



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Library of Things: En fullstack
applikasjon for utlån av utstyr

Library of Things: A fullstack application
for lending items

Jan-Petter Dāvøy

Fredrik Enes

Susanne Marie Mangle Kvame

Dataingeniør

Fakultet for teknologi, miljø- og samfunnsvitskap

Veileder Sven-Olai Høyland

13.mai 2024

Vi 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*, § 12

TITTELSIDE FOR HOVEDPROSJEKT

<i>Rapportens tittel:</i> Library of Things: En fullstack applikasjon for utlån av utstyr Library of Things: A fullstack application for lending items	<i>Dato:</i> 13.05.2024
<i>Forfatter(e):</i> Jan-Petter Dávøy, Fredrik Enes, Susanne Marie Mangle Kvame	<i>Antall sider u/vedlegg:</i> 57 <i>Antall sider vedlegg:</i> 76
<i>Studieretning:</i> Dataingeniør	<i>Antall disketter/CD-er:</i> 0
<i>Kontaktperson ved studieretning:</i> Sven-Olai Høyland	<i>Gradering:</i> Ingen
<i>Merknader:</i> x	

<i>Oppdragsgiver:</i> Fredrik M. Salhus	<i>Oppdragsgivers referanse:</i> Ingen
<i>Oppdragsgivers kontaktperson:</i> Ingen	<i>Telefon:</i> +47 920 34 694

Sammendrag:

Dette bachelorprosjektet har som mål å utvikle en internasjonalisert og globalt skalerbar applikasjon for utlån av utstyr som støtter innlogging og autentisering gjennom Lightning Wallet. Lightning Wallet er en lommebok for å kjøpe og selge kryptovaluta, men enkelte har også støtte for autentisering. Prosjektets formål er å bidra til et bærekraftig samfunn ved å oppfordre og tilrettelegge for utlån, til fordel av egeneie. Prosjektet har benyttet en tverrfaglig-tilnærming, og er utført i samarbeid med oppdragsgiver og designer Fredrik M. Salhus. Ved å utforske forskjellige applikasjonsløsninger og autentiseringsmuligheter har prosjektet undersøkt om en utlånstjeneste basert på Lightning Wallet vil føre til en brukervennlig og tilgjengelig tjeneste.

Stikkord:

React	Firebase	Lightning Wallet
-------	----------	------------------

Abstract

The goal of this bachelor's project is to develop an internationalized scalable application for lending and borrowing items utilizing a login- and authentication-service through a Lightning Wallet. A Lightning Wallet is predominantly used to trade cryptocurrency, however some of them also support authentication. This project wishes to be a contributing factor in creating a sustainable future by encouraging the sharing of resources in favor of individual ownership. This project has had an interdisciplinary approach, where Fredrik M. Salhus has had the role of client and designer, and the group was responsible for developing and implementing the application. By exploring different types of applications and possible authentications the project has examined if a lending service based on Lightning Wallet will result in a user friendly and available service.

FORORD

Denne rapporten gir en gjennomgang av bacheloroppgaven *Library of Things* gitt av Fredrik M. Salhus. *Library of Things* er et utlånsbibliotek, og er inspirert av tidligere oppgaver som *Tingotek* og *Pigify*. Arbeidet i prosjektet er utført av Jan-Petter Dåvøy, Fredrik Enes og Susanne Marie Mangle Kvame.

Vi ønsker å rette en spesiell takk til oppdragsgiver og designer Fredrik M. Salhus for samarbeidet på denne oppgaven. Fredrik M. Salhus har designet logoer, grafikk og designet til applikasjonen. Samtidig rettes en stor takk til vår veileder Sven-Olai Høyland for hans tilgjengelighet og gode samarbeid i denne perioden.

Til slutt ønsker vi å rette vår takknemmelighet til familie og venner for deres støtte, og alle som deltok underveis i brukertesting.

Innholdsfortegnelse

Abstract.....	ii
Ordliste.....	viii
Akronymer.....	x
Figurliste.....	xi
Tabelliste.....	xii
1 INNLEDNING.....	1
1.1 Kontekst.....	1
1.2 Motivasjon.....	1
1.3 Prosjekteier.....	2
1.4 Problembeskrivelse, mål og forskningsspørsmål.....	2
1.5 Oppbygging av rapporten.....	2
2 PROSJEKTBEKRIVELSE.....	3
2.1 Praktisk bakgrunn.....	3
2.1.1 Tidligere arbeid.....	3
2.1.2 Initielle krav.....	3
2.1.3 Initielle løsnings-idé.....	4
2.2 Avgrensninger.....	4
2.3 Ressurser.....	5
2.4 Litteratur om problemstillingen.....	5
2.4.1 Autentisering.....	6
2.4.2 Lightning Wallet.....	6
3 DESIGN AV PROSJEKTET.....	7
3.1 Forslag til løsning.....	7
3.1.1 Alternativ løsning 1.....	7
3.1.2 Alternativ løsning 2.....	7
3.1.3 Alternativ løsning 3.....	7
3.1.4 Alternativ løsning 4.....	8
3.1.5 Diskusjon av alternativene.....	8
3.2 Valgt løsning.....	10
3.3 Valg av verktøy.....	11
3.3.1 Frontend-løsning.....	11
3.3.2 Backend-løsning.....	12
3.3.3 Autentisering.....	12

3.3.4	Database løsning	12
3.3.5	Versjonskontrollsystem og sky basert-tjener	13
3.4	Prosjektmetodikk	13
3.4.1	Utviklingsmetodikk.....	13
3.4.2	Prosjektplan.....	15
3.4.3	Risikovurdering.....	15
3.5	Evalueringsplan.....	16
4	DETALJERT LØSNING.....	17
4.1	Avgrensning av problemstillingen.....	17
4.2	Autentisering.....	17
4.3	System arkitektur	18
4.3.1	Brukstilfellediagram	18
4.3.2	Domenemodell i form av et klassediagram	19
4.3.3	Databasemodell.....	20
4.4	Utvikling	21
4.4.1	Frontend-utvikling	21
4.4.2	Backend-utvikling.....	22
4.4.3	Implementering av autentisering.....	23
4.4.4	Cloud Functions for Firebase.....	23
4.4.5	Database.....	24
4.4.6	Mellomlagring av data	25
4.4.7	Implementering av design.....	26
4.4.8	Internasjonalisering.....	27
4.5	Dokumentasjon	28
4.5.1	Dokumentasjonsstrenger.....	28
4.5.2	README.md.....	29
4.5.3	Versjonskontroll.....	30
4.6	Sikkerhet	31
4.6.1	GDPR.....	31
4.6.2	Sikkerhet i Firebase.....	32
4.6.3	Begrenset tilgang til applikasjon.....	32
4.6.4	Kryptering av QR-koder	32
4.6.5	Sikkerhet i versjonskontrollsystem	33
4.6.6	Secure local storage	34
4.7	Universell utforming.....	34

5	RESULTATER.....	36
5.1	Evalueringsmetode.....	36
5.1.1	Brukertesting.....	36
5.1.2	Systemtesting.....	36
5.1.3	Akseptansetesting.....	37
5.1.4	Dybdeintervju.....	37
5.2	Evalueringsresultat.....	37
5.2.1	Evalueringsresultat av brukertesting.....	37
5.2.2	Evalueringsresultat av system- og akseptansetesting.....	38
5.2.3	Evalueringsresultat av dybdeintervju.....	38
5.3	Prosjektresultat.....	39
5.3.1	Endelig implementasjon av design.....	39
5.3.2	Funksjonelle og ikke-funksjonelle krav.....	40
5.3.3	Dokumentasjon.....	41
6	DISKUSJON.....	42
6.1	Avgrensninger og valg.....	42
6.1.1	Valgte verktøy.....	42
6.2	Applikasjonsresultat.....	43
6.3	Avvik fra initielle løsnings-idé.....	44
6.4	Brukertesting.....	44
6.5	Prosjektmetodikk.....	45
6.6	Forbedringer i endelig løsning.....	46
6.7	Design.....	46
6.8	Samarbeid på tvers av disipliner.....	48
7	KONKLUSJON OG VIDERE ARBEID.....	50
7.1	Forskningsspørsmål.....	50
7.2	Konklusjon.....	50
7.3	Videre arbeid.....	51
7.3.1	Optimalisering.....	51
7.3.2	Internasjonalisering.....	51
7.3.3	Utforming av design for andre enheter.....	52
7.3.4	Tilleggsfunksjoner.....	52
	REFERANSER.....	53
	VEDLEGG.....	58
	Vedlegg 1. Brukertesting.....	58

Vedlegg 1.1. Instruksjoner til brukertest.....	58
Vedlegg 1.2. Spørreundersøkelse med svar.....	59
Vedlegg 2. Demonstrasjonsvideo og skjermbilder av applikasjon.....	67
2.1 Demonstrasjonsvideo.....	67
2.2 Skjermbilder fra applikasjon.....	67
Vedlegg 3. Dybdeintervju.....	72
Vedlegg 4. Kravdokument.....	75
Vedlegg 5. Prosjekthåndbok.....	75
Vedlegg 6. Systemdokumentasjon.....	75
Vedlegg 7. Visjonsdokument.....	76

Ordliste

Tabell 1: Ordliste

Ord	Forklaring / definisjon
Administrator	En bruker med tillatelse til å låne ut, opprette, endre og slette utstyr. Kan også endre og slette brukere.
Akseptansetest	Teste om brukerens behov er møtt.
Autentisering	Autentiseringsløsningen i applikasjonen basert på LNURL-auth og innlogging med Lightning Wallet.
Backend	Delen av applikasjonen som ikke er synlig for en bruker, men som håndterer logikken og behandling av data.
Black box testing	En funksjonell testmetodikk hvor testkriteriene er basert på informasjon fra oppgitte spesifikasjoner. Testeren skal ikke ha tilgang til den interne kildekoden [1].
Blokkjede	«En desentralisert og distribuert digital «regnskapsbok» som gjør det mulig å registrere, spore og synliggjøre alle digitale transaksjoner.» [2]
Brukergrensesnitt (User interface)	Det en bruker ser på applikasjonen.
Brukeropplevelse (User experience)	Opplevelsen brukeren har av applikasjonen.
Brukertest	Teste funksjoner for å evaluere brukeropplevelsen.
BaaS (Backend as a Service)	En tjeneste som lar utviklere fokusere på frontend-utvikling, ved at backend oppgaver håndteres automatisk.
Cloud Firestore	Databasetjenesten til Google Firebase for å lagre, hente og oppdatere data.
Cloud Function	En tjeneste som kan brukes for å kjøre en funksjon på tjenerne til Google.
Frontend	Det sluttbrukeren vil se og interagere med på nettsiden. Utviklet gjennom HTML, CSS og JavaScript.
Fullstack	Kombinasjon av både frontend og backend i en applikasjon.
Funksjonsgap	Beskriver forskjellen mellom samfunnets forventninger, og enkeltmenneskets forutsetninger.
GANTT-diagram	Et diagram som viser prosjektet sin tidsplan og forskjellige arbeidsoppgaver.
Hurtiglager (Cache)	Lagrer data i nettleseren slik at de raskere kan hentes ut.
Implementere	Legge til konsepter og design i en applikasjon.
Internasjonalisering	Gjøre applikasjonen tilgjengelig på flere språk.

Kanban-tavle	Agil metode for å organisere og tildele oppgaver.
Lightning Network	Desentralisert nettverk som gjør det mulig å utføre raske og kostnadseffektive transaksjoner med krypto.
Lightning Wallet	Kryptolommebok som kan brukes for å sende og motta krypto. Enkelte lommebøker har også innebygd støtte for autentisering.
LNURL-auth	Åpen kildekode-løsningen som benyttes for autentisering på applikasjonen.
Local storage	Objekt som lar deg lagre nøkkel/verdi par til nettleseren for hurtigtilgang.
Markeringsspråk (Markup language)	Er et standard tekst-kodingssystem som består av et sett med symboler som blir satt inn i et tekstdokument for å kontrollere struktur, formatering eller forholdet mellom delene [3].
Native mobilapplikasjon	En programvare som er spesifikt satt opp til et operativsystem.
NoSQL	En database som benyttes til å lagre semi-strukturelle data. Kan leses som <i>ikke-SQL</i> .
Peer-to-peer (P2P)	En gruppe datamaskiner som hver for seg fungerer som en node for deling av filer innad i gruppen [4].
Plattform spesifikk applikasjon	En applikasjon som er utviklet for et spesifikt operativt system og/eller en type enhet. Dette kan for eksempel være en iOS eller Android enhet.
Progressiv webapplikasjon	En applikasjon bygget med webapplikasjon prinsipper som gir brukeropplevelse lik en plattform spesifikk applikasjon.
Prop	Informasjon som kan sendes som argument til en JSX-tag. For eksempel kan <i>className</i> , <i>src</i> og <i>alt</i> sendes som <i>egenskaper (props)</i> til en <i>img-tag</i> [5].
React komponent	En gjenbrukbar del av kode som representerer en del av brukergrensesnittet.
Relasjonsdatabase	Database som består av tabeller med (relasjoner) som er forbundet med henvisninger, eller nøkler, mellom seg. [6]
RESTful API	Tjeneste som brukes for å utveksle data over nettet ved å benytte http eller https metoder.
Scrum	En <i>smidig (agil)</i> utviklingsmetode for prosjektgjennomføring.
Service worker	En service worker fungerer som en proxy server mellom webapplikasjonen, nettleser og nettverket.
Superadministrator	En bruker med tillatelse til å opprette, endre og slette lokasjoner. Kan tildele administrator tillatelser.
Systemtest	Teste systemet i sin helhet.

UML (Unified Modeling Language)	Industristandard for modellering av datasystemer.
Versjonskontrollsystem	Felles system for håndtering av forskjellige versjoner av koden.
Åpen kildekode	Programkoden er tilgjengelig og åpen for alle.

Akronymer

Tabell 2: Tabell over akronymer

Akronym	Betydning
API	Application programming interfase
BaaS	Back-end-as-a-Service
CCPA	California Consumer Privacy Act
CRUD	Create, read, update, delete
CSS	Cascading style sheets
GDPR	General Data Protection Regulation
HTML	Hypertext markup language
HTTPS	Hypertext Transfer Protocol Secure
JSX	JavaScript XML (Extensible Markup Language)
LOT	Library of Things
MVP	Minimal viable product
NFC	Near field communication
NoSQL	Not only SQL
PWA	Progressive Web Application
SEO	Søkemotoroptimalisering
SQL	Structured query language
UI	Brukergrensesnitt (User interface)
UML	Unified modeling language
UX	Brukeropplevelse (User experience)
WCAG	Web Content Accessibility Guidelines

Figurliste

Figur 2.1: Prototype av applikasjonen.	4
Figur 3.1: Illustrasjon av flyten til en PWA. Hentet fra: [25].....	8
Figur 3.2: Redigert bilde for å illustrere forskjellene mellom alternativene. Hentet fra: [30]...9	
Figur 3.3: Illustrasjon av responsiv webapplikasjon. Hentet fra: [34].....	11
Figur 3.4: Prosjektets GitHub Kanban-tavle.....	13
Figur 3.5: Illustrasjon av Scrum sprint. Hentet fra: [52].....	14
Figur 4.1: Visuell fremstilling autentiseringsprosessen.....	18
Figur 4.2: Brukstilfelle for en bruker.....	19
Figur 4.3: Brukstilfellediagram for administrator og superadministrator.....	19
Figur 4.4: Klassediagram iterasjon 1.	20
Figur 4.5: Databasemodell tilhørende databasen.	21
Figur 4.6:Eksempel fra database hvor identifikatorer er brukt.....	24
Figur 4.7: Viser hvordan en egenskap (prop) kan brukes i React.....	25
Figur 4.8: Eksempel på hvordan brukeren sin plukklister lagres i local storage.	26
Figur 4.9: Strukturen på mappen som inneholder stilarkene.	26
Figur 4.10: Eksempel på filstrukturen til oversettelsesfilene.....	27
Figur 4.11: Eksempel oversatt JSON-fil.	27
Figur 4.12: Eksempel på hvordan useTranslation brukes i applikasjonen.....	28
Figur 4.13: Eksempel av dokumentasjonsstreng fra prosjektet.	28
Figur 4.14: Eksempel av hvordan dokumentasjonsstrengen ser ut dersom man fokuserer musepekeren på en funksjon.	29
Figur 4.15: README-filen tilhørende GitHub-prosjektet.....	29
Figur 4.16: Illustrasjon over hvordan en git forgrening kan se ut. Hentet fra:[74].....	30
Figur 4.17: Et eksempel av gitmeldinger.....	31
Figur 4.18: Eksempel på en Security Rule.....	32
Figur 4.19: Eksempel på ukryptert identifikator til en reservasjon.	33
Figur 4.20: Eksempel på kryptert identifikator til en reservasjon.	33
Figur 4.21: Illustrasjon over funksjonsgap. Hentet fra:[83]	34
Figur 5.1: Resultatene av akseptansetest.	38
Figur 5.2: Figma design av innloggingsside.	40
Figur 5.3: Implementert innloggingsside design.	40
Figur 6.1: Hjemmesiden til applikasjonen.	43
Figur 6.2: Velg land-siden til applikasjonen.....	43
Figur 6.3: Velg lokasjon-siden til applikasjonen.	43
Figur 6.4: Hvordan man kan legge til en sirkelsag i plukklister.	44
Figur 6.5: Utstyret tilgjengelig på "garden tools" kategorien.	44
Figur 6.6: Kategorivalg siden til applikasjonen.....	44
Figur 6.7: Førsteutkast av design.	47
Figur 6.8: Førsteutkast av brukerprofil.	47
Figur 6.9: Siste iterasjon av brukerprofil.	47
Figur 6.10: Siste utkast av designet.	47
Figur 6.11: Oppdragiver sitt bord med utstyr og plakater.....	49
Figur 6.12: Del av oppdragsgiver sin fremvisning.	49

Tabelliste

Tabell 1: Ordliste	viii
Tabell 2: Tabell over akronymer.....	x
Tabell 3: Oversikt over krav som ikke ble oppfylt	41

1 INNLEDNING

Dette kapitlet tar for seg bakgrunnen og motivasjonen for prosjektet, presenterer oppdragsgiveren og diskuterer gruppens eierskap over resultatet. Videre defineres problemstillingen, målet og forskningsspørsmålet til prosjektet. Avslutningsvis blir strukturen i rapporten presentert.

1.1 Kontekst

Oppdragsgiver Fredrik Salhus har tidligere utviklet idéen bak *tingotek.no* som er en nettside som tilrettelegger for utlån av utstyr [7]. Utfordringene oppdragsgiveren har erfart med denne løsningen er at den er geografisk begrenset til Norge og at innloggingsløsningen er avhengig av at brukeren har tilgang på *Vipps* [8]. Oppdragsgiver ønsker derfor å internasjonalisere legge til rette for skalering av konseptet bak *tingotek.no*.

Fredrik M. Salhus var ansvarlig for utformingen av designet til applikasjonen, og designet var en sentral del av Salhus sin egen bacheloroppgave. Denne tilnærmingen, hvor Salhus var ansvarlig for designet, og gruppen var ansvarlig for å utvikle funksjonalitet til applikasjonen, ga gruppen muligheten til å jobbe tverrfaglig med prosjektet.

1.2 Motivasjon

Etter hvert som alvoret rundt klimautfordringene har blitt klarere, har FN presentert en rekke bærekraftsmål for å motvirke de negative konsekvensene tilknyttet disse utfordringene [9]. Det forbrukes mye mer ressurser enn hva som er bærekraftig for kloden, og fra 2000 til 2017 har det globale materielle fotavtrykket økt med 70 prosent. «For å sikre gode levekår for nåværende og fremtidige generasjoner må også hver enkelt forbruker endre livsstil. Det innebærer å minske ressursbruken, miljøødeleggelsen og klimautslippene som et samfunn og som enkeltperson» [10]. Tanken bak *Library of Things (LOT)* er å bidra til en global delingskultur i et forsøk på å redusere ressursforbruket til hver enkelt person.

Worldbank rapporterte at i 2021 var det 1,4 milliarder mennesker som ikke hadde tilknytning til en bankkonto eller mobil pengeoverføringsmuligheter [11]. Det er ikke urimelig å anta at flere av disse har tilgang på en mobiltelefon selv, eller kan låne av familie eller venner. I et forsøk på å nå ut til disse brukerne, er det ønskelig at LOT skal tilby innlogging med en *Lightning Wallet*, siden den ikke behøver tilknytning til en bankkonto. Mange applikasjoner i dag samler og bruker mye mer persondata enn nødvendig. Derfor ønsker Library of Things også å redusere mengden persondata som innhentes.

1.3 Prosjekteier

Prosjekteier for dette prosjektet er i utgangspunktet oppdragsgiver Fredrik M. Salhus. Gruppens eierskap og mulighet for en åpen kildekode-løsning har blitt diskutert med veileder og oppdragsgiver. Resultatet av disse samtaler er at prosjektet ikke skal være en åpen kildekode-løsning siden oppdragsgiver ikke ønsker å fraskrive seg eierskap til produktet. Spørsmålet om gruppens eierskap i prosjektet skal være åpent for diskusjon gjennom prosjektets forløp.

1.4 Problembeskrivelse, mål og forskningsspørsmål

Prosjektet LOT skal adressere problemet med begrenset tilgang på lån av utstyr, uavhengig av brukernes geografiske lokasjon og økonomiske situasjon. Det er også ønskelig å legge til rette for en innloggingstjeneste som er tilgjengelig for alle. Dagens løsninger er ofte basert på teknologier som enten krever tilknytning til et spesifikt geografisk område, som for eksempel Vipps, eller er styrt av en organisasjon som ikke nødvendigvis representerer folkets interesser.

Målet med prosjektet er å utvikle en internasjonal og skalerbar applikasjon som skal tilrettelegge for delingskultur uavhengig av brukernes geografiske lokasjon og språk. Brukerne av applikasjonen skal kunne velge en lokasjon, legge ønskede gjenstander i en plukkliste og reservere disse gjenstandene. En administrator skal kunne legge til nye gjenstander, legge til og fjerne lokasjoner, og bistå i utlånsprosessen.

Med utgangspunkt i problembeskrivelsen, vil denne oppgaven ta for seg følgende spørsmål:

I hvilken grad vil en applikasjon som tilbyr Lightning Wallet-innlogging bidra til en mer brukervennlig og tilgjengelig utlånstjeneste?

1.5 Oppbygging av rapporten

Rapporten så langt har tatt for seg bakgrunnen og motivasjonen for prosjektet, diskusjon om gruppens eierskap, tydeliggjøring av problemstillingen, målene og forskningsspørsmålet. Kapittel 2 tar for seg tidligere arbeid, prosjektets omfang og avgrensninger. I kapittel 3 blir verktøy og metodikker som er anvendt gjennom prosessen beskrevet og vurdert, inkludert risikovurdering og evaluering av prosjektets fremgang. Deretter, i kapittel 4, presenteres den detaljerte løsningen, med modeller og en innsikt i utviklingsprosessen av applikasjonen. I kapittel 5 blir evalueringsmetodene og resultatene av dem beskrevet, og kapittelet avsluttes med en gjennomgang av prosjektresultatene. I kapittel 6 diskuteres valgene som ble tatt, applikasjonens resultat, eventuelle avvik, forbedringer og samarbeid på tvers av fagfelt. Forskningsspørsmålet besvares, konklusjoner trekkes og forslag til videre arbeid presenteres i kapittel 7. Rapporten avsluttes med en referanseliste etterfulgt av vedlegg.

2 PROSJEKTBESKRIVELSE

Dette kapitlet gir en gjennomgang av tidligere arbeid relatert til prosjektet. Videre defineres omfanget til prosjektet, og nødvendige avgrensninger. Til slutt kartlegges ressursene som ble benyttet under gjennomføringen av prosjektet.

2.1 Praktisk bakgrunn

For å gi en grundigere forståelse for prosjektet vil dette kapitlet beskrive tidligere arbeid relevant for prosjektgjennomføringen, samt kravene som ble satt ved prosjektets start. Til slutt presenteres den initielle løsnings-idéen.

2.1.1 Tidligere arbeid

Oppdragsgiver har tidligere utviklet konseptet bak *Pigify*, som er en applikasjon som tilrettelegger for utlån av gjenstander mellom private aktører [12]. Oppdragsgiver har også utviklet idéen bak *Tingotek* [7].

Flere av funksjonene fra Tingoteket vil danne grunnlaget for utviklingen av Library of Things. I dagens løsning er det mulig for brukerne å se alle tilgjengelige kategorier og deres tilhørende gjenstander. Disse funksjonene skal videreføres til LOT.

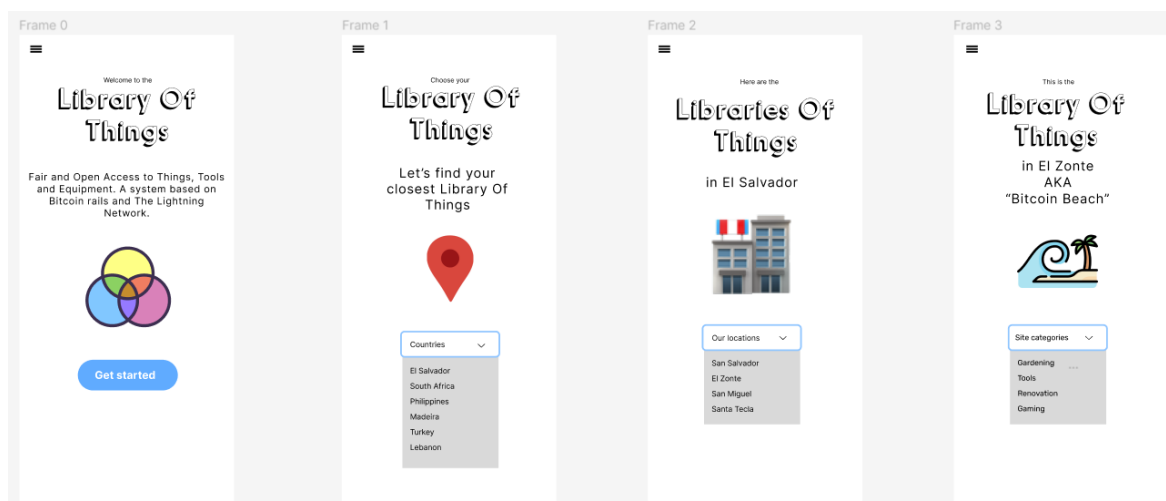
2.1.2 Initielle krav

For å sikre en grundig forståelse av produktet og for å oppfylle oppdragsgiverens krav, var det avgjørende å definere hvilke funksjoner applikasjonen burde inneholde. Oppdragsgiver formulerte følgende krav som dannet grunnlaget for den initielle løsningen:

- Applikasjonen skal være globalt skalerbar.
- Applikasjonen skal være internasjonalisert.
 - Det skal være mulig å velge språk og lokasjon.
- Applikasjonen skal benytte seg av en innloggingstjeneste basert på Lightning Network-teknologi.
 - Løsningen skal være basert på at brukeren kan bruke sin Lightning Wallet til å registrere seg og logge inn på applikasjonen.
- Det skal være mulig å låne gjenstander basert på en valgt lokasjon.

2.1.3 Initielle løsnings-idé

Oppdragsgiver ønsket en applikasjon som tilbyr funksjonalitet som beskrevet i punkt 2.1.2. Applikasjonen skal ha en sømløs innloggingsløsning som lar brukerne skanne en *QR-kode* for å sikre en rask innlogging. Brukerne skal selv kunne velge språk og ønsket lokasjon, og applikasjonen skal tilpasses valgene brukerne har gjort. Et annet ønske fra oppdragsgiver var å begrense mengden persondata som lagres i applikasjonen. Det skal derfor være mulig for brukerne å ha full oversikt over alle persondata som er lagret om brukeren. Brukerne skal også ha muligheten til å slette brukeren sin og alle tilhørende data.



Figur 2.1: Prototype av applikasjonen. Hentet fra: Eget Figma prosjekt.

Oppdragsgiver hadde ansvaret for å designe en prototype som skisserer både designet og brukerflyten for applikasjonen. Den initielle løsningen av applikasjonen fulgte denne prototypen.

2.2 Avgrensninger

Prosjektet strekker seg til utviklingen av en skalerbar *fullstack applikasjon* som også inkluderer implementeringen av en fremtidsrettet og sikker innloggingstjeneste. På grunn av dette omfanget, har gruppen og oppdragsgiveren valgt å fokusere på å utvikle et *Minimum Viable Product (MVP)* [13] for å sikre at prosjektet er gjennomførbar innen prosjektets tidsramme. Ved å bruke en MVP-tilnærming sikres det at produktet inneholder de kritiske funksjonene og kan enkelt settes i produksjon. Dette gjorde det mulig å iterativt avgrense og videreutvikle produktet slik at det kan tas i bruk underveis i prosessen.

For å utvikle en best mulig MVP samarbeidet gruppen tett med oppdragsgiveren gjennom ukentlige arbeidsmøter. Under disse møtene ble de viktigste funksjonene avklart, inkludert følgende:

- Autentisering ved hjelp av en Lightning Wallet.
- Visning av land, lokasjoner, kategorier og gjenstander.

- Mulighet for brukere å låne gjenstander.
- Mulighet for administratorer å legge til, endre, og fjerne gjenstander.
- Mulighet for en superadministrator å legge til, endre og fjerne lokasjoner, samt administrere brukere og administratorer.
- Applikasjonen skal være tilgjengelig på flere språk.

I arbeidet med å sikre brukervennlighet i det endelige produktet, har gruppen sammen med oppdragsgiver valgt å prioritere kjernefunksjonalitet fremfor ytelse i de tidlige stadier av utviklingen. Dette innebærer at mens utviklingsprosessen konsentrerer seg om å etablere nødvendige funksjoner, vil valget av teknologier sikre tilstrekkelig ytelse uten at dette er det hovedfokuset i de første iterasjonene.

2.3 Ressurser

For å planlegge prosjektet har gruppen benyttet seg av verktøy som gjorde redigering og samarbeid mulig i sanntid. Gruppemedlemmene benyttet seg av *Office-pakken* for samskriving på dokumenter og for oppbevaring av filer. Møteinnkalling og annen kommunikasjon med oppdragsgiver og veileder foregikk over e-post. For å få oversikt over applikasjonens arkitektur ble *Visual Paradigm* [14] benyttet til å lage *Unified Modeling Language (UML)-diagrammer* [15]. *Canva* [16] ble benyttet som et verktøy for å lage figurer og illustrere eksempler.

Etter oppdragsgivers ønske ble Google sine tjenester benyttet som samarbeidsverktøy mellom gruppen og oppdragsgiver. Oppdragsgiver utformet designet til applikasjonen i *Figma* [17] slik at gruppen hadde tilgang til designet under hele prosjektet.

Til å organisere og fordele arbeidsoppgaver underveis i utviklingen, valgte gruppen å ta i bruk *Scrum-rammeverket* og det ble benyttet en *Kanban-tavle* [18]. Siden *GitHub* [19] integrerer *Kanban-tavlen* direkte i arbeidsflyten opp mot versjonskontrollsystemet, ble dette en naturlig og effektiv måte for gruppemedlemmene å fordele arbeidsoppgaver og holde kontroll over fremgangen.

Utviklingen av applikasjonen ble gjort på datamaskiner som gruppemedlemmene selv disponerer. Det var heller ikke behov for lån av utstyr fra HVL eller oppdragsgiver. Dersom det oppsto spørsmål, komplikasjoner eller behov for veiledning underveis var det veileder Sven-Olai Høyland gruppen henvendte seg til.

2.4 Litteratur om problemstillingen

I dette kapittelet utforskes relevant litteratur knyttet til problemstillingen om autentisering og bruk av *Lightning Wallet*. Formålet med kapittelet er å gi en grunnleggende forståelse på hvordan autentiseringsløsningen fungerer, og hvorfor den kan være et alternativ til andre autentiseringsløsninger. Kapittelet gir også en introduksjon i hva en *Lightning Wallet* er.

2.4.1 Autentisering

Den vanlige måten i dagens løsninger er å autentisere seg med brukernavn og passord. Dette kan imidlertid føre til sikkerhetsrisikoer, da brukere ofte velger enkle passord eller gjenbraker passordene på flere nettsteder. Brukere på internett gjenbraker det samme passordet 32% av tiden på tvers av 5 til 10 nettsteder og applikasjoner [20]. Hvis en bruker sine kontodetaljer blir delt i en datalekkasje, kan uvedkommende bruke informasjonen til å få tilgang til ulike tjenester. LOT sin autentisering tilrettelegger for at bruker kun trenger å skanne en QR-kode med en Lightning Wallet.

2.4.2 Lightning Wallet

Lightning Wallet er en type *kryptolommebok* for å oppbevare, sende og motta kryptovaluta. Dette foregår ved å generere og behandle et par av private og offentlige nøkler. Den private nøkkelen benyttes til å signere transaksjoner og er et bevis på at kryptovaluta (for eksempel *Bitcoin*) som sendes kommer fra riktig *Lightning Wallet*.

Bitcoin i *Lightning Wallet* overføres over et *Lightning Network*. «Lightning Network (LN) er en «layer 2»-betalingsprotokoll designet for å legges på toppen av en blokkjedebasert kryptovaluta som bitcoin eller litecoin.» [21]. For betaling med en *Lightning Wallet* over *Lightning Network* åpnes en *peer-to-peer(P2P)* [22]-kanal mellom aktørene. Disse kanalene tillater flere transaksjoner uten umiddelbar registrering på *blokkjeden*, og de blir administrert av *Lightning-protokollen*.

Ved å utføre transaksjoner utenfor blokkjeden, kan *Lightning Network* håndtere et større antall transaksjoner samtidig, som fører til en mer skalerbar og effektiv betalingsopplevelse.

Selv om *Lightning Wallet* tilbyr raskere og billigere transaksjoner, er det viktig å være oppmerksom på risikoen knyttet til å benytte åpne kanaler. Dersom en av de involverte aktørene bestemmer seg for å stenge kanalen, kan dette resultere i midlertidig tap av midler. Enkelte *Lightning Wallets* tilbyr mulighet for transaksjon med andre kryptovalutaer og betalingsnettverk. Utvalgte har også støtte for autentisering.

3 DESIGN AV PROSJEKTET

Dette kapitlet inneholder en beskrivelse og vurdering av verktøyene og metodikkene som gruppen har benyttet underveis i prosjektet. Kapitlet inneholder også en risikovurdering og tilhørende tiltak. Avslutningsvis blir evalueringsplanen til prosjektet presentert.

3.1 Forslag til løsning

Dette delkapitlet vil utforske de forskjellige alternative løsningene til hvordan applikasjonen kan struktureres. Kapitlet vil også diskutere fordelene og ulempene ved de forskjellige applikasjonstypene. Gjennom en grundig gjennomgang av de ulike alternativene vil målet være å identifisere den mest hensiktsmessige og effektive løsningen for prosjektets formål.

3.1.1 Alternativ løsning 1

Den første alternative løsningen er en *native mobilapplikasjon*. En native applikasjon er beskrevet som en applikasjon som lastes ned og kjøres på en enhet. Dersom applikasjonen er spesifikt utviklet for en mobil enhet, regnes det som en native mobilapplikasjon [23]. Spesifikt for en native mobilapplikasjon er at det er mulig å få tilgang til enhetens maskinvare og interne operativsystem. Dette gjør det mulig å interagere med enhetens springssystem, mikrofon, kamera og andre funksjoner.

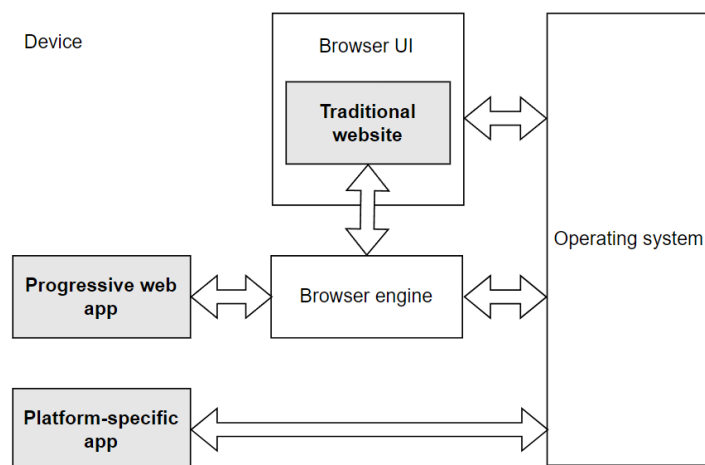
3.1.2 Alternativ løsning 2

Den andre alternative løsningen gruppen har utforsket er en *webapplikasjon*. En webapplikasjon er en applikasjon som er lagret på en ekstern server, og som blir kjørt av en bruker gjennom en nettleser [24]. En webapplikasjon gir brukerne muligheten til å interagere med de innebygde funksjonene til en nettleser. Et eksempel på dette er muligheten for brukerne å bokmerke nettsider for å sikre enkel tilgang i fremtiden.

3.1.3 Alternativ løsning 3

Den tredje løsningen er en *progressiv webapplikasjon (PWA)*. En PWA vil ha samme funksjonaliteter som en webapplikasjon, men vil også tilby en tilsvarende brukeropplevelse som en plattform spesifikk applikasjon [25]. Eksempelvis vil en PWA lastet ned på en mobil enhet tillate brukerne å benytte berøringsskjermen til å manøvrere seg gjennom applikasjonen.

En PWA er bygget på de samme grunnleggende funksjonaliteter som en nettside. Dette betyr at den består av minst en *HTML-basert* side, *CSS* og *JavaScript*. I tillegg inkluderer den en *web app manifest-fil* som gir nettleseren informasjon om at PWA-en skal installeres, samt en *service worker* som tilrettelegger for *nettverksuavhengig (offline)* funksjonalitet.



Figur 3.1: Illustrasjon av flyten til en PWA. Hentet fra: [25]

3.1.4 Alternativ løsning 4

Det siste alternativet er en *responsiv webapplikasjon*. En responsiv webapplikasjon er en nettside som kan aksesseres fra en desktop, mobil eller andre enheter [26]. Denne løsningen vil fortsatt tilby de samme funksjonene som en tradisjonell webapplikasjon, men vil også ha et design, *brukergrensesnitt (UI)* og *brukeropplevelse (UX)* tilpasset andre enheter.

3.1.5 Diskusjon av alternativene





Ytelsen til *native applikasjoner* er som regel bedre enn ytelsen til *web-applikasjoner*. Dette gjelder både utnyttningen av prosessorkraft og båndbredde [27]. Kombinasjonen av god ytelse og tilgang på de innebygde funksjonene til enheten kan føre til at mange vil foretrekke brukeropplevelsen en *native applikasjon* tilbyr. Selv om en native applikasjon vil ha bedre ytelse og være lett tilgjengelig gjennom mobilenhetene, kan det ekskludere eller gi en dårlig opplevelse til brukere som benytter seg av andre enheter.

En annen mulig løsning ville derfor være å utvikle en *webapplikasjon*. Gruppen har tidligere erfaring med webutvikling, og forhåndskunnskapene deres vil gjøre det lettere å håndtere utfordringer som kan oppstå underveis i utviklingsprosessen. Gruppens tidligere erfaringer vil føre til at gruppen kan gi et mer realistisk tidsestimat på når de forskjellige funksjonene til applikasjonen kan forventes å være ferdigstilt.

Videre er utviklingsprosessen til en webapplikasjon som regel billigere enn en native applikasjon. Utviklingsprosessen har også en lavere kompleksitet [23]. Dette fører til at tiden fra utvikling til produktet er klar til å settes ut i markedet er kortere enn ved en *native applikasjonsløsning*.

En stor utfordring ved utviklingen av en webapplikasjon er å tilpasse applikasjonen til forskjellige nettlesere. Nettlesere har ulike måter å tolke HTML-kode, og ikke alle nettlesere støtter de samme bibliotekene og rammeverkene som kan brukes under utviklingsprosessen [28]. Spesielt kan utformingen av dynamiske brukergrensesnitt og brukeropplevelser være utfordrende. Det må også legges til ekstra funksjoner som gjør applikasjonen responsiv på andre enheter. Dette kan motvirke fordelene en tradisjonell webapplikasjon tilbyr med at den er rask, billig og enkel å utvikle.

For å sikre gode brukeropplevelser og samtidig levere en ytelse på linje med en native mobil applikasjon, kan en PWA være et godt alternativ. Ytelsen til en PWA vil ikke være like god som en native applikasjon, men en PWA vil fremdeles tilby en bedre ytelse enn en webapplikasjon [29]. Siden en PWA kan lastes ned på andre enheter, har brukerne muligheten til å velge selv om de vil kjøre applikasjonen i nettleser eller på ønsket enhet. Dette vil gi brukerne mer autonomi i valget av ønsket brukeropplevelse.

Capabilities	Responsive Website	Native App	Progressive Web App
 Mobile-Friendly	✓	✓	✓
 Installation	✗	✓	✓
 SEO-Indexed	✓	✗	✓
 Offline Mode	✗	✓	✓

Figur 3.2: Redigert bilde for å illustrere forskjellene mellom alternativene. Hentet fra: [30].

Det er enklere å vedlikeholde og oppdatere en PWA enn en native applikasjon [31]. Oppdateringer til en PWA kan utføres på tjenersiden, som vil føre til at brukerne kan få tilgang på den nyeste versjonen uten å måtte laste ned oppdateringene først. I en native applikasjon må brukerne laste ned og installere oppdateringene. Dette kan ha en negativ innvirkning på brukeropplevelsen.

En PWA, dersom den er riktig utviklet, vil føre til en bedre *søkemotoroptimalisering (SEO)* enn *webapplikasjons-alternativene*. En bedre *SEO-rate* vil øke synligheten til applikasjonen på de forskjellige søkemotorene, som igjen vil føre til flere besøkere og mulige brukere. Blant de viktigste kriteriene for en bedre *SEO* er at applikasjonen skal være responsiv i design, applikasjonen skal laste innholdet raskt, og nettsiden skal være mobil-vennlig. Dette er funksjoner en PWA tilbyr [32].

Utviklingstiden og kostnadene ved å utvikle en PWA er raskere og billigere enn ved en native mobil applikasjon, men en tradisjonell webapplikasjon vil utkonkurrere alternativene på akkurat disse punktene [31].

Et kompromiss mellom en PWA og en *tradisjonell webapplikasjon*, vil være en *responsiv webapplikasjon*. Denne typen webapplikasjon vil være rask og billig å utvikle, og vil tilby en brukeropplevelse som er tilpasset flest mulige enheter. Applikasjonen vil derimot ikke tilby samme nettverksuavhengig funksjonalitet som en PWA. Applikasjonens ytelse vil også være noe lavere enn alternativene, men siden tidsrommet prosjektet skal utvikles i er såpass kort, blir utviklingstid en avgjørende faktor i valget av løsninger.

3.2 Valgt løsning

Et av hovedkriteriene til prosjektet er at applikasjonen skal være tilgjengelig for så mange som mulig. For å oppnå dette er det naturlig at applikasjonen er tilpasset flest mulige enheter. På bakgrunn av dette ble en tradisjonell webapplikasjonsløsning fjernet som en passende løsning, siden den ikke er tilpasset mobile enheter.

Siden *Library of Things* skal være tilgjengelig for så mange som mulig, ble det enighet i at omfanget ved å lage en native mobilapplikasjon til både *Android* og *iOS* ville forhindre gruppen i å levere et fungerende produkt innen tidsfristen. Gruppens manglende erfaring med mobilutvikling forsterket denne formeningen. Videre ble det avgjort gjennom diskusjoner med oppdragsgiver at en native mobil applikasjonsløsning ikke ville oppfylle kravene for applikasjonen.

Derfor ble valget av den endelige løsningen mellom en PWA og en responsiv webapplikasjon. Oppdragsgiver har vært tydelig med gruppen underveis i utvalgsprosessen at mange personer har for mange applikasjoner på enhetene sine. I gjennomsnitt har en person 80 apper på mobilen sin, og av disse 80 appene blir 62% av dem ikke tatt i bruk månedlig, og i 2019 ble 25% av applikasjoner som var nedlastet bare brukt en gang [33], [34].

Gruppen har derfor valgt å se vekk fra en PWA som en mulig løsning. Selv om en PWA vil tilby et responsiv design, bra ytelse og være nedlastbar på forskjellige enheter, valgte gruppen og oppdragsgiver å tilrettelegge for en mer bærekraftig applikasjon i form av en *responsiv webapplikasjon*.



Figur 3.3: Illustrasjon av responsiv webapplikasjon. Hentet fra: [34]

3.3 Valg av verktøy

Oppdragsgiver, som ikke har bakgrunn som utvikler, ga gruppen stor frihet til å selv velge hvilke teknologier de ønsket å benytte. Det eneste kravet fra oppdragsgiver til teknologier var at autentisering skulle gjøres gjennom en *Lightning Wallet* dersom det var hensiktsmessig og gjennomførbart.

3.3.1 Frontend-løsning

Ved start av prosjektet vurderte gruppen hvilket programmeringsspråk de ønsket å benytte under utviklingen av *frontend-løsningen*. Gruppemedlemmene har brukt *JavaScript* tidligere og ønsket å bygge videre på den kompetansen. Det ble vurdert to tilnæringer innenfor JavaScript: *TypeScript* [35] og *React* [36]. TypeScript utvider JavaScript med statiske typer og vedlikeholdes av Microsoft. React er et bibliotek for utvikling av brukergrensesnitt vedlikeholdt av Meta. Basert på en kombinasjon av popularitet i dagens marked [37], kravene til oppgaven og det eksisterende kunnskapsnivået bestemte gruppen å benytte *React*. Valget ble ytterligere støttet av React sitt kontinuerlige vedlikehold og den oppdaterte støttedokumentasjonen.

3.3.2 Backend-løsning

Gruppen har gjennom studieemner på *HVL* brukt *Java-rammeverket Spring Boot* [38] og Microsoft sitt *.NET-rammeverk* [39]. Til tross for gruppemedlemmene sin tidligere erfaring med disse rammeverkene, var det ønskelig å utforske Google sin tjeneste, *Firebase* [40].

Firebase er en *Backend as a Service (BaaS)-tjeneste* [41] som effektivt håndterer *backend-operasjoner*, for eksempel henting og lagring av data. Dette gjør den attraktiv i tilfeller hvor en enkel backend-løsning er tilstrekkelig. Dersom *Firebase* ikke omfatter alle nødvendige tjenester, tilbyr *Firebase* muligheten til å koble opp alternative backend-løsninger. *Firebase* gjør skalering til ulike geografiske lokasjoner mulig ved å legge til rette for distribuering av applikasjonen på flere tjenerne. *Firebase* sin lave kompleksitet og muligheten for å skalere løsningen globalt førte til at *Firebase* ble valgt som den primære backend-løsningen.

Det var også behov for en alternativ backend-løsning for funksjonalitet som ikke kunne implementeres i *Firebase*. *Java-rammeverket Spring Boot* ble vurdert, men på bakgrunn av støttedokumentasjon for autentiseringsløsning ble *Express.js* [42] undersøkt. *Express.js* er fleksibelt og regnes som det raskeste *Node.js-rammeverket* [43]. Basert på støttedokumentasjonen for autentiseringsløsning og hastigheten til rammeverket, ble *Express.js* valgt som den alternative backend-løsningen.

3.3.3 Autentisering

Oppdragsgiver ønsket autentisering med *Lightning Wallet* i applikasjonen. Det krever at brukeren har tilgang til en *Lightning Wallet* med innebygd funksjonalitet for innlogging. Andre alternative løsninger for autentisering ble undersøkt i tilfelle autentisering med *Lightning Wallet* ikke passet for prosjektet sitt behov. *Firebase*, som er benyttet i backend, tilbyr flere forhåndsdefinerte autentiseringsmetoder som innlogging med epost og passord, Google, Apple, Facebook, Twitter og GitHub. Dette gir fleksibilitet til å kunne endre til en eller flere av *Firebase* sine løsninger raskt og uten vesentlig tidsbruk om det skulle bli nødvendig.

Etter ønske fra oppdragsgiver ble åpenkildekodeløsningen *LNURL-auth* [44] vurdert som en autentiseringsløsning for bruk med *Lightning Wallet* [44]. Det ble også undersøkt autentisering gjennom *Metamask* [45], basert på kryptovalutaen *Etherum*, og *Bitcoin Connect* [46], basert på Bitcoin. Oppdragsgiver ga uttrykk for at han ikke ønsket tjeneste basert på *Etherum*. Etter testing av disse alternativene på nettsider med tilsvarende funksjonalitet ble det sammen med oppdragsgiver bestemt at *LNURL-auth* skulle benyttes.

3.3.4 Database løsning

Gjennom utdanningsløpet har gruppen tilegnet seg kunnskap og kjennskap til relasjonsdatabasen *PostgreSQL* [47]. *Firestore*, som er en tjeneste tilbudt av *Firebase*, er basert på *NoSQL*-teknologi i likhet med *MongoDB* [48] eller *DynamoDB* [49]. Gitt at beslutningen allerede var tatt om å bruke Google *Firebase* i prosjektet, ble det derfor bestemt

å benytte dokumentdatabasen Firestore. Dette valget sikret en mer helhetlig og integrert løsning med Firebase-plattformen.

3.3.5 Versjonskontrollsystem og sky basert-tjener

Versjonskontrollsystemer og sky baserte-tjenere spiller en avgjørende rolle i programvareutviklingsprosjekter. Disse verktøyene tilrettelegger for et effektivt samarbeid på kildekoden under utviklingen. Verktøyene gir også mulighet for å administrere kildekoden og det er mulig å gjenopprette tidligere versjoner ved behov [50].

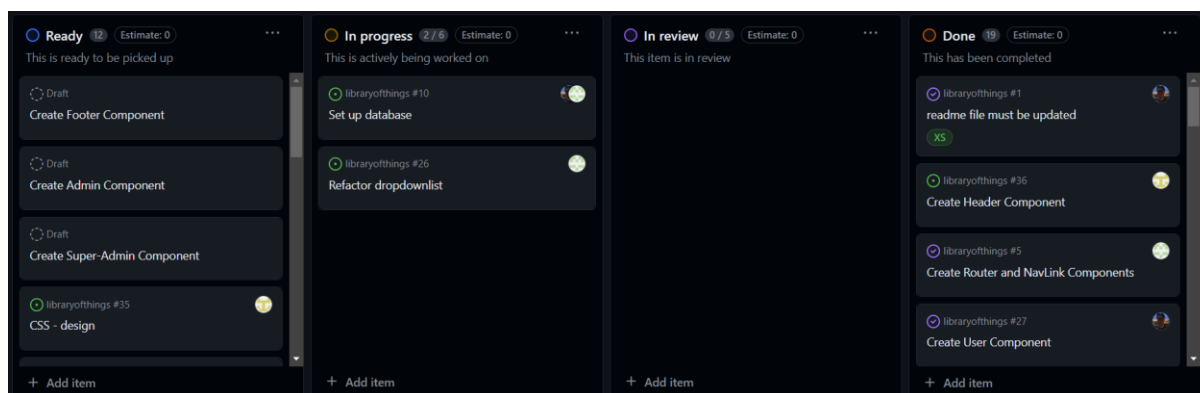
Gruppen har tidligere kjennskap til *Git* som versjonskontrollsystem og GitHub [19] som *skybasert-tjener*, og har tidligere samarbeidet ved hjelp av disse teknologiene. Tidligere erfaring med GitHub effektiviserte oppstarten av prosjektet uten behov for å sette opp egen skybasert-tjener for versjonskontroll.

3.4 Prosjektmetodikk

Dette kapittelet gir en forståelse av prosjektmetodikken som har blitt benyttet i prosjektet. Ved å ta i bruk en *smidig (agil)* tilnærming har gruppen satt seg et mål om å imøtekomme kravene og utfordringene som oppstår underveis. Avslutningsvis blir risikoene i prosjektet gjennomgått.

3.4.1 Utviklingsmetodikk

På grunn av prosjektets behov for fleksibilitet og tilpasningsevne, har gruppen valgt en *smidig tilnærming* til utviklingsprosessen, som kombinerer elementer fra både *Kanban-* og *Scrum-*rammeverket. Den agile tilnærmingen gjør det mulig å tilpasse seg endringer raskt og kontinuerlig, noe som er avgjørende i et prosjekt med skiftende krav og utfordringer [51].



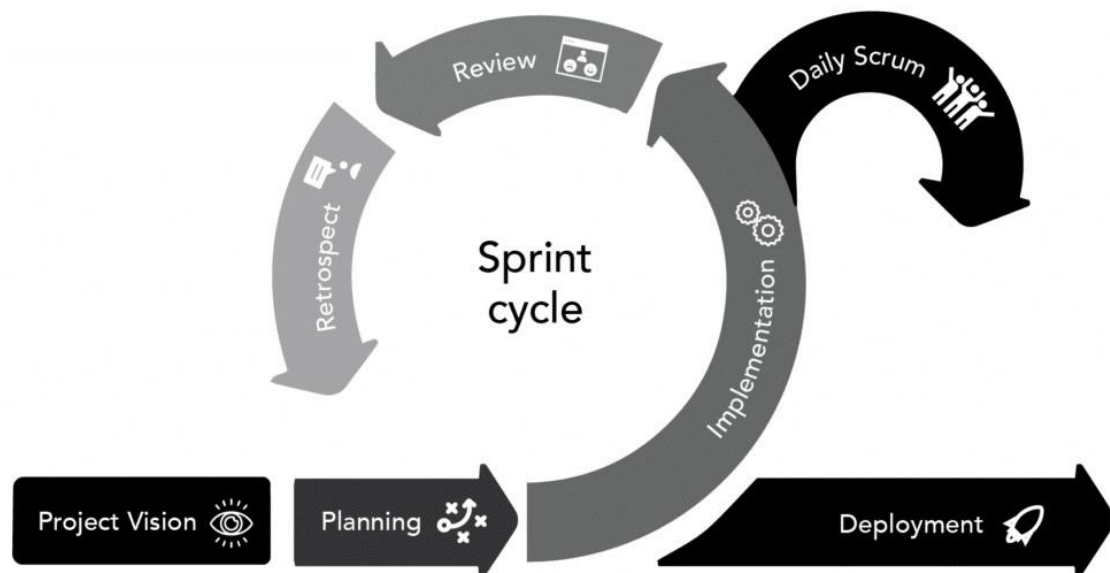
Figur 3.4: Prosjektets GitHub Kanban-tavle. Hentet fra: Eget GitHub prosjekt.

Kanban-tavlen ble benyttet som et verktøy for å gi en visuell presentasjon av oppgavene som skulle gjennomføres, samtidig som den ga en oversikt over arbeidsflyten fra idé til

implementering [51]. Ved å dra nytte av Kanban-metodikken, har gruppen etablert en effektiv måte å prioritere, organisere og spore fremdriften på.

Samtidig har gruppen tatt i bruk daglige *stand-up møter*, inspirert av Scrum, for å opprettholde kontinuerlig kommunikasjon og oppdatere hverandre om fremdriften av prosjektet, utfordringer og neste steg. Disse møtene har bidratt til at hele gruppen var informerte om prosjektets utvikling og handlingene som måtte utføres for å møte kravene.

Ved slutten av hver uke satt gruppen av tid til å planlegge den påfølgende uken. Denne prosessen besto av at gruppen lagde en ukeplan over hvordan dagene skulle disponeres, i tillegg til å sette opp ukentlige milepæler for Scrum-sprintene. En Scrum-sprint vil i utgangspunktet gå over 2-4 uker, men på grunn av tidsomfanget til prosjektet, valgte gruppen å bruke ukentlige sprinter.



Figur 3.5: Illustrasjon av Scrum sprint. Hentet fra: [52].

En viktig del av Scrum-prosessen er *retrospekt-delen*, hvor gruppen og prosjekteier møttes for å gå igjennom resultatet av sprinten. Ved å inkludere oppdragsgiveren i ukentlige møter for planlegging og fremdrift, kunne gruppen sikre at prosjektet ble gjennomført i tråd med oppdragsgiverens krav og forventninger. Disse møtene har gjort det mulig for gruppen å samarbeide tett med oppdragsgiver.

Gjennom hele prosjektet har gruppen valgt å benytte en felles arbeidsplass, noe som har lagt til rette for kontinuerlig kommunikasjon innad i gruppen. Det *smidige (agile) manifestet* sier at «Den mest effektive måten å formidle informasjon inn til og innad i et utviklingsteam, er å snakke ansikt til ansikt.» [52]. Ved å ha alle medlemmene til stede i samme rom, kunne gruppen dra nytte av ulike perspektiver og kompetanser. Dette gjorde det mulig å raskt identifisere de beste løsningene og effektivt implementere dem.

3.4.2 Prosjektplan

Gjennom prosjektet benyttet gruppen et *GANTT-diagram* (se vedlegg 5) for å få en helhetlig og strukturert oversikt over prosjektets omfang. Et GANTT-diagram er et stolpediagram som viser både planlagt og fullført arbeid over en periode [51]. Dette diagrammet ble nøye strukturert for å organisere arbeidsflyten gjennom prosjektets løp og ble delt inn i tre kategorier: dokumentasjon, utvikling og møter.

Dokumentasjonsdelen omfattet alle obligatoriske innleveringer, støttedokumentasjon og rapportskriving. Utviklingsdelen dekket undersøkelse av teknologier, selve utviklingen, design av applikasjonen og brukertesting. Gruppen mente det var viktig å ha en strukturert oversikt over møtene for å sikre god kommunikasjon med oppdragsgiver og veileder. Derfor ble det laget en egen kategori for møter.

3.4.3 Risikovurdering

En avgjørende faktor for prosjektets gjennomførbarhet er gruppens evne til å forutse og håndtere risikoer. For å få en oversikt over de mest kritiske risikoene knyttet til prosjektet har gruppen utført en omfattende risikoanalyse (se vedlegg 5). De mest kritiske faktorene som ble identifisert i analysen er:

- Tredjepartsløsningen som blir tatt i bruk feiler.
- Autentiseringsløsningen blir utdatert.
- Problemer relatert med avgrensning av prosjektet.
- Teknologivalget kan føre til utfordringer.

Prosjektet benytter seg av tredjepartsløsninger for innlogging og autentisering av brukere. En utfordring ved å bruke tredjepartsløsninger i en applikasjon er at eieren av løsningen kan velge å avslutte prosjektet eller endre tilgangen til løsningen. Gruppen har derfor brukt tid under valget av teknologier til å finne løsninger som tilrettelegger for en smidig overgang til en alternativ innlogging og autentiseringstjeneste. Valget av Firebase som en *BaaS* tilrettelegger for denne overgangen. Dersom den valgte autentiseringsløsningen, LNURL-auth, feiler eller ikke lengre blir vedlikeholdt av utviklerne, er det raskt og enkelt å endre til en annen løsning.

Det å utvikle en global skalerbar *fullstack-applikasjon* som dekker alle ønskene til oppdragsgiver på tre måneder med utvikling er en utfordrende oppgave. For å sikre at det blir mulig å levere et fungerende produkt har gruppen, gjennom regelmessige møter med oppdragsgiver og veileder, fokusert på å begrense omfanget til prosjektet. I oppstartsfasen til prosjektet brukte gruppen mye tid og ressurser på å planlegge systemarkitekturen, sette opp ukentlige milepæler og lage en utviklingsplan for å tilrettelegge for prosjektets suksess.

3.5 Evalueringsplan

For å evaluere prosjektets fremgang, og for å forsikre seg at produktet gruppen utvikler alltid er i en fungerende tilstand, benyttet gruppen seg av utvalgte metodikker fra *Scrum-rammeverket*. Spesielt ble *stand-up møter* benyttet for å holde gruppe-medlemmene informerte om statusen til prosjektet. Dette bidro til at gruppen holdt hverandre ansvarlig, som sikret en akseptabel fremdrift. Det ble utført ukentlige møter med oppdragsgiver, og etter behov med veileder, hvor statusen av prosjektet ble diskutert. Tilbakemeldingene fra oppdragsgiver ble tatt med videre i fremtidige iterasjoner av produktet.

Denne evalueringsprosessen, hvor gruppen viser frem produktet, og oppdragsgiver får muligheten til å teste applikasjonen, var en viktig del av evalueringsprosessen for produktet. Ved å utføre *systemtesting* på denne formen, vil gruppen kunne evaluere om kravene på bakgrunn av brukstilfellene ligger an til å bli oppfylt, eller om det må gjøres endringer underveis.

Gruppen har også planlagt å utføre *brukertester* på eksterne aktører for å få tilbakemelding på brukeropplevelsen og brukergrensesnittet til produktet. Den siste delen av evalueringsplanen vil foregå på formen av en *akseptansetest* sammen med oppdragsgiver. Akseptansekriteriene vil være basert på bakgrunn av de initiale kravene i delkapittel 2.1.2, i tillegg til uvalgte krav fra kravdokumentasjonen (se vedlegg 4).

4 DETALJERT LØSNING

Dette kapitlet gir en omfattende gjennomgang av løsningen, fra avgrensninger av problemstillingen til hvordan applikasjonen er strukturert. Autentiseringsløsningen beskrives nærmere, og kapitlet gir et detaljert innblikk i utviklingsprosessen. Sikkerheten og dokumentasjonsprosessen til applikasjonen blir også gjennomgått i kapitlet.

4.1 Avgrensning av problemstillingen

Det ble tidlig avklart at prosjektets gjennomførbarhet var avhengig av at gruppen, og oppdragsgiver, prioriterte de viktigste funksjonene fremfor design og ytelse. Scrum sin iterative tilnærming til prosjektgjennomføring gjorde det mulig for gruppen å fokusere på å implementere de kritiske funksjonene til applikasjonen. Samtidig bidro tilnærmingen til at gruppen kunne tilrettelegge for de ikke-kritiske funksjonene dersom milepælene for uken ble oppfylt.

Da oppdragsgiver presenterte nye utkast av designet til applikasjonen, diskuterte gruppen og oppdragsgiver hvilke elementer fra designet som best representerte tankene bak LOT. Designelementene det var enighet om ble iverksatt i de følgende iterasjonene av utviklingsprosessen.

4.2 Autentisering

Et krav fra oppdragsgiver var å benytte autentisering med *Lightning Wallet* i applikasjonen. For å oppfylle dette kravet ble den åpne kildekode-løsningen *LNURL-auth* [22] valgt. LNURL-auth er en åpen autentiseringsprotokoll som tillater brukere å autentisere seg på tjenester og applikasjoner ved å benytte en Lightning Wallet. Denne protokollen fjerner behovet for mer tradisjonelle autentiseringsmetoder som innlogging med brukernavn og passord. I stedet for tilbyr den en brukervennlig *desentralisert* autentiseringsprosess som tar hensyn til brukerens private data ved å benytte en offentlig nøkkel fra Lightning Wallet som identifikator. Autentiseringsprosessen illustreres i figur 4.1



Figur 4.1: Visuell fremstilling autentiseringsprosessen. Hentet fra: Eget prosjekt i Canva.

4.3 System arkitektur

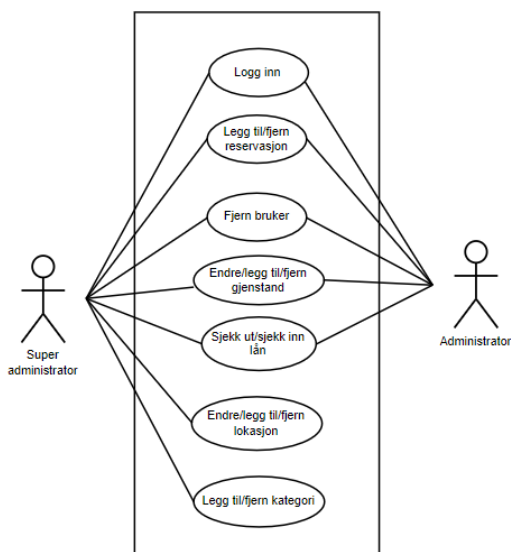
Systemet applikasjonen tilhører består av en *Firestore* database, en *BaaS* levert av *Firebase* og en frontend-løsning i *React*. For å sikre at de forskjellige komponentene som utgjør systemet interagerer på en sikker og effektiv måte, har gruppen utformet diagrammer ved hjelp av UML [53].

4.3.1 Brukstilfellediagram

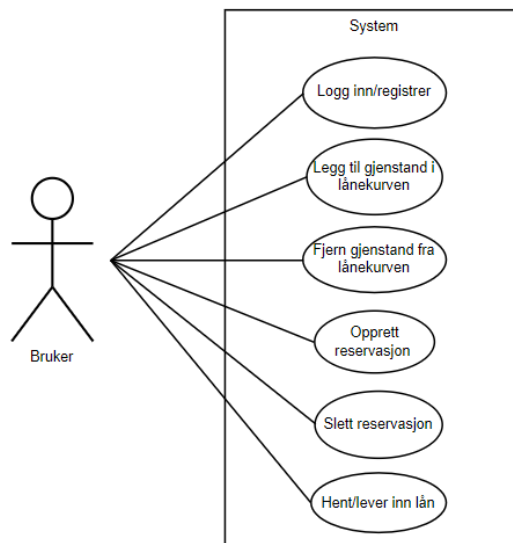
Et brukstilfelle beskriver en sekvens av handlinger utført av et system eller en aktør for å oppnå et resultat. En aktør er en rolle som står utenfor systemet. Ved å lage et brukstilfelle er det mulig å få en strukturert representasjon av interaksjonen mellom aktørene og systemet [54]. Gruppen har derfor valgt å representere hvordan aktørene vil samhandle med systemet gjennom brukstilfeller.

De sentrale aktørene i systemet inkluderer brukere av applikasjonen, administratorer og superadministratorer. Brukerne av applikasjonen vil benytte seg av forskjellige funksjoner, for eksempel utlån av utstyr og innlogging. En administrator vil ha mulighet til å håndtere og organisere utlånet av utstyret. Superadministratorer skal ha full tilgang til applikasjonen,

blant annet muligheten til å administrere brukerkontoer og lokasjoner. Se figur 4.2 og figur 4.3 som viser brukstilfellediagram for superadministrator og administrator, og for en bruker.



Figur 4.3: Brukstilfellediagram for administrator og superadministrator. Hentet fra: Eget Visual Paradigm prosjekt.

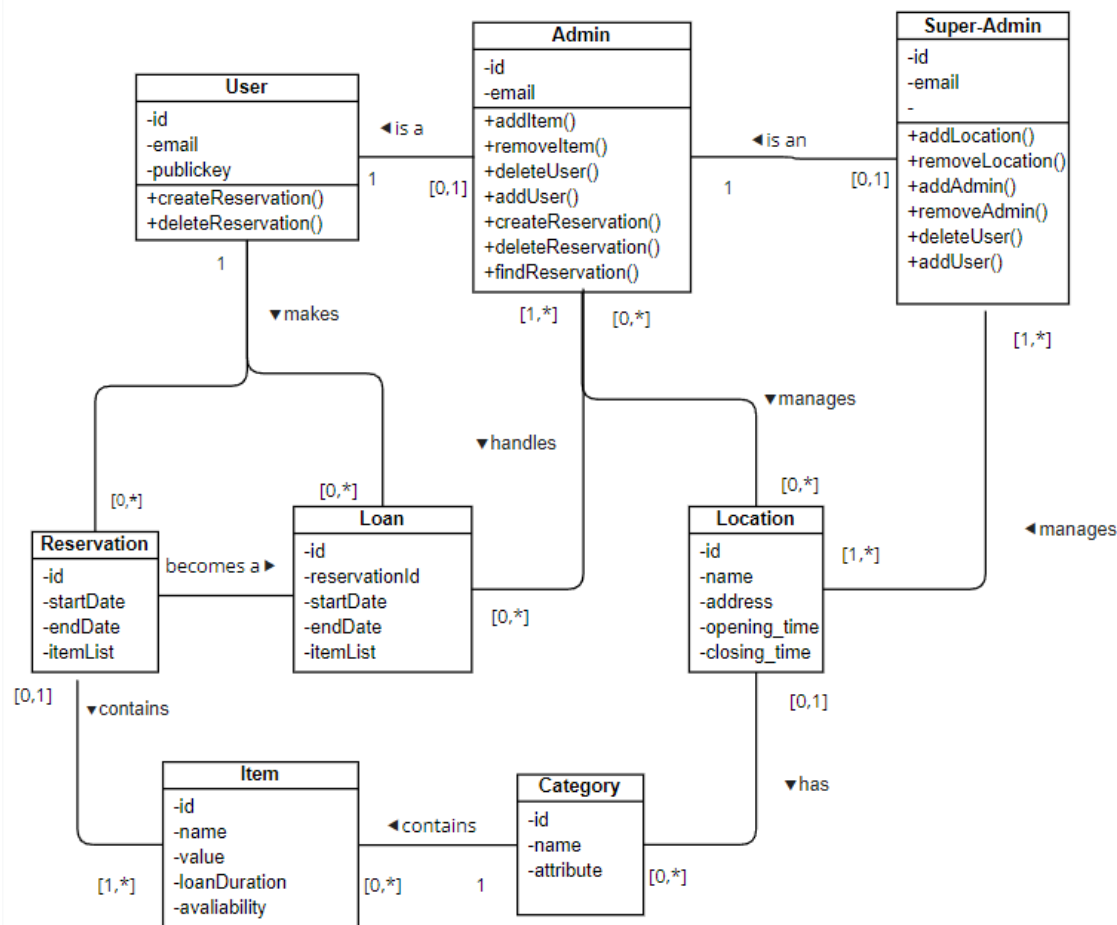


Figur 4.2: Brukstilfelle for en bruker. Hentet fra: Eget Visual Paradigm prosjekt.

Hvert brukstilfelle er detaljert beskrevet i vedlegg 5. Ved å representere hvordan disse aktørene samhandler med applikasjonen gjennom brukstilfeller, fikk gruppen en tydelig strukturert modell av systemets funksjonalitet. Dette ga gruppen muligheten til å identifisere og forstå ulike scenarier, og sørget for at systemet møtte behovene til de forskjellige aktørene.

4.3.2 Domenemodell i form av et klassediagram

Et klassediagram blir brukt for å modellere de ulike klassene til applikasjonen, hvordan de er utformet og assosiasjonene mellom klassene [55]. Klassene i diagrammet til systemet vil representere brukerne, administratorene og lokasjonene til de forskjellige bibliotekene. Diagrammet vil også beskrive de forskjellige tingene tilgjengelig på bibliotekene og hvilke kategorier de tilhører. Se figur 4.4 for klassediagram.



Figur 4.4: Klassediagram iterasjon 1. Hentet fra: Eget prosjekt i Visual Paradigm.

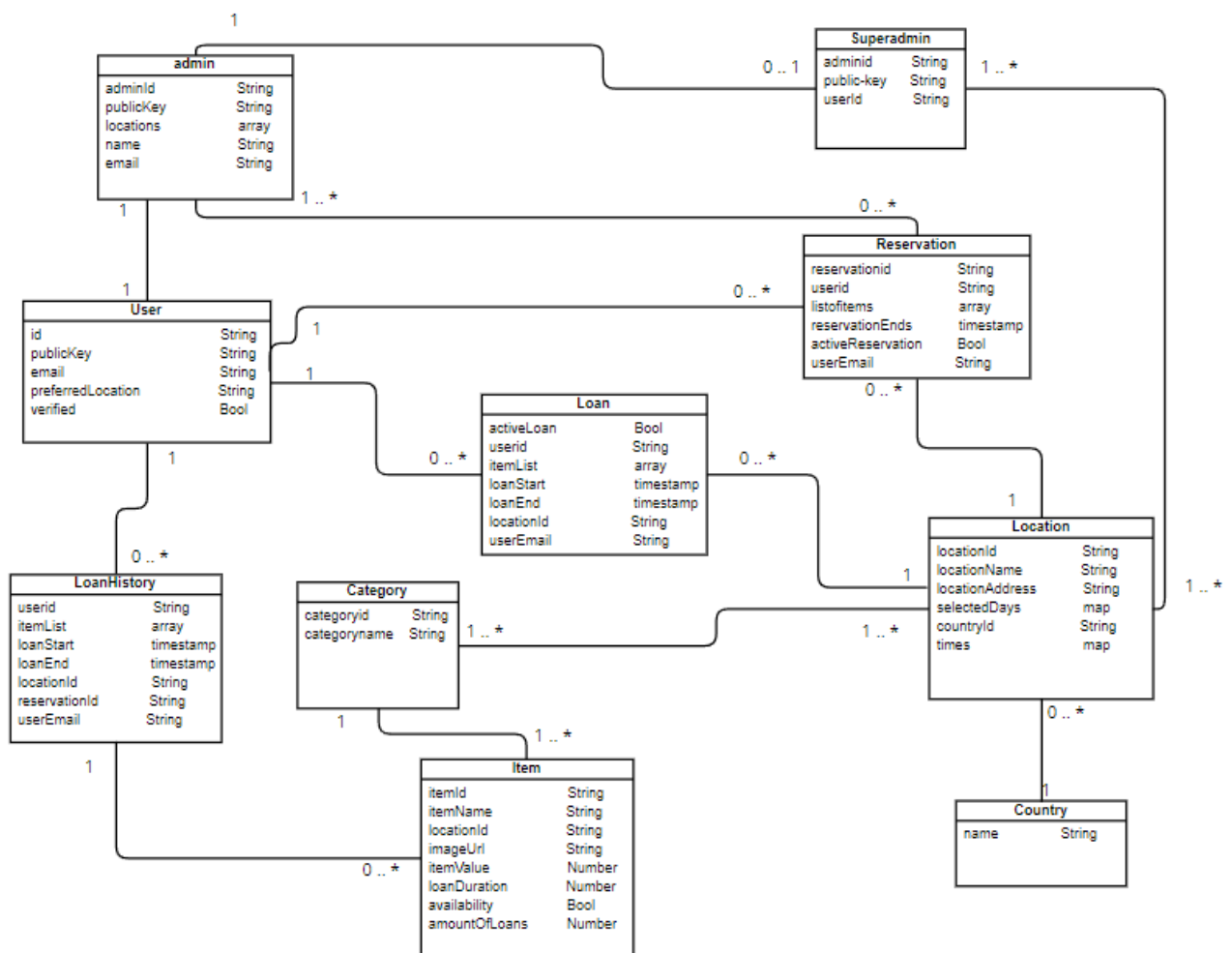
Klassediagrammet dannet grunnlaget for utviklingen av applikasjonen. Diagrammet gjorde det mulig for gruppen å strukturere kodebasen på en best mulig måte, uten å måtte planlegge applikasjonens struktur parallelt med utviklingsprosessen. Dersom den modellerte løsningen viste seg å ikke være gjennomførbar, eller det oppsto en bedre løsning, valgte gruppen å avvike fra modellen.

Det ble klart underveis i prosjektet, at applikasjonen ikke vil følge den objekt-orienterte strukturen modellen er utformet på. Klassediagrammet dannet istedenfor grunnlaget til *React* komponentene som ble tatt i bruk i *frontend-løsningen*.

4.3.3 Databasemodell

En viktig del av applikasjonens arkitektur er oppbyggingen av databasen. For å illustrere sammenhengen i databasen, har gruppen laget en databasemodell. Figur 4.5. viser entitetene til systemet, samt multiplisiteten mellom dem. Se systemdokumentasjon (vedlegg 6) for en mer detaljert forklaring.

Modellen har blitt oppdatert og utvidet i takt med endringer underveis. Databasemodellen viser hvordan databasen vil se ut ved prosjektets slutt, og har vært en viktig ressurs for å veilede utviklingen av prosjektet.



Figur 4.5: Databasemodell tilhørende databasen. Hentet fra: Eget Visual Paradigm prosjekt

4.4 Utvikling

Dette delkapittelet vil gi en grundig gjennomgang over hvordan gruppen har utviklet applikasjonen, hvilke teknologier som ble valgt og hvorfor akkurat disse ble valgt.

4.4.1 Frontend-utvikling

For å generere visningen av applikasjonen valgte gruppen å benytte seg av biblioteket React [36]. React er et JavaScript-bibliotek som består av *komponenter*. Hver komponent er en del av brukergrensesnittet og har sin egen logikk og utseende [56]. Komponentene er JavaScript funksjoner som returnerer en *markering* (*markup*). Et *markeringsspråk* består av regler for å strukturere tekstdokument gjennom tagger, og en markering representerer disse taggene. En av hovedforskjellene mellom *JSX*, som er React sin markering, og *HTML*, er at *JSX* er strengere enn *HTML*. Dette innebærer blant annet at hver *JSX*-tag må pakkes inn i en felles

forelder, som for eksempel en *div*-tag eller et *React-fragment*. Gruppen valgte derfor å benytte seg av React sin JSX markering til fordel for HTML sin markering.

En av de største fordelene ved å bruke React er at den komponentvise tilnærmingen til utviklingen tilrettelegger for å lettere følge designmønstrene til metodikker som *SOLID* [57]. Spesielt vil det være lettere å oppfylle kravene for *single responsibility* fra *SOLID*, hvor hver klasse, eller komponent, skal ha ett enkelt veldefinert ansvar [57].

React tilbyr også mange innebygde funksjoner for å optimalisere generering og fremvisning av nettsidene (*useRef*, *useMemo*, *useCallback*). Disse funksjonene tilbyr i varierende grad måter å optimalisere frontend-løsningen til applikasjonen. Det vil for eksempel være mulig å forhindre unødvendig oppdatering av variabler ved bruk av *useRef*. Funksjonen *useCallback* vil også forhindre unødvendig utførelse av funksjoner dersom ytterligere kall til funksjonen ikke vil gi nye resultater. Dette vil være svært verdifullt for funksjoner som krever mye ytelse eller båndbredde.

4.4.2 Backend-utvikling

Firebase er *BaaS-tjenesten* som ble benyttet til interaksjon med *Firestore* databasen i prosjektet. Interaksjon med *backend-løsningen* foregår gjennom frontend-løsningen, og *Firebase* har allerede flere innebygde funksjoner som kan brukes til å utføre *Create*, *Read*, *Update*, *Delete* (*CRUD*)-operasjoner.

Denne type backend-teknologi egner seg godt for mindre applikasjoner, og bidrar til rask utvikling med lav kostnad. Det gjør behovet for en egendefinert backend-løsning mindre og et MVP vil ferdigstilles tidligere. *Firebase* benytter seg av en *JSON-struktur* på spørringene. *JSON*-strukturen gjør at filstørrelsene på spørringene er små og egner seg godt for *LOT* som skal benyttes i sanntid [58].

Der *Firebase* ikke tilbyr nødvendig innebygd funksjonalitet, er det opprettet spørringer i en egen løsning som er blitt gjort tilgjengelig via et *RESTful API* (*Application Programming Interface*). Denne løsningen implementeres ved hjelp av *Express.js*. Løsningen omfatter funksjoner som e-postverifisering ved første gangs pålogging, kvittering på e-post ved ny reservasjon, kryptering av QR-koder og autentisering.

Konsekvensen av å bruke *Firebase* er at hvis ikke *Firebase* har ønsket innebygd funksjonalitet, må løsningen settes opp på egenhånd. Applikasjonen kan ikke dra nytte av den innebygde autentiseringstjenesten til *Firebase*, og kan derfor ikke bruke e-post tjeneren til *Firebase*. Gruppen implementere en egen løsning for e-post-utsending ved hjelp av backend-rammeverket *Nodemailer* [59]. Denne tjenesten blir gjort tilgjengelig gjennom et *RESTful API* som er koblet opp mot frontend-løsningen for å sende e-post.

Nodemailer ble benyttet for å verifisere brukeren sin e-post. Siden den direkte tilknytningen brukerne har til applikasjonen er gjennom epost, er det viktig at de bruker en epost som kan verifiseres. Dette er spesielt viktig siden tjenesten kan gi tilgang til verdifulle gjenstander,

som for eksempel en sirkelsag. Dersom brukerne ikke verifiserer e-posten sin, får de ikke tilgang til å låne gjenstander eller sider som krever autentisering.

Nodemailer fungerer ved at man oppretter et *transporter objekt* som tilbyr en innebygd funksjon for å sende e-post. Denne funksjonen aksepterer forskjellige argumenter, blant disse et data objekt. Gruppen brukte dataobjektet til å strukturere innholdet til e-posten. Siden formatet til objektet kunne være på HTML-format, var det relativt enkelt å omgjøre innholdet i dokumentene fra databasen til korrekt HTML-format. Dette var svært nyttig når gruppen sendte ut bekreftelse e-post på reservasjoner og lån, siden det var enkelt å tilpasse innholdet i dataobjektet.

4.4.3 Implementering av autentisering

Autentiseringsløsningen er satt opp med forskjellige stier som kalles på ved hjelp av et RESTful API. Stien */api* vil sjekke om en bruker allerede er autentisert, */api/login* vil gi tilbake en innloggingsside, */api/user* vil respondere med den offentlige nøkkelen til brukeren sin Lightning Wallet. For å logge ut av tjenesten benyttes */api/logout* og brukers sesjon ødelegges.

Bruker blir vist en *QR-kode* for innlogging fra stien */api/login*. Verdien på QR-koden er en *bech32* [60] -kryptert lenke. Bruker skanner QR-koden med en Lightning Wallet, og gir fra seg Lightning Wallet sin offentlige nøkkel. Den offentlige nøkkelen vil være unik for hvert domene den brukes på. Ved å kreve innlogging gjennom Lightning Wallet, blir det vanskeligere å omgå autentiseringen siden løsningen sikrer at brukeren ikke kan autentiseres uten tilgang til tilknyttet Lightning Wallet.

Første gang en bruker autentiserer seg vil brukeren måtte oppgi en e-post. Deretter vil brukeren motta en e-post som må bekreftes ved å trykke på en lenke. Prosessen sikrer at e-postadressen eksisterer og at brukeren har tilgang til denne. Videre må brukeren oppgi en foretrukket lokasjon. Den foretrukne lokasjonen kan endres og hindrer ikke brukeren fra å låne gjenstander på andre lokasjoner.

Hvis en bruker allerede er autentisert, så vil brukeren bli sendt videre inn i tjenesten og kunne starte å interagere med hele applikasjonen. Om en bruker ikke allerede er autentisert, vil brukeren ikke ha mulighet til å se gjenstandene.

En bruker kan logge seg ut av sesjonen ved å benytte en logg ut knapp som kaller på *api/logout* som fjerner brukeren sin sesjon. Brukeren er nødt til å autentisere seg på nytt for å få tilgang til tjenesten. Sesjonen vil ikke fjernes før brukeren logger ut eller skrur av enheten.

4.4.4 Cloud Functions for Firebase

Applikasjonen krever tilgang til funksjoner som kjøres regelmessig, for eksempel hvert 10. minutt. For å gjøre dette mulig ble *Cloud Functions* benyttet [61]. Cloud Functions er et rammeverk som kjører backend-kode som kan kobles opp mot ved hjelp av *https-spørringer*

eller *Cloud Scheduler Jobs* [62]. *Cloud Scheduler Jobs* er en metode for å automatisere jobber eller funksjoner. For å sjekke om en reservasjon har utløpt, kjøres denne funksjonen hvert 10. minutt.

4.4.5 Database

For oppbevaring av data ble *Google Firestore* benyttet. Denne integreres mot *Firebase* ved å benytte de eksisterende funksjonene fra *Firebase*. *Firestore* er en *NoSQL* skydatabase. Fordelen med *NoSQL* teknologi er at de har fleksible datamodeller, raske spørringer og er enkel å jobbe med for utviklerne [63]. Data lagres som *JSON*-dokumenter med nøkkel-verdi par.

Firestore er en *NoSQL* database som er en hierarkisk strukturert og består av *dokument*, *kolleksjon* og *subkolleksjon* [64]. Et dokument kan inneholde flere datatyper som for eksempel strenger, tall eller lister. Flere dokumenter kan samles i en kolleksjon. En subkolleksjon i *Firestore* referer til en samling av dokumenter som er underordnet et dokument i en kolleksjon.

Applikasjonen sin database består av kolleksjoner og dokumenter. For å referere til dokumenter i andre kolleksjoner så benytter databasen auto-genererte identifikatorer for å opprette relasjoner mellom dokumenter i ulike kolleksjoner.

Databasen er blant annet satt opp med en bruker- og en lånekolleksjon. Dokumentene i lånekolleksjonen inneholder identifikatoren til en bruker. Dette kobler relaterte data sammen på tvers av forskjellige kolleksjoner uten å ha relasjonell struktur i databasen. Figur 4.6 viser data til et lån i applikasjonen, og kobling mellom bruker- og lånekolleksjon.

```
activeLoan: true
itemList
  0 "u050Jkw43Y7YqP7Gwg2t"
loanEnd: April 6, 2024 at 1:50:15 PM UTC+2
loanStart: March 22, 2024 at 12:50:15 PM UTC+1
locationId: "uwLoMaOnSbvBafC3wQQn"
reservationId: "6AV7YLyeciVilG5m8zjd"
userId: "9EjM0Qq8JbzuOqmAqfJa"
```

Figur 4.6: Eksempel fra database hvor identifikatorer er brukt. Hentet fra: Eget *Firebase* prosjekt.

Dette er en vanlig tilnærming i NoSQL databaser, spesielt når man håndterer data som ikke har streng-struktur eller hvor relasjonene er dynamiske eller komplekse. Ved å benytte de auto-genererte identifikatorene opprettholdes fleksibilitet og skalerbarhet i systemet samtidig som det gjør det mulig for effektiv datahenting og manipulering.

For å øke ytelsen og redusere kompleksiteten i spørringer som involverer flere databasetabeller, har hyppig etterspurte data blitt denormalisert. Dette innebærer å gjøre ofte aksessert data tilgjengelig i flere kolleksjoner. Dette fører til raskere og mer effektive spørringer.

4.4.6 Mellomlagring av data

Mellomlagring av data spiller en viktig rolle i å redusere forespørsler til databasen. Ved å mellomlagre data som er hentet fra databasen i *hurtiglageret* (*cache*), blir ressursforbruket til applikasjonen redusert. Det finnes mange forskjellige måter å sende og mellomlagre data i React. En av de vanligste måtene er å sende data som en *egenskap* (*prop*) mellom foreldre og barnekomponenter. Figur 4.7 viser et eksempel på hvordan en egenskap kan brukes.

```
function Knapp(props) {
  const knappStil = {
    color: props.color,
    fontSize: props.fontSize + 'px'
  };
  return (<button style={knappStil}>{props.text}</button>);
}
export default function App() {
  return (<div><Button text="Klikk meg!" color="blue" fontSize={16} /></div>);
}
```

Figur 4.7: Viser hvordan en egenskap (*prop*) kan brukes i React: Hentet fra: Egen kildekode

En utfordring ved å sende data mellom komponentene i React gjennom *egenskaper* (*props*) er et fenomen kjent som *prop drilling*. Prop drilling er prosessen av å sende data eller *tilstander* (*state*) gjennom flere lag i *komponent-hierarkiet*. Dette kan føre til problemer dersom hierarkiet er veldig dypt eller kompleks [65]. Et av problemene som kan oppstå gjennom prop drilling er at det blir vanskeligere å vedlikeholde koden, for eksempel kan det bli vanskeligere å gjøre endringer i kodebasen uten å påvirke andre komponenter i hierarkiet [65].

For å løse problemene med prop drilling tilbyr React en innebygd måte å sende data og tilstander gjennom komponent-treet uten å måtte sende disse manuelt gjennom egenskaper [66]. Denne funksjonen er kjent som en *kontekst* (*context*), og er designet for å dele data som er ønskelig å bruke på tvers av hele applikasjonen. Gruppen benyttet seg av konteksten blant annet for å håndtere den autentiserte brukeren når den er logget inn, *plukklisten*, og til å lagre resultatene fra andre kostbare funksjoner.

Ved å bruke kontekst, var det mulig å sjekke konteksten direkte for relevante data, til fordel for å måtte utføre en spørring til databasen. Dette var spesielt nyttig for sider som krevde at

en bruker var autentisert, og for å sjekke om brukeren skal ha tilgang til siden den ville aksessere.

For å gjøre brukerdata som er lagret i konteksten vedvarende dersom brukeren lukker nettleseren eller mister nettilkobling, har gruppen brukt *secure local storage* [67]. Secure local storage er en kryptert skrivebeskyttet attributt som tilhører window grensesnittet. Den tillater brukerne å aksessere et *Storage-objekt* som tilhører dokumentets origin [68]. Data i Storage-objekter blir lagret på tvers av nettleserens sesjoner. Data i bruker sin plukkliste er ikke kryptert og lagres i *local storage* [68].

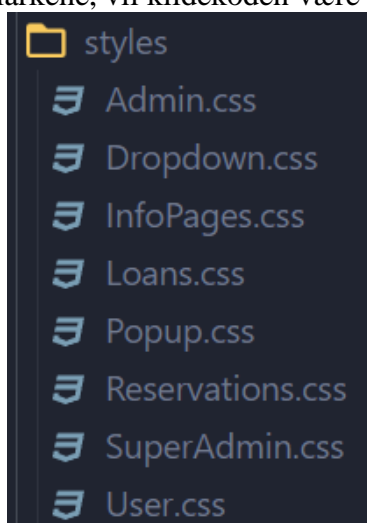
```
localStorage.setItem("Picklist", list);  
const pickList = localStorage.getItem("PickList");  
localStorage.removeItem(pickList);
```

Figur 4.8: Eksempel på hvordan brukeren sin plukkliste lagres i local storage. Hentet fra: Egen kildekode

4.4.7 Implementering av design

Utforming av applikasjonen ble basert på oppdragsgivers design. For å implementere dette designet valgte gruppen å bruke *CSS (Cascading Style Sheets)*, fremfor andre styling-bibliotek som Tailwind [69].

En fordel med CSS er muligheten til å implementere et generelt design på tvers av alle sidene, og deretter legge til mer spesifikke designelementer etter behov. For å opprettholde en oversiktlig og gjenbrukbar kode, sørget gruppen for å dele opp *stilarkene (style sheets)* der det var nødvendig. Dette gjorde det enklere og mer oversiktlig for gruppen å implementere designet ved å isolere endringene til komponentene som benyttet stilarkene. Ved å dele opp stilarkene, vil kildekode være mer oversiktlig og lettere å vedlikeholde for fremtidige



utviklere.

Figur 4.9: Strukturen på mappen som inneholder stilarkene. Hentet fra: Eget prosjekt

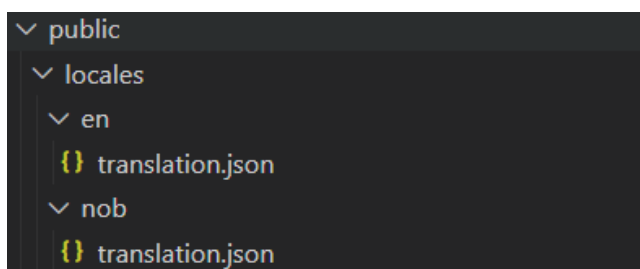
Et av kravene for applikasjonen var å oppnå bred tilgjengelighet for brukere. Oppdragsgiverens design var rettet mot mobilplattformer, men gruppen tok ansvar for å tilpasse og implementere designet for enheter med større skjermstørrelser. Dette sørget for at applikasjonen kunne nå ut til flest mulig brukere, uavhengig av enhetens skjermstørrelse.

Designet ble implementert med tanke på å bruke semantisk riktig HTML, noe som gir bedre struktur og organisering av nettsidens innhold. Dette inkluderte riktig bruk av HTML-elementer (for eksempel `<header>`, `<nav>`, `<footer>`) for å markere viktige deler av applikasjonen. Ved å implementere riktig struktur i designet, bidrar dette til å forbedre nettstedets rangering i resultatene til søkemotorer. Gruppen forsøkte å sikre at applikasjonen ikke bare opprettholdt brukervennlighet, men også at den var tilrettelagte for synlighet i søkeresultatene ved å strekke seg mot kravene til universell utforming.

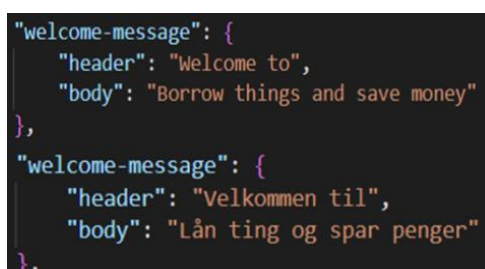
4.4.8 Internasjonalisering

En undersøkelse utført av CSA-Research i 2020 sier at 76% av netthandlere foretrekker å kjøpe produkt hvor produktinformasjonen er på deres morsmål. I tillegg så vil 40% av brukerne aldri kjøpe noe fra en nettside som er på et fremmedspråk [70]. Disse funnene indikerer at dersom man ikke internasjonalsiserer nettsiden sin, kan man risikere å miste 40% av markedet. Selv om LOT ikke selger produkt, men låner ut produktene, er det ikke urimelig at disse tallene også vil representere sentimentet til brukerne av LOT.

Tanken bak LOT er at det skal være mulig å opprette biblioteker uavhengig av hvor man er i verden, og det vil derfor være viktig at applikasjonen kan tilpasses den geografiske lokasjonen biblioteket er opprettet på. For å implementere internasjonalsisering i applikasjonen er rammeverket: *react-i18next* [71] brukt.



Figur 4.10: Eksempel på filstrukturen til oversettelsesfilene. Hentet fra: Eget prosjekt



Figur 4.11: Eksempel oversatt JSON-fil. Hentet fra: Egen kildekode.

React-i18next rammeverket tilbyr flere funksjoner for å internasjonalsisere og oversette innholdet i applikasjonen. Det er mulig å bruke de innebygde funksjonene til å hente ut brukernes foretrukne språk, bestemme et standard språk for applikasjonen, og å opprette egne JSON filer med tilhørende oversettelser for språkene applikasjonen skal støtte.


```
import { useTranslation } from 'react-i18next';
const { t } = useTranslation();
<div className="home-header">
  <h1>{t('welcome-message.header')}</h1>
  <img src={LOTLogo} className="home-logo"
    alt="image of the library of things logo" />
  <p className="home-welcome-message">{t('welcome-message.body')}</p>
</div>
```

Figur 4.12: Eksempel på hvordan *useTranslation* brukes i applikasjonen. Hentet fra: Egen kildekode.

Ved å importere *useTranslation* funksjonen og lagre den i en variabel som illustrert i figur 4.12 kan man enkelt hente ut den riktige oversettelsen fra den ønskede JSON-filen. Dette gjør det raskt og enkelt å tilpasse nettsiden til det valgte språket.

Prosjektfilen til applikasjonen inneholder en *README.md*-fil (se vedlegg 6) som beskriver prosessen av å legge til nye språk i detalj. Denne guiden ble utformet for å gjøre det mulig for oppdragsgiver, og andre mindre datakyndige, skal kunne legge til nye språk uten behov for en dedikert utvikler. README-filen inneholder også forslag på hvordan *internasjonaliseringen* i applikasjonen kan forbedres.

4.5 Dokumentasjon

En viktig del for å sikre at applikasjonen er vedlikeholdbar, og for å tilrettelegge for overtakelsen av prosjekt og kodebase, er å tydelig dokumentere prosjektet underveis. Gruppen har benyttet seg av dokumentasjonsstrenger, en README-fil og versjonskontrollsystem for å sikre at kodebasen er tilstrekkelig dokumentert og forklart.

4.5.1 Dokumentasjonsstrenger

Dokumentasjonsstrenger er tekststrenger som beskriver hva et segment av kode eller en funksjon gjør. Ved å bruke dokumentasjonsstrenger i form av JSDoc [72] i kodebasen, vil det være lettere for andre utviklere å forstå hva den tilhørende koden faktisk gjør.

Dokumentasjonsstrenger gjør det også mulig for andre å vedlikeholde og gjenbruke koden dersom utvikleren som skrev koden ikke er tilgjengelig.

```
/**
 * This method checks if a user by a given publickey is an administrator or not.
 * @param {*} publicKey the publickey of the user
 * @returns true if user is admin, false otherwise
 */
```

Figur 4.13: Eksempel av dokumentasjonsstreng fra prosjektet. Hentet fra: Egen kildekode

```
function isUserAdminByPublickey(publicKey: any): Promise<boolean | undefined>

This method checks if a user by a given publickey is an administrator or not.

@param publickey — the publickey of the user
```

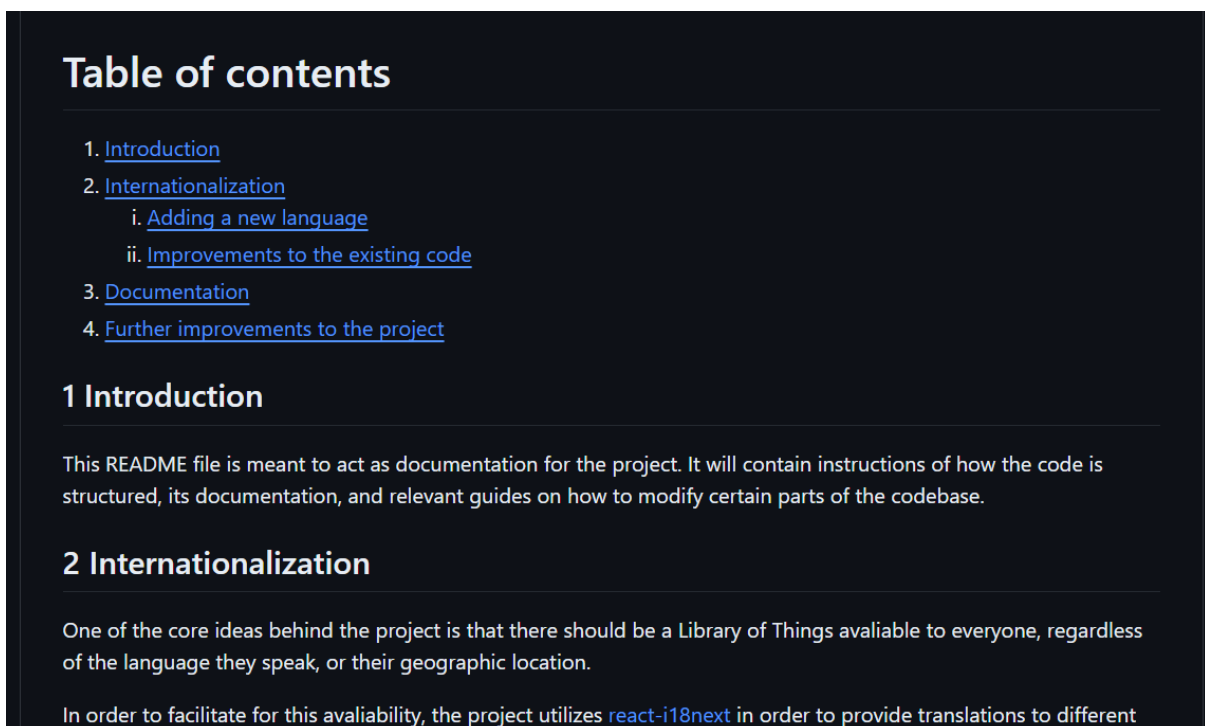
Figur 4.14: Eksempel av hvordan dokumentasjonsstrengen ser ut dersom man fokuserer musepekeren på en funksjon. Hentet fra: Egen kildekode

4.5.2 README.md

Gruppen har valgt å legge til en *README.md*-fil i applikasjonen. En README fil blir ofte brukt for å formidle viktig informasjon om prosjektet. Dette kan innebære en overordnet forklaring over prosjektet, hvilket formål prosjektet skal oppfylle, hvorfor prosjektet er viktig og hvordan det kan brukes [73].

Gruppen brukte README filen til å gi en grundig introduksjon over prosjektet og dets formål. README filen inkluderer også relevant informasjon på hvordan applikasjonen kan vedlikeholdes og videreutvikles. Filen inkluderer også en guide på hvordan det kan legges til nye språk og oversettelser i applikasjonen.

Figur 4.15 viser README-filen som tilhører prosjektet. For flere bilder av README-filen, se vedlegg 6 *Systemdokumentasjon*.



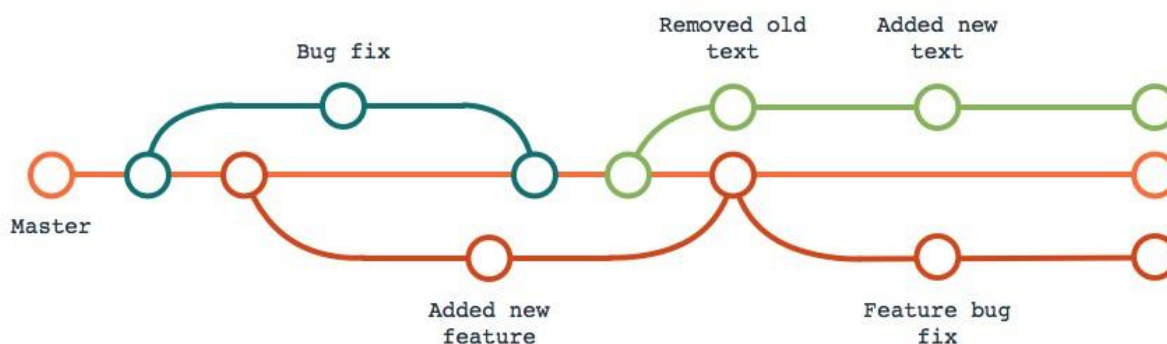
The image shows a dark-themed README file. At the top, there is a 'Table of contents' section with a list of links: '1. Introduction', '2. Internationalization' (with sub-items 'i. Adding a new language' and 'ii. Improvements to the existing code'), '3. Documentation', and '4. Further improvements to the project'. Below this, section '1 Introduction' begins with the text: 'This README file is meant to act as documentation for the project. It will contain instructions of how the code is structured, its documentation, and relevant guides on how to modify certain parts of the codebase.' Section '2 Internationalization' follows, starting with: 'One of the core ideas behind the project is that there should be a Library of Things available to everyone, regardless of the language they speak, or their geographic location.' The bottom of the image shows the start of a paragraph: 'In order to facilitate for this availability, the project utilizes [react-18next](#) in order to provide translations to different

Figur 4.15: README-filen tilhørende GitHub-prosjektet. Hentet fra: Eget GitHub-prosjekt.

4.5.3 Versjonskontroll

Gjennom prosjektet forløp var det avgjørende å finne et verktøy som ville legge til rette for effektiv og iterativt samarbeid med applikasjonen. Som nevnt i kapittel 3.3.5 ble det bestemt å benytte Git som versjonskontrollsystem.

En *gren (branch)* i Git er en uavhengig arbeidsgren som gir muligheten til å jobbe med kodeendringer og nye funksjoner uten å påvirke hovedkoden som tilhører *hovedgrenen (main branch)*. Ved å bruke forskjellige grener, kunne hvert medlem i gruppen jobbe med sin egen gren uten å påvirke koden i hovedgrenen. Dette var spesielt nyttig når medlemmene måtte utforske forskjellige rammeverk og funksjonalitet. Ved å bruke gren-funksjonaliteten til Git, ble det gjort tydelig hvem som jobbet med hva. Figur 4.16 viser hvordan en Git forgrening kan fungere.

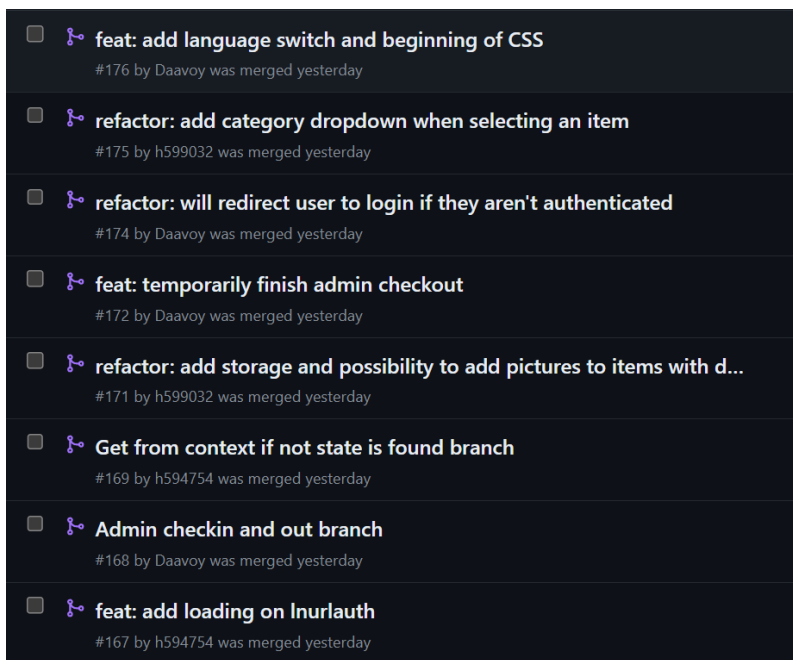


Figur 4.16: Illustrasjon over hvordan en git forgrening kan se ut. Hentet fra:[74]

Gruppen valgte også å knytte Git opp med GitHub sin kanban tavle, slik at en gren kunne opprettes direkte fra den aktuelle oppgaven på tavlen. Når en oppgave ble fullført, kunne endringene fra grenen *flettes (merge)* inn i hovedgrenen, gjennom en *pull-request*. En pull-request er en prosess hvor endringene i den lokale grenen blir sammenlignet med innholdet i hovedgrenen, og et av medlemmene må godta de innkommende endringene før de flettes inn i hovedgrenen.

En annen fordel med Git er muligheten til å holde den lokale koden oppdatert med de siste endringene i hovedgrenen. Ved å bruke *pull-funksjonaliteten* i Git, kan den lokale versjonen enkelt oppdateres med alle de nye endringene som ble gjort av de andre medlemmene. Dette bidro til at gruppemedlemmene alltid hadde tilgang til den nyeste versjonen av koden.

Git sin kommentarfunksjonalitet har også blitt benyttet, som ga muligheten til å legge ved utfyllende forklaringer til endringene som ble gjort. Dette har vært nyttig for å holde oversikt over hva som er gjort og gjorde det enklere for andre medlemmer i gruppen å forstå endringene. Alle kommentarene blir lagret i en historikk, dette sikrer at fremtidige utviklere kan få tilgang til kommentarene. Kommentarene kan brukes som dokumentasjon ved feilsøking, samt brukes under videreutvikling av applikasjonen. Figur 4.17 viser et eksempel på Git *commit*-meldinger i GitHub-prosjektet.



Figur 4.17: Et eksempel av gitmeldinger. Hentet fra: Eget GitHub prosjekt.

4.6 Sikkerhet

Dette avsnittet tar for seg tiltakene som er gjort for å sikre data og integriteten i applikasjonen og LOT. Fra å overholde *GDPR* til å benytte seg av kryptering av sensitiv informasjon. Hver del av sikkerheten er gjennomtenkt og implementert for å gi en trygg og sikker opplevelse for brukerne.

4.6.1 GDPR

General Data Protection Regulation (GDPR) er et sett regler som gjelder for alle EU/EØS-land [75]. Loven gjelder ved automatisk og ikke-automatisk behandling av personopplysninger. Brukerne til LOT gir kun fra seg ett tilfelle av persondata som skal behandles, og dette er e-postadressen. For å sikre at GDPR og andre personvernlover overholdes, benyttes *HTTPS-protokollen* for å kryptere dataoverføringer mellom brukernes enheter og LOT sine tjenere.

Land utenfor EU/EØS kan ha egne lokale regler for hvordan data skal behandles. Ved skalering av applikasjonen utenfor EU/EØS må disse følges. Et eksempel på dette er *California Consumer Privacy Act (CCPA)* [76] som er en lov for å beskytte data og personvern for innbyggere i California. Hvis GDPR følges har applikasjonen et bra fundament til å følge alle lokale regler, men ved skalering må de lokale reglene undersøkes.

For å kunne låne en gjenstand krever det at brukeren har en e-postadresse knyttet opp til applikasjonen. Beskyttelse av e-postadressen vil gjøres gjennom regulering av hvem som har tilgang til å aksessere databasen og at databasen har tilstrekkelige sikkerhetstiltak på plass.

Registrerte brukere har rett til innsikt i informasjon tjenesten har lagret om brukeren. Den registrerte brukeren kan kreve at LOT utleverer alle opplysninger som er lagret. Oppretting av konto krever samtykke fra bruker. Bruker har også mulighet til å slette kontoen sin.

4.6.2 Sikkerhet i Firebase

For å sikre at alle lagrede og oppbevarte data hos Firebase tilhørende applikasjonen er trygg, har sikkerhetstiltakene til Firebase blitt grundig gjennomgått.

Alle Firebase tjenester har fullført den internasjonale *ISO 27001 standarden* [77]. *ISO 27001 standarden* hjelper organisasjoner å etablere, implementere, opprettholde og forbedre informasjonssikkerheten. Standarden danner et rammeverk for hvordan en bedrift kan håndtere konfidensiell informasjon. Videre sikrer den at bedriften sine IT-systemer beskyttes mot uautorisert tilgang, endringer, ødeleggelser eller andre sikkerhetstrusler.

Firebase Security Rules er en tjeneste mellom applikasjonen sine data og brukere [78].

Security Rules kan benyttes til å lage regler for hvem som har tillatelse til å lese eller skrive data til spesifikke deler av databasen. Reglene skrives i et spesialformat som ligner JSON og er illustrert i figur 4.18.

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true;
    }
  }
}
```

Figur 4.18: Eksempel på en Security Rule. Hentet fra: Eget Firebase prosjekt.

4.6.3 Begrenset tilgang til applikasjon

Applikasjonen forhindrer brukere fra å se gjenstander på en lokasjon før de er autentisert. Denne sikkerhetsmekanismen er implementert for å forhindre at tingene kan fremvises til besøkere av applikasjonen som er ute etter å gjøre skade. Besøkeren av applikasjonen vil kunne fremvise land og lokasjoner hvor det finnes et bibliotek, og hvilke kategorier som er på lokasjonen, men ikke hvilke gjenstander som tilhører lokasjonen.

4.6.4 Kryptering av QR-koder

For å sikre en god brukeropplevelse i applikasjonen brukes det QR-koder som enkel tilgang til funksjoner som innlogging, og visning av låne- og reservasjonsdetaljer. Når en bruker blir presentert en QR-kode er det ønskelig at den ikke viser stien til hvor QR-koden leder. For å

unngå dette har verdiene til QR-kodene knyttet til reservasjon og lån blitt kryptert med *AES-256-CBC* [79] kryptering. QR-koden som presenteres for en bruker ved autentisering er kryptert med en *bech-32* [60] kryptering. Figur 4.19, viser et eksempel på en *ukryptert* QR-kode. Figur 4.20, viser et eksempel på en *AES-256-CBC kryptert* QR-kode.



Figur 4.19: Eksempel på ukryptert identifikator til en reservasjon. Hentet fra: egen applikasjon.



Figur 4.20: Eksempel på kryptert identifikator til en reservasjon. Hentet fra: egen applikasjon.

4.6.5 Sikkerhet i versjonskontrollsystem

Sammen med oppdragsgiver ble det bestemt at kildekode skulle være privat for å unngå at kildekode brukes til å finne sikkerhetshull. Prosjektet er likevel satt opp slik at dersom oppdragsgiver ønsker å åpne kildekode for offentligheten, skal det ikke være historikk i applikasjonen som utfordrer sikkerheten. Ved hjelp av en *.gitignore*-fil i *GitHub*-oppbevaringen (*repository*), har konfigurasjonsfilene til tjenesten blitt skjult slik at ikke legges til i *GitHub*-oppbevaringen. Dersom konfigurasjonsfilene lastet opp i oppbevaringen burde oppbevaringen fjernes og opprettes på nytt.

For å sikre at URL-er, passord og annen informasjon ikke ble eksponert direkte i kildekode, ble disse bli lagret i en *.env-fil*. Filen ble brukt til å lagre konfigurasjonsinnstillinger, miljøvariabler og annen sensitiv informasjon sikkert [80]. Dette bidrar til å øke sikkerheten ved å holde sensitiv applikasjonsdata adskilt fra kildekode. Dersom URL-er eller andre innstillinger oppdateres, trenger man kun å gjøre endringer i *.env-filen*, og ikke på flere steder i kildekode. Dette fører til bedre organisering av konfigurasjonsdata og gjør det enklere å vedlikeholde applikasjonen.

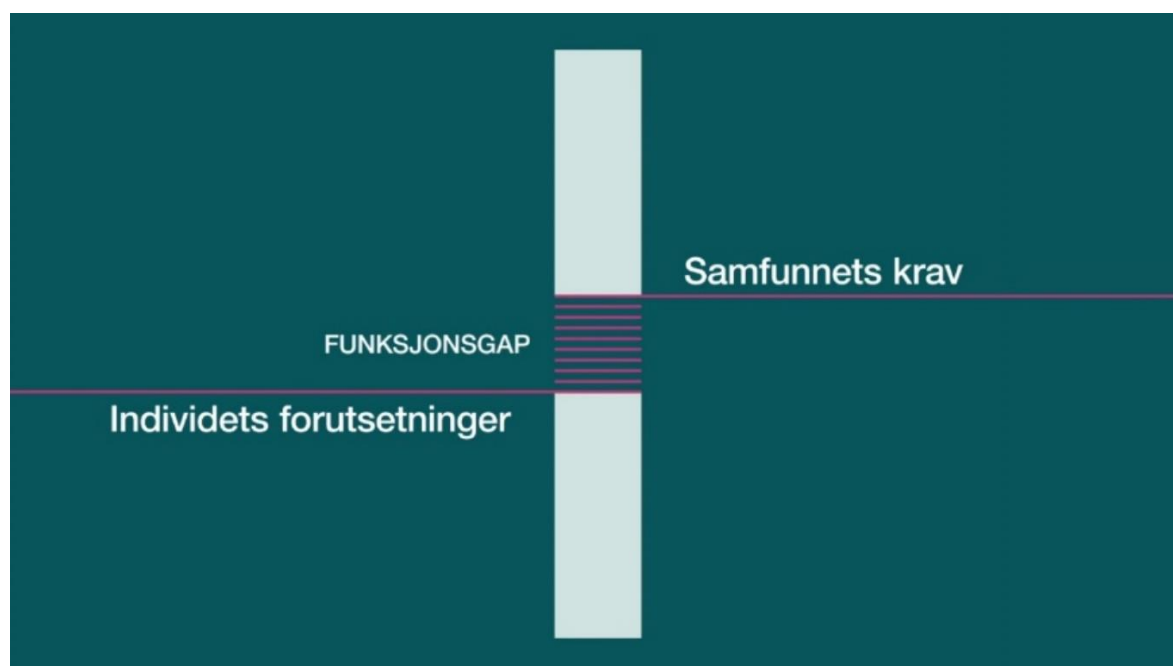
4.6.6 Secure local storage

Applikasjonen benytter *secure local storage* [67] for å mellomlagre brukerdata i applikasjonen (se delkapittel 4.4.6 for mer informasjon). For å sikre at brukerne ikke kan lese eller modifisere mellomlagret brukerdata benyttes kryptering. *Secure local storage* genererer en unik sikker nøkkel for hver nettleser og bruker nøkkelen til å kryptere data før de lagres i *Secure local storage*. Dekryptering blir gjort ved å benytte den genererte nøkkelen til nettleseren.

4.7 Universell utforming

Rettferdig nett-tilgjengelighet har fått et økende fokus de siste årene, og spiller en sentral rolle for at webapplikasjonen er tilgjengelig for så mange som mulig uavhengig av funksjonsnivået til personen. For å sikre at en applikasjon oppfyller disse kravene, har organisasjoner som Datatilsynet [75] i Norge, og *World Wide Web Consortium* [81] internasjonalt, utformet en rekke retningslinjer på hvordan en applikasjon skal utvikles.

Spesielt skal retningslinjene spesifisert i *Web Content Accessibility Guidelines (WCAG) 2.1* [82] følges for å tilrettelegge for brukere som har nedsettelse som påvirker brukerens evne til å benytte seg av applikasjonen. Dette er for å tilrettelegge for brukere som har nedsettelse som kan føre til et *funksjonsgap* mellom brukernes forutsetninger og samfunnets krav, som vist i figur 4.21.



Figur 4.21: Illustrasjon over funksjonsgap. Hentet fra:[83]

Applikasjonen Library of Things skal være tilgjengelig for flest mulig brukere, og det er derfor viktig at applikasjonen oppfyller kravene spesifisert i WCAG 2.1. Dette er viktig, ikke bare for tilgjengelighetens skyld, men også fordi mange av kravene er lovpålagte. Ønsket er å tilrettelegge for å redusere funksjonsgapet ved å styrke brukernes forutsetninger.

Søkemotoroptimalisering (SEO) er også noe man bør ha i bakhodet når man utvikler en applikasjon. En stor fordel med en universelt utformet applikasjon er at SEO-raten til applikasjonen vil økes. Dette kan medføre at applikasjonen blir mer synlig for fremtidige brukere.

5 RESULTATER

Kapittelet beskriver metodene som ble benyttet for å evaluere prosjektet. Dette inkluderer bruker-, system- og akseptanse-testing og dybdeintervju med oppdragsgiver. Resultatene fra evalueringsmetodene presenteres, og prosjektresultatet blir gjennomgått.

5.1 Evalueringsmetode

For å evaluere prosjektet har gruppen gjennomført brukertesting, *system- og akseptansetesting*, og dybdeintervju med oppdragsgiver. Brukertesten ga innblikk i hvordan applikasjonen opplevdes, og ga en pekepinn på om applikasjonen oppleves som tilgjengelig, sikker, og enkel å bruke. System- og akseptansetestene som ble utført ble brukt for å vurdere om applikasjonen tilfredsstilte kravene til applikasjonen. Avslutningsvis avdekker dybdeintervjuet om gruppen har oppfylt ønskene oppdragsgiver hadde for prosjektet.

5.1.1 Brukertesting

Gruppen valgte å gjennomføre brukertesting for å evaluere brukeropplevelsen til applikasjonen. Dette valget ble tatt med hensyn til at brukertesting kan vise hvor godt applikasjonen fungerer i praksis og om den oppfylder brukernes behov. Det var 12 personer som deltok i testen og testgruppens alder var fra 18 – 50+. Testene ble gjennomført fysisk siden deltakerne skulle utføre oppgaver som krevde at en administrator var til stede for å sjekke ut og inn lån.

For å sikre at brukertesting ble gjennomført på en organisert måte, ble deltakerne gitt et sett med instruksjoner på papir. Instruksjonene veiledet dem gjennom bestemte oppgaver, og gruppen var tilgjengelig dersom det var spørsmål. Deltakerne ble bedt om å gjøre oppgaver som å logge inn i applikasjonen og å gjennomføre låneprosessen fra start til slutt. Deltakerne ble også bedt om å bytte språk og å slette brukeren sin. Det var viktig å få en oversikt om det var enkelt å bytte språk for å avklare om applikasjonene er internasjonalisert.

Etter brukertesting ble deltakerne bedt om å delta i en spørreundersøkelse for å gi gruppen tilbakemeldinger om applikasjonens funksjoner og brukervennlighet. Spørreundersøkelsen inkluderte spørsmål om deltakerens opplevelse rundt å utføre spesifikke oppgaver og deres generelle inntrykk av applikasjonen. Det var også ønskelig å få informasjon om brukerens kjennskap til Lightning Wallet og åpenhet for en slik innlogging, samt hvor viktig personvern er for deltakerne.

5.1.2 Systemtesting

Systemtesting er prosessen av å teste systemet i sin helhet basert på dets funksjonelle og ikke-funksjonelle spesifikasjoner. *Brukstilfeller*, *klassediagram* og andre *UML-diagrammer* kan forme kriteriene for systemtestingen [83]. Gruppen fokuserte på *black box testing* [1], som er en prosess hvor testeren kun vet at det er mulig å gi informasjon inn i boksen, og at boksen

vil sende noe tilbake. Denne testmetodikken kan utføres ved at testeren vet hvilke resultat som forventes, og at utfallet samsvarer med forventningene. Brukstilfellene som beskrevet i delkapittel 4.3.1 vil representere det forventede resultatet for systemtestingen.

5.1.3 Akseptansetesting

For å best representere oppdragsgivers interesser, har gruppen benyttet seg av en akseptansetest. En akseptansetest brukes for å teste at det endelige systemet utfører funksjonene kunden ønsket. Brukstilfeller brukes ofte som en intensjonsavtale mellom kunden og utviklerne for enighet om hva systemet skal gjøre [84].

Akseptansetesten gruppen og oppdragsgiver utførte ble derfor formet på bakgrunn av brukstilfellene, hvor hoved-, alternativ- og unntaks-flyten beskrevet i kravdokumentasjon (se vedlegg 4) ble en viktig del av akseptanskriteriene. Oppdragsgiver gikk igjennom applikasjonene som bruker, administrator og superadministrator som en del av akseptansetesten.

5.1.4 Dybdeintervju

For å evaluere prosjektets resultat valgte gruppen å gjennomføre et dybdeintervju med oppdragsgiver, Fredrik M. Salhus. Hensikten med intervjuet var å vurdere om oppdragsgiver var fornøyd med applikasjonen. Intervjuet dekket oppdragsgivers generelle inntrykk, vurdering av hvordan gruppen implementerte designet og funksjonalitet. Videre ble det stilt spørsmål om oppdragsgiverens ønsker for videre arbeid av applikasjonen.

5.2 Evalueringsresultat

Dette kapittelet presenterer resultatene fra evalueringsmetodene gjennomført i prosjektet. Gjennom system- og akseptansetesting, brukertesting, og dybdeintervju med oppdragsgiver ble applikasjonens funksjonaliteter og brukeropplevelse vurdert.

5.2.1 Evalueringsresultat av brukertesting

Den første observasjonen var at alle deltakerne klarte å utføre hovedfunksjonene i applikasjonen, som å legge til og låne gjenstander. Dette indikerte at funksjonene fungerer som ønsket. Ved reservering av gjenstander foreslo noen deltakere å inkludere en filtreringsfunksjonalitet for å gjøre det enklere å finne ønskede gjenstander. Det ble også foreslått å ha en tydeligere tilbakemelding når en gjenstand legges inn i plukklisten.

Brukertesten avdekket en feil der to deltakere ikke kunne se sine reserverasjoner og lån i applikasjonen. Enkelte testbrukere opplevde utfordringer med navigeringen til applikasjonen, dette var et kjent problem for gruppen.

Når det gjaldt innloggingen, var flertallet av deltakerne enig i at denne prosessen var enkel og effektiv. Noen av deltakerne mente likevel det var unødvendig å bruke denne type innlogging da de allerede brukte innloggingstjenester som Vipps og BankId. Enkelte av deltakerne som var skeptiske til løsningen, mente de bare trengte å bli vant med å bruke Lightning Wallet.

Til slutt ble brukerne bedt om å slette kontoen sin. Selv om det kun ble gitt en instruks om å slette kontoen uten ytterligere veiledning, klarte alle deltakerne å gjennomføre denne oppgaven.

5.2.2 Evalueringsresultat av system- og akseptansetesting

Flere av arbeidsmøtene med oppdragsgiver besto av å utføre systemtesting av applikasjonen. Den iterative tilnærmingen gruppen hadde til utviklingsprosessen førte til at systemtestingen startet med at brukerflyten til applikasjonen samsvarte med kravene og brukstilfellet som avtalt. Etter hvert som gruppen fikk på plass ytterligere funksjonaliteter, ble disse også en del av systemtestingen. Den iterative tilnærmingen til utviklingen, og testingen, resulterte i at gruppen alltid hadde en fungerende applikasjon i tråd med oppdragsgiver sine ønsker.

Resultatene av akseptansetesting vil avgjøre om applikasjonen regnes som en suksess fra oppdragsgiver sitt perspektiv. Figur 5.1 viser kriteriene for akseptansetesten samt om applikasjonen oppfylte disse.

Bruker	ikke		Administrator og superadministrator	ikke	
	akseptert	akseptert		akseptert	akseptert
Registrer/logg inn	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Logg inn	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Legge til gjenstand i lånekurv	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Legg til/fjern reservasjon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Fjern gjenstand fra lånekurv	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Fjern bruker	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Opprett reservasjon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Endre/legg til/fjern gjenstand	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Slett reservasjon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Endre/legg til/fjern lokasjon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
			Endre/legg til/fjern kategori	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figur 5.1: Resultatene av akseptansetest. Hentet fra: Egen illustrasjon i Canva.

Fra figur 5.1, og gjennom tilbakemelding fra oppdragsgiver, kan akseptansetesting regnes som en suksess.

5.2.3 Evalueringsresultat av dybdeintervju

I dybdeintervjuet med oppdragsgiver bekrefter han at kravene har truffet hans ønsker, og han er spesielt fornøyd med at applikasjonen er nærmere ferdig produkt enn et MVP. Han uttrykker stolthet over flere funksjoner, spesielt QR-kode skanning, som han mener forenkler interaksjonen mellom bruker og administrator. Videre uttrykker han tilfredshet med løsningen gruppen har implementert for autentisering gjennom Lightning Wallet. Generelt

sett overstiger applikasjonen hans forventninger, og han er særlig imponert over den samlede innsatsen fra gruppen. Til slutt foreslår oppdragsgiver forbedringer som implementering av *NFC-brikker (Near Field Communication - tagger)* [85]. Han gir også ros til samarbeidet med gruppen og uttrykker håp om å utvide bruken av applikasjonen til ulike lokalsamfunn.

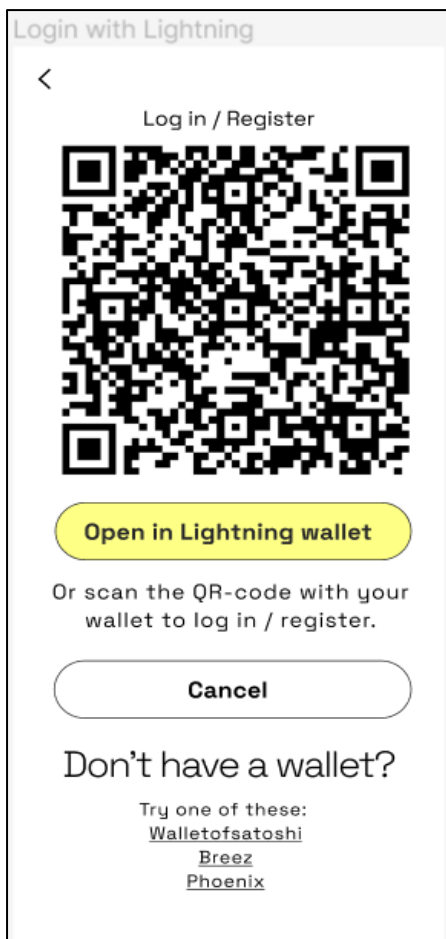
5.3 Prosjektresultat

Resultatet av prosjektet er en internasjonalisert *progressiv webapplikasjon* for utlån av gjenstander. Applikasjonen er tilrettelagt for å kunne skaleres på et globalt nivå, og har i stor grad lagt til rette for universelt utformet funksjonalitet. Dette kapittelet gir en oversikt over prosjektets resultat, hvordan det endelige designet var implementert, applikasjonens oppfylte krav, og dokumentasjonen til prosjektet.

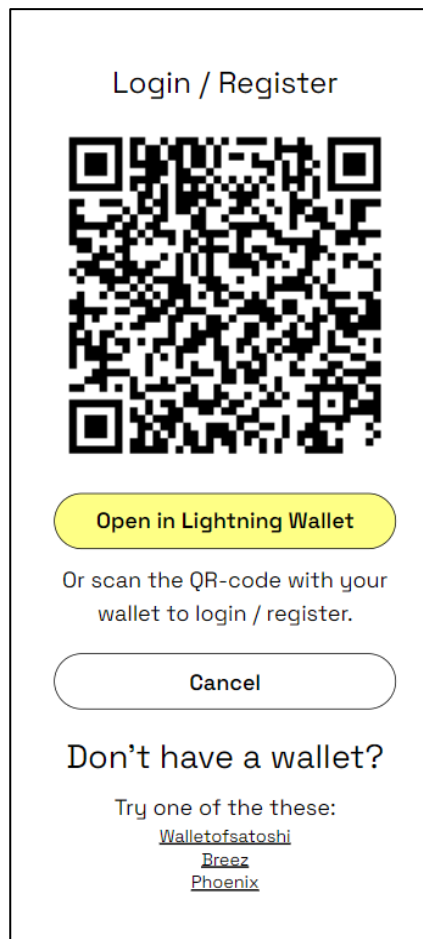
5.3.1 Endelig implementasjon av design

Oppdragsgiver ønsket kun at applikasjonen skal være designet og tilpasset mobile enheter i denne omgang. Gruppen har likevel tilrettelagt for at designet er tilpasset desktop og større mobile enheter. Dette medførte at gruppen brukte mer tid på implementeringen av designet. Samtidig følte gruppen at det var riktig å prioritere implementering av designet ettersom tilgjengelighet var en av de viktigste prinsippene for prosjektet.

For å vurdere om designet gruppen har implementert stemmer overens med designet, har gruppen fått løpende tilbakemeldinger fra oppdragsgiver. Gruppen har også utført et dybdeintervju med oppdragsgiver, hvor oppdragsgiver opplyste at han var særdeles fornøyd med implementasjonen av designet. Videre ble oppdragsgiver og gruppen enig om at designet til enkelt av administratorsidene ikke skulle prioriteres, og ble derfor ikke designet av oppdragsgiver og heller ikke implementert av gruppen. Figur 5.2 viser hvordan designet til innloggingssiden er utformet i Figma og figur 5.3 viser en skjermdump av innloggingsiden til applikasjonen.



Figur 5.2: Figma design av innloggingsside. Hentet fra: Eget Figma prosjekt



Figur 5.3: Implementert innloggingsside design. Hentet fra: Egen kildekode

5.3.2 Funksjonelle og ikke-funksjonelle krav

Ved oppstart av prosjektet ble det spesifisert en rekke krav for applikasjonen, disse kravene er beskrevet i visjonsdokumentet (se vedlegg 7). De initielle kravene i kapittel 2.1.2 ble formet med utgangspunkt fra kravene beskrevet i dokumentet. Det endelige produktet oppfylte alle de initielle kravene, men det var enkelte funksjonelle- og ikke-funksjonelle krav som ikke ble prioritert. Disse kravene er beskrevet i tabell 3, og kan brukes som et utgangspunkt til videre arbeid med applikasjonen.

Tabell 3: Oversikt over krav som ikke ble oppfylt

Krav type	Beskrivelse:
Funksjonell	De som ønsker å låne noe skal få beskjed dersom en gjenstand ikke er levert inn til nytt utlån.
Funksjonell	Legge inn en lånekø på utlånt utstyr.
Funksjonell	Det skal sendes en påminnelse når innleveringsfristen nærmer seg.
Ikke-funksjonell	Kravene om universell utforming må følges.

5.3.3 Dokumentasjon

Ettersom applikasjonen i utgangspunktet skulle utforme et *minimum viable product*, var det viktig for gruppen å dokumentere kodebasen og utviklingsprosessen slik at det var mulig og oversiktlig for andre å overta kodebasen. Selv om gruppen og applikasjonen avveik fra MVP-tilnærmingen, og gruppen produserte en applikasjon nærmere et sluttprodukt, vil dokumentasjonen være nyttig for å videreutvikle applikasjonen.

Det var spesielt tre områder av dokumentasjonen gruppen fokuserte på. Disse er oversiktlig presentert i *README.md*-filen, se vedlegg 6. Det første var å skrive *dokumentasjonsstrenger* på formen av *JSDoc* [72] til alle globale funksjoner, samt andre funksjoner som kunne virke uklare for andre utviklere. Videre har gruppen laget en oversikt over alle installasjoner som er utført underveis i prosjektet. Til slutt har gruppen laget en *README.md*-fil som inneholder en oversikt over denne dokumentasjonen, samt annen relevant informasjon tilhørende prosjektet.

Commit-historikken i GitHub kan også brukes for å få en oversikt over hvordan gruppen har jobbet med oppgaven. Historikken inneholder kodeendringer, kommentarer på hva disse endringene innebærer, og tidspunktene de ble utført. Siden Git fungerer som et versjonskontrollsystem, kan kommende utviklere få tilgang til tidligere versjoner av kildekoden gjennom commit-historikken.

6 DISKUSJON

Dette kapitlet diskuterer ulike aspekter ved prosjektet, inkludert avgrensninger, valg av verktøy og metodikk. Videre blir resultatene av applikasjonen analysert, samt utfordringer som oppsto underveis. Diskusjonen inkluderer også gjennomgang av brukertesting, foreslåtte forbedringer i løsningen og hvordan gruppen har opplevd samarbeid på tvers av disipliner.

6.1 Avgrensninger og valg

Omfanget av å lage en applikasjon som ville oppfylle idéen bak Library of Things, ble tidlig avklart av gruppen og oppdragsgiver som for omfattende til å fullføre på den gitte tiden. Derfor har det vært stort fokus på avgrensninger og valg av riktige teknologier gjennom prosjektet. De initielle kravene som spesifisert i delkapittel 2.1.2, var funksjonene oppdragsgiver ønsket applikasjonene skulle inneholde.

Selv om gruppen og oppdragsgiver ble enige om å utvikle et MVP, ble det gjennom de ukentlige arbeidsmøtene lagt til ytterligere funksjoner til applikasjonen. Gruppen kom blant annet selv med idéen om å bruke QR-koder for inn- og utsjekk av gjenstandene. Arbeidsmøtene har bidratt til at gruppen kunne bedre dekke behovene til oppdragsgiveren, samtidig førte de også til at kompleksiteten til applikasjonen økte. Dette førte til at applikasjonen avveik fra MVP-tilnærmingen gruppen hadde planlagt.

6.1.1 Valgte verktøy

Verktøyene som ble benyttet i prosjektet ble foreslått av gruppen til oppdragsgiver. Gruppen hadde i stor grad fritt valg til å bestemme teknologier, bortsett fra at autentiseringen skulle tilrettelegge for innlogging med *Lightning Wallet*. Det ble bestemt at *BaaS-tjenesten Firebase* [40] skulle bli benyttet, men gruppen følte underveis at andre alternativer burde vært utforsket nærmere.

Tjenestene som gjør Firebase unik, kunne ikke bli utnyttet fullstendig grunnet manglende løsning for autentisering med *Lightning Wallet*. Konsekvensen var at en alternativ backend-løsning måtte settes opp for å legge til rette for funksjonaliteten som Firebase ikke støttet, som for eksempel e-post verifisering.

Det kan diskuteres om en backend-løsning kun burde vært basert på *Express.js* siden flere av Firebase sine funksjoner likevel ikke kunne brukes i prosjektet. Gruppen mener likevel at selv om løsningen kunne dekket behovene bedre ville omfanget av å lære seg både *React* og *Express.js* gått på bekostning av fremgangen til prosjektet. Det er heller ingen garanti at backend-løsningen kunne levert samme ytelse som *Firebase*.

Oppdragsgiver kommuniserte tidlig i prosjektet et ønske om at applikasjonen skulle være en webapplikasjon med design tilpasset for mobiltelefoner. Dybdeintervjuet som ble gjort med oppdragsgiver understreker også dette. Designet ble derfor tilpasset mobil og designet på PC og store enheter ble ikke prioritert. Gruppen synes ikke denne tilnærmingen nødvendigvis var

riktig og føler at en PWA hadde passet oppgavebeskrivelsen bedre. Fordelen med løsningen som ble valgt er at bruker unngår å laste ned applikasjonen. Dette reduserer kostnadene med å gjøre applikasjonen tilgjengelig for mobile enheter.

6.2 Applikasjonsresultat

Dette kapittelet gir en visuell representasjon av løsningen, og vises gjennom figurer hentet fra applikasjonen. Applikasjonen er blitt gjort tilgjengelig på domenet <https://libraryofthings.com/>. Vedlegg 2 inneholder flere bilder av applikasjonen, samt en video som viser applikasjonen.

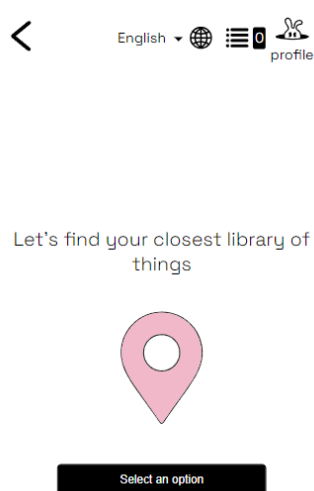
Fra delkapittel 1.4 og 2.1.2 ble følgende funksjoner definert som en del av målet for prosjektet:

- Det skal være mulig å velge en lokasjon og å legge ønskede gjenstander inn i en plukklister.
- Det skal være mulig å reservere gjenstandene i plukklister.
- En administrator skal kunne legge til nye gjenstander.
- En administrator skal kunne legge til og fjerne lokasjoner.

Figurene 6.1-6.6 viser flyten til hvordan en bruker kan velge lokasjon, finne ønskede gjenstander og legge dem inn i plukklister. Vedlegg 2 inneholder skjermdumper for administratorflyten, samt øvrige bilder av applikasjonen. Dette vedlegget vil også inneholde en demonstrasjonsvideo av applikasjonen.



Figur 6.1: Hjemmesiden til applikasjonen. Hentet fra: Egen kildekode



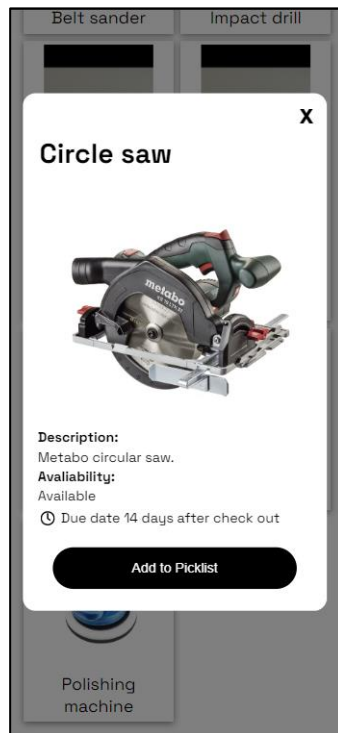
Figur 6.2: Velg land-siden til applikasjonen. Hentet fra: Egen kildekode



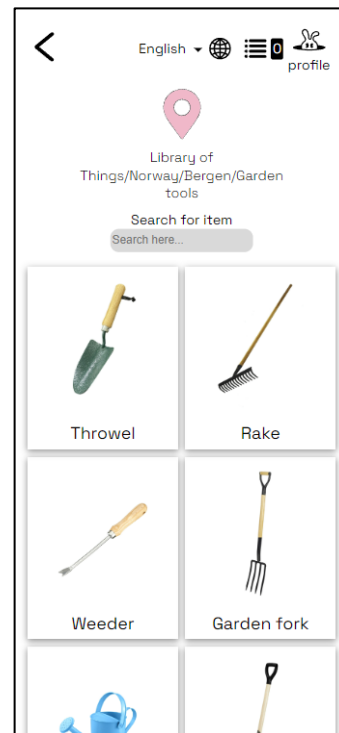
Figur 6.3: Velg lokasjon-siden til applikasjonen. Hentet fra: Egen kildekode



Figur 6.6: Kategorivalg siden til applikasjonen. Hentet fra: Egen kildekode



Figur 6.4: Hvordan man kan legge til en sirkelsag i plukklisten. Hentet fra: Egen kildekode



Figur 6.5: Utstyret tilgjengelig på "garden tools" kategorien. Hentet fra: Egen kildekode

6.3 Avvik fra initielle løsnings-idé

Løsningen gruppen har utviklet har oppfylt alle kravene spesifisert i den initiale løsnings-idéen. Avvikene fra disse kravene har vært supplerende tiltak gruppen selv har valgt å legge til. Gruppen har laget en QR-kode-løsning for inn- og utsjekk, en epost-verifiserings funksjon ved bruk av *Nodemailer-rammeverket*, og tilrettelegging for å legge til *NFC-brikker* [85] til gjenstandene.

Funksjonene gruppen valgte å legge til, har ført til at applikasjonen bedre oppfyller alle ønskene til oppdragsgiveren. Dette har resultert i et mer fullstendig sluttprodukt. Ulempen ved å legge til disse funksjonene, er at det har økt antall arbeidstimer gruppen har brukt på selve utviklingen. Det er mulig at ved å prioritere tiden annerledes så kunne gruppen ytterligere prioritert andre deler av utviklingsprosessen, som testing, bug-identifisering og bug-fiksing.

6.4 Brukertesting

Brukertest ble utført som evalueringsmetode for å besvare forskningsspørsmålet og for å undersøke om problemstillingen ble besvart. Det ble lagt vekt på å prøve å samle data fra varierte aldersgrupper og nivå på datakunnskapene.

Brukertestingen ble forsinket på grunn av manglende design på sidene som skulle testes. Opprinnelig var planen å gjennomføre brukertesten mot slutten av mars måned, men ble

gjennomført mot slutten av april. Gruppen har i etterkant av brukertesting reflektert over muligheten for å gjennomføre brukertester på et tidligere tidspunkt, uten implementering av design, for å kartlegge styrker og svakheter i applikasjonen. Da ville det vært tid til å legge til rette for iterativ implementasjon av større endringer i funksjonalitet. Dette kunne ført til at færre funksjoner ble tatt med til videre arbeid. Flere iterasjoner kunne også ha avklart om de nye implementeringene har løst de identifiserte problemene fra tidligere iterasjoner.

Under brukertesting mottok deltakerne et instruksjonsark som gruppen hadde laget på forhånd. Det ble observert at noen av deltakerne misforstod deler av instruksene, noe som kan ha resulterte i feilaktige resultater. Derfor ville det vært til fordel for gruppen å få en tredjepart til å lese gjennom instruksene og gitt tilbakemelding på hva som kunne vært uklart.

Samlet sett dekket resultatene svarene på hva gruppen ønsket å oppnå gjennom testingen, og det ble identifisert problemer som gruppen ikke var klar over på forhånd. I tillegg mottok gruppen forslag fra deltakerne til forbedringer som kan tas med i det videre arbeidet.

6.5 Prosjektmetodikk

Prosjektet ble gjennomført med sprinter som hadde varighet på en uke. For å separere arbeidet ble sprintukene delt opp i to fokusområder: utvikling og rapportskrivning. En negativ konsekvens av å benytte denne tilnærmingen var at gruppen opplevde at noen deler av arbeidet utført i tidligere sprinter ble bortglemt og måtte friskes opp. Til tross for dette, ble den foretrukne måten valgt for å sikre et tydelig skille mellom utvikling- og rapportskrivingsprosessen. Dette viste seg å være en effektiv måte for å oppnå jevn fremgang på begge fokusområdene, og de daglige stand-up møtene bidro til å redusere konsekvensene av tilnærmingen.

For å bidra til jevn fremgang, var det viktig for gruppen å ha klare mål for hver uke. For å følge opp disse ble en *Kanban-tavle* benyttet til å fordele oppgaver. Denne metodikken krevde at alle kritiske oppgaver ble utført først. Likevel opplevde gruppen at noen kritiske oppgaver ble fullført senere enn ønsket, på bakgrunn av medlemmenes egne preferanser. En positiv side ved å benytte *Kanban-tavlen* var at fordeling av oppgaver ble lettere. Gruppen kunne tydelig se progresjon, identifisere problemer og justere prioriteringene underveis.

Gruppen fulgte en iterativ tilnærming gjennom å følge de ukentlige sprintene som ble satt. Startfasen av prosjektet fokuserte på å få på plass hovedfunksjonene i applikasjonen uten tanke på å involvere designet inn i den iterative prosessen. Det førte til at designer ikke fikk den samme iterative tilnærmingen og at designet ble gjennomført i kun to iterasjoner. Det var perioder hvor det var usikkerhet rundt designet og gruppen ventet på at den andre iterasjonen skulle ferdigstilles. For fremtidige prosjekter vil det være en fordel å inkludere designeren i den iterative prosessen.

Samarbeidet i prosjektet foregikk i stor grad ved å sitte i samme rom. Dette følger et av prinsippene i *det smidige manifestet*, «Den mest effektive måten å formidle informasjon inn til og innad i et utviklingsteam, er å snakke ansikt til ansikt.» [52]. Tidvis kunne det være utfordrende å sitte sammen på grunn av forskjellige timeplaner. Med tanke på de forutsetningene som lå til grunn, fungerte denne tilnærmingen godt da samtlige

gruppemedlemmer var i stor grad til stede fysisk og problemer kunne løses med diskusjon ansikt til ansikt.

6.6 Forbedringer i endelig løsning

Dersom gruppen skulle utvikle applikasjonen på nytt, er det aspekter av prosessen som har forbedringspotensial. Når det kommer til prosjekt- og utviklings-metodikkene, er gruppen stort sett fornøyd, men det ville vært naturlig å involvere designet tettere i den iterative tilnærmingen gruppen benyttet. Dette ville ført til at gruppen forhåpentligvis ikke måtte vente på designet i like stor grad.

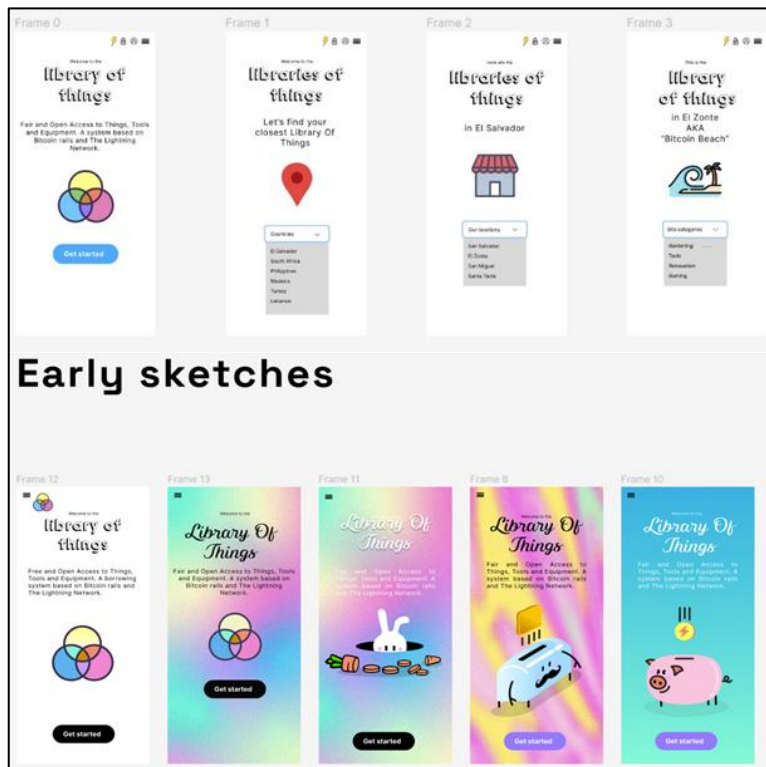
Selv om ikke oppdragsgiver ønsket å fokusere på ytelse, og oppdragsgiver ble fornøyd med hastigheten til den endelige løsningen, ville gruppen brukt mer tid på å utforske måter å gjøre applikasjonene raskere. Spesielt så ville mellomagringsmulighetene utforskes dypere, da gruppen erfarte mangler i den valgte løsningen. *React* sine innebygde funksjoner for å optimalisere ytelsen burde også blitt benyttet i større grad.

Den endelige system-arkitekturen til applikasjonen avveik en del fra hvordan den var modellert. Dersom gruppen skulle laget applikasjonen på ny, kunne det vært fornuftig å bruke mer tid til å planlegge arkitekturen.

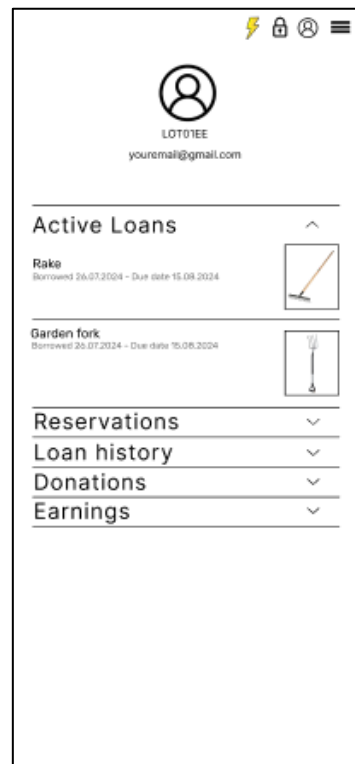
Sikkerheten til applikasjonen har også forbedringspotensial. Selv om gruppen har valgt verktøy som i stor grad håndterer sikkerheten til applikasjonen, finnes det sikkerhetshull som muligens kan utnyttes. Dersom gruppen hadde tid og ressurser, ville det vært ønskelig å utføre forskjellige sikkerhetsøkende tiltak, for eksempel å utføre *penetrasjonstesting*. Det ville også vært ønskelig å utføre *integrasjon-* og *enhetstester* i applikasjonen, i tillegg til *system-* og *akseptansetestene* gruppen utførte. Ved å utføre flere automatiske tester, vil det øke sikkerheten til applikasjonen, og det vil forhindre at uønskede endringer påvirker funksjonene til applikasjonen.

6.7 Design

Designet til applikasjonen ble utformet av oppdragsgiver. Selv om gruppen hadde en stemme i utformingen, var det oppdragsgiver som laget og leverte designet til gruppen. Et problem med denne tilnærmingen var at gruppen var avhengig av at oppdragsgiver kunne levere designet til ønsket tid, noe som ikke alltid var tilfellet. Den første utkastet av designet gruppen fikk ble levert i midten av februar, og er vist i figur 6.7 og 6.8.

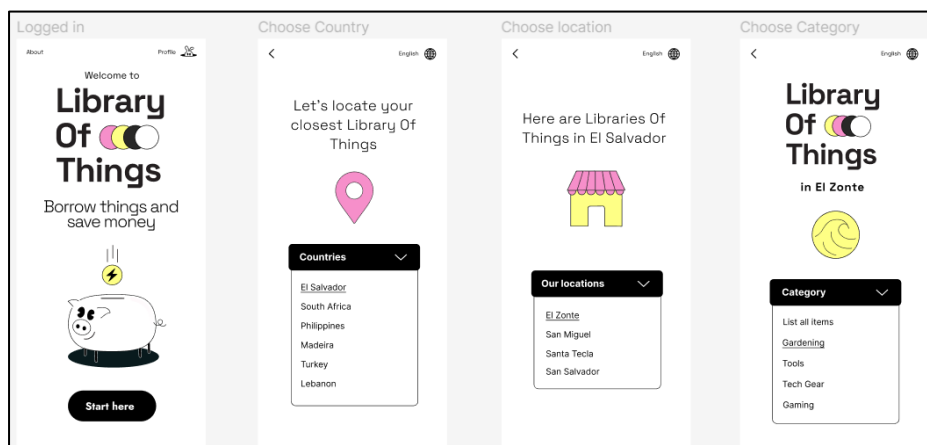


Figur 6.7: Førsteutkast av design. Hentet fra: Eget Figma prosjekt

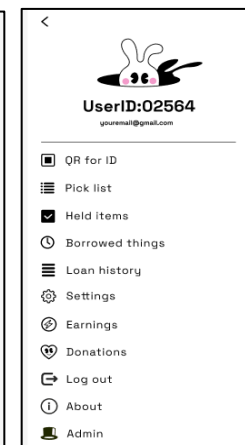


Figur 6.8: Førsteutkast av brukerprofil. Hentet fra: Eget Figma prosjekt

Etter at gruppen hadde begynt å implementere dette designet, ble ikke designet nevneverdig endret før i løpet av påskeferien. Da fikk designet store endringer, se figur 6.9 og 6.10. Dette førte til at gruppen måtte gjøre endringer i kodebasen, og stilarkene for å gjenspeile forandringene. Dersom gruppen hadde vært flinkere til å inkludere oppdragsgiver i den iterative tilnærmingen, kunne designprosessen fulgt utviklingsprosessen i tråd.



Figur 6.10: Siste utkast av designet. Hentet fra: Eget Figma prosjekt



Figur 6.9: Siste iterasjon av brukerprofil. Hentet fra: Eget Figma prosjekt

Designet oppdragsgiver laget var spesifikt tilpasset mobile enheter, og hvordan designet til applikasjonen på andre enheter skulle utformes ble overlatt til gruppen. Dette førte til at gruppen måtte bruke tid på både utforming og implementering av designet parallelt. Det er mulig at tiden brukt på denne prosessen ville vært bedre anvendt på andre deler av applikasjonen. Dette førte også til at gruppen ikke fikk oppfylt kriteriene for *universell utforming* i like stor grad som ønsket.

Under prosjektet stod gruppen overfor utfordringer knyttet til å definere applikasjonens funksjoner, da oppdragsgivers ønsker om funksjoner ikke var tydelige i begynnelsen. Det krevde mye innsats fra gruppens side for å definere og utforme funksjonalitet som møtte prosjektets mål.

6.8 Samarbeid på tvers av disipliner

Ved å ha en oppdragsgiver som også fylte rollen som designer for mobilvisning av applikasjonen, fikk gruppen en unik mulighet å jobbe i et tverrfaglig team gjennom prosjektet. Samarbeidet mellom gruppen og designeren opplevdes som veldig bra, og spesielt arbeidsmøtene bidro til at det var mulig å dele erfaringer og kunnskap på tvers av disipliner. Denne tilnærmingen gjorde at applikasjonen ble tilpasset endringer på en dynamisk måte gjennom prosjektets forløp.

Samarbeidet var verdifullt for prosjektgruppen som lærte mye av denne måten å arbeide på. Gruppen fikk en dypere forståelse for hvordan designvalg kan skape en mer helhetlig og brukervennlig opplevelse. Denne erfaringen utvidet ikke bare faglig kunnskap, men styrket også evnen til å samarbeide med andre som man ikke kjenner fra før.

Oppdragsgiver hadde en utstilling av sin bachelor oppgave som omhandlet designet til LOT. Gruppen var til stede på denne, figur 6.7 og figur 6.8 viser deler av oppdragsgiver sin utstilling.



Figur 6.11: Oppdragsgiver sitt bord med utstyr og plakat. Hentet fra: Privat.



Figur 6.12: Del av oppdragsgiver sin fremvisning. Hentet fra: Privat

7 KONKLUSJON OG VIDERE ARBEID

Bachelorprosjektet hadde som formål å utvikle en fullstack applikasjon for utlån av utstyr. Det var ønskelig at applikasjonen skulle være internasjonalisert og globalt skalerbar. Videre har prosjektet tatt sikte på å utforske om en applikasjon som tilbyr innlogging med *Lightning Wallet* vil bidra til en mer brukervennlig og tilgjengelig utlånstjeneste. Dette kapittelet vil undersøke om applikasjonen, og prosjektet, oppfyller kravene på bakgrunn av resultatene og diskusjonene som ble presentert i rapporten.

7.1 Forskningsspørsmål

Gruppen vil gjøre oppmerksom på at tjenesten er tenkt å bli tatt i bruk i andre land enn Norge. For å få et representativt og helhetlig resultat, anbefaler gruppen å utføre brukertester også i disse landene.

Spørsmålet som ble formulert ved starten av prosjektet var:

I hvilken grad vil en applikasjon som tilbyr Lightning Wallet-innlogging bidra til en mer brukervennlig og tilgjengelig utlånstjeneste?

Applikasjonen som er utviklet fungerer godt til utlån av utstyr. Gjennom evaluering av *Lightning Wallet-innloggingen* kom det frem at testere som ikke har blitt eksponert for denne typen innlogging tidligere ønsket flere lignende innloggingstjenester. Evalueringen viste også at det var en brukervennlig prosess for å kunne låne utstyr. Dette indikerer at prosjektet i stor grad har nådd målet om å bidra til at utlån av utstyr er mer brukervennlig og tilgjengelig.

7.2 Konklusjon

Formålet med prosjektet *Library of Things* var å utvikle en applikasjon som var tilgjengelig for alle uavhengig av deres geografiske lokasjon og språket de behersker. Applikasjonen skulle benytte seg av en innlogging- og autentiserings-tjeneste basert på kryptoteknologi, og løsningen skulle være globalt skalerbar og internasjonalisert.

På bakgrunn av problembeskrivelsen og målet beskrevet i delkapittel 1.4 og 2.1.2 har gruppen identifisert tre kriterier for å vurdere prosjektets suksess:

1. Oppfyller applikasjonen kravene som ble satt?
2. Er oppdragsgiver fornøyd med applikasjonen?
3. I hvilken grad svarer applikasjonen på problemstillingen?

Resultatene fra akseptansetesten viser at applikasjonen svarer på alle kravene som ble satt ved prosjektets start. Diskusjonene rundt den endelige løsningen har avdekket at selv om kravene er møtt, finnes det forbedringspotensial for applikasjonen. Spesielt må sikkerheten til applikasjonen gjennomgås før den rulles ut.

Gjennom dybdeintervjuet med oppdragsgiver (se vedlegg 3), kommer det tydelig frem at oppdragsgiver er svært fornøyd med applikasjonen. Den oppfyller alle kravene han hadde, og applikasjonen tilbyr tilleggsfunksjoner som gjør interaksjonen mellom administrator og

brukerne bedre. Fra dybdeintervjuet uttalte oppdragsgiver at «*Jeg er mer enn fornøyd med arbeidet som er gjort, og det overskrider det jeg hadde forventet*».

Ved å velge teknologier som er skalerbare, har gruppen oppfylt målet om å gjøre applikasjonen tilgjengelig uavhengig av geografisk lokasjon. Autentiseringsløsningen basert på innlogging med *Lightning Wallet* har redusert behovet for en spesifikk geografisk tilknytning. Samtidig er autentiseringen uavhengig fra de store aktørene.

Rammeverket *React-i18next* har gjort det raskt og enkelt å tilpasse applikasjonen flere forskjellige språk. Videre inneholder README-filen instruksjoner på hvordan dette kan bli gjort. Derfor kan det konkluderes med at applikasjonen i stor grad svarer på problemstillingen ved å tilby en tilgjengelig og internasjonalisert løsning.

Fra diskusjon og resultater, kan applikasjonene regnes å ha oppfylt alle kriteriene. Prosjektet kan derfor vurderes som en suksess.

7.3 Videre arbeid

Delkapittelet tar for seg videre arbeid gruppen anbefaler bør implementeres før applikasjonen settes i bruk. For å gjøre det lettere for oppdragsgiver å iverksette disse forslagene, har gruppen oppdatert *Kanban-tavlen* til prosjektet, og lagt til relevant informasjon om enkelte av forslagene i *README-filen*.

7.3.1 Optimalisering

En viktig del av å gjøre en applikasjon skalerbar er at kildekoden er optimalisert. Dette innebærer at *databasespørringer*, *backend-forespørsler* og *klientside-kode* er skrevet for best mulig ytelse. Gruppens manglende erfaring med *React*, har ført til at det er mange områder som kan forbedres i *frontend-løsningen*. I kapittel 4.4.1 ble det nevnt funksjoner som kan føre til en bedre optimalisert *frontend-kode*, og det vil være naturlig å *refaktorere* kodebasen til å bruke disse funksjonene der det passer.

Det er også mulig å optimalisere bildefremvisningen i applikasjonen. Bilder som legges inn i applikasjonen blir lagret i tre ulike størrelser i *Firestore* sin *Storage*. Fremvisningen av bilder kan forbedres ved å hente ut riktig størrelse til de ulike fremvisningene. Dette kan gjøre fremvisningen raskere som kan bidra til en bedre brukeropplevelsen.

7.3.2 Internasjonalisering

Gruppen har tilrettelagt for *internasjonalisering* av applikasjonen. En viktig del av internasjonalisering, er at oversettelsene på applikasjonen ikke bare beholder ordenes betydning, men også deres oppfattelse. Dette betyr at dersom applikasjonen skal bli tatt i bruk globalt, må oppdragsgiver finne fagfolk som kan oversette applikasjonen på korrekt vis. For å tilrettelegge for dette, har gruppen inkludert en guide til hvordan man kan legge til flere språk i *README.md* filen.

En utfordring med denne guiden, er at for å legge til oversettelser for nytt språk, så må man inn i kodebasen for å legge til en ny mappe for språket, samt den tilhørende JSON filen. Det

kan være ønskelig å utforske, eller utvikle, andre alternativer som gjør denne prosessen enklere for personer uten erfaring med utvikling.

7.3.3 Utforming av design for andre enheter

For å sikre en god brukeropplevelse og økt tilgjengelighet på ulike enheter, er det nødvendig å utvikle design tilpasset de ulike plattformene. Som tidligere nevnt, så utformet gruppen selv designet for større enheter. Gruppen mangler erfaringer innen design, brukeropplevelser og brukergrensesnitt. Derfor burde designet gjennomgås og tilpasses der det er nødvendig.

7.3.4 Tilleggsfunksjoner

Før applikasjonen gjøres tilgjengelig i andre land foreslår gruppen at brukertesting blir utført i disse landene. Spesielt land med mennesker som benytter *Lightning Wallet* regelmessig kan gi et bedre inntrykk om levedyktigheten til applikasjonen.

Oppdragsgiver ønsker funksjonalitet for *NFC-brikke* i applikasjonen. Anbefalingen fra gruppen er å bruke et *React-bibliotek* for å skanne *NFC-brikken* og legge til *NFC-id* i gjenstanden sitt dokument i databasekolleksjonen *Item*.

Applikasjonen er nødt til å bli grundig testet for å tette eventuelle sikkerhetshull før den blir tatt i bruk. Dette har blitt nedprioritert av gruppen grunnet tidsrammen og ønske fra oppdragsgiver. Gruppen sitt forslag er at sikkerhetskyndige tar en gjennomgang av applikasjonen og *penetrasjonstester* den.

Forsinkelsesgebyr skal benyttes i applikasjonen som forsikring om at gjenstandene kommer tilbake til biblioteket til avtalt tid. Gruppen sitt forslag er at dette gjøres gjennom en *betalingsforespørsel (pay request)* hos *LNURL* [86].

REFERANSER

- [1] S. Nidhra, «Black Box and White Box Testing Techniques - A Literature Review», *Int. J. Embed. Syst. Appl.*, bd. 2, s. 29–50, jun. 2012, doi: 10.5121/ijesa.2012.2204.
- [2] K. S. Lie og H. Øverby, «blokkjede», *Store norske leksikon*. 8. februar 2023. Åpnet: 10. mai 2024. [Online]. Tilgjengelig på: <https://snl.no/blokkjede>
- [3] «Markup language | Definition, Examples, & Facts | Britannica». Åpnet: 10. mai 2024. [Online]. Tilgjengelig på: <https://www.britannica.com/technology/markup-language>
- [4] «Peer-to-Peer nettverk (P2P)», Techopedia. Åpnet: 12. mai 2024. [Online]. Tilgjengelig på: <https://www.techopedia.com/no/ordliste/peer-to-peer-nettverk-p2p>
- [5] «Passing Props to a Component – React». Åpnet: 10. mai 2024. [Online]. Tilgjengelig på: <https://react.dev/learn/passing-props-to-a-component>
- [6] «Relasjonsdatabase», *Wikipedia*. 22. november 2023. Åpnet: 10. mai 2024. [Online]. Tilgjengelig på: <https://no.wikipedia.org/w/index.php?title=Relasjonsdatabase&oldid=23995941>
- [7] «Tingotek». Åpnet: 31. januar 2024. [Online]. Tilgjengelig på: <https://www.tingotek.no/hjemmeside>
- [8] «Vipps – veldig, veldig enkelt». Åpnet: 20. februar 2024. [Online]. Tilgjengelig på: <https://vipps.no/>
- [9] «FNs bærekraftsmål». Åpnet: 7. februar 2024. [Online]. Tilgjengelig på: <https://fn.no/om-fn/fns-baerekraftsmaal>
- [10] «Ansvarlig forbruk og produksjon». Åpnet: 31. januar 2024. [Online]. Tilgjengelig på: <https://fn.no/om-fn/fns-baerekraftsmaal/ansvarlig-forbruk-og-produksjon>
- [11] «Report», World Bank. Åpnet: 7. februar 2024. [Online]. Tilgjengelig på: <https://www.worldbank.org/en/publication/globalindex/Report>
- [12] «Pigify - Library of Things», pigify. Åpnet: 23. februar 2024. [Online]. Tilgjengelig på: <https://www.pigify.com>
- [13] A. N. Duc og P. Abrahamsson, «Minimum Viable Product or Multiple Facet Product? The Role of MVP in Software Startups», i *Agile Processes, in Software Engineering, and Extreme Programming*, H. Sharp og T. Hall, Red., i *Lecture Notes in Business Information Processing*. Cham: Springer International Publishing, 2016, s. 118–130. doi: 10.1007/978-3-319-33515-5_10.
- [14] «Ideal Modeling & Diagramming Tool for Agile Team Collaboration». Åpnet: 15. april 2024. [Online]. Tilgjengelig på: <https://www.visual-paradigm.com/>
- [15] «Welcome To UML Web Site!» Åpnet: 10. mai 2024. [Online]. Tilgjengelig på: <https://www.uml.org/>
- [16] «Canva: Visual Suite for alle», Canva. Åpnet: 11. mai 2024. [Online]. Tilgjengelig på: https://www.canva.com/nb_no/
- [17] «Figma: The Collaborative Interface Design Tool», Figma. Åpnet: 26. april 2024. [Online]. Tilgjengelig på: <https://www.figma.com/>
- [18] «What is Scrum? - Scrum Methodology Explained - AWS», Amazon Web Services, Inc. Åpnet: 23. februar 2024. [Online]. Tilgjengelig på: <https://aws.amazon.com/what-is/scrum/>
- [19] «Build software better, together», GitHub. Åpnet: 6. mars 2024. [Online]. Tilgjengelig på: <https://github.com>
- [20] «50+ Password Statistics: The State of Password Security in 2024», Exploding Topics. Åpnet: 15. april 2024. [Online]. Tilgjengelig på: <https://explodingtopics.com/blog/password-stats>

- [21] «Lightning Network», *Wikipedia*. 27. februar 2023. Åpnet: 11. april 2024. [Online]. Tilgjengelig på: https://no.wikipedia.org/w/index.php?title=Lightning_Network&oldid=23354236
- [22] «Peer-to-peer», *Wikipedia*. 27. september 2023. Åpnet: 11. mai 2024. [Online]. Tilgjengelig på: <https://no.wikipedia.org/w/index.php?title=Peer-to-peer&oldid=23854697>
- [23] «Web Apps vs. Native Apps vs. Hybrid Apps - Difference Between Types of Web and Mobile Applications - AWS», Amazon Web Services, Inc. Åpnet: 16. februar 2024. [Online]. Tilgjengelig på: <https://aws.amazon.com/compare/the-difference-between-web-apps-native-apps-and-hybrid-apps/>
- [24] «Web app | Definition, History, Development, Examples, Uses, & Facts | Britannica». Åpnet: 24. februar 2024. [Online]. Tilgjengelig på: <https://www.britannica.com/topic/Web-application>
- [25] «What is a progressive web app? - Progressive web apps | MDN». Åpnet: 16. februar 2024. [Online]. Tilgjengelig på: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/What_is_a_progressive_web_app
- [26] «What are Responsive Apps?», BrowserStack. Åpnet: 25. februar 2024. [Online]. Tilgjengelig på: <https://browserstack.wpengine.com/guide/what-are-responsive-apps/>
- [27] «What Is a Native App? (Beginner's Guide + Examples)», Mighty Networks. Åpnet: 16. februar 2024. [Online]. Tilgjengelig på: <https://www.mightynetworks.com/resources/native-app>
- [28] W. Technologies, «Techniques for Web Application Compatibility Across Browsers», Medium. Åpnet: 24. februar 2024. [Online]. Tilgjengelig på: <https://medium.com/@workboxtech/techniques-for-web-application-compatibility-across-browsers-8739e4891df4>
- [29] J. Tran, «PWA vs Native App and how to choose between them», World's #1 POS for Magento. Åpnet: 24. februar 2024. [Online]. Tilgjengelig på: <https://www.magestore.com/blog/pwa-vs-native-app-and-how-to-choose-between-them/>
- [30] «PWA vs Native apps». Åpnet: 5. mai 2024. [Online]. Tilgjengelig på: <https://www.barkweb.co.uk/blog/pwa-vs-native-apps>
- [31] «Progressive Web Apps vs Native Apps: What Should You Pick? | Turing». Åpnet: 24. februar 2024. [Online]. Tilgjengelig på: <https://www.turing.com/blog/progressive-web-apps-vs-native-apps/>
- [32] S. Walsh, «The Top 3 Google Ranking Factors That Really Matter», Search Engine Journal. Åpnet: 24. februar 2024. [Online]. Tilgjengelig på: <https://www.searchenginejournal.com/ranking-factors/top-ranking-factors/>
- [33] «Mobile App Download Statistics & Usage Statistics (2024)», BuildFire. Åpnet: 25. februar 2024. [Online]. Tilgjengelig på: <https://buildfire.com/app-statistics/>
- [34] «Mobile apps that have been used only once 2019», Statista. Åpnet: 25. februar 2024. [Online]. Tilgjengelig på: <https://www.statista.com/statistics/271628/percentage-of-apps-used-once-in-the-us/>
- [35] «JavaScript With Syntax For Types.» Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://www.typescriptlang.org/>
- [36] «React». Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://react.dev/>
- [37] «Stack Overflow Developer Survey 2023», Stack Overflow. Åpnet: 24. februar 2024. [Online]. Tilgjengelig på: https://survey.stackoverflow.co/2023/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2023
- [38] «Spring Boot», Spring Boot. Åpnet: 7. mai 2024. [Online]. Tilgjengelig på: <https://spring.io/projects/spring-boot>

- [39]. «NET | Build. Test. Deploy.», Microsoft. Åpnet: 7. mai 2024. [Online]. Tilgjengelig på: <https://dotnet.microsoft.com/en-us/>
- [40]. «Firebase | Google's Mobile and Web App Development Platform», Firebase. Åpnet: 2. mai 2024. [Online]. Tilgjengelig på: <https://firebase.google.com/>
- [41]. T. Kalyanasundaram, «Firebase and Beyond: The Evolving Landscape of Backend as a Service (BaaS)», Medium. Åpnet: 6. mars 2024. [Online]. Tilgjengelig på: <https://medium.com/@kalyanasundaramthivaharan/firebase-and-beyond-the-evolving-landscape-of-backend-as-a-service-baas-1af0b730e359>
- [42]. «Express - Node.js web application framework». Åpnet: 25. februar 2024. [Online]. Tilgjengelig på: <https://expressjs.com/>
- [43]. «What Is Express.js? Everything You Should Know», Kinsta®. Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://kinsta.com/knowledgebase/what-is-express-js/>
- [44]. «luds/04.md at luds · Inurl/luds», GitHub. Åpnet: 25. februar 2024. [Online]. Tilgjengelig på: <https://github.com/Inurl/luds/blob/luds/04.md>
- [45]. «Home | MetaMask developer documentation». Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://docs.metamask.io/>
- [46]. «Bitcoin Connect». Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://bitcoin-connect.com/>
- [47]. P. G. D. Group, «PostgreSQL», PostgreSQL. Åpnet: 16. april 2024. [Online]. Tilgjengelig på: <https://www.postgresql.org/>
- [48]. «MongoDB: The Developer Data Platform | MongoDB». Åpnet: 7. mai 2024. [Online]. Tilgjengelig på: <https://www.mongodb.com/>
- [49]. «Fast NoSQL Key-Value Database – Amazon DynamoDB – Amazon Web Services», Amazon Web Services, Inc. Åpnet: 7. mai 2024. [Online]. Tilgjengelig på: <https://aws.amazon.com/dynamodb/>
- [50]. «About Git», GitHub Docs. Åpnet: 6. mars 2024. [Online]. Tilgjengelig på: <https://docs.github.com/en/get-started/using-git/about-git>
- [51]. C. G. Cobb, *The project manager's guide to mastering agile: principles and practices for an adaptive approach*, 1st edition. Hoboken, New Jersey: Wiley, 2015.
- [52]. «Prinsippene bak Det smidige manifestet.» Åpnet: 5. mai 2024. [Online]. Tilgjengelig på: <https://agilemanifesto.org/iso/no/principles.html>
- [53]. «Unified Modeling Language», *Wikipedia*. 15. april 2024. Åpnet: 26. april 2024. [Online]. Tilgjengelig på: https://en.wikipedia.org/w/index.php?title=Unified_Modeling_Language&oldid=1219021783
- [54]. T. Weilkiens og B. Oestereich, *UML 2 Certification Guide: Fundamental and Intermediate Exams*. San Diego, UNITED STATES: Elsevier Science & Technology, 2006. Åpnet: 5. mars 2024. [Online]. Tilgjengelig på: <http://ebookcentral.proquest.com/lib/hogskbergen-ebooks/detail.action?docID=282119>
- [55]. D. Pilon og N. Pitman, *UML 2.0 in a Nutshell*. O'Reilly Media, Inc., 2005.
- [56]. «Quick Start – React». Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://react.dev/learn>
- [57]. S. Henneken, *Use of the SOLID principles with the IEC 61131-3: 5 Principles for Object-Oriented Software Design in the PLC Programming*. BoD – Books on Demand, 2023.
- [58]. «JSON vs XML - Difference Between Data Representations - AWS», Amazon Web Services, Inc. Åpnet: 16. april 2024. [Online]. Tilgjengelig på: <https://aws.amazon.com/compare/the-difference-between-json-xml/>
- [59]. «Nodemailer :: Nodemailer». Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://www.nodemailer.com/>

- [60] «Bech32 - Bitcoin Wiki». Åpnet: 4. april 2024. [Online]. Tilgjengelig på: <https://en.bitcoin.it/wiki/Bech32>
- [61] «Cloud Functions for Firebase», Firebase. Åpnet: 4. april 2024. [Online]. Tilgjengelig på: <https://firebase.google.com/docs/functions>
- [62] «Cloud Scheduler», Google Cloud. Åpnet: 9. mai 2024. [Online]. Tilgjengelig på: <https://cloud.google.com/scheduler>
- [63] «NoSQL Vs SQL Databases», MongoDB. Åpnet: 13. mai 2024. [Online]. Tilgjengelig på: <https://www.mongodb.com/resources/basics/databases/nosql-explained/nosql-vs-sql>
- [64] «Firestore Data Model: An Easy Guide | Hevo». Åpnet: 5. april 2024. [Online]. Tilgjengelig på: <https://hevodata.com/learn/firestore-data-model/>
- [65] «What is prop drilling in React?», DEV Community. Åpnet: 11. april 2024. [Online]. Tilgjengelig på: <https://dev.to/codeofrelevancy/what-is-prop-drilling-in-react-3kol>
- [66] «Context – React». Åpnet: 11. april 2024. [Online]. Tilgjengelig på: <https://legacy.reactjs.org/docs/context.html>
- [67] «react-secure-storage», npm. Åpnet: 29. april 2024. [Online]. Tilgjengelig på: <https://www.npmjs.com/package/react-secure-storage>
- [68] «Window: localStorage property - Web APIs | MDN». Åpnet: 12. april 2024. [Online]. Tilgjengelig på: <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>
- [69] «Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.» Åpnet: 12. mai 2024. [Online]. Tilgjengelig på: <https://tailwindcss.com/>
- [70] «CSA Research», CSA Research. Åpnet: 16. april 2024. [Online]. Tilgjengelig på: <https://csa-research.com/>
- [71] «Introduction | react-i18next documentation». Åpnet: 16. april 2024. [Online]. Tilgjengelig på: <https://react.i18next.com>
- [72] «Use JSDoc: Index». Åpnet: 2. mai 2024. [Online]. Tilgjengelig på: <https://jsdoc.app/>
- [73] «About READMEs», GitHub Docs. Åpnet: 3. april 2024. [Online]. Tilgjengelig på: <https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-readmes>
- [74] cPanel Community, «Git Version Control series: What is Git?», cPanel. Åpnet: 5. mai 2024. [Online]. Tilgjengelig på: <https://cpanel.net/blog/products/git-version-control-series-what-is-git/>
- [75] «Datatilsynet - personvern og informasjonssikkerhet», Datatilsynet. Åpnet: 10. april 2024. [Online]. Tilgjengelig på: <https://www.datatilsynet.no/>
- [76] «California Consumer Privacy Act (CCPA) | State of California - Department of Justice - Office of the Attorney General». Åpnet: 30. april 2024. [Online]. Tilgjengelig på: <https://oag.ca.gov/privacy/ccpa>
- [77] «ISO/IEC 27001», *Wikipedia*. 8. mars 2024. Åpnet: 11. april 2024. [Online]. Tilgjengelig på: https://en.wikipedia.org/w/index.php?title=ISO/IEC_27001&oldid=1212457641
- [78] «Firebase Security Rules», Firebase. Åpnet: 11. april 2024. [Online]. Tilgjengelig på: <https://firebase.google.com/docs/rules>
- [79] «Advanced Encryption Standard», *Wikipedia*. 25. mars 2024. Åpnet: 10. april 2024. [Online]. Tilgjengelig på: https://en.wikipedia.org/w/index.php?title=Advanced_Encryption_Standard&oldid=1215543815
- [80] S. Mudadla, «What is the use of .env», Medium. Åpnet: 16. april 2024. [Online]. Tilgjengelig på: <https://medium.com/@sujathamudadla1213/what-is-the-use-of-env-8d6b3eb94843>

- [81] «W3C», W3C. Åpnet: 10. april 2024. [Online]. Tilgjengelig på: <https://www.w3.org/>
- [82] «Web Content Accessibility Guidelines (WCAG) 2.1». Åpnet: 10. april 2024. [Online]. Tilgjengelig på: <https://www.w3.org/TR/WCAG21/>
- [83] L. Briand og Y. Labiche, «A UML-Based Approach to System Testing», *Softw. Syst. Model.*, bd. 1, nr. 1, s. 10–42, sep. 2002, doi: 10.1007/s10270-002-0004-8.
- [84] R. W. Miller og C. T. Collins, «Software Developer RoleModel Software, Inc. 342 Raleigh Street Holly Springs, NC 27540 USA +1 919 557 6352».
- [85] «Nærfeltskommunikasjon», *Wikipedia*. 18. juni 2023. Åpnet: 8. mai 2024. [Online]. Tilgjengelig på: <https://no.wikipedia.org/w/index.php?title=N%C3%A6rfeltskommunikasjon&oldid=23638607>
- [86] «luds/06.md at luds · Inurl/luds». Åpnet: 7. mai 2024. [Online]. Tilgjengelig på: <https://github.com/Inurl/luds/blob/luds/06.md>

VEDLEGG

Vedlegg 1. Brukertesting

Vedlegg 1.1. Instruksjoner til brukertest

Velkommen til brukertesting av applikasjonen “Library of Things”!

Prosjektet "Library Of Things" handler om å dele ressurser i lokale samfunn. Vi har utviklet en applikasjon hvor hvem som helst kan komme og låne ting fra et utlånsbibliotek helt kostnadsfritt.

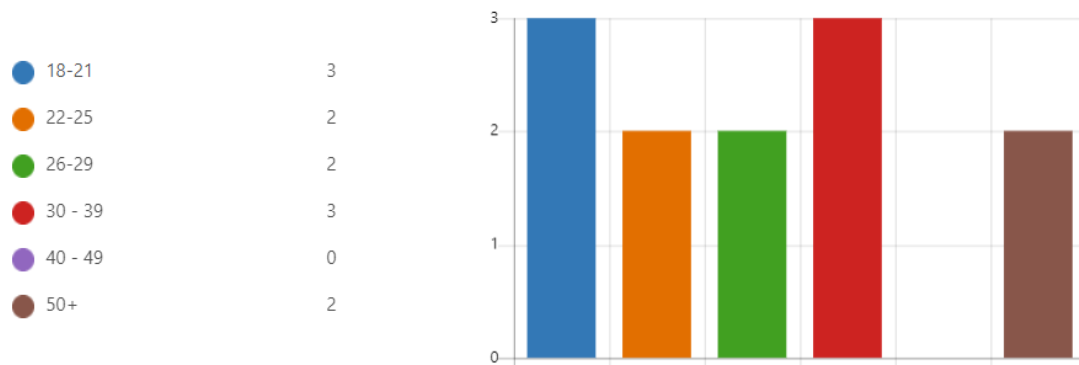
Nedenfor finner du en enkel guide med steg-for-steg instruksjoner for å hjelpe deg gjennom testen.

Instruksjoner:

1. Opprett en ny brukerprofil. Eposten dere oppgir skal være “test@gmail.com”. Velg “Bergen” som foretrukket lokasjon.
2. Trykk “start her” og deretter velg et ønsket språk før du går videre. (Vær oppmerksom på at ikke alle sidene er oversatt i applikasjonen)
3. Legg til en “Shovel” fra Bergen, Norge, i plukklisten.
4. Gå til plukklisten for å se tingen du la til.
5. Legg til ytterligere to valgfrie gjenstander i plukklisten din.
6. Slett “Shovel” fra plukklisten.
7. Deretter opprett en reservasjon fra plukklisten.
8. Gå til brukerprofilen din, og finn reservasjonen du nettopp opprettet.
9. Vis frem QR-koden som tilsvarer din reservasjon.
10. Når QR-koden er scannet av observatøren, gå til brukerprofilen og finn lånet ditt.
11. Vis frem QR-koden som tilsvarer ditt lån. Når QR-koden er scannet av observatøren gå til neste steg.
12. **Avslutt og slett brukerkonto:**
 - a. Gå tilbake til brukerprofilen din.
 - b. Finn alternativet for å slette brukeren og følg prosessen for å slette kontoen din.

Vedlegg 1.2. Spørreundersøkelse med svar

1. Alder?



2. Hvor gode datakunnskaper har du?

- God
- Svært god
- Svært god
- God
- Middels
- Svært god
- God
- God
- Dårlig
- God
- God
- Middels

3. Hvordan gikk det å finne frem på nettsiden? Hva gikk bra eller dårlig?

- Bra, lett å finne fram.
- Meny var lett tilgjengelig, som gjorde det lett å finne fram. Noen navn på elementer i listen hadde like navn som gjorde det litt forvirrende hva som ligger hvor.
- Ganske rett frem! Men å opprette konto med et krypto trading profil forsto jeg ikke poenget med.

- Litt vanskelig å kjønne innloggingen i starten, men ellers fint og intuitivt
- Slet litt med navigasjon
- Det gikk fint.
- Ofte litt utydelig hvor man skulle gå eller hva man skulle gjøre.
- Helt ok. Lett når man fikk instruksjoner. Manglet en enkel måte å komme tilbake til startsidene. Fant en vei etterhvert, men måtte gå via lånekurv. Burde vært mulig å komme dit med pilen øverst til venstre uavhengig av hvor man er (kanskje ved å trykke flere ganger, men å ende opp der til slutt).
- Bra, very easy to find things
- Det gikk bra, litt vanskelig å følge bruksanvisningen til tider, men fant fram etter relativt kort tid
- Det gikk fint
- Gikk greit, men kanskje litt mange operasjoner. Fint om alt var på norsk.

4. **Fikk du til å lage en reservasjon?**

● Ja	12
● Nei	0

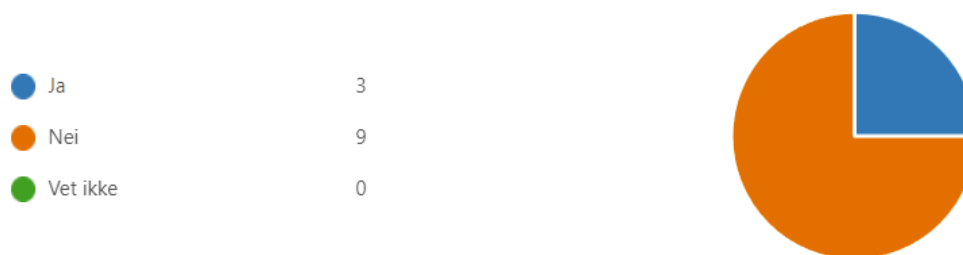


5. **Hvordan synes du det var å finne frem til gjenstandene og reservere disse?**

- Enkelt
- Lett å finne fram og reservere. Men det hadde kanskje vært lettere å bruke et filter på gjenstand-siden slik at man ikke må gå tilbake for å bytte gruppe.
- Helt grei, skulle gjerne sett at det var flere filtre, eventuelt en søkemotor.
- Fin
- Skulle ønske det var litt mer tydelig når man trykker på et gjenstand og reserverer at den blir faktisk reservert
- Lett

- Gikk greit å finne fram. Litt utydelig på når man hadde lagt en gjenstand inn i handlekurven og om det var reservert eller ikke.
- Enkelt nok, var ganske selvforklarende. Synes lånene burde vises med hva som er lånt som overskrift i stedet for LånID. Som bruker bryr jeg meg ikke om ID-en, men jeg ønsker å enkelt kunne se hva jeg har lånt i listen over reservasjoner/historikk uten å måtte klikke på hver reservasjon for å se detaljer.
- Litt trøbbel og kom ikke opp på bruker profil men utenom det var det fint
- Det var kjapt og enkelt, men reservasjonen dukket ikke opp når jeg trykket inn på reservasjoner fra brukerprofilen, måtte trykke direkte fra reservasjonen får å få opp qr koden
- Det var enkelt
- Det gikk greit

6. **Har du tidligere benyttet en lommebok for å selge eller kjøpe bitcoin?**

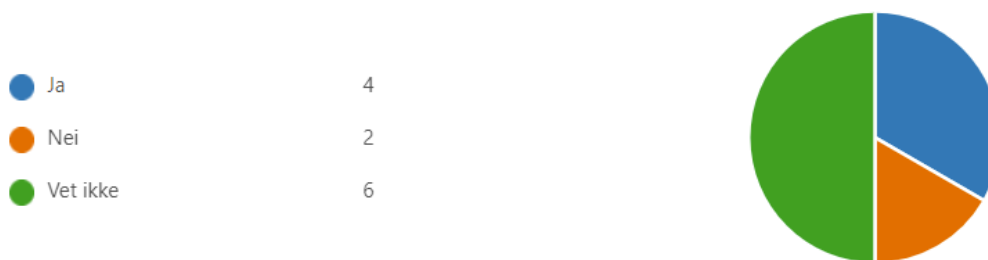


7. **Hva synes du om registrering og innloggingstjenesten(lightning wallet) i applikasjonen?**

- Enkelt
- Lett å finne fram og reservere. Men det hadde kanskje vært lettere å bruke et filter på gjenstand-siden slik at man ikke må gå tilbake for å bytte gruppe.
- Helt grei, skulle gjerne sett at det var flere filtre, eventuelt en søkemotor.
- Fin
- Skulle ønske det var litt mer tydelig når mann trykker på et gjenstand og reserverer at den blir faktisk reservert
- Lett
- Gikk greit å finne fram. Litt utydelig på når man hadde lagt en gjenstand inn i handlekurven og om det var reservert eller ikke.

- Enkelt nok, var ganske selvforklarende. Synes lånene burde vises med hva som er lånt som overskrift i stedet for LånID. Som bruker bryr jeg meg ikke om ID-en, men jeg ønsker å enkelt kunne se hva jeg har lånt i listen over reserverasjoner/historikk uten å måtte klikke på hver reserverasjon for å se detaljer.
- Litt trøbbel og kom ikke opp på bruker profil men utenom det var det fint
- Det var kjapt og enkelt, men reserverasjonen dukket ikke opp når jeg trykket inn på reserverasjoner fra brukerprofilen, måtte trykke direkte fra reserverasjonen får å få opp qr koden
- Det var enkelt
- Det gikk greit

8. **Kunne du ønske at flere tjenester tilbydde en lignende innloggingstjeneste?**

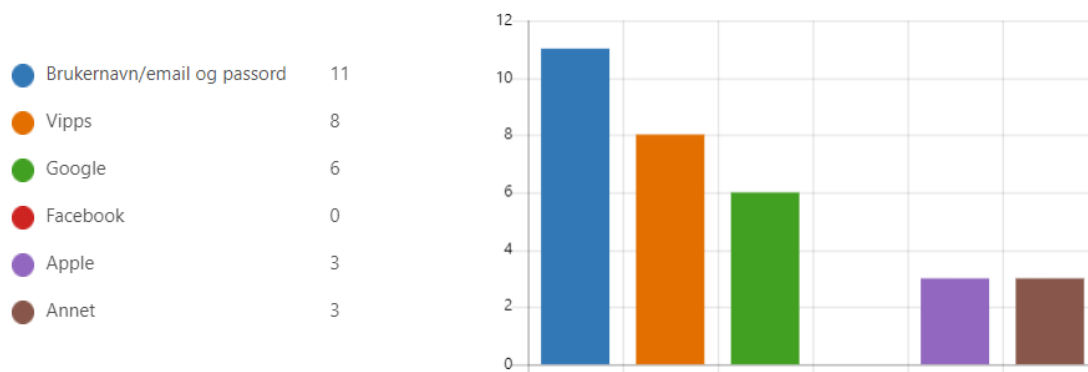


9. **Har du brukt noen tilsvarende innloggingstjenester før? Hvis ja, hvilke?**

- Vet ikke.
- Ikke lightning wallet tjenester men andre innlogginstjenester som Vipps, Google, Microsoft.
- Coinbase
- Minnet litt om digital bank id
- Nei
- Vipps tror jeg
- BANKID og Google brukes mye
- Vipps? Ble også tvunget til å bruke Authenticator av UiB, men syntes det var plagsomt. Vipps er ok siden jeg likevel har det, men for de som ikke allerede har en slik app er nok brukernavn/email og passord den beste løsningen.
- Nei
- Vipps var lignende vil jeg si, men ikke noen andre typer innlogginger som dette
- Nei

- Nei

10. **Hvilke av disse innloggingstjenestene foretrekker du? Du kan velge flere.**



11. **Hva er din opplevelse av innloggingstjenesten til Library of Things sammenlignet med andre tjenester du har brukt tidligere?**

- Merket ikke så veldig stor forskjell.
- Enkel å bruke, men ikke nødvendigvis bedre enn eksisterende tjenester. Har ikke prøvd å sette opp wallet, så vet ikke hvor mye jobb det er.
- Den er avhengig av du allerede har laget en bruker på det du skal logge inn gjennom, så her ville det vært naturlig å velge noe de fleste.
- Kjapp og enkel, veldig grei når en blir vant med den
- Nøytral
- Litt tungvindt med en egen app for å logge inn. Skjønner at man må verifisere hvem som er logga inn i en utlånstjeneste, men siden man kan være helt anonym forstår jeg ikke helt formålet likevel. Ellers veldig bra!
- finnes bedre alternativer
- Jeg likte den ikke. Syns det var tungvindt å måtte laste ned en egen app og forstod ikke helt poenget. Syns også at det ikke var tydelig forklart på nettsiden. Trodde nesten at innlogging med lightning wallet var en egen greie og ikke et krav. Hvis observatør ikke hadde sagt at det var et krav så ville jeg brukt mer tid på å finne ut om det fantes en alternativ innlogging på nettsiden, for det fristet ikke å måtte laste ned en egen app og lage en bruker i den. Nå husker jeg ikke helt om det var slik at jeg måtte logge inn før jeg kunne se utvalget av ting, men jeg tror det? Jeg ville absolutt foretrukket å kunne se på tingene før jeg logger inn. Når jeg kommer til en nettside som krever at jeg lager bruker før jeg kan se innholdet så dropper jeg den nettsiden med én gang. Med det ekstra hinderet at jeg må laste ned en ukjent app først, hadde jeg nok bare sagt "fuck dette, dette gidder jeg ikke".

- Eg syns det var bra med lite til ingen problemer
- Ganske lik i forhold til vipps, mye enklere og effektiv enn andre sosiale medier som facebook, instagram osv
- Ganske likt som andre tjenester
- Ryddig, oversiktlig og informativ

12. **Fikk du til å bytte språk i applikasjonen?**

● Ja	9
● Nei	3



13. **Hvordan vil du beskrive opplevelsen med å bytte språk?**

- Ganske enkelt
- Verken enkelt eller vanskelig
- Ganske enkelt
- Ganske enkelt
- Enkelt
- Enkelt
- Verken enkelt eller vanskelig
- Enkelt
- Vanskelig
- Verken enkelt eller vanskelig
- Enkelt
- Verken enkelt eller vanskelig

14. **Fikk du til å slette brukeren?**

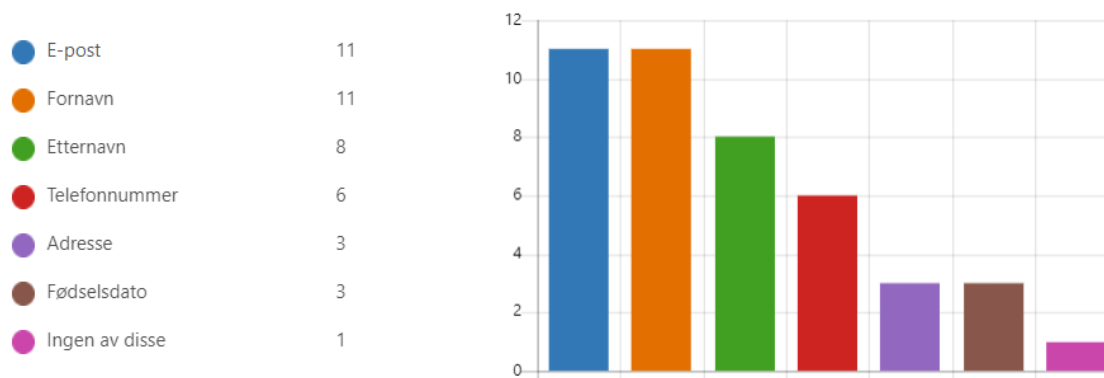
● Ja	12
● Nei	0
● Vet ikke	0



15. **Hvor viktig er personvern på internett for deg?**

- Ganske viktig
- Nøytral
- Svært viktig
- Ganske viktig
- Nøytral
- Ganske viktig
- Ganske viktig
- Nøytral
- Svært viktig
- Ganske viktig
- Svært viktig
- Ganske viktig

16. **Hvilken persondata er du komfortabel med å gi fra deg til en applikasjon? Du kan velge flere.**



17. Hva er ditt helhetlige inntrykk av applikasjonen?

- Bra
- Ser veldig bra ut, og ganske grei flyt i programmet.
- Virker veldig grei å bruke, men hvordan den vil fungere i praksis er jeg litt usikker på. Man er avhengig av å ha en utlåner i nærheten som har utstyret du trenger tilgjengelig for å bruke den. Jeg tror at mange heller vil dra 10 minutter til den lokale jernvarehandelen å kjøpe en spade for 200 kroner, enn å kjøre en halv time for å låne en spade, kjøre hjem, bruke den, så levere den igjen. Vinningen går litt opp i spinningen.
- Veldig nyttig og grei å bruke
- Fint uidesign, flott ide! Slet litt med navigasjon og flyten av appen, men de kan kanskje skyldes på at det var første gang jeg brukte appen.
- "(Svaret på nr 16 er veldig avhengig av hvilken applikasjon det er).
- Veldig bra! "
- God idé og mye som var bra. Var litt vanskelig å navigere gjennom siden og vite hvor man skulle gå til en hver tid. Brukervennligheten kan gjøres bedre / mer intuitivt
- Fungerer greit når man har kommet over innloggingshinderet. God måte å velge lokasjon på. Fint med bilder av tingene man kan låne. Lett å finne profilsiden. Relativt oversiktlig profilside. (Ville kanskje byttet rekkefølge på "Lånehistorikk" og "Brukerreservasjoner", og brukt et handlekurv-ikon til "Lånekurv"?) Fint med oversikt over hva som er reservert, hva som er under utlån og hva man har lånt før, med mulighet til å se detaljer. Lurt med QR-koder for lån og tilbakelevering. Språk var ikke så bra på norsk, manglet oversettelser og hadde noen særskrivingsfeil. Søt kanin.
- Bra. Enkelt å bruke. Fint plassert, veldig simpelt å enkelt å finne frem
- Gjennomtenkt og god applikasjon, utenom "bug" med reservasjonen og lånet.
- Enkel og grei

- Fin og oversiktlig

- 18. **Dersom du har noe mer du ønsker å tilføye om applikasjonen så kan du gjøre det her**

- Var litt tungvidt å finne tilbake til tingene man kunne låne etter man hadde loka rundt i profilen sin. Følte jeg trykte på tilbake mange ganger og bare gikk inn og ut av lån og reservasjonene som jeg hadde kikket på.

Ble spurt om hvor jeg ville låne ting rett etter jeg hadde sagt at jeg var i Bergen (hvis jeg husker rett), som følte litt dobbelt opp.

Og hadde vært fint om man kunne sjekke ut hva som finnes før man må registrere seg.

Vedlegg 2. Demonstrasjonsvideo og skjermbilder av applikasjon

2.1 Demonstrasjonsvideo


Demonstrasjonsvideoen er tilgjengelig her: <https://youtu.be/rBDAHbOPHjs>

2.2 Skjermbilder fra applikasjon

Her kommer noen flere skjermbilder fra applikasjonen.

Sider bruker har tilgang til:

English profile



UserID: 02967c

bachelor@hvl.no

- Active loans
- Pick List
- Reservations
- Loan history
- Settings
- Log out
- Admin

English profile

Active loans

Loan ID: QQDNXaamAPAILK5fu7NC

From: Mon May 13 2024 12:49
To: Mon May 27 2024 12:49

[Display loan details](#)

English profile

Loan details

LoanID: QQDNXaamAPAILK5fu7NC
From: Mon May 13 2024 12:49
To: Mon May 27 2024 12:49


Items:

Power jigsaw

QR-code



English profile



UserID:02967c

bachelor@hvl.no

- Update preferred location
- Delete account

When you delete your account you will lose all your borrowing history and the donation records.

English profile

Pick list

Weeder

Availability: Available
 Due date 14 days after checkout

[Add more](#)

[Reserve items!](#)

English profile

Reservation confirmed

You have reserved the following items:

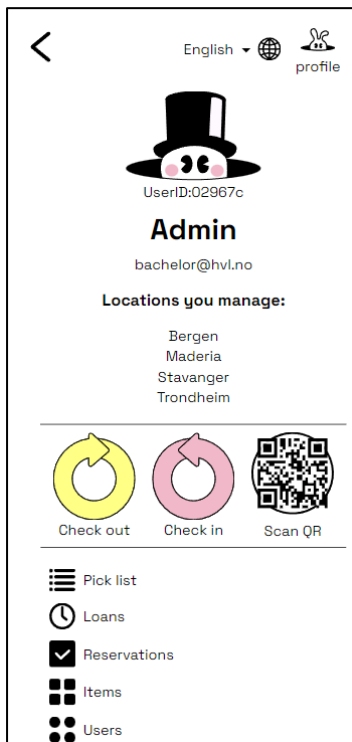
Weeder

At this location: Bergen
This reservation ends at 13.5.2024, 14:53:01


[Go to reservation](#)

[Go to your profile](#)

Administratorsider:






English profile







 UserID:02967c
Admin
 bachelor@hvl.no

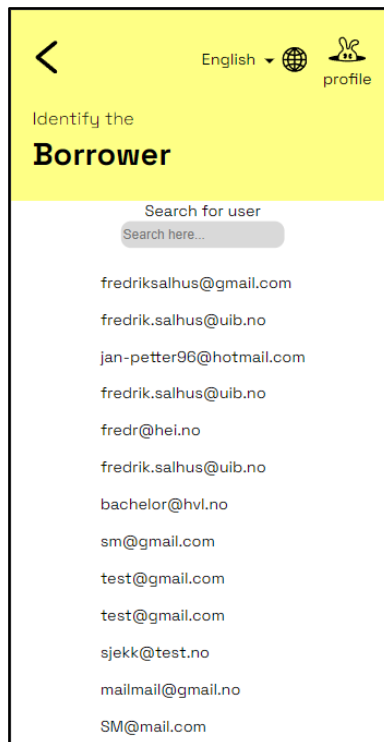
Locations you manage:

- Bergen
- Maderia
- Stavanger
- Trondheim

Check out Check in Scan QR

-  Pick list
-  Loans
-  Reservations
-  Items
-  Users

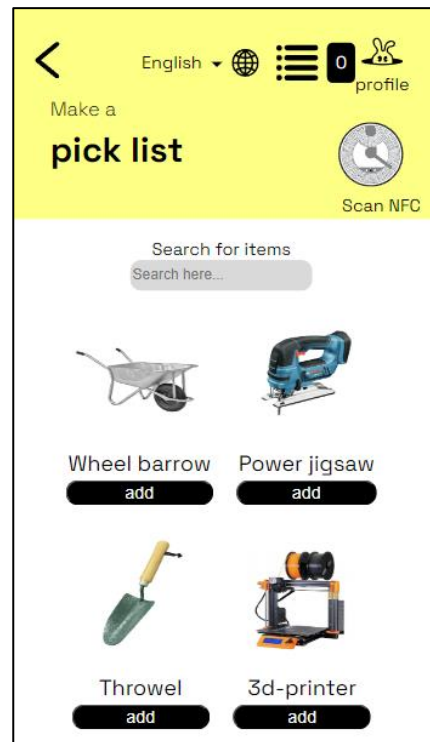


English profile


Identify the
Borrower

Search for user
 Search here...





- fredriksalhus@gmail.com
- fredrik.salhus@uib.no
- jan-petter96@hotmail.com
- fredrik.salhus@uib.no
- fredr@hei.no
- fredrik.salhus@uib.no
- bachelor@hvl.no
- sm@gmail.com
- test@gmail.com
- test@gmail.com
- sjekk@test.no
- mailmail@gmail.no
- SM@mail.com






English profile


Make a
pick list  Scan NFC

Search for items
 Search here...


-  **Wheel barrow**
add
-  **Power jigsaw**
add
-  **Throwel**
add
-  **3d-printer**
add



English   1  profile




Make a
pick list

 Scan NFC

User email: bachelor@hvl.no

3d-printer 

 Availability: Available
 Due date 12 days after check out

English   0  profile




Loans

Search for e-mail

ID: NDCCa7uzminn7z45NSMM
 User email: fredrik.salhus@uib.no
 Location: Bergen

ID: IN48NcxplNc3WrTII4E2
 User email: fredrik.salhus@uib.no
 Location: Bergen


ID: 3ABknJshgJDB8f2aelXF
 User email: fredrik@gmail.com
 Location: Bergen

English   0  profile


Administer Items

Edit items




Bergen




Item name: Wheel barrow Location:
3JslioPfrSoSxMu6g2xC




Item name: Power jigsaw Location:
3JslioPfrSoSxMu6g2xC

English   0  profile

Edit Wheel barrow 

Select category



Name


Description

Value of item





Duration of loan

Upload Image

Superadmin:



UserID:r9g2vL
Super admin
mailmail@gmail.no

-  Edit locations
-  Edit Categories
-  Assign admin rights
-  User profile

Manage locations

+ Add location

Find location
Search here...

Bergen

Maderia

Stavanger

Trondheim

Troms

Lisboa

Quebeck

Manage locations

Edit Maderia

Delete

Monday --:-- ☹️ ☹️

Tuesday --:-- ☹️ ☹️

Wednesday --:-- ☹️ ☹️

Thursday 08:00 ☹️ 21:00 ☹️

Friday --:-- ☹️ ☹️

Saturday --:-- ☹️ ☹️

Sunday --:-- ☹️ ☹️

Name

Address

Description

Submit

Administrare categories

Add category

Find category
Search here...

Garden tools

Baby gear

Electric tools

Tech gear

sport

Administrare categories

Edit Garden tools

Delete

Name

Submit

Tech gear

sport

Vedlegg 3. Dybdeintervju

Dybdeintervju er gjengitt i sin helhet med tillatelse fra oppdragsgiver.

Samsvarer disse initielle kravene (fra rapporten) med dine ønsker for applikasjonen?

Hvis ikke, hvilke krav mangler?

Når jeg ser de så har vi jo fått til at den er skalerbar, internasjonalisert og med mulighet for å endre språk osv. Ja disse intielle kravene har truffet med mine ønsker for applikasjonen.

Sammen ble vi enige om å forsøke å danne et MVP, hvordan følte du denne tilnærmingen gikk?

Jeg følte at det ble mer en prototype enn et MVP, og at det nesten har blitt til et ferdig produkt som fungerer helt, selv om vi ikke får testet det i skala og stresstestet det. Men jeg tenker at det er jobbet for at applikasjonen skal være en slags tynn klient.

Hvilke funksjonalitet(er) liker du best med applikasjonen?

Jeg er veldig glad i Superadmin funksjonen og muligheten for å skalere hele applikasjonen, og at man kan være flere roller samtidig. Liker også godt minimalismen i applikasjonen, det å ha et utgangspunkt og kunne gjøre mange kritiske funksjoner fra et sted. Veldig stolt av skann QR-kode flyten, jeg synes det løser veldig mye av interaksjonen mellom en bruker og en admin for da slipper man spørsmål og språkbarrierer og andre ting og gjør ting enklere.

Et av kravene i prosjektbeskrivelsen var å benytte logg inn med Lightning Wallet, hva synes du om den valgte løsningen for autentisering?

Jeg synes den er magisk sånn som den vises på siden. Det har vært et ønske fra min side å forenkle innlogging i lang tid. Det jo skummelt å bruke teknologi som er nisje fordi man er redd for at det ikke blir adoptert og at man blir sett ned på og at noen ikke har lyst å bruke det fordi de er usikre. Det er mange aspekter med dette, men en fundamental forståelse for at Lightning Wallet er en slags matematisk og ganske sikret og kryptert plattform som har disse nodene som verifiserer og sikrer nettverket. Dette gjør at det er den eneste virkelige smarte og lure måten å logge seg inn på med, ved at man autentiserer seg med Lightning Wallet og jeg er egentlig veldig fornøyd med denne løsningen.

Jeg synes det forebygger litt lite dokumentasjon på nettsiden

deres(<https://github.com/lnurl/luds/blob/luds/04.md>) og at det virker som at det er et nisje

prosjekt, det at man kan logge inn med LNURL. Det er ikke sånn som ligger ute på flere store nettsider som mange vet om, men jeg synes det fungerer særdeles bra. Jeg er veldig stolt og glad for denne løsningen.

Hva er ditt helhetlige inntrykk av applikasjonen? Føler du at applikasjonen har blitt slik du forventet ved prosjektets start?

Applikasjonen har blitt bedre tenker jeg. Jeg visste at jeg hadde tanker og ideer, men at dere ville også komme med innspill og deres tekniske kyndige kompetanse. Sånn at vi sammen har utviklet noe som fungerer bra med alle sin feedback og sin kompetanse som har gjort at det har blitt en god applikasjon. Den har blitt mye bedre enn jeg noensinne kunne forestilt meg, på alle måter. Jeg er mer enn fornøyd med arbeidet som gjort, det overskrider det jeg hadde forventet.

Hvilke funksjonaliteter savner du mest i applikasjonen slik den er nå?

Siden jeg har stått i Tingoteket så tror jeg at det hadde vært spennende med NFC-tagging, men det er ikke kritisk. Siden vi har gått fra å lage en MVP til en prototype så trenger vi ikke å lage et helt ferdig konsept, men det er en funksjon som er fancy og fin å bruke. Når det er sagt så har vi laget nye måter å håndtere inn- og utlånsprosessen på, som gjør det mye enklere siden man slipper å skanne fysiske ting. Slik kan man gjøre raskere klikk, men jeg vil tro at NFC-teknologi har et bruksområde i denne applikasjonen.

Hva synes du om flyten som administrator?

Den er veldig bra, den fungerer helt topp. Når vi får på plass de siste knappene på plass så tenker jeg at det blir helt optimalt.

Hva synes du om flyten som en superadministrator?

Den er også veldig god, så lenge jeg kan skifte rolle fra superadministrator til administrator og at det er tydelig så er det veldig greit.

Hva synes du om flyten som bruker?

Flyten som bruker er noe jeg har tenkt mye på. Jeg vil at folk skal bruke minst mulig tid på mobilene, dette skal bare være et verktøy for å kunne få oversikt over hva som er tilgjengelig i nærsamfunnet, hva man kan låne, lage egne plukklistor og å kunne lage reservasjoner slik at man kan hente det. Sett ut ifra det så gjør appen akkurat det. Sånn sett så svarer den helt 100% konsist på oppgaven. Jeg synes bildene og utlistingene av tingene er

gode, beskrivelsene er gode og det er lett å navigere og å lage en plukk liste. Jeg føler som bruker at det er rom for å gjøre feil, man blir ikke redd for å trykke på noe feil fordi man kan ikke opprette flere lån, masse reserverasjoner, det skjer ikke noe kluss. Sånn sett så er det en app som ikke tuller seg til, det er også veldig positivt.

Hvor viktig var det for deg at applikasjonens design var tilpasset desktop i tillegg til mobile enheter?

Ikke i det hele tatt faktisk, for min del er det kun tenkt på som en mobilapplikasjon. Jeg har null fokus på desktop, web app for bare mobil var utgangspunktet hele veien.

Synes du applikasjonen ligner det designet som du har laget i Figma?

Ja, det ser veldig likt ut og med et par små justeringer så blir det helt perfekt.

I hvilken grad syntes du at gruppen har tilpasset applikasjonen basert på tilbakemeldingene under arbeidsmøtene?

Jeg syns dere har gjort en forenkende, veldig kalkulert og faglig tilnærmet god jobb med å gjøre det på den måten som er smartest og best fra deres side. Dere har vært flinke med å kommunisere det overfor meg også, det er jeg veldig glad for. Det har også vært rom for å diskutere ting i hvert møte for ting som har vært usikre og ting som måtte revurderes. Jeg håper at dere har opplevd at jeg har vært fleksibel når dere har hatt mer kompetanse og visst hva som fungerer best.

Samarbeid

Hva er din tidligere erfaring innen web design?

Jeg tok webdesign kurs i 2000, deretter gikk jeg på Noroff med fokus på multimedia design. Også har jeg jobbet i 20 år med interaksjonsdesign, webdesign og applikasjonsdesign deriblant i kinoindustrien og leketøysindustrien. Jeg har vært interessert i å lage egne apper og applikasjoner, dette er noe som har fulgt meg de siste 25 årene. Jeg kjenner på at erfaringer fra dette gjør at jeg ønsker å optimalisere for et enkelt brukergrensesnitt som er grafisk tilpasset, men som er egnet for at det er rom for at utviklere kan komme med innspill for at applikasjonen skal bli bedre. Tydelighet har vært noe jeg ser i designet mitt, det er grådig viktig for meg at jeg ikke lager unødvendig abstrahering eller usikkerhet rundt navigasjon, ikoner og grafikk.

Hvordan har det vært å jobbe med en prosjektgruppe fra Dataingeniør?

Det har vært en sann glede, dere har stilt på alle møtene og aldri vært forsinket. Ingen problemer på noen måte og dere har jobbet jevnt og trutt. Dere har kommet med gode løsninger og vært veldig flinke på å kommunisere de overfor meg. Det har vært et kjempefint samarbeid, det setter jeg veldig pris på. Jeg prøvde å ha ingen forventninger fordi folk er veldig forskjellige, men jeg syns det har gått kjempefint.

Hvordan synes du arbeidsmøtene har vært?

De har vært veldig effektive, vi har fått gjort veldig mye på kort tid. Tatt mange gode og vanskelige avgjørelser på kort tid og vært flinke på å planlegge framover.

Føler du at vi kunne fått til mer i løpet av denne tidsperioden?

Absolutt ikke, jeg føler vi har tynt strikken maks i forhold til hva som har vært mulig. Når jeg nevnte dette prosjektet for en student som gikk på HVL for to år siden og som jobbet med meg på Tingotek-prosjektet, så sa han “Oi Fredrik, det var veldig ambisiøst. Lykke til!”. Han visste at dere som studenter tar den utfordringer hvis jeg får med noen. For meg var det kjempeviktig å få med noen her (HVL) fra. Jeg var så glad når dere tok kontakt for det var på ingen måte gitt. Tidligere har jeg vært utelatt fra pitcheprosessen uten at de har visst om det. Så de har hatt pitcheprosessen uten at de har husket å si ifra til meg, så jeg var livredd for at jeg skulle gå glipp av det. Når jeg lagte video og når jeg visste at jeg skulle presentere det så prøvde jeg å legge litt tid og krefter inn i det slik at det kunne være spisbart og overkommelig. Jeg var utrolig glad for at dere valgte å jobbe med dette prosjektet med meg.

Hva tenker du veien er videre for applikasjonen?

Jeg tenker at nå når vi ferdigstiller så er den blitt til en brukbar app hvis vi klarer å utplassere applikasjonen. Jeg tenker at den er bygget såpass åpen og med et rammeverk som gjør at dersom det ikke er dere som fortsetter så kan jeg få kyndig hjelp utenfra til å gjøre nye ting i applikasjonen. Det hadde vært superspennende hvis jeg kunne reist til steder der de bruker bitcoin som en del av lokal samfunnet for eksempel Madeira, Fillipinene, Tyrkia eller andre steder. Der det er nisjer av lokalsamfunn som bruker dette og at vi kunne opprettet noen små LOT og begynt å låne ut ting, det hadde vært superspennende. Jeg er fremdeles av den oppfatning at alle nabolag trenger et Tingotek, et sted hvor de kan låne ting og dette er på en måte et initiativ for å tilrettelegge for dette, på en sikker og god måte. Jeg vet at det er et veldig idealistisk prosjekt, jeg hadde nesten ikke trodd at jeg kunne være så idealistisk selv. Det bikker nesten over på sosialisme og kanskje til og med kommunisme, at man skal låne ting og “be happy”, og at man ikke skal eie noe, men det er ikke det det handler om i det hele tatt. Jeg har satt meg ned tusen ganger og tenkt “hva er det jeg egentlig holder på”, det fundamentale tror jeg er at vi har masse ting som vi ikke trenger hele tiden. Det å gi fra oss noe er helt greit når vi ikke bruker de hele tiden og det å låne tingene når man trenger tingene er en fin måte å dele med lokalsamfunnet på. Jeg tenker at det er en helt naturlig utnyttelse som sparer oss for penger og ressurser. Det er fint med tilgang til ting som vi ikke nødvendigvis trenger å kjøpe. Det er et flott prosjekt og de fleste er veldig entusiastiske så det er veldig hyggelig og kjekt når folk gir slike tilbakemeldinger. Det er vanskelig å ikke like dette prosjektet fordi det er så mye godhjertet ting i det. Jeg tror det ligger noe veldig positivt i det.

Vedlegg 4. Kravdokument

Inkludert som eksternt vedlegg ved innlevering av prosjektet.

Vedlegg 5. Prosjekthåndbok

Inkludert som eksternt vedlegg ved innlevering av prosjektet.

Vedlegg 6. Systemdokumentasjon

Inkludert som eksternt vedlegg ved innlevering av prosjektet.

Vedlegg 7. Visjonsdokument

Inkludert som eksternt vedlegg ved innlevering av prosjektet.