



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Utlånssystem for elektrolab

Library system for electronics lab

Magnus Leira, Daniel L. Jacobsen, Ingebrigt S. Holstad

Informasjonsteknologi/Dataingeniør

Institutt for datateknologi, elektroteknologi og realfag

Bjarte Kileng

04.06.2021

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle

kilder som er brukt i arbeidet er oppgitt, jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 10.

Utlånssystem for elektrolab

Rapportens tittel: Utlånssystem for elektrolab	Dato: 04.06.2021
Forfatter(e): Daniel L. Jacobsen Magnus Leira Ingebrigt S. Holstad	Antall sider u/vedlegg: 30
	Antall sider m/vedlegg: 44
Studieretning: Dataingeniør/informasjonsteknologi	Antall disketter/CD-er: 0
Kontaktperson ved studieretning: Bjarte Kileng	Gradering: ingen
Merknader: Ingen	

Oppdragsgiver: HVL IDER	Oppdragsgivers referanse: ingen
Oppdragsgivers kontaktperson: Endre Håland	Telefon: +47 922 39 154

Sammendrag: Rapporten tar for seg utviklingsprosessen av et utlånssystem for elektrolabben ved HVL, hvor oppdragsgiver ønsket et enkelt og oversiktlig program for å holde styr på utstyr og hvem som låner det. Planleggingen og valgene som ble gjort i prosjektet baserte seg på noen initielle krav fra oppdragsgiver, for å komme frem til en løsning som samsvarte med hva som var ønsket. Studenter har tilgang til å reservere utstyr og se brukerhistorikken, mens administratorer kan endre, legge til og slette utstyr, kategorier og brukere.

Summary: The report describes the development process of a library system for the electronics lab at HVL, where the client wanted a simple and lucid program to keep track of equipment and who borrowed it. The planning and choices made in the project were based on some initial requirements from the client to find a solution that matched with what was wanted. Students have the ability to make reservations on equipment and see their user history, while administrators can change, add and delete equipment, categories and users.

Stikkord:

Utlånssystem	Webapplikasjon	React.js	ASP.Net core	Azure
--------------	----------------	----------	--------------	-------

Høgskulen på Vestlandet, Fakultet for ingeniør- og naturvitenskap

Postadresse: Postboks 7030, 5020 Bergen

Besøksadresse: Inndalsveien 28, Bergen

Tlf 55587500

Fax 55 58 77 90

E-post: post@hvl.no

Hjemmeside: <http://www.hvl.no>

Forord

Vi ønsker å takke Endre Håland for muligheten til å løse oppgaven, samt mye engasjement rundt prosjektet. Dette prosjektet har gitt oss mye erfaring og lærdom.

Takk til Bjarte Kileng for god veiledning og mye god innsikt, både med tanke på rapporten og selve prosjektutviklingen.

Vi vil også takke brukertesterne for mange gode innspill og at de tok seg tid til å gjøre det.

1 Innledning	6
1.1 Motivasjon og mål	6
1.2 Kontekst	7
1.3 Avgrensninger	7
1.4 Ressurser	8
1.5 Oppbygging av rapporten	8
2 Prosjektbeskrivelse	9
2.1 Praktisk bakgrunn	9
2.1.1 Om prosjekteier	9
2.1.2 Initielle krav	9
2.1.3 Initiell løsnings-idé	10
2.2 Litteratur om problemstillingen	11
2.2.1 Hva er UX?	11
2.2.1.1 Relevans	12
2.2.1.2 Brukervennlighet	12
2.2.1.3 Navigering	12
2.2.1.4 Tilgjengelighet	12
2.2.1.5 Kjennskap	13
3 Design av prosjektet	13
3.1 Forslag til løsning	13
3.1.1 Mobilapplikasjon	13
3.1.2 Webapplikasjon	14
3.2 Valgt løsning	14
3.3 Valg av verktøy	14
3.4 Prosjektmetodikk	15
3.4.1 Utviklingsmetodikk	15
3.4.2 Prosjektplan	16
3.4.3 Risikovurdering	16
3.5 Evalueringsplan	16
4 Detaljert Design	17
4.1 UI	17
4.1.1 Reservasjonsside	18
4.1.2 Admin	20
4.1.3 Min side	21
4.2 Klientside-rammeverk	22
4.3 Server og API	23
4.3.1 Deployment	23
4.3.2 Web-API	23

4.3.3 Database	23
4.3.4 Login og registrering	24
5 Evaluering	26
5.1 Evalueringsmetode	26
5.2 Evalueringsresultat	26
6 Diskusjon	28
7 Konklusjon og videre arbeid	29
8 Referanser	31
8.1 Veiledninger/Litteratur	31
9 Appendix	32
9.1 Risikoliste	32
9.2 GANTT-diagram	34
9.3 Brukermanual	34
9.3.1 Min side	34
9.3.2 Reservasjon	35
9.3.3 Admin	35

1 Innledning

Dette kapittelet er en introduksjon til prosjektet. Det inneholder en oversikt over motivasjon og mål, kontekst, avgrensninger, ressurser og oppbyggingen av prosjektet.

1.1 Motivasjon og mål

HVL IDER Elektroteknikk har mye verdifullt utstyr, men har ingen god måte å holde styr på det. Dette gjør det vanskelig å ha oversikt over hva slags utstyr labben eier, hvem som låner det og det går mye tid på å lete. Oppdragsgiver ønsker et nettsted hvor utstyret på elektrolaben skal kunne registreres, for så å lånes ut til brukere. Hva er en god måte å organisere utlånssystemet på med tanke på sortering av de ulike utlånproduktene og logistikken rundt det?

Prosjektet skal være et velfungerende og brukervennlig lånebibliotek der man kan låne ut og holde oversikt over labutstyr. Det vil bli lagd ulike metoder for å logge inn på systemet. I første omgang vil det være en simpel innlogging med brukernavn og passord, som en får tilgang til å bruke etter en har registrert seg. På sikt vil det være aktuelt å implementere Feide-innlogging, da nettsiden bare skal bli brukt av studenter og ansatte ved HVL.

Oppdragsgiver ønsker at utlånssystemet skal være tillitsbasert. Gruppen trenger derfor ikke å ta forbehold om at enkelte kan komme til å bevisst ødelegge i systemet.

Brukere av systemet skal bestå av administratorer og studenter som låner utstyr, eventuelt andre personer. En administrator skal ha mulighet til å legge til, fjerne eller endre utstyr. I tillegg skal de ha mulighet til å veksle mellom tre oversikter, en for alle utstyrtyper, en for alle brukere registrert i systemet, og en for alle kategorier. Innholdslisten av utstyret skal være oversiktlig slik at brukere enkelt kan reservere utstyr, og få oversikt over hvilke dager/perioder utstyret er ledig for utlån. Dersom en prøver å låne et allerede utlånt utstyr, vil en få muligheten til å låne et tilsvarende utstyr. Det vil si at dersom en prøver å låne et mikroskop som ikke er tilgjengelig i valgt tidsperiode, vil en få opp alternative mikroskoper som eventuelt er ledige. En tilleggsfunksjon vil også være at man kan melde fra om utstyr er ødelagt, og legge ved notat om hva som ikke fungerer. Admin skal ha mulighet til å få oversikt

over defekt utstyr samt se vedlagt notat. Dette gir admin muligheten til å sjekke utstyret, og eventuelt fjerne denne meldingen dersom utstyret har blitt fikset.

1.2 Kontekst

Slik det er nå, er elektrolaben uoversiktlig, og ingen har kontroll på hvor utstyr er, og om det er fungerende eller ei. Studenter går inn og henter utstyret de ønsker, uten å nødvendigvis gi beskjed til noen om at de har tatt det. Dette skaper problemer dersom en trenger noe, men ikke finner det.

På grunn av dette, er prosjektet viktig å få på plass for oppdragsgiver. Løsningen vil gjøre det lettere å holde styr på alt utstyret, og gi muligheten til å reservere utstyr ved behov. I tillegg gir det en god oversikt over ødelagt utstyr, slik at det kan fikses eller erstattes.

1.3 Avgrensninger

Hovedavgrensingen for prosjektet er tid, samt vanskeligheter for å møtes for å jobbe med prosjektet sammen. Om prosjektet hadde gått over en lengre tid, kunne gruppen med sikkerhet vite at en kom i mål med alt, men grunnet tidsbegrensninger vil det være fokus på å få de grunnleggende funksjonene på plass.

Grunnet korona er gruppen på forskjellige steder i landet, og må utnytte tiden ekstra godt når en først har mulighet til å møtes. Det krever også litt ekstra av hvert gruppemedlem, i form av at det blir en del individuelt arbeid, for så å møtes for gjennomgang av hva som har blitt gjort.

En annen begrensning vil være enkelte områder hvor gruppen ikke er helt sikker på om kompetansen er tilstrekkelig god nok, da det blir brukt en del nye teknologier gruppen ikke har vært så mye borti ved tidligere anledninger. Det må derfor settes av ekstra tid for å lære seg teknologiene, for å være trygg på at alt fungerer som det skal.

1.4 Ressurser

For å kunne gjennomføre prosjektet er det nødvendig med en del informasjon om hva arbeidsgiver ser for seg, og hvilke funksjoner som skal på plass. Generell informasjon om elektrolaben sitt oppsett, samt hvilket utstyr de ønsker å få orden på er nyttig informasjon å ha før en begynner på prosjektet. Det vil også være nødvendig med serverplass som HVL stiller med, med tilhørende informasjon om hvilke teknologier som fungerer på disse. Programmeringen vil skje i Visual Studio Code, og bruke rammeverkene React.js og ASP.NET core mvc. Rapportskriving og bachelorskriving foregår i Google Docs, mens koden blir pushet til et felles repository i Github. En del av møtene vil foregå over Discord, grunnet begrensninger med tanke på korona. En annen viktig ressurs vil være oppdragsgiver, som kommer med innspill om endringer og forbedringer til prosjektet, for å få det "skreddersydd" til å passe behovet.

1.5 Oppbygging av rapporten

Kapittel 1: En introduksjon til prosjektet.

Kapittel 2: Mer dybdebeskrivelse av prosjektet.

Kapittel 3: Prosjektets design og oppsett.

Kapittel 4: Mer i dybden på designet av nettsiden

Kapittel 5: Hvordan prosjektet har blitt evaluert, og resultatet

Kapittel 6: Tilbakeblikk på prosjektet og oppbyggingen av det - hva som gikk bra og hva som var dårlig

Kapittel 7: Sammendrag av prosjektet og diskusjon om videre arbeid

Kapittel 8: Referanser/kilder som har blitt brukt i rapporten

Kapittel 9: Vedlegg til rapporten - Risikoliste, GANTT-diagram og brukermanual

2 Prosjektbeskrivelse

2.1 Praktisk bakgrunn

Dette systemet ble laget for Høgskulen på Vestlandets institutt for datateknologi, elektroteknologi og realfag. Prosjekteier ønsket et digitalt lånebibliotek som ga studentene muligheten til å låne utstyr fra elektrolaben, samt at admin-brukere fikk oversikt over utstyret.

2.1.1 Om prosjekteier

Høgskulen på Vestlandets institutt for datateknologi, elektroteknologi og realfag er et institutt med rundt 900 studenter og over 100 ansatte, som tilbyr utdanningsprogram i Førde, Bergen og Haugesund. Det tilbys utdanningsprogram på bachelor-, master-, og ph.d.-nivå. (*Institutt for datateknologi, elektroteknologi og realfag, 2020*)

2.1.2 Initielle krav

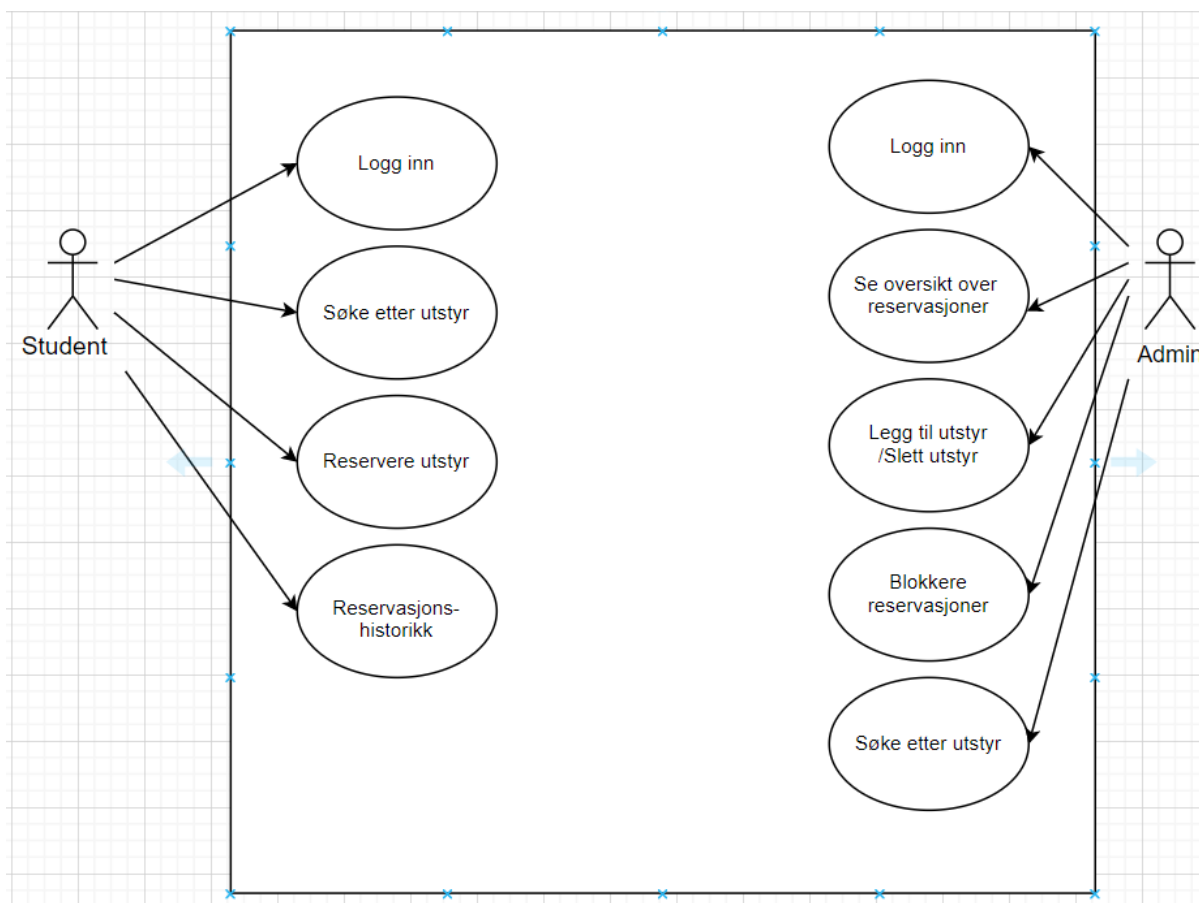
Oversikt over krav fra prosjekteier:

- Brukervennlighet for utlåner/ansatte som registrerer utstyr etc.
- Brukervennlighet for de som låner.
- Skille mellom utlån og forbruksmateriell.
- Oversiktlig database over utstyr/komponenter med tilhørende datablad.
- Søkbart innen kategorier.
- Registrering av eventuell feil og mangler ved utstyr.
- Oversikt over hvem som låner med datoer og leveringsfrist.
- Implementert under HVL sine systemer
- Mulig å vedlikeholde/modifisere/utvide systemet etter at denne oppgaven er ferdig.
- Må kunne manuelt overkjøres.

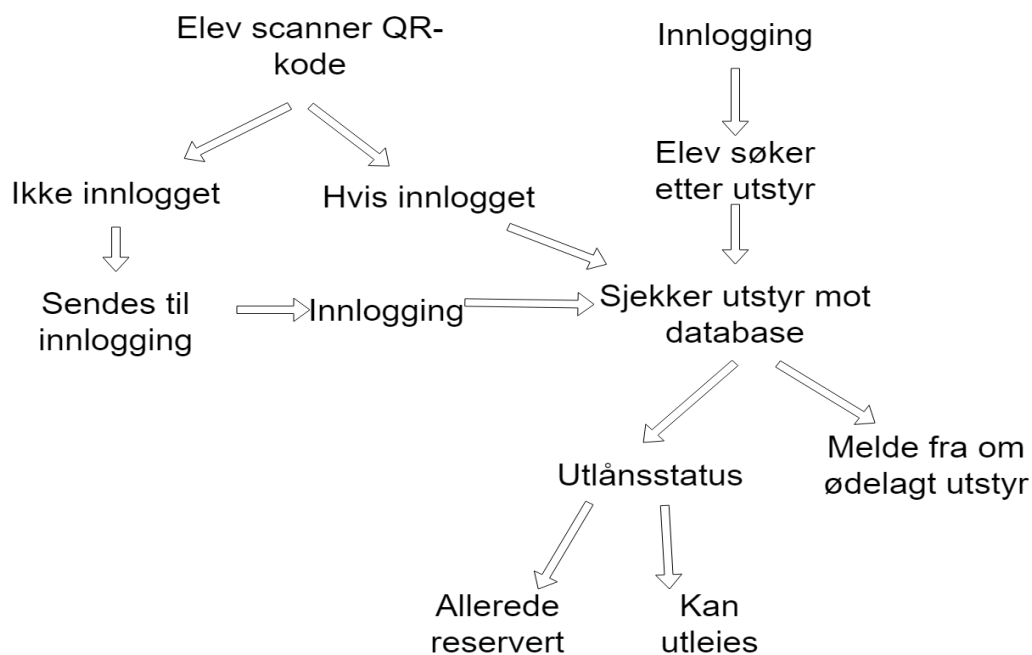
2.1.3 Initiell løsnings-idé

I listen av initielle krav fra prosjekteier kan man se at det er fokus på brukervennlighet. Det ble derfor laget en enkel designløsning med forståelige funksjoner slik at brukerne ikke ble overveldet av et komplisert UI. Funksjonene til studentbrukerne var å kunne søke etter utstyr, reservere utstyr og se brukerens historikk.

Admin-bruker skal kunne ha oversikt over alt utstyr. Planen er å fikse funksjoner for å kunne se oversikt over reserverasjoner, legge til/slette utstyr, blokkere reserverasjoner og søke etter utstyr. Blokkering av reserverasjoner fungerer slik at når utstyr trengs til en spesifikk dag, kan admin blokkere reserverasjon slik at ingen kan låne det spesifikke utstyret på den dagen. Dette hindrer at det ikke mangler utstyr på lab-dager.



Figur 1: Eksempel 1 for brukstilfeller



Figur 2: Eksempel 2 for brukstilfelle

Figur 2 beskriver brukstilfelle for reservasjonen. Da kan man se at det er en utlånsstatus, som peker på "Allerede reservert" og "Kan utleies". Reservasjonen fungerer slik at når man skal reservere utstyr, trykker man først inn en dato for reservasjonsperioden. Etter det kan man navigere i et filter, som består av kategorier og underkategorier, for finne fram til utstyret. Når utstyret velges, blir enhetene til utstyret filtrert etter om de er reservert i perioden som er valgt av brukeren. Da kan utstyret velges og bli lagt i handlekurven. Når bruker har valgt utstyret, kan bruker bekrefte reservasjonen og hente utstyret på labben. Reservasjonen har både brukeren selv og admin oversikt over.

2.2 Litteratur om problemstillingen

Hva er en god måte å organisere utlånssystemet på med tanke på sortering av de ulike utlånssystemene og logistikken rundt det? Det skal lages et digitalt utlånssystem med fokus på brukervennlighet, men hva er brukervennlighet?

2.2.1 Hva er UX?

UX referer til om et nettsted er enkelt og brukervennlig, og består av 5 prinsipper. Det handler om å hjelpe besøkende med å finne det de leter etter og oppnå sine mål. En del av

dette påvirker nettstedets brukergrensesnitt. Dårlig UX fører ofte til høyere avvisningsrater og dermed lavere konverteringsfrekvens og lavere inntjening. *(Hva er UX? Pluss 5 grunnleggende prinsipper for brukeropplevelse for nettsteder, ukjent årstall)*

2.2.1.1 Relevans

Relevans er et viktig nøkkelord i UX. Brukerne skal komme på nettstedet med et spesifikt mål. Man må da tilby UX av høy kvalitet, slik at de kan oppnå det raskere. Av den grunn vil man ikke overbelaste nettstedet med mange tilfeldige elementer som ikke er nyttig for brukerne. Relevans handler om å gi brukerne det de vil ha. *(Hva er UX? Pluss 5 grunnleggende prinsipper for brukeropplevelse for nettsteder, ukjent årstall)*

2.2.1.2 Brukervennlighet

Brukervennlighet består av faktorer som gjør nettstedet en enkel og fin opplevelse. Det er viktig å stille seg selv spørsmålene

- Er det enkelt og rasjonalisert for enkel shopping?
- Hvor lett er det å finne, fylle ut og sende inn skjemaer som brukerne skal fylle ut?
- Kan brukere finne påloggingssiden, og er prosessen enkel og grei?
- Hvor vanskelig er det for brukere å bytte fra ett blogginnlegg til et annet eller fra en produktside til en annen?

(Hva er UX? Pluss 5 grunnleggende prinsipper for brukeropplevelse for nettsteder, ukjent årstall)

2.2.1.3 Navigering

Navigering skal være synlig og ta liten plass på skjermen. Jo større nettsiden er, desto viktigere er navigeringsmenyen. Det er viktig at navigeringen forklare på en forståelig måte hvor man blir sendt til. *(Hva er UX? Pluss 5 grunnleggende prinsipper for brukeropplevelse for nettsteder, ukjent årstall)*

2.2.1.4 Tilgjengelighet

Å få tilgang til nettsted er en høflighet for personer med nedsatt funksjonsevne som kan bruke skjermlesere eller på annen måte endre online-opplevelser. Det er derfor viktig at enhver bruker av nettstedet skal kunne nå sitt endelige mål. Den beste måten å gjøre

nettstedet ditt tilgjengelig er å følge instruksjonene til “Web Content Accessibility Guideline”.
(Hva er UX? Pluss 5 grunnleggende prinsipper for brukeropplevelse for nettstader, ukjent årstall)

2.2.1.5 Kjennskap

De fleste eiere av et nettsted ønsker at det skal se unikt ut. Selv om dette er et forståelig og beundringsverdig mål, er det fortsatt viktig at nettstedet oppfyller noen av de forventede standardene for webdesign. Gitt nåværende trender på internett, har folk forventninger til hvordan det skal se ut og hvor visse elementer er sannsynlige. Hvis det ikke oppfyller forventningene, er det en sannsynlighet for at brukerne blir forvirret, frustrert og får vanskeligheter med å finne det de trenger. (Hva er UX? Pluss 5 grunnleggende prinsipper for brukeropplevelse for nettstader, ukjent årstall)

En klassisk eksempel på dagens trend er:

- Logo øverst til venstre
- Meny for navigering øverst på siden
- En rask vei til brukerens mål for nettsiden
- En søkefunksjon i overskriften

3 Design av prosjektet

3.1 Forslag til løsning

For å finne ut i hvilken retning gruppen skulle gå med design av prosjektet ble det lagt vekt på de initielle kravene samt tidligere erfaring som gruppen hadde.

3.1.1 Mobilapplikasjon

En av idéene var å utvikle en mobilapplikasjon, men det viste seg raskt at det var store ulemper med denne løsningen. En av ulempene var at brukere ville vært bundet til å bruke en smarttelefon istedenfor å kunne velge mellom smarttelefon og PC, noe som ville resultert

i et mindre brukervennlig produkt. I tillegg var det ingen av gruppemedlemmene som hadde noe erfaring med å utvikle mobilapplikasjoner for både IOS og Android.

3.1.2 Webapplikasjon

En annen idé som ble vurdert var en webapplikasjon. Alle gruppemedlemmene hadde tidligere kunnskap som kunne gjøre det lettere å utvikle en sånn løsning, og det ville i tillegg være mulig å bruke fra både mobil og PC, noe som er en fordel med tanke på brukervennlighet. Et annet krav som ble tenkt på var å ha muligheten å utvide systemet etter at oppgaven var ferdig. Dersom webapplikasjonen ble utviklet på en god måte, så ville det vært lett å utvide.

3.2 Valgt løsning

Mobilapplikasjon ble ganske tidlig i prosjektplanleggingen valgt bort, og det ble bestemt å utvikle en webapplikasjon istedenfor. En webapplikasjon gjør det lett å dele opp prosjektet i tjener- og klientside, som kunne vært en fordel når arbeidsoppgaver skulle fordeles. Når det kom til design av prosjektet bestemte gruppen seg for å lage et Web-API som var ansvarlig for å sende spørringer til databasen, og for å motta og sende svar på forespørsler sendt fra frontend.

3.3 Valg av verktøy

For å finne ut hvilke server-muligheter HVL kunne tilby ble det holdt et møte med server-teamet på IT-hjelp. På møtet ble gruppen anbefalt å bruke Microsoft Azure, dette er en Microsoft-plattform for skytjenester hvor man kan publisere webapplikasjonen og få tilgang til en database. Det ble derfor bestemt å benytte ASP.NET Core som er en Microsoft-teknologi, noe som kunne gjøre det lettere å få de forskjellige teknologiene til å samarbeide enn om det hadde blitt brukt et annet rammeverk. Det ble også gitt informasjon om muligheten til å benytte Feide som innloggingsmetode, noe som hadde vært et godt alternativ for brukere av programmet. Til frontend ble det bestemt å bruke React.js som er et javascript-bibliotek som forenkler utvikling av brukergrensesnitt, valget av React.js var basert på at det var enkelte på gruppen som hadde erfaring med det fra før, dokumentasjonen var bra, og det var mye i bruk, så det ville være lettere å finne løsninger på potensielle

problemer. Videre ble det planlagt at utviklingen av webapplikasjonen skulle gjøres gjennom eget Microsoft Azure-abonnement, og mot slutten søke om å få legge det ut på HVL sitt abonnement.

For å få en god oversikt over de forskjellige verktøyene som skulle bli brukt og lært, satt gruppen opp en liste.

- Microsoft Azure, for tjener og database.
- Microsoft ASP.NET Core, tjenersiderammeverk for kommunikasjon mellom tjener og klientside.
- React.js, rammeverk for klientside. For å forenkle prosessen med å utvikle brukergrensesnittet.
- Bootstrap, CSS-rammeverk for å gjøre sidene finere.
- Sublime Merge, for å pushe, pulle og merge
- Postman, program for å sende HTTP requests
- Visual Studio, IDE for C#
- Visual Studio Code, IDE for React.js-utvikling
- Figma, et verktøy for å planlegge brukergrensesnittet og for å kunne lettere kommunisere ideer til oppdragsgiver.
- Github, versjonskontroll, oppbevaring av kode og for å forbedre samarbeidet.
- Discord, kommunikasjonsverktøy
- Google docs, felles tekstskrivingsprogram
- Docker, for å kunne kjøre SQL server på mac

3.4 Prosjektmetodikk

3.4.1 Utviklingsmetodikk

For prosjektet var tanken å benytte en agile utviklingsmetodikk. Målet til gruppen med å bruke agile utvikling var blant annet å raskt kunne ha et fungerende program, noe som gjør det lettere for oppdragsgiver å gi tilbakemelding under utvikling slik at nødvendige endringer kunne gjøres. Gruppen ville også teste applikasjonen kontinuerlig, slik at det lettere ville være mulig å finne ut når eventuelle problemer inntreffer, da feilsøkingen ville vært begrenset til nylige endringer.

3.4.2 Prosjektplan

For å få en god oversikt over hva som trengte å gjøres i prosjektet, satte gruppen opp et Gantt-diagram (Se Appendix 9.2). På diagrammet ble frister for rapportskrivning lagt til, sammen med hovedprogrammeringsoppgavene som måtte gjøres. Tidsanslaget for programmeringen var veldig vanskelig å forutse, spesielt med mangel på erfaring. For å gjøre det lettere var tanken å dele opp større oppgaver i mindre gjøremål, og det ble bestemt å ha ukentlige møter hvor gruppen diskuterte hvilke av de mindre gjøremålene som måtte jobbes med, fremfor å lage en strukturert plan som kunne blitt vanskelig å følge. Et annet viktig aspekt var å planlegge hva hver person på gruppen skulle gjøre under utviklingen, noe som medførte at ikke alle ble like godt kjent med de forskjellige delene av prosjektet, men som ga hver person en klar idé om hva som måtte læres. Det var forventet at hver person skulle ha en god nok forståelse av sitt spesialområde til å kunne forklare hva som ble gjort til de andre. I tillegg var det viktig at hvert av gruppemedlemmene dokumenterte hva de hadde gjort, i form av kommentarer i kildekoden. Disse skulle være beskrivende og godt formulert.

3.4.3 Risikovurdering

For å håndtere risiko ble det lagd en risikoliste (Se Appendix 9.1) som inneholdt potensielle problemer som kunne oppstå underveis, samt en plan for hvordan disse eventuelt skulle håndteres. Det var likevel en stor sannsynlighet for at det fantes problemer som gruppen ikke hadde tenkt på, enten grunnet mangel på kunnskap, eller på grunn av eksterne faktorer som det var vanskelig å forutse. Disse problemene måtte løses etterhvert som de oppsto. Det var høy risiko knyttet til mangel på kunnskap rundt de forskjellige teknologiene som skulle brukes, det var derfor stor fokus på å tilegne seg kunnskap for å minimere risikoen.

3.5 Evalueringsplan

Planen for evaluering av prosjektet var først å vurdere om programmet var teknisk bra. For API'et var tanken å bruke Postman for å teste om HTTP-forespørsler returnerte det de skulle, og for React.js skulle det skrives enhetstester. Når det kom til brukergrensesnittet ville gruppen hovedsakelig fokusere på tilbakemelding fra oppdragsgiver, men håpet også på å få utført brukertester med bekjente som ikke nødvendigvis hadde en elektrobakgrunn. Tanken

bak dette var at selv om det i hovedsak er elektrostudenter som skal bruke applikasjonen, så er det ikke en selvfølge at nye studenter på elektro har kunnskap om utstyret som blir utleid.

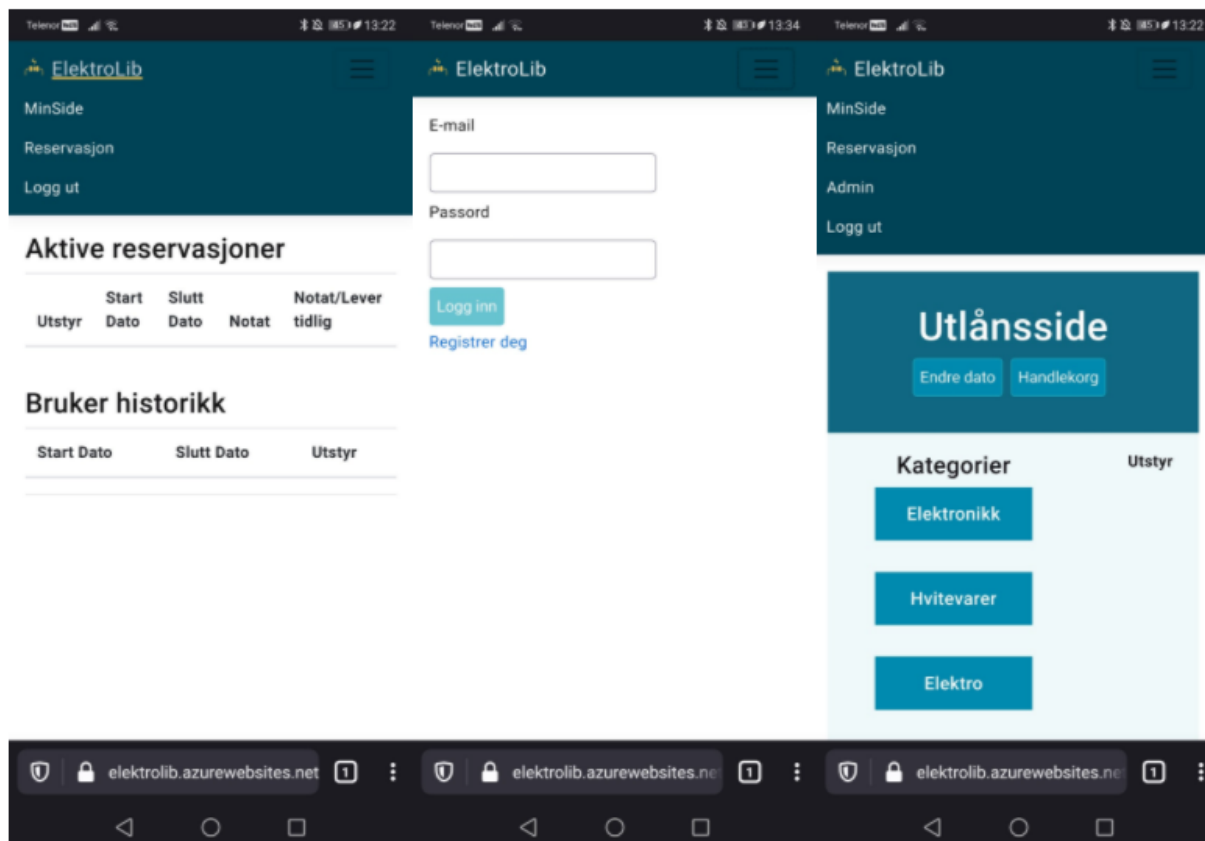
4 Detaljert Design

4.1 UI

Målet med prosjektet var å finne en god måte å organisere utlånssystemet på, samt lage et brukervennlig design. I kapittel “2.2 Litteratur om problemstilling” står det om hva UX er. UX har følgende fem prinsipper:

- Relevans
- Brukervennlighet
- Navigering
- Tilgjengelighet
- Kjennskap

Generelt handler det om systemet er lett å bruke i form av navigasjon, enkelt å nå formålet, og tilgjengelighet. For navigering ble det lagd en navigasjonsbar som inneholder relevante sider på høyre side, og logo på venstre side. Brukerne har da oversikt over hvilken sider de har tilgang til. Navigasjonsbaren endrer seg etter om man er en vanlig bruker eller en adminbruker. Bilde 1 viser hvordan navigasjonsbaren ser ut i de forskjellige tilstandene. På venstre side ser man et eksempel på når du er en vanlig bruker, der man har tilgang til ‘Min side’ og ‘Reservasjonssiden’. På høyre side viser hvordan den ser ut når du er admin, der du tillegg har tilgang til ‘Admin’-siden i navigasjonsbaren. Midten av bildet viser hvor du kommer når du besøker nettsiden.



Bilde 1: Eksempel på hvordan navigasjonsbaren er i forskjellige tilstander

Det første man støter på er logg inn-siden. Der kan man logge inn på en eksisterende brukeren eller registrere seg. Vanlige brukere har tilgang til vanlige sider som 'Min side' og 'Reservasjon'. Dette er to sentrale sider man bruker. Adminbrukere har i tillegg tilgang til 'Admin', hvor brukeren har ekstra funksjoner. Dersom man allerede er innlogget, kommer man direkte til 'Min side'.

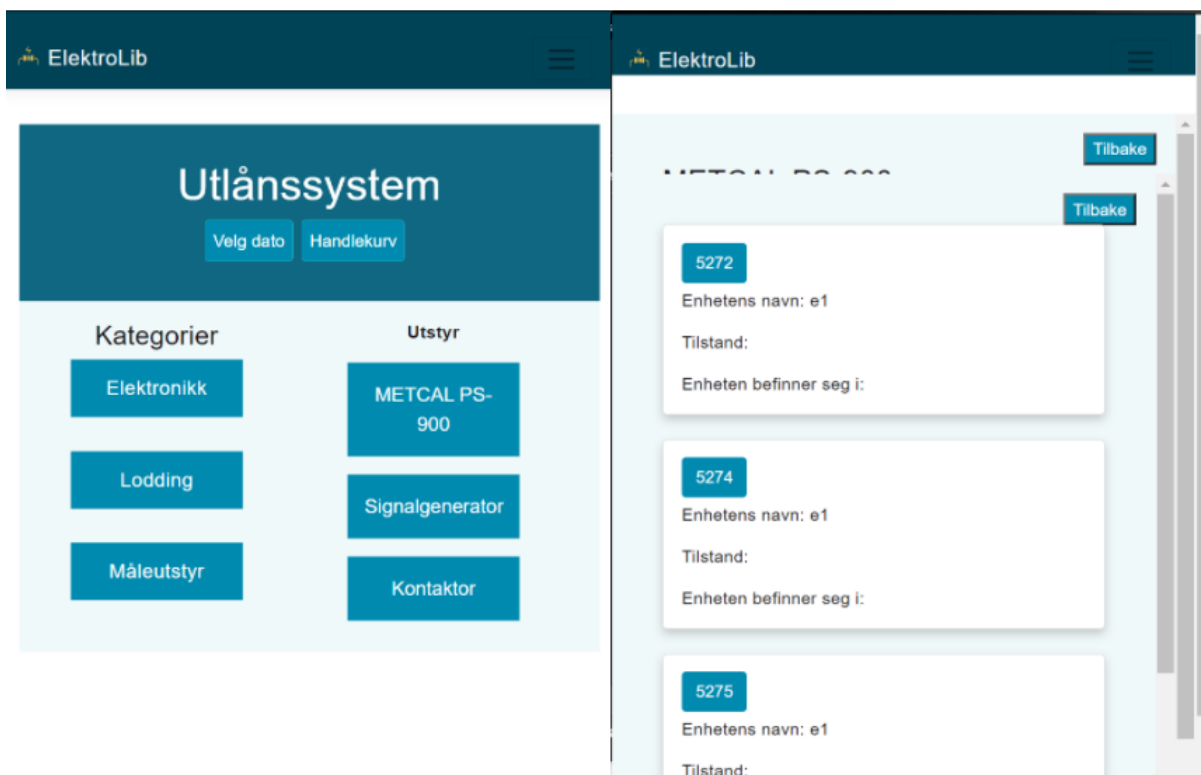
Brukere av systemet vil vanligvis bruke systemet på en mobilenhet. Det er derfor satt fokus på å gjøre 'Reservasjon' og 'Min side' enkel å bruke på mobil. Det er ikke nødvendig på 'Admin'-siden, da den hovedsakelig vil bli brukt av PC-brukere.

4.1.1 Reservasjonsside

På reservasjonssiden får du muligheten til å reservere utstyr. Man kommer første til en side hvor man setter inn en start- og sluttdato for perioden man ønsker å låne et utstyr i. Deretter kommer man til hovedsiden for reservasjon. Den består av tre deler. Som man kan se i Bilde 2 består øvre delen av det første bildet en overskrift og to knapper. 'Endre dato'-knappen

sender brukeren tilbake til siden hvor man velger dato. 'Handlekurv'-knappen sender brukeren til siden hvor man kan bekrefte reservasjonen for valgt utstyr. Til venstre på bildet finner man kategoriene. Disse er fordelt på hoved- og underkategorier. Hovedkategoriene er de blå knappene, og om man trykker på en av de, vil de tilhørende underkategoriene komme som en drop-down. Dersom man trykker på en av underkategoriene, vil den filtrere ut utstyrstyper som hører til den, som man kan se til høyre på bildet. Siste delen er altså utstyrstypene. Når man trykker på en av de, vil Popup-komponenten bli kalt og en side med informasjon om utstyrstypen og en knapp med "Vis alle ledige enheter".

Popup-komponenten er en komponent som genererer en side over innholdet. Dette er for at man ikke skal forsvinne fra reservasjonssiden. Trykker man på knappen "Vis alle ledige enheter" vil en ny Popup-komponent bli kalt som inneholder alle enheter som hører til utstyrstypen, som kan se til høyre i Bilde 2. Innholdet er en liste over alle enheter som tilhører utstyret og er fordelt med kort. Kortet inneholder Id, enhetens navn, tilstand og hvor den befinner seg. Tilstand er om denne enheten er ødelagt og ikke bør lånes ut. Etterhvert skal det finnes en løsning på å filtrere ut ødelagt utstyr også. Hvor enheten befinner seg beskriver hvilket rom, skap, hylle og boks enheten ligger i. Om det ikke er relevant vil det ikke stå på kortet. Etter å ha valgt et utstyr kommer man tilbake til hovedsiden igjen. Da kan man se at en enhet er valgt på forsiden. Deretter kan man trykke seg inn på 'Handlekurv' for å bekrefte reservasjonen. Når den er bekreftet blir brukeren sendt til 'MinSide', hvor utstyret ligger i reservasjonslisten.



Bilde 2: Forsiden av reservasjonssiden og eksempel på Popup-komponent

Det er laget en logikk for hvordan man skal filtrere utstyr som allerede er reservert i en spesifikk periode. Løsningen består av få faktorer og ble implementert som et API-kall.

Utstyr vil bli filtrert vekk om:

- Startdato eller sluttdato er innenfor perioden utstyret er lånt i
- Startdato er før og sluttdatoen er etter perioden utstyret er lånt i

4.1.2 Admin

Adminsiden er kun tilgjengelig for brukere med administratorstatus. Her kan brukeren navigere mellom 3 forskjellige oversikter, en for utstyrskategorier, en for brukere, og en for kategorier. Disse er organisert ved hjelp av Bootstraps' komponent "Breadcrumb". Denne komponenten kan sees på Bilde 3 under, med gyllen bakgrunnsfarge. En får da navigert mellom de forskjellige oversiktene på en interaktiv og dynamisk måte. I tillegg har man to knapper øverst på siden, en for å legge til ny utstyrskategori, og en for å legge til ny kategori.

Oversikten over utstyrskategori gir muligheten til å slette utstyrskategorier, eller trykke innpå den enkelte av disse for å få opp alle de individuelle utstyrene under utstyrskategorien. Disse utstyrene kan også endres hver for seg.

Id	Navn	Beskrivelse	Produsent	Kategorinavn	Slett
20	Kjøleskap	Samsung-kjøleskap med innebygd isbitfunksjon	Samsung	Diverse	Slett
21	Test	kjdnad	McDonalds	Test	Slett
22	A	hkjawh	nmdan,	Test	Slett

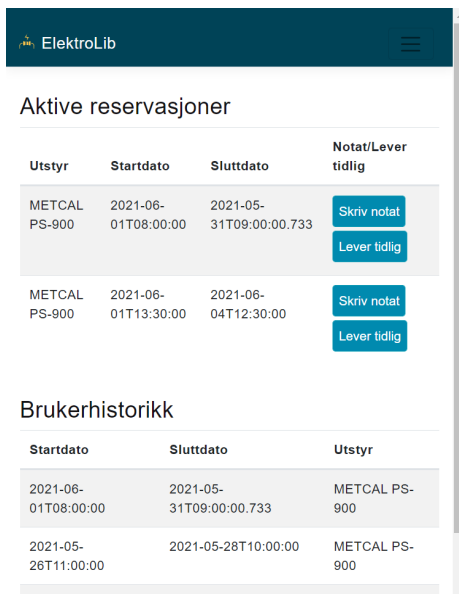
Bilde 3: Skjermdump av Admin-side med oversikt over Utstyrskategori

Brukeroversikten fungerer på en tilsvarende måte, men har ikke funksjonen hvor en kan trykke på en enkelt rad for å endre underenhetene enkeltvis. Oversikten viser for- og etternavn, e-post og om brukeren har administratorrettigheter eller ei. Siden har to knapper, en for å slette brukeren, og en for å gjøre brukeren til administrator.

Kategorioversikten viser hvilken id og navn kategorien har, samt id'en til forelderen og tilhørende navn. Dersom foreldre-id'en er lik 0, er dette en forelder uten noen overkategori.

4.1.3 Min side

'Min side' er en enkel side som inneholder brukerens utlånshistorikk og en liste over aktive reserverasjoner som man kan se på Bilde 4. Listen over aktive reserverasjoner har man oversikt over når utstyret skal leveres, muligheten til å notere skader på utstyret og muligheten til å levere utstyret tidligere. Leverer man tidlig blir sluttdatoen endret til datoen og tiden man bekrefter kallet.



Bilde 4: Min side

Målet med denne siden er at brukere skal kunne ha god oversikt og kontroll over sine reserverasjoner. Det skal være lett å skille mellom aktive reserverasjoner og brukerhistorikken.

4.2 Klientside-rammeverk

React JS blir brukt som rammeverket i dette prosjektet. "React.js er et JavaScript-bibliotek for lagning av brukergrensesnitt." (*Getting started, 2021*) Fordelen med React.js er at den er deklarativ, som betyr at man kan ha kontroll over flyten og tilstanden til applikasjonen. React gjør det enklere å lage interaktiv UI. Deklarative synspunkter gjør koden forutsigbar og enklere å debugge. Det som menes med uttrykket deklarativ, er at regler og relasjoner i systemet ikke kan endres. (*React, 2021*)

React er komponentbasert der hver komponent har sin egen tilstand. Det fungerer slik at om man har en komponent som viser en liste av bøker og en annen komponent som viser en liste av filmer, kan man ha en komponent som kaller på de to komponentene som viser en side av innholdet. Siden komponentlogikken er skrevet i Javascript istedenfor maler, kan man enkelt passere data gjennom applikasjonen og holde tilstanden ut av dokumentobjektmodellen. (*React, 2021*)

4.3 Server og API

4.3.1 Deployment

Hele applikasjonen vil bli lagt ut på Azure sin plattform som en app service. App service er en tjeneste som gjør det raskt og enkelt å få en nettside ut på Azure, og håndterer blant annet sikkerhet og skalering. I tillegg tilbyr app services kontinuerlig “deployment” som gjør at hver gang noe blir oppdatert i githubprosjektet så oppdateres nettsiden. Gjennom Azure fikk gruppen også tilgang til en SQL-database. Azure tilbyr studenter begrenset tilgang til tjenestene sine gratis, så gruppen brukte Azure kun for å teste hvordan nettsiden så ut på mobil, og ellers ble en lokal database og server brukt.

4.3.2 Web-API

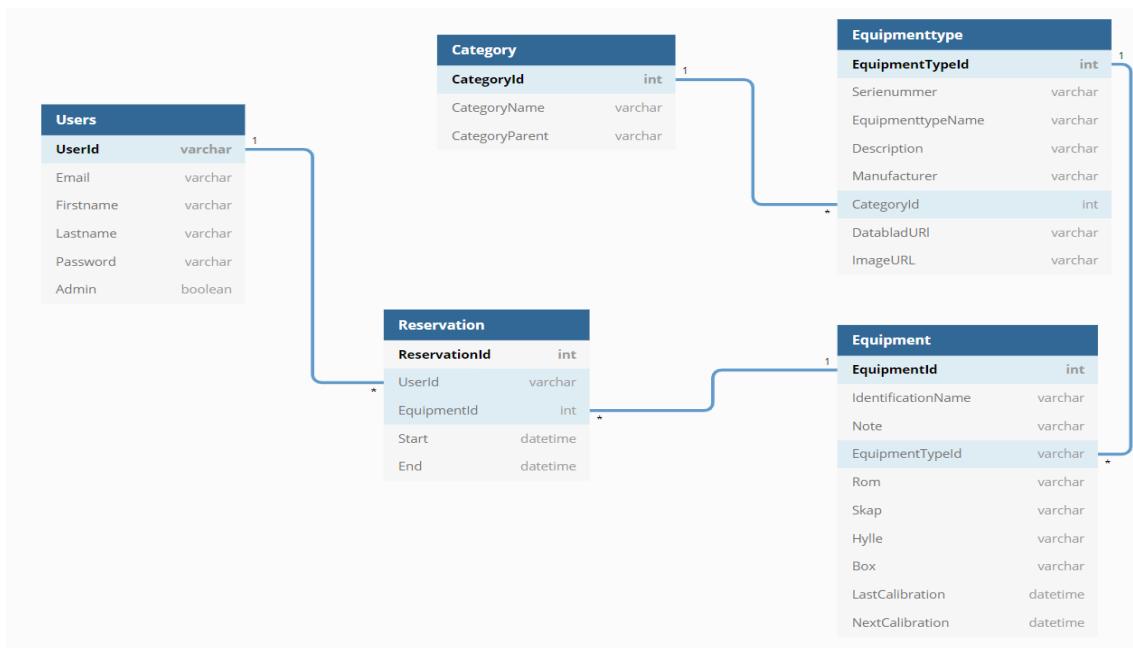
Web-APIet ble utviklet med ASP.NET Core som er et rammeverk som gjør det enklere å designe et web-api på en strukturert måte. Hovedansvaret til APIet er å motta spesifikke HTTP-forespørsler og sende tilbake strukturert data. Det er særlig to hovedmapper med kode som er viktig, disse er modeller og kontrollere. Modeller beskriver hvordan hvordan databasetabellene skal se ut, og kontrollere spesifiserer hvordan forespørsler og svar må være strukturert. En pakke som gjør store deler av APIet enklere å utvikle er Entity framework Core, dette er en ORM(object-relational mapper) som blant annet forenkler kommunikasjonen med databasen, og gjør det mulig å designe databasetabellene basert på modeller og generere kode for kontrollere.

Når det blir sendt en forespørsel om å for eksempel få alt utstyr i databasen, så mottar utstyrskontrolleren forespørselen, og gjør deretter en spørring til databasen og sender tilbake dataen som JSON via en HTTP-respons til frontend.

4.3.3 Database

Databasedesignet har forandret seg mye i løpet av prosjektet, etterhvert som oppdragsgiver har gitt tilbakemeldinger på hva som må legges til eller endres. Det endelige designet vises i Figur 3 og er satt opp sånn at “EquipmentType” er en overordnet utstyrskategori for utstyr og “Equipment” inneholder det som er unikt for like utstyr, som for eksempel om utstyret er

ødelagt, hvor det ligger og når det ble kalibrert. Brukere er delt opp i to grupper, admin og studenter. Adminbrukere har tilgang til alt av data og kan også legge til og endre data, mens studenter kun har tilgang til å se dataen og legge til reserverasjoner. Kategorier er satt opp som en trestruktur hvor "Categoryparent" peker på forelderkatogorien sin id.



Figur 3: Endelig databaseskjema

4.3.4 Login og registrering

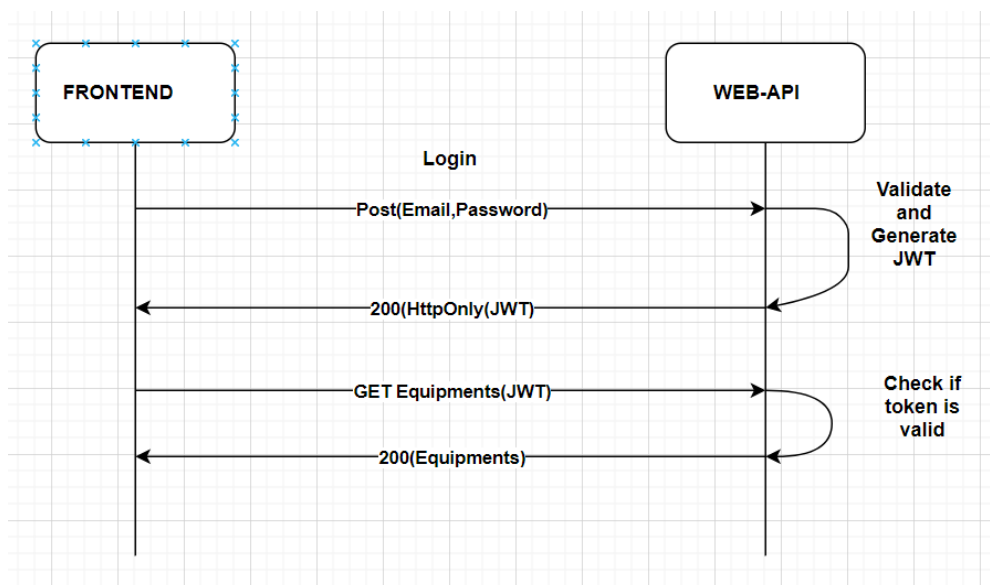
Autentisering og autorisering blir håndtert ved hjelp av JWT (Json Web Tokens).

“JWT (JSON Web Token) er en JSON-basert tokentype som kan brukes både for autentisering av en bruker, og for å kunne overføre informasjon om brukeren på en sikker og enkel måte. Et JWT kan inneholde informasjon om for eksempel brukernavn, utstedelsestidspunkt for tokenet, hvor lenge tokenet er gyldig, og annen informasjon som kan være relevant.

Informasjonen i tokenet kalles "claims". Sikkerhetsmessig baserer JWT seg på digitale signaturer.”(Bekk, 2017). Når en bruker logger inn så blir epost og passord verifisert, og det blir generert en JWT som blir plassert i en HTTPOnly Cookie. Dette er en type cookie som gjør at frontend javascript ikke har tilgang til innholdet av cookien, men den blir sendt med alle forespørsler til WEB-APIet, så det kan autentisere brukeren og sjekke at JWTen ikke har

blitt endret. I Figur 4 ser man et eksempel på en bruker som logger inn, får en token og sender med denne for å få tak i alt utstyr.

En stor sårbarhet med JWT er at hvis noen andre får tak i JWTen så har de full tilgang til alt den spesifikke brukeren har tilgang til. En måte å minimere de potensielle problemene dette kan forårsake er ved å la tokenen kun være aktiv i en kort periode, og generere en refresh token. Refresh tokens er en token som lagres i databasen og er typisk aktiv over en lengre periode. Når JWTet har blitt inaktiv så sjekker APIet om refresh token fortsatt er gyldig, og sender en ny kortlevd JWT. Det var en del problemer rundt implementeringen av refresh tokens, og det ble derfor bestemt å prioritere andre oppgaver. Løsningen ble istedenfor å generere en token som var aktiv i 10 dager. Dette sikrer at tokenen ikke lever for lenge med tanke på sikkerhet, men forhindrer at brukeren må logge inn alt for ofte, da dette hadde gått utover brukervennligheten.



Figur 4: Eksempel på login og henting av data

5 Evaluering

5.1 Evalueringsmetode

Når det kommer til evaluering av designet til systemet har gruppen fått frirom til løse dette selv. Evalueringen skjer ved ukentlige møter med oppdragsgiver. Oppdragsgiver kommer med tilbakemeldinger om hva han liker og innspill om hva han ønsker å se mer av.

I uke 21 ble det satt opp brukertesting av utlånssystemet. Hensikten med dette var å få oversikt over brukeropplevelsen av dette systemet. Det ble valgt ut noen vanlige brukere som skulle prøve systemet og en adminbruker, som er arbeidsgiveren selv. De tre brukerne fikk muligheten til å prøve systemet helt ut, i form av leting etter bugs og finne feil. Målet med testingen av admin var at han kunne få muligheten til å oppleve hvordan systemet fungerte, og komme med tilbakemeldinger om hva han likte eller hva som må forbedres.

Postman ble brukt til å evaluering av prosjektets API. Postman er en samarbeidsplattform for API-utvikling. Den forenkler hvert trinn i å bygge et API, og effektiviserer samarbeidet slik at man kan opprette bedre APIer (About Postman, 2021). Ved å sende API-forespørsel får man oversikt over hvordan man kan bruke forespørslene det i prosjektet. Dette hjalp testing av 'Reservasjon', diverse funksjoner i 'Admin' og 'Min side'.

5.2 Evalueringsresultat

Som skrevet i "5.1 Evalueringsmetode" ble det utført brukertesting av systemet. Det var to studenter som testet som vanlig brukere, mens oppdragsgiver testet som admin. De to studentene er ikke fra elektrolinjen, men fra andre studier. Hensikten var at webapplikasjonen skulle være brukervennlig for alle som ville låne, og lett å forstå for folk som ikke går på elektrolinjen. Det ble mottatt flere gode innspill fra testerne. Følgende innspill på reservasjonssiden:

- Knappene og fonten under kategori er ulik knappene og fonten på utstyr. Dette ga et inntrykk om at det var rotete.
- Gjøre oversikten på meny til ethvert utstyr mer oversiktlig.

- Popup-siden som kommer når man trykker “Liste over alle ledige enheter” gjemmer ikke hele siden, noe som fører til at det ser rotete ut.
- Når man velger utstyr blir man tatt tilbake til hovedsiden. Det ble usikkerheter om utstyret ble valgt. Det ble da altså ønske om en oversikt over den valgte enheten til enhver tid.
- Skille vekk utstyrsenheter som allerede er utlånt.
- Ønske om et ikon av en handlevogn på handlekurv-knappen

Man kan se at det er en del CSS-koding som må bli gjort for å fikse flere innspill, men også noe funksjonelt. Filtrering av allerede lånte utstyr ble veldig kritisk siden det ville blitt mye rot om to personer lånte den samme enheten, men bare en av de fikk låne den. Det ble førsteprioritet for gruppen, og det ble løst ved å filtrere reserveringer i backend. Videre på følgende innspill på ‘Min side’:

- Litt mer informasjon om utstyret man låner.
- Større boks når man skal rapportere en feil på utstyret man låner.
- Ønske om en knapp som registrerer at utstyret er ødelagt, som fører til at den blir fjernet fra reserveringssiden.
- Lever tidlig fungerer ikke.

Her er det altså å få ‘lever tidlig’-knappen til å fungere. Dette er ikke veldig kritisk siden man har informasjon om når utstyret skal bli levert. Informasjon om utstyret man velger kan bli gjort på en mer ryddig måte i en senere versjon. Det ble diskusjon om knappen som registrerer at utstyret er ødelagt. Dette kan bli gjort, og det ble også implementert en funksjon i backend for filtrering av ødelagt utstyr.

Resterende innspill:

- Ønske om en “glemt passord”-knapp
- Trykker man på tilbakeknappen i nettleseren, ble man logget ut. Det må fikses.
- Obs på å skille norsk og engelsk. Om siden skal være på norsk, bør man endre de engelske ordene.
- Når man har reservert et utstyr, kommer man til MinSide. Av og til vises ikke reserveringen med en gang før man har lastet siden på ny.
- Skaleringen på siden er veldig rotete på mobil.

Denne webapplikasjonen skal være mobilvennlig. Det må derfor bli fokus på skaleringen for mobilbrukere. Fra oppdragsgiver ble det ingen innspill på 'Admin'-siden og han er fornøyd med designet av webapplikasjonen. Det skal likevel bli flere møter for å vise den nylige versjonen gruppen har fått på plass.

Bruken av Postman har vært verdifull for gruppen. Den har hjulpet oss med å teste API-kallene i prosjektet. Når det trengtes et API-kall, testet vi først i Postman for å se hva man trengte for å utføre forespørselen. Deretter ble det lagt inn i koden.

6 Diskusjon

React.js-rammeverket har vært veldig nyttig gjennom hele utviklingsfasen, og ble godt likt av hele gruppen. Eneste ulempen var at det var mye nytt å lære seg, noe det gikk mye tid til. ASP.Net Core har også fungert veldig bra, men det gikk mye tid til å prøve å få til RefreshTokens, som til slutt ble vraket da det ikke fungerte som det skulle, og bare skapte problemer. Ellers var det et bra valg som har hjulpet mye med å hente data fra databasen, for så å bli vist i UI'et. Azure-serveren har i seg selv vært bra, men med abonnementet studenter har, har en kun 100 dollar en kan bruke, og overstiger en dette går det av egen lommebok. En konsekvens av dette var at alle gruppemedlemmer måtte kjøre prosjektet på localhost under utvikling, og prosjektet ble sjeldent oppdatert på Azure-serveren, for å forhindre å overstige det som var inkludert i abonnementet. I tillegg var det et problem at om man tidligere hadde vært innom en eldre utgave av nettsiden på Azure-serveren, så var det denne som ble vist for brukeren selv etter det var lastet opp en ny utgave. For å fikse dette, måtte en åpne nettsiden i en ny browser, hvor en igjen støter på samme problem dersom Azure-serveren blir oppdatert igjen. En litt mindre viktig konsekvens er at å oppdatere serveren til den nyeste prosjektversjonen tar en del tid, da deployment tar noen minutter, og at det ofte er unødvendig mye error-handling, hvor en etterpå må forsøke å deploye på ny.

Dersom gruppen skulle gjort prosjektet på ny, ville det blitt prioritert å begynne med frontend-arbeid tidligere i utviklingsfasen. Det var mye nytt som måtte læres, og det gikk mye tid til å sette seg inn i ny teknologi, fremfor å prøve seg på "learning by doing", som

antakeligvis hadde vært en bedre løsning. I tillegg burde gruppen hatt brukertester på et tidligere stadiet, da resultatene av disse viste at en del av CSS-arbeidet ikke var godt nok når nettsiden både skulle fungere bra på datamaskin og mobil. Det kom også en del kommentarer på mindre justeringer som ville fått resultatet til å se bedre ut, men som det ikke vil være tid til i sluttfasen, da viktigere ting prioriteres. Gruppen burde også sett mer på de initielle kravene. Noen av kravene ble glemt litt ut i prosjektet, da gruppen hadde dannet seg et bilde for hvordan nettsiden skulle se ut, og hvilke funksjoner den skulle ha. En konsekvens av dette, er at ikke alle de initielle kravene er på plass, selv om det meste er som det skal.

7 Konklusjon og videre arbeid

Problemstillingen handlet om hvordan man på en god måte kunne organisere et utlånssystem på med tanken på sortering av de ulike utlånproduktene og logistikken rundt det. Det ble også satt opp følgende initielle krav fra oppdragsgiver:

- Brukervennlighet for utlåner/ansatte som registrerer utstyr etc.
- Brukervennlighet for de som låner.
- Skille mellom utlån og forbruksmateriell.
- Oversiktlig database over utstyr/komponenter med tilhørende datablad.
- Søkbar innen kategorier.
- Registrering av eventuell feil og mangler ved utstyr.
- Oversikt over hvem som låner med datoer og leveringsfrist.
- Implementert under HVL sine systemer
- Mulig å vedlikeholde/modifisere/utvide systemet etter at denne oppgaven er ferdig.
- Må kunne manuelt overkjøres.

Det ble laget et system med tre dynamiske sider. Reservasjonssiden for utlån av utstyr og forbruksmateriell. 'Min side' slik at brukere selv kan ha oversikt over både aktive reserverasjoner og historikk. 'Admin' for å endre, slette og registrere utstyrstyper, kategorier og hver utstyrsenhet til systemet, samt ha oversikt over disse.

Før brukertesting var det noen intielle krav som ikke hadde blitt utført. Filtrering av utstyr som allerede var på utlån var gruppens hovedfokus etter brukertesting, og det ble utført med en funksjon i backend. På 'Admin'-siden kan man registrere kategorier, underkategorier, utstyrstyper og enheter til utstyrstypene. Dette gjør systemet oversiktlig og lett å bruke for registrering av utstyr og søking etter utstyr ved hjelp av kategoriene. Derimot er det ikke fikset slik at man kan skille forbruksmateriell med utstyr. Dette kan bli fikset i senere versjon. Admin skal kunne ha oversikt over hvem som låner med dato og leveringsfrist. Dette kan man bare gjøre ved å utføre et API-kall i postman. I en senere versjon av prosjektet kan det bli implementert.

På 'Min side' har man oversikt over utlånshistorikk og aktive reserveringer. Det er enkelt å se hva man låner, men ut fra brukertestene var det ønske om mer informasjon om utstyret man låner på siden. Det er enkelt å rapportere feil ved utstyr, men 'Lever tidlig'-knappen fungerer ikke. På både 'Min side' og reserveringssiden har det vært et problem med skalering for mobilbrukere. Dette gjør siden uprofesjonell og lite brukervennlig. Før den eventuelt blir implementert på HVL sin server, må dette fikses. Webapplikasjonen er brukbar og man kan låne utstyr en om gangen uten problem. Gruppen har ikke nådd hele målet enda, da mobilskaleringen gjør webapplikasjonen lite brukervennlig, og at den ser mindre fin og profesjonell ut.

For videre arbeid blir fokuset å rette på ting som ikke fungerer komplett, men også å fikse skaleringen for mobilbrukere. I slutten av uke 22 skal gruppen jobbe mer på prosjektet og ha et møte med oppdragsgiver om det avsluttende arbeidet. Da blir fokuset skalering av sidene for mobilbrukere, implementasjon av en liste av hvem som låner med dato og leveringsfrist, og implementasjon av forbruksmateriell. Dette arbeidet skal sikre at vi når målene for arbeidet og de initielle kravene. Gruppen skal også rydde opp og kommentere i koden slik at det er oversiktlig for neste person å sette seg inn i, dersom applikasjonen senere skal vedlikeholdes, modifiseres eller utvides.

8 Referanser

8.1 Veiledninger/Litteratur

Institutt for datateknologi, elektroteknologi og realfag (2020). Tilgjengelig fra:

<https://www.hvl.no/om/organisering/fin/institutt-for-datateknologi-elektroteknologi-og-realfag/> (Hentet: 16.04.2021)

React (2021). Tilgjengelig fra:

<https://reactjs.org/> (Hentet: 16.04.2021)

Getting started (2021). Tilgjengelig fra:

<https://reactjs.org/docs/getting-started.html> (Hentet: 16.04.2021)

Hva er UX? Pluss 5 grunnleggende prinsipper for brukeropplevelse for nettstader (ukjent årstall)

Tilgjengelig fra:

<https://secinfinity.net/nor/hva-er-ux-pluss-5-grunnleggende-prinsipper-for-brukeropplevelse-for-nettsteder/> (Hentet: 29.04.2021)

About Postman (2021) Tilgjengelig fra:

<https://www.postman.com/company/about-postman/> (Hentet: 21.05.2021)

Bekk(2017) *JWT(JSON Web Token)* Tilgjengelig fra:

<https://radar.bekk.no/tech2017/arkitektur-og-plattform/jwt-json-web-token> (Hentet: 27.05.2021)

9 Appendix

9.1 Risikoliste

Risiko	Sannsynlighet	Innvirkningsgrad	Alvorlighet	Håndtering
Lite kunnskap om Azure	2	5	10	Vi må lese oss opp på hvordan man bruker Azure, og jevnlig teste og feile for å få en forståelse over teknologien
Lite kunnskap om React JS	3	5	15	Jevnlig gå gjennom kurs om react js på nett, og kode mye selvstendig i forkant
At websiden ikke blir ferdig	3	5	15	Huske og sette tid til arbeid med prosjektet og ha en god balanse mellom skriving av rapport og programmering
Overføre prosjektet fra gruppens Azure til HVL sin Azure	3	4	12	Må i god tid vite at prosjektet kan kjøres på begge server
Lite kunnskap om ASP.NET	2	5	10	2 personer har allerede et kurs innen ASP.NET, men man må fortsatt drille på det for å holde kunnskapen fersk

Gruppen får problemer med å dele prosjektet med hverandre	1	2	2	Om dette skjer, kan vi dele filen med en minnepenn
Prosjektet vil ikke kjøres	2	5	10	Bruke agile metoden for å ha oversikt om når eventuell når prosjektet ikke kjøres, og fikse det så raskt så mulig
Rapporten blir ikke ferdig	1	5	5	Sette av ekstra tid i helgene for rapportskrivning
Lite fokus på agile utvikling	3	4	12	

9.2 GANTT-diagram

uke	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Prosjektplanlegging	█	█	█	█				█	█											
UI-skisser		█																		
Eksamensforberedelse					█	█	█													
OA 7-1 og OA 7-2								█	█											
Påskeferie										█										
Forprosjektrapport											█	█								
Zoom-presentasjon													█							
Implementering av backend													█	█	█					
Statusrapport														█	█					
Implementering av frontend														█	█	█	█			
Enhetstesting																	█	█		
Utkast-1 og blogg innhold											█	█	█	█	█	█	█	█		
Utkast-2 og EXPO poster											█	█	█	█	█	█	█	█	█	
Endelig rapport											█	█	█	█	█	█	█	█	█	█

9.3 Brukermanual

Dette er en oversikt over hvordan man bruker diverse funksjoner på nettsiden.

9.3.1 Min side

Aktive reserveringer

- Liste over aktive reserveringer
- Inneholder en knapp hvor man kommer til en side der man kan notere skader på utstyret.

Brukerhistorikk

- Liste over alle brukerens reserverasjoner

9.3.2 Reserverasjon

Inneholder diverse sider:

Velg dato

- Velg en startdato og sluttdato
- Man må velge om man skal videre til reserverasjonssiden

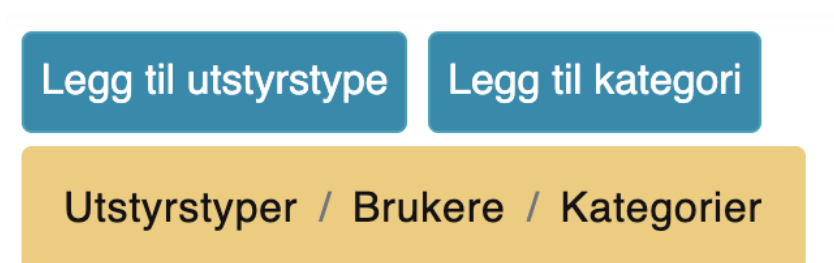
Reserverasjonssiden

- Inneholder en liste over kategorier, utstyrstyper og handlekurv
- Trykke på en av kategoriene vil filtrere en liste over utstyrstyper
- Trykke på en utstyrstype vil poppe opp en boks med info om utstyret og alle ledige enheter
- Velg en enhet og gå til handlekurv for å bekrefte reserverasjonen

9.3.3 Admin

På adminssiden har en flere muligheter. Det er to knapper, en for å legge til utstyrstype/utstyrskategori, og en for å legge til kategori. Under kan man navigere mellom hvilke tables man ønsker å se, disse er Utstyrstype, Brukere og Kategorier. Default er satt til Utstyrstype.

Noe som er ganske viktig å ha i bakhodet før en begynner å skulle legge til utstyr, er at alt utstyr som blir lagt til havner i et "tre" hvor Kategori > Utstyrstype > Utstyr. Det vil derfor være mest logisk å begynne med å legge til kategori, så utstyrstype i kategorien, og deretter hvert enkelt utstyr i utstyrstypen.



Legg til kategori-knapp

Trykker man på "Legg til kategori"-knappen vil en bli sendt til følgende side:

Legg til ny kategori

Her kan du legge nye kategorier til i databasen.

Lagre

Tilbake til adminsiden

Her fylles det ut informasjon som kategorinavn, samt om kategorien skal ha en forelder (og evt. hvilken forelder).

Legg til utstyrstype-knapp

Trykker en på "Legg til utstyrstype"-knappen vil en bli møtt med denne:

Legg til ny utstyrstype

Her kan du legge ny utstyrstype til i databasen.

 ▼

Her vil det være mulig å legge til nye utstyrstyper. Man setter et navn på utstyrstypen, serienummer, beskrivelse at typen, merke, bilde-URL og Datablad-URL. Bilde-URL kan en hente ved å kopiere en bilde-URL fra et bilde man finner på google eller et annet sted. Datablad-URL hentes fra URL'en hvor informasjon om databladet ligger. Eksempeldata kan

være:

Legg til ny utstyrstype

Her kan du legge ny utstyrstype til i databasen.

 ▼

Utstyrstype/Utstyrskategori

På utstyrstypedelen av siden vil følgende være vist:

Utstyrstype

Oversikt over utstyrstypene.

Id	Navn	Beskrivelse	Produsent	Kategorinavn	
20	Kjøleskap	Samsung-kjøleskap med innebygd isbitfunksjon	Samsung	Diverse	Slett
21	Test	kdjnad	McDonalds	Test	Slett
22	A	hkjawh	nmdan,	Test	Slett
23	B	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent sagittis tellus ut erat eleifend sodales. Proin dignissim arcu odio, vitae luctus justo rutrum nec. Donec auctor magna vitae erat porttitor, in porttitor sapien vehicula. Nam posuere tristique fermentum. Proin rutrum massa sem, id sagittis massa luctus at. Suspendisse neque mauris, vulputate vitae pharetra ac, suscipit a felis. Nunc viverra quis lectus et consequat. Proin auctor odio a fringilla eleifend. Donec eget lobortis arcu, sit amet consectetur arcu.	Lorem	Test	Slett
24	Test	tuashdsl	jlkdashdkn	Test	Slett
25	Kjøleskap	LG-kjøleskap med innebygd isbitfunksjon	LG	Diverse	Slett

Denne delen gir følgende muligheter:

- Trykke på selve raden til en utstyrstype for å se hvilke utstyr som ligger under denne.
Er også mulig å legge til flere utstyr her.
- Slette en utstyrstype (og alt utstyr som ligger under denne)

Trykker en f.eks på Kjøleskap-raden med Id 20 på bildet over, vil en få opp følgende:

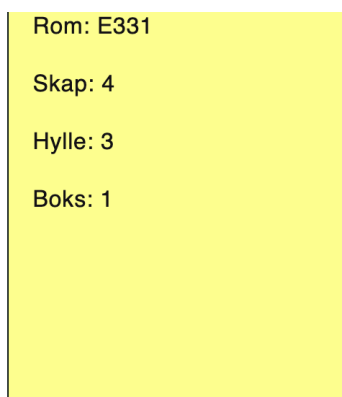
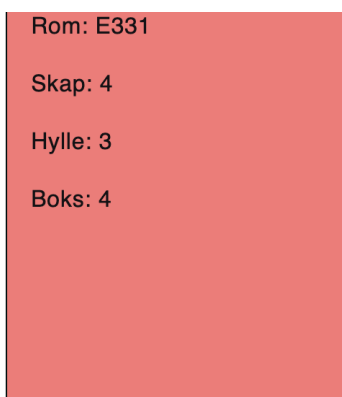
Kjøleskap

Identifikasjonsnavn: Samsung2 Notat: Fungerer Forrige kalibrering: 22/05/2021 11:00 Neste kalibrering: 29/05/2021 11:00 Rom: E331 Skap: 3 Hylle: 2 Boks: 1	Identifikasjonsnavn: Samsung1 Notat: Fungerer Forrige kalibrering: 22/05/2021 11:00 Neste kalibrering: 29/05/2022 11:00 Rom: E331 Skap: 3 Hylle: 2 Boks: 1	Identifikasjonsnavn: Samsung2 Notat: Fungerer Forrige kalibrering: 22/05/2021 11:00 Neste kalibrering: 29/07/2021 11:00 Rom: E331 Skap: 3 Hylle: 2 Boks: 1	Identifikasjonsnavn: Samsung1 Notat: Fungerer Forrige kalibrering: 22/05/2021 11:00 Neste kalibrering: 29/05/2021 11:00 Rom: E331 Skap: 3 Hylle: 2 Boks: 1
Identifikasjonsnavn: Samsung1 Notat: Fungerer Forrige kalibrering: 22/05/2021 11:00	Identifikasjonsnavn: Samsung6 Notat: Ødelagt Forrige kalibrering: 01/01/0001 00:00	Identifikasjonsnavn: Samsung6 Notat: Ødelagt Forrige kalibrering: 01/01/0001 00:00	Identifikasjonsnavn: Samsung6 Notat: Ødelagt Forrige kalibrering: 01/01/0001 00:00

Hvert av disse kortene representerer et individuelt utstyr. Fargekodene er til for å lettere se om et utstyr skal/bør kalibreres ut fra "Neste kalibrering".

- **Rød** fargekode betyr at tiden er utløpt, og at utstyret må kalibreres.
- **Gul** fargekode betyr at det er under 1 måned til utstyret skal kalibreres.
- **Grønn** fargekode betyr at det er over 1 år til utstyret skal kalibreres.
- **Blå/nøytral** fargekode betyr at utstyret skal kalibreres om mer enn 1 måned, og under 1 år.

Siden har også følgende plusstegn for å legge til flere utstyr/kort.



Disse kortene kan også klikkes på, som videresender deg til en side hvor det utstyret som hører til det tilhørende kortet kan endres.

Eksempel:

56. Equipment

En kan her endre informasjonen, og trykke lagre, så vil kortet være endret. Her kan man endre navn, notat, forrige kalibrering, neste kalibrering, romnummer, skapnummer, hyllnummer og boksnummer. Datoene er i String-format.

Trykker en derimot på plusstegnet, vil følgende side komme opp:

. Equipment








Her har en ikke noen tilsendte verdier, da denne siden er for å legge til nytt utstyr/kort. Her kan man endre navn, notat, forrige kalibrering, neste kalibrering, romnummer, skapnummer, hyllenummer, boksnummer og antall. Altså ganske likt som når en endrer et utstyr, bare at en her også har et felt for antall. Skriver en 40 som antall, vil 40 utstyr med den angitte informasjonen bli lagret i databasen.

Brukere

Under "Brukere" får man følgende informasjon:



Brukere

Oversikt over brukere.

Admin	Fornavn	Etternavn	E-post		
	Jkddd	Heiis	ldjksaj@hotmail.com	Slett	Endre adminstatus
	Ingebrigt	Holstad	ingebrigt-holstad@hotmail.com	Slett	Endre adminstatus
	jdlks	ndij	hdjskl@hotmail.com	Slett	Endre adminstatus
	Endre	Håland	ehal@gmail.com	Slett	Endre adminstatus
	Hei	Heiiii	Heisann@hotmail.com	Slett	Endre adminstatus
	knlsalkn	ndjko	asdnlka@hotmail.com	Slett	Endre adminstatus
	hksjahdk	ibjdhbm	hehkhja@hotmail.com	Slett	Endre adminstatus

Denne tabellen viser forskjellig informasjon om hver enkelt bruker, som adminstatus, fornavn, etternavn og e-post.

Adminstatus

-  for brukere som ikke er admin
-  for brukere som er admin.

Adminstatus kan gis og tas til/fra brukere ved å trykke “Endre adminstatus”.

Brukere kan slettes ved å trykke “Slett”. Utlånshistorikken til brukeren vil da også bli slettet.

Kategorier

Kategoridelen har følgende tabell:

Kategorier

Oversikt over kategorier.

Id	Navn	Foreldre-id	Foreldre-navn	
37	Test	0	Dette er en forelder	Slett
44	Test	37	Test	Slett
46	Diverse	0	Dette er en forelder	Slett
47	Test3	0	Dette er en forelder	Slett
48	ns	46	Diverse	Slett
49	mals	44	Test	Slett
50	Underunderkategori	49	mals	Slett

En har kategoriens id og navn, samt foreldre-id og foreldre-navn. Her er Id'en og foreldre-id'en ganske nyttig å ha med, siden foreldre ikke kan slettes før alle barn er slettet. Et eksempel på dette vil være at kategorien med Id 46 (Diverse) ikke kan slettes med mindre kategorien med Id 48 (ns) slettes først. Dette er for å forhindre at en admin sletter en forelder av et barn, som igjen kunne skapt problemer i senere tid. Det er mulig å lage barn av barn her, men dette er ikke ordnet på Reservasjonssiden, og anbefales derfor ikke (før det evt fungerer der også).