

PRO203

SMIDIG PROSJEKT

PROSJEKTRAPPORT

Høyskolen Kristiania



Gruppenummer: 58

Gruppemedlemmer & kandidatNr.:

Ayanle Mohamed - 549

Barkhad Isse - 486

Mikkel Villard - 460

Christoffer Heien - 32

Tobias Kløfta - 197

Kristian Melgård - 628

Johannes Evensen – 293

1.0 Innledning	3
1.1 Hva var oppdraget?	3
1.2 Visjon?	3
1.3 Hva skulle vi oppnå?	4
1.4 Hvem er målgruppen?	4
1.5 Andre tanker?	4
2.0 Iterasjonene	5
2.1 Design Sprint.....	5
2.2 Scrum sprintene	6
2.2.1 Sprint 1	6
2.2.2 Sprint 2	6
3.0 Prosess	7
3.1 Hvordan har vi brukt Scrum?	7
3.2 Hvordan har bruken endret seg?	7
3.3 Hvilke valg har vi gjort?.....	8
3.4 Har vi avviket fra anbefalingene?	8
3.5 Refleksjoner om prosessen	9
3.6 Referanser til vedlegg.....	9
4.0 Veien videre	9
5.0 Kilder.....	11
6.0 Vedlegg.....	11
Vedlegg 1 - Link til Mirobrett	11
Vedlegg 2 - Sprint 1 produktbacklog.....	12
Vedlegg 3 - Sprint 1 burndowngraf	12
Vedlegg 4 - Sprint 1 retrospective.....	13
Vedlegg 5 - Sprint 2 produktbacklog.....	13
Vedlegg 6 - Sprint 2 burndowngraf	15
Vedlegg 7 - Sprint 2 retrospective.....	16

1.0 Innledning

1.1 Hva var oppdraget?

Da vi startet opp prosjektet fikk vi presentert en problemstilling fra Frostbyte: dagens rutine for inn- og utsjekk i barnehagen foregikk i et delt Excel-ark. Alle med lenken kunne se og endre alt, noe som både skapte forvirring og var helt uforenlig med krav til personvern. Vår oppgave var å erstatte dette med en digital løsning der foresatte enkelt kan registrere barna sine inn og ut og ansatte får en tydelig oversikt, samtidig som dataene behandles sikkert. Vi skulle definere behovene, designe en prototype, bygge en første versjon av både frontend og backend, og dokumentere prosessen gjennom smidige sprint-iterasjoner.

1.2 Visjon?

Visjonen for løsningen har vært å utvikle et system som er enkelt, intuitivt og tilgjengelig for alle brukere, uavhengig av alder og teknisk erfaring. Brukerne skal raskt forstå hvordan systemet fungerer, uten behov for opplæring eller forklaringer, og kunne utføre sentrale handlinger som innlogging, innsjekk/utsjekk og meldinger på en trygg og effektiv måte.

Samtidig har det vært viktig at løsningen håndterer sensitive data på en sikker og ansvarlig måte. Systemet er derfor tenkt bygget rundt rollebasert tilgang, hvor foreldre kun får tilgang til informasjon om egne barn, og ansatte kun ser barn og data som er relevante for deres avdeling og rolle. Personvern og datasikkerhet har vært sentrale hensyn i utformingen av både frontend og backend.

I tillegg har vi hatt et mål om å lage en fleksibel og skalerbar løsning. Arkitekturen og strukturen skal gjøre det mulig å videreutvikle systemet med nye funksjoner og utvidelser over tid, uten at det går på bekostning av brukervennlighet eller sikkerhet. Visjonen har dermed vært å balansere enkelhet i brukeropplevelsen med robuste tekniske valg i bunn.

1.3 Hva skulle vi oppnå?

I løpet av prosjektet satte vi oss tre hovedmål:

1. Lage en fungerende MVP hvor foresatte kan sjekke inn og ut egne barn, se status og sende beskjer, og hvor ansatte kan krysse barna inn og ut og skrive interne notater.
2. Etablere en robust produkt- og sprint-backlog i Scrumwise, med tydelige epics, PBIs og tasks, slik at vi jobber strukturert og kan følge progresjonen med burn-down-grafer.
3. Sikre at vi følger opp faglige krav som GDPR og rollebasert tilgang, og dokumenterer hele prosessen slik at en sensor kan se hvordan vi har resonert og hva vi har lært.

1.4 Hvem er målgruppen?

Primærmålgruppen er foresatte som leverer og henter barn i barnehagen, og ansatte som jobber i avdelingene. Løsningen skal gjøre hverdagen enklere for dem. Foresatte skal kunne registrere barna uten å stå i kø eller ringe, og ansatte skal kunne følge med på alle barna i avdelingen sin med ett tastetrykk. Sekundært er også barnehageledelsen en målgruppe, ettersom de trenger et administrativt verktøy for å sette opp brukere og roller.

1.5 Andre tanker?

Underveis i arbeidet ble det tydelig at prosjektet handlet om mer enn å utvikle en teknisk løsning. Det dreide seg også om å bygge et team som klarer å jobbe smidig sammen. Allerede i design-uka erfarte vi hvor krevende det kan være å samle alle samtidig og holde tempo i en stram tidsplan. Samtidig ble det klart hvor effektivt samarbeidet blir når vi møtes fysisk og kan fokusere på en oppgave om gangen.

I Sprint 1 valgte vi bevisst å prioritere teori, planlegging og datamodellering fremfor å gå rett i koding. Dette ga oss en felles forståelse av problemstillingen og et solid fundament å bygge videre på. I Sprint 2 merket vi tydelig hvordan arbeidsflyten forbedret seg når vi møtte fysisk, fordelte ansvar tydeligere og jobbet med mer presise tasks. Disse erfaringene danner bakteppet for resten av rapporten. Vi har lært like mye av prosessen som av det ferdige resultatet, og dette gjenspeiles i hvordan vi beskriver oppdraget, visjonen og målene våre.

2.0 Iterasjonene

2.1 Design Sprint

Helt i starten av prosjektet gjennomførte vi en design-sprint for å få en grunnleggende forståelse av caset og avklare hvilken retning vi skulle ta. Sprinten var strukturert som en Google Design Sprint og foregikk hovedsakelig i Miro. Der organiserte vi ideer, sorterte dem etter relevans og stemte over ulike løsninger før vi tok en felles beslutning om hva vi skulle gå videre med. Vi laget enkle skisser i Miro som senere ble overført til Figma, hvor vi bygde en mer gjennomarbeidet prototype. Denne prosessen gav oss flere viktige gevinster:

Felles retning og prioritering: Å jobbe sammen i Miro hjalp oss å forstå utfordringene i caset og strukturere arbeidet. Det ble tydelig at kjernefunksjonen skulle være enkel inn- og utkryssing av barn, og vi bestemte oss for å prioritere dette fremfor å legge til ekstra funksjoner som kalender eller værmelding. Dette var i tråd med visjonen, som var å lage en så simpel app som mulig.

Visualisering av løsningene: Skisser og prototyper i Figma ble brukt som et sentralt verktøy for å visualisere hvordan brukerflyt, funksjonalitet og samspillet mellom de ulike rollene i systemet skulle fungere. Gjennom prototypene fikk vi tidlig testet og diskutert navigasjon, skjermoppsett og rekkefølgen på brukerhandlinger, noe som gjorde det lettere å avdekke uklarheter før videre utvikling.

Vi tok i bruk Figma sine AI-verktøy for å generere flere alternative designutkast på kort tid. Dette ga oss et bredere utgangspunkt for diskusjon og gjorde det enklere å sammenligne ulike løsninger. Etter flere runder med justeringer og felles vurderinger landet gruppen på et design som alle kunne enes om, og som balanserte brukervennlighet, enkelhet og funksjonelle behov. Prototypen fungerte dermed både som et kommunikasjonsverktøy internt i gruppen og som et tydelig grunnlag for videre teknisk arbeid.

Brukerundersøkelse: Vi lot potensielle brukere teste prototypen vår. Undersøkelsen ga dem mulighet til å prøve funksjonaliteten og vurdere om informasjonen var tydelig.

Tilbakemeldingene ga oss innsikt i hvordan løsningen ble oppfattet og avdekket små ting vi måtte justere. Resultatene fra undersøkelsen ble brukt aktivt i videre design og gjorde at vi satt igjen med en testbar prototype og en tydelig retning for utviklingen.

Læringspunkter: Sprinten gjorde det enklere å strukturere arbeidet videre. Den viste oss også at vi jobbet mer effektivt når vi møttes fysisk og at tidsrammen for en design-sprint kan være for knapp når teamet er spredt. Noen oppgaver måtte tas to ganger fordi ikke alle var til stede, og det var utfordrende å få en Figma-prototype som fungerte godt på mobil uten at brukerne måtte logge inn. Til slutt fant vi løsninger på dette, men det tok tid.

Vi vil i sluttproduktet inkludere skjermbilder fra Miro-brettet, skissene og Figma-prototypen som illustrerer både idemyldringen og det visuelle uttrykket vi landet på.

2.2 Scrum sprintene

2.2.1 Sprint 1

Planlegging og grunnmur: Første sprint ble brukt til å bygge et solid fundament. Vi fokuserte på å sette opp produkt- og sprint-backloger i Scrumwise, dele arbeidet inn i epics, PBIs og tasks, og modellere databasen. Vi diskuterte teknologivalg og rollefordeling, og brukte tid på å forstå GDPR-krav og RBAC-modell. Arbeidet var hovedsakelig teoretisk; vi valgte å la koding vente til vi hadde en felles forståelse og en gjennomtenkt plan.

2.2.2 Sprint 2

Implementasjon og forbedring:

I den andre sprinten gikk vi fra planlegging og teori til mer praktisk implementasjon. Vi møttes fysisk hver dag, noe som ga bedre arbeidsflyt, raskere avklaringer og et tettere samarbeid i gruppen. Frontend-teamet startet med å kode skjermene basert på Figma-prototypen, mens backend-teamet jobbet med oppsett av database og utvikling av API-endepunkter.

Underveis koblet vi sammen flere sentrale funksjoner, som inn- og utsjekk av barn, statusvisning og grunnleggende meldingsflyt, og testet disse i en enkel webserver. Samtidig ble det tydelig at Sprint 2 var mer omfattende enn først antatt. Flere av kravene vi hadde definert viste seg å være mer tidkrevende å implementere enn forventet, særlig når frontend og backend skulle fungere sømløst sammen.

Selv om vi ikke rakk å fullføre alle kravene fullt ut, opplevde vi stor progresjon i hvordan systemet hang sammen teknisk. Vi ble også flinkere til å bryte ned oppgaver i mindre tasks

og jobbe mer strukturert, og vi erfarte at arbeidet som ble lagt ned i Sprint 1 med planlegging og datamodellering gjorde Sprint 2 betydelig mer effektiv. Erfaringene fra denne sprinten ga oss samtidig et tydelig bilde av hva som gjenstår og hvilke deler som må prioriteres videre.

3.0 Prosess

3.1 Hvordan har vi brukt Scrum?

Hver sprint startet med et planleggingsmøte der vi gikk gjennom produkt-backloggen i Scrumwise og plukket ut de epics og PBIs vi skulle jobbe med. Deretter brøt vi PBIs ned i konkrete tasks og estimerte med story points. Vi gjennomførte daglige stand-ups (enten i Teams eller fysisk) hvor vi fortalte hva vi hadde gjort, hva vi skulle gjøre, og om vi hadde hindringer. Vi avsluttet hver sprint med en review der vi demonstrerte det vi hadde fått til, og en retrospektiv der vi diskuterte hva som fungerte og hva vi skulle forbedre. I løpet av sprintene logget vi også tidsbruk og oppdaterte burn-down-grafen. GitHub ble brukt for versjonskontroll; hver utvikler hadde sin egen gren og vi gjorde pull requests til main når vi hadde ferdige funksjoner.

3.2 Hvordan har bruken endret seg?

I Sprint 1 brukte vi mye tid på å bli kjent med verktøyene og etablere en fungerende struktur. Scrumwise fremstod som relativt enkelt i teorien, men det tok tid før vi fant en god måte å organisere epics, PBIs, tasks og story points på. Tidsloggingen var også noe ujevn i denne sprinten, og flere førte timer samlet mot slutten av uka i stedet for fortløpende. Dette gjorde at burn-down-grafen i liten grad reflekterte faktisk fremdrift.

I Sprint 2 tok vi lærdom av disse erfaringene. Tasks ble definert mer konkret og presist, og vi skrev dem slik at de i praksis beskrev hva som måtte være gjort for at en PBI kunne regnes som ferdig. Vi møttes fysisk hver dag, logget timer jevnligere og brukte morgenmøtene til å gå gjennom arbeidet fra dagen før for å sikre at timeføringen stemte. Vi ble også mer konsekvente med å flytte tasks til *Done* når de faktisk var ferdige, selv om vi ser at dette fortsatt er et område med forbedringspotensial.

Burn-down-grafer ble fortsatt brukt i Sprint 2, men mer som et visuelt støtteverktøy enn som hovedstyring. I praksis var det task-boardet i Scrumwise som fungerte som vårt viktigste verktøy for oversikt, prioritering og oppfølging av arbeidet.

3.3 Hvilke valg har vi gjort?

Vi tok noen bevisste valg som preget prosessen:

- I Sprint 1 valgte vi å bruke mesteparten av tiden på teori: produkt-backlog, datamodell, RBAC-modell, GDPR-krav og teknologivalg. Vi tenkte at et solid fundament ville spare oss for tid senere.
- I sprint 2 bestemte vi oss for å jobbe mer i grupper enn hver for oss. Det gjorde at frontend- og backend-gruppene hele tiden visste hva de andre gjorde, og vi unngikk misforståelser.
- Vi valgte å møte fysisk i Sprint 2, da vi så hvor mye tid som gikk med til å “hente inn folk” i Sprint 1. Det ga en helt annen rytme.
- Vi delte opp PBIs i mange små tasks med klare sluttkriterier. Dermed ble tasks i praksis en DoD, altså når tasken var gjort, var PBIen ferdig med akkurat den delen. Dette gjorde samtidig at tasks ble mer omfattende enn i Sprint 1
- Når det gjaldt GitHub, valgte vi å jobbe på feature-grener og heller vente med å merge til funksjonaliteten var komplett og testet. Vi var litt usikre på push/pull i starten og redde for å overskrive hverandre, men lærte underveis å bruke pull requests.

3.4 Har vi avveket fra anbefalingene?

Vi har ikke fulgt Scrum-boka slavisk. I design-sprinten gikk vi over tiden på flere øvelser og måtte gjøre enkelte voteringer om igjen fordi folk var borte. I Sprint 1 brukte vi nesten hele sprinten på planlegging, noe som kanskje bryter med tanken om at man alltid skal levere noe “shippable” etter hver sprint. Vi gjorde det likevel fordi vi mente at et gjennomtenkt grunnlag var viktigere enn å levere halvferdig kode som måtte skrives på nytt. Vi har også valgt å se på burndown-grafen mer som en illustrasjon enn et styringsverktøy, ettersom grafen viste “flyttinger” fremfor faktisk arbeid. Dette skyldtes at vi ofte jobbet på en task i flere dager før vi flyttet den til Done. Til slutt bestemte vi at tasks skulle være små nok til å kunne bli flyttet hyppigere, men vi lot fortsatt PBIs bli stående til alle deler var ferdige.

3.5 Refleksjoner om prosessen

Det viktigste vi har lært, er at prosessen er et levende verktøy. I starten trodde vi at Scrumwise, burndown-grafer og story points var fasiten. Etter hvert skjønnte vi at verktøyene bare fungerer hvis vi bruker dem riktig og tilpasser dem til teamet. Vi erfarte at fysisk oppmøte og klare tasks gjorde underverker for effektiviteten. Vi ser også at vi fortsatt har en vei å gå og spesielt når det gjelder å registrere timer fortløpende, flytte tasks til Done med en gang de er ferdige, og holde backloggen oppdatert. Men fremfor alt har prosessen gjort oss mer bevisste på å kommunisere, planlegge og evaluere kontinuerlig.

3.6 Referanser til vedlegg

I vedleggene til sluttrapporten vil vi inkludere:

- Skjermbilder av produkt- og sprint-backlogene våre fra Scrumwise (epics, PBIs, tasks og tilhørende story points).
- Burn-down-grafene for Sprint 1 og Sprint 2 som viser hvordan arbeidsmengden ble redusert over tid.
- Eksempler på brukerhistorier som stammer fra Figma-prototypen.
- En oversikt over definisjonen av “done” slik vi praktiserte den, via oppdelingen i små tasks.
- Skjermbilder av timeføringen i Scrumwise for å dokumentere tidsbruk per oppgave.
- En kort oppsummering av vår Git-workflow med beskrivelser av branch-strategi, pull requests og hvordan vi løste push/pull-utfordringene.

4.0 Veien videre

Selv om vi i løpet av prosjektet har fått på plass en fungerende MVP og en solid struktur for videre arbeid, er vi fullt klar over at mye gjenstår før løsningen kan tas i bruk i en ekte barnehage. I det videre arbeidet ser vi særlig for oss følgende fokusområder:

Fullføre registreringsmodulen: Vi må implementere hele førstegangsregistreringen med forhåndsdefinerte brukere, engangskode via SMS/e-post og automatisk tilknytning av

foresatte og ansatte til riktige barn og avdelinger. Slik det er nå, mangler vi funksjonalitet for å sende koder, matche identifikatorer og opprette brukere i databasen.

Styrke sikkerheten: GDPR og RBAC er grunnpilarer. Neste steg blir å kryptere all trafikk med HTTPS og lagre sensitive data (som telefonnumre) kryptert i databasen. Vi vil også utvide autorisasjonslogikken fra dagens enkle header-sjekker til en mer robust token-basert løsning, og sørge for logging av alle endringer.

Bytte ut midlertidige løsninger: Vi har jobbet parallelt med en JSON-basert MVP og en SQL-database. Nå må vi ferdigstille SQL-versjonen og fjerne JSON-midlertidigheten. Vi vurderer også å flytte databasen til en skyløsning (for eksempel Azure) for å teste skalerbarhet og sikkerhet i en realistisk driftsplattform.

Finpolere frontend: Brukergrensesnittet fungerer, men designet kan forbedres. Vi vil teste med flere foresatte og ansatte, justere tekst og layout, og sikre at appen er responsiv på alle skjermstørrelser. I tillegg planlegger vi å utvide språkstøtten og forbedre tilgjengeligheten (f.eks. skjermleser-støtte).

Utforske nye funksjoner: I design-sprinten diskuterte vi tilleggsfunksjoner som kalender, historikk, værvarsel og meldingssystem. Nå som kjernefunksjonaliteten er på plass, kan vi evaluere om disse bør prioriteres videre. Vi vil basere beslutningen på tilbakemeldinger fra brukere og barnehageledere.

Forbedre prosessen: Vi har lært mye om Scrum underveis, men vi kan bli enda bedre på å logge tid fortløpende, flytte tasks til Done så snart de er ferdige og holde burn-down-grafen oppdatert. Vi vil også fortsette å jobbe fysisk når det er mulig, holde daglige stand-ups korte og fokuserte, og bruke retrospektivene til å justere kursen underveis.

Automatisering og testing: For å sikre kvalitet ønsker vi å sette opp kontinuerlig integrasjon der nye endringer automatisk testes mot backend og frontend. Vi bør også utvikle enkle enhetstester for API-ene og bruke brukertester med foresatte og ansatte for å avdekke eventuelle usability-problemer.

Sluttrapport og presentasjon: Det siste vi skal gjøre, er å fullføre rapporten og forberede en muntlig presentasjon av prosjektet. Her vil vi samle alle vedleggene vi har lovet som backlogger, burn-down-grafer, brukerhistorier, skjermbilder og refleksjoner. For å gi et helhetlig bilde av både resultatet og prosessen vår.

Med disse stegene har vi et klart veikart for hvordan prosjektet kan utvikles fra MVP til et fullverdig produkt. Målet er å levere en trygg, brukervennlig og driftssikker app som barnehager og foresatte faktisk vil ta i bruk, samtidig som vi fortsetter å lære og forbedre arbeidsmåten vår.

5.0 Kilder

Kjærnes, N. (2025). *Forelesning 4: Sprintutførelse, gjennom gang og retrospektiv*. (Intern forelesning, Høyskolen Kristiania). Høyskolen Kristiania.

6.0 Vedlegg

Vedleggene under viser link til mirobrettet og bilder fra sprint 1 og 2.

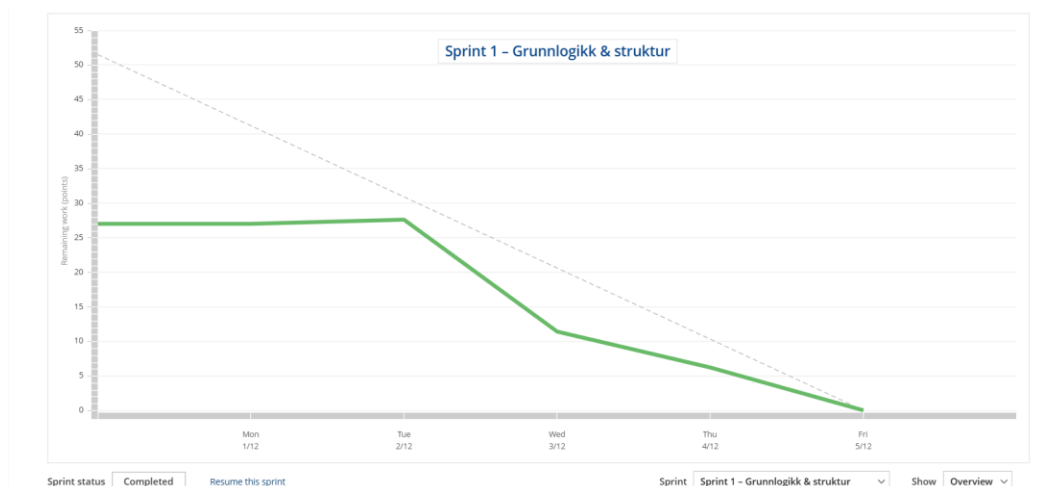
Vedlegg 1 - Link til Mirobrett

https://miro.com/app/board/uXjVJofFbCc=/?share_link_id=867516856128

Vedlegg 2 - Sprint 1 produktbacklog



Vedlegg 3 - Sprint 1 burndowngraf



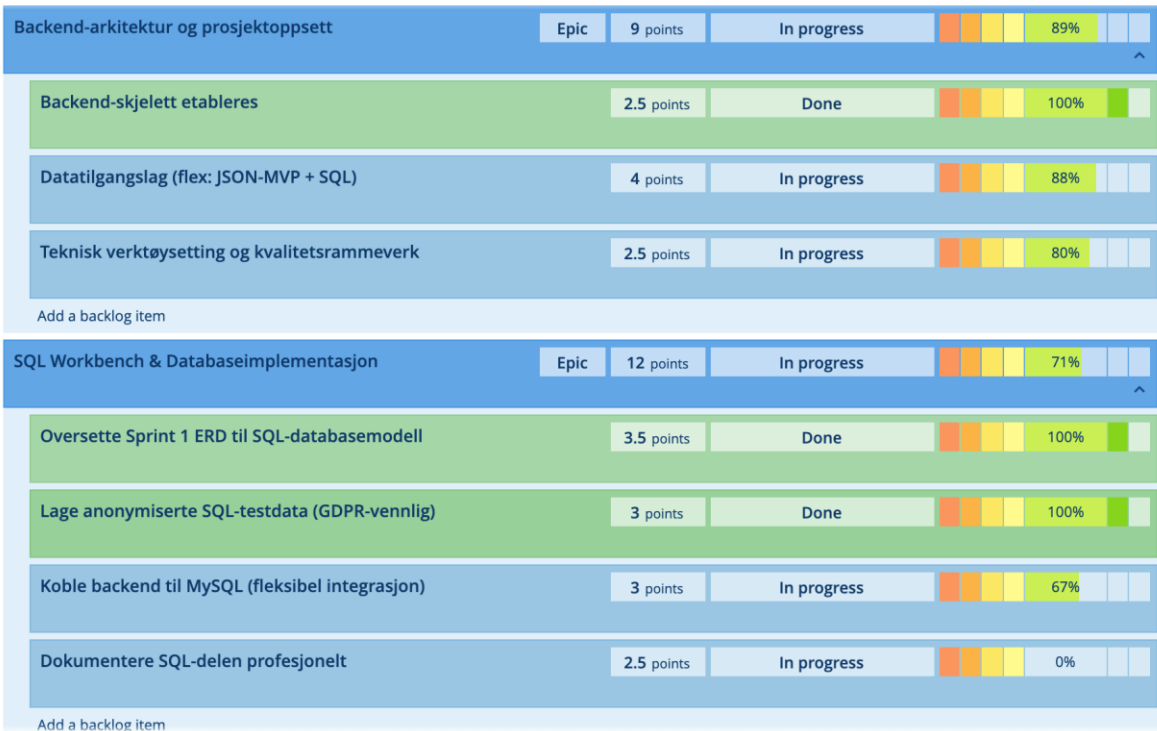
Vedlegg 4 - Sprint 1 retrospective

Retrospective for Sprint 1 – Grunnlogikk & struktur		
What went well?	What went wrong?	What should we do?
Fysisk oppmøte ga bedre flyt og samarbeid	Flere jobbet på overlappende oppgaver - førte til dobbelt arbeid	Dele opp oppgaver tydeligere for å unngå overlap
Teamet ble mer samkjørt enn i designuka	Scrumwise tillater kun én "assigned to", selv om flere jobbet sammen	Logge timer fortløpende og bruke morgenmøtene til gjennomgang
Raskere avklaringer og mindre misforståelser når vi satt sammen	Timer ble loggført for sent (som regel samlet på fredag), gjorde det vanskeligere å være nøyaktig	Fortsette med mest mulig fysisk oppmøte
Alle fikk god kontroll på RBAC, GDPR og struktur pga. overlappende arbeid	Totaloversikten i Scrumwise ble litt misvisende pga. manglende fortløpende logging	Forbedre gruppens kommunikasjon og fordeling av oppgaver
Effektiv deling og gjennomgang av dokumenter		Legge inn mer detaljerte kommentarer i Scrumwise for bedre logging
Add a card	Add a card	Add a card

Vedlegg 5 - Sprint 2 produktbacklog

Sprint 2							
Oppstart, planlegging og Scrum-rammeverke for sprint 2	Epic	12.25 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Sette opp Backlog		4 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Organisere teamroller og arbeidsflyt		2 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Scrum-aktiviteter og seremonier		5 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Daily Scrum – møter hver dag i sprinten		1.25 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Add a backlog item							
Overgang fra Sprint 1 til Sprint 2	Epic	6 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Gjennomgang av MVP-frontend (React/Figma) før backend-start		3 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Mapping av teori fra Sprint 1 - backend-implementasjon		3 points	Done	<div><div></div><div></div><div></div><div></div><div></div></div>	100%		
Add a backlog item							
Backend-arkitektur og prosjektoppsett	Epic	9 points	In progress	<div><div></div><div></div><div></div><div></div><div></div></div>	89%		

Sprint 2



Sprint 2



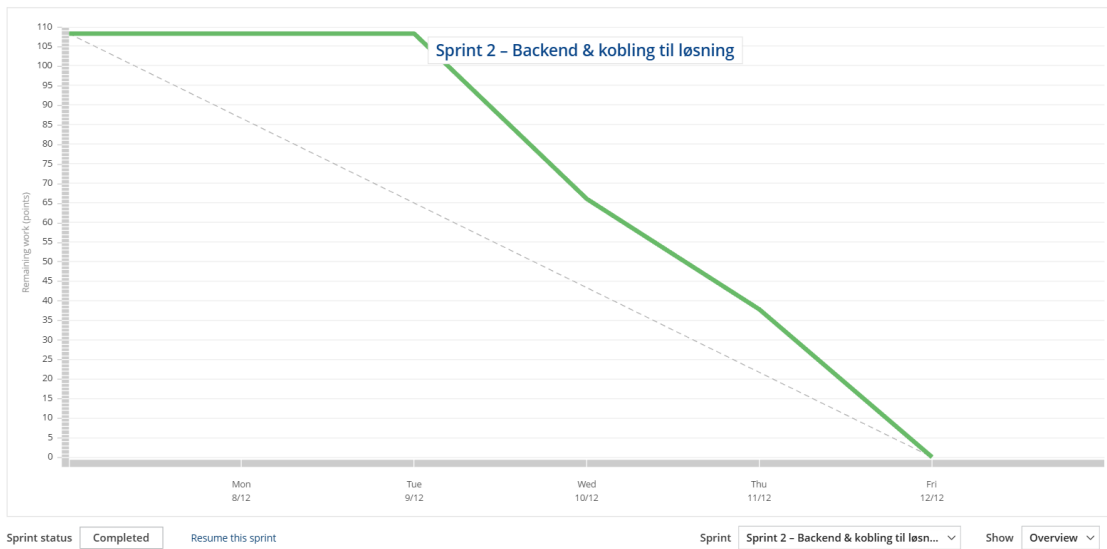
Sprint 2



Sprint 2



Vedlegg 6 - Sprint 2 burndowngraf



Vedlegg 7 - Sprint 2 retrospective

Retrospective for sprint 2 - backend løsning		
What went well?	What went wrong?	What should we do?
Godt gruppe arbeid på backend, database og frontend ga god felles forståelse.	Flere oppgaver ble flyttet sent til Done, noe som gjorde burndown-grafen lite representativ for fremdrift.	Fortsette med fysisk oppmøte og gruppearbeid.
Teamet møtte fysisk hver dag, som førte til færre avbrytelser og høyere effektivitet.	Backend viste seg å være mer krevende enn estimert.	Flinkere til å flytte tasks til Done fortløpende når de faktisk er ferdige.
En enkel frontend MVP gjorde det mulig å teste backend i praksis.	Usikkerhet rundt GitHub skapte skepsis for å overskrive andres arbeid.	Bli tryggere på GitHub for å redusere usikkerhet.
Backend, frontend og databasen ble koblet sammen, som gjorde at systemflyten fungerte som forventet.	Noe dokumentasjon og logging ble forsinket fordi tekniske deler ikke var ferdige før sent i sprinten.	Prioritere dokumentasjon og opprydding sammen med utvikling.
Oppgavene ble fordelt tydelig sånn at det ikke var noe som overlappet.	Add a card	Add a card
Add a card		