

Bachelorprosjekt våren 2021

Tilgjengelighet

Fri

Begrenset

Emnekode og emnenavn	BAO301 Bachelorprosjekt
Tittel norsk	Rebel Cash: praktisk betalingsapplikasjon som samler nyttige verktøy.
Tittel engelsk	Rebel Cash: convenient payment application that contains useful tools.
Oppdragsgiver	Shortcut
Innleveringsdato	20 mai 2021
Antall sider	67 (pluss referanser og vedlegg)
Antall ord	11963 (alt fra forord til konklusjon, inkludert overskrifter)

Sammendrag (maks 100 ord)

Denne rapporten beskriver prosessen gjennom vårt Bachelorprosjekt fra januar til juni 2021. Den gjennomgår reisen vår gjennom utforming og utvikling av Rebel Cash applikasjonen vi laget for vår kunde. Dette prosjektet var et proof of concept for vår kunde Rebel (Entra ASA), for å vise dem mulighetene de har med denne applikasjonen. Vårt mål var å produsere en

betalingsapplikasjon som kunden kunne videreutvikle etter overlevering fra oss, med en in-house valuta. Vi har hatt fokus på funksjonalitet som kan vise potensialet med applikasjonen og brukervennlighet. Dette kommer frem i rapporten gjennom beskrivelse av designsprintene, brukertestene og forankring i forskning.

Gruppenummer	29	
Studentnavn	Kandidatnummer	Signatur
Kjetil Haug Terjesen	8029	<i>Kjetil Haug Terjesen</i>
Jonas Say	8124	<i>Jonas Say</i>
Lasse Hovden	8061	<i>Lasse Hovden</i>
Maya Maria Tjøme Kjær	8001	<i>Maya Kjær</i>
Stine Mathisen	8176	<i>Stine Mathisen</i>

Denne [oppgaven/bacheloroppgaven] er gjennomført som en del av utdannelsen ved Høgskolen Kristiania. Høgskolen er ikke ansvarlig for oppgavens metoder, resultater, konklusjoner eller anbefalinger.



Forord	6
Innledning	6
Om prosjektet	7
Hvem er Rebel?	7
Hvem er Shortcut?	7
Hva er oppgaven?	7
Hva skal vi gjøre?	8
Forventninger fra oppdragsgiver	8
Om gruppen	8
Målsetting og ambisjonsnivå	9
Problemstilling	10
Knytning til forskningsområder	10
Brukervennlighet	10
Kontaktløs betaling	10
NFC og QR kode	11
NFC	11
QR kode	11
Prosess og metode	11
Utviklingsmetode	12
Metode	12
Hva er Scrumban?	12
Teknikker i Scrumban?	12
Hvilke teknikker skal vi bruke i Scrumban?	12
Hvorfor skal vi bruke Scrumban?	13
Valg av teknologier og Verktøy	14
Notion	14
Figma.	14
Slack	14
Google Docs	15
Digitalt videomøte	15
Git, Github og Git Workflow	15
Betalingsteknologi	17
Plattform	17
Database valg	18

Editor (IDE) til Utvikling	18
SwiftUI og MVVM	18
Gjøre rede for testing	19
Brukertesting	19
Enhetstesting	20
Prosjektplan	20
Risiko og tiltaksplan	20
Datainnsamling	21
Brukerinnsikt	21
Personas	21
Brukerhistorier	21
Brukerreise	21
Analyse og utforming	22
Personas	22
Brukerreisen	23
Design	24
Designmal	24
Farger	24
Designprinsipper	25
Universell utforming	25
Workshop	28
Empatikart	28
“Low hanging fruit”	29
Designsprint	32
Klientapplikasjon	32
“Need to have”	32
“Nice to have”	34
Terminal	35
Prototype	36
Lavnivå prototype	36
Høynivå prototype	38
Versjon 1	38
Versjon 2	39
Versjon 3	41
Brukertest av lavnivå prototype - Analyse	41
Tabelloversikt av funn	42

Konklusjon	43
Brukertest av høynivå prototype - Analyse	43
Tabelloversikt over funn	44
Funn og anbefalinger	45
Konklusjon	47
Koding av terminal og Klient	47
Terminal	47
Klient	47
NFC problematikk	48
Løsning	48
Prototype	48
Utforming og funksjonalitet	52
Terminal applikasjon	52
Klientapplikasjon	53
Arkitektur	57
Kommunikasjon terminal og klient	58
Database	58
Vurdering	60
Vurdering av prosess	60
Planlegging og metode	60
Workshop	61
Designsprint, prototyping og brukertesting	62
SwiftUI og Versjonskontroll	63
Vurdering av Løsningen	64
Veien videre	65
Konklusjon	66
Referanser	67
Vedlegg	69

Forord

I løpet av dette prosjektet har vi brukt kunnskapen vi har tilnærmet oss over de siste tre årene og utviklet en moderne løsning for Rebel som er eid av Entra ASA gjennom vår arbeidsgiver Shortcut.

Vi ønsker å takke Shortcut AS for muligheten vi har fått til å skrive til et reelt prosjekt som en bachelorgruppe. En ekstra stor takk ønsker vi å gi til vår eksterne veileder Asle Kinnerød, Tech lead hos Shortcut. Han har vist oss en enorm gjestfrihet, inkludert oss som gruppe i hele Shortcut-teamet og virkelig gjort sitt beste for at vi skulle ha det best mulig under dette oppdraget i en periode hvor teamet måtte sitte hjemme grunnet Covid-19. Vi ønsker også å rette en takk til vår interne veileder Rolando Gonzales som stiller opp uansett hvilken dag det er. Han har kommet med grundige og spesifikke tilbakemeldinger som har vært veldig verdifulle for oss.

1. Innledning

De siste fem månedene av skolegangen ved institutt for teknologi ved Høgskolen Kristiania avsluttes med bachelorprosjekt. Prosjektet ble utført hos Shortcut, en bedrift som er ledende og unik i Norge innenfor apputvikling. Vi er stolte av å kunne jobbe med prosjektet Rebel, et fremtidsrettet bygg beregnet til IT-bransjen, hvor kreativitet og sosialisering står i fokus. Gjennom bachelorprosjektet vil vi kunne lære av de beste samt briljere med kunnskapen gruppen sitter inne med. Begreper som brukes gjennom rapporten er forklart i en begrepsliste i vedlegg A.

1.1. Om prosjektet

Under dette bachelorprosjektet “jobber” vi hos et konsultentselskap kalt Shortcut. Entra er en av kundene til Shortcut og vi har fått i oppdrag å lage en applikasjon til deres bygg, Rebel.

1.1.1. Hvem er Rebel?

“De beste IT-hodene” (Rebel, n.d.) 1500 av dem samles under ett tak. Rebel skal leie ut kontorer med fleksible medlemskap samt faglige, flerfoldige fellesskap, Co-working og communities. Dette innebærer at de har konferansesaler og auditorier, møteplasser og treffpunkter fullt med samarbeid og samspill. Utenom det faglige skal Rebel huset også disponere kaffebarer, spisesteder, hangouts og noen barer som er mer en bare bar.

1.1.2. Hvem er Shortcut?

Shortcut ble grunnlagt i 2007 og har per dags dato 77 registrerte ansatte i Norge. I 2019 åpnet selskapet sitt første kontor utenfor Oslo og det samme året åpnet de i København. Selskapet har en enorm vekst både blant ansatte og kunder.

De er ledende i Norge innenfor apputvikling og har store kunder som Vipps og Rema 1000. Deres applikasjoner er rettet direkte mot sluttbrukeren og anmeldelsene deres på Google Store og App Store snakker for seg. Shortcut jobber i hovedsak native utvikling til Android og iOS, men er imøtekommende med kundene sine behov og har egne ansatte som er spesialiserte på kryssplattform utvikling.

1.1.3. Hva er oppgaven?

Entra har en visjon om å samle “de beste IT-hodene” inn i et bygg kalt Rebel, for å skape kreativitet og fellesskap mellom de ulike firmaene. Rebel-bygget er fremtidsrettet, de vil blant annet gi brukerne muligheten til å booke arbeidsplasser, åpne dører ved bruk av NFC, samt kunne betale med mobilen i en og samme applikasjon. Shortcut har gitt oss i oppdrag å utvikle et proof of concept av betalingsløsningen, som vil være en del av en større applikasjon. Vi skal



utvikle en klientapplikasjon samt en terminal. Entra har også presisert at de ønsker en in-house valuta kalt Rebel Cash, som skal brukes i Rebel-bygget.

1.1.4. Hva skal vi gjøre?

Vi skal lage morgendagens betalingsløsning for Rebel-bygget. I og med at prosjektet er såpass stort vil det være ønskelig å sette noen rammer for prosjektet og konkretiserer hva som er viktigst for kunden og brukerne. Det vil være til nytte å kartlegge hvem brukerne er, samt avdekke nødvendige behov i vår løsning. Ettersom vi allerede vet at Shortcut er en av de bedriftene som skal inn i Rebel-bygget vil dette være en kilde vi kommer til å benytte oss mye av.

1.1.5. Forventninger fra oppdragsgiver

Shortcut er opptatt av trivsel, selvstendighet, motivasjon samt punktlighet. Deres forventninger til oss er at vi arbeider selvstendig gjennom hele bacheloroppgaven. Det vil si at vi booker inn møtene med veiledere, kunden samt nøkkelpersoner som kan hjelpe oss videre i prosessen av eget initiativ. Shortcut forventer utvikling av en applikasjon for løsningen og mye av konseptet rundt denne.

1.2. Om gruppen

Gruppen består av fem studenter hvor fire av studentene går frontend- og mobilutvikling og en av studentene går interaktivt design som vist i *tabell 1*. Vi er en gruppe som utfyller hverandres kompetanse godt og samarbeider godt sammen. Se vedlegg B for gruppekontrakt.

Gruppemedlemmer	Studieretning	Interesseområder
Kjetil Haug Terjesen	Frontend og mobilutvikling	iOS, spesielt SwiftUI
Jonas Say	Frontend og mobilutvikling	Web, JS (React/Vue)
Lasse Hovden	Frontend og mobilutvikling	Backend, Kotlin, Java
Maya Maria Tjøme Kjær	Frontend og mobilutvikling	App design, Kotlin, SwiftUI
Stine Mathisen	Interaktivt Design	App design, brukervennlige løsninger

Tabell 1: Oversikt over gruppemedlemmer med studieretning samt interesser.

1.3. Målsetting og ambisjonsnivå

Hovedmålet vårt er å kunne levere fra oss noe konkret som kunden har bruk for og kan videreutvikle. Vi ønsker å møte kunden sitt behov om en betalingsløsning som er smidig å ta i bruk, samt deres ønske om en in-house valuta som kan brukes både til å justere valutaen i løpet av dagen og vil være med på å skape engasjement blant brukerne. Under hovedmålet har vi noen delmål. Læringsmålet er å dra nytte av kunnskapen til alle gruppemedlemmene samt lære fra ekspertene i Shortcut og Entra. Vi setter høye forventninger både når det gjelder innsats og kvalitet til dette prosjektet og vi ønsker å lære hvordan arbeidsprosessen er i bransjen.

Både ved bruk av kunnskap vi sitter med, innsats og ny kunnskap ønsker vi å skape verdi i prosjektet vi leverer fra oss. Samtidig ønsker vi å lære av bransjen for å levere en løsning med potensiale til videreutvikling.

1.4. Problemstilling

Det finnes allerede en del betalingsløsninger i dag, mange er enkle å bruke og de dekker det viktigste behovet, det å kunne betale. Derfor vil det være mulig at flere ikke ser behovet med enda en betalingsløsning. Derfor vil det være lurt av oss å kunne se nærmere på andre behov og kartlegge hvordan brukerne kan dra nytte av en ny betalingsløsning. På bakgrunn av dette lyder problemstillingen slik:

Hva vil gjøre vår betalingsapplikasjon brukervennlig og praktisk for brukerne av Rebel-bygget? For å nå frem til en praktisk betalingsapplikasjon, ser vi at vi trenger å fokusere på enkelte ting:

- Hvor enkel bør løsningen være for å sikre god brukervennlighet
- Hvilke funksjoner vil vekke interesse hos bruker.
- Hvordan gagnar løsningen Entra

1.5. Knytning til forskningsområder

1.5.1. Brukervennlighet

Et forskningsområdet som vil være nyttig for oss å se nærmere på er brukervennlighet. Vi ønsker å sikre gode brukeropplevelser gjennom bruk av WCAG 2.1 og designprinsipper. Vi vil også se nærmere på retningslinjene til Apple for applikasjon design, hvor brukervennlighet er et av fokusområdene. Ved å følge retningslinjer fra WCAG 2.1 vil kunne designe en løsning som er brukervennlig for de med nedsatte funksjonsnedsettelse, samtidig vil vi kunne sikre en brukervennlig løsning for alle.

1.5.2. Kontaktløs betaling

Kontaktløs betaling er betaling uten fysisk kontakt med terminal.

1.5.3. NFC og QR kode

NFC (Near Field Communication) og QR kode (Quick Response code) er brukt i andre betalingsapplikasjoner i dag og begge er populære ut i fra sine egenskaper.

1.5.3.1. NFC

NFC er en trådløs teknologi for å utveksle data mellom sender og mottaker. Mobilen kan brukes til å lese eller skrive data til NFC tagg. Mobilen kan være en NFC tagg i seg selv.

1.5.3.2. QR kode

QR-Kode (*ref bilde 1*) er enkelt forklart fungerer det som en todimensjonal strekkode.



Bilde 1: QR-kode

2. Prosess og metode

I dette kapittelet tar vi for oss prosesser og metoder som vi kommer til å benytte oss av gjennom prosjektet. Vi kommer til å utrede en prosjektplan, risikoplan og en tiltaksplan samt dokumentere valg av teknologier og datainnsamling med blant annet brukertester. Vi vil også ta for oss de ulike verktøyene vi vil benytte oss av under prosjektet.

2.1. Utviklingsmetode

2.1.1. Metode

2.1.1.1. Hva er Scrumban?

Scrumban er en blanding av Scrum og Kanban som brukes i systemutviklingsprosjekter. Formålet er å forme en metode som er mer effektiv og produktiv. Dette gjøres ved at teamet velger ut teknikker fra Scrum og Kanban som passer best til prosjektet. Metoden blir da friere enn Scrum sitt system, men mer strukturert enn Kanban sitt frie oppsett (Alqudah and Razali 2018).

2.1.1.2. Teknikker i Scrumban?

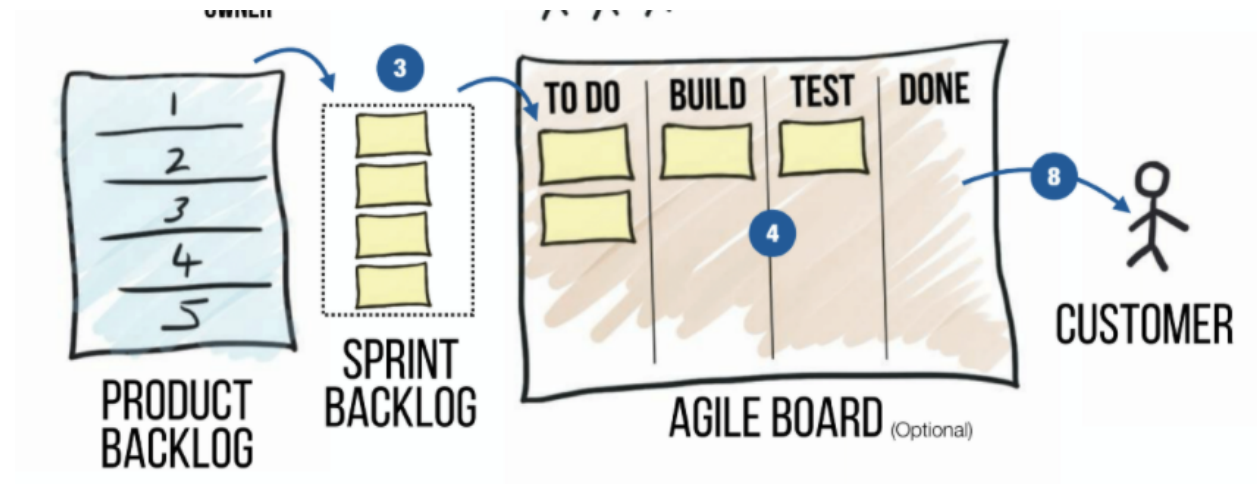
Scrumban teknikkene vil variere fra prosjekt til prosjekt. Det avhenger helt av prosjektets størrelse og tidsforløp. I Scrumban er det vanlig å ha en backlog med oversikt over hvilke oppgaver som skal utføres (Daly, n.d.). Kanban-brett er en nyttig verktøy for å dele opp oppgavene mer detaljert ved bruk av kolonner og kort, dette gjør det enklere for teamet å forplikte seg til riktig mengde arbeid (Rehkopf, n.d.). Burndown-graf er ikke noe som er påkrevd, men kan brukes hvis teamet synes de trenger det. Som vi ser i tabellen i vedlegg C så fokuserer Scrumban mer på kontinuerlig flyt kontra sprinter, men et team kan velge om de ønsker å bruke sprinter (Alqudah and Razali 2018, 3).

I tillegg kan det implementeres sprint planlegging, review og Retrospective. Dette er nyttige elementer både for å få oversikt, evaluere og lære underveis. I Scrumban brukes disse elementene når man trenger dem og de kan legges til og fjernes underveis. For å sikre god kommunikasjon er Daily Standup, eller daglige møter gode teknikker. Når det kommer til roller, så er det helt opp til teamet å bestemme om det er behov for det, eventuelt hvilke som skal brukes.

2.1.1.3. Hvilke teknikker skal vi bruke i Scrumban?

Shortcut bruker et prosjektstyrings program, Notion, som gruppen kommer til å benytte seg av. Her er det innebygget mulighet for Kanban-brett. Som vi ser på *bilde 2* under, kan vi dele opp Kanban-brettet i kolonner, se kapittel 2.1.1.2 Teknikker i Scrumban, slik som Backlog og Sprint

Backlog. Vi ønsker å bruke teknikker fra Kanban-brettet slik som: In progress, For approval og Complete/Done.



Bilde 2, Eksempel på kanban-brett i Scrumban (Kevcodez, n.d.)

Kommunikasjon og læring er ekstremt viktig, derfor ønsker vi å benytte oss av Daily Standup, samt retrospektive teknikker som vår smidige metodikk tilbyr. Når det gjelder sprinter er det ønskelig å benytte seg av en ukes sprinter. Dette vil gjøre prosjektet mer håndterbart, samt teamet vil ha lettere for å sende hyppigere leveranser til kunden (Rehkopf, n.d.). Videre skal vi ha weekly sync med ekstern veileder ukentlig og med kunde minst annenhver uke. Vi kommer til å bruke "Limit the work in Progress" (Alqudah and Razali 2017, 3) fra Kanban.

Når det kommer til roller så har gruppa bestemt seg for at vi skal definere noen roller for å opprettholde arbeidsflyt, fremdrift, samt det å få tatt avgjørelser. Rollene er: Prosjektleder, som gjør Product Owners jobb med backlogen, Decider som kan ta enstemmig avgjørelser, Risikoansvarlig som har ansvar for at risikoplanen blir oppdatert og Dokumentasjonsansvarlig som har ansvar fremdriften av dokumentasjonen.

2.1.1.4. Hvorfor skal vi bruke Scrumban?

Det er mange fordeler med smidig metodikk. I "Project Management for Information Systems" (Cadle and Yeates 2008, 73 og 79) antyder de at smidig metodikk er et bedre alternativ enn



Fossefallsmetoden dersom kunden har en utfordring med å definere kravspesifikasjonene detaljert nok, eller hvis det er stor sannsynlighet for at det trengs å gjøres endringer underveis i prosjektet. I og med at vår oppgave er veldig åpen (ref kapittel 1.1) er det stor sannsynlighet for at det må gjøres endringer underveis. Her vil teknikkene sprinter, retrospective og jevn involvering av kunde (ref punkt 2.1.1.3) hjelpe oss til å produsere det kunden faktisk trenger. Dette er noe gruppen anser som viktig, da vi ønsker å lage et produkt som gjør at brukerne i Rebel-bygget ønsker å benytte seg av betalingsapplikasjonen og den in-house valutaen.

Det å jobbe i sprinter med hyppige små leveranser og evaluering underveis gir mulighet for læring og minimerer sannsynligheten for feil. God kommunikasjon er noe vi ser på som viktig for å lykkes, noe Scrumban muliggjør gjennom Daily Standup og Retrospective (ref punkt 2.1.1.2).

2.2. Valg av teknologier og Verktøy

2.2.1. Notion

Notion er et prosjektstyringsverktøy som gir god oversikt i prosjektet. Programmet er komponentbasert slik som Kanban-brett og kalender, hvor vi kan fritt velge hvilke komponenter vi skal ta i bruk. Notion gir oss muligheten til å samle informasjonen på ett sted, samt hente inn linker til alle dokumentene vi trenger. Dette vil gi oss god oversikt.

2.2.2. Figma.

Figma er et nettbasert prototype- og designverktøy som lar gruppen jobbe sammen over nettet i sanntid. Dette er viktig for oss, ettersom vi ikke kan møtes fysisk på grunn av Covid-19. Flere i gruppen har brukt Figma tidligere blant annet til brukertesting, hvor det er enkelt for testdeltakerne å dele skjerm og teste prototypen som om det var en ferdig applikasjon. Vi ønsker å benytte Figma til å lage trådkisser, lage prototyper samt gjennomføre designsprintene.



2.2.3. Slack

Slack er en kommunikasjonskanal hvor vi kan opprettholde kommunikasjon innad i gruppen, med ansatte hos Shortcut og ikke minst kunden vår. Ettersom all kommunikasjon er samlet på en plattform, vil det bli enkelt å følge med for alle i gruppen.

2.2.4. Google Docs

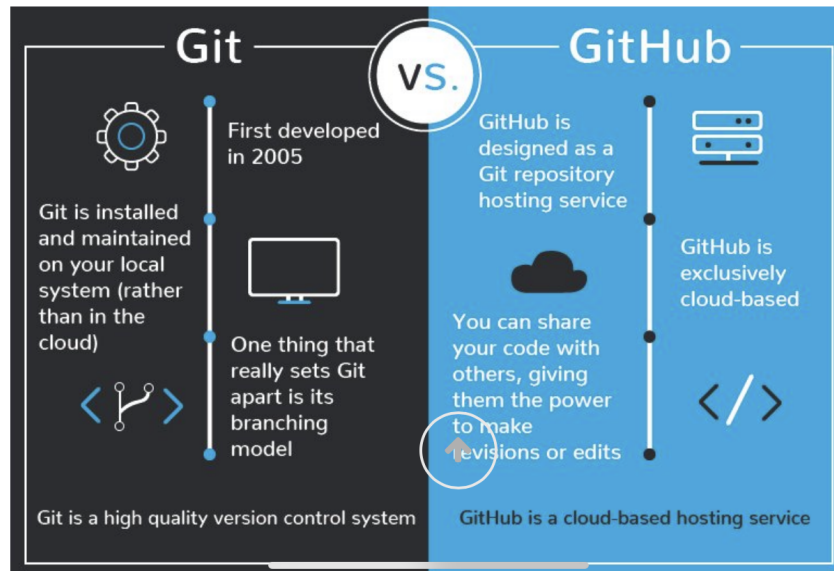
Google Docs er Google sitt nettbaserte skriveprogram. Det lar gruppen jobbe i samme dokument i sanntid, slik at gruppen vil være kontinuerlig oppdatert med nyeste versjon av dokumentet. En av funksjonene vi kommer til å dra nytte av er muligheten til å lage en referanseliste vi kan referere fra rett inn i teksten og opprette referanselisten i bunnen av dokumentet når vi er ferdig.

2.2.5. Digitalt videomøte

Gruppen kommer primært til å bruke Google Meets til møtevirksomhet. Vi kan booke videomøter rett fra Google Calendar, hvor vi blir en del av Shortcut sin kalender. Med Google Meet kan vi gjennomføre møter og dele skjerm for å vise frem nødvendig informasjon. Dette blir spesielt viktig i og med at Covid-19 situasjonen mest sannsynlig gjør at vi ikke kan møtes fysisk.

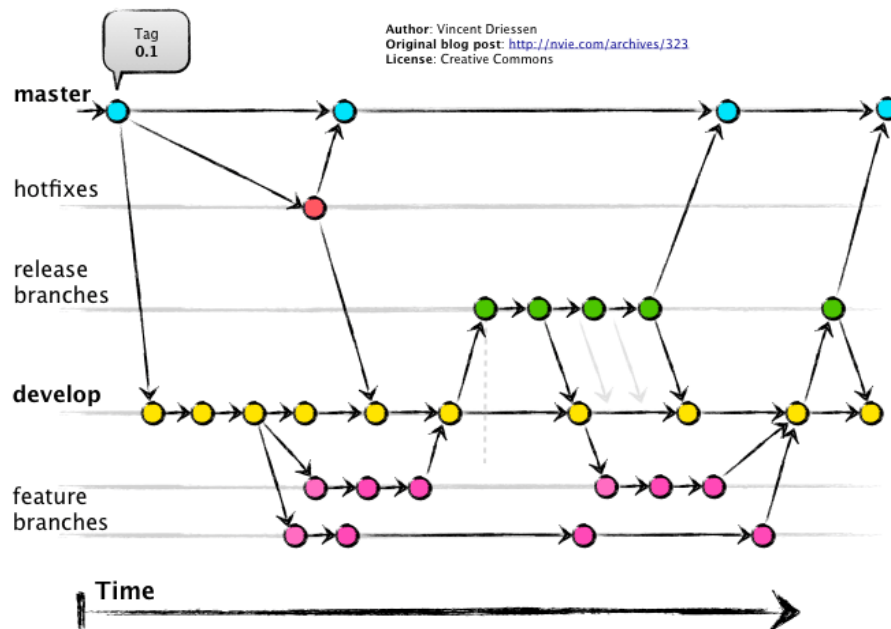
2.2.6. Git, Github og Git Workflow

Vi kommer til å benytte oss av Git som versjonskontroll med Github som vår skybaserte lagringsplattform. Det vil gi oss sikkerhet mot utilsiktet endring og tap av kode. Hvis noe slikt skulle oppstå kan vi rulle tilbake til tidligere versjoner. Som illustrert på *bilde 3*, gir verktøyene oss muligheten til å jobbe sammen på samme kodebase hvor alle har tilgang på den nyeste versjonen. Vi kommer til å få opprettet Github repositories av Shortcut på deres Github konto i og med at det er de som skal eie koden etter prosjektet.



Bilde 3: Oversikt over Git og GitHub (Devmountain, n.d.)

Vi kommer også til å sette opp prosjektet etter Git Workflow. Som vi kan se på *bilde 4*, kommer vi til å jobbe i en branch deretter flytte det til en ferdigstilt branch, slik at vi alltid har en fungerende versjon. De fungerende versjonene vil være sperret for direkte endringer.



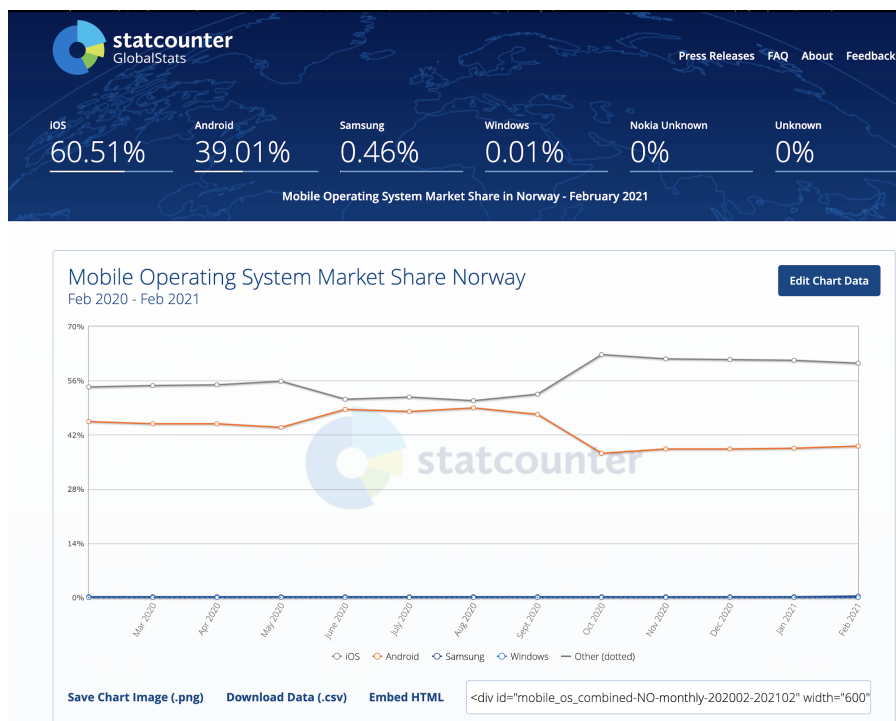
Bilde 4: Oversikt over Git Workflow branching struktur (Driessen 2010)

2.2.7. Betalingsteknologi

NFC og QR er to alternativer til trådløs betaling. Gruppen lener mot å bruke NFC i løsningen, da dette er en raskere prosess en QR. Videre i utvikling av applikasjonen bør det undersøkes om NFC kan brukes flerfunksjonelt i bygget for eksempel åpne dører.

2.2.8. Plattform

Vi ønsker å utvikle terminalen i Android og klienten i iOS, sistnevnte har størst markedsandel i Norge i dag som vist på *Bilde 5* og det gir derfor mening å utvikle klienten i iOS.



Bilde 5: Mobil operativsystem markedsandel Norge ("Statcounter", n.d.)

NFC funksjonaliteten er mer åpen på Android enn iOS (Google 2019) (Apple, n.d.), det vil derfor være mer aktuelt å bruke Android på terminalen. Gruppemedlemmene ser for seg å jobbe med Android og iOS etter endt skolegang, derfor vil videre læring på feltene komme til nytte. I tillegg vil det være tilgjengelige veiledere fra Shortcut til iOS og Android.

2.2.9. Database valg

Google Firebase er Google sin skybaserte databasetjeneste. Gruppen ønsker å benytte seg av Firestore på grunnlag av at den er fleksibel, skalerbar og holder dataene på klientapplikasjonen oppdatert via “realtime listeners” (“Cloud Firestore”, n.d.). I tillegg har Shortcut Google Partners-merke, som gir oss tilgang på unik kunnskap og hjelp.

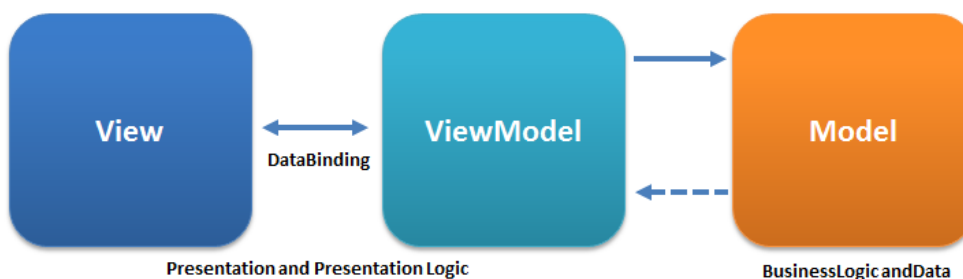
2.2.10. Editor (IDE) til Utvikling

Android Studio og Xcode er to IDE'er (integrated development environment). Vi kommer til å benytte oss av disse da de er spesifikt laget for Android og iOS utvikling. For mer detaljert innsikt i hva Android Studio og Xcode er og fungerer se vedlegg D

2.2.11. SwiftUI og MVVM

Vi kommer til å benytte SwiftUI med MVVM design struktur på klientapplikasjonen. SwiftUI er et moderne språk som er lett å lese og naturlig å skrive (“Apple Developer”, n.d.). Et moderne og fremtidsrettet bygg som Rebel trenger en applikasjon i det nyeste rammeverket, også med tanke på vedlikehold fremover. Der hvor SwiftUI ikke er hundre prosent utviklet enda kan vi bruke UIKit inn i SwiftUI gjennom `UIViewRepresentable()`.

For å få en god mappestruktur på filene vil vi bruke oppsettet til MVVM (Model View ViewModel), dette vil passe godt med SwiftUI. I MVVM og SwiftUI er tanken å koble View med ViewModel via DataBinding/ObjectBinding. View'et i modellen (*ref bilde 6*) får kontakt med ViewModel via DataBinding, og ViewModel har kontakt med Model.



Bilde 6: MVVM modell (Kouraklis 2016)

2.3. Gjøre rede for testing

2.3.1. Brukertesting

Det finnes ulike måter å få innsikt i hvordan brukere anvender de ulike løsningene. For eksempel blir Google Analytics mye brukt til innsiktsarbeid, men gir ikke tilbakemelding på hvorfor en bruker for eksempel hopper av i en betalingsprosess. Ved å gjennomføre brukertester er det lettere å få vite hva brukerne tenker og hvordan de interagerer med en løsning (Nordbø 2017, 158).

Det finnes flere måter å gjennomføre brukertester på. To eksempler er brukertester i lab og Geriljatesting. Sistnevnte er vanlig å gjennomføre ved å plukke ut folk på gaten og ber dem om å teste løsningen. Brukertesting i lab er en mer omfattende brukertest som krever mer forberedelser, for eksempel: mål for brukertesten, hente inn testdeltakere, klargjøre testplan og testoppgaver (Nordbø 2017, 159).

Grunnet Covid-19 situasjonen vil gjennomføring av Geriljatesting være vanskelig og brukertesting i lab vil heller ikke være ideelt. Vi ønsker derfor å ta utgangspunkt i brukertesting i lab, men gjennomføre de gjennom Google Meets.

Vi vil ta utgangspunkt i brukertestvettreglene beskrevet i "Interaksjonsdesign" (Nordbø 2017, 158).

Vi vil lage en grundig testplan som inneholder introduksjon til testen, testdeltakere, når testene skal gjennomføres, spørsmål og oppgaver og ikke minst målet for testen. Det er ønskelig å dele inn testene i tre: innledning hvor vi introduserer oss og stiller noen få spørsmål, hoveddel med oppgaver og avslutning hvor vi stiller spørsmål rundt løsningen testdeltakerne har vært igjennom. Alle oppgaver og spørsmål vil være laget på bakgrunn av målet for testen, da dette vil gi oss best innsikt ut fra hva vi ønsker å teste (Nordbø 2017, 161).



2.3.2. Enhetstesting

Enhetstesting brukes for å teste deler av kildekoden. Selv om det er viktig i en utviklingsfase tenker gruppen å vente med dette, da prosjektet er et proof of concept.

2.4. Prosjektplan

Vi ønsker å lage et GANTT diagram (vedlegg E) over prosjektet for å strukturere sprintene. Diagrammet er oversiktlig og vil inkludere de viktigste fasene av prosjektet. Vi er klar over at endringer kan forekomme utover i prosjektet og dette er noe vi må følge nøye med på. Milepælene i diagrammet vil bli inkludert ved bruk av stjerner. For å opprettholde hyppige leveranser tar vi utgangspunkt i at sprintene varer en uke.

2.5. Risiko og tiltaksplan

En Risikoplan er et grunnleggende verktøy i prosjekter. Det blir brukt til å kartlegge risiko som prosjektgruppen og prosjektet kan få konsekvenser av, hvor stor sannsynlighet det er for at risikoen faktisk skjer, hvilke tiltak man bør ta for å være rustet og hvem som er ansvarlig for å håndtere risikoen. Det kan medføre store forsinkelser og skader hvis det ikke er laget noen mal for hvordan man skal håndtere risiko. Hele gruppen kommer til å ha ansvar for å håndtere og følge med på risiko, men som nevnt i kapittel 2.1.1.3, vil vi utnevne en person som har ansvar for risikoplanen. En utarbeidet risikoplan for gruppen vil ta for seg de fleste tilfeller som stiller en risiko for prosjektet, fra liten til stor påvirkning. Den krever oppfølging og jevnlig re-evaluering. Hver fredag ønsker vi å ha gjennomgang av planen for å oppdatere og diskutere rundt den. Vi kommer også til å utarbeide en tiltaksplan som i detalj beskriver håndteringen av risiko om den inntreffer.



2.6. Datainnsamling

2.6.1. Brukerinnsikt

2.6.1.1. Personas

Ved å utarbeide ulike personas vil vi få god oversikt over målgruppen. En persona er en beskrivelse av en fiktiv person som utarbeides på bakgrunn av det vi allerede vet om målgruppen. For at personas skal ha en verdi er det viktig at de er mest mulig realistiske og spesifikke (Cooper 2004). Det er naturlig å inkludere: navn, alder, arbeidssted, stilling, ferdigheter, digital erfaring, interesser og bilde eller en skisse når man utarbeider en persona.

2.6.1.2. Brukerhistorier

Brukerhistorier (user stories) er vanlig å benytte seg av når man bruker en smidig metode. En brukerhistorie skal på en setning si hva en bruker har behov for å gjennomføre ved hjelp av det systemet vi skal lage. Enhver brukerhistorie skal inneholde:

- "Hvem som skal bruke løsningen"
- "Hva de trenger av løsningen"
- "Hvorfor de trenger det" (Nordbø 2017, 109)

En brukerhistorie formuleres slik:

"Som en <rolle> trenger jeg å <oppgave> fordi <mål>." (Nordbø 2017, 110)

Det er flere fordeler med å utarbeide brukerhistorier. Blant annet vil det gjøre det enklere for designerne å konkretisere brukerbehov og det kan brukes som en del av konseptet. Det vil også være nyttig for resten av teamet i og med at det kan brukes som kravspesifikasjon senere i prosessen (Nordbø 2017, 110). Vi ønsker å definere en rekke brukerhistorier som kan hjelpe oss med å avgrense løsningen og fokusere på brukerbehovene.

2.6.1.3. Brukerreise

En brukerreise er en visualisering av brukerens opplevelse av et produkt. Det kan være nyttig "for å identifisere problemområder og brukerbehov" (Nordbø 2017, 98).



Det å utarbeide et brukerreisekart gir en visuell oversikt over brukerreisen (Nordbø 2017). Vi kommer til å ta i bruk brukerreisekart da dette vil gi både oss og kunden en god oversikt over hva som bør være med i løsningen til slutt, samt hvilke deler som bør prioriteres fremfor andre.

3. Analyse og utforming

I dette kapittelet vil vi bli bedre kjent med personas'ene vi har utbrodert samt brukerreisen som har hjulpet oss med å visualisere hvordan en bruker opplever å ta i bruk applikasjonen. Videre vil ta for oss designmalen og noen av dets utfordringer. Her vil vi snakke litt om universell utforming og hvilke retningslinjer som er viktig for vår løsning. Kapittelet vil ta for seg workshopene og designsprintene vi har vært igjennom samt hvordan vi endte opp med den endelige prototypen. Vi vil se nærmere på analysen av brukertestene. Videre tar vi for oss kodingen av terminal og klientapplikasjonen og dets utfordringer.

3.1. Personas

For å få en dypere forståelse av målgruppen kan vi utarbeide personas (ref punkt 2.6.1.1). Rebel hadde utarbeidet ulike personas fra deres datainnsamling. Ettersom prosjektet er avgrenset var ikke alle personas'ene som Rebel hadde laget, like relevant for oss. Vi bestemte oss derfor for å ta utgangspunkt i deres personas, men gjøre endringer slik at de er tilpasset prosjektet. Se vedlegg F for å lese om utformingen.

Det er ulike meninger om det å bruke personas er bra eller ikke (Nordbø 2017, 101,102).

Ettersom vi skal lage en løsning hvor brukerne har nokså ulike ferdigheter når det kommer til IT-kompetanse, ønsker vi å utarbeide fem personas for å være sikre på å nå ut til målgruppen vår. På denne måten får vi bedre forståelse av brukerbehovene.



3.2. Brukerreisen

Brukerreisekart er en oversikt over hvordan vi ser for oss at en bruker kommer til å bruke løsningen fra de oppdager den til de er ferdig med å bruke den. Som vi ser på bildet i vedlegg G, så er det to scenarioer. Enten kommer brukeren som gjest til bygget, for eksempel for å møte noen, eller at brukeren jobber der fast. I begge tilfellene blir brukeren oppmerksom på applikasjonen på et tidspunkt. Neste steg blir at brukeren bestemmer seg om hun skal laste ned applikasjonen, eller bruke App-Clips, som vil si at man bruker deler av applikasjonen uten å laste den ned. Herfra blir brukeren litt forskjellig. I og med at sistnevnte alternativet er veldig mye enklere, så blir det i denne sammenhengen å aktivere App-Clips på tapping og gjennomføre betalingen. Her vil brukeren også få muligheten til å laste ned applikasjonen, så Rebel kan bruke dette for å reklamere for applikasjonen.

Hvis brukeren velger å laste ned applikasjonen så må hun registrere seg og logge inn. Registrering må være så enkel som mulig for at det ikke skal være noe hinder. Etter innlogging får brukeren fire muligheter som vist på *bilde 7*;

1. Bruker kan overføre penger til Rebel Cash. Det gjøres i få steg for at det skal være enkelt, samt oversiktlig slik at brukeren har full forståelse.
2. Bruker kan se sine lojalitetsfordeler. Det kan være klippekort på kaffe, eller rabatt på visse tidspunkt og andre tilbud, men også hvilken Rebel Level som er nådd.
3. Bruker kan betale. Her har brukeren to muligheter. Enten å betale med NFC tapping mot en terminal, hvor butikken har scannet varene og brukeren kan velge ønsket tilbud.
4. Bruker kan se sine kvitteringer. Her kan brukeren få full oversikt over hva som er kjøpt og hva det koster. Her er det også en mulighet for å dele de kvitteringene man ønsker med arbeidsgiver.

Bilde 7: Brukerreisen



Dette brukerreisekartet vil hjelpe oss til å ha en klarere oversikt over hva vi skal lage og hva vi skal fokusere på fremover i prosjektet. For Rebel så vil denne oversikten hjelpe med å forstå hva vi kommer til å lage, hva som er muligheter for applikasjonen og bruke det i en salgsargument for sine sjefer

3.3. Design

3.3.1. Designmal

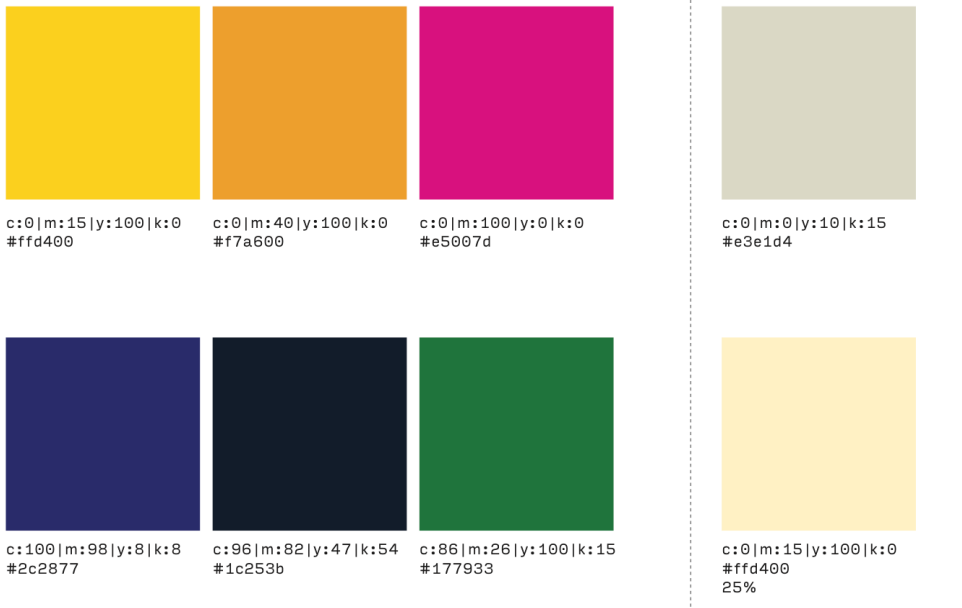
Etter et møte med Entra fikk vi tilsendt en designmal. Designmalen består av logoer, plassering, grafiske støtteelementer, farger, fonter og fotomanér vedlegg H. Ved bruk av designmalen vil vi ha lettere for å ta valg på hvordan ting skal se ut og det er en måte å fastsette ting.

3.3.1.1. Farger

Bilde 8 nedenfor viser fargene Rebel ønsker at vi skal ta i bruk. I løsningen vil man kunne se flere farger som går igjen i større grad, blant annet rosa, orange og blå. Det var noe vanskelig å skulle kombinere de sterke fargene med bakgrunnsfargene som er i designmalen. Gruppen bestemte seg derfor å ta kontakt med en designer hos Shortcut for rådgivning. Designeren var helt enig i at det ikke var de enkleste fargene og foreslo derfor at bakgrunnsfargen på hovedsidene kunne holdes hvite, men at innlogging- og "Register"-sidene kan designes med de sterke fargene.

For å møte WCAG sine retningslinjer sjekket vi alle fargene (*bilde 8*) vi brukte opp mot hverandre. Det viste seg at den orange fargen ikke møtte kravene. Derfor bestemte vi oss for å gå utenom designmalen og finne en farge som har større kontrast mot hvit.

Les mer om dette under punkt 3.6.3.



Bilde 8: Fargene i designmalen fra Rebel

3.3.2. Designprinsipper

For å sikre god brukeropplevelse er det lurt å ta utgangspunkt i designprinsipper.

Designprinsipper er retningslinjer for god design, de er basert på hvordan mennesker lærer, husker og oppfatter ting (Johnson 2014). I designprosessen har vi vektlagt designprinsipper: “synlighet”, “hint” og “tilbakemelding” ettersom det er ønskelig at løsningen skal være lettlært og selvforklarende. Brukerne skal kunne navigere seg logisk gjennom løsningen. Det er gjort bevisste valg rundt tekst og knapper, for å tydeliggjøre hva man kan trykke på. Vi er opptatt av å gi tilbakemelding til bruker for å unngå frustrasjon og eller forvirring, blant annet vil brukeren etter å ha endret den firesifrede koden, få tilbakemelding om at den nye koden er lagret.

3.3.3. Universell utforming

Universell utforming går ut på å tilpasse løsningen slik at den blir tilgjengelig for alle, inkludert funksjonsutfordrede. WCAG (Web content accessibility guidelines) er retningslinjer vi må følge for å tilgjengeliggjøre løsningen, det er tre ulike grader av oppnådd accessibility: A, AA og AAA.

Kandidatnummer: 8029, 8124, 8061, 8001, 8176

Sistnevnte er den høyeste graden. Man er nødt til å følge flere retningslinjer desto mer man ønsker å tilgjengeliggjøre løsningen. Her er det også viktig å huske på at løsningen blir ikke bare mer brukervennlig og tilgjengelig for de med funksjonsutfordringer, løsningen vil bli enda bedre å bruke for alle (Nordbø 2017, 59). Det finnes ulike funksjonsutfordringer;

- “Sensoriske nedsettelse som er knyttet til, syn, lukt, hørsel, balanse og berøring”
- “Motoriske nedsettelse som er knyttet til alt som har med motorikk og bevegelse å gjøre”
- “Kognitiv nedsettelse som kan være nedsatt læringsevne eller hukommelse eller konsentrasjonsvansker” (Nordbø 2017, 62)

Vi vet at løsningen vil bli tatt i bruk i et bygg hvor brukerne kommer fra ulike steder i verden, det er derfor logisk at løsningen vil være på engelsk slik at vi når ut til flere.

Med utgangspunkt i retningslinjene for universell utforming beskrevet av W3C i WCAG 2.0 (Caldwell et al. 2008), har vi i stor grad fokusert på kontraster i tekst og grafiske elementer. Etter å ha testet fargene fra designmalen så vi at den orange fargen (fargekode #F7A600) ikke hadde høy nok kontrast mot fargen hvit (#FFFFFF) og motsatt. Vi bestemte oss derfor for å endre denne fargen til (#E96201) som vist på *bilde 9* under. Etter å ha endret denne fargen oppnår alle fargene “AA18pt”, som betyr at fargen består AA dersom fonten er større enn punkt 18. Derfor vil teksten i løsningen med farge ha punkt 19 eller større.

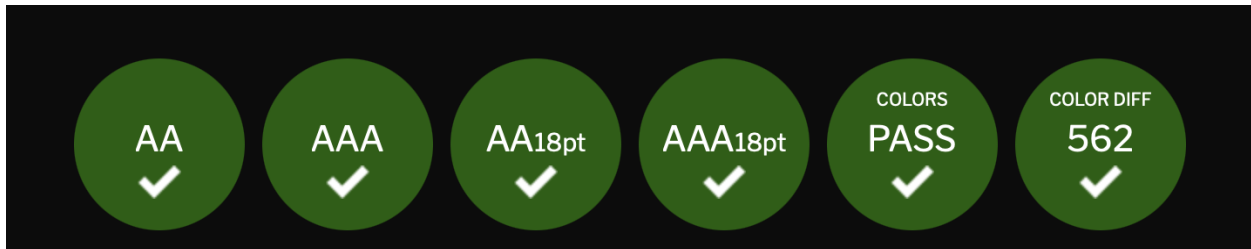


Bilde 9: Endring av farge, bilde viser den nye fargen.



Kandidatnummer: 8029, 8124, 8061, 8001, 8176

Når det gjelder resten av fargene, oppnår blå (#2C2877) full uttelling som vist på *bilde 10*. Rosa (#E5007D) består alle kriteriene utenom AAA og “Color diff”, hvilket betyr at fargen består AA men er ikke like lett leselig for folk som er fargeblinde. Ettersom at rundt 10% av menn er fargeblinde (Preece, Rogers, and Sharp 2015) er det ønskelig å ha høy kontrast på fargene vi bruker. Den vanligste formen for fargeblindhet er evnen til å skille mellom rødt og grønt (Nordbø 2017, 60,61). Fordi rosafargen blir brukt mot sort og hvit bakgrunn har vi fokusert mindre på “Color diff” kontra AA.



Bilde 10: Fremvisning av full uttelling på testing av farge

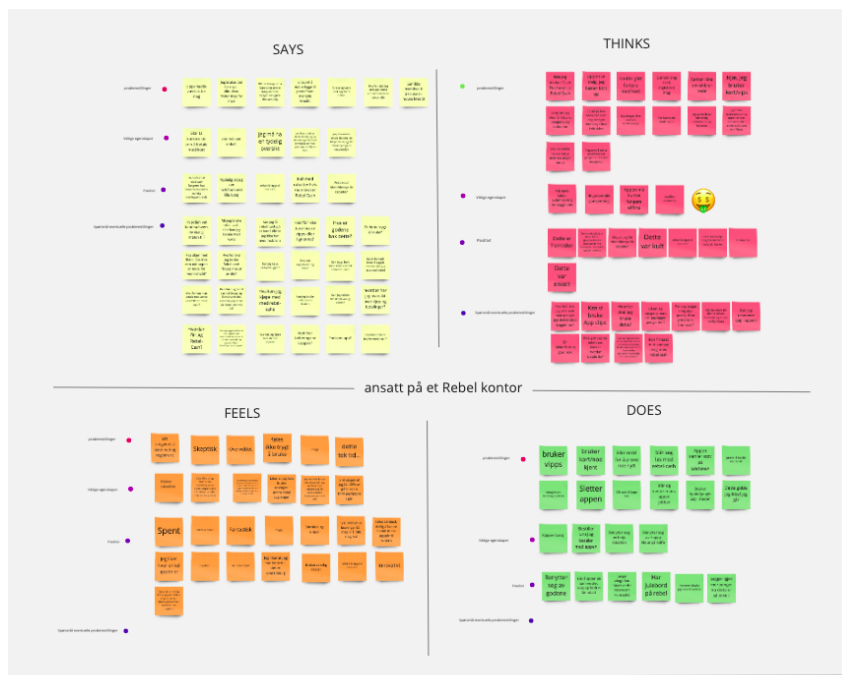
For å gjøre det enklere for de som har motoriske funksjonsutfordringer, som for eksempel betennelse i håndledd, har vi lagt navigeringen til hovedsidene i en tab bar som er plassert nederst i applikasjonen. På denne måten slipper brukerne og strekke tommelen øverst på mobilen, eller i verste fall bytte grep på mobilen.

Generelt i løsningen har vi hatt fokus på prinsippene “Mulig å betjene” og “Forståelig”. “Mulig å betjene” går ut på at “det må være mulig å betjene brukergrensesnittkomponenter og navigeringfunksjoner” (Nordbø 2017, 66). Løsningen er navigerbar slik at brukerne finner innhold samt vet hvor i løsningen de befinner seg. For eksempel ikonet i tab baren får en annen farge når brukeren befinner seg på den siden. “Forståelig” betyr at “det må være mulig å forstå informasjon og betjening av brukergrensesnitt”. Prinsippet oppfylles da innholdet i løsningen er forståelig og leselig samt løsningen hjelper brukerne å unngå feil. Et eksempel er når brukeren har laget en firesifret kode og ser at koden ble feil, vil brukeren få muligheten til å endre den med en gang (Nordbø 2017, 64, 65).

3.4. Workshop

3.4.1. Empatikart

I starten av prosjektet synes vi at det var vanskelig å vite hva som bør prioriteres og vite hva vi skulle lage. Vi bestemte oss derfor for å dra nytte av ekspertene vi har til rådighet. Vi kontaktet en designer i Shortcut som hadde flere gode råd. Blant annet fikk vi tips om workshops og verktøy vi kunne benytte oss av, i og med at vi ikke kunne møtes. Et at verktøyene var “Miro” som lar oss gjennomføre workshops digitalt. Her lagde vi et empatikart for å bedre forstå brukerne av konseptet. Et empatikart er en visualisering som brukes for å fremstille det vi vet om en bestemt type bruker. Vi delte kartet inn i fire deler: Says, thinks, feels og does (ref bilde 11). Under alle rubrikkene gikk vi inn i rollen som en bruker av løsningen. Vi gav oss selv både konstruktiv og positive tilbakemeldinger, slik som: “er løsningen sikker å bruke”, “jeg føler applikasjonen lagrer for mye av mine opplysninger” og “dette var enkelt”.



Bilde 11: resultatet fra “empati kart” workshopen



Bilde 12: Oversikt over ting vi må fokusere på videre i prosessen

Workshopen hjalp gruppen å bryte ned prosjektet til mindre deler. Slike kommentarer som nevnt ovenfor (*bilde 12*), lot oss jobbe mer målrettet mot løsningen og vi forsto mer av brukeropplevelsen av konseptet. Vi fikk en bedre oversikt over hva som er viktig å fokusere på videre. Vi fikk også god innsikt i hvordan brukerne kan reagere på en slik applikasjon. Vi kom over flere fallgruver vi ønsker å unngå, for eksempel innhente for mye informasjon om brukerne. Ved å unngå å samle inn mye informasjon av brukeren under en registrering, vil terskelen for å registrere seg være lavere, med tanke på sparing av tid og sikkerhet. For å minimere feil og risikoen for forsinkelser er slike fallgruver en fordel å finne ut av så tidlig som mulig. Som vist på *bilde 12* er dette ting vi ønsker å ha fokus på videre, ettersom at dette er potensielle grunner til at brukere ikke ønsker å anvende applikasjonen.

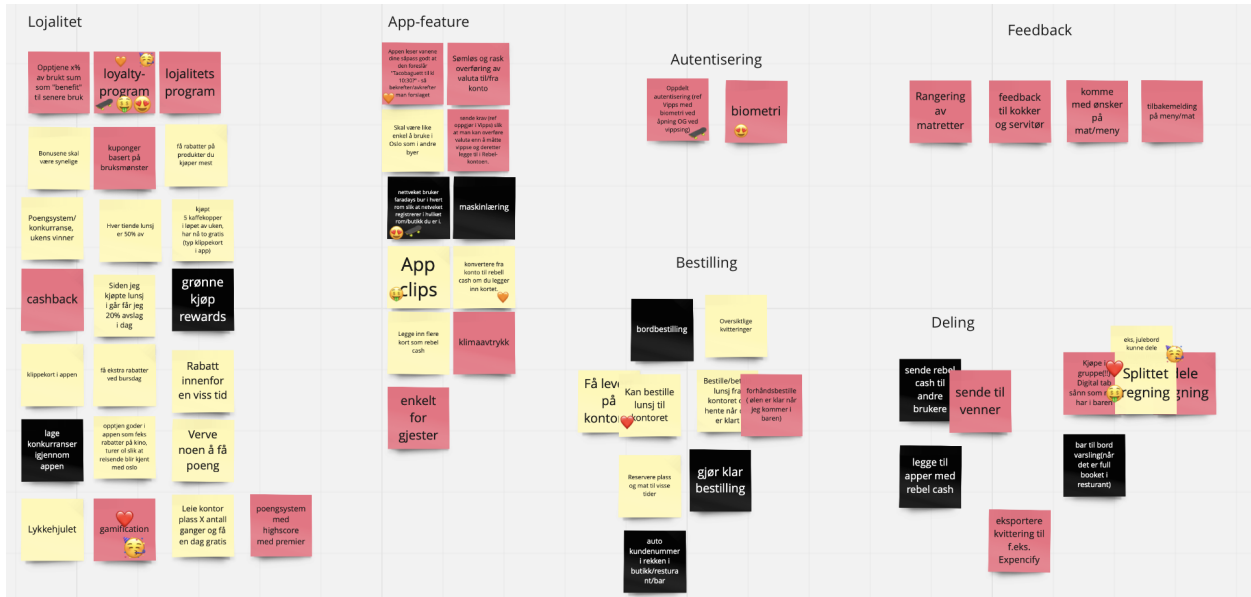
3.4.2. “Low hanging fruit”

Vi gjennomførte workshopen “Low hanging fruit” for å finne ut av hvilke funksjoner vi har kapasitet til å implementere i løsningen, basert på tid og ressurser. Den går ut på å kartlegge hvor lang tid, hvor komplisert og hvor viktig de ulike funksjonene er for løsningen. Her begynte



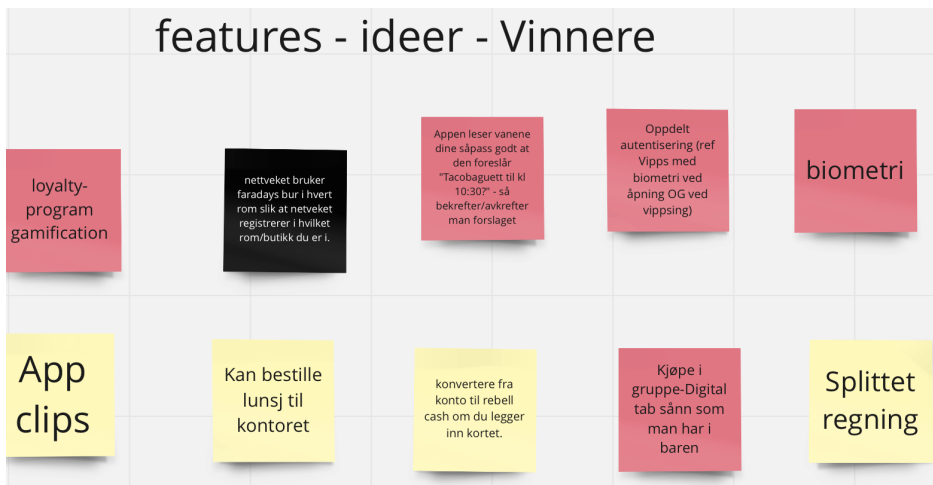
Kandidatnummer: 8029, 8124, 8061, 8001, 8176

vi med å skrive ned alle funksjoner vi ønsket å ha med, deretter sorterte vi de som vist på *bilde 13*.



Bilde 13: viser resultatet av alle ideer og funksjonalitet gruppen kom opp med

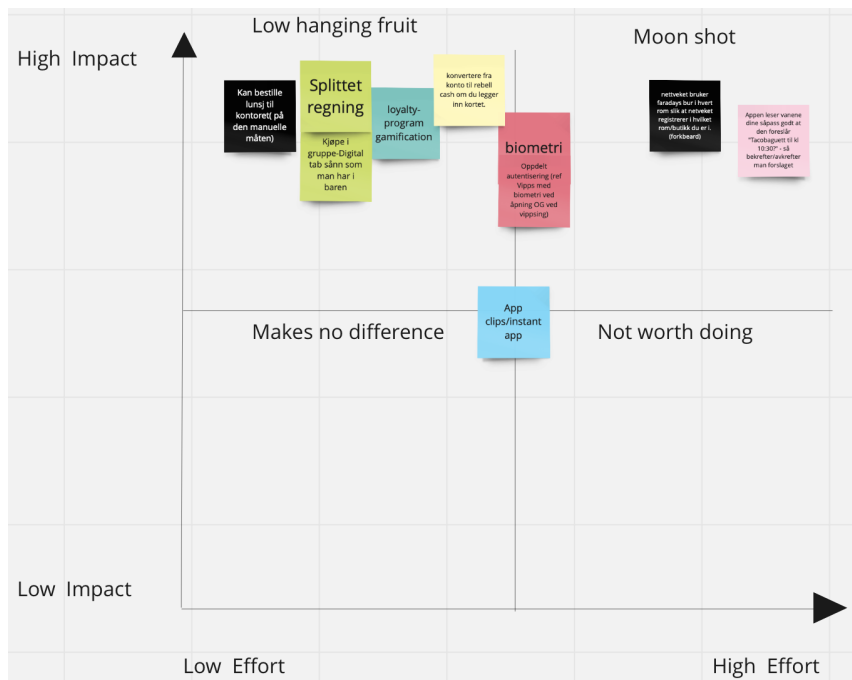
Deretter gikk vi gjennom de enkelte funksjonene og ideene og kom frem til noen post-it lapper vi ønsket å gå videre med, som vist på *bilde 14*.



Bilde 14: viser ideene vi valgte å ta med videre i workshopen.

Kandidatnummer: 8029, 8124, 8061, 8001, 8176

Videre sorterte vi funksjonene ut fra hvor viktige de er for løsningsens sluttresultat og hvor vanskelig det vil være å implementere de. Her kom vi frem til flere ting som vil være gunstig for oss å gjennomføre, for å gjøre applikasjonen mer interessant og lettere for brukeren å anvende. Vi fant ut at to av ideene vil være vanskelig for oss å gjennomføre grunnet mangel på ressurser og tid, dette gjorde at vi la fra oss disse ideene i første omgang. Videre ønsker vi å ha fokus på lojalitetsprogram samt gi brukeren muligheten til å overføre penger til Rebel Cash, da sistnevnte er essensiell for applikasjonen.



Bilde 15: resultat av "low hanging fruit".

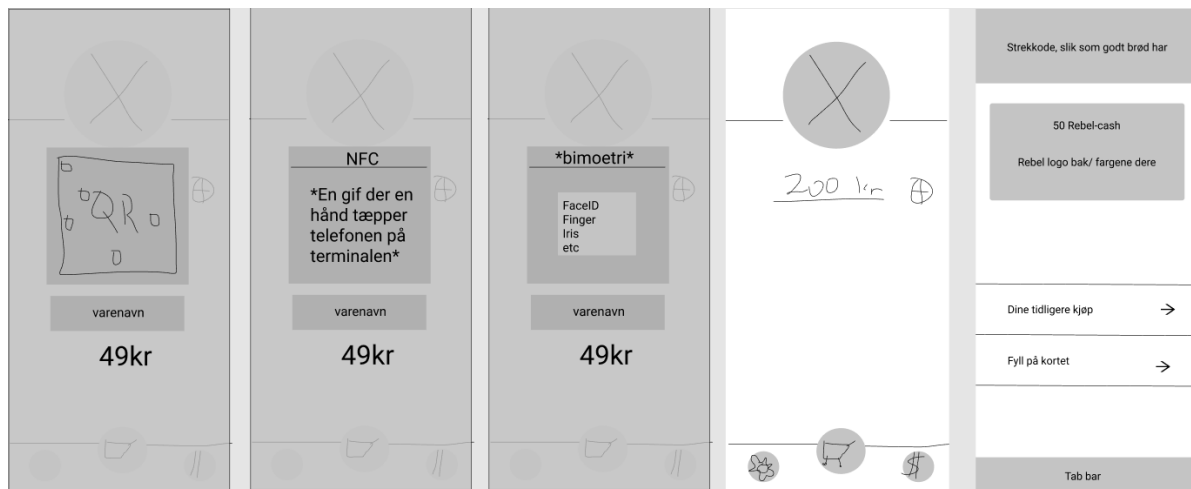
Uten denne workshopen er det fare for at vi ville tatt på oss for mye arbeid. Grafen som vist på bilde 15 hjalp gruppen å finne ut av hvor lang tid og hvor viktig de ulike funksjonene vil være for løsningen. Dette var en enklere måte å komme frem til hva vi kan og vil ha med videre, uten lange diskusjoner.

3.5. Designsprint

3.5.1. Klientapplikasjon

3.5.1.1. "Need to have"

Ettersom at vi fikk konkretisert ønsket funksjonalitet i workshopene kunne vi begynne med designsprint. Første sprinten ble utført på klientapplikasjonen, hvor vi i første omgang skisserte opp hva vi anser som "need to have", som vil si hva som absolutt må være med for å møte Entra sine ønsker. Dette innebærer; visning av saldo, initiere en betaling og se transaksjoner. *Bilde 16* viser sammensetningen av skissene vi utbroderte og *bilde 17* er prototypen som ble skissert opp senere og gjort klikkbar.



Bilde 16: sammensetning av skissene vi utbroderte. Skissene visualiserer førsteutkast av hvordan en betaling kan gjennomføres samt hvordan forsiden kan se ut.



Bilde 17: klikkbar prototype. Bilde til venstre viser hvordan forsiden vil se ut, bilde nummer to fra venstre viser de ulike transaksjonene, om brukeren trykker på en av disse transaksjonene vil brukeren komme til bilde som er i midten som fremstiller en kvittering. De to bildene til høyre viser hvordan brukeren kan overføre fra norske kroner til Rebel Cash.

Ut i fra workshopene hadde vi nå flere ideer på hva vi ønsket å ha med i løsningen.

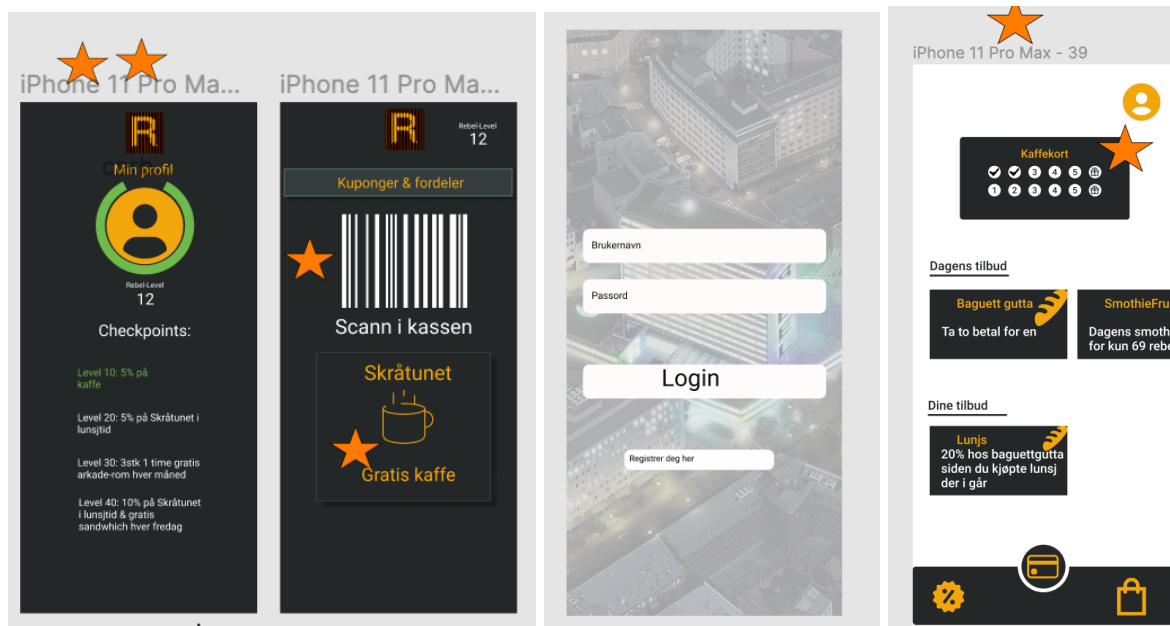
Designsprinten hjalp oss med å komme frem til ønsket oppsett, hvordan flyten skulle være og hvordan gjøre løsningen så enkel som mulig å ta i bruk. For å begynne med oppsettet var dette essensielt for både flyten og hvordan løsningen er å bruke. Ved at alle på gruppen kom med forslag, klarte vi å bygge på hverandre sine ideer og finne et oppsett på de ulike sidene. Vi valgte å stemme på det oppsettet vi synes var best, deretter sette det sammen på en logisk måte. Ved å starte med oppsettet har vi muligheten til å brukerteste om flyten i løsningen er logisk for brukerne, og finne ut av hva brukerne forventer å kunne gjøre i løsningen. Ved å brukerteste tidlig får vi muligheten til å tilrettelegge løsningen ut i fra hvilke behov brukerne har, slik at vi kan designe en løsning som er brukervennlig.

Videre kan vi implementere tilleggsfunksjoner for å skape mer interesse i applikasjonen, da applikasjonen ikke bare skal brukes til betaling, men samle nyttige verktøy for å forenkle hverdagen til brukerne.

3.5.1.2. “Nice to have”

Gruppen bestemte seg for å gjennomføre en designsprint på de elementene vi anser som “nice to have” funksjoner, altså ting som er fint å ha men ikke nødvendig for å møte kunden sine behov. Selv om vi hadde mange tanker på ønsket funksjonalitet etter workshopene vi har gjennomført bestemte vi oss for å ikke konkretisere elementer som skulle være med. For oss var dette en måte å stimulere kreativitet og komme opp med ulike ideer.

Bilde 18 viser noen ideene vi ønsker bygge videre på.



Bilde 18: Viser ulike ideer på hvordan kuponger og tilbud fremstilles. Bilde nummer 3 fra venstre viser en ide på hvordan logg inn siden kan se ut.

Denne designsprinten gikk ut på å visualisere ideer gruppemedlemmene satt inne med. Her fikk vi en pekepinn på hvilke tilleggsfunksjoner flertallet i gruppen så for seg var nyttig for brukere. Sprinten var en gylden mulighet for gruppemedlemmene å visualisere hvordan “nice to have” funksjonaliteten kunne se ut. Ved å skissere opp ulike ideer, fikk flere på gruppen flere kreative ideer. Dette var en av grunnen til at valgte å ikke konkretisere hva vi skulle skissere opp på forhånd. Som en følge av det førte dette til et belønningssystem kalt Rebel Level. De ulike skissene med “nice to have” funksjonene ble stemt over. Resultatet ble deretter satt sammen

Kandidatnummer: 8029, 8124, 8061, 8001, 8176

med “need to have” funksjonene og brukertestet. Vi ønsket å få tilbakemelding på om funksjonene vi har kommet frem til er nyttige for brukerne og om de er gunstige å ta i bruk, mer om dette i brukertestene under punkt 3.7.

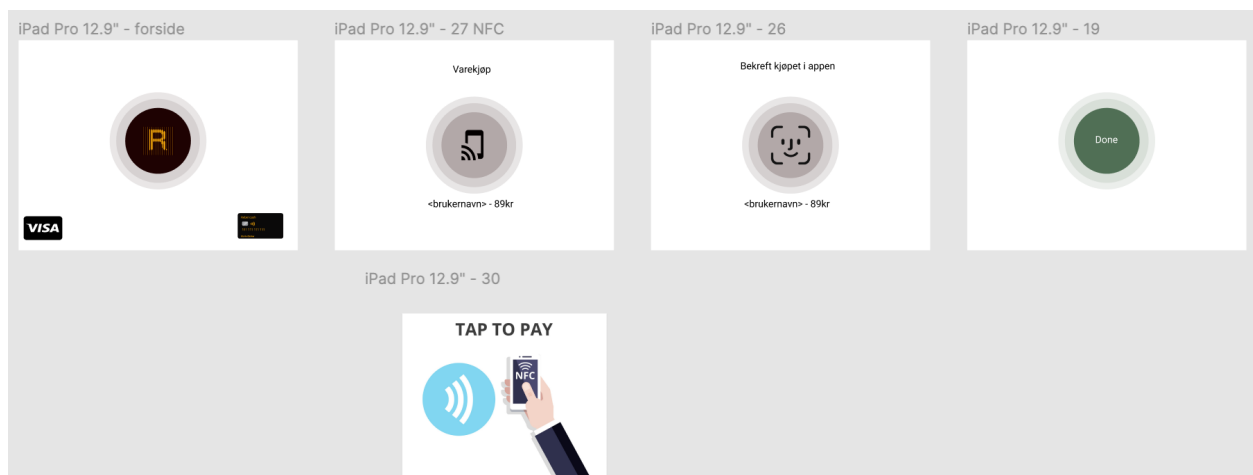
3.5.2. Terminal

Designsprinten for å konstruere terminalen ble gjennomført på samme måte som klientapplikasjonen. Det var to av tegningene som fikk flest stemmer, vi endte opp med en kombinasjon av disse to løsningene. Den ferdige wireframen av terminalen ble lagt inn i en egen page i figma, sammen med wireframen av klientapplikasjonen.



Bilde 19: skissene fra designsprinten som fikk flest stemmer

Ut fra tegningene skissert vi opp en kombinasjon av de ulike ideene. *Bilde 19* ovenfor er skissene som fikk flest stemmer. *Bilde 20* nedenfor er resultatet av ferdig oppsett på terminalen som viser hvordan terminalen tar brukeren gjennom en betaling fra start til slutt.



Bilde 20: Klikkbar prototype som tar brukerne gjennom en betaling.

Da vi gjennomførte designsprint på terminalen, var det for å definere kundereisen når en bruker betaler med en mobil mot en terminal. Her var alle enige om at få steg var det viktigste å tenke på, dette tok vi med oss videre i prosessen. Vi ønsket at løsningen skulle være så enkel som mulig og kun den mest nødvendige informasjonen skulle vises. Ved å skissere opp ulike ideer klarte gruppen å komme frem til den enkleste reisen gjennom en betaling. Her kom det også frem ulike ønsker for hele reisen, slik som få klikk og enkelt oppsett.

(Se vedlegg I for å lese mer om designsprintene)

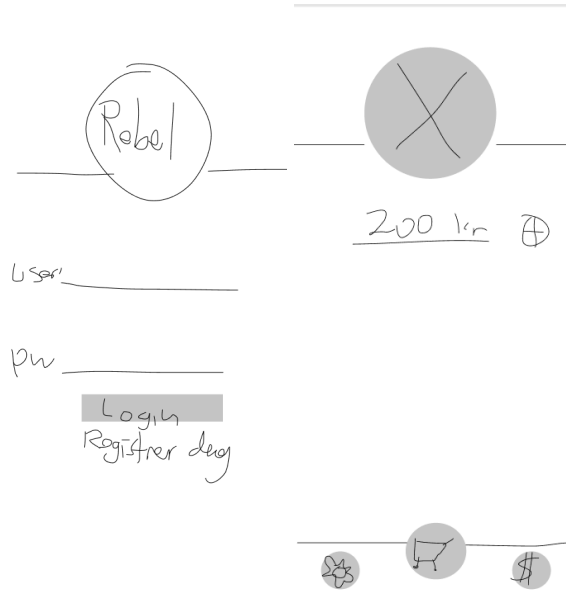
3.6. Prototype

En prototype kan være med på å vise hvordan hele eller deler av en løsning er tenkt til å fungere. Det kan være alt fra skisser på papir til fungerende løsning i en nettleser. Det er flere grunner til at man lager en prototype:

- “Ha noe vi kan bruke som grunnlag for å diskutere med team og kunde/produkteier”
- “Kunne teste ideer for oss selv og finne ut hvilken som fungerer best”
- “Kunne teste løsningen med representanter for målgruppene”
- “Kunne teste ut teknisk funksjonaliteter” (Nordbø 2017, 147)

3.6.1. Lavnivå prototype

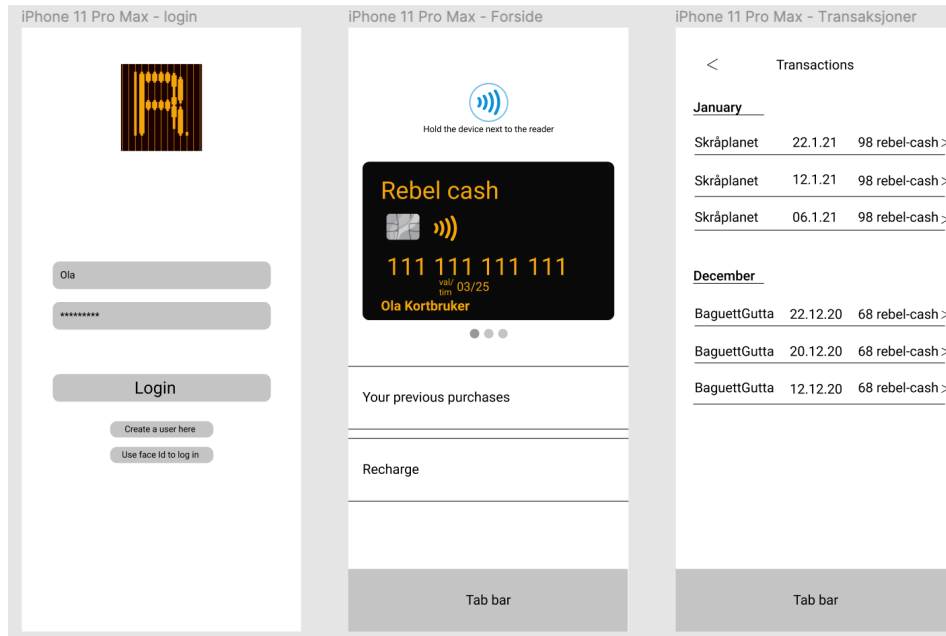
Vi bestemte oss for å lage en lavnivå prototype. Grunnet Covid-19 lagde gruppemedlemmene hvert sitt forslag på hvordan sidene skulle se ut ved bruk av Figma. Vi skisset opp ulike forslag med enkle tegninger som kun skulle visualisere plasseringen til ulike elementer, som vist på *bilde 21*.



Bilde 21: viser skissene gruppe medlemmen lagde i designsprinten. Bilde til venstre viser hvordan en logg inn kan se ut, bilde til høyre viser "Home"-siden.

Ut fra de skissene som ble stemt frem i designsprinten, lagde vi en litt mer detaljert klikkbar prototype, slik at vi kunne brukerteste strukturen og navigasjonen. Vi skiller mellom *vertikal* og *horisontal prototyping*. *Horisontal prototyping* vil si at man legger inn flere funksjoner, med få detaljer, i motsetning til vertikal prototyping hvor man fokuserer på få funksjoner med flere detaljer (Sandnes 2018). Vi valgte vertikal prototyping ettersom vi i første omgang fokuserte på de viktigste funksjonene for å møte kundens behov. [Link til lavnivå prototype](#)

Ut i fra lavnivå prototypen (Bilde 22) kunne vi teste løsningens navigasjon og brukernes forventninger til løsningen før vi begynte med høynivå prototypen. Under dette stadiet hadde vi noe å vise frem til kunde for å få tilbakemelding.



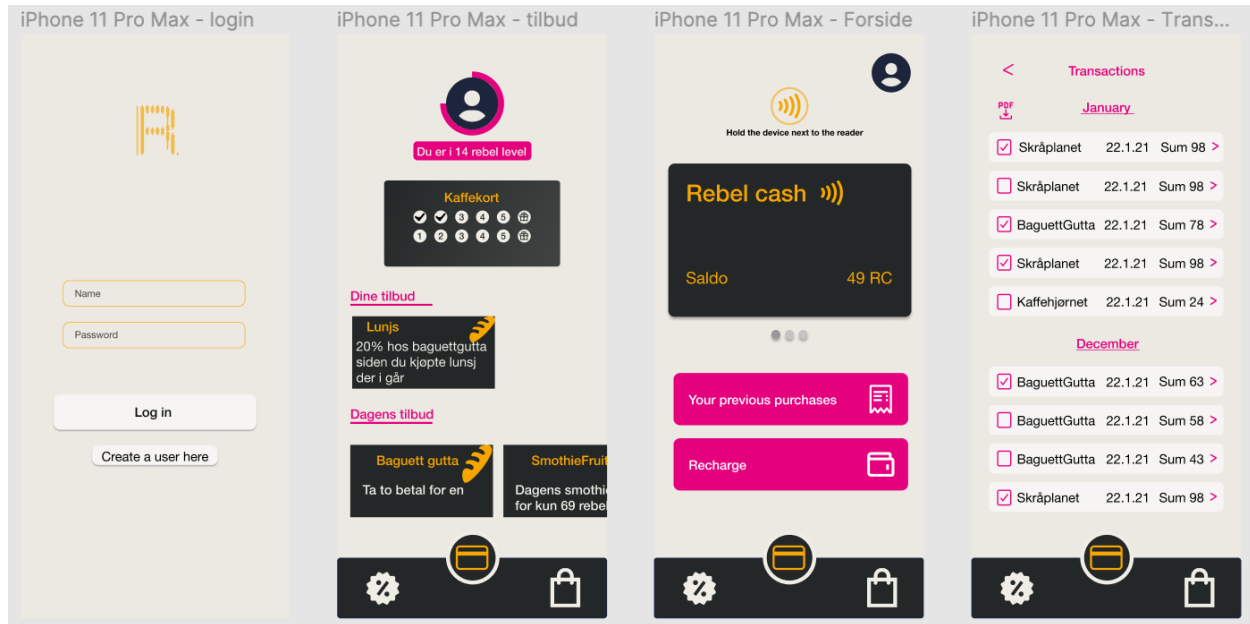
Bilde 22: Viser lavnivå skissene som ble laget med utgangspunkt i designsprinten.

3.6.2. Høynivå prototype

Høynivå prototyper blir gjerne laget med likt materiale som den ferdige løsningen skal lages med. Altså vil det være noe som vises frem digitalt og som gjerne går an å interagere med. Derfor vil det også ligne på den ferdige løsningen (Preece, Rogers, and Sharp 2015). Vi valgte derfor å lage prototypen i Figma. Ved å bruke figma kan vi legge inn ulike funksjonaliteter og en bruker kan interagere med løsningen som om det var en ferdig applikasjon.

3.6.2.1. Versjon 1

Vi begynte å eksperimentere med de ulike fargene fra designmalen. På *bilde 23* kan man se første høynivå prototype med fargene fra designmalen.



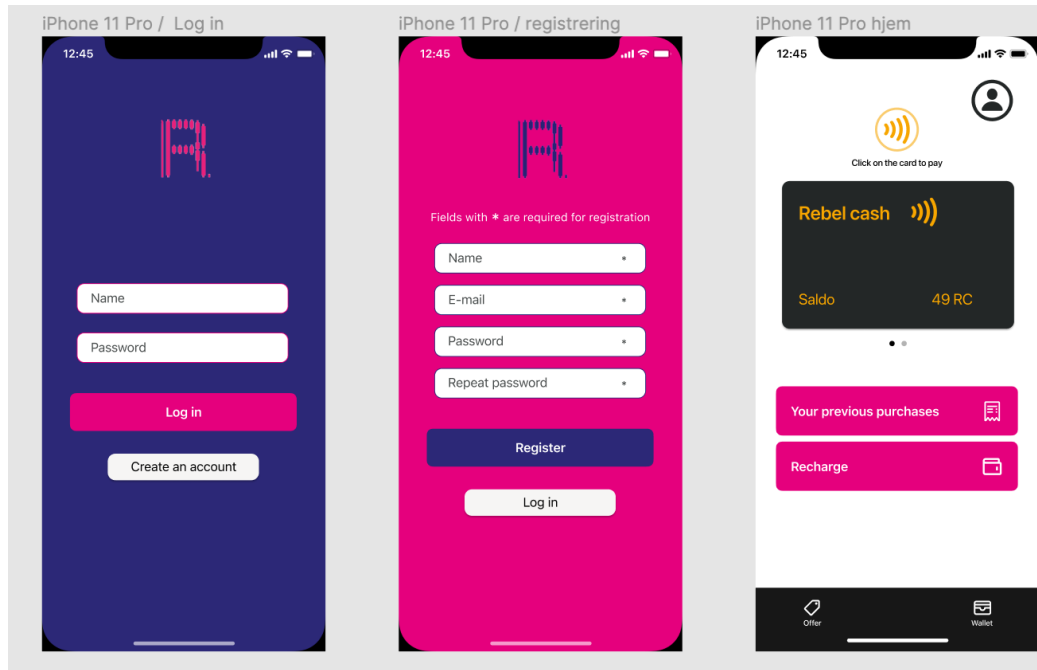
Bilde 23: Første høynivå prototype med farger fra designmalen. Bilde til venstre viser logg inn skjermen, neste bilde viser tilbudssiden, deretter "Home"-siden og helt til høyre er oversikt over tidligere kjøp.

Ettersom at fargene er nokså rebelske ga det noen utfordringer (ref punkt 3.5.1).

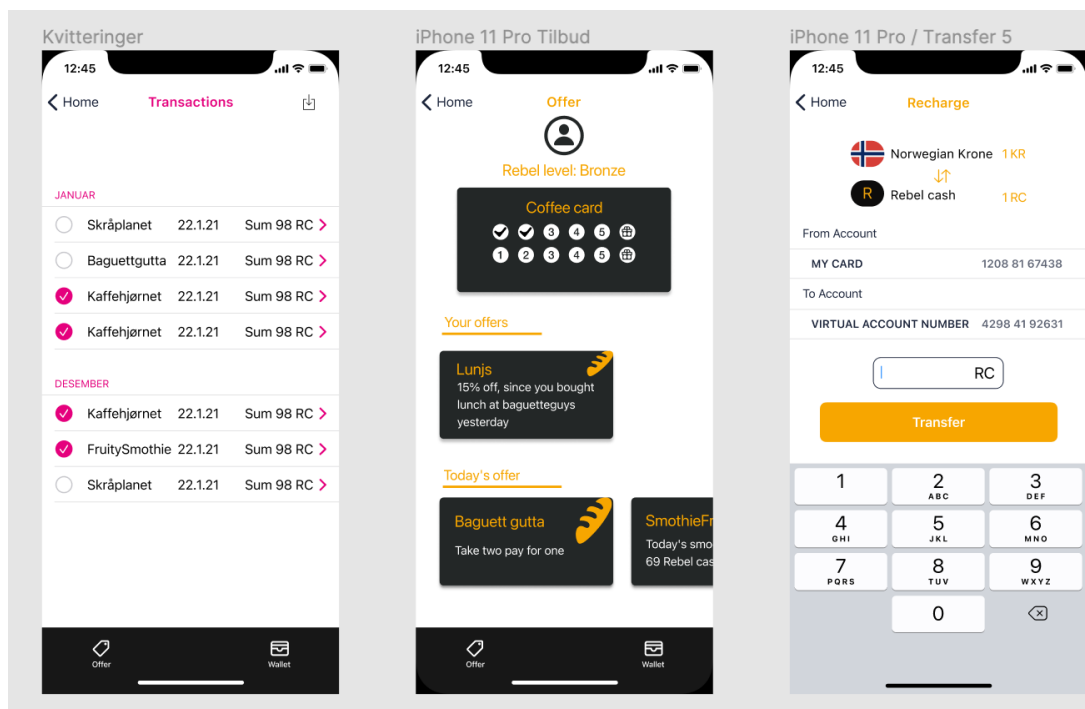
[Link til høynivå prototype Versjon 1](#)

3.6.2.2. Versjon 2

Etter å ha kontaktet en designer i Shortcut for rådgivning om fargevalgene, endte vi opp med større fokus på de vågale fargene på "Login"- og "Register"-siden. Vi vil også minimere de sterke rebelske fargene på hovedsidene, slik som vist på *bilde 24*. I og med at applikasjonen blir i første omgang utviklet for iOS, har vi brukt flere elementer i løsningen som er standard oppsett for iOS. Vi lastet ned "iOS/iPadOS 13 Design UI Kit (Community)" fra Figma, for å lettere kunne ta i bruk elementene. For eksempel "tab bar" som er plassert nederst på telefonen hvor man enkelt kan gå fra en side til en annen. Elementet "status bar" er plassert øverst i applikasjonen og viser klokkeslett, batteri og wifi som vist på bilde *bilde 25*. Vi gjennomførte en brukertest på denne løsningen der vi fikk mange gode tilbakemeldinger (ref punkt 3.8).



Bilde 24: Prototypen med nye farger, registreringen og logg inn har fått sterk bakgrunnsfarge, men hovedsidene når bruker har logget inn er hvite.



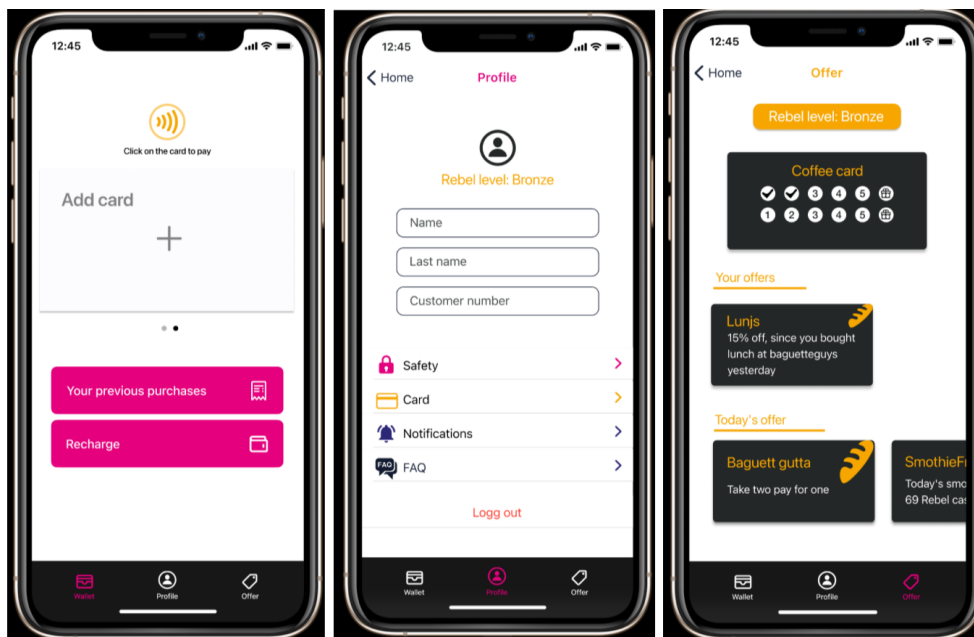
Bilde 25: bildet til venstre viser hvordan brukeren kan huke av på kvitteringer for å sende til arbeidsgiver eller andre. Bilde i midten er en oppdatert versjon av "offer"-siden. Bilde til høyre er når brukeren overfører penger til Rebel Cash.

[Link til høynivå prototype Versjon 2](#)

3.6.2.3. Versjon 3

Som vist på *bilde 26* gir ikonet i tab baren en indikasjon på hvilken side brukeren befinner seg på.

I brukertesten fikk vi tilbakemelding på at det er vanlig at første siden brukeren kommer til ligger til venstre i Tab baren. Vi har derfor valgt å flytte den dit fremfor i midten.



Bilde 26: Viser endringene, rosa markering i tab baren for å gi tilbakemelding på hvor brukeren befinner seg i løsningen. "Wallet" er også flyttet til helt til venstre i tab baren.

[Link til høynivå prototype Versjon 3](#)

3.7. Brukertest av lavnivå prototype - Analyse

Vi har valgt å fremstille funnene i en tabelloversikt. Hvert funn er rangert med en alvorlighetsgrad som går fra Lav til Kritisk. På denne måten vil vi enklere kunne se hvilke funn vi bør prioritere først for å forbedre løsningen.

1. **Kritisk** - Løsningens funksjonalitet mangler eller fungerer ikke som tiltenkt.

2. **Høy** - Brukeren er hindret fra å utføre oppgaven eller at løsningen ikke hindret brukeren i å gjøre feil.
3. **Middels** - Begrensede problemer som ikke hindret brukeren i å utføre oppgaven, men som skaper frustrasjon eller forvirring.
4. **Lav** - Ikke kritiske problemer, som kan være avvik fra standarder eller visuelle normer. (Nordbø 2017, 172)

3.7.1. Tabelloversikt av funn

Tabellen (*tabell 2*) tar for seg observert problem/tilbakemeldinger, anbefalt forbedring og problemets alvorlighetsgrad.

Obsvert problem/ tilbakemeldinger	Anbefalt forbedring	Alvorlighetsgrad
Forvirret om registreringen ble fullført	Gi tilbakemelding til bruker om at registreringen er fullført	Medium
Bruker forvirret om hvordan man lager en firesifret kode	Legge inn placeholder + forklarende tekst om hva den fire sifrede koden gjør	Medium
Usikker på hvor mye 1 Rebel Cash er i norske kroner	Vise tidlig/tydelig i appen hvor mye en rebel cash er i forhold til nok. Viser en oversikt over de ulike valutaene sammenlignet med Rebel cash.	Høy
Oversiktlig, hvor mye rebel cash har jeg?	Vise saldo før og etter overføring	Kritisk
Usikker på hva Rebel cash er	før registrering, kort info om hva rebel cash er	Medium

Tabell 2: Tabelloversikt over funn.

Se vedlegg J for fullstendig tabell med funn.

3.7.2. Konklusjon

Til tross for at brukertestene ble gjennomført over Google Meets fikk vi verdifulle tilbakemeldinger. Nøkkelen til gode brukertester er forarbeid. Vi hadde laget en god testplan som gjorde at vi var godt forberedt og førte til vellykket gjennomføring. Organiseringen av funnene og dets alvorlighetsgrad i en tabell ga oss god oversikt over hva som må jobbes videre med. Se vedlegg J for en mer detaljert analyse.

3.8. Brukertest av høynivå prototype - Analyse

Vi fikk god oversikt ved å fremstille funnene fra første brukertest i en tabell, vi valgte derfor å gjøre det samme denne runden se *tabell 3*.

3.8.1. Tabelloversikt over funn

Observert problem/ tilbakemeldinger	Anbefalt forbedring	Alvorlighetsgrad
Kunne tenkt meg en onboarding skjerm som forklarer applikasjonen før bruk.	Lage en onboarding som man kan velge å se på, to alternativer "next tip" og "Skip"	Medium
Tab baren gir ingen indikator på hvilken skjerm jeg er på.	Vise hvor i applikasjonen du står, farge på ikonet i tab bar på hvor du står. Flytte "Wallet" til venstre i og med at det er første siden.	Medium
Finner ikke ut av hvor man finner info om hva en rebel cash er	Kan ligge i onboardingen, eventuelt legge det i profil ettersom at flere av testdeltakerne tenkte det kunne ligge her. Legge inn en FAQ	Medium
Kvitteringer må ha en gyldig valuta	Ha en synlig konvertering eller bytte ut RC med KR	Høy
Forvirrende å skulle laste ned kvitteringer, hvor havner de?	Lage en standard iphone last ned, ha muligheten til å velge mellom å sende kvitteringer direkte eller laste de ned i filer. Rett til Expensify?	Medium

Tabell 3: Tabelloversikt over funn fra brukertestene

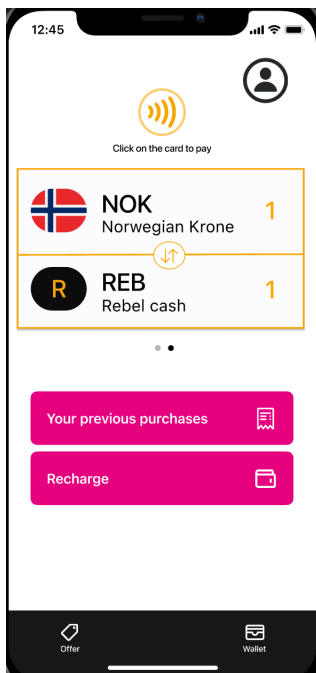
Se vedlegg K for fullstendig tabell med funn.



3.8.2. Funn og anbefalinger

Digitale kvitteringer var en gjentakende tilbakemelding. Flere var lei av å passe på kvitteringer over lengre perioder og det ville vært enklere å ha det digitalt. Det forteller oss at tilgang på kvitteringer digitalt er en motivasjonsfaktor for brukerne for å anvende applikasjonen. Flere av testdeltakerne forteller at de bruker Expensify på jobb, for å få igjen for utlegg, derfor ville det vært smart å kunne sende kvitteringer fra Rebel Cash direkte til Expensify. Vi fikk også bekreftet at muligheten for å kunne legge inn flere kort i applikasjonene er viktig.

Det var kun en av testdeltakerne som klarte å finne frem til valutakalkulatoren som vist på *bilde 27*. Det er mulig å se hvor mye 1 Rebel Cash er verdt på "Recharge"-siden, som flere av testdeltakerne fant til slutt. Derfor har vi bestemt oss for at når du sveiper på kortet vil du ha muligheten til å legge til flere kort. Valutakalkulatoren blir fjernet fra "wallet", men vil bli en del av en "FAQ" på profilsiden. Det er også ønskelig å legge inn en onboarding med denne informasjonen.



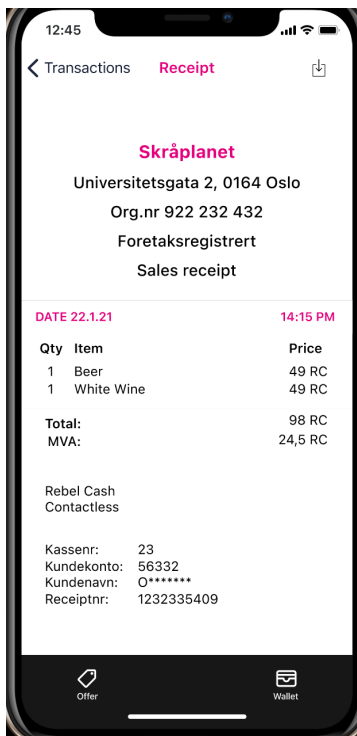
Bilde 27: Bilde fra prototypen, valutakalkulatoren

Underveis i brukertesten var det flere testdeltakere som ikke visste hvilken side de var på i tab baren. Derfor flyttes "Wallet" til venstre i tab baren, da dette er første siden brukerne kommer

Kandidatnummer: 8029, 8124, 8061, 8001, 8176

til når de logger inn. Vi vil også legge til en farge på ikonet i tab baren for å gi en indikasjon på hvor bruker befinner seg i applikasjonen.

Etter at testdeltakerne hadde gjennomført en betaling fikk de i oppgave å se på hva de hadde kjøpt (*bilde 28*). Flere testdeltakere fortalte at det må være konvertert tilbake til norske kroner for at det skal være en gyldig kvittering. Vi vil derfor endre fra Rebel Cash til norske kroner på kvitteringene.



Bilde 28: bilde fra prototypen, kvittering

Da testdeltakerne fikk i oppgave å laste ned en kvittering lurte flere på hvor filen havnet. Alle testdeltakerne nevnte at de ønsket å sende kvitteringen rett til Expensify. Her vil vi gi muligheten til å bestemme hvilken applikasjon brukeren vil anvende for å sende kvitteringen.



3.8.3. Konklusjon

Analysen av brukertesten ga oss flere tilbakemeldinger på hvordan vi kan forbedre løsningen. Flere av funnene har fått Høy og Kritisk alvorlighetsgrad, som forteller oss at det er forbedringspotensiale i applikasjonen. En mer detaljert analyse ligger i Vedlegg K.

3.9. Koding av terminal og Klient

3.9.1. Terminal

Terminalen har i oppgave å ta i mot en betaling og godkjenne/avslå. Det er tiltenkt at terminalen skal plasseres alene eller med andre betalingsterminaler. Av betalingsteknologi stod det mellom QR-kode og NFC. Det ble konkludert med at NFC gir en bedre flyt i prosessen enn QR. Siden løsningen kun er for demonstrasjon valgte vi å bruke ren Kotlin.

Vi valgte å implementere NFC teknologien først i løsningen. Da dette fungerte, utformet vi databasen. I løpet av prosjektet ble vi opptatte av å kunne ha tekst i applikasjonen. Teksten skulle forandre seg når klientapplikasjon brukes opp mot terminal.

For å bruke data fra databasen måtte den først modelleres. Dette for at vi skulle kunne bruke data mer effektivt utover koden. Vi lager dataklasser for hver butikk, bruker og kvittering.

3.9.2. Klient

I Klient applikasjonen satte vi opp MVVM sammen med SwiftUI som nevnt i kapittel 2.2.11. SwiftUI bruker State og object binding for å holde på en tilstand og oppdaterer View'et/brukergrensesnittet. Dette er samme tankegang som MVVM har (ref *bilde 6* punkt 2.2.11). Dette resulterte i at vi bygget opp mappestrukturen i Views, ViewModels og Models. RebelCash Applikasjonen ligger utenfor disse mappene, da det er startpunktet til applikasjonen. Denne er en erstatte til UIKit sin AppDelegate fil. I Views mappen har vi lagt alle filer som viser noe til brukeren og som brukeren interagerer med.



Bruken SwiftUI lar oss se endringene i koden i sanntid og er tidsbesparende for oss når vi utviklinger klienten.

3.9.3. NFC problematikk

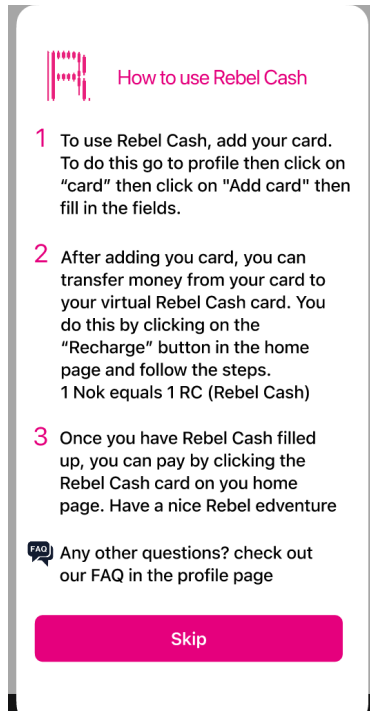
Når vi skulle implementere NFC i klientapplikasjonen måtte vi lese oss opp på hvordan dette gjøres på apple. Vi kan ikke finne noen dokumentasjon på HCE på apple sine utviklersider (Apple, n.d.). HCE (Host Card Emulation), etter mye research kan vi se at dette ikke ligger åpent i NFC API til Apple kun for de med spesiell tilgang. For å løse dette må bruke lese funksjonalitet på iOS for å lese terminal som bruker HCE. Dette kan la seg gjøre men må implementeres seinere utover prosjektet.

4. Løsning

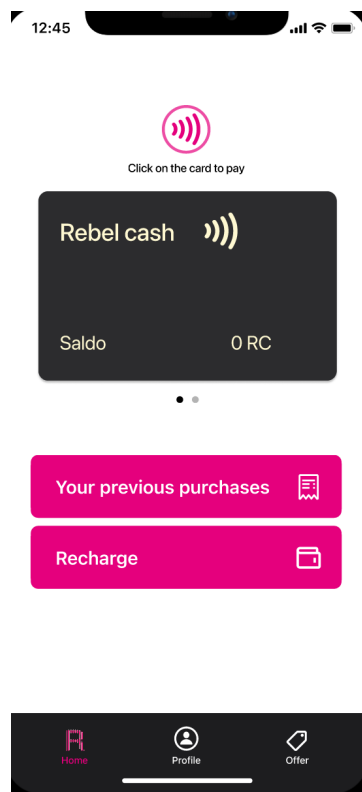
I dette kapittelet presenterer vi løsningen med fokus på utforming og funksjonalitet. I tillegg vil vi presentere arkitekturen for løsningen vi har laget for klient og terminal applikasjonen, med støttende forklaring til løsningen. I og med at prosjektet er et proof of concept vil prototypen i Figma samt kodet løsning være en del av løsningen.

4.1. Prototype

I en utviklingsprosess forekommer det endringer underveis på designet. Sent i prosessen fant vi ut av vi ønsket å gjøre endringer i prototypen. Det første vi la inn var en onboarding, den vil komme opp første gangen bruker registrer seg. Her står det enkelt forklart hvordan brukerne tar i bruk applikasjonen ved hjelp av enkle steg som vist på *bilde 29*.



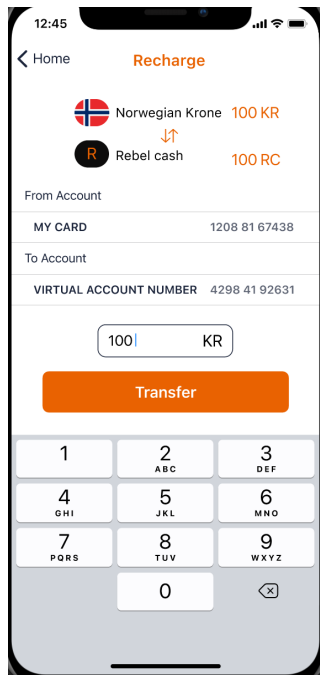
Bilde 29: Onboarding skjerm, ved hjelp av enkle steg tar den brukeren gjennom funksjoner.



På bilde 30 er teksten på det sorte kortet gjort om til en lys gul farge (#FEF8D1). Kortet har også fått fargen gråsort (#2C2C2E) da det er en standardfarge til iOS. "Home"-siden er helt enkel med få elementer og lite støy i bakgrunnen. Rosa er fargen som går igjen og har en god kontrast til den hvite bakgrunnsfargen. På denne siden kan bruker gå til tilbud, profil, tidligere transaksjoner, gjennomføre en betaling og gå til siden hvor man kan overføre penger.

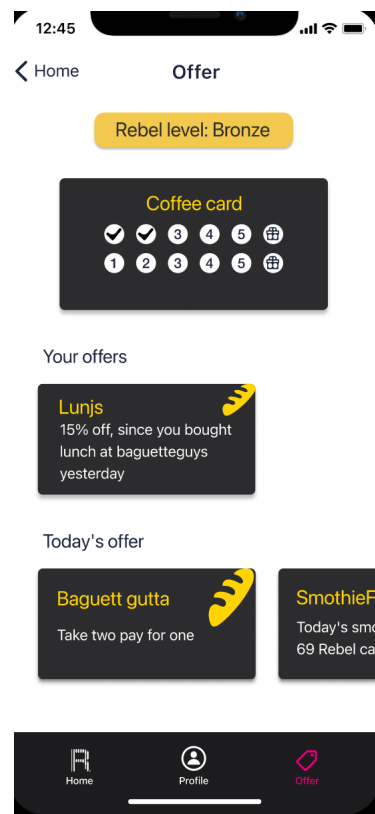
Tab baren befinner seg nederst på stort sett alle sidene, vi valgte å endre ikonet til rebel sin R logo samt "Home" som vist på bilde 30. Tab baren har fått fargen lys sort (#1C1C1E) som er standardfarge for iOS. Tab baren vil bruke rosa fargen på ikonet og teksten for å indikere hvor brukeren befinner seg i applikasjonen.

Bilde 30: "Home"-siden har et enkelt design med få elementer.



“Recharge”-siden som vist på *bilde 31* har en sterk oransje farge (#E96201), den har høy kontrast mot mørke og lyse farger. Denne fargen er den mest fremtredende fargen på sidene hvor bruker kan overføre penger fra norske kroner til Rebel Cash som vist på *bilde 31*. Her kan bruker overføre penger til sitt virtuelle kort.

Bilde 31: “Recharge” siden, med en sterk oransje farge.



Skriften på tilbuskortene er endret til gul fargen (#FFD400) til Rebel, da den passer bedre mot den mørke bakgrunnen på kortene. Her vil bruker kunne stemple kaffekortet sitt, lese mer om Rebel Level, se dagen tilbud samt personlige tilbud slik som vist til venstre på *bilde 32*.

Bilde 32: “Offer”-siden, her kan bruker benytte seg av tilbudene.



Bilde 33 viser skjermen der bruker kan legge til et nytt kort. Siden er gjort om litt, for å forkorte ned prosessen. Når bruker trykker på “+ Add Card” vil et overlay som vist på *bilde 34* komme opp, hvor brukeren kan fylle inn i feltene og kortet vil legges inn som vist på *bilde 33*.

< Profile **Card**

My card

**** * 5868

10/30

+ Add Card

Bilde 33: bruker har lagt til et kort.

Add card

Name your card *

Payment card number *

Valid date *

CVC *

Add card

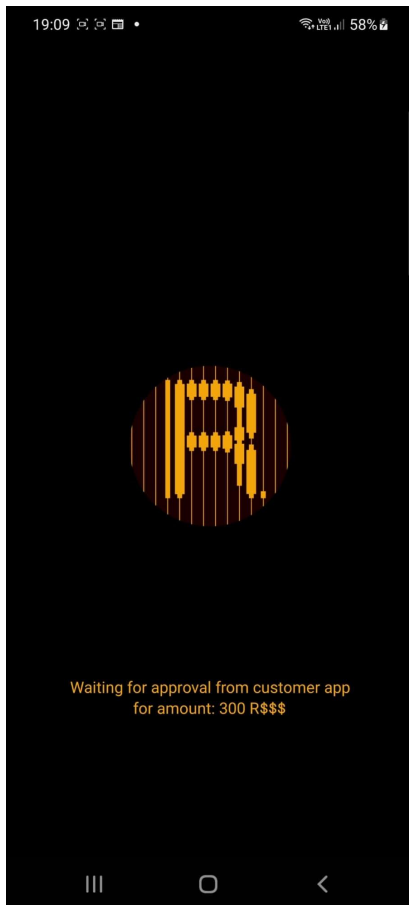
Bilde 34: Overlay som kommer opp når bruker trykker på “+ Add card”

[Link til siste prototype Rebel Cash prototype](#)

4.2. Utforming og funksjonalitet

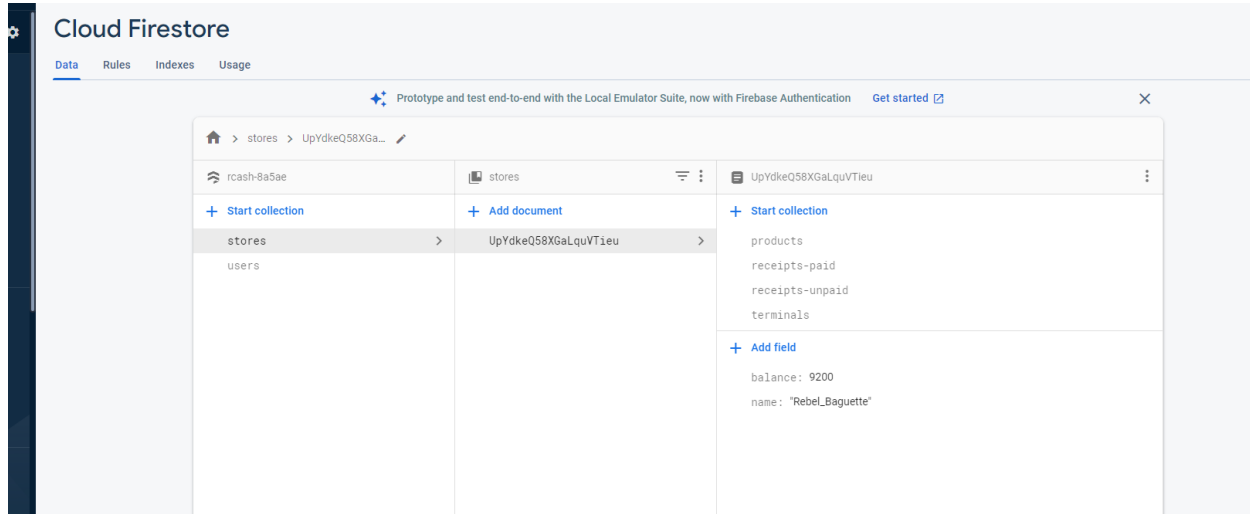
4.2.1. Terminal applikasjon

Terminal applikasjon (*bilde 35*) er utviklet i Kotlin. Applikasjonens hovedteknologi bruker NFC for å kunne lese NFC informasjon fra klientapplikasjonen som sender sitt kundenummer til terminal og setter eierskap til kjøpet. Etter klientapplikasjonen har bekreftet kjøpet vil en variabel i databasen som terminal lytter på bli satt som “true”. Transaksjonen blir da gjennomført av terminalen hvor kunde og butikk utveksler Rebel Cash for varer.



Bilde 35: Bilde av terminalen

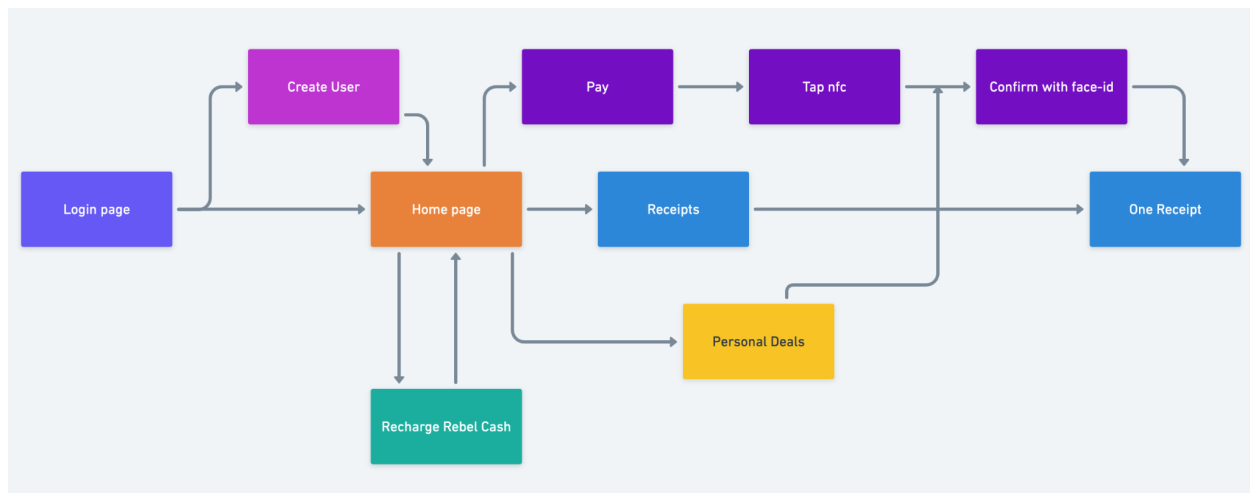
Når transaksjonen er utført, vil kvitteringen som ligger i temp-receipts i databasen (*bilde 36*) flyttes til receipt hos users og receipt-paid hos Stores med riktig ID på hvilken butikk eller kunde det gjelder. Hver kunde og butikk har sin unike ID i databasen det samme gjelder terminalene og kvitteringene.



Bilde 26: Firebase Firestore database

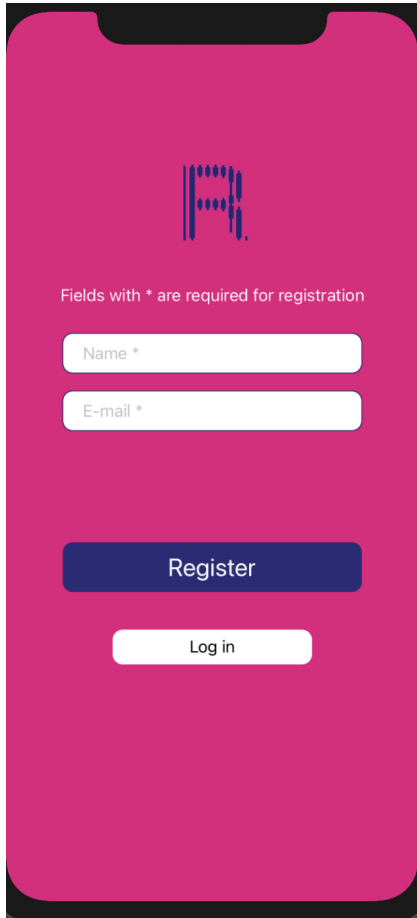
4.2.2. Klientapplikasjon

Som vist på *bildet 37* har klientapplikasjonen hovedsidene “Login”, “Register”, “Home”, “PersonalDeals”, “Receipts” og “Recharge”.



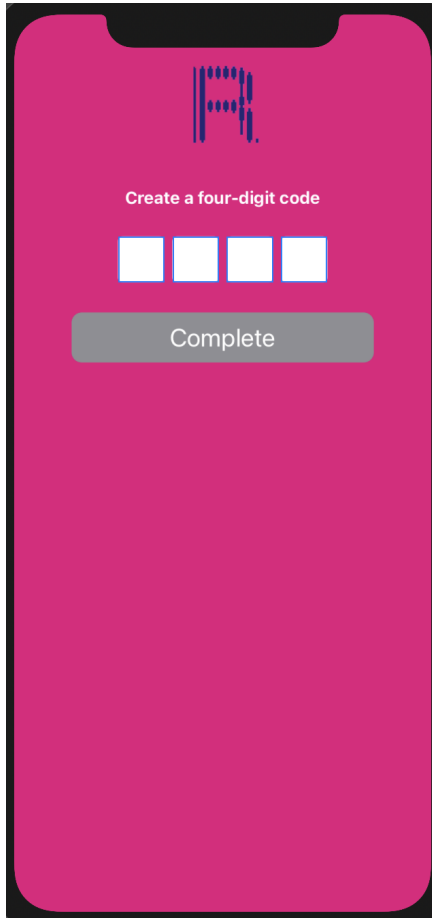
Bilde 37: Oversiktsflyt, en overordnet flyt oversikt over de viktigste sidene i applikasjonen.

Løsningen har en logg inn side hvor brukeren kan navigere seg til “Home”-siden eller til “Register”-siden. Det er ingen logikk i innloggingen da dette er et proof of concept.



Bilde 38: "Register"-siden

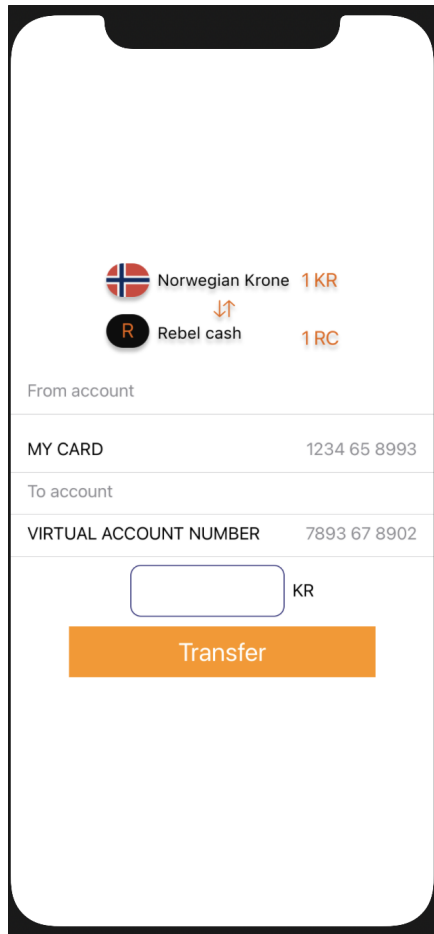
På "Register"-siden som vist på *bilde 38* kan brukeren registrere seg med navn og epost eller gå tilbake til logg inn. Da bruker trykker på "Registrer" blir bruker navigert til en ny side hvor man blir bedt om å lage en firesifret kode som vist på *bilde 39*. Når den firesifrede koden er opprettet blir knappen for å navigere videre aktiv og bruker blir navigert til "home"-siden.



Bilde 39: Bruker kan lage en firesifret kode

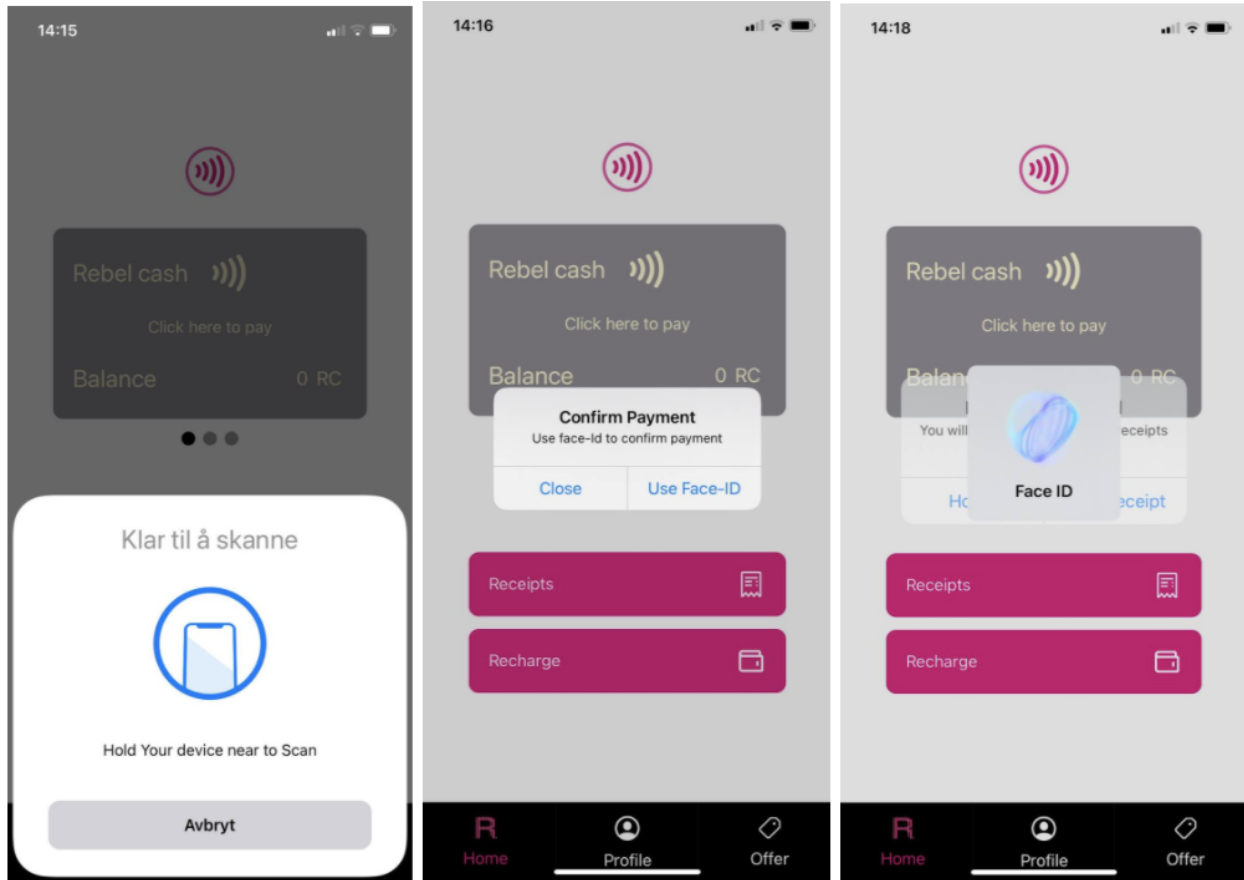
Applikasjonen har en navigasjonslinje øverst og en tab bar nederst på hovedsidene, som navigasjonen enklere. Fra “Home”-siden kan bruker navigere seg til “Reciept”-, “Profile”-, “PersonalDeals”- og “Recharge”-siden. Ved å trykke på Rebel Cash kortet kan brukeren betale med NFC/tæpping. Saldo på kortet viser hvor mye Rebel Cash kunden har tilgjengelig.

Fra “Recharge”-siden kan bruker overføre penger til Rebel Cash som vist på *bilde 40*. Bruker kan se hvor mye norske kroner er i Rebel Cash så fort beløpet tastes inn. Saldoen på kortet blir oppdatert etter overføring. Det er ikke implementert noen kontoløsning med ekte penger.



Bilde 40: Her kan bruker overføre penger til Rebel Cash

Når bruker trykker på Rebel Cash kortet for å betale kommer det opp et sheet/modal som ber bruker om å scanne/tæppe telefonen mot terminalen. Når terminal (NFC tag) er scannet kommer det opp en vaslingsboks hvor bruker enten kan lukke varslingen eller godkjenne med Face ID. Her er det den innebygde Face ID'en i telefonen som aktiveres, som vist på *bilde 41*. Ved godkjent Face ID får bruker mulighet til å velge å gå til enten "Receipt"-siden eller "Home"-siden.



Bilde 41: viser reisen fra brukeren trykker på kortet til betalingen blir gjennomført med Face ID

Fra “Receipts” får brukeren oversikt over alle transaksjonene. Her kan man trykke på transaksjonen og få opp kvitteringen, her kan bruker dele kvitteringer til andre applikasjoner, eller sende direkte til personer.

“Personal deals” er en oversikt over tilbudene man får fra bedriftene i Rebel-bygget, i vår løsning har vi kun statiske elementer.

4.3. Arkitektur

Vi vil nå redegjøre arkitekturen bak løsningen vår og hvordan løsningen mellom klient, terminal og database samhandler slik at en betaling gjennomføres.



4.3.1. Kommunikasjon terminal og klient

1. Terminal står ventende og venter på at en midlertidig kvittering skal lages i databasen.
2. Kunde tar eierskap av kvittering i databasen ved å skanne eller lese med NFC på sin mobil mot terminal.
3. Kunden får så en forespørsel på sin klientapplikasjon som kunde da blir nødt til å bekrefte via Face ID.
4. Når kunden har bekreftet kjøpet, forandres en variabel på kvitteringen hetende "paid" til en boolsk verdi "true".
5. Terminalen har under hele tiden lyttet til forandring i denne verdien. Når denne er "true" sletter terminal applikasjonen kvitteringen fra "temp-receipts" i databasen å oppretter en kvittering under "receipts- paid" både hos kunde, butikk og i databasen.
6. Deretter tar terminalen av seg transaksjonen av midler ved å fjerne Rebel Cash fra kunde og legge til butikkens likviditet .

Se vedlegg L for Visualisering av kommunikasjon terminal og klient.

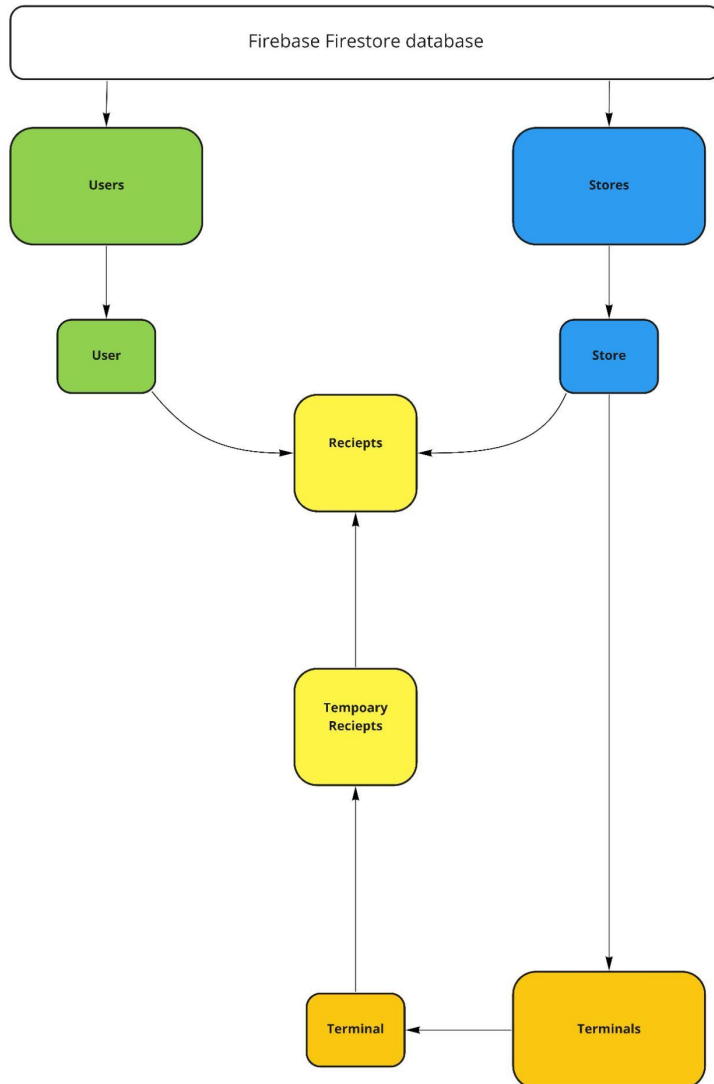
4.3.2. Database

Under denne seksjonen forklarer vi databasen sin struktur og endelig format.

Databasen er en Google Firebase Firestore database som er satt opp på en slik måte at man har "collections" samlinger og "docs" dokumenter. Slik det fungerer er at hver samling har hver sine dokumenter med sine egne egenskaper som igjen kan utvides til samlinger innenfor dokumentene.



Strukturen bygger på at man har to hoved samlinger som er “stores” og “users” som vist på *bilde 42*. Disse samlingene har igjen flere dokumenter med en unik id knyttet til de forskjellige brukerne og butikkene innenfor samlingene. For hver bruker og butikk er det lagt til flere verdi felter og egenskaper som definerer kunden eller butikken med tanke på midler, kvitteringer og varer.



Bilde 42: Visualisering av database

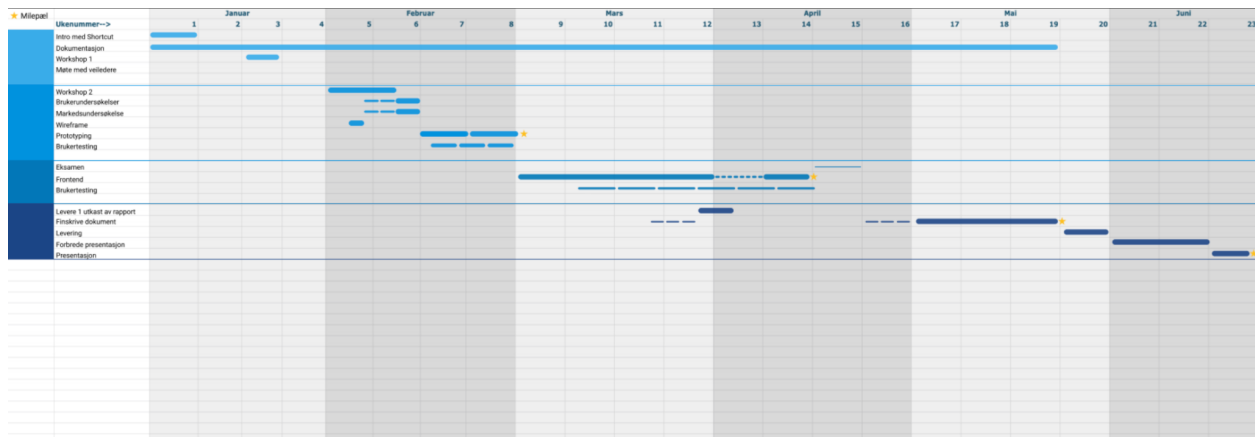
5. Vurdering

I dette kapittelet tar vi for oss prosessen, løsningen og utformingen av prosjektet. Vi går i detalj på viktige teknologier, teknikker og metoder som vi har brukt for å utvikle og strukturere oppgaven på best mulig måte. Vi fikk beskjed om at vi skulle lage "morgendagens betalingsløsning" og det var opp til oss å finne ut hva det er og hvordan vi skulle lage det.

5.1. Vurdering av prosess

5.1.1. Planlegging og metode

Gantt diagrammet (*Bilde 43*) vi laget i planleggingsfasen ga oss en god oversikt over hva vi skulle gjøre i de forskjellige fasene. Det ga oss en visuell fremstilling av arbeidsmengden samt tidsbruk, og vi fikk en ide om når hver del måtte avsluttes. I og med at vi la inn "slack" i planen ga det oss det rommet vi trengte til å gjøre endringer underveis samt håndtere uforutsette ting som for eksempel sykdom.



Bilde 43: Gantt diagram

Etter 5 måneder har vi gjort opp kun gode erfaringer med Scrumban. En av erfaringene vi har fått er at korte og effektive sprinter gir god oversikt i et stort prosjekt som det vi har hatt. Dette gjorde at vi kunne ha små leveranser, som hjalp oss å se hva som var ferdig og hva som trengte mer arbeid. Vi brukte sprintene og kanban-brettet tett opp mot hverandre. I slutten av hver sprint planla vi neste ukes sprint ved å tømme sprint backlogen og legge inn de oppgavene vi

skulle gjøre uken etter. Disse ble hentet fra den overordnede backloggen. Dette hjalp oss til å avgrense arbeidsoppgavene per uke og resulterte i en god og effektiv arbeidsflyt samt bedre resultater. Det at vi delte opp kanban-brettet i grupper (ref punkt 2.1.1.3) gjorde at hele gruppen visste hva alle jobbet med, som hadde en positiv innvirkning på kommunikasjonen i gruppa.

En annen ting som hadde positiv innvirkning på kommunikasjonen var Daily Standup. I begynnelsen av prosjektet hadde vi to standuper per dag, men så raskt at dette var unødvendig.

Siste dagen i hver sprint hadde vi retrospective møte. Her fikk gruppen reflektert over hva som hadde gått bra og hva som kunne gått bedre, pluss at vi fikk skrevet ned de viktigste tingene vi hadde oppnådd. Dette hjalp oss med å bli bedre underveis, samt å holde moralen og arbeidsinnsatsen oppe. På slutten av dette møtet gikk vi gjennom risikoplanen (ref vedlegg M), hvor vi så på viktige elementer og regulerte risikoen av disse opp eller ned. Et konkret eksempel på endring i risiko var under punktet “sykdom hos Decider”. Da dette inntraff så vi i tiltaksplanen (ref vedlegg N) at en av de andre gruppemedlemmene skulle overta denne rollen. Ved å følge tiltaksplanen klarte vi å fortsette arbeidet som normalt. For mer info rundt risikooppdateringer se vedlegg O

5.1.2. Workshop

Som nevnt i kapittel 3.4.1 brukte vi Empatikart i Miro for å konkretisere nøkkelfunksjonene. Dette hjalp oss med å være kreative og komme på mange gode ideer samt at vi fant ut hva som kan motiverer brukerne til å bruke applikasjonen. Et av utbyttene vi fikk fra workshopen var at den opprinnelige “Register”-siden hadde alt for mange felt å fylle inn. Etter vi hadde fikset dette punktet og brukertestet det, fikk vi kun positive tilbakemeldinger. Dette ga oss en god indikasjon på at workshopen kom oss godt til nytte.

Videre i prosessen brukte vi “Low hanging fruit” workshopen (ref punkt 3.4.2) som hjalp oss med å tydeliggjøre hva som var gunstig å ha med i betalingsapplikasjonen og viktigheten av



funksjonaliteten, samt tiden det ville tatt å utvikle. Resultatet av workshopene førte til at vi sparte mye tid, fremfor om vi skulle ha diskutert frem og tilbake i løse luften. Dette resulterte i at vi fikk fremdrift i prosjektet etter en litt stagnerende periode.

5.1.3. Designsprint, prototyping og brukertesting

Vi har hatt to designsprinter dedikert til klientapplikasjonen og en sprint på terminalen (ref punkt 3.4). Ved at alle gruppelemmene fikk skissert opp ulike oppsett klarte vi å lage en god flyt, med få elementer, gjennom løsningen. Som nevnt i kapittel 3.5.1.1 stemte vi på de beste forslagene som hjalp oss med å komme frem til de mest brukervennlige løsningene. I designsprint nummer to kom det flere ønsker fra gruppen om en tab bar, da dette gjør det enkelt å trykke seg gjennom ulike sider og raskt kunne å gå til "Home"-siden hvor brukerne kan betale. Dette påvirket flyten positivt og løsningen ble enklere å bruke. Designsprintene har vært viktige for oss i startfasen for å danne grunnmuren for designet av løsningen. Det har derfor vært viktig å brukerteste etter hver iterasjon.

Prosesen fra enkle trådkisser til ferdig høynivå prototype i Figma var basert på mange valg og endringer tatt på bakgrunn av tilbakemeldinger fra brukertester, designprinsipper og WCAG sine retningslinjer. Vi startet med å skissere et oppsett for den nødvendige funksjonaliteten slik som beskrevet i punkt 3.5.1.1. De enkle skissene var et godt utgangspunkt for testing, ettersom vi ønsket tilbakemelding fra brukerne tidlig i prosessen. Underveis i prosessen har vi gjennom brukertesting fått tilbakemeldinger som har ført til endringer i prototypen. Dette har vært viktig for å kunne skreddersy løsningen på best mulig måte for å vekke interesse hos bruker. Et eksempel på dette var å implementere muligheten til å kunne dele kvitteringer fra Rebel Cash til ønsket applikasjon eller direkte til en person. Videre har vi tatt et valg om å endre enkelte av fargene i designmalen til Rebel, dette var nødvendig for at løsningen skulle være universelt utformet. Vi hadde noen utfordringer med bakgrunnsfargene mot de sterke fargene, da det fort så utdatert ut les mer om dette under punkt 3.3.1.1. Vi konkluderte med å holde bakgrunnen hvit slik at løsningen ser moderne ut og står i stil til det moderne Rebel-bygget. Vi tok en avgjørelse og endret noen av hovedfargene i designmalen slik at de møter WCAG sine



Kandidatnummer: 8029, 8124, 8061, 8001, 8176

retningslinjene og opprettholder god kontrast. Nettsiden “contrast-finder” (Tanguru, n.d.) hjalp oss med å finne fargen med bedre kontrast, som var mest mulig lik den originale fargen til Rebel. Disse utfordringene var ikke store, men avgjørende for løsningsens tilgjengelighet. Uten disse endringene ville applikasjonen ikke sett helt ut som den gjør i dag, og de ville ikke fylt kravene beskrevet i WCAG sine retningslinjer.

Gjennom prosjektet har vi hatt fokus på å involvere brukerne. Vi brukertestet derfor på ansatte i Shortcut som i fremtiden har kontorer i Rebel-bygget. Dette har gitt oss veldig god og ikke minst ærlig innsikt fordi de selv vil være reelle brukere av applikasjonen. Brukertestene ble gjennomført på en ryddig og god måte; vi hadde en god testplan og gikk igjennom brukertesten før vi testet på testdeltakerne. Dette ga oss muligheten til å rette opp feil før brukertestene, les mer om brukertestene under punkt 3.7 og 3.8. Vi fikk gode data til videre analysering ved at tre personer fra gruppa var tilstede. To av personene tok notater av hva testdeltakerne sa, hvor de trykket og atferden deres.

Mesteparten av innsikten gruppen samlet inn var gjennom brukertestene. Personasene vi laget var basert på de vi fikk tilsendt fra kunden, da de allerede hadde dette på plass. Vi gjorde noen forandringer tilpasset vårt prosjekt. På bakgrunn av dette, ser vi i etterkant at vi gjerne skulle sendt ut en brukerundersøkelse, for å kartlegge brukerne utenfor IT-bransjen. Selv om store deler av kundegruppen vil være fra IT-bransjen, vil blant annet andre som jobber i bygget som for eksempel ansatte i de ulike restaurantene/kafeene og andre avdelinger som HR være en del av kundegruppen.

5.1.4. SwiftUI og Versjonskontroll

Under utviklingen av klienten benyttet vi oss av SwiftUI. Dette ga oss muligheten til å enkelt lage et godt brukergrensesnitt, håndtere “state” samt “databinding”. Dette resulterte i at vi fikk kodet opp de relevante sidene på kortere tid.



Som versjonskontroll verktøy brukte vi Git med GitHub som skybasert lagring. Videre brukte vi Git Workflow (ref *Bilde 3* i punkt 2.2.6) for å ha felles retningslinjer på hvordan vi skulle bruke Git. For mer info om branching strategien vår, se vedlegg P. Dette hjalp oss med å kunne jobbe sammen på samme kodebase og å sikre koden fra å bli ødelagt eller slettet uten hensikt. Dette resulterte i at alle til enhver tid hadde tilgang til en oppdatert versjon av koden.

5.2. Vurdering av Løsningen

Som nevnt i kapittel 4 er siste prototype i Figma og kodet løsning en del av den endelige løsningen vår. Vi valgte å gjøre det på denne måten for å dekke hvert spekter av løsningen.

Ved å fokusere på designprinsippet "synlighet", som vil si høy kontrast på fargene, hjelper dette mennesker med sensoriske nedsettelse, motoriske nedsettelse er også en viktig del av universell utforming. Vi løste dette ved å legge til enkel navigering gjennom en tab bar som gir brukeren navigeringsmuligheter uten å måtte bytte grep på mobilen samt med få klikk. Vi har også tatt hensyn til de med kognitive nedsettelse, ved å lage enkle sider med få elementer slik at brukerne har lett for å lære og huske hvordan applikasjonen fungerer. Vi har hatt fokus på designprinsippene "hint" som gir applikasjonen et selvforklarende design og "tilbakemelding" som gir informasjon om hva brukerne har gjort og hva dette fører til. Et eksempel på bruk av designprinsippet hint i applikasjonen er på "home"-siden hvor det er tre prikker under kortet, dette gir brukeren et hint om at det befinner seg noe mer til høyre. Designprinsippet "tilbakemelding" kan vi se når brukeren har overført penger til Rebel Cash og det kommer opp en melding hvor brukeren får tilbakemelding om vellykket overføring. Det er viktig for oss at applikasjonen er lett å ta i bruk for alle. Ved å designe for de som har plager med ledd eller er svaksynt, vil vi gjøre applikasjonen mer brukervennlig for alle, som beskrevet i punkt 3.3.3. På bakgrunn av endringene resulterte dette i at vi fikk tatt hensyn til brukerne av applikasjonen, uansett funksjonsnedsettelse eller ikke.

Som beskrevet i punkt 4.2.2 kan klientapplikasjonen lese NFC brikken i terminalen. Dette er det vi kommer til å fokusere på her da betaling med NFC er den viktigste implementasjonen i vår



Kandidatnummer: 8029, 8124, 8061, 8001, 8176

løsning. Det første suksess kriteriet var å sørge for lesing av NFC tag'en mellom terminal og klientapplikasjonene fungerte. Neste suksess kriterie var å sjekke om det fungerte stabilt, altså at den leser hver gang klienten kommer nærmere enn 4 centimeter fra terminalen. Vi testet dette med 20 korte tester og resultatet var at det fungerer hver gang, men at telefonen må innenfor 3 centimeter for å lese. Vi vurderer dette for å være godt nok i en betalings situasjon, også med tanke på sikkerheten til brukeren. For mer om NFC funksjonalitet og sikkerhet se vedlegg Q.

Det at vi har klart å benytte NFC i betalingen vil gagne Entra ved at de muligens kan bruke NFC i andre deler av bygget. De vil også kunne benytte applikasjonen i flere av sine bygg.

Terminalen ble utviklet til Android grunnet bedre støtte for NFC muligheten. Kotlin er et godt utviklet språk som er enkelt og kan kombineres med java som også kan brukes som backend i løsningen. Ulempen ved bruken av kotlin uten rammeverk, er at vi manuelt måtte sette opp layouts og i dette tilfelle tok det lengre tid. Alt i alt er vi veldig fornøyd med hvordan løsningen ble og vi fikk implementert NFC til både klienten og terminalen som vi så på som vår hovedutfordring i selve utviklingen.

5.3. Veien videre

Dagens løsning begrenser seg til funksjonaliteten beskrevet i kapittel 4.2. Infrastrukturen til det nåværende produktet er ikke optimalisert for lansering.

Neste steg i prosessen med Rebel Cash ser vi for oss at blir å implementere resten av funksjonaliteten fra prototypen. Det vil være ting som registrering og logg inn funksjonalitet, lojalitets funksjonaliteten og kvitteringene. Videre så vil et naturlig steg være å legge til App Clips funksjonalitet se vedlegg G (brukerreisen ende til ende). Det vil også være naturlig å videreutvikle applikasjonen og funksjonene til det Rebel trenger. Her vil implementering av Rebel Cash inn i hovedapplikasjonen til Rebel være et godt alternativ, blant annet for å se på muligheten for å bruke NFC til å åpne dører inne i bygget. I tillegg ser vi muligheten til å



implementere BLE beacons (Bluetooth low energy beacon), slik at brukeren for eksempel kan bestille mat og drikke direkte til sin lokasjon med relativ høy presisjon basert på antall beacons.

Det vil også være ønskelig å gjennomføre brukertester underveis i utviklingen, samt på ferdig løsning for å avdekke feil, og å være sikker på at man lager det brukeren og kunden trenger.

6. Konklusjon

I bachelorperioden har vi laget et Proof of Concept av en betalingsløsning i form av to applikasjoner for Rebel gjennom Shortcut. Målet med applikasjonene har vært å lage “morgendagens betalingsløsning”. Punkter som ble vektlagt har vært at de skal være lettvinne, enkle, nyttige og brukervennlige. Gjennom samtaler med Entra og Shortcut konkluderte vi med at løsningen skulle være en betalingsløsning med både en klient og en terminal. Prosjektet startet med noen workshops, korte design-sprinter, skissering i Figma, brukertesting av klikkbar Figma-prototyp, koding av terminal og klient, brukertesting av sistnevnte og til slutt ferdigstilling av applikasjonene basert på observasjonene fra brukertestene.

Basert på observasjoner, brukertester og samtaler med mentor og interessenter i Shortcut konkluderte vi med at applikasjonen skulle inneholde en form for intern valuta, gamification og trådløs betaling. Med funksjonaliteten implementert vil det tiltrekke brukere, gjøre det lettvinnt å bruke applikasjonen og gagne Entra i form av kuponger som fører til gjenkjøp. Under utviklingen av løsningen har det vært et sterkt fokus på at Rebel Cash ikke skal tape mot andre betalingsapplikasjoner (ref Google Pay og Apple Pay). Brukertestene ga oss gode tilbakemeldinger og kritikk som førte til at vi endret på utseende og plassering av knapper til å tilfredsstille kritikken.

Vi hadde hyppige møter med Entra og Shortcut som sørget for at vi hele tiden var på riktig vei og at prosjektet vårt føltes viktig. Etter mange måneder med dokumentering, skissering,

brukertesting og utvikling fikk vi bekreftet Entra's tilfredshet da de viste stor takknemlighet og interesse i resultatet vårt.

7. Referanser

- Alqudah, Mashal, and Rozilawati Razali. 2017. "A comparison of scrum and Kanban for identifying their selection factors."
https://www.researchgate.net/publication/323817820_A_comparison_of_scrum_and_Kanban_for_identifying_their_selection_factors.
- Alqudah, Mashal, and Rozilawati Razali. 2018. "An Empirical Study of Scrumban Formation based on the Selection of Scrum and Kanban Practices."
https://www.researchgate.net/publication/329506016_An_Empirical_Study_of_Scrumban_Formation_based_on_the_Selection_of_Scrum_and_Kanban_Practices.
- Apple. n.d. "NFC - IOS." Core NFC. Accessed 03 26, 2021.
<https://developer.apple.com/documentation/corenfc>.
- "Apple Developer." n.d. <https://developer.apple.com/xcode/swiftui/>.
- Cadle, James, and Donald Yeates. 2008. *Project Management for Information Systems*. fifth edition ed. N.p.: Pearson.
- Caldwell, Ben, Michael Cooper, Loretta G. Reid, and Gregg Vanderheiden. 2008. "Web Content Accessibility Guidelines (WCAG) 2.0." W3.org. <https://www.w3.org/TR/WCAG20/>.
- "Cloud Firestore." n.d. <https://firebase.google.com/docs/firestore>.
- Cooper, Alan. 2004. *Inmates Are Running the Asylum, The Why High Tech Products Drive Us Crazy and How to Restore the Sanity*. 2nd ed. N.p.: Pearson Education (US).
- Daly, Laura. n.d. "Kanplan: where your backlog meets kanban." Atlassian.com. Accessed 03 19, 2021. <https://www.atlassian.com/agile/kanban/kanplan>.



Kandidatnummer: 8029, 8124, 8061, 8001, 8176

Devmountain. n.d. "Git vs. GitHub: What's the Difference?" devmountain.
<https://blog.devmountain.com/git-vs-github-whats-the-difference/>.

Driessen, Vincent. 2010. "A successful Git branching model." nvie.
<https://nvie.com/posts/a-successful-git-branching-model/>.

"Gitflow WorkFlow." n.d.
<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>.

Google. 2019. "Near field communication overview." developer.android.com.
<https://developer.android.com/guide/topics/connectivity/nfc>.

Johnson, Jeff. 2014. *Designing with the Mind in Mind*. 2nd ed. N.p.: Morgan Kaufmann Publishers In.

Kevcodez. n.d. "Agile Development: Scrumban." kevcodes.
<https://kevcodes.de/inside-agile-product-development-series/05-agile-development-scrumban/>.

Kouraklis, John. 2016. "MVVM as Design Pattern."
https://www.researchgate.net/publication/310493153_MVVM_as_Design_Pattern.

Krogstad, Julie R. n.d. "IKKE AKKURATT PÅ SKINNER."
<https://core.ac.uk/download/pdf/30900485.pdf>.

Nordbø, Tone. 2017. *Introduksjon til interaksjonsdesign*. Oslo: Universitetsforlaget.

Preece, Jenny, Yvonne Rogers, and Helen Sharp. 2015. *Interaction design: beyond human-computer interaction*. Chichester: John Wiley & Sons, Inc.

"Professional Guides: Workflow Strategies." 2017.
<https://www.youtube.com/watch?v=aJnFGMclhU8>.

Rebel. n.d. ""De beste IT-hodene."" Rebel. <https://rebel.no/>.

Rehkopf, Max. n.d. "Sprints." atlassian.com. Accessed 03 19, 2021.
<https://www.atlassian.com/agile/scrum/sprints>.



Kandidatnummer: 8029, 8124, 8061, 8001, 8176

Rehkopf, Max. n.d. "What is a kanban board?" atlassian.com. Accessed 03 19, 2021.

<https://www.atlassian.com/agile/kanban/boards>.

Rubin, Kenneth S. 2013. *Essential Scrum*. N.p.: Addison-Wesley.

Sandnes, Frode E. 2018. *Universell utforming av IKT-systemer*. N.p.: Universitetsforlaget.

"Statcounter." n.d. <https://gs.statcounter.com/os-market-share/mobile/norway>.

Tanguru. n.d. "Contrast finder." contrast-finder.tanguru. <https://contrast-finder.tanaguru.com>.

Toftøy-Andersen, Eli, and Jon G. Wold. 2011. *Praktisk brukertesting*. N.p.: Cappelen Damm akademisk.

8. Vedlegg

A = Begrepsliste

B = Gruppekonsert

C = Tabell III, Scrum Practices Formation

D = Android Studio og Xcode

E = Gantt

F = Personas

G = Brukerreise

H = Designmal

I = Designsprint

J = Bruksskvalitetrapport - Lavnivå prototype

K = Bruksskvalitetrapport - Høynivå prototype

L = Visualisering av kommunikasjon terminal og klient.

M = Risikoplan

N = Tiltaksplan

BAO301

Kandidatnummer: 8029, 8124, 8061, 8001, 8176



O = Risikooppdateringer

P = Branching Strategi - Fra Smidig prosjektet til Kjetil og Jonas vår 2020

Q = NFC funksjonalitet og sikkerhet