



Forside

IS-304: 2023

Tittel: Digitalisering av intern trappekurransse

Emnekode	IS-304
Emnenavn	Bacheloroppgave i informasjonssystemer
Emneansvarlig	Hallgeir Nilsen
Veileder	Sofie Wass
Oppdragsgiver	Capgemini Kristiansand

Studenter:

Etternavn	Fornavn
Berge Stamnes	Eirik
Larsen	Sebastian
Lauvhjell	Andreas
Linnerud	Anna

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeid uten at dette er oppgitt, og at alle referanser er oppgitt i litteraturlisten.	JA	
Kan besvarelsen brukes til undervisningsformål?	JA	
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA	

Forord

Vi vil gjerne få lov til å takke alle som har bidratt og hjulpet oss med å fullføre både bachelorprosjekt og bacheloroppgave. En spesiell takk til vår flotte veileder fra Capgemini, Vegard Sætre, som har hjulpet oss å strukturere og prioritere arbeidet vårt, samt kommet med gode tilbakemeldinger og forslag (og dratt oss med på sosiale arrangementer med resten av Capgemini gjengen). Takk til Andreas og Selim som har vært vår faglige støtte fra bedriften, og som har passet på at vi har hatt alt vi trenger og ikke minst at vi har hatt det bra. Og takk til resten av de ansatte hos Capgemini som har tatt oss imot, vært med på brukertester, holdt kurs for oss og generelt vært veldig engasjert i arbeidet vårt. Vi vil også få takke Rakel i Quadrum kafé som kokte en stor kaffe til oss hver gang vi var hos Capgemini.

Til slutt vil vi også gjerne takke Sofie Wass og Hallgeir Nilsen for all akademisk støtte og oppmuntring gjennom hele arbeidet. Studiet har vært utfordrende og har hatt sine oppturer og nedturer, men vi sitter alle igjen med god kunnskap og erfaringer som vi vil dra nytte av i arbeidslivet.

Sammendrag

I emnet IS-304 Bacheloroppgave i IT og Informasjonssystemer, har vi i Gruppe 9 fått lov til å utføre bachelorprosjektet sammen med Capgemini. Capgemini er et internasjonalt og etablert IT-selskap, samt en viktig global aktør innenfor en rekke områder. De har hovedkontor i Frankrike, men har flere konsulenthus rundt omkring i verden. Grunnen til at gruppen ønsket å jobbe med Capgemini var at oppgaven virket omfattende og spennende. Samt var det stort rom for teamet til å være kreative og ta selvstendige valg. Prosjektet gikk ut på å digitalisere en konkurranse som omhandlet hvilken ansatt som besteg flest trapper i løpet av en måned.

Rapporten tar for seg hvordan teamet gikk frem, viktige metodikker anvendt, og valgte verktøy. Samt hvordan gruppen har benyttet scrum for å opprettholde iterativ fremgang. Teamet var også kontinuerlig i samtaler med produkteier for å opprettholde ønsket kvalitet. Til slutt kommer resultatet av prosjektet, som teamet i stor grad er fornøyd med.

Det ble gjennomført flere intervjuer og brukertester for å forstå hva som var ønsket. Deretter anvendte teamet sin egen kunnskap og erfaring for å forme et komplett system. Det vil alltid være rom for forbedring av applikasjonen, men sluttproduktet svarte på alle kriteriene stilt av oppdragsgiver.

Innholdsfortegnelse

1. Introduksjon	6
1.1. Capgemini	6
1.2. Oppgavebeskrivelse	6
1.3. Problemdomene	7
1.4. Det nye produktet.....	7
2. Sentrale valg og avgjørelser.....	8
2.1. Forventningsavklaring	8
2.1.1. Forventninger rundt teknologibruk	8
2.1.2. Kvalitet.....	9
2.1.3. Kontrakter	9
2.2. Prosjektstyring	9
2.2.1. Scrum	10
2.2.2. Rollefordeling	10
2.2.3. Prosjektstyringsverktøy.....	10
2.2.4. Kommunikasjonsverktøy	11
2.2.5. Estimering	11
2.2.6. Risikoanalyse	12
2.2.7. Sprint-plan/Forventet fremgang.....	13
2.3. Teknologi	14
2.3.1. Frontend og backend.....	14
2.3.2. Database.....	14
2.3.3. Utviklingsmiljø	15
3. Kvalitet.....	15
3.1. Kommunikasjon.....	15
3.2. Kodestandard	16
3.3. Versjonskontroll.....	16
3.4. Team-oppsett.....	16
3.5. Brukertestning	17
4. Prosjektgjennomføring.....	17
4.1. Gjennomføring av scrum	18
4.1.1. Produkt Backlogg.....	18
4.1.2. Sprint Backlogg	18

4.1.3. Sprint Planning.....	19
4.1.4. Daily Scrum	19
4.1.5. Sprint Review og Retrospekt	20
4.2. Analyse og applikasjonsdesign	22
4.2.1. Forstå Problemdomenet	22
4.2.2. Intervju	22
4.2.3. Persona.....	23
4.2.4. Brukerhistorier	23
4.2.5. Systemkrav.....	26
4.2.6. Skisser	26
4.2.7. Navigasjonskart.....	27
4.2.8. Designprinsipper	28
4.2.9. Prototype	29
4.2.10. Databasedesign	29
4.3. Implementering	31
4.3.1. MudBlazor	31
4.3.2. Dapper.....	32
4.3.3. Systemarkitektur	33
4.3.4. Filstruktur.....	34
4.3.5. Testing.....	35
4.4. Gjennomføring del 1: Analyse og design (Pre-Sprint - Sprint 2).....	36
4.4.1. Pre-sprint.....	36
4.4.2. Sprint 1	36
4.4.3. Sprint 2.....	37
4.4.4. Retrospekt Pre-sprint til Sprint 2	37
4.5. Gjennomføring del 2: Utvikling og testing (Sprint 3 - Sprint 5)	37
4.5.1. Sprint 3.....	38
4.5.2. Sprint 4.....	38
4.5.3. Sprint 5.....	39
4.5.4. Retrospekt Sprint 3 til Sprint 5	39
4.6. Gjennomføring del 3: Ferdigstilling (Sprint 6 til Sprint 7).....	39
4.6.1. Sprint 6.....	39
4.6.2. Sprint 7.....	40
4.6.3. Retrospekt Sprint 6 til Sprint 7	40
4.7. Brukertestning.....	40

4.7.1. Brukertest 1	41
4.7.2. utfordringer under utførelse av brukertest	41
4.7.3. Brukertest 2	42
4.7.4. Endringer som resulterte fra brukertesting	43
5. Refleksjon	47
5.1. Utelatte Funksjoner	48
5.2. Kvalitet og Kvalitetssikring	48
5.3. Rollefordeling	49
5.4. Prosjektgjennomføring.....	49
5.4.1. Risikohåndtering	49
5.4.2. Estimering med Story Points	50
5.4.3. Testing.....	51
5.4.4. Konklusjon og læringsutbytte	51
6. Uttalelse fra oppdragsgiver	52
7. Egenvurdering	54
8. Litteraturliste.....	56
9. Appendiks	62
9.1. Appendiks 1 – Risikoanalyse.....	62
9.2. Appendiks 2 – Skisser.....	62
9.3. Appendiks 3 – Oppsummering av brukerundersøkelse i Sprint 1	63
9.4. Appendiks 4 – Brukerundersøkelse mal, prototype v1.....	66
9.5. Appendiks 5 – MoSCoW	68
9.6. Appendiks 6 – Risikologg.....	70
9.7. Appendiks 7 – Interessentanalyse og Kommunikasjonskart	71
9.8. Appendiks 8 - Demo av applikasjon.....	71

Figurliste

Figur 1: Utkast av gruppens sprint-board	11
Figur 2: Utklipp av scrum-poker.....	12
Figur 3: 5x5 Risk Matrix Sample (SafetyCulture, 2023a).....	13
Figur 4: Sprint-plan med forventet fremgang	13
Figur 5: Brukerhistorie eksempel.....	18
Figur 6: Sprint 3 backlogg	19
Figur 7: Burndown Chart av sprint-periode.....	21
Figur 8: Cumulative flow-diagram	21
Figur 9: Persona satt opp etter brukerintervjuene	23
Figur 10: Tabell med to brukerhistorie maler	24
Figur 11: Tabell fylt med brukerhistoriene benyttet	25
Figur 12: Tabell med Must-Have brukerhistorier	26
Figur 13: Navigasjonskart versjon 1	27
Figur 14: Navigasjonskart versjon 2	28
Figur 15: ER-Diagram Versjon 1.....	30
Figur 16: Versjon 2, ferdig ER-diagram	31
Figur 17: SQL-spørring for å hente brukerresultat	32
Figur 18: Systemarkitektur for Captrapp	33
Figur 19: Overordnet mappestruktur i prosjektet.....	35
Figur 20: Medaljer der bare en var klikkbar i prototypen.....	42
Figur 21: Progresjon av leaderboard.....	43
Figur 22: Progresjonen av frontsidene.....	44
Figur 23: Streak popup.....	45
Figur 24: Progresjon av 'Min Profil'	46
Figur 25: Progresjon for fellesmål	47

1. Introduksjon

I løpet av vårsemesteret 2023 har gruppen gjennomført bacheloroppgaven sammen med Capgemini. Målet til prosjektet var å digitalisere en konkurranse slik at ansatte i bedriften kunne registrere antall etasjer de besteg i løpet av en arbeidsdag. Teamet benyttet også kunnskap, og erfaring fra studiet på en mer praktisk måte. Dette prosjektet dro i tillegg inn ny kunnskap ettersom det oppstod behov.

1.1. Capgemini

Capgemini er et multinasjonalt IT konsultentselskap som opererer i over 50 land, og er en av de større IT konsulent selskapene i verden. Capgemini beskriver seg selv som ‘Den strategiske partneren som hjelper å stille de vanskelige spørsmålene og finne de rette svarene, ved å utnytte teknologi’ (Capgemini, u.å.). I Norge har de kontorer rundt om i landet og er cirka 1400 ansatte. De kom til Kristiansand i fjor, og her bistår de virksomheter i den digitale transformasjonen (Næringsforeningen, 2022).

1.2. Oppgavebeskrivelse

Produktet Capgemini ønsket at gruppen skulle utvikle var en digital løsning for deres interne trappek konkurranse. Den skulle minimum ha en oversikt over antall etasjer de forskjellige ansatte i bedriften har gått opp på vei til kontoret.

Oppgaven slik den ble presentert av Capgemini var åpen i forhold til hvordan den skulle løses. Utenom dette var det bare noen forslag for å få gruppen til å være kreative i hvordan medlemmene skulle løse dette, og bygge videre på konseptet. Sammen med produkteier kom teamet fram til denne oppgavebeskrivelsen for å beskrive det viktigste som gruppen skulle levere.

‘Lage en webapplikasjon for Capgemini, slik at ansatte daglig kan registrere antall etasjer de har besteget. Selve registreringen av trapper skal være enkel og rask for brukerne.

Applikasjonen skal gi en oversikt over hvor mye hver enkelt ansatt og alle ansatte samlet har gått i løpet av en periode. Dette kan vises i form av en rangstige(r). Dataen tilgjengelig kan Capgemini for eksempel bruke til premieutdelinger eller felles aktivitet for veldedighet.’

1.3. Problemdomene

Trappekonkurransen i Capgemini gikk ut på å gå flest trapper i løpet av en måned. For å holde oversikten over poengene til hver ansatt hadde de en tavle i gangen hvor de måtte tegne et rutenett med måned og datoer. Poengene ble oppført ved at de ansatte skrev initialene sine inne i den ruten som tilsvarte dagens dato med antall trapper gått den dagen. På slutten av måneden måtte noen regne ut hvor mange trapper alle hadde gått, og kåre en vinner. Det måtte også bli tegnet et nytt rutenett for den nye måneden. De hadde ingen steder de skrev ned og holdt oversikt over tidligere vinnere eller poeng, så denne informasjonen forsvant ved månedsskifte.

Dette systemet var problematisk fordi det krevde at en ansatt tok ansvaret for å tegne en ny måned på tavlen og regne ut alle poengsummene hver måned. Det var ikke alltid de hadde tid eller husket å gjøre dette i tide, og konkurransen startet dermed ofte senere ut i måneden. Det kunne være leit for de som hadde gått mange trapper de dagene hvor konkurransen ikke var startet ennå. Det kunne også hende at ansatte hadde glemt hvor mange trapper de gikk disse dagene, og fikk ikke skrevet opp riktig antall som resultat. Hovedproblemen var at det tok for lang tid i forhold til hva det gav, og oppsettet var rotete og overkomplisert. De ønsket i tillegg å gjøre konkurransen mer attraktiv og moro for deltakerne.

1.4. Det nye produktet

For å løse de eksisterende problemene i trappekonkurransen skal gruppen digitalisere den i form av en webapplikasjon. Her kommer gruppen til å fjerne behovet for at noen må telle poengsummer, tegne på tavlen hver måned, og all statistikk vil bli lagret permanent. De ansatte skal kunne registrere trapper fra kontorplassen eller hvor som helst, så lenge de har tilgang til internett. De vil ikke trenge å gå til gangen for å registrere, som også kan spare tid. Konkurransen blir også mer oversiktlig, ettersom hver deltaker kan se sin egen og andres statistikk i forskjellige formater. Til slutt vil gruppen dekke behovet for å gjøre konkurransen mer gøy ved å legge til gamification i applikasjonen.

2. Sentrale valg og avgjørelser

Dette kapittelet består av sentrale valg som har effekt på systemets helhet. Da med majoriteten i forhold til valg som ble gjort tidlig, slik som forventningsavklaring, interne forventninger, og anvendt teknologi. Disse er sentrale for prosjektet, og medfølgende beslutninger tatt her hadde i stor grad innflytelse på prosjektets fremgang, og utfall.

2.1. Forventningsavklaring

Under pre-sprinten hadde gruppen blant annet møte med prosjekteier og satte forventningsavklaringer relatert til prosjektet. I disse møtene ble det avklart hvilke forventninger Capgemini hadde til bachelor-gruppen og hva gruppen kunne forvente av dem. Blant kravene satt til gruppen var at gruppen skulle utføre 'Must haves' i MoSCoW (som blir utdypet senere i 4.2.5), gi beskjed om gruppen ikke kunne møte, svare i rimelig tid, være ærlige og gi beskjed hvis gruppen slet, og holde tidsplaner oppdatert.

Kravene som gruppen stilte til Capgemini, var å få svar på spørsmål og meldinger innen rimelig tid. De skulle bistå med kunnskap som gruppen hadde behov for, og var oppdatert på hva som lå i Teams-kanalen. Produkteier skulle også delta i de relevante møtene, som sprint-review og statusmøter med gruppens UiA-Veileder. I tillegg var det forventet at gruppen skulle få god og konstruktiv kritikk, som forhåpentligvis bidro til at produktet ville leve opp til forventningene deres.

2.1.1. Forventninger rundt teknologibruk

En sentral beslutning var å velge hva slags teknologi som skulle bli brukt. Capgemini uttrykte tidlig at de ikke hadde noen preferanser for programmeringsspråk, og at gruppen stod fritt til å velge det de følte seg mest komfortable med. Gruppen hadde opprinnelig tenkt til å bruke React Native og C#, men endte til slutt opp med å anvende Blazor-rammeverket som bruker C# i både front- og backend. Valget falt på Blazor, siden det hørtes spennende ut å lære C# og ettersom det virket praktisk å ha det samme språket i både frontend og backend.

2.1.2. Kvalitet

Gruppen begynte tidlig å ha møter med produkteier, hvor det ble avklart hva som ble forventet ut av gruppen i form av arbeid, innsats og sluttprodukt. Her kom det frem at produkteier ønsket seg et produkt som var brukende, opprettholdbar og med mulighet for å utvides på et senere tidspunkt. Det ble fastsatt at det var bedre å levere et produkt med høy kvalitet, men med begrenset omfang, enn et produkt av lav kvalitet med stort omfang. Gruppen startet med høye mål, men etter gjentakende møter med produkteier, ble gruppen over tid overtalt til å kutte ned omfang for å forsikre at det som ble levert holdt høy kvalitet.

2.1.3. Kontrakter

Etter tidligere erfaringer var gruppen raske på å bli enige om en felles gruppekontrakt, hvor alle deltakere ble enige om rammene for hvordan en skal forholde seg til eventuelle problemer som måtte oppstå. I kontrakten stod det blant annet konkrete punkter for hvordan en skal gå frem hvis et gruppemedlem ikke følger reglene, som for eksempel hvis noen skulle vært sene eller ikke levert avtalt arbeid gjentatte ganger. Hensikten med denne var ikke å drive skremsel internt i gruppen, men heller å ha noe å referere til hvis et gruppemedlem skulle falle helt ut slik at dette problemet ikke faller på resten.

I samkjør med dette ble også en kontrakt undertegnet i samarbeid med Capgemini, og ansvarlige fra UiA hvor de relevante aktørene ble enige om å sikre gjensidig utbytte. Her kom det frem hvem som var ansvarlig kontaktperson i bedriften, og hvem som skulle være universitetets representant. Dette ble så undertegnet skriftlig av alle parter.

2.2. Prosjektstyring

For gjennomføringen av prosjektet bestemte gruppen seg for å bruke en agil prosjektstyringsmetode. Agil metodikk i prosjektstyring er en framgangsmåte som involverer stadig samarbeid med interessenter og arbeid i iterasjoner. Nøkkelen ligger i tankegangen om kontinuerlig forbedring gjennom prosjektets livssyklus, hvor endringer kan bli gjort raskt og responsivt (Wrike, u.å.a).

2.2.1. Scrum

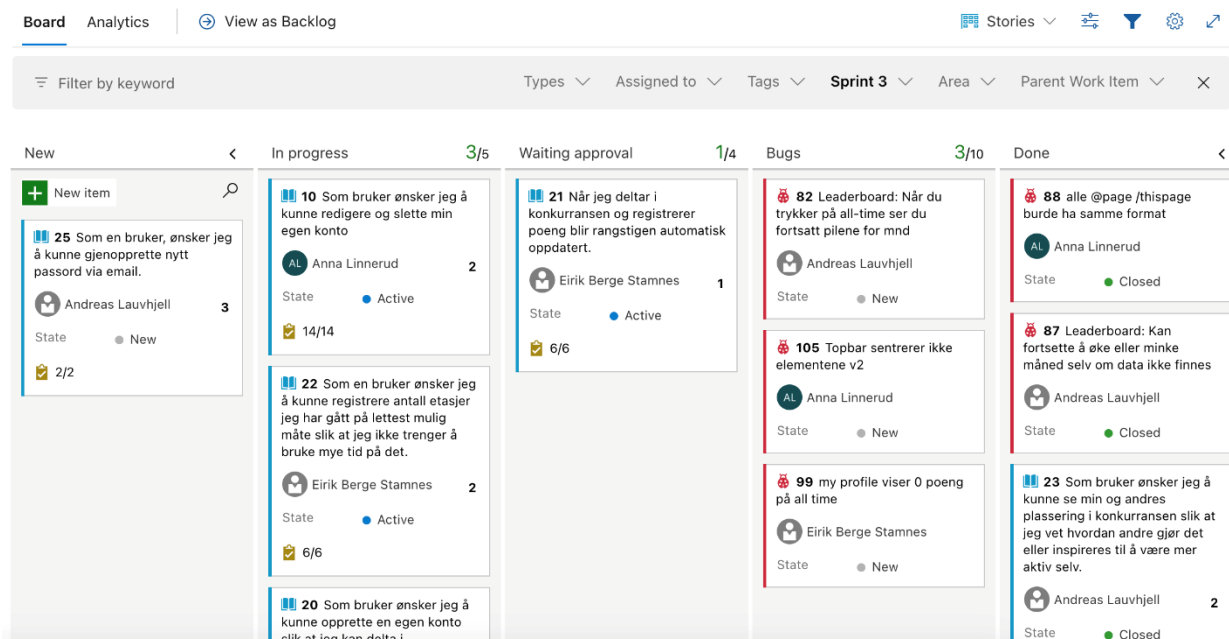
Scrum er et agilt rammeverk (Scrum, u.å.e) som gruppen benyttet i prosjektet. Dette involverer å jobbe i sprints, ha daglige framdriftsmøter og revidere og reflektere over valgene som har blitt gjort etter hver iterasjon (Schwaber & Sutherland, 2020). Scrum ble valgt ettersom det er et responsivt rammeverk som gir mulighet for forandringer underveis. Prosjektets tidsramme var relativt kort, og det var da ønskelig å kunne endre prioritet på funksjoner og oppgaver etter ønske fra interessenter. På den måten ville de mest attraktive delene av applikasjonen komme på plass først, noe som sikret kvalitet og god brukeropplevelse. Scrum legger også opp til god kommunikasjon både innad i gruppen og med andre involverte, som kan hindre misforståelser og bortkastet arbeid (Wallen, 2021). Ytterligere var det også dette rammeverket de brukte mest på Capgemini, og ved å velge scrum fikk gruppen både kursing og assistanse rundt problemstillinger i prosjektstyringen.

2.2.2. Rollefordeling

Gruppen har jobbet sammen tidligere på flere andre prosjekter. Derfor var erfaringen at en mer flat-hierarkisk struktur fungerte best, men teamet hadde også faste roller som kom mer frem da det var hektisk. I starten ble en gruppeleder og en vara stemt frem. På disse falt ansvaret for å passe på at oppgaver slik som møtereferater og lignende administrativt arbeid ble gjort. Da det kom til scrum master-rollen gikk denne på rundgang, som vil si at hvert gruppemedlem hadde rollen i minst én sprint. Hensikten med dette var at alle gruppemedlemmene skulle få muligheten til å sitte med samme ansvar for å bedre forstå rollen en scrum master innebar. Dette bidro også til at én person slapp å sitte med alt det administrative arbeidet gjennom hele prosjektet.

2.2.3. Prosjektstyringsverktøy

Teamet benyttet Azure DevOps for å holde kontroll på sprintene, samt de medfølgende sprint-brettene (boards) for å se hva som var gjort, og hva som skulle bli gjort. Disse brettene bestod av fem kolonner: en for brukerhistorier som ikke var påbegynt, en for påbegynte brukerhistorier, en for 'avventer godkjenning', en for bugs, og en siste for brukerhistorier og bugs som var fullførte. Statusen for elementene, eller brukerhistoriene, ble også markert med en tilstand for å gjøre det lettere å se i den overordnede backloggen av brukerhistorier. Disse tilstandene var 'ny/ikke påbegynt', 'aktiv', og 'ferdig'.



Figur 1: Utkast av gruppens sprint-board

2.2.4. Kommunikasjonsverktøy

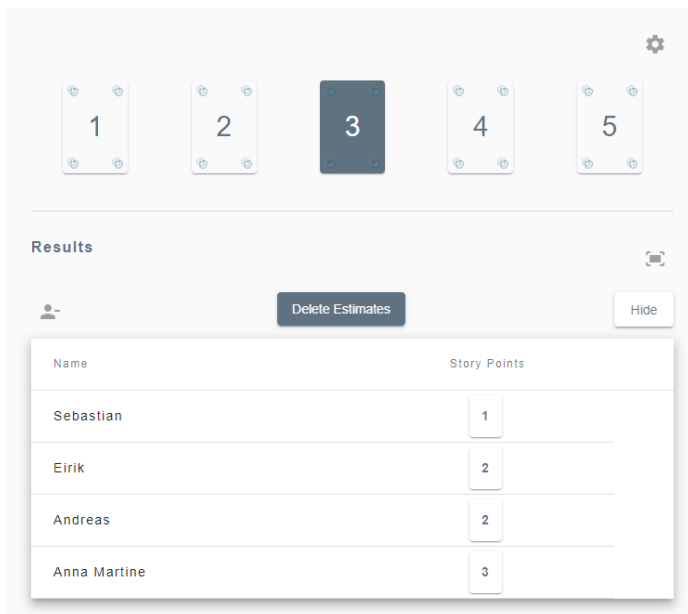
I mer offisiell kommunikasjon mellom produkteier, mentorer, og UiA-veileder ble epost og Microsoft Teams brukt som kommunikasjonskanaler. Internt i gruppen ble en miks av Discord for mer løs planlegging, samt Teams for konkrete tidspunkt benyttet. Møtereferat ble brukt for å føre tidspunkt og sted for neste møte.

2.2.5. Estimering

Gruppen benyttet seg ikke av tidsestimering til brukerhistorier, men brukte heller 'Story Points' som er innebygd i Azure DevOps. Dette ble gjort etter at gruppen fikk en gjennomgang med en scrum master på Capgemini om hvordan de estimerte sine arbeidsoppgaver. De hadde gått bort fra å bruke kvantitativ tid som estimat og gått over til å måle oppgaven i kompleksitet (A. Rueness, personlig kommunikasjon, 19. januar 2023). Story Points estimerer heller den samlede innsatsen som kreves for å fullføre en oppgave, og faktorer som kompleksitet, jobbmengde, risiko og usikkerhet blir tatt i betraktning (Radigan, u.å.).

For å estimere alle brukerhistoriene brukte gruppen Planning Poker. Planning Poker er en estimering- og planleggingsteknikk der hvert medlem får en rekke med kort med påtrykte verdier. Verdiene representerer antall Story Points (Mountain Goat Software, u.å.). Gruppen

bestemte seg for at Story Points skulle gå i en liten Fibonacci-skala (1, 2, 3, 5), der en ener kunne bli gjort på én dag eller mindre, og en femmer burde bli delt inn i mindre deler og estimert på nytt. En brukerhistorie ble lest høyt av scrum master, og hvert medlem trykket på det kortet som samsvarte med det de trodde var riktig kompleksitetsnivå for funksjonaliteten. Den som hadde estimert lavest, og den som hadde estimert høyest måtte diskutere over hvorfor de hadde valgt som de hadde gjort, og alle måtte være enige i sluttestimateret.



Figur 2: Utklipp av scrum-poker

2.2.6. Risikoanalyse

Risikoanalyse er en metode som er laget for å senke usikkerhet ved å analysere mulige risikoer i prosjektet. Deretter å få bedre forståelse på hvordan man kan løse eller forhindre disse risikoene om de skulle oppstå (Karlsen, 2018. s. 431).

For å kunne forutse og redusere mulig skadelige prosjekthendelser, har gruppen utført en risikoanalyse. Modellen som ble brukt i analysen kalles '5x5 Risk Matrix' (figur 3). 5x5 Risk Matrix visualiserer risiko-nivået fra lav til høy ved å kombinere sannsynlighet og konsekvens av hendelsen (SafetyCulture, 2023b).

konsekvens		Ubetydelig	Liten	Moderat	Stor	Katastrofal
sannsynlighet	Veldig sannsynlig	Medium 5	Høy 10	Veldig høy 15	Ekstrem 20	Ekstrem 25
	Sannsynlig	Lav 4	Medium 8	Høy 12	Veldig høy 16	Ekstrem 20
	Mulig	Lav 3	Medium 6	Medium 9	Høy 12	Veldig høy 15
	Usannsynlig	Veldig lav 2	Lav 4	Medium 6	Medium 8	Høy 10
	Veldig usannsynlig	Veldig lav 1	Veldig lav 2	Lav 3	Lav 4	Medium 5

Figur 3: 5x5 Risk Matrix Sample (SafetyCulture, 2023a)

Det har også blitt ført en logg der gruppen skriver ned risikohendelser som oppstod i løpet av prosjektet. Disse blir videre gjennomgått i kapittel 5.

2.2.7. Sprint-plan/Forventet fremgang

Før prosjektet ble satt i gang hadde gruppen et møte med produkteier der tidsrammene ble planlagt. Her ble det enighet om at prosjektet skulle ha åtte sprints som alle var to uker lange. Hver sprint hadde hvert sitt overordnede tema og en beskrivelse av hvilke deler av prosjektet som skulle bli gjennomført i de forskjellige iterasjonene (figur 4).

Sprint	Tema
1	Planlegging og oppsett
2	Design/prototype/brukerundersøkelser
3	Implementering/utvikling
4	Implementering/utvikling
5	Testing/utvikling
6	Produksjonsfase
7	'Backup sprint'
8	Overlevering og evaluering

Figur 4: Sprint-plan med forventet fremgang

2.3. Teknologi

Dette kapitlet tar for seg i mer detalj de teknologiene og hjelpemidlene gruppen tok i bruk under utviklingen. Valgene her tok utgangspunkt i egne erfaringer og forslag fra veiledere i Capgemini.

2.3.1. Frontend og backend

Som nevnt i 2.1.1 ble det valgt å prøve noe nytt. Valget landet på å ta i bruk C# og .NET for selve programmeringen. Valget av C# og .NET kommer blant annet fra et inntrykk om at dette er mye brukt blant bedrifter i Norge. Rammeverket som ble valgt var Blazor, som gjorde det mulig at både frontend og backend programmeringen kunne foregå på samme språk. Blazor er et rammeverk for utvikling av web-applikasjoner med C# og HTML og er laget av Microsoft (Sainty, 2019).

Programmering i samme språk for både frontend og backend, var nytt for hele gruppen, og gjorde det lettere å kunne hjelpe hverandre på tvers av stacken. Blazor har også mange eksisterende biblioteker som ble benyttet, blant annet MudBlazor (som utdypes i 4.3.1), noe som var behjelpelig på brukergrensesnittet.

2.3.2. Database

Det ble bestemt at det skulle settes opp en standard MySQL database. Valget av database var hovedsakelig basert på at gruppen hadde mest kjennskap til denne varianten av SQL. I tillegg var det noen fellestrekk mellom den planlagte databasen og et tidligere prosjekt, slik at gruppen ville være i stand til å gjenbruke eller ta inspirasjon fra tidligere kode, events og oppsett gruppen hadde laget.

Databasen ble i produksjonsfasen knyttet opp mot Azure, hvor faktisk brukerdata ble lagret. Det ble brukt en lokal database for lokal testing som alle kunne sette opp gjennom en SQL-fil med testdata. Det sistnevnte var spesielt relevant for utviklingsfasen av prosjektet, ettersom det ikke var gunstig å bruke ressurser på å kjøre en full server under utvikling. Dette ville også skapt mer konflikter under aktiv utvikling hvis gruppen brukte samme database underveis ved justeringer.

2.3.3. Utviklingsmiljø

Etter løpende dialog både internt i gruppen og med relevante veiledere, kom teamet frem til at Visual Studio 2022 ville bli brukt som IDE (Integrated Development Environment). Grunnen til at denne IDE'en ble valgt var at gruppen skulle bruke en Azure utviklerplattform, som fungerer best i samkjør med Visual Studio. I Azure er det også flere andre teknologier som ble anvendt, blant annet DevOps. Azure DevOps ble brukt som sprint-brett for å se fremgang samt sette opp arbeidsmål og brukerhistorier for de forskjellige sprintene. Her er det også innebygd Git-funksjonalitet i likhet med GitHub som gruppen kunne bruke til versjonskontroll for utviklingsarbeidet (Microsoft Learn, 2022).

3. Kvalitet

Det var viktig for teamet og interessenter at kvalitet ble opprettholdt i prosjektet. Dette kapittelet inneholder punktene som var nødvendige for at dette ble sikret gjennom hele prosessen. Et spesielt viktig punkt var den jevnlige kommunikasjonen med Capgemini som sikret at prosjektet levde opp til standarden de ønsket. Gruppen hadde selv flere egne tiltak, blant annet team-oppsettet og versjonskontroll.

3.1. Kommunikasjon

God kommunikasjon var sentralt for at systemet skulle bli bra og ville dekke kravene til bedriften. Aktiv kommunikasjon var med på å unngå feil og misforståelser, i utviklingsprosessen og i sluttproduktet. Gruppen hadde god kommunikasjon med veileder/produkteier fra bedriften, gjennom både uformell daglig dialog og sprint reviews omtrent annenhver uke. Dette medførte til rask og effektiv oppstart av prosjektet, samt stødig progresjon med forventet kvalitet gjennom hele prosessen.

Kommunikasjonen i gruppen foregikk blant annet gjennom daily scrum. I et daily scrum møte forteller gruppe medlemmene etter tur, hva de har gjort siden sist, hva de planlegger å gjøre i løpet av dagen og om de har støtt på noen utfordringer (Wrike, u.å.b). Resultatet fra dagens arbeidsøkt og planene for neste møte ble også skrevet ned i hvert møtereferat på slutten av dagen.

3.2. Kodestandard

En annen metode gruppen tok i bruk for å sikre kvalitet i produktet var å komme frem til en kodestandard og holde seg til denne (Salian, 2020). Gruppen spurte Capgemini om det var mulig å få retningslinjer til selskapets egen kodestandard for deres C#-kode. Hvis dette ble gjort, kunne koden enklere bli opprettholdt og bygd på av Capgeminis egne ansatte.

Dessverre ble et slikt dokument aldri mottatt, noe som gjorde at gruppen selv måtte bestemme hvordan man skulle korrekt skrive koden. Gruppen bestemte derfor å gå for C#s tradisjonelle kodestandard som var litt annerledes enn det gruppen var vant til.

Programmeringsspråket Java, som gruppen er mest kjent med, bruker camelcase, mens C# bruker såkalt pascalcase når det kommer til navngivning. Disse to navnestandardene har likheter når det kommer til flerordsnavn der hvert nye ord får en stor bokstav. Forskjellen mellom disse er at i camelcase skal starte med en liten bokstav på alle navn unntatt klasser, mens man med pascalcase skal ha stor bokstav på alt unntatt variabel- og parameternavn (Educative, u.å.).

3.3. Versjonskontroll

Versjonskontroll hjelper utviklere å holde kontroll på endringene som blir gjort i koden og reduserer konflikter som ellers kunne ha skadet eller ødelagt koden (Microsoft, 2022). For dette prosjektet bestemte gruppen i samtale med prosjekteier at det skulle brukes Git i Azure Repos for å samkjøre koden. Alle 'branches' måtte ha et forklarende navn, og skulle helst være tilsvarende en task fra en brukerhistorie. Dette gjorde det lettere å holde styr på hva som ble gjort og eventuelt hva som hadde gått galt etter en sammenslåing av koden. Gruppen benyttet også en branch-policy som sørger for at ingen fikk lov til å merge til main uten at minst en annen på teamet har gått gjennom, og godkjent pull requesten deres.

3.4. Team-oppsett

For å maksimere fremgang og kvalitet av produktet, delegerte gruppen seg i to primære puljer. Dette ble et frontend fokusert team, og et backend team. Effekten av dette var at hvert gruppemedlem ikke trengte å bruke unødvendig tid på å sette seg inn i de diverse bibliotekene, dokumentasjon, og medfølgende syntaks-logikk før arbeidet kunne påbegynne. Dette hadde diverse fordeler samt noen ulemper. Så lenge brøkparten av frontend-funksjonalitet var inne, kunne en enkelt påbegynne samsvarende backend-logikk.

Den mest markante fordel var at gruppen som en helhet raskt ble kompetente og effektive på sine respektive områder. Dette ledet til en stødig inkrementell fremgang på brukerhistoriene. Problemstillinger som i starten kunne tatt flere arbeidsdager ble gjort på kortere tid enn hva som originalt var estimert. Denne fordelingen av ansvarsområder mener gruppen førte til et produkt av høyere kvalitet enn hvis gruppen ikke hadde fordelt inn i frontend og backend.

Til tross for at teamet var delt opp i to hovedområder var det fremdeles flere ganger de resterende brukerhistoriene for sprintene krevde at alle jobbet i back- eller frontend. Dette gjorde at teamet fikk en bedre helhetlig forståelse av produktet, noe som spilte positivt inn på utviklingen. Det er tvilsomt om denne kryssjobbingen som trengtes enkelte ganger hadde fungert hvis gruppen gikk for et rammeverk som ikke hadde likt programmeringsspråk i hele stacken.

3.5. Brukertestning

Det er vanskelig å vite akkurat hva brukeren vil ha. Selv om en tror en har funnet den perfekte løsningen, er det ikke sikkert den fungerer like godt blant målgruppen. Brukertestning er derfor viktig å utføre for å finne ut av hva som fungerer og hva som ikke fungerer, samt å avdekke skjult innsikt og øker sjansen for et suksessfullt produkt (Allard, 2021).

I løpet av prosjektperioden har gruppen utført en rekke brukerintervjuer for å sørge for at forventningene fra brukerne matchet produktets retning, og for å forsikre at de fikk et produkt de ville bruke. Ettersom prosjektet fokuserer på brukerorientert kvalitet, som går ut på å tilfredsstille brukerens behov og ønsker (ISTQB Glossary, u.å.), vil det være essensielt å legge vekt på brukertestning.

4. Prosjektgjennomføring

Dette kapittelet omhandler den administrative tilnærmingen og medfølgende struktur gruppen benyttet gjennom prosjektet. Blant annet teamets tilnærming til flere scrum-prinsipper, og en overordnet struktur rundt analyse og applikasjonsdesign. Samt inndeling av sprinter, og de diverse testene som ble utført, og implementering av funn.

4.1. Gjennomføring av scrum

Som tidligere nevnt har gruppen benyttet scrum for å strukturere arbeidsdagen og forsikre fremgang. Her vil rapporten gå over gruppens formelle tilnærming til konsepter som produkt backlogg, sprint backlogg, daily scrum, sprint review og retrospekt, og grunnngi disse valgene og fordelene dette gav.

4.1.1. Produkt Backlogg

Produkt backlogg er en overordnet, dynamisk liste som omhandler de diverse oppgavene som trengs å gjøre innad i prosjektet. Dette ble benyttet som grunnlag for oppsett og planlegging av sprint backloggen (Scrum, u.å.a).

Backloggen tok henhold til brukerhistoriene som hadde blitt fullført, og medfølgende prioritering. Teamet hadde også definert akseptansekriterier til de diverse oppgavene. I akseptansekriteriene definerte gruppen hva som måtte oppnås før brukerhistorien var ferdig; eksempel på dette kan ses i figur 5.

The screenshot shows a Jira user story card for 'USER STORY 22'. The title is '22 Som en bruker ønsker jeg å kunne registrere antall etasjer jeg har gått på lettest mulig måte slik at jeg ikke trenger å bruke mye tid på det.' The author is 'Eirik Berge Starnes' and it has '1 comment'. The card is in 'Closed' status, with 'Reason' 'Acceptance tests pass' and 'Iteration' 'Captrapp\Sprint 3'. The card is divided into several sections: 'Description' (with a 'Click to add Description' link), 'Acceptance Criteria' (listing three criteria: 'Bruker skal kunne registrere antall etasjer kjøpt etter opnet webapplikasjon', 'Forsiden inkluderer "quick options" for å legge til.', and 'Bruker blir spørt om å bekrefte input.'), 'Discussion', 'Planning' (with 'Story Points' 2, 'Priority' 1, and 'Risk'), 'Classification' (with 'Value area' 'Business'), 'Deployment' (with a note 'To track releases a Releases and turn Boards in your pip about deployment'), and 'Development' (with '+ Add link' and two recent updates: 'Registrer trapper tat Created 24.02.2023, ✓' and 'b3b6c14e Merged PI Created 24.02.2023').

Figur 5: Brukerhistorie eksempel

4.1.2. Sprint Backlogg

Sprint backloggen består av sprintmålet og produkt backlogg-elementene valgt for sprinten. Sprint-backloggen ble stadig oppdatert og gav ett sanntidsbilde av arbeidet teamet hadde planlagt å oppnå i løpet av sprinten for å oppnå sprintmålet. Det skulle gi et klart bilde, slik at det var mulig å analyse fremgang i daily scrum-møtene (Scrum, u.å.b). I figur 6 vises et utkast av backloggen for Sprint 3.

Order	Title	State	Assigned To	Rema...
1	> Som en bruker, ønsker jeg å kunne gjenopprette nytt p...	Closed	Andreas Lauv...	
2	> Som bruker ønsker jeg å kunne opprette en egen konto slik ...	Closed	Sebastian Lars...	
3	Leaderboard: Kan fortsette å øke eller minke måned selv om...	Closed	Andreas Lauv...	
4	Leaderboard: Når du trykker på all-time ser du fortsatt pilen...	Closed	Andreas Lauv...	
5	Topbar sentrerer ikke elementene v2	Closed	Anna Linnerud	
6	my profile viser 0 poeng på all time	Closed	Eirik Berge Sta...	
7	> Som bruker ønsker jeg å kunne redigere og slette min egen ...	Closed	Anna Linnerud	
8	> Som en bruker ønsker jeg å kunne registrere antall etasjer je...	Closed	Eirik Berge Sta...	
9	Som en bruker, ønsker jeg å få en oversikt over ukentlige, m...	Closed	Eirik Berge Sta...	
10	> Når jeg deltar i konkurransen og registrerer poeng blir rangs...	Closed	Eirik Berge Sta...	

Figur 6: Sprint 3 backlogg

4.1.3. Sprint Planning

Sprintplanleggingen skjedde i all hovedsak innenfor pre-sprinten, her satte gruppen, i samarbeid med veileder fra bedriften, opp en generell mal for hvordan fremgangen og medfølgende oppgaver skulle se ut. Konkrete brukerhistorier og deres relevante oppgaver ble satt opp i slutten av tidligere sprint samt starten av den nye (Scrum, u.å.f).

Teamet hadde mye fokus på sprintplanlegging fra starten av, ettersom enkelte oppgaver var avhengige av at noen andre oppgaver var fullført først. Som blir nevnt senere i kapittel 4.2.4 benyttet teamet seg av MoSCoW-prioritering som bidro til mer oversikt i sprint planleggingen med tanke på hva som burde bli gjort, og i hvilken rekkefølge.

4.1.4. Daily Scrum

Daily scrum er kjappe møter holdt hver dag for alle team-medlemmene. Hvor teamet i plenum går gjennom deres individuelle fremgang mot sprintmålet. Dette er for at alle skal vite hva de skal jobbe med for neste dag. Om teamet lå bak planen, måtte ressurser bli relokert slik at målet kunne oppnås. Hvis noen var ferdige med sine oppgaver, ble de tildelt nye elementer fra sprint backloggen (ProductPlan, u.å.).

Daily scrum ble ført i møtereferatene, hvor alle skrev hva de hadde gjort siden sist møte, samt hva de hadde gjort for den gjeldende dagen. Det ble også ført opp hva som skulle bli gjort på

neste møte, og om noen oppgaver skulle bli gjort før neste møte var det også et felt for det. Dette gjorde at alle visste hva de andre medlemmene holdt på med, og om det var spørsmål rundt hva noen hadde gjort ble dette oppklart under føringen av møtereferatet.

En slik tilnærming gjorde at ingen trengte å tvile på om noe var under arbeid, og en fikk rom for å påminne medlemmene om å føre inn deres fremgang i Azure DevOps. Dette var en av flere faktorer som bidro til en god og oversiktlig arbeidsstruktur.

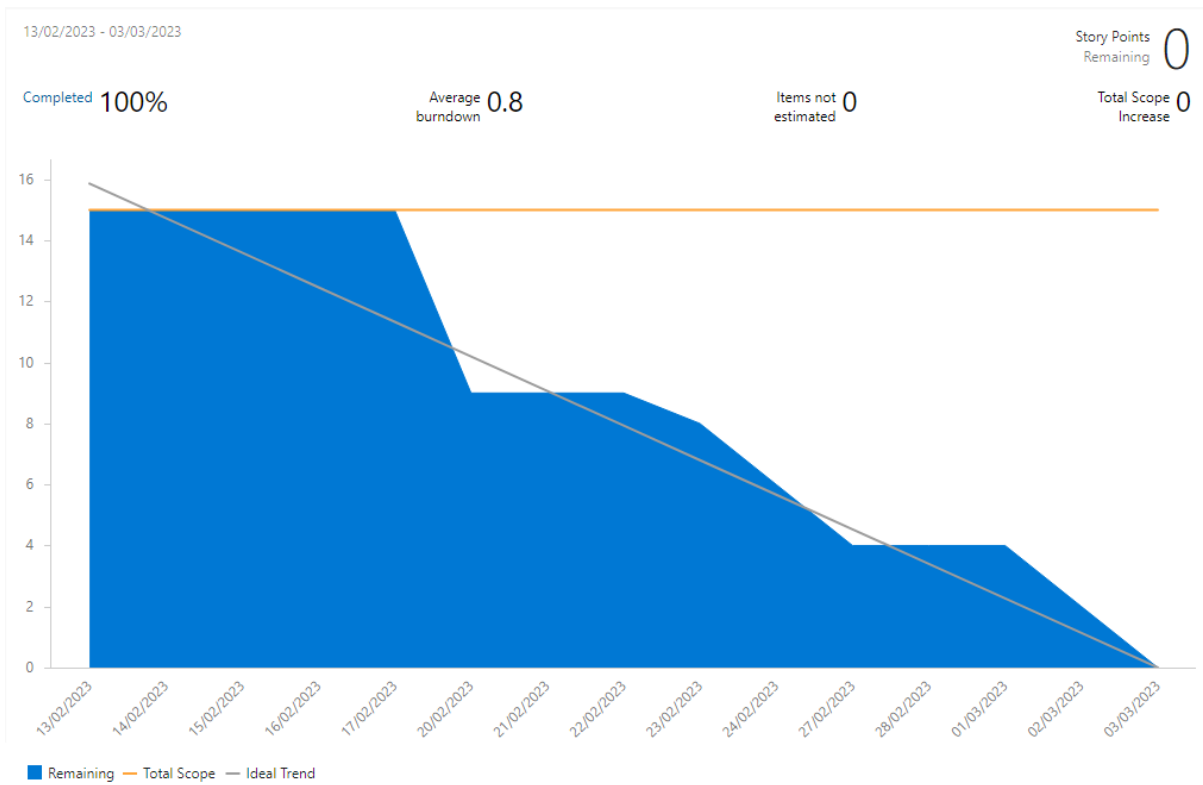
4.1.5. Sprint Review og Retrospekt

Sprint review er en av de siste hendelsene i sprint-omløpet. Her går alle interessentene gjennom hva som hadde blitt oppnådd i sprinten, samt hva som hadde endret seg. Med denne informasjonen kunne deltakerne planlegge hva som skulle gjøres i neste sprint. I denne delen kunne også endringer i produkt backloggen være nødvendig for å reflektere nye oppgaver som oppstod (Scrum, u.å.d).

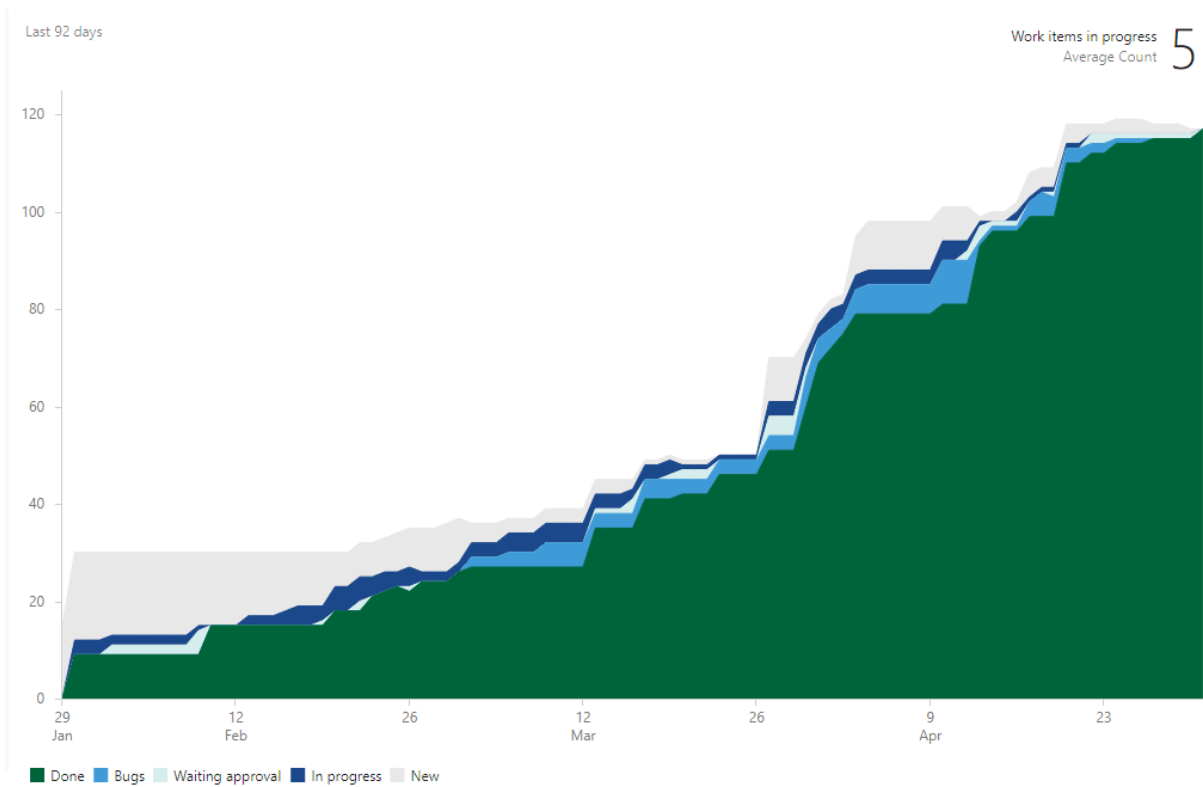
Sprint retrospekt er hva som konkluderer en sprint. Her gjennomgikk gruppen hvordan sist sprint gikk med fokus på individuelle medlemmer, interaksjoner, prosesser, verktøy, og deres akseptansekriterier. Deretter diskuterte teamet hva som gikk bra, hvilke problemer som oppstod, hvordan disse ble løst, eller hvorfor de ikke ble løst. Med dette tatt i betraktning kunne teamet finne nye hjelpelige endringer som kunne bidra til økt effektivitet (Scrum, u.å.c).

I figur 7 er det visualisert en burndown-chart av en sprint-periode. Dette viser fremgangen, men reflekterer ikke oppgaver som var delvis fullført. Fremdeles viser den at sprinten var, gitt konteksten, planlagt godt og fremgangen var som forventet. Ellers viser figur 8 et cumulative flow diagram som belyser i høyere grad arbeidsflyten og fremgang gjennom hele prosjektet.

Enkelte ganger falt gruppen bak eller foran estimer. Det sistnevnte var ofte lettere å håndtere ettersom løsningen var å begynne på oppgaver ment for fremtidige sprinter. Om gruppen lå bak på skjema, så gruppen på hva problemene var og gjorde justeringer i samhold med produkteier.



Figur 7: Burndown Chart av sprint-period



Figur 8: Cumulative flow-diagram

4.2. Analyse og applikasjonsdesign

Dette kapittelet handler om forarbeidet som ble gjort før gruppen kunne sette i gang med selve prosjektet. Slik som brukerens behov, estimeringer og risiko burde bli tatt i betraktning før en kan gå videre til design og utviklingsfasen.

4.2.1. Forstå Problemdomenet

Før gruppen gikk i gang med å designe, var det viktig å sette seg inn i problemdomenet som applikasjonen skulle løse. Uten en klar innsikt i hva problemdomenet er, er sjansen høy for at tjenesten en produserer ikke løser de problemene den var ment til å løse. For å skaffe innsikt i problemdomenet, valgte derfor gruppen å utføre en rekke intervjuer med de ansatte hos Capgemini for å kunne bedre forstå hva slags system brukerne forventet og ønsket.

4.2.2. Intervju

I utgangspunktet var det ikke en detaljert oppgavebeskrivelse gruppen kunne gå etter, men en del oppklaringer og forventinger ble gitt fra produkteier. En av disse forventningene var å lage et 'user-centered design' (UCD), eller et brukerorientert design. I en brukerorientert designprosess involverer en brukere i hele prosessen gjennom ulike undersøkelser og designteknikker, slik at applikasjonen blir brukbar og tilgjengelig for dem som skal bruke den (Interaction Design Foundation, u.å.).

For å oppnå dette utførte teamet seks semistrukturerte intervjuer i Sprint 1 med målgruppen. Intervjuene ble utført og analysert før noen konkrete avgjørelser ble tatt for videre utvikling av applikasjonen. Semistrukturerte intervjuer er en kvalitativ forskningsmetode der en utformer et sett med spørsmål på forhånd, men intervjueren har mulighet til å spørre tilleggsspørsmål eller gå litt vekk fra manus (KnowFife, 2018). Intervjuet som ble holdt startet med noen generelle spørsmål, før de gikk litt mer spesifikt inn på deres forhold til trappekonkurransen. Informantene ble også spurt om de hadde noen forslag til funksjoner eller egenskaper de ville ha med i applikasjonen, og her fikk teamet mange gode forslag.

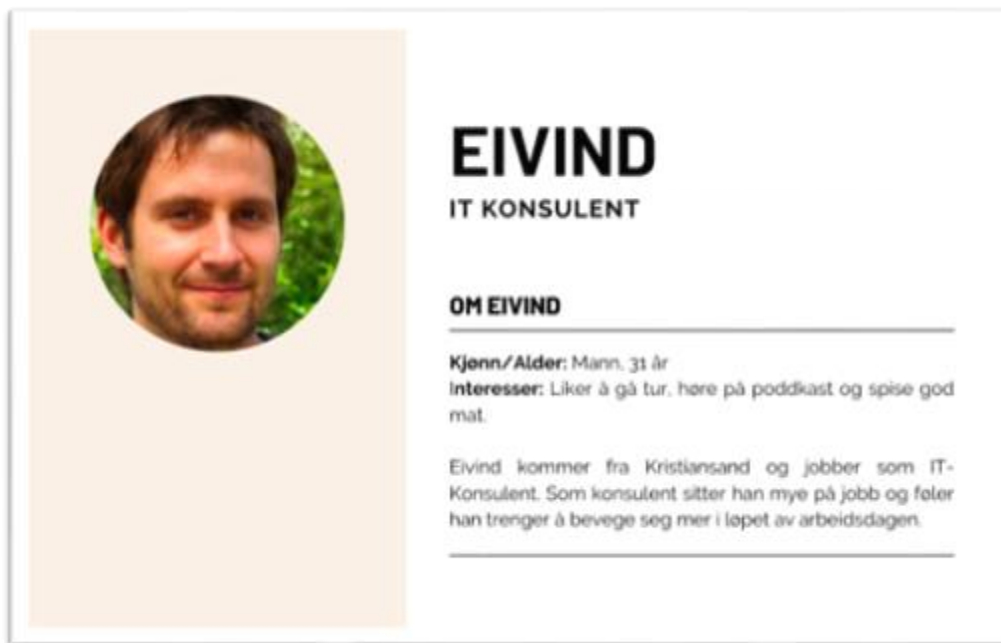
Etter intervjuene samlet gruppen alle svarene i ett dokument og oppsummerte dem.

Oppsummeringene er vedlagt i Appendiks 3. På de mer åpne spørsmålene brukte gruppen fargekoding, der hvert nye tema som informantene nevnte fikk sin egen farge. Dette gjorde

det enklere å sammenligne og identifisere mønstre til videre arbeid, som for eksempel å lage persona eller brukerhistorier (Barcelona Field Studies Centre, u.å.).

4.2.3. Persona

Ut ifra oppsummeringene som ble hentet fra intervjusvarene lagde teamet en persona. En persona er en fiksjonell karakterer som representerer en type bruker som skal forholde seg til ditt system. Personaen hjelper med forståelsen av brukerens behov, erfaringer og mål (Dam & Siang, 2022). I figur 9 kan en se personaen teamet benyttet for å gi ett innblikk i hvem den generelle brukeren av systemet var. Dataen og Informasjonen anvendt ble satt sammen med informasjon hentet i brukerintervjuer fra oppstarten av prosjektet.



Figur 9: Persona satt opp etter brukerintervjuene

4.2.4. Brukerhistorier

En brukerhistorie er en forklaring på en funksjon eller egenskap som er skrevet fra brukerens perspektiv. Grunnen til at en gjør dette er for å forstå hvordan systemet eller produktet skal gi verdi til brukeren. I tillegg, når teamet leser en brukerhistorie fremfor bare en rekke med oppgaver, får de en mer helhetlig forståelse av hva de skal lage, hvorfor de lager det og hvilken verdi de skaper (Rehkopf, u.å.).

Dette prosjektets brukerhistorier tar utgangspunkt i intervjusvarene, slik som personaen.

Teamet har fulgt to maler for å skrive brukerhistoriene, vist i figur 10 (Clayton, 2021, 2:08).

Mal 1	Mal 2
Som en... [bruker/persona] Ønsker jeg å... [handling/oppgave] Slik at... [fordel]	Når jeg... [handling] Skjer dette... [utfall]

Figur 10: Tabell med to brukerhistorie maler

Grunnen til at det ble brukt to maler og ikke en, selv om de dekker samme behov, er for å unngå misforståelser om hva oppgaven faktisk går ut på. Mal 1 fokuserer mest på brukerens verdi og ønske, mens Mal 2 kan bli sett på som litt mer teknisk og er mer rettet mot utvikleren. Den forklarer hvordan den tenkte funksjonen vil fungere og gir et innblikk i hvordan en ferdig funksjon vil se ut (Clayton, 2021, 2:55). I Tabellen under (figur 11) vises alle brukerhistorier i begge formater.

Brukerhistorier	
Mal 1	Mal 2
Som en bruker, ønsker jeg at det er en automatisk overgang til ny poengtelling for hver ny måned.	Når jeg går inn i en ny måned, vil jeg at applikasjonen starter en ny runde med poengtelling automatisk.
Som en bruker, ønsker jeg å kunne gjenopprette nytt passord via email.	Når jeg glemmer passordet mitt, vil jeg å kunne gjenopprette dette på egenhånd ved bruk av mail-verifisering.
Som en bruker, ønsker jeg å få en oversikt over hvem av mine medarbeidere som fikk 1-3. plass forrige måned, gjerne som en pop-up.	Når jeg går inn på applikasjonen, vil jeg få en oversikt over hvem som fikk 1-3. plass sist måned.
Som en bruker ønsker jeg å få en oversikt over ukentlige, månedlige, samt all-time rangstiger for alle brukerne med navn og tilhørende poengsum.	Når jeg går inn på applikasjonen, vil jeg få en oversikt over ukentlige, månedlige, og all-time rangstiger med både poeng og navn.
Som en bruker, ønsker jeg å se statistikken til andre brukere.	Når jeg trykker på en brukerprofil, vil jeg få opp informasjonen til den brukeren.
Som en bruker, ønsker jeg å oppnå medaljer ettersom jeg utfører forskjellige oppgaver som er definert i appen.	Når jeg utfører forskjellige oppgaver i appen vil jeg kunne få medaljer.
Som en bruker ønsker jeg å få en funfact når jeg har oppnådd en milepæl av registrerte etasjer.	Når jeg har oppnådd en milepæl, vil jeg få tildelt en funfact som jeg kan vise på profilen min.

Som bruker ønsker jeg å kunne redigere og slette min egen konto.	Når jeg trykker på slett konto, vil kontoen min bli fjernet fra systemet. Når jeg trykker på rediger informasjon, vil jeg få muligheten til å permanent endre min egen, lagrede informasjon.
Som en administrator ønsker jeg å administrere brukere slik at brukere og brukerinformasjon holdes oppdatert.	Når jeg redigerer eller sletter en bruker så oppdateres lista med brukere
Som en bruker ønsker jeg å se en logg over tidligere vinnere og poengsummer slik at jeg kan se hvordan jeg eller andre brukere har gjort det tidligere.	Når jeg trykker meg inn i 'historikk' så får jeg opp en oversikt over historikken jeg kan trykke inn videre på.
Som en bruker ønsker jeg å se at jeg gjør det godt ved at bakgrunnsfargen endres, slik at jeg kan skryte til andre.	Da jeg kom på førsteplass forrige måned, endret bakgrunnsfargen seg på profilen min til gull.
Som en bruker ønsker jeg at antall trapper gått for alle som er med blir lagt sammen og blir ført i en felles statistikk slik at vi kan jobbe mot et felles mål.	Når jeg registrerer at jeg har gått opp noen trapper, økes fellessummen for trapper gått med tilsvarende poeng som jeg kan se på en side.
Som en bruker ønsker jeg å kunne registrere antall etasjer jeg har gått på lettest mulig måte, slik at jeg ikke trenger å bruke mye tid på det.	Når jeg har gått opp 4 etasjer og skal registrere det, åpner jeg appen og trenger bare bruke to klikk for å gjennomføre handlingen.
Som bruker ønsker jeg å kunne se min og andres plassering i konkurransen slik at jeg vet hvordan andre gjør det eller inspirer til å være mer aktiv selv.	Når jeg vil se hvilken plass meg selv eller andre ligger på i konkurransen, går jeg inn på en leaderboard side hvor alle brukere med sine tilhørende poeng blir vist.
Som en bruker, ønsker jeg at rangstigen i leaderboardet blir oppdatert automatisk.	Når jeg deltar i konkurransen og registrerer etasjer, blir rangstigen automatisk oppdatert.
Som bruker ønsker jeg å kunne opprette en egen konto slik at jeg kan delta i konkurransen.	Når jeg trykker på 'registrer bruker' og fyller inn informasjonen min, oppretter jeg en ny konto, slik at jeg kan være med i konkurransen.

Figur 11: Tabell fylt med brukerhistoriene benyttet

4.2.5. Systemkrav

Ut ifra brukerhistoriene lagde teamet, i samarbeid med produkteier, noen krav for hva systemet måtte, burde og kunne inneholde samt hva det ikke skulle inneholde. Denne måten å kategorisere brukerhistorier på kalles MoSCoW-metoden (Must have, Should have, Could have, og Won't have). De brukerhistoriene som ble kategorisert under 'must' var de funksjonene som var kritiske for at applikasjonen i det heletatt skulle fungere. 'Should have' var de funksjonene som mest sannsynlig ville bli implementert i systemet, men som ikke var like kritisk som 'must'. Videre var brukerhistorier som ble satt til 'Could' mindre prioritert, men de var funksjoner som hadde vært fint å få implementert hvis tiden strakk til. Til slutt var det 'Won't', som ikke ville bli implementert i dette prosjektet, men kunne eventuelt bli lagt inn på et senere tidspunkt (O' Connor, 2016). Alle tabellene er vedlagt i Appendiks 5, under er tabellen for 'must have' funksjonaliteten til applikasjonen (figur 12).

Must have
Som en bruker ønsker jeg å kunne registrere antall trapper jeg har gått.
Som en bruker ønsker jeg å få en oversikt over ukentlige, månedlige, samt alltime rangstiger for alle brukerne med navn og tilhørende poengsum.
Som bruker ønsker jeg å kunne opprette en egen konto slik at jeg kan delta i konkurransen.
Legg inn automatisk oppdatert brukerhistorie her
Som en bruker, ønsker jeg å kunne gjenopprette nytt passord via email.
Som en bruker, ønsker jeg at det er en automatisk overgang til ny poengtelling for hver ny måned.
Som en bruker ønsker jeg å kunne registrer antall etasjer jeg har gått på lettest mulig måte slik at jeg ikke trenger å bruke mye tid på det.
Som bruker ønsker jeg å kunne se min og andres plassering i konkurransen slik at jeg vet hvordan andre gjør det eller inspirer til å være mer aktiv selv.

Figur 12: Tabell med Must-Have brukerhistorier

4.2.6. Skisser

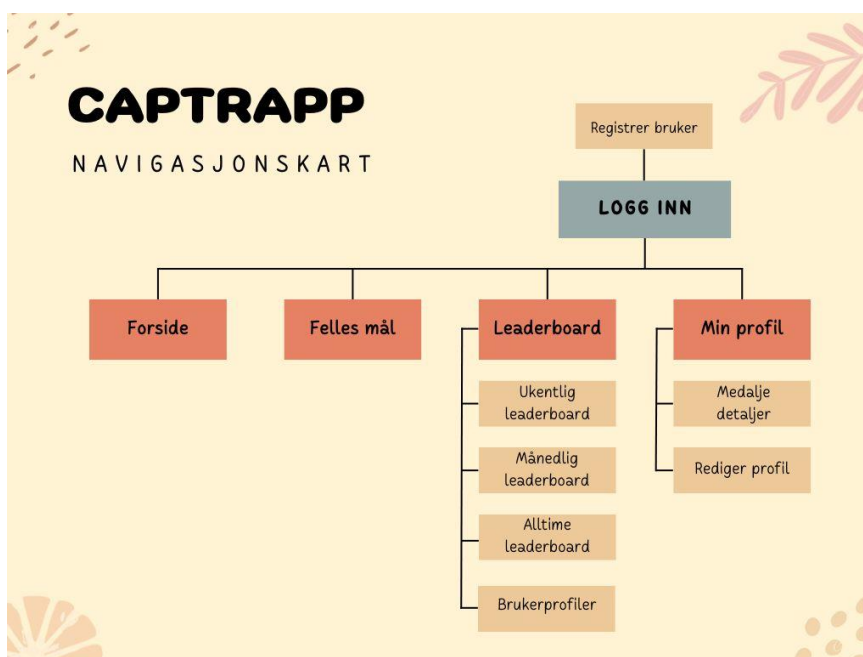
Skissering av idéer er en vanlig og effektiv praksis. Det tillot teamet en enkel og rask måte å få visualisert systemets design. Det gjorde det også mulig å observere problemer, og å kritisere valg som kunne oppfattes som uortodokse. Målet var å oppdage disse mulige problemene før mye tid blir brukt på prototype eller applikasjonen selv. Ettersom skisseringer

var raske å utføre, tillot dette et innovativt og raskt-endrende prosjektdesign, som teamet tok med som erfaring gjennom resten av prosjektet (Tufts, 2014).

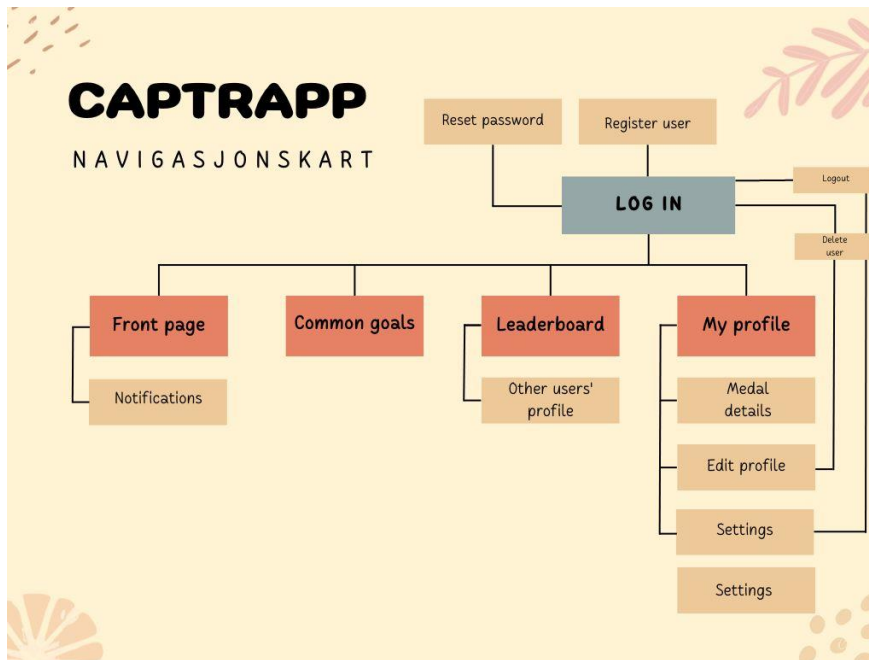
Teamet lagde skisser i løpet av Sprint 1 for å sammenstille idéene, og disse kan sees i Appendiks 2. Disse tillot gruppen å se hva som fungerte med tanke på blant annet navigasjonsmenyen. Før denne prosessen var det forskjellige idéer relatert til posisjonering av denne. Ved å benytte skissering fikk teamet bedre oversikt over de mulige designvalgene, og kom frem til en felles enighet om det beste alternativet.

4.2.7. Navigasjonskart

For å få bedre overblikk over hvordan systemet så ut, lagde gruppen et navigasjonskart. Et navigasjonskart er strukturen på applikasjonen, eller en liste over alle sidene og hvordan en kan navigere seg mellom dem (DesignTLC, 2019). Med et navigasjonskart sikret teamet at strukturen var godt planlagt og at alle hadde samme tankegang rundt systemets oppbygning. Figur 13 er den første versjonen. Etter brukertestene og videre utvikling så navigasjonskartet mer ut som figur 14. Den nyeste versjonen hadde flere navigasjonsmuligheter. Det ble blant annet blitt lagt til en notifikasjonsside på forsiden og en innstillingsside på brukerens profil. Ukentlig-, månedlig- og alltime leaderboard sidene ble fjernet, ettersom dette endte opp som faner innunder leaderboard siden, og ikke individuelle sider.



Figur 13: Navigasjonskart versjon 1



Figur 14: Navigasjonskart versjon 2

4.2.8. Designprinsipper

Gruppen fulgte Benyons 12 designprinsipper under utviklingen av prototypen og applikasjonen. Benyons 12 designprinsipper kan deles i tre underkategorier som er Learnability, Effectiveness og Accommodation. Learnability fokuserer på at applikasjonen skal være lett å lære seg og at det skal være likhet i designet. Effectiveness-kategorien handler om at brukeren skal kunne lett navigere seg rundt, føle at de har kontroll og få feedback på de forskjellige handlingene de har eller skal utføre. Siste kategori er accommodation, og det omhandler fleksibilitet, attraktivt design og vennlighet (Rahman, 2019).

Noen eksempler på learnability som ble realisert var kjennskap i designet og konsekvent utseende og oppbygning. Applikasjonen har en notifikasjonsbjelle som symbol for varsler som er et ofte brukt ikon for varsler i systemer. Den samme fargepaletten går igjen i alle sidene, og tilbakeknappene og tittel er alltid i toppbaren. På effektivitet ble det lagt inn en bunnlinje med en navigasjonsmeny slik at bruker lett kan navigere seg gjennom de ulike sidene, samt feedback på for eksempel vellykket opplasting av nytt profilbilde. Når det kommer til accommodation har gruppen blant annet inkludert en velkomsthilsen til brukere på forsiden, og prompts er alltid formulert høflig og formelt.

4.2.9. Prototype

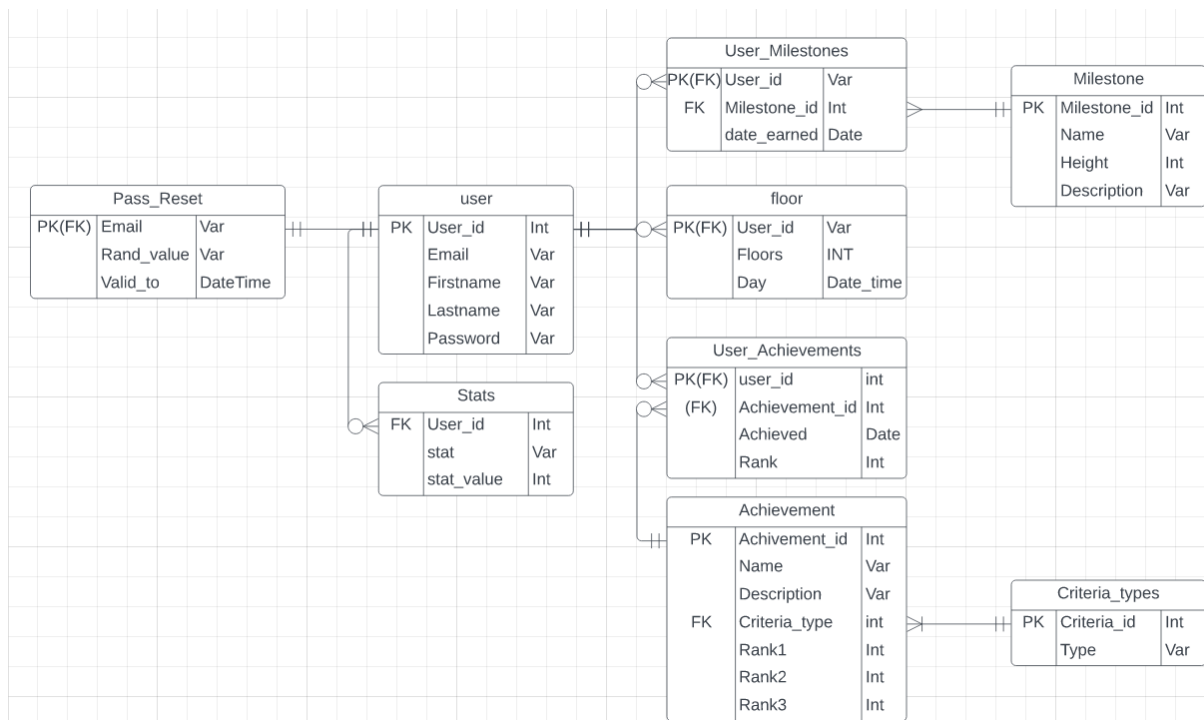
Prototypen ble laget i Figma som er et program på nettet laget for å designe blant annet wireframes og prototyper (Flinn, u.å.). Gruppen bestemte seg tidlig for å fokusere på mobil. Dette ble grunnlagt med at alle de ansatte allerede hadde mobiler lett tilgjengelig, noe gruppen mente ville gjøre at terskelen for bruk av applikasjonen ville bli lavere enn en applikasjon designet for datamaskiner.

Derfor ble størrelsen på sidene valgt til å være det samme som en iPhone 14 Max Pro og prototypen ble designet deretter. Figma er et såpass effektivt designprogram at prototypen ble laget som hi-fi fra begynnelsen av. Med hi-fi (high-fidelity) menes det at den var en fullt funksjonell prototype, og ikke bare for eksempel tegninger av applikasjonen (Usability, u.å.). Prototypen ble designet med farger som gruppen fikk tilsendt og var fargene Capgemini brukte for sitt eget designuttrykk. Ikoner ble valgt ut fra en plugin i Figma kalt 'Iconify', som hadde en database med gratis vektorgrafikk man kunne bruke.

Prototypen skulle originalt bli designet i løpet av Sprint 2, men grunnet rask fullføring av Sprint 1, så ble prototypen påbegynt en uke tidligere enn først planlagt. Naturlig nok var det frontend-laget i gruppen som designet mesteparten av prototypen, men backend kom med innspill i designvei der det var greit å kunne ha flere synspunkter på.

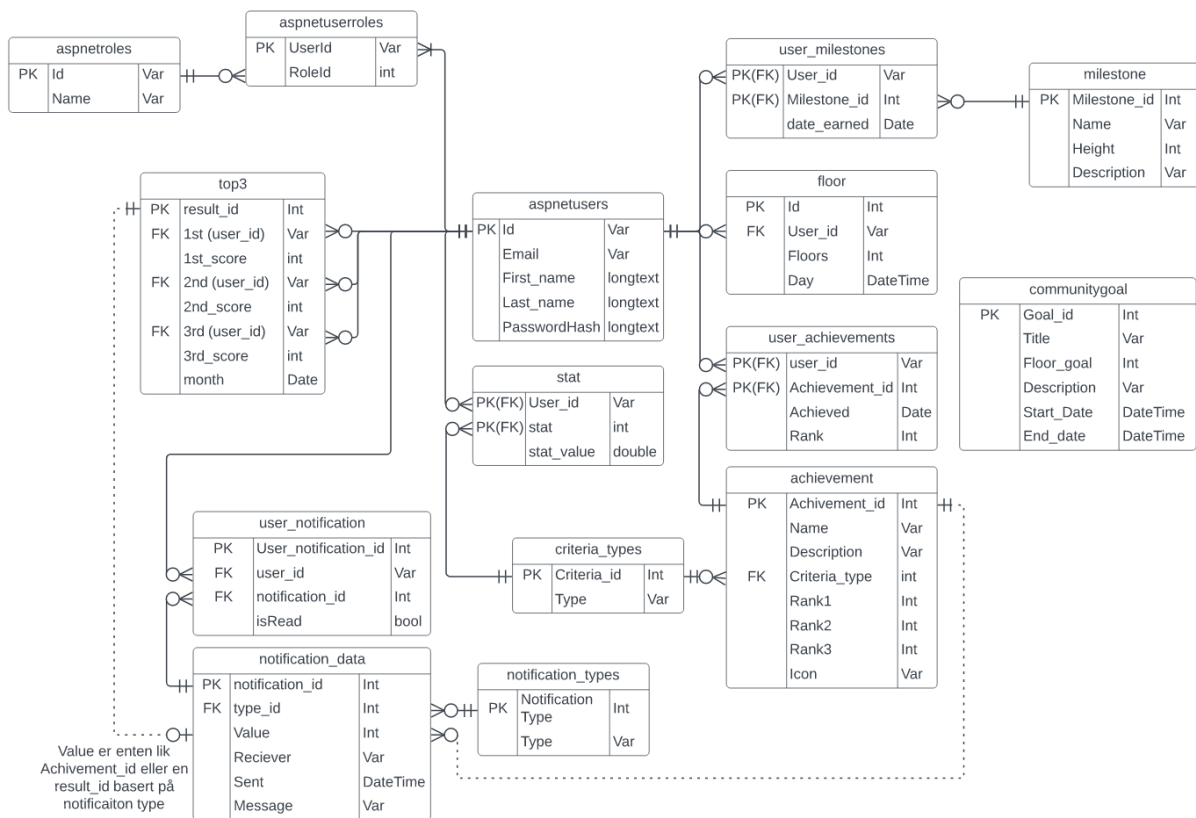
4.2.10. Databasedesign

For å få bedre oversikt over hvilke data som var nødvendig i databasen, lagde gruppen et ER-diagram. Et ER-diagram illustrerer hvordan 'entities' som for eksempel mennesker, objekter eller konsepter relaterer til hverandre (Lucidchart, u.å.). Under Sprint 1 ble det laget et første utkast av ER-Diagrammet for systemet, dette ble så videre justert i Sprint 2, etter hvert som teamet fikk mer innblikk i systemet. Her diskuterte gruppen en del om den mulige strukturen av databasen og hvilke tabeller som kom til å være nødvendige for funksjonaliteten i applikasjonen. Dette ble et godt utgangspunkt for å komme i gang med utviklingen, men det viste seg at diagrammet inneholdt noe feil. ER-diagrammet er illustrert i figur 15.



Figur 15: ER-Diagram Versjon 1

I løpet av sprintene ble det nødvendig å gjøre nye endringer. Endringene kom i flere former som å måtte endre datatyper på noen av feltene, som da enten hadde fått feil type i diagrammet fra før eller gruppen fant ut at måtte være en annen type. Videre ble det gjort endringer som å legge til composite keys, altså to forskjellige felt sammen som blir kombinert til en primary key. For eksempel som en kan se på figur 15 så var user_milistones bare listet opp med user_id som primary key med en feiltakelse. Dette fungerer ikke da primary key er nødt til å være unik og en bruker kan ha flere milepæler. Det ble også lagt til en del nye tabeller som 'top3', 'communitygoal' og notifikasjonsrelaterte tabeller for funksjonalitet som ble utviklet under sprintene. En av de andre større endringene kom fra at bachelorgruppen tok i bruk ASP.NET Identity Framework. Dette førte til en ny brukertabell. Denne inkluderte tabeller som roller og 'claims', pass_reset ble også fjernet ettersom dette blir håndtert av rammeverket. På versjon 2 av ER-Diagrammet (figur 16) er noen av ASP.NET-tabellene ikke inkludert da gruppen har valgt å oppsummere de viktigste feltene fra rammeverket som ble brukt i prosjektet.



Figur 16: Versjon 2, ferdig ER-diagram

4.3. Implementering

Dette kapittelet omhandler noen av rammeverkene som ble brukt i utviklingen for forskjellig funksjonalitet og lettere implementasjon i applikasjonen. Blant annet MudBlazor, Dapper, og overordnet filstruktur. Her nevnes også programvaretesting.

4.3.1. MudBlazor

Gruppen besluttet å ta i bruk et komponentbibliotek for å gjøre programmeringen enklere. Et komponentbibliotek er en skybasert mappe som inneholder ferdiglagde ‘programdeler’ som har et konsistent design som en kan ta i bruk i prosjekter (Brake, 2017). Disse programdelene blir kalt for komponenter og har et fokus på brukergrensesnitt, men inneholder ofte integrert støtte for backend. Eksempler på komponenter som ble brukt i applikasjonen er tabeller, knapper, ikoner og navigasjonsmenyer. Fordelen ved å bruke et komponentbibliotek er at det er en rask og enkel måte å få et estetisk design samtidig som det kan by på mye nyttig funksjonalitet som andre har forhåndsprogrammert (UXPin, 2021). Gruppen ble tipset om MudBlazor fra en av mentorene i Capgemini. Dette er et komponentbibliotek laget for Blazor

som gruppen likte for sitt enkle design og oversiktlige nettside, som gjorde det enkelt for frontend-teamet å finne de komponentene som trengtes.

4.3.2. Dapper

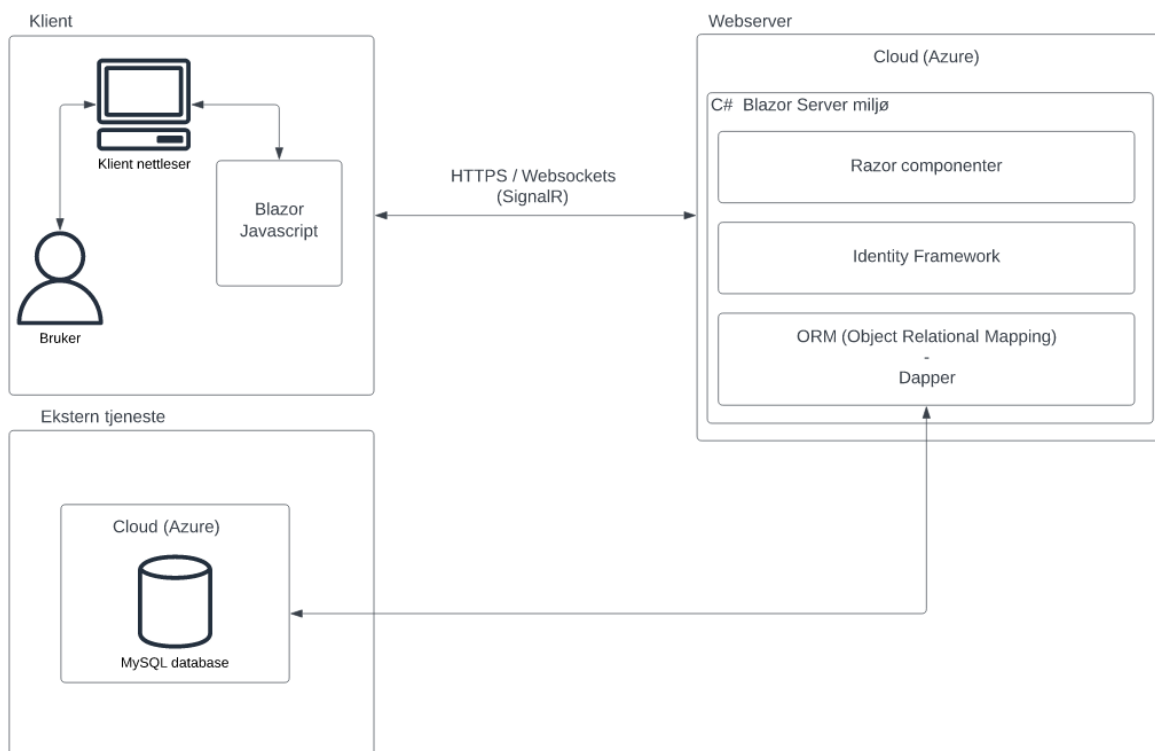
I prosessen med å sette opp et system i koden for å kommunisere med databasen, ble det valgt å ta i bruk en Object-Relation Mapper (ORM). ORM er en måte å lage en bro mellom databasen og et objekt orientert program. Å ta i bruk et ORM verktøy vil forenkle interaksjonen mellom database og koden i det objekt-orienterte språket (Abba, 2022). Dette vil gjøre koden som drar nytte av SQL-spørringer og resultater lettere å bruke, og vil være med på å unngå SQL-injeksjoner. Her fant gruppen Dapper som er et ORM for .NET applikasjoner. Dapper setter opp for å kunne kjøre SQL-spørringer og knytte resultatet fra spørringene direkte til objekter. Det er verdt å nevne av Dapper er definert som et micro-ORM og har bare 'spørring til objekt' funksjonaliteten og ikke resterende funksjonalitet funnet i en fullverdig ORM (ZZZ Projects, 2023).

Basert på spørringene lagrer Dapper dataen i objekter. Disse objektene er definert av gruppen i form av modeller som definerer hvilke data de skal hente fra databasen. Objektene holdt da dataen gruppen trengte på de diverse sidene. Et eksempel på dette var å hente ut brukerinformasjonen og antall etasjer brukeren hadde gått totalt. På figur 17 ser man en SQL-spørring som blir sendt til LoadData-funksjonen som returnerer resultatet i form av en UserLeaderboardModel.

```
string sql = @"SELECT u.id, sum(f.floors) as floors, u.first_name, u.last_name
              FROM aspnetusers u Left JOIN floor f on u.id=f.user_id
              WHERE u.id = @UserId GROUP BY user_id DESC";
user = await _data.LoadData<UserLeaderboardModel, dynamic>(sql, new { UserId = user_id });
```

Figur 17: SQL-spørring for å hente brukerresultat

4.3.3. Systemarkitektur



Figur 18: Systemarkitektur for Captrapp

Systemets arkitektur kan deles opp i flere deler. Disse er illustrert i figur 18. På brukerens side har de en klient i form av en mobiltelefon eller annen enhet med en nettleter, som ved første interaksjon med Blazor-applikasjonen setter opp nødvendige elementer for fremtidig kommunikasjon. Blazor-javascript vil starte en SignalR-tilkobling som vil kommunisere med webserveren ved alle fremtidige tilkoblinger til serveren.

SignalR er en teknologi som muliggjør oppdatering i sanntid for webapplikasjoner (Goswami, 2017). Dette betyr at applikasjonen lar brukeren ha interaksjoner med nettsiden uten at det kreves å sende nye HTTP-forespørsler for hver handling.

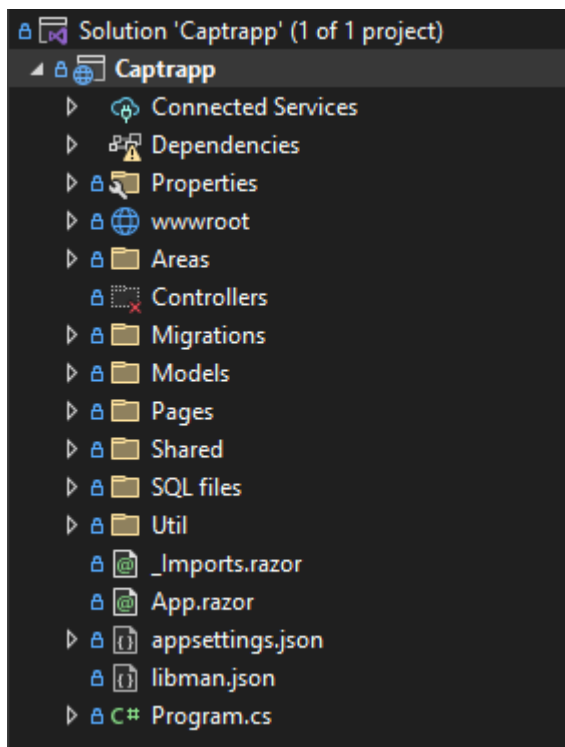
På server-siden er applikasjonen delt opp i Razor-komponenter (med filformatet .razor) som er Blazor-komponentene/sidene i applikasjonen, og Razor-sider (med filformatene .cshtml og .cshtml.cs) som for Captrapp er sidene som ble implementert av Identity-rammeverket, og deretter tilpasset etter produktets behov. Både Razor-komponentene og sidene inneholder logikken og funksjonaliteten, samt det visuelle innholdet for de enkelte sidene.

Den siste hoveddelen av applikasjonen er kommunikasjonen med databasen. Her benyttes

gruppen ORM-implementeringen til Dapper som ble beskrevet i 4.3.2. Dapper tar SQL-spørringene definert i applikasjonen og kommuniserer med en spesifikk database ut ifra en definert 'connection string' i applikasjonen. Resultatene fra spørringene blir returnert til serveren som videre håndterer dataen før den blir sendt til klienten slik den skal vises.

4.3.4. Filstruktur

Filstrukturen i prosjektet er relativt rett frem. Selve frontend-sidene kan deles inn i to kategorier. Hovedsidene ligger i '/Pages'. Disse er sidene brukerne vil benytte mesteparten av tiden, og kan bli ansett som hovedsidene. Core Identity-rammeverket ble automatisk lagt til mappen '/Areas' da rammeverket ble implementert. Ettersom det er et såpass omfattende rammeverk, ble det bestemt at gruppen skulle beholde mappestrukturen rammeverket kom med, for å unngå å ødelegge noe funksjonalitet. I 'wwwroot' ligger profilbildene til brukerne siden gruppen ikke ønsket å lagre bilder i databasen basert på tidligere erfaringer. Her ligger også andre bilder som trengs til applikasjonen, og javascript-kode samt stylesheet til Identity-sidene. Videre så inneholder mappen '/Models' alle modellene som brukes til å representere objekter som er hentet fra databasen. I '/Shared' ligger ting som blir brukt flere plasser i koden, slik som egenlagde komponenter og side-layouts. '/SQL files' inneholder henholdsvis tabellene og dataen for å generere databasen, mens '/Util' inneholder øvrige klasser og interfaces som trengs.



Figur 19: Overordnet mappestruktur i prosjektet

4.3.5. Testing

Testing er en viktig del av utviklingsprosessen og spiller en sentral rolle i prosjekter (Gorný, 2020). Testing tar ofte flere former i et utviklingsprosjekt. For gruppens prosjekt var dette i form av testing utført av medlemmene som utviklere under prosjektarbeidet, samt testing av brukerne i dedikerte brukertester. Brukertestene blir diskutert dypere i kapittel 4.7.

Testingen gruppen utførte selv var hovedsakelig statisk testing og ende-til-ende testing. Statisk testing referer til testing som blir utført uten å kjøre koden. Vanlige måter å utføre denne typen tester på er 'linting' og sjekking av typer (Gathoni, 2022). Linting er en automatisk sjekk av kildekoden for feil i programmet eller stilbaserte feil (Wanjala, 2023). Mange IDE-er har lint-funksjonalitet bygget inn og vil da varsle om feil som ugyldige typer, feil format på navngiving, manglede tegn eller parameter osv. Videre så var det hovedsakelig ende-til-ende testing gruppen utførte. Ende-til-ende testing er en testmetode for å sjekke om flyten i applikasjonen fungerer fra start til slutt fra brukerens perspektiv. Her sjekker en at alle delene av systemet fungerer slik som det skal, og at informasjonsflyten mellom komponentene og mellom bruker og applikasjonen er riktig (Gathoni, 2022). Dette ble gjort i en liten skala etter ny funksjonalitet eller oppdateringer på de enkelte sidene. Hensikten med

dette var å sjekke at systemet oppførte seg som forventet, men også en mer full gjennomgang etter sprinter eller større endringer.

4.4. Gjennomføring del 1: Analyse og design (Pre-Sprint - Sprint 2)

Dette kapittelet går over hva arbeidet var og gjennomføringen i pre-sprinten frem til Sprint 2. Denne perioden i prosjektet var i all hovedsak basert på analyse og design av Captrapp-applikasjonen. Denne delen av prosjektet startet med gjennomgang av hva gruppen visste, intervju om hva fremtidige brukere så for seg og til slutt design og brukertesting av prototypen.

4.4.1. Pre-sprint

Pre-sprinten bestod hovedsakelig av planlegging for en grunnleggende struktur, samt introduksjon til Capgemini som et firma. Gruppen ble mottatt tidlig i januar, hvor Capgemini sine forventinger samt gruppens egne kom frem. Dette ble anvendt videre i sprint-planleggingen som ble gjort i samarbeid med mentorene fra bedriften hvor teamet, som tidligere nevnt, fikk satt opp temaer for alle fremtidige sprinter. Det vil si at medlemmene allerede fra starten kunne gi et estimat på hvor en burde være flere uker i fremtiden. I dette segmentet fikk teamet også på plass en konkret idé på hvilke programmer, samt rammeverk var mest ideelt å bruke.

4.4.2. Sprint 1

For Sprint 1 var temaet planlegging og oppsett. Her benyttet Gruppen intervjuer med brukerne av det fremtidige systemet. Resultatene av dette ble videre bearbeidet og anvendt for å lage brukerhistorier og en persona. Målet var å få en oversikt over hvem brukerne var, samt deres mangfoldige interesser og tanker rundt den fremtidige applikasjonen.

Brukerhistoriene ble deretter vurdert i forhold til MoSCoW-metoden for hva gruppen skulle prioritere. Videre laget gruppen flere skisser for å få en idé om hvordan applikasjonen skulle se ut. Disse skissene ble grunnlaget for videre arbeid med applikasjonens design.

I denne sprinten ble også diverse viktige oppgaver for prosjektet utført. Blant disse var å lage en risikoanalyse (Appendiks 1), og lage et kommunikasjonskart og interessentanalyse

(Appendiks 7). Dermed kunne gruppen ha en viss oversikt over mulige problemer og hvem som skulle kontaktes i hvilke situasjoner.

4.4.3. Sprint 2

Teamet benyttet Sprint 2 til å utvikle en prototype i Figma, samt teste denne på diverse fremtidige brukere fra Capgemini. Samtidig ble alle de opprinnelige brukerhistoriene estimert med kompleksitetsgrad som ble diskutert i avsnittet 2.2.5. Sprint 2 hadde bedre delegering av oppgaver sammenlignet med tidligere. Dette medførte til en forbedring i produktivitet. Denne sprinten ble også brukt til å forberede gruppen til å starte utviklingen i Sprint 3. Noen av forberedelsene gjort her var å ferdigstille versjon 1 av ER-diagrammet, sette opp all nødvendig programvare, Git-oppsett via Azure DevOps og legge inn et felles konfigurert grunnprosjekt.

4.4.4. Retrospekt Pre-sprint til Sprint 2

I Pre-sprint og Sprint 1 følte gruppen det var litt lite å gjøre, mens i Sprint 2 var det grei arbeidsmengde. Noe gruppen kunne ha gjort var å ha satt inn en 'lærings-sprint' som fokuserte på å sette seg inn i den valgte teknologien. Dette kunne ha fylt ut mangel på oppgaver, samt hjulpet å kickstarte utviklingssprinten. Gruppen er veldig glad i å kode, og alle syntes de første sprintene gikk veldig tregt med mye pirkearbeid i Figma og udefinerte oppgaver. Det var også ikke satt av noen pauser utenom lunsjen som gruppen mente ble for tungt. Dette førte til mindre motivasjon og konsentrasjon hos medlemmene, og er noe som ble tatt i betraktning for neste iterasjon.

4.5. Gjennomføring del 2: Utvikling og testing (Sprint 3 - Sprint 5)

Denne perioden av prosjektet dekket hoveddelen av utviklingen av applikasjonen. Det meste av backend funksjonaliteten og frontend-arbeidet med utseende ble gjort her, og avsluttet med en ny runde brukertesting. Selv om det meste ble ferdig i denne perioden, var det en del som ikke gikk etter planen og arbeid ble forskjøvet.

4.5.1. Sprint 3

Sprint 3 var starten på selve utviklingen av produktet. Grovt sett delte teamet seg opp med halvparten på frontend-utvikling og halvparten på backend. Disse rollene var ikke statiske, men delegerte hovedansvarsområder. Det var klart tidlig i sprinten at alle i gruppen hadde behov for mer opplæring i C#, noe som betydde at det var en relativt treig start. Gruppen hadde blant annet estimert en brukerhistorie til å ha Story Point 1 (veldig enkel og raskt gjennomført), men på grunn av manglende kunnskap ble denne ene oppgaven jobbet med i nesten hele Sprint 3. Siste dagen av sprinten hadde teamet sprint review med produkteier der det ble bestemt at ettersom ikke alle brukerhistorier planlagt i Sprint 3 var ferdig enda, skulle sprinten utvides med en uke. Avgjørelsen påvirket ikke planlagt fremdrift ettersom prosjektets sprintplanlegging inkluderte en 'buffer sprint' på to uker som kunne bli brukt til slike situasjoner.

I Sprint 3 fikk teamet i tillegg iverksatt en god praksis på gjennomgang av hverandre sin kode. Git pull-forespørsler krevde godkjenning av andre for å kunne bli pushet til hovedgrenen av repoet. Dette førte til at det var større sjanse for å oppdage feil med pushet kode før den ble integrert inn i hovedgrenen. En annen nyttig ting var at gruppemedlemmene fikk en større forståelse for hva innholdet var i den nye koden som ble integrert.

4.5.2. Sprint 4

Sprint 4 bestod av mindre 'Must have' og mer 'Should have'-punkter. Dermed ble funksjonalitet som loggføring av statistikk, medfølgende medaljer og administratorfunksjonalitet prioritert og fullført i denne sprinten.

Da Sprint 4 var ferdig hadde gruppen som vanlig en sprint review med produkteier. I denne sprinten var det tre brukerhistorier som ikke ble helt leveringsklare. Disse ble alle enige om at skulle bli overført til Sprint 5. I løpet av møtet ble det utarbeidet en prioriteringsliste og plan for neste sprint. Gruppen hadde ikke opplevd noen store problemer denne sprinten, men heller blitt utfordret av tidsklemma. Det har blitt jobbet godt og effektivt, men tidsestimeringen var ikke realistisk nok denne sprinten heller.

Risikologgen ble også oppdatert, ettersom et gruppemedlem var i Heimevernet i fire dager på starten av Sprint 5. Han planla å jobbe på kveldene og i helgen for å ta igjen arbeid, men realistisk sett vil det ikke være mulig å ta igjen 8x3 timer med arbeid i tillegg til tjeneste. Dermed vil det ha en liten påvirkning på fremgangen.

4.5.3. Sprint 5

Sprint 5 var definert av refaktorering og rest-arbeid for å muliggjøre en tidlig utgivelse av applikasjonen. Det største systemet som ble inkludert i applikasjonen var notifikasjoner. Samtidig som dette ble ferdigstilt ble en ny runde med brukerintervjuer også gjennomført, for å se om applikasjonen var fullverdig nok for de fremtidige brukerne. Resultatet av brukerintervjuene var positive, men det var mange forslag til forbedringer og det ble oppdaget en del bugs mens brukerne holdt på. Den resterende delen av sprinten gikk dermed til å løse flest mulig av de nyoppdagede manglene. På et oppklaringsmøte med produkteier fikk gruppen laget en prioriteringsliste der de mest prioriterte oppgavene stod øverst. På dette møte ble det også fjernet en del oppgaver som produkteier mente ikke var nødvendig å gjennomføre i denne omgangen. Det ble ikke en tidlig utrulling av applikasjonen slik som planlagt, ettersom det viste seg å være mye å ordne opp i.

4.5.4. Retrospekt Sprint 3 til Sprint 5

Generelt er teamet fornøyd med fremgangen i sprint 3-5. Det var her majoriteten av applikasjonen ble laget, noe som reflekteres i de siste sprintene, hvor det var lite ny funksjonalitet å legge til. Sprint 3 hadde noen overraskelser i form av feil-estimeringer, og dette ble en viktig erfaring å dra med seg videre. Oppstarten var heller ikke optimal, ettersom mye tid ble brukt på opplæring, og forståelse av rammeverkene som ble benyttet. Dette gjorde imidlertid at gruppen stilte sterkere gjennom resten av sprintene.

4.6. Gjennomføring del 3: Ferdigstilling (Sprint 6 til Sprint 7)

Sprint 6 og 7 dekket de siste tre ukene av aktivt arbeid med prosjektet hos Capgemini. Prosjektet skulle egentlig også hatt en Sprint 8, men denne falt bort, ettersom teamet og prosjekteier bestemte at prosjektet var ferdig, og kunne bli levert inn.

4.6.1. Sprint 6

Sprint 6 bestod i stor grad av bug-fixing, refaktorering og justeringer i forhold til resterende tilbakemeldinger fra brukertesten i Sprint 5. I tillegg ble en testversjon av applikasjonen satt opp i Azure, som lot gruppen teste systemet med flere personer aktive samtidig. Her dukket det opp flere problemer som ikke kom naturlig frem i den lokale testen. Blant annet ble

skyggene på medaljene innrammet, websockets var ikke skrudd på og for brukere som brukte Safari på iPhone fungerte appen bare noen ganger. Dette er feil teamet hadde løst hvis prosjektet skulle ha fortsatt over i en betatestingsfase.

4.6.2. Sprint 7

I den siste sprinten var det bare litt småting igjen som måtte fikses. De siste av bugs ble rettet på og den siste refaktoreringen ble gjort, samt ble det skrevet dokumentasjon for produktet. Ettersom det var veldig lite igjen å få gjort, delte teamet seg opp i to, og det var bare halve teamet som jobbet med prosjektet den siste perioden. Den siste uken hadde gruppen en presentasjon for Capgemini der de presenterte produktet, deres reise og deres lærdommer. Helt til slutt ble det holdt et overleveringsmøte der produktet offisielt ble overlevert til bedriften.

4.6.3. Retrospekt Sprint 6 til Sprint 7

Alle i gruppen syntes ferdigstillingen hadde gått greit, og det er ikke mye de mente kunne bli gjort annerledes. Alt som var planlagt for de siste sprintene ble gjennomført. Hadde dette vært et ordentlig forretningsprosjekt hadde nok to av utviklerne ‘trappet ned’ mot slutten og latt de andre fullføre de siste finjusteringene. Dette gjorde gruppen ved at noen begynte å jobbe med bachelor. De gruppemedlemmene som gjorde dette opplevde det som litt kjedelig, men er glade for at det ble sånn i etterkant.

4.7. Brukertesting

Brukertesting handler om å teste løsningene på ekte brukere i en realistisk kontekst. Det er ryggraden i brukersentrert design. Uavhengig av dyktigheten og erfaringene til teamet, lærer en alltid noe ved brukertester. De bidrar til et bedre innsyn i brukernes ønsker samt smertepunkter (Hasan, 2022).

I løpet av prosjektet ble det utført to forskjellige runder med brukertester. Den første ble utført på slutten av Sprint 2 og den andre i løpet av Sprint 5. Frontend-laget i gruppen utførte disse. Brukertesting ble utført med én bruker om gangen. Gruppen hadde en ordstyrer som spurte spørsmålene og en skribent som tok notater.

4.7.1. Brukertest 1

I denne runden med brukertester ble det utført brukertest med til sammen fem brukere. Alle deltakerne i brukertesten jobber for Capgemini i Kristiansand og er dermed de direkte sluttbrukerne av applikasjonen. Malen for brukerundersøkelsen ligger i Appendiks 4. Da brukerne kom inn ble de takket for at de kunne delta og deretter fortalt hva planen var. Brukertesten ble utført på en PC. Sluttproduktet er tenkt å være primært designet for mobiler, så Figma ble satt til å simulere en mobil. Brukerne satt mellom ordstyrer og skribent slik at en kunne se skjermen til brukeren, og følge med på hva brukeren gjorde. Brukerne ble først bedt om å utforske applikasjonen på egenhånd og oppfordret til å snakke høyt om hva de så, tenkte og følte. Del to av brukertesten var forhåndsdefinerte oppgaver som gruppen hadde laget. Siste del av brukertesting var en oppsummering med generelle spørsmål der brukeren ble spurt en rekke med mer generelle spørsmål. Disse var for eksempel hvordan de ville ha beskrevet og poenggitt applikasjonen, og å be brukeren liste opp noen elementer ved applikasjonen som trekker opp og som trekker ned. Etter spørsmålene ble brukeren takket for at de kom. Mellom hver brukertest diskuterte frontend gruppen hva som kunne bli gjort annerledes, hvilke spørsmål som fungerte og ikke fungerte. For eksempel kom gruppen til konklusjonen etter et par brukertester om at en ikke skulle spørre de konkrete oppgavene hvis brukeren tidligere allerede hadde utført oppgaven under del en. Dette ble for repeterende og enkelt. Det gruppen derimot gjorde var å notere seg hvilke av disse oppgavene brukeren ikke hadde dekket under del en og i så fall be brukeren gjøre disse oppgavene.

Etter brukertesting var ferdig, begynte gruppen med koding av svarene de hadde fått. Notatene ble samlet i et dokument, og fellestrekk ble framhevet med en særskilt farge. Dette gjorde at et klarere bilde av designforbedringer manifesterte seg. Den siste dagen for Sprint 2 ble brukt til å lage en forbedret versjon av prototypen basert på tilbakemeldingene gruppen fikk i brukertestene.

4.7.2. utfordringer under utførelse av brukertest

Under brukertesting kom gruppen over noen få utfordringer. Blant annet merket gruppen fort at brukere ble forvirret av at ikke alle elementer i prototypen var klikkbare. Gruppen hadde som nevnt tidligere laget en hi-fi prototype, men det var allikevel elementer som brukeren ikke kunne aktivere (figur 20). Disse manglene i prototypen var grunnet begrenset

tid. En løsning kunne vært at navigasjonen på elementene endte opp på samme endepunkt. Det ene stedet der dette ble gjort i prototypen oppstod det forvirring rundt sammenhengen mellom valgt element og endepunkt. Dermed vil kanskje begge løsningene ha sine negative og positive sider. Et annet virkemiddel som ville gjort interaktive elementer mer tydelig er utnyttning av ikoner eller fremheving. Dette ville skapt ett mer tydelig skille mellom aktive, og inaktive elementer.



Figur 20: Medaljer der bare en var klikkbar i prototypen

En annen utfordring var det at gruppen var usikre på om de skulle svare på spørsmål underveis som brukeren lurte på, eller om de ikke skulle gi noen hint eller ekstern informasjon. Tanker gruppen hadde rundt dette var:

- hjelpe og så ødelegge for resultatet av hvorvidt appen er logisk og intuitiv.
- Eller ikke hjelpe og la det være pinlig og litt ukomfortabelt, men få et klarere resultat
- Eller hjelpe etter hvert og så heller skrive ned at ‘dette måtte vi hjelpe dem med. De klarte ikke dette..’

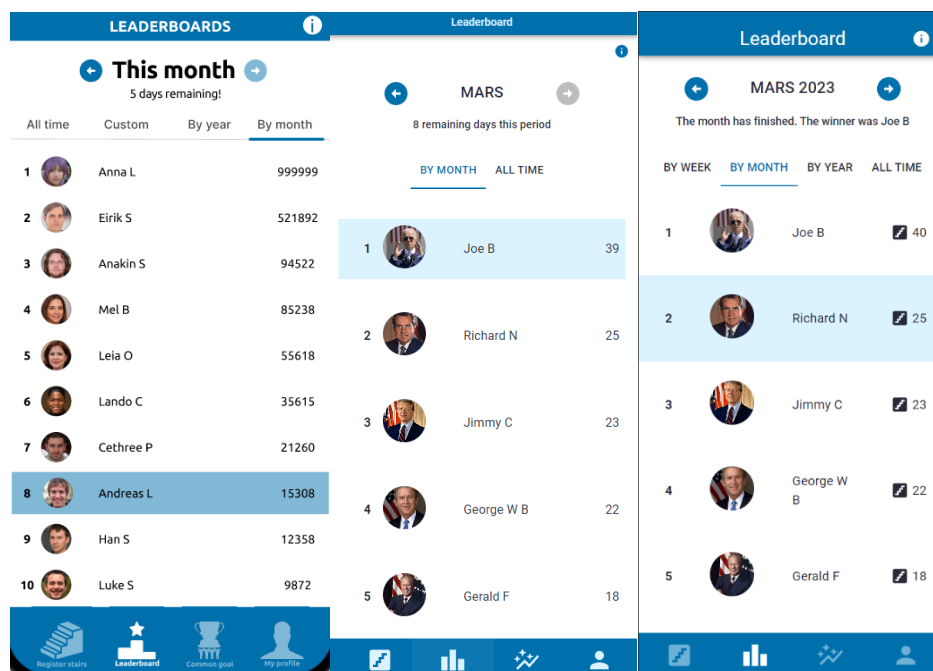
4.7.3. Brukertest 2

Den andre brukertesten gruppen gjennomførte, ble holdt i slutfasen av prosjektet. I motsetning til den første brukertesten, som fokuserte på å oppdage nye konsepter og implementere disse, fokuserer denne brukertesten på redesign og å gjøre forbedringer i applikasjonen (Gautam, u.å.). Brukertest 2 bestod av tre deler. En ‘førsteintrykksdel’, der

brukeren kunne bevege seg fritt gjennom appen. Her noterte skribent brukerens tanker, navigasjon i applikasjonen og eventuelle utfordringer de støttet på. I den andre delen hadde intervjuholderne forberedt noen oppgaver som brukerne skulle gjennomføre, og i den siste delen skulle de svare på et SUS-skjema. Et SUS-skjema er et spørreskjema som blir brukt til å evaluere brukervennligheten til et produkt (Bhat, u.å.). Ved å ha alle disse tre delene med i testingen fikk gruppen innblikk i uventede hindringer, forventede hindringer og den generelle brukervennligheten til applikasjonen.

4.7.4. Endringer som resulterte fra brukertesting

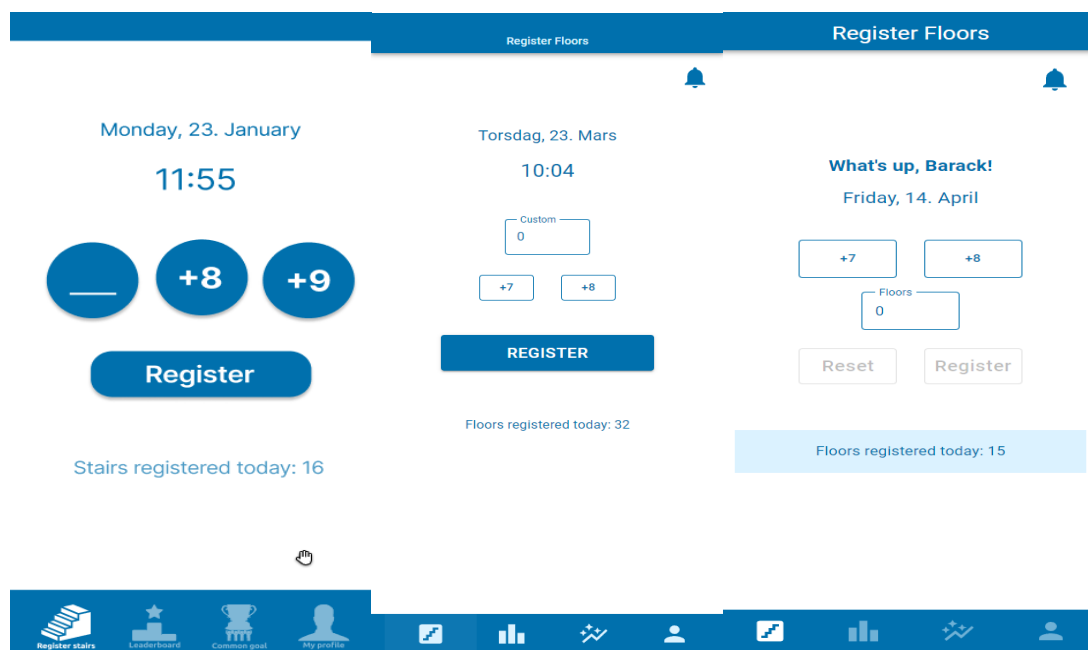
I dette kapittelet blir progresjonen for diverse sider av applikasjonen vist frem. Progresjonen og endringene baserte seg på tilbakemeldingene gruppen fikk fra begge brukertestene. De fleste bildene her vil vise utviklingen fra prototypen benyttet i brukertest 1, hvordan det så ut før brukertest 2 og hvordan sluttproduktet så ut.



Figur 21: Progresjon av leaderboard

I den første brukertesten var som nevnt en Figma prototype benyttet. Her fikk gruppen blant annet tilbakemelding på at leaderboardet var veldig tettpakket, slik som vist til venstre i figur 21. Dette fikk designet til å virke rotete ut. Dermed ble det utført endringer for å gjøre det mer åpent og luftig, slik som en ser i midt-bilde på figur 21. Dette fikk gode tilbakemeldinger

under brukertest 2, og det var bare noen få endringer brukerne ønsket å se på denne siden. Den siste versjonen av leaderboardet er vist helt til høyre i figur 21. Her er det blitt implementert blant annet en egen seksjon for uker i tillegg til år slik som opprinnelig planlagt, og et lite ikon for å illustrere at tallet betydde etasjer besteget. Dette ikonet samsvarer med ikonet i navigasjonsbaren til forsiden hvor registreringer blir utført.



Figur 22: Progresjonen av frontsidene

En av de andre sidene med substansielle endringer var forsiden (figur 22). Hvor brukertestene indikerte at flere ikke forstod hva knappene betydde, spesielt ikke den egendefinerte knappen helt til venstre. Derfor ble dette endret slik at bruker fikk medfølgende tekst, og tydeliggjøring om at det var et input-felt. Det som også ses her, er at navigasjons-baren ble gjort tynnere og mer enkel i den faktiske applikasjonen. Etter runde to med brukertester, var det endringer som at 'custom' ble endret til 'floors' for å indikere at det er etasjer det gjelder. Samtidig vil verdien i 'floors' oppdatere seg basert på +7 og +8 valgknappene slik at bruker kjapt kan kompensere for en glemte registrering. I tillegg ble det lagt til en nullstillingsknapp. Andre endringer var hovedsakelig stilendringer, tidspunkt ble fjernet, og en tilfeldig velkomstmelding ble lagt til.



You did not register last workday

This is just an honesty based check for streak

If you have been away or worked from home you can keep the streak.

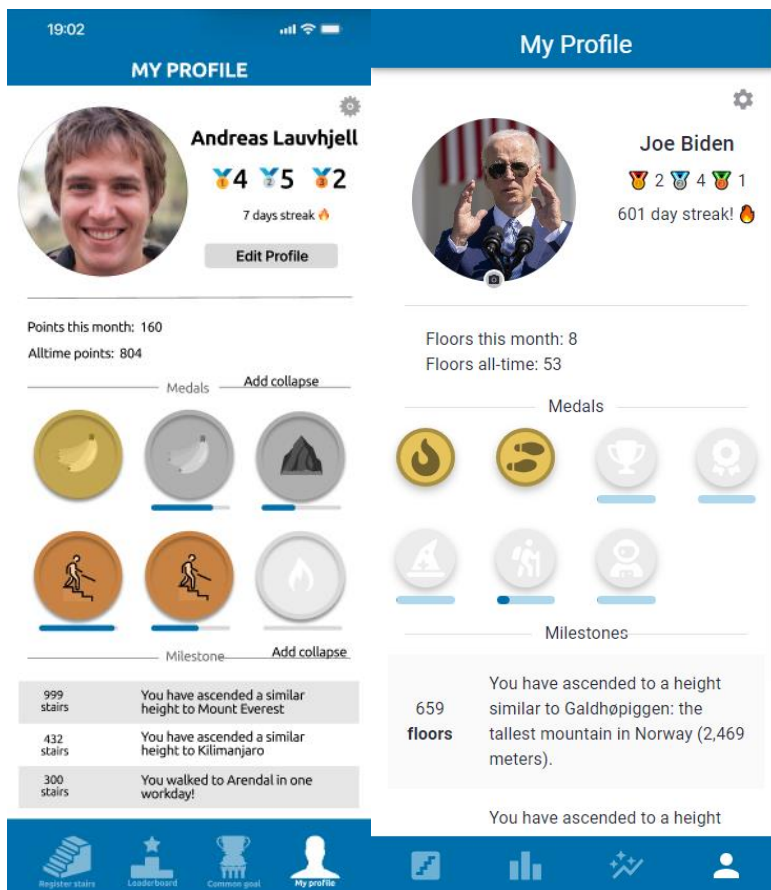
If you skipped the stairs last workday reset streak

Note: If you forgot to register a day, streak increase is lost for that day

RESET STREAK **KEEP STREAK**

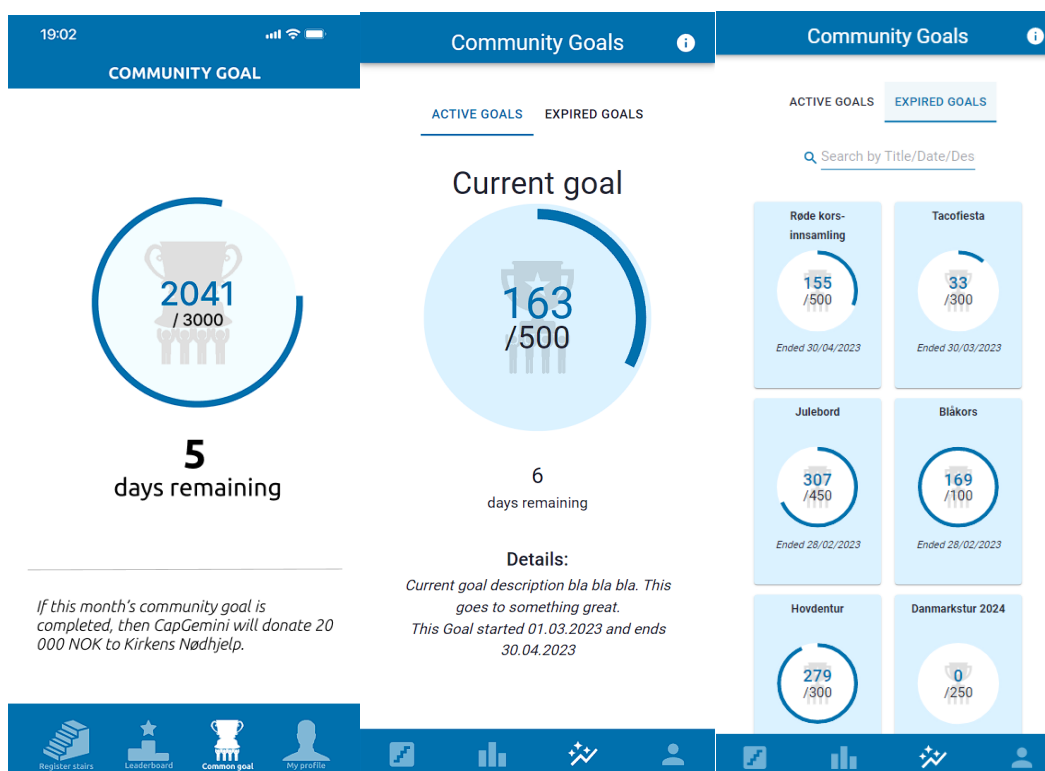
Figur 23: Streak popup

Det kom og opp spørsmål rundt streak funksjonen og hvordan den skulle fungere. Slik det er i sluttproduktet vil brukeren bli presentert med en pop-up boks som vist i figur 23, hvis de har 'mistet' streaken sin. Med streak menes det dager som er blitt registrert på rad, altså hver arbeidsdag. Dette pop-up vinduet gir brukeren muligheten til å beholde streaken hvis de for eksempel har tatt ut ferie og ikke har hatt mulighet til å registrere. Dette baserer seg på ærlighet fra brukernes side. Etersom Capgemini helst ikke ønsket å ha begrensninger internt i applikasjonen (anti-jukse funksjoner) ble dette sett på som den greieste løsningen. Det sparer også applikasjonen for en ekstra query på streak-sjekken, samt at bruker slipper å måtte registrere feriedager på applikasjonen.



Figur 24: Progresjon av 'Min Profil'

På profilsiden til brukerne (figur 24), var de fleste endringene stilvalg. Denne siden forble relativt lik prototypen. Til tross for dette har den hatt endringer som å flytte 'endre profil' knappen inn til innstillinger, som brukeren når via tannhjulet. Etter tilbakemeldingene ble kollapsmenyen til medaljene fjernet, og bare de siste tre milepælene vises uten å måtte trykke på 'vis mer'.



Figur 25: Progresjon for fellesmål

Community goal, eller fellesmål, er også en side som har fått mye endringer. Originaldesignet i prototypen viste bare det ene aktive fellesmålet. Her fikk gruppen klar tilbakemelding på at de ønsket å kunne se de gamle målene også. Som en ser i midten av figur 25, var det mulige å bytte mellom aktive og gamle mål i versjonen benyttet i brukertest 2. Videre fikk gruppen en del tilbakemeldinger på implementeringen her også. Slik det var satt opp før brukertest 2, kom de utgåtte målene i full størrelse og vertikalt etter hverandre. Brukerne syntes dette var lite oversiktlig. Dette førte til at det i sluttversjonen av applikasjonen ble introdusert kompakte kort som kunne trykkes på for å bli utvidet, i tillegg til at det ble inkludert et søkefelt.

5. Refleksjon

Her legges gruppens egne refleksjoner rundt utført arbeid frem, med fokus på hvordan prosjektgangen gikk. Spesielt med tanke på utfordringer, og hvordan disse ble håndtert. Videre nevnes en omfattende funksjon som ble utlatt, samt kvalitet og kvalitetssikring, rollefordeling, og selve prosjektgjennomføringen.

5.1. Utelatte Funksjoner

Den største utelatte funksjonen var et 'arkade'-orientert system, som var inspirert av Mario Kart. I denne tenkte modusen kunne brukerne sabotere hverandre, benytte seg av power-ups, og lignende spillfunksjoner. Dette var også en modus som hadde dekket et ønske noen Capgemini-ansatte hadde om å kunne likestille mellom de som var på jobb hver dag og de som delvis har hjemmekontor. Det viste seg nokså klart rundt Sprint 5 at dette var noe som dessverre måtte nedprioriteres, og fokuset ble heller lagt på å fullføre samt finpusse de nåværende funksjonene for å opprettholde ønsket kvalitet i produktet.

5.2. Kvalitet og Kvalitetssikring

Etter gruppens mening er det viktig for kvalitetssikring at gruppen har samme oppfatning som produkteier i forhold til hvordan sluttproduktet skal se ut, samt prioriteringsstruktur som reflekter dette. Dette spiller spesielt inn på den løpende dialogen gruppen har hatt med Capgemini under prosjektet. Her har det kontinuerlig vært forsøk på å legge seg så nærme Capgemini sin praksis som mulig. Til gjengjeld var Capgemini veldig behjelpelige på å sende ned ansatte for å dele sine kunnskaper. Blant annet fikk gruppen lære om hvordan de taklet problemer, og hvordan en skulle forholde seg til programmeringsstandarder.

Gruppen var klar på at enkelte scrum teknikker, samt en agil prosjektgjennomgang måtte bli satt opp tidlig. Det vil si at gruppen forsikret seg om at alle hadde pågang innenfor sine felt, eller å delegerer hjelp til oppgaver som viste seg å være mer utfordrende enn forventet. Ved å dele opp prosjektet i forskjellige sprints med overordnede mål, var det tydelig hva som skulle bli gjort i hvilken fase.

Videre hadde gruppen også et møte med en scrum master fra Capgemini tidlig i prosjektfasen, for å høre på noen tips og triks relativt til hva som fungerte for dem. Der fikk gruppen blant annet vite hvilken praksis de brukte for estimering, oppgavedelegasjon, og sprint-planlegging. Videre fikk gruppen også se på deres definisjon for hva som avgjorde om en brukerhistorie var ferdig (akseptansekriterier).

Sammen med produkteier hadde gruppen gått gjennom idéer og tanker om hva produktet kom til å inneholde og noen tanker om funksjonalitet. For å sikre kvalitet i prosjektet ble det utført intervjuer med ansatte i bedriften, slik at gruppen kunne se hvilke tanker brukerne hadde for en slik applikasjon. Gjennom disse intervjuene ble det bekreftet at mye av det gruppen og

prosjekteier hadde kommet frem til var ønsket funksjonalitet. De ga også nye idéer til funksjoner som gruppen kom frem til at burde være med. Dette ga et godt grunnlag for videre planlegging, design og utvikling. Dette ble senere bekreftet gjennom brukertester av applikasjonen. Disse var med på å bekrefte at prosjektet var på rett spor, og kunne bli brukt til revidering og forbedring av funksjonalitet.

5.3. Rollefordeling

I starten var det diskusjoner rundt hvordan rollene skulle fordeles i gruppen. Som nevnt tidligere opererte gruppen med to forskjellige team, ett frontend, og ett backend. Dette fungerte fint i praksis, og medførte jevn fremgang ettersom alle visste hvor de skulle spisse seg inn. Til tross for dette er det vesentlig å nevne at en slik tilnærming reduserte individuell breddekunnskap. Samtidig betydde det derimot at alle jobbet der de hadde mest lyst, og fikk fordypet kunnskap innenfor disse feltene. Teamet følte også at dette bidro til en mer givende prosess, ettersom det var mer interesse for oppgavene en fikk tildelt.

Da det kom til teamets tilnærming til prosjektets rollefordeling valgte gruppen som nevnt å ha scrum master rollen på rotering slik at alle skulle få denne erfaringen. Dette fungerte fint, men gruppen hadde fremdeles en formell gruppeleder som var ansvarlig for kommunikasjonen med veiledere. Ellers ble alle store valg tatt i plenum, og alle tok høyde for medlemmenes individuelle behov, frister, intervjuer, og generelle tidsplan.

5.4. Prosjektgjennomføring

Prosjektgjennomføringen har vært en givende og lærerik prosess. Tidligere har risikomatriksen blitt nevnt, og tilnærmingen til denne blir utdypet i risikohåndtering. Dette samt hvordan estimeringen gikk i ettertid er hva som kommer til å bli dekket i dette kapittelet. Deretter skrives det om flere læringsutbytter, erfaring, og en avsluttende konklusjon.

5.4.1. Risikohåndtering

Som nevnt tidligere, har gruppen i løpet av prosjektperioden ført en risikologg (Appendiks 6). Loggen har blitt fylt ut med alle hendelsene som har oppstått som har bremset fremgangen av prosjektet. Det har forekommet totalt ti hendelser som er ført opp i loggen. Det har ikke

oppstått problemer med veldig stor risiko konsekvens, men de som ble ført ned var av typen ‘mangel/feil på utstyr’, ‘Mac kompatibilitet’, ‘manglende oppfølging av bedriften’ og ‘mild sykdom’. Den risikoen som påvirket framgangen mest, var Mac problemer. Dette er noe gruppen har hatt erfaring med tidligere. Det var blant annet alltid feil med phpMyAdmin, events i databasen ville ikke fungere, og ‘hot reload’ i Visual Studio var ikke implementert i Mac versjonen av programmet. Dette gjorde at et gruppelem verken kunne kjøre nyeste versjon av database eller jobbe effektivt på laptopen sin. Ifølge risikoanalysen skal denne risikotypen håndteres ved å ‘google/finne en løsning’. Etter å ha gjort dette i flere dager uten noen løsning, ble det bestemt at personen måtte heller gå over til hjemmekontor på den stasjonære PC-en, for ikke å miste mer tid på prosjektet.

5.4.2. Estimering med Story Points

Å bruke Story Points fremfor å estimere i timer har fungert veldig godt for teamet. Alle jobber i forskjellig tempo, og dermed virker det mer logisk å heller sortere brukerhistorier etter kompleksitet. Gruppen har aldri estimert med Story Points før, så det var noen oppgaver som ikke reflekterte den sanne kompleksiteten, men på den generelle basisen var estimeringen en suksess. Av brukerhistorier som gruppen bommet ganske mye på var det blant annet denne: *‘som bruker ønsker jeg å kunne opprette en egen konto slik at jeg kan delta i konkurransen’*. Når denne ble estimert brukte gruppen sine PHP-ferdigheter (et annet programmeringsspråk) som referanse, og tenkte at denne oppgaven kunne bli gjort relativt raskt. Dermed ble den gitt Story Point ‘1’. Oppgaven endte opp med å bli løst ved bruk av Identity Framework, hvilket tok en del lengre tid å finne ut av enn i PHP. Skulle brukerhistorien ha blitt estimert på nytt, ville gruppen gitt den Story Point 5, ettersom den tok nesten en hel sprint å løse. Grunnet den høye poengsummen hadde det kanskje også vært gunstig å dele brukerhistorien inn i mindre oppgaver, slik at disse kunne estimeres med en lavere sum og bli gjennomført i løpet av maks en dag.

Videre overestimerte gruppen også hvor komplisert det var å skifte temafarger i applikasjonen. Denne ble også satt til Story Point ‘1’, men den krevde mer arbeid grunnet problemer med informasjonskapsler og hardkodet CSS. Estimering er vanskelig uansett om det er Story Points, timer eller arbeidsdager. Det var vanskelig å beregne kompleksitet for et programmerings språk som gruppen aldri har brukt før. Noe en kunne ha gjort var å heller overestimere på poengene, slik at man også tar med i beregningen selve prosessen av å lære seg det man trenger for å kunne løse oppgaven.

5.4.3. Testing

Det er verdt å nevne at gruppen ikke har brukt enhetstester i utviklingen av dette prosjektet. Gruppen valgte å ikke inkludere enhetstester basert på flere grunner. Den første grunnen er at prosjektet er kortvarig og gikk kjapt mot prosjektets frist. Gruppen bestod av fire personer og det var mange oppgaver å utføre, så gruppen fokuserte på utviklingen og å skape et godt sluttprodukt. Den andre grunnen var at gruppen allerede hadde aktiv manuelltesting underveis i utviklingen. Både for funksjonene etter hvert som de ble laget, og under de fulle ende-til-ende testene. Denne manuelle testingen plukket opp de fleste bugs, både på det visuelle og funksjonelle. Den siste viktige grunnen er at applikasjonen i stor grad kommuniserer aktivt med en database i de fleste funksjonene og sidene. Dermed kunne det bli en tidskrevende jobb å opprette relevante endepunkter via «stub» eller «mock»-metoder. I tillegg var det ikke garanti på at disse testene ville avdekket feil som vi ikke allerede oppdaget kjapt i de manuelle testene, spesielt visuelle feil.

5.4.4. Konklusjon og læringsutbytte

Produktet Capgemini mottok fra gruppen oppfylte alle de grunnleggende punktene gruppen hadde fått. Til tross for dette var det mye mer som gjerne skulle bli gjort, men dette er noe som nesten alltid vil være tilfellet i slike systemer. Dermed anser verken teamet eller oppdragsgiver dette som et nederlag, og tar heller med seg erfaringene inn i fremtidige prosjekter. Alle gruppemedlemmene sitter igjen med verdifulle ferdigheter og erfaringer innenfor Blazor, både front- og backend. Samt har gruppen spesielt fått en videreutvikling i arbeidsmetodikk og administrering. Før starten av prosjektet hadde gruppen tidligere jobbet i formelle strukturer, og dette prosjektet tillot teamet å benytte all denne kunnskapen for å forsikre en givende fremgang. Målene var kanskje i starten litt optimistiske, og dette vil være et vesentlig punkt å ta med seg videre. Generelt vil alle erfaringene være viktig for fremtidige prosjekter, oppgaver innenfor lignende felt, og ikke minst arbeidslivet. En videodemonstrasjon av applikasjonen kan bli funnet i Appendix 8.

6. Uttalelse fra oppdragsgiver

Capgemini er et konsultentselskap med over 380 000 medarbeidere i over 50 land. I Norge er vi cirka 1600 ansatte og etablerte vårt kontor i Kristiansand høsten 2021. I forbindelse med inngåelse av kontrakt for leie av lokaler i Quadrum ble vi også gjort oppmerksom på hvordan huseiere møter nye behov i forbindelse med utleie av moderne lokaler i en post-Covid sammenheng.

Siden 2021 har kontoret i Kristiansand hatt en god vekst og med flere ansatte skapes også nye behov. For å aktivisere de ansatte har det blitt kjørt konkurranser i å gå opp alle etasjene til kontoret. Etter hvert som dette ble mer populært, har behovet for en applikasjon til dette forformålet blitt løftet. Capgemini tok med seg forespørselen til 'RefreshIT'. 'RefreshIT' er et arrangement i regi av Universitetet i Agder, og en mulighet for bedrifter å presentere seg for bachelorstudenter. Her inviteres bachelorstudenter inn til prosjekter hvor de kan arbeide sammen med bedrifter i forbindelse med bacheloroppgaven. Capgemini møtte her mange interessante studentgrupper, og hadde også intervjuer med flere av disse i etterkant.

Valget vårt falt på gruppen bestående av Andreas Lauvhjell, Anna Martine Støme Linnerud, Eirik Berge Stamnes, og Sebastian Larsen. Gruppen ble valgt fordi de fremsto faglig dyktige, strukturerte og viste en entusiasme for både Capgemini og prosjektet. Det var også en gruppe som hadde arbeidet mye sammen før, og hadde en god fordeling ift. interesser og kompetanseområder. Det var også en klar rollefordeling i gruppen.

I gjennomføringsperioden har Capgemini bistått prosjektgruppen med ressurser tilknyttet veiledning. Disse har gitt opplæring og veiledning i Azure, prosjektledelse, og presentasjonsteknikk. I tillegg til dette har prosjektgruppen gjennomført en rekke intervjuer med ansatte i Capgemini for å sikre kvalitet og kontinuerlig feedback underveis.

Prosjektgruppen har utvist stor grad av selvstendighet og evne til å ta disse valgene selv, men også evnet å spørre om råd og veiledning ved behov. Gruppen har bekreftet de initiale inntrykkene fra intervjuene og har gjennom hele prosjektet opptrådt svært profesjonelle og entusiastiske. De har også vist en stor grad av læringsvillighet i forhold til nye metoder og verktøy som har blitt introdusert for dem gjennom prosjektet.

Prosjektet har vært gjennomført etter smidig metodikk ('Scrumban') hvor Azure DevOps har blitt brukt for både kildekodekontroll og prosjektstyring. De har benyttet Capgemini sine lokaler under sine arbeidsmøter 2 ganger per uke. Representanter fra Capgemini har hatt regelmessig kontakt med prosjektgruppen og fulgt fremdriften gjennom sprint reviews og

styringsmøter med veileder fra UiA. Capgemini sin observasjon er at gruppen har utvist stort engasjement og forpliktelse i prosjektet. Gruppen har tilpasset seg de utfordringer og endringer som har oppstått underveis, samtidig som de har vært bevisst på tidsbruk og levert på tidsfrister. Gruppen har hatt tydelige roller og gjennomført prosjektet eksemplarisk.

Capgemini er svært fornøyd med det prosjektgruppen har levert og den innsatsen de har lagt ned. Gruppen presenterte sluttproduktet under et fellesmøte for alle ansatte i Kristiansand hvor de fikk svært gode tilbakemeldinger. Vi er veldig stolte av å være en del av dette bachelorprosjektet og ønsker å utvise takknemlighet overfor gruppen for godt samarbeid.

Lykke til videre!

Kristiansand, 13. mai 2023

Med vennlig hilsen

Vegard Sætre

Consultant, Capgemini

7. Egenvurdering

Andreas Lauvhjell

Gjennom prosjektet har det vært interessant å endelig sette seg inn i frontend-arbeid, som jeg synes har vært nedprioritert i fagene på studiet. Det har vært utfordrende, men lærerikt å holde på med. Jeg føler jeg nå er i en posisjon der jeg har en del forståelse for frontend-koding, noe som jeg ikke hadde da jeg gikk inn i prosjektet. Sett bort ifra programmeringa så har jeg vært med å bygge prototypen og deltatt på samtlige brukerintervjuer og tester. Det beste med prosjektet har vært å få kunne lære seg C# som jeg har fått veldig sansen for og som jeg har sett mange jobbutlysninger ettersøker. Jeg er takknemlig for Java-programmeringa vi har holdt på med tidligere semestre, da det er svært stor overføringsgrad mellom disse språkene. Gruppen har hatt en svært god dynamikk og har jobbet effektivt med tida som har vært til rådighet. Jeg vil si at dette semesteret har soleklart vært det semesteret der jeg har lært mest og jeg setter utrolig pris på at det virker som en veldig virkelighetsnær måte å jobbe, framfor å måtte sitte på en skolebenk.

Anna Martine Støme Linnerud

Det har vært veldig moro å få jobbe med bachelorprosjekt hos Capgemini. Jeg har fått lære mye om hvordan bedriften jobber og hvordan det er å samarbeide med en 'kunde' om et sluttprodukt. I gruppen har jeg vært gruppeleder, men vi har hatt en ganske løs ordning til det. På selve prosjektet har jeg jobbet med Frontend. Vi brukte mye MudBlazor, som både hadde sine fordeler og sine ulemper. Det var interessant å bruke C# over hele stacken, men jeg tror nok ikke det er favorittspråket mitt i frontend. Vi som jobbet i frontenden, hadde også hovedansvaret for å holde brukerintervjuer og undersøkelser samt prototype. Samarbeidet innad i gruppen har funket veldig godt, og jeg er glad for at vi bestemte oss for å dele oss i frontend og backend. Det har gjort at vi har fått til så mye mer enn hvis alle skulle ha satt seg inn i begge sider.

Eirik Berge Stamnes

Denne Bacheloroppgaven, og medfølgende prosjekt har vært svært givende. Jeg har fått mye bedre innsikt i hvordan private bedrifter tar i bruk arbeidsmetodikk, samt hva en kan forvente å møte i arbeidslivet. På selve web-applikasjonen har jeg og Sebastian jobbet mest på backend delen av systemet. Det var kunnskapsrikt å lære C# samt flere av rammeverkene og bibliotekene vi benyttet oss av, samt å forholde seg til kvalitet forventingen Capgemini hadde. Å jobbe med Azure DevOps har også gitt meg et mye mer dybdesyn på bruk av scrum-metodikk og medfølgende prosjektadministrering.

Sebastian Larsen

Eg syntest bacheloroppgåva og prosjektet har vore veldig interessant å jobbe med. Vi begynte med ei relativt fri oppgåve som har hatt sine fordeler og ulemper, men har gjort at vi har hatt eit spennande prosjekt heile vegen. Prosjektet lott oss lære mye nytt om C#/.NET og Blazor, og ga en god arena for å anvende det vi kunne frå før og fordjupe kunnskapane rundt mange deler av prosjektarbeid. Samtidig hadde vi ei god fordeling mellom backend og frontend noko som let oss spela på styrkane til gruppemedlemane og gav god framgang i prosjektet. Eg har i stor grad jobba backend saman med Eirik gjennom det meste av prosjektet, men har og hatt hovudansvar med databasen.

Det har blitt mye god lærdom av å jobbe saman med Capgemini, spesielt da vi fekk en flink rettleiar / produkteiar som vi har hatt mye kontakt med gjennom heile prosjektet. Dette har gjort at vi har fått reell erfaring på korleis scrum og prosjektarbeid kan bli utført hos bedrifter, som er godt relevant erfaring å ta med seg vidare.

8. Litteraturliste

Abba, I.V. (2022, 21. oktober) What is an ORM – The meaning of object relational mapping database tools. Hentet fra <https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/>

Allard, L. (2021). How to Improve Your Product With User Testing. Hentet fra <https://avocode.com/blog/how-to-improve-your-product-with-user-testing>

Barcelona Field Studies Centre. (u.å.). Qualitative Research in Geography using Coding Analysis. Hentet fra <https://geographyfieldwork.com/CodingAnalysis.htm#:~:text=The%20coding%20process%20involves%20searching,that%20can%20be%20investigated%20further>

Bhat, A. (u.å.). What is System Usability Scale? Hentet fra <https://www.questionpro.com/blog/system-usability-scale/#:~:text=Advantages%20of%20System%20Usability%20Scale&text=Since%20there%20are%20definitive%2010,various%20software%2C%20hardware%20or%20websites>

Brake, R. (2017). Advantages of a Component Library. Hentet fra <https://www.getopensocial.com/blog/open-source/advantages-component-library/>

Capgemini. (u.å.). What we do. Hentet 23.01.23 fra <https://www.capgemini.com/about-us/who-we-are/what-we-do/>

Clayton, M. (2021, 22. juli). What are Epics, User Stories, and Story Points? [Video]. YouTube. <https://www.youtube.com/watch?v=ogV2r9579WI&t=129s>

Dam, R. F., & Siang, T. Y. (2022). Personas – A Simple Introduction. Hentet fra <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them#:~:text=Creating%20personas%20will%20help%20you,user%20you're%20designing%20for.>

DesignTLC. (2019). What Is a Sitemap and Why Do You Need One? Hentet fra <https://designtlc.com/what-is-a-sitemap-and-why-do-you-need-one/>

Educative. (u.å.). C# naming conventions. Hentet fra <https://www.educative.io/answers/c-sharp-naming-conventions>

Flinn, F. (u.å.). Figma Prototype: What is it and why use it for design? Hentet fra <https://designproject.io/blog/figma-prototype#:~:text=You%20can%20use%20it%20to,save%20you%20time%20and%20stress.>

Gathoni, M. (2022, 13. september). 4 Testing methods every developer should know. Hentet fra: <https://www.makeuseof.com/testing-methods-developers-should-know/>

Gautam, V. (u.å.). The Difference between Early Stage and Late Stage User Testing. Hentet fra <https://www.userexperior.com/blog/the-difference-between-early-stage-and-late-stage-user-testing>

Gorný, D. (2020). Testing: An important part of the development process. Hentet fra <https://medium.com/@damian.gorny/testing-an-important-part-of-the-development-process-eebff3fd7109>

Goswami, A. (2017). What, Why and How About SignalR. Hentet fra <https://www.c-sharpcorner.com/UploadFile/abhijmk/what-why-and-how-about-signalr/>

Hasan, D. (2022). Hva er brukertesting? Hentet fra <https://www.netlife.com/blogg/hva-er-brukertesting>

Interaction Design Foundation. (u.å.). User Centered Design. Hentet fra [https://www.interaction-design.org/literature/topics/user-centered-design#:~:text=User-centered%20design%20\(UCD\),and%20accessible%20products%20for%20them](https://www.interaction-design.org/literature/topics/user-centered-design#:~:text=User-centered%20design%20(UCD),and%20accessible%20products%20for%20them)

ISTQB Glossary. (u.å.). User-Based Quality. Hentet fra <https://istqb-glossary.page/user-based-quality/#:~:text=A%20view%20of%20quality%2C%20wherein,unlikely%20to%20find%20a%20ny%20users.>

Karlsen, J.T. (2018). Prosjektledelse (4.utg.). Oslo: Universitetsforlaget.

KnowFife. (2018). Semi-Structured Interviews. Hentet fra https://know.fife.scot/_data/assets/pdf_file/0028/177607/KnowHow-Semistructured-interviews.pdf

Lucidchart. (u.å.). What is an Entity Relationship Diagram (ERD)? Hentet fra <https://www.lucidchart.com/pages/er-diagrams>

Microsoft. (2022). What is Azure Repos? Hentet fra <https://learn.microsoft.com/en-us/azure/devops/repos/get-started/what-is-repos?view=azure-devops>

Microsoft Learn. (2022). What is Azure DevOps? Hentet fra <https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>

Mountain Goat Software. (u.å.). Planning Poker. Hentet fra <https://www.mountangoatsoftware.com/agile/planning-poker>

Næringsforeningen. (2022). Capgemini etablerer seg på Sørlandet. Hentet fra <https://www.nikr.no/aktuelt-nyheter/capgemini-etablerer-seg-pa-sorlandet>

O'Connor, J. (2016). Prioritizing Your User Stories with the MoSCoW Method. Hentet fra <https://sax1johno.medium.com/prioritizing-your-user-stories-with-the-moscow-method-8bf42d427da6>

ProductPlan (u.å.). Daily Scrum. Hentet fra <https://productplan.com/glossary/daily-scrum/>

Radigan, D. (u.å.). Story points and estimation. Hentet fra <https://www.atlassian.com/agile/project-management/estimation>

Rahman, M. T. (2019). Digital Interaction Design IS-104. Hentet fra <https://useg.it/t/it-og-informasjonssystemer/raw/branch/master/IS-104/2019-Autumn/viewer/files/Lectures/Lecture%206%20-%20Design%20Principles%20for%20usability.pdf>

Rehkopf, M. (u.å.). User stories with examples and a template. Hentet fra <https://www.atlassian.com/agile/project-management/user-stories#:~:text=software%20user's%20perspective,-.A%20user%20story%20is%20an%20informal%2C%20general%20explanation%20of%20a,value%20back%20to%20the%20customer.>

SafetyCulture. (2023a). *5x5 Risk Matrix Sample* [Figur]. Hentet fra <https://safetyculture.com/topics/risk-assessment/5x5-risk-matrix/>

SafetyCulture. (2023b). A Guide to Understanding 5x5 Risk Matrix. Hentet fra <https://safetyculture.com/topics/risk-assessment/5x5-risk-matrix/>

Sainty, C. (2019). What is Blazor and why is it so exciting? Hentet fra <https://chrissainty.com/what-is-blazor-and-why-is-it-so-exciting/>

Salian, V. (2020). Product quality vs Code quality. Hentet fra <https://vijeshsalian.medium.com/product-quality-vs-code-quality-3c7a7a662656>

Schwaber, K. & Sutherland, J. (2020). The 2020 Scrum Guide™. Hentet fra <https://scrumguides.org/scrum-guide.html>

Scrum. (u.å.a). What is a Product Backlog? Hentet fra <https://www.scrum.org/resources/what-is-a-product-backlog/>

Scrum (u.å.b). What is a Sprint Backlog? Hentet fra <https://www.scrum.org/resources/what-is-a-sprint-backlog/>

Scrum (u.å.c). What is a Sprint Retrospective? Hentet fra <https://www.scrum.org/resources/what-is-a-sprint-retrospective>

Scrum (u.å.d). What is a Sprint Review? Hentet fra <https://www.scrum.org/resources/what-is-a-sprint-review>

Scrum. (u.å.e). What is Scrum? Hentet fra <https://www.scrum.org/resources/what-scrum-module>

Scrum. (u.å.f). What is Sprint Planning? Hentet fra <https://www.scrum.org/resources/what-is-sprint-planning/>

Tufts, K. (2014). Why sketching is an important part of the design process. Hentet fra <https://www.dnnsoftware.com/blog/why-sketching-is-an-important-part-of-the-design-process>

Usability. (u.å.). Prototyping. Hentet fra <https://www.usability.gov/how-to-and-tools/methods/prototyping.html>

UXPin. (2021). What Is a Component Library, and Why Should You Use One for UI Development? Hentet fra <https://www.uxpin.com/studio/blog/ui-component-library/>

Wallen, J. (2021). What is a scrum, and why should your teams use it? Hentet fra <https://www.techrepublic.com/article/what-is-a-scrum-and-why-should-your-teams-use-them/>

Wanjala, A. (2023, 10.januar). What is linting and why is it important for your programming projects? Hentet fra: <https://www.makeuseof.com/what-is-linting/>

Wrike. (u.å.a). What Is Agile Methodology in Project Management? Hentet fra <https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/>

Wrike. (u.å.b). Daily Scrum Meetings. Hentet fra www.wrike.com/scrum-guide/daily-scrum/

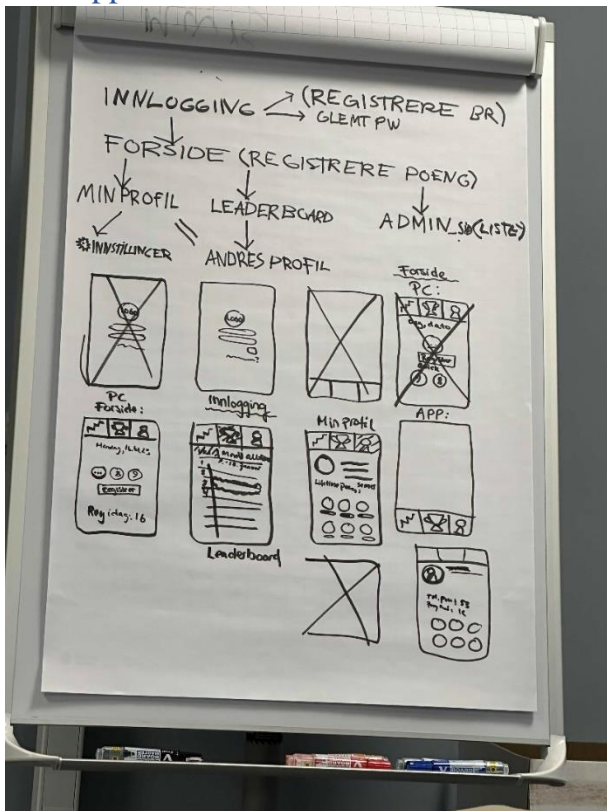
ZZZ Projects. (2023, 8. januar). *Welcome to learn dapper*. Hentet fra <https://www.learndapper.com/>

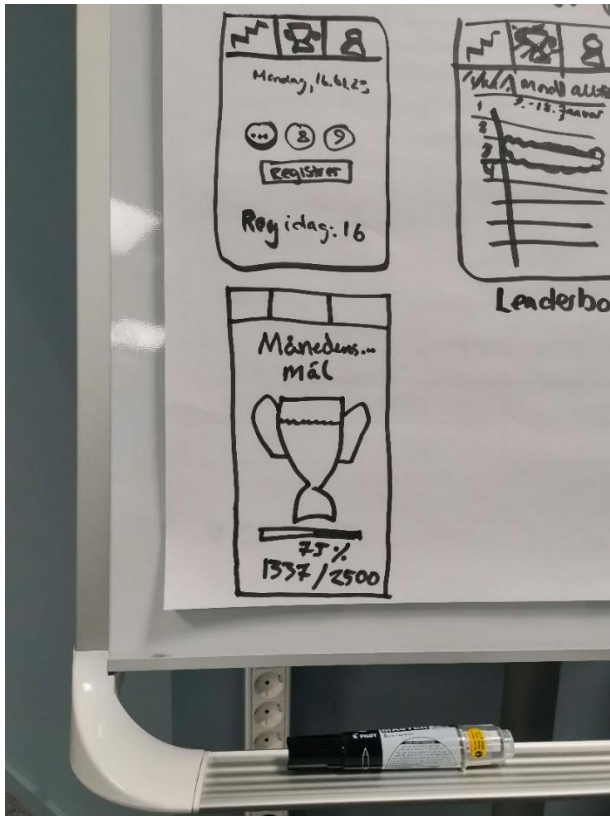
9. Appendiks

9.1. Appendiks 1 – Risikoanalyse

Nr.	Hva kan gå galt?	Sannsynlighet (1-5)	Konsekvens (1-5)	Totalt risiko-nivå	Hvordan unngå risiko	Hvordan håndtere hendt risiko
1	Forsentkommelse	4	1	4	Gi straff til de som kommer for sent	Ring/starthe uten de
2	Mild sykdom	2	3	6	Spise sunt, sove nok.	Hjemmekontor
3	Generell sykdom	2	5	10	Spise sunt, sove nok.	Være ekstra flinke på møtereferat og ellers fortsette uten syk person
4	Mac kompatibilitet	4	3	12	Små bokstaver i database	Google/finne en løsning
5	Mangel/feil på utstyr	2	4	8	Alltid sjekk at du har med alt. Ta vare på utstyr.	Hente manglende utstyr/kjøpe nytt.
6	Miste data	1	5	5	Autolagring og versjonskontroll	Se etter backup, evt starte på nytt
7	Manglende motivasjon	2	2	4	Holde lagånd oppe, spille hverandre gode, motivere	Snakke med personen, komme til bunns i problemer
8	Manglende kompetanse	4	2	8	Gi oppgaven til en som har kompetanse	Spør om hjelp eller google
9	Manglende oppfølging fra bedriften	2	4	8	Minne de på at vi trenger noe	Prøve å gjøre andre ting i mens vi venter
11	Manglende kommunikasjon	2	2	4	Skrive møtereferat/daily scrum	Snakke om misforståelsen og fikse opp i det
12	Internettproblemer	1	5	5	Ute av vår kontroll	Gjøre andre oppgaver som ikke innebærer internett. Ta en pause. Bytte lokasjon.
13	Lover knyttet til personlig data	2	2	4	Ha med så lite personlig data som mulig	Fjerne/gjemme/beskytte sensitiv data
14	Feilestimering	4	3	12	Heller underestimere enn overestimere sprinter	Fullføre oppgavene i neste sprint
16	Lav kodelstandard/dokumentasjon	3	5	15	Skrive god dokumentasjon og bli enige om kodelstandarder	Legge til dokumentasjon, endre hele koden til en akseptabel kodelstandard.
17	Generelt fravær	4	4	16	Ikke avtale møter dager man skal ha møte	Si ifra i god tid

9.2. Appendiks 2 – Skisser





9.3. Appendiks 3 – Oppsummering av brukerundersøkelse i Sprint 1

Spørsmål:

1. Hvor lenge har du jobbet her?

- A. 1. Desember 2021
- B. 1. November 2022
- C. 1.oktober 2022 (jobba i Capgemini i Trondheim siden des 2019).
- D. 12 år, kom hit september 2022.
- E. 6 mnd har jobba i en annen startup

Oppsummering: De fleste har jobbet i Capgemini Kristiansand i under ett år.

2. Hvor ofte er du på kontoret i uka?

- A. 3 dager i uka.
- B. 4-5 dager i uka.
- C. Stort sett hver dag. 5 dager i uka.
- D. Hver dag, 5 dager.

E. 2-3 dager

Oppsummering: Vi ser at det lager seg en gruppering på folk som går 2-3 dager i uka og folk som går 4-5 dager.

3. Hvor gammel er du?

A. 31 år gammel

B. 38 år gammel

C. 29 år.

D. 34 år.

E. 25 år

Oppsummering: Gjennomsnittlig alder er 31,4

4. Hvor aktiv er du i trappek konkurransen?

A. En gang om dagen.

B. Mest aktiv/veldig aktiv (to ganger om dagen).

C. Prøver å gå trappene hver gang 1-2 ganger daglig. Greit med poeng bare opp.

D. Ikke veldig aktiv, middels. Prøver å gå 2-3 ganger uken.

E. Jeg vant ikke, men jeg prøver å ta de et par ganger når jeg er på kontoret

Oppsummering: De fleste går 1-2 ganger om dagen (når de er på kontoret).

4a. Om aktiv nå: Hva vil du si er det som motiverer deg til å delta aktivt?

A. Ikke trengte å trene, føler at hvis jeg går ned trappen en gang om dagen gjelder det som trening. Har ikke konkurranse instinkt, gjør det for egen helse.

B. Ikke bli en feit slask. Skulle gjerne hatt timer på det. Men jeg skal ikke gi dere noen idéer.

C. Syntes det at man for å vite hvordan en ligger an, det er en ekstra bonus at det spiller på konkurranseinstinkt.

D. Noe aktivitet er bedre enn ingenting. Alt hjelper.

E. 2 ganger hver gang jeg er på kontoret. Jeg sitter så mye stille ellers at det er godt å bevege seg litt.

Oppsummering: Majoriteten finner motivasjon i at det er bra for helsen å ta trappene.

4aa. Hvilken plass pleier du å komme på i trappekonsurransen?

- A. Kommer langt ned, er for lite på kontoret
- B. Jeg har ikke vært med mens det var offisiell plassering. Men kom kanskje på nummer to tror jeg?
- C. Vet ikke, i starten var jeg der ikke så mye.
- E. Jeg vet ikke, men jeg vinner aldri.
- D. Ikke aktiv.

Oppsummering: De fleste har ikke styr på hvilken plass de kommer på.

4b. Om ikke aktiv lenger: Hva er det som gjør at du i dag ikke har deltatt like aktivt som før? Har du noen tanker om hva som skal til for å motivere deg til å delta mer?

4c. Om aldri vært /veldig lite aktiv: Har du noen tanker om hva som ville fått deg interessert i å delta i konkurransen.

- D. Felles mål å gå etter.

Oppsummering: En person hadde vært motivert av et felles mål for å bli aktiv.

6. Hvis vi hadde implementert en mariokart-modus, hadde du brukt den?

- A. Ja, må være en måte å utligne på.
- B. Ja, finne et mobbeoffer på kontoret og alltid skyte på de
- C. C. Tror kanskje ikke det.
- D. Tror for min del at standard modus er mest fristende.
- E. Jeg liker ikke at folk kan sabotere meg, er greit når jeg kan sabotere andre da. Jeg foretrekker at det poengsummen reflekterer egenskap og innsats.

Oppsummering: Litt over halvparten hadde ikke brukt mariokartmodus.

8. Har du noen idéer til features i applikasjonen du ville hatt implementert?

- A. Tenker det må være enkelt, ikke alltid jeg gidder å skrive på tavla "Kan ikke vinne noe som helst sånn som det er nå". NFC-tagg ville vært lett, fast-knapp for 8 og 9. "Så enkelt som mulig". Har lyst til å angripe spesifikke personer.
- B. Hvis du hadde hatt bluetooth beacons.. Tidtakning. tid det tar å gå etasjene. Samle data og så kan du gjøre noe gøy med dataen.. Han går saktere på toppen
- C. Inspirasjon fra tidligere konkurranse. Lodd for antall trapper en hadde gått, på slutten var det loddtrekning. I starten var det en som gikk ekstremt mye, en måte å

involvere flere til tross for at ikke alle er “aktive”. Folk følte at de måtte fordi det var kollektivt; altså totalt antall trapper for alle samlet.

D. Så enkelt som mulig.

E. Topplister kan se alle, Automatisk trappegjenkjenning så du slapp å logge, statistikk sånn som på den helseappen.

Oppsummering: De to temaene som går igjen på features er statistikk og lettvinthet.

9. Hvilke features i applikasjonen vil du helst IKKE ha implementert?

A. – (kom ikke på)

D. Egentlig ikke.

B. Kanskje noen ikke vil vise poengene sine hvis de har veldig lite poeng?

C. Ingenting merkverdig.

D. - (kom ikke på)

E. mariokartmodus, masse varslinger vil jeg ikke ha

Oppsummering: Det er bare noen få som hadde et forslag til features de ikke ville ha.

10. Hvordan ser du for deg at det er mest praktisk å registrere trappetrinn? (Fks fast maskin i gangen, webapplikasjon på mobil/PC, Standalone applikasjon som må lastes ned)

A. NFC klistremerker på -1 og 1, så opp til toppen for å registrere. Ville vært mulig å sammenligne tid da også.

B. Noe kontaktløst, eller en touchskjerm. To trykk på en tablet med bilde av oss på.

C. Tenker at det kan være greit å ha en fast maskin i gangen, for å gjøre det lettere å huske.

D. Enten QR/NFC, app, jo mer automatisk jo bedre. Så enkelt som mulig.

E. Enten at det er første sida etter at du har registrert appen. Widget på telefonen. Default 8/9 trapper. Geolokasjon trapp track. Krever app på telefonen mest sansynlig.

Oppsummering: De vil enten ha automatisk registrering eller minimal klikk på mobil/felles enhet.

9.4. Appendiks 4 – Brukerundersøkelse mal, prototype v1

Person X

Takke personen for at de kunne komme. Vi har delt undersøkelsen inn i to deler: Først skal du bla litt fritt i appen, etter dette har vi noen spesifikke oppgaver vi ønsker at du fullfører. Vi kommer også til å stille noen spørsmål underveis. Det er viktig at du er ærlig, vi setter veldig pris på konstruktiv kritikk.

Førsteintrykk

Hva er din første reaksjon om appen? Snakk høyt om hva du tenker når du klikker rundt.

Hvordan er appen sammenliknet med forventningene dine?

Har du noen spørsmål akkurat nå?

Hvorfor ville noen ha brukt denne appen?

Hvordan kan appen hjelpe deg?

Hva er det første du ville gjort når du fikk appen?

Oppgavefokuset

1. Hvis du skulle ha sjekket ut en annen bruker sin profil, hva ville du ha gjort?
2. Hvordan kunne vi ha presentert denne informasjonen på en mer meningsfull måte?
3. Hva var det vanskeligste med denne oppgaven?

1b. Kan du finne oversikt over milepæler?

2b. Hvordan kunne vi ha presentert denne informasjonen på en mer meningsfull måte?

3b. Hva var det vanskeligste med denne oppgaven?

1c. Finn ut hvor mange poeng du mangler for å få fullført en medalje.

2c. Hvordan kunne vi ha presentert denne informasjonen på en mer meningsfull måte?

3c. Hva var det vanskeligste med denne oppgaven?

1b. Hvordan ville du gått frem for å registrere trapper du akkurat har gått?

2b. Hvordan kunne vi ha presentert denne informasjonen på en mer meningsfull måte?

3b. Hva var det vanskeligste med denne oppgaven?

4. Er det noe du ville ha forandret/lagt til/fjernet for å gjøre det bedre for deg?

5. Var det noe overraskende eller uventet?
6. Fra en skala på 1-5 hva ville du ratet appen, og hvorfor? Nevn minst 2 ting som trekker ned, og 2 ting som trekker opp.

Oppsummering

Hadde du brukt appen?

Føles appen ut som om den var designet for deg?

Er det noe som mangler?

Hvilke adjektiv ville du brukt for å beskrive appen?

Takk for at du kom!

9.5. Appendiks 5 – MoSCoW

Must have
Som en bruker ønsker jeg å kunne registrere antall trapper jeg har gått.
Som en bruker ønsker jeg å få en oversikt over ukentlige, månedlige, samt alltime rangstiger for alle brukerne med navn og tilhørende poengsum.
Som bruker ønsker jeg å kunne opprette en egen konto slik at jeg kan delta i konkurransen.
Legg inn automatisk oppdatert brukerhistorie her
Som en bruker, ønsker jeg å kunne gjenopprette nytt passord via email.
Som en bruker, ønsker jeg at det er en automatisk overgang til ny poengtelling for hver ny måned.
Som en bruker ønsker jeg å kunne registrer antall etasjer jeg har gått på lettest mulig måte slik at jeg ikke trenger å bruke mye tid på det.
Som bruker ønsker jeg å kunne se min og andres plassering i konkurransen slik at jeg vet hvordan andre gjør det eller inspirer til å være mer aktiv selv.

Should have
Som en bruker, ønsker jeg å få en oversikt over hvem av mine medarbeidere som fikk 1-3. plass forrige måned, gjerne som en pop-up.
Som en bruker, ønsker jeg å se statistikken til andre brukere.
Som en bruker, ønsker jeg å oppnå medaljer ettersom jeg utfører forskjellige oppgaver som er definert i appen.
Som en bruker ønsker jeg å få en funfact når jeg har oppnådd en milepæl.

Som bruker ønsker jeg å kunne redigere og slette min egen konto.
Som en administrator ønsker jeg å administrere brukere slik at brukere og brukerinformasjon holdes oppdatert.
Som en bruker ønsker jeg å se en logg over tidligere vinnere og poengsummer slik at jeg kan se hvordan jeg eller andre brukere har gjort det tidligere.
Som en bruker ønsker jeg å se at jeg gjør det godt ved at bakgrunnsfargen endres slik at jeg kan skryte til andre.
Som en bruker ønsker jeg at antall trapper gått for alle som er med blir lagt sammen og blir ført i en felles statistikk slik at vi kan jobbe mot et felles mål

Could have
Som en bruker ønsker jeg at applikasjonen har en Mariokart liknende modus, slik at jeg kan ha det moro med kollegaer, og har sjansen til å komme meg opp på leaderboardet ved å tenke taktisk.
Som bruker, ønsker jeg å kunne se statistikk over en egendefinert tidsperiode, slik at jeg kan sjekke for eksempel hvor mye poeng jeg har fått når en halv måned har gått, og ikke må forholde meg til de satte poengvisningene.
Som en bruker ønsker jeg å få pushvarsler når noen går forbi meg på leaderboardet, slik at jeg kan gripe sjansen og registrere noen ekstra etasjer for å holde meg på toppen.
Som bruker, ønsker jeg å kunne laste opp mitt eget profilbilde, slik at profilen min blir mer unik.
Som en bruker ønsker jeg en egen konkurranse for de trappene som er inne på Capgemini kontoret, slik at jeg lettere kan registrere noen trapper i pausen.
Som bruker vil jeg kunne ha en avatar som jeg kan tilpasse, slik at jeg føler at profilen min er mer unik.
Som en administrator ønsker jeg å kunne opprette felles mål rett fra applikasjonen, slik at jeg slipper å bruke så mye tid på det.
Som en bruker ønsker jeg å kunne opprette en 1v1 med andre brukere, slik at vi kan ha en intern konkurranse om å gå flest trapper.
Som en bruker ønsker jeg å kunne opprette en "fjelltur" som blir fullført når jeg har registrert nok trapper, slik at jeg får motivasjon og inspirasjon til å gå i trappene.
Som en bruker ønsker jeg å kunne velge ut personer jeg vet kommer til å gå mye i trappene denne perioden slik at jeg kan oppnå poeng selv om jeg ikke har anledning til å gå i trappene selv.
Som en bruker ønsker jeg at trapper jeg går blir registrert automatisk av nfc-tags, slik at jeg sparer litt tid.

Won't have

Som en bruker ønsker jeg at det finnes ulike anti-cheating funksjoner i appen, slik at jeg stoler mer på resultatene i konkurransen.

Som en bruker ønsker jeg å kunne spesifisere i profilen min alder, kjønn, telefonnummer, email og sivilstatus, slik at de andre brukerne vet mer om meg.

Som bruker ønsker jeg at applikasjonen skal kunne ta tiden på hvor lang tid jeg bruker opp trappene, slik at jeg kan få en statistikk og se om jeg har forbedret meg siden sist.

9.6. Appendiks 6 – Risikologg

nummer	risikonumme	Risikotype	Startdato	Sluttdato	Kommentar
1	5	Mangel/feil på utstyr	19/01/2023	19/01/2023	Eirik glemte laderen til PCen. Måtte reise hjem og hente
2	5	Mangel/feil på utstyr	26/01/2023	27/01/2023	Anna hadde ikke satt opp MFA på den nye telefonen. Måtte bruke gjestebroker.
3	9	Manglende oppfølging fra bedriften	27/01/2023	27/01/2023	Fikk ikke devops, har satt opp en devops selv
4	9	Manglende oppfølging fra bedriften	30/01/2023	2/2/2023	Vegard ble syk, fikk ikke hatt oppstartsmøte
5	4	Mac kompatibilitet	9/2/2023	9/2/2023	Problemer med å få prosjektet til å fungere på Anna sin macbook
6	4	Mac kompatibilitet	20/02/2023	vedvarende	Anna sin mac har ikke hotrefresh, så når ho skal kode må ho rebuilde for å se endringer
7	5	Mangel/feil på utstyr	27/02/2023	27/02/2023	Håndtaket på døra til det faste møterommet vårt var gått tom for strøm, så vi kom oss ikke inn. Vi måtte sette oss i et annet rom, og så må vi forlate rommet litt senere i dag når det er booket av noen andre.
8	17	generelt fravær	20/03/2023	24/03/2023	Andreas måtte tjenestegjøre i Heimevernet. Han tok igjen arbeidet ved å jobbe det inn igjen i helgene.
9	2	Mild sykdom	12/4/2023	13/04/2023	Andreas ble smittet av en forkjølelse da han var hjemme på ferie. Det var en usikkerhet om han ville være fysisk på Capgemini torsdag 4/13, men følte seg pigg nok til å likevel stille opp
10	5	Mangel/feil på utstyr	14/04/2023	14/04/2023	Andreas sølte kaffe på veggen som rant ned i strømsokkelen. Dette kortslettet strømmen og hovedsikringen gikk. Dette ble løst ved åpne sokkelen og la den tørke fram til klokka 12

9.7. Appendiks 7 – Interessentanalyse og Kommunikasjonskart

Interessentanalyse			
Interessent(rolle)	Navn	Behov som skal ivaretas	
Sentrale interessenter			
Produkteier	Vegard S.	Ha kontroll på planer/fremdrift i prosjekt	
Capgemini, (mentor)	Selim, Andreas H	Informere om problemer som behøver oppfølging	
Sluttbruker	Capgemini Ansatte	Generelle ønsker, og interesse for bruk	
UiA Veileder	Sofie	Status og progresjon i prosjektet, men ingen direkte interesse for bruk	
UiA Emneansvarlige	Hallgeir og Geir Inge	Vite at vi jobber med prosjektet og har en viss fremdrift	
Capgemini, leder	Øyvind	Informere, hvis interesse og tid	
Kommunikasjon			
Kommunikasjonsbehov	Kanal og form	Frekvens	Ansvarlig
Stort behov	Teams, møter, mail: muntlig/skriftlig	Daglig	Gruppe 9
Lite behov	muntlig, teams, mail	ad hoc	Gruppe 9 og Vegard
Generelt behov	muntlig	ad hoc	Gruppe 9
Stort behov	Møter/ Styresmøter / Mail	Bi ukentlig	Gruppe 9
Generelt behov	Presentasjoner / Canvas	Månedlig	Gruppe 9
Lite behov	Mail	Ukentlig	Vegard

9.8. Appendiks 8 - Demo av applikasjon

Her er det vedlagt en link til en kort video gjennomgang av applikasjonen.

https://1drv.ms/u/s!AsI4Twsz_ancgdhTWKRA1JLT59t5qw?e=9j5DUQ