

IS-304: Våren 2021
**Utvikling av profilside og forside med innleggsfeed i samarbeid med
Læringsbanken AS**

Emnekode:	IS-304
Emnenavn:	Bacheloroppgave i Informasjonssystemer
Emneansvarlig:	Hallgeir Nilsen
Veileder:	Janis Gailis
Oppdragsgiver:	Læringsbanken AS

Studenter:

Etternavn	Fornavn
Opheim-Halsne	Sven
Nguyen	Khai Hoang Joachim
Høiland	Christer
Perez	Andreas Sheppard
Nuland	Arian Skaara

Jeg/vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeider uten at dette er oppgitt og at alle referanser er oppgitt i litteraturlisten.	JA <input checked="" type="checkbox"/>	NEI <input type="checkbox"/>
Kan besvarelsen brukes til undervisningsformål?	JA <input checked="" type="checkbox"/>	NEI <input type="checkbox"/>
Vi bekrefter at alle i gruppa har bidratt til besvarelsen	JA <input checked="" type="checkbox"/>	NEI <input type="checkbox"/>

Forord

Dette semesteret har vært annerledes for oss alle. Skolene har holdt stengt og mange har måttet tilpasse seg en ny hverdag med digital undervisning og hjemmekontor.

Først, så ønsker vi å takke Læringsbanken som valgte oss til å jobbe på dette prosjektet med dem. Læringsbanken har gitt oss en fin praksisplass og inkludert oss i deres arbeid om å lage en plattform hvor lærere skal kunne dele sin kunnskap. Vi har fått god veiledning og oppfølging av dem og de har gitt oss rom til å være kreative. Vi er svært takknemlige for tiden vi har hatt hos Læringsbanken og hvor godt vi har blitt inkludert og ivaretatt.

Vi ønsker å gi en ekstra takk til utviklerne Xohan Otero Barbosa og Tord Vetle Gjertsen i Læringsbanken, som har veiledet og hjulpet oss med den tekniske delen av prosjektet vårt. De har delt masse av sin kompetanse, som har vært lærerikt for oss.

Vi ønsker også å rette en takk til Carl August Arnesen Tronrud og Bård Bergkvist, som har gitt oss god oppfølging i den administrative delen av prosjektet og styrt oss mot deres visjon for nettsiden. Alle disse har vært svært tilgjengelige og hjelpsomme for oss gjennom hele prosjektet, noe som har gjort praksistiden til en god opplevelse for gruppen.

Vi har også fått veldig god oppfølging og støtte fra vår veileder fra UiA, Janis Gailis. Han har gitt oss gode, konstruktive tilbakemeldinger på det vi gjennomfører. Dette har hjulpet oss med å holde riktig kurs gjennom prosjektet.

Til slutt ønsker vi å takke våre familier og nærmeste, som har støttet oss gjennom prosjektet. Det er ikke alltid like lett å ha flere dager hvor man ikke ser hverandre på grunn av mange timer foran dataskjermen, men de har alltid holdt oss oppe og støttet oss. I en utfordrende tid for oss alle, har de hjulpet oss med å holde motivasjonen og motet oppe.

Igjen, takk for støtten.

Abstrakt

Denne rapporten beskriver våre erfaringer gjennom utviklingen av bachelorprosjektet i faget IS-304 ved Universitetet i Agder. Bachelorprosjektet ble tilbudt til oss av Læringsbanken AS, som jobber med å lage en plattform hvor lærere over hele landet skal kunne dele kunnskap på et sted av og for lærere. Eksempler på innhold som kan deles på Læringsbanken kan være undervisningsopplegg, ideer eller øvelser som lærere kan bruke i undervisningen sin.

Oppgaven vi ble enig om med Læringsbanken var å lage en forside og en innleggsfeed, som senere skal brukes på nettsiden deres. Her skal brukerne ha muligheten til å følge andre brukere eller fag. I tillegg til dette skal også en profilside implementeres.

Fremgangsmåten til gruppen var å bruke Scrum-rammeverket. Prosjektgjennomføringen ble lagt opp over fem sprinter som strakk seg gjennom fire måneder. Disse inneholdt en detaljert plan som inkluderte oppgavebeskrivelser, kostnadsstyring, kvalitetssikring, risikoanalyse, flowchart, wireframes og senere en prototype. Vi sørget for å opprettholde kommunikasjon med Læringsbanken gjennom hele prosessen og forskjellige ansvarsområder innad i gruppen ble fordelt.

En agil prosjektstyring ble opprettholdt gjennom hele prosjektet. Det var viktig for oss å ha kontinuerlig kommunikasjon med oppdragsgiveren, for å kunne gjøre potensielle endringer eller omstrukturering av koden på kort tid. Vi opplevde naturligvis innspill fra oppdragsgiver underveis og vi klarte å løse disse i stor del uten problemer.

Programmeringsspråket vi valgte å bruke for gjennomføring av prosjektet var C# og .NET, med Blazor som rammeverk. For å gjøre nettsiden så brukervennlig som mulig, brukte vi CSS til å lage en interaktiv prototype. Designet skal hovedsakelig videreføres fra Læringsbanken sin allerede eksisterende løsning. Vi satte også en viss kodestandard for å sikre at kvaliteten ble vedlikeholdt gjennom hele prosjektet.

Tidlig i prosjektet opprettet vi en risikoanalyse, for å få oversikt over potensielle utfordringer som kunne inntreffe. Flere av disse ble en realitet gjennom prosjektet, blant annet at ett av gruppene våre ble sykemeldt. Prosjektgjennomføringen klarte derimot å videreføres på tross av dette og vi klarte å implementere de viktigste funksjonene vi utarbeidet med læringsbanken.

Resultatet av dette ble en forside og innleggsfeed hvor brukerne kan se og lage innlegg. Brukerhistoriene ble implementert og kravene i disse ble tilstrekkelig møtt. Scrum-rammeverket ble opprettholdt gjennom hele prosjektet og arbeidsfordelingen ble fulgt gjennom sprintene. Vi mener at resultatet vårt er av god kvalitet og at sluttproduktet kan implementeres av Læringsbanken i deres endelige nettside.

Innholdsfortegnelse

Forord	1
Abstrakt	2
1. Introduksjon	6
1.1 Beskrivelse av prosjektet og oppdragsgiver	6
2. Sentrale avgjørelser	8
2.1.1. Prosjektplan	9
2.1.2. Prosjektstyringsverktøy - Azure DevOps og Trello	10
2.1.3. Risikoanalyse og potensielle utfordringer	10
2.1.4 Kildebruk	11
2.2. Designplan	12
2.3. Kvalitetssikring	12
2.3.1. Versjonskontroll	13
2.3.2 Kodestandarder	14
2.3.3. White- og black box testing	14
2.3.4. Involvering av oppdragsgiver	15
2.4. Stack	15
2.4.1. Lucidchart	16
2.4.2. Visual Studio	16
2.4.3. C#	16
2.4.4. ASP.NET	17
2.4.5. CSS	17
2.4.6. Blazor	17
2.4.7. MSSQL	17
2.4.8. BitBucket/GitHub	17
2.5. Andre sentrale avgjørelser	18
2.5.1. Discord	18
2.5.2. Slack	19
3. Prosjektgjennomføring	20
3.1. Planlegging og analyse	20
3.1.1. Planleggingsfasen	20
3.1.2. Forventinger til gruppearbeidet	21
3.1.3. Tidsstyring	21
3.1.4. Brukerhistorier	22
3.2. Design	23
3.2.1. Skisser	23
3.2.2. Prototype	23
3.3. Testing	25
3.4. Implementering	25

3.5. Kommunikasjon	26
3.6. Scrum	27
3.6.1. Sprint-mål	28
3.6.2. Scrum-master	29
3.6.3. Product backlog	29
3.6.4. Sprint backlog	30
3.6.5. Daily- og weekly Scrum	31
3.6.6. Burndown chart	31
4. Resultat	33
5. Refleksjon	34
5.1. Forventninger	34
5.2. Prosjektplan	34
5.3. Prosjektstyring	36
5.4. Prosjektet	38
5.5. Risikohåndtering	39
5.6. Samarbeid med oppdragsgiver	40
5.7. Utfordringer	40
5.7.1. Digitale verktøy	40
5.7.2. Scrum og timeplanlegging	41
5.7.3. Pandemien	41
5.8. Gruppen	42
5.9. Egevaluering	42
5.9.1. Sven Opheim-Halsne	42
5.9.2. Khai Hoang Joachim Nguyen	42
5.9.3. Christer Høiland	43
5.9.4. Andreas Sheppard Perez	43
5.9.5. Arian Skaara Nuland	43
6. Uttalelse fra Læringsbanken	44
7. Referanser	45
8. Vedlegg	47
Figur 1 - Agile modellering prinsipper	47
Figur 2 - MoSCoW tabellen med time estimering	47
Figur 3 - Eksempel på timeregistrerings-skjema	48
Figur 4 - Prosjektplan	49
Figur 5 - Risikoanalysen	50
Figur 6 - Nielsens brukbarhets heuristikker	51
Figur 7 - Benyons 12 designprinsipper	52
Figur 8 - MVC	53
Figur 9 - Gruppekontrakt	53

Figur 10 - Brukerhistorier	54
Figur 11 - Skisser	56
Figur 12 - Prototype/wireframe	58
Figur 13 - Møtereferat fra styringsmøte 1	65
Figur 14 - Møtereferat fra styringsmøte 2	67
Figur 15 - Product Backlog	68
Figur 16 - Burndown chart Sprint 2	68
Figur 17 - Burndown chart Sprint 3	69
Figur 18 - Burndown chart Sprint 4	69
Figur 19 - Kode eksempel	70

1. Introduksjon

1.1 Beskrivelse av prosjektet og oppdragsgiver

I siste semesteret av studiet IT og informasjonssystemer, skal alle studentene gjennomføre et praksisprosjekt som siste ledd av å oppnå bachelorgraden. Hensikten med dette er å gi studentene et innsyn i hvordan det er å planlegge, gjennomføre og implementere et produkt i samarbeid med en ekte bedrift. Denne prosessen er ment for å gi studentene et innblikk i hva de kan forvente seg i arbeidslivet og rapporten vil være grunnlaget for vurderingen av bacheloroppgaven.

Prosjektet vårt gjennomføres for Læringsbanken AS (<https://www.laringsbanken.no>). De er en privat bedrift som ønsker å tilby en plattform for lærere hvor de kan dele undervisningsopplegg og lignende, i tillegg til å kunne diskutere fagstoff med hverandre. Læringsbanken er en gründerbedrift på Universitetet i Agder og har støtte fra blant annet UiA Nyskaping. Bedriften ble startet av to lærerstudenter etter en av studentene skulle planlegge en matematikkdag. Etter timesvis av planlegging, var studenten usikker på om oppgavearkene skulle i søpla eller samle støv i ei hylle. Derfor ønsker de å skape en plattform hvor lærere kan dele disse oppgavearkene og annet undervisningsopplegg.

Læringsbanken er en bedrift oppbygd av lærerstudenter selv, noe som fører til at de har et bredt nettverk av lærere og lærerstudenter. Dette bidrar til enkel testing av plattformen, og gir gode muligheter for jevnlig oppdateringer og tilbakemeldinger gjennom utviklingsprosessen. Dette kan benyttes som et kvalitetssikrende verktøy i prosjektet.

Opgaven vi avtalte med Læringsbanken går ut på å videreutvikle denne plattformen ved hjelp av deres to utviklere, slik at plattformen er klar til estimert lansering i høsten 2021. Vi skal implementere en ny forside som inneholder en innleggsfeed samt en profilside hvor man kan se og endre sin egen profil og egne innlegg. Hensikten med dette er å lage et bærekraftig system som deres utviklere kommer til å videreutvikle.

Dagens løsning som brukes er innlogging via Feide for å identifisere brukeren og rollen deres og dette er en løsning vi ønsker å forbedre ved å lage en profilside, hvor brukerne kan utdype og endre profilen på toppen av informasjonen fra Feide. Feide er en tjeneste for identitetsforvaltning for lærere og elever i utdanningssektoren. På profilsiden i systemet skal man kunne legge til informasjon om hvilken skole man jobber på, hvilke fag man underviser i, lage nye innlegg og se sine egne innlegg. Man skal også kunne dele undervisningsopplegg som er relevant for fagene man underviser i eller emner man er interessert i.

Forsiden med innleggsfeed som skal implementeres i prosjektet skal la brukerne skrive innlegg samt vurdere dem etter hvordan de synes kvaliteten på innholdet er. Eksempler på slike innlegg kan være læringsinnhold, eller generelle spørsmål som skal kunne diskuteres i kommentarfeltet på innlegget. Innleggene skal også kunne deles, for eksempel med en gruppe lærere. Dersom innlegget er upassende eller useriøst skal det også kunne rapporteres til administratorene av siden, for å sikre at kvaliteten på innleggene opprettholdes.

Videreutviklingen vil bidra til fremtidig implementering og skal bestå av en profilside med diverse informasjon og en funksjonell innleggsfeed med forskjellige interaksjoner. Brukeren skal da ha mulighet til å sette opp en profilside samt delta i et samfunn av lærere hvor en kan diskutere, dele undervisningsopplegg og annen relevant informasjon.

Vi ble bedt om å underskrive en taushetsavtale for å sikre at informasjonen rundt produktet deres skulle forbli konfidensiell. Enigheten om dette var et krav før vi fikk starte prosjektutviklingen.

2. Sentrale avgjørelser

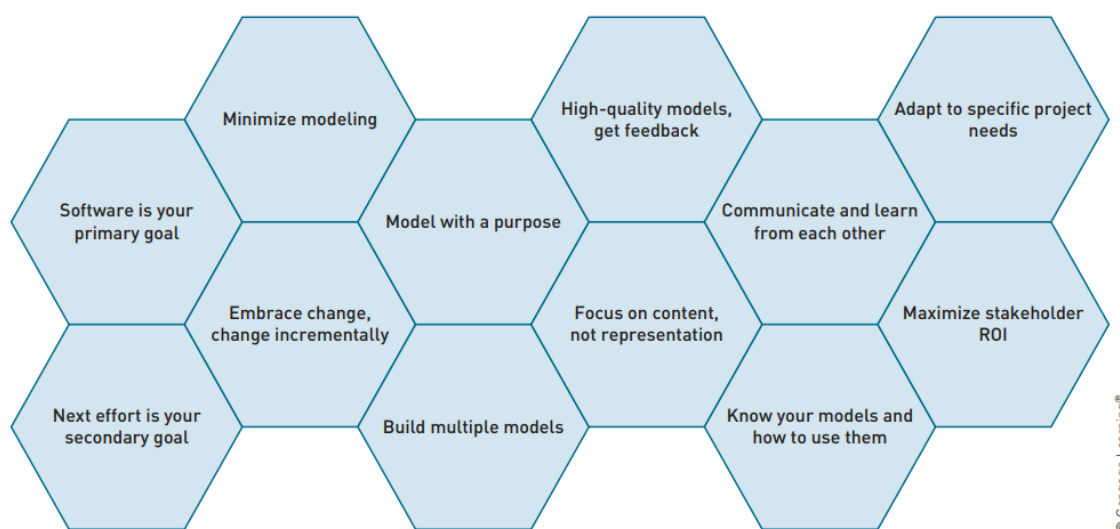
Dette kapitlet beskriver standardene vi har satt for oss selv og valgene vi har gjort i løpet av prosjektet i tillegg til begrunnelser og konsekvenser av disse valgene. Dette inkluderer prosjektmetodikk, designplan, kvalitetssikring og tekniske standarder.

2.1. Prosjektmetodikk

I samarbeid med oppdragsgiver valgte vi å bruke Scrum-rammeverket, da vi mener dette er best egnet for prosjektet. Dette rammeverket inneholder timeboxing, som gir oss oversikt over hvilke oppgaver som skal fullføres i bestemte tidsperioder. Her er det også enkelt å gi oppdragsgiver innblikk i prosessen ved hjelp av prosjektstyringsverktøy, planleggingsmøter, weekly- og daily Scrums. Utviklingen skjer da ved bruk av en “sprint”, som er tidskontrollerte små prosjekter som skal implementere spesifikke porsjoner av et system (Satzinger, Jackson & Burd, 2015, s. 318).

Vi valgte å ha en agil prosjektstyring gjennom hele utviklingsprosessen vår. Det å ha en agil prosjektstyring innebærer å dele opp prosjektet i små biter, som i vårt tilfelle er sprinter og deretter prioritere etter viktighet. I motsetning til et tradisjonelt prosjekt, hvor målet blir satt og utarbeidet derfra med god struktur, er agil prosjektstyring mye mer fleksibelt (Satzinger et al, 2015, s. 304).

Når prosjektet er agil kan man enklere gjøre endringer underveis og tilpasse seg uventede situasjoner som kan oppstå. Agil prosjektutvikling støtter kontinuerlig iterasjon, hvor utvikling og testing vil skje hånd-i-hånd underveis. Denne kontinuerlige iterasjonen skaper en god arbeidsflyt for oss. Vi ønsket å ha noen av de agile modellerings prinsippene høyt prioritert, eksempelvis å kommunisere og lære av hverandre, omfavne forandring og forandre trinnvis (Satzinger et al, 2015, s. 305). Agile modellerings prinsippene kan ses i figur 1 (Satzinger et al, 2015, s. 305).



Figur 1 - Agile modellerings prinsipper

Arbeidsprosessen bestod av en pre-sprint og fire sprints deretter. Hver sprint begynte møter både innad i gruppen og med oppdragsgiver. Her kom vi frem til hvilke oppgaver som skulle gjøres i denne sprinten fra product backloggen og fordeling av hvem som skulle gjøre de forskjellige oppgavene. Disse oppgavene er allerede vurdert etter MoSCoW metoden som kan ses under vedlegg som figur 2 (Benyon, 2014, s. 140). Hver sprint varte i omkring en måned.

Alle oppgavene var time-estimert, men for å kunne spore hvor stor forskjell det var mellom de planlagte og faktiske timene ble alle timene loggført i et timeregistrerings skjema og deretter ført inn sammen med de estimerte timene. De faktiske timene gir oss et inntrykk av om oppgavene var mer eller mindre utfordrende enn det vi vurderte i forkant og kontroll over hvor mange timer hvert gruppemedlem jobber. Disse ligger vedlagt i figur 3.

Hver sprint blir avsluttet med et styringsmøte/sprint review sammen med både oppdragsgiver og veileder, hvor vi går gjennom arbeidet som har blitt gjort. Her er det også rom for innspill og tilbakemeldinger fra begge parter. En del av vårt valg om å ha en agil prosjektutvikling, er fordi vi har opplevd at det gjør prosjektet mer toveis og lett justerbart for både oss og oppdragsgiver.

2.1.1. Prosjektplan

Gjennom tidligere prosjekter har vi erfart at det er viktig med en god prosjektplan. Vi fokuserte derfor på en god planleggingsfase fra starten av, i håp om å få kontroll og oversikt over prosjektet som skulle gjennomføres. I starten av prosjektet lagde vi et planleggingsark hvor vi fordelte ansvar og retningslinjer for hvordan arbeidsmiljøet i gruppen skal være. Der planla vi også hvordan eventuelle konflikter i gruppen skal håndteres. Senere utviklet vi også en risikoanalyse for å videre analysere dette. Konkrete mål for prosjektet ble bestemt og forventet arbeidsmengde per gruppemedlem ble avtalt. Dette resulterte i at alle hadde kontroll på deres ansvars- og fokusområder før vi gikk i gang med prosjektet. Prosjektplan ligger vedlagt som figur 4.

Vår oppgave bestod av å videreutvikle systemet og nettsiden Læringsbanken allerede har utviklet. Vi fikk i oppgave å utvikle en forside med innleggsfeed samt en profilside for brukere av nettsiden. Vi fikk beskjed om at vi kun skulle fokusere på nettserversjonen av siden deres og ikke involvere en mobilversjon i vårt prosjekt.

Nettsiden er for øyeblikket ikke tilgjengelig for offentligheten da Læringsbanken ikke ønsker å tilgjengeliggjøre den før systemet er komplett og testing har blitt utført. Derfor fikk vi tilgang til deres repository for systemet via BitBucket og det ble opprettet en egen branch til oss der, slik at våre endringer i systemet ikke vil påvirke det eksisterende systemet.

Vi fordelte ansvaret etter områdene vi mener gruppemedlemmene hadde mest kompetanse i. Denne fordelingen ble satt opp som følger:

- Front-end, utforming av nettsiden: Arian
- Back-end, utforming av funksjonalitet: Khai
- Fullstack-utvikler: Sven
- Skrivning, utforming av bacheloroppgaven: Andreas
- Gruppeleder: Arian
- Scrum-master: Christer
- Kontaktpersoner: Arian og Christer

Uavhengig av ansvarsområdene, så ønsket vi at alle hadde en viss grad av kompetanse og innblikk om hverandre sitt arbeid. Gruppen arbeidet også på tvers av ansvarsområdene sine etter behov. Dette er viktig dersom det skulle oppstå uforutsette problemer med et av gruppemedlemmene og for å sørge for at arbeidet kan gå videre dersom en av oss skulle være utilgjengelig eller ute av stand til å være med i prosjektgjennomføringen. Alle måtte ha kunnskap om programmeringsspråket, de digitale verktøyene og alle modellene og tegningene som ble satt opp.

For å sikre at prosjektet ble gjennomført på riktig måte, hadde vi styringsmøter sammen med veileder og produkteier. Her fikk vi presentert hva vi hadde gjort og redegjøre for veien videre i prosjektet. Hensikten med disse styringsmøtene var å gi både produkteier og veileder et innblikk i prosessen i tillegg til å få tilbakemeldinger på prosjektgjennomføringen for å kunne videreutvikle systemet på best mulig måte. Disse ble holdt digitalt grunnet den pågående pandemien.

2.1.2. Prosjektstyringsverktøy - Azure DevOps og Trello

Fra starten av prosjektet ønsket gruppen å ha effektivt arbeid ved hjelp av Scrum-rammeverket. I utgangspunktet skulle vi bruke Trello som prosjektstyringsverktøy, men bestemte oss for å bytte til Azure DevOps ettersom programvaren enklere tilbyr oss å lage modeller direkte fra prosjektstyringsverktøyet, som for eksempel burndown charts. Trello hadde svært begrensede funksjoner for vårt bruk og mye av funksjonene vi ønsket oss lå bak en betalingsvegg.

Azure DevOps tilbyr funksjoner vi kan bruke for å utvikle noe ved bruk av Scrum-rammeverket, ble det et veldig vellykket valg. Azure DevOps tilbyr blant annet product- og sprint backlog, flowcharts og burndown-charts. Prosjektstyringsverktøyet er vitktig for å ha full oversikt over sprintene og tidsfristene og sørge for at oppgaver ikke blir glemt. Senere i prosessen byttet vi tilbake til Trello.

2.1.3. Risikoanalyse og potensielle utfordringer

For å kunne identifisere og håndtere potensielle risiko knyttet til både gruppen og prosjektet, er det viktig at man har en oversikt over hva slags innvirkning dette vil ha. Risikoene vil kunne påvirke både gruppen, prosjektet og gruppemedlemmet og derfor er det lurt å ha

kontroll på hvilke situasjoner som kan måtte bli håndtert. For å løse og visualisere dette lagde vi en risikoanalyse. Vi satte opp potensielle risikoer som kan oppstå, fordelt under forskjellige temaer. Vi utviklet deretter en tabell med hvor stor risiko de forskjellige totalverdiene er på en skala fra 1 til 25; 1-3 regnes som lav risiko, 4-6 regnes som medium risiko, 7-16 regnes som høy risiko og 17-25 regnes som veldig høy risiko.

Risikoene ble først vurdert etter alvorlighet, som vil si hvor stor innvirkning det har både på gruppen og enkeltpersonen. Det inkluderer også hvordan det vil påvirke gruppen og prosjektgjennomføringen i sin helhet. Deretter ble risikoen vurdert etter sannsynlighet, altså hvor sannsynlig det er at denne risikoen vil inntreffe. Under risikovurderingen fokuserte vi på å relevans til de forskjellige risikoene, blant annet med hensyn til COVID-19 pandemien. Dette førte til en kombinert risikofaktor ut ifra tabellen, basert på hvor sannsynlige og alvorlige de forskjellige risikoene var. Risikoanalysen ligger vedlagt som figur 5.

Basert på våre risikovurderinger konkluderte vi med at følgende risikoer, som vi måtte legge ekstra vekt på å overvåke og forhindre, var følgende:

- Risiko 1: Psykisk utmattelse
- Risiko 3: Motivasjonsmangel
- Risiko 5: Sykdom
- Risiko 7: Programvarefeil
- Risiko 8: Lagringsfeil

Det kommer tydelig frem i risikoanalysen vår at psykiske utfordringer er den største risikoen vi potensielt måtte kunne håndtere. Dette har blitt forsterket av pandemien og mange synes det er utfordrende når det er så mye hjemmekontor og mangel på sosial kontakt. Derfor vil vi som gruppe tilrettelegge for at dette ikke skal oppstå og gjøre vårt beste for å forhindre at gruppemedlemmene går på en psykisk smell på grunn av faktorer som for eksempel overarbeid eller stress. Flere av disse risikoene ble en realitet for oss, da vi hadde et gruppemedlem som ble sykemeldt i den første måneden av prosjektet på grunn av sin psykiske helse. På grunn av at vi vurderte dette som en høyrisiko i starten av semesteret ved hjelp av risikoanalysen hjalp det oss med å håndtere denne situasjonen som oppstod på best mulig måte. Dette gjorde at vi kunne fortsette prosjektet som planlagt selv om vi var ett medlem mindre og enklere tilrettelegge for at gruppemedlemmet kom tilbake i prosjektet igjen på best mulig måte.

2.1.4 Kildebruk

Det var viktig for oss å velge sikre og pålitelige kilder gjennom prosjektet. Derfor prioriterte vi å bruke relevant innhold fra pensumbøkene fra forskjellige fag gjennom studiet. For eksempel så var dette viktig for å få en dypere forståelse av scrum og agil prosjektutvikling før vi gikk i gang med prosjektet. Dette gav oss et bedre utgangspunkt for hvordan vi kunne involvere oppdragsgiver i større grad og hva som kunne være potensielle utfordringer med dette.

2.2. Designplan

Vi oppfatter det som viktig å benytte god heuristikk og gode strategier for å oppnå ønskede resultater i utviklingen et produkt. For at brukerne skal kunne ta i bruk og utnytte prosjektet vårt på best mulig måte, er det viktig at vi tilgjengeliggjør funksjonene på en god måte. Derfor er godt design helt avgjørende.

Vi ønsket å involvere en del av designteorien, men fikk ikke anvendt dette da oppdragsgiver alt hadde et design vi måtte forholde oss til. Noe av teorien vi ønsket å inkludere er blant annet Nielsens 10 brukbarhets heuristikker som er generelle retningslinjer for interaksjonsdesign (Nielsen, 2020). Disse kalles “heuristikker” som betyr læren om hvordan en best oppnår og lagrer kunnskap. Alle av Nielsens 10 brukbarhets heuristikker ligger i figur 6 (Nielsen, 2020). Benyons 12 designprinsipper (figur 7) kan hjelpe å orientere designeren til nøkkelegenskaper av godt design og gjøre designeren klar over viktige problemer (Benyon, 2015, s. 86). Vi ble også informert om et designspråk Google skapte i 2014 kalt “Material Design” (Google, 2014).

2.3. Kvalitetssikring

Kvalitetssikring er meget viktig i de fleste felt, også for oss i dette bachelorprosjektet. Etersom dette prosjektet innebærer å samarbeide med en bedrift, er det ekstra viktig å kvalitetssikre da eventuelle problemer også kan gå utover bedriften. Manglende kvalitetssikring kan være svært kritisk for prosjekter og derfor er det viktig å utøve gode rutiner for kvalitetssikring gjennom hele prosjektgjennomføringen.

For å sikre kvalitet, valgte vi å lage en plan for å følge en satt kodestandard. Det innebærer å navngi filene på en relevant og variert måte for å gi bedre oversikt. Dette skaper en god sammenheng og gjør hele koden enklere å forstå. Under utvikling har vi arbeidet med en del av “must have”-funksjonene fra MoSCoW-metoden i hver separate “branch”. Dette bidrar til redusert risiko for at en pågående del av utviklingen ikke komme i konflikt med en annen, og unngår kodeduplisering. Siste nivå av kvalitetssikringen innebærer at en av utviklerne fra Læringsbanken går igjennom koden og godkjenner før det blir sendt til “master” branchen.

Da vi utformet product backloggen til prosjektet var det viktig for oss å utforme arbeidsoppgavene basert på Læringsbanken sine ferdig utviklede brukerhistorier. Dette er svært viktig for å sikre at vår tolkning av oppgaven og utviklingen vi gjør samsvarer mest mulig med visjonen Læringsbanken har for nettsiden. Ved å ta utgangspunkt i deres brukerhistorier, i tillegg til å ha en agil prosjektstyring, sørger vi for at det blir færre misforståelser og mer dialog under prosjektutviklingen.

Som tidligere nevnt utviklet vi også en risikoanalyse i starten av prosjektet. Dette er også et kvalitetssikrende tiltak, da det vil gi oss et innblikk i hvor krevende det vil være å håndtere uforventede situasjoner som kan oppstå. Under utviklingen av risikoanalysen diskuterte gruppen naturligvis konsekvensene av de gitte scenarioene dersom disse problemene skulle oppstå. Dette belyste at det er viktig å være løsningsorientert selv i krevende situasjoner og

viktigheten av å være forberedt på slike risikoer ble desto tydeligere. På grunn av risikoanalysen ble vi bedre forberedt på eventuelle uforutsette hendelser som kan oppstå og lærte hvordan vi skulle takle dette på best mulig måte for å ivareta kvaliteten i prosjektet. Risikoanalysen ligger vedlagt som figur 5.

En viktig del av kvalitetssikringen, i tillegg til kodenstandard og versjonskontroll, er å tilgjengeliggjøre prosjektet for testing av en gitt gruppe testere. Vi utførte whitebox-testing og blackbox-testing (GeeksforGeeks, 2020) under utviklingsprosessen, og ønsket å utføre testing av systemet med en gruppe lærere mot enden av prosjektet. Dette fikk vi dessverre ikke tid til under vår praksistid, men det vil bli utført i ettertid av Læringsbanken.

Oppdragsgiver har et nettverk av andre lærere og lærerstudenter som var villige og tilgjengelige til å teste vårt system, det betyr at målgruppen vår ville vært testere. Testernes opplevelser av prosjektet og hvordan de utnytter det vil gi Læringsbanken verdifulle tilbakemeldinger som kan hjelpe til å gjøre forbedringer for å videre kvalitetssikre systemet. Disse tilbakemeldingene gir en bekreftelse på hvilke deler av prosjektet som fungerer bra i praksis og hva som kan forbedres.

2.3.1. Versjonskontroll

For å sikre kvaliteten på utviklingen som blir gjort er det viktig å kontinuerlig gjennomføre versjonskontroll av alt arbeidet som blir gjort. Store feil i koden kan være fatale for prosjektet og derfor er det viktig at koden som produseres må gjennomgås av andre for å sikre at det som blir gjort er riktig og følger kodenstandard til systemet. Versjonskontrollen bidrar også til å forsikre at det som pushes inn bidrar med den funksjonen den er ment til å gjøre, uten å påvirke eller ødelegge de andre funksjonene i prosjektet.

I dette prosjektet brukte vi programvaren BitBucket for versjonskontroll og lagring av kildekode. Grunnen til at vi brukte dette systemet og ikke et system vi var kjent med fra før av, som for eksempel GitHub, var at Læringsbanken allerede brukte BitBucket som versjonskontroll. Deres BitBucket inneholdt kildekoden til prosjektet i master-branchen, som vi brukte som utgangspunkt for vår utvikling av siden. Master-branchen ble klonet til en bachelor-branch, som vi kunne jobbe på. Når vårt bidrag til systemet er ferdigstilt og klart for å bli merget med Læringsbankens system, kan dette skje ved hjelp av BitBucket og versjonskontrollen som gjennomgås der.

Bitbucket oppdateres automatisk når et medlem pusher koden sin inn, noe som fører til at de andre som jobber med koden alltid har siste oppdaterte versjon av den.

Dette løste vi blant annet med å lage en ny branch for hver oppgave som ble gjort, for ikke å påvirke master-branchen hver gang koden ble pushet. At hver oppgave var tilknyttet sin egen branch gjorde det også lettere å hente ut hver funksjon i forhold til design eller forbedringer før det ble merget til master. I tillegg til dette hadde vi også en egen bachelor-branch, som krevde gjennomgang og godkjenning av koden før det ble pushet videre til master.

Versjonskontroll av branchene våre og godkjenning fra utviklerne Læringsbanken var svært viktig, da en feil i en branch kan påvirke hele prosjektet og i verste fall ødelegge det helt. Versjonskontrollen er svært viktig for å sørge for at arbeidet som blir gjennomført er funksjonelt og for å sjekke at kodenstandardene blir opprettholdt.

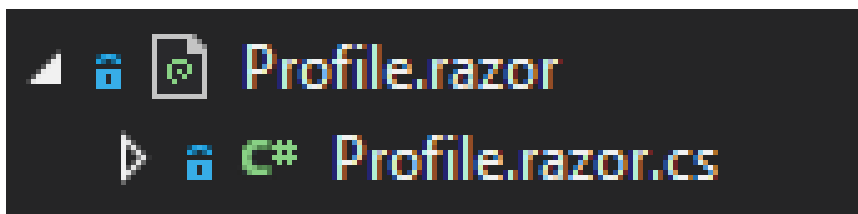
Litt over halvveis i semesteret bestemte Læringsbanken seg for å bytte versjonskontroll fra BitBucket til GitHub, da BitBucket er en betalt tjeneste og GitHub er gratis. Dette skapte ingen komplikasjoner.

2.3.2 Kodestandarder

For å sikre kvaliteten i koden og at alt er lettlest, fulgte gruppen kodenstandarder gjennom hele prosjektet. Kodestandardene er viktig for å sørge for at prosjektet er ryddig og greit for de som skal godkjenne det og eventuelt utvikle videre. Kodestandarden vår involverte blant annet å ha beskrivende kallenavn på både filer og brancher som ble opprettet. Koden ble også dokumentert og kommentert, slik at funksjonene på hver klasse blir lett forståelig og dette gjør at versjonskontrollen blir enklere.

I tillegg til dette brukte vi også CamelCase til filnavn og navngivningen av variabler. Dette innebærer å navngi filer og objekter uten mellomrom eller tegnsetting, noe som innebærer en stor bokstav for å skille ordene. For eksempel så hadde vi en mappe for controller filer og en for modell filer. Begge disse tiltakene bidrar til å gi mer oversikt over prosjektet og sørge for å hindre kodeduplisering. Det å ha fokus på kodenstandarder fra starten av hjalp oss med å lage et ryddig prosjekt, som enkelt kan implementeres eller utvikles fra Læringsbanken sin side i fremtiden.

Både det Læringsbanken hadde fra før og det vi har vært med å implementere bruker MVC-arkitekturen. De har api som Model, som tar seg av alt av kommunikasjon mellom nettsiden og databasen. View er standard HTML og CSS som er implementert i filene, mens Controller ligger i klasse-filene. I klasse-filene var alt av kode lagt nederst i filen under “@code”, men utviklerne holder på med en overgang der de legger all koden i en under-fil; Hoved-filen blir for eksempel “Profil.razor” mens under-filen med all kode blir kalt “Profil.razor.cs”. Visual studio (Microsoft, 2019) vil da automatisk legge CS-filen under razor-filen som vist i figur 8.



Figur 8 - MVC

2.3.3. White- og black box testing

For å sørge for at prosjektet har gjennomgående god standard har vi hatt kontinuerlig fokus på testing. Et tiltak for å oppnå god standard var ved å gå gjennom koden med bruk av White

box testing av det interne koden. Med white box testing er det essensielt at alle delene i prosjekt fungerer som de skal, før de blir endelig implementert. Dersom delene ikke fungerer som de skal, vil ikke systemet funke som det skal. Da vi bruker denne metoden for testing går vi gjennom linje for linje og tester, slik at vi er sikre på at koden fungerer som den skal. Dermed er det mulig for oss å vite med en gang om noe som ikke fungerer eller kjører.

En av de metodene under white box testing er å sette en breakpoints på hver enkelte funksjon under debuggingen av systemet og optimalisere på ut fra resultatene av dette. Teste-rutinene fungerte bra og spesielt siden det har vært mange nye verktøy i dette prosjektet. Det å begynne med nye programmeringsspråk og rammeverk for oss var krevende, men takket være en del testing fra tidlig i prosjektet så fikk vi svar på om koden fungerte eller ikke.

Med bruk av black box testing går vi gjennom det visuelle delen av prosjektet om metoder og funksjoner fungere som det skal til før den blir sendt ut til sluttbrukeren.

2.3.4. Involvering av oppdragsgiver

For å sørge for at prosjektet ble utviklet i tråd med oppdragsgiver sine ønsker, hadde vi hyppige møter med Læringsbanken. Vi hadde ofte møter med deres utviklere, som bidro med teknisk hjelp. Dette hjalp oss mye gjennom prosjektet og gjorde det enklere for oss å forstå mye av de nye systemene vi måtte sette oss inn i. Etter ønske fra Læringsbanken bestemte vi oss for at ett av gruppemedlemmene ble utpekt som gruppeleder, Lederen hadde hovedansvaret for å opprettholde en kontinuerlig dialog med oppdragsgiver.

Vi hadde også møter med daglig leder i Læringsbanken, altså vår oppdragsgiver. Formålet med disse møtene var å sette kursen videre og høre på tilbakemeldinger og innspill i arbeidet som var blitt gjort så langt. Under møter med bedriften Læringsbanken ble det avtalt at dette prosjektet og vårt produkt vil være med å danne grunnmuren for deres ferdige produkt (nettside). Det innebærer at det vil være andre som ser på, endrer på og baserer seg på produktet vi leverer.

Ettersom Scrum-rammeverket var i stor fokus under utviklingen, etterspurte vi jevnlig møter med arbeidsgiver for å forsikre oss om at vi hadde rett fokus i prosjektet og at produkteier var fornøyd. Vår løsning på dette var å møte med utviklere tidlig i sprint og møte med oppdragsgiver i slutten av sprinten. Ettersom pandemien har hatt stor effekt semesteret 2021, har det hovedsakelig vært digitale møter. Dette har gitt mulighet for å møtes uten å tenke på fysiske grenser og selv om vi helst hadde foretrukket fysiske møter har digitale møter fungert tilstrekkelig.

2.4. Stack

Oppdragsgiver gav oss frihet til å velge den teknologiske stacken vi ville bruke selv. Det eneste kravet de hadde til oss var å bruke BitBucket som et kildekode-lager. Vi vurderte alternativene for hva slags språk vi skulle bruke og hvilket rammeverk som kunne være relevant for oss. Vi ønsket å lage et moderne produkt, med standarder som ville vært svært

likt det vi kommer til å møte i arbeidslivet etter endt studie. Vi valgte å følge Læringsbankens standarder, da dette gjorde det lettere for oss å ta i bruk og utvikle etter deres systemer. Produktet ble utviklet etter følgende standarder:

- C# (Microsoft, 2020) og ASP.NET som programmeringsspråk (Microsoft, 2019)
- CSS som designspråk og UI design (W3Cs CSS Working Group, 2016)
- Blazor som rammeverk (Microsoft, 2018)
- Bitbucket som kildekode-lager (Bitbucket, 2020)
- Visual Studio 2019 som IDE (Microsoft 2019)
- MSSQL som database (Microsoft, 2019)

2.4.1. Lucidchart

Lucidchart (Lucid, n.d.), en plattform med mer enn 10 millioner brukere, tilbyr en tjeneste hvor brukerne kan utvikle diagrammer, flowcharts, wireframes, tankekart og diverse andre ikoner og bilder. Lucidchart har vært meget verdifullt nemlig fordi man kan gjøre alt dette samtidig med andre. Dette gjorde eksempelvis utarbeidingen av low-fidelity prototyper betydelig mye lettere og mer praktisk for gruppen.

2.4.2. Visual Studio

Visual Studio er en IDE (Integrated Development Environment), eller utviklingsmiljø, som er utviklet av Microsoft. Programmet gir utviklere en plattform for å lage, endre, teste og debugge kode. Dette hjelper utviklere med å utvikle programmer, spesielt rettet mot Windows. Vi valgte å bruke Visual Studio fordi vi har erfaring med et liknende program, Visual Studio Code, fra tidligere prosjekter. Grunnen til at vi byttet over til Visual Studio var fordi dette programmet er bedre optimalisert for C# og .NET-rammeverket.

2.4.3. C#

C# er et objektorientert språk og i likhet med Visual Studio er det utviklet av Microsoft (Microsoft, 2020). Vi hadde ikke noe erfaring med C# fra før og dette viste seg å bli en utfordring for oss gjennom prosjektutviklingen. Det var vanskelig å utnytte alle funksjonene C# har å by på og i tillegg var det vanskelig å vite hvordan man skulle bruke det til i samarbeid med de andre digitale verktøyene. Heldigvis hadde Læringsbanken utviklet sitt foreløpig system med dette språket, noe som gjorde at vi kunne bruke det som inspirasjon. I tillegg var Læringsbankens utviklere svært hjelpsomme og tålmodige med oss.

Det finnes også utallige læringsressurser og innføringskurs på nett for å lære og utnytte dette. Grunnen til at vi brukte dette programmeringsspråket som standard er fordi Læringsbanken allerede hadde brukt dette til å utvikle siden sin og vi fulgte derfor deres rutiner på dette. C# er et nyere og mer moderne språk enn Java og vil også være mer relevant for oss i arbeidslivet. Kombinert med .NET rammeverket vil dette effektivisere utviklingsprosessen og derfor brukte vi det som vår standard.

2.4.4. ASP.NET

I likhet med både C# og Visual Studio er også ASP.NET utviklet av Microsoft (Microsoft, 2019), noe som gjør at de går sømløst sammen og er optimalisert for hverandre. Dette er en plattform hvor man kan bruke flere språk til å utvikle programmer for flere plattformer og brukes også til å kjøre programmene. Programmeringsspråket vi valgt å bruke, C#, er helt avhengig av .NET for å kunne fungere med API. Språket inneholder også et stort klassebibliotek, som vi kan utnytte gjennom prosjektet vårt. Visual Studio er spesielt rettet mot .NET utvikling, noe som gjorde at valget av rammeverk kombinert med C# ble enda enklere for oss.

2.4.5. CSS

Vi valgte å bruke CSS (Cascading Style Sheets) som designspråk gjennom prosjektet ettersom det er et kjent og velbrukt språk. Det kan brukes til å endre blant annet oppsett, farger og stiler på flere sider om gangen. Dette er et språk vi har god kompetanse i og har brukt i tidligere prosjekter. CSS er også allerede godt innarbeidet hos Læringsbanken og dette er noe vi ønsker å bygge videre på med vår kompetanse.

2.4.6. Blazor

Blazor er et åpent rammeverk rettet mot webutvikling ved hjelp av C# og ASP.NET. Også dette er en eksisterende standard hos Læringsbanken, som gjorde at vi måtte lære oss dette. Grunnet at vi har brukt JavaScript tidligere, så gikk denne overgangen ganske fint. I tillegg til dette så er Blazor spesielt rettet mot utvikling i ASP.NET og C#, noe som var svært relevant for vårt prosjekt.

2.4.7. MSSQL

Under utvikling ble MSSQL brukt som databasesystem ettersom Læringsbanken brukte dette i sin eksisterende database. Gruppen har tidligere erfaring med lignende program og det var dermed meget overkommelig. Databaser blir brukt til å lagre og hente informasjon og MSSQL blir brukt til å strukturere koden.

2.4.8. BitBucket/GitHub

For at samarbeidet gjennom utviklingen blant både gruppemedlemmene og Læringsbanken er det viktig at man har et kildekode lager, hvor man kan pushe og pulle kode slik at alle får med seg funksjonene som legges inn. Vi har brukt GitHub tidligere, men BitBucket var lett å tilpasse seg til da disse to deler mye av de samme bruksfunksjonene. Vi brukte BitBucket til versjonskontroll og hadde krav om godkjenning før koden ble merget med master-branchen for å sikre kvaliteten på koden.

Av økonomiske årsaker valgte Læringsbanken å bytte kildekode lager fra BitBucket til GitHub underveis i prosjektet. All kildekode ble overført dit og vi tok plattformen i bruk fortløpende. Denne overgangen gikk fint for oss, da vi som nevnt har erfaring med GitHub fra tidligere prosjekter.

Andre digitale verktøy ble vurdert i prosessen var Trello i stedet for Azure DevOps. Grunnen til at vi til slutt valgte å gå for sistnevnte var fordi dette prosjektstyringsverktøyet inneholdt funksjoner som gjorde det enklere å utforme grafer som f.eks burndown chart. Ulempen med å velge DevOps var derimot at det ikke fantes innebygde verktøy for å estimere timer der, noe som enkelt kan gjøres på Trello. Derfor måtte vi beregne timene separat i et eget dokument.

Grunnen til at vi brukte de utviklingsverktøyene vi gjorde var i stor grad fordi disse er rutiner Læringsbanken allerede praktiserer. En annen viktig faktor for valget av digitale verktøy er fordi vårt prosjekt mest sannsynlig skal videreutvikles etter at vi er ferdige med det. For å sørge for flyt i videreutviklingen av prosjektet er det viktig at vi fulgte deres kodestandarder og brukte samme system for å sikre enkelhet i videreutviklingen. For å kunne bygge videre på deres nettside hadde det vært vanskelig dersom vi skulle begynne å gjøre store endringer på f.eks. språket som brukes. Det hadde gjort det vanskeligere for oss å få riktig opplæring av dem, da de kanskje ikke hadde hatt like mye erfaring hvis vi f.eks hadde valgt Python. Det er heller ikke gitt at dette hadde passet inn med deres system hvis de velger å implementere vårt prosjekt i deres system.

2.5. Andre sentrale avgjørelser

I denne delen av rapporten skal vi gå igjennom de andre valgene vi gjennomførte for å sørge for best mulig prosjektgjennomføring. Under vår studietid har vi lest at kommunikasjon har en essensiell rolle for at vi skal få fremgang og resultater gjennom prosjektet. Dette oppfattet vi som riktig, da vi i ettertid så hvor stor effekt kommunikasjonen hadde, ved at oppgaver og idéer ble godt kommunisert både oss imellom og med oppdragsgiver. Grunnet den pågående COVID-19 pandemien har kommunikasjon over nett vært viktigere enn noensinne og vi har vært takknemlige for å ha såpass gode teknologiske kommunikasjonsmidler tilgjengelig. Disse vil vi diskutere i denne delen av kapittelet.

2.5.1. *Discord*

Som studenter under en pågående pandemi er det viktig å finne alternative måter å kommunisere på. Et program som har blitt betydelig mye brukt er Discord, et populært program for kommunisering over nett. Discord tilbyr gratis tjenester som samtaler med og uten video, chat, strømming av skjerm og applikasjoner. Sammen med dette så gir Discord brukerne sine muligheten til å lage egne private servere som gir muligheten til å organisere seg.

Ettersom det er få fysiske møter under pandemien ble Discord brukt daglig. Det ble et verdifullt hjelpemiddel, spesielt da applikasjonen tilbyr skjermdeling. Skjermdeling er har vist seg å gjøre kommunikasjon og kontrollering av arbeid betydelig lettere. Applikasjonen gir en lavterskel mulighet for å ha møter og samtaler, og vi brukte det for å gjennomføre alle møtene våre med både veileder og oppdragsgiver.

2.5.2. *Slack*

Etter ønske fra oppdragsgiver, brukte vi kommunikasjonsplattformen Slack for å opprettholde dialog med dem. Her kunne vi kommunisere med de enkelte teammedlemmene både individuelt og i grupper med flere medlemmer. Dette er en eksisterende løsning Læringsbanken har brukt lenge og derfor var det både praktisk og enkelt for begge parter å involvere oss på Slack. Enkelte av oss i gruppen hadde også erfaring med det fra før av, noe som gjorde prosessen enklere.

3. Prosjektgjennomføring

Prosjektet ble gjennomført gjennom en pre sprint og fire sprints deretter. Denne delen av rapporten vil beskrive planleggingsfasen og gjennomføringen av prosjektet

3.1. Planlegging og analyse

For å sikre kvalitet og flyt i prosjektet er god planlegging helt avgjørende, som er grunnen til at vi startet prosjektet med en pre-sprint. Pre-sprinten fungerte som en planleggingsfase, hvor hele prosjektet ble delt opp i sprints og datoer. Her la vi opp hvor mange sprints prosjektet skulle fordeles utover og hvilke oppgaver fra product backloggen vi ønsket å oppnå i hver av sprintene. Denne fasen av prosjektet inneholdt mye dialog med oppdragsgiver for å få en forståelse av arbeidet som skulle gjøres og sette opp forskjellige arbeidsoppgaver som skulle gjøres gjennom prosjektet.

Som prosjekteringsverktøy brukte vi Azure DevOps, hvor vi satte opp en kolonne for hver sprint og en kolonne hvor ferdige oppgaver skal legges. Brukerhistoriene dannet grunnlaget for hvordan vi formulerte og vurderte oppgavene etter MoSCoW metoden. Oppgavene ble i samarbeid med oppdragsgiver, vurdert etter MoSCoW metoden for å avgjøre viktigheten og prioriteringen av de forskjellige oppgavene i prosjektet. Det vil si at de aller viktigste oppgavene, som må være tilstede i prosjektet ble vurdert til "Must have". De med mindre viktighet, som helst skal være med ble prioritert som "Should have". Deretter kommer "Could have" og til slutt "Won't have" for oppgavene med lavest prioritering.

I vårt prosjekt valgte vi å legge hovedfokuset vårt på oppgavene som vi vurderte etter "Must have" og deretter "should have". Disse oppgavene er ikke kun prioritert fordi de er mest kritiske for systemet, men de er også de mest tidkrevende oppgavene for å få inn den ønskede funksjonaliteten. Derfor er en slik prioritering viktig for at ikke tidkrevende og viktige oppgaver blir utarbeidet på et feil eller for sent tidspunkt i prosjektgjennomføringen.

3.1.1. Planleggingsfasen

Det viktigste aspektet in planleggingsfasen vår var å kartlegge hvordan produktet skulle bli og hvordan vi skulle utvikle det. Gruppen laget tidlig i prosjektet en prosjektplan som fordelte arbeidet over fire sprints, pluss en pre sprint for planlegging. Oppgavene ble utarbeidet i tråd med brukerhistoriene og product backloggen ble implementert i Azure DevOps. Rutiner for kvalitetssikring av koden ble også avtalt.

For å danne prosjektplanen er det viktig å ha tett dialog med oppdragsgiver for å få informasjon om hvordan de ønsker at prosjektet skal bli. Dette løste vi med å ha flere planleggingsmøter med Læringsbanken, hvor vi så på skisser av hvordan de ønsket at nettsiden deres skulle se ut. De sendte oss brukerhistoriene som de hadde utarbeidet selv, slik at vi kunne innblikk i hvordan deres visjon for Læringsbanken er.

Vi avtalte også faste veiledningsmøter med Læringsbanken sine utviklere en gang i uken, for å få veiledning til det tekniske aspektet av prosjektet. Som tidligere nevnt hadde vi ikke så

mye erfaring med mange av de digitale verktøyene vi brukte i dette prosjektet fra tidligere, så derfor har disse møtene hjulpet oss mye gjennom utviklingsprosessen. I tillegg til dette ble vi enig med både Læringsbanken og veilederen vår om å ha et styringsmøte i slutten av hver sprint. Her skulle arbeidet som har blitt gjort i den tidligere sprinten bli gjennomgått og gruppen får tilbakemeldinger på hva som kan forbedres i neste sprint. Scrum-master hadde ansvaret for å kalle inn til og organisere disse møtene.

Utover dette ble det noen flere møter da Læringsbanken ønsket å sjekke status eller om de hadde noe de ønsket å tilføye til prosjektet. Dette er i tråd med at vi har en agil prosjektstyring og det skal være like enkelt for oppdragsgiver å henvende seg til oss som det er for oss å henvende oss til dem.

3.1.2. Forventinger til gruppearbeidet

Under planleggingsfasen av prosjektet hadde naturligvis alle gruppen forventinger til hvordan gruppearbeidet skulle legges opp og gjennomføres. Vi ønsket alle å oppnå gode resultater og ville tidlig bli enig om tiltak for å sikre dette på best mulig måte. Det var mange nye systemer og verktøy å sette seg inn i starten av dette prosjektet og alle var innstilt på å lære disse på best mulig måte. Som tidligere nevnt delte vi oss også opp i forskjellige ansvarsområder for å utnytte den kunnskapen vi allerede satt på. Samtidig forventet vi at alle i gruppen satt seg inn i og hadde kunnskap om hverandres arbeid. Dette gjorde vi for at alle gruppemedlemmene skal være i stand til å ta over en oppgave dersom uforutsette hendelser skal oppstå, eller at et av ansvarsområdene trenger hjelp med krevende utfordringer. Vi måtte være fleksible samtidig som vi fokuserte på våre arbeidsområder. Dette ble lagt stort fokus på i planleggingsprosessen.

Gruppen vår oppstod i starten av tredje semester og vi har jobbet på lignende prosjekter sammen tidligere. Vi innså derimot at det trengs tiltak etter erfaringer fra tidligere arbeid. Dette prosjektet var mer omfattende enn de forrige vi har hatt og siden vi har en oppdragsgiver i dette tilfellet kreves det at vi har organiserte rutiner for prosjektgjennomføringen vår. Derfor lagde vi en ny gruppekontrakt som inneholdt regler for hvordan vi skal arbeide og oppføre oss mot hverandre gjennom dette semester og denne ligger vedlagt som figur 9.

3.1.3. Tidsstyring

I starten av prosjektet ble alle oppgavene i product backlogen timeestimert. I samarbeid med Læringsbanken så vi på hvor stor tidsramme vi hadde på prosjektet i sin helhet. Vi så også på hvordan vi kunne prioritere oppgavene MoSCoW metoden og hvor krevende oppgavene er. Ut fra det lagde vi et tidsestimat (figur 2) for hver oppgave, med en estimert minimum antall timer og maks antall timer på å gjennomføre den. Dette gir oss et utgangspunkt om hvor lang tid vi kan bruke på de forskjellige oppgavene for å få resultatene til å samsvare med timene som brukes. Tidsestimatet vil hjelpe oss selv med å tilrettelegge arbeidsmengden vår og gi oppdragsgiver innsyn i hvilke oppgaver som er krevende.

Vi estimerte at hvert gruppelem burde ha rundt seks timer tilgjengelig hver ukedag for arbeid på prosjektet. Dette gir oss en total tidsramme for prosjektgjennomføringen på omkring 2850 timer.

Siden dette prosjektet ikke har økonomisk budsjett å forholde seg til, men er bundet opp mot en tidsramme, ble vi tidlig enige om at vi måtte ha noen form for oversikt over den individuelle tidsbruken. Derfor valgte vi å opprette regneark, ett for hver sprint, hvor gruppelemmene loggførte timene sine. For å fylle det ut skal man skrive start, slutt og pause for hver enkelt dag man jobbet. Vi la også inn et felt for eventuelle ting som kan oppstå som påvirker arbeidstimene, som for eksempel sykdom. Alle på gruppen hadde en felles utarbeidet tabell hver for å skrive timer. Eksempel på tidsskjemaet ligger vedlagt i figur 3.

Vi satte opp fire sprinter med et tidsrom på omkring en måned for hver av sprintene. I etterkant ser vi at vi var litt for ambisiøse med timene vi estimerte på hver oppgave opp mot de faktiske timene vi brukte. Vi hadde for store oppgaver og de fleste gikk over de oppsatte timene, noe som kunne bli løst ved å dele opp de store oppgavene.

3.1.4. Brukerhistorier

Vår oppfatning av brukerhistorier er at det brukes som et format for å beskrive oppgavens akseptkriterier og verdier. Det beskriver brukerens rolle, hvilken funksjon som ønskes og hvorfor denne funksjonen er av verdi. Vår forståelse av dette er at det kan brukes for å enkelt kommunisere hvordan systemet ønskes å bli brukt og for å forstå hva brukeren har behov for. Brukerhistorier ble tilbudt av Læringsbanken, da prosjekteiere hadde laget disse tidligere. Vi valgte å beholde disse til prosjektet da gruppen ikke fant det fornuftig å bruke tid på å lage nye. Brukerhistoriene var i et annet format enn gruppen er vant til, men vi valgte å bruke de for å følge brukerhistoriene med høyest prioritert funksjoner. Disse er tilgjengelige i figur 10.

Gjennom prosjektet baserte vi arbeidsoppgavene våre på allerede utviklede brukerhistorier fra Læringsbanken. Dette er tre forskjellige brukerhistorier, som er sortert i prioritert rekkefølge etter ønskene de har for den endelige nettsiden. Det vil si at brukerhistorien som er førsteprioritet er den aller viktigste og de ønskede funksjonene som er beskrevet i den vil være svært ønsket i det endelige produktet. De andre brukerhistoriene vil også være viktige, men disse vil bli nedprioritert under utviklingen av prosjektet for å få inn de viktigste funksjonene først.

Grunnen til at vi valgte å bruke disse ferdige brukerhistoriene i stedet for å lage våre egne var for å hjelpe til med å realisere Læringsbankens visjon om hvordan de ønsker at plattformen deres skal være. Vår oppgave er å videreutvikle nettsiden ved å implementere en innleggsfeed og en profilside, begge disse funksjonene er en del av den høyest prioriterte brukerhistorien. Det vil si at selve prosjektet vårt og oppgavene dette innebærer er utarbeidet med denne brukerhistorien i bakhodet. Dette førte til at de fleste av funksjonene som er har den høyeste prioriteten i brukerhistoriene ble vurdert som "Must have" i MoSCoW metoden vår.

Alle brukerhistoriene ligger vedlagt og brukerhistorien vi valgte å bruke som utgangspunkt gjennom prosjektgjennomføringen er vedlagt som figur 10 og 11.

3.2. Design

Produktet er designet med det eksisterende designet i bakhodet. Produkteier gir hyppige design-innspill, slik at plattformen blir designet som ønsket. Nye deler av systemet følger derfor utseendet under utviklingen. Vi har gjort tiltak for å utføre denne prosessen, eksempelvis er Benyons 12 designprinsipper. Dette er fordi til tross for at designprinsippene kan virke selvforklarende, som for eksempel prinsippet om respons. Dette kan eksempelvis gjenspeiles i systemet i knapper, navigasjonsbar og “footer”, i bunnen av nettsiden. Disse tingene skaper en respons, slik at brukeren forstår hva som skjer.

Vi valgte å hovedsakelig basere oss på brukergrensesnittet som Læringsbanken allerede bruker. Dette gjorde vi hovedsakelig med tanke på å gjøre videreutviklingsprosessen så enkel som mulig for oppdragsgiver. Vi mener at å beholde deres brukergrensesnitt gjennom prosjektgjennomføringen vil sikre kvalitet og et brukbart design. Ved å ha sammenheng i designet mellom systemet som allerede eksisterer og systemet vi utvikler, vil det ikke være store forandringer under brukeropplevelsen. Dette er viktig for å sikre flyt og gjøre brukeropplevelsen smidigere.

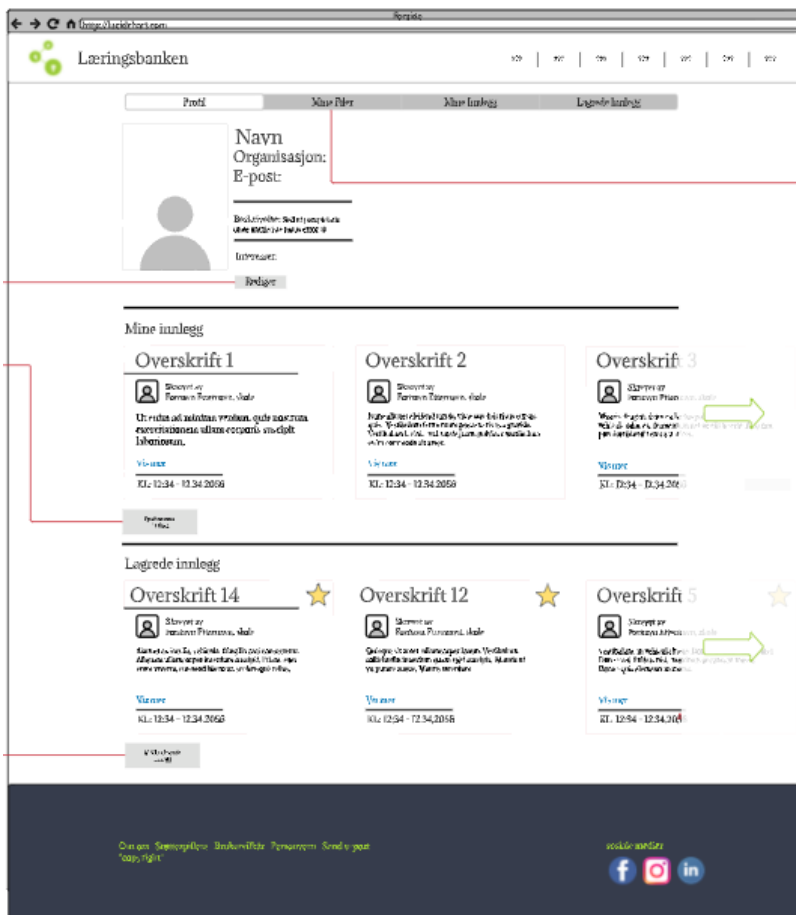
Vi utførte design ut fra det som ble utviklet og underveis tok ønsker fra oppdragsgiver. Noen ønsker tok mer tid enn andre, eksempelvis et ønske om å lage post-it lapper som et grafisk element til alle innlegg. Denne oppgaven ble først delt opp, da dette måtte gjøres på back-end i motsetning til front-end. Denne oppgaven ble til slutt nedprioritert.

3.2.1. Skisser

Ettersom systemet var under utvikling da gruppen ble involvert i prosjektet, ble vi godt supplert med skisser fra arbeidsgiver. De hadde allerede utviklede skisser for hvordan de ville at deres system skulle se ut. Dette ble supplementert av Læringsbanken og fungerte som inspirasjon og utgangspunkt for prototypen og systemet. Vi valgte derimot å ta en litt annerledes vri på prosjektet enn det vi fikk i skissene. Dette var mest fordi deres design var svært mye mer avansert enn det vi hadde tid og ressurser til å implementere. Vi utviklet derimot en low-fidelity prototype om hvordan vi så for oss at vårt system kom til å bli, slik at Læringsbanken fikk et inntrykk av dette. Vårt produkt kan sannsynligvis bli videreutviklet etter at vi er ferdig med det, da mesteparten av selve funksjonaliteten for deres skisser blir implementert av oss. Skissene befinner seg under vedlegg, som Figur 11.

3.2.2. Prototype

Vi begynte utvikling av low-fidelity UX prototype tidlig i februar. Dette var på etterspørsel av arbeidsgiver, da de ønsket en kombinasjon av prototype, high fidelity skisser med forklart funksjonalitet. Prototypen benytter også røde piler for å visualisere funksjoner og navigering på nettsiden. Etter denne ble fullført tok gruppen en gjennomgang med oppdragsgiver og utviklere og forklarte de diverse funksjonene. Et eksempel inkluderes under, som Figur 12. Resterende deler av prototypen vil inkluderes under vedlegg, som Figur 12.



Figur 12 - Prototype

Figuren viser ved bruk av røde piler, fra en knapp (med funksjon) til et nytt vindu som illustrerer hvordan denne funksjonen og knappen vil reagere. To røde piler som kommer fra venstre, utenfor bildet, peker mot et innlegg. Dette innlegget har da to røde piler, en som peker på og viser til funksjonen “del innlegg” og hvordan den funksjonen vil se ut. Den andre pilen viser til funksjonen “rapporter innlegg” og en får da se hvordan denne funksjonen vil se ut. Ettersom dette er prosessen gjennom hele prototypen fungerer det også som et kart over hele systemet.

En viktig del av prototypen er at den skal vise alle planlagte funksjoner i tillegg til små deler og funksjoner som ikke ville blitt vist i eksempelvis wireframe. Et eksempel på dette er hver gang det blir vist et tekstfelt kan en se verktøylinje for tekst. Denne inneholder muligheter for å formatere teksten man skal skrive, ved å gjøre den fet, kursiv eller lignende. Dette er en viktig del av prototypen, ettersom det er en funksjon som ønskes å implementeres men ikke nødvendigvis ville blitt vist i en prototype.

3.3. Testing

Underveis i prosjektet er testing en veldig viktig del av å gjennomgå over koden for å kvalitet sikre arbeid vårt og for å sjekke over for eventuelle feil eller bugs. Med bruk av white box og black box testing som ble nevnt i 2.3.3 fikk vi gjennomgått koden grundig gjennom før vi sender det videre til bli oversett av andre i gruppen før det merges til bachelor branchen.

Læringsbanken har et godt forhold til sin brukerbase og tilbyr derfor god og rask testing av systemet. De har nylig begynt med lukket beta-testing av nettsiden en begrenset brukergruppe. Dette vil fungere som et kvalitetssikrende tiltak, fordi brukerne kan prøve seg frem og rapportere forslag til endringer og feil med systemet.

Under planleggingsfasen hadde vi og Læringsbanken noen tanker om å kunne bruke brukergruppen av lærere til testing underveis i prosjektutviklingen. Der vi kunne gjennomføre en liten brukertest av funksjonalitetene ved slutfasen av bachelorprosjektet for å se om funksjonene som blir lagt til samhandler godt mellom Læringsbanken og brukerne.

3.4. Implementering

Før vi begynte implementeringsfasen av prosjektet, satte vi oss ned sammen med oppdragsgiver og planla hvilke funksjonaliteter som skulle inkluderes. Videre godkjente de også prototypen. Gruppen satte raskt i gang med å lære oss hvordan C#, ASP.NET og Blazor fungerer, ettersom gruppen hadde lite erfaring med dette fra tidligere. Vi måtte også finne ut mer om hvordan vi kunne trekke ut informasjonen om brukeren av systemet fra Feide API-en.

Gjennom sprint 3 og 4 fortsatte vi med implementering av brukerhistoriene Læringsbanken hadde laget og rangert sammen med oss i begynnelsen av pre-sprinten og sprint 1. I sprint 3 fikk gruppen kommet i gang med utvikling av funksjonene og utseendet på nettsiden. Ved slutten av hver sprint har vi og oppdragsgiver gått gjennom hva vi har jobbet med i sprinten. Der vi får en tilbakemeldinger fra Læringsbanken på hva de likte med arbeidet og eventuelle endringer de ønsker forandre.

Det som differensierer prosjektet vårt fra andre prosjekter er at dette ikke er en oppbygging av et helt nytt system. Dette er en videreutvikling av et system som allerede eksisterer, der vi fikk tilgang til kildekode som er allerede laget av utviklerne hos Læringsbanken gjennom BitBucket.

Videre laget utviklerne en egen branch som vi arbeidet i. Ved fullført prosjekt vil den branchen ligge tilgjengelig for utviklerne, slik at de kan gå inn å gjøre de endringene de selv vil før de eventuelt bestemmer seg for å implementere branchen i systemet. Denne løsningen sikrer at det prosjektet vi utvikler ikke har noe påvirkning på Læringsbankens nettside før de selv ønsker det.

3.5. Kommunikasjon

For å forsikre et godt styrt prosjekt bestemte vi oss tidlig for å opprettholde hyppig kommunikasjon, både med oppdragsgiver og innad i gruppen. Etersom COVID-19 pandemien satte en stopper for de fleste typer samlinger, måtte gruppen ta programmet Discord i bruk. Dette gjorde at dersom to eller flere skrev på rapporten samtidig, kunne en drøfte sammen. Dersom en arbeidet koding og trengte hjelp kunne andre komme inn i samtalen og dele skjerm med hverandre. Dette var et godt tiltak for kvalitetssikring. Discord ble også jevnlig brukt for å kommunisere med arbeidsgiver og veileder.

Etter flere semestre av samarbeid med samme gruppe, er alle klar over hvor viktig kommunikasjon kan være. Vi satt derfor en regel om lavterskel kommunikasjon, som også gjaldt for vår oppdragsgiver, Læringsbanken. Innad i gruppen ved hjelp av Discord, lagde vi egne kanaler til hvert gruppemedlem som representerer kontorer. Her kan gruppemedlemmene sitte når de jobber, slik at de andre gruppemedlemmene kan se hvem som er mest tilgjengelig der og da.

For å avtale møter og fastsette en agenda ble det sendt møteinnkallelse til arbeidsgiver og veileder. Dette ble skrevet og sendt av Scrum master. Møteinnkallelser inneholder informasjon som hvem som ønskes tilstede, hva som skal diskuteres og hvor møtet tar plass, som i vårt tilfelle var over Discord. Under møtene ble det skrevet møtereferater, som inkluderes under vedlegg i figur 13 og 14. Gruppen verdsetter møtereferat meget høyt, ettersom det kvalitetssikrer kommunikasjon da en kan enkelt tilbake på hva som har blitt sagt og hvilke tilbakemeldinger gruppen får.

Den gode kommunikasjonen med både Læringsbanken og oppdragsgiver ledet til utviklingen av en applikasjon som kombinerte kvalitet og funksjon på en. Kommunikasjonen mellom gruppen og oppdragsgiver var som regel uformell og lavterskel. Vi hadde noen fysiske møter med hele teamet i Læringsbanken tidligere i semesteret og ideelt sett hadde dette vært noe vi kunne fortsatt med gjennom hele prosjektgjennomføringen. Heldigvis har vi klart å opprettholde god kommunikasjon med dem på tross av at det meste av arbeidet har blitt utført hjemmefra.

De viktigste tiltakene vi gjorde for å opprettholde kommunikasjonen både innad og utad gjennom prosjektet var:

- Styringsmøter med både Læringsbanken og veileder
- Møter med Læringsbanken for justeringer i prosjektet
- Sprint planning meeting, daily Scrum og weekly Scrum med gruppen

Gruppen opparbeidet seg også mer kunnskap og erfaring gjennom prosjektet, noe som gjorde at vi ikke ble fullt så avhengige av hjelp fra oppdragsgiver. Likevel har det vært svært viktig å holde utviklingsprosessen agil ved å kommunisere jevnlig med dem, for å få innspill og eventuelle justeringer. Dette har ført til en god og åpen prosjektgjennomføring, til fordel for både oss og oppdragsgiver.

3.6. Scrum

Scrum er et teambasert utviklingsrammeverk som bygger på gjennomføre gitte oppgaver innenfor et visst tidsrom, kalt sprinter. Under planleggingen av prosjektet bestemte vi oss for å spre prosjektet ut over fem sprinter; en pre-sprint og fire sprinter deretter. Scrum-masteren vil ha det overordnede ansvaret for oppgavene som gjøres sørge for en kontinuerlig dialog blant gruppemedlemmene og med oppdragsgiver. Vi er godt kjent med Scrum-rammeverket før og det har blitt brukt til vellykkede prosjekter for gruppen tidligere. Rammeverket er også veldig transparent og gir både oppdragsgiver og veileder innblikk i hva som foregår. Disse faktorene gjorde avgjørelsen med å bruke Scrum også i dette prosjektet veldig mye enklere.

Gjennom hele prosjektgjennomføringen har det vært viktig for oss å ha god kommunikasjon og utfyllende planlegging, både innad i gruppen og med oppdragsgiver. Scrum-rammeverket gjør dette mye lettere for oss, ettersom vi alltid vet hva som kan forbedres for å få best mulig resultat. For å sikre dette har vi blant annet hatt hyppige planleggingsmøter, både innad i gruppen og med oppdragsgiver. Etter hver sprint holdt vi et styringsmøte (sprint review) med både veileder og oppdragsgiver, hvor vi gikk gjennom arbeidet som har blitt gjort i sprinten. Her fikk vi tilbakemeldinger og innspill på arbeidet som ble gjort i tillegg til tips på hva vi kunne gjøre for å forbedre prosjektet videre.

I tillegg til pre-sprinten hadde vi også sprint planning meeting i starten av hver sprint. I disse møtene er kun gruppen til stede og Scrum-masteren hadde ansvar for å styre også disse møtene. Her blir møtereferatet gjennomgått og tilbakemeldingene fra styringsmøtet blir prosessert. Disse tilbakemeldingene skal utbedres og i de fleste tilfellene førte disse tilbakemeldingene til at vi måtte gjøre endringer i planene for neste sprint. Sprint planning møtene er også en viktig del av det å jobbe agilt, da oppgavene og målene vi kom frem til i pre-sprinten mest sannsynlig har endret seg gjennom prosjektutviklingen. Ved å ha både styringsmøter og pre-sprint møter kan vi tilpasse utviklingen vår etter tilbakemeldingene som kommer og legge opp sprintene etter potensielle endringer i produktet eller prosjektgjennomføringen.

Utenom styringsmøtene og pre-sprintene har vi også hatt weekly Scrum innad i gruppen hver uke. De fleste av oppgavene vi setter opp i pre-sprinten gjennomføres individuelt gjennom prosjektet og derfor er det viktig med møter for å samle gruppen og sjekke status på både oppgavene og gruppemedlemmene. Her går vi gjennom hva som har blitt gjort uken før og tar opp og håndterer eventuelle problemer som kan ha oppstått.

Styringsmøter, sprint planning meeting, daily- og weekly Scrum er alle nøkkelfaktorer til å sikre kvalitet i prosjektet. Derfor har vi sørget for å opprettholde kommunikasjonen både innad og med både oppdragsgiver og veileder og terskelen for å stille spørsmål fra begge sider har vært veldig lav. Dersom prosjektet skulle blitt gjennomført uten kontinuerlig input fra oppdragsgiver eller med dårlig kommunikasjon og arbeidsfordeling i gruppen, ville resultatet av prosjektet vært helt annerledes.

Vi mener at det å bruke en agil prosjektstyring ved hjelp av Scrum-rammeverket var svært effektivt og riktig i vårt prosjekt. Disse fungerte på en god måte og førte til en veldig åpen og lavterskel dialog mellom oss, veileder og oppdragsgiver. Agiliteten i prosjektet gjorde at vi var mye mer fleksible for potensielle endringer og utfordringer som kunne oppstå samtidig som vi hadde en allerede godt planlagt prosjektgjennomføring. Dersom vi skulle gjort noe annerledes til neste gang hadde vi tatt litt mer hensyn til slike potensielle endringer og utfordringer allerede i pre-sprinten. I etterkant av prosjektet ser vi at vi ble litt for ambisiøse på hvor mye arbeid vi kunne få unnagjort tidlig i prosjektet, noe som førte til at enkelte av oppgavene ble dyttet videre fra sprint til sprint. Dette er upraktisk for prosjektstyringen og kan føre til at oppgavene blir større og mer tidkrevende enn agile.

3.6.1. Sprint-mål

Prosjektet vårt ble som tidligere nevnt delt opp i fire sprinter, i tillegg til pre-sprinten. Hver sprint varte i omkring 1 måned, med unntak av sprint 4. Grunnen til dette er fordi den ikke er like omfattende da den kun dreier seg om ferdigstilling av prosjektet. Datoene sprintene var delt opp i gjennom semesteret er som følger;

- Pre-sprint: 06.01 - 01.02
- Sprint 1: 02.02 - 01.03
- Sprint 2: 01.03 - 26.03
- Sprint 3: 05.04 - 01.05
- Sprint 4: 01.05 - 30.05

Sprint 1 baserer seg mest på å få lagt grunnmuren i prosjektet og innebærer blant annet utarbeiding av low-fidelity prototype og wireframes. I tillegg til dette var også sprint 1 en slags utforskningsfase for C# og de andre digitale verktøyene vi ikke hadde erfaring med fra tidligere. Derfor var målet med sprint 1 å bli kjent med verktøyene som skulle brukes og å legge et grunnlag for videre utvikling i form av for eksempel å opprette nye brancher.

Sprint 2 og 3 innebar hovedmengden av utviklingen som ble gjort. Her hadde vi som mål å utvikle et helhetlig produkt og få implementert alle must-have oppgavene. I disse sprintene var det også viktig å ha hyppige møter innad i gruppen for å sørge for at arbeidsfordelingen fungerte. De fleste av must-have funksjonene ble implementert i sprint 2 og deretter videreutviklet og designet i sprint 3.

Sprint 4 var den korteste sprinten og varte i underkant av to uker, her var hovedfokuset å finpusse og ferdigstille produktet samt få de siste innspillene fra både oppdragsgiver og veileder på ting vi kan endre på før endelig innlevering. I slutten av sprint 4 hadde vi også et møte med hele teamet til Læringsbanken hvor vi gikk gjennom og evaluerte hvordan prosjektgjennomføringen har gått.

Vi klarte derimot ikke alltid å oppnå de ønskede målene i hver sprint, noe som førte til at enkelte av oppgavene vi hadde satt opp som mål i en sprint måtte bli overført til neste sprint. Dette førte til at en del av arbeidet måtte bli tatt igjen, men likevel fikk vi implementert alle

must-have oppgavene våre. Dersom vi skulle gjennomført prosjektet igjen ville vi lagt mer press på å få gjennomført oppgavene i de sprinteren de var satt opp i, slik at vi også hadde hatt tid til å implementere flere av “could-have” oppgavene.

3.6.2. Scrum-master

Tidlig i planleggingsprosessen ble det fastslått at et medlem skulle være Scrum-master. Scrum-masteren vil ha overordnet ansvar for arbeidsflyten i prosjektet og sørge for at arbeidsfordelingen fungerer på en god måte. Han har også ansvaret for å sette opp sprinter og kalle inn til sprint-møter.

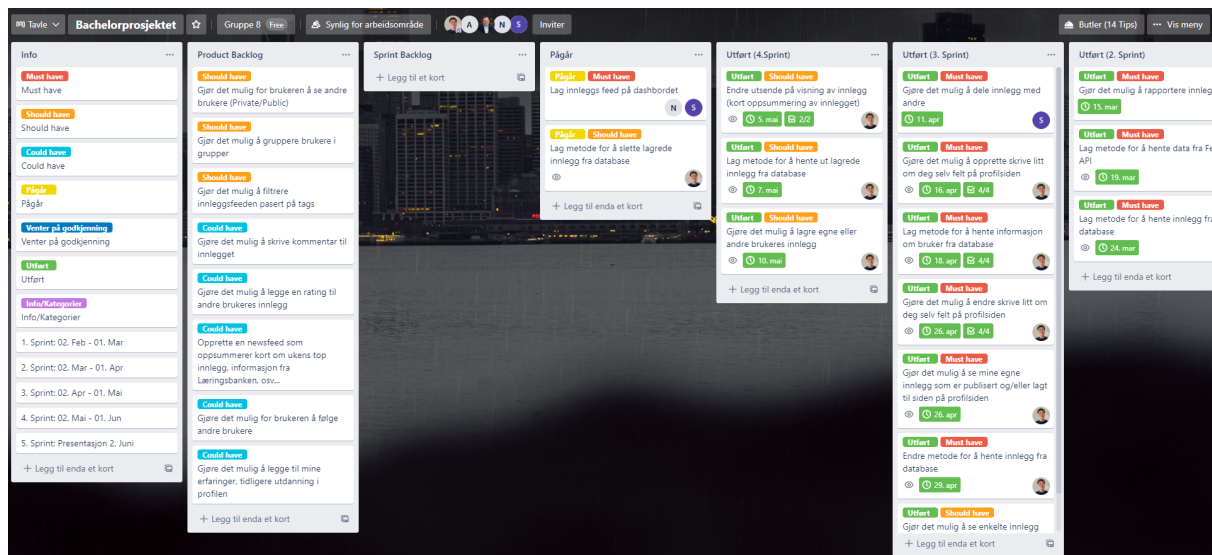
I vårt prosjekt har vi også gjennom hele prosjektet hatt dialog med to forskjellige veiledere, oppdragsgiver og veileder. Scrum-masteren fungerer som et slags knutepunkt mellom disse og arbeidet som gjøres og gjør Scrum-prosessen mer tilgjengelig for begge parter ved hjelp av for eksempel styringsmøter. Dette bidrar til bedre arbeidsflyt og sørger for at oppgavene formes etter oppdragsgivers ønsker og veileders veiledning.

Vi har under tidligere prosjekter rullert på rollen som Scrum-master, men ettersom det til tider kunne være kaotisk valgte vi å beholde ett medlem som hadde rollen gjennom hele prosjektet. Gruppen ble enig om å ha Christer som Scrum-master, for han har tidligere erfaring med ansvarsfordeling og oppfølging og ønsket derfor å ta på seg dette ansvaret.

En slik rolle som dette er avgjørende for arbeidsflyten gjennom prosjektet og det er enklere for både medlemmene i gruppen og oppdragsgiver å henvende seg til en Scrum-master dersom det skulle oppstå problemer som går utover Scrum, som for eksempel utsatt tidsfrist eller endringer i oppgavene.

3.6.3. Product backlog

Product backlogen vår består av funksjoner og oppgaver som skal bli implementert i prosjektet og kan ses som figur 15 i gruppens Scrum board på Trello. Disse har beskrivende navn som er basert på brukerhistoriene vi fikk av Læringsbanken og funksjonaliteten de skal ha i sluttproduktet. For å få bedre oversikt over oppgavene er de også gradert med en rød, gul eller grønn “tag” etter prioritering i MoSCoW metoden.



Figur 15 - Scrum board og product backlog

For funksjonene og brukerhistoriene som skulle implementeres på en god måte var det viktig at backloggen ble velstrukturert. Flere av oppgavene hadde også flere sider av seg som må utvikles for å passe til brukerhistoriene. Dette løste vi ved å ha flere “underoppgaver” under selve oppgavene. Disse inneholdt betingelser og vilkår fra oppgaven som må være akseptert for at vi kunne sette oppgaven som fullført.

Vi valgte også å opprette oppgaver som omfatter bachelor rapporten i backloggen vår, selv om disse ikke har noe med selve produktet å gjøre. Grunnen til dette var at vi ønsket en oversikt over hvor langt vi ønsker ha kommet i rapporten i slutten sprint og det har fungert på en god måte. Disse er også markert med egne farger i product backloggen for å tydeliggjøre at disse ikke har noe med programmering å gjøre, slik at de som er ansvarlig for programmeringen ikke trenger å fokusere på disse.

Oppgavene varierte veldig i hvor omfattende de var, men de fleste var store nok til å kunne gjennomføres på en uke. Oppgavene er også tidsestimert i forkant og kunne endres dersom det skulle være avvik i forhold til beskrivelsen. Enkelte kunne også gjennomføres på et par dager. Det var derimot noen som var svært omfattende, som for eksempel utviklingen av forsiden og profilsiden i sin helhet. Vi forstod fort at disse oppgavene måtte bli med gjennom alle sprintene, ettersom de var svært tidkrevende. I figur 15 ser du den fullstendige versjonen av product backloggen vår, inkludert “underoppgaver”.

3.6.4. Sprint backlog

Sprint backlog har vært en effektiv måte å visualisere hva som mangler for å fullføre sprinten. Vi kunne også se hvilke oppgaver som har blitt fullført, som i kombinasjon med sprint backlog gir et godt overblikk til hvordan vi lå an i sprinten. Et problem med dette var størrelsen på oppgavene i sprinten. Dette skapte en flaskehals, da det ble for mye tid brukt på for store oppgaver. Etter fullføring av Must have oppgavene som ble satt, innså gruppen at MoSCoW-modellen burde blitt produsert på en annerledes måte ettersom noen av must

have-oppgavene ble litt for store i seg selv. Profilsiden for eksempel inneholdt veldig mange mindre oppgaver og tok opp store deler av semesteret. Derfor ble vårt største problem i sprint backloggen at vi endte opp med å forskyve oppgavene fra sprint til sprint. Vi ser i ettertid at vi skulle delt oppgavene mer opp og vært forsiktede med hvor mange oppgaver vi inkluderte i sprint backloggen.

3.6.5. Daily- og weekly Scrum

Tidlig i prosessen var alle enige om at det er viktig med god kommunikasjon i gruppen. For å forsikre god kommunikasjon og arbeidsflyt valgte vi å dele opp gruppen i to hvor en del hadde fokus på utvikling og en annen del hadde ansvar for rapporten. Dette i kombinasjon med at alle i gruppen byttet på oppgaver, for å gi alle en god oversikt over prosjektet. Derfor hadde Christer daily Scrum med Andreas under arbeid med rapporten. Dette innebærer daglige møter hvor arbeidsmengde og arbeidsplan blir gått igjennom. I tillegg ble det hyppig kommunikasjon med gruppemedlemmene som hadde fokus på utvikling. Dette var viktige tiltak for å sikre at alle gruppemedlemmer er på samme side.

Minimum en gang i uken holdt gruppen et weekly Scrum, hvor alle i gruppen oppdaterte hverandre på hvordan arbeidet gikk. Dette ukentlige møtet hadde varierende varighet, ettersom kommunikasjonen var hyppig nok til å sikre en god oversikt over prosjektet. I tillegg til dette ble det tidligere nevnte tiltaket om å bruke egne digitale kontorer over Discord, en meget god ordning for å sikre kommunikasjonen. Disse møtene ble det ikke skrevet referat fra.

Her stiller gjerne Scrum-master spørsmål som:

- Hva har du oppnådd siden forrige møte?
- Hva skal du gjøre frem til neste møte?
- Har du støtt på noen utfordringer?

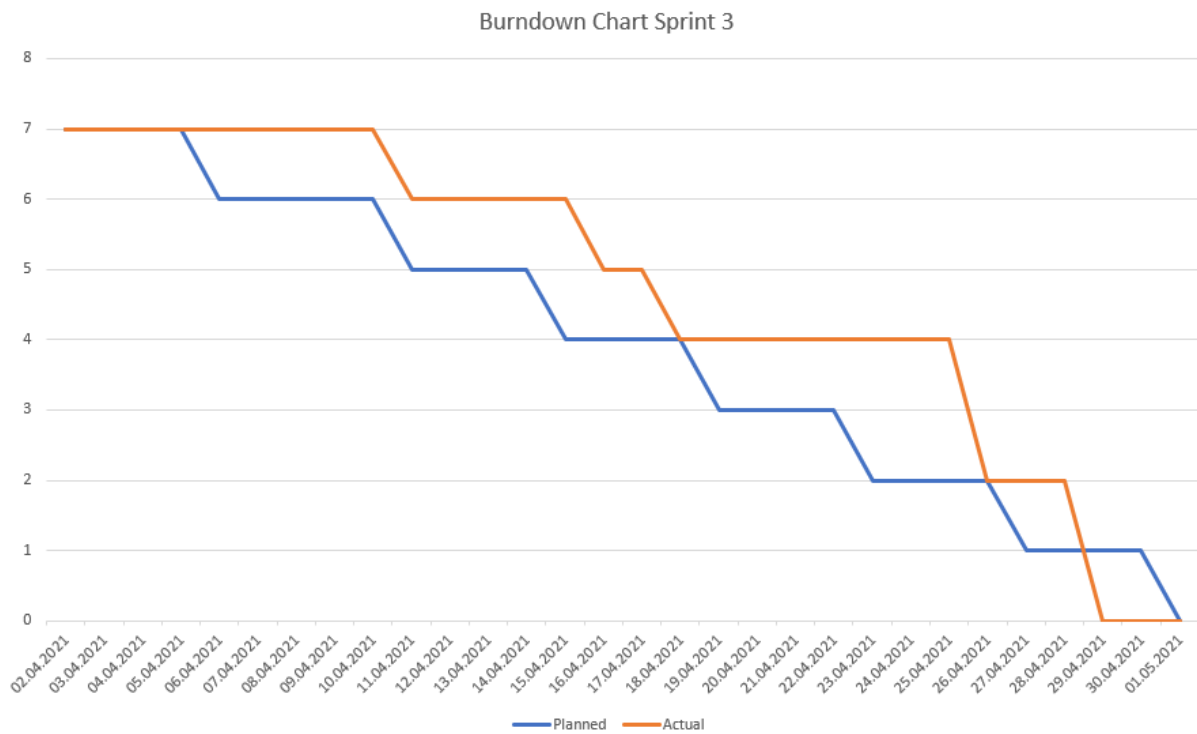
I tillegg til disse jevnlig møtene med gruppemedlemmene, ble det holdt styremøter i slutten av hver sprint. I disse møtene deltar både oppdragsgiver og veileder, og får en statusoppdatering fra prosjektutviklingen. Etter disse møtene blir det skrevet referat slik at all informasjon fra arbeidsgiver angående prosjektet og eventuelle forandringer og tanker blir inkludert, og veileder med ekstra informasjon. Scrum møter og styremøter er et viktig verktøy for å skape en transparent arbeidsprosess og ettersom gruppen opprettholder fordelte ansvarsområder er Scrum meget viktig for å sikre en god balanse i arbeidet.

3.6.6. Burndown chart

Burndown chart er en grafisk måte å vise hvilke oppgaver en har samtidig som tidsestimering for å kunne gi en visuell representasjon av arbeid som skal bli og har blitt utført. Burndown chart er ofte brukt i sammenheng med Scrum. I vårt tilfelle har vi slått sammen burndown chart med vårt "Scrum board". Dette er en mer delt opp versjon av burndown chart men med meget lik funksjonalitet i form av tidsestimering. Scrum board gir oss mulighet til å dele opp oppgaver, sammen med tidsestimering og kan ses i figur 15. Deretter la vi de individuelle oppgavene inn i en rute som beskriver når det skal bli ferdig. Dersom det ikke er fullført

innen tiden, må man skrive hvor lang tid som er brukt slik at en kan sammenligne faktisk tid som er brukt og estimert tid.

Burndown chart er et meget godt verktøy for å visualisere arbeid utført i forhold til estimert arbeid utført. Dette kan gi innsikt til eksempelvis arbeidsmengde, tid til arbeid, for store oppgaver og lignende. Vi lagde burndown charts for hver enkelt sprint og disse ligger vedlagt som figur 16, 17 og 18.



Figur 17 - Burndown chart, sprint 3

4. Resultat

I denne videoen demonstreres funksjoanliteten av innleggsfeeden og profilsiden.

Link til videoen: <https://youtu.be/IqJknzkP5T0>

5. Refleksjon

I denne delen av rapporten vil vi reflektere over hvordan gjennomføringen av prosjektet har fungert. Vi redegjøre for på hva som har fungert, hva som ikke har fungert og læringsutbyttet vårt fra prosjektgjennomføringen.

5.1. Forventninger

Før semesterets oppstart gjorde gruppen visse forberedelser for å forsikre et godt planlagt prosjekt. Det ble satt møter med Læringsbanken som satte gode forventninger, spesielt når det gjaldt kommunikasjon. Gruppens forventninger til hverandre ble hyppig diskutert, da alle ønsket å oppnå et godt prosjekt og spesielt god rapport. Disse forventningene ble tatt opp tidlig underveis ettersom gruppen ønsket å sette gode rutiner. Dessverre ble disse forventningene ikke helt møtt, da vi ble møtt med utfordringer fra tidlig av, eksempelvis et sykemeldt gruppemedlem.

Det ble også satt forventninger til arbeidsinnsats. Disse forventningene utviklet seg og ble forandret underveis, hvor en generell utmattelse traff gruppen. Dette var en naturlig utvikling og noe som var forventet i henhold til risikoanalysen. Vi forventet å komme i gang med prosjektet ganske raskt, men ettersom vi ikke fikk tilgang til Læringsbankens systemer før taushetsavtalen var signert av begge parter ble utviklingsstarten forsinket. Konsekvensen av dette ble en betydelig redusering av tidsrommet som var satt opp for sprint 1.

5.2. Prosjektplan

Fra starten av prosjektet har gruppen alltid hatt fokus på hvordan prosjektet skal styres og gjennomføres på best måte. Dette startet med å tidlig bestemme en gruppeleder og en Scrum-master. Dette innebærte at Christer meldte seg frivillig til denne rollen. Andreas ble ansvarlig for prosjektdokumentasjon, Arian tok ansvar for å være gruppeleder med fokus på front-end utvikling. I starten av prosjektet prøvde vi å involvere alle på systemutviklingen og kun legge fokus på dette. Vi innså derimot tidlig at dette kom til å bli vanskelig ettersom vi også hadde en rapport som skulle skrives, i tillegg til et avansert forskningsprosjekt i IS-305. Det ble da naturlig at når flere arbeidet med prosjektdokumentasjon samtidig, arbeidet de tett og hadde stort sett konstant kommunikasjon sammen. For å kunne arbeide parallelt med utviklerne var det enklere at utviklerne ble involvert i prosjektdokumentasjonen nødvendigvis, om ikke utviklerne allerede har dokumentert sitt arbeid. Vi så oss derfor nødt til å dele opp gruppen i forskjellige grupper for å få utarbeidet de forskjellige delene i prosjektet parallelt, henholdsvis i en utviklingsgruppe og en dokumentasjonsgruppe.

Balansen innad i gruppen skapte en forholdsvis god oversikt for gruppemedlemmene. Dette er noe gruppen gjerne kunne gjort annerledes, til tross for at alle hadde oversikt kunne nok alle hatt bedre kontroll. Dette var noe som ble tatt litt mer hånd om senere i prosjektet. Da valgte gruppen å sette systemutvikling til sides for å fokusere på rapporten, fremfor at de som arbeidet med utvikling byttet mellom utvikling og rapportskrivning. Det ble tidlig klart at det skulle jobbes med frihet under ansvar. Dette førte derimot til at flere av gruppemedlemmene fikk skylapper og mistet fokus på de andre ansvarsområdene utenom deres egne. I ettertid

opplevdes dette som et problem, til tross for hardt arbeid, da vi kunne tatt mer ansvar selv utenfor våre ansvarsområder. Dette hadde gjort skiftet i fokus fra koding til rapport mye enklere.

De positive sidene av planleggingen vi gjennomførte var at ansvarsfordelingen fungerte fint og ble opprettholdt gjennom hele prosjektet. På tross av at gruppemedlemmene mistet litt fokus på hverandres ansvarsområder, sørget ansvarsfordelingen for kvalitet og fremgang på alle delene av prosjektet. Ved hjelp av den satte vi også standarder for både de tekniske og de praktiske aspektene av prosjektet. Scrum-delen av prosjektplanen vår fungerte også fint og vi klarte å opprettholde de alle møtene som ble planlagt. Prosjektplanen har også hjulpet oss med å jobbe jevnt og ingen av delene i prosjektgjennomføringen vår har blitt et skippertak.

I etterkant ser vi derimot at vi ikke har klart å oppfylle alle elementene i den prosjektplanen vi la, spesielt når det kommer til forventet arbeidsmengde per gruppemedlem. Vi burde vært mer realistiske når det kommer til hvor mye tid man har til disposisjon hver dag og den forventede arbeidsmengden vi satte var alt for høy i forhold til det som faktisk ble resultatet. De felles arbeidsøktene vi planla ble raskt for ulike for å opprettholdes som et resultat av at vi fordelte oss i team. Vi ønsket å holde de forskjellige arbeidsprosessene parallelle, men dette viste seg å være utfordrende da gruppemedlemmene la inn ulik arbeidsmengde i prosjektet til tider. Dette førte til at de forskjellige ansvarsområdene til teamene kunne oppleve mye mer fremgang enn de andre, basert på hvor mye innsats som faktisk ble lagt inn.

Ved bruk av Scrum-rammeverket organiserte gruppen og Scrum-master en plan for uken. Dette ble gjort etter utført MoSCoW-modell, slik at denne gir et innblikk til full prosjektutførelse og en oversikt over sprinten. Etter sprint ble det avtalt styremøte som innebærte at gruppen møtte med oppdragsgiver og veileder. Disse møtene fikk i ettertid referat skrevet som vil inkluderes i appendiks. Etter fullført møte og sprint tar Scrum-master og resten av gruppen videre til arbeid. Da kan sprint backlog og resterende oppgaver som passer i henhold til MoSCoW-modellen bli fullført.

Dersom vi skulle gjennomført prosjektet igjen finnes det prinsipper og metoder vi burde ha involvert tidlig i prosjektet. Et eksempel på dette er Googles designprinsipper kunne vi utnyttet for å forbedre brukeropplevelsen og designet av brukergrensesnittet i systemet vårt. Disse prinsippene kunne ha vært en inspirasjon for utviklingen av en enklere brukeropplevelse av nettsiden. Hovedgrunnen til at vi ikke kunne brukt dette prinsippet er fordi Læringsbanken allerede har et design på deres side som vi følger og store endringer i brukergrensesnittet kunne blitt utfordrende for dem med tanke på videreutvikling. Vi er uansett fornøyd med resultatet vårt og mener at systemet vi har utviklet har tilstrekkelig brukervennlighet. Veileder informerte oss om Googles designprinsipper under det første styringsmøtet, men da var vi allerede godt i gang med utviklingen. Prinsippene var noe vi selv burde tenkt på tidlig i prosessen og ville vært et utmerket supplement til å utvikle designet.

5.3. Prosjektstyring

Vi valgte å bruke Scrum-rammeverket sammen med agil styringsmetode gjennom hele prosjektgjennomføringen vår. Scrum har vi mye erfaring med fra tidligere prosjekter, men å jobbe agilt var ganske nytt for oss. Vi har som regel jobbet med den tradisjonelle waterfall-metoden, men for å sikre åpenhet ble vi tipset om å heller jobbe mer agilt. Dette er også første gangen vi har jobbet direkte med en oppdragsgiver og grunnen til at vi valgte å jobbe agilt sammen med Scrum var for å sikre framgang parallelt med å opprettholde kommunikasjon.

Effekten av daily- og weekly Scrum møtene har vært svært positive, til tross for at planleggingen kunne blitt utført bedre. Møtene ga godt utbytte, da gruppe medlemmene følte seg godt oppdaterte etter utført møte. Oppdragsgiver ble også informert og ga innspill med fremtidige ønsker, planer og eventuelle andre poeng. Veileder ga gode tips og god informasjon slik at gruppen kunne utføre jobben sin på best mulig måte. Dette var et viktig kvalitetssikringstiltak gjennom prosjektgjennomføringen.

Det ble kun utført to formelle styringsmøter da gruppen ble kontaktet av Læringsbanken for statusoppdatering og eventuelle ønsker. Rundt disse møtene var vi også i kontakt med veileder Janis Gailis og det følte overflødig med formelle møter i tidene rett rundt disse statusmøtene. Etter refleksjon rundt dette ble gruppen enig i at styringsmøtene likevel burde bli prioritert for å gjennomføre prosessen så nøyaktig som mulig. Til tross for denne refleksjonen virket det mer som en formalitet å utføre styringsmøtene for nylig separate møter med oppdragsgiver og veileder.

Sprint planning meeting fungerte også på en god måte. Det å legge opp arbeidsukene på starten av sprinten i stedet for å fastsette alle oppgavene i pre-sprinten har hjulpet oss med å opprettholde agiliteten i prosjektet. Det sikret også til at vi alltid kunne ta de gjenværende oppgavene fra product backloggen inn med høyest prioritet inn i sprint backloggen, slik at vi alltid jobbet med implementeringen av de viktigste funksjonene.

Vi har alle lært masse om Scrum-rammeverket og hvordan dette kan bli brukt. Mye av det vi har lært, er som resultat av å tilpasse oss under utvikling. Blant annet MoSCoW-modellen og burndown chart ble reflektert godt over i ettertid, da vi eksempelvis innså at "Must have"-delen kunne vært større. Burndown chart og Scrum board ble brukt tidlig, men visse ting burde blitt delt mer opp. Dette gjelder noen oppgaver som ble satt opp, da disse var for store. For å enklere kunne orientere seg rundt og fullføre oppgaven, ville Scrum board blitt mer oversiktlig og enklere å fullføre med mindre arbeidsoppgaver.

I stedet for å bruke Scrum-rammeverket kunne vi brukt "waterfall-metoden". Den metoden baserer seg på en lineær sekvens fase. Det gir en oversikt over prosjektaktiviteter og hver fase avhenger av resultat fra tidligere fase. Til tross for at vi innså hvordan ting kunne bli gjort bedre ved bruk av Scrum, hadde vi ikke valgt waterfall metoden dersom det skulle gjøres om igjen. Waterfall metoden kunne fungert ettersom den er mer støttende for ett enkelt prosjekt

og Scrum med agil utvikling kan passe bedre til flere små prosjekter. Den agile utvikling passet oss bedre ettersom den støtter kontinuerlig testing og utvikling, i motsetning til waterfall metoden hvor ting skjer sekvent.

Dessverre har ikke utviklingsprosessen vært så agil som vi hadde ønsket. En direkte konsekvens av dette er som tidligere nevnt at det ble vanskeligere å overlapse hverandres ansvarsområder. Dette kan være som resultat av dårlig arbeidskraft eller for store arbeidsoppgaver. Det som har fungert innen det agile må være den hyppige kommunikasjonen mellom Scrum-master og oppdragsgiver. Dette har ført til raskere utførelse av ønskede forandringer. Mange av problemene innen utviklingsprosessen handler om de tidligere nevnte arbeidsoppgavene.

I ettertid ser vi at det å bruke Scrum-rammeverket har vært en delvis god løsning. Vi beregnet nok tid på hver sprint til å sikre fremgang og møtene vi hadde både innad i gruppen og med oppdragsgiver fungerte fint. Det har derimot også vært flere negative sider av å bruke Scrum. Et eksempel på dette hadde som tidligere nevnt vært å lage mindre oppgaver for å forhindre at store oppgaver ble pushet fra sprint til sprint. Vi mener imidlertid at Scrum har vært det riktige rammeverket å bruke, men at vi kunne og burde utnyttet det bedre.

Dersom vi skulle gjennomført prosjektet ved bruk av Scrum-rammeverket på nytt ville vi gjort oppgavene mye mindre for å få delt opp bedre. For eksempel så ble det å ha en oppgave som heter "Profil" feil, da dette var en annen oppgave som inneholdt implementering av veldig mange forskjellige funksjoner, pluss at front-end og design av siden kom på toppen av dette. Dette resulterte i at vi måtte flytte oppgavene fra sprint til sprint og brukte enda flere timer enn det vi hadde estimert. Neste gang ville vi delt opp oppgavene enda mer, slik at vi kunne gjennomført mindre deler av de store og omfattende oppgavene gjennom hver sprint. Dette hadde resultert i at vi fikk et mer nøyaktig antall timer, da vi kunne beregnet og loggført tiden på de mindre oppgavene hyppigere i stedet.

Det å opprettholde agil prosjektstyring har også vist seg å ha svært positiv innvirkning. Vi har opprettholdt kontinuerlig dialog med oppdragsgiver og endringer i prosjektet har vært lett å implementere. Ved å ikke fastsette det ferdige produktet eller prosessen rundt det var hele tiden prosjektet veldig enkelt tilpasselig. Dette sørget for bedre kvalitet og bidro til kontinuerlige endringsforslag fra både oss og oppdragsgiver for å sikre et best mulig sluttproduktet. Kreative løsninger kom mye enklere frem. Dette er første gangen vi har brukt agil prosjektutvikling overfor en klient og det har ført til mye mer åpenhet. Det viste seg også at fungerte svært bra å jobbe med oppdragsgiver ved bruk av kombinasjonen av Scrum og agil prosjektstyring. Denne kombinasjonen kunne vi brukt igjen og er noe vi burde tenkt på også i tidligere prosjekter.

I løpet av dette bachelorprosjektet har gruppen fått en dypere forståelse av Scrum rammeverket og hvordan det kan brukes til å administrere utviklingsprosjekter. Hele gruppen hadde erfaring med Scrum fra tidligere prosjekter, men dette var det første prosjektet der gruppen var helt selvorganisert og jobbet for en ekte klient som hadde forventninger og krav til sluttproduktet.

5.4. Prosjektet

Prosjektet og utviklingen viste seg å være mer utfordrende enn det vi så for oss før vi begynte. Det Læringsbanken stilte med var et stort og sånn sett “ferdig” produkt, som vi skulle være med å videreutvikle. Noe som ville si at ikke bare måtte vi lære oss nye kodespråk, men vi måtte også sette oss inn i deres kodestandarder og hvordan Læringsbanken strukturerte kode og mapper og lignende.

Utvikling av funksjonalitet har gått veldig greit og vi klarte og holde et greit tempo på dette. Men måten vi fordelte arbeidet på skapte en liten flaskehals når disse oppgavene skulle videre på design. Arian som hadde ansvar for design og front-end, arbeidet på dette tidspunktet med wireframes, som var et prioritert ønske fra læringsbanken. Det gjorde at front-end utviklingen vår gikk litt tregt en liten stund. Men siden wireframes var et prioritert ønske fra så var det viktigere at vi prioriterte dette.

Videre arbeid med design og frontend etter fullførte wireframes gikk veldig greit, da vi kunne jobbe hver for oss og bruke wireframes til å se hvordan vi ønsket at designet skulle være. I ettertid skulle vi ønske at vi fikk ønsket om wireframes, eller at vi bestemte oss selv for å lage det tidligere i prosjektet(helst i pre-sprint), da det tok opp veldig mye arbeidskapasitet og gjorde at programmeringen gikk litt tregere enn vi skulle ønske i en periode.

Design av systemet var tidlig et stort tema. Vi var forberedt på å ta diverse designprinsipper og heuristikker i forbehold, for å oppnå målet vårt. I ettertid har vi innsett at designprinsippene og heuristikken skulle hatt større fokus, da vi mistet litt fokus på dette ettersom det kom ønsker om fokus på spesifikke ting fra arbeidsgiver. Til tross for mangel av fokus, ble designprinsipper og heuristikker fulgt så godt som mulig. Eksempler på disse er heuristikkene om konsistens og standarder og system status m.m. og noen designprinsipper er blant annet synlighet, konsistens og tilbakemelding.

Mesteparten av tiden i prosjektet har vi satt stort fokus med testing av det vi har jobbet med. Gjennom hver oppgave vi gjør har vi satt opp slik at alle må gå gjennom koden sin før den blir sendt videre til andre gruppemedlemmer for å sikre kvaliteten av koden. Gjennom prosjektet har vi for det meste brukt white box og black box testing for å sikre kvaliteten, men vi innså at det ble brukt mye tid når hver enkelte må gå gjennom for hver linje med kode. Vi kunne derfor revurdert om vi skulle brukt noen andre metoder for testing og eventuelt planlagt annerledes for testing med kontrollgruppen som oppgradsgiver har tilgjengelig.

Vi endte på et produkt som vi er fornøyde med. Vi føler at kvaliteten på arbeidet vi har gjort er bra og slik vi har forstått det er arbeidsgiver også fornøyd. Dersom vi skulle gjort et lignende prosjekt igjen ville vi derimot innført ytterligere tiltak for å sikre enda høyere kvalitet. Vi burde ha prioritert å gjennomføre mere enhetstesting av koden, da dette ikke ble gjort i så stor grad som det burde blitt. Gruppen burde også ha satt seg bedre inn i de teknologiske standardene før semesterstart, da vi visste om flesteparten av de tekniske standardene vi skulle bruke før jul, som for eksempel C#. Kurs og opplæring i dette før semesterstart hadde gjort oss bedre forberedt, og ført til at vi hadde brukt mindre av prosjekttiden til å måtte lære dette.

5.5. Risikohåndtering

I starten av prosjektet utviklet vi en risikoanalyse for å forberede oss på og kunne håndtere potensielle risikoer mot prosjektet. En god risikoanalyse har vært spesielt viktig på grunn av pandemien. Tiden med nedstengninger, begrensninger og mindre sosial kontakt har vært utfordrende for de fleste; inkludert oss selv. Dette ble reflektert i risikoanalysen vår, da vi så på psykiske utfordringer som en av de største risikoene mot prosjektgjennomføringen vår. Vi fremhevet spesielt fem risikoer som høye risikoer og disse måtte vi være spesielt forsiktige og forebyggende mot.

For å prøve å forhindre at disse risikoene skulle inntreffe ble vi enige om gode rutiner og retningslinjer tidlig i prosjektet. Vi delte oss inn i team for å sørge for at alle jobbet med noe som de var komfortable med og planla regelmessige gruppemøter for å sikre åpen dialog innad i gruppen. Vi kom fram til at gruppen sitter på mye kunnskap og at det er veldig viktig at vi jobber som en gruppe og bruker hverandre. Alle måtte også ha en viss kontroll og innsikt i hverandres arbeid, slik at dersom noen falt fra var det alltid noen andre til å steppe inn. Dette gav oss alle en trygghet i arbeidet vi gjorde.

Dessverre så vi gjennom semesteret at flere av risikoene ble en realitet for flere av gruppemedlemmene. Mange av oss slet med utmattelse og motivasjonsmangel til tider. Vi ser på pandemien og hjemmekontor som en av de største årsakene til dette. Når man sitter hjemme er det flere faktorer som kan distrahere fra arbeidet som gjøres. Dersom vi hadde hatt muligheten til å møtes og jobbe sammen fysisk, vil distraksjonene bli minimale. Vi hadde også blitt enige med Læringsbanken om å gjennomføre det meste av arbeidet i nærhet til dem, ved å bruke lokalene til Colab på UiA. Virkningen av fysisk samvær er undervurdert, noe vi har merket gjennom denne prosessen.

Motivasjonsmangel kan også være en negativ side av Scrum. Hvis et gruppemedlem blir satt på en oppgave som er for utfordrende til at den kan gjennomføres, eller at man er for psykisk utslitt, kan dette resultere i motivasjonstap. Som nevnt er dette problemer vi alle støtte på, men som vi klarte å håndtere med å omorganisere oss til oppgaver vi faktisk klarte å gjennomføre. Vi tilpasset også arbeidsmengden på hvert gruppemedlem slik at ingen skulle bli overarbeidet og hadde en svært åpen og lavterskel dialog innad i gruppen om status på den psykiske helsen hos hver enkelt.

Vi hadde som tidligere nevnt også et sykemeldt gruppemedlem tidlig i prosessen, noe som vi måtte håndtere rett etter utviklingen av risikoanalysen. Vi hadde derimot planlagt risikoanalysen godt nok til at vi klarte å håndtere det. Flexibiliteten i gruppen var god nok til at vi klarte å justere arbeidsmengden og arbeidsområdene til at prosjektutviklingen gikk videre, selv om vi var en person mindre.

Tekniske feil i prosjektet ble også vurdert som høye risikoer i risikoanalysen. Denne delen av prosjektet gikk ganske fint. Vi hadde selvfølgelig noen små utfordringer på grunn av de fleste av verktøyene vi brukte i systemutviklingen var helt nye for oss. En av de største faktorene som hjalp oss var at Læringsbanken hadde allerede implementerte rutiner og systemer som vi kunne bruke som inspirasjon til vårt prosjekt. Læringsbanken sine egne utviklere var også enkelt tilgjengelige og hjalp oss kjapt dersom vi trengte det.

Vi mener at risikoanalysen vår har blitt utført på en god måte og risikoene vi analyserte har vært svært relevant overfor de uforutsette problemene vi måtte håndtere. Vi mener også at de forebyggende tiltakene vi har gjort for å forhindre disse risikoene har virket. Kommunikasjonen med Læringsbanken har fungert fint og det har ikke vært alvorlige konflikter innad i gruppen. Vi har heldigvis støtt på minimalt med tekniske problemer, men synes likevel at det var viktig at vi vurderte disse potensielle risikoene i forkant. Det er derimot noen risikoer som er så usannsynlige at vi ikke hadde trengt å bruke tid på å vurdere dem, som for eksempel risikoen for dødsfall. Neste gang ville vi brukt mer tid på å vurdere flere faktisk sannsynlige risikoer enn å ha med veldig usannsynlige risikoer.

5.6. Samarbeid med oppdragsgiver

Det har vært meget fornøylig å samarbeide med oppdragsgiver. De har vært lett tilgjengelig, positive og hjelpelige. Det nærmeste til et problem som har oppstått, vil være ved et par forespørsler som heldigvis var greit overkommelig med agil utvikling. Et eksempel på dette er ved utvikling av prototype, da dette ble ønsket ferdig i løpet av få dager. Som nevnt, var dette overkommelig grunnet den agile utviklingen.

Samarbeidet ble ikke som forventet, da gruppen ønsket å arbeide fysisk sammen med oppdragsgiver i nærheten. Dette ble satt en stopper av pandemien, da en måtte begrense fysisk kontakt og tilfeller hvor universitetets lokaler ble stengt. Situasjonen ble taklet så godt som mulig med digitale møter.

5.7. Utfordringer

Under prosjektutvikling vil det alltid oppstå utfordringer, både i og utenfor systemet. Det oppstod problemer underveis, som for eksempel sykemeldt gruppemedlem, lite erfaring med programmerings stacken og en pågående pandemi som førte til at vi måtte jobbe hjemmefra.

5.7.1. Digitale verktøy

Under prosjektet har vi hatt en del utfordringer, både digitale og fysiske. Innen det digitale møtte vi problemer med Azure DevOps. Problemene startet ved vurdering av Trello, da Trello

ikke tilbyr mulighet til å få burndown chart uten betaling. Derfor valgte vi tidlig å bruke DevOps ettersom den tjenesten tilbyr premium funksjoner til studenter uten betaling. Denne forandringen ble utfordrende ettersom ingen hadde erfaring med DevOps. Det ga oss fortsatt muligheten til å utvikle agilt, men ikke til sitt fulle potensiale. Dette var et resultat av oppgavene ikke ble fullført og flyttet på som ønsket og at mange oppgaver var mye større enn de burde vært. Underveis ble det oppdaget at burndown chart som ble tilbudt ikke fungerte som ønsket. Dette i kombinasjon med tidligere lite effektivt bruk av DevOps førte til et bytte til Trello.

Disse problemene burde ikke oppstått, da det tryggere valget fra starten ville vært å fullføre prosjektet med Trello og til slutt lage burndown chart manuelt. Ettersom burndown chart ble laget manuelt i vårt tilfelle uansett ville dette spart oss for mye tid. En annen løsning hadde vært å bli bedre kjent med Azure DevOps, enten ved bruk av en ordentlig gjennomgang og testing eller ved hjelp av Læringsbanken hvor noen hadde mer erfaring. Ettersom tiden er noe begrenset, ville nok hjelp av Læringsbanken vært det optimale valget.

For å følge Læringsbankens system måtte gruppen lære å kode med C# og Blazer. Dette var en gøy utfordring, da ingen hadde erfaring med det. Derfor dedikerte Khai og Sven de to første sprintene til å lære seg C# og Blazer. Denne utfordringen viste seg å være overkommelig ettersom de gjorde seg forberedt og fikk god hjelp fra utviklerne til Læringsbanken. C# har det vært funksjonelt programmeringsspråk å skrive i, siden det meste av syntaxen er veldig lik det vi har brukt tidligere i sammenheng med Java.

5.7.2. Scrum og timeplanlegging

Sprinter ble planlagt og gjennomgått med oppdragsgiver og veileder, i tillegg til at fremtidige planer som ble diskutert. Timeplanleggingen burde vært gjennomført bedre. Det vi gjorde feil i starten var oppgavene vi satt opp i alt for brede og omfattende. For eksempel oppgaven "Profil" inneholdt 4-5 forskjellige funksjoner som skulle implementeres på profilsiden, pluss at front-end og design på siden kom på toppen av dette. Denne store oppgaven burde vi heller fordelt i flere mindre deloppgaver.

Disse oppgavene skapte en utmattelse som måtte løses. For å takle denne utfordringen ble gruppen oppfordret til å være tilgjengelig og bruke sine "kontorer" på Discord, en taktikk som ble brukt av Andreas og Christer under utvikling av rapporten. Det løste problemet relativt raskt, da medlemmene var lettere tilgjengelig under samme arbeidstid nærmere tidsrommet 09-17. Noe som var nok tid til å oppnå en rapport vi kunne være fornøyde med.

5.7.3. Pandemien

Ettersom pandemien var godt i gang da prosjektet startet, måtte det kjøpt bestemmes hvordan gruppen skal arbeide. Da det er begrenset hvor mange en kan være på en plass, bestemte gruppen seg tidlig om å begrense møter til å ta plass digitalt. Så lenge regjeringens anbefalinger tillater det har gruppen hatt møter med hverandre og arbeidsgiver fysisk. Dette ble en utfordring, spesielt da alle ønsket å arbeide tettere med hverandre og med oppdragsgiver.

Pandemien har også medbrakt diverse utfordringer for studenter i hele landet, mer spesifikt psykiske utfordringer, som tidligere nevnt i rapporten. Dette skapte noen utfordringer underveis og det blir tydelig og ofte tatt opp i gruppen for å forsikre at det ikke skjer igjen mot slutten.

5.8. Gruppen

I starten av prosjektet måtte gruppen takle en krevende utfordring, da et av gruppemedlemmene ble sykemeldt under planleggingsfasen av prosjektet. Gruppen var noenlunde forberedt på sykemeldingen, da et gruppemedlem har vært en del frem og tilbake i helsesystemet siste året. Gruppen taklet denne utfordringen meget godt, ettersom det tok plass i pre-sprint og Sprint 1. Han var i jevnlig kommunikasjon med gruppen under sykemeldingen og fikk jevnlig oppdateringer tilbake. Han kom tilbake til gruppen i starten av sprint 2 og gjennomførte prosjektet derfra.

5.9. Egenevaluering

5.9.1. *Sven Opheim-Halsne*

I prosjektet har jeg jobbet mest med back-end programmering og litt front-end. Sammen med Khai har jeg jobbet med å programmere det meste av funksjonene i prosjektet vårt. Jeg hadde ansvaret for å få implementert de viktigste funksjonene på innleggs sidene. Etter at vi fullførte å programmere funksjonene har jeg jobbet med designet.

Gjennom dette prosjektet har jeg lært å bruke et nytt programmeringsspråk til utvikling(C#). Jeg har lært mer om å jobbe med Scrum-rammeverket og fått en bedre forståelse for hvordan man får dette til å fungere bra i praksis sammen med bedrift/produkteier. Gjennom prosjektet har vi hatt svært god kontakt og et godt samarbeid med produkteier, noe som har gjort dette til et svært lærerikt prosjekt for hvordan et slikt prosjekt kan fungere bra i praksis.

5.9.2. *Khai Hoang Joachim Nguyen*

I dette prosjektet har jeg for det meste vært med på de aller fleste ledd. Jeg har hovedsakelig hatt hovedansvaret for backend-programmeringen sammen med Sven, men har også hjulpet til med å utarbeide med frontend delen av prosjektet. Jeg har hatt for det meste hovedansvaret til profilsiden med å implementere diverse funksjoner som hentet informasjon fra feide, muligheten til å kunne legge og endre informasjon om seg selv, vise egne publiserte eller upubliserte innlegg samt muligheten til å kunne lagre egne eller andre innleggene til minne.

Gjennom prosjektet har jeg lært mye nytt og fått bedre forståelse av hvordan det er å utvikle for oppdragsgiveren, i tillegg fått et bredere spekter med erfaring hvordan Scrum og agile systemutvikling er i praksis. Samtidig har jeg fått lært meg et helt nytt programmeringsspråk som er litt annerledes i med det vi har lært fra før i studiet.

5.9.3. Christer Høiland

Gjennom prosjektet har jeg vært Scrum-master, i tillegg til å arbeidet med dokumentasjonsdelen av prosjektet sammen med Andreas. Det vil si at jeg har hovedsakelig dokumentert arbeidet som har blitt gjort, parallelt med utviklingen. Jeg har hatt ansvaret for å sikre fremgang i prosjektet ved hjelp av Scrum og opprettholde kontakt med Læringsbanken. I tillegg hadde jeg også hovedansvaret for risikoanalysen og hvordan disse risikoene skulle håndteres. Da gruppemedlemmet vårt som var sykemeldt kom tilbake tok jeg også på meg ansvaret for å gjøre denne prosessen så enkel som mulig for han og tilrettelagt arbeidet etter dette.

Jeg fått en bedre forståelse av hvordan det er å jobbet tett med en oppdragsgiver gjennom et prosjekt og fått bedre erfaring med hvordan Scrum og agil systemutvikling fungerer i praksis. Jeg har også fått bedre erfaring av hvordan det er å være Scrum-master og viktigheten av å opprettholde god kommunikasjon under denne typen prosjektutvikling.

5.9.4. Andreas Sheppard Perez

I løpet av prosjektet har jeg hatt varierende arbeidsoppgaver. Fra starten hjalp jeg med å produsere prototypen. Deretter ble det en periode hvor jeg ble sykemeldt, men holdt kontakt med gruppen. Ved overgangen tilbake fra sykemelding ble det daglig arbeid med økende timer for hver uke. Fokuset var da prosjektdokumentasjon. Det har vært et tett samarbeid med Christer for å fullføre prosjektdokumentasjonen i tillegg til litt å bistå Arian innen designteori.

Ved å arbeide i et slikt prosjekt har jeg lært utrolig masse. Jeg har fått mer erfaring med Scrum, større respekt for kommunikasjon og hvor viktig det er. Jeg har også lært mer teori enn forventet, da prosjektdokumentasjon krever en del lesing og reflektering. Prosjektet har vært meget spennende, lærerikt og krevende.

5.9.5. Arian Skaara Nuland

Som prosjektleder har jeg hatt et overordnet ansvar i å holde arbeidsflyten i gruppen. For å oppnå dette har jeg lagt mye arbeid i planleggingen av prosjektet, dette innebærer kartlegging av hva som kreves av oss som en gruppe for å oppnå den ønskede kvaliteten. Sammen med dette så har mitt fokus vært på front-end utviklingen, design, kvalitetssikring og prosjektstyring underveis, i tillegg til dette så har jeg tatt del i de fleste områdene prosjektet omgår.

Prosjektet i sin helhet har gitt meg et solid innblikk i hvordan arbeidslivet fungerer innenfor Informasjonssystemer og hvordan man i praksis arbeider med en reell oppdragsgiver. I tillegg til dette så har jeg har lært mye om hvordan agile systemutvikling fungerer i praksis . Utover det så har prosjektet gitt meg et fantastisk innblikk i hvordan prosjektstyring og problemløsning underbygger kvaliteten i et produkt.

6. Uttalelse fra Læringsbanken

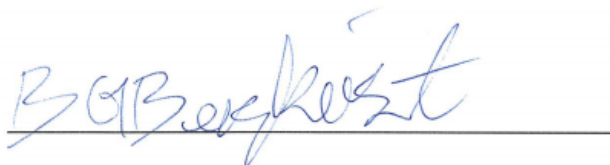
BÅRD GJERUSTAD BERGKVIST
12.05.21

Sluttrapport

Læringsbanken AS er et gründerprosjekt startet av lærerstudenter ved Universitetet i Agder for å bygge et knowledgemanagemant-system for bruk i grunnskolen. For tiden bygger vi vår MVP og vi har tatt imot et tilbud av gruppe 8 for å bygge deler av systemet.

Prosjektet startet med en forventningsavklaring mellom Læringsbanken og bachelorgruppen hvor gruppen ytret et ønske om autonomi, men god dialog. Under forventningsavklaringen Gruppe 8 rammene for prosjektorganisering, dialog og ledelse.

Ut fra Moscow-modellen vi utarbeidet med dem har de levert alt som står på «Must have» og deler av «should have». Gruppen har levert oppdateringer på tiden, stilt kritiske og avklarende spørsmål samt svart på våre spørsmål. Videre har gruppen fulgt mange av elementene i Scrum-organisering. Våre utviklere melder at gruppen har lært C# raskt og fulgt god kodestandard. Vi opplever å bli inkludert i utviklingen og at teamet har jobbet agilt rundt våre ønsker. Prosjektansvarlig fra læringsbanken melder om en tett dialog hvor små og store tilpasninger er blitt gjort underveis. Sluttproduktet som er levert lever opp til våre forventninger.



Bård Bergkvist
CXO
Læringsbanken AS

Universitetsveien 13
4630 Kristiansand
Org.nr: 925 028 614
Tlf.: 416 66 026

7. Referanser

Satzinger, J., Jackson, R., Burd, S. (2016). *Systems Analysis and Design* (7.).

Figur 1. Satzinger, J., Jackson, R., Burd, S. (2016). *Systems Analysis and Design* (7.).

Nielsen, J. (2020). 10 Usability Heuristics for User Interface Design. Hentet fra: <https://www.nngroup.com/articles/ten-usability-heuristics/>

Figur 6. Nielsen, J. (2020). 10 Usability Heuristics for User Interface Design. Hentet fra: <https://www.nngroup.com/articles/ten-usability-heuristics/>

Benyon, D. (2014). *Designing Interactive Systems* (3.). Edinburgh: Pearson Education Limited

Figur 7. Benyon, D. (2014). *Designing Interactive Systems* (3.). Edinburgh: Pearson Education Limited

Google (2014). Material System. Hentet fra: <https://material.io/design/introduction>

GeeksforGeeks. (2020). Differences between Black Box Testing vs White Box Testing. Hentet fra: <https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/>

Microsoft. (2019). Visual Studio IDE, Code Editor, Azure DevOps, & Apps Center. Hentet fra: <https://visualstudio.microsoft.com/>

Microsoft. (2020). C# Documentation. Hentet fra: <https://docs.microsoft.com/en-us/dotnet/csharp/>

Microsoft. (2019). What is ASP.NET? Hentet fra: <https://dotnet.microsoft.com/apps/aspnet>

W3Cs CSS Working Group. (2016). What is CSS? Hentet fra: <https://www.w3.org/Style/CSS/>

Microsoft. (2018). Blazor. Hentet fra: <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>

Atlassian. (n.d.). Bitbucket. Hentet fra:

<https://bitbucket.org/product/>

Microsoft. (2019). SQL Server 2019. Hentet fra:

<https://www.microsoft.com/nb-no/sql-server/sql-server-2019>

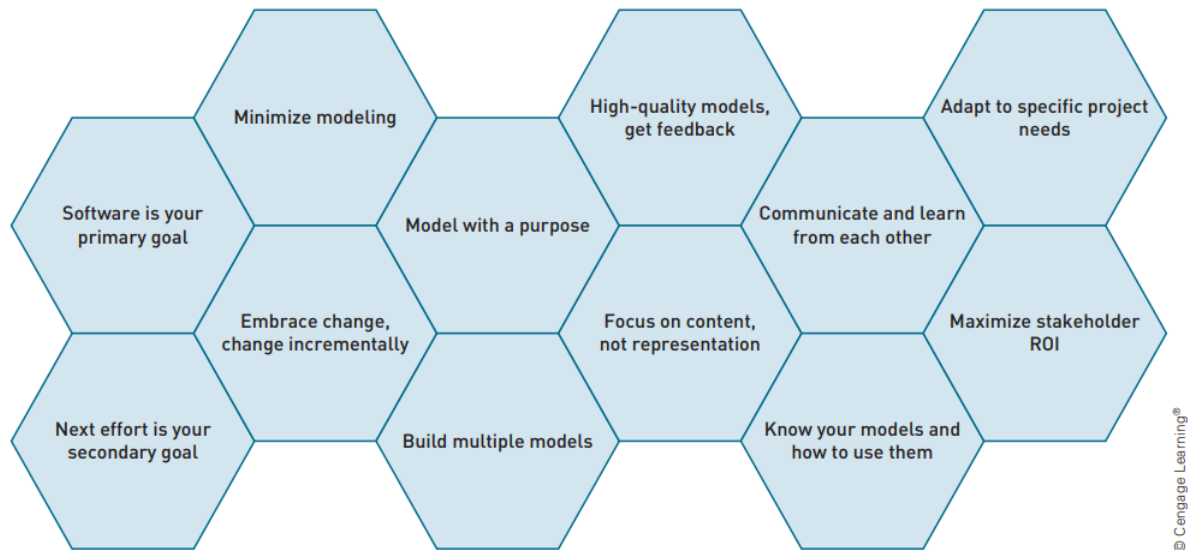
Lucid Software Inc. (2008). Lucid Software. A Visual Collaboration Suite. Hentet fra:

<https://lucid.co/>

Cockburn, A. (2007). *Agile Software Development. The Cooperative Game* (2.). Boston: Addison-Wesley

8. Vedlegg

Figur 1 - Agile modellering prinsipper



Figur 2 - MoSCoW tabellen med time estimering

Time Estimate		Estimate min hour	Estimate max hour	Actual hour
Must have	1. Gjør det mulig å rapportere innlegg	10	40	30
Must have	2. Lag metode for å hente data fra Feide API	10	40	25
Must have	3. Lag metode for å hente innlegg fra database	10	40	20
Must have	4. Gjør det mulig å dele innlegg med andre	10	40	20
Must have	5. Gjør det mulig å opprette skrive litt om deg selv felt på profilsiden	20	60	35
Must have	6. Lag metode for å hente informasjon om bruker fra database	5	20	10
Must have	7. Gjør det mulig å endre skrive litt om deg selv felt på profilsiden	20	60	40
Must have	8. Gjør det mulig å se mine egne innlegg som er publisert og/eller lagt til siden på profilsiden	10	30	25
Must have	9. Endre metode for å hente innlegg fra database	5	20	15
Must have	10. Lag innleggs feed på dashbordet	25	75	
Should have	11. Gjør det mulig å se enkelte innlegg utenfor login	1	10	1
Should have	12. Endre utsende på visning av innlegg (kort oppsummering av innlegget)	5	15	10
Should have	13. Lag metode for å hente ut lagrede innlegg fra database	10	40	15
Should have	14. Gjør det mulig å lagre egne eller andre brukers innlegg	10	40	30
Should have	15. Lag metode for å slette lagrede innlegg fra database	10	40	
Should have	16. Gjør det mulig for brukeren å se andre brukere (Private/Public)	10	40	
Should have	17. Gjør det mulig å gruppere brukere i grupper	10	40	
Should have	18. Gjør det mulig å filtrere innleggsfeeden pasert på tags	20	60	
Could have	19. Gjør det mulig å skrive kommentar til innlegget	20	60	
Could have	20. Gjør det mulig å legge en rating til andre brukers innlegg	10	40	
Could have	21. Opprette en newsfeed som oppsummerer kort om ukens top innlegg, informasjon fra Læringsbanken, osv...	20	60	
Could have	22. Gjør det mulig for brukeren å følge andre brukere	10	40	
Could have	23. Gjør det mulig å legge til mine erfaringer, tidligere utdanning i profilen	10	40	
Total Hours		271	950	276

Figur 3 - Eksempel på timeregistrerings-skjema

Dag	Start	Slutt	Pause	Tilleggstid	Timer	Eventuelt	Totalt i sprinten
01.03.2021	09:30	14:30	00:30		0	5 Jeg var syk	230
02.03.2021							
03.03.2021							
04.03.2021							
05.03.2021							
06.03.2021							
07.03.2021							
08.03.2021							
09.03.2021							
10.03.2021							
11.03.2021							
12.03.2021							
13.03.2021							
14.03.2021							
15.03.2021							
16.03.2021							
17.03.2021							
18.03.2021							
19.03.2021							
20.03.2021							
21.03.2021							
22.03.2021							
23.03.2021							
24.03.2021							
25.03.2021							
26.03.2021							
27.03.2021							
28.03.2021							
29.03.2021							
30.03.2021							
31.03.2021							
01.04.2021							

Figur 4 - Prosjektplan

Planleggingsark

Når jobber vi?

- Arbeidsområde 08:00-18:00
- 7,5(8) timer med jobbing mellom arbeidsområdet (f.eks. 08-16 eller 10-18)
- Mandag - Fredag, fredag fungerer som en dag hvor folk kan jobbe og gjøre litt annet, det blir kompensert med lengre arbeidstid mandag-torsdag
- Legg inn ting som jobb og legetimer i Google Calendar
- Felles arbeidsøkt onsdager 10-15 - alle er på discord og hjelper hverandre ut der det trengs, og orienterer hverandre om hvordan ståa er.

Hvordan jobber vi?

- Foreløpig digitalt arbeid, ser an corona restriksjonene og muligheten for et eget rom på UIA(via læringsbanken)
- Må være på discord i arbeidstiden
- Arbeidstiden skal være produktiv/effektiv tid - lov med pauser som f.eks. lunsj, middag
- C# og Blazor vil være hoved kodespråkene
- Jobbe jevnt i stedet for skipertak
- Sette deadlines og delmål, bruke Trello aktivt
- Alle gruppe medlemmene skal evaluere seg selv og hverandre i slutten av semesteret

Hvem har ansvar for hva?

- Front-end, utforming av nettsiden: Arian
- Back-end, utforming av koding: Khai
- Skrivning, utforming av bacheloroppgaven: Andy
- Gruppeleder: Arian + Khai
- Scrum master/planlegger: Christer
- Kontaktperson: Arian + Christer(backup)
- IT-konsulent: Sven
- Uansett viktig at alle har innsikt og kontroll over hva hverandre driver med, og kan bidra der det trengs på tvers av ansvarsområder.
- Dersom du ikke har noe å gjøre, finn noe å gjøre, eller hjelp noen andre. Det er alltid noe som kan gjøres.
- Si ifra hvis du trenger hjelp!

Scrum

- Weekly scrum med læringsbanken hver mandag - hva har vi jobbet med den forrige uken, og hva skal være fokuset denne uken?
- Vi må møte læringsbanken før klokken 13 (mandager), skal vi si klokka 11? Kl 09 for gruppen til å planlegge hva vi skal snakke om og evaluere uken
- Vær presis til møter, scrum osv, foreløpig vår definitivt største svakhet.
- Gi beskjed asap dersom uforventede ting dukker opp, og man ikke kan arbeide.

Figur 5 - Risikoanalysen



Navn på risiko	Alvorlighet	Sannsynlighet	Total risiko
Helsetilsikringer			
1. Psykisk utmattelse	4	4	Høy (16)
2. Fysisk utmattelse	2	2	Medium (4)
3. Motivasjonsmangel	3	4	Høy (12)
4. Uventet fravær	1	3	Lav (3)
5. Sykdom	3	3	Høy (9)
6. Død	5	1	Medium (5)
Tekniske risikoer			
7. Programvarefeil	4	3	Høy (12)
8. Maskinvarefeil	4	1	Medium (4)
9. Lagringsfeil	4	2	Høy (8)
10. Internettfeil	2	2	Medium (4)
11. Mangel på nødvendige verktøy	4	1	Medium (4)
Kommunikasjonsrisikoer			
12. Dårlig kommunikasjon med Læringsbanken	4	1	Medium (4)
13. Dårlig kommunikasjon med veilederen	2	1	Lav (2)
14. Svake tilbakemeldinger på testing	3	2	Medium (4)
15. Svake tilbakemeldinger på intervjuer	2	2	Medium (4)
16. Oppmøtemangel i forelesninger	2	2	Medium (4)
17. Oppmøtemangel i gruppeøvelser	3	1	Lav (3)
18. Konflikter innad i gruppen	4	1	Medium (4)
Planleggingsrisikoer			
19. Ukorrekte møter	3	1	Lav (3)
20. Lite planleggingsmøter	2	1	Lav (2)
21. Dårlig tidsestimerting	2	2	Medium (4)
22. Ikke holde tidsfrister	3	2	Medium (6)
23. Dårlig arbeidsfordeling	3	1	Lav (3)
24. Dårlig planlagte sprinter	2	2	Medium (4)

Figur 6 - Nielsens brukbarhets heuristikker

1 Visibility of System Status

Designs should **keep users informed** about what is going on, through appropriate, timely feedback.

2 Match between System and the Real World

The design should speak the users' language. Use words, phrases, and concepts **familiar to the user**, rather than internal jargon.

5 Error Prevention

Good error messages are important, but the best designs **prevent problems** from occurring in the first place.

8 Aesthetic and Minimalist Design

Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface **competes** with the relevant units of information.

Nielsen Norman Group

Jakob's Ten Usability Heuristics

3 User Control and Freedom

Users often perform actions by mistake. They **need a clearly marked "emergency exit"** to leave the unwanted state.

6 Recognition Rather Than Recall

Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.

9 Recognize, Diagnose, and Recover from Errors

Error messages should be expressed in **plain language** (no error codes), precisely indicate the problem, and constructively suggest a solution.

4 Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. **Follow platform conventions**.

7 Flexibility and Efficiency of Use

Shortcuts — hidden from novice users — may **speed up the interaction** for the expert user.

10 Help and Documentation

It's best if the design **doesn't need** any additional explanation. However, it may be necessary to provide documentation to help users understand how to complete their tasks.

Figur 7 - Benyons 12 designprinsipper

- 1 *Visibility*. Try to ensure that things are visible so that people can see what functions are available and what the system is currently doing. This is an important part of the psychological principle that it is easier to recognize things than to have to recall them. If it is not possible to make it visible, make it observable. Consider making things 'visible' through the use of sound and touch.
- 2 *Consistency*. Be consistent in the use of design features and be consistent with similar systems and standard ways of working. Consistency can be something of a slippery concept (see the Further thoughts box). Both conceptual and physical consistency are important.
- 3 *Familiarity*. Use language and symbols that the intended audience will be familiar with. Where this is not possible because the concepts are quite different from those people know about, provide a suitable metaphor to help them transfer similar and related knowledge from a more familiar domain.
- 4 *Affordance*. Design things so it is clear what they are for; for example, make buttons look like push buttons so people will press them. Affordance refers to the properties that things have (or are perceived to have) and how these relate to how the things could be used. Buttons afford pressing, chairs afford sitting on, and Post-it notes afford writing a message on and sticking next to something else. Affordances are culturally determined.

Giving them the sense of being in control, knowing what to do and how to do it

- 5 *Navigation*. Provide support to enable people to move around the parts of the system: maps, directional signs and information signs.
- 6 *Control*. Make it clear who or what is in control and allow people to take control. Control is enhanced if there is a clear, logical mapping between controls and the effect that they have. Also make clear the relationship between what the system does and what will happen in the world outside the system.
- 7 *Feedback*. Rapidly feed back information from the system to people so that they know what effect their actions have had. Constant and consistent feedback will enhance the feeling of control.

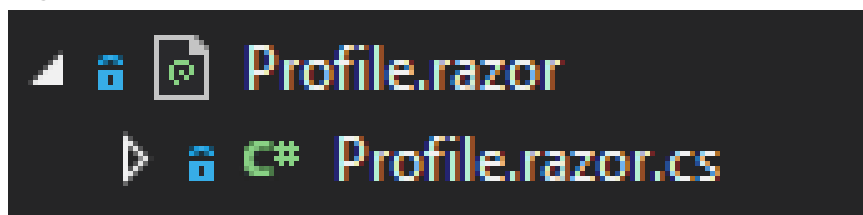
Safely and securely

- 8 *Recovery*. Enable recovery from actions, particularly mistakes and errors, quickly and effectively.
- 9 *Constraints*. Provide constraints so that people do not try to do things that are inappropriate. In particular, people should be prevented from making serious errors through properly constraining allowable actions and seeking confirmation of dangerous operations.

In a way that suits them

- 10 *Flexibility*. Allow multiple ways of doing things so as to accommodate people with different levels of experience and interest in the system. Provide people with the opportunity to change the way things look or behave so that they can personalize the system.
- 11 *Style*. Designs should be stylish and attractive.
- 12 *Conviviality*. Interactive systems should be polite, friendly and generally pleasant. Nothing ruins the experience of using an interactive system more than an aggressive message or an abrupt interruption. Design for politeness (see Box 4.4). Conviviality also suggests joining in and using interactive technologies to connect and support people.

Figur 8 - MVC



Figur 9 - Gruppekontrakt

Gruppekontrakt

Gruppe 8

Mål

Gruppens mål er at alle innleveringer skal ha gjennomgående god kvalitet. Hvis alle legger inn en god innsats burde toppkarakter være oppnåelig.

Orden

Alle gruppemedlemmer skal melde fra om fravær, dette gjelder også hvis man er mer enn 10 minutter for sen til gruppetimen. Hvis et medlem kommer for sent uten å melde fra gjentatte ganger, vil dette være gyldig grunn til straff. Straffen vil bli valgt av gruppen, og basert på omfanget av fraværet, samt omstendighetene rundt regelbruddet. Ved hyppig fravær med melding kan det også ende med en reaksjon, men dette må bedømmes av resten av gruppen, med et sterkt fokus på omstendighetene rundt fraværet.

Andre viktige punkter rundt orden:

- Alle gruppemedlemmer skal møte forberedt til gruppetimene.
- Et gruppemedlem burde ta ansvar for å reservere møterom i god tid.
- Det er viktig at alle gruppemedlemmer fokuserer på oppgaven de har foran seg og fullfører innen tidsrammen. Hvis man ikke holder deadlines vil man bli straffet.
- Det er viktig at gruppen kommuniserer bra, og at alle vet hva deres oppgave er der og da.
- På gruppetimer skal alle gruppemedlemmer jobbe strukturert ifølge tidene som gruppen setter opp.

Beslutninger

Hvert gruppemedlem har en stemme. Alle større beslutninger kan gjøres med et overtall av stemmer, men i de fleste tilfeller ønsker gruppen helst å inngå et kompromiss.

Miljø

Alle medlemmer av gruppen skal vise respekt for andre, og konflikter skal løses så raskt som mulig. Alle skal behandles likt, så godt det lar seg gjøre. Dette vil forhindre at noen føler seg urettferdig behandlet.

Straff

Hvis et gruppemedlem ikke følger de reglene som de selv har skrevet under på i denne kontrakt, vil medlemmet være pliktig til å gjøre dette opp ved en senere anledning. Medlemmet kan for eksempel måtte jobbe mer ved en annen oppgave, eller hjelpe gruppen på en annen måte, for eksempel ved å kjøpe mat til de andre medlemmene.

1. BRUKERHISTORIER

Førsteprioritering: Estimert ferdig 01.08.20

- Som bruker ønsker jeg å registrere en bruker og logge inn på plattformen, så jeg kan konsumere innhold.
- Som bruker ønsker jeg å kunne scrolle meg gjennom en blandet feed med relevante innlegg for mine interesser, for å få tips og innspill jeg kan bruke i læreryrket.
- Som en bruker ønsker jeg å ha min egen brukerprofil, slik at jeg kan se hvilke innlegg jeg har lagret på og hvilke innlegg jeg selv har delt.
- Som bruker ønsker jeg en nettside som ser elegant og brukervennlig ut, slik at jeg har en positiv opplevelse når jeg er på siden og slik at jeg ønsker å komme tilbake.
- Som bruker ønsker jeg en intuitiv måte å søke på innlegg, så jeg finner undervisningsopplegg lettere.
- Som bruker ønsker jeg å kunne se gjennom kategorier som et alternativ til søking.
- Som bruker ønsker jeg å publisere innlegg, så jeg kan dele erfaringen min med andre.
- Som bruker ønsker jeg å sortere egne innlegg i kategorier så de blir enklere å finne.
- Som lærerstudent ønsker jeg en database med undervisningsopplegg jeg kan teste ut i praksis, for å finne min egen lærerstil.
- Som lærerstudent ønsker jeg en database med undervisningsopplegg jeg kan teste ut i praksis, for å være så kreativ og motiverende som mulig overfor elevene.
- Som bruker ønsker jeg å spesifisere innlegget mitt i
 - o Alder/ klasstrinn
 - o Tidsbruk
 - o Gruppe/ individuelt
 - o Nøkkelord
 - o Kompetansemål
- Som bruker ønsker jeg å vurdere andres innlegg for å sikre kvalitet.
- Som administrator ønsker jeg å kunne legge til nye tagger og definisjoner

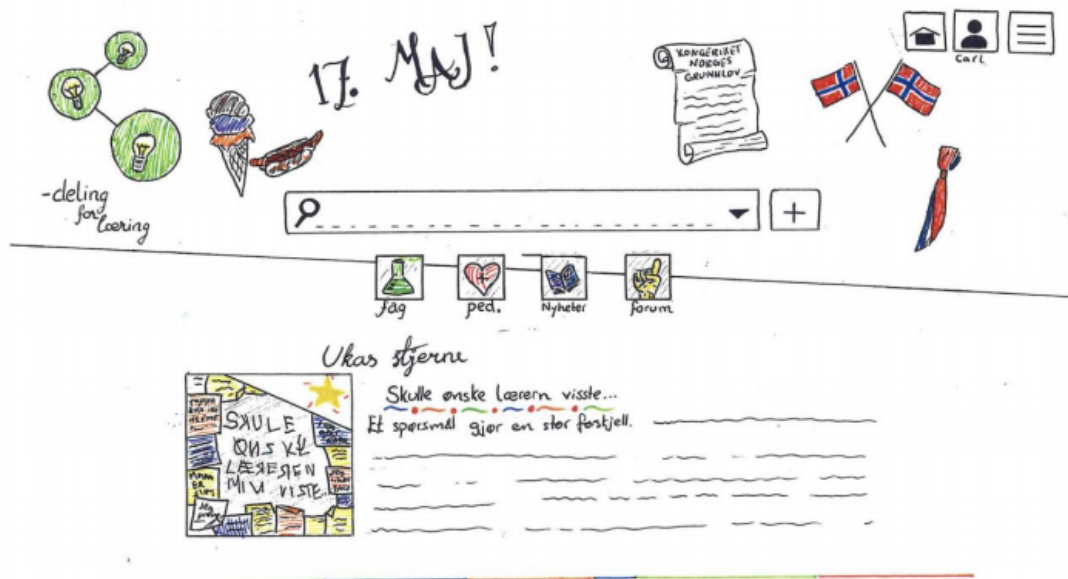
Andreprioritering: Estimert ferdig 01.01.21

- Som bruker ønsker jeg å lagre innlegg, så jeg kan finne dem igjen senere på min profil
- Som bruker ønsker jeg å se andres profiler, så jeg kan se alle innleggene deres og informasjon om dem.
- Som bruker ønsker jeg å kommentere, gi tilbakemeldinger og legge opp til diskusjon under hvert innlegg.
- Som bruker ønsker jeg å dele innlegg direkte i andre kanaler, som for eksempel sosiale medier.

Tredjeprioritering: Bør vente

- Som bruker ønsker jeg å velge favorittkategorier, for å videre spesifisere relevant innhold
- Som bruker ønsker jeg å få notasjoner når nytt innhold i ønskede kategorier blir publisert
- Som bruker ønsker jeg å spore hvilke kompetansemål jeg har vært gjennom med innleggene jeg har prøvd ut i min profil.
- Som bruker vil jeg ha muligheten til å foreslå nye tagger.
- Som bruker ønsker jeg muligheten til å ha en gratisversjon av nettsiden, så jeg kan få en smakebit av hva tjenesten vil gi meg
- Som bruker ønsker jeg et forum hvor jeg kan komme i kontakt med andre lærere, fagpersonell og eksperter, for å diskutere skoletematikk.

Figur 11 - Skisser



Fag

PERIODISK BAKESHIP!

ELEFANT-SKUM!

Du trenger:

- Gjør
- Vann
- Zink
- H₂O₂

Portal til Underverden!

Bevart velkommen

NH₄Cr₂O₇
H₂SO₄

Min favorittfeil ✓

✗

Pedagogikk

BRÅKORT

BRA!

Klasseromsbelønninger!

- Bake Julekaker i 15 minutter
- Teke opp hendene 15 ganger
- Tygg tyggis i 30 min
- Hjelp sidemann
- Gjør alle lekser

Skjønner!

Vanstelig!

GA OPP!

Nyheter

Masteroppgave:
Hvordan kjempe igjen barn med stort potensial?
Karakter A

Hvordan snakke om:
PARISTERRØREN

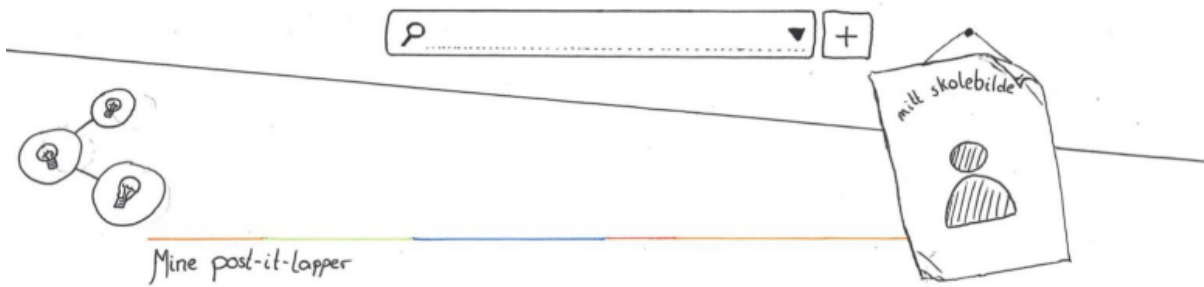
Fjerne rett til spesialundervisning?
HVA SKJER?

Stortingsmelding:
Vil fornye fagene

29.04.19

Skoleblogg - Det gjør vi
Reyse skole, tirsdag.

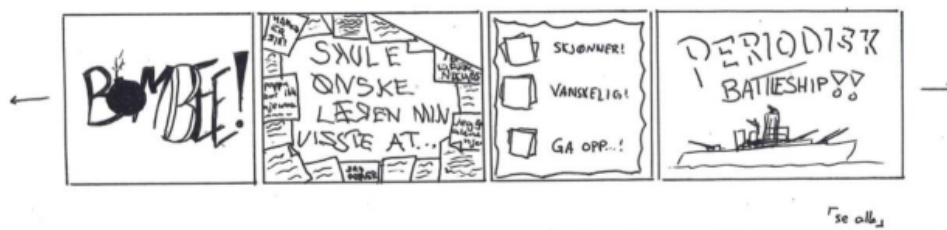
Hvordan snakke om:
Fred i Korea?



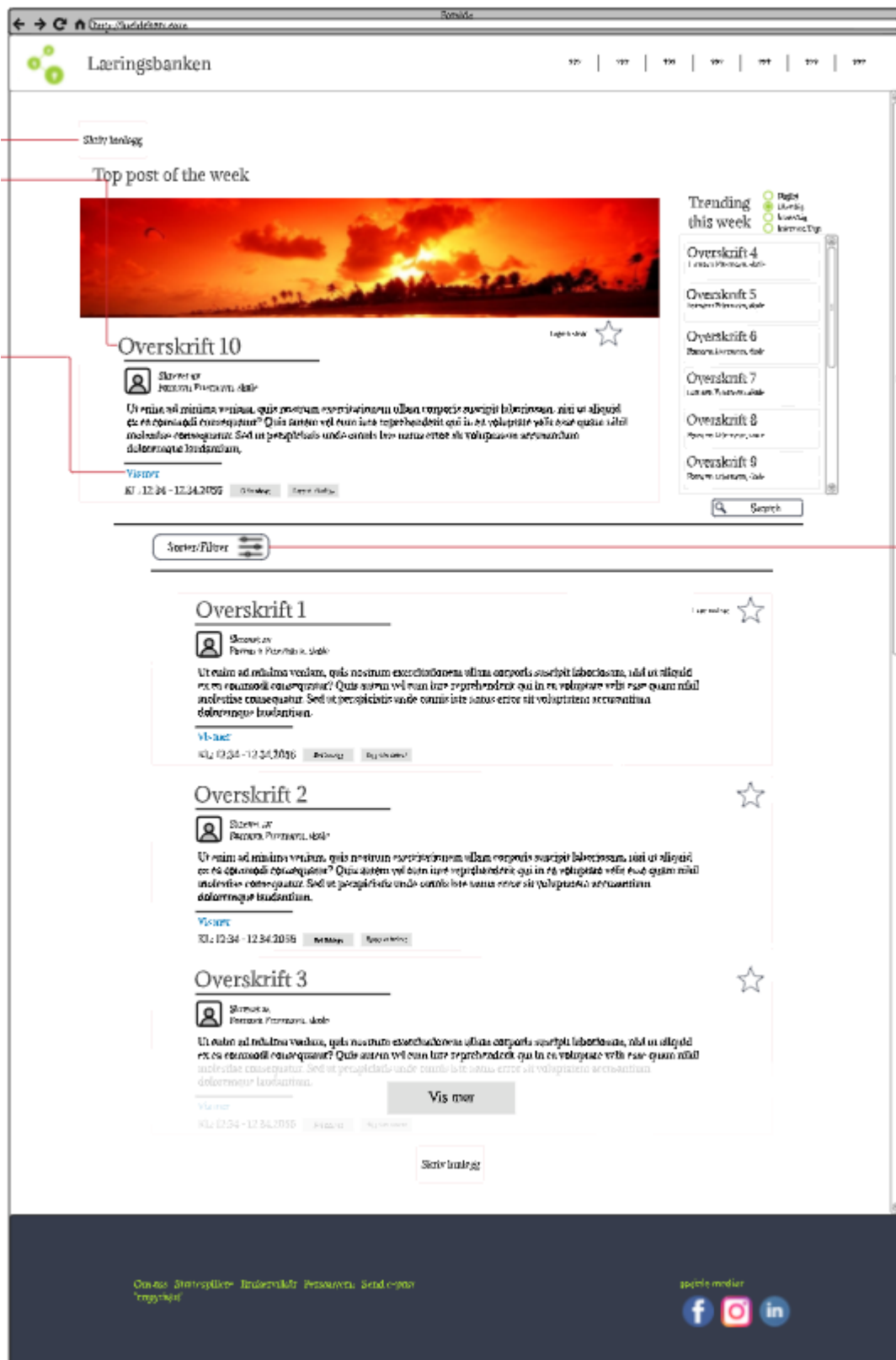
Mine post-it-lapper



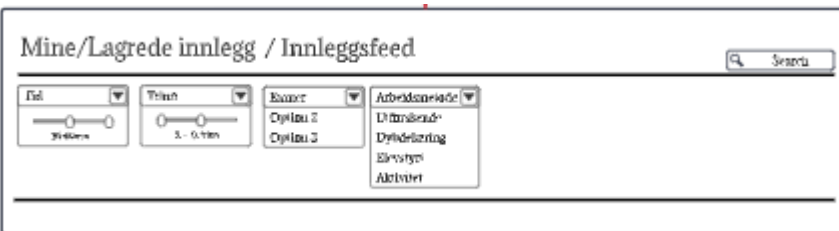
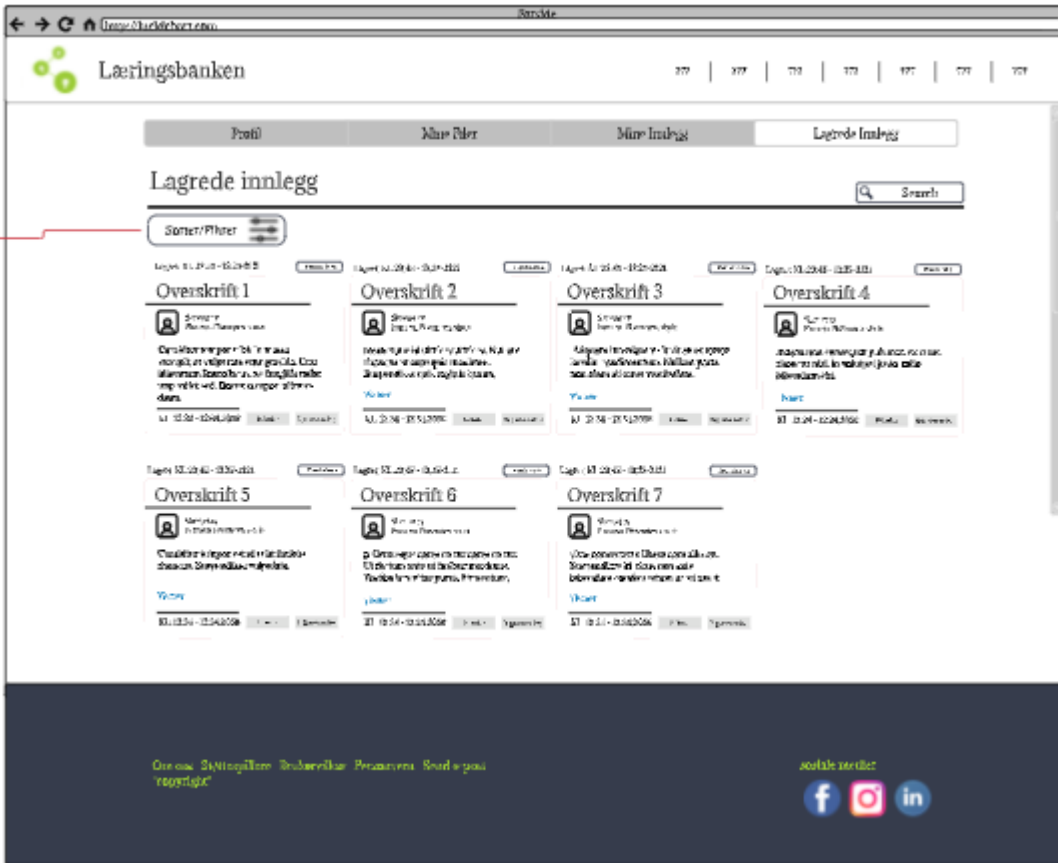
Mine innlegg



Figur 12 - Prototype/wireframe








← → ↻ 🏠 <http://studiebanke.no> Studie

Læringsbanken 100 | 101 | 102 | 103 | 104 | 105 | 106


Profil **Mine filer** Mine lenker Lagrede lenker

Mine filer


Mapper




Læringsplan



Eksempeloppgaver






Eksempeloppgaver



Quiz, XXXX.XX.XX

Du kan støtte opp for brukerne våre. Prosjektene våre er gratis tilgjengelig!

Andre medier:


← → ↻ 🏠 <http://studiebanke.no> Studie

Læringsbanken 100 | 101 | 102 | 103 | 104 | 105 | 106


Profil **Mine filer** Mine lenker Lagrede lenker

Mine filer


Mapper




Læringsplan



Eksempeloppgaver









Eksempeloppgaver






Quiz, XXXX.XX.XX

Filer

Filnavn	Filtype ▼	Dato	Tilgjengelighet
 Wordfil_1.docx	MS Word	12.21.20 - 14-25	Pivat
 Wordfil_2.docx	MS Word	12.21.20 - 14-25	Offentlig
 Powerpointfil_1.pptx	MS Powerpoint	12.21.20 - 14-25	Pivat
 Powerpointfil_2.pptx	MS Powerpoint	12.21.20 - 14-25	Pivat
 Powerpointfil_3.pptx	MS Powerpoint	12.21.20 - 14-25	Offentlig
 Tekstfil_1.txt	Notepad	12.21.20 - 14-25	Pivat

Du kan støtte opp for brukerne våre. Prosjektene våre er gratis tilgjengelig!

Andre medier:

Figur 13 - Møtereferat fra styringsmøte 1

Referat av styringsmøte

Prosjekt: "Læringsbanken.no" bachelor prosjekt
Dato: 23.03.21
Sted: Digitalt møte

Til stede:

Carl August Arnesen Tronrud (Læringsbanken), Janis Gailis (UiA), Arian Nuland, Andreas Sheppard Perez, Christer Høiland, Khai H.J. Nguyen, Sven Opheim-Halsne

Tirsdag, 23. Mars 2021, ble det holdt et digitalt styringsmøte med oppdragsgiver og veileder for å vise status av prosjektet. Det ble også tatt opp spørsmål og veiledning av både oppdragsgiver og veileder.

Planen for møtet gikk som følger:

- Gjennomgang av system
- Vise prototype og fremtidige idéer
- Prosjekt status
- Gjennomgang av Sprint 2
- Innspill av oppdragsgiver og veileder, med videre ønsker
- Planlegging for Sprint 3

Oppsummering

Gjennomgang av systemet, dokumenter og prototype

Gjennomgang av systemet ble gjennomført, og diverse ny og gammel funksjonalitet ble tatt opp. Knapper som "Rapporter innlegg" ble vist frem, tilpasning av brukerprofil. De midlertidige innleggene ble tatt opp, men disse innleggene skal designes til noe mer gjenkjennelig. Deretter ble risikoanalyse delt, og en gjennomgang av prototypen tok plass. Under dette stadiet i prosjektet er noen av funksjonene fra prototypen lagt til. Den gir også et bedre bilde til hvordan innleggene skal ses ut.

Under gjennomgang av systemet tok veileder Janis Gailis opp Nielsens brukbarhets-heuristikker, Google sitt designspråk kalt "Material Design". Dette gjelder utvikling av design for systemet, retningslinjer og måter en kan oppnå et velfungerende, fint system. Dette har siden blitt lest mer opp på og inkludert i rapporten.

Deretter ble det tipset om V-model og gode tips angående testing. Dette innebærte å ha forskjellig typer testere, erfarne databrukere, avanserte brukere. Det ble også anbefalt å kjøre oppgaver og spørsmål under testing, eksempelvis: "Følg oppgaven: skriv et innlegg, del noen andres innlegg, rapporter et innlegg. Hva skjedde under X funksjon. Hva ble du møtt på etter X knapp, hvilket vindu kom opp etter X."

Scrum prosessen

Gruppen informerte om arbeidsflyten, ved hjelp av Azure DevOps som Scrum-board, dette inneholder burndown chart og gir meget god oversikt over hvilke oppgaver som blir jobbet med, hvor lang tid som blir brukt og om de blir sett på som ferdig eller ikke. For å fortsatt ha god oversikt, valgte vi å legge til et punkt kalt "Design". Det innebærer at funksjoner som skal designes i front-end plasseres her, slik at alle er klar over at funksjonen er nærmest ferdig men ikke helt i mål enda. Christer Høiland har rollen som Scrum master og vil kalle inn til nytt møte i april.

Design og kvalitetsplanlegging

For å forsikre kvalitet av prosjektet og systemet har gruppen blitt enige om at alle skaper én branch per funksjon, slik at funksjoner ikke skal skape problemer for hverandre og gi oversikt over hvilke funksjoner som kan gi problemer. En annen del av kvalitetsplanlegging kom i forslag av Janis Gailis, og gjaldt "acceptance testing". Dette innebærer at ved publisering av branch, kjøres en test for å passe på at all kode fungerer som den skal.

Mottatte tilbakemeldinger|

- Design, diverse prinsipper
- Testing
- Godt bruk av Scrum-board

Figur 14 - Møtereferat fra styringsmøte 2

Referat av styringsmøte

Prosjekt: "Læringsbanken.no" bachelor prosjekt
Dato: 27.04.21
Sted: Digitalt møte

Til stede:

Carl August Arnesen Tronrud (Læringsbanken), Janis Gailis (UiA), Arian Nuland, Andreas Sheppard Perez, Christer Høiland, Khai H.J. Nguyen, Sven Opheim-Halsne

Tirsdag, 27. April 2021, ble det holdt et digitalt styringsmøte med oppdragsgiver og veileder for å vise status av prosjektet. Det ble også tatt opp spørsmål og veiledning av både oppdragsgiver og veileder. Grunnet misforståelser ute av vår kontroll, ble styringsmøtet delt i to; første møtet 27.04.2021 med arbeidsgruppe og oppdragsgiver, Carl August Arnesen Tronrud. Det ble holdt et annet møte, 03.05.21 med arbeidsgruppe og veileder. Referatet vil derfor inneholde informasjon fra to separate møter.

Planen for møtet gikk som følger:

- Gjennomgang av forandringer i systemet
- Fremtidige idéer
- Prosjekt status - fremtidig testing
- Gjennomgang av Sprint 3
- Innspill av oppdragsgiver og veileder, med videre ønsker
- Planlegging for Sprint 4, siste uken før innlevering av rapport

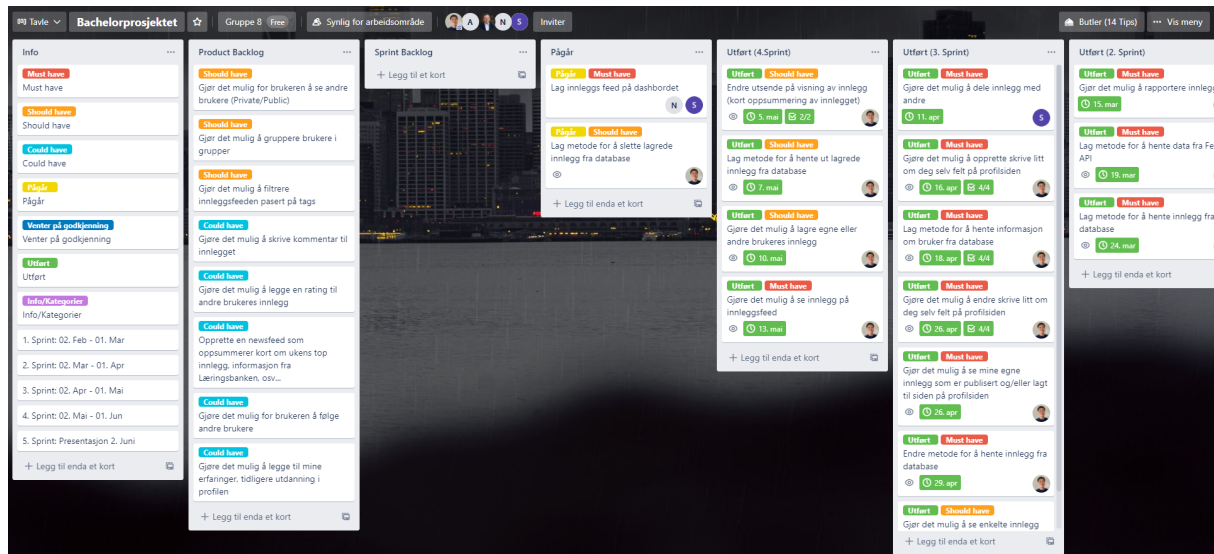
Oppsummering

Gjennomgang av systemet og fremtidig testing

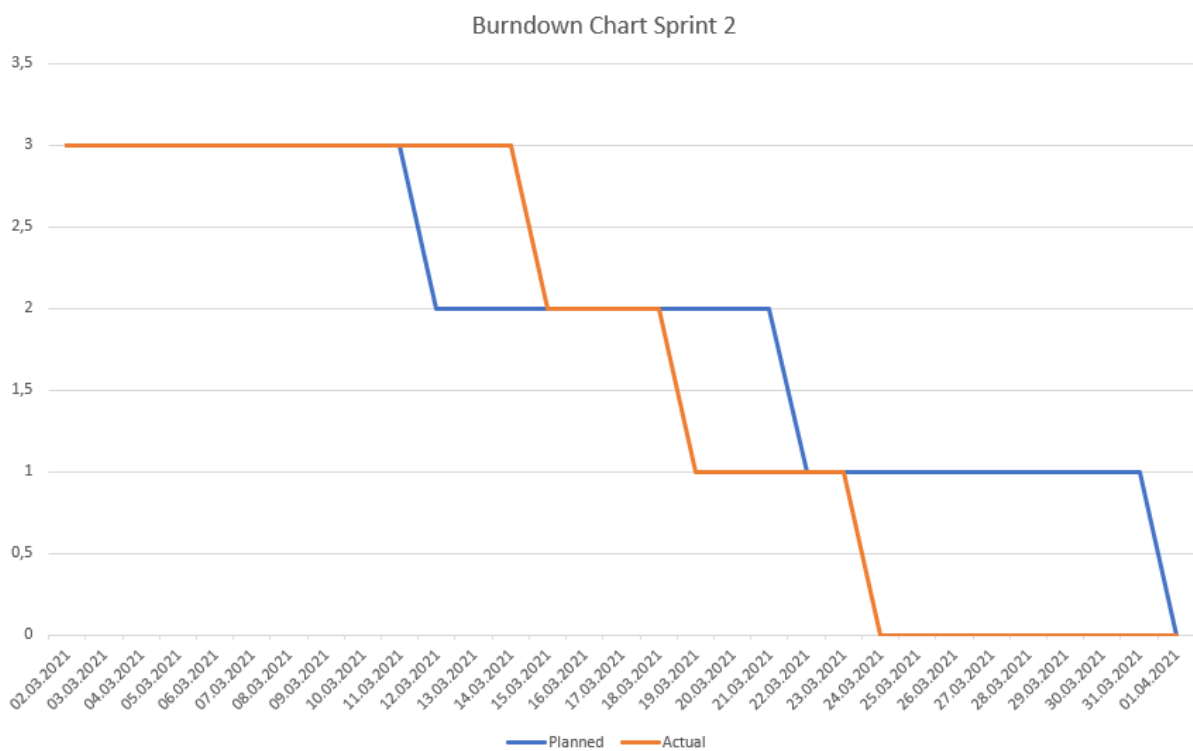
Gjennomgang av systemet ble gjennomført med veileder, hvor nye funksjoner ble vist. Dette innebærer eksempelvis tittel og innhold i innlegg, med fullstendig profil og personalisering. Deretter fikk gruppen veiledning angående tidsestimering som skal inkluderes i arbeidsrapporten. Her fikk alle en god gjennomgang av viktige emner i tidsestimering som å dele oppgaver opp i riktig størrelse i henhold til tidsbegrensning, og sammenligning mellom tidsestimering og faktisk tidsbruk. Avsluttende i møtet med veileder ble diverse pirk i henhold til muntlig eksamen som tar plass 02.06.21.

Møtet med oppdragsgiver startet ved å oppdatere progresjon i systemet, og små innspill fra oppdragsgiver. Deretter gikk temaet over til eventuelle kurs som kunne avtales, slik at gruppemedlemmene kunne få innleid kurs hvis det var ønskelig. Utførelse av testing ble diskutert, da oppdragsgiver fortalte om tilgjengelig prosess. Dette innebærte grupper på Facebook, med medlemmer som kunne utføre testing på en effektiv måte. Diverse avgjørelser

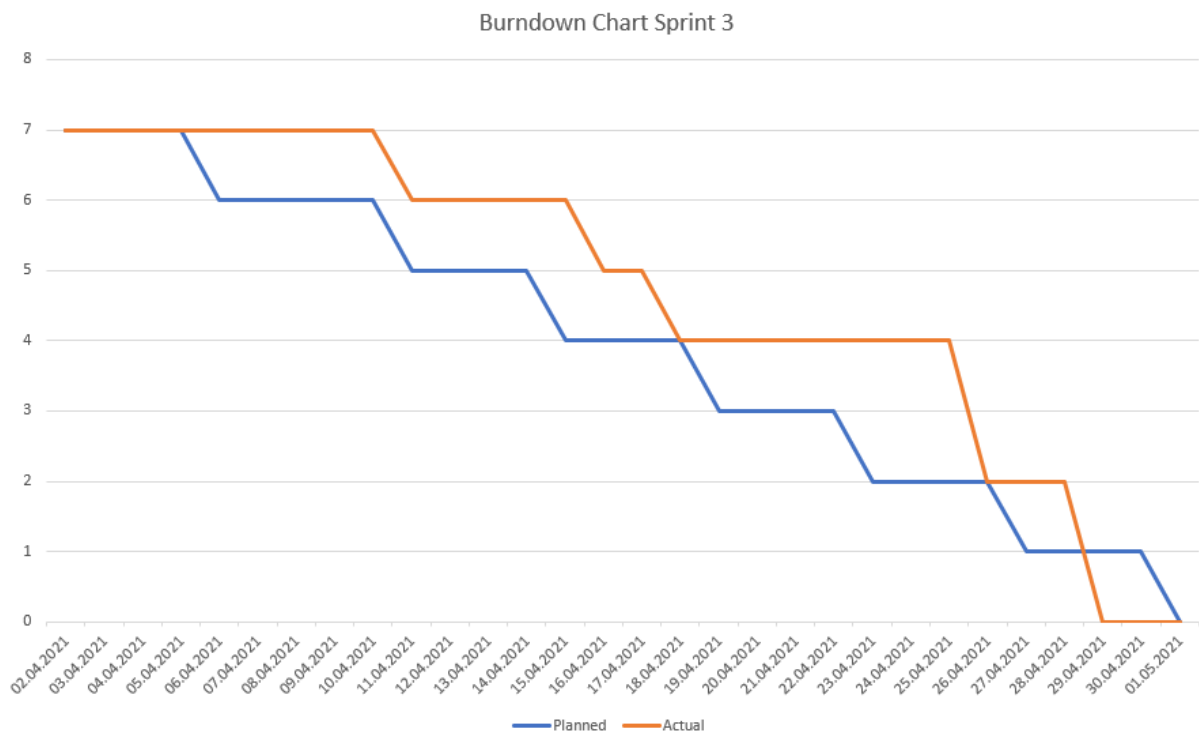
Figur 15 - Product Backlog



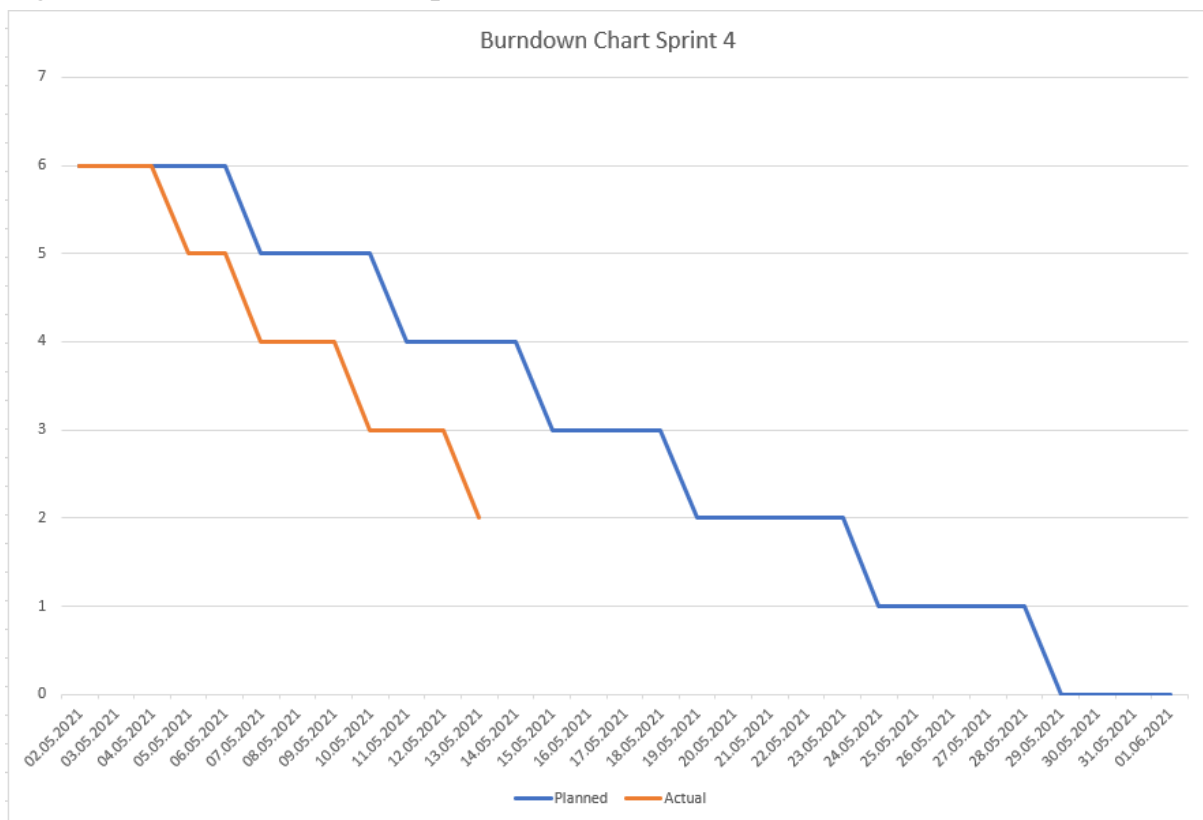
Figur 16 - Burndown chart Sprint 2



Figur 17 - Burndown chart Sprint 3



Figur 18 - Burndown chart Sprint 4



Figur 19 - Kode eksempel

```
2 references
private async void SaveUser() // When we return the user that we are creating we can make this an async again.
{
    var state = await AuthState.GetAuthenticationStateAsync();
    this.usermodel.UserId = UserId;
    this.usermodel.About = About;
    this.usermodel.Interests = Interests;
    this.usermodel.Grade = Grade;
    this.usermodel.Organization = Organization;

    try
    {
        loadFailed = false;
        var createdUser = await httpClient.CreateUserAsync(this.usermodel);

        StateHasChanged();
        NavigationManager.NavigateTo("/profil", true);
    }
    catch (Exception ex)
    {
        loadFailed = true;
        NavigationManager.NavigateTo("/profil", true);
    }
}
```