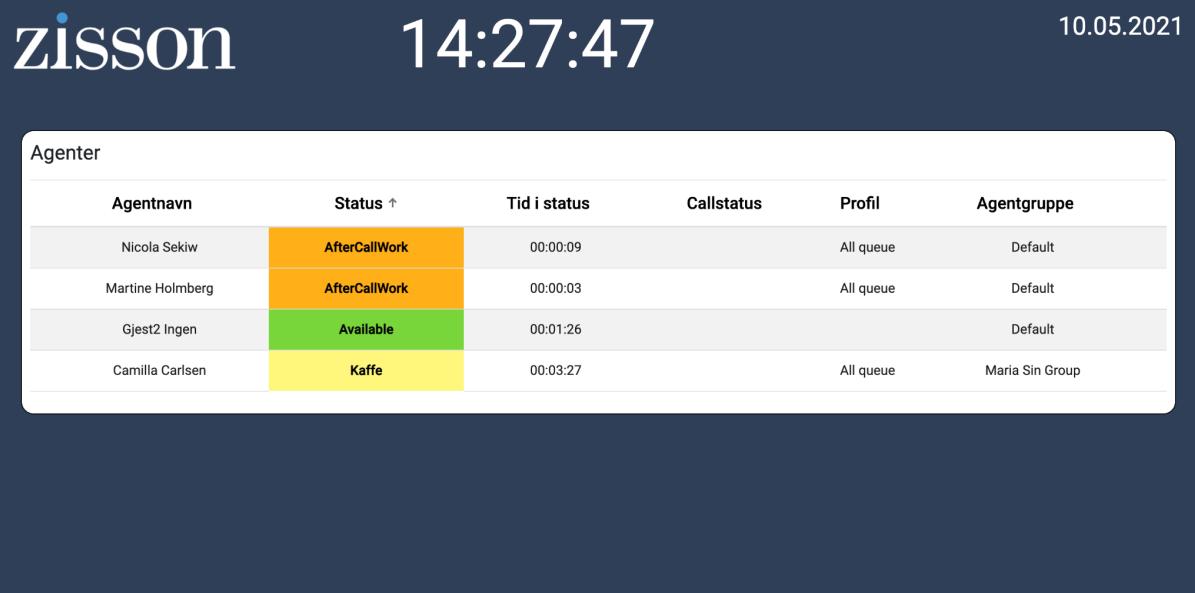
Figur 41 viser at brukeren kan velge å laste opp en logo til oppsettet av wallboardet.



Figur 41: Last opp logo

6.2 Wallboard

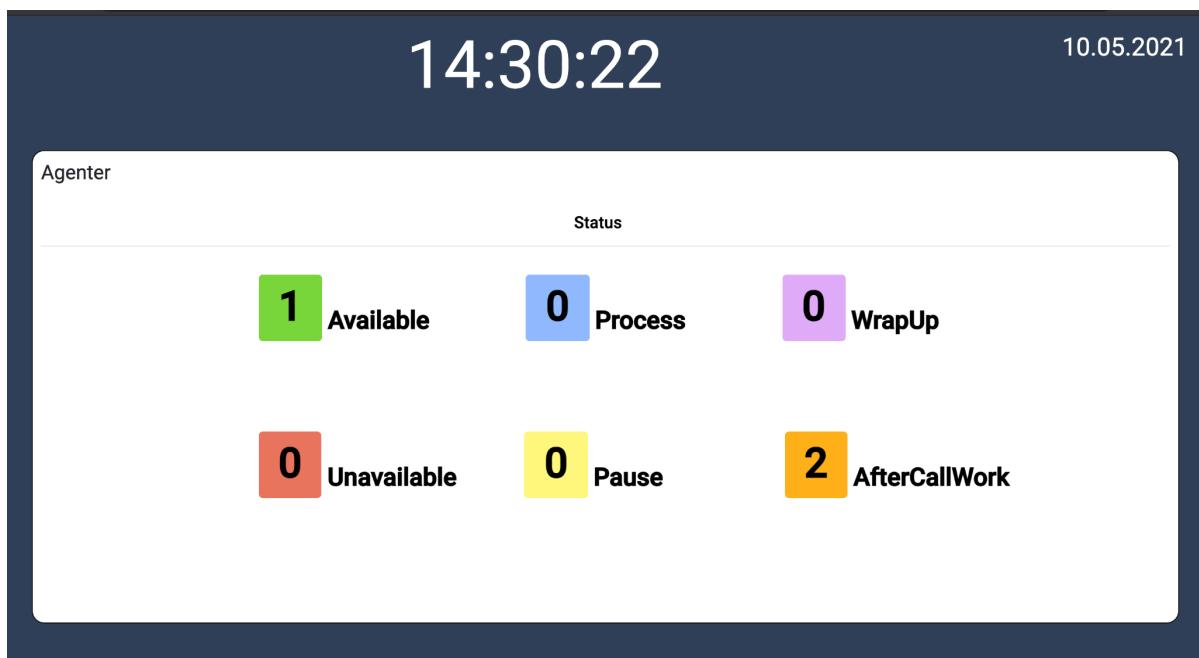
Figur 42 viser wallboardet etter at oppsettet fra Zisson Interact er lagret og URL-lenken er åpnet i et nytt nettleservindu. Wallboardet viser sortering på status og ønsket opplastet logo, samt kolonnene: Agentnavn, status, tid i status, callstatus, profil og agentgruppe.



Figur 42: Wallboard i tabell-form med agenter

Figur 43 viser wallboardet etter at diagramform er valgt i Zisson Interact.

Wallboardet viser antall agenter i hver status.



Figur 43: Wallboard i diagramform med agenter

7.0 Designvalg av sluttprodukt

7.1 Tabell

Tabellen er designet med en tanke på å fremstille dataene mest mulig synlig og ryddig for å gjøre den lett lesbar for brukeren. Det ble utarbeidet to forskjellige varianter som viste farger over hele raden og kun farge over status. Den ene varianten ble designet med sebra-linjer i hvit og grå, da dette er en god måte å hjelpe brukeren med å holde seg på linjen når brukeren leser (Hellmuth, 2019). Farge over hele raden bidrar også med den samme synligheten som en tabell med sebra-linjer gjør. Ved den tilnærmingen hvor det var ment å ha farge over hele raden var det ment å ha horisontale linjer. Dette for å gi minst mulig visuelle forstyrrelser (Hellmuth, 2019).

Etter tilbakemeldinger på A/B testing av prototypene, var det et flertall for farge på kun status og derfor ble det valgt å beholde sebra-tabellen med farge på kun statuskolonnene. A/B testing baserer seg på å vise frem to varianter av et brukergrensesnitt og måle hvem som gir best resultat (Optimizely, u.å.) Se punkt [6.1 Ikke-funksjonelle tester](#) for utdypning av hvordan A/B testingen ble utført.

For å tydeliggjøre hva som er overskrift og hva som er elementene i tabellen, ser overskriftene for hver kolonne annerledes ut enn radene (Hellmuth, 2019). Dette bidrar til at det er lettere for brukeren å oppdage feltene. (Hellmuth, 2019).

7.2 Ikoner

Ikonene gruppen valgte var pil for sortering, checkbox for valg av agentgrupper og valg av sorteringsretning (A-Å), sjekkboks for valg av diagram og radio buttons for valg av sortering. Disse ble valgt med tanke på standardisert bruk av slike ikoner.

Best practices innen UX design viser at checkbokser blir brukt der det er mulig å velge ingen eller flere elementer fra en liste (Babich, 2016). Derfor er det valgt å bruke dette som en del av brukergrensesnittet i Zisson Interact. Dette står også i tråd med deres tidligere design av konfigurasjonssiden.

Det var et ønske at begge konfigurasjonssidene skulle være konsistente. I brukertester av prototypene ble det blant annet presentert at det skulle være brukt "toggle"-knapp på valg av diagramform og to radio buttons for sorteringsretning, hvor det kunne velges å enten krysse av for stigende eller synkende sortering. Dette var ikke noe Zisson brukte i den gamle konfigurasjonssiden, og derfor ble det valgt å gjøre det likt som det tidligere wallboard oppsettet, ved å bruke check-bokser istedenfor, da brukerne av systemet er kjent med dette fra før av.

Radio buttons blir brukt i sammenhenger hvor man kun skal velge én av flere valg (Nielsen, 2004). Siden det kun skal være mulig å sortere på én kolonne om gangen er dette hensiktsmessig i valget av hvilken kolonne det skal sorteres på.

Tidlig i prototypeprosessen ble det valgt å implementere søkefelt i oppsettet av wallboardet. Dette var etter tilbakemeldinger fra en brukertest, hvor det kom frem at det hadde vært en fordel å kunne søke opp agentgrupper hvis det var mange grupper å velge mellom. Dette designvalget ble viderført da det var et ønsket at oppsettet av wallboardet skulle integreres i Zisson Interact. Det ble valgt å følge retningslinjene til Beggs om blant annet å ha placeholder tekst for søkefeltet som

beskriver handlingen som skal utføres (Beggs, 2019). Det ble også valgt å bruke ikonet forstørrelsesglass, som kan være en fordel i forhold til erfaringer brukeren har fra før med søkefelt (Beggs, 2019).

I følge Sandnes er det viktig at ikonets betydning er kjent for brukeren, det er først da det kan brukes for å støtte det visuelle søker etter funksjonalitet (Sandnes, 2018, s. 92). Sandnes skriver videre at iconer kan ta mindre plass i brukergrensesnittet enn det teksten kan og derfor er de mye brukt i de mindre brukergrensesnittene (Sandnes, 2018, s. 92). Sandnes legger også til at en fordelaktig egenskap ved iconer er at de utnytter menneskets egenskaper til gjenkjenning (Sandnes, 2018, s. 94).

Bruken av iconer i kontekst gjør det også enklere for brukeren å vite hva ikonet betyr (Sandnes, 2018, s. 95). I vårt design gjelder dette spesielt for pilene som står ved hver kolonne. Når disse forstås i sammenheng med en tabell så kan dette gjøre at det er enklere for brukeren å forstå betydningen av ikonet. Det er også brukt en varseltrekant i feilmeldingen som kommer opp ved mislykket tilkobling til websocket.

7.3 Fargevalg

Da prosjektgruppen skulle velge fargene for status i agenttabellen ble det enighet i samråd med Kim, om hvilke farger som kunne være riktig å bruke på wallboardet.

Fargevalgene i tabellen ble valgt på grunnlag av hva Zisson har brukt fra før, både i det gamle wallboardet, og i andre deler av Zisson Interact. Spesielt gjaldt dette fargene som blir brukt ved agentstatus i det gamle wallboardet.

En av tilbakemeldingene gruppen fikk ved brukertesting var at fargene var vanskelig å skjønne for noen av testpersonene. En av grunnene til dette var at disse

testpersonene ikke var kjent med systemet fra før. Fargevalgene ble likevel stående etter gjennomgang av prototypene med nøkkelpersonene fra Zisson; Kim, Christer, Moritz og Thomas. De mente at fargene passet godt overens med Zisson sitt design, og at brukerne av systemet kjenner fargekodene godt.

Beste praksis for bruk av farger i design viser at man burde ta hensyn til fargesvakhet hos bruker og kultur (Kucheravy, u.å.). Kucheravy skriver også at man burde unngå design som er avhengig av fargene rød, grønn og blå. Designet vårt er ikke avhengig av fargene, men de bidrar til å gi et raskt overblikk over statusene til agentene, som også kan oppfattes godt på avstand. Med hensyn til fargesvakhet og fargeblindhet er det inkludert en forklarende tekst under statuskolonnen, slik at både farge og tekst kan indikere hva status tilsier. Eksempelvis vil statusen "Opptatt" indikere med et tekstfelt som sier "Opptatt", samt fargen rød på gjeldende status. Det er ikke tatt hensyn til andre tolkninger av fargene, som f.eks. kulturforskjeller, da fargene er bestemt i tråd med hva oppdragsgiver ønsker og bruker fra før.

For at teksten som står over fargene skal være tydelig og lesbar er det valgt å nedtone bakgrunnsfargene til statusene, slik at fargeverdiene mellom tekst og bakgrunn får større kontrast. Tekstfargen som er valgt er fargen svart fordi den skal være tydelig å lese og ha nok kontrast i forhold til bakrunnen.

7.3.1 Fargene som er brukt på status

Grønn - markerer at agenten er ledig/ available.

I følge Babich indikerer fargen grønn ofte sikkerhet og suksess, og er et symbol på vekst (Babich, 2019). Babich skriver videre at grønn betyr "gå", og i vår kontekst er grønn en farge som indikerer at agenten er ledig og klar for nye oppdrag. Den skal

være sterkt og tydelig etter ønske fra oppdragsgiver, slik at man raskt kan se hvilke agenter som er ledige.

Rød - markerer at agenten er opptatt.

Om fargen rød sier Babich at det er en synlig farge som er i stand til å fokusere på oppmerksomheten raskt og få folk til å ta raske beslutninger (Babich, 2019). Den indikerer også på advarsel eller fare. I vår kontekst er rød en farge som indikerer at agenten er opptatt, så fargen rød advarer mot at denne personen ikke må forstyrres, og skal være motsetningen til grønn som indikerer ledig.

Blå - markerer at agenten jobber, men ikke nødvendigvis er opptatt.

Blå er brukt for å vise agentstatus for agenter som jobber med skriftlige henvendelser. I følge Cameron Chapman kan fargen blå knyttes til kommunikasjon (Chapman, u.å.). Palliyaguru skriver også at blå indikerer tillit, lojalitet og sannhet, og kan gi en følelse av ro og avslapning (Palliyaguru, 2018). Det er derfor valgt å bruke blå for å indikere at en agent verken er helt opptatt eller ledig. En agent kan svare på flere skriftlige henvendelser samtidig, men kan også ta seg av flere underveis. Denne statusen er ikke integrert i Zisson sitt system enda, men er gjort klart på wallboardet for videre utvikling.

Lilla - brukt av Zisson for å indikere at en agent er i "wrap up".

Denne fargen er allerede brukt av Zisson for å indikere at agenten nettopp er ferdig med en samtale og er i en status de kaller "Wrap Up". Det er en 3 sekunders pause for å gi agenten nok tid til å gjøre seg klar igjen for å svare på nye henvendelser, og er en viktig status å inkludere i wallboardet.

Oransje - brukt av Zisson for å indikere at en agent er i "after call work".

Denne fargen er allerede brukt av Zisson for å indikere at agenten jobber med etterarbeid etter en telefonsamtale eller skriftlig henvendelse. Den skal gi agenten mulighet til å markere at det trengs å gjøre seg ferdig med nødvendig etterarbeid.

Gul - brukt av Zisson for å indikere at en agent er i pause.

Denne fargen er allerede brukt av Zisson for å indikere at agenten er i pause. Her vil pause tilsi at agenten er borte fra plassen sin eller utilgjengelig, f.eks. i lunsjpause eller henter kaffe. Gul er assosiert med glede, energi og lykke (Palliyaguru, 2018), som også kan være en god assosiasjon på pause.

7.3.2 Andre fargevalg basert på Zisson Interact

Det ble valgt å implementere de fargene som Zisson allerede bruker i Zisson Interact. Blant annet er det brukt samme grønnfarge som allerede finnes i brukergrensesnittet til Zisson, på bruk på avhukingsbokser, radioknapper og toggleknapp. Her var det tidligere brukt mørkeblå på prototypene. Det er også valgt å bruke rød for å markere feil på format eller manglende input, da fargen rød symboliserer advarsel eller error-tilstand (Babich, 2019). Som default på bakgrunnsfarge til wallboardet er det også valgt å bruke samme farge som Zisson Interact bruker på bakrunnen sin. Dette for å skape en helhet og sammenheng i produktene til Zisson.

7.4 Overordnet design

Overordnet ble det valgt å bruke farger som Zisson allerede bruker på sine tjenester. Dette for å skape en helhet og en rød tråd for produktet som blir utviklet.

7.4.1 Font

Fonten som er valgt å bruke på wallboardet er *Open Sans*, og det er det flere grunner til. *Open Sans* ligner på *Areal*, men det er derimot mer futuristisk på grunn

av de lange slagene. Det tilbyr også god lesbarhet og dette gjør at den egner seg spesielt godt for steder som apper og nettsider (Martina, 2021). I følge Martina ser også Open Sans rent og minimalistisk ut, og slike skrifter sørger for at nettsteder ser bra ut på alle enheter (Martina, 2021).

7.4.2 Diagram

En utfordring med tabellen på wallboardet var å kunne vise alle agenter i kun ett vindu dersom det ble veldig mange agenter. Dette ble løst med å visualisere agentene gjennom et diagram, som administrator kunne velge som en mulighet i innstillinger ved oppsett av wallboardet. Diagrammet ble utformet i samarbeid og etter ønske fra produktavdelingen.

7.4.3 Feilmelding

Dersom wallboardet skulle miste tilkobling til websocket eller server var det ønskelig å gi en tydelig tilbakemelding om at agentinformasjonen ikke lenger var i sanntid. Feilmeldingen ble utformet på bakgrunn av samme design som Zisson også bruker i Zisson Interact ved feil på server eller tilkobling. Det ble valgt å ikke integrere en teller, da websocket prøver å reconnecte hvert 5. sekund, og telleren da blir nullstilt for hver gang.

7.4.5 Språk

I sluttproduktet ble det valgt å endre status språket til engelsk da det i første JSON modellen var på engelsk. Engelsk er også et internasjonalt språk og det er viktig at alle agenter kan forstå hva som står på wallboardet.



OSLOMET

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITET

Testdokumentasjon

Wallboard for Zisson

Bachelorprosjekt 2020 / 2021

Gruppe 32

Martine Holmberg s236808
Camilla H. Carlsen s197235
Synne Angell s331153
Nikola Sekiw s333975

E. TESTDOKUMENTASJON

1.0 Identifikasjon

Laget: 08.02.2021

Sist oppdatert: 12.05.2021

Hva testes her: ReWave, WaveWebAPI, WallboardService,

AgentRealTimeWallboard

Versjonsnummer av denne planen: 1.0

2.0 Innledning

Testing har vært en viktig del av prosjektet for å avdekke eventuelle feil i kritiske deler av systemet, kvalitetssikre produktet og kontrollere at systemet blir utviklet slik at det oppfyller oppdragsgivers krav.

For oppdragsgiver vil testing kunne gi en grad av tillit til systemet. Zisson har i første omgang ytret at enhetstesting er det viktigste for dem, da de har en egen spesialisert avdeling for å gjøre de større testene. Samtidig setter de pris på alt av testing som studentene har mulighet til å gjøre.

Testingen som er utført har tatt utgangspunkt i kravspesifikasjonen som er utviklet i samråd med Zlsson. Se [C. KRAVSPESIFIKASJON](#) for utdyping.

Testdokumentasjonen inneholder følgende:

- En forklaring av de ulike begrepene som er brukt
- Viser de ulike testobjektene i prosjektet
- Tidsplanlegging

- Fremgangsmåte
- Kriterier for testene
- Resultatet av testingen
- Og til slutt en oppsummering.

3.0 Begreper

Enhetstest

I enhetstester testes individuelle komponenter hver for seg. Da fokuserer man på funksjonaliteten til individuelle objekter/ metoder. Slike tester er ofte automatiserte og man tester da på forskjellig input. Testene kjøres ofte for å forsikre seg til enhver tid at alt fungerer som det skal, i tillegg til at man verifiserer at hver enhet oppfyller spesifikasjonen. Det finnes to typer enhetstester; en for normal bruk for å sjekke om komponentene oppfører seg som forventet, og en for unormal bruk, for eksempel ved å skrive inn ugyldig input for å sjekke at komponentene ikke krasjer (Spillner et al., 2014, s. 42-48).

Integrasjonstest

I integrasjonstester testes sammensatte komponenter. Det er fokus på at grensesnittet til disse komponentene oppfører seg i henhold til spesifikasjonen og man tester da hvordan deler av systemet fungerer sammen (Spillner et al., 2014, s. 50).

Systemtest

I systemtesting er det systemkravene som testes. Her ligger fokuset på interaksjonen mellom komponentene. Det sjekkes at interaksjonen er korrekt, komponentene er kompatible og at det blir overført riktig data. Det fullstendige kjørbare systemet testes som en helhet og det sjekkes at systemet oppfyller de funksjonelle og ikke-funksjonelle kravene (Spillner et al., 2014, s. 58-60).

Akseptansetest

Kunden skal verifisere at systemet når til deres forventninger og oppfyller kravene. Brukerkravene testes og tilliten til systemet etableres. Systemet skal verifiseres før levering og det bedømmes om systemet er klart for å bli installert og brukt (Spillner et al., 2014, s. 61-62).

Statisk test

Dette gjøres uten å kjøre koden som skal analyseres. Finner feil i systemet direkte med at en eller flere personer utfører en intensiv undersøkelser eller ved bruk av verktøy, istedenfor å finne dem via symptomer. Utviklerne sjekker programmeringsstandarder og forhåndsdefinerte regler (Spillner et al., 2014, s. 79).

Brukertest

Kunden skal verifisere at systemet når til deres forventninger og oppfyller kravene. Brukerkravene testes og tilliten til systemet etableres. Systemet skal verifiseres før levering og det bedømmes om systemet er klart for å bli installert og brukt (Spillner et al., 2014, s. 63-64).

4.0 Testobjekter

Testobjektene består av kode som befinner seg i alle lagene av systemet og de ulike testene vi skal utføre vil dekke flere komponenter i systemet, samhandlingen mellom komponentene samt de funksjonelle og ikke-funksjonelle kravene som er satt for oppgaven. Prosjektene som testene skal lages for er ReWave, WaveWebAPI, AgentRealTimeWallboard og WallboardService.

Brukertest

Testes på prototyper og ferdig produkt.

[Enhetstest](#)

Testes på alle kritiske komponenter og tilhørende kode i AgentRealTimeWallboard og WallboardService.

[Integrasjonstest](#)

Tester delsystemene opp mot hverandre. Dette innebærer alle de fire overnevnte prosjektene som studentene skal arbeide i og interaksjonen mellom disse.

[Systemtest](#)

Tester hele systemet i sin helhet, fra opprettelse av wallboard til det publiserte wallboardet som mottar sanntidsdata.

[Akseptansestest](#)

Tester kravspesifikasjon, prototype og inkrementer fra hver sprint.

[Statisk analyse](#)

Tester alt av arbeid som blir produsert.

5.0 Tentativ tidsplanlegging

Oppgaven	Start	Slutt	Tidsbruk	Kommentarer (hvordan)
Ferdigstille kravspesifikasjon	Uke 1	Uke 6	6 uker	Gjennom møter med Zisson.
Brukertesting av prototype	Uke 5	Uke 6	2 uker	Tester prototypen på bekjente for generelle tilbakemeldinger. Mer konkrete brukertester utføres på ansatte i Zisson.
Enhetstester	Hver utviklings sprint fra uke 7	Uke 16	10 uker	xUnit i backend, Karma og Yasmine i frontend.
Integrasjonstester	På slutten av hver sprint. Fra uke 7	Uke 16	10 uker	Testcafé.
Systemtest	Siste sprint	Uke 16	2 uker	Tester systemet i en helhet i de siste sprintene. Verktøy: Katalon..
Akseptansetest av prototyper og kravspesifikasjon	Uke 6	Uke 6	1 uke	Møte med Zisson for å få godkjent prototyper og kravspek.
Akseptansetest av inkrementer	Etter hver sprint: fra uke 7	Uke 16	10 uker	Få aksept av alle inkrementer implementert i sprintene.
Akseptansetest av sluttprodukt	Siste sprint (uke 15)	Uke 16	1 uke	Akseptansetest av MVP gjøres gjennom et siste avsluttende møte med Zisson.
Statisk analyse	Hver sprint fra uke 7	Uke 16	10 uker	Kontinuerlig gjøre statisk analyse. Gå over hverandres kode, parprogrammere, bruke IDE til å oppdage feil.

Figur 44: Tabell med arbeidsplan

6.0 Fremgangsmåte

6.1 Ikke-funksjonelle tester

Brukertest av prototype

- Et intervju/ undersøkelse har blitt utført sammen med product manager i Zisson.
- Neste brukertest har blitt utført på bekjente for å teste grunnleggende design, plassering av elementer og farger på wallboardet. Det er stilt

spørsmål vedrørende ikke-funksjonelle krav som brukervennlighet og universell utforming.

- De to neste brukertestene er blitt utført på nøkkelpersoner i Zisson som har kunnskap om produktet og om brukerne av systemet.
- En av brukertestene er basert på A/B testing som er blitt utført av bekjente for å finne ut hvordan fargene best skal kunne vises på wallboardet. A/B testing baserer seg på å vise frem to varianter av et brukergrensesnitt og måle hvem som gir best resultat (Optimizely, u.å.). A/B testingen i dette tilfellet måler resultatet ved å dokumentere svarene fra testgruppen om hvilke av de to variantene de likte best.
- Brukertestene av prototypene blir laget ved å vise frem de ulike bildene som tilhører prototypen. Det var fordelt roller som intervjuer, datamaskin, sekretær og observatør . Der de to sistnevnte produserte dokumentasjonen av testene. Intervjueren stilte åpne spørsmål som var planlagt på forhånd. Grunnet situasjonen med COVID-19 ble brukertestene gjennomført over Teams.

[Brukertest av brukervennlighet](#)

- Heuristisk analyse av prototypen.

[System Usability Scale test](#)

- Det er utført en system usability scale test for å teste brukervennlighet av det fungerende produktet etter siste sprint. Fordelene ved en slik test er at det gjør det mulig å teste og måle brukerens persepsjon av brukervennlighet (UsabiliTest, 2021).
- Testpersonene ble bedt om å rangere de følgende spørsmålene (UsabiliTest, 2021) etter følgende kriterier: strongly disagree, disagree, neutral, agree, og strongly agree.

1. I think that I would like to use this system frequently.
 2. I found the system unnecessarily complex.
 3. I thought the system was easy to use.
 4. I think that I would need the support of a technical person to be able to use this system.
 5. I found the various functions in this system were well integrated.
 6. I thought there was too much inconsistency in this system.
 7. I would imagine that most people would learn to use this system very quickly.
 8. I found the system very cumbersome to use.
 9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Resultatet ble regnet ut etter denne oppskriften:

“Your users will have ranked each of the 10 templates questions above from 1 to 5, based on their level of agreement.”

- *For each of the odd numbered questions, subtract 1 from the score.*
- *For each of the even numbered questions, subtract their value from 5.*
- *Take these new values which you have found, and add up the total score. Then multiply this by 2.5.” (Thomas, 2021).*

Figur 45 viser hvordan de forskjellige scorene ble tolket (UsabiliTest, 2021).

SUS Score	Letter Grade	Adjective Rating
Above 80.3	A	Excellent
Between 68 and 80.3	B	Good
68	C	OK
Between 51 and 67	D	Poor
Below 51	F	Awful

Figur 45: Tabell for tolkning av SUS score

6.2 Funksjonelle tester

Enhetstester

- Det planlegges at kritisk kode som produseres skal enhetstestes. Det lages automatiserte enhetstester som kan kjøres som regresjonstest ved implementasjon av ny kode.
- Testene dokumenteres ved å ta skjermbilder av enkelte av testene som produseres, samt resultatene av disse.

Integrasjonstester

- Integrasjonstestene skal teste integrasjonen mellom alle de fire prosjektene som studentene arbeider med. Så langt det lar seg gjøre skal flest mulige utfall testes, og flest mulige tester automatiseres. Testene med resultater dokumenteres i excel.

Systemtest

- Det planlegges at det skal testes på alle brukerhistoriene som er definert i kravspesifikasjonen. Det er de funksjonelle kravene som skal bli testet, da de ikke-funksjonelle kravene i kravspesifikasjonen blir dekket av brukertester.

- Testene med resultater dokumenteres i excel og med skjermbilder fra de automatiserte testene i Katalon.

6.3 Akseptansetest

Akseptanseteste kravspesifikasjon

Kravspesifikasjonen blir akseptansetestet gjennom at den blir utarbeidet sammen med oppdragsgiver. All dokumentasjon av kravspesifikasjon og valgte krav for hver sprint dokumenteres med verktøyet Jira, i tillegg til at det legges i kravspesifikasjonen under [punkt 3.0 Endelig kravspesifikasjon](#).

Akseptansetest prototypen

Prototypen godkjennes av oppdragsgiver gjennom møter der vi viser frem prototypen. Dokumenteres med møtereferat og under [punkt 8.5.2, Akseptansetest av prototype](#).

Akseptanseteste inkrementer

På hvert sprint review-møte skal inkrementene som er implementert presenteres og eventuelt godkjennes. Dokumenteres i Jira og med møtereferat, samt som vedlegg under [punkt 8.5.3 Akseptansetest av inkrementer](#).

Akseptanseteste sluttproduktet

Det holdes et avsluttende møte der sluttproduktet presenteres og Zisson gir tilbakemelding på hvorvidt de godkjener arbeidet studentene har gjort eller ikke. Dokumenteres gjennom møtereferat og under [punkt 8.5.4, Akseptansetest av sluttprodukt](#).

6.4 Statisk analyse

Det skal kontinuerlig gjennomføres statisk analyse av koden som produseres. Dette gir kvalitetssikring gjennom at studentene i prosjektet gjennomgår hverandres kode. Funkjonsnavn, klassenavn og skrivefeil blir gjennomgått. I tillegg vil programvaren det produseres kode i sjekke automatisk for kodefeil underveis i utviklingen, og det lastes ned et tillegg i “TSLint” som er en veileder for hvordan kode skal skrives i Angular.

7.0 Kriterier og miljø for testene

7.1 Brukertest

Kriterier for at testene kan kjøres

At det er laget prototyper og spørsmål knyttet til disse.

Kriterier for at testene er kjørt

Når planlagte brukertester er utført og det er mottatt tilbakemeldinger på prototypene som igjen kan brukes for å gjøre nye endringer.

Kriterier for grundigheten av testene

Utvalgt funksjonalitet må testes på brukere av systemet.

Godkjenningskriterier

Brukertestene er godkjent når de er gjennomført med testpersonene og testene er skriftlig dokumentert. Det skal videre gjøres en vurderingen på alle tilbakemeldingene som kommer frem under testingen. Noen av brukertestene kan komme med motstridende tilbakemeldinger, og da skal dette tas opp med oppdragsgiver for å komme til enighet om hvilke endringer som skal gjøres.

Testmiljø

- **Prototype**

Prototypene utvikles i Miro, samt digitale workshops som utføres i samråd med produktansvarlig i Zisson i forbindelse med disse. De videre brukertestene av prototypene utføres ved bruk av powerpoint der bilder og utsnitt av wallboardet vises frem. Bakgrunnen for dette valget er at det ikke skal testes interaksjon, men statisk design. Prototypen for delene av systemet som er i Zisson Interact skal ikke brukertestes, da oppsettet skal være likt som systemet er fra før, og det er ikke ønsker om at dette skal endres på.

- **Inkrement**

Teamsmøte med skjermdeling.

- **Ferdig produkt**

Teamsmøte med skjermdeling.

7.2 Enhetstest

Kriterier for at testene kan kjøres

Så fort kode og enhetstest er produsert kan testene kjøres.

Kriterier for at testene er kjørt

Resultat må vises på alle testene i xUnit og Karma.

Kriterier for grundigheten av testene

Enhetstestene skal dekke de delene av systemet der det er laget metoder med logikk. Det er ikke nødvendig å teste de delene av systemet der data kun blir sendt til/ fra, uten at noe mer logikk skjer, da feil i de delene vil oppdages fort ved integrasjons- og systemtest. Fokuset skal være å enhetsteste koden i prosjektene tilknyttet wallboardet: AgentRealTimeWallboard og WallboardService. Prosjektene

for ReWave er tidkrevende å enhetsteste da det er mange avhengigheter som går utenom koden studentene skal produsere, derfor blir dette prioritert vakk.

Godkjenningskriterier

- 90% av de produserte testene må være godkjente.
- Metodene med logikk som befinner seg i prosjektene
AgentRealTimeWallboard og WallboardService har blitt testet.

Testmiljø

- Backend: xUnit
- Frontend: Karma og Jasmine

7.3 Integrasjonstest

Kriterier for at testene kan kjøres

- Modulene som skal testes er ferdig integrert.
- Testplan, testscenarier og testtilfeller er satt opp i excel.
- Testmiljøet er satt opp og klart.

Kriterier for at testene er kjørt

Alle testtilfeller må ha fått et testresultat i TestCafé.

Kriterier for grundigheten av testene

All funksjonalitet knyttet til brukerhistorier laget i kravspesifikasjonen skal testes.

Testmiljø

- Testcafe
- Manuelle tester

7.4 Systemtest

Kriterier for at testene kan kjøres

Systemtesten kan lages når alle brukerhistorier tilhørende MVP har blitt implementert gjennom sprinter og satt opp i et testdokument i excel. Testene skal settes opp i Katalon.

Kriterier for at testene er kjørt

Alle brukerhistoriene må ha fått et testresultat.

Kriterier for grundigheten av testene

Alle brukerhistorier laget i kravspesifikasjonen skal testes.

Godkjenningskriterier

Godkjenningskriteriene for integrasjons- og systemtest er godkjenningskriterier som studentene tidligere har brukt i faget “Testing av programvare” utdelt fra læreren i faget (T. Krattebøl, personlig kommunikasjon, 2020).

Godkjenningskriterier ved integrasjons- og systemtest

Testene ansees godkjent når 90% av de planlagte tester er utført og det ikke finnes noen åpne A og B feil.

Ved rapportering av feil, skal feil og avvik kategoriseres. Dette vil bli brukt som underlag for godkjennings- og underkjenningskriterier av testene. Følgende kategorier kan f.eks. benyttes for feil og avvik:

- Kategori A: Feil i en eller flere funksjoner som forårsaker stopp. For eksempel maskin/ system/ funksjon må startes på nytt.
- Kategori B: Feil har alvorlige konsekvenser i systemet eller tilstøtende systemer, for eksempel feil i databaser, alvorlige feil i rapporter slik at disse tolkes feil etc.
- Kategori C: Feil med mindre alvorlige konsekvenser, for eksempel mindre grad av ustabilitet, mindre funksjons-/ egenskapsmangler etc.

- Kategori D: Feil uten alvorlige konsekvenser, for eksempel layout-/ trykkfeil i skjermbilder, rapporter eller dokumentasjon som ikke har konsekvenser for forståelsen av disse.

Testmiljø

- Katalon
- Manuelle tester

7.5 Akseptansetest

Kriterier for at testene kan kjøres

- **Prototype**

Når brukertester er ferdig og endringer er gjort med tanke på tilbakemeldinger, kan prototypen akseptansetestes.

- **Inkrementer**

Sprint review etter hver sprint, hvor man går igjennom alt som har blitt utviklet og får aksept eller tilbakemeldinger på endringer som må gjøres til neste sprint.

- **Sluttpunkt**

Når alle krav fra kravspesifikasjon er implementert i produktet og alle andre testnivåer er utført kan sluttpunktet akseptansetestes.

Kriterier for at testene er kjørt

Tilbakemelding fra oppdragsgiver på hvorvidt testobjektet blir akseptert eller ikke.

Kriterier for grundigheten av testene

- **Prototype**

Den ferdige prototypen må vises frem for Zisson.

- **Inkrement**

Alle inkrementer må vises frem i møter med Zisson.

- **Sluttprodukt**

Hele sluttproduktet må vises frem for Zisson.

[Godkjenningskriterier ved integrasjons- og systemtest](#)

Godkjenningskriteriene for akseptansetestene er at oppdragsgiver godkjenner testobjektene.

[Testmiljø](#)

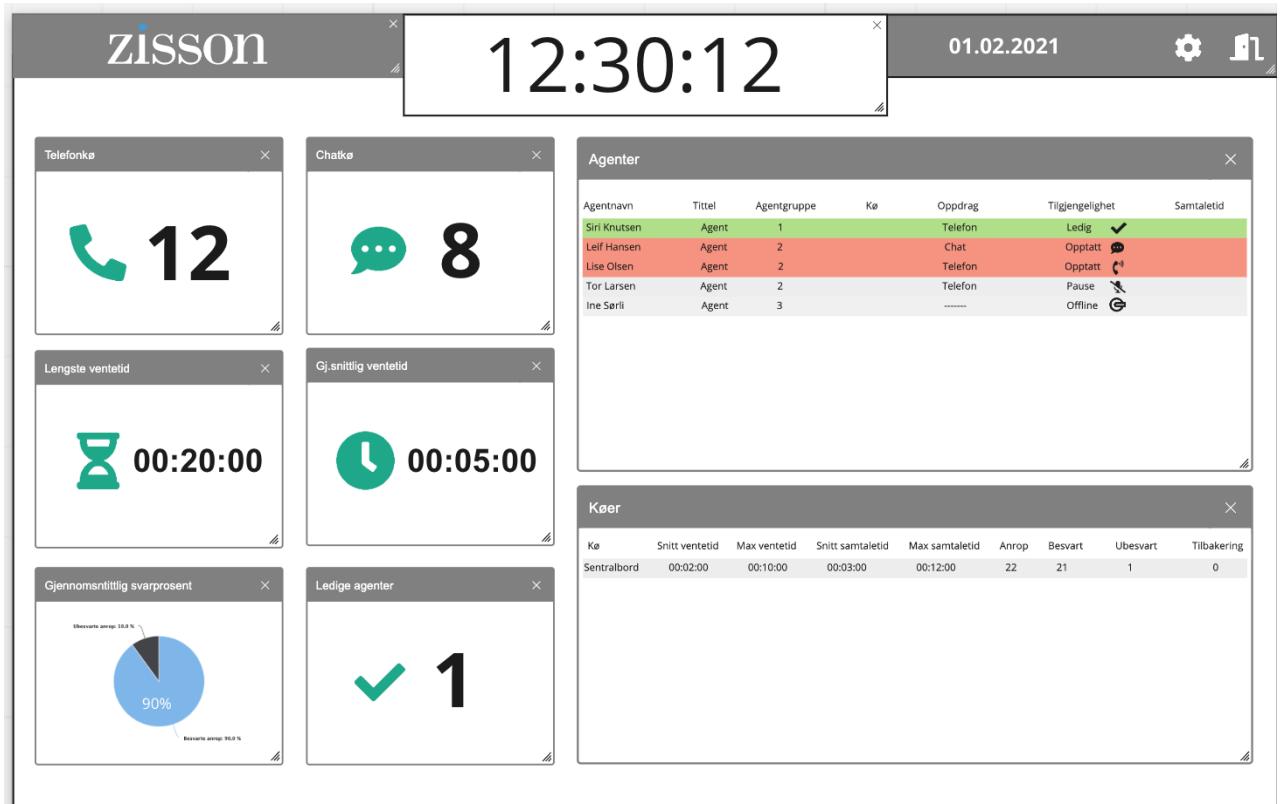
Teams med skjermdeling

[8.0 Resultater](#)

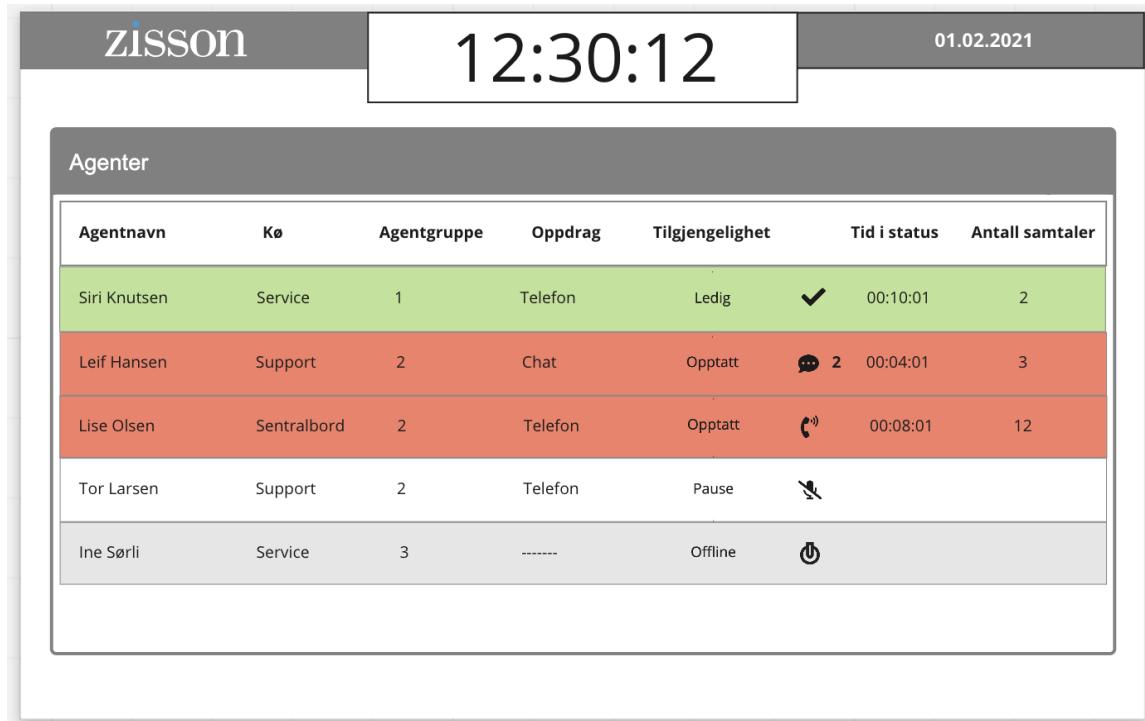
[8.1 Brukertester](#)

[8.1.1 Intervju/ samtale om første prototype](#)

Figur 46 og 47 viser bilde av de første prototypene.



Figur 46: Første prototype med flere elementer.



Figur 47: Første prototype med agentoversikt.

Se spørsmålene vedrørende brukertestene, punkt [4.3.1 Intervju/samtale om første prototype](#), i vedleggene.

Tilbakemeldinger og endringer

- Unødvendig med bruk av ikoner, da de ikke bidrar med noen beskrivelse.
- Fokus på at wallboardet skal være mulig å se langt unna. Derfor viktig med store kontraster.
- Kø-attributten måtte bytte navn til profil istedenfor, fordi en profil kan ha mange køer.
- Bytte navn fra "chat" til "skriftlige henvendelser" ettersom det kan være flere typer "chat" - feks e-mail, facebook, instagram, twitter, kundeservice, etc.
- Det måtte skilles bedre mellom skriftlig henvendelser og telefonsamtaler:
Der telefonsamtaler kunne defineres som opptatt, mens skriftlige henvendelser ikke kan defineres som opptatt, da man kan håndtere flere skriftlige henvendelser på en gang og over lengre tid, men likevel være tilgjengelig for flere.
- Det ble også gjort endringer på layout og farger.

8.1.2 Første brukertest - MVP

I figur 48 - 51 vises bilder av prototypene ved første brukertest.

Endret layout og integrert en innstillingermeny
for filtrering av agentgrupper

1. Oppsett filtrering før agentoversikt-> trykk på innstillinger

12:30:12 01.02.2021

Agenter

Agentnavn Profil Agentgruppe Status Tid i status Besvart Mottatt skriftlig Gjennomsnittlig samtaletid Gjennomsnittlig ringetid

Valg Agentgruppe

- Support
- Salg
- Service
- Sentralbord

Figur 48: Tom agentoversikt med pil som peker på innstillinger-knappen.

3. Oppsett filtrering før agentoversikt

12:30:12

Agenter

Agentnavn Profil Agentgruppe Status Tid i status Besvart Mottatt skriftlig Gjennomsnittlig samtaletid Gjennomsnittlig ringetid

Valg Agentgruppe

- Support
- Salg
- Service
- Sentralbord

Figur 49: Tom agentoversikt med innstillinger for agentgrupper til høyre.

Implementert nye statusfarger, samt fjernet ikoner

4. Wallboard etter oppsett-> trykket på innstillinger (Få agenter)

The screenshot shows the 'Agenter' (Agents) section of a dashboard. At the top right, there is a small 'Valg' (Selection) button. A red arrow points from the text 'Implementert nye statusfarger, samt fjernet ikoner' to this button. A modal window titled 'Valg' is open, showing a list of 'Agentgruppe' (Agent Group) names. Some names have a green checkmark next to them, indicating they are selected. A teal box on the right contains the text 'Oppdatering in real time' (Real-time update). The main table below shows agent details like name, profile, group, status, and duration.

Agentnavn	Profil	Agentgruppe	Status	Tid i status	Besvart	Mottatt skriftlig	Max samtale
Lise Johansen	Sentralbord	Sentralbord	PAUSE	00:04:10	4	4	00:04:10
Bent Stiansen	Kjøkken	Kjøkken	BEHANDLER	00:04:10	4	4	00:04:10
Thomas Alsgaard	Vintersport	Ski	OPPTATT	00:04:10	4	4	00:04:10
Jenny Skavland	Mote	Smykker og klær	ACW	00:04:10	4	4	00:04:10
Espen Lind	Musikk	Guitar	LEDIG	00:04:10	4	4	00:04:10
Kurt Nilsen	Musikk	Guitar	WRAPUP	00:04:10	4	4	00:04:10

Figur 50: Agentoversikt med valgte agentgrupper til høyre.

(5). Mange agenter-> uten innstillinger

The screenshot shows the 'Agenter' section with a different set of agents. The table now includes additional columns: 'Gjennomsnittlig samtaletid' (Average call duration) and 'Gjennomsnittlig ringetid' (Average ring time). The agents listed are Christian Strand, Lise Johansen, Bent Stiansen, Thomas Alsgaard, Jenny Skavland, Espen Lind, Kurt Nilsen, Lise Johansen, Bent Stiansen, Thomas Alsgaard, Jenny Skavland, Espen Lind, Kurt Nilsen, and Espen Lind. All agents are currently in the 'BEHANDLER' (Handling) status.

Agentnavn	Profil	Agentgruppe	Status	Tid i status	Besvart	Mottatt skriftlig	Gjennomsnittlig samtaletid	Gjennomsnittlig ringetid
Christian Strand	TV	Programleder	BEHANDLER	00:04:10	4	4	00:04:10	00:04:10
Lise Johansen	Sentralbord	Sentralbord	PAUSE	00:04:10	4	4	00:04:10	00:04:10
Bent Stiansen	Kjøkken	Kjøkken	BEHANDLER	00:04:10	4	4	00:04:10	00:04:10
Thomas Alsgaard	Vintersport	Ski	OPPTATT	00:04:10	4	4	00:04:10	00:04:10
Jenny Skavland	Mote	Smykker og klær	ACW	00:04:10	4	4	00:04:10	00:04:10
Espen Lind	Musikk	Guitar	LEDIG	00:04:10	4	4	00:04:10	00:04:10
Kurt Nilsen	Musikk	Guitar	WRAPUP	00:04:10	4	4	00:04:10	00:04:10
Lise Johansen	Sentralbord	Sentralbord	PAUSE	00:04:10	4	4	00:04:10	00:04:10
Bent Stiansen	Kjøkken	Kjøkken	BEHANDLER	00:04:10	4	4	00:04:10	00:04:10
Thomas Alsgaard	Vintersport	Ski	OPPTATT	00:04:10	4	4	00:04:10	00:04:10
Jenny Skavland	Mote	Smykker og klær	ACW	00:04:10	4	4	00:04:10	00:04:10
Espen Lind	Musikk	Guitar	LEDIG	00:04:10	4	4	00:04:10	00:04:10
Kurt Nilsen	Musikk	Guitar	WRAPUP	00:04:10	4	4	00:04:10	00:04:10
Espen Lind	Musikk	Guitar	LEDIG	00:04:10	4	4	00:04:10	00:04:10

Figur 51: Agentoversikt med aktive agenter.

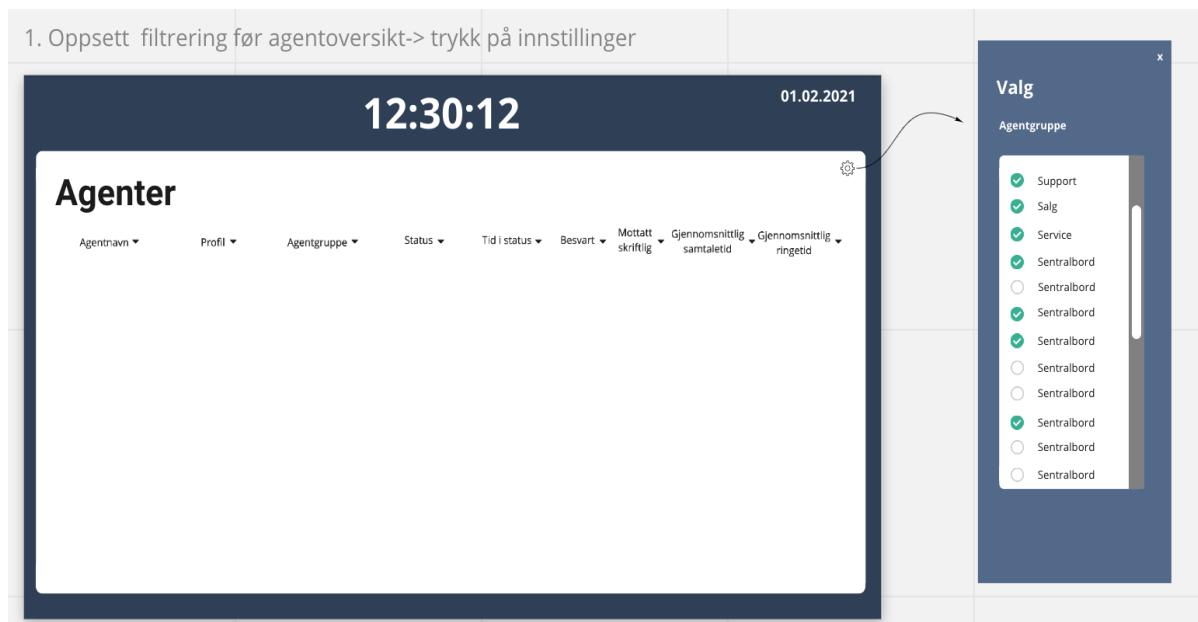
Se spørsmålene vedrørende brukertestene, punkt [4.3.2 Første brukertest - MVP](#), i vedleggene.

Tilbakemeldinger og endringer

- Få tydeligere frem filtrering, innenfor valg menyen
- Valg for å kunne velge alle/fjerne alle valg
- Liten skriftstørrelse, øke denne ut i fra antall agenter i koden
- Vise sorteringspil kun på én kolonne av gangen og vise en svak pil hvis man holder over en annen kolonne man kan sortere på.
- Sortere på kun agentnavn, profil og agentgruppe

8.1.3 Heuristisk evaluering av MVP prototype

Figur 52 - 55 viser bilde av videre utarbeidelse av prototypene.



Figur 52: Tom agentoversikt med pil som peker på innstillinger-knappen.

3. Oppsett filtrering før agentoversikt

The screenshot shows the 'Agenter' (Agents) section of a dashboard. At the top, there is a dark header with the time '12:30:12'. Below it is a white header bar with dropdown menus for 'Agentnavn', 'Profil', 'Agentgruppe', 'Status', 'Tid i status', 'Besvart', 'Mottatt skriftlig', and 'Gjennomskriv'. To the right of the header is a modal window titled 'Valg' (Selection) with the heading 'Agentgruppe'. The list contains several entries, each with a checked checkbox and a name: Support, Salg, Service, Sentralbord, and multiple entries for Sentralbord. A vertical scrollbar is visible on the right side of the modal.

Figur 53: Tom agentoversikt med innstillingen for agentgrupper til høyre.

4. Wallboard etter oppsett-> trykket på innstillingen (Få agenter)

The screenshot shows the same 'Agenter' section after applying the filter. The table now displays seven rows of agent data. Each row includes the agent's name, department, current status, duration, and other metrics. To the right of the table is the same 'Valg' (Selection) modal as in Figure 53. A large green callout box with the text 'Oppdatering in real time' (Real-time update) has an arrow pointing towards the modal, indicating that the selection will refresh the data in real time.

Agentnavn	Profil	Agentgruppe	Status	Tid i status	Besvart	Mottatt skriftlig	Max samtaletid
Lise Johansen	Sentralbord	Sentralbord	PAUSE	00:04:10	4	4	00:04:10
Bent Stiansen	Kjøkken	Kjøkken	BEHANDLER	00:04:10	4	4	00:04:10
Thomas Alsgaard	Vintersport	Ski	OPPTATT	00:04:10	4	4	00:04:10
Jenny Skavland	Mote	Smykker og klær	ACW	00:04:10	4	4	00:04:10
Espen Lind	Musikk	Guitar	LEDIG	00:04:10	4	4	00:04:10
Kurt Nilsen	Musikk	Guitar	WRAPUP	00:04:10	4	4	00:04:10

Figur 54: Agentoversikt med valgte agentgrupper til høyre.

(5). Mange agenter-> uten innstillinger



Figur 55: Agentoversikt-vindu med aktive agenter.

Heuristisk evaluering handler om å evaluere designet til et brukergrensesnitt, ved å legge vekt på anbefalte retningslinjer (Rosenfield et al., 2015, s. 324). Personene som evaluerer har som oppgave å identifisere utfordringer og kartlegge forbedringspotensialet ved designet.

Retningslinjene som ble fulgt (Nielsen Norman Group, 1994) og funnene

1. Synlighet og systemstatus

- Ha med tannhjulet i menyen etter at tannhjulet har blitt trykket på, på selve wallboardet.
- En tilbakemelding om hvilke agentgrupper som har blitt valgt som vises på siden eller under tabellen over agentgrupper, slik at en kan se en oversikt over valgt agentgrupper. Kan da også vises en feilmelding hvis ingen agentgrupper har blitt valgt.

2. Match mellom systemet og den virkelige verden

- Ord og konsepter som er brukt i prototypen er brukt bevisst med tanke på at dette er brukt i Zisson sine produkter tidligere og er kjent for kunden. Derfor er det også valgt noen ord som er engelsk, da dette er valgt fra tidligere av hos Zisson.
- Eneste nye som er introdusert er fargene rød og blå som status på agent. Samt statusen med navn “Behandler”.

3. Brukerkontroll og frihet

- Flere muligheter for å krysse seg ut av innstillinger boksen. En mulighet er å ha “avbryt” knapp, men da er det også nødvendig med “lagre” og ting man da velger kan ikke lenger vises dynamisk på wallboardet, men heller etter endringene som er gjort er lagret.

4. Konsistens og standarder

- Teksten “Alle” inne på valg kan med fordel byttes til “Velg alle” for en bedre forståelse.

5. Feil forebygging

- Det ble oppdaget at det ikke var noen feilmelding til brukeren hvis det ikke var valgt noen agentgrupper. Et forslag var å implementere dette.
- Det var ikke laget noen fatal feilmelding for brukeren dersom dette skulle skje.

6. Anerkjennelse i stedet for å huske

- I innstillinger er agentgruppene synlige for brukeren. Brukeren kan derfor velge de agentgruppene som de kjenner igjen, og dette gjør at man kan bruke “recognition” istedenfor “recall”.

- I innstillinger er det også forklart hva man skal gjøre: "Velg filtrering for agentgruppe".
- Pil-ikonets retning kan være vanskelig å forstå ift. sorteringsretning, og hvordan brukeren skal kunne velge en annen kolonne. Det er tenkt at dette skal løses med å holde "musen" over kolonnen for å få opp en pil i den ønskede kolonnen.
- Fargene på radene kan være vanskelig for brukerne å forstå. Dette gjelder sammenhengen mellom status og farge. Fargene er også elementer ved designet som brukeren må lære/ huske. Fargevalgene kommer til å være forklart i en egen brukerdokumentasjon.
- Det er ikke lagre knapp i innstillingen. Tanken bak er at lagringen skal skje "in real time" og at ønsket sortering skal dukke opp med en gang.

7. Fleksibilitet og effektivitet i bruken

- Vurderingen av dette er at systemet er så lite at det ikke er nødvendig å effektivisere systemet ytterligere.

8. Estetisk og minimalistisk design

- Sette inn enten en farge eller gjøre innstillinger-ikonet større når man kjører over ikonet med musa (on hover).

9. Hjelp brukere å gjenkjenne, diagnostisere og gjenopprette feil

- Få inn en varsemelding hvis f.eks. sanntidsdata ikke kan vises ved feil på server- eller websocket-tilkobling.

10. Hjelp og dokumentasjon

- En mulig løsning er å lage en brukermanual slik at brukerne av systemet kan lese der hvis de føler at noe er uklart, eller for nye brukere som potensielt ikke er like godt kjente med systemet.

8.1.4 Andre runde brukertest - MVP

Figur 56 - 59 viser bilder over prototypene ved andre runde brukertester med fokus på MVP.

Innstillinger-ikonet er flyttet på

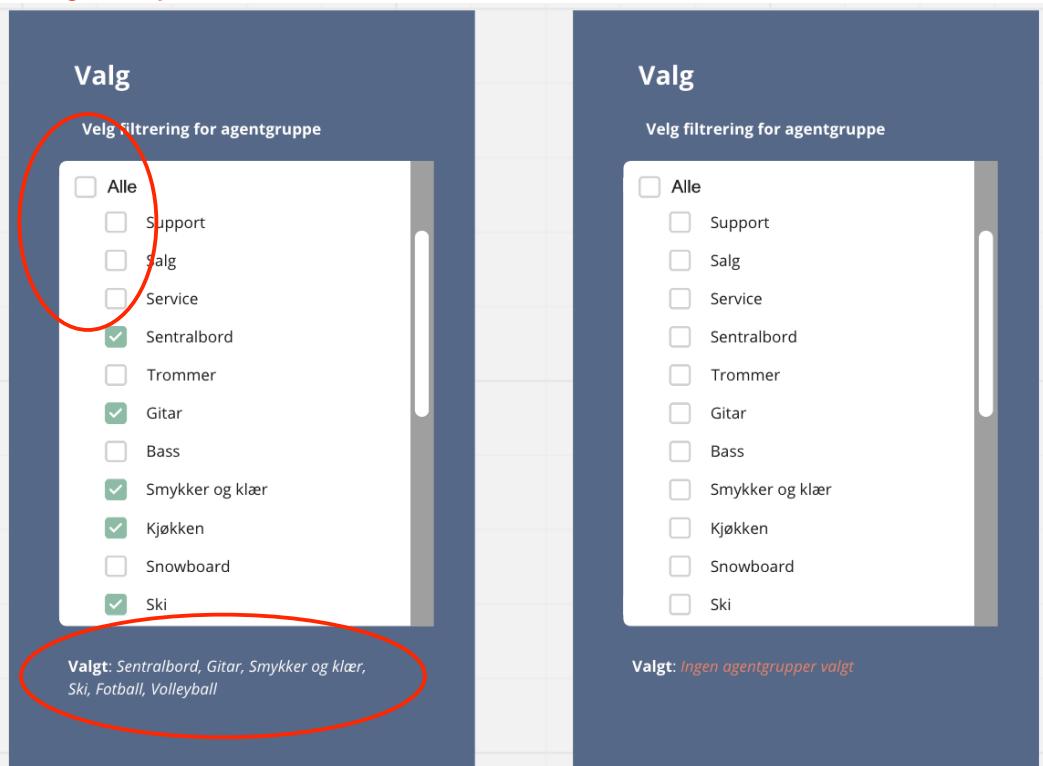
12:30:12
01.02.2021

Agenter

Agentnavn ↑	Profil	Agentgruppe	Status	Tid i status	Besvart	Mottatt skriftlig	Gjennomsnittlig samtaletid	Gjennomsnittlig ringetid
Du må velge minst én agentgruppe i innstillinger:								

Figur 56: Tom agentoversikt.

Endret fra runde til firkantede
check-boxer, samt integrert innrykk



Implementert tilbakemelding til
bruker

Figur 57: To forslag til innstillinger for valg av agentgrupper

Agentnavn	Profil	Agentgruppe	Status	Tid i status	Besv.
Christian Strand	Vintersport	Ski	BEHANDLER	00:04:10	4
Lise Johansen	Vintersport	Skiskyting	PAUSE	00:04:10	4
Bent Stiansen	Vintersport	Vintersport	BEHANDLER	00:04:10	4
Thomas Alsgaard	Vintersport	Ski	OPPTATT	00:04:10	4
Jenny Skavland	TV	TV	ACW	00:04:10	4
Espen Lind	Sentralbord	Sentralbord	LEDIG	00:04:10	4
Kurt Nilsen	Musikk	Gitar	WRAPUP	00:04:10	4
Lise Johansen	Musikk	Gitar	PAUSE	00:04:10	4
Kristian Eriksen	Musikk	Piano	BEHANDLER	00:04:10	4
Stian Staysman	Musikk	Sanger	OPPTATT	00:04:10	4
Tom Eriksen	Musikk	Gitar	ACW	00:04:10	4
Solveig Kloppen	Musikk	Gitar	LEDIG	00:04:10	4
Maria Mena	Kjøkken	Kjøkken	WRAPUP	00:04:10	4
Erlend Haaland	Fotball	Fotball	LEDIG	00:04:10	4

Figur 58: Agentoversikt med valgte agentgrupper til høyre.

Agentnavn	Profil	Agentgruppe	Status	Tid i status	Besvart	Mottatt skriftlig	Gjennomsnittlig samtaletid	Gjennomsnittlig ringetid
Christian Strand	Vintersport	Ski	BEHANDLER	00:04:10	4	4	00:04:10	00:04:10
Lise Johansen	Vintersport	Skiskyting	PAUSE	00:04:10	4	4	00:04:10	00:04:10
Bent Stiansen	Vintersport	Vintersport	BEHANDLER	00:04:10	4	4	00:04:10	00:04:10
Thomas Alsgaard	Vintersport	Ski	OPPTATT	00:04:10	4	4	00:04:10	00:04:10
Jenny Skavland	TV	TV	ACW	00:04:10	4	4	00:04:10	00:04:10
Espen Lind	Sentralbord	Sentralbord	LEDIG	00:04:10	4	4	00:04:10	00:04:10
Kurt Nilsen	Musikk	Gitar	WRAPUP	00:04:10	4	4	00:04:10	00:04:10
Lise Johansen	Musikk	Gitar	PAUSE	00:04:10	4	4	00:04:10	00:04:10
Kristian Eriksen	Musikk	Piano	BEHANDLER	00:04:10	4	4	00:04:10	00:04:10
Stian Staysman	Musikk	Sanger	OPPTATT	00:04:10	4	4	00:04:10	00:04:10
Tom Eriksen	Musikk	Gitar	ACW	00:04:10	4	4	00:04:10	00:04:10
Solveig Kloppen	Musikk	Gitar	LEDIG	00:04:10	4	4	00:04:10	00:04:10
Maria Mena	Kjøkken	Kjøkken	WRAPUP	00:04:10	4	4	00:04:10	00:04:10
Erlend Haaland	Fotball	Fotball	LEDIG	00:04:10	4	4	00:04:10	00:04:10

Figur 59: Agentoversikt-vindu med aktive agenter.

Se spørsmålene vedrørende brukertestene, punkt [4.4 Andre runde brukertest - MVP](#), i vedleggene.

Tilbakemeldinger og endringer

Ut i fra andre brukertest kom det frem nye synspunkt fra testpersonen:

- Testpersonen refererte til excel, og ønsket en mer komprimert tabell lik som excel har.
- Fargene ble for dominerende og var ikke nødvendigvis så viktig å få fremhevet så mye som det hadde blitt gitt uttrykk for fra tidligere samtaler med Zisson.
- Ønsket mulighet for å kunne søke i agentgrupper ved filtrering og et ønske om at systemet ligner mest mulig på det allerede eksisterende systemet deres.
- Ønsket en mer grafisk fremstilling av agentene ved mange agenter.

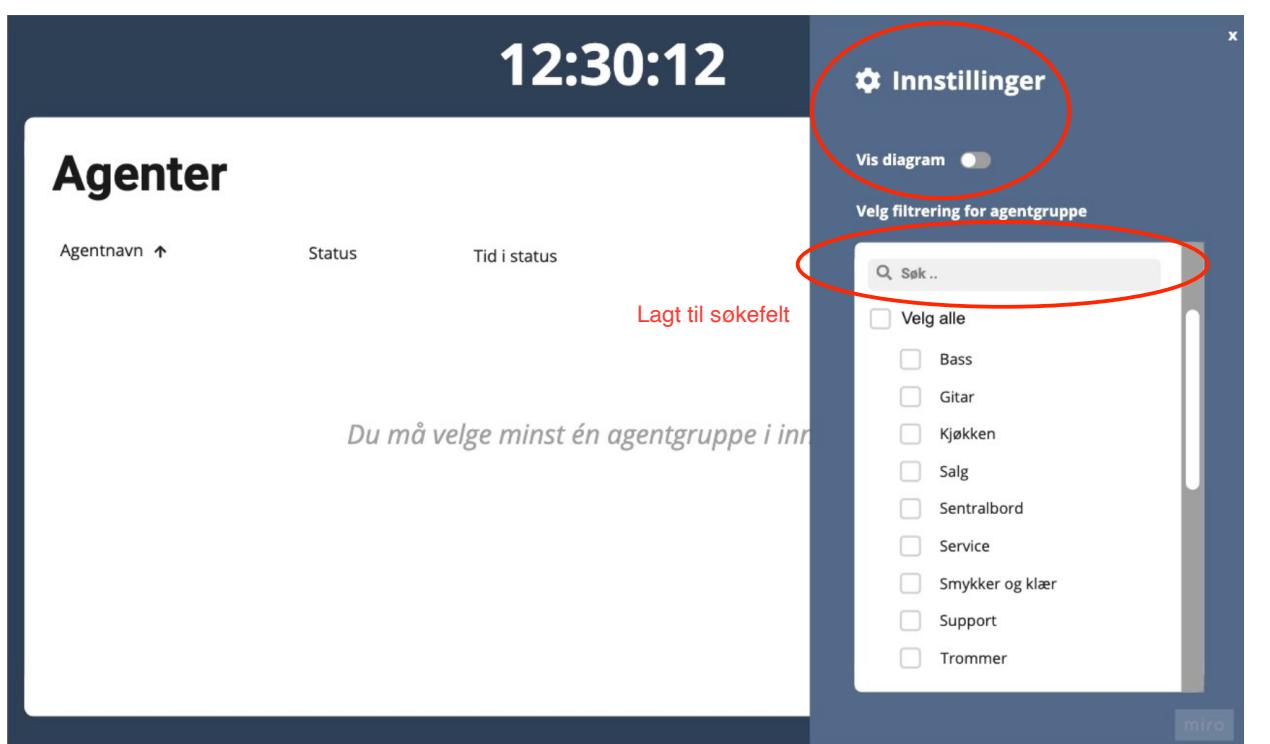
8.1.5 Tredje runde brukertester - MVP

Figur 60-70 viser bilder av prototypene ved tredje runde brukertester med fokus på MVP.

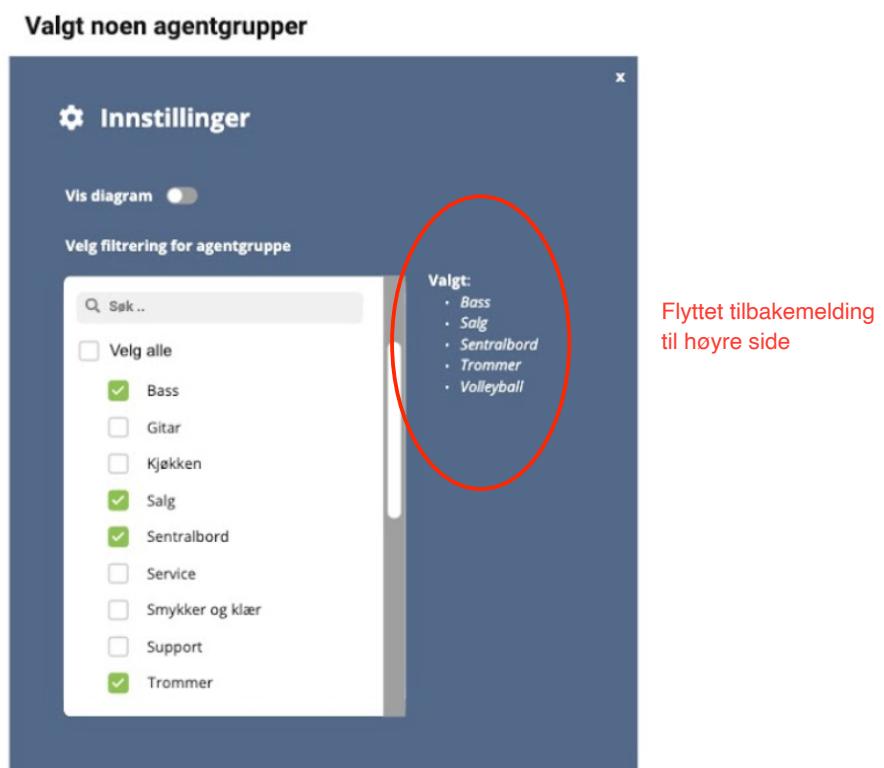
The screenshot shows a digital interface with a dark blue header bar. In the top right corner, it displays the date '01.02.2021' and a gear icon. The main content area has a white background and a dark blue header labeled 'Agenter'. Below the header is a table with six columns: 'Agentnavn ↑', 'Status', 'Tid i status', 'Profil', 'Besvart', and 'Mottatt skriftlig'. A red oval highlights the first two columns ('Agentnavn ↑' and 'Status'). In the bottom right corner of the table area, there is a red text message: 'Endret på kolonner' (Changed columns). Below the table, a message reads: 'Du må velge minst én agentgruppe i innstillinger:' followed by a gear icon. The bottom right corner of the interface features a small 'miro' logo.

Agentnavn ↑	Status	Tid i status	Profil	Besvart	Mottatt skriftlig
Endret på kolonner					

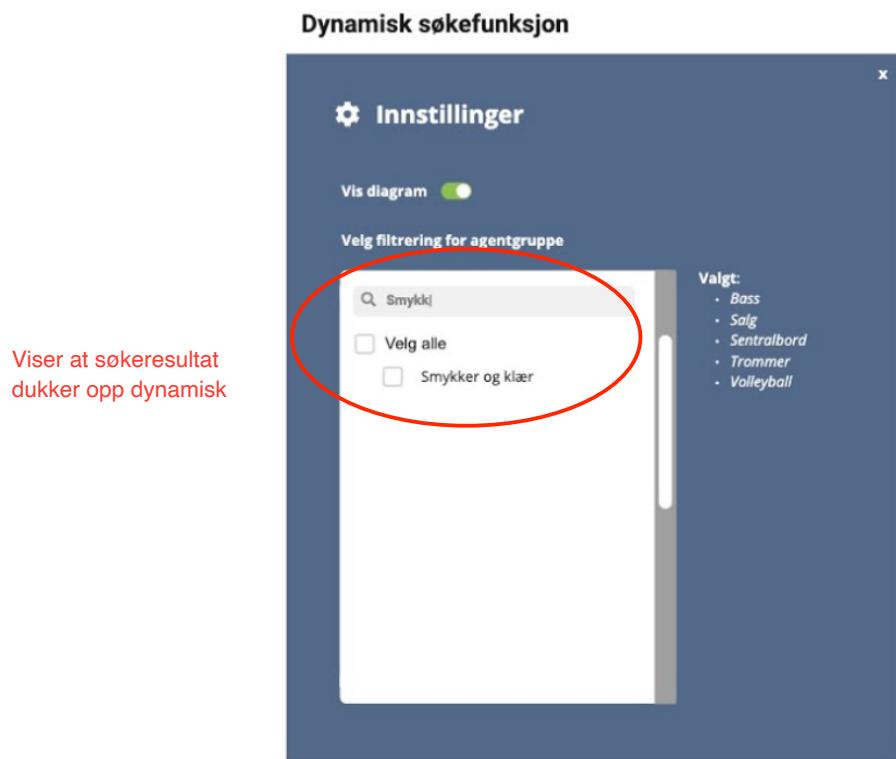
Figur 60: Tom agentoversikt



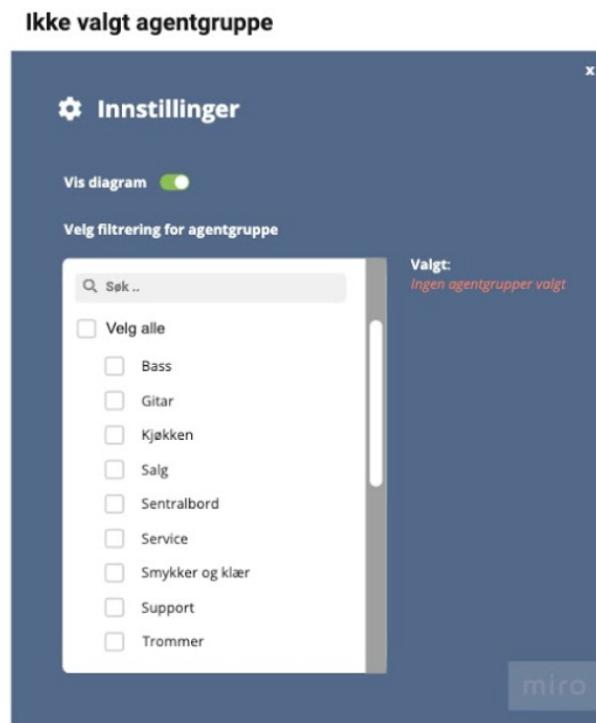
Figur 61: Tom agentoversikt med innstillinger til høyre



Figur 62: Innstillinger med valgte agentgrupper til høyre



Figur 63: Innstillinger med filtrert søk



Figur 64: Innstillinger med tilbakemelding til høyre om ingen valgte agentgrupper.

12:30:12

01.02.2021 

Agenter

Agentnavn	Status ↑	Tid i status	Profil	Besvart	Mottatt skriftlig
Christian Strand	LEDIG	00:04:10	Vintersport	4	4
Lise Johansen	LEDIG	00:04:10	Vintersport	4	4
Bent Stiansen	LEDIG	00:04:10	Vintersport	4	4
Thomas Alsgaard	BEHANDLER	00:04:10	Vintersport	4	4
Jenny Skavland	BEHANDLER	00:04:10	TV	4	4
Espen Lind	BEHANDLER	00:04:10	Sentralbord	4	4
Kurt Nilsen	WRAPUP	00:00:03	Musikk	4	4
Lise Johansen	ACW	00:04:10	Musikk	4	4
Kristian Eriksen	ACW	00:04:10	Musikk	4	4
Stian Staysman	PAUSE	00:04:10	Musikk	4	4
Tom Eriksen	OPPTATT	00:04:10	Musikk	4	4
Solveig Kloppen	OPPTATT	00:04:10	Musikk	4	4
Maria Mena	OPPTATT	00:04:10	Kjøkken	4	4
Erlend Haaland	OPPTATT	00:04:10	Fotball	4	4

Figur 65: Agentoversikt med farge over hele raden

12:30:12

01.02.2021 

Agenter

Agentnavn	Status ↑	Tid i status	Profil	Agentgruppe	Besvart	Mottatt skriftlig	Gjennomsnittlig samtaletid	Gjennomsnittlig behandlingstid
Christian Strand	LEDIG	00:04:10	Vintersport	Ski	4	4	00:04:10	00:04:10
Lise Johansen	LEDIG	00:04:10	Vintersport	Skiskyting	4	4	00:04:10	00:04:10
Bent Stiansen	LEDIG	00:04:10	Vintersport	Vintersport	4	4	00:04:10	00:04:10
Thomas Alsgaard	BEHANDLER	00:04:10	Vintersport	Ski	4	4	00:04:10	00:04:10
Jenny Skavland	BEHANDLER	00:04:10	TV	TV	4	4	00:04:10	00:04:10
Espen Lind	BEHANDLER	00:04:10	Sentralbord	Sentralbord	4	4	00:04:10	00:04:10
Kurt Nilsen	WRAPUP	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Lise Johansen	ACW	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Kristian Eriksen	ACW	00:04:10	Musikk	Piano	4	4	00:04:10	00:04:10
Stian Staysman	PAUSE	00:04:10	Musikk	Sanger	4	4	00:04:10	00:04:10
Tom Eriksen	OPPTATT	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Solveig Kloppen	OPPTATT	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Maria Mena	OPPTATT	00:04:10	Kjøkken	Kjøkken	4	4	00:04:10	00:04:10
Erlend Haaland	OPPTATT	00:04:10	Fotball	Fotball	4	4	00:04:10	00:04:10

Figur 66: Agentoversikt med farge over status-kolonnen

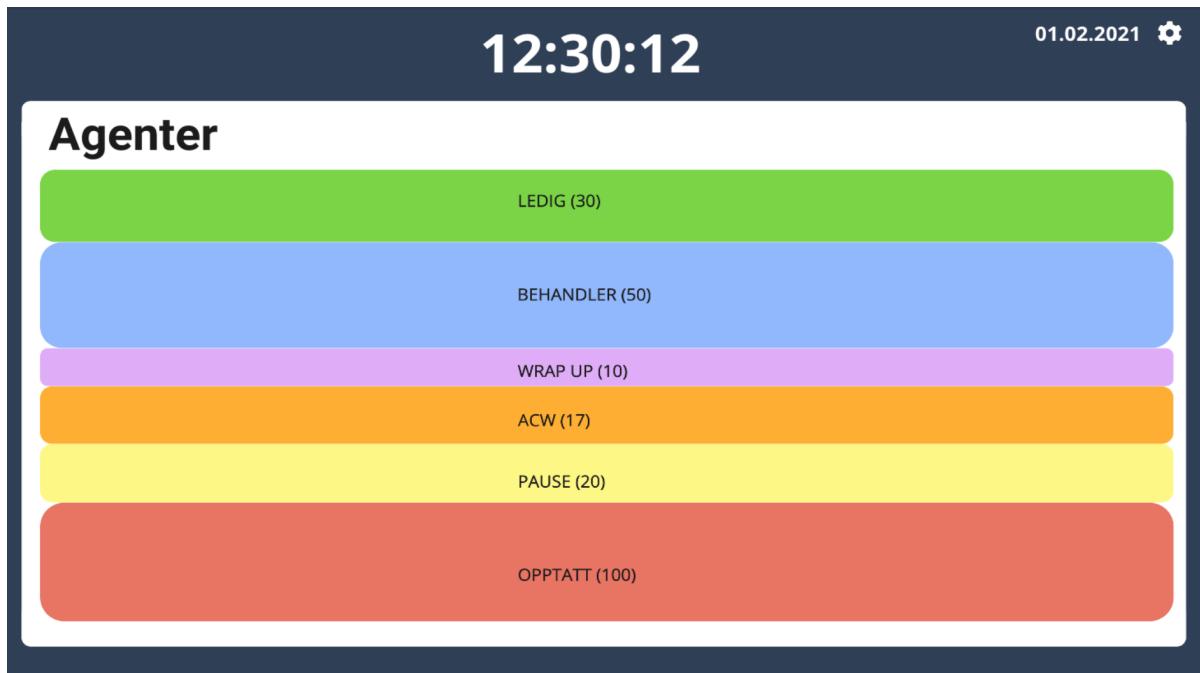
12:30:12

01.02.2021 

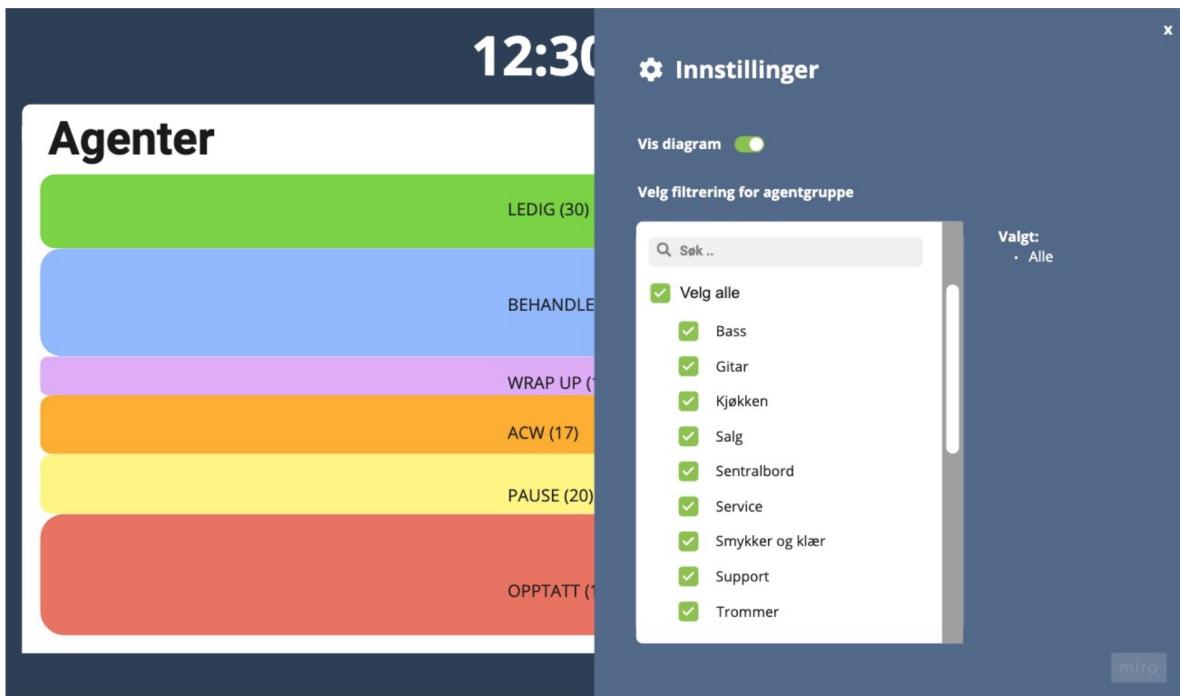
Agenter

Agentnavn	Status	Tid i status	Profil ↓	Agentgruppe	Besvart	Mottatt skriftlig	Gjennomsnittlig samtida	Gjennomsnittlig ringetid	Gjennomsnittlig behandlingstid
Christian Strand	BEHANDLER	00:04:10	Vintersport	Ski	4	4	00:04:10	00:04:10	00:10:10
Lise Johansen	PAUSE	00:04:10	Vintersport	Skiskyting	4	4	00:04:10	00:04:10	00:10:10
Bent Stiansen	BEHANDLER	00:04:10	Vintersport	Vintersport	4	4	00:04:10	00:04:10	00:10:10
Thomas Alsgaard	OPPTATT	00:04:10	Vintersport	Ski	4	4	00:04:10	00:04:10	00:10:10
Jenny Skavland	ACW	00:04:10	TV	TV	4	4	00:04:10	00:04:10	00:10:10
Espen Lind	LEDIG	00:04:10	Sentralbord	Sentralbord	4	4	00:04:10	00:04:10	00:10:10
Kurt Nilsen	WRAPUP	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10	00:10:10
Lise Johansen	PAUSE	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10	00:10:10
Kristian Eriksen	BEHANDLER	00:04:10	Musikk	Piano	4	4	00:04:10	00:04:10	00:10:10
Stian Staysman	OPPTATT	00:04:10	Musikk	Sanger	4	4	00:04:10	00:04:10	00:10:10
Tom Eriksen	ACW	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10	00:10:10
Solveig Kloppen	LEDIG	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10	00:10:10
Maria Mena	WRAPUP	00:04:10	Kjøkken	Kjøkken	4	4	00:04:10	00:04:10	00:10:10
Erlend Haaland	LEDIG	00:04:10	Fotball	Fotball	4	4	00:04:10	00:04:10	00:10:10

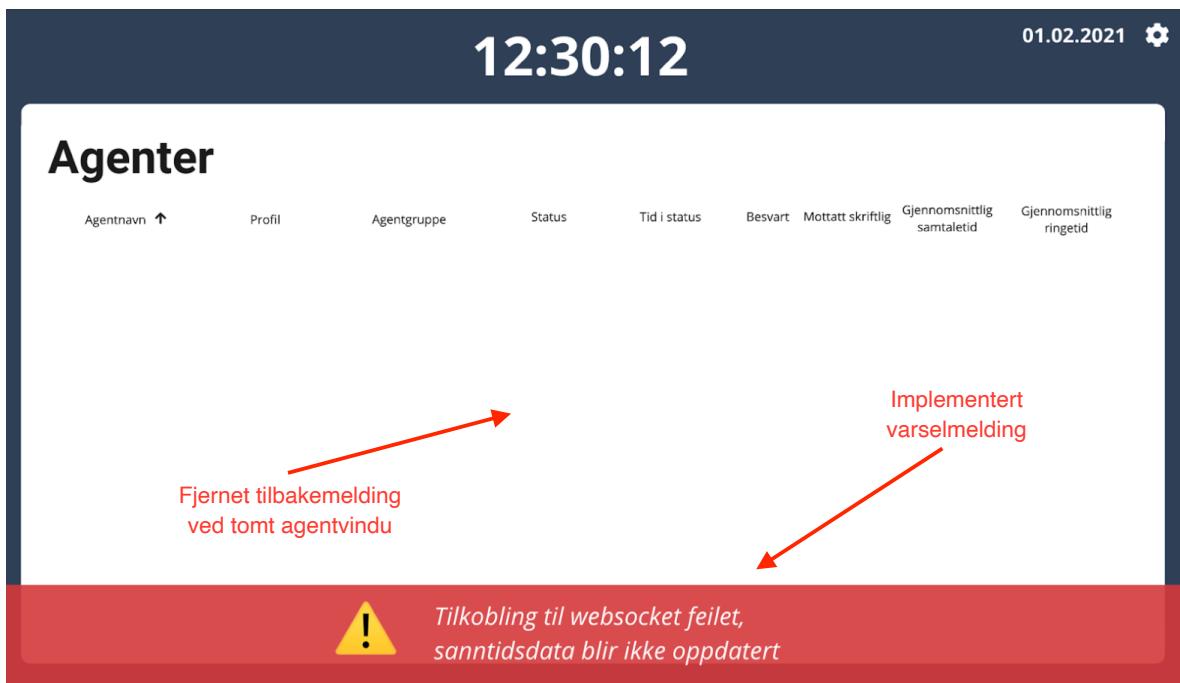
Figur 67: Agentoversikt med pastellfarger over hele raden



Figur 68: Diagramform av agentoversikten



Figur 69: Diagramform med innstillinger til høyre.



Figur 70: Tom agentoversikt med feilmelding om feilet tilkobling til websocket.

Ved tredje runde ble prototypen brukertestet på Thomas (CTO).

Tilbakemeldinger og endringer

- MVP kolonnene skulle være disse attributtene: Agentnavn, status, tid i status, profil, besvart, mottatt skriftlig, agentgruppe, Callstatus.
- Innstillinger skal integreres i ReWave.
- Velg alle, bør gjøres om til å være velg alle resultater.
- Var usikker på farge over hele, men mente det var opp til oss å se hvordan det ble.
- Gruppen ble oppfordret til å gjøre A/B tester eller geriljatester på dette.
- Sortering inn i Rewave.
- Logo også en del av MVP.

8.1.6 Fjerde runde brukertester - MVP

Figur 71 og 72 viser bilder av prototypene ved fjerne runde brukertester med fokus på MVP.



The screenshot shows a mobile application interface titled "Agenter". At the top, it displays the time "12:30:12" and the date "01.02.2021" with a gear icon for settings. The main content is a table with the following columns: Agentnavn, Status ↑, Tid i status, Profil, Agentgruppe, Besvart, Mottatt skriftlig, Gjennomsnittlig samtaletid, and Gjennomsnittlig behandlingstid. The table lists 15 agents with their respective details. The "Status ↑" column uses color-coded boxes to represent different states: green for "LEDIG", blue for "BEHANDLER", purple for "WRAPUP", orange for "ACW", yellow for "PAUSE", red for "OPPTATT", and grey for "Vintersport" and "Skiskyting".

Agentnavn	Status ↑	Tid i status	Profil	Agentgruppe	Besvart	Mottatt skriftlig	Gjennomsnittlig samtaletid	Gjennomsnittlig behandlingstid
Christian Strand	LEDIG	00:04:10	Vintersport	Ski	4	4	00:04:10	00:04:10
Lise Johansen	LEDIG	00:04:10	Vintersport	Skiskyting	4	4	00:04:10	00:04:10
Bent Stiansen	LEDIG	00:04:10	Vintersport	Vintersport	4	4	00:04:10	00:04:10
Thomas Alsgaard	BEHANDLER	00:04:10	Vintersport	Ski	4	4	00:04:10	00:04:10
Jenny Skavland	BEHANDLER	00:04:10	TV	TV	4	4	00:04:10	00:04:10
Espen Lind	BEHANDLER	00:04:10	Sentralbord	Sentralbord	4	4	00:04:10	00:04:10
Kurt Nilsen	WRAPUP	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Lise Johansen	ACW	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Kristian Eriksen	ACW	00:04:10	Musikk	Piano	4	4	00:04:10	00:04:10
Stian Staysman	PAUSE	00:04:10	Musikk	Sanger	4	4	00:04:10	00:04:10
Tom Eriksen	OPPTATT	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Solveig Kloppen	OPPTATT	00:04:10	Musikk	Gitar	4	4	00:04:10	00:04:10
Maria Mena	OPPTATT	00:04:10	Kjøkken	Kjøkken	4	4	00:04:10	00:04:10
Erlend Haaland	OPPTATT	00:04:10	Fotball	Fotball	4	4	00:04:10	00:04:10

Figur 71: Agentoversikt med farge over status-kolonnen

Agentnavn	Status ↑	Tid i status	Profil	Besvart	Mottatt skriftlig
Christian Strand	LEDIG	00:04:10	Vintersport	4	4
Lise Johansen	LEDIG	00:04:10	Vintersport	4	4
Bent Stiansen	LEDIG	00:04:10	Vintersport	4	4
Thomas Alsgaard	BEHANDLER	00:04:10	Vintersport	4	4
Jenny Skavland	BEHANDLER	00:04:10	TV	4	4
Espen Lind	BEHANDLER	00:04:10	Sentralbord	4	4
Kurt Nilsen	WRAPUP	00:00:03	Musikk	4	4
Lise Johansen	ACW	00:04:10	Musikk	4	4
Kristian Eriksen	ACW	00:04:10	Musikk	4	4
Stian Staysman	PAUSE	00:04:10	Musikk	4	4
Tom Eriksen	OPPTATT	00:04:10	Musikk	4	4
Solveig Kloppen	OPPTATT	00:04:10	Musikk	4	4
Maria Mena	OPPTATT	00:04:10	Kjøkken	4	4
Erlend Haaland	OPPTATT	00:04:10	Fotball	4	4

Figur 72: Agentoversikt med farge over hele raden

Det ble utført A/B tester hvor det ble vist frem to forskjellige wallboard. Et hvor det var farge over hele raden, og et hvor det kun var farge på statuskolonnen.

Tilbakemeldinger og endringer

Seks personer ble spurta (fire bekjente og to fra Zisson) om følgende: *Hvilken av wallboardene foretrekker du med tanke på fargevalg?* Hvorav fire svarte at de foretrak farge på status fremfor farger over hele raden. Mens to svarte at det gav bedre synlighet å ha statusfarge over hele raden. Ut i fra dette ble det valgt å bruke farge på statuskolonnen.

8.1.7 System usability scale test

Gjennomsnittet fra SUS-testene var 83,125 som tilsvarer graden A (utmerket). Ut i fra modellen 6.1 *Tabell for tolkning av SUS score* ser man at en poengsum over 68 er kategorisert som OK, Good, og excellent noe som betyr at testscoren bør ligge

over 68.

Fordi det ble utført tester på personer som er bekjente av gruppens medlemmer, kan resultatene være noe påvirket av dette. Likevel gir det en indikasjon for hvor brukervennlig tjenesten er.

Da testpersonene fra Zisson kun var personer som enten utviklet eller jobbet tett med systemet fra før og dermed hadde god kjennskap til det, ble det valgt å kun teste på bekjente personer, slik at de kunne gi en objektiv evaluering av systemet, hvor de ikke var påvirket av tidligere erfaring eller kjennskap.

For mer dokumentasjon rundt testen som ble utført se vedlegg [punkt 4.4 SUS spørreundersøkelse](#).

8.2 Enhetstester

8.2.1 Backend

Enhetstestene ble laget og gjennomført i prosjektet WallboardService. Det ble brukt xUnit og alle testene ble automatisert. Det har blitt laget enhetstester både for normal bruk (_OK) og tester for unormalt bruk (_FALSE).

Figur 73 viser skjermbilde av xUnit testene.



Figur 73: Kjørte enhetstester i wallboard service prosjektet

Eksempler på enhetstesting i backend

deleteCache_OK() er en test som sjekker om det fungerer å slette cache. Dette testes ved at en cache blir lagt til med metoden AddCache for så deretter å bli slettet. I assert-delen blir det sjekket om cache som ligger i lista med cacher er lik null. Siden cache ble slettet, og resultatet er lik null, er testen godkjent. Dette markeres også med den grønne V-en som vist på bildet i figur 74.

```
[Fact]
0 references | Run Test | Debug Test | ✓
public void deleteCache_OK()
{
    caches.AddCache(wallboardKey, cache, agentmapping);
    caches.deleteCache(wallboardKey);
    Assert.Equal(null, caches.getCacheFromCaches(wallboardKey));
}
```

Figur 74: C# enhetstest deleteCache_OK()

`findCache_FALSE()` er en test som skal returnere false ved forsøk på å finne en bestemt cache. Dette fordi den prøver å finne en cache som ikke finnes i listen over alle cache som har blitt lagt til/ opprettet. Det brukes “`Assert.false`” for å sjekke om cach har blitt funnet. Denne blir ikke funnet, og dermed er det riktig at resultatet er false. Testen blir dermed grønn, som man kan se i figur 75.

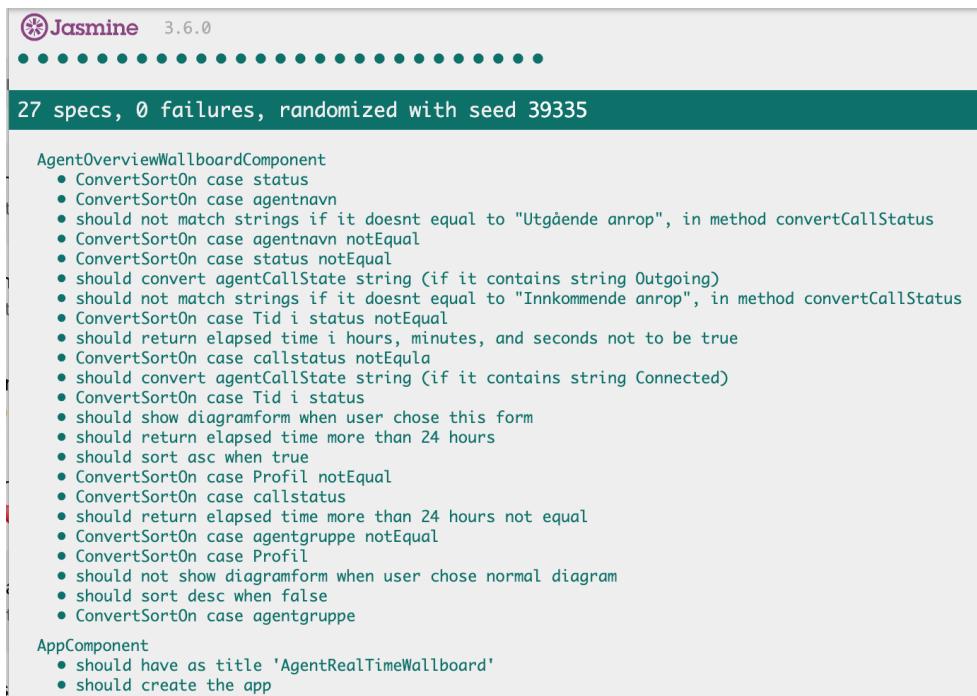
```
[Fact]
0 references | Run Test | Debug Test | ✓
public void findCache_FALSE()
{
    Guid notExistingWallboard = new Guid("11111111-93f2-4671-96ae-f3b158934455");
    var cacheFound = caches.findCacheInCaches(notExistingWallboard);

    Assert.False(cacheFound);
}
```

Figur 75: C# enhetstest `findCache_FALSE()`

8.2.2 Frontend

I figur 76 er resultatene av testene som er gjort i Karma og Jasmine, i prosjektet AgentRealTimeWallboard.



Figur 76: Resultater fra testene gjort i Karma og Jasmine

Eksempler på enhetstester i Angular:

I figur 77 blir det testet på at funksjonen getElapsedTime returnerer "Over 24t" dersom det har gått mer enn 24 timer siden det ble byttet status. Her blir det brukt toMatch() for å teste at strengene er like.

```
it('should return elapsed time more than 24 hours', () => {
  // Arrange
  let startTime = new Date('2021-04-06T07:28:39.2304777Z');
  // Act
  let time="Over 24t";
  // Assert
  expect(component.getElapsedTime(startTime)).toMatch(time);
});
```

Figur 77: Angular enhetstest av metoden getElapsetTime

I figur 78 blir det testet at funksjonen convertSortOn returnerer "Agentstate". Det blir brukt toEqual for å sjekke at de to strengene er like.

```
it('ConvertSortOn case status', () => {
  // Arrange
  const sorting= "Status";
  // Act
  const convertedFunc= component.convertSortOn(sorting);
  // Assert
  expect(convertedFunc).toEqual('Agentstate');
});
```

Figur 78: Angular enhetstest av metoden convertSortOn

I figur 79 blir det testet at convertCallStatus ikke returnerer "utgående anrop". Det er brukt not.toEqual for å teste at de to strengene ikke er like.

```
it('should not match strings if it doesnt equal to "Utgående anrop", in method convertCallStatus', () => {
  const agentCallState = 'Outgoing';

  const newString = component.convertCallStatus(agentCallState);

  expect(newString).not.toEqual('Outgoing');
});
```

Figur 79: Angular enhetstest av metoden convertCallStatus

Dekningsgrad

Vi har valgt å enhetsteste de mest kritiske områdene av programvaren da omfanget av å enhetsteste alt blir for stort i dette prosjektet. Enhetstestene er laget i prosjektene AgentRealTimeWallboard og WallboardService. Her testes alle metoder som innebærer logikk utover å kun hente/sende data.

Oppdagede feil

Det ble ikke oppdaget noen kritiske feil ved enhetstesting. Det som kom tydelig frem fra enhetstesting var at metodene burde gi bedre tilbakemeldinger på hva som skjer. For eksempel ved sletting av cache ble det oppdaget at dersom man sletter en cache som ikke eksisterer, kommer det ingen feilmelding på at sletting ikke tar sted. På denne måten virket det som at testen tilsier at cache ble slettet, noe som virker misvisende når man tester.

8.3 Integrasjonstester

Henviser til dokumentasjon av integrasjonstestene i GitHub:

[Vedlegg: Integrasjonstest-dokumentasjon \(ekstern lenke\)](#)

Integrasjonstestene ble kjørt både gjennom TestCafe og manuelle tester. Det ble gjort noen manuelle tester da endringer på wallboardet skulle skje i sanntid, som f.eks. endring av agentstatus og tid i status.

Oppdagede feil

- **Feil ved sortering av kolonne:**
 - Her var sortering på "Agentnavn" satt som default på sortering, noe som gjorde at brukeren ikke kunne anstre handlingen dersom det ble trykket på en radioknapp under valg av sortering.

- Siden det ikke var implementert en egen radioknapp for sortering på agentnavn, ble dette implementert som en egen, slik at dette ble en valgmulighet.
- **Feil ved lagre:**
 - Ved endring av bl.a. valg som “Sorter stigende” og “Agentgrupper” ble ikke alltid endringene lagret når man trykket på “Lagre”.
 - Dette ble fikset, slik at lagre alltid vil lagre endringer gjort på wallboardet.
- **Feil ved kloning:**
 - Kloning av wallboard var ikke implementert riktig, slik at den alltid gikk på edit til “Gammelt wallboard”-edit når man trykket på klon-ikonet.
 - Dette ble fikset, slik at man kom inn på riktig edit ved kloning av wallboard tilhørende “Nytt Wallboard”-edit.

TestCafe

Det ble brukt TestCafe på AgentRealTimeWallboard. Her ble innholdet som vises i wallboard-tabellen på frontend testet. 13 av 13 tester ble grønne - en prosentandel på 100%. I figur 80 vises skjermbilde av testene i TestCafe.

```
Running tests in:
- Chrome 90.0.4430.93 / macOS 10.15.7

Zisson Wallboard
✓ Check property of AgentName - OK
✓ Check property of AgentName - NOT OK
✓ Check property of AgentState - OK
✓ Check property of AgentState - NOT OK
✓ Check property of AgentStateStartTime - OK
✓ Check property of AgentStateStartTime - NOT OK
✓ Check property of AgentStateStartTime - OK
✓ Check property of AgentCallStateStartTime - NOT OK
✓ Check property of QueueProfile - OK
✓ Check property of QueueProfile - NOT OK
✓ Check property of AgentGroup - OK
✓ Check property of AgentGroup - NOT OK
✓ Check property of AgentGroup - NOT OK
```

13 passed (16s)

Figur 80: Kjørte tester i TestCafe

Testene sjekker at informasjonen som blir sendt til agenttabellen gjennom et agent-event enten samsvarer eller ikke samsvarer med den informasjonen som blir sendt inn. Eksempelvis så sjekker en test at status på agenten er riktig og samsvarer (OK), mens en annen test sjekker at hvis statusen ikke samsvarer så skal den ikke være riktig (NOT OK).

Figur 81 viser kodeeksempel fra prosjektet:

```
test('Check property of AgentState - OK', async t => {
  const title = Selector('h1');
  const tableRows = Selector('tbody > tr');
  const agentState = Selector('tbody > tr').withText('Available');
  // check title, table rows and first name exists
  await t
    .expect(title.innerText).eql("Agenter")
    .expect(tableRows.count).eql(1)
    .expect(agentState.exists).ok();
});

test('Check property of AgentState - NOT OK', async t => {
  const title = Selector('h1');
  const tableRows = Selector('tbody > tr');
  const agentState = Selector('tbody > tr').withText('Pause');
  // check title, table rows and first name exists
  await t
    .expect(title.innerText).eql("Agenter")
    .expect(tableRows.count).eql(1)
    .expect(agentState.exists).notOk();
});
```

Figur 81: Eksempel på integrasjonstest som sjekker agentstatus

Manuelle tester

Endringer som skjedde i sanntid på wallboardet, f.eks. endring av agentstatus, samt oppsett, redigering og distribuering av et wallboard ble testet manuelt. Forventede resultat, faktiske resultat og eventuelle kommentarer ble dokumentert i et excel-dokument, se [Vedlegg: Integrasjonstest-dokumentasjon \(ekstern lenke\)](#).

Resultat og dokumentasjon

- Antall grønne tester (passed): 52 tester = 96 %
- Antall røde tester (failed): 2 tester = 4%

Tester som ble røde

- Legge til logo ved opprettelse av nytt wallboard - kategori C
- Legge til logo i feil format - kategori C

Begge feilene er kategorisert som C: Feil med mindre alvorlige konsekvenser, for eksempel mindre grad av ustabilitet, mindre funksjons-/ egenskapsmangler etc. I tillegg var begge de røde testene av typen feil som allerede fantes i ReWave, og var derfor ikke en feil som gruppen prioriterte å rette opp i.

8.4 Systemtester

Henviser til dokumentasjon av integrasjonstestene i GitHub:

[Vedlegg: Systemtest-dokumentasjon \(ekstern lenke\)](#)

Resultatene av grønne tester i Katalon er vedlagt under [punkt 4.1, Skjermbilder av tester i Katalon.](#)

Systemtestene ble kjørt gjennom Katalon Recorder, samtidig som det ble gjort noen manuelle tester. De manuelle testene ble utført da noen tester måtte gjøres i sanntid, slik som å se klokkeslett og tid i status.

Oppdagede feil

- **Feil ved status:**
 - Statusen Pause ble rettet opp etter testing 04.05.2021, deretter retestet etter endringer.

- **Feil ved opplasting av logo:**
 - En feil som gjør at man må laste opp logo to ganger for at det skal bli lagret. Kategorisert som en C feil.

Resultat og dokumentasjon

- Antall grønne tester (passed): 34 tester = 94 %
- Antall røde tester (failed): 1 test = 3%
- Antall ikke-kjørte tester (remaining): 1 test = 3%

Tester som ble røde

- Legge til logo ved opprettelse av nytt wallboard - kategori C

Feilen er kategorisert som en C: Feil med mindre alvorlige konsekvenser, for eksempel mindre grad av ustabilitet, mindre funksjons-/egenskapsmangler etc.

Tester som ikke ble kjørt

- Se tid i callstatus

Testen ble ikke kjørt grunnet interne feil i systemet til Zisson. Callstatus kunne ikke testes da det ikke var mulig å ringe inn til en agent (agenten endret verken status eller callstatus).

8.5 Akseptansetester

8.5.1 Akseptansetest av kravspesifikasjon

Kravene blir satt opp i Jira og i dokumentet "Kravspesifikasjon". Endelig kravspesifikasjon som er grunnlaget for akseptansetest av kravspesifikasjon kan sees under [punkt 3.0, Endelig kravspesifikasjon](#).

Møte med Thomas og Moritz:

- Kravspesifikasjon blir gjennomgått og satt opp sammen slik at alle er inneforstått med hva kravene er for oppgaven.

[8.5.2 Akseptansetest av prototype](#)

Møte 15.02 med Thomas og Moritz:

- Gruppen viser frem prototypen og får klarsignal til å begynne utviklingen, med forbehold om at noen endringer må gjøres:
 - Modulen med innstillinger, samt sorteringsfunksjonaliteten, må flyttes inn i Zisson Interact.
 - Hvorvidt agentens statusfarge skal dekke hele raden med en agent eller kun i statuskolonnen er det usikkerheter rundt, og det blir viktig å teste senere i utviklingsforløpet når grensesnittet kan brukertestes.
 - Utover det godkjennes prototypen.

Møte 01.03 med Thomas og Moritz:

- Gruppen viser frem den oppdaterte prototypen der modulen med innstillinger er flyttet inn i Zisson Interact:
 - Prototypen godkjennes under møtet.

[8.5.3 Akseptansetest av inkrementer](#)

For hver sprint ble det gjort akseptansetester der inkrementene ble gjennomgått og satt til godkjent/ ikke godkjent. Se vedlegg under [punkt 8.5.3, Akseptansetest av inkrementer](#) for utdypende informasjon.

Dersom et inkrement ikke ble godkjent, ble det videreført til neste sprint, hvor det tas en ny gjennomgang og avgjørelse på inkrementet. På denne måten ble det sikret at alle inkrementer til slutt ble godkjent.

8.5.4 Akseptansetest av sluttprodukt

Det avholdes et avsluttende møte med Zisson, tilstede er Thomas Tretli og Moritz Lilleby, samt alle medlemmene i bachelorgruppen. Alle prosjektene blir gjennomgått i sin helhet og studentene får godkjent sluttproduktet.

9. Oppsummering

Alt i alt har testingen av prosjektet vært et viktig verktøy for å sikre kvalitet til produktet som har blitt utviklet. De fleste feilene ble rettet opp i, og sikret på denne måten bedre kvalitet på produktet. Testingen vil også ha en høy nytteverdi for de som skal videreutvikle produktet. Feilene som ble oppdaget, men som gruppen ikke kunne gjøre noe med, ble rapportert til oppdragsgiver for videre utvikling. Man kan også anta at testingen har sikret prosessen med å levere et tilstrekkelig produkt gjennom jevnlige akseptansetester som kontinuerlig ga en pekepinn på hvordan ting lå an. På denne måten var det lite usikkerhet knyttet mot akseptansetest av sluttproduktet, da akseptansetester underveis hadde gitt en bekreftelse på at ting levde opp til forventningene til oppdragsgiver

F. AVSLUTNING

1.0 Resultatet

I løpet av prosjektperioden har gruppen laget en ny wallboardløsning for oppdragsgiver Zisson. Etter ønske fra oppdragsgiver var oppgaven å fornye og forbedre det eksisterende wallboardet, både med tanke på design og estetikk, men også teknologi. Samtidig måtte det integreres i Zissons eget brukergrensesnitt.

Designet av wallboardet er fornyet. Enkelte valg ble gjort på bakgrunn av teori om hvordan en tabell bør representeres ut ifra informasjonen som skal visualiseres.

Andre valg ble gjort med støtte i de funnene som kom frem gjennom arbeidsmetodikken design thinking og undersøkelser, brukertester og akseptansetesting som ble utført. Ønsker fra oppdragsgiver har gått på bekostning av enkelte funn som kom frem, og derfor har ikke alle funn fra brukertestene blitt implementert i den siste løsningen. Funnene har allikevel blitt tatt med i prototyper underveis i designprosessen og vist som en løsning til oppdragsgiver. Gjennom akseptansetesting har Zisson hatt siste ord på hva de ønsker å ha med. På denne måten har det sikret at oppgaven som leveres lever opp til kravene og forventningene oppdragsgiver har hatt, og det er det som har vært av betydning.

Oppdragsgiver har bekreftet at de kommer til å videreutvikle prosjektet slik at det kan bli brukt som en del av deres kontaktsenterløsning. Zisson har gitt uttrykk for å ha vært imponert og fornøyd med både produktet og innsatsen fra bachelorgruppen, og har samtidig hatt stor forståelse for noen utfordringer i forhold til endring i krav underveis og andre interne problemer i systemet.

2.0 Hva kunne blitt gjort annerledes?

Det kan ikke bekreftes at produktet er optimalisert for kundene til Zisson, da det ikke har vært direkte brukertesting eller dialog med disse. Dette med bakgrunn i at brukertestene er utført av personer som jobber i Zisson, samt at det er kravene fra oppdragsgiver gruppen har jobbet ut ifra. Skulle man forbedret prosjektet ville det vært hensiktsmessig å kunne gjøre undersøkelser og brukertester direkte på brukere og kunder av systemet. Samtidig kunne undersøkelser tidlig i startfasen vært utført på en mer grundig måte, ved å intervju direktekunder av Zisson om hva som er deres utfordringer med dagens system og hva de hadde ønsket av funksjonalitet for wallboardet. Samtidig som det kunne vært utført grundigere litteratursøk på området.

Valget av arbeidsmetodikken Scrum har for gruppen virket som en god arbeidsmetodikk å jobbe etter, da det underveis har kommet flere endringer i krav fra oppdragsgiver. Dersom gruppen hadde jobbet med en mer planmessig arbeidsmetodikk, kunne dette skapt utfordringer og manglende implementasjon av endringene som kom. Det ble allikevel valgt å gjennomføre hele designprosessen i første sprint, noe som ikke ble gjennomførbart, da kravendringene også satte krav til endringer i designet underveis. Dette kunne kanskje vært løst på en bedre måte ved å ikke designe hele systemet fra begynnelsen, men å jobbe med inkrement for inkrement og utvikle designet deretter.

Gjennomføringen av System Usability Scale-testen kunne vært utført på flere testpersoner for å få et bredere datagrunnlag for vurderingen av brukervennlighet av systemet. Grunnlaget for testene kan derfor være et grunnlag for videre utforskning av brukervennlighet i fremtiden.

Til ettertanke kunne det også vært fint å ha gjennomført flere brukertester, med større grad av observasjoner, samt brukt flere metoder for bruktertesting slik som eyetracking-testing, og feltobservasjoner. Slik testing kunne vært utført dersom designprosessen hadde vært gjennomført samtidig med Scrum sprintene. Dette er en viktig lærdom for prosjektgruppen å ta med seg videre.

3.0 Forbedringspotensiale

Dersom Zisson ønsker å videreutvikle produktet kan det være hensiktsmessig at de brukertester på sine egne kunder og jobber ut ifra eventuelle tilbakemeldinger de får ut fra det.

Forslaget som kom frem i designprosessen med å endre navn og farger på statusene er også et forslag som studentene tenker kan løfte wallboard ytterligere og gi en høyere grad av brukervennlighet, da statusene kan gi mer informasjon enn det de gjør i dag. Det hele ble utarbeidet i samråd med Kim som var enig at dette kunne være en god løsning. Det kom senere frem at dette ble en for omfattende endring og det hele ville gått utenfor oppgavens omfang, da de opprinnelige statusene allerede ble brukt i Zisson Interact og at disse burde samsvare med statusene på wallboardet. Løsningen for behandler/process er implementert i AgentRealTimeWallboard prosjektet, dette dersom det skulle være ønskelig for Zisson ved en senere anledning å innføre funksjonaliteten i deres kontaktsenterløsning Zisson interact. Man kan tenke seg at en innføring av dette kunne gitt en enda bedre representasjon av agentene på wallboardet, noe som igjen kunne gitt økt kundetilfredshet.

Ved videreutvikling av wallboardet ville det vært optimalt å kunne implementere flere komponenter på wallboardet. slik som statistiske fremvisninger, agenthistorikk og mulighet for å endre oppsettet av komponentene på wallboardet.

4.0 Læringsutbytte

Perioden har gitt gruppen et stort læringsutbytte hvor gruppen har lært nye teknologier, språk og verktøy, samt tatt i bruk teoretisk kunnskap i praksis. Å sette seg inn i andres kode og systemer har vært en erfaring som både har vært utfordrende og lærerikt, noe som kan komme godt med i arbeidslivet som venter. Å følge en smidig arbeidsmetodikk har ført til en ryddig utviklingsprosess. Kommunikasjon og møter med oppdragsgiver har i tillegg gitt gruppen kunnskap om bedriften, og hvordan et utviklingsprosjekt foregår i praksis fra design, til implementering av design i kode, og testing.

Gruppen har hatt stort fokus på MVP og kravspesifikasjonene da det har vært ønskelig å tilfredsstille oppdragsgiverens krav. Fokuset på MVP har også gitt gruppen viktig lærdom om viktigheten av å avgrense omfanget av oppgaven.

COVID-19-situasjonen har gitt gruppen noen utfordringer når det gjelder kommunikasjon og samarbeid, både mellom studentene, men også med oppdragsgiver. Det å ikke sitte sammen fysisk har ført til at kommunikasjonen har tatt lengre tid enn det ellers ville gjort, samtidig så har det vært vanskeligere å følge med på hva hver enkelt gjør. Pandemien har også bydd på utfordringer knyttet til brukertesting. Samtidig har det gitt gruppen god trening på digitale videomøter, noe som har vært en viktig lærdom å ta med seg videre i arbeidslivet. Pandemien har også gitt gruppen viktige erfaringer knyttet til å løse utfordringer i en vanskelig periode, sammen som et team.

Alt i alt har prosjektet vært spennende og lærerikt, og gitt gruppen mange gode erfaringer som er relevant for både videre studier og arbeidslivet.

G. REFERANSER

Angular (u.å). Hentet 2021, 11. mai. <https://angular.io/>

Angular (u.å). *Observables in Angular*. Hentet 2021, 11 mai.
<https://angular.io/guide/observables-in-angular>

Angular Material. (2021). *Table*. Hentet 2021, 10. Mai.
<https://material.angular.io/components/table/overview>

Angular (u.å.). *The RxJS library*. Hentet 2021, 10. Mai.
<https://angular.io/guide/rx-library>

Aven, T. (2020, 5.november). *Risikoanalyse*. Hentet 12.mai 2021
<https://snl.no/risikoanalyse>

Babich N. (2016, 22. Juni). *UX Design: Checkbox and Toggle in forms*. UX planet.
Hentet 2021, 12. Februar.
<https://uxplanet.org/checkbox-and-toggle-in-forms-f0de6086ac41>

Babich, N. (2019, 17. januar). *Using Red and Green in UI Design*. UX planet. Hentet 2021, 12. Februar.
<https://uxplanet.org/using-red-and-green-in-ui-design-66b39e13de91>

Beggs. D. (2019, 11. Februar). *Best UX practices for search inputs*. UX collective.
Hentet 2021. 19. Februar.
<https://uxdesign.cc/best-ux-practices-for-search-inputs-c44dba565448>

Bootstrap (u.å). *Build fast, responsive sites with Bootstrap*. Hentet 2021, 11. mai.
<https://getbootstrap.com/>

CarreerRide. (u.å.). *Design phases*. Hentet 2021, 24. Mai.
<https://www.careerride.com/page/design-phases-637.aspx>

Chapman, C. (u.å). *The Role of Color in UX*. Toptal. Hentet 2021, 25. februar.
<https://www.toptal.com/designers/ux/color-in-ux>

Computer Hope. (2020, 6. februar). *Draw.io*. Computer Hope. Hentet 2021, 12 Mai,
fra <https://www.computerhope.com/jargon/d/drawio.htm>

Dam F. R. og Siang Y. T. (2021). *5 Stages in the Design Thinking Process*. Interaction design foundation. Hentet 2021, 1. Februar.

<https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>

Doorley, S., Holcomb. S., Klebahn. P., Segovia. K., og Utley. J. (2018). *Design thinking bootleg*. Design school at Stanford University. Hentet 2021, 21. Mai.

https://static1.squarespace.com/static/57c6b79629687fde090a0fdd/t/5b19b2f2aa4a99e99b26b6bb/1528410876119/dschool_bootleg_deck_2018_final_sm+282%29.pdf

Drolshammer, E., Olasveengen, M. F., (2016). *4 pluss 1 View Model*. Hentet 2021, 24. Mai. <https://wiki.cantara.no/display/dev/4+plus+1+View+Model>

Git (u.å). Hentet 2021, 11. mai. <https://git-scm.com/>

GitHub (2020, 24. juli). *Hello World*. Hentet 2021, 11. mai.
<https://guides.github.com/activities/hello-world/>

Hellmuth, M. (2019, 1.oktober). The Ultimate Guide to Designing Data Tables.
<https://medium.com/design-with-figma/the-ultimate-guide-to-designing-data-tables-7db29713a85a>

Hevner. A. R., March. S. T., Park. J., og Ram. S. (2004). *Design Science In Information Systems Research*. Hentet 2021, 19. Mai.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.1725&rep=rep1&type=pdf>

HubSpire (u.å.). *Application programming interface*. Hentet 2021, 18. Mai.
<https://www.hubspire.com/resources/general/application-programming-interface/>

Jira (u.å). Hentet 2021, 11. mai.

https://www.atlassian.com/software/jira?&aceid=&adposition=&adgroup=9500365689&campaign=9124878870&creative=415542763132&device=c&keyword=jira&matchtype=e&network=g&placement=&ds_kids=p51242194730&ds_e=GOOGLE&ds_eid=700000001558501&ds_e1=GOOGLE&gclid=CjwKCAjw1uiEBhBzEiwAO9BHckGENXjLdhHt9SQuKD0VAii_3HHve_MxXPfASc6bPNOSMcjAHUe6xoCo8AQAvD_BwE&qclsrc=aw.ds

Karma (u.å). Hentet 2021, 11. mai. <https://karma-runner.github.io/latest/index.html>

Katalon (u.å). Hentet 2021, 11. mai. <https://www.katalon.com/>

Kontio, M. (2005). *Architectural manifesto. Designing software architectures, Part 5.* Hentet 2021, 24. Mai.

<https://www.scribd.com/document/84656947/Architectural-manifesto-Designing-software-architectures-Part-5>

Kruchten. P. (1995). *Architectural blueprints- The “4+1” View Model of Software Architecture.* Hentet 2021, 24. Mai.

<https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>

Kucheriyav, A. (u.å.). *Color Accessibility – UX Best Practices for Using Color in Design.* Intechnic. Hentet 2021, 25.februar.

<https://www.intechnic.com/blog/color-accessibility-ux-best-practices-for-using-color-in-design>

Kukhnavets. P. (2019, 6. Juni). *How to run Scrum efficiently in 2019? Quick guide for beginners.* Hentet 2021, 11. Mai. <https://habr.com/en/company/hygger/blog/455022/>

Khorikov. V. (2016, 4. mai). Singleton vs Dependency Injection. Enterprise Craftsmanship. Hentet 2021, 10. Mai.

<https://enterprisecraftsmanship.com/posts/singleton-vs-dependency-injection/>

Martina (2021, 9. februar). 19 of the best fonts for Websites. Hentet 2021, 14. Februar. <https://zyro.com/blog/best-fonts-for-websites/>

McElroy K. (2017). *Prototyping for Designers, Developing the best digital & physical products.* USA: O'Reilly.

Microsoft (u.å.). *A Tour of C#.* Hentet 2021, 11. mai.

<https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

Microsoft. (u.å.). *Dictionary<TKey,TValue> Class.* Hentet 2021, 10. Mai.

<https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2?view=net-5.0>

Microsoft (u.å.). *Language Integrated Query (LINQ) (C#).* Hentet 2021, 11. mai.

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>

Microsoft. (u.å.). *List<T> Class.* Hentet 2021, 10. Mai.

<https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?view=net-5.0>

Microsoft (u.å). *Velkommen til Microsoft Teams*. Hentet 2021, 12. mai.
https://support.microsoft.com/nb-no/office/velkommen-til-microsoft-teams-b98d533f-118e-4bae-bf44-3df2470c2b12?wt.mc_id=otc_microsoft_teams

Microsoft (u.å). *Visual Studio*. Hentet 2021, 11. mai.
<https://visualstudio.microsoft.com/>

Microsoft (u.å). *Visual Studio Code*. Hentet 2021, 11 mai.
<https://code.visualstudio.com/>

Microsoft (u.å). *What Is .Net?*. Hentet 2021, 11. mai.
<https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>

Miro (u.å.). Hentet 2021, 12.mai. <https://miro.com/about/>

Nielsen. J. (2004, 26. September). *Checkboxes vs. Radio Buttons*. Nielsen Norman group. Hentet 2021. 19. Februar.
<https://www.nngroup.com/articles/checkboxes-vs-radio-buttons/#:~:text=Radio%20buttons%20are%20used%20when,previously%20selected%20in%20the%20list.>

Nielsen Norman Group. (1994, 24. april). *10 Usability Heuristics for User Interface Design*. (<https://www.nngroup.com/articles/ten-usability-heuristics/>).

Nishadha (2020, 21. Desember). *Use Case Diagram Relationships Explained with Examples*. Hentet 2021, 15. mai.
<https://creately.com/blog/diagrams/use-case-diagram-relationships/>

OpenVPN (u.å). Hentet 2021, 11. mai. <https://openvpn.net/>

Optimizely. *A/B testing*. Hentet 2021. 11. Mai.
<https://www.optimizely.com/optimization-glossary/ab-testing/>

Palliyaguru, C. (2018, 30. mars). *Color basics and psychology*. UX collective. Hentet 2021, 25. februar. <https://uxdesign.cc/color-effect-b78fae8bb72f>

Pello. R. (2018, 31. oktober). *Design science research — a short summary*. Hentet 2021, 19. Mai.
<https://medium.com/@pello/design-science-research-a-summary-bb538a40f669>

Product plan.(u.å.) *Minimum Viable Product*. Hentet 2021. 19. Februar.
<https://www.productplan.com/glossary/minimum-viable-product/>

RabbitMQ (u.å). Hentet 2021, 11. mai. <https://www.rabbitmq.com/>

Rosenfield, L., Morville, P., Arango, J. (2015). *Information architecture: For the web and beyond.* (4. utg.). USA: O'Reilly Media Inc.

RxJS (2020). *Introduction.* Hentet 2021, 11. mai. <https://www.learnrxjs.io/>

RxJS (u.å). RxJS Introduction. Hentet 2021, 13.mai.
<https://rxjs-dev.firebaseio.com/guide/overview>

Sandnes. E. F. (2018). *Universell utforming av IKT-systemer. Brukergrensesnitt for alle.* Oslo: Universitetsforlaget.

Sing, T. (u.å.). Interaction Design Foundation: Design thinking. Hentet 2021, 12, Februar. <https://www.interaction-design.org/literature/topics/design-thinking>

Sommerville I. (2016). *Selected chapters from Software Engineering.* Edinburgh: Pearson.

Spillner, A., Linz, T., Schaefer, H. (2014). *Software Testing Foundations: A Study Guide for the Certified Tester Exam* (4. utgave). Rocky Nook Inc.

S. Mohammadi (2020, 17. november) *Miro – virtuelle workshops og brainstorming i en hverdag med hjemmekontor.* Hentet 2021, 11. mai.
<https://www.thenewcompany.no/post/miro-virtuelle-workshops-og-brainstorming-i-en-hverdag-med-hjemmekontor>

TestCafe (u.å). Hentet 2021, 11. mai. <https://testcafe.io/>

Thomas. N. How To Use The System Usability Scale (SUS) To Evaluate The Usability Of Your Website. UsabilityGeek. Hentet 2021. 6. Mai.
<https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>

UsabiliTest. System Usability Scale (SUS) Plus. Hentet 2021. 6. Mai.
<https://www.usabilitest.com/system-usability-scale>

Uttersrud U. (2021) Avsnitt: 1.1.1 Hva er en algoritme? Hentet 2021, 10. Mai.
<https://www.cs.hioa.no/~ulfu/appolonius/kap1/1/kap11.html#1.1.1>

UX planet. (2016, 5. September.) *The art of UX sketching and paper prototyping.* Hentet 2021, 1. Februar.

<https://uxplanet.org/the-art-of-ux-sketching-and-paper-prototyping-5dae5a1efc7d>

Vera Tank. (u.å.). VERA Risikoanalyse. Hentet 2021, 12.mai.

<https://veratank.no/risikoanalyse/>

xUnit (u.å). Hentet 2021, 11. mai. <https://xunit.net/>

Yaestar. (2020, 2. Desember). Call Center Wallboard: The Do's and Don'ts. Hentet 2021, 12. Mai. <https://www.yeastar.com/blog/call-center-wallboard-best-practices/>

Friis. D. R, Siang. T. Y (2021). 5 Stages in the Design Thinking Process. Hentet 2021, 30. April.

<https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>



OSLO
OMET

OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITET

Vedlegg

Wallboard for Zisson

Bachelorprosjekt 2020 / 2021

Gruppe 32

Martine Holmberg s236808
Camilla H. Carlsen s197235
Synne Angell s331153
Nikola Sekiw s333975

H. VEDLEGG

1.0 Dokumentasjon av Scrum

Nedenfor er dokumentasjon av de ulike sprintene gruppen gikk gjennom i Scrum-prosessen med utdypende informasjon rundt inkrementene som ble implementert og sammendrag fra møtene som ble holdt.

1.1 Sprint 1 (1.februar - 12. februar)

Sprint planning

Målsettingen var å gjennomføre alle stegene i design thinking i denne sprinten. Samtidig som det var et hovedmål å komme godt i gang med prototypene, slik at prosjektgruppen kunne nærme seg et design som fungerte før utviklingsfasen startet. Se [3.1 Designprosess \(design thinking\)](#) for utdypning.

Sprint review

Etter sprint review ble det tydelig at wallboard-innstillinger skulle være en del av Zisson Interact. Det ble samtidig avgjort at kolonnene som skulle være en del av MVP var: Agentnavn, status, tid i status, callstatus, profil, agentgruppe, besvart og besvart skriftlige. Derfor måtte endringer av prototypen gjøres i neste sprint.

1.2 Sprint 2 (15.februar - 26.februar)

Sprint planning

Følgende funksjonalitet ble valgt ut fra backlog:

1. Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.
2. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet ut i fra hva man ønsker å vise frem.

Annet:

1. Endringer i prototypene

Implementasjon

1. **Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.**

1.1 Oppsett av websocket

Første implementasjon ble oppsett av websocket på frontend. En websocket krever ikke en HTTP-request for hver forespørsel som sendes/ mottas, sammenlignet med vanlige HTTP-kall. Det er derfor et effektivt og godt verktøy ved behov for sanntidsdata (Carnahan, 2016). Gruppen får et prosjekt overlevert fra utvikler i Zisson med et “skjelett” for server-delen, delvis klar for å kobles mot en websocket på frontend.

Gruppen måtte opprette et angular-prosjekt fra bunnen av, som skulle bli det nye wallboardet. For å opprette tilkobling til websocket på klient brukes RxJS biblioteket som er tilgjengelig i angular.

1.2 Initialisering av data

For å teste at websocket fungerer gjøres et forsøk på initialisering av data. Optimal initialisering ble vanskelig, da det kommer frem av studentene må få tilgang til Zisson sin VPN. Derfor ble det i mellomtiden satt opp en RabbitMQ med data skrevet direkte inn i JSON, for å simulere data som egentlig skulle kommet fra MQ/ kø på server og videre til frontend. Server mottar elementer fra køen gjennom “eventer” som avfyrer, hvor det er modellen “AgentStateEvent” det skal jobbes med. Den avfyrer for hver endring som

skjer på en agent, og sender da inn hele agenten med endringene som et JSON-objekt.

2. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.

Det ble i første omgang laget en modell for et wallboard, der modellen skulle brukes i ReWave og sendes over til databasen. Modellen videresendes til teknisk veileder, som gjør endringer i Zisson sin database. Databasen har ikke studentene tilgang til, derfor må eventuelle endringer i databasen gå gjennom teknisk veileder. Modellen som lages inneholder de verdiene som har kommet frem i kravspesifikasjonen at et wallboard bør inneholde.

Annet:

1. Endringer av prototypene

Da det kom nye krav fra oppdragsgiver på hvilke kolonner som skulle være med i agenttabellen, ble endringene gjort i prototypen. Kolonnene som skulle være med var: Agentnavn, status, tid i status, callstatus, profil og agentgruppe. Det ble samtidig laget et forslag til design av konfigurasjon av et wallboard i ReWave.

[Sprint review](#)

Sprint review-møte avholdes og hvert inkrement gjennomgås. Det ble en enighet om at inkrementet knyttet til sanntidsdata må videreføres til neste sprint. Studentene fikk tilgang til en del eksempladata (JSON-objekt) som kan brukes for å midlertidig simulere mer realistiske data gjennom RabbitMQ, fram til det kan hentes ekte sanntids-eventer.

[Sprint retrospective](#)

Gruppen avholdt et retrospective-møte, der det ble enighet om at sprinten har gått tregere enn forventet. Både fordi websocket og RabbitMQ var nytt for alle, men også fordi mye tid ble brukt til å vente på svar slik at arbeidet har blitt stående stille i perioder. Oppdragsgiver skulle undersøke manglende VPN-tilgang nærmere til neste sprint. Prosjektgruppen ble enige om å fordele oppgavene i større grad i neste sprint.

1.3 Sprint 3 (1.mars - 12.mars)

Sprint planning

Følgende ny funksjonalitet ble valgt ut fra backlog:

1. Som agentleder ønsker jeg at agentene lagres til cache, slik at jeg ved tilkobling får inn de siste dataene som har vært registrert.
2. Som agentleder ønsker jeg et wallboard som viser frem agentene i en tabell for å ha en oversikt over agentene som er på jobb.
3. Som agentleder ønsker jeg at wallboardet gjør et nytt forsøk på tilkobling til websocket dersom tilkobling mislykkes.
4. Som agentleder ønsker jeg å se agentnavn for å skille mellom agentene.
5. Som agentleder ønsker jeg å se profil for å vite hvilken profil agentene tilhører.
6. Som agentleder ønsker jeg å se agentgruppe for å vite hvilken agentgruppe agentene tilhører.

Følgende funksjonalitet videreføres fra forrige sprint:

1. Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.
2. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet ut i fra hva man ønsker å vise frem.

Implementasjon

- 1. Som agentleder ønsker jeg at agentene lagres til cache, slik at jeg ved tilkobling får inn de siste dataene som har vært registrert.**

Data om agentene måtte lagres i cache. Dette ble gjort ved å lagre data som kommer inn ved bruk av *Dictionary* i C#. *Dictionary* er en type *Collection* som holder på et nøkkel- og verdi-par (Microsoft, u.å.) I dette tilfellet er nøkkelen agentene sin login-id, og verdi er agent-objektet. Det ble også laget kode for at objektene i cache skulle oppdateres i sanntid. På denne måten ble det sikret at de siste lagrede data av agentene ble vist på skjermen ved oppstart av et wallboard som tidligere hadde kjørt. Dette var ønskelig, da det uten cache kun ville kommet opp én og én agent på skjermen ut i fra om de utløste et event. Videre ble det utviklet funksjonalitet slik at server sjekket om det lå noe lagret i cache ved oppkobling til websocket, og ved det tilfelle sendte dataene til klienten.

- 2. Som agentleder ønsker jeg et wallboard som viser frem agentene i en tabell for å ha en oversikt over agentene som er på jobb.**

2.1 Lage wallboard-komponent på frontend

Det ble implementert en ny komponent som var en del av selve wallboardet. Det legges her opp til å motta en logo, i tillegg til at dato og klokkeslett vises. Komponenten er kalt header-wallboard-component.

2.2 Lage agentabell komponent på frontend

Komponenten for agentoversikt med tabell og kolonner utvikles i angular med et API (MatTableModule). Komponenten ble kalt agent-overview-component. MatTable utvikles slik at den er klar for å motta

agentobjekter som kommer inn i JSON-format fra websocket.

3. Som agentleder ønsker jeg at wallboardet gjør et nytt forsøk på tilkobling til websocket dersom tilkobling mislykkes.

Det ble laget kode slik at frontend prøver på en ny tilkobling dersom det skjer en feil ved tilkobling av websocket. Dette gjøres med en SetInterval-metode som finnes i angular. SetInterval brukes for å sikre at klienten kontinuerlig med et gitt intervall prøver å koble seg opp til websocket igjen.

4. Som agentleder ønsker jeg å se agentnavn for å skille mellom agentene.

Fra JSON-objektet som mottas gjennom websocket ble det opprettet et AgentStatus-objekt på klient. Fra dette objektet hentes det ut navn som vises frem i tabellen under kolonnen "Agentnavn".

5. Som agentleder ønsker jeg å se profil for å vite hvilken profil agentene tilhører.

På lik linje med agentnavn, hentes også profil ut fra AgentStatus-objektet og vises frem i tabellen under kolonnen "Profil".

6. Som agentleder ønsker jeg å se agentgruppe for å vite hvilken agentgruppe agentene tilhører.

På lik linje med agentnavn og profil, hentes agentgruppe ut fra AgentStatus-objektet og vises frem i tabellen under kolonnen "Agentgruppe".

Inkrementer videreført fra forrige sprint:

1. Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.

1.1 Initialisering

Inkrementet fortsetter fra forrige sprint. Det ble satt opp en metode i AgentController på server, som hentet data knyttet til modellen AgentStatus. Denne modellen inneholder informasjon om hver enkelt agent som videre skal lagres til cache. Samtidig som det lagres til cache, ble dataene også oversendt med websocket til frontend, der agentstatus ble vist i nettleseren.

- 2. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet ut i fra hva man ønsker å vise frem.**

2.1 Lage en mockup config

Det ble laget en klasse, ConfigMockup, en midlertidig løsning på et objekt med data fra et wallboard på server. Opprinnelig er dette data som blir opprettet i ReWave ved opprettelsen av et wallboard, og deretter blir lagt inn i databasen som gjør det mulig å hente konfigurasjonen på serversiden. Ettersom studentene ikke hadde tilgang til VPN for å få tak i data fra databasen og ReWave ikke var ferdigstilt, ble det nødvendig med en midlertidig klasse for å lage en mockup av et wallboard objekt.

[Sprint review](#)

Sprint review-møte ble avholdt og arbeidet som ble utført under sprintet ble gjennomgått. Oppdragsgiver godkjente arbeidet som var gjort. Det kom frem at cache må håndtere flere wallboards basert på en nøkkel (wallboardkey), og dette måtte jobbes videre med i neste sprint. Deretter måtte det fortsatt arbeides med konfigurasjonsdelen av et wallboard i ReWave.

[Sprint retrospective](#)

Gruppen avholdt et retrospective-møte etter sprint review der foregående sprint ble diskutert. Det var en enighet om at arbeidsfordelingen fungerte godt og gruppen var fornøyd med å ha utført en del arbeid. Det ble diskutert at det til neste sprint ble viktig å sette av tid til å sette seg inn i ReWave, slik at alle i gruppen hadde en forståelse av systemet det skulle jobbes i.

1.4 Sprint 4 (15.mars - 26.mars)

Sprint planning

Følgende funksjonalitet ble valgt ut fra backlog:

1. Som agentleder ønsker jeg å se tid i agentstatus for å vite hvor lenge agenten har vært i en status.
2. Som agentleder ønsker jeg å se callstatus for å vite hvor lenge hver agent er i hver telefonsamtale og se hvem agenten er i samtale med.
3. Som administrator ønsker jeg å kunne filtrere på agentgruppe for å vise ønskede agenter på wallboardet.
4. Som agentleder ønsker jeg å se en overordnet oversikt over statusene til agentene i diagramform, slik at jeg har bedre oversikt ved mange agenter.
5. Som agentleder ønsker jeg at cache håndterer ulike wallboards ut i fra en wallboard nøkkel, slik at jeg kun får dataene som tilhører et gitt wallboard.
6. Som administrator ønsker jeg å ha en knapp som sier noe om jeg skal opprette et wallboard med den nye eller den gamle konfigurasjonen.
7. Som agentleder ønsker jeg at wallboardet fremstår responsivt på store skjermer, slik at wallboardet ser bra ut når vises frem på storskjerm.

Følgende funksjonalitet ble videreført fra forrige sprint:

1. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet ut i fra hva man ønsker å vise frem.

Annet:

1. Enhetstesting

Implementasjon

- 1. Som agentleder ønsker jeg å se tid i agentstatus for å vite hvor lenge agenten har vært i en status.**

Det ble laget kode for å telle tiden en agent har vært i en status ut i fra tiden man fikk inn fra server. Tiden kom inn som en string fra databasen og ble gjort om til datatypen Date i angular, og ut i fra dette sammenlignet man nåværende tid med tiden som kom inn.

- 2. Som agentleder ønsker jeg å se callstatus for å vite hvor lenge hver agent er i hver telefonsamtale**

Det ble laget kode for callstatus med en teller som teller hvor lenge en agent har vært i samtale. Koden ble laget på samme måte som tidtelleren knyttet til status.

- 3. Som administrator ønsker jeg å kunne filtrere på agentgruppe for å vise ønskede agenter på wallboardet.**

På server og klient ble det jobbet ut ifra en mockup av et wallboard, da tekniske problemer har forsinket prosjektet med tanke på å få ekte data fra databasen. Ut i fra wallboard-mockup-objektet ble det implementert funksjonalitet på server som filtrerer ut agenter basert på hvilke agentgrupper som er valgt for wallboardet. På denne måten mottar klienten kun de agentene som skal vises frem.

- 4. Som agentleder ønsker jeg å se en overordnet oversikt over statusene til agentene i diagramform, slik at jeg har bedre oversikt ved mange agenter.**

Det ble utviklet en diagramform som viste agentene kun basert på status. Diagramformen ble utviklet i en egen komponent, og var en forenkling av tabellen som først hadde blitt utviklet. Diagramformen viste kun kolonnen status, og legger agentene med samme status i rader tilknyttet hverandre. På denne måten fremstår det som et slags diagram, der man lett får oversikten over antall i statuser.

5. Som agentleder ønsker jeg at cache håndterer ulike wallboards ut i fra en wallboard nøkkel, slik at jeg kun får dataene som tilhører et gitt wallboard.

Cache ble videreutviklet slik at det ble opprettet en cache for hvert wallboard, og cache ble hentet ut ved å finne cache med gitt wallboardkey. Tidligere har cache kun bestått av en Dictionary. Koden ble endret til å ha en Caches-klasse som er en beholder for alle cacher. Caches består derfor av en Dictionary med et nøkkel-verdipar (wallboardkey, cache).

6. Som administrator ønsker jeg å ha en knapp som sier noe om jeg skal opprette et wallboard med den nye eller den gamle konfigurasjonen.

Det ble implementert en dropdown-menü som ga brukeren et valg mellom å opprette et wallboard av den nye eller den gamle typen. Det ble også gjort endringer slik at det er en generell wallboard-modell (WallboardEntry) som den gamle og den nye arver fra, i tillegg til at det er ekstra verdier i modellen for både den nye (Wallboard) og den gamle (AgentRealTime) konfigurasjonen som er spesifikke for hver av dem.

7. Som agentleder ønsker jeg at wallboardet fremstår responsivt på store skjermer, slik at wallboardet ser bra ut når det vises frem på storskjerm.

Det ble lagt til @media queries i CSS for å gjøre tabellen responsiv på større skjermer. Det var ikke et behov for å tilrettelegge for mindre skjermer, da wallboardet alltid skal vises frem med stor skjermstørrelse.

Inkrementer videreført fra forrige sprint:

- 1. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.**

1.1 Sette opp konfigurasjon i Rewave

For å sette opp konfigurasjon måtte gruppen først sette seg inn i to nye prosjekter: ReWave og WaveApiServer. Det ble opprettet en ny komponent som skulle være hovedsiden for å opprette et wallboard av den nye typen. Her ble det lagret elementene på frontend som var nødvendige for å ha mulighet til å velge konfigurasjon med de feltene som var ønsket. Routing i angular måtte tilpasses til den nye komponenten.

Annet:

1. Enhetstesting

Ved siden av arbeid med inkrementer fra backlog, starter arbeidet med enhetstester. Enhetstesting ble ikke lagt inn som et mål for at et inkrement skulle bli godkjent, og derfor gikk det ikke innunder hvert inkrement. Testing var allikevel arbeid som skulle gjøres parallelt med utvikling for å sikre kvalitet.

Sprint review

Arbeidet ble gjennomgått og det ble diskusjoner hvorvidt det var nødvendig med fremvisning av agentene i diagram. Det ble konkludert med at det var ønskelig å representere diagrammet med store bokser i statusfarger med en teller for antall

agenter i hver status, istedenfor det forslaget som var utarbeidet til nå. Det ble en enighet om å fjerne selve diagrammet enn så lenge, og kun ha bokser som viser antall i hver status, og komme tilbake til det om noe annet skal implementeres for å vise frem agentene i diagramform. Arbeidet med Rewave og konfigurasjon av et wallboard måtte også videreføres. Det ble til stadighet gjort endringer i Zisson sine systemer og databasen, noe som førte til at det måtte gjøres endringer i prosjektet gruppen jobbet med.

Sprint retrospective

Sprintet ble gjennomgått og det ble diskutert hvordan man skulle jobbe videre i neste sprint. Det ble en enighet om at det måtte brukes mer tid på testing i de påfølgende sprintene.

1.5 Sprint 5 (6.april - 16.april)

Sprint planning

Følgende funksjonalitet ble valgt ut fra backlog:

1. Som administrator ønsker jeg å legge til en logo til wallboardet fordi det er en del av selskapets profil.
2. Som bruker ønsker jeg å sortere på de ulike kolonnene for å kunne tilpasse sorteringen ut i fra hva som er relevant.
3. Som agentleder ønsker jeg å se callstatus for å vite hvilket telefonnummer en agent er i samtale med slik at jeg har oversikt.
4. Som administrator ønsker jeg å kunne endre konfigurasjonen til et wallboard slik at jeg ikke må opprette et nytt dersom endringer er nødvendig.

Følgende funksjonalitet videreføres fra forrige sprint:

1. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.

2. Som agentleder ønsker jeg å se en overordnet oversikt over statusene til agentene i diagramform, slik at jeg har bedre oversikt ved mange agenter.

Andre arbeidsoppgaver:

1. Enhetstester
2. Integrasjonstester
3. Endringer i diagramform i prototyper

Implementasjon

- 1. Som administrator ønsker jeg å legge til en logo til wallboardet fordi det er en del av selskapets profil.**

Det ble laget funksjonalitet for å laste opp logo i ReWave og å motta denne på klient. Logoen ble mottatt som en Base64-string fra databasen, og måtte konverteres til en SafeUrl for så å konverteres til en jpeg fil, for tilslutt å kunne vises frem på wallboardet.

- 2. Som bruker ønsker jeg å sortere på de ulike kolonnene for å kunne tilpasse sorteringen ut i fra hva som er relevant.**

Det ble laget funksjonalitet for å sortere de ulike kolonnene og å motta informasjon om hvilken kolonne det skulle sorteres på, samt sorteringsretning det skulle sorteres på. Sorteringsmekanismen ble bygget på MatTable sin MatSort, som tok i mot kolonnen og sorteringsretningen, for så å sorterte etter det.

- 3. Som agentleder ønsker jeg å se callstatus for å vite hvilket telefonnummer en agent er i samtale med slik at jeg har oversikt.**

Informasjon som sa noe om det var et innkommende eller utgående anrop,

samt telefonnummer, ble hentet frem for å vises i kolonnen “Callstatus” i tabellen. Dette er data som ble sendt med et agentevent.

4. Som administrator ønsker jeg å kunne endre konfigurasjonen til et wallboard slik at jeg ikke må opprette et nytt dersom endringer er nødvendig.

Det ble laget funksjonalitet i ReWave som gjorde det mulig å oppdatere et allerede eksisterende wallboard av den nye typen. Når man trykker på “Endre” knappen for et wallboard, vil det bli gjort et API-kall for å hente det gitte wallboardet ut ifra en id. Videre navigeres man til samme komponent som ved opprettelse av wallboard, bare at verdiene i komponentene da er fylt ut med verdiene som det eksisterende wallboardet inneholder. Ved lagring gjøres et HTTP PUT kall for å oppdatere wallboardet i databasen.

Inkrementer videreført fra forrige sprint:

1. Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet ut i fra hva man ønsker å vise frem.

Siden for å konfigurere et wallboard i ReWave ble ferdigstilt. Det ble lagt til valg som: vis som diagram, navn på wallboard, et array for valg av kolonne det skulle sorteres på, samt sorteringsretning. I tillegg kunne man velge hvilke agentgrupper man ønsket å filtrere på, som deretter skulle vises på wallboardet. Det ble jobbet videre med å få hentet ut objektet som opprettes fra databasen. Grunnet mange avhengigheter og endringer i Zisson sin opprinnelige database var dette arbeid som tok tid.

2. Som agentleder ønsker jeg å se en overordnet oversikt over statusene til agentene i diagramform, slik at jeg har bedre oversikt ved mange agenter.

Grunnet endringer i krav fra oppdragsgiver ble diagrammet som var utviklet

fjernet, og erstattet med bokser som representerte hver sin status og som inneholdt tall for antall agenter som er i en status.

Annet:

1. **Enhetstester:** Arbeidet med enhetstester ble videreført, parallelt med arbeid knyttet til sprinten.
2. **Integrasjonstester:** Arbeid med integrasjonstester ble startet på. Testcafe ble satt opp slik at testene kunne automatiseres.
3. **Endringer i diagramform i prototyper:** Etter endringer i krav fra oppdragsgiver ble prototypen for et wallboard med diagramform endret på.

Sprint review

Funksjonaliteten som var implementert gjennomgås. Det kom frem at filtreringslogikken måtte endres da utviker i Zisson hadde laget filtrering annerledes i databasen enn studentene hadde tiltenkt.

Sprint retrospective

Sprinten ble gjennomgått og det ble planlagt hvordan den neste sprinten skulle foregå med tanke på at det nå nærmet seg prosjektslutt. Det ble enighet om at fokus på testing var viktig. Testingen ble satt som egne brukerhistorier i sprintene, da testing knyttet til et inkrement i utgangspunktet ikke har vært et mål for om inkrementet ble godkjent.

1.6 Sprint 6 (19.april - 30.april)

Sprint planning

Følgende funksjonalitet ble valgt ut fra backlog:

1. Som administrator ønsker jeg å se ikoner for det nye og det gamle wallboardet i oversikten over wallboards for å kunne skille mellom de ulike typene.
2. Som administrator ønsker jeg å distribuere wallboardet via en unik URL-lenke til den som skal overvåke den, slik at de jeg sender den til får tilgang til det spesialtilpasset wallboard jeg opprettet.
3. Som administrator ønsker jeg å kunne klone det nye wallboardet for å raskt kunne opprette et nytt wallboard av samme type.
4. Som agentleder ønsker jeg å kunne oppdatere cache med nye agenter hvis det har kommet en endring på en wallboardkonfigurasjon som blir hentet fra databasen.
5. Som administrator ønsker jeg å kunne endre på et wallboard og da få inn de verdiene som tidligere var skrevet inn.
6. Som agentleder ønsker jeg å se agentstatus for å vite hva agentene foretar seg.
7. Som administrator ønsker jeg under oppretting/oppdatering av wallboard å kun se felt som er relevante dersom det er valgt diagramform.

Følgende funksjonalitet videreføres fra tidligere sprint:

1. Som administrator ønsker jeg å kunne filtrere på agentgruppe for å vise ønskede agenter på wallboardet.
2. Som agentleder ønsker jeg at agentene lagres til cache, slik at jeg ved tilkobling får inn de siste dataene som har vært registrert.
3. Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.

Annnet:

1. Integrasjonstest

2. Systemtest

Implementasjon

- Som administrator ønsker jeg å se ikoner for det nye og det gammel wallboardet i oversikten over wallboards for å kunne skille mellom de ulike typene.**

Under wallboard-liste, der alle opprettede wallboard kan sees i en tabell, ble det lagt til ikon i en ny kolonne kalt "Type" som indikerer om et wallboard er av typen gammelt wallboard eller nytt. Dette ga brukeren nyttig informasjon, da det ellers ikke var noen opplagt måte å se hvilken type et wallboard var,, sett bort i fra ved opprettelsen av et wallboard, eller i det et wallboard åpnes.

- Som administrator ønsker jeg å distribuere wallboardet via en unik URL-lenke til den som skal overvåke den, slik at de jeg sender den til får tilgang til det spesialtilpasset wallboard jeg opprettet.**

Etter at et wallboard av enten type nytt eller gammelt blir lagret, skal det distribueres. Modellen for wallboardet ble endret av teknisk veileder, og dette måtte derfor endres i både ReWave, samt server og klient for wallboardet. Det ble laget en "routing"-funksjonalitet hvor ReWave kobler seg til porten hvor wallboard-prosjektet kjører. Når det trykkes på "lenke"-ikonet inne på ReWave, åpnes denne porten og en wallboardKey, som er en unik nøkkel knyttet til wallboardet, legges på bakerst i URL-en. I WallboardService-prosjektet er det kode som da henter wallboardKey fra URL-en og får på denne måten tak i wallboardet som skal hentes fra databasen. Det blir så sendt til frontend og det valgte wallboardet blir visualisert på nettleseren.

3. Som administrator ønsker jeg å kunne klone det nye wallboardet for å raskt kunne opprette et nytt wallboard av samme type.

Det ble laget funksjonalitet for å klone et wallboard ved å trykke på “klone”-knappen i wallboard-listen. Da kommer man til oppretting av wallboard siden som da er av samme typen som det wallboardet som ble klonet. Kloning gjøres ved at det blir laget en ny routing i administration.routing-module. Dette gjør at hvis man trykker på “klone”-knappen, så vet systemet til hvilken del av nettsiden den skal gå til fordi det er laget en sti for å håndtere dette.

4. Som agentleder ønsker jeg å kunne oppdatere cache med nye agenter hvis det har kommet en endring på en wallboardkonfigurasjon som blir hentet fra databasen.

Cache ble videreutviklet til å kunne håndtere endringer i filtrering av agentgrupper. Dersom det gjøres endringer på hvilke agentgrupper som skal filtreres til et wallboard, må cache se etter endringer, for så å slette og opprette en ny cache for den nye filtreringen.

5. Som administrator ønsker jeg å kunne endre på et wallboard og da få inn de verdiene som tidligere var skrevet inn.

Ved ønske om endring på et wallboard blir det laget kode som fyller tekstfelt, knapper og lister med de verdiene som det valgte wallboardet opprinnelig har. På denne måten er det lett for brukeren å se hvilke valg som allerede eksisterer for wallboardet for så å gjøre de eventuelle endringene som er ønsket. Metoden det jobbes mest med er “isGroupSelected”, og er en metode som henter inn filteringsvalgene fra wallboardet, som ligger i et agentmapping-array som kun inneholder agentene sin id. Metoden må derfor hente ut alle agentene som eksisterer i systemet for så å sammenligne med

agentmapping-arrayet, slik at man får hentet ut agentene sine agentgrupper.

På denne måten får man fylt listen med de valgte agentgruppene.

6. Som agentleder ønsker jeg å se agentstatus for å vite hva agentene foretar seg.

En agent kan ha flere ulike statuser. Statusene har hele veien blitt hentet inn, men det er flere utfall som må håndteres for at statusene skal vises riktig i tabellen. Under statusen "pause" kan man velge mellom "pause", "personlig" og "kaffe". Fra server kommer dette kun inn som statusen "unavailable", men med en pause-id som skiller mellom de ulike pausetypeene. Det ble lagt inn kode for å håndtere dette, samt å håndtere at statusene "LoggedOn", "LoggedOff" og "LoggingIn" ikke vises i tabellen, da dette ikke er ønsket på wallboardet.

7. Som administrator ønsker jeg å under oppretting/oppdatering av et wallboard kun se felt som er relevante for diagramform.

Dersom det blir valgt at wallboardet skal vises i diagramform vil det på wallboardet kun vises bokser med ulike statusfarger, samt antallet agenter på hver status i. Dersom diagramform er huket av i konfigurasjonen på et wallboard blir det dermed overflødig å ha med videre konfigurasjon om sortering. Løsningen på dette ble å "skjule" elementene for sortering når checkbox-en for diagramform ble krysset av.

Inkrementer videreført fra forrige sprint:

1. Som administrator ønsker jeg å kunne filtrere på agentgruppe for å vise ønskede agenter på wallboardet.

Filtreringslogikken på server ble endret, da oppdragsgiver hadde utviklet filtreringslogikk i databasen på en annen måte enn hva som var tiltenkt.

Zisson har i sitt system, ReWave, laget filtrering slik at et wallboard sender med id til agentene som er med i de agentgruppene det er filtrert på.

Filtreringslogikken på server hadde opprinnelig denne logikken, og derfor måtte dette endres slik at server kun sjekket id-ene som kommer inn i et filtreringsmap opp mot id-ene på agentene som kommer inn som eventer.

2. Som agentleder ønsker jeg at agentene lagres til cache, slik at jeg ved tilkobling får inn de siste dataene som har vært registrert.

Koden tilhørende cache ble endret fra å bruke Singleton designmønster til å heller bruke Dependency injection for opprettelsen av Caches, som er en beholder for alle cachene til de ulike wallboardene. Caches-klassen blir en dependency som blir injected i startup.cs filen der programmet starter opp. Her kjøres metoden addSingleton og på denne måten blir det opprettet et objekt av Caches ved oppstart, som holder på alle cachene i en dictionary.

Tidligere har det kun blitt brukt singleton designmønster. Singleton er et designmønster brukt for å sørge for at det kun er mulig å opprette en instans av en klasse, slik at det kun er én forekomst av en gitt klasse. Mens dependency injection er et designmønster som står for å overføre avhengigheter til objekter som bruker dem. En avhengighet er et objekt som et annet objekt er avhengig av (Khorikov, 2016).

3. Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.

Studentene fikk hjelp til, i form av tilgang og endring av oppsett, å få tak i de virkelige dataene fra Zisson sin MQ, i stedet for å bruke RabbitMQ. Prosjektet har hittil testet med data som simulerte ekte data fra Zisson sin MQ, og det har ikke derfor den store betydning for utviklingen av prosjektet. Men det har

betydning for troverdigheten av integrasjon- og systemtest, samt sluttproduktet i sin helhet, og det var derfor viktig å få ordnet dette.

Sprint review

Det ble gjennomført et siste møte med Zisson. Her ble prosjektet i sin helhet gjennomgått og oppdragsgiver uttrykte at de er fornøyd med produktet og ønsket at dette overleveres til dem, slik at de kunne bruke prosjektet som en grunnlag for videre arbeid og utvikling.

Sprint retrospective

Sammen med Zisson delte alle noen tanker om hvordan prosjektet i sin helhet hadde vært, hva som har fungert bra og hva som kunne vært bedre. Zisson ga uttrykk for at de har vært fornøyd med studentenes innsats og samarbeid, og resultatet av prosjektet. De fortalte at det er utfordrende å gi en oppgave med et passende omfang, da det ofte er mye mer arbeid enn man ser for seg, noe studentene var enige i. Det har vært flere ting med prosjektet som har tatt tid, da prosjektgruppen har måttet vente på at Zisson skulle gi tilgang til verktøy eller at de har oppdatert sine systemer ut i fra de endringene som studentene har måttet gjøre. Men alt i alt er studentene fornøyd med oppgavens omfang og læringsutbytte.

2.0 Prosjektdagbok

Her ligger prosjektdagboken hvor gjøremålene for hver arbeidsdag er skrevet ned. Dagboken har fungert som en intern oversikt over dagens gjøremål, og for å holde en oversikt over hva som er gjort. Det er hovedsakelig jobbet med bacheloren fire dager i uken; mandag, onsdag, torsdag og fredag, med noen unntak. Dager som ikke er loggført har blitt brukt til å jobbe med andre fag. Dagboken holder en overordnet liste over arbeidsoppgaver som er gjennomført, hvor også noen av gruppens medlemmer har hatt egne, interne møter med oppdragsgiver. I disse tilfeller er navn nevnt. Ikke alle dager er fylt ut like detaljert, der noen dager kun nevner generelt hva som er jobbet med.

Uke 1

07.01.2021 - torsdag

- Daily Stand Up
- Denne uken her startet vi å jobbe med Forprosjektrapporten. Vi tenker å starte på den slik at vi kan være forberedt til første møte for året med Zisson og den interne veilederen vår. På denne måten kan vi ha klart evt spørsmål.
- Vi har fordelt oppgaver for forprosjektet.

08.01.2021 - fredag

- Daily Stand Up
- I dag fikk vi mail fra vår interne veileder om at vi skal ha møter på torsdager mellom kl. 15:00-16:00. På fredag uke 2 skal vi ha møte med alle gruppene som har samme veileder og planlegge veien videre og stille eventuelle spørsmål vi skulle ha.
- Holder samtidig på å planlegge et møte med Zisson og veileder.
- I tillegg har vi jobbet mer med utforming av arbeidsplanen. Venter på kravspesifikasjonen fra Zisson sin produktavdeling.

Uke 2

11.01.2021 - mandag

- Daily Stand Up
- I dag definerte vi ulike roller og fordele hovedansvar.
- Vi har også jobbet mer med forprosjektrapporten slik at den kan sendes inn til Zisson før møtet på tirsdag.

12.01.2021 - tirsdag

- Daily Stand Up
- Planleggingsmøte med Zisson. Fikk mer informasjon om hvordan de tenker vi skal løse oppgaven, samt informasjon om ulike punkter relatert til forprosjektrapporten. Zisson sender oss dokument med det de har kommet frem til, og vi sender Zisson fremdriftsplanen.
- Zisson etterspør møte med veileder for å få skrevet en NDA kontrakt slik at vi kan få tilgang til utviklingsmiljøet deres. Martine sender mail til veileder og spør om dette.
- Jobber videre med forprosjektrapporten.
- Vi har også avtalt et møte internt i gruppa på torsdag for å ferdigstille forprosjektet slik at vi kan sende det videre til Zisson og den interne veilederen vår.

13.01.2021 - onsdag

- Daily Stand Up
- Zisson har sendt NDA-avtale sammen med prosjektdokument.
- Vi skriver under på OsloMet-avtale og NDA, og sender dette videre til veileder og Zisson for videre signering.

- Møte med alle parter involvert, torsdag 21.januar kl. 15.30, er bekreftet med veileder og gruppemedlemmer. Avventer en siste bekreftelse fra Zisson på dette tidspunktet.

14.01.2021 - torsdag

- Daily Stand Up
- Gruppemedlemsmøte der vi jobber med/ ferdigstiller forprosjektsrapporten. Forhåpentligvis blir forprosjektet klart for sending til veileder og Zisson for tilbakemeldinger.

15.01.2021 - fredag

- Daily Stand Up
- Gruppemedlemsmøte på teams fra 09.00 til 12.00.
- Vi jobbet med Forprosjektet og ferdigstilte det første utkastet. Det ble deretter sendt til Zisson med spørsmål forskjellige spørsmål som vi lurte på.
- Videre opprettet vi Prosjektrapport, Produktdokumentasjon og Testdokumentasjon, og satte opp strukturen på disse.
- Kickstart-møte med veileder 13.30-14.30.

Uke 3

18.01.2021 - mandag

- Daily Stand Up
- Vi begynte med designprosessen med å lage empathy map i Miro, innhente informasjon om brukere og lage personas.
- Gruppemedlemmene jobber hver for seg med personas.
- Lager 4 personas som senere skal gjennomgås i fellesskap på teams for bli enige om utfyllende informasjon.

19.01.2021 - tirsdag

- Satt opp planleggingsverktøy: opprettet Trello-tavle for å kunne sette opp oppgave-lister.

20.01.2021 - onsdag

- I dag ble det avtalt at vi skal ha et møte med Zisson på onsdag neste uke etter klokken 12.30 for å lage brukerhistorier sammen.
- Fikk tilbakemeldinger fra veileder på forprosjektet, endret litt på noen punkter i rapporten.

21.01.2021 - torsdag

- Gruppemedlemmene møttes for utarbeiding og ferdigstilling av personas, samt forprosjektrapporten.
- Utover dagen var det også et møte sammen med veileder og Zisson der gikk vi gjennom samarbeidet videre framover og hvordan dette best burde løses (f.eks. metodikk og MVP), samt det formelle.

22.01.2021- fredag

- Lagt inn litt om utviklingsprosessen i sluttrapporten. Herunder første steget i designprosessen.
- Vi fikk tilgang til Jira. Utforsket Jira.

Uke 4

25.01.2021 - mandag

- Jobbet med prosjektrapporten, herunder steget fremdrift- og arbeidsplan.
- Opprettet referanseliste i sluttrapporten.
- Startet her også å skrive litt på tekstdokumentet, laget et oppsett for dokumentet og skrev litt om fremgangsmåten på noen av testene.

27.01.2021 - onsdag

- Møte for og sammen lage brukerhistorier kl 11.00.
- Skrev ned spørsmål til Zisson.
- Gruppen forberedte seg til møtet med å skrive opp forslag til brukerhistorier på forhånd.
- Møte med Zisson 12.30 for å utvikle brukerhistorier sammen i JIRA.
- Satte opp backlog og epics som skal brukes i sprintene.
- Thomas skal gå igjennom nærmere for å gi oss tilbakemeldinger.

28.01.2021 - torsdag

- Til fredag skal alle i gruppa lese seg opp på flytdiagram og use case diagram og lage disse hver for seg slik at vi kan gå gjennom de på fredag og utarbeide en endelig versjon.
- Vi begynner å lage flytdiagram og sekvensdiagram.
- Kontakter Ola for å purre på en presentasjon vi ble vist under første møtet, som er interessant for oss å ha under designprosessen.

29.01.2021 - fredag

- Møte innad i gruppa for å se gjennom diagrammene vi har laget og utarbeide de ferdig sammen.
- Laget aktivitetsdiagram, sekvensdiagram og use case diagram.
- Delte inn hvem som skulle skrive om hva i sluttrapporten.
- Endret på struktur i sluttrapporten.

Uke 5: Sprint 1 - Design, brukertester og teknisk oppsett

01.02.2021 - mandag

- Daily Stand Up

- Hatt intern workshop hvor vi utarbeidet skisser til prototyper, og begynte å utforme utkast til første prototype som skal testes.
- Sendt mail til Thomas ang. tilgang til wave og kontaktpersoner ifm. testing.
- Skrevet om skisser, workshops og brainstorming i sluttdokumentet.

02.02.2021 - tirsdag

- Daily Stand Up
- Møte med Zisson (Ola og Thomas). Fikk tydeligere definisjon på MVP og at det skal være fokus på agentoversikten på wallboardet i første omgang.
- Fikk tilsendt powerpoint fra produkt.
- Fikk beskjed om at vi skulle få tilsendt kontaktpersoner.
- Kom med ønsker om at wallboardet også burde vise en oversikt over agentgrupper. (må legges inn i kravspesifikasjonen). Samt nice to have: bytte av bakgrunnsbilde og mulighet for forstørrelser, flytting av komponenter.

03.02.2021 - onsdag

- Daily Stand Up
- I dag har gruppa utarbeidet flere prototyper av komponenten "agentoversikt".
- Vi har også tatt kontakt med veilederen for å ha et møte på torsdag og sendt mail til Kim som jobber i Zisson, for å diskutere prototypene.
- Gjorde endringer på prototypene ut i fra informasjon fra møtet tirsdag. Fokus på agentoversikt.
- Laget spørreundersøkelse til testing og sende ut undersøkelser eller avtale tid for intervjuer.
- Se over kravspesifikasjon. Dele inn i must haves og nice to have.
- Begrunnet hvorfor vi valgte den prototypen å jobbe videre med.

04.02.2021 - torsdag

- Daily Stand Up
- Workshop med Kim.
- Fortsette på utforming av prototypen.
- Sendte ut undersøkelser eller avtale tid for intervju - Christer.
- Så over kravspesifikasjon. Delte inn i must haves og nice to have.
- Veiledningsmøte
- Begrunnet hvorfor vi valgte den prototypen å jobbe videre med, og gjøre research til fredag.
- Klokken 10 i dag har gruppa vært i møte med Kim og diskutert prototypene vi hadde utarbeidet på Miro. Vi fikk mye bra tilbakemelding som vi jobbet med videre etter møte.
- Klokken 15 har vi møte med veilederen hvor vi skal diskutere veien videre når det gjelder prototyping.

05.02.2021 - fredag

- Daily Stand Up
- I dag møttes gruppa klokken 09:00 for å jobbe videre med prototypene etter å ha fått gode tilbakemeldinger fra Kim.
- Laget prototyper av filtrering av agentgruppe.
- Fikset opp i kravspesifikasjon (MVP, must haves, nice to have, lage mer konkrete brukerhistorier). Endret disse i Jira.
 - Sendte til Thomas & Kim
- Forberedte brukertester/spørreundersøkelser
- Startet på test-planlegging

Over helgen:

- Brukerteste/ vise frem og stille spørsmål

Uke 6

Fokus på design/ brukertester/ oppsett av tekniske programmer/ test-planlegging.

08.02.2021 - mandag

- Daily Stand Up
- I dag har vi gått igjennom resultatene fra brukertestene vi gjorde i helgen. Ut i fra dette videreutviklet vi prototypen og lagde en brukertest som vi skal bruke på Christer fra Zisson.
- Vi har også startet på testrapporten og laget en plan for testingen.
- Endret prototype
- Planlagt møte med Christer
- Planlagte sprintreview møte med Thomas.
- Startet på analyse av prototypen/ heuristisk evaluering av universell utforming, klart til onsdag
- Ryddet opp i prosjektrapporten og sendte til Aiko.

09.02.2021 - tirsdag

- Daily Stand Up
- Alle jobbet videre hver for seg med en heuristisk evaluering av prototypen.

10.02.2021 - onsdag

- Daily Stand Up
- I dag gikk vi først igjennom den heuristiske evalueringen sammen og la til endringer.
- Etter dette hadde vi møte med Zisson der vi brukertestet Christer. Han hadde mange tilbakemeldinger og mye nye tanker, som var veldig annerledes enn hva Kim hadde gitt oss av tilbakemelding.
 - Ut i fra dette lagde vi noen nye forslag til prototyper.

- Mot slutten av dagen brukte vi litt tid på å rydde opp i rapporten.
- Avtale møte med Kim, Thomas.

11.02.2021 - torsdag

- Daily Stand Up
- Møte med Kim.
 - Diskuterte tilbakemeldinger fra Christer og videreutviklet prototypen.
- Møte med veileder kl. 16.30, fikk tilbakemelding om å tenke mer over hvordan brukerteste produktet.
 - Gikk igjennom:
 - Testplanlegging
 - Kravspek til MVP
 - Vise prototype

12.02.2021 - fredag

- Daily Stand Up
- Testplan (oppdaterte denne og sendte til Aiko)
- Snakket med Thomas om MVP og brukerteste (Thomas og Kim)
- Diskuterte UX-testing og hvordan det skal gjøres(i detalj)

Uke 7: Sprint 2 - Utvikling av kode og testing av prototyper

Agenda for uken:

- Lage wave innstillinger side som prototype.
- Endre i jira hvilke kolonner som skal være med som del av MVP.
- Skrive ferdig testplan.
- Fylle inn i rapporten.
- Følge sprinten i jira.
- Gjøre om på diagrammer.

15.02.2021 - mandag

- Daily Stand Up
- Sprint planleggingsmøte med Zisson
 - Ble enig om MVP
 - Viste frem prototypen
 - Spurte om mulighetene for å begynne utviklingen

16.02.2021 - tirsdag

- Daily Stand Up
- Videre sprintplanlegging med Thomas
 - Viste forslag til modell, spurte om spørsmål relatert til den
 - Definerte en ferdig sprint og hva den skal inneholde
- Gruppe jobber videre med å oppdatere diagrammene slik at de samsvarer med den endelige kravspesifikasjonen og lager GUI for wallboard-innstillinger inne på wave.

17.02.2021 - onsdag

- Daily Stand Up
- Møte med Moritz etter 14:15
 - Fikk forklart litt av koden og veien videre for oss når vi skal starte på utviklingen.
 - Gruppa har deretter lastet ned repository-en og opprettet et angular prosjekt slik at vi kan starte å jobbe med utviklingen.
- Fordelte oppgaver fra sprinten
- Gruppa jobbet med testdokumentasjonen og fylte inn noen punkter som manglet.

18.02.2021 - torsdag

- Daily Stand Up
- I dag har gruppa jobbet sammen med å prøve å sette opp websocket.

Uke 8

Agenda for uken:

- Følge sprint i Jira
- Sprintreview-møte fredag.
- Lage geriljatester.
- Fylle inn i rapporten.

22.02.2021 - mandag

- Daily Stand Up
- Satt opp websocket.
- Nikola hadde møtet med Moritz og fikk oppklaring i en del ting vi lurte på.
Fikk beskjed om at vi må få tilgang til VPN for å få hentet "ekte" data.

24.02.2021 - onsdag

- Daily Stand Up
- Satt opp vår egen rabbitmq mens vi venter på å få tilgang til VPN.
- Fikk hentet data fra dette og ut til frontend, tenker da at vi har fått til oppgaven om initialisering.

25.02.2021 - torsdag

- Daily Stand Up
- Så på hvordan vi skal håndtere cache
- Reconnect ved feilet tilkobling
- Sendte mail om møte med Zisson fredag

26.02.2021 - fredag

- Sprint review møte med Zisson
 - Vise prototype fra Wave
 - Gå igjennom implementasjoner gjort i sprinten
 - Avtale sprint planning for mandag

Uke 9: Sprint 3

01.03.2021 - mandag

- Sprint planleggingsmøte
- Sendt rapport til veileder
- Initialisering av agentdata

03.03.2021 - onsdag

- Daily Stand Up
- Ble opprettet mockup av wallboardet med agenttabellen på frontend
- Ble opprettet en mockup config på server-siden, som filtrerer agenter og forteller GUI hvordan ting skal se ut
- Startet å se på caching på server-siden
- Gjort endringer på prototype
- Fylt inn i Sluttrapporten
- Synne møte med Thomas
- Fylt inn siste oppdatering av prototyper til Vedlegg-dokumentet

04.03.2021 - torsdag

- Daily Stand Up
- Ble gjennomført møte for å fordele flere tasks og usecases fra backlogen til sprinten.
- Hver og en jobber med oppgavene fra backlogen.

05.03.2021 - fredag

- Fikset initialisering, og dynamisk farge på status
- Fikset Angular-tabell og gjøre den responsiv
- Startet å merge sammen (mockup-)tabellen og initialisering av data

Uke 10

08.03.2021 - mandag

- Daily Stand Up
- Det ble fordelt videre oppgaver og det ble jobbet både på frontend og backend.
- I backend har to på grupper jobbet med cache og debugging
- To på grupper jobbet med agenttabellen - og få agentdata initialisert

09.03.2021 - tirsdag

- Daily Stand Up
- Fortsatte å jobbe med backend (cache) og frontend (agenttabellen)
- Initialisering i Angular-tabellen fikset

10.03.2021 - onsdag

- Daily Stand Up
- Fordelte nye tasks fra backlog
 - Startet på unittest backend
 - Startet på å lage diagram som viser agenter med status (og antall) på frontend i agenttabellen
 - Startet med å fikse add eller update på frontend av agentstatus

11.03.2021 - torsdag

- Daily Stand Up
- Jobbet videre med frontend

12.03.2021 - fredag

- Daily Stand Up
- Sprint review møte
- Lest om testCafe på frontend
- lest om spec.ts tester på frontend.

[Uke 11: Sprint 4](#)

15.03.2021 - mandag

- Sprint 3 planleggingsmøte.
- Valgt ut brukerhistorier til sprinten.
- Alle fikk bruker på wave og har blitt bedre kjent med programmet.
- Lest om enhetstester på frontend og prøvd å skrive noen tester.
- Sendt mail til veileder om spørsmål til rapporten.
- Testet vpn mot wallboard.
- Nikola møte med Moritz.

17.03.2021 - onsdag

- Daily Stand Up
- Vi jobber selvstendig med ulike inkrementer.
- Zisson har problemer med Wave (Zisson Interact), så man får ikke hentet ut eventer.
- Jobber derfor videre med RabbitMQ.
- Jobbet med frontend: gjøre tabellen responsiv + startet på å få opp en vars melding hvis backend disconnecter
- Jobbet på tid i status

18.03.2021 - torsdag

- Daily Stand Up
- Møte med Thomas fra Zisson for gjennomgang av de to nye prosjektene vi har fått som tilhører Zisson Interact systemet.
- Jobbet med frontend: få til en varselmelding hvis backend disconnecter, fullført
- Fikset på agent-tabellen etter endring i krav fra oppdragsgiver: tatt bort to kolonner
- Jobbet på tid i status
- Opprettet og startet også på en presentasjon til midtveisevaluering og -presentasjon

19.03.2021 - fredag

- Daily Stand Up
- Jobbet videre på tid i status.

Uke 12

Jobbe med midtveisevaluering

22.03.2021 - mandag

- Daily Stand Up
- Planlegger videre arbeid for uka
- Utdelte nye tasks og brukerhistorier i Jira
- Alle fikk satt opp Wave i backend og frontend
- Jobbet med frontend i Wave: lage ny settingsside for oppretting av wallboard (tilpasset vår Config-modell)

24.03.2021 - onsdag

- Daily Stand Up
- Delte problemstillinger - hadde noen problemer
- Jobbet med midtveispresentasjonen

25.03.2021 - torsdag

- Daily Stand Up
- Jobbet videre med sortering
- Jobbet med ReWave

26.03.2021 - fredag

- Sprint 3 review-møte (bachelorgruppen)
- Ferdigstilte midtveis-presentasjonen
- Nikola hadde et møte med Moritz, gikk igjennom endringer han skulle gjøre i ReWave

Uke 13 - Påske

- Påskeferie

Uke 14: Sprint 5

06.03.2021 - tirsdag

- Gjennomført midtveisfremføring med veileder og de andre gruppene.
- Research sortering, og enhetstester frontend og opplasting av logo.

7.04.2021 - onsdag:

- Sprint planleggingsmøte med Zisson, kl. 11.00-12.00
 - Sprint review
 - Sprint planlegging

- Fordeling av oppgaver).

8.04.2021 - torsdag

- Daily Stand Up
- Fikset sort på frontend
- Fikset endringer fra Thomas etter sprintmøtet på onsdag - ifbm diagram-form (laget ny prototype på Miro og fikset i koden)
- Laget nytt oppsett for enhetstesting etter at nye funksjoner ble implementert i frontend
- Fikset på kolonnen Callstatus på frontend etter ønske fra Thomas
- Jobbet med config i ReWave
- Jobbet med innsett av logo i ReWave
- Nikola møte med Moritz angående routing i AgentRealTimeWallboard prosjektet

9.04.2021 - fredag

- Daily Stand Up
- Fortsetter med enhetstesting i Angular
- Fortsetter med ReWave

Uke 15

12.04.2021 - mandag

- Daily Stand Up
- Starte på integrasjonstesting, herunder lest litt om testCafé
- Lagt inn flere testscenarioer i exceldokumentet.
- Skrevet litt om research i sluttrapporten.

13.04.2021 - tirsdag

- Daily Stand Up
- Prøvde å løse Plasterboard modell i angular.
- Skrevet litt om research i rapporten.
- Gjorde noen små endringer i diagramform.
- Nikola møte med Thomas.

14.04.2021 - onsdag

- Daily Stand Up
- Begynte på integrasjonstester frontend Wallboard

15.04.2021 - torsdag

- Daily Stand Up
- Enhetstester backend.
- Integrasjonstester.

16.04.2021 - fredag

- Daily Stand Up
- Jobbet med integrasjonstest frontend Wallboard
- Jobbet med excel-dokumentet for integrasjonstest
- Satte opp Systemtest-excel-dokument
- Sprint reviewmøte med Zisson

[Uke 16: Sprint 6 - Siste sprint](#)

Agenda for uka:

- Fokus på integrasjonstester og systemtester.
- Overlevering av produkt.
- Jobbe med testing

19.04.2021- mandag

- Sprint planleggingsmøte med Zisson
- Startet på Sytemtester
- Enhetstester på backend.

21.04.2021 - onsdag

- Daily Stand Up
- Jobbet med set up av default config i wave
- Ferdigstilling av ikon-funksjonalitet for nytt wallboard i tabell i ReWave
- Endre filtreringslogikk på server
- Jobbet med esting

22.04.2021 - torsdag

- Daily Stand Up
- Skrevet på slutt dokumentasjon, testdokumentasjon og slutt rapport.
- Config i reWave
- Jobbe med editsiden: Sette opp wallboard og distribuere
- Jobbet med testing

23.04.2021 - fredag

- Daily Stand Up
- Jobbet med testing, både integrasjon og enhets

Uke 17

26.04.2021 - mandag

- Daily Stand Up
- Integrasjonstesting
- Finpusset på småfeil som dukket opp

27.04.2021 - tirsdag

- Daily Stand Up
- Jobbet videre med småfeil
- Jobbet med config i Wave
- Testing

28.04.2021 - onsdag

- Endret på cache etter tilbakemeldinger fra Moritz.
- Systemtester
- Jobbet med kravene fra sprint.
- Testing

29.04.2021 - torsdag

- Daily Stand Up
- Jobbet videre med småjusteringer
- Cache
- Testing

30.04.2021 - fredag

- Daily Stand Up
- Testing
- Småjusteringer og feil i cache og config
- Nikola og Synne møte med Moritz angående routing i AngularRealTimeWallboard prosjektet og det å få sendt nøkkelen til wallboard service.

Uke 18

Fokus på Systemtesting

03.05.2021 - mandag

- Daily Stand Up
- Systemtester
- Jobbet med dokumentasjoner

04.05.2021 - tirsdag

- Sprint review og siste møtet med Zisson.
- Fremvisning av system.
- Endringer etter tester.

05.05.2021 - onsdag

- Daily Stand Up
- Systemtester
- Ryddet i kravspesifikasjon.
- Enhetstesting av backend
- Jobbet med dokumentasjon og rapporter

06.05.2021 - torsdag

- Daily Stand Up
- Rydde i testdokumentasjon.
- Enhetstesting av backend
- Småfix på ReWave
- Testing
- Jobbet med dokumentasjon og rapporter

07.05.2021 - fredag

- Enhetstesting av backend
- Siste finish av småfeil og justeringer
- Jobbet med dokumentasjon og rapporter

Uke 19 - Rapport og dokumentasjon

10.05.2021 - mandag

- Laget video der vi har en demo av systemet.
 - Alle måtte bidra med å være agenter som skifter statuser i sanntid
- Jobbet med rapporter og dokumentasjoner

11.05.2021 - tirsdag

- Jobbet med rapporter og dokumentasjoner

12.05.2021 - onsdag

- Jobbet med rapporter og dokumentasjoner

14.05.2021 - fredag

- Jobbet med rapporter og dokumentasjoner
- Mål å ferdigstille første utkast for vurdering og gjennomgang av veileder

Uke 20 og 21

- Sende inn utkast til veileder (tirsdag 18.05)
- Jobbet med eventuelle justeringer som trengs
- Jobbe mot ferdigstilling av rapporter og dokumentasjoner
- Levere prosjektrapport

Uke 22 og 23

- Jobbe med og ferdigstille fremføringen av prosjektet fram mot presentasjonsdag

3.0 Arbeidsplan

UKE	DATO	OPPGAVER
2	11. - 17.01	<p><u>Planleggingsfase</u></p> <p>Arbeide med forprosjektet:</p> <ul style="list-style-type: none"> - Møte med Zisson 12.01 - Skrive på forprosjektet - Lage arbeidsplan - Kickstart-møte med veileder 15.01
3	18. - 24.01	<p><u>Planleggingsfase</u></p> <p>Arbeide med forprosjektet:</p> <ul style="list-style-type: none"> - Sende forprosjekt til veileder minimum to dager før 23.01 - Levere forprosjekt innen 23.1 - Ferdigstille arbeidsplan - Møte med veileder og Zisson: <ul style="list-style-type: none"> - Skrive under NDA - Spørre Zisson noterte spørsmål som har kommet opp under planleggingen <p>Utarbeidelse av kravspesifikasjon:</p> <ul style="list-style-type: none"> - Avtale med Zisson møte for å definere brukerhistorier og krav sammen, samt finne ut av mulighetene for om vi kan komme i kontakt med noen brukere av systemet.
4	25. - 31.01	<p><u>Planleggingsfase</u></p> <p>Utarbeidelse av kravspesifikasjon:</p> <ul style="list-style-type: none"> - Datainnsamling: Sette oss inn i brukerens behov, få informasjon fra Zisson og evt. snakke med noen brukere - Lage brukerhistorier - Møte med Zisson for å utarbeide brukerhistorier sammen <p>Oppsett av programmer:</p> <ul style="list-style-type: none"> - Installere og sette opp all nødvendig programvare - Bli kjent med programvare vi er ukjent med - Få tilgang av Zisson til Git og andre ressurser/verktøy - Avtale med Zisson vedrørende Sprint planleggingsmøte <p>Arbeide med sluttrapport</p>
5	01. - 07.02	<p><u>Gjennomføringsfase (Sprint 1)</u></p> <p>Sprint planleggingsmøte</p> <p>Utarbeidelse av kravspesifikasjon:</p> <ul style="list-style-type: none"> - Lage brukerhistorier

		<p>Oppsett av programmer:</p> <ul style="list-style-type: none"> - Dersom noen har problemer eller mangler programvare skal dette ordnes denne uken - Sette oss inn i programvare vi er ukjent med <p>Prototyping:</p> <ul style="list-style-type: none"> - Workshop med free thinking - Skissere forslag til prototyper: lage low-fi prototyper - Bli enig om hvilken vi liker best, be om tilbakemelding fra Zisson - Lage high-fi prototype <p>Arbeide med sluttrapport</p>
6	08. - 14.02	<p><u>Gjennomføringsfase (Sprint 1)</u></p> <p>Prototyping:</p> <ul style="list-style-type: none"> - Lage high-fi prototype - Be om tilbakemelding fra Zisson - Brukerteste prototypen <p>Sprint review møte</p> <p>Arbeide med sluttrapport</p>
7	15. - 21.02	<p><u>Gjennomføringsfase (Sprint 2)</u></p> <p>Sprint planleggingsmøte:</p> <ul style="list-style-type: none"> - Velge ut oppgaver fra backlog <p>Internt møte:</p> <ul style="list-style-type: none"> - Delegere oppgavene som er valgt ut fra sprint backlog <p>Prototyping:</p> <ul style="list-style-type: none"> - Brukerteste prototypen - Gjøre endringer dersom det er behov - Ferdigstille prototypen <p>Arbeide med sluttrapport</p>
8	22. - 28.02	<p><u>Gjennomføringsfase (Sprint 2)</u></p> <p>Utvikling</p> <p>Sprint review møte</p> <p>Arbeide med sluttrapport</p>
9	01. - 07.03	<u>Gjennomføringsfase (Sprint 3)</u>

		<p>Sprint planleggingsmøte:</p> <ul style="list-style-type: none"> - Velge ut oppgaver fra backlog <p>Internt møte:</p> <ul style="list-style-type: none"> - Deleger oppgavene som er valgt ut fra sprint backlog <p>Utvikling</p> <p>Arbeide med sluttrapport</p>
10	08. - 14.03	<p><u>Gjennomføringsfase (Sprint 3)</u></p> <p>Utvikling</p> <p>Sprint review møte</p> <p>Arbeide med sluttrapport</p>
11	15. - 21.03	<p><u>Gjennomføringsfase (Sprint 4)</u></p> <p>Sprint planleggingsmøte:</p> <ul style="list-style-type: none"> - Velge ut oppgaver fra backlog <p>Internt møte:</p> <ul style="list-style-type: none"> - Deleger oppgavene som er valgt ut fra sprint backlog <p>Utvikling</p> <p>Testing: Lage enhetstester før koden produseres</p> <p>Arbeide med sluttrapport</p>
12	22. - 28.03	<p><u>Gjennomføringsfase (Sprint 4)</u></p> <p>Utvikling</p> <p>Sprint review møte</p> <p>Arbeide med sluttrapport</p>
13	29.02 - 4.04	Påskeferie
14	05. - 11.04	<p><u>Gjennomføringsfase (Sprint 5)</u></p> <p>Sprint planleggingsmøte:</p> <ul style="list-style-type: none"> - Velge ut oppgaver fra backlog <p>Internt møte:</p> <ul style="list-style-type: none"> - Deleger oppgavene som er valgt ut fra sprint backlog <p>Utvikling</p>

		<p>Testing: Enhetstest</p> <p>Arbeide med sluttrapport</p>
15	12. - 18.04	<p><u>Gjennomføringsfase (Sprint 5)</u></p> <p>Utvikling</p> <p>Testing: Fullstendig test av systemet</p> <p>Sprint review møte</p> <p>Arbeide med sluttrapport</p>
16	19. - 25.04	<p><u>Gjennomføringsfase (Sprint 6)</u></p> <p>Sprint planleggingsmøte:</p> <ul style="list-style-type: none"> - Velge ut oppgaver fra backlog <p>Internt møte:</p> <ul style="list-style-type: none"> - Delegere oppgavene som er valgt ut fra sprint backlog <p>Utvikling: Rette opp i feil oppdaget ved testing?</p> <p>Testing</p> <p>Sprint review møte</p> <p>Arbeide med sluttrapport</p>
17	26.04 - 02.05	<p><u>Avslutningsfase</u></p> <p>Sluttrapport:</p> <ul style="list-style-type: none"> - Internt møte: Delegere arbeid for sluttrapporten
18	03. - 09.05	<p><u>Avslutningsfase</u></p> <p>Sluttrapport:</p> <ul style="list-style-type: none"> - Skrive på sluttrapport
19	10. - 16.05	<p><u>Avslutningsfase</u></p> <p>Sluttrapport:</p> <ul style="list-style-type: none"> - Skrive på sluttrapport
20	17. - 23.05	<p><u>Avslutningsfase</u></p> <p>Sluttrapport:</p> <ul style="list-style-type: none"> - Gjennomgå sluttrapporten - Sende sluttrapport til veileder for tilbakemeldinger
21	24. - 30.05	<p><u>Avslutningsfase</u></p>

		Sluttrapport: Innlevering 25.mai - Begynne på presentasjon
22	31.05 - 6.06	- Jobbe med presentasjon
23	07 - 10.06	- Jobbe med presentasjon - Presentasjon

4.0 Dokumentasjon av tester

4.1 Skjermbilder av tester i Katalon

A - seSanntidsdata

Se [Vedlegg: Systemtest-dokumentasjon \(ekstern lenke\)](#) for dokumentasjon.

B - seProfil

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
AssertText	//main[@id='agentTable']/div/table/tbody/tr/td[4]	All queue

Figur 82: seProfil test i katalon

C - seAgentgruppe

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//tr[14]/td[4]/span/a/mat-icon	
assertText	//main[@id='agentTable']/div/table/tbody/tr/td[5]	Maria Sin Group

Figur 83: seAgentgruppe test i katalon

D - seAgentstatus: Som bruker ønsker jeg å se agentstatus

1.0 seAgentStatus: Available

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
assertText	//main[@id='agentTable']/div/table/tbody/tr[2]/th	Available

Figur 84: seAgentStatus available test i katalon

2.0 seAgentStatus: Unavailable

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
assertText	//main[@id='agentTable']/div/table/tbody/tr/th	Unavailable

Figur 85: seAgentStatus unavailable test i katalon

3.0 seAgentStatus: Pause

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
AssertText	//main[@id='agentTable']/div/table/tbody/tr[3]/th	Pause

Figur 86: seAgentstatus pause test i katalon

4.0 seAgentStatus: WrapUp

Command	Target	Value
open	http://localhost:4200/?wallboardKey=9a3a0785-ef65-4e50-9479-016359b75340	
assertText	//main[@id='agentTable']/div/table/tbody/tr/th	WrapUp

Figur 87: seAgentstatus WrapUp test i katalon

5.0 seAgentStatus: AfterCallWork

Command	Target	Value
open	http://localhost:4200/?wallboardKey=9a3a0785-ef65-4e50-9479-016359b75340	
assertText	//main[@id='agentTable']/div/table/tbody/tr/th	AfterCallWork

Figur 88: seAgentStatus AfterCallWork test i katalon

6.0 seAgentStatus: Personlig

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
AssertText	//main[@id='agentTable']/div/table/tbody/tr/th	Personlig

Figur 89: seAgentStatus personlig test i katalon

7.0 seAgentStatus: Kaffe

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
AssertText	//main[@id='agentTable']/div/table/tbody/tr[2]/th	Kaffe

Figur 90: seAgentStatus Kaffe test i katalon

E - seTidlAgentstatus

Manuelt testet, se [Vedlegg: Systemtest-dokumentasjon \(ekstern lenke\)](#).

F - seAgentnavn: Som bruker ønsker jeg å se agentnavn

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
AssertText	//main[@id='agentTable']/div/table/tbody/tr[4]/td	C:

Figur 91: seAgentnavn test i katalon

G- setUpWallboard: Som administrator ønsker jeg å sette opp et wallboard

Kommentar: fikk ikke kjørt "open localhost" pga feil på Zisson sitt system, men får åpnet det manuelt og kjørt testen fra der.

Command	Target	Value
open	http://localhost:30200/	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//input[@type='text']	
type	//input[@type='text']	TestWallboard
click	//mat-radio-button[@id='mat-radio-5']/label/div/div	
click	//div/div[2]/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
assertText	//tr[last()]/td	TestWallboard

Figur 92: setUpWallboard test i katalon

H - URLsetUpWallboard: Som administrator ønsker jeg å se wallboard uten diagram

1.0 Som tabell

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGro up=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//tr[16]/td[4]/span/a/mat-icon	
assertText	//main[@id='agentTable']/div/table/thead/tr/th/div/div	Agentnavn

Figur 93: URLsetUpWallboard i tabellform test i katalon

I - setUpWallboardDiagram: Som administrator ønsker jeg å sette opp et wallboard i diagramform

Kommentar: fikk ikke kjørt "open localhost" pga feil på Zisson sitt system, men får åpnet det manuelt og kjørt testen fra der.

Command	Target	Value
open	http://localhost:30200	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/span	
click	//wave-mdi	
click	//wave-menu-page	
type	//input[@type='text']	TestDiagramWallboard
click	//mat-radio-button[@id='mat-radio-2']/label/div/div	
click	//div[2]/wave-checkbox/wave-mdi	
click	//div/div[2]/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
assertText	//tr[last()]/td	TestDiagramWallboard

Figur 94: URLsetUpWallboard i diagramform test i katalon

J - URLsetUpWallboardDiagram: Som administrator ønsker jeg å se wallboard med diagram

2.0 Som diagram

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[19]/td[4]/span/a/mat-icon	
assertElementPresent	//th[@id='AgentState']/div/div/div/div	true

Figur 95: URLsetUpWallboardDiagram i diagramform test i katalon

K - leggeTilLogo

Manuelt testet, se [Vedlegg: Systemtest-dokumentasjon \(ekstern lenke\)](#).

L - sortereFelter: Som administrator ønsker jeg å velge kolonner jeg vil sortere på wallboardet.

1.0 SorterePåAgentnavn

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//wave-row[2]/div[2]/div	
type	//input[@type='text']	SorterAgentnavn
click	//mat-radio-button[@id='mat-radio-3']/label/div/div	
click	//div/div/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
click	//tr[14]/td[4]/span/a/mat-icon	
AssertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th/div/div[2]	true

Figur 96: SorterePåAgentnavn test i katalon

2.0 SorterePåStatus

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//wave-row[2]/div[2]/div	
type	//input[@type='text']	SorterStatus
click	//mat-radio-button[@id='mat-radio-5']/label/div/div	
click	//div/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
click	//tr[16]/td[4]/span/a/mat-icon	
assertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th[2]/div/div[2]/div	true

Figur 97: SorterPåStatus test i katalon

3.0 SorterPåAgentrgruppe

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	link=Wallboard-list	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//input[@type='text']	
type	//input[@type='text']	SorterAgentgruppe
click	//div/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
click	//tr[15]/td[4]/span/a/mat-icon	
assertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th[6]/div/div[2]/div	true

Figur 98: SorterOpAgentgruppe test i katalon

4.0 SorterPåCallstatus

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div	
click	link=Wallboard-list	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//wave-menu-page	
type	//input[@type='text']	SorterCallstatus
click	//div/div/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
click	//tr[14]/td[4]/span/a/mat-icon	
assertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th[4]/div/div[2]/div	true

Figur 99: SorterPåCallstatus test i katalon

5.0 SorterPåTidIStatus

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//input[@type='text']	
type	//input[@type='text']	SorterTidIStatus
click	//div/div/div/wave-checkbox/wave-mdi	
click	//mat-radio-button[@id='mat-radio-5']/label/div/div	
click	//wave-toolbar/div/button	
click	//tr[13]/td[4]/span/a/mat-icon	
assertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th[3]/div/div[2]	true

Figur 100: SorterPåTidIStatus test i katalon

6.0 SorterPåProfil

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//mat-radio-button[@id='mat-radio-7']/label/div/div	
click	//wave-menu-page	
type	//input[@type='text']	SorterProfil
click	//div/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
click	//tr[13]/td[4]/span/a/mat-icon	
AssertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th[5]/div/div[2]/div	true

Figur 101: SorterPåProfil test i katalon

7.0 SorterPålngen

Command	Target	Value
open	http://localhost:30200/?smallTabGroup=queueListTab&bigTabGroup=agentsTab	
click	//button	
click	//wave-accordion[2]/div/span[2]	
click	//div[2]/a/span[2]	
click	//wave-toolbar/div/button	
click	//div[@id='mat-menu-panel-0']/div/button/mat-icon	
click	//wave-row[2]/div[2]/div	
type	//input[@type='text']	SorterIngen
click	//div/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
click	//tr[13]/td[4]/span/a/mat-icon	
assertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th/div/div[2]/div	true

Figur 102: SorterPålngen test i katalon

M - filtrereAgentgrupper: Som administrator ønsker jeg å filtrere på agentgruppe

1.0 Agent som tilhører agentgruppen Default er logget på en kø - sjekker at agenten eksisterer i wallboardet

Command	Target	Value
open	http://localhost:4200/?wallboardKey=1284d5dc-d431-4c11-aee1-0eba546b46c2	
assertText	//main[@id='agentTable']/div/table/tbody/tr/td[5]	Default

Figur 103: filtrerAgentgrupper på agentgruppe test i katalon

2.0 Agent som tilhører agentgruppen Default er logget av kø - sjekker at agenten ikke eksisterer i wallboardet lenger

Command	Target	Value
open	http://localhost:4200/?wallboardKey=1284d5dc-d431-4c11-aee1-0eba546b46c2	
assertElementNotPresent	//main[@id='agentTable']/div/table/tbody/tr/td[5]	Default

Figur 104: sjekk på ikke eksisterende agentnavn test i katalon

N - EndreWallboard: Som administrator ønsker jeg å endre wallboard innstillinger

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[15]/td[6]/span/button	
click	//input[@type='text']	
type	//input[@type='text']	EndreWallboardNY
click	//mat-radio-button[@id='mat-radio-4']/label/div/div[2]	
click	//div/div/wave-checkbox/wave-mdi	
click	//wave-toolbar/div/button	
AssertText	//tr[15]/td/span	EndreWallboardNY
click	//tr[15]/td[4]/span/a/mat-icon	

AssertElementPresent	//main[@id='agentTable']/div/table/thead/tr/th[2]/div/div[2]/div[2]/div[2]	true
----------------------	--	------

Figur 105: EndreWallboard test i katalon

O - ÅpneFlereWallboardURL

Manuelt testet, se [Vedlegg: Systemtest-dokumentasjon \(ekstern lenke\)](#).

P - SeTidICallStatus

Ble ikke testet grunnet feil på Zisson Interact, som førte til at denne ikke kunne testes.

Q - SeDato: Som bruker ønsker jeg å se dato.

Command	Target	Value
open	http://localhost:4200/?wallboardKey=ecc1d28b-d31f-49bc-8299-303194b988b1	
AssertText	id=date	05.05.2021

Figur 106: SeDato test i katalon

R - SeKlokkeslett

Manuelt testet, se [Vedlegg: Systemtest-dokumentasjon \(ekstern lenke\)](#).

S - feilmeldingVedDisconnected: Som bruker ønsker jeg å få tilbakemelding om Websocket er nede

Command	Target	Value
open	http://localhost:30200/administration/wallboards/main	
click	//tr[6]/td[4]/span/a/mat-icon	
setTimeout	http://localhost:4200/?wallboardKey=c1c30ff0-e21d-4bf0-bb05-19aaa72791ab	1000
AssertElementPresent	//app-error-wallboard/div/p	//app-error-wallboard/div/p

Figur 107: feilmeldingVedDisconnected test i katalon

4.2 Akseptansetester av inkrementer

Sprint 1

Hele designprosessen ble fulgt under denne sprinten for å lage prototyper.

Herunder ble det gjort brukertesting.

Tilbakemeldinger etter sprint review:

- Prototypene ble gjennomgått under sprint review og oppdragsgiver var fornøyd med det som ble presentert.

Sprint 2

Inkrementer implementert:

- Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.
- Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.

Tilbakemeldinger etter sprint review møte:

- Grunnet manglende tilgang til nødvendige verktøy og prosjekter, blir begge inkrementene tatt med til neste sprint, da enkelte ting må settes opp på en midlertidig måte mens studentene arbeider. Dette gjør at inkrementene ikke kan godkjennes enda.

Sprint 3

Inkrementer implementert:

- Som agentleder ønsker jeg at agentene lagres til cache, slik at jeg ved tilkobling får inn de siste dataene som har vært registrert.

- Som agentleder ønsker jeg et wallboard som viser frem agentene i en tabell for å ha en oversikt over agentene som er på jobb.
- Som agentleder ønsker jeg at wallboardet gjør et nytt forsøk på tilkobling til websocket dersom tilkobling mislykkes.
- Som agentleder ønsker jeg å se agentnavn for å skille mellom agentene.
- Som agentleder ønsker jeg å se profil for å vite hvilken profil agentene tilhører.
- Som agentleder ønsker jeg å se agentgruppe for å vite hvilken agentgruppe agentene tilhører.

Følgende funksjonalitet videreføres fra forrige sprint:

- Som agentleder ønsker jeg å se sanntidsdata for å få et overblikk over situasjonen i nåtid.
- Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.

Tilbakemeldinger etter sprint review møte:

- De fleste inkrementene som er utført under sprintet godkjennes. Cache må jobbes videre med da det kom ny informasjon fra oppdragsgiver på hvordan de ønsker at dette skal fungere.

Sprint 4

Inkrementer implementert:

- Som agentleder ønsker jeg å se tid i agentstatus for å vite hvor lenge agenten har vært i en status.
- Som agentleder ønsker jeg å se callstatus for å vite hvor lenge hver agent er i hver telefonsamtale og se hvem agenten er i samtale med.

- Som administrator ønsker jeg å kunne filtrere på agentgruppe for å vise ønskede agenter på wallboardet.
- Som agentleder ønsker jeg å se en overordnet oversikt over statusene til agentene i diagramform, slik at jeg har bedre oversikt ved mange agenter.
- Som agentleder ønsker jeg at cache håndterer ulike wallboards ut i fra en wallboard nøkkel, slik at jeg kun får dataene som tilhører et gitt wallboard.
- Som administrator ønsker jeg å ha en knapp som sier noe om jeg skal opprette et wallboard med den nye eller den gamle konfigurasjonen.
- Som agentleder ønsker jeg at wallboardet fremstår responsivt på store skjermer, slik at wallboardet ser bra ut når vises frem på storskjerm.

Følgende funksjonalitet blir videreført fra forrige sprint:

- Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.

Tilbakemeldinger etter sprint review møte:

- Samtlige inkrementer blir godkjent. Arbeidet med brukerhistorien knyttet opp mot å sette opp wallboard i ReWave blir videreført, da det i forrige sprint ble gjort en del endringer på prosjektet som skal jobbes med for å få implementert inkrementet.

Sprint 5

Inkrementer implementert:

- Som administrator ønsker jeg å legge til en logo til wallboardet fordi det er en del av selskapets profil.
- Som bruker ønsker jeg å sortere på de ulike kolonnene for å kunne tilpasse sorteringen ut i fra hva som er relevant.

- Som agentleder ønsker jeg å se callstatus for å vite hvilket telefonnummer en agent er i samtale med slik at jeg har oversikt.
- Som administrator ønsker jeg å kunne endre konfigurasjonen til et wallboard slik at jeg ikke må opprette et nytt dersom endringer er nødvendig.

Følgende funksjonalitet videreføres fra forrige sprint:

- Som administrator ønsker jeg å sette opp wallboard med mulighet til å velge ut ulik funksjonalitet i fra hva man ønsker å vise frem.
- Som agentleder ønsker jeg å se en overordnet oversikt over statusene til agentene i diagramform, slik at jeg har bedre oversikt ved mange agenter.

Tilbakemeldinger etter sprint review møte:

- Inkrementene godkjennes. Ny informasjon om filtrering kommer frem, og det må gjøres endringer på inkrementet knyttet til dette.

Sprint 6

Inkrementer implementert:

- Som administrator ønsker jeg å se ikoner for det nye og det gamle wallboardet i oversikten over wallboards for å kunne skille mellom de ulike typene.
- Som administrator ønsker jeg å distribuere wallboardet via en unik URL-lenke til den som skal overvåke den, slik at de jeg sender den til får tilgang til det spesialtilpasset wallboard jeg opprettet.
- Som administrator ønsker jeg å kunne klone det nye wallboardet for å raskt kunne opprette et nytt wallboard av samme type.
- Som agentleder ønsker jeg å kunne oppdatere cache med nye agenter hvis det har kommet en endring på en wallboardkonfigurasjon som blir hentet fra databasen.

- Som administrator ønsker jeg å kunne endre på et wallboard og da få inn de verdiene som tidligere var skrevet inn.
- Som agentleder ønsker jeg å se agentstatus for å vite hva agentene foretar seg.
- Som administrator ønsker jeg under oppretting/oppdatering av wallboard å kun se felt som er relevante dersom det er valgt diagramform.

Følgende funksjonalitet videreføres fra tidligere sprint:

- Som administrator ønsker jeg å kunne filtrere på agentgruppe for å vise ønskede agenter på wallboardet.
- Som agentleder ønsker jeg at agentene lagres til cache, slik at jeg ved tilkobling får inn de siste dataene som har vært registrert.

Tilbakemeldinger etter sprint review møte: Inkrementene blir godkjent.

4.3 Spørsmål fra brukertester

Ved brukertestene ble det gjort kvalitative undersøkelser, der svarene foreligger i form av tekst. Hensikten med kvalitative studier er å oppnå dybdekunnskap, samt få en helhetlig forståelse av spesifikke kontekster (Grønmo, 2020). Det ble satt opp intervju, samt samtaler og workshops, hvor gruppen kunne gjøre endringer underveis for rask tilbakemelding og felles forståelse.

4.3.1 Intervju/ samtale om første prototype

Etter samtale og intervju med Kim Josefson (product manager hos Zisson) ble det vist frem de første utkastene til wallboard og spurt om følgende:

1. Hvordan fungerer ikonene?
2. Hvilke av utkastene av wallboardene fungerer best?
3. Hva tenker du om fargevalg og kontrast?
4. Andre kommentarer/ tilbakemeldinger?

4.3.2 Første brukertest - MVP

Spørsmål:

Design

- Hva synes du om ikonene? Hva betyr de?
- Er det noe som er uklart?
- Er det noe som burde være mer fremhevet?
- Er det noe du synes burde vært annerledes?
- Hva synes du om tekststørrelsen?
- Hvordan oppleves tabellen?
- Hva synes du om størrelsen på klokkeslettet?
- Hvordan oppleves wallboardet på avstand?

Funksjonalitet

- Hvordan ville du gått frem for å sortere?
- Hvordan ville du gått frem for å filtrere?
- Hva synes du om plasseringen på innstillinger-knappen?

4.4 Andre runde brukertest - MVP

Spørsmål

Bilde 1: Oppsett av agentgrupper før filtrering

- Hva tenker du om at dette er det første bildet man kommer til når man skal sette opp et wallboard?
- Hvordan ville du gått frem for å filtrere agentgrupper?
- Hva tenker du om plasseringen på innstillinger-knappen? (Bør den ligge inne i agentvinduet og tilhøre kun innstillinger knyttet til agenter, eller bør det være en generell innstillinger knapp?)

Bilde 2: Valg av innstillinger/filtreringsmeny:

- Hvilken av innstillinger/filtreringsmeny alternativ liker du best?
 - Liker du best runde/firkantede sjekkbokser?
 - Liker du best innrykk/ikke innrykk?

Bilde 3: Oppsett filtrering med innstillinger uten agenter

- Hva synes du om krysset øverst til høyre for å krysse deg ut av innstillingene?
- Er det noen annen måte du skulle ønske du kunne krysse deg ut av innstillinger på?(Kanskje også: Ville det vært ønskelig å kunne trykke hvor som helst på skjermen i tillegg, for å krysse seg ut av innstillinger?)

Bilde 3.1 Agentoversikt med filtrering

Her ser du oversikt over wallboard med agenter og innstillinger.

Bilde 4: Mange agenter - uten innstillinger

- Hvordan oppleves tabellen?
- Hva synes du om kategoriene i tabellen?
- Hva synes du om fargekategoriene? Kan du beskrive hva de forskjellige fargekategoriene betyr?
- Hva synes du om ikonene? Hva tenker du at de betyr?
- Hva synes du om tekststørrelsen og fonten?
- Hva tenker du skjer hvis du trykker på pil-ikonet?
- Hva synes du om størrelsen og plasseringen på klokkeslettet og dato?
- Hvordan oppleves tabellen på avstand?

Bilde 5: Wallboard med få agenter

- Hva tenker du om hvordan vi har løst det med tanke på antall agenter og størrelsen på hver rad i tabellen?

Andre tilbakemeldinger:

- Er det noe som er uklart?
- Er det noe som burde være mer fremhevet?
- Er det noe du synes burde vært annerledes?

4.4 SUS spørreundersøkelse

4.4.1 Skjermbilder av SUS spørreundersøkelse (Survey monkey)

SP1

I think that I would like to use this system frequently.

strongly agree**SP2**

I found the system unnecessarily complex.

strongly disagree**SP3**

I thought the system was easy to use.

strongly agree**SP4**

I think that I would need the support of a technical person to be able to use this system.

strongly disagree**SP5**

I found the various functions in this system were well integrated.

agree**SP6**

I thought there was too much inconsistency in this system.

strongly disagree**SP7**

I would imagine that most people would learn to use this system very quickly.

strongly agree**SP8**

I found the system very cumbersome to use.

strongly disagree**SP9**

I felt very confident using the system.

strongly agree**SP10**

I needed to learn a lot of things before I could get going with this system.

strongly disagree

Figur 108: Skjermbilde av SUS spørreundersøkelse, testperson 1

SP1

I think that I would like to use this system frequently.

agree

SP2

I found the system unnecessarily complex.

disagree

SP3

I thought the system was easy to use.

strongly agree

SP4

I think that I would need the support of a technical person to be able to use this system.

strongly disagree

SP5

I found the various functions in this system were well integrated.

strongly disagree

SP6

I thought there was too much inconsistency in this system.

strongly disagree

SP7

I would imagine that most people would learn to use this system very quickly.

strongly agree

SP8

I found the system very cumbersome to use.

strongly disagree

SP9

I felt very confident using the system.

strongly agree

SP10

I needed to learn a lot of things before I could get going with this system.

strongly disagree

Figur 109: Skjermbilde av SUS spørreundersøkelse, testperson 2

SP1

I think that I would like to use this system frequently.

agree

SP2

I found the system unnecessarily complex.

disagree

SP3

I thought the system was easy to use.

Agree

SP4

I think that I would need the support of a technical person to be able to use this system.

agree

SP5

I found the various functions in this system were well integrated.

strongly agree

SP6

I thought there was too much inconsistency in this system.

strongly disagree

SP7

I would imagine that most people would learn to use this system very quickly.

agree

SP8

I found the system very cumbersome to use.

disagree

SP9

I felt very confident using the system.

neutral

SP10

I needed to learn a lot of things before I could get going with this system.

neutral

Figur 110: Skjermbilde av SUS spørreundersøkelse, testperson 3

SP1

I think that I would like to use this system frequently.

agree

SP2

I found the system unnecessarily complex.

neutral

SP3

I thought the system was easy to use.

Agree

SP4

I think that I would need the support of a technical person to be able to use this system.

strongly disagree

SP5

I found the various functions in this system were well integrated.

strongly agree

SP6

I thought there was too much inconsistency in this system.

disagree

SP7

I would imagine that most people would learn to use this system very quickly.

strongly agree

SP8

I found the system very cumbersome to use.

disagree

SP9

I felt very confident using the system.

agree

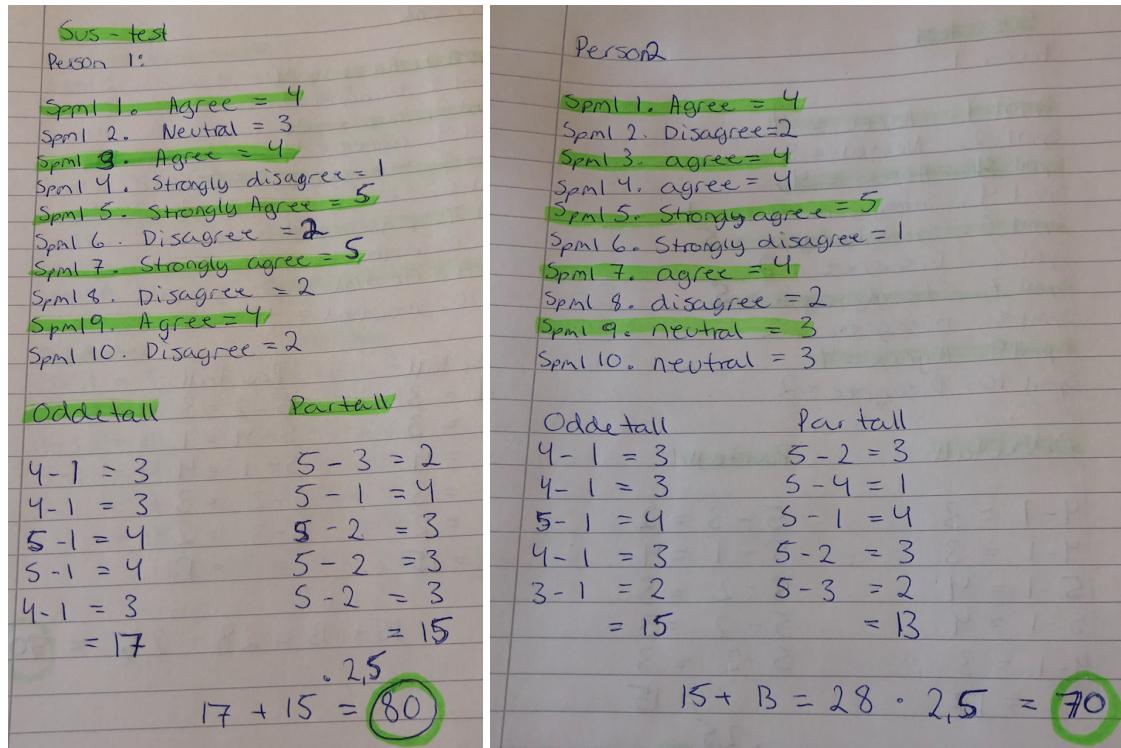
SP10

I needed to learn a lot of things before I could get going with this system.

disagree

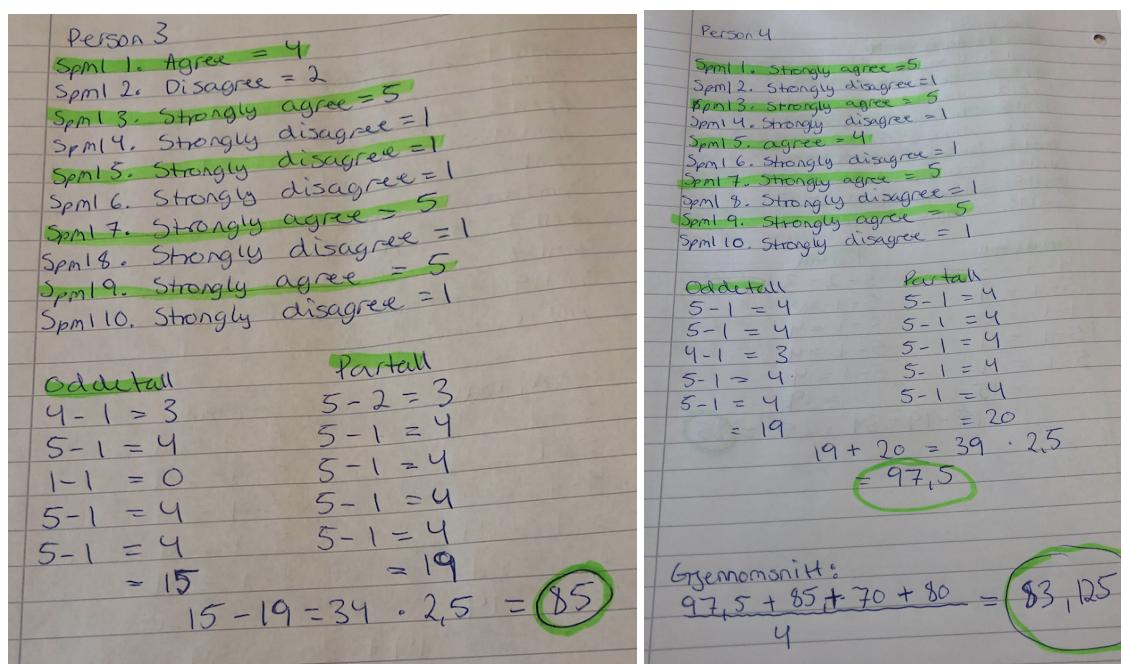
Figur 111: Skjermbilde av SUS spørreundersøkelse, testperson 4

4.4.2 Utregning SUS spørreundersøkelse



Figur 112:(Venstre) Utregning av SUS spørreundersøkelse, testperson 1

Figur 113:(Høyre) Utregning av SUS spørreundersøkelse, testperson 2



Figur 114: (Venstre) Utregning av SUS spørreundersøkelse, testperson 3

Figur 115: (Høyre) Utregning av SUS spørreundersøkelse, testperson 4

5.0 Utkast prototyper

I GitHub-repositoriet Vedlegg ligger alle prototypene fra hele designprosessen, og inkluderer alle forslag; alt fra papirskisser til sluttprototyper. Vedlegget er delt inn etter alle prototyperundene, med et eget punkt for mock ups.

Her finnes en ekstern lenke til GitHub-repository med [Alle utkast av skisser og prototype.](#)

6.0 Kildekode

Som et eksternt vedlegg ligger kildekode som prosjektgruppen har utviklet. Det er utelatt annen intern basekode etter ønske fra Zisson. Vedlegget er en zip-fil som inneholder en PDF-fil av skjermbilder av koden i ReWave- og WallboardService-prosjektene, samt angularprosjektet AgentRealTimeWallboard.

I. REFERANSER FOR VEDLEGG

Carnahan, J. (2016, 14.mars) *When to use a HTTP call instead of a WebSocket.*

Windows blogs.

<https://blogs.windows.com/windowsdeveloper/2016/03/14/when-to-use-a-http-call-instead-of-a-websocket-or-http-2-0/>

Grønmo, S. (2020, 3. november). *Kvalitativ metode.* Hentet 15.mai 2021

https://snl.no/kvalitativ_metode

Khorikov, V. (2016, 4. mai). Singleton vs Dependency Injection. Enterprise Craftsmanship. Hentet 2021, 10. Mai.

<https://enterprisecraftsmanship.com/posts/singleton-vs-dependency-injection/>

Microsoft. (u.å.). *Dictionary<TKey, TValue> Class.* Hentet 2021, 10. Mai.

<https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2?view=net-5.0>