

William Andersson  
Øivind Haugerøy Larsson  
Jonathan Midttun  
Bjørnar Østtveit

## Loyalty — Fordelskort i en digital verden

Bacheloroppgave i Dataingeniør

Veileder: Majid Rouhani

Mai 2020



William Andersson  
Øivind Haugerøy Larsson  
Jonathan Midttun  
Bjørnar Østtveit

## **Loyalty — Fordelskort i en digital verden**

Bacheloroppgave i Dataingeniør  
Veileder: Majid Rouhani  
Mai 2020

Norges teknisk-naturvitenskapelige universitet  
Fakultet for informasjonsteknologi og elektroteknikk  
Institutt for datateknologi og informatikk



Kunnskap for en bedre verden





## Forord

Denne oppgaven ble gjennomført som en bacheloroppgave, som del av studieplanen for 3. års dataingeniører, og fremstilt av RealNett AS i samarbeid med Institutt for Datateknologi og Informatikk ved NTNU. Oppgaven kom fram som en av oppgavene instituttet tilbød til studentene sine, og ikke som resultat av samarbeid med bedrift og studenter.

Bacheloroppgaven har gitt teamet mange gode erfaringer i løpet av arbeidsperioden. Vi har først og fremst fått innblikk i hvordan det er å jobbe med reelle problemstillinger, og utvikling i samarbeid med en oppdragsgiver. Oppgaven har også gitt oss mulighet til å utfordre oss selv med nye teknologier og videre prøvd ut evnene våre til å arbeide i team.

Vi ønsker å takke de ansatte ved RealNett AS som vi har jobbet med, Robert Estenstad og Christian Fjeseth, som har gitt oss muligheten til å jobbe i deres kontor på Tiller, med sponset lunsj og god stemning i arbeidstiden. De har også vært spesielt lette å kommunisere med gjennom prosjektet, som vi setter stor pris på. Selv om oppgaven ikke var førstevalg originalt, ser vi på oss som heldig som endte opp med denne oppgaven! Videre vil vi takke Majid Rouhani, veilederen vår fra IDI, som har gitt oss god og konkret tilbakemelding når vi har spurt om hjelp. Til slutt ønsker vi å takke Hira og Charlie fra Google Support som har vært veldig hjelpsomme når det har oppstått problemer og spørsmål med Google Pay.

William Andersson

Øivind H. Larsson

Jonathan Midttun

Bjørnar Østtveit



---

*Signatur*

---

*Signatur*

---

*Signatur*

---

*Signatur*

---

20.05.2020

---

20.05.2020

---

20.05.2020

---

20.05.2020

---

Dato

---

Dato

---

Dato

---

Dato

*Trondheim, Norge*

# Oppgavetekst

## Oppdragsgiver

Vår oppdragsgiver er RealNett AS representert av Robert Estenstad og Christian A. Fjeseth.

## Hensiktet med oppgaven

Hensikten med oppgaven er å utvikle et kommersielt fordelskort (ikke en app) som skal lanseres i hele Norge. Kortet vil gi innehaverne utvalgte fordeler hos utsteder (vår kunde) og kan bestå av digitale stempel, betalingskort for festivaler eller VIP-kort. Det digitale fordelskortet skal installeres på smarttelefoner og skal være basert på digital lommebok. Fordelskortets registreringsprosess samler inn kontaktopplysninger som senere, om sluttbruker godkjenner slik bruk, kan benyttes til utsendelse av eksklusive tilbud og rabatter, og nyhetsbrev eller annen relevant informasjon. For iOS benyttes Wallet - for Android benyttes tredje-parts elektronisk lommebok tilsvarende iOS' Wallet. Det er et poeng at det ikke er behov for å utvikle en egen app for løsningen. Markedsundersøkelser viser at produktet passer godt til bedrifter innen salg av tjenester/produkter som har høy omløpshastighet, herunder f.eks kafe og restaurant, nattklubb, idrettsarrangementer, festivaler etc. Muligheten til å utstede fordelskort til sluttbruker, samt ett eller flere designtilpassete fordelsprogram skal selges til bedrifter i SMB-markedet og vil være et supplement til bedriftenes øvrige markedsaktiviteter. Fordelsprogrammet vil gi bedriftene en unik mulighet til å kartlegge og kommunisere med sine faktiske kunder.

Merk: Dette er den originale oppgaveteksten, som vedlagt i vedlegg A

## Sammendrag

Denne bacheloroppgaven går ut på å utvikle et produkt for en oppdragsgiver, RealNett AS. Produktet, ved navn Loyalty/LoyaltyCard, bidrar til digitaliseringen av fordelsprogrammer, og gruppen har derfor valgt å undersøke følgende problemstilling: **Kan digitalisering av fordelskort øke nytteverdien kortene gir for små og mellomstore bedrifter og forbrukere?** Videre definerer gruppen nytteverdi som forbedring i tilgjengelighet, oversikt og logistikk.

Vi vil gå gjennom hva digitalisering er, hvordan man kan lykkes med digitalisering og hvorfor Loyalty som produkt er noe som er verdt å digitalisere. Gruppen hadde et ønske om å jobbe smidig, og falt til slutt på Lean utvikling med Kanban som utviklingsmetode. Gruppen har gjennomgående valgt ukjente og relevante teknologier for utviklingsprosessen, slik at vi får nye erfaringer å ta med oss videre. Det var viktig for oppdragsgiver at Loyalty ikke skulle være en enkeltstående app, og vi bruker dermed eksisterende elektroniske lommebokløsninger i form av Google Pay og Apple Wallet. Senere i rapporten diskuterer teamet fordeler og ulemper ved digitalisering av fordelsprogrammer, med problemstillingen i tankene.

Vi har endt opp med et produkt som vi synes løser digitalisering av fordelskort på en god måte, hvor utstedere har muligheten til å hente statistikk de tidligere ikke har hatt tilgang på ved et analogt fordelsprogram, og sluttbrukere får fordelskort som er lettere tilgjengelig enn fysiske.

## Abstract

This bachelors assignment is concerned with developing a product for a project owner, RealNett AS. The product, named Loyalty/LoyaltyCard, contributes to the digitalization of loyalty programmes, and therefore the team has chosen to indulge in the following research question: **Could the digitalization of loyalty cards increase the value of cards for small and medium sized businesses, and consumers?** Value is further defined as an improvement to availability, access to information and logistics.

We will explain what digitalization is, how to succeed with digitalization, and why there is value in digitalizing Loyalty as a product. The group wanted to use an agile working process, and Lean development with Kanban became the process of choice. The group has generally chosen unknown and relevant technologies for the development process, and have gained a lot of new experiences as a result. It was important for the product owner that the finished Loyalty product used existing electronic wallet options such as Google Pay and Apple Wallet, instead of being an isolated application. We will discuss the pros and cons of the digitalization of loyalty programmes, using the research question at hand, later in the report.

In the end, we have developed a system which we believe produce digitalized loyalty cards in a convenient manner, where issuers have the opportunity to collect detailed statistics, and end users can enjoy a greater ease of access, compared to analog cards.

# Innholdsfortegnelse

<b>1</b>	<b>Innledning og relevans</b>	<b>1</b>
1.1	Resultatmål . . . . .	1
1.2	Effekt mål . . . . .	1
1.3	Prosessmål . . . . .	1
1.4	Problemstilling . . . . .	2
1.5	Målgruppe . . . . .	2
1.6	Rapportens struktur . . . . .	3
<b>2</b>	<b>Teori</b>	<b>3</b>
2.1	Bakgrunn . . . . .	4
2.1.1	Lojalitetsprogrammer . . . . .	4
2.1.2	Hvorfor/hvordan lykkes lojalitetsprogram? . . . . .	5
2.1.3	Stempelkort som lojalitetsprogram . . . . .	6
2.1.4	RealNetts kilder . . . . .	8
2.2	Digitalisering . . . . .	8
2.2.1	Hva er digitalisering? . . . . .	8
2.2.2	Digitaliseringsprosessen . . . . .	9
2.2.3	Fordelene ved digitalisering . . . . .	11
2.2.4	Loyalty som digitaliseringsprosjekt . . . . .	13
2.3	Metodikk . . . . .	13
2.3.1	Smidig utvikling og Agile Manifesto . . . . .	13
2.4	Teknologier . . . . .	14
2.4.1	Versjonskontroll . . . . .	14
2.4.2	Kontinuerlig integrasjon . . . . .	14
2.4.3	Skyløsninger . . . . .	15
<b>3</b>	<b>Valg av teknologi og metode</b>	<b>15</b>
3.1	Systemutvikling . . . . .	15
3.1.1	Node.js med Express . . . . .	15
3.1.2	Vue.js . . . . .	15
3.1.3	React Native . . . . .	16
3.1.4	MySQL . . . . .	16
3.1.5	GitHub . . . . .	16
3.1.6	Jest . . . . .	16
3.1.7	AWS Elastic Beanstalk . . . . .	16
3.1.8	AWS CodePipeline . . . . .	16
3.1.9	Swagger . . . . .	17
3.2	Elektronisk lommebok . . . . .	17
3.2.1	Google Pay . . . . .	17
3.2.2	Apple Wallet . . . . .	17
3.3	Utviklingsmetode . . . . .	17
3.3.1	Lean systemutvikling med kanban . . . . .	17
3.3.2	Trello . . . . .	17
3.4	Arbeids- og rollefordeling . . . . .	18
<b>4</b>	<b>Resultater</b>	<b>19</b>

4.1	Vitenskapelige resultater . . . . .	19
4.1.1	Fordelsprogramtyper . . . . .	19
4.2	Ingeniørfaglige resultater . . . . .	27
4.2.1	Resultatmål . . . . .	27
4.2.2	Effekt mål . . . . .	28
4.2.3	Prosessmål . . . . .	28
4.2.4	Tester . . . . .	29
4.3	Administrative resultater . . . . .	29
4.3.1	Timeregnskap . . . . .	29
4.3.2	Fremdriftsplan . . . . .	29
4.3.3	Utviklingsprosess . . . . .	30
<b>5</b>	<b>Diskusjon</b>	<b>30</b>
5.1	Drøfting av vitenskapelige resultater . . . . .	30
5.2	Drøfting av ingeniørfaglige resultater . . . . .	31
5.2.1	Resultatmål . . . . .	31
5.2.2	Effekt mål . . . . .	32
5.2.3	Prosessmål . . . . .	32
5.3	Drøfting av administrative resultater . . . . .	32
5.3.1	Timeregnskap . . . . .	32
5.3.2	Fremdriftsplan . . . . .	33
5.3.3	Utviklingsprosess . . . . .	34
5.4	Gruppearbeidet . . . . .	35
5.4.1	Kommunikasjon . . . . .	35
5.5	Etiske problemstillinger . . . . .	36
5.5.1	Personvern . . . . .	36
5.5.2	Miljømessige konsekvenser . . . . .	36
5.5.3	Samfunnsmessige konsekvenser . . . . .	36
<b>6</b>	<b>Konklusjon og videre arbeid</b>	<b>36</b>
6.1	Konklusjon . . . . .	36
6.2	Videre arbeid . . . . .	37
	<b>Referanser</b>	<b>39</b>
<b>7</b>	<b>Vedlegg</b>	<b>41</b>
A	Den opprinnelige oppgaveteksten . . . . .	41
B	Kontrakt . . . . .	44
C	Visjonsdokument . . . . .	46
D	Kravdokumentasjon . . . . .	59
E	Systemdokumentasjon . . . . .	72

## Figurer

1	Grunnleggende formelen for beregning av CLV for SaaS-bedrifter ( <i>Loyalty programs to maximize lifetime value</i> 2020) . . . . .	5
2	Grafen viser tiden det tok rotter å komme seg fra en seksjon på banen til den neste. Start er 0 og belønningen er etter 5 (Panagiotidi 2016) . . . . .	7
3	Effekten av digitalisering (Parviainen mfl. 2017) . . . . .	12
4	Skjerm bilde fra Trello . . . . .	18
5	Programinformasjon . . . . .	20
6	Kortdesign . . . . .	20
7	Forhåndsvisning av Google og Apple pass . . . . .	21
8	Registreringsdesigner med forhåndsvisning . . . . .	22
9	Forhåndsvisning av ferdigdesignet produkt . . . . .	22
10	Kundens statistikkside . . . . .	23
11	Mobilapplikasjonen ved scanning av en UUID . . . . .	24
12	Ansatt siden til mobilapplikasjonen . . . . .	25
13	Registreringssiden på mobil . . . . .	26
14	Registreringssiden etter brukeren er registrert . . . . .	26
15	Google og Apple pass slik de ser ut i Google Pay og Apple Wallet . . . . .	27

## Tabeller

1	Dekningsgrad av user stories . . . . .	28
2	Antall timer per person ved endt prosjekt . . . . .	29
3	Planlagte timer mot faktiske timer per aktivitet . . . . .	29
4	Planlagt avslutningsdato mot faktisk avslutningsdato per aktivitet . . . . .	30



## Akronymer og forkortelser

- **SMB:** Små og mellomstore bedrifter. Definisjon: Små bedrifter har 1-20 ansatte og mellomstore bedrifter har 21-100 ansatte (*Fakta om små og mellomstore bedrifter 2020*)
- **IDI:** Institutt for datateknologi og informatikk
- **AWS:** Amazon Web Services, tilbyr digital infrastruktur via skyløsninger.

## Begrepsavklaringer

- **Veileder:** Majid Rouhani
- **Utviklingsteam:** William Andersson, Øivind Larsson, Jonathan Midttun og Bjørnar Østtveit.
- **Oppdragsgiver:** RealNett AS, representert ved Robert Estenstad og Christian A. Fjeseth.
- **Brukere av systemet:** Definert i visjonsdokumentet (Vedlegg C, kapittel 3.2). Som en oppsummering har vi fire hovedbrukere:
  - **Superadmin** har full kontroll over systemet, og representeres av oppdragsgiver.
  - **Administrator/admin:** Representeres av markedsfirmaer. Disse ser verdien i produktet og selger det videre til kunder.
  - **Kunde:** Virksomhetene som vil ta i bruk Loyalty-kort i deres daglige drift. Eksempler på dette kan være Studio 26 og Dromedar Kaffebar.
  - **Sluttbruker/Forbruker:** Brukerne av Loyalty-kortene som registrerer seg og installerer de i sin digitale lommebok.

# 1 Innledning og relevans

RealNett AS er et trøndersk firma som tilbyr tele- og datakommunikasjon i form av fiber/bredbånd med tilhørende wifi- og nettverkssystem (*Realnett* 2020). Målgruppen deres er i hovedsak bedrifter i SMB-markedet, men de tilbyr også tjenester til private. RealNett har identifisert digitale fordelskort som et mulig, fremtidig supplement til deres nåværende tjenester.

Hovedpoenget med fordelskortet er å unngå en frittstående mobilapplikasjon, men heller utvikle et installerbart kort for den mobile enhetens digitale lommebok. Dermed unngår vi at sluttbruker trenger å laste ned en applikasjon, gitt de allerede bruker en elektronisk lommebokløsning. Avgjørelsen for dette baserer seg på RealNetts erfaring med eksisterende fordelskort-applikasjoner og relevante markedsundersøkelser i USA. De ser det som hensiktsmessig å samle fordelskortene i den digitale lommeboken istedenfor å ha en applikasjon for hvert fordelskort.

## 1.1 Resultatmål

Vårt system skal fungere i nettleser og ha utvidelser til mobilapplikasjon for kunder. Etter endt prosjektperiode skal utviklingsteamet ha oppnådd disse resultatmålene:

- Digitalisert prosessen for fordelskort ved å installere dem i en digital lommebok.
- Tilby produkteier, markedsfirmaer og kunder muligheten til å innhente relevant statistikk og kvalitetsdata for videre utvikling og analyser.
- Kartlagt systemets muligheter for videreutvikling i henhold til produkteiers ønsker.

## 1.2 Effektmål

RealNett AS ønsker å digitalisere fordelsprogrammer. De har i hovedsak identifisert et behov for digitalisering gjennom egne erfaringer med eksisterende løsninger, og i tillegg begrunnet dette med flere relevante markedsundersøkelser og artikler. Som en effekt av digitaliseringen er det ønskelig å se en økning i fullførte fordelskort, dvs. fordelskort hvor gevinsten blir innkassert. Produktet skal også utnyttes kommersielt av produkteier ved å selge det videre til markedsfirmaer. Med dette prosjektet utvider RealNett AS horisontene sine utover tele- og datakommunikasjon, som også anses for å være et mål for prosjektet.

- Utvide forretningsmessig horisont.
- Øke bruken av fordelskort hos små og mellomstore bedrifter.

## 1.3 Prosessmål

Prosessmålene kan deles inn i to deler. En for produkteiers mål og en for teamets mål. Disse målene vil ikke sammenfalle da utviklingsteamet har andre forutsetninger for prosessen enn produkteier.

Loyalty-prosjektet er RealNetts første bachelorprosjekt ved NTNU. Deres prosessmål blir å bygge kompetanse rundt interaksjonen mellom bachelorstudenter og bedriften. Dette er kompetanse som kan legge grunnlag for videre samarbeid med andre bachelorgrupper på IDI.

- Opparbeide erfaring med bachelorprosessen og bachelorstudenter ved IDI.

For utviklingsteamet står kompetansebygging sterkt i prosessmålene. Vårt studie har et høyt fokus på prosjekter, samarbeid og utviklingsprosesser. Gjennom tre år har teamet opparbeidet seg en grunnleggende kompetanse innenfor temaene. Bachelorprosjektet bygger videre på dette ved å inkludere en reell

oppdragsgiver i utviklingsprosessen. Dette gir utviklingsteamet verdifull erfaring i samspillet mellom arbeidsgiver og utvikler.

- Opparbeide verdifulle erfaringer med teknologier, samspill mellom arbeidsgiver og utvikler, team-samarbeid og utviklingsprosesser.

## 1.4 Problemstilling

### Formulering av problemet

Opgaven gikk ut på å digitalisere en allerede eksisterende teknologi i form av fordelskort. Vi ønsker å se på data og statistikk, og få førstehåndserfaring på hvorvidt denne digitaliseringen kan bidra til å løse problemer fysiske fordelskort har, samtidig som det kan gi åpning for ny nytteverdi.

Et av de største problemene for sluttbrukere er tilgjengelighet. Forbrukerene må oppbevare kort fysisk i en lommebok eller liknende. Dette skalerer dårlig, da fysisk plass er en faktor, i tillegg må man alltid huske å ha med alle kortene man potensielt trenger. Mister man kortet risikerer man å miste fordeler, eller hvertfall fremgangen man har oppnådd på kortet.

For bedrifter som er utsteder av fordelskortene er kortene essensielt reklame, og et tiltak for å øke forbruk. Dette er selvfølgelig ikke et problem, men digitalisering av kortene kan bidra til at bedrifter ser nye nytteverdier av å iverksette et fordelsprogram. Videre er logistikken bak designprosessen og distribusjonen av fordelskortene barrierer for bruk, både hos kunde og sluttbruker

### Problemstilling

Til slutt falt problemstillingen på følgende:

### **Kan digitalisering av fordelskort øke nytteverdien kortene gir for små og mellomstore bedrifter og forbrukere?**

Når vi snakker om å øke nytteverdi er det tre sentrale tema vi tror digitaliseringen av fordelskort kan spesielt bidra med.

- Tilgjengelighet - Hvorvidt fordelskortet kan brukes eller ei i en gitt situasjon.
- Oversikt - Generell oversikt over bruken av fordelskortet, statistikk. Eksempelvis hvilken aldersgruppe som er mest sannsynlig å bruke kortene.
- Enklere logistikk - Designprosessen og distribusjon.

## 1.5 Målgruppe

Vårt produkt er delt opp i flere deler. Vi skal ha en adminkonsoll som gir tilgang til de administrative funksjonene for systemets brukere. Målgruppen for denne delen av applikasjonen vil da bli markedsfirmaer og RealNetts superadmin. I tillegg vil kunder ha tilgang til en liten del av dette systemet hvor de kan se statistikk og sende inn forespørsler om Loyalty-kort.

Applikasjonen vi skal lage vil være tilgjengelig for de ansatte hos systemets kunder. Den vil inneholde funksjoner for å scanne Loyalty-kort. Denne applikasjonen vil leveres på en mobil enhet fra markedsfirmaene eller RealNett. Denne mobile enheten vil kun inneholde vår applikasjon og eventuelle andre nødvendige funksjoner. Ved å begrense funksjonene på enheten senker vi terskelen for kunnskapsnivået hos de ansatte. Visjonen er at applikasjonen skal være så enkel at alle ansatte kan benytte seg av den uten opplæring.

Siste delen av systemet vil være en registreringsside for Loyalty-kortene. Her vil sluttbrukerne sende inn nødvendig informasjon i bytte med Loyalty-kort hos ønsket virksomhet. Disse sluttbrukerne er i hovedsak alle som handler hos en bedrift og har mulighet til å installere et Loyalty-kort.

Vår oppdragsgiver har en klar visjon for hvilke virksomheter de skal selge produktet til. Hvis vi ser på større virksomheter som Rema 1000 og COOP, har allerede disse egne løsninger for å belønne lojalitet. Loyalty-kort skal være et minimalistisk alternativ for mindre aktører som ikke har mulighet til å utvikle egne, bærekraftige løsninger.

## 1.6 Rapportens struktur

Her går vi kort over kapitlene som gjenstår og hva leser kan forvente under hvert kapittel.

### Kap. 2 - Teori

Her vil vi gå gjennom den relevante teorien for prosjektet. Dette omfatter spesielt digitalisering og metodikken vår. Etter å ha lest dette kapitlet, bør leseren ha forståelse over teorien bak resten av innholdet.

### Kap. 3 - Valg av teknologi og metode

Her går vi over hvilken teknologi og hvilken metode teamet har valgt for å utvikle og løse problemstillingen vår. Her vil vi også begrunne valgene vi tar, slik at leser får innblikk i hvorfor spesielt disse valgene anses som gode.

### Kap. 4 - Resultater

Resultatkapitlet deles opp i 3 deler, vitenskaplige, ingeniørfaglige og administrative resultater, og her skal vi gå gjennom alle resultatene våre, uten videre diskusjon.

### Kap. 5 - Diskusjon

Dette kapitlet deles opp likt som forrige, men her diskuterer vi alle resultatene, og går gjennom eventuelle avvik fra den originale planen.

### Kap. 6 - Konklusjon og videre arbeid

Her oppsummerer gruppen resultatene og diskusjonen og trekker en sluttkonklusjon basert på arbeidet vi har gjort. Til slutt vil vi se på videre arbeid som kan gjøres med det vi har produsert.

## 2 Teori

Kapittel to inneholder fire underkapitler. Første underkapittel tar for seg bakgrunnen for utviklingsprosjektet, med hovedvekt på RealNetts kildehenvisninger og lojalitetsprogrammer. Neste underkapittel går i dybden på hva digitalisering er og i hvilken grad oppgaven er et digitaliseringsprosjekt. Tredje underkapittel tar for seg metodikk og teorien rundt smidig utvikling. Siste underkapitlet handler om teorien bak flere av teknologiene vi har benyttet oss av under prosjektperioden.

## 2.1 Bakgrunn

Ideen om å digitalisere lojalitetsprogrammer er ikke ny, og det eksisterer allerede mange digitale varianter. RealNett oppdaget en fellesnevner ved de eksisterende tjenestene, nemlig at de tilhørte store markedsaktører. Rema 1000, COOP, Amazon Prime og Uber er eksempler på store selskap med gjennomførte og gode digitaliserte lojalitetsprogrammer. Til sammenligning kan vi se på Dromedar Kaffebar. I dag bruker de pappkort med fysisk stempeling for å belønne lojalitet. Den økonomiske kraften til Dromedar Kaffebar kan gjøre vanskelig å leie inn konsulenter for digitaliseringsprosjekter.

RealNett har identifisert en mulighet for å utvikle tilleggstjenester de kan tilby sine kunder. I dag tilbyr de fiber- og bredbåndstjenester til bedrifter i SMB-markedet. Blant annet benytter Mormors Stue og Trondheim Torg seg av deres tjenester (*Realnett - utvalgte kunder 2020*). Ideen baserer seg på at RealNett vil tilby digitale lojalitetsprogrammer direkte til sine kunder, eller til markedsfirmaer hvis muligheten oppstår.

I våre samtaler med RealNett kom det frem at ideen deres kommer fra markedsundersøkelser og egne erfaringer. De har ikke gjennomført egne undersøkelser eller brukt norske kilder i deres forarbeid (foruten deres erfaringer med norske aktører). Kildene deres er amerikanske, men gir et godt bilde av hvorfor lojalitetsprogrammer er viktige for å holde på kunder, og øke omsetning.

### 2.1.1 Lojalitetsprogrammer

Lojalitetsprogram brukes for å oppmuntre kunder til å returnere til den aktuelle bedriften for å handle mer. De lokkes med tilbud, kampanjer og eventuelle kuponger. Eksempelvis er stempelkort en enkel form for lojalitetsprogram. Et stempelkort på kaffe gir deg hver femte espresso gratis, da vil forbrukeren få et incentiv til å fullføre stempelkortet for å motta sin belønning.

*“Lojalitetsprogram tilbyr to nøkkelfunksjoner: De belønner kunden for lojalitet, og de tilbyr bedriften en overflod av kundeinformasjon”* (Chen 2019).

Et stempelkort tilbyr belønning for lojale kunder. Eksempelvis kan kunden få en gratis kaffekopp for hver femte de kjøper. Problemet med stempelkort oppstår når vi ser på James Chens andre nøkkelfunksjon. Med analoge (fysiske) stempelkort vil ikke kjøpet bli registrert på kunden, og bedriften mister muligheten til å innhente kundeinformasjon. Ved å digitalisere stempelkort gis kundene mulighet til å registrere seg, og dermed kan bedriften binde kjøpshistorikk opp mot de registrerte kundene. Et slikt datasystem vil kunne tilby en overflod av kunde- og kjøpsinformasjon, i tillegg til belønning for lojale kunder.

Et eksempel på et godt fungerende lojalitetsprogram er Trumf. Trumf AS ble startet i 1996 og har inngått partnerskap med flere aktører. Med Trumf mottar kunden belønning tilsvarende 1% av kjøpssummen for hvert enkelt kjøp som blir gjennomført. I tillegg har de flere trippel-trumf torsdager i året, hvor belønningen er økt til 3%. Løsningen er enkel og digital, etter registrering vil alle kjøpene lagres hos Trumf, og belønningen lagt inn i kundens trumf-saldo. Denne saldoen kan til enhver tid innkasseres i form av overføring til bankkonto, eller goder hos noen av Trumfs partnere. Lojalitetsprogrammet Trumf kan ses på som vellykket fordi den treffer begge nøkkelfunksjonene til James Chen. De belønner lojale kunder med 1% av hver handel, samtidig som den registrerer og lagrer all kjøpshistorikk til medlemmene sine. Noen eksempler på firmaer som benytter seg av trumf er: Kiwi, Joker, Tui, Shell og Viaplay (*Trumf.no 2020*).

Et lojalitetsprogram med mange og aktive medlemmer er sannsynlig å tilføye den aktuelle bedriften verdier i flere sektorer. I en undersøkelse gjort av Bond Brand Loyalty i samarbeid med VISA kommer det frem at: *“73% av medlemmer i lojalitetsprogram sannsynligvis vil anbefale bedrifter med gode lojali-*

tetsprogram til andre, mens 66% tilrettelegger sine innkjøp for å maksimere lojalitetsbonus (The Loyalty Report 2019 2019). Markedsføring er en viktig bieffekt av lojalitetsprogrammer. Slik det kommer frem i “The Loyalty Report 2019” ser vi at nesten tre fjerdedeler av medlemmer sier det er sannsynlig at de anbefaler bedriften videre. En undersøkelse gjort av Nielsen i 2012 viser at 92% av de spurte forbrukerne stoler på anbefalinger de får av venner og familie (Customer Trust 2012). Dermed er det sannsynlig at et gjennomført lojalitetsprogram vil føre til en økning i kunder.

Det gir i hovedsak merverdi i form av lojale kunder og markedsføring, men et annet viktig punkt er at bedriften får full oversikt over medlemmers kjøpshistorikk. Bedriftene velger selv hvordan de vil bruke den verdifulle dataen, og i neste delkapittel skal vi se hva som er “riktig” bruk.

### 2.1.2 Hvorfor/hvordan lykkes lojalitetsprogram?

For å forstå hvordan lojalitetsprogrammer lykkes må vi først forklare *Customer Lifetime Value* (CLV). CLV er et estimat for inntektene en bedrift kan forvente av en vilkårlig kunde.

$$\text{Lifetime Value} = \text{avg. customer lifetime} \times \text{avg. monthly spend} \times \% \text{ gross margin}$$

Figur 1: Grunnleggende formelen for beregning av CLV for SaaS-bedrifter (*Loyalty programs to maximize lifetime value 2020*)

For en SaaS-bedrift (forklart i 2.4.3 Skyløsninger) beregnes CLV med formelen i figur 1. Variablene i formelen er gjennomsnittlig levetid for kunder, gjennomsnittlig forbruk hver måned og bruttomarginen på salgene i prosent. Bruttomargin er forskjellen mellom innkjøpspris (for en SaaS-bedrift tilsvarer dette driftskostnader for programvaren) og utsalgspris. Denne marginen er relativt konstant med mindre bedriften endrer drastisk på deres tjenester, eller utsalgspris. For å øke CLV er det ønskelig å endre på de to første variablene. Lojalitetsprogrammer kan hjelpe bedrifter til å øke gjennomsnittlig levetid og forbruk ved å gi kunder et incentiv til å forsette handlingen ved den aktuelle bedriften.

Lojalitetsprogrammer påvirker CLV ved å øke hvor mye og hvor ofte et medlem handler i den aktuelle bedriften. Som nevnt kom “The Loyalty Report 2019” frem til at 66% av medlemmer i lojalitetsprogrammer tilrettelegger sine innkjøp for å maksimere bonus. Med tilrettelegging menes at kundene legger til varer for å nå en terskel, eller neste nivå, i lojalitetsprogrammet. Et eksempel på dette er nettbutikker som tilbyr deg gratis frakt hvis handlekurven din er over en viss sum. Et konkret eksempel er Platekompaniet som tilbyr gratis frakt på alle kjøp som overskrider en totalsum på 1000 kr (*Platekompaniet - frakt 2020*). På denne måten gir Platekompaniet et incentiv til sine kunder, og belønner de som er villig til å tilrettelegge sine innkjøp.

For en bedrift koster det fem ganger så mye å tilegne nye kunder, som å beholde en tidligere kunde (Link 2017). Ved å fokusere på de eksisterende kundene kan bedriften kutte utgifter og bygge en lojal

kundebase. Et annet viktig punkt Margaret Link poengterer er at lojale kunder kjøper for 67% mer enn nye kunder. Derfor er det viktig for en bedrift å beholde sine lojale kunder, da kostnaden er lavere og inntektene høyere. Selvfølgelig burde ikke en bedrift slutte å budsjettere penger til innhenting av nye kunder, men for bedrifter med en god kundebase er det muligheter for en økonomisk vekst ved å investere i eksisterende kunder fremfor nye. Et tiltak for å beholde eksisterende kunder er nettopp lojalitetsprogrammer.

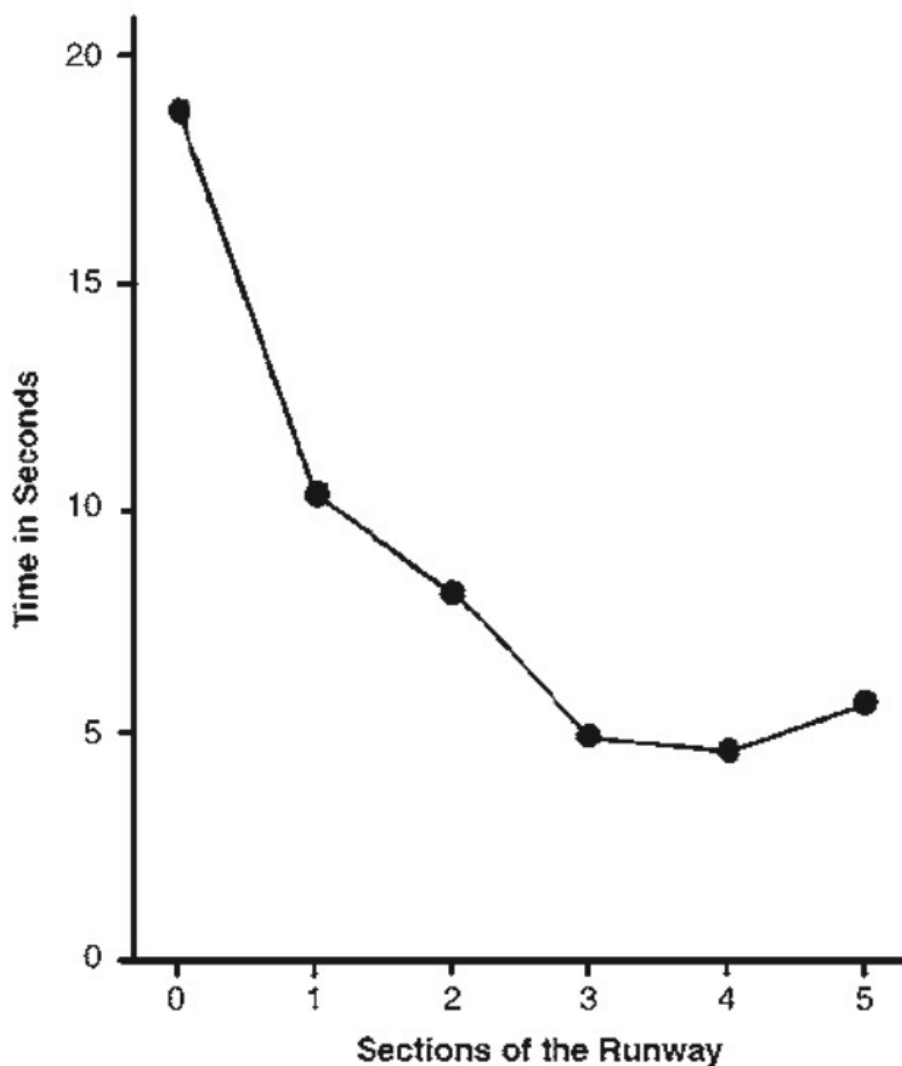
Hvordan klarer lojalitetsprogrammer å oppmuntre eksisterende kunder til å vende tilbake til den aktuelle bedriften? I følge en av artikkelene RealNett refererte til oss, er svaret delt inn i to hovedpunkter (*Loyalty programs to maximize lifetime value* 2020). Det ene punktet benytter seg av kundenes FOMO (Fear of Missing Out) ved å ha tidsbegrensede tilbud. Et psykologisk prinsipp kalt *Loss Aversion* forteller oss at mennesker føler mer smerte hvis vi mister en gjenstand enn gleden vi føler når vi mottar den samme gjenstanden («How Fear Drives Action» 2020). Eksempelvis føler vi mer glede av å **ikke** miste 100 kr, enn vi føler av å finne 100 kr på bakken. FOMO og *Loss Aversion* er i bunn og grunn det samme og baserer seg på samme tankegang. Vi mennesker opplever redsel for å bli utelatt, eller gå glipp av noe. Ved å tilby sine kunder tidsbegrensede tilbud, slik som når Komplett.no har nye tilbud hver time på Black Friday, benytter bedriftene seg av kundenes FOMO for å trigge impuls kjøp.

Det andre punktet nevnt i artikkelen, handler om å redusere antallet kunder som melder seg ut av lojalitetsprogrammet og bytter tjenesteleverandør. En vilkårlig kunde kan når som helst bytte til en annen leverandør, men hvordan kan bedrifter hindre leverandørbytte og beholde dem som kunder? For å hindre dette bruker bedriftene lojalitetsbonuser som er begrenset til den aktuelle bedriften. Hvis du for eksempel skal på flyreise med Norwegian, så gir de deg “cashpoints” en valuta som kun kan brukes hos Norwegian. I tillegg kan bedrifter som benytter seg av VIP-programmer, eller nivåer basert på kjøpsmengde og frekvens, “true” sine kunder med å frata deres opparbeidede lojalitetsstatus hvis de ikke fortsetter å handle hos dem. I praksis settes det ofte en tidsfrist på opparbeidet bonus. Med mindre det gjennomføres en transaksjon hos bedriften innen tidsfristen, så vil den opparbeidede bonusen bli fratatt kunden.

### 2.1.3 Stempelkort som lojalitetsprogram

Analoge stempelkort er en enkel form for lojalitetsprogram. Det er enkelt å forstå og distribuere, men det kan argumenteres for om fysiske stempelkort er miljøvennlig. Her skal vi ikke gå inn på miljøaspektet med stempelkort, men heller se på psykologien som ligger bak, og hvorfor det bidrar til å skape tilbakevendende kunder.

Psykologien bak stempelkort kommer fra “The Goal Gradient Hypothesis”. Konseptet med hypotesen er at mennesker (og andre biologiske skapninger) vil bruke mer krefter når de nærmer seg en belønning. Hypotesen er gammel og på 1930-tallet gjennomførte Clark L. Hull en vitenskapelig undersøkelse for å bevise den (Panagiotidi 2016). I undersøkelsen brukte de rotter som forsøksobjekter, og målte opp en rett bane med mat i enden. Deretter slapp han rottene løs i den andre enden og registrerte tiden de brukte på hver seksjon på banen (se figur 2 for resultatene av undersøkelsen).



Figur 2: Grafen viser tiden det tok rotter å komme seg fra en seksjon på banen til den neste. Start er 0 og belønningen er etter 5 (Panagiotidi 2016)

På grafen er det en tydelig reduksjon i tidsbruk når rottene får ferten av belønningen i enden. Gjennom flere litteratursøk har vi ikke funnet lignende undersøkelser gjennomført på mennesker, men teamet var samstemte i at vi har opplevd effekten. Konkurranseløp er et eksempel på en situasjon hvor effekten kan oppleves. Gjennom de første 1500 meterne blir deltakeren stadig mer og mer sliten, kanskje begynner han å sette spørsmålstegn ved om målgang er mulig. Så rundes den siste svingen og målstreken er synlig. Visualisering av målstreken trigger en følelse hos deltakeren, den er så nære og alt han trenger er en siste kraftanstrengelse. Deltakeren finner plutselig en reserve med krefter og begynner sluttspurten i et *høyere* tempo enn tidligere. Stempelkort utnytter denne effekten ved å ha en synlig “målstrek” i enden. I tillegg benytter bedrifter med stempelkort seg av en teknikk for å fremskyve effekten. De gir deg flere stempler enn forventet slik at du plutselig bare mangler tre stempler istedenfor fire. Ved å dytte deg nærmere målstreken benytter de seg av “The Goal Gradient Effect”, fordi vi mennesker legger inn en ekstra innsats for å komme i mål.



Stempelkort gir kunder et incentiv for å returnere til den aktuelle bedriften. De fleste personer er glade i produkter som er gratis, og stempelkort tilbyr ofte et gratis produkt som belønning. Prosessen er enkel å forstå for kundene, dermed er terskelen lav for deltagelse i lojalitetsprogrammet. Til sammenligning stiller digitale lojalitetsprogrammer flere krav til maskinvare og teknologisk kunnskap. Ved å benytte seg av den nevnte “Goal Gradient Effect” bidrar stempelkort til å skape tilbakevendende kunder.

#### 2.1.4 RealNetts kilder

I dette kapitlet har det hovedsaklig blitt referert til kildene RealNett har brukt, med noen innslag av eksterne kilder teamet har funnet. RealNetts kilder var:

- *Essential Loyalty Marketing Statistics for 2019* 2019
- *Loyalty programs to maximize lifetime value* 2020
- *The Loyalty Report 2018* 2018 (Dette er The Loyalty Report for 2018, ved et kjapt søk fant vi en lik rapport, utført av samme firma, for 2019 og brukte den istedenfor.

## 2.2 Digitalisering

Da internett ble gjort tilgjengelig for offentligheten i 1991, markerte det starten på en eksplosiv utvikling innenfor teknologi. Internett var revolusjonerende på mange områder, men spesielt begrepene kommunikasjon og tilgjengelighet fikk nye definisjoner. Det som tidligere var analogt ble digitalt og lå kun et tastetrykk unna. I år 2000 var nesten halvparten av USAs befolkning på internett. 16 år senere hadde det økt til 76 % av befolkningen. Samme år er 97 % av alle som bor i Norge koblet til internett. (Max Roser og Ortiz-Ospina 2020) Som en reaksjon på denne eksplosive veksten, så de fleste virksomheter seg nødt til å iverksette digitaliseringsprosesser for å holde seg konkurransedyktige. I dette kapitlet skal vi gå i dybden på hva digitalisering er, hvorfor det er nødvendig og hva som må ligge til rette for en vellykket digitaliseringsprosess.

### 2.2.1 Hva er digitalisering?

Digitalisering er et vidt begrep med flere forskjellige betydninger. Mange vil assosiere digitalisering med overgangen fra det analoge til det digitale, altså fra det skrevne materiale til vårt kjære binære tallsystem. Den norske regjeringen (*Digitalisering i offentlig sektor* 2014) har definert digitalisering slik: “Digitalisering handler om å bruke teknologi til å fornye, forenkle og forbedre. Det handler om å tilby nye og bedre tjenester, som er enkle å bruke, effektive, og pålitelige.” De setter ikke fokus på digitaliseringen av materiale, men hvordan teknologien kan forbedre prosesser innad i allerede definerte økosystem. Selv om det kommer tydelig frem hvor fokuset for digitalisering ligger, er definisjonen vag. Hvordan vet virksomheter at deres digitaliseringsprosjekt er effektive, enkle å bruke og pålitelige?

I dag brukes ofte begrepene digitisering, digitalisering og digital transformasjon om hverandre. Ved å se på forskjellen mellom begrepene kan vi isolere effekten et digitaliseringsprosjekt skal ha for en virksomhet.

Gartner har definert digitisering i deres IT-ordliste. “... digitization takes an analog process and changes it to a digital form without any different-in-kind changes to the process itself.” (*Digitization, glossary* 2020) Her ser vi tydelige forskjeller fra regjeringens definisjon av digitalisering. Fokuset for digitisering er å gå fra analog form til digital form, uten å gjøre endringer på selve prosessen. Under digitalisering ligger fokuset på å benytte seg av ny teknologi for å forbedre en eksisterende prosess. Et av premissene for et digitaliseringsprosjekt er at det i forkant er gjennomført digitisering.

Digital transformasjon inneholder typisk flere digitaliseringsprosjekter, og har fokus på å bruke ny teknologi for å endre forretningsmodeller og strategier. "...this broader term refers to the customer-driven strategic business transformation that requires cross-cutting organizational change as well as the implementation of digital technologies." (Bloomberg 2018) Tenker vi tilbake på regjeringens definisjon av digitalisering, vet vi at målet med digitalisering er å fornye, forenkle og forbedre. I digital transformasjon er fokuset på endringer i bedriften, basert på implementasjon av ny teknologi. Den nye teknologien betyr nye muligheter for å optimalisere bedrifters kjernefunksjonalitet, samtidig som det åpner opp for implementasjon av nyttig tilleggsfunksjonalitet.

Så hva er digitalisering? Kortfattet er det implementasjon av teknologi for å gjøre prosesser eller tjenester digitale. Samtidig er det viktig å tenke på at en digitaliseringsprosess er kostbar og sjelden kortfattet. Med implementasjon av nye systemer følger det med en ny opplæringsprosess av ansatte. Derfor er det viktig å identifisere i hvilken grad digitaliseringen forbedrer de eksisterende tjenestene, og hvorvidt det er brukevennlig og håndterbart for ansatte og bedriftens kunder.

### 2.2.2 Digitaliseringsprosessen

Selve digitaliseringsprosessen gjennomføres trinnvis, og i dette underkapitlet har vi gått ut ifra nettstedet Prosjektveiviseren (*Prosjektveiviseren* 2020). Nettstedet er utarbeidet av Digitaliseringsdirektoratet (Difi).

*"Prosjektveiviserens formål er å bidra til flere vellykkede prosjekter. Den beskriver virksomhetens overordnede styring av sine prosjekter, basert på faser og beslutningspunkter"*

Sitatet er hentet fra nettstedet og forklarer at dets formål er å hjelpe virksomheter til å finne suksess i deres digitaliseringsprosjekter. Veiviseren er oppdelt i seks trinn:

- Idé
- Konsept
- Planlegging
- Gjennomføring
- Avslutning
- Realisering

I slutten av hvert trinn inntreer virksomheten i en beslutningsfase. Her skal de involverte bestemme premissene for neste trinn. Vi skal ikke gå i dybden på trinnene, men heller kjapt gjennomgå de viktigste, og mest relevante underpunktene.

#### **Trinn 1 - Idé**

Dette trinnet handler hovedsakelig om å identifisere behov virksomheten har, og forstå hvor behovet kommer fra. En virksomhet kan ha nytte av å utbedre, og revidere, sine behov ved eventuelle nasjonale endringer. Ofte kan behovsutredninger utløses av endringer i loven, nye nasjonale strategier eller nye utviklingstiltak som direkte omhandler deler av virksomhetens daglige drift. Gjennom utredningen kan det komme frem nye idéer til fremtidig drift eller problemer ved den daglige driften. I slutten av trinnet er det en beslutningsfase hvor teamet, med virksomhetsledelsen, tar stilling til om behovsutredningens resultater kan utredes nærmere i konseptfasen.

#### **Trinn 2 - Konsept**

Etter trinn 1 har virksomheten identifisert et problem, idé eller behov som krever videre utredning. I den videre utredningen skal de få svar på om prosjektet er gjennomførbart. Dette gjøres ved å utarbeide flere konseptidéer, hvor alle konseptene er en mulig løsning på problemet. Hvert konsept vil bli gransket for å se om de er gjennomførbare. Her undersøker teamet om virksomheten besitter nødvendig kompetanse, om prosjektet er økonomisk forsvarlig og hvorvidt konseptet faktisk løser problemet, eller tilfredsstillende behovet. Beslutningsfase to kommer i slutten av trinnet, hvor virksomheten må bestemme seg for om konseptet de har utredet er gjennomførbart og realistisk.

### **Trinn 3 - Planlegging**

I dette trinnet skal konseptet detaljeres og konkretiseres. Det første og viktigste punktet er å etablere teamet som skal gjennomføre prosjektet. Deretter skal teamet utarbeide et styringsdokument som inneholder prosjektets mål, prosjektplan, gevinstrealiseringsplan og gjennomføringsstrategi. Disse dokumentene setter premissene for gjennomføringen av prosjektet som kommer i neste trinn. Det siste som skjer i dette trinnet, er at produkteier må godkjenne styringsdokumentet.

### **Trinn 4 - Gjennomføring**

Her gjennomføres prosjektet i henhold til styringsdokumentet som ble utarbeidet i forrige trinn. Selve gjennomføringen deles ofte opp i flere delfaser. Teamet gjør lurt i å se på hvilken type prosjekt som skal gjennomføres og velge delfaser i henhold til dette. Et eksempel er å bruke smidig metodikk (se 2.3) med korte og hyppige delleveranser. Avslutningen på dette trinnet er en beslutningsfase hvor produkteier skal godkjenne leveransene, og se om resultatene er i tråd med styringsdokumentet.

### **Trinn 5 - Avslutning**

I avslutningsfasen går teamet gjennom styringsdokumentet og sammenligner med produktet. Her skal de øvrige dokumentene som prosjektregnskap og prosjektdokumentasjon ferdigstilles og arkiveres. Samtidig utarbeides det en sluttrapport med prosjektevaluering og prosjektresultater. Er teamet fornøyd med produktet vil det bli sendt til en linjeorganisasjon som tar seg av forvaltning, drift og videre utvikling. Trinnet avsluttes med at produkteier beslutter at prosjektet kan avsluttes og teamet kan oppløses.

### **Trinn 6 - Realisering**

I realiseringstrinnet følger virksomheten opp videre gevinster av produktet og vurderer oppnåelsen av effektivitetsmålene. Her brukes gevinstrealiseringsplanen til å iverksette aktiviteter for å realisere ønskede gevinster. Eksempler på aktiviteter er opplæring av ansatte, og etablere og følge opp endrede arbeidsprosesser. Det siste som skjer er at produkteier vurderer de oppnådde gevinstene mot gevinstrealiseringsplanen og beslutter videre tiltak for å fortsette økningen av gevinster.

Regjeringen har også definert fem prinsipper for digitaliseringsprosjekter de mener må være på plass for å sikre suksess (*Prinsipper for digitaliseringsprosjekter* 2016). Det første prinsippet er å starte med å definere og kartlegge behov. Her ser vi at trinn 1 i prosjektveiviseren dekker dette punktet ved å starte med behovsidentifisering.

Punkt to handler om tenke stort, men starte smått. Regjeringen mener det er viktig å sette seg ambisiøse mål og tenke helhetlig. Blir prosjektet komplisert er det hensiktsmessig å dele det opp i mindre delprosjekter. Dette sikrer også hyppige leveranser og tidlig gevinstinnhenting.

Punkt tre handler om å velge riktig samarbeidspartnere for utviklingen av systemet. Her burde det tenkes på hvilken leverandør som passer prosjektet, men for å gjøre et kvalifisert valg trenger produkteier å identifisere utviklingsprosessen og hvilke kompetanser som kreves. Samtidig er det viktig å være åpen for endringer underveis som kan føre til at andre leverandører blir mer attraktive.

Punkt fire handler om å sørge for riktig kompetanse og god lederforståelse. Fordi digitaliseringsprosjekter er virksomhets- og forvaltningsutvikling, er det viktig å sikre at prosjektet styres på en kompetent måte. For å sikre det trenger de tverrfaglig kompetanse, forståelse og forankring hos ledelsen.

Punkt fem handler om å levere hyppig. Som nevnt i punkt to er det hensiktsmessig å dele opp prosjektet i delprosjekter for å hindre at det jobbes med for komplekse systemer. Ved å levere hyppig vil det bli skapt nytte underveis i prosjektet. Da åpnes det for brukertesting av delprosjektene, som kan gi verdifulle tilbakemeldinger som hjelper med den videre utviklingen.

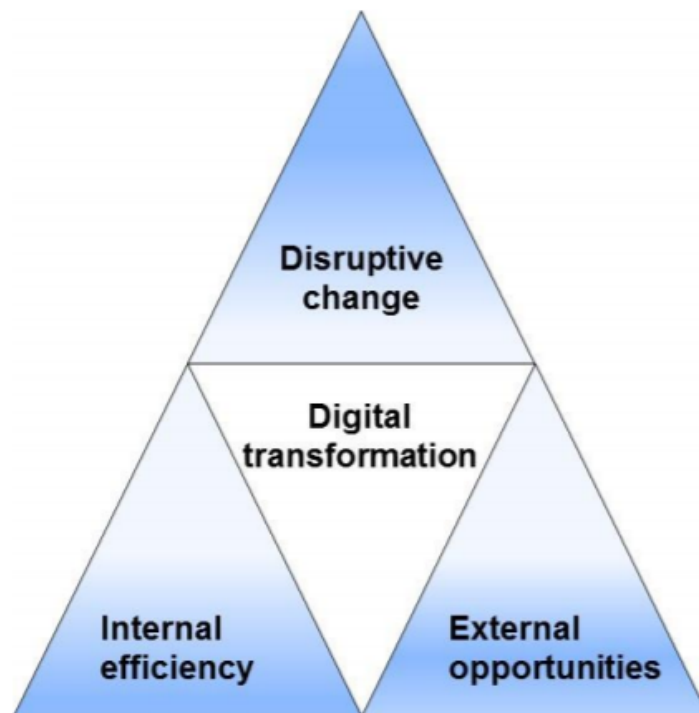
### **2.2.3 Fordelene ved digitalisering**

Overgangen fra analoge til digitale prosesser byr på et mangfold av fordeler. En rapport fra 2017 visualiserer fordelene ved digitalisering i et triangel (se figur 3). Fordelene er delt inn i tre hovedkategorier: Disruptive Change, Internal Efficiency og External Opportunities.

Disruptive change omhandler endringene i driftsmiljøet til virksomheten. Et digitaliseringsprosjekt kan gjøre tidligere arbeidsoppgaver utdatert. Ved en oppgradering fra manuell fakturabehandling til et helautomatisk system, vil virksomheten kunne frigjøre arbeidskraft til andre oppgaver. Samtidig åpner digitalisering for nye jobber. Et helautomatisk fakturasystem blir ofte levert av et eksternt firma som vedlikeholder og drifter systemet.

Internal efficiency omhandler forbedret effektivitet i prosesser, kvalitet og stabilitet. Dette skjer fordi virksomheten utsletter flere manuelle steg til fordel for et automatisk system med høyere presisjon. En virksomhet som utfører digitisering av sine data, vil ha bedre tilgang til den nevnte dataen og relevant statistikk vedrørende deres arbeidsprosesser. Ved å i tillegg automatisere flere av arbeidsprosessene, vil relevant data tilgjengeliggjøres og lagres digitalt. Virksomheten kan benytte seg av en sky-løsning for lagring av data, noe som vil redusere risikoen for datatap fordi slike løsninger har innebygde backup-løsninger.

External opportunities omhandler de nye mulighetene virksomheten tilegner seg ved digitalisering. Ved å benytte seg av ny teknologi skapes muligheter for å tilby nye tjenester, eller mer avanserte versjoner av eksisterende tjenester. Samtidig vil en digital tjeneste kunne tilby bedre responstid og klientbehandling fordi automatiske prosesser kan gjennomføres på noen få sekunder, i motsetning til tiden en manuell prosess ville tatt.



Figur 3: Effekten av digitalisering (Parviainen mfl. 2017)

I artikkelen “8 Advantages of Digitalization of Business” (Dominguez 2018) finnes det et par punkter som konkretiserer flere av fordelene digitalisering tilbyr virksomheter. Artikkelforfatteren Alberto Dominguez nevner blant annet digital tilstedeværelse, nye kanaler for å oppnå kontakt med kunder, forenkling av teamarbeid og intern kommunikasjon og bedre beslutningsgrunnlag.

For å kunne tilby kunder digitale tjenester trenger virksomheten en digital plattform. En digital plattform kan for eksempel være en nettside eller en applikasjon. Samtidig kan virksomheten benytte seg av andre digitale kanaler for å utvide sin digitale tilstedeværelse. Ved å ha en høy digital tilstedeværelse skapes det også nye kanaler hvor kunder kan oppnå kontakt med virksomheten. Eksempler på digitale kanaler er applikasjoner, e-post, sosiale medier og relevante blogger. LinkedIn kan nevnes som eksempel på et sosialt medium hvor bedrifter og kunder kan kommunisere. Her kan enkeltpersoner finne bedriftsrepresentanter og søke på jobber som er utlyst (*LinkedIn* 2020). De digitale tjenestene bidrar også til økt tilgjengelighet. Før i tiden da virksomheter bare hadde mulighet til å tilby analoge tjenester, var de avhengig av fysisk kommunikasjon. Tidligere var det nødvendig med fysisk oppmøte i banken for å kunne overføre mellom kontoer. Da begrenses tilgjengeligheten til tjenesten fordi faktorer som åpningstid og møtetidspunkter spiller inn. I dag er alt tilgjengelig gjennom bankens nettside eller applikasjon, som er tilgjengelig 24 timer i døgnet.

Digitalisering gir virksomheter muligheten til å forenkle teamarbeid og sin interne kommunikasjon. De nye digitale kommunikasjonskanalene hjelper ikke bare kunder å ta kontakt med bedrifter, men det hjelper også den interne kommunikasjonen i bedriften. I arbeidssammenheng er det ikke en selvfølge at teamet har mulighet til å arbeide i fysisk nærhet. Digitale verktøy som Google Docs og Microsoft Teams effektiviserer den interne kommunikasjonen i tilfeller hvor fysisk tilstedeværelse er umulig.

Ved å benytte seg av digitale tjenester gjør virksomheter det mulig å innhente relevant kundedata. Denne kundedataen kan senere bli kategorisert og visualisert i form av grafer. Da har virksomheten tilgang til all statistikk vedrørende sine digitale tjenester, og kan brukes som grunnlag for videre beslutninger. Noen

bedrifter tar steget videre og benytter seg av *Big Data* for å ta sine beslutninger. Big Data som konsept kan relateres til Business Intelligence (se *Hva er Business Intelligence?* 2020 for definisjon) og Data Mining (se *Hva er data mining?* 2020 for definisjon). “stordata-begrepet blir til forskjell fra de to andre først relevant når datamengdene, antall transaksjoner og antall datakilder blir så store og komplekse at de krever egne metoder og teknologier for å kunne trekke mening ut av dataene” (*Kartlegging og vurdering av stordata i offentlig sektor* 2015). Definisjonen er hentet fra en rapport utarbeidet av Viviento AS og Agenda Kaupang AS i 2015. Big Data blir av virksomheter brukt for å aggregere relevant statistikk, som deretter kan brukes som grunnlag for beslutninger.

#### 2.2.4 Loyalty som digitaliseringsprosjekt

Denne bacheloroppgaven omfatter utviklingen av en webapplikasjon med funksjonalitet for å designe, administrere, og distribuere digitale fordelskort. Dette er en spesifikk digitaliseringsprosess som skal fornye de nåværende, “analoge” eller fysiske løsningene i form av papir- og plastkort. På et bredere nivå befinner dette seg under digitalisering av lommebøker, som i nyere tid har hatt ekstremt stor vekst med teknologier som Apple Pay og Google Pay. Målt i prosentandel av smarttelefoneiere som tar i bruk digitale lommebokløsninger, har denne veksten vært større i Norge enn alle andre land i Europa, og alle land i verden utenom Kina per november 2018 i følge statistikksamling utført av Merchant Machine. (*The Rise of Digital & Mobile Wallets* 2018) I Norge har 42% av smarttelefoneiere benyttet seg av digitale lommebøker, som mest sannsynlig er takket være bankenes samarbeid i bruken av Vipps som betalingsmulighet, og buss- og flyselskapenes utvidelse av tilgjengelighet på digitale billettløsninger. Denne statistikken viser til at nordmenn er stadig mer villig til å ta i bruk nye digitale lommebokløsninger, spesielt blant yngre aldersgrupper.

### 2.3 Metodikk

#### 2.3.1 Smidig utvikling og Agile Manifesto

Mye av systemutvikling i dag baserer seg i stor grad på smidig utvikling, og prinsipper fra *Agile Manifesto*, eller *Manifestet for smidig programvareutvikling*. De viktigste prinsippene slik manifestet (Beck mfl. 2001) beskriver er:

**Personer og samspill** fremfor prosesser og verktøy

**Programvare som virker** fremfor omfattende dokumentasjon

**Samarbeid med kunden** fremfor kontraktsforhandlinger

**Å reagere på endringer** fremfor å følge en plan

“Dette vil si: Selv om punktene som står til høyre har verdi, så verdsetter vi punktene til venstre enda høyere.” (sst.).

Smidig utvikling og manifestet for smidig programvareutvikling kan ses på som en direkte reaksjon på hvordan utvikling generelt ble gjennomført tidligere. Tradisjonell utvikling, for eksempel vannfallsmetoden, er sekvensiell, og de forskjellige stegene i en utviklingsprosess gjennomføres én gang. For vannfallsmetoden har man følgende sekvens: Definer krav, design, implementasjon, testing, distribuer/opprethold. Man beveger seg ikke fra et steg til et annet før man er **helt** ferdig. Selv om dette kan fungere for visse prosjekter, kommer det klare problemer med denne utviklingsmetoden: Nye krav midt i prosjektet er vanskelig å implementere, ingen prototyper, høy risiko, skalering etc. (Stoica, Mircea og Ghilic-Micu 2013). For eksempel: Dersom man har et prosjekt som er antatt å ta 5 år, er det godt mulig at kravene man spesifiserte i begynnelsen av prosjektet nå har endret seg grunnet ny teknologi. Det er blant annet disse problemene som smidige utviklingsmetoder forsøker å løse. Det er verdt å nevne at tradisjonelle

utviklingsmetoder fortsatt har sin plass i programvareutvikling. (Stoica, Mircea og Ghilic-Micu 2013) nevner for eksempel at vannfallsmetoden er fint å bruke i et prosjekt der kravene er godt spesifiserte, teknologien er forstått, og prosjektet er kort. De smidige utviklingsmetodene ser derimot ut til å øke effektiviteten til utviklingen, samtidig som kommunikasjon med kunder og interessenter holdes mer sentralt

Smidige utviklingsmetoder så en popularitetsøkning rundt tusenårsskiftet. Vi kan se på et eksempel fra Hsu og Lin 2018, som undersøker Titansoft, et singaporsk firma stiftet i 2005. Titansoft tok i bruk smidig utviklingsmetode, mer spesifikt Scrum, i flere av kontorene sine i 2014-15, og så spesielt god utvikling for effektivitet av håndtering av problemer, og generell nedetid for prosessene de leverte. Det er selvfølgelig vanskelig å kreditere all forbedring eksklusivt til bruken av smidig utvikling, da Titansoft er et stort firma under utvikling. Det er viktig å legge merke til at de fikk en belastning på effektivitet det første året etter endringen av utviklingsmetode, men kvaliteten fikk derimot en positiv forbedring tross dette. Dette kan forklares med det personlige prinsippet til manifestet, der personer og samspill, og samarbeid med kunde står sentralt. Dersom man har sterkere kommunikasjon med kunder, er det gjerne en god byggestein for å kunne forbedre kvaliteten på produktet sitt.

Det er kort verdt å nevne følgende: Dybå og Dingsøy 2008 gjennomførte et studie av studier som så på effektiviteten til agile utviklingsmetoder (i all hovedsak eXtreme Programming). Selv om de merker seg generelt god oppfatning av smidig utvikling, merker de seg også at studiene som blir gjort på dette er mangelfulle. De ser at det er utappet potensiale for å lage sterke kilder til effektiviteten av utviklingsmetodene.

## 2.4 Teknologier

### 2.4.1 Versjonskontroll

Versjonskontrollsystemer registrerer endringer i en eller flere filer over tid, slik at man kan finne tilbake spesifikke versjoner senere. For systemutvikling betyr dette at dersom man gjør en endring som ødelegger hele eller deler av systemet kan man enkelt rulle tilbake til forrige fungerende versjon. Ytterligere problemer som kan oppstå ved systemutvikling er at lokale filer blir korrumpert, og problemet med hvordan flere kan samarbeide på samme prosjekt. Derfor benytter de fleste systemutviklingsprosjekter seg av distribuerte versjonskontrollsystem. Disse løser problemene ovenfor ved at alle som bidrar har en komplett historikk av alle versjoner lokalt på sin maskin, slik at det ikke finnes et *single point of failure*. I tillegg håndterer det sammenfletting av lokale endringer slik at man effektivt kan samarbeide på samme prosjekt (Chacon og Straub 2014).

### 2.4.2 Kontinuerlig integrasjon

Kontinuerlig integrasjon forkortet CI (Continuous Integration på engelsk) snakker man gjerne om sammen med kontinuerlig utrulling, CD (Continuous Delivery/Deployment på engelsk). CI og CD handler om å automatisere ulike prosesser i utviklingsarbeidet. Dette gjøres ved hjelp av en såkalt *Pipeline*, som gjerne starter med å hente koden fra versjonskontrollsystemet. Denne prosessen initieres automatisk enten ved at ny kode sjekkes inn i versjonskontrollsystemet eller det kan skje ved definerte tidspunkter. Deretter, vil koden kompiles/bygges og det vil automatisk kjøres enhetstester som rapporteres til utviklingsteamet. Disse delene inngår i CI-delen. Videre, kan man velge at dersom koden ble compilert/bygget uten problemer, og enhetstestene ble godkjent, at applikasjonen også rulles ut automatisk. Dette er det man mener med kontinuerlig utrulling eller CD. Ved å automatisere prosesser på denne måten sparer man seg for tregt manuelt arbeid hver gang man gjør endringer, og gjør det mulig å rulle ut endringer

raskere enn ellers. Det senker også sannsynligheten for at bugs blir med i utrullingene ved at dersom ett steg i prosessen feiler, vil resten av pipelinen avbrytes. (CodeShip 2020)

### 2.4.3 Skyløsninger

Skyløsninger er et alternativ til *on premise*-løsninger, og handler om at man kjøper løsninger som tjenester tilgjengelig via internett, i stedet for å kjøpe produkter man eier og håndterer selv lokalt. Dette kan inkludere alt fra programvare til digital infrastruktur, og man skiller gjerne mellom forskjellige typer skyløsninger på følgende måte:

- **SaaS:** Software as a Service er applikasjoner som kjører i skyen som man som regel får tilgang til via en nettleser, og som man som regel betaler et månedlig eller årlig abonnement for å bruke. Dette kan være et alternativ til å kjøpe programvare-lisenser, og fordelen er at i stedet for å betale mye på forhånd, betaler man bare så lenge man har bruk for programvaren. I tillegg slipper man å installere programvaren lokalt.
- **PaaS:** Platform as a Service er en skybasert løsning for å kjøre applikasjoner. PaaS-tjenester sørger gjerne for alt av servere, databaser, nettverk, lagring og operativsystem automatisk. Fordelen med slike løsninger er at man slipper å investere i hardware, og man slipper å sette opp all nødvendig programvare selv. Man betaler som regel månedlig for plattformen.
- **IaaS:** Infrastructure as a Service tilbyr grunnleggende dataressurser slik at man kan bygge opp komplekse digitale systemer akkurat som man vil uten å måtte betale for hardware på forhånd. Slike tjenester senker derfor den marginale kostnaden for datainfrastruktur betraktelig. Man betaler som regel kun for de ressursene man bruker til enhver tid og er da som regel mer kostnadseffektivt. Det er også hurtigere å sette opp, og enklere å skalere enn å bygge ut sin egen fysiske infrastruktur. (IBM 2020)

## 3 Valg av teknologi og metode

### 3.1 Systemutvikling

#### 3.1.1 Node.js med Express

Teamet var samstemte rundt valget av JavaScript som programmeringsspråk til backend. Bakgrunnen for dette var at vi ønsket et gjennomgående programmeringsspråk. Løsningens backend ble utformet som et REST API for kommunikasjon med databasen. Gjennom flere studieprosjekter har teamet opparbeidet seg erfaring med bruk av Node.js og rammeverket Express til backendutvikling. Express tilbyr utviklere en minimalistisk måte å sette opp web- og mobilkonfigurasjoner. Node.js og Express er godt dokumentert med fyldige APIer, og har gode muligheter for å integrere eksterne rammeverk.

#### 3.1.2 Vue.js

Vue.js er et JavaScript frontend-rammeverk som brukes for å utvikle webgrensesnitt. JavaScript ble valgt foran andre programmeringsspråk fordi det tilbyr utallige, nyttige biblioteker for applikasjonsutvikling. En egenskap som Vue tilbyr er deres direktiver. Dette er egendefinerte html-attributter som tilbyr utvikleren unike muligheter for å legge til reaktiv oppførsel i html-elementer. I tillegg er rammeverket open-source med god dokumentasjon. Valget av frontend-rammeverk falt hovedsakelig på teamet da oppdragsgiver kun hadde få betingelser. De ønsket et rammeverk som var “cutting edge” og enkelt å videreutvikle. Teamet har erfaringer med React.js gjennom tidligere studieprosjekter, men Vue.js ble valgt etter anbefaling fra oppdragsgiver om å utfordre oss selv med et nytt rammeverk.



### 3.1.3 React Native

For å skanne fordelskortene kom vi sammen med oppdragsgiver frem til at den beste løsningen var å utvikle en egen mobilapp. Merk at i oppgaveteksten står det spesifikt at det er et poeng at det ikke skal utvikles en egen mobilapp for løsningen; dette gjelder derimot for kortene til forbrukerne, at disse skal integreres i eksisterende lommebok-løsninger og ikke eksistere i en frittstående app. Vi har tidligere brukt Hoopla sin billettløsning og har erfart at å skanne billetter gjennom en mobilapp fungerer bra. Alternativet ville vært å bygge det inn i webløsningen, men dette ville trolig ikke vært like stabilt eller brukervennlig. For å utvikle mobilapplikasjonen tenkte vi først at det var nærliggende å velge Vue Native, da vi bruker Vue.js for web-løsningen. Ved videre undersøkelser fant vi ut at Vue Native fungerer som en *wrapper* for React Native og ikke tilbyr like mange funksjonelle egenskaper. Med *wrapper*, mener vi at Vue.js-koden bare blir oversatt til React Native. Dermed valgte vi bort Vue Native til fordel for React Native. Dette rammeverket gir oss mulighet til å utvikle én JavaScript-applikasjon som fungerer både på iOS og Android. Nytteverdien av dette baserer seg på at iOS og Android bruker forskjellige programmeringsspråk og rammeverk, og ved å bruke React Native slipper vi å utvikle to apper. React Native er også et modent rammeverk med mye innebygget funksjonalitet. Dette gjør at utviklingen av en helt egen app til vårt formål blir like trivielt som å skulle gjøre det samme som den del av webløsningen.

### 3.1.4 MySQL

Valget av databaseteknologi ble tatt på grunnlag av teamets erfaring. I løpet av studiet har vi gjennomgående brukt MySQL som databasetjener. Samtidig er det godt dokumentert, open-source og populært blant eksisterende systemer.

### 3.1.5 GitHub

Vi bruker Git som versjonskontrollsystem siden dette er standard for systemutviklingsprosjekter og det er dette gruppen har brukt før. Videre, måtte vi velge hvilken Git-tjeneste vi skulle benytte oss av. Da sto det hovedsakelig mellom GitHub og GitLab. Gruppen har brukt begge systemene før i ulike sammenhenger. Fordeler med GitLab er at det har innebygget CI/CD-løsning noe GitHub ikke har. Grunnen til at vi valgte GitHub foran GitLab var at gruppen liker denne løsningen bedre, og at vi endte opp med å velge en annen løsning for CI/CD uansett.

### 3.1.6 Jest

For enhetstesting i JavaScript finnes det utallige ulike rammeverk. Jest er et populært rammeverk vedlikeholdt av Facebook som også er det eneste rammeverket gruppen har erfaring med. Dermed ble valget av testrammeverk ganske enkelt.

### 3.1.7 AWS Elastic Beanstalk

Oppgavestiller ønsket at vi skulle bruke AWS for å levere løsningen fordi de bruker dette fra før til andre formål. AWS har mange forskjellige løsninger for å *hoste* nettsider, og det sto mellom EC2, Lightsail og Elastic Beanstalk. Grunnen til at vi valgte Elastic Beanstalk er at den automatiserer store deler av serverkonfigurasjonen, og kan enkelt skaleres opp og ned avhengig av trafikk. Samtidig har vi full kontroll over konfigurasjonen av miljøet siden vi kan legge ved system-konfigurasjonsfiler i versjonskontroll-systemet.

### 3.1.8 AWS CodePipeline

CodePipeline er en del av AWS sin kontinuelig integrasjons-løsning. Det finnes andre populære løsninger for kontinuerlig integrasjon som for eksempel Travis CI eller GitLab CI. Grunnen til at vi valgte Amazon

sin løsning var fordi at det er godt integrert mot Elastic Beanstalk.

### **3.1.9 Swagger**

Swagger brukes for å dokumentere REST APIet. Swagger produserer en interaktiv API-dokumentasjon som kjører i nettleseren og som har innebygget funksjonalitet for å prøve ut APIet mot backend-serveren, og få ut resultatet umiddelbart. Dette gjør det enkelt for andre utviklere å sette seg inn i og prøve ut APIet vårt. Grunnen til at vi valgte dette er at vi har brukt andre API-er dokumentert på denne måten før, og opplevde det som svært brukervennlig.

## **3.2 Elektronisk lommebok**

### **3.2.1 Google Pay**

For Android telefoner så falt valget av elektronisk lommebok på Google Pay. Den største grunnen til dette er at det er den klart mest populære elektroniske lommeboken for Android, og ble ansett som spesielt hensiktsmessig, fordi vi ikke ønsket å øke terskelen for å kunne bruke de digitale fordelskortene. Etter litt utforsking så vi også at det var enkelt å håndtere APIet til Google Pay, og dokumentasjonen var også godt gjennomført. Videre var strukturen til Google kort også hensiktsmessig, fordi den skilte mellom en template og et kort. Det medførte at oppdateringer på all informasjon som ikke hadde med brukere å gjøre ble enklere. Alt dette gjorde Google Pay til et åpenbart valg for Android telefoner.

### **3.2.2 Apple Wallet**

Oppdragsgiver ønsket at forelderskortene skulle være tilgjengelig for iPhoneer gjennom Apple Wallet fordi det kommer med alle Apple produkter som har iOS 6 eller nyere. Dette operativsystemet ble gitt ut i 2012, og er for lengst utdatert. Dermed vil nærmest alle iPhoneer ha tilgang på Apple Wallet. Samtidig kan man utnytte flere muligheter Apple tilbyr gjennom Wallet sammenlignet med andre tredjepartsapplikasjoner. Fordelskortet kan for eksempel holde seg på låst skjerm med en varslingsmelding rundt et visst tidspunkt, eller hvis man er i nærheten av butikken kortet gjelder for. Ettersom oppdragsgiver spesifikt ønsket å bruke Apple Wallet, og fordi den tilbyr fler muligheter enn andre tredjeparts lommebøker for iOS, så ble valget enkelt.

## **3.3 Utviklingsmetode**

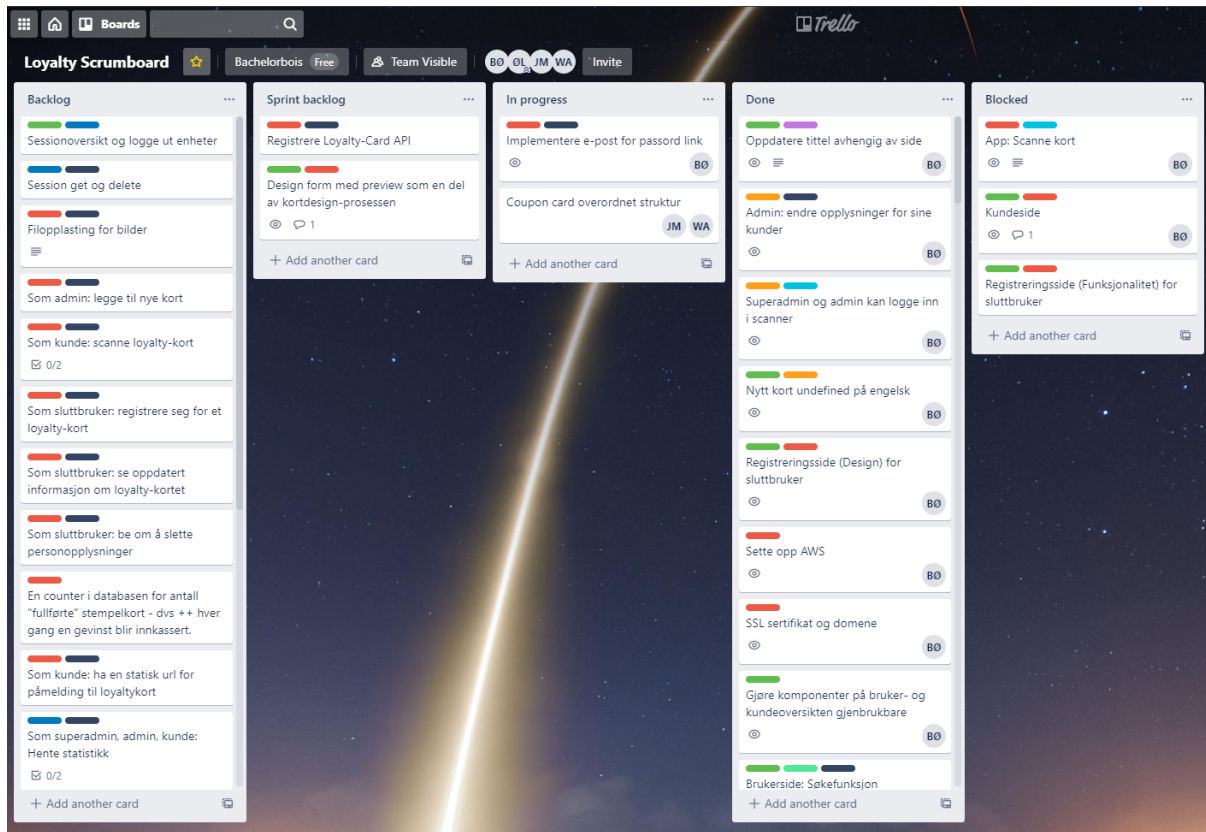
### **3.3.1 Lean systemutvikling med kanban**

Lean systemutvikling med kanban er en flytbasert prosess som egner seg godt for uforutsigbart arbeid, og har en høy toleranse for endringer underveis. Denne metoden baserer seg på å ha en kanban-tavle med oversikt over oppgaver i prosjektet. I stedet for å planlegge sprinter som i scrum, har man heller en kontinuerlig flyt av oppgaver mellom kolonnene på tavlen. I tillegg, er det vanlig å begrense maksimalt antall oppgaver per kolonne, spesielt det som er under arbeid. Dette er for å gi høyere gjennomstrømming av oppgaver i stedet for å ha mange halvferdige oppgaver liggende. (Skaug 2013) Det sto mellom dette og scrum, og grunnen til at vi valgte dette var at vi innså tidlig at utviklingsarbeidet skulle vise seg å bli ganske uforutsigbart fordi vi måtte lære oss nye teknologier og eksterne APIer, og det skjedde mange endringer underveis. I tillegg, hadde vi ikke tilgang til en scrum-master i bedriften vi jobbet for.

### **3.3.2 Trello**

For å holde styr på oppgavene trengte vi en form for tavle der vi kunne føre opp leveransene i prosjektet med status for gjennomføring. Dette kan enten være en fysisk tavle eller en digital. I et tidligere prosjekt

har vi brukt en fysisk tavle med post-it lapper, men det ble fort uoversiktlig fordi man hele tiden må reise seg for å lese lappene. I tillegg er dette lite fleksibelt siden det krever at alle jobber på samme sted til enhver tid. Av digitale tavle-verktøy sto det mellom to alternativer: Microsoft Teams med Microsoft Planner-tavler, eller Trello. Grunnen til at vi valgte Trello er fordi det er et mindre komplisert verktøy som likevel har all funksjonaliteten vi trenger. På Trello-tavlen vår hadde vi kolonnene: *Backlog*, *Sprint backlog* (prioriterte gjøremål), *In progress*, *Done*, og *Blocked* (venter på andre leveranser).



Figur 4: Skjerm bilde fra Trello

### 3.4 Arbeids- og rollefordeling

Den viktigste fordelingen vi gjorde i starten av prosjektet gikk ut på å fordele hvem som hadde ansvar for å sette seg inn i de ulike teknologiene utviklingsteamet ikke hadde erfaring med fra før. Dette var hovedsakelig: Apple Wallet API, Google Pay API, AWS, React Native, og Vue.js. Utover dette jobbet hvert medlemmene med individuelle oppgaver der det var nødvendig, uten en formell ansvarsfordeling.

- **William Andersson** har hatt hovedansvaret for integreringen mot Google Pay. En av grunnene til dette er at han selv eier en Android-telefon, som er nødvendig for å kunne teste kortene fra Google Pay. Han har sammen med Jonathan hatt hovedansvar for det meste som gjelder selve kort-lagingen, både front-end og back-end.
- **Øivind Larsson** har hatt hovedansvar for oppsett og utvikling innenfor frontend med Vue.js. Mot slutten av prosjektet tok han ansvar for rapportskrivning.
- **Jonathan Midttun** har hatt hovedansvaret for integreringen mot Apple Wallet. For å teste integrasjonen er det nødvendig å ha tilgang på en Mac maskin for å generere nødvendige sertifikater til

utsteding av kort, og en iPhone for testing, og siden Jonathan hadde begge deler var det naturlig at han fikk ansvaret for Apple Wallet. Som nevnt ovenfor utviklet William og Jonathan store deler av kortadministrasjonen i systemet, og sammen har de vært ansvarlig for kortlaging helt fra design i front-end til utsteding gjennom back-end til Google Pay og Apple Wallet.

- **Bjørnar Østtveit** har hatt hovedansvar for *DevOps*, dette innebærer: kontinuerlig integrasjon i AWS CodePipeline, bygging og testing i AWS CodeBuild, og utrulling til AWS Elastic Beanstalk. I tillegg til automatisk henting av sertifikater. Han hadde også hovedansvar for scanner-appen i React Native.

## 4 Resultater

I dette kapitlet skal vi gå gjennom resultatene vi har oppnådd gjennom arbeid med prosjektet. Dette deles inn i 3 seksjoner: Vitenskapelige, ingeniørfaglige og administrative resultater. De vitenskapelige resultatene omhandler hvilke resultater vi har oppnådd i forbindelse med problemstillingen vi har jobbet med. De ingeniørfaglige resultatene omhandler målene som ble satt i kapittel 1.1, 1.2 og 1.3. De administrative resultatene omhandler utviklingsprosessen vår.

### 4.1 Vitenskapelige resultater

Vi har laget en web-applikasjon og en mobilapplikasjon. Avhengig av hvilken systembruger du er, vil du ha tilgang til og ha bruk for forskjellige deler av applikasjonene. Vi skal her gå gjennom alle deler av det vi har utviklet som omhandler digitaliseringen av fordelskort.

#### 4.1.1 Fordelsprogramtyper

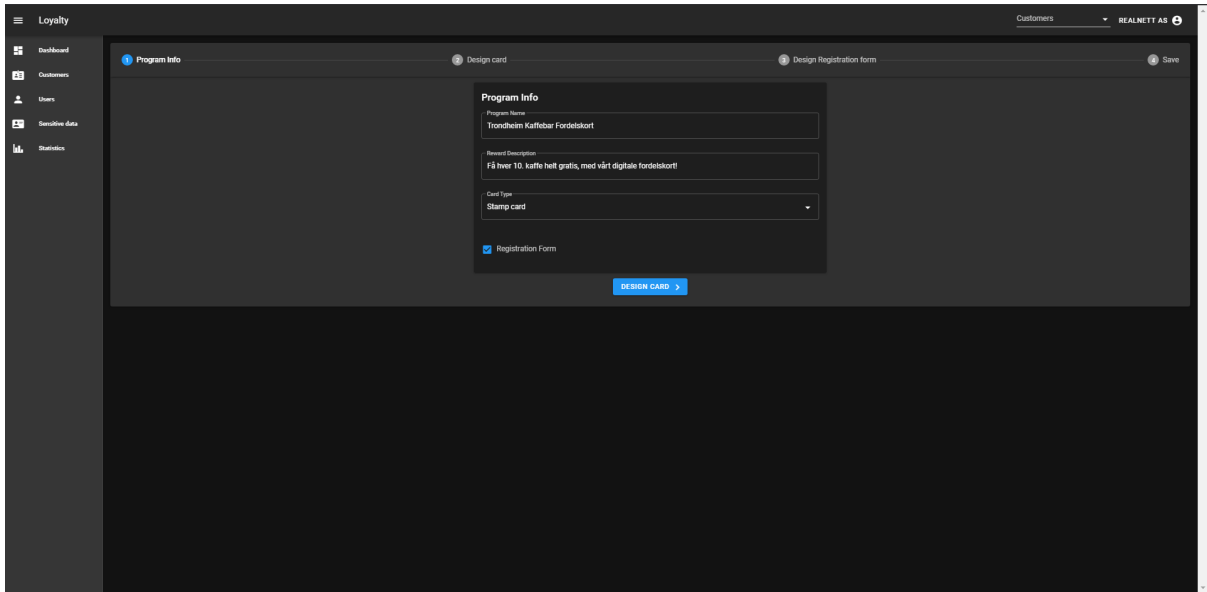
Programtypene som er implementert er:

- **Stempelkort:** Her får forbrukeren en fordel etter et visst antall stempel som kan opptjenes ved kjøp av en vare eller en tjeneste. Når stempelkortet er brukt opp blir kortet tilbakestilt til å ha null stempel slik at de kan begynne å tjene seg opp igjen på nytt.
- **Medlemskort:** Forbrukeren får en fordel av å være medlem, som bevises ved å vise kortet slik at en ansatt kan scanne det.
- **Kupong:** Forbrukeren får en fordel én gang ved å vise kupongen. Kupongen ugyldiggjøres etter bruk. Dette er funksjonelt implementert, men er ikke tilgjengelig i applikasjonen siden det ikke er laget en egnet distribusjonskanal for kuponger.
- **VIP-kort:** Forbrukeren får en fordel ved å vise kortet. Den som scanner kortet kan se navnet til den som er registrert slik at de kan sjekke dette opp mot legitimasjon hvis ønskelig. Dette er veldig likt et medlemskort bortsett fra at VIP-kort er eksklusive. Dette er funksjonelt implementert, men er ikke tilgjengelig i applikasjonen siden det ikke er laget en egnet distribusjonskanal for VIP-kort.

Den eneste måten å distribuere fordelskort på som er implementert er et egendefinert registreringsskjema med en valgfri statisk url.

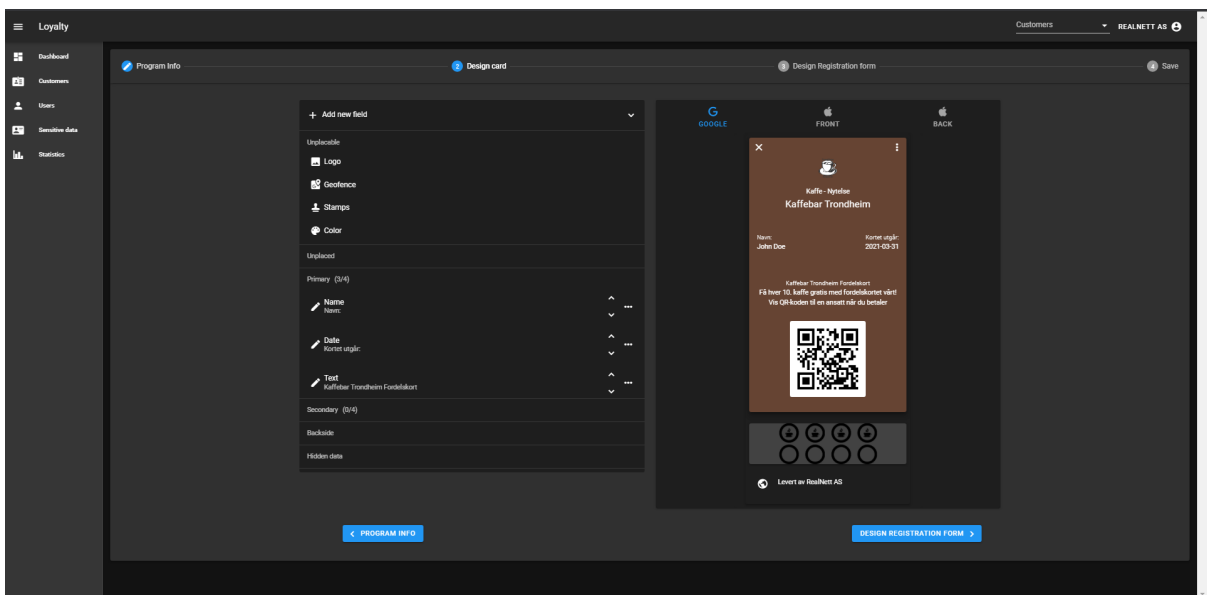
#### Design av fordelsprogram

Vi starter på begynnelsen av prosessen, som er design av selve programmet. Dette gjøres av superadministratorer og administratorer, etter kundenes ønsker. Designet gjøres i 3 steg: Programinformasjon, kortdesign og registreringsdesign.



Figur 5: Programinformasjon

På figur 5 ser vi programinformasjonen avbildet. Her kan en administrator enkelt velge programnavn, forklaring av fordel og hva slags type fordelsprogram det skal være. Man kan legge til registreringsskjema ved å krysse av for dette. Når man er ferdig her går man videre til kortdesign.



Figur 6: Kortdesign

Kortdesigneren på figur 6 er hoveddelen av designprosessen. På venstre side er det liste der man kan legge til felter inn i kortet. Disse “oversettes” så til hvordan kortet hadde sett ut når man først laster det ned til Google Pay eller Apple Wallet. Man kan legge inn tekstfelt, datofelt og navnefelt. Datofeltet kan også brukes som en utløpsdato, slik som er gjort i eksempelbildet. Skulle man ønske det, kan utløpsdatoen også settes til å være x antall dager etter registrering. Alle felter som er avhengig av brukerdata, for

eksempel navn og registreringsdato, vil fylles inn i kortet når man registrerer seg.

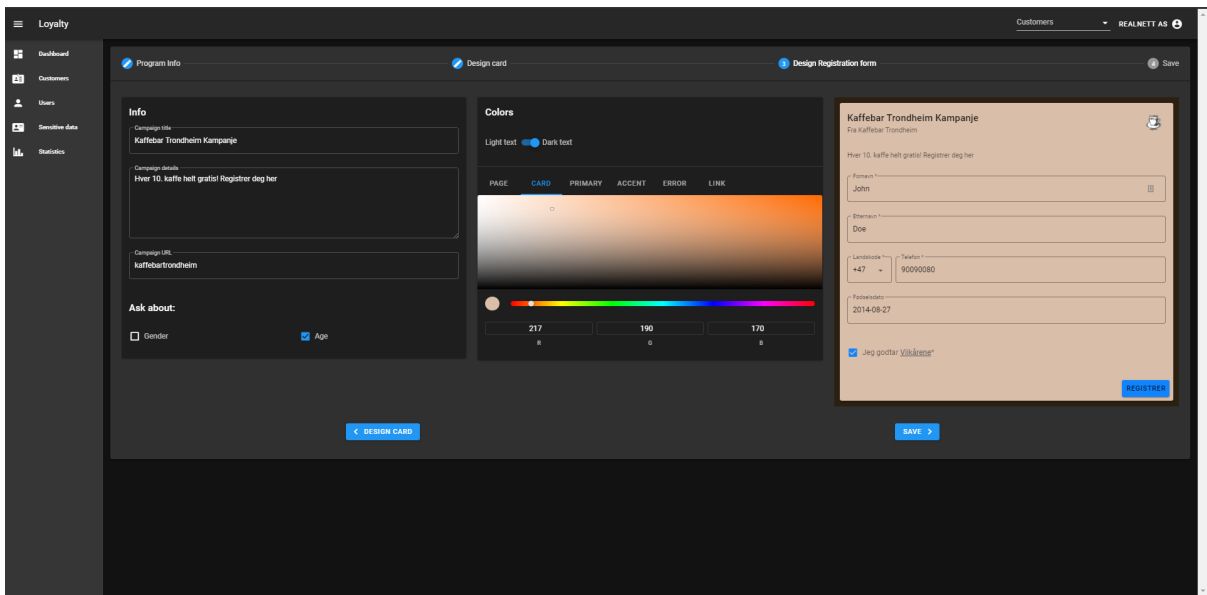
På høyre side ser vi forhåndsvisningen av et Google pass, og forsiden og baksiden til et Apple pass. Disse er ikke ekte Google og Apple kort, men heller vårt forsøk på å kopiere stilen slik at designeren enklere kan gjøre valg som gjør at utseendet blir bedre. På figur 7 ser vi forhåndsvisningen for Google og forsiden til Apple kort.



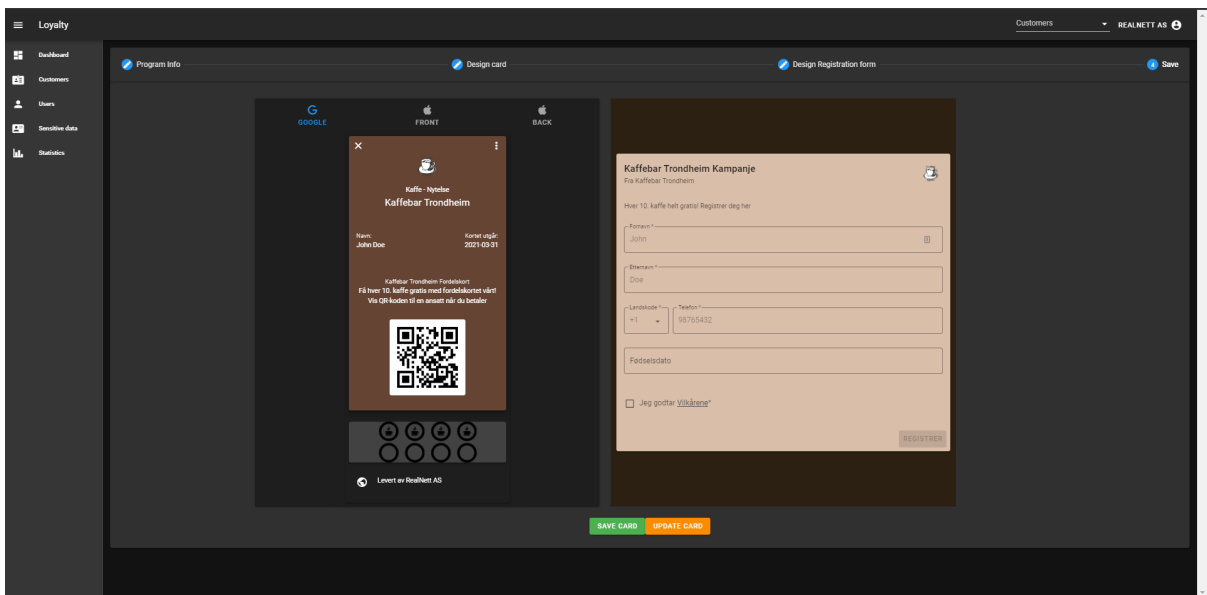
Figur 7: Forhåndsvisning av Google og Apple pass

Når man er fornøyd med designet av kortet, går man videre. Dersom man har valgt at kortet skal ha et registrerings skjema blir man sendt til registreringsdesignet. Design av registrering vises på figur 8. Her kan man velge utseendet på registreringsskjemaet sitt. I tillegg lar vi kunder velge mellom ekstra felter de skulle vært interessert i, for eksempel kjønn eller alder. Dette kan kunden videre se statistikk på, som vi kommer til senere.

Når man er ferdig med dette er designprosessen over, og man får en siste oversikt over produktet man har designet, som vist på figur 9. Er man fornøyd, kan man lagre kortet. All endring av kort foregår i samme miljø, slik at man enkelt kan endre på samme måte som man ville endret felter under selve designet.



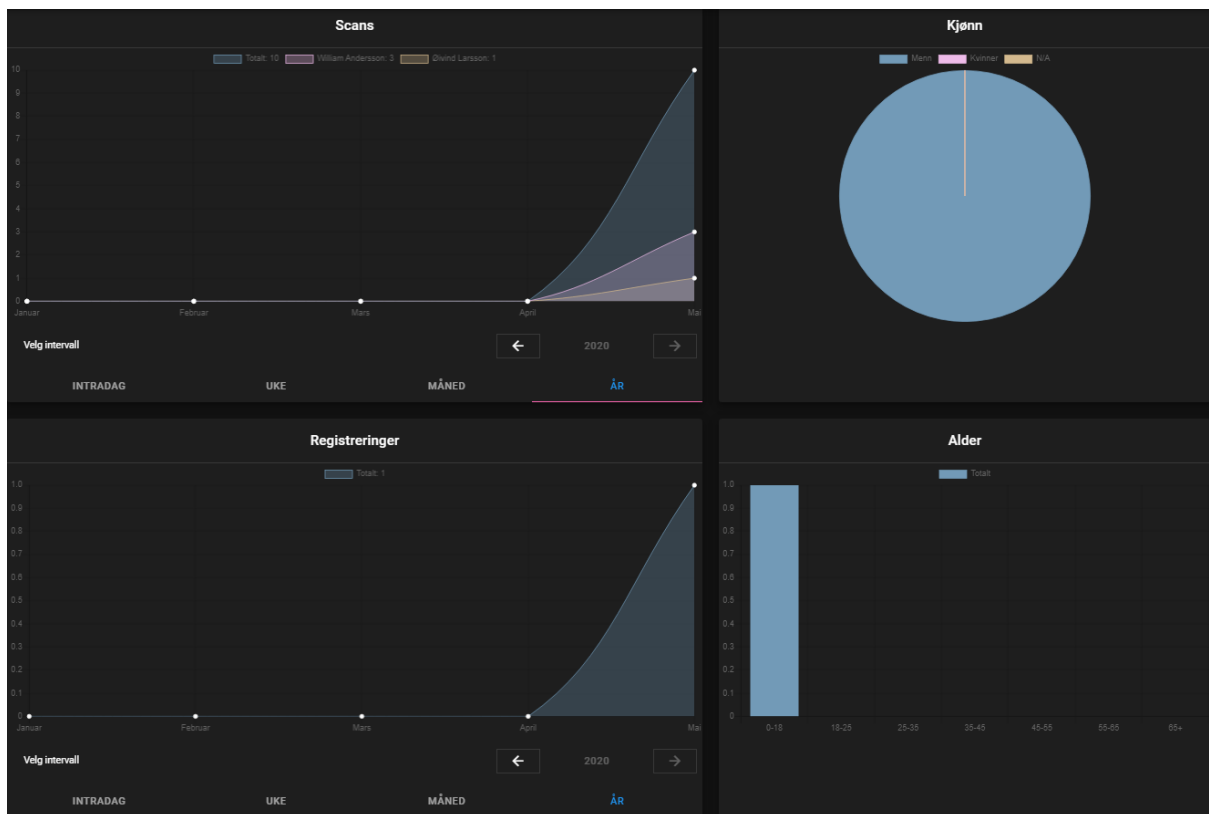
Figur 8: Registreringsdesigner med forhåndsvisning



Figur 9: Forhåndsvisning av ferdigdesignet produkt

## Kundeside og Mobilapplikasjon

Vi skal nå gå gjennom det som angår kunden av det vi har utviklet. Kundesiden er den samme webapplikasjonen som administratorer, men man har ikke like mange muligheter. Mobilapplikasjonen er en egenstående applikasjon til Android som brukes for å scanne kort.



Figur 10: Kundens statistikkside

På figur 10 kan vi se kundens statistikkside for et spesifikt kort. Her vil det eksempelvis være mulig å se hvor mange registreringer kortet til kunden hadde en spesifikk måned. Statistikken er for øyeblikket delt opp i scans og registreringer, og kjønn/alder dersom registreringskjema hadde dette. Man kan dele opp statistikken etter kort, eller ha all statistikk for alle kort aggregert. Scanning kan videre deles opp etter hvilken ansatt som har gjennomført scanningen. Videre skal vi se på mobilapplikasjonen hvor man gjennomfører disse scannene.

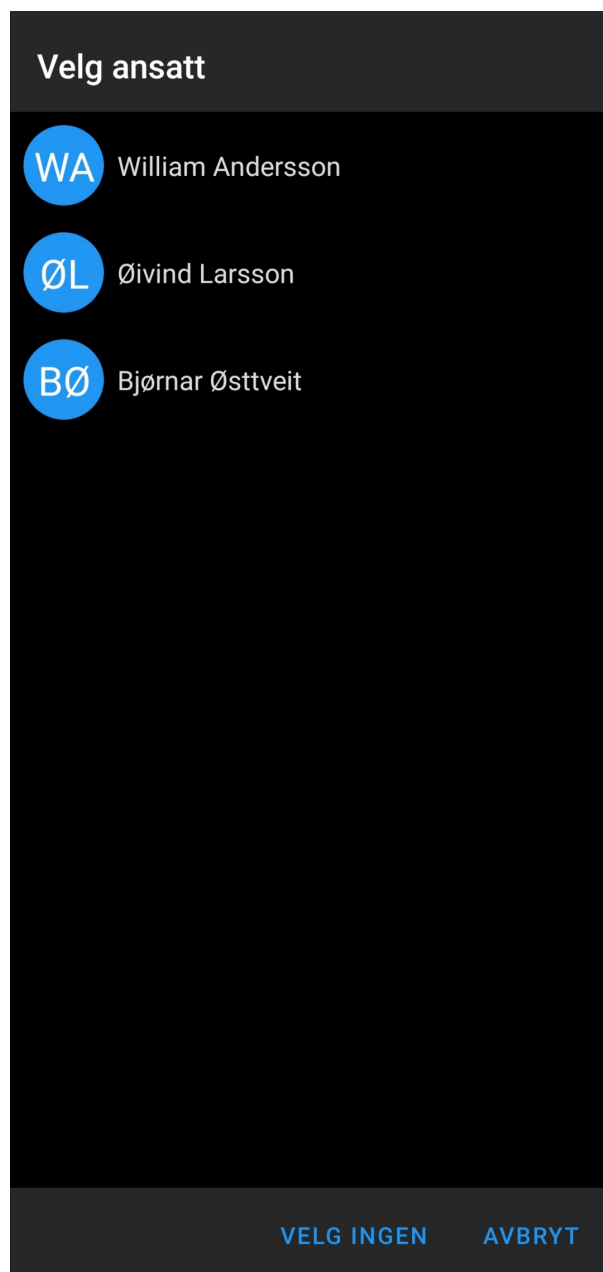
Mobilapplikasjonen kan kun lastes ned på Android-telefoner med en .apk fil. Denne applikasjonen brukes eksklusivt til å scanne sluttbrukerens kort. På figur 11 ser vi hvordan det ser ut etter en ansatt har scannet et kort. Alle kort scannes via en QR-kode som er UUID-en til en registrert bruker. Hva denne scannen faktisk betyr, håndteres av backenden. Applikasjonen leser en UUID spesielt fort, så selv om den kun reagerer på UUID-er, kan det være fort gjort å scanne ved uhell. Av den grunn er det mulig å angre scanningen sin, i tilfellet det skulle være ved uhell.





Figur 11: Mobilapplikasjonen ved scanning av en UUID

For videre oppdeling og mer statistikk har vi gjort at det er mulig å registrere flere ansatte i bedriften sin, slik at du kan fordele scanning per ansatt. På figur 12 ser vi hvordan dette ser ut.



Figur 12: Ansattssiden til mobilapplikasjonen

### Sluttbrukernes opplevelse

For sluttbrukere er prosessen enkel. Dersom kortet har registrering, er denne siden en statisk URL som kunden kan besøke på sin mobiltelefon. Registreringssiden vi så i 4.1.1, Design av fordelsprogram, er vist på mobil i figur 13. Når man registrerer seg og alt er i orden blir man sendt videre til en side der man får en knapp som gjør det mulig å legge til kortet i enten Google Pay eller Apple Wallet. Google Pay er avbildet på figur 14. Når man trykker på denne knappen, blir man videreført til Google Pay applikasjonen på telefonen sin, dersom den er lastet ned.

**Kaffabar Trondheim Kaffepanjen**  
Fra Kaffabar Trondheim

Hver 10. kaffe helt gratis! Registrer deg her

Fornavn \*  
John

Etternavn \*  
Doe

Landskode \*  
+1

Telefon \*  
98765432

Fødselsdato

Jeg godtar [Vilkårene\\*](#)

REGISTRER

Figur 13: Registreringssiden på mobil

**Gratulerer!**  
Registreringen var vellykket

Du kan laste ned kortet til mobil-lommeboken din ved å trykke på knappen under.

G Pay | Lagre på telefonen

Figur 14: Registreringssiden etter brukeren er registrert

Når man har trykket på knappen og lagt til i sin elektroniske lommebok, kan man se kortet vi designet i 4.1.1, Design av fordelsprogram på figur 15. Vi kan se at de er relativt like hverandre, men ikke perfekt. Skal du da som sluttbruker vise fram kortet ditt for en ansatt, vil du åpne din elektroniske lommebok og finne fram kortet. Dette kan også gjøres uten internett, men da vil ikke kortet bli oppdatert før mobilen er på nett igjen, men alt annet vil bli gjennomført.



Figur 15: Google og Apple pass slik de ser ut i Google Pay og Apple Wallet

## 4.2 Ingeniørfaglige resultater

Målene for dette prosjektet er delt opp i tre deler og er beskrevet i kapittel 1.1 Resultatmål, 1.2 Effektmål, og 1.3 Prosessmål.

### 4.2.1 Resultatmål

Som beskrevet i 1.1 Resultatmål ble resultatmålene våre som følger:

- Digitalisert prosessen for fordelskort ved å installere de i en digital lommebok.
- Tilby produkteier, markedsfirmaer og kunder muligheten til å innhente relevant statistikk og kvalitetsdata for videre utvikling og analyser.
- Kartlagt systemets muligheter for videreutvikling i henhold til produkteiers ønsker.

I vedlegg C Visjonsdokument, definerte vi at en vellykket løsning vil: “Sørge for at kunder som mottar lojalitetskort alltid har dette tilgjengelig.”

Vår løsning digitaliserer fordelskort, slik at sluttbrukere kan få fordelskortet sitt installert i en digital lommebok. Ser vi på eksemplene fra forrige kapittel kan vi se digitale fordelskort i en elektronisk lom-

mebok, som kunne blitt brukt i en reell situasjon. Dermed kan vi anse produktet vi har lagd som en vellykket løsning, og det første resultatmålet regnes også for å være vellykket.

Vår løsning tilbyr også alle superadministrator-, administrator- og kundebrukere statistikk for fordelskortene de har utsendt til sluttbrukere. Statistikken er relevant fordi den tilbyr kundene informasjon de tidligere ikke hadde tilgang til, dersom det samme fordelsprogrammet ble gjennomført med fysiske fordelskort.

Når det kommer til kartlegging av systemets muligheter for videreutvikling, gjør vi nettopp dette her i rapporten i diskusjonskapitlet som følger.

Videre kan vi se på alle user stories som ble beskrevet i Kravspesifikasjonen til prosjektet. Vi delte user stories opp etter systembrukerne: superadmin, admin, kunde og sluttbruker. Dette ser vi oppsummert i tabell 1. Omtrent alle user stories som ikke er dekket omhandler kuponger, hvorfor dette er tilfellet vil bli beskrevet i tilsvarende diskusjonskapittel. Utover dette er nesten all planlagt funksjonalitet dekket.

<b>Bruker</b>	<b>User story dekning</b>
Superadmin	3/4
Admin	5/11
Kunde	5/9
Sluttbruker	4/7

Tabell 1: Dekningsgrad av user stories

Vi kan kort oppsummere at løsningen er en vellykket løsning, men at all ønsket funksjonalitet ikke er innfridd. Vi har lagt et godt grunnlag for videre utvikling, fordi vi har forsøkt å gjøre strukturen på fordelskortene, og den generelle dataen vi lagrer, spesielt fleksibel.

#### **4.2.2 Effektmål**

Effektmålene for prosjektet var skrevet med oppdragsgiver i tankene. Målene ble beskrevet som følger:

- Øke bruken av fordelskort hos små og mellomstore bedrifter.
- Utvide forretningsmessig horisont.

I skrivende stund er det spesielt vanskelig å si i hvilken grad disse målene er/blir oppnådd. Begge målene er avhengig av bruken av produktet vi har utviklet, og fordi vi ikke vil se produktet i bruk før levering av rapport, er det vanskelig å henvise til konkrete resultater av effekten. Vi vil likevel diskutere målene i tilsvarende diskusjonskapittel.

#### **4.2.3 Prosessmål**

I 1.3 Prosessmål satte vi opp to overordnede prosessmål for prosjektet. Ett for oppdragsgiver, RealNett AS, og ett for oss som team og individer.

Prosessmålet til oppdragsgiver var å: “Opparbeide erfaring med bachelorprosessen og bachelorstudenter ved IDI.” Gjennom prosjektet med RealNett AS, har de fått erfaring med bachelorstudenter som de tidligere ikke har hatt. Erfaringen de har fått med seg nå kan legge grunnlag for om oppdragsgiver ønsker å jobbe videre med bachelorgrupper på IDI i fremtiden.

Teamets mål for prosessen var å “Opparbeide verdifulle erfaringer med teknologier, samspill mellom arbeidsgiver og utvikler, teamsamarbeid og utviklingsprosesser.” Teamet og teamets individer har fått

gode erfaringer gjennom bachelorprosessen. Ingen av deltakerene har utviklet for en reell arbeidsgiver tidligere, og bacheloroppgaven har gitt oss et godt innsyn i denne prosessen. Evnen til å dele opp et stort prosjekt i overkommelige deler er en spesielt nyttig evne for systemutviklere. Dette har teamet fått oppleve gjennom bacheloroppgaven også. Vi har alle jobbet med teknologier vi tidligere ikke har brukt, eksempelvis AWS og Vue.js. I tillegg til å være lærerikt, er dette teknologier som brukes hyppig i utviklingsprosjekter.

#### 4.2.4 Tester

Det er ikke gjennomført formelle brukbarhetstester, men tilbakemeldinger fra oppdragsgiver fremgår av møtereferatene i prosjekthåndboken (*Prosjekthåndbok* 2020). Det er skrevet enhetstester for backenden til løsningen. Hvordan testene kjøres, hva som er og ikke er testet, samt testdekning er dokumentert i vedlegg E Systemdokumentasjon, kapittel 9.1.

### 4.3 Administrative resultater

#### 4.3.1 Timeregnskap

Utviklingsteamet har ført timer fortløpende igjennom prosjektet med en liten statusrapport for hver dag. Timelistene finnes i prosjekthåndboken. (sst.) I tabell 2 ser vi antall timer per person ved endt prosjekt.

Person	Timer brukt ved endt prosjekt
William Andersson	491
Øivind Larsson	508
Jonathan Midttun	497
Bjørnar Østveit	528

Tabell 2: Antall timer per person ved endt prosjekt

#### 4.3.2 Fremdriftsplan

Fremdriftsplanen i form av et GANNT-diagram ligger i prosjekthåndboken. (sst.) Her delte vi opp prosjektet i aktiviteter, og planla når og hvor mange timer vi skulle jobbe med hver. I tabell 3 ser vi aktivitetene, og planlagte timer mot faktiske timer.

Aktivitet	Planlagte timer	Faktiske timer
Oppstartsfasen	40	57
Kravspesifikasjon	100	31
Research	300	182
Design	100	19
Utvikling	810	1162
Testing	150	12
Sluttdokumentasjon	500	561

Tabell 3: Planlagte timer mot faktiske timer per aktivitet

I tabell 4 kan du se når vi planla å bli ferdig med de ulike aktivitetene mot når vi faktisk ble ferdig.

Aktivitet	Planlagt avslutningsdato	Faktisk avslutningsdato
Oppstartsfase	07.02	06.02
Kravspesifikasjon	14.02	20.02
Research	24.03	05.04
Design	25.02	20.02
Utvikling	13.05	18.05
Testing	18.05	16.05
Sluttdokumentasjon	20.05	20.05

Tabell 4: Planlagt avslutningsdato mot faktisk avslutningsdato per aktivitet

### 4.3.3 Utviklingsprosess

Gruppen har brukt lean systemutvikling med kanban som prosess, dette innebærer en kontinuerlig flyt av oppgaver. Hva som er gjort, og til hvilken tid fremgår av timelistene med statusoppdateringer i prosjekthåndboken. (*Prosjekthåndbok* 2020)

## 5 Diskusjon

I dette kapitlet vil vi drøfte resultatene vi fremviste i 4 Resultater. Dette vil forekomme i samme rekkefølge som resultatene. Her vil vi også diskutere eventuelle avvik fra planen, og hvorfor de forekom.

### 5.1 Drøfting av vitenskapelige resultater

Som beskrevet i 1.4 Problemstilling er problemstillingen vår som følger:

**Kan digitalisering av fordelskort øke nytteverdien kortene gir for små og mellomstore bedrifter og forbrukere?**

Økt nytteverdi ble beskrevet som forbedring i: Tilgjengelighet, oversikt og logistikk. Vi skal her drøfte hvordan vår løsning med digitale fordelskort bidrar til å forbedre disse punktene, ved å sammenlikne prosessene som må bli gjort med fysiske fordelskort kontra digitale.

#### Tilgjengelighet

Tilgjengelighet er en av de største faktorene for bruk av fordelskort. Som nevnt har fysiske fordelskort et par problemer med tilgjengelighet. Den fysiske plassen som kreves kan være problematisk dersom du melder deg på spesielt mange fordelsprogram. Dette problemet elimineres fullstendig ved digitalisering. Strengt tatt kan en mobil fylles opp av alle kortene som lagres på mobilen, men det er snakk om såpass store mengder med fordelskort at det er lite usannsynlig. For å utnytte fordelskortet sitt må man ha det med seg på stedet. Det er ikke uvanlig å glemme dette fordelskortet, og så ikke få brukt fordelene sine. Med en digital løsning på elektronisk lommebok er alt man trenger tilgang til en mobiltelefon. I vår vestlige verden er det spesielt uvanlig å ikke ha med seg mobilen. Ser vi på et studie fra IDC og Facebook om mobilbruk i USA, viser det seg at 79% av voksne i alderen 18-44 har mobilen sin på seg minst 22 timer om dagen (*Always Connected - How Smartphones And Social Keep Us Engaged* 2013). Den elektroniske lommeboken er omtrent alltid med deg. Lommebokløsningene vi har valgt vil også fungere uten internett, dermed må man ikke ha tilgang til mobildata på stedet. Dersom man går tom for strøm på mobilen, vil derimot ikke digitale fordelskort kunne benyttes. Vi mener at det positive aspektet veier opp for det negative, og digitale fordelskort har en klar fordel på tilgjengelighet, i den vestlige verden.

## Oversikt

Det er under oversikt vi mener at digitale fordelskort har en spesielt stor fordel. Vår løsning tilbyr statistikk som ellers ikke vil være tilgjengelig med et fysisk fordelskort. Utstedere vil nå ha tilgang til informasjon som ville vært spesielt vanskelig å utnytte med fysiske fordelskort. Eksempelvis kan en utsteder se hvor mange stempler kundene deres bruker i gjennomsnitt, eller periodevis pågang for et visst program.

Dersom vi hadde rukket å implementere kupongutsending, hadde det også vært en mulighet for utstedere. Vi kan for eksempel se på en frisørsalong som har et fordelsprogram. De ønsker å gi kupong på billigere hårklipp til alle menn som deltar på fordelsprogrammet, fordi de har hatt lite pågang fra mannlige kunder. Med en digital løsning vil dette være enkelt, fordi kjønnet til brukeren er lagret i databasen, og kupongutsendingen vil kunne filtreres. Dette gir frisørsalongen en mulighet de tidligere ikke hadde hatt, dersom kortet var fysisk. Tilbudet kunne vært det samme, men det er vanskeligere å få informasjonen ut.

## Logistikk

Flere deler av logistikken bak fordelsprogram ser endringer ved vår digitale løsning. Først og fremst er designprosessen annerledes. Designet kan ikke være helt fritt, fordi vi må følge Google Pay og Apple Wallet sine design. Her gjør fysiske fordelskort det bedre, fordi mulighetene for design er endeløse. I vår løsning blir designprosessen forenklet, ettersom man er begrenset i mulighetene sine. Dette gjør at omtrent hvem som helst kan gjennomføre designprosessen enkelt, dersom de får et par retningslinjer på utseendet. I tilfellet av stempelkort er registreringsprosessen også vanskeligere, da man med et fysisk stempelkort kun trenger å plukke opp en lapp. Digitaliseringen gir også en terskel med tanke på at man må oppgi personlig informasjon man muligens ikke ønsker at utstederen skal ha tilgang på, selv om den kan slettes i ettertid. En stor fordel med design på et digitalt fordelsprogram er mulighetene for endringer i ettertid. Viste det seg at den forklarende teksten på kortet var for vag? Da kan man bare sende ut en oppdatering og alle kunder med kortet vil få oppdateringen med en gang.

Distribusjonen av digitale fordelskort gjennomføres også annerledes. Registrering foregår på en statisk URL, som for eksempel kan vises via en QR-kode. Det var også ønskelig å distribuere program som ikke har åpen registrering (som VIP kort) direkte via SMS, men dette rakk vi ikke. Den viktigste forskjellen med distribusjonen er at den kan foregå tross avstand, og kan gjøres helt fjernstyrt. I tillegg er det ikke et behov for å printe ut kortene, man kan ikke gå tomt for digitale fordelskort å gi til kundene sine.

## 5.2 Drøfting av ingeniørfaglige resultater

### 5.2.1 Resultatmål

Som nevnt var kriteriene for suksess, i følge C Visjonsdokument: "Sørge for at kunder som mottar lojalitetskort alltid har dette tilgjengelig." Dette kan vi i ettertid anse som noe manglende, da det er flere kriterier for at produktet skal være en suksess. Det betyr ikke at vi ikke har oppnådd hva vi anser for å være en god løsning, men vi burde vært mer presise fra starten. Først og fremst var det ansett som spesielt viktig for oppdragsgiver at kundene kunne få fleksibel statistikk, så dette skulle blitt vist tydeligere i visjonsdokumentet. Ser vi derimot i resultatmålene våre kommer dette kriteriet bedre frem. I kapittel to av D Kravdokumentasjon snakker vi om oppdragsgivers ønsker og krav. Her nevner vi at de viktigste ønskene og kravene er hvor kommunikasjon og interaksjon med produktet foregår for de forskjellige brukerne, at statistikk er fleksibel og lett tilgjengelig, og hvilke kunder produktet kan appellere til. Vi har gjennom prosjektets løp forsøkt å utvikle produktet med disse ønskene i tankene.



Vi mener at produktet vi har utviklet stemmer godt overens med disse kravene og ønskene, så med resultatmål, visjonsdokument og kravdokumentasjon i tankene anser vi produktet som en suksess.

Vi nevnte user stories i Resultatmål, og at omtrent alle user stories som ikke er fullført faller under kategorien kuponger og eksklusive tilbud. Det er først og fremst verdt å nevne at mye av funksjonaliteten til kuponger og eksklusive tilbud er implementert allerede. Dette har med at vi fokuserte arbeidet vårt på stempelkort først. Stempelkort var det mest omfattende fordelsprogrammet vi så for oss med tanke på funksjonalitet. Vi tenkte derfor at om stempelkort hadde all ønsket funksjonalitet, var grunnlaget for videre utvikling på andre fordelsprogram veldig sterk. Per definisjon er det nå mulig å lage kuponger og VIP-kort i kortdesigneren om vi åpner for det, samt scanning av disse korttypene, men det vil fortsatt kreve at du registrerer deg for programmet via et skjema, som gjør at litt av poenget med disse programmene forsvinner. Originalt var det også et mål å ha distribusjon per SMS, men dette rakk vi dessverre ikke. Det vi ønsker å presisere er at kuponger ikke er tilgjengelig enda fordi vi heller prioriterte større fleksibilitet for videre utvikling. Det samme gjelder andre fordelsprogram som lojalitetskort eller VIP-programmer. Disse har et veldig godt grunnlag for design og funksjonalitet, men distribusjonen og små finpussinger mot de spesifikke fordelsprogrammene mangler.

### 5.2.2 Effektmål

Som nevnt er det vanskelig å si noe om effektmålene til prosjektet i skrivende stund. Dersom vi gjør noen antakelser om fremtiden kan vi vurdere målene. Det er verdt å nevne at dette er i beste fall synsing, og bør tas med en klype salt. Verdt å merke seg at dette bare er antakelser, og ikke 100% sannsynlig.

Dersom vi antar at produktet videreutvikles utover vårt produkt, og oppdragsgiver til slutt ruller det ut, er det klart at den forretningsmessige horisonten til RealNett AS utvides. Som et telekommunikasjonsfirma er ikke digitale fordelskort en del av deres forretningsmodell, så en utrulling av LoyaltyCard som et kommersielt produkt vil utvide deres forretningsmodell.

Det andre målet var å øke bruken av fordelskort hos små og mellomstore bedrifter. Vi kan anta at den økte nytteverdien vi har sett på, kan gjøre fordelsprogrammene mer interessante for bedrifter som tidligere kun ville brukt fordelsprogrammet som en slags reklame. Det er vanskelig å si noe håndfast om tall på potensiell økning, så det velger vi å unngå. Vi tror at økt nytteverdi kan ha en god korrelasjon med økt forbruk, hvertfall med tanke på at interessen for digitale fordelskortløsninger er relativt høy blant små og mellomstore bedrifter.

### 5.2.3 Prosessmål

I resultatkapitlet nevnte vi at vi har utfordret oss selv relativt bra, og har lært mange nye teknologier. I ettertid tror vi at det kunne vært interessant å utfordre oss selv på en annen databaseteknologi enn MySQL. MySQL er databaseteknologien som teamet har jobbet med siden vårt første databaseprosjekt på studiet, og er dermed det eneste et par av oss kan. Tross dette er teamet fornøyd med alt vi har klart å lære oss i løpet av prosjektet, i alt fra teknologier til prosjektskriving, og å jobbe med en reell oppdragsgiver.

## 5.3 Drøfting av administrative resultater

### 5.3.1 Timeregnskap

Arbeidsbelastningen per person for prosjektet skulle i følge fellesveiledningen være på 501,5 timer. (*Fellesveiledning for bacheloroppgaver ved IDI AIT 2020*) Før eksamenen den 18.03.2020 hadde vi andre obligatoriske aktiviteter som gjorde at vi ikke fikk jobbet jevnt med prosjektet i denne perioden. Etter

eksamenen har gruppen jobbet jevnt, med en målsetning om å jobbe åtte timer hver arbeidsdag. Avvik fra dette skyldes for det meste deltidsjobb noen ukedager, og at vi jobbet noen helger for å følge fremdriftsplanen.

### 5.3.2 Fremdriftsplan

I fremdriftsplanen er det noen ganske store avvik mellom hvor mange timer vi planla å bruke, og hvor mange timer vi faktisk brukte per aktivitet. Estimatenes for kalendertid derimot, stemmer ganske godt overens med planen. Videre ser vi en gjennomgang av hver aktivitet med diskusjon om hvorfor de ble eller ikke ble som planlagt.

#### – **Oppstartsfase**

Oppstartsfasen innebar oppstartsmøtet, visjonsmøtet med oppdragsgiver, og utarbeiding av visjonsdokument.

##### **Avvik**

Vi brukte litt mer tid enn planlagt på denne aktiviteten. Dette kommer av at vi forsto tidlig at visjonsdokumentet trengte mer arbeid enn kravspesifikasjonen. Forklaringen kommer i neste avsnitt.

#### – **Kravspesifikasjon**

Kravspesifikasjon innebar å kartlegge kravene og ønskene til oppgavestiller, samt utarbeide kravspesifikasjonsdokumentet.

##### **Avvik**

Her endte vi opp med å bruke en del mindre tid enn forventet. Noe av dette kan tillegges det faktum at oppgavestiller hadde en klar visjon for prosjektet, og at vi derfor tidlig fikk oversikt over krav og ønsker. Vi kunne nok ha brukt mer tid på å utarbeide et mer grundig kravspesifikasjonsdokument, men tidlig i prosjektet satt vi mye hos oppgavestiller og hadde mye god muntlig kommunikasjon, og vi så derfor ikke et sterkt behov for å produsere så mye formell dokumentasjon av kravene. Oppgavestiller ønsket svært mye funksjonalitet og vi visste at vi ikke kom til å ha tid til å lage alt de ville ha. Derfor fokuserte vi tidlig på å gjøre løsningen vår så fleksibel som mulig slik at den enkelt kunne videreutvikles i fremtiden. Vi var generelt ganske usikre på hvor mye funksjonalitet vi kom til å rekke å implementere, og derfor er kravspesifikasjonen litt mer overordnet.

#### – **Research**

Dette innebærer hovedsakelig research på teknologier utviklingsteamet ikke hadde erfaring med fra før, som for eksempel: Apple Wallet API, Google Pay API, AWS, React Native, og Vue.js. Research på teori i forbindelse med sluttrapporten er timeført på *Sluttdokumentasjon*.

##### **Avvik**

Her har vi timeført en del færre timer enn planlagt. For det første, er det vanskelig å anslå hvor mye tid som trengs til å lære seg noe nytt, og vi antok at det kom til å ta ganske mye tid. I tillegg, så går ofte research på teknologi hånd i hånd med utviklingen, og det har nok blitt gjort veldig mye mer research enn hva timetallet antyder, bare at det er ført under *Utvikling* i stedet for.

#### – **Design**

Med design, mener vi planlegging av hvordan brukergrensesnittet skal se ut, og ikke selve frontend-implementasjonen.

##### **Avvik**

Denne aktiviteten har vi såvidt ført timer på, og det er flere grunner til dette. Oppgavestiller hadde en ganske klar visjon for hvordan de ville at designet av brukergrensesnittet skulle se ut, og kunne i tillegg vise til eksempler fra andre løsninger der de ville ha noe tilsvarende. Det var dermed ikke så mye designarbeid som måtte gjøres i dette prosjektet, arbeidet lå mer på implementasjonen.

Bakgrunnen for estimatet var et tidligere prosjekt vi har hatt ved NTNU, der vi trengte flere designiterasjoner før produkteier var fornøyd. I ettertid så ser vi at denne aktiviteten var litt overflødig, og kunne vært slått sammen med *Kravspesifikasjon*.

#### – **Utvikling**

Utviklingsaktiviteten innebærer alt av koding, det vil si, backend, frontend, app. I tillegg, innebærer det også alt av kontinuerlig integrasjon, serveroppsett og utrulling.

##### **Avvik**

Denne aktiviteten ble mer omfattende enn vi hadde sett for oss. Noe av grunnen til dette er at det tok ganske mye tid å få til en fleksibel grunnmur, før vi rakk å ferdigimplementere noe av funksjonaliteten. Etter hvert som vi kom godt i gang med implementasjonen, har vi jobbet mest med utvikling. Noen dager jobbet vi parallelt med utvikling og dokumentasjon (eller en annen aktivitet), og da har nok alle timene den dagen blitt ført på *Utvikling*, selv om andre aktiviteter ble jobbet med i løpet av dagen. I tillegg, var utviklingsgruppen flere ganger trege på å føre timene, slik at hva som ble gjort den aktuelle dagen ikke lenger satt friskt i minnet. Heldigvis var både kildekode og dokumentasjonen versjonskontrollert, så vi kunne gå tilbake å se hvem som hadde gjort hva om nødvendig. Det må likevel understrekes at vi brukte mye mer tid på utviklingen enn planlagt. Grunnen til dette er for dårlig estimering av omfanget av utviklingsdelen, dårlig rapportering av hvilke aktiviteter vi har brukt tid på, og for høye ambisjoner for hvor mye funksjonalitet vi realistisk sett kunne klare å implementere med den tiden vi hadde. Neste gang ville vi prøvd å senke ambisjonsnivået, og vært mer realistiske. Samtidig kunne timelistene blitt ført annerledes for å representere faktisk akkumulert timebruk per aktivitet. Denne aktiviteten ble avsluttet litt senere enn planlagt, men det var svært lite som ble gjort etter planlagt avslutningsdato.

#### – **Testing**

Testing innebærer enhetstesting og brukbarhetstesting.

##### **Avvik**

Dette mener utviklingsteamet at vi brukte mindre tid på enn det vi burde. For det første ble det ikke gjennomført noen formelle brukertester. Mesteparten av brukergrensesnittet er laget for profesjonelle i markedsføringsbransjen, og det var ikke enkelt å få tak i folk fra denne målgruppen under koronavirus-pandemien. Vi fikk derimot dekket noe av behovet for brukertesting gjennom muntlig tilbakemelding fra oppdragsgiver da vi satt ved deres lokaler å jobbet, samt ved videokonferanse da vi måtte ha hjemmekontor. Enhetstesting kom vi senere i gang med enn planlagt, dette har mye med hvor omfattende utviklingsarbeidet ble.

#### – **Sluttdokumentasjon**

Denne aktiviteten innebærer alt arbeid på hovedrapport, prosjekthåndbok og systemdokumentasjon.

##### **Avvik**

Her har vi brukt omtrent så mye tid som vi planla, men vi kom i gang litt senere enn planlagt. Grunnen til dette er at vi fokuserte på å fullføre et fungerende produkt først. Vi har likevel jobbet jevnt med hovedrapporten gjennom prosjekt-perioden, men det ble ikke brukt så mange timer i uka på den før den ble hovedfokuset i mai.

### **5.3.3 Utviklingsprosess**

Opprinnelig planla vi å bruke scrum som utviklingsmetode, og i starten av prosjektet prøvde vi å følge denne metodikken. Vi så derimot tidlig at dette ville bli vanskelig. Grunnen til dette var at det ble vanskelig å planlegge sprinter med konkrete leveranser fordi vi måtte sette oss inn i nye teknologier og

finne ut av begrensningene og mulighetene innefor de to lommebok-teknologiene vi valgte, og dette viste seg å legge sterke føringer for hva vi kunne levere. Det var med andre ord uforutsigbart, spesielt i starten. I tillegg, hadde vi ikke tilgang på en scrum-master i bedriften vi jobbet for. Derfor gikk vi over til å bruke lean systemutvikling med kanban som metodikk.

#### **Hva fungerte bra med metodikken:**

- Metodikken var fleksibel og robust til å håndtere endrende forutsetninger gjennom prosjektet, noe som var viktig på grunn av prosjektets natur.
- Så lenge trello ble brukt aktivt av alle deltakere hadde vi god oversikt over fremdrift, og hva hver enkelt deltaker jobbet med.

#### **Hva fungerte dårlig med metodikken:**

- I perioder var ikke gruppen like flinke med å bruke trello, og da ble det fort uoversiktlig. Da var det vanskelig å vite hva andre jobbet på, og hvordan fremdriften var for ulike deler av prosjektet.
- Når oppgavene flyter kontinuerlig, og det ikke planlegges sprinter i plenum som i scrum, er det litt vanskeligere å estimere når planlagt funksjonalitet blir ferdigimplementert, og litt vanskeligere å koordinere med andre prosjektdeltakere.

## **5.4 Gruppearbeidet**

Vi startet med å planlegge systemarkitekturen i plenum, vi identifiserte delvis uavhengige deler av prosjektet, og hvordan de skulle henge sammen. Deretter, fordelte vi ansvarsområder i henhold til kapittel 3.4, og kunne dermed jobbe individuelt med hvert vårt ansvarsområde med stor grad av selvstendighet. På starten fungerte dette særdeles godt med utviklingsmetoden vi valgte. Senere, ble grensene mellom de ulike delene mer uklare, og vi måtte jobbe mer på tvers av ansvarsområdene. Da hadde det en tendens til å bli mer uoversiktlig. Vi har alle jobbet sammen på prosjekter tidligere, og kom derfor godt overens med hverandre. Vi har jevnt over klart å bli enige om tekniske beslutninger, uten at dette har ført til konflikt innad i gruppen. Generelt har prosessen vært svært lærerik, i og med at alle fikk prøve seg på noe nytt.

### **5.4.1 Kommunikasjon**

Vi har generelt hatt veldig god kommunikasjon med oppdragsgiver. Mye av dette kan tillegges at vi satt ofte ved deres lokaler, og at de var veldig tilgjengelige for spørsmål og tilbakemelding. I tillegg til at vi hadde jevnlig statusmøter med alle deltakere og oppdragsgiver. Denne gode kommunikasjonen ble dessverre ikke opprettholdt ut over prosjektet. Grunnen til dette var at koronavirus-pandemien gjorde at vi måtte ha hjemmekontor de ti siste ukene.

Dermed måtte vi kommunisere digitalt med oppdragsgiver og til dels hverandre store deler av prosjektperioden. Her har vi hatt litt blandede erfaringer. Vi var heldige med at hele utviklingsgruppen bor i samme kollektiv, dermed kunne vi fortsette å jobbe sammen på samme måte selv med hjemmekontor. Det var derimot en tre-ukers periode da alle deltakerene satt hver for seg i sin hjemby, og da måtte vi naturligvis kommunisere digitalt. Da brukte vi Discord, som er en kommunikasjonsplattform som er populær i gaming-miljøet. Denne plattformen lar oss kommunisere over tekst, tale, samt dele skjerm. Dette fungerte veldig bra oss imellom. Kommunikasjonen med oppdragsgiver derimot, ble merkbart dårligere. Vi kommuniserte hovedsakelig med oppdragsgiver over Slack, en plattform som ligner Discord, men uten støtte for tale-kommunikasjon. Vi opplevde at det var en høyere terskel for å ta kontakt hos begge parter,

og dermed kommuniserte vi mindre og sjeldnere enn før. Vi hadde ikke lenger den samme gode kontinuerlige kommunikasjonen som vi hadde før, men heller et par meldinger her og der, samt videomøter en gang i blant. Det viktigste er at vi hadde veldig god kommunikasjon i starten, som gjorde at vi likevel klarte å jobbe selvstendig med prosjektet mot slutten, da vi allerede hadde en veldig god oppfatning av oppdragsgivers ønsker.

## 5.5 Etiske problemstillinger

### 5.5.1 Personvern

En av de viktigste etiske problemstillingene for dette prosjektet er personvern. Dette er fordi at det er et poeng å samle inn person- og kontaktopplysninger for demografisk bruksstatistikk og for å sende ut tilbud og nyhetsbrev. Det er svært viktig at denne informasjonen behandles sikkert og forsvarlig. Vi må derfor sørge for at sensitive personopplysninger lagres sikkert, og at de ikke kan hentes ut av administratorer eller kunder uten videre. I tillegg, er det viktig at denne informasjonen kan slettes dersom forbrukeren ønsker dette i henhold til General Data Protection Regulation (GDPR).

### 5.5.2 Miljømessige konsekvenser

Hvordan denne løsningen påvirker miljø er vanskelig å fastslå, og kunne vært et forskningsprosjekt i seg selv. Dette er fordi løsningen kan redusere produksjonen av fysiske kort, samt logistikken som kreves av hele denne verdikjeden. Dersom løsningen fungerer etter hensikten kan den føre til økt forbruk av varer og tjenester, som trolig vil påvirke miljøet negativt. I tillegg, bruker digital infrastruktur mye energi, og miljøkonsekvensen av dette vil avhenge av hvilke energikilder som brukes til å drive datasentrene. Hvorvidt de totale miljømessige konsekvensene av denne løsningen er positive eller negative kan vi dermed ikke svare på, men de er trolig marginale i den store sammenhengen.

### 5.5.3 Samfunnmessige konsekvenser

Denne løsningen er laget for små- og mellomstore bedrifter, som generelt ikke har nok kapital til å utvikle digitale fordelsprogrammer på egenhånd, slik som mange større aktører allerede har gjort. Vår løsning kan dermed hjelpe SMB-segmentet i konkurransen mot de store etablerte aktørene i markedet. Det kan også gi et sterkt etterlengtet konkurransefortrinn i kampen mindre lokale bedrifter befinner seg i mot de store e-handelsaktørene, ved å gi kundene en grunn til å komme tilbake. Vi mener derfor at vår løsning kan hjelpe lokalt næringsliv og på den måten skape positive ringvirkninger for lokalsamfunn i hele Norge.

## 6 Konklusjon og videre arbeid

### 6.1 Konklusjon

Vi vil anse dette digitaliseringsprosjektet som en suksess. Vi har klart å implementere den funksjonaliteten som oppdragsgiver prioriterte høyest, og utover dette har vi laget en fleksibel grunnmur som enkelt kan utvides i fremtiden for å dekke oppdragsgivers øvrige ønsker. Oppdragsgiver ønsket svært mye funksjonalitet, og det vi ikke rakk å implementere er beskrevet kapittel 6.2 Videre arbeid. Problemstillingen fra kapittel 1.4 lyder som følger:

**Kan digitalisering av fordelskort øke nytteverdien kortene gir for små og mellomstore bedrifter og forbrukere?**

Hvor økt nytteverdi var definert som: økt tilgjengelighet, bedre oversikt, og enklere logistikk.

Ut i fra resultatet og diskusjonen kan man se at vår digitale løsning for fordelskort kan øke nytteverdien etter disse kriteriene på flere måter.

- **Økt tilgjengelighet** fordi fordelskortet alltid er tilgjengelig på mobilen til forbrukeren, og ikke trenger å ta opp fysisk plass i en lommebok. Man risikerer altså ikke å miste eller glemme kortet, slik man risikerer med fysiske kort. Fordelskortet må heller ikke konkurrere om den begrensede plassen i forbrukeren sin lommebok. Dette bidrar til økte fordeler for forbrukeren, som nå får utnytte sine fordelskort i større grad enn før. Bedriftene som utsteder kort kan derfor forvente økt bruk av deres fordelsprogrammer.
- **Bedre oversikt** fordi utstedere av fordelskort får tilgang til data og statistikk de ikke ellers ville hatt. De vil ha muligheten til å vurdere effekten av fordelsprogrammer med et kvantitativt grunnlag, og tilpasse programmene deretter.
- **Enklere logistikk** fordi man ikke trenger å opprettholde et inventar av fysiske kort. I tillegg kan man endre på fordelsprogrammer etter at kort er gitt ut, og kortene oppdateres deretter. Man har altså nye muligheter for kontinuerlig forbedring av fordelsprogrammer på en måte som ikke er mulig med fysiske kort.

Løsningen skal selges som et SaaS-produkt, som betyr at det ikke er noen store forskuddskostnader eller investeringskostnader for bedrifter som ønsker å benytte seg av produktet. Dette gjør at kostnaden blir forutsigbar og overkommelig, og løsningen er derfor spesielt godt egnet for små og mellomstore bedrifter.

## 6.2 Videre arbeid

Systemet er laget med videreutvikling i bakhodet helt fra starten av, og vi mener at vårt resultat er spesielt godt egnet til et videreutviklingsprosjekt. Dette er både fordi at applikasjonen er laget slik at ny funksjonalitet enkelt kan legges inn uten å måtte gjøre strukturelle endringer, og fordi at vi har laget en god og velfungerende infrastruktur for kontinuerlig integrasjon og utrulling som kan brukes videre. Basert på oppdragsgivers ønsker har vi en rekke forslag til ny funksjonalitet som kan bygge videre på det vi har laget:

### – **Distribusjon av kort**

Nåværende funksjonalitet:

- Registreringsskjema med statisk url

Forslag til ny funksjonalitet:

- Distribusjon til eksisterende kortinnehavere via SMS eller push-varsel, gjerne med mulighet til å avgrense hvem som får tilbud basert på kortets fremgang eller basert på demografi. Egner seg godt for eksklusive tilbud som for eksempel kuponger.
- Manuell registrering i adminsidene, lenke til nedlasting kan sendes via SMS eller e-post. Egner seg til eksklusive fordelsprogrammer som for eksempel VIP-kort.
- Registreringsskjema med mulighet til å generere engangs-tokens for å begrense registrerings-tilbudet til de som har fått tilsendt en egen lenke. Egner seg til eksklusive fordelsprogrammer som for eksempel VIP-kort.

### – **Korttyper**

Nåværende funksjonalitet:

- Stempelkort

- Medlemskort
- Kupong (ikke i bruk da det mangler en godt egnet distribusjonskanal)
- VIP-kort (ikke i bruk da det mangler en godt egnet distribusjonskanal)

Forslag til ny funksjonalitet:

- Festivalkort for helt kontaktløs betaling. Kortet burde vise saldo, og kan for eksempel bruke vår eksisterende geolokasjonsløsning til å vises på låst-skjermen på hele festivalområdet. Det vil være nødvendig å implementere en måte for forbrukerne å sette inn penger på kortet; til dette vil vi anbefale at man kan betale inn via Vipps, eller en tilsvarende løsning man kan gjøre rett fra mobilen. Dette er fordi at litt av nytteverdien her sammenlignet med eksisterende løsninger er at man da slipper å stå i kø for å kjøpe og fylle opp et fysisk kort.
- Lojalitetskort med poengopptjening. Her burde poengene vises på kortet. Det kan for eksempel gis et visst antall poeng for å kjøpe spesifikke varer, det må i så fall kunne legges inn varer med antall poeng, som kan velges ved scanning. Forbrukeren må kunne velge å bruke sine opptjente poeng for å motta en fordel. Det kan også være en mulighet å ha flere nivåer avhengig av hvor mange poeng forbrukeren har opptjent, og at de får bedre fordeler på et høyere nivå.

#### – **Tilbud og nyhetsbrev**

Forslag til ny funksjonalitet:

- Mulighet til å sende ut nyhetsbrev eller informasjon om tilbud til eksisterende kortinnehavere via SMS eller push-varsel, gjerne med mulighet til å avgrense hvem det blir sendt ut til basert på demografi. Veldig viktig at forbrukeren kan melde seg av fordelsprogrammet dersom de ikke ønsker å motta reklame på denne måten, da vil kunden imidlertid også miste fordelene. Her er det også viktig at det utvises moderasjon så det ikke oppleves som spam av forbrukerne.

#### – **Statistikk**

Nåværende funksjonalitet:

- Registreringer over tid per kort og aggregert
- Scans over tid per kort og aggregert
- Kjønnfordeling per kort og aggregert
- Aldersfordeling per kort og aggregert

Forslag til ny funksjonalitet:

- Antall aktive kortbrukere over tid per kort og aggregert. Aktiv kan defineres som at kortet har bedt om oppdatering innen en viss periode, på denne måten kan du vite at kortet ikke er slettet.
- Gjennomsnittlig antall scans per kortbruker per kort
- Fordeling mellom iPhone og Android enheter per kort og aggregert
- Mulighet til å filtrere statistikk basert på kjønn, alder eller mobiloperativsystem

## Referanser

- Always Connected - How Smartphones And Social Keep Us Engaged* (2013). URL: <https://www.nu.nl/files/IDC-Facebook%5C%20Always%5C%20Connected%5C%20%5C%281%5C%29.pdf> (sjekket 14.05.2020).
- Beck, Kent mfl. (2001). *Manifestet for smidig programvareutvikling*. URL: <https://agilemanifesto.org/iso/no/manifesto.html> (sjekket 26.03.2020).
- Bloomberg, Jason (2018). *Digitization, Digitalization, And Digital Transformation: Confuse Them At Your Peril*. URL: <https://www.forbes.com/sites/jasonbloomberg/2018/04/29/digitization-digitalization-and-digital-transformation-confuse-them-at-your-peril/#2abf20fa2f2c> (sjekket 13.04.2020).
- Chacon, Scott og Straub, Ben (2014). *Pro Git*. 2nd Edition. Apress. Kap. 1.1 Getting Started - About Version Control.
- Chen, James (2019). «Loyalty Program». I: URL: <https://www.investopedia.com/terms/l/loyalty-program.asp> (sjekket 10.05.2020).
- CodeShip (2020). *What is Continuous Integration?* URL: <https://codeship.com/continuous-integration-essentials> (sjekket 11.05.2020).
- Customer Trust* (2012). URL: <https://www.nielsen.com/us/en/insights/article/2012/consumer-trust-in-online-social-and-mobile-advertising-grows/> (sjekket 12.05.2020).
- Digitalisering i offentlig sektor* (2014). URL: <https://www.regjeringen.no/no/tema/statlig-forvaltning/ikt-politikk/digitaliseringen-i-offentlig-sektor/id2340245/> (sjekket 13.04.2020).
- Digitization, glossary* (2020). URL: <https://www.gartner.com/en/information-technology/glossary/digitization> (sjekket 27.04.2020).
- Dominguez, Alberto (2018). *Digitalization of Business*. URL: <https://ehorus.com/digitalization-of-business/> (sjekket 18.05.2020).
- Dybå, Tore og Dingsøy, Torgeir (2008). «Empirical studies of agile software development: A systematic review». I: T. Dybå, T. Dingsøy, Empirical studies of agile software development: A systematic review, Inform. Softw. Technol. (2008), doi:10.1016/j.infof.2008.01.006.
- Essential Loyalty Marketing Statistics for 2019* (2019). URL: <https://www.saasquatch.com/blog/essential-customer-loyalty-statistics-for-2019/#program> (sjekket 05.10.2020).
- Fakta om små og mellomstore bedrifter* (2020). URL: <https://www.nho.no/tema/sma-og-mellomstore-bedrifter/artikler/sma-og-mellomstore-bedrifter-smb/> (sjekket 29.04.2020).
- Fellesveiledning for bacheloroppgaver ved IDI AIT* (2020). URL: <https://studntnu.sharepoint.com/sites/teamsite2/10817/Dokumenter/Fellesveiledning%20bacheloroppgaver%20ved%20IDI%20AIT.pdf> (sjekket 13.05.2020).
- «How Fear Drives Action» (2020). I: (). URL: <https://www.saasquatch.com/blog/loss-aversion-how-fear-drives-action/> (sjekket 14.05.2020).



- Hsu, Hwai-Jung og Lin, Yves (jul. 2018). «How Agile Impacts a Software Corporation: An Empirical Study». I: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), s. 20–25. DOI: [10.1109/COMPSAC.2018.10197](https://doi.org/10.1109/COMPSAC.2018.10197).
- Hva er Business Intelligence?* (2020). URL: <https://www.visma.no/business-intelligence/hva-er-business-intelligence/> (sjekket 19.05.2020).
- Hva er data mining?* (2020). URL: <https://inspari.no/data/data-mining/> (sjekket 19.05.2020).
- IBM (2020). *Cloud computing: A complete guide*. URL: <https://www.ibm.com/cloud/learn/cloud-computing> (sjekket 11.05.2020).
- Kartlegging og vurdering av stordata i offentlig sektor* (2015). URL: [https://www.regjeringen.no/contentassets/7a30f56668634d8c96ad660f92ffd508/bruk\\_av\\_stordata\\_i\\_offentlig\\_sektor.pdf](https://www.regjeringen.no/contentassets/7a30f56668634d8c96ad660f92ffd508/bruk_av_stordata_i_offentlig_sektor.pdf) (sjekket 19.05.2020).
- Link, Margaret (2017). *Ultimate Guide to Customer Loyalty*. URL: [https://www.slideshare.net/margaretlink/the-ultimate-guide-to-customer-loyalty-in-2017?qid=e7e8506b-32a3-4865-8ce1-ff4664e2f79d&v=&b=&from\\_search=2](https://www.slideshare.net/margaretlink/the-ultimate-guide-to-customer-loyalty-in-2017?qid=e7e8506b-32a3-4865-8ce1-ff4664e2f79d&v=&b=&from_search=2) (sjekket 14.05.2020).
- LinkedIn* (2020). URL: <https://no.linkedin.com/> (sjekket 19.05.2020).
- Loyalty programs to maximize lifetime value* (2020). URL: <https://www.saasquatch.com/blog/loyalty-programs-to-maximize-lifetime-value/> (sjekket 10.05.2020).
- Max Roser, Hannah Ritchie og Ortiz-Ospina, Esteban (2020). «Internet». I: *Our World in Data*. URL: <https://ourworldindata.org/internet>.
- Panagiotidi, Maria (2016). *Psykologien bak lojalitetskort*. URL: <https://medium.com/its-all-in-your-mind/the-psychology-behind-loyalty-cards-5e330e59d568> (sjekket 02.04.2020).
- Parviainen, Päivi mfl. (2017). *Tackling the digitalization challenge: how to benefit from digitalization in practice*. URL: <https://ijispm.sciencesphere.org/archive/ijispm-050104.pdf> (sjekket 15.05.2020).
- Platekompaniet - frakt* (2020). URL: <https://www.platekompaniet.no/faq/gratis-frakt-pa-alle-orderer-over-kr.-1000-/> (sjekket 14.05.2020).
- Prinsipper for digitaliseringsprosjekter* (2016). URL: <https://www.regjeringen.no/no/tema/statlig-forvaltning/ikt-politikk/prinsipper-for-digitaliseringsprosjekter/id2484486/> (sjekket 18.05.2020).
- Prosjekthåndbok* (mai 2020). Se vedlagte zip-fil.
- Prosjektveiviseren* (2020). URL: <https://www.prosjektveiviseren.no/> (sjekket 16.05.2020).
- Realnett* (2020). URL: <https://www.realnett.no> (sjekket 10.05.2020).
- Realnett - utvalgte kunder* (2020). URL: <https://www.realnett.no/utvalgtekunder> (sjekket 10.05.2020).
- Skaug, Ingvald (12. apr. 2013). *Hva er kanban?* URL: <https://www.bouvet.no/bouvet-deler/utbrudd/hva-er-kanban> (sjekket 14.05.2020).

Stoica, Marian, Mircea, Marinela og Ghilic-Micu, Bogdan (2013). «Software Development: Agile vs. Traditional.» I: *Informatica Economica* 17.4, s. 66–67.

*The Loyalty Report 2018* (2018). URL: [https://cdn2.hubspot.net/hubfs/352767/Loyalty%20Report%202018%20Files/Bond\\_TheLoyaltyReport%202018%20US\\_Exec%20Summary.pdf](https://cdn2.hubspot.net/hubfs/352767/Loyalty%20Report%202018%20Files/Bond_TheLoyaltyReport%202018%20US_Exec%20Summary.pdf) (sjekket 10.05.2020).

*The Loyalty Report 2019* (2019). URL: [https://cdn2.hubspot.net/hubfs/352767/TLR%202019/Bond\\_US%20TLR19%20Exec%20Summary%20Launch%20Edition.pdf](https://cdn2.hubspot.net/hubfs/352767/TLR%202019/Bond_US%20TLR19%20Exec%20Summary%20Launch%20Edition.pdf) (sjekket 10.05.2020).

*The Rise of Digital & Mobile Wallets* (2018). URL: <https://merchantmachine.co.uk/digital-wallet/> (sjekket 13.04.2020).

*Trumf.no* (2020). URL: <https://www.trumf.no/> (sjekket 11.05.2020).

## 7 Vedlegg

Merk at hyperlenker ikke fungerer i sammenslåtte pdf-filer. Samtlige vedlegg under kan også finnes i vedlagte zip-fil, der lenkene er klikkbare.

### A Den opprinnelige oppgaveteksten

**Arbeidstittel:**

Loyalty

**Hensikten med oppgaven:**

Hensikten med oppgaven er å utvikle et kommersielt fordelskort (ikke en app) som skal lanseres i hele Norge. Kortet vil gi innehaverne utvalgte fordeler hos utsteder (vår kunde) og kan bestå av digitale stempel, betalingskort for festivaler eller VIP-kort. Det digitale fordelskortet skal installeres på smarttelefoner og skal være basert på digital lommebok. Fordelskortets registreringsprosess samler inn kontaktopplysninger som senere, om sluttbruker godkjenner slik bruk, kan benyttes til utsendelse av eksklusive tilbud og rabatter, og nyhetsbrev eller annen relevant informasjon.

For iOS benyttes Wallet - for Android benyttes tredje-parts elektronisk lommebok tilsvarende iOS' Wallet. Det er et poeng at det ikke er behov for å utvikle en egen app for løsningen.

Markedsundersøkelser viser at produktet passer godt til bedrifter innen salg av tjenester/produkter som har høy omløpshastighet, herunder f.eks kafe og restaurant, nattklubb, idrettsarrangementer, festivaler etc. Muligheten til å utstede fordelskort til sluttbruker, samt ett eller flere designtilpassete fordelsprogram skal selges til bedrifter i SMB-markedet og vil være et supplement til bedriftenes øvrige markedsaktiviteter. Fordelsprogrammet vil gi bedriftene en unik mulighet til å kartlegge og kommunisere med sine faktiske kunder.

**Kort beskrivelse av oppgaveforslag:**

Oppgaven krever god forståelse av det kommersielle aspektet ved vår virksomhet. Vi anbefaler derfor at man benytter seg utstrakt av muligheten til å ha arbeidsplass hos oss. Her får du aktivt være med i diskusjoner av forretningsmessig karakter, samt muligheten til å påvirke beslutninger som vil forme sluttproduktet.

RealNett skal etablere et SaaS produkt som skal håndtere et ubegrenset antall kunder med hver sine unike fordelsprogram. Ved å opprette en kunde i systemet, og legge inn bedriftens grafiske profil i form av farger, logoer, ikoner med mer, vil fordelsprogrammene hente grunndesign fra denne profilen. Uavhengig av kundens grunndesign må det være mulig å gjøre individuelle tilpasninger ved behov.

Hvert fordelsprogram skal ha sitt unike registrerings skjema og datafelter slik at fordelsprogrammene skal kunne driftes parallelt og adskilt fra hverandre. En sluttbruker vil derfor kunne registrere seg i flere fordelsprogram, og vil motta en unik id i hvert program.

Noen stikkord:

- Web-basert administrasjon for vedlikehold og opprettelse av nye kunder og fordelsprogram
- Grafisk grensesnitt - både for utsteder og sluttbruker
- Tilgangsstyring
- Eksport/presentasjon av data - med filtermuligheter basert på tilgangsstyring

(RealNett eller RealNetts kunder)

- Nettbasert registrering av sluttbrukere - automatisk utsendelse/nedlasting av fordelskort
- Utarbeide dokumentasjon for systemets funksjoner
- Bidra med teknisk tilbakemelding for å sikre en effektiv og strategisk riktig utvikling av systemet
- Tilpassing til online betalingstjenester
- Apple Developer sertifikater for Apple Wallet

**Oppgaven passer for (kryss av de(t) som passer og skriv evt. en kommentar til oss):** - Bacheloroppgave

**Kan oppgavestiller stille arbeidsplass med nødvendig utstyr og programvare:** Arbeidsplass, ikke PC og programvare

**Hvis ikke, hva kreves av maskin og programvare:**

Maskin som takler ev programmeringsspråk - studentene kan komme innspill til valg av optimalt programmeringsspråk

**Begrensninger i tilgjengelighet av opplysninger o.l.:** Opplysninger knyttet til prosjektet er å anse som taushetsbelagt for alle involverte om ikke annet er uttrykt

**Oppgaven passer best for, antall studenter:** - 3  
- 4

**Opplysninger om oppgavestiller**

**Navn på bedrift eller organisasjon:** RealNett AS

**Adresse** Vestre Rosten 85

**Postnummer** 7075

**Poststed** TILLER

**Navn på kontaktperson/veileder:** Robert Estenstad

**Telefon:** 94843242

**Epost:** robert@realnett.no

**Navn på kontaktperson 2/veileder 2:** Christian A. Fjeseth

**Telefon kontaktperson 2/veileder 2:** 90074908

**Epost kontaktperson 2/veileder 2:** christian@realnett.no

**Utfyllende kommentarer til hva oppgaven gjelder:**

IPR og ev andre rettigheter knyttet til produktet i sin helhet tilfaller RealNett AS etter endt prosjekt.

For utfyllende opplysninger kan RealNett kontaktes direkte.

Det er ønskelig med to grupper, ev en kombinasjon av to fagretninger. Gjerne en gruppe som jobber med selve utviklingen av systemet og en gruppe som kan jobbe med krysningspunktet mellom system og den kommersielle delen av prosjektet.

## B Kontrakt

# Avtale mellom partene

Bedrift/virksomhet: RealNett

Student(ene): William Andersson, Øivind Larsson, Jonathan Middleton, Bjørnar Østveit  
og NTNU, IDI AIT

Oppg.nr: 62 Studentprosjekt Loyalty

Studieprogram: Dataingeniør

## Gjennomføring

Studenten skal gjennomføre et studentprosjekt i samarbeid med bedriften/virksomheten. IDI AIT veileder arbeidet faglig. Det er utarbeidet retningslinjer for gjennomføring av studentprosjekt som beskriver oppgavefordeling og hvordan studentprosjekter gjennomføres. Retningslinjene tar også opp ansvarsfraskrivelse, opphavsrettigheter og tilgjengelighet med muligheter for individuelle avtaler.

## Ansvarsfraskrivelse

Instituttet er ikke ansvarlig for eventuelle ødeleggelser som studenten måtte påføre oppgavestillers utstyr direkte eller som følge av programvare studenten lager og/eller bruker, eller som studenten på annen måte medvirker til.

## Opphavsrett og tilgjengelighet

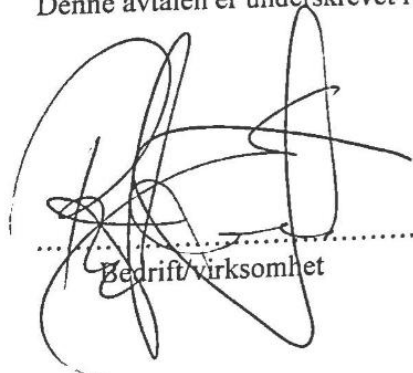
Når ikke annet er avtalt, eier studenter selv den IPR (immaterielle rettigheter) de skaper som en del av studier/studieopphold ved IDI AIT. Alle resultater er åpent tilgjengelig. Opphavsretten reguleres av Åndsverksloven. Avtaler som inngås mellom IDI AIT og studenter skal som minimum sikre instituttet rett til å bruke generert IPR til utdannings- og forskningsformål. IDI AIT skal også motta en vurderingskopi av arbeidet inkludert eventuell kildekode. NTNU oppfordrer sterkt til publisering av alle bachelor- og masteroppgaver.

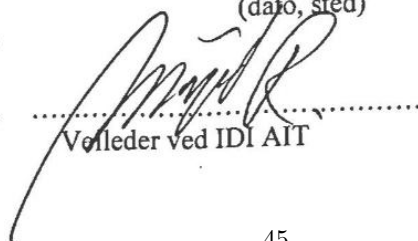
Marker med kryss det som gjelder denne oppgaven:


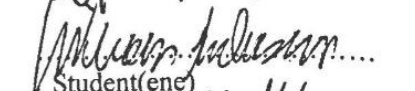
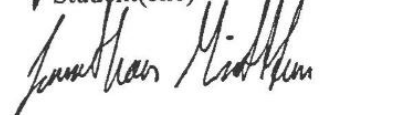
- Normalsituasjonen: Studentene har selv alle rettigheter knyttet til resultatet fra bacheloroppgaven, med de unntak som er beskrevet over.
- Oppdragsgiveren har rettighetene og kan utnytte produktet kommersielt og videreutvikle produktet/metoden. Instituttet vil ikke utnytte produktet kommersielt, men vil kunne arbeide videre med den grunnlagskompetansen som er vunnet gjennom prosjektet, som beskrevet over.
- Resultatene fra arbeidet legges ut som OpenSource iht lisens \_\_\_\_\_ (Se <http://creativecommons.no/lisenser>).
- Bacheloroppgaven (det skriftlige arbeidet) skal være undergitt utsatt offentliggjøring i \_\_ år (maks 3).

Oppdragsgiver er selv ansvarlig for å avtale håndtering av eventuelle konfidensielle opplysninger med veileder/sensor og studenten(e).

Denne avtalen er underskrevet i 3 – tre - eksemplarer hvor partene skal ha hver sin.

  
Bedrift/virksomhet

23.01.2020 Trondheim  
(dato, sted)  
  
Veileder ved IDI AIT

  
Bjørnar Østveit  
  
Student(ene)  
  
Jonathan Middleton

## C Visjonsdokument

# Visjonsdokument BA Loyalty, Gruppe 062

William Andersson

Øivind Larsson

Jonathan Midttun

Bjørnar Østtveit

Januar/Februar 2020 - v1.2



## Revisjonshistorie

<b>Dato</b>	<b>Versjon</b>	<b>Beskrivelse</b>	<b>Forfatter</b>
03.02.2020	1.0	Visjonsdokument lagd, 1. utkast	William Andersson, Øivind Larsson, Jonathan Midttun, Bjørnar Østtveit
04.02.2020	1.1	Laget funksjonelle egenskaper og innledning, 2. utkast	William Andersson, Øivind Larsson, Jonathan Midttun, Bjørnar Østtveit
17.02.2020	1.2	Redigert ikke-funksjonelle egenskaper, 3. utkast	Øivind Larsson

# Innholdsfortegnelse

<b>1</b>	<b>Innledning</b>	<b>4</b>
<b>2</b>	<b>Sammendrag problem og produkt</b>	<b>4</b>
2.1	Problemsammendrag . . . . .	4
2.2	Produktsammendrag . . . . .	4
<b>3</b>	<b>Overordnet beskrivelse av interessenter og brukere</b>	<b>6</b>
3.1	Oppsummering av interessenter . . . . .	6
3.2	Oppsummering av brukere . . . . .	7
3.3	Brukermiljøet . . . . .	7
3.4	Sammendrag av brukerens behov . . . . .	8
3.5	Alternativer til vårt produkt . . . . .	9
<b>4</b>	<b>Produktoversikt</b>	<b>10</b>
4.1	Produktets rolle i brukermiljøet . . . . .	10
4.2	Forutsetninger og avhengigheter . . . . .	10
<b>5</b>	<b>Produktets funksjonelle egenskaper</b>	<b>10</b>
<b>6</b>	<b>Ikke-funksjonelle egenskaper</b>	<b>11</b>
6.1	Testing . . . . .	11
6.2	Hosting av kode . . . . .	11
6.3	Ytelse . . . . .	11
6.4	Sikkerhet . . . . .	12
6.5	Server . . . . .	12
6.6	Skalerbarhet . . . . .	12
6.7	Innstalling og kjøremiljø . . . . .	12
6.8	Brukskvalitet . . . . .	12
6.9	Språk . . . . .	12
6.10	Sms-tjeneste . . . . .	12

# 1 Innledning

Dette dokumentet beskriver de overordnede kravene og målene til Bachelorprosjektet 062 Loyalty. Hensikten med oppgaven er å utvikle et digitalt kommersielt fordelskort. Kortet vil gi utvalgte fordeler hos utsteder, dette kan for eksempel være et digitalt stempelkort, betalingskort for festivaler, eller et VIP-kort. Fordelskortet skal installeres på smarttelefonen til innehaver i deres digitale lommebok (Apple Wallet eller tilsvarende for Android-enheter). Fordelskortets registreringsprosess skal samle kontaktopplysninger (som sluttbruker må godta) som kan benyttes til utsendelse av eksklusive tilbud eller nyhetsbrev. Hvert fordelsprogram skal ha sitt eget registrerings skjema og de skal være adskilt fra hverandre. Tjenesten skal selges til små og mellomstore bedrifter.

## 2 Sammendrag problem og produkt

### 2.1 Problemsammendrag

I dag er det mange større virksomheter som har utviklet digitale lojalitetsprogrammer for eksempel gjennom en app. Hos mange mindre virksomheter brukes det fremdeles fysiske lojalitetskort som for eksempel stempelkort. Fysiske kort blir ofte tapt eller ikke medbrakt, som fører til at slike fysiske kort er mindre effektive til å få kundene tilbake. For små virksomheter er det ikke lønnsomt å utvikle sin egen digitale løsning. Det er vanskelig å vite hvor effektive lojalitetsprogrammer er når man ikke har mulighet til å få statistikk på bruk.

Problem med	Mange kunder mister eller glemmer å medbringe sitt fysiske lojalitetskort.
berører	Virksomheter som utsteder lojalitetskort og kunder som ønsker å bruke lojalitetskort.
som resultat av dette	Lojalitetsprogrammer er ineffektive til å skape tilbakevendende kunder.
en vellykket løsning vil	Sørge for at kunder som mottar lojalitetskort alltid har dette tilgjengelig.

### 2.2 Produktsammendrag

En webbasert plattform som gir markedsfirmaer muligheten til å designe digitale lojalitetskort for sine kunder. Virksomheter lar kunder registrere seg for å få kortet, kundene får dette til sin mobil-lommebok (e.g. Apple Wallet). Deretter kan utsteder scanne kortene ved bruk via en egen app, og kan se statistikk over bruken på webplattformen.

For	virksomheter.
som	utsteder lojalitetskort.
Loyalty	er en webapplikasjon.
som	distribuerer digitale lojalitetskort for virksomheter som scannes ved bruk.
I motsetning til	at kundene får fysiske kort som må medbringes og stemples.
Er vårt produkt	digitalt, tilgjengelig på mobil, og gir virksomheter statistikk på kunders bruksmønster.

### 3 Overordnet beskrivelse av interessenter og brukere

#### 3.1 Oppsummering av interessenter

Navn	Utdypende beskrivelse	Rolle under utviklingen
RealNett AS	De er firmaet som sendte inn bacheloroppgaven til NTNU. Oppgaven er en ide de vil utvikle, men ikke har kapasitet til. De vil ha rettighetene til sluttproduktet og kan utnytte dette kommersielt, derfor er de interessert i at sluttproduktet er klart for kommersielt bruk.	De vil representere produkteier i utviklingsprosessen. Dermed vil de ha en sentral rolle fordi det er deres visjoner produktet skal bygges på.
NTNU, IDI	Dette er en bacheloroppgave og dermed er IDI en interessent. De vil se sluttproduktet vårt og vurdere det, men er ikke interessert i å utnytte produktet kommersielt.	Vi har en veileder fra IDI som skal være til stede under prosessen og hjelpe oss til å nå vurderingskriteriene. Veilederen vil ikke hjelpe til med koding og selve sluttproduktet, men dokumentasjon og prosess.
Markedsfirmaer	Representerer firmaer som kan videreselge systemet til sine kunder etter behov. De er interessert i sluttproduktets verdi ved videresalg.	De vil ikke ha en rolle under utviklingen.
Kunder	Representerer virksomheter som betaler for produktet. Dette kan kjøpes igjennom markedsfirmaer eller direkte fra RealNett AS. De er interessert i hvordan produktet fungerer i praksis og om det er gunstig for deres bruk.	De vil ikke ha en rolle under utviklingen.
Brukere	Representerer personene som installerer Loyalty-kort og bruker dette. De er kun interesserte i at produktet er gunstig for deres bruk.	De vil ikke ha en rolle under utviklingen.

### 3.2 Oppsummering av brukere

Navn	Utdypende beskrivelse	Rolle under utviklingen	Representert av
Superadmin	Full kontroll av systemet, kan administrere og opprette kontoer for markedsfirmaer og deres kunder, samt sine egne kunder. Lager Loyalty-kort for sine kunder og har mulighet til det samme for markedsfirmaenes kunder.	Produkteier	RealNett AS
Admin	Opprette og administrere kundekontoer, lager og sender ut Loyalty-kort etter sine kunders behov.	Ingen	Markedsfirmaer
Kunde	Skanner Loyalty-kort ved bruk i deres virksomhet. Kan logge inn og se statistikk på bruk av Loyalty-kort.	Ingen	Utsteder - Virksomheter som bruker Loyalty-kort.
Bruker	Innehaver av Loyalty-kort, kan registrere seg og laste ned Loyalty-kort.	Testing	Hvem som helst med tilgang til Apple Wallet eller Passbook

### 3.3 Brukermiljøet

Brukermiljøet til produktet vil variere basert på bruker. Superadmin og admin vil bruke valgfri nettleser på PC, da produktet leveres webbasert. Løsningen må dermed fungere i de vanligste nettleserene. Det er ikke ønskelig at administrerende behov gjennomføres på mobil.

Kunder vil ønske å kunne gjennomføre scanning av kort på mobil/nettbrett, via en app koblet mot applikasjonens serverside.

Sluttbrukerene derimot, vil kunne bruke det webbaserte systemet for å registrere seg og logge inn for å se en oversikt over sine Loyalty-kort og vil ha muligheten for å legge de til i sin Apple Wallet eller Passbook. Derfor er det ønskelig at denne funksjonaliteten er optimalisert for mobile enheter.

### 3.4 Sammendrag av brukerens behov

Behov	Prioritet	Vedrører	Dagens Løsning	Foreslått løsning
<b>Superadmin/Admin</b>				
Opprette Admin (Superadmin)	Medium	Struktur	Ingen	RealNett AS vil kunne opprette nye underadminer som kan opprette sine egne kunder.
Lage Loyalty-kort	Høy	Loyalty-kort	Ingen	Kunde sender inn ønskede spesifikasjoner, Superadmin/Admin lager kortet i webgrensesnitt
Utsende kuponger til spesifikke Loyalty-kort	Medium	Loyalty-kort	Ingen	Kunde sender inn ønskede spesifikasjoner, Superadmin/Admin lager kupongen i webgrensesnitt
Opprette konto for kunde	Høy	Struktur	Ingen	Kunde kontakter Superadmin/Admin med spesifikasjoner.
<b>Kunde</b>				
Scanne QR-kode	Høy	Loyalty-kort	Ingen/Stemplestempelkort	Bruker finner fram kortet sitt i elektronisk lommebok, og får en kode scannet av kasseoperatør hos Kunde.
Se statistikk for Loyalty-kort og kuponger	Høy	Statistikk	Ingen	Kunde vil kunne, i en ønsket periode, se statistikk for bruken av sitt Loyalty program. Statistikk som f. eks. hvor mange stemplinger, hvor mange nedlastinger, hvor mange kuponger brukt.
Sende forespørsel om Loyalty-kort	Lav	Loyalty-kort	Ingen, bruker e-post e.l for kommunikasjon	Tross at Superadmin/Admin direkte kan gjøre dette, kan det være nyttig at man sender inn forespørsler direkte via et skjema på web.

Behov	Prioritet	Vedrører	Dagens Løsning	Foreslått løsning
Skille mellom ansatte	Lav	Statistikk	Ingen	Gjennom sin konto kan kunde skille mellom ansatte slik at de kan fordele statistikken.
<b>Bruker</b>				
Registrere/Laste ned Loyalty-kort	Høy	Loyalty-kort	Ingen	Fra kundes registreringsside vil bruker kunne utsendes en direkte nedlasting eller en SMS med URL for nedlasting.
Laste ned kupong	Medium	Loyalty-kort	Ingen	Ut fra kundes spesifikasjoner, vil spesifikke kort få pushvarsler eller SMS om at de har en tilgjengelig kupong, og disse kan lastes ned direkte på kortet i den elektroniske lommeboken.
Se og slette personlig data	Medium	Personlig data	Ingen	Fra GDPR må det være mulighet for å se og slette all personlig data lagret om sluttbrukeren.

### 3.5 Alternativer til vårt produkt

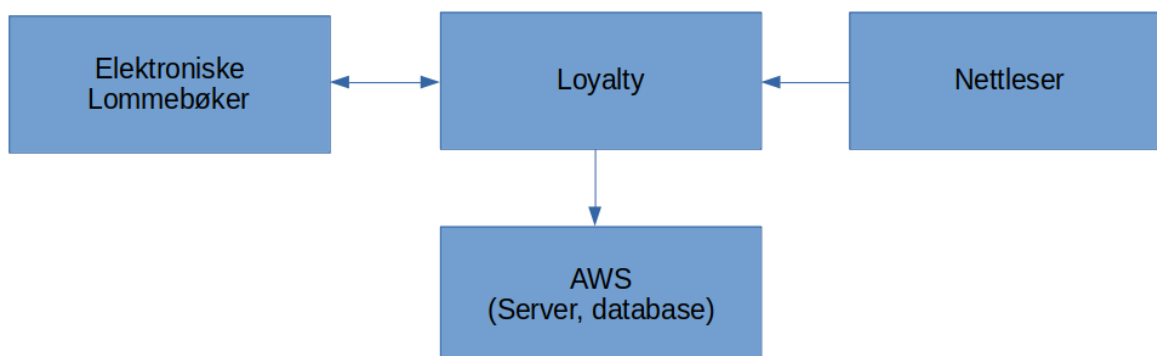
Lokalt (i Norge) finnes det ingen direkte alternativer, annet enn digitaliserte medlemskort og kupong-samlinger som eksisterer som egne apper. F.eks. [Coop Medlem app](#) og [Studentkalender app](#).

Ellers vil alternativene være medlemskort, stempelkort, kuponger, og andre fordelskort i fysisk form.



## 4 Produktoversikt

### 4.1 Produktets rolle i brukermiljøet



### 4.2 Forutsetninger og avhengigheter

Forutsetter at kunder som ønsker å laste ned et lojalitetskort har en iPhone med IOS 6 eller nyere, eller en Android enhet med en tredjeparts lommebok-app. Forutsetter at virksomheter som utsteder kort har tilgang på en mobil enhet til å scanne kort med. Forutsetter at brukere gir samtykke for bruk av personlig data (GDPR), og har e-post og telefonnummer til å unikt identifisere brukeren.

## 5 Produktets funksjonelle egenskaper

Funksjonelle egenskaper
Superadmin kan opprette adminbrukere.
Superadmin kan slette sensitiv brukerinformasjon.
Superadmin kan se all statistikk om Loyalty-kortene.
Superadmin kan se all statistikk om eksklusive tilbud, kuponger og nyhetsbrev.
Admin kan designe Loyalty-kort etter kundens spesifikasjoner.
Admin kan opprette Loyalty-kort.
Admin kan opprette VIP Loyalty-kort.
Admin kan utsende Loyalty-kort til spesifikke brukere.
Admin kan utsende Loyalty-kort til kunder.
Admin kan opprette kuponger etter kunden spesifikasjoner.
Admin kan utsende kuponger til spesifikke Loyalty-kort.
Admin kan utsende eksklusive tilbud og/eller nyhetsbrev til brukere etter forespørsel fra kunde.

Admin kan opprette kundebrukere.
Admin kan se statistikk for Loyalty-kortene de har utsendt.
Admin kan se statistikk om eksklusive tilbud, kuponger og nyhetsbrev hos deres kunder.
Admin skal kunne bytte språk i webgrensesnittet.
Kunder kan scanne QR-kode fra brukere.
Kunder kan se statistikk for deres Loyalty-kort.
Kunder kan se statistikk for deres kuponger.
Kunder kan sende inn forespørsler om Loyalty-kort til admin.
Kunder kan sende inn forespørsler om kuponger til admin.
Kunder kan sende inn informasjon om eksklusive tilbud og/eller nyhetsbrev til admin
Kunder kan skille mellom ansatte ved scanning av QR-koder.
Kunder kan filtrere statistikk på ansatte.
Kunder kan digitalt stemple kuponger fra brukere.
Brukere kan registrere seg for Loyalty-kort.
Brukere kan motta lenke med nedlasting for sitt Loyalty-kort.
Brukere kan laste ned kuponger.
Brukere kan se stemplene på sine Loyalty-kort.
Brukere kan se personlig data.
Brukere kan be om å slette personlig data.

Merk at Superadmin innehar samme rettigheter som Admin.

## 6 Ikke-funksjonelle egenskaper

### 6.1 Testing

Enhetstesting må være implementert på serversiden. Ønsket kodedekning er over 50%.

### 6.2 Hosting av kode

All kode skal lagres på GitHub.

### 6.3 Ytelse

Løsningen bør gi en tilgjengelighet på minst 99.99% og en responstid på under ett sekund med 1000 samtidige brukere.

## 6.4 Sikkerhet

Minstekrav for sikkerhet er at løsningen følger OWASP (2013) A1 og A3.

## 6.5 Server

Løsningen skal lanseres på Amazon Web Services.

## 6.6 Skalerbarhet

Løsningen skal kunne dynamisk skalere opp og ned avhengig av pågang.

## 6.7 Innstallering og kjøremiljø

En plattformuavhengig nettleser med støtte for HTML5-standarden må kunne brukes som klient mot applikasjonen. Løsningen skal være optimalisert for mobile enheter for minimum kunder som laster ned kort, og virksomheter som utsteder kort. Innstallering av Loyalty-kort vil foregå via en nedlastingslenke som sluttbruker får tilsendt på sms. Nedlastingslenken skal bli ugyldig etter en nedlastning. Viktig at nedlastingslenken kun kan benyttes på kundens registrerte enhet.

## 6.8 Brukskvalitet

Løsningen skal ha god brukskvalitet, spesielt med tanke på sluttbrukere som laster ned kortene, og virksomheter som utsteder kortene.

## 6.9 Språk

Løsningen og tilhørende kodedokumentasjon skal skrives på engelsk. Dokumenter som hovedrapport, kravdokumentasjon og systemdokumentasjon skal skrives på norsk. WEB-løsningen skal ha mulighet for både norsk og engelsk.

## 6.10 Sms-tjeneste

Løsningen skal ha en automatisert sms-tjeneste gjennom Sveve AS sitt sms-API. Se deres nettside [Sveve.no](http://Sveve.no) for mer informasjon

## D Kravdokumentasjon

# Kravdokumentasjon BA Loyalty, Gruppe 062

William Andersson

Øivind Larsson

Jonathan Midttun

Bjørnar Østtveit

April 2020 - v1.2

## Revisjonshistorie

<b>Dato</b>	<b>Versjon</b>	<b>Beskrivelse</b>	<b>Forfatter</b>
04.02.2020	1.0	Kravdokumentasjon lagd, 1. utkast	William Andersson, Øivind Larsson, Jonathan Midttun, Bjørnar Østtveit
15.04.2020	1.1	Oppdatert ER-modell og Systemarkitektur	William Andersson, Øivind Larsson, Jonathan Midttun, Bjørnar Østtveit
29.04.2020	1.2	Oppdatert ER-diagram og Systemarkitektur	William Andersson

# Innholdsfortegnelse

<b>1</b>	<b>Introduksjon</b>	<b>4</b>
<b>2</b>	<b>Oppdragsgivers Ønsker og Krav</b>	<b>4</b>
<b>3</b>	<b>User Stories</b>	<b>4</b>
<b>4</b>	<b>Systemarkitektur</b>	<b>7</b>
4.1	Systemarkitekturdiagram . . . . .	7
4.2	ER-Diagram . . . . .	8
<b>5</b>	<b>Prototyper</b>	<b>9</b>
5.1	Wireframes . . . . .	9

# 1 Introduksjon

Dette dokumentet skal gi leseren en oversikt over systemets arkitektur, kommunikasjonsflyt og kravene rundt produktet. Det skal vise hvilke muligheter man, som bruker av systemet, ønsker å kunne gjennomføre. Dokumentet kan ses på som utviklernes utarbeiding av hva som ønskes å lage, ut i fra oppdragsgivers ønsker og krav.

## 2 Oppdragsgivers Ønsker og Krav

Produktet skal ikke designes som en app, men en web-app. Super-administratorer, administratorer og kunder skal kunne gjøre alt administrativt gjennom denne web-appen. I tillegg skal kunder ha en egenstående app som brukes til scanning av Loyalty-kort. Sluttbrukere ønskes ikke som brukere av web-appen, med unntak av registrering til Loyalty-programmer. Alt av sluttbrukerkommunikasjon utover dette vil bli gjort via. SMS og Google Pay/Apple Passes.

Statistikk står sentralt for Loyalty, og det skal være den enklest tilgjengelige informasjonen. Det er viktig at statistikken både administratorer og kunder kan se, er lett tilgjengelig og fleksibel. Sentral statistikk omhandler bruk av Loyalty-kort og kuponger. Dette vil eksempelvis være, antall ny-opprettede Loyalty-kort i en periode eller hvor mange stemplinger som er gjort i en periode.

Produktet er lagd med en spesifikk kundegruppe i tankene. Det er ikke slik at Loyalty vil være den absolutt beste mulige løsningen for en hver kunde, i et vakuum. Det er derimot slik at Loyalty er såpass fleksibelt, at det kan treffe mange kunder, som ellers ikke ville kunne ha råd til å produsere sitt helt egne system, da dette er en stor økonomisk investering. Eksempler på kunder som står sentralt er kaffebarer, som kan ha stempelkort eller lojalitetsprogram eller klubber som kan ha VIP-kort.

## 3 User Stories

<b>Som superadministrator</b>
Ønsker jeg å opprette administratorbrukere
Slik at jeg kan gi andre tilgang til systemet for behandling av Loyalty-kort
Ønsker jeg å slette sensitiv brukerinformasjon
Slik at GDPR opprettholdes og brukere kan slette informasjon lagret i databasen
Ønsker jeg å se statistikk om Loyalty-kortene
Slik at jeg kan kartlegge bruken og planlegge videre utvidelser
Ønsker jeg å se statistikk om eksklusive tilbud, bruk av kuponger og nyhetsbrev
Slik at jeg kan gi kartlegge bruken og planlegge videre utvidelser



<b>Som administrator</b>
<p>Ønsker jeg å designe Loyalty-kort etter kundes spesifikasjoner</p> <p>Slik at jeg kan møte kundens krav</p>
<p>Ønsker jeg å opprette Loyalty-kort</p> <p>Slik at jeg har mulighet til å utsende de til kunder</p>
<p>Ønsker jeg å opprette VIP Loyalty-kort</p> <p>Slik at jeg har mulighet til å opprette kort som inneholder ekstra fordeler</p>
<p>Ønsker jeg å utsende Loyalty-kort til spesifikke brukere</p> <p>Slik at jeg kan sende ut VIP Loyalty-kort til riktig person</p>
<p>Ønsker jeg å utsende Loyalty-kort til kunder</p> <p>Slik at kundene kan åpne opp registrering for brukere</p>
<p>Ønsker jeg å opprette kuponger</p> <p>Slik at kunden kan gi brukerne tilbud utenom Loyalty-kortets fordeler</p>
<p>Ønsker jeg å utsende kuponger til spesifikke Loyalty-kort</p> <p>Slik at brukerne mottar kuponger de kan bruke hos kunden</p>
<p>Ønsker jeg å utsende eksklusive tilbud og nyhetsbrev til brukere</p> <p>Slik at brukerne kan benytte seg av de hos kunden</p>
<p>Ønsker jeg å se statistikk for Loyalty-kortene jeg har utsendt</p> <p>Slik at jeg ser til hvilken grad kundene og brukerne benytter seg av Loyalty-kortene</p>
<p>Ønsker jeg å se statistikk om eksklusive tilbud, kuponger og nyhetsbrev</p> <p>Slik at jeg ser til hvilken grad kundene og brukerne benytter seg av dette</p>
<p>Ønsker jeg å opprette kundebrukere</p> <p>Slik at jeg kan gi mine kunder tilgang til systemet</p>

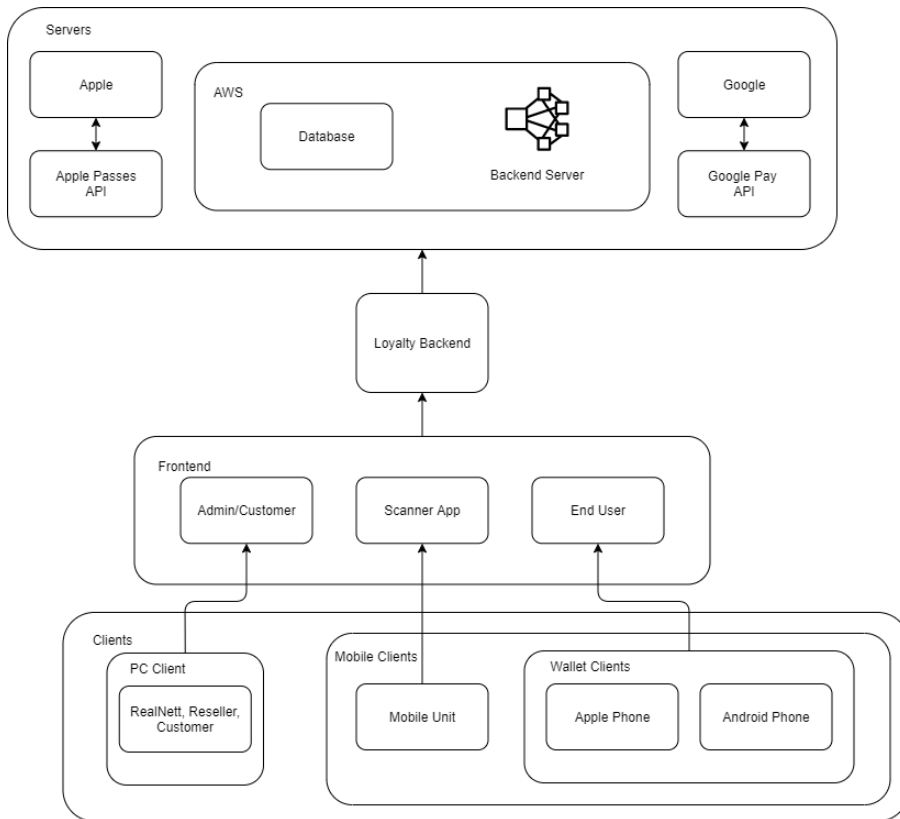
<b>Som kunde</b>
<p>Ønsker jeg å scanne QR-koder fra brukere via. en egen app</p> <p>Slik at jeg kan scanne deres Loyalty-kort</p>
<p>Ønsker jeg å se statistikk for mine Loyalty-kort</p> <p>Slik at jeg kan kartlegge bruken</p>
<p>Ønsker jeg å se statistikk om kuponger</p> <p>Slik at jeg kan kartlegge bruken</p>
<p>Ønsker jeg å sende inn forespørsler om Loyalty-kort til admin</p> <p>Slik at jeg kan stille krav til design og admin kan utsende de til brukere</p>
<p>Ønsker jeg å sende inn forespørsler om kuponger til admin</p> <p>Slik at de kan utsende de til mine brukere</p>
<p>Ønsker jeg å sende inn informasjon om eksklusive tilbud og nyhetsbrev til admin</p> <p>Slik at de kan utsende de til mine brukere</p>
<p>Ønsker jeg å filtrere statistikk på ansatte</p> <p>Slik at jeg kan se hvilke ansatte som bruker fordelsprogrammet mest</p>
<p>Ønsker jeg å skille mellom ansatte ved scanning av QR-koder</p> <p>Slik at det alltid er en ansvarlig for hver scanning</p>
<p>Ønsker jeg å digitalt stemple kuponger fra brukere</p> <p>Slik at brukere ikke kan benytte seg av kuponger flere ganger</p>

<b>Som sluttbruker</b>
Ønsker jeg å registrere meg for Loyalty-kort Slik at jeg kan benytte meg av fordelsprogrammer
Ønsker jeg å motta nedlastninglenke for mitt Loyalty-kort Slik at jeg enkelt kan installere det i min Wallet
Ønsker jeg å motta kuponger og eksklusive tilbud Slik at jeg kan benytte meg av de
Ønsker jeg å laste ned kuponger Slik at utsender kan stemple de ved bruk
Ønsker jeg å se informasjon om mitt Loyalty-kort Slik at jeg kan ha oversikt over bruken
Ønsker jeg å kunne se min personlige informasjon Slik at jeg har oversikt over informasjonen utsender har lagret om meg
Ønsker jeg å be utsender om å slette min personlige data Slik at dataen blir slettet fra deres database

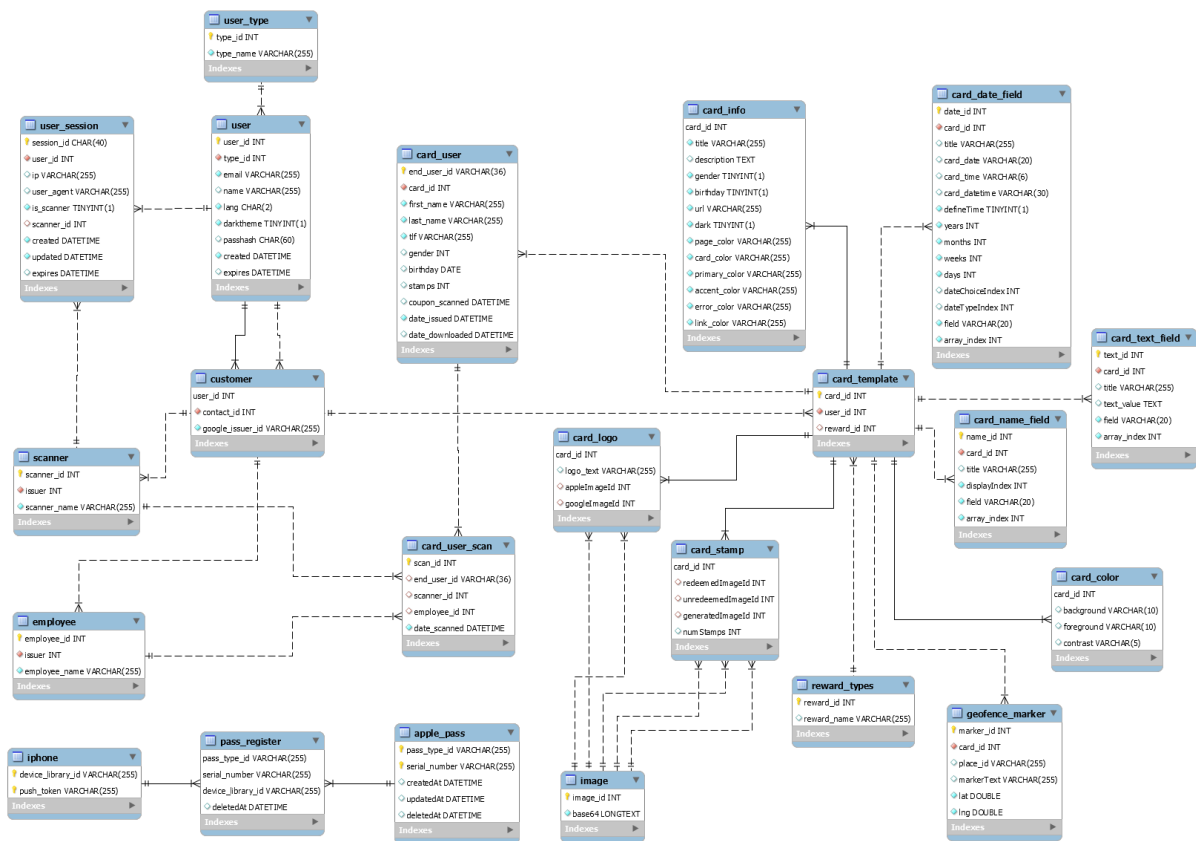
## 4 Systemarkitektur

### 4.1 Systemarkitekturdiagram

Visuell avbildning av kommunikasjonsflyten for prosjektet. Pilene viser hvilken vei kommunikasjonen går. Eksempelvis er det Apple Passes API som snakker med vår back-end.



## 4.2 ER-Diagram



## 5 Prototyper

### 5.1 Wireframes

Et par enkle illustrasjoner på hvordan det generelle utseende vil bli værende. I illustrasjonene er Superhero Burger brukt som eksempel på kunde.

**SUPER HERO  
SHB  
BURGER** Registrer ditt loyalty-kort hos {customer.name}

Mobilnummer:\*

E-post:\*

Denne informasjonen er ikke nødvendig for registrering, men vil kunne gi deg bedre, målrettede tilbud!

Kjønn:

Alder:

Jeg samtykker til at min personlige data lagres, og kan brukes til å markedsføre målrettet mot meg

Her ser vi eksempel på hvordan registreringssiden til en kunde kunne sett ut. Registreringssidene vil inneholde logoen til kunden, og en form for all informasjon fra sluttbruker. Det er ønskelig at, basert på hva kunden kan markedsføre på, at de feltene som ikke kreves kan være fleksible. Det er mer interessant for en frisørsalong å vite kjønnnet til personen som eier Loyalty-kortet, enn en kaffebar. Fargene til siden vil variere basert på spesifikasjon gitt fra kunde, og vil samsvare med den grafiske profilen til kunden. Her er tekst valgt som hvit tekst med svart kant, da dette leses godt på alle farger.



Adminpanelet vil inneholde alt admin kan gjøre. Vi ser det ikke som hensiktsmessig at alt av muligheter på spesifikke kunder ligger i dette overordnede panelet, da man heller bør velge gitt kunde først.

Brukernavn	Brukertype	
bjornost	Superadmin	⋮
williaan	Superadmin	⋮
ReKom	Admin	⋮

Liste over brukere, filtreringsmuligheter skal legges inn.

Navn	Antall Loyalty-kort	
Dromedar kaffebar	20 000 000	⋮
Superhero Burger	20 000 000	⋮

Liste over kunder, filtreringsmuligheter skal legges inn.



## Superhero Burger

20 000 000 aktive kort | 35 748 utsendte kuponger

### Nytt kort

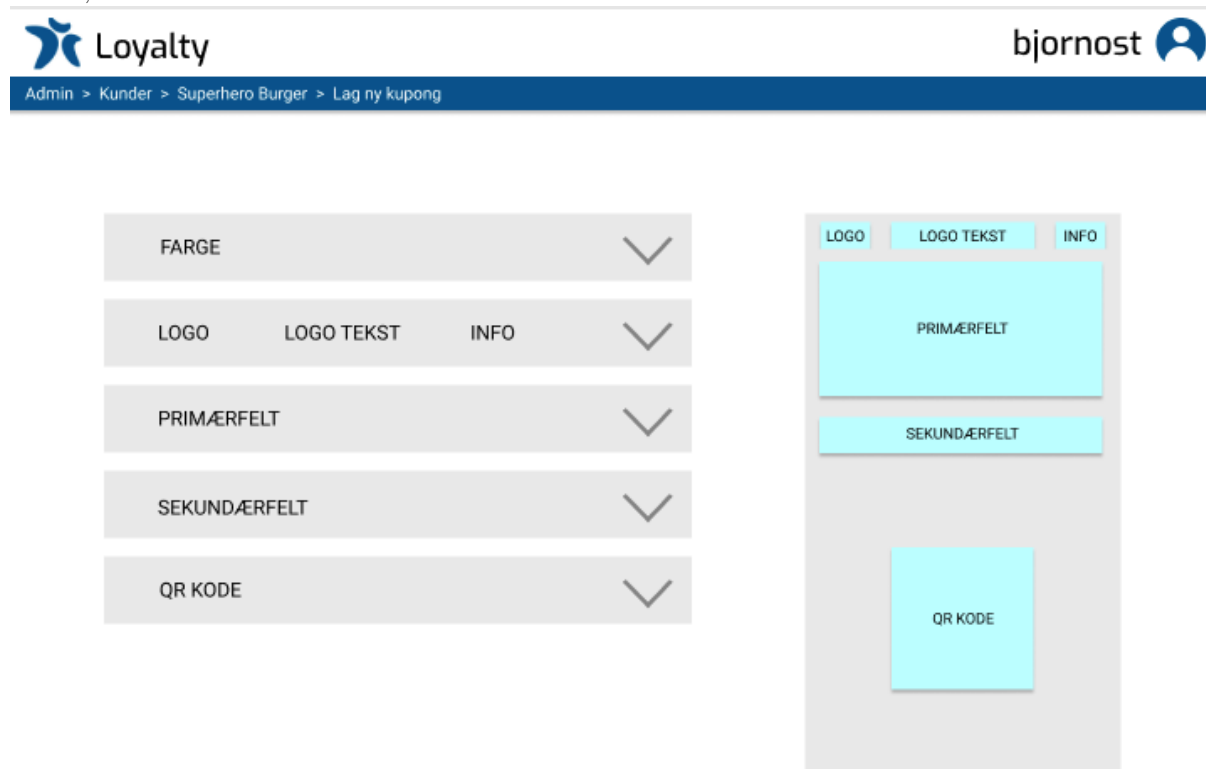


Vis statistikk

Send kupong

Kontakt kunde

Her er et forslag til grensesnitt for én kunde. Her vil man kunne gjøre alt som skulle være nødvendig med en kunde. Dersom man hoverer over et kort vil det kunne trykkes på, og man kan redigere kortet eller se statistikk for det spesifikke kortet. Det kan hende det er hensiktsmessig å ha en liste a la kunde/bruker-listene, slik at man kan filtrere kortene.



Trykket man på nytt kort vil man bli tatt til denne siden. Logo, navn etc. vil automatisk fylles inn, da man kommer fra en kundeside. Samme grensesnitt vil brukes for redigering av eksisterende kort.



## E Systemdokumentasjon

# Systemdokumentasjon BA Loyalty, Gruppe 062

William Andersson

Øivind Larsson

Jonathan Midttun

Bjørnar Østtveit

Mai 2020 - v1.1

## Revisjonshistorie

<b>Dato</b>	<b>Versjon</b>	<b>Beskrivelse</b>	<b>Forfatter</b>
29.04.2020	1.0	Systemdokumentasjon lagd, 1. utkast	William Andersson, Øivind Larsson, Jonathan Midttun, Bjørnar Østtveit
16.05.2020	1.1	Oppdatert kapittel 8 og 9.1	Bjørnar Østtveit

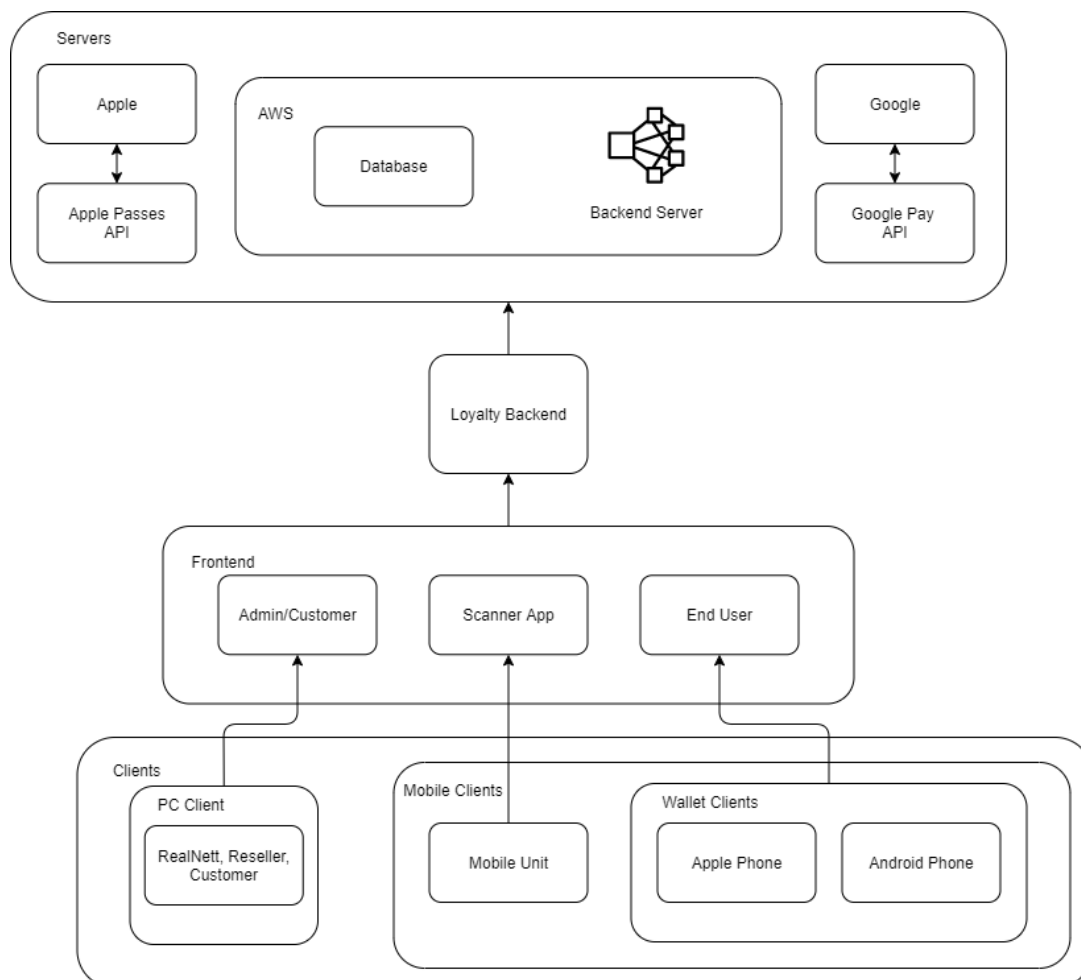
# Innholdsfortegnelse

<b>1</b>	<b>Introduksjon</b>	<b>4</b>
<b>2</b>	<b>Arkitektur</b>	<b>4</b>
<b>3</b>	<b>Prosjektstruktur</b>	<b>4</b>
<b>4</b>	<b>Databasemodell</b>	<b>5</b>
<b>5</b>	<b>Server-tjenester</b>	<b>6</b>
<b>6</b>	<b>Sikkerhet</b>	<b>12</b>
6.1	Kommunikasjon . . . . .	12
6.2	Passordhåndtering . . . . .	12
6.3	Autentisering . . . . .	13
6.4	Autorisasjon . . . . .	13
6.5	Beskyttelse mot angrep . . . . .	13
<b>7</b>	<b>Installasjon og kjøring</b>	<b>13</b>
7.1	Avhengigheter . . . . .	14
7.2	Lokal installasjon . . . . .	14
7.3	Staging/Produksjon . . . . .	17
<b>8</b>	<b>Dokumentasjon av kildekode</b>	<b>18</b>
8.1	Backend . . . . .	18
8.2	Frontend . . . . .	20
<b>9</b>	<b>Kontinuerlig integrasjon og testing</b>	<b>21</b>
9.1	Testing . . . . .	21
9.2	Kontinuerlig integrasjon . . . . .	22
	<b>Referanser</b>	<b>23</b>

# 1 Introduksjon

Dette dokumentet er skrevet i sammenheng med Bacheloroppgaven Loyalty. Dette er en oversikt over de mest sentrale delene av systemet som Loyaltyprosjektet bruker. Det vil blant annet bli gjennomgått: overordnet struktur, API, sikkerhet, tester, CI/CD og hvordan kjøre dokumentasjonen til kildekode.

## 2 Arkitektur



Avbildet her er de viktigste komponentene i systemet, og kommunikasjonen mellom disse. Retningen på pilene henviser til hovedkommunikasjonsvei. De forskjellige delene av frontend er kun tilgjengelig for klientene det er piler fra. For eksempel kan ikke en tilfeldig android telefon ha tilgang til Admin/Customer-paneler. Det er en liten forskjell i kommunikasjon fra Apple og Google: Google har sine egne endepunkter fra Google Pay API for Passes, som du kommuniserer med. Apple derimot ønsker at du lager endepunkter etter deres spesifikasjoner, som de kan kommunisere med. Dette gjør at enkelte ting, som oppdateringer av kort, ikke kan gjennomføres på samme måte for Android telefoner og Apple telefoner. Backend og databasen er hostet på en AWS server.

## 3 Prosjektstruktur

Prosjekt-repoet består i all hovedsak av tre deler: et React Native prosjekt (app), et Express.js prosjekt (backend), og et Vue.js prosjekt (frontend). I tillegg, ligger det en del konfigurasjonsfiler i rotmappa og under `.ebextensions`. Under ser du katalogstrukturen noe forenklet:

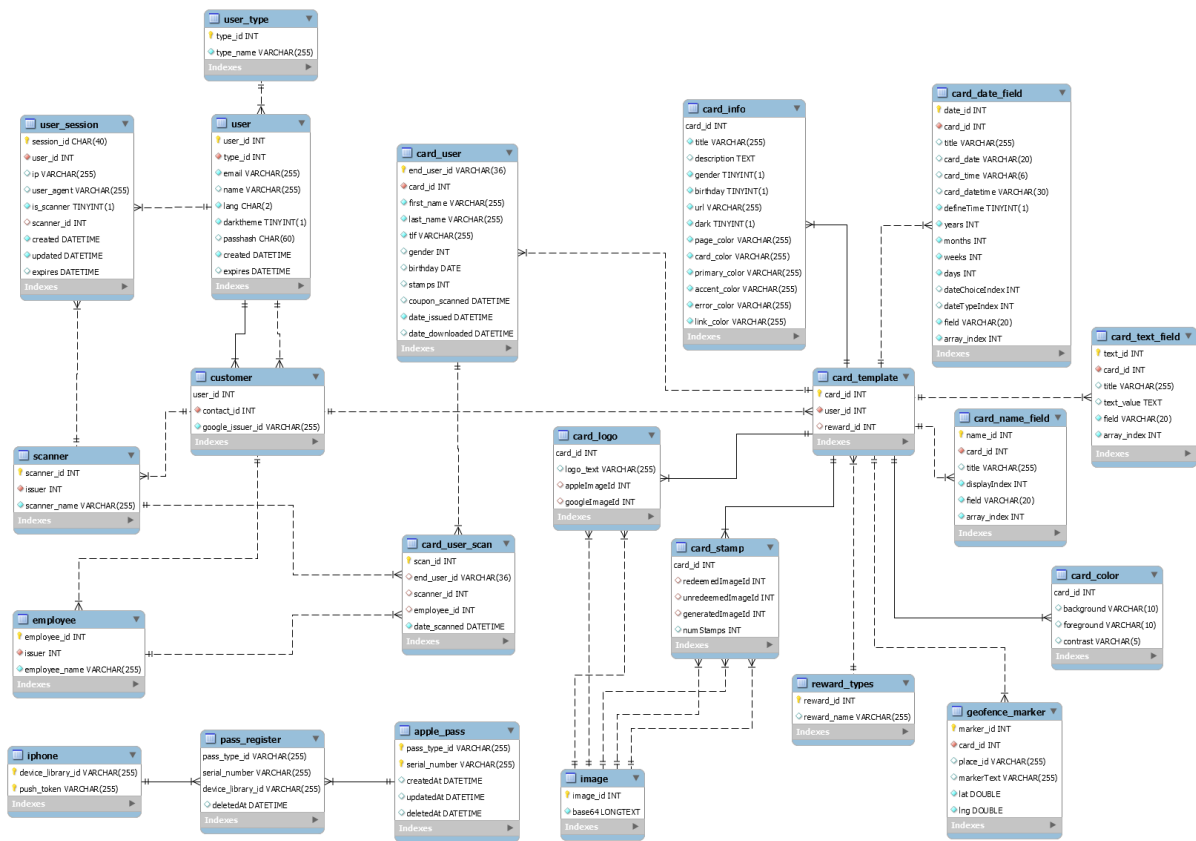
```

LoyaltyCard
├── .ebextensions: konfigurasjonsfiler for AWS Elastic Beanstalk
├── app
│   ├── api: her defineres axios-repositories
│   ├── components: her ligger egenutviklede React Native-komponenter
│   ├── screens: egne React Native-komponenter som utgjør sider i appen
│   ├── util: diverse gjenbrukbare funksjoner
│   ├── App.js: scriptet som starter når appen åpnes
│   ├── environment.js: (ikke i versjonskontroll) definerer api-url for ulike miljøer
│   └── ...
├── backend
│   ├── config
│   │   ├── authorization.js: middleware som håndterer sesjoner
│   │   ├── database.js: Eksporterer sequelize instans
│   │   ├── gapi.authorization.js: middleware som håndterer autentisering mot Google Pay
│   │   └── api
│   │       ├── nocache.js: middleware som hindrer klienten i å lagre responsen i cache
│   │       └── requireLogin.js: middleware som brukes på endepunkter som krever innlogging
│   ├── models: her defineres sequelize-modellene som brukes for tilkobling til database
│   ├── routes: her defineres api-endepunkter
│   ├── tests: her ligger enhetstestene
│   ├── utils: diverse funksjoner
│   ├── app.js: her brukes routes og middlewares for å kjøre applikasjonen
│   └── ...
├── frontend/src
│   ├── api: her defineres axios-repositories
│   ├── components: her ligger egenutviklede Vue-komponenter
│   ├── lang: her ligger oversettelsesfiler
│   ├── store: her defineres vuex-modulene
│   ├── util: diverse gjenbrukbare funksjoner
│   ├── views: egne Vue-komponenter som utgjør selvstendige sider
│   ├── App.vue: rot-komponenten i Vue-frontend
│   ├── main.js: initialiseringsscriptet som instansierer App.vue i nettleseren
│   ├── router.js: definerer url-adresser for de forskjellige sidene
│   └── ...
├── db_setup.sql: brukes for å opprette database-tabellene
├── db_test_data.sql: setter inn test-data i databasen
├── README.md
└── ...

```

## 4 Databasemodell

I figur 1 ser vi en modell av databasen i form av et ER-Diagram. Denne viser struktur og innhold for de forskjellige entitetene vi har i databasen. Lagring av Apple-kort gjøres på de 3 separate tabellene vi ser nederst til venstre. Øverst til venstre ser vi brukerstrukturen. Disse brukerne er forbeholdt administratører og kunder, ikke sluttbrukere. Sluttbrukere lagres i card\_user. Høyre side ser man card\_template, som brukes for å lagre den generelle strukturen til lojalitetsprogram.



Figur 1: ER-Diagram for Loyalty Database

## 5 Server-tjenester

Server-tjenestene består av en omvendt proxy server som kjører NGINX og en backend server som kjører Node + Express. Alle forespørsler går via NGINX og behandles i følge reglene i tabell 1. NGINX håndterer også all SSL/TLS kryptering.

NGINX - Omvendt Proxy	
Endepunkt	Beskrivelse
/*	Frontend: leverer statiske filer for Vue frontend
/api/*	REST API: vidresender forespørsler til Express backend

Tabell 1: NGINX endepunkter

<b>Express - Backend</b>	
/api/apple/v1/	
<b>Endepunkt</b>	<b>Beskrivelse</b>
/devices/:deviceLibraryIdentifier/registrations/:passTypeIdentifier	GET: henter alle serienummer for Apple kort registrert på en iPhone
/devices/:deviceLibraryIdentifier/registrations/:passTypeIdentifier/:serialNumber	POST: registrerer iPhone for varslinger på et kort
/devices/:deviceLibraryIdentifier/registrations/:passTypeIdentifier/:serialNumber	DELETE: avregistrerer iPhone fra å motta varslinger på et kort
/passes/:passTypeIdentifier/:serialNumber	GET: henter nyeste versjon av et Apple kort
/log	POST: skriver feilmeldinger fra Apple til logg

Tabell 2: Express endepunkter under /api/apple/

<b>Express - Backend</b>	
/api/auth/	
<b>Endepunkt</b>	<b>Beskrivelse</b>
/	GET: sjekker autentiserings-status basert på session-cookies
/login	POST: logger inn bruker med e-post og passord
/logout	POST: logger ut bruker og sletter session
/resetpassword	POST: sender passordresett-lenke til e-post
/setpassword	POST: setter et nytt passord med resett-token fra e-post
/password	PUT: endrer passord hvis du oppgir riktig nåværende passord

Tabell 3: Express endepunkter under /api/auth/



<b>Express - Backend</b>	
<i>/api/users/:user_id/cards/</i>	
<b>Endepunkt</b>	<b>Beskrivelse</b>
/	GET: henter alle kortene til en kunde
/	POST: oppretter et nytt kort for en kunde
/:card_id	GET: henter ett spesifikt kort
/:card_id	PUT: oppdaterer ett spesifikt kort

Tabell 4: Express endepunkter under */api/users/:user\_id/cards/*

<b>Express - Backend</b>	
<i>/api/data/</i>	
<b>Endepunkt</b>	<b>Beskrivelse</b>
/	GET: henter ut personopplysninger for kortinnehavere
/:end_user_id	DELETE: sletter alle personopplysninger til en kortinnehaver

Tabell 5: Express endepunkter under */api/data/*

<b>Express - Backend</b>	
<i>/api/endusers/</i>	
<b>Endepunkt</b>	<b>Beskrivelse</b>
/:end_user_id	GET: returnerer et Apple-Wallet kort som kan installeres på IOS enheter

Tabell 6: Express endepunkter under */api/endusers/*

<b>Express - Backend</b>	
/api/google/:user_id/	
<b>Endepunkt</b>	<b>Beskrivelse</b>
/google_card/	
/	GET: returnerer alle Google LoyaltyClass instanser fra en gitt issuer
/	POST: oppretter en Google LoyaltyClass instanse og returnerer denne
/:id	GET: returner en instanse av Google LoyaltyClass
/:id	PATCH: oppdaterer en instanse av Google LoyaltyClass og returnerer denne
/google_card_object/	
/	GET: returnerer alle Google LoyaltyObject instanser under en Google LoyaltyClass
/:object_id	GET: returnerer et Google LoyaltyObject
/:card_id	PATCH: oppdaterer alle Google LoyaltyObject under en Google LoyaltyClass
/:card_id/:object_id	PATCH: oppdaterer ett Google LoyaltyObject
/google_general/	
/issuer	GET: returnerer en liste av alle Google Issuers under eieren av autentiseringstokenet
/issuer	POST: oppretter en instanse av Google Issuer og returnerer denne
/issuer/:issuerId	GET: returnerer én Google Issuer
/issuer/:issuerId	PATCH: oppdaterer én Google Issuer og returnerer denne

Tabell 7: Express endepunkter under /api/google/:user\_id/

<b>Express - Backend</b>	
/api/image/	
<b>Endepunkt</b>	<b>Beskrivelse</b>
/:image_id	GET: henter ett bilde
/	POST: oppretter ett bilde
/stamps	POST: genererer og returnerer et stempelbilde som vises på kort

Tabell 8: Express endepunkter under /api/image/

<b>Express - Backend</b>	
<i>/api/registration/</i>	
<b>Endepunkt</b>	<b>Beskrivelse</b>
<i>/:url</i>	GET: henter nødvendig data for å vise registreringsskjemaet for et lojalitetsprogram
<i>/:url</i>	POST: registrerer en ny sluttbruker i et lojalitetsprogram, og returnerer en lenke for å laste ned kort i mobil-lommebok

Tabell 9: Express endepunkter under */api/registration/*

<b>Express - Backend</b>	
<i>/api/stats/</i>	
<b>Endepunkt</b>	<b>Beskrivelse</b>
<i>/cards/:card_id/registrations</i>	GET: henter alle registreringer (kun dato og tid) for ett spesifikt kort
<i>/cards/:card_id/scans</i>	GET: henter alle scans (dato, tid og ansatt-id) for ett spesifikt kort
<i>/cards/:card_id/gender</i>	GET: henter antall registrerte menn, kvinner, og uspesifisert kjønn for ett spesifikt kort
<i>/cards/:card_id/age</i>	GET: henter antall registrerte innenfor definerte aldersgrupper for ett spesifikt kort
<i>/customers/:user_id/registrations</i>	GET: henter alle registreringer (kun dato og tid) for alle kortene til én spesifikk kunde samlet
<i>/customers/:user_id/scans</i>	GET: henter alle scans (dato, tid og ansatt-id) for alle kortene til én spesifikk kunde samlet
<i>/customers/:user_id/gender</i>	GET: henter antall registrerte menn, kvinner, og uspesifisert kjønn for alle kortene til én spesifikk kunde samlet
<i>/customers/:user_id/age</i>	GET: henter antall registrerte innenfor definerte aldersgrupper for alle kortene til én spesifikk kunde samlet
<i>/admins/:user_id/registrations</i>	GET: henter alle registreringer (kun dato og tid) for alle kortene til alle kundene til én spesifikk administrator samlet
<i>/admins/:user_id/scans</i>	GET: henter alle scans (kun dato og tid) for alle kortene til alle kundene til én spesifikk administrator samlet
<i>/admins/:user_id/gender</i>	GET: henter antall registrerte menn, kvinner, og uspesifisert kjønn for alle kortene til alle kundene til én spesifikk administrator samlet
<i>/admins/:user_id/age</i>	GET: henter antall registrerte innenfor definerte aldersgrupper for alle kortene til alle kundene til én spesifikk administrator samlet
<i>/registrations</i>	GET: henter alle registreringer (kun dato og tid) totalt
<i>/scans</i>	GET: henter alle scans (kun dato og tid) totalt
<i>/gender</i>	GET: henter antall registrerte menn, kvinner, og uspesifisert kjønn totalt
<i>/age</i>	GET: henter antall registrerte innenfor definerte aldersgrupper totalt

Tabell 10: Express endepunkter under */api/stats/*

Express - Backend	
/api/users/	
Endepunkt	Beskrivelse
/	GET: henter alle brukere
/:user_id	GET: henter en bruker basert på bruker-id
/:user_id	PATCH: oppdaterer en bruker sin informasjon
/:user_id	DELETE: sletter en bruker
/:user_id/set-expiration	PUT: endrer utløpsdato på bruker
/admins	GET: henter alle brukere med rollen Admin
/admins	POST: oppretter en bruker med rollen Admin
/customers	GET: henter alle brukere med rollen Kunde
/customers/:user_id	PUT: endrer en Kunde sin overordnede Admin
/admins/:user_id/customers	GET: henter en Admin sine underordnede Kunder
/admins/:user_id/customers	POST: oppretter en bruker med rollen Kunde under en Admin
/customers/:user_id/employees	GET: henter en Kunde sine ansatte
/customers/:user_id/employees	POST: oppretter en ansatt for en Kunde
/employees/:employee_id	PUT: endrer navn på en ansatt
/employees/:employee_id	DELETE: sletter en ansatt

Tabell 11: Express endepunkter under /api/users/

## 6 Sikkerhet

### 6.1 Kommunikasjon

All kommunikasjon med serveren går over https og er da kryptert med SSL/TLS. Dersom en bruker prøver å koble til over http vil de videresendes til https. SSL/TLS sertifikater skaffes gjennom LetsEncrypt, som er en non-profit *Certificate Authority* som tilbyr gratis sertifikater. Sertifikatene som leveres av LetsEncrypt varer kun i 90 dager, og kan fornyes når det er mindre enn 30 dager igjen til de utløper. Vår løsning sjekker om den trenger å fornye sertifikatet hver søndag kl. 04:00, slik at den aldri risikerer at sertifikatet utløper.

### 6.2 Passordhåndtering

For passord-håndtering følger vi rådene til OWASP. Passordene hashes med bcrypt-algoritmen med en arbeidsfaktor på 12. Arbeidsfaktoren bestemmer hvor mye CPU-kraft som kreves for å generere hashen. En høyere arbeidsfaktor vil gjøre *brute-force* angrep vanskeligere. (OWASP 2020a) NPM-pakken

vi bruker, *bcrypt*, genererer en salt automatisk som den bruker for å hashe passordet. Hashen og saltene lagres sammen som én streng i databasen. Dersom en bruker har glemt passordet sitt, kan de få en e-post med en lenke som inneholder en *JWT-token*. Med denne lenken kan brukeren sette et nytt passord uten å kunne sitt gamle. JWT-tokenet er signert med en hemmelig nøkkel som gjør at vi kan verifisere at den ikke er forfalsket, den er kun gyldig i 48 timer, og kan kun brukes én gang. Dette antar at brukeren sin e-post-konto er sikker, og mulige utbedringer her kunne vært å implementert 2-faktor autentisering og/eller sikkerhets-spørsmål. Dette ville for øvrig også gjort løsningen sikrere generelt.

### 6.3 Autentisering

Web-løsningen bruker sesjons-basert autentisering. Her følger vi rådene til OWASP om sesjons-håndtering. Sesjons-tokenet er 240 bits og genereres av en kryptografisk sterk algoritme fra Node-pakken *crypto*. Brukersesjoner lagres i databasen på serversiden, og sesjons-tokenet lagres i cookies på klient-siden. For å sikre at sesjons-tokens ikke kommer på avveie har vi satt følgende attributter i cookieene: (OWASP 2020b)

- **Secure**: Dette gjør at cookien kun kan sendes over en sikker tilkobling (https).
- **HttpOnly**: Dette gjør at cookien ikke er tilgjengelig for script som kjører i nettleseren, og sikrer dermed at sesjons-tokenet ikke kan stjeles ved et *Cross-site Scripting* angrep.
- **SameSite=strict**: Dette gjør at cookien ikke kan sendes fra et annet domene enn vårt, noe som hindrer *Cross-site request forgery* angrep.

### 6.4 Autorisasjon

For å begrense tilgang til ressurser brukes *Express-middlewares*, eller *mellomvare*. I Express.js går forespørslene “gjennom” en rekke funksjoner helt til forespørselen blir besvart. Man kan derfor legge inn funksjoner, eller mellomvare mellom innkommende forespørsel og svar. På denne måten kan man for eksempel først sjekke at brukeren er pålogget, og dersom brukeren er pålogget så går forespørselen videre til funksjonen som henter ressursen. Hvis brukeren derimot ikke er pålogget, vil mellomvaren svare på forespørselen med errorkode 401, og forespørselen kommer ikke videre derfra. På samme måte kan vi sjekke brukertype, og hvorvidt brukeren “eier” ressursen de forespør, for å bestemme om brukeren skal få tilgang til ressursen de ber om eller ikke.

### 6.5 Beskyttelse mot angrep

Frontend-løsningen bruker Vue, som er et moderne frontend-rammeverk som får kontinuerlige oppdateringer for å sikre seg mot *Cross-site scripting* angrep. Dette gjør den blandt annet ved å *rense* data før det kjøres i nettleseren slik at tegn fra javascript-syntaksen ikke tolkes som kode. På backend brukes sequelize for kommunikasjon med MySQL databasen, og dette biblioteket sørger for å forhindre *SQL-injection* ved å *rense* input slik at tegn som er en del av MySQL-syntaksen ikke blir tolket som kode. Det er kun NGINX som er tilgjengelig på internett på port 80 og 443, backend-serveren og databasen er blokkert utad av brannmur. Dette begrenser angrepsflaten betydelig og gjør det vanskeligere å angripe selve plattformen applikasjonen kjører på. I tillegg sørger AWS for beskyttelse av selve infrastrukturen.

## 7 Installasjon og kjøring

(*Installasjon og kjøring er også dokumentert i README.md i prosjekt-repoet*)

## 7.1 Avhengigheter

- **npm** er pakkebehandleren vi bruker
- **Vue.js** brukes som frontend-rammeverk
- **Vuex** brukes for håndtering av global tilstand i Vue.js
- **Vuetify** brukes som et komponent-rammeverk for Vue.js
- **Node.js** brukes som *runtime-system* for backend
- **Express.js** brukes som backend-rammeverk
- **Sequelize** brukes av backend for kommunikasjon med MySQL-databasen
- **React Native (Expo)** er et rammeverk for Node.js som brukes til skanner-appen

## 7.2 Lokal installasjon

1. Installer Node.js versjon 12 fra [nodejs.org](https://nodejs.org)
2. Installer MySQL versjon 5.7 fra [mysql.com](https://mysql.com)
3. Last ned eller clone prosjekt-repoet fra [GitHub](https://github.com):  

```
git clone https://github.com/EstRob1982/LoyaltyCard.git
```
4. Kjør `db_setup.sql` fra prosjekt-repoet i MySQL databasen for å sette opp tabell-strukturen. Eventuelt også kjør `db_test_data.sql` dersom du ønsker test-brukere og kort.

### 7.2.1 Backend

Naviger til `backend`-mappen i prosjektet. Kjør `npm install` for å installere avhengighetene til backend. Videre, må du lage en `keys`-mappe der du legger `apple_key.pem`, `apple_cert.pem`, og `gapi_key.pem`. Generering av Apple sertifikat og privatnøkkel beskrives i neste delkapittel. Deretter må du sette opp miljøvariablene som trengs av backend. Dette kan gjøres ved å lage en `.env` fil, der du skriver inn variabler på formen **NØKKEL=VERDI**. Backend trenger følgende miljøvariabler:

```
NODE_ENV=development
SESSION_EXPIRATION (i minutter)
JWT_SECRET (for signering av jwt-tokens)
APPLE_PASS_TYPE_IDENTIFIER
APPLE_TEAM_IDENTIFIER
APPLE_PEM_PRIVATE_KEY_PASSPHRASE
GAPI_CLIENT_EMAIL
GAPI_TOKEN_URI=https://oauth2.googleapis.com/token
CERT_DOMAINS=localhost:8080
RDS_HOSTNAME=localhost
RDS_DB_NAME=ebdb
RDS_USERNAME (brukernavn til database)
RDS_PASSWORD (passord til database)
MAIL_USER (for sending av passord-reset)
MAIL_PASSWORD
```

Nå kan du starte backend ved å kjøre `npm start`.

## 7.2.2 Apple sertifikat og privatnøkkel

Følg [Apple Wallet Developer Guide](#) for å laste ned et Pass Type ID sertifikat. Lagre Pass Type IDen i miljøvariabelen `APPLE_PASS_TYPE_IDENTIFIER`. Hent også Team Identifiser til Apple Developer kontoen, og lagre den i miljøvariabelen `APPLE_TEAM_IDENTIFIER`.

Installer sertifikatet på en Mac datamaskin ved å dobbelttrykke på den. Finn deretter sertifikatet i Keychain Access og eksporter den som `.p12` fil. Bruk Pass Type IDen som filnavn slik at filen heter noe som `pass.development.loyalty.no.p12`. Den vil be deg skrive inn et passord, dette kan være hva som helst, men skriv det ned så du husker det. Det kan hende du også blir bedt om passordet til datamaskinen, så ikke bland disse. Kopier den eksporterte `.p12` filen til en ny midlertidig `certificates`-mappe plassert i backend mappen. Fra backend mappen skal man kunne kjøre kommandoen:

```
passkit-keys certificates

# Hvis det ikke fungerer, prøv med full path til passkit-keys programmet:
./node_modules/@walletpass/pass-js/bin/passkit-keys certificates
```

Du vil bli bedt om et “Import password”, som er passordet du husket fra tidligere. Så blir du bedt om å lage en “pass phrase” som ikke kan være for lett å gjette, så en kombinasjon av ulike ord som f.eks. “edderkopp pyramide” anbefales her. Lagre dette valgte passordet i miljøvariabelen `APPLE_PEM_PRIVATE_KEY_PASSPHRASE`.

Åpne `.pem` filen som ble lagd i en tekstbehandler og pass på at det er en `CERTIFICATE` blokk med data og en `ENCRYPTED PRIVATE KEY` blokk med mer data. Lag to filer, `apple.cert.pem` og `apple.key.pem` i `keys`-mappen hvis de ikke eksisterer allerede. Kopier hele `CERTIFICATE` blokken fra `.pem` filen til `apple.cert.pem` slik at innholdet ser slikt ut:

```
-----BEGIN CERTIFICATE-----
MIIF9DCCBNygAwIBAgIIQgOtXzFvBQUwDQ.....
.....
-----END CERTIFICATE-----
```

Kopier deretter `ENCRYPTED PRIVATE KEY` blokken til `apple.key.pem` slik at innholdet ser slik ut:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBAbGkqhkiG9w0BBQ0wMzAbBgkq.....
.....
-----END ENCRYPTED PRIVATE KEY-----
```

Du kan nå slette den midlertidige `certificates`-mappen. Merk at feilinstallasjon av Apple sertifikater kan feile uten noe melding, så det anbefales å bruke iOS Simulator på en Mac datamaskin og ha åpen Console programmet for å kunne sjekke eventuelle feilmeldinger ved testing av Apple Pass.

## 7.2.3 Google sertifikat og privatnøkkel

Det man trenger av Google sertifisering for å bruke APIet til Google vil komme fra en service-account. Hvordan man lager denne står forklart i [Basic Setup - Get access to REST API - 1. Register Your Application](#). Ved steg 11. i prosessen vil du få lastet ned en privatnøkkel ved navn `privatekey.json`. Denne vil ha følgende form:



```

{
  "type": "service_account",
  "project_id": "Navnet til prosjektet",
  "private_key_id": "Generert ID",
  "private_key": "-----BEGIN PRIVATE KEY-----
                  MIIFDjBAbgkqhkiG9wOBBQ0wMzAbBgkq.....
                  .....
                  -----END PRIVATE KEY-----",
  "client_email": "brukernavn@prosjekt-id.iam.gserviceaccount.com",
  "client_id": "Generert ID",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/klient-epost"
}

```

Privatnøkkelen som skal lagres i `keys/gapi_key.pem` er `private_key` feltet i `privatekey.json`. Videre må `client_email` lagres i `.env`. De andre feltene i `privatekey.json` er ikke nødvendig for å gjøre API kall til Google Pay for Passes.

Følg så det neste steget på siden, [2. Tie your service account to your Google Pay API for Passes account](#). Deretter vil resten håndteres av backend.

#### 7.2.4 Frontend

Naviger til `frontend`-mappen i prosjektet og kjør `npm install`. Som nevnt i sikkerhets-delen vil ikke sesjons-tokens fungere uten SSL/TLS-kryptering, og i produksjonsmiljøet er det NGINX-serveren som håndterer dette. Heldigvis har Vue.js en innebygd proxy-server for å kjøre lokalt, som også kan håndtere dette. Du trenger derfor å lage et selv-signert SSL/TLS sertifikat for å kjøre applikasjonen lokalt. Lag en `keys`-mappe der du legger `key.pem` og `cert.pem`. Dette kan lages ved hjelp av OpenSSL:

```
openssl req -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365 -subj '/CN=localhost'
```

Nå kan du starte frontend ved å kjøre `npm run serve`.

#### 7.2.5 App

Først må du installere Expo-Cli globalt: `npm install expo-cli -g`. Deretter, navigerer du til `app`-mappen og kjører: `npm install`. Videre, må du opprette en `environments.js` fil med følgende innhold:

```

import Constants from 'expo-constants';

const ENV = {
  dev: {
    apiUrl: '',
  },
  staging: {
    apiUrl: '',
  },
  prod: {
    apiUrl: '',
  },
};

const getEnvVars = (env = Constants.manifest.releaseChannel) => {
  if (__DEV__) {
    return ENV.dev;
  }
}

```

```

    } else if (env === 'staging') {
      return ENV.staging;
    } else if (env === 'prod') {
      return ENV.prod;
    }
  };
}

export default getEnvVars;

```

Fyll inn `apiUrl` med endepunkter for å nå de forskjellige miljøene. **NB!** du kan ikke bruke `localhost` som endepunkt, siden det er appen på mobilen som kobler til backend-serveren. Bruk derfor din lokale ip-adresse på nettverket. Du kan starte utviklingsmiljøet til Expo ved å kjøre `npm start` eller `expo start`. For å få appen på mobilen må du laste ned Expo-appen fra App Store eller Google Play. På denne måten får du konsoll-output fra appen til pc-en. Hvis du ønsker en *standalone* versjon du kan installere, kan du kjøre: `expo build:android -t apk --release-channel staging|prod`. Da får du en *apk* fil du kan installere på android.

### 7.3 Staging/Produksjon

Staging- og produksjonsmiljøet er Node.js-plattformer på AWS Elastic Beanstalk. Følg [Deploying Node.js applications to Elastic Beanstalk](#) for hvordan dette kan konfigureres. Det må også settes opp en MySQL-database for miljøet, følg [Adding a database to your Elastic Beanstalk environment](#) for hvordan dette kan gjøres. I tillegg, må du spesifisere miljøvariabler, dette gjøres under **Configuration > Software > Environment properties**. Miljøvariablene du må spesifisere er som følger:

```

CERT_DOMAINS
CERT_EMAIL
NODE_ENV=prod|staging
SESSION_EXPIRATION (minutes)
JWT_SECRET
APPLE_PASS_TYPE_IDENTIFIER
APPLE_TEAM_IDENTIFIER
APPLE_PEM_PRIVATE_KEY_PASSPHRASE
GAPI_CLIENT_EMAIL
GAPI_TOKEN_URI
MAIL_USER
MAIL_PASSWORD

```

Elastic Beanstalk gjør følgende miljøvariabler tilgjengelig for applikasjonen automatisk:

```

PORT
API_ENDPOINT
RDS_HOSTNAME
RDS_PORT
RDS_DB_NAME
RDS_USERNAME
RDS_PASSWORD

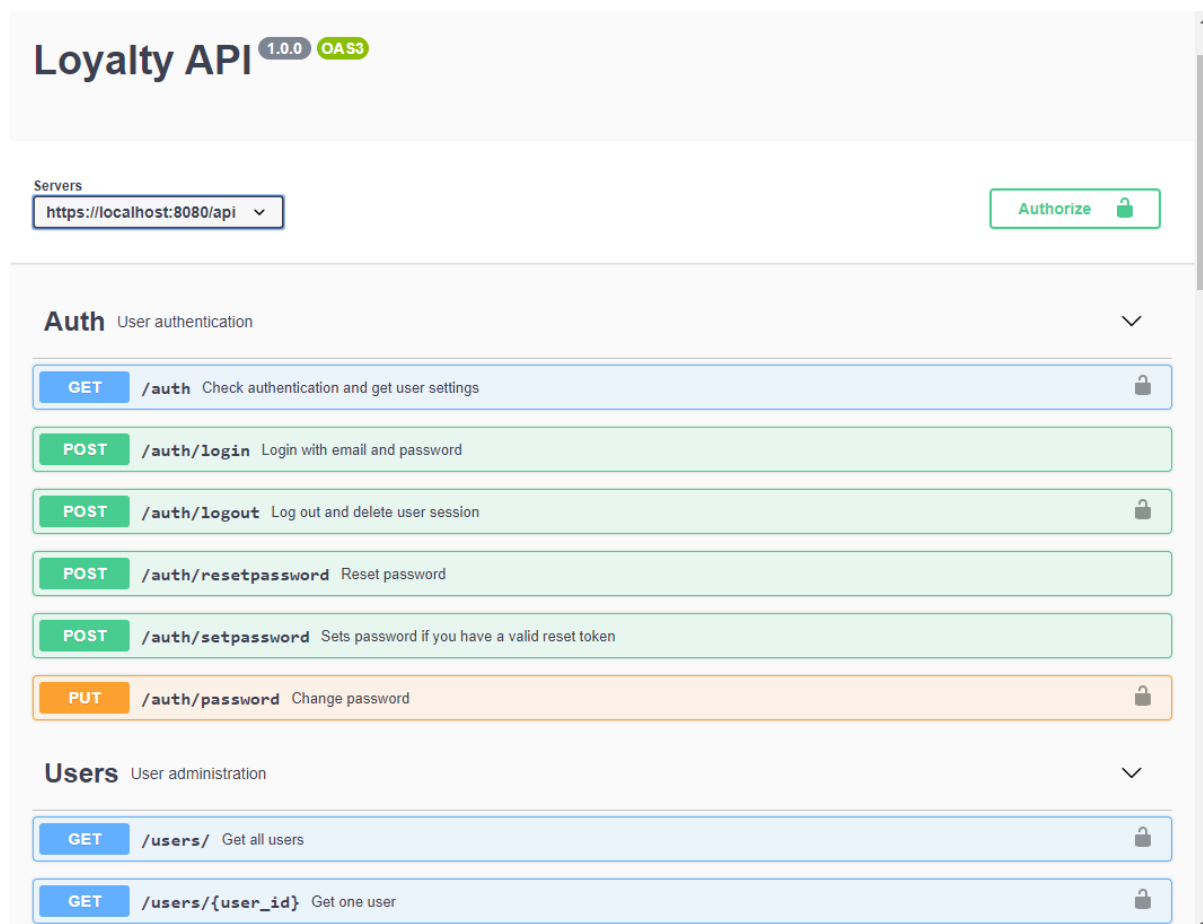
```

SSL/TLS-sertifikater lastes ned automatisk fra LetsEncrypt ved utrulling, ved hjelp av *certbot*. Elastic Beanstalk oppretter automatisk en *storage bucket* i AWS S3 for miljøet. Dette er enkelt og greit et sted man kan lagre filer for applikasjonen. I denne storage-bucketen, under `keys`, må `apple_key.pem`, `apple_cert.pem`, og `gapi_key.pem` lagres. Applikasjonen laster ned disse filene automatisk ved utrulling. For å kjøre applikasjonen i staging- eller produksjonsmiljøet må koden kjøres gjennom CI/CD-pipelinen beskrevet i kapittel 9.2.

## 8 Dokumentasjon av kildekode

### 8.1 Backend

APIet er dokumentert med *Swagger* ved hjelp av pakkene: *swagger-jsdoc*, som lager dokumentasjon ut i fra jsDoc-kommentarer i koden, og *swagger-ui-express*, som gjør at Express.js backenden kan levere dokumentasjonen som en webside. For å vise API-dokumentasjonen må du først starte backenden med `NODE_ENV=development`, deretter, kan du vise dokumentasjonen i en nettleser ved å gå til `/api/docs`. I figur 2 kan du se hvordan dette ser ut.



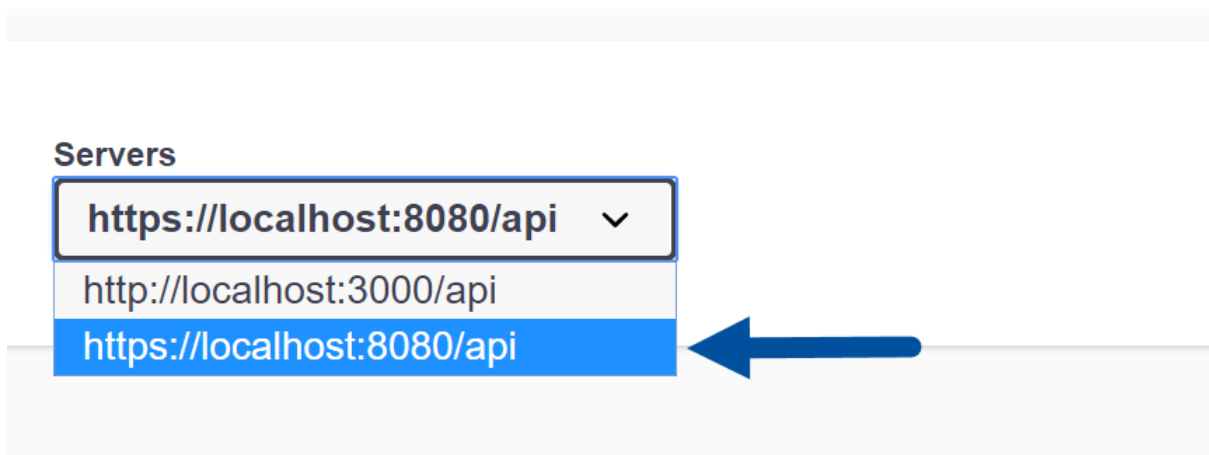
Figur 2: Skjermbilde fra Swagger-dokumentasjon

Her kan du se de ulike endepunktene som utgjør APIet, hvilke parametere de ulike endepunktene tar, og hvilke resultater de kan gi. Swagger lar deg også prøve ut endepunktene i nettleseren. Hvis du ønsker å prøve ut endepunkter som krever innlogging (merket med hengelås-symbol) må du kjøre frontend-proxyserveren beskrevet i kapittel 7.2.4, og åpne dokumentasjonen via den. Dette er fordi at cookies kun sendes over https. Deretter, kan du enten trykke på **Authorize**-knappen og oppgi en gyldig sesjonsid, eller du kan bruke `/api/auth/login` endepunktet til å logge inn med brukernavn og passord; sesjonsid-en vil da lagres av nettleseren. Videre kommer en steg for steg forklaring på hvordan dette kan gjøres:

Først må du åpne dokumentasjonen via frontend-proxyserveren, dette er vanligvis på:

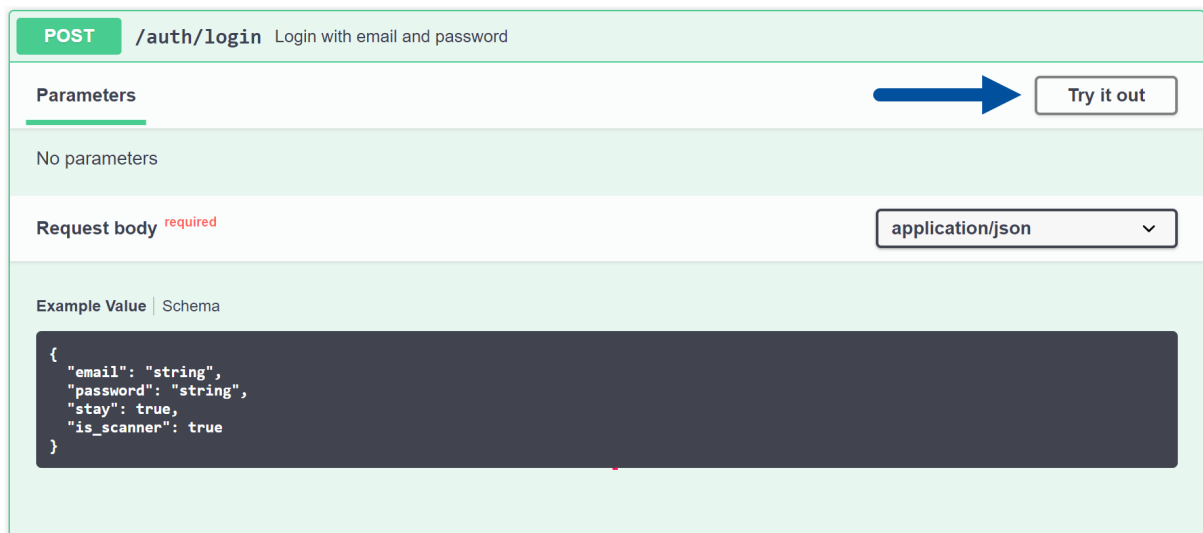
```
https://localhost:8080/api/docs
```

Deretter, må du velge riktig server i Swagger slik at den sender forespørsler gjennom proxyen over https:



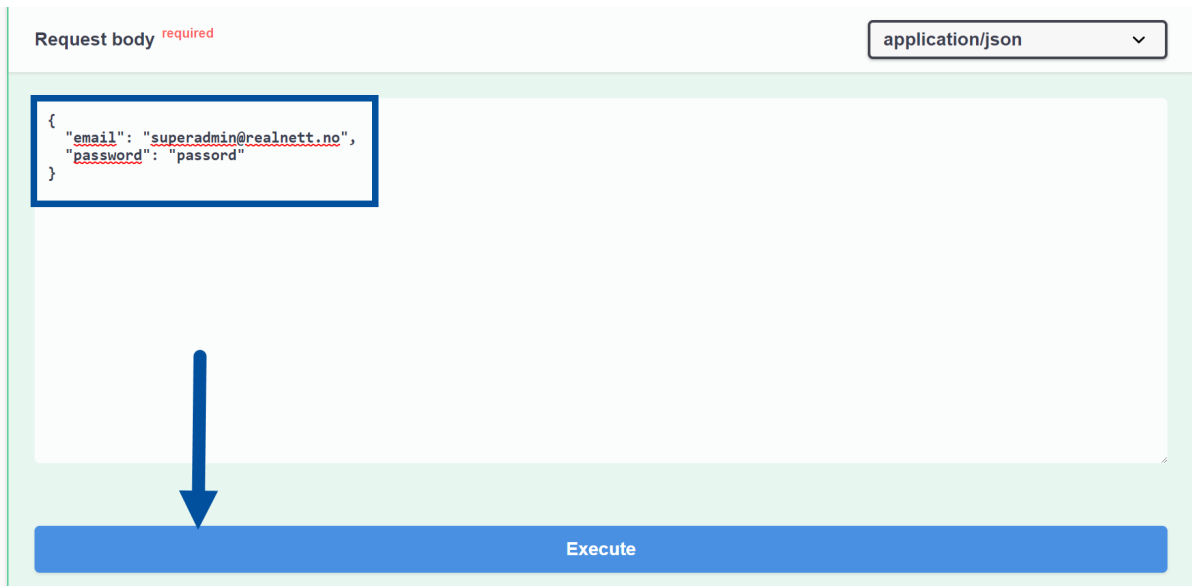
Figur 3: Valg av server — Swagger

Videre, kan du finne endepunktet `/api/auth/login`, og trykke på **Try it out**.



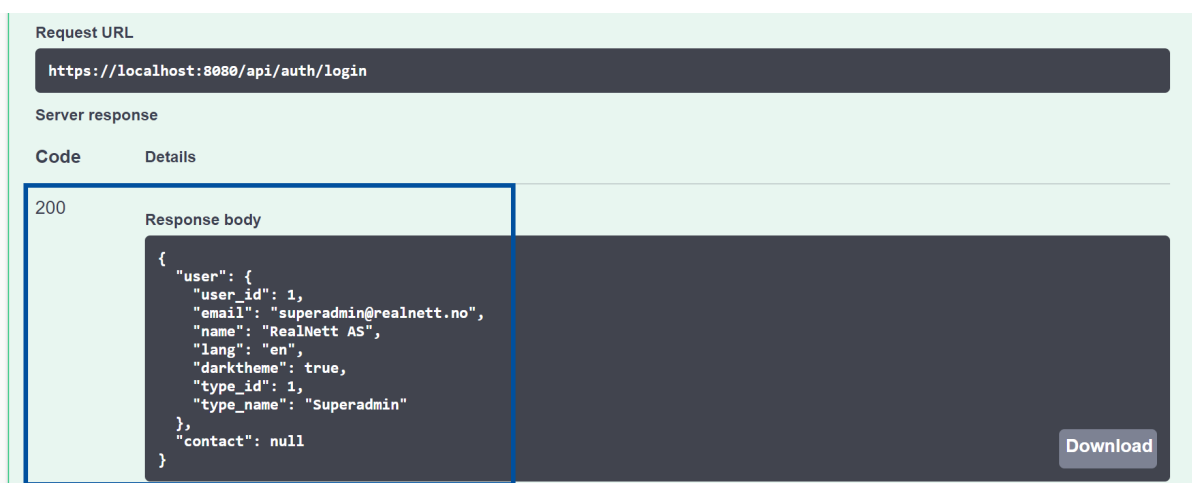
Figur 4: Trykk på Try i out — Swagger

Deretter, skriver du inn e-post og passord og trykker på **Execute**.



Figur 5: Trykk på execute — Swagger

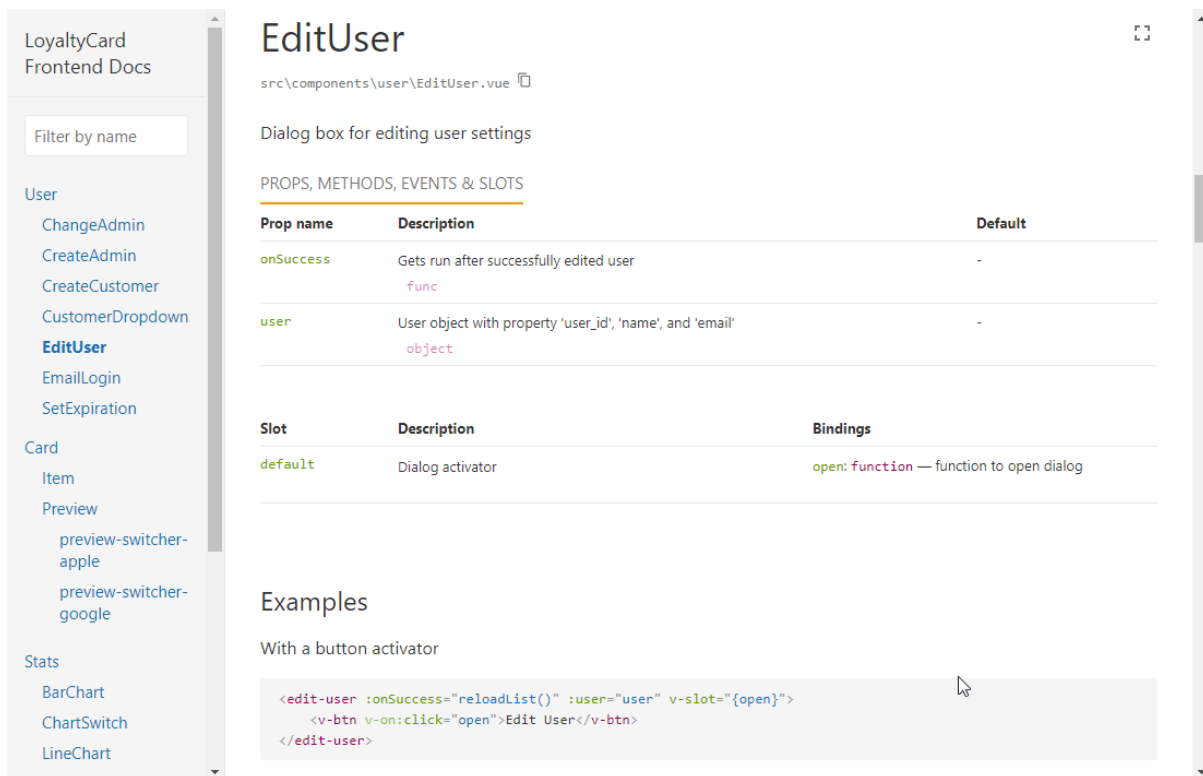
Hvis du gjorde alt riktig skal du få statuskode 200, samt informasjon om brukeren. Da skal sesjonsid-en være lagret i nettleseren og du kan prøve ut resten av APIet.



Figur 6: Vellykket innlogging — Swagger

## 8.2 Frontend

Gjenbrukbare Vue-komponenter vi har laget er dokumentert med *Vue Styleguidist*. Dette er en plugin for Vue som genererer dokumentasjon ut i fra jsDoc-lignende kommentarer i koden. For å vise frontend-dokumentasjonen kan du enten kjøre `npm run styleguide` for å få dokumentasjonen levert som en nettside, eller du kan kjøre `npm run styleguide:build` for å generere statiske html-filer du kan vise i en nettleser. Begge disse kommandoene må kjøres fra `frontend`-mappen. For hver komponent kan du se *props*, metoder, *events*, *slots*, og brukseksempler. I figur 7 kan du se et eksempel på hvordan dokumentasjonen ser ut.



Figur 7: Skjerm bilde fra Vue Styleguidist-dokumentasjon

## 9 Kontinuerlig integrasjon og testing

### 9.1 Testing

For enhetstesting brukes Jest, et populært testrammeverk vedlikeholdt av Facebook. Vi har fokusert på testing av backend fremfor frontend, siden det er i backend det er viktig med enhetstester med tanke på sikkerhet og personvern. Testene ligger i `backend/tests` i prosjekt-repoet. Testene kan kjøres lokalt ved å kjøre: `npm test` fra `backend`-mappen.

#### 9.1.1 Enhetstesting av backend

For testing av Express.js-backend brukes Supertest. Supertest setter opp en lokal testserver som kjører backend-koden og tilbyr metoder for å sende forespørsler til testserveren. Resultatet av forespørslene kan da testes med Jest. På denne måten testes samtlige endepunkter under `auth.js`, `gdpr.js`, `users.js`, `stats.js`, og litt av `scan.js`. Her testes det at resultatene av forespørslene stemmer, og at tilgangsstyringen fungerer som den skal; dette er spesielt viktig for endepunkter som håndterer sensitiv data, som for eksempel `gdpr.js`. Endepunkter som er avhengige av Apple eller Google API er ikke enhetstestet, da disse avhengighetene gjør enhetstesting vanskelig. Det burde imidlertid være mulig å teste dersom man klarer å *mocke* Apple og Google APIene, men dette har ikke vi gjort. I figur 8 kan du se testdekningen.

## All files

37.74% Statements 762/2019 18.21% Branches 148/769 28.04% Functions 129/468 37.88% Lines 752/1985

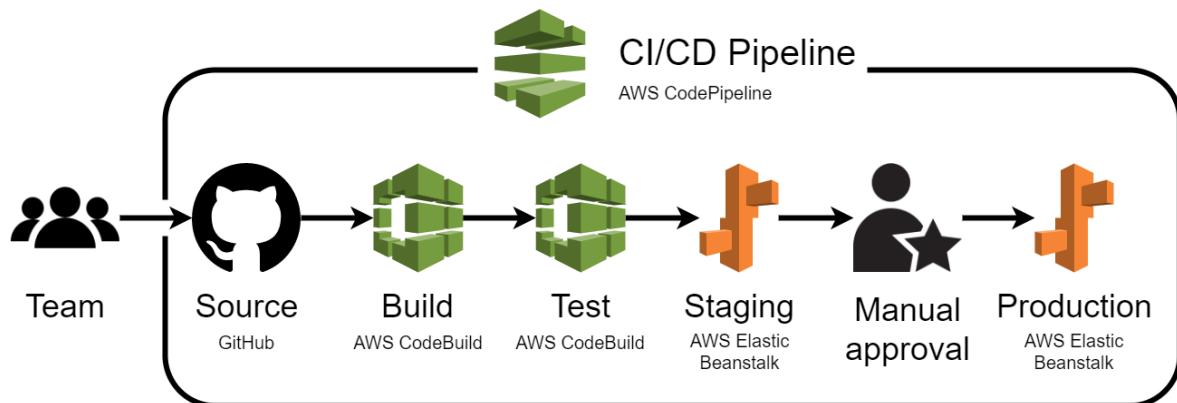
Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

File	Statements	Branches	Functions	Lines				
backend	93.02%	40/43	50%	1/2	100%	0/0	93.02%	40/43
backend/cards/apple	5.68%	5/88	0%	0/53	0%	0/19	5.81%	5/86
backend/config	47.95%	35/73	50%	17/34	28.57%	4/14	47.95%	35/73
backend/models	100%	80/80	100%	0/0	100%	0/0	100%	80/80
backend/models/apple	50%	24/48	100%	0/0	0%	0/8	52.17%	24/46
backend/models/cardtemplate	35.69%	91/255	0%	0/178	0%	0/25	36.84%	91/247
backend/routes	44.57%	443/994	32.97%	122/370	40.19%	125/311	44.46%	433/974
backend/routes/google	9.18%	39/425	0%	0/130	0%	0/80	9.22%	39/423
backend/utlis	38.46%	5/13	0%	0/2	0%	0/3	38.46%	5/13

Figur 8: Testdekning av backend

## 9.2 Kontinuerlig integrasjon

For kontinuerlig integrasjon benyttes AWS CodePipeline og AWS CodeBuild. I figur 9 kan du se oppsettet. Når kode sjekkes inn i GitHub på master, vil CodePipeline automatisk laste ned nyeste versjon. Deretter, sendes det videre til AWS CodeBuild som bygger koden og gjør det klart til å kjøre på serveren. Før det går videre kjøres enhetstestene i et eget CodeBuild-miljø. Dersom koden ble bygget uten problemer og testene gikk gjennom uten feil blir koden automatisk rullet ut på *staging*-serveren. Staging-serveren er så likt produksjons-miljøet som overhodet mulig, og dermed kan løsningen testes som en helhet før oppdateringen rulles ut til produksjonsmiljøet. Dette gjøres ved at oppdragsgiver godkjenner utrullingene manuelt.



Figur 9: Oppsettet i AWS CodePipeline

### 9.2.1 Konfigurasjon

*Build*-konfigurasjonen er definert i `buildspec.yml` i prosjekt-repoet. Her bygges Vue-frontend for produksjon, backend trenger ikke å bygges. CodeBuild-miljøet for byggingen trenger også et par miljøvariabler:

```
NPM_AUTH_TOKEN
NPM_EMAIL
```

Du må altså knytte CodeBuild-miljøet til en npm-bruker, ellers vil ikke npm installere pakkene som er nødvendig for å bygge koden. Følg [Creating and viewing authentication tokens](#) for hvordan du kan opprette en autentiserings-token for npm-brukeren din.

Test-konfigurasjonen er definert i `testspec.yml` i prosjekt-repoet. Her installeres og konfigureres korrekt versjon av MySQL før testene kjøres.

## Referanser

OWASP (2020a). *Password Storage*. URL:

[https://cheatsheetseries.owasp.org/cheatsheets/Password\\_Storage\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html) (sjekket 30.04.2020).

— (2020b). *Session Management*. URL:

[https://cheatsheetseries.owasp.org/cheatsheets/Session\\_Management\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html) (sjekket 30.04.2020).



