

Ludvik Schanke Blunck (8174)  
Daniel Wendt Kjellid (8153)

Emnekode: BAO301

Emnenavn: Bachelorprosjekt

Oppdragsgiver: Zeipt

Innleveringsdato: 20.05.2021

Antall sider: 83

Antall ord: 11995

Tilgjengelighet: Fri  Begrenset

Høyskolen Kristiania

Web basert kvitteringsmal-bygger for Zeipt - En brukersentrert prototype  
Web based receipt template builder for Zeipt - A user-centered prototype



Våren 2021

Denne bacheloroppgaven er gjennomført som en del av utdannelsen ved Høyskolen Kristiania. Høyskolen er ikke ansvarlig for oppgavens metoder, resultater, konklusjoner eller anbefalinger.

## Innholdsfortegnelse

<b>Sammendrag .....</b>	<b>4</b>
<b>1. Innledning.....</b>	<b>5</b>
1.1. Presentasjon av oppdragsgiver.....	5
1.2. Presentasjon av prosjektet.....	5
1.3. Presentasjon av gruppen.....	6
1.4. Gruppens målsettinger .....	6
1.5. Problemstilling .....	7
1.6. Teoretisk forankring .....	7
1.7. Forskningsetikk.....	9
<b>2. Prosess og metode .....</b>	<b>11</b>
2.1. Prosess og prosjektplan .....	11
2.2. Valg av utviklingsmetode .....	11
2.3. Vår bruk av Scrum.....	11
2.3.1. Sprinter .....	12
2.3.2. Daily Scrum .....	13
2.3.3. Product review .....	13
2.4. Verktøy .....	14
<b>3. Analyse og utforming .....</b>	<b>16</b>
3.1. Målgruppe.....	16
3.2. Designprosessen.....	16
3.4. Ekstern brukertesting av prototype .....	20
3.4.1. Deltakere.....	21
3.4.2. Utforming av brukertest .....	22
3.4.3. Funn i brukertest.....	24
3.4.4. Konklusjon.....	26
<b>4. Teknisk løsning.....</b>	<b>30</b>
4.1. Utvikling.....	30
4.1.1. Utvikling av GUI.....	30
4.1.2. Utvikling av oversetter.....	34
4.1.3. Dokumentasjon av kode .....	39
4.1.4. Utdfordringer .....	41
4.2. Forslag til videre utvikling .....	41
<b>5. Vurdering av løsning og prosjekt .....</b>	<b>43</b>
5.1. Teknisk resultat .....	43
5.2. Nytte for oppdragsgiver .....	43
5.3. Vurdering av metode, prosjektgjennomføring og verktøy .....	44
5.4. Vurdering av resultat i forhold til teori .....	44
<b>6. Konklusjon .....</b>	<b>46</b>

<b>Referanseliste .....</b>	<b>47</b>
<b>Vedlegg .....</b>	<b>50</b>
<b>Vedlegg A: Begrepsliste .....</b>	<b>50</b>
<b>Vedlegg B: Gruppekontrakt .....</b>	<b>51</b>
<b>Vedlegg C: Prosjektplan .....</b>	<b>54</b>
<b>Vedlegg D: Risikoanalyse .....</b>	<b>56</b>
<b>Vedlegg E: Utviklet prototype .....</b>	<b>59</b>
<b>Vedlegg F: Spørsmål og oppgaver i brukertest .....</b>	<b>65</b>
<b>Vedlegg G: Resultat og analyse fra ekstern brukertest .....</b>	<b>69</b>
Resultat og analyse .....	69
Individuelle svar .....	74
<b>Vedlegg H: Utdrag fra Git.....</b>	<b>79</b>
<b>Vedlegg I: Oppdragsgivers vurdering .....</b>	<b>82</b>

## **Sammendrag**

Denne rapporten omhandler hvordan vi har utviklet en web basert kvitteringsmal-bygger hos Zeipt - en løsning som Zeipt kan bruke til å senke onboardingstiden for fremtidige kunder og løser et problem ved at kunder slipper å oversette deres kvitterings-format lokalt før de sender det til endepunktene, og kan istedenfor benytte seg av et GUI som veileder dem gjennom prosessen. Målet med resultatet er at det skal være av svært god kvalitet og som både er tilfredsstillende og gir en nytteverdi for oppdragsgiver. Prosjektet har bestått av en omfattende innsiktsfase gjennom en iterativ designprosess og ekstern brukertesting. Etter prosjektets første fase har vi utviklet løsningen i programmeringsspråket JavaScript. Ved å hele tiden sette sluttbrukeren i sentrum har vi kommet frem til vår løsning.

## 1. Innledning

### 1.1. Presentasjon av oppdragsgiver

Vår oppdragsgiver Zeipt er en startup som fokuserer på å digitalisere kvitteringer. Målet deres er å automatisk levere kvitteringer digitalt, rett etter kjøpet er gjort. På denne måten sikter de mot å eliminere behovet for printing av papirkvitteringer. Foretaket ble stiftet i 27.01.2016 som et Aksjeselskap og er i dag en alumni til TheFactory Accelerator. De har mottatt softfunding fra Innovasjon Norge, har en Seal of Excellence fra EU og er et løpende SkatteFUNN-prosjekt. Zeipt er registrert i bransjen Programmeringstjenester og har base i Oslo sentrum. Selskapets sist registrerte driftsinntekter fra 2019 var 494 000 kr (Proff.no 2021). Teamet deres består av 11 ansatte, hvor Sebastian Rundqvist (veileder/produkteier) er Co-founder & CTO sammen med David Alexandre Salvail som er fungerende daglig leder.

### 1.2. Presentasjon av prosjektet

Zeipt ønsket å utvide dashboardet deres med en web basert kvitteringsmal-bygger, som skal anvende seg av en stegvis tilnærming for å grafisk hjelpe brukeren med å gjøre det selv. Utviklingen av denne byggeren vil kraftig senke onboardingstiden for Zeipt sine fremtidige kunder da de slipper å oversette deres kvitterings-format lokalt før de sender det til endepunktene, og kan istedenfor benytte seg av et GUI (Graphical User Interface) som veileder dem gjennom prosessen. Som Sebastian Rundqvist (produkteier) sier, "Løsningen vil effektivt redusere tidsbruk og kostnader for integrasjon i og med at kundene slipper å gjøre større inngrep i sine egne kodebaser og kan istedenfor fokusere på selve leveransen av kvitteringen på Zeipt sine endepunkter".

GUIet det er snakk om er Zeipt sitt dashboard. Dashboardet er kundenes administrasjonsområde hvor data, analyser, veiledninger, fakturaer og innstillinger samles på ett sted. Dashboardet skal videreutvikles med et nytt navigeringsalternativ i menyen, altså kvitteringsmal-byggeren. Kvitteringsmal-byggeren skal kunne skape en mal som lagres i databasen slik at Zeipts backend vet hvordan brukerens kvitteringsformat skal oversettes online ved behov. Dersom vi løser denne oppgaven på en god måte vil Zeipt bruke våre løsninger i deres dashboard.

Løsningen som utvikles skal bygge videre på Zeipt sin kodebase. Prosessen vil inkludere utvikling av nye dashboard sider. Designet baseres på Zeipt sitt nåværende design ved bruk av bestemte farger og komponenter, men vi har fått frie tøyler til å være kreative i designprosessen.

Ettersom oppgaven som skal løses er ganske teknisk vil vi bruke en del begreper sentralt rundt programvareutvikling gjennom rapporten. En begrepsliste kan finnes i Vedlegg A.

### 1.3. Presentasjon av gruppen

Vår bachelorgruppe består av Daniel Wendt Kjellid og Ludvik Schanke Blunck. Begge går Interaktivt design på Høyskolen Kristianas institutt for teknologi. Gjennom studieløpet har vi opparbeidet oss en bred kunnskap i utvikling, prototyping og brukertesting. På grunn av gruppens størrelse har vi ikke hatt stort fokus på fordeling av ansvar. Vi har jobbet om hverandre og samarbeidet både med utvikling og rapportskrivning. Dette fordi vi hele tiden kan hjelpe og lære av hverandre, slik at det endelige resultatet blir av best mulig kvalitet.

Gjennom bachelorstudiet har vi alltid valgt å jobbe sammen på gruppe når det har vært mulig, som har ført til en god dynamikk oss imellom. Vi har også utarbeidet en gruppekontrakt gjengitt i Vedlegg B.

Tabellen under viser alle personer involvert i prosjektet:

Navn	Stilling	Tilknytning	Rolle i prosjektet
Sebastian Rundqvist	Co-founder & CTO	Zeipt	Veileder/produkteier
Bjørn Reitzer	Head of frontend	Zeipt	Mentor frontend utvikling
Hanne Sørum	Førsteamanuensis/Associate Professor	Høyskolen Kristiania	Intern veileder
Daniel Wendt Kjellid	Student	Høyskolen Kristiania	Utvikling og rapportskrivning
Ludvik Schanke Blunck	Student	Høyskolen Kristiania	Utvikling og rapportskrivning

Tabell 1: Oversikt over personer involvert i prosjektet

### 1.4. Gruppens målsettinger

Målet med prosjektet er å levere et resultat med svært god kvalitet og som er tilfredsstillende for oppdragsgiver. Den ferdige løsningen skal ha en nytteverdi for Zeipt, og skal samtidig ha gitt gruppens medlemmer størst mulig læringsutbytte.

For å sikre at vi oppnår at oppdragsgiver og sluttbrukere er tilfreds med løsningen er det viktig at vi har en god kommunikasjon med nøkkelpersoner. Det er også et ønske om å få en bedre forståelse for hvilke prosesser og krav arbeidslivet setter i praksis, samt å lære relevant teknologi og teknikker.

### 1.5. Problemstilling

Oppgaven fra Zeipt kan beskrives som todelt, oversetting og kartlegging. Oversettelsesdelen er den delen av oppgaven som skal gjøre et ikke foretrukket filformat om til et foretrukket filformat, og kartleggingen er den delen som kobler nøkler og verdier fra det foretrukne filformatet til Zeipt sitt kvitteringsformat. Ved å ha fokus på brukervennlighet kan man effektivisere både bygge- og onboardingsprosessen. Dette vil gi en nytteverdi for Zeipt og deres kunder.

Vår problemstilling blir derfor; *Hvordan utvikle en web basert kvitteringsmal-bygger som er brukervennlig for sluttbruker?*

For løse problemstillingen vil vi ta utgangspunkt i sluttbrukerens ønsker og behov for å bygge en høynivå prototype. Denne skal brukertestes og utvikles iterativt gjennom prosessen.

### 1.6. Teoretisk forankring

For å kunne utarbeide en prototype i henhold til de målene vi har satt oss trenger vi å innhente kunnskap innen en rekke felter, inkludert interaksjonsdesign, informasjonsarkitektur, programmering og brukertesting.

Vi vil her ta utgangspunkt i litteratur innenfor de nevnte fagfeltene, men hovedvekt på forskning fra Jakob Nielsen og Donald Norman, og da spesielt deres samarbeid på nettstedet nngroup.com (Nielsen Norman Group). Videre støtter vi oss på boken "Interaction Design: Beyond Human-Computer Interaction" (Preece, Rogers og Sharp 2019) og "Introduksjon til interaksjonsdesign" (Nordbø 2017). I prosjektets første faser hvor vi har stort fokus på design og en brukersentrert designprosess hviler vi oss på teori fra boken "The Elements of User Experience: User-Centered Design for the Web and Beyond" (Garret 2010).

Vi har et ønske om at brukerne som benytter seg av løsningen skal oppleve en så god brukeropplevelse som overhodet mulig. En brukeropplevelse kan defineres som hvordan en bruker opplever et produkt/tjeneste (Preece, Rogers og Sharp 2019). For å ivareta brukerens behov har vi tatt utgangspunkt i en brukersentrert designprosess. Brukersentrert design er en filosofi som baserer seg på brukerens behov og interesser, og har som mål å utvikle produkter og tjenester som er anvendbare og forståelige (Nordbø 2017). For å ivareta dette er brukeren hele veien involvert i prosessen, og før selve utviklingen av løsningen startet gjorde vi en omfattende ekstern brukertest for å kartlegge om brukeropplevelsen vi hadde designet for fungerte slik vi hadde tenkt. Resultat og funn i denne brukertesten kommer vi nærmere inn på i punkt 3.4.

I designet av løsningen var vi hatt brukervennlighet, på engelsk *usability*, i sentrum for å sikre at løsningen skal være nettopp anvendbar og forståelig. For å sikre at man skal oppnå dette er det hensiktsmessig å utforme spørsmål ut fra usability goals (Preece, Rogers og

Sharp 2019). Vi har utformet følgende spørsmål for løsningen vår, forankret i teori om de forskjellige målene for brukskvalitet (Preece, Rogers og Sharp 2019):

- 1. Anvendbarhet:** Har løsningen et sett med tilpassede funksjoner som lar brukerne gjøre oppgavene sine på den måten de ønsker? Løsningen har et stort fokus på å prøve og forstå format og sammenheng i forhold til det brukeren laster opp. Den vil derfor akseptere mye forskjellig, og deretter gjøre det om til et format som Zeipt forstår. På den måten gis brukeren stor frihet til å "gjøre som de vil", men resultatet vil fortsatt bli det samme.
- 2. Effektivitet:** Når kunden har lært hvordan de bruker systemet for å opprette maler, kan de opprettholde en høy grad av effektivitet? Løsningen er delt opp stegvis hvor vi forsøker å gi god informasjon om hva som skjer i de forskjellige stegene. Når dette er sagt er prosessen i seg selv relativt kort, og det burde derfor være greit å huske om man skal gjøre dette flere ganger.
- 3. Sikkerhet:** Hvor mange feil er det mulig å støte på underveis når man bruker løsningen, og hvilke tiltak er iverksatt for å la brukeren løse feilen? Man vil hele veien få feilmeldinger dersom noe er galt. Her vil feilmeldingen beskrives, og det skal foreslås for brukeren hvordan man løser det.
- 4. Nyttighet:** Er den utviklede løsningen i stand til å la kunder av Zeipt opprette egne maler for hvordan Zeipt skal håndtere hver individuell kundes kvitteringsfelt? Løsningen prøver å tilrettelegge for dette ved å forstå forskjellige formater av opplastede maler, enten det er ren tekst, JSON eller XML. Deretter veiledes kunde gjennom et GUI hvor de vil få opp feilmeldinger, og muligheten til å rette opp disse, dersom det skulle være noe.
- 5. Lett å lære:** Er det mulig for brukeren å finne ut hvordan man bruker systemet ved å utforske grensesnittet og prøve ut forskjellige handlinger? Før den endelige versjonen av løsningen var klar ble den brukertestet. Tilbakemeldingene fra testen var at man skjønte hvordan man skulle opprette en mal når man hadde gjort det en gang tidligere. Funn og resultater fra testen kan finnes i punkt 3.4.
- 6. Lett å huske:** Hva slags grensesnitt-støtte har blitt tilbudt for å hjelpe brukerne å huske hvordan man utfører handlinger, spesielt for operasjoner de ofte bruker? Tilsvarende funksjonalitet og betydning er utformet likt gjennom hele løsningen.

For å kunne designe og utvikle gode løsninger kan man ta utgangspunkt i designprinsipper. Designprinsipper er basert og utformet på bakgrunn av psykologi med tanke på hvordan mennesker oppfatter, lærer og husker. Det finnes flere som har utarbeidet ulike designprinsipper, men vi har valgt å hvile oss på prinsippene utformet av Donald Norman:

- 1. Synlighet (visibility):** Synlighet skal fungere som en påminnelse om hva som skal gjøres, og hvordan (2013). Eksempler på dette i løsningen er vi skiller vanlig tekst fra klikkbare lenker som utfører handlinger. Vi har også hatt et stort fokus på visuell struktur, samt at elementer skal berike funksjoner, og ikke forstyrre dem.



2. **Sammenheng (mapping):** Sammenheng handler om sammenhengen mellom forskjellige elementer, altså kontroller og det de kontrollerer (Nordbø 2017). I løsningsens mapping-steg vises input- og label felt i sammenheng, og i en logisk rekkefølge.
3. **Tilbakemelding (feedback):** Ved å gi brukeren tilbakemelding i forbindelse med handlingen som er blitt utført, og hva det har ført til, gjør at brukeren kan fortsette med å utføre aktiviteten (Norman 2013). I løsningen gir vi brukeren en tilbakemelding både når ting går som forventet, og når det skjer en feil. Når det skjer en feil får brukeren beskjed om hva som gikk galt, og hvordan man skal unngå at dette skjer igjen.
4. **Konsistent design (consistency):** Konsistent design handler om at lignende funksjonalitet skal fungere på lignende måte (Nordbø 2017). Ved å gjøre dette, kan løsningen oppleves lettere å lære og lettere å bruke (Preece, Rogers og Sharp 2019). I løsningen er alle handlings-funksjoner fargelagt i samme farge, og det skal være tydelig at disse fører til noe. Dette praktiseres jevnt over applikasjonen.
5. **Begrensninger (constraints):** Begrensninger handler om å begrense hva brukeren har mulighet til å gjøre, for å unngå at det skjer feil som kan ende opp med å hindre fremgangen (Preece, Rogers og Sharp 2019). Vi har valgt å gå for en stegvis tilnærming, og man har ikke mulighet til å fortsette til neste steg dersom det tidligere steget ikke er tilfredsstillende.
6. **Hint (affordance):** Hint, også kjent som selvforklarende design, handler om oppfattede eller faktiske egenskaper ved noe som gjør at løsningen oppfattes selvforklarende (Nordbø 2017). I alle stegene er det en forklarende tekst i forhold til hva som foregår, og hvordan man skal komme videre. Det varsles også om feil som gjøres, og hvordan disse kan rettes opp.

### 1.7. Forskningsetikk

Vi har i løpet av prosjektperioden utviklet og gjennomført en ekstern brukertest, som gjør at vi må ta stilling til det forskningsetiske aspektet rundt dette.

Alle som deltar i studien burde behandles både ærlig og rettferdig, og vi som forskere har i oppgave å være transparente rundt nettopp de etiske problemstillingene (Oates 2006).

Samtidig som vi må ta etiske hensyn gjennom studien fordi vi "burde", er det også flere lover og regler som regulerer nettopp dette. Nasjonalt har vi "Forskningsetikkloven", "Helseforskningsloven" og "Personopplysningsloven", og De nasjonale forskningsetiske komiteene som et forvaltningsorgan. Internasjonalt reguleres opplysninger om personvern for institusjoner og bedrifter hvor deltakeren er tilknyttet EU, i stor grad av General Data Protection Regulation (GDPR). GDPR er den strengeste loven i verden som omhandler personvern og sikkerhet. Ved brudd av GDPR kan det sanksjoneres enormt store bøter/straffer (European Union 2021).

Vi kommer ikke til å melde inn studien til NSD, da vi ikke skal behandle personopplysninger på den måte at dataene kan spores tilbake til enkeltpersoner.

Det er viktig for oss at deltakerne i studien forholdes anonyme, deltar av fri vilje, og at de får informasjon om resultatene i etterkant, før de blir publisert. Videre vil ikke studien fremme våre egne synspunkter eller interesser, men vil være objektiv slik at resultatene bærer preg av validitet og reliabilitet. Deltakerne gjennomfører også undersøkelsen uten oppsyn og overvåking. Det er derfor heller ikke lagt inn et samtykkeskjema.

## 2. Prosess og metode

I dette kapitlet forklares og utdypes det rundt valg av metode, samt redegjøres for valg og bruk av ulike verktøy og alternativer.

### 2.1. Prosess og prosjektplan

Prosjektplanen vår viser den overordnede planen for prosjektet som vil løpe i 20 uker. Prosessen i prosjektet er fordelt i aktivitetene oppstart, analyse og forarbeid, utvikling, akseptansetester, levering og rapportering.

I oppstartsfasen (fase 1) lå fokuset på å definere og konkretisere oppgaven, og beskrivelsen rundt hva som skal utføres og lages. Deretter gikk vi over til å analysere problemstillingen og finne ut hvilke verktøy og teknologier som kunne brukes til å løse problemstillingen (fase 2). I denne fasen utviklet vi prototyper og brukertestet de. Videre bevegde vi oss inn i utviklingsfasen (fase 3) som bestod av koding av prototypen. For å sikre at kvaliteten på løsningen var god og at oppdragsgivers forventninger ble oppfylt på en tilfredsstillende måte ble det gjennomført akseptansetester (fase 4). Siste fase, levering (fase 5), bestod av klargjøring av produkt og rapport. Rapporteringsfasen (fase 6) var aktiv under hele prosjektet, og gikk parallelt med de andre fasene. For en mer detaljert gjennomgang av fasene se Vedlegg C.

Aktivitet	Uker																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Oppstart	█																			
Analyse og forarbeid			█																	
Utvikling				█																
Akseptansetester																				█
Levering																				█
Rapportering	█																			

Illustrasjon 1: Prosjektplan tabell

### 2.2. Valg av utviklingsmetode

Under utviklingen av et informasjonssystem kan det være en fordel å velge en passende utviklingsmetode slik at prosjektet blir fullført til avtalt tid, bruker riktig mengde ressurser og innfrir kravene spesifisert på forhånd. I utgangspunktet kategoriserer vi utviklingsmetoder inn i to retninger, vannfallsmetoder og smidige metoder. Vannfallsmetoden adopterer en steg for steg tilnærming med forhåndsdefinerte kravspesifikasjoner og har gjerne en prosess med forhåndsbestemt rekkefølge. Smidige metoder fokuserer på en iterativ prosess hvor kravspesifikasjonene kan endre seg underveis i utviklingen. I slike metoder benytter man ofte prototyper for å involvere brukerne av systemet som bygges. Dette gjør at brukerne kan definere og bekrefte krav ved å demonstrere mulighetene som finnes. Slik vil man oppnå en mer brukersentrert prosess (Cadle og Yeates 2008).

Hvilke av disse metodene man benytter seg av kommer mye an på prosjektets omfang, og hva som skal utvikles. Vannfallsmetodikk legger til rette for en forhåndsbestemt plan slik at de involverte kan vite hva man skal jobbe med til enhver tid. Denne typen metoder gir ikke rom for endringer underveis. I smidige metoder derimot er det lettere å endre kravspesifikasjoner underveis i prosjektet, i og med at utviklingen foregår i iterasjoner og involverer brukergrupper kan man få mye verdifull innsikt i løpet av utviklingen, dermed kan man hindre store kostbare feil (Cadle og Yeates 2008). Nettopp av disse grunnene valgte vi å ta i bruk en smidig utviklingsmetode, slik at sluttproduktet kan oppnå god brukervennlighet gjennom inkludering av sluttbrukere i utviklingsprosessen.

### **2.3. Vår bruk av Scrum**

Generelt sett involverer smidige metoder forskjellige teknikker som gjør at kravspesifikasjonene og deretter systemet kan utvikles gjennom en rekke iterative aktiviteter som vanligvis involverer bruken av prototyper. Prototyper legger til rette for at systemet kan bli testet og vurdert av brukere, slik at justeringer og modifikasjoner kan bli gjort raskt og enkelt (Cadle og Yeates 2008).

En manifestasjon av den smidige tilnærmingen til systemutvikling er Scrum (Cadle og Yeates 2008). Scrum er det ledende rammeverket for system og programvareutvikling. Ifølge Nes (2019) er Scrum basert på empirisme, som vektlegger at kunnskap kommer fra erfaring og beslutninger som baseres på det som er kjent. Grunnet tidligere erfaring med bruk av Scrum, valgte vi å ta det i bruk, men med tanke på at vi bare var to på gruppen måtte vi adaptere en litt egen versjon. Vanligvis i Scrum deles gruppen inn i roller, hvor product backlogen er i stort fokus gjennom utviklingen. Product backlogen er en prioritert og estimert liste over funksjonalitet (Nes, 2019). Som nevnt var vi kun to på gruppen, og valgte derfor å ikke dele inn i roller, eller ta noe hensyn til en oversiktlig product backlog. Vi valgte Scrum for å ta i bruk de effektive teknikkene slik som sprinter, product review og daily scrum.

Før vi tok fatt i utviklingen hadde vi sprinter hvor alt fokus lå på å etablere et godt design og avdekke eventuelle forbedringer ved hjelp av en brukertest. Slik kunne vi være sikre på at designet var logisk å navigere og ta i bruk før vi satt i gang med selve kodingen. Å avdekke flest mulig feil i designet fra tidlig av gjorde at vi slapp å redesigne underveis i utviklingen.

#### **2.3.1. Sprinter**

En sprint kan bli sett på som en syklus som starter med et planleggingsmøte, hvor scrum-teamet fastsetter hva som skal leveres i løpet av sprinten (Nes, 2019). Våre sprinter ble planlagt gjennom videomøter med produkteier (Zeipt). Dette ga oss et overblikk over hvilke oppgaver vi skulle prioritere gjennom sprinten. Sprintene startet med en gang møtet ble avsluttet, og varte helt til neste møte. Som regel var det 7-10 dager mellom hvert møte, som da ble tidsrommet for sprintene. Vi tok ikke oppgavefordelingen under sprintene veldig høytidelig, og heller diskuterte hvilke oppgaver vi kunne tenke oss å utføre. Det førte til en god fordeling, hvor begge kunne velge oppgaver ut ifra tidligere kunnskap.

På slutten av en sprint ble det holdt et møte med Zeipt, som kan ligne en sprint review. På disse møtene som vi kalte "demo day" presenterte vi arbeidet som hadde blitt gjort under sprinten. Slik kunne Zeipt komme med tilbakemeldinger og innspill, som ofte førte til at nye oppgaver ble oppdaget, og det ble diskutert i fellesskap hva neste sprint burde inneholde.

For å sette et slags lokk på sprintene, slik at vi kunne ta fatt i neste sprint, hadde vi sprint retrospektiv. Dette var et møte mellom oss på gruppen, hvor vi tok et lite tilbakeblikk på sprinten som hadde vært, og diskuterte hva som hadde gått bra, og hva som kunne ha gått bedre. Disse kjappe møtene hjalp oss med å forbedre måten vi jobbet på i neste sprint.

Å dele opp utvikling av både design og kode i sprinter fungerte godt. Uten sprintene ville vi ikke hatt noen retningslinjer for hva som skal gjøres, og ville ført til dårlig arbeidsflyt. Sprinter hjelper oss å innkapsle arbeidsoppgaver over en bestemt tidsperiode, sånn at man ikke blir oversvømt med arbeid, men heller utvikler i iterasjoner med et overkommelig antall arbeidsoppgaver.

### **2.3.2. Daily Scrum**

For å kunne kartlegge hvordan vi lå an i løpet av en sprint, hadde vi jevnlig møter i gruppen. I følge Scrum.org (2021) er Daily Scrum et 15-minutters møte daglig mellom utviklerne for å overse fremgangen mot sprintens mål. Vi satt ingen tidsramme for våre Daily Scrum møter, og vi fikk heller ikke hatt slike møter daglig. Vi hadde en tilnærming til disse møtene som minner mer om en vanlig Daily Standup, hvor det er i følge geekbot.com (2020) 3 spørsmål som står i fokus:

1. Hva har du gjort siden sist?
2. Hva skal du gjøre fremover?
3. Er det noe som hindrer deg i å gjøre det?

Ved å svare på disse spørsmålene fikk vi en god oversikt over hva slags arbeid som gjensto for å nå sprintens mål. Som nevnt ble det ikke mulighet for disse møtene hver dag, noe vi skulle ønske vi hadde tid til. Uansett fikk vi veldig mye positivt ut av møtene når vi hadde mulighet til det. Daily Scrum fungerte bra for oss fordi vi fikk mulighet til å kartlegge hva som hadde blitt gjort, hva som skulle gjøres og hva som eventuelt kunne være utfordrende. På denne måten fikk vi en oversikt over hvordan vi skulle dele opp arbeidet ut ifra kompleksiteten av det. For at vi skulle få enda bedre utbytte av Daily Scrum måtte vi ha hatt det hver dag, slik kunne vi hatt full kontroll over hva slags arbeid som gjensto i sprintene. Daily Scrum ble holdt over verktøyet Discord (punkt 2.4.).

### **2.3.3. Product review**

Møtene med Zeipt etter avsluttet sprint fungerte som et product review, som i følge Scrumdictionary.com (2021) er et møte hvor produkteier og scrum teamet går over inkrementet i utviklingen hvor produkteier kan komme med tilbakemeldinger og tips om

veien videre, hva man kan gjøre bedre osv. Gjennom prosjektet har vi hatt en åpen dialog med Zeipt som alltid har kommet med innspill og pekepinner. Under disse videomøtene demonstrerte vi løsningen og eventuell kode, slik at Zeipt enkelt kunne komme med spørsmål og tilbakemeldinger.

## **2.4. Verktøy**

I dette punktet har vi valgt å presentere de ulike verktøyene vi har benyttet oss av i utviklingsprosessen, og hvordan de har gagnet oss.

### **Figma**

I starten av prosjektet, skulle vi lage skisser for løsningen, som ble gjort ved hjelp av Figma. Figma er en web basert grafikk-redigering og grensesnitt design app, som kan brukes til å gjøre all slags grafisk design arbeid. Figma skiller seg ut fra andre lignende verktøy ved at Figma kan brukes direkte i nettleseren, dette gjør det utrolig enkelt å tilgjengeliggjøre prosjekter. Figma har på denne måten gjort det enkelt for oss å inkludere produkteier i designprosessen ved å gi de tilgang til prosjektet.

Figma gjør det også mye enklere for oss som utviklere å samarbeide i designprosessen i og med at flere personer kan jobbe i samme prosjekt samtidig. Figma har også innebygd støtte for å lage fullt funksjonelle prototyper, som ble brukt i brukertesting.

### **Git og GitLab**

GitLab er en samarbeidsutviklingsplattform. GitLab er basert på Git versjonskontrollsystem. Dette gjør det mulig å gjennomføre en ordnet utvikling. Git i seg selv er programvare for sporing av endringer i et hvilket som helst sett av filer, og er vanligvis brukt for å koordinere arbeid blant utviklere.

Zeipt benytter seg av GitLab som plattform for deres kodebase, i begynnelsen av prosjektet fikk vi tilgang til de nødvendige filene på GitLab, slik at vi kunne komme i gang med programmeringen. Etter vi fikk tilgang brukte vi Git for å lage en ny "branch" i deres kodebase. En "branch" representerer en uavhengig versjon av prosjektet hvor våre endringer ikke påvirker Zeipt sin kode. Dette gir oss frie tøyler til å prøve og feile, uten at Zeipt må ta stilling til endringene vi gjør. Bare når Zeipt ser seg fornøyd med arbeidet vårt, kan de velge å merge (slå sammen) vår "branch" med deres master, som er den originale kodebasen. Git loggen i løpet av prosjektet kan finnes i Vedlegg H.

### **Google Docs**

For at vi kunne samarbeide med rapportskrivningen, benyttet vi oss av Google Docs. Google Docs legger til rette for å enkelt samarbeide om det samme dokumentet samtidig. Dokumentet er tilgjengelig i nettleseren for alle med tilgang, og alle endringer blir gjort i sanntid. Lagring er også utrolig enkelt med Google Docs, i og med at alt lagres automatisk.

**Discord**

Covid-19 pandemien gjorde det dessverre vanskelig for vår gruppe å møtes, men ved bruk av Discord har vi kunnet kommunisere jevnlig i utviklingsprosessen. Discord er en kommunikasjonsapp for samtaler, videosamtaler og meldinger brukt av over 100 millioner mennesker rundt om i verden (Discord.com 2021).

Underveis i prosjektet hjalp Discord oss med å løse ulike utfordringer vi hadde i plenum. Discord var også det vi brukte for våre daily standups, hvor vi reflekterte over arbeid som var gjort, eventuelle utfordringer og veien videre i utviklingen.

**E-post (Gmail)**

E-post er også verdt å nevne som verktøy, i og med at det var slik vi kommuniserte med Zeipt i utviklingsprosessen, utenom de ukentlige videomøtene. Gjennom e-post kunne vi stille spørsmål angående oppgaven, som fungerte godt for oss fordi Zeipt alltid var flinke til å gi oss tilbakemeldinger og tips over e-post.

**Visual Studio Code**

For å sikre god arbeidsflyt, er det viktig med en god IDE (Integrated Development Environment). Visual Studio Code er en lett men kraftig IDE med innebygd støtte for hundrevis av programmeringsspråk (Visual Studio Code 2021). På grunn av tidligere erfaring med verktøyet, ble valget enkelt. VSCode har et intuitivt og brukervennlig design, som hjalp oss å opprettholde gode kodestandarder i programmeringsdelen av prosjektet.

**Maze Design**

Maze Design ble brukt til brukertesting av prototypen, som er en plattform som lar oss gjennomføre brukertester online. Maze har innebygd støtte for prototyper laget i Figma, som gjorde det enkelt å ferdigstille brukertesten innen kort tid.

### 3. Analyse og utforming

#### 3.1. Målgruppe

For at vi skal kunne lage noe bra må vi vite hvem vi skal lage det til. Målgrupper hjelper oss med å tilspisse tjenesten ved å ha et overblikk over hvilke personer eller grupper man faktisk lager løsningen for. For å prøve å sortere brukere i målgruppen inn i mindre målgrupper kan vi i tillegg til behov, benytte oss av forskjellige kriterier som skiller brukerne fra hverandre. Disse kriteriene kan eksempelvis være alder, kjønn, utdanningsnivå, sivilstatus og inntekt. Disse kalles for demografiske kriterier. I tillegg til å sortere på demografiske kriterier, kan vi også sortere på psykografiske. Dette kan for eksempel være oppførsel, meninger, erfaring med teknologi og kunnskap om det gjeldende fagområdet (Nordbø 2017).

Videre er personas et sentralt begrep når man snakker om segmentering og målgrupper. En persona er en personalisering av en typisk person i målgruppen. Personaliseringene er basert på kundedata, og disse dataene brukes til å visualisere og konstruere en skikkelse med personlighet, preferanser, vaner, hobbyer osv. Målet med å opprette personas er å gjøre det lettere å forstå målgruppen, samt skjønne deres behov og atferd og sentrere utviklingen av produktet eller tjenesten deretter (Fagerstrøm, Eg, Johannessen og Vogt 2020). For at personas skal være av verdi må de være mest mulig spesifikke, og mest mulig realistiske (Nordbø 2017).

Når vi segmenterer inn i målgruppe tar vi utgangspunkt i målgruppen for oppgaven vi har løst, og ikke nødvendigvis Zeipt sin målgruppe som en helhet. Vi kan derfor definere målgruppen som menn og kvinner i alderen 21-67 år, og som er utdannet eller har erfaring innenfor teknologi og/eller systemutvikling. Videre jobber de i en bedrift som driver salg av varer eller tjenester til privatpersoner. Gruppen må være tilhengere av teknologi, og gjerne være interessert i miljøaspektet ved å kutte tradisjonelle kvitteringer. Målgruppen trenger faglig kunnskap til fagområdet.

En relevant persona i forhold til målgruppen, kan for eksempel være: **Jon, 29 år:** Jon er 29 år og jobber som systemutvikler i Bohus og har en mastergrad i anvendt datavitenskap ved NTNU. Han er gift med Marianne (31 år) og de har en datter sammen. Jon er lidenskapelig ovenfor teknologi og teknologiens fremskritt. Han er opptatt av at datteren hans skal ha de beste forutsetningene når hun vokser opp. Jon er klimabevisst, og gjør små ting i hverdagen for å bidra til en bedre klimasituasjon. Eksempler på dette er kildesortering, kjøp av kortreist mat, sykler til jobb, osv. Jon har nylig fått øye på Zeipt, og synes først og fremst at konseptet er kult og spennende, men liker også fokuset på bærekraft.

#### 3.2. Designprosessen

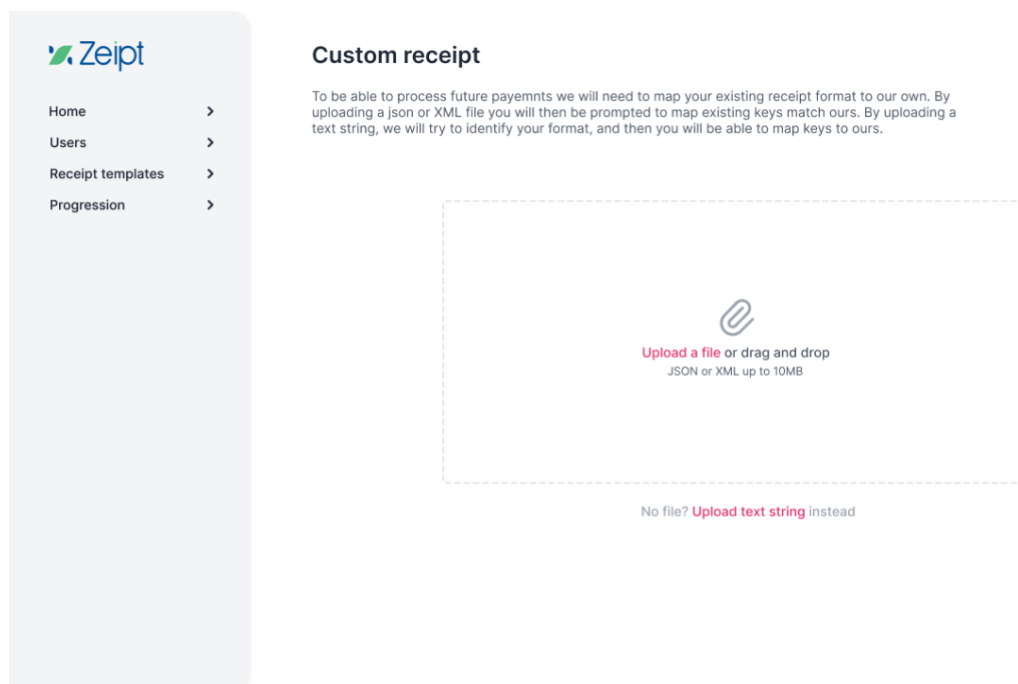
I designprosessen har vi hatt mye fokus på designprinsipper og usability goals (punkt 1.6), for å sikre god brukervennlighet. Designskissene ble konstruert i verktøyet Figma (punkt 2.4) under sprinter tidlig i prosjektets livssyklus. Vi har gått for en "design first" metodikk som



legger til rette for god bruk av designprinsipper før vi rettet fokuset mot koding, samtidig som det gir oss en mulighet til å kontinuerlig være på bølgelengde med oppdragsgiver. Begynner man å kode før designet er ferdigstilt, risikerer man å oppdage eventuelle forbedringer i designet for sent, som kan føre til at man må kode noe på nytt.

For at dashboardet til Zeipt skulle ha et sammenhengende design, tok vi utgangspunkt i deres fargepalett og typografi i designprosessen.

For å demonstrere hvordan vi tok hensyn til designprinsipper og usability goals i designprosessen vil vi herunder vise frem noen skisser fra Figma, og deretter knytte designet opp mot anvendt designteori.



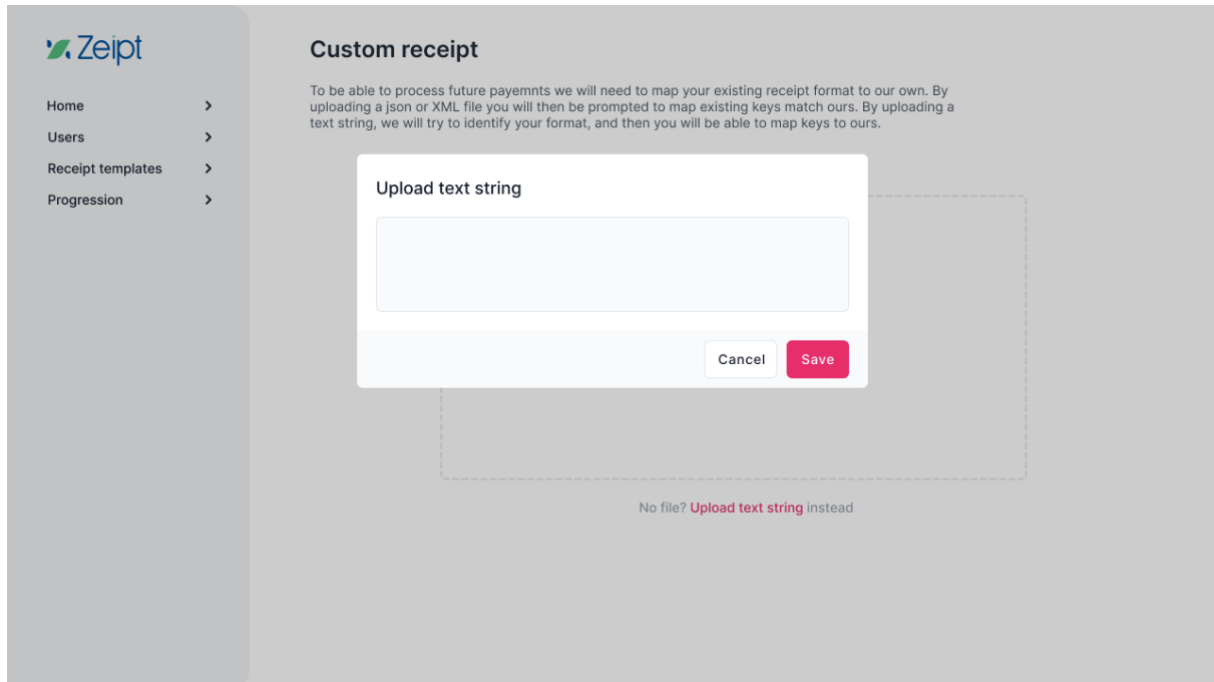
*Illustrasjon 2 - landingsside for løsningen*

For at nye brukere skal kunne komme raskt i gang og forstå hva løsningen faktisk gjør, la vi stor vekt på designprinsippet **hint**. I skissen ovenfor ser vi en klar og tydelig overskrift med en forklarende tekst under, som forteller brukeren hva han/hun kan gjøre på dette første steget.

**Synlighet** er også et prinsipp vi hadde fokus på i designet av landingssiden. Først og fremst ved bruk av et ikon og de stiplede linjene, som skal signalisere at det skal utføres filopplastning. Bruken av den rosa fargen (som er en del av fargepaletten til Zeipt) på tekst knappene gjør at de skiller seg ut fra resten av siden, og gjør det åpenbart for brukeren at disse har en "Call to Action" (CTA) effekt. CTA i webdesign er et begrep brukt om elementer som ber om en handling fra brukeren, populært i form av en knapp (Gube 2009). Gube (2009) poengterer også i sin artikkel at bruken av kontrasterende farger er viktig i bruken av

CTA-knapper. Slik sørger man for at brukeren skiller knappene fra omkringende elementer og legger godt merke til hvor det er meningen å klikke.

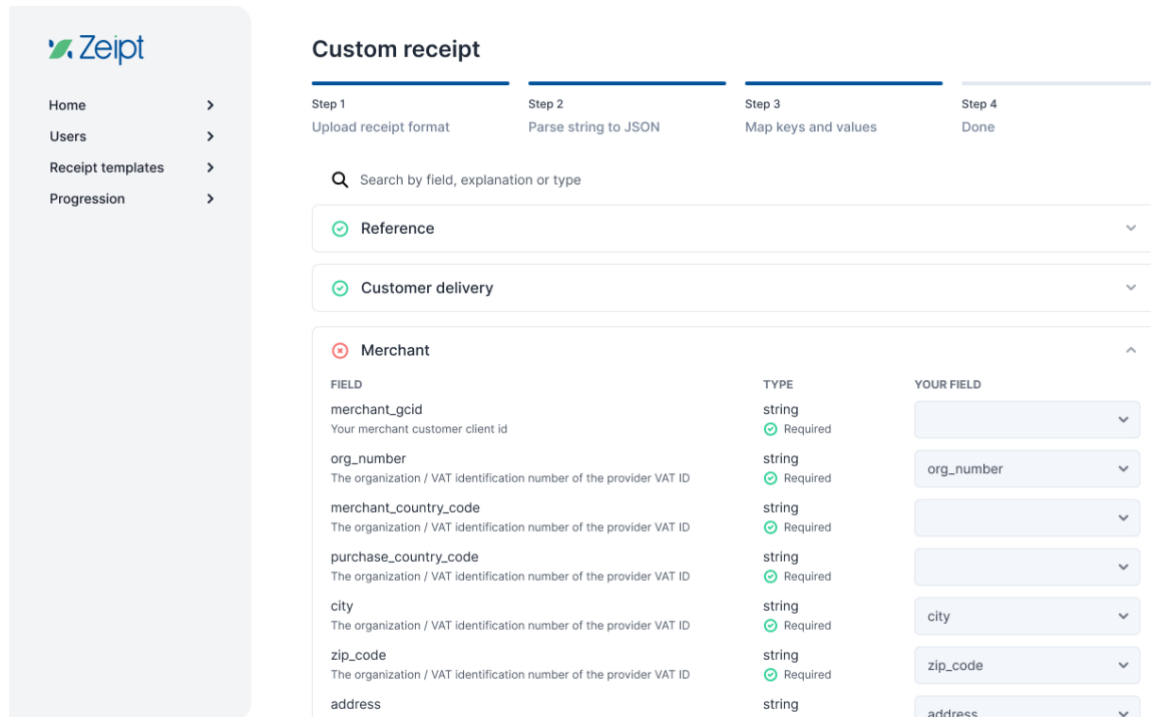
Den rosa fargen på knappene er gjennomgående i designet, slik at brukeren alltid gjenkjenner en CTA-knapp. Dette går under designprinsippet **konsistent design** som handler om at lignende elementer burde ha samme funksjon. Under ser vi et eksempel på hvordan den rosa fargen opprettholder konsistent design i løsningen.



*Illustrasjon 3 - modal*

I eksempelet ovenfor ser vi også at "Cancel" knappen har et mer nøytralt design, som gjør det tydeligere for brukeren hvilken knapp som utfører en slags funksjon.

Videre skal vi ta en titt på en annen skisse, og eksemplifisere bruken av de gjenværende prinsippene.



Illustrasjon 4 - mapping av nøkler

Ovenfor ser vi skissen av det siste steget i løsningen, hvor nøkler skal mappes til Zeipt sitt format. **Begrensninger** er et viktig designprinsipp å ta i bruk i et slikt steg, hvor brukeren fort kan bli forvirret eller gjøre feil. Et eksempel er måten vi har delt inn feltene under sine respektive overskrifter (headers), i egne dropdown-felt. Grunnen til at vi valgte å designe det på denne måten er fordi brukeren da kan lukke og åpne de forskjellige feltkategoriene som de ønsker, slik at de slipper å bli oversvømt med informasjon.

At brukeren har oversikt over hva som har blitt gjort, og hva som gjenstår å gjøres, faller under designprinsippet **tilbakemelding**. Engness (2014) skriver at tilbakemelding handler om at en handling alltid skal ha en reaksjon, at en utført handling skal føre til et resultat. Måten vi benyttet oss av dette prinsippet baserer seg rundt bruken av ikonene i illustrasjon 4. Når bruker har suksessfullt klart å kartlegge sine felter med Zeipt sine, vil det indikeres med et grønt ikon. Mangler det mapping av felter i en dropdown, vil ikonet være rødt.

Designprinsippet **sammenheng** er også praktisert i illustrasjon 4. Altså sammenhengen mellom kontroller og det de kontrollerer. Å ha en dropdown med nøkler, på samme linje som de forskjellige nøklene til Zeipt, hjelper brukeren med å forstå hvilken dropdown som kontrollerer hver nøkkel.

Bruken av designprinsipper er gjennomgående i designet av løsningen. Å benytte designteorien godt, hjalp oss å utforme et brukervennlig design vi og oppdragsgiver ble fornøyd med, men for å finne ut om designet innfrir usability goals på en god måte, var vi nødt til å ty til en ekstern brukertest.

### 3.4. Ekstern brukertesting av prototype

I oppstartsfasen, før selve utviklingen av løsningen startet, gjennomførte vi en ekstern brukertest ved hjelp av forskningsstrategien undersøkelse (survey). Grunnen til at valget falt på en ekstern brukertest fremfor en tradisjonell en, er tiden vi er i. På grunn av den pågående covid-19 pandemien begrenses vi i forhold til fysisk kontakt. Målet bak en slik undersøkelse er at man vil hente inn data standardisert og systematisk. Deretter kan man finne mønstre i dataene og generalisere disse mønstrene, slik at de gjøres gjeldende for en større del av befolkningen (Oates 2006).

Formålet med testen var å legge til rette for god brukervennlighet i løsningen vår og avdekke problemer i løsningen. For å oppnå dette var vi avhengige av å vite hvordan forskjellige mennesker interagerer med første versjon av løsningen vi har utviklet, for så å videreutvikle løsningen. På bakgrunn av dette trenger vi å utforske meninger og tendenser assosiert med individer, og dette er data som er vanskelig å innhente i en kvantitativ setting (Cresswell 2014).

Fordeler med en slik brukertest er at man kan teste der deltakeren befinner seg, og slipper å bruke tid og energi på å fysisk møte opp. Fordi testdeltakeren da befinner seg i sitt "naturlige" miljø, og ikke opplever å være under observasjon, kan svarene bære preg av at de er mer realistiske og ærlige enn hva de kanskje hadde vært i en tradisjonell setting. Videre har man mulighet til å nå en bredere populasjon mer effektivt fordi man ikke er avhengig av et fast sted (Carr 2020). En slik test er også betydelig mer kostnadseffektiv da man ikke har behov for en hel del utstyr for å teste løsningen, og mange tjenester tilbyr verktøy for ekstern brukertesting har også en funksjon hvor du kan spesifisere målgruppe og andre relevante data, og tjenestene vil automatisk innhente testdeltakere for deg.

En klar ulempe ved en ekstern brukertest er at man vil miste en viss dybde og helhet i forhold til innhenting av data. I motsetning til en tradisjonell brukertest, vil man ikke kunne se hvordan brukeren reagerer, eller kartlegge hvorfor brukeren gjør som en gjør.

Den eksterne brukertesten er utviklet i verktøyet Maze Design som er en plattform som lar oss gjennomføre brukertester online. Maze benytter seg av en rekke byggeblokker som kan brukes i forbindelse med en brukertest. Disse blokkene er:

1. **Oppdragsbasert:** En oppdragsbasert oppgave er et oppdrag testdeltakere skal gjennomføre. Her setter man oppgavetittel, beskrivelse og ønsket måte deltakeren skal interagere med prototypen for å løse oppgaven.

2. **Åpent spørsmål:** Åpent spørsmål brukes når man ønsker en lengre, åpen, tekstopp-gave. Kan være gunstig for å samle data som kontaktinformasjon eller tilbakemeldinger.
3. **Flervalg:** Flervalgsoppgave lar deltakeren velge en eller flere alternative svar til en åpen oppgave.
4. **Meningskala:** Meningskala lar testdeltakeren uttrykke sin mening i forhold til en skala. Skalaen strekker seg fra 0 til 10.
5. **Ja/Nei:** Ja/Nei brukes ved lukkede spørsmål, og kan måles mellom to alternativer. Deltakeren velger et alternativ.
6. **Kontekstbasert:** Kontekstbaserte spørsmål gir deltakeren en ekstra skjerm underveis i testen for å gi mer kontekst der det er nødvendig.

Ettersom brukertesten gjennomføres som en undersøkelse, og at vi tar en mindre mengde data og generaliserer det til en større del av befolkningen/brukergruppen, må vi ta nøyaktighetsområdet og konfidensnivå i betraktning. Nøyaktighetsområdet (feilmarginal/konfidensintervall) forteller noe om hvor nær den sanne populasjonsverdien vi er. Hvis vi måler at 80% av befolkningen mener noe, og vi har et nøyaktighetsområde på +/- 4%, vil den virkelige verdien ligge et sted mellom 76% og 84%. For å ha et nøyaktighetsområde på +/- 0% må hele befolkningen eller brukergruppen testes, avhengig av omfanget av undersøkelsen (Oates 2006). Konfidensnivå er hvor sikre vi er på at den sanne populasjonsverdien faller innenfor verdiområdet oppnådd fra utvalget (Oates 2006).

For å kunne svare på problemstillingen er vi avhengige av å vite hvordan forskjellige mennesker interagerer med første versjon av løsningen vi har utviklet, for så å videreutvikle løsningen. Derfor trenger vi å utforske meninger og tendenser assosiert med individer, og dette er data som er vanskelig å innhente i en kvantitativ setting (Cresswell 2014). Derfor har vi valgt å benytte oss av en brukertest som i all hovedsak produserer kvalitative data.

#### **3.4.1. Deltakere**

Det er normalt for studier å gjennomføre med et nøyaktighetsområde på +/- 3% og et konfidensnivå på 95% (Oates 2006). Når det er sagt, finnes det bred forskning innenfor fagfeltet interaksjonsdesign som kartlegger hvor mange det er hensiktsmessig å brukerteste med. Ettersom dette er mest relevant for testen vi har gjennomført, kombinert med prosjektets tidsperspektiv, har vi valgt å hvile oss på forskning fra Nielsen (2002 og 2012), men er oppmerksom på at nøyaktighetsområde og konfidensnivå tradisjonelt sett er veldig relevant. I forskningen til Nielsen (2002 og 2012) konkluderes det med at man får maksimalt ut av testen ved å teste med fem deltakere. Dette er fordi jo flere brukere som tester, jo mindre vil du etter hvert lære da du kommer til å observere det samme resultatet gjentatte ganger. Ved å bruke fem brukere vil man avdekke 80% av problemene i løsningen.

Under er det lagt ved en tabell over deltakerne, samt noe relevante opplysninger. Som tabellen viser, er 40% av deltakerne testet ikke i målgruppen. Grunnen til at vi ønsket å teste både i og utenfor målgruppen, er at løsningen hviler tungt på teknisk terminologi og begreper, og vi ønsker å se om dette utgjør en stor forskjell mellom gruppene, og hvilke grep vi eventuelt kan gjøre for å gjøre løsningen mer tilgjengelig.

Deltakernr.	Alder	Kjønn	Kunnskap til fagområdet	IT-kunnskap	I målgruppe
34370590	29	Mann	God	God	Ja
34385140	31	Mann	God	God	Ja
34334346	25	Kvinne	Dårlig	God	Nei
34326779	23	Mann	Dårlig	God	Ja
34326459	24	Mann	Dårlig	Dårlig	Nei

Tabell 2: Oversikt over deltakere i brukertest

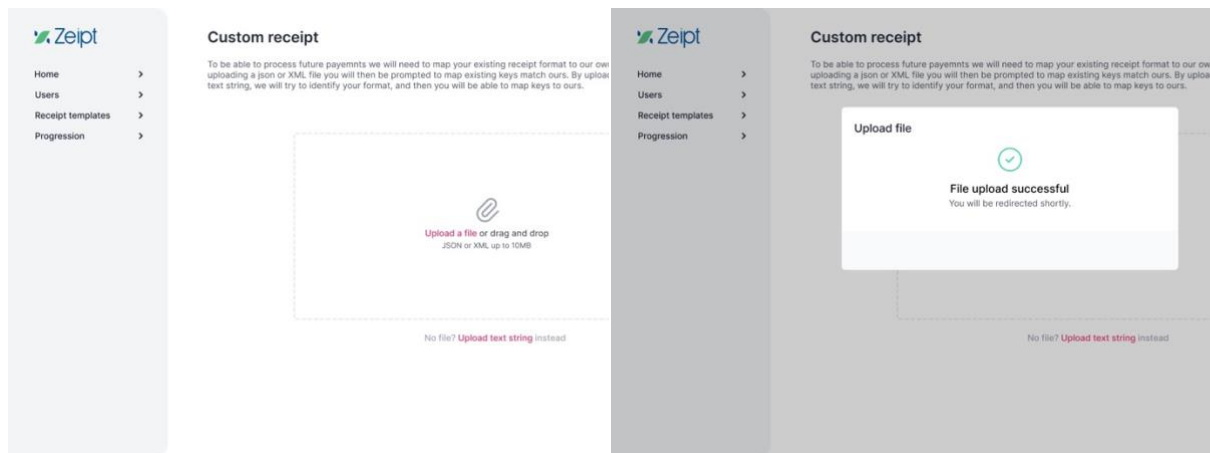
### 3.4.2. Utforming av brukertest

Vi forhåndsdefinerte ti spørsmål/oppgaver, i spesifikk rekkefølge, som ble brukt i brukertesten for å innhente data. Denne måten å hente inn data på er ofte brukt i sammenheng med forskningsstrategien undersøkelse fordi den tilbyr en effektiv måte å hente inn data på (Oates 2006).

Vi har valgt å gjengi de mest relevante oppgavene og deres målsettinger. En komplett oversikt over alle oppgaver utformet i forbindelse med brukertesten kan finnes i Vedlegg F. Større versjoner av bildene under kan finnes i Vedlegg E.

### Oppgave 4: Du vil mappe ditt kvitteringsformat til Zeipt sitt, last opp formatet som JSON/XML.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hvordan han/hun initialiserer opplastningsflyten når det ikke er noen eksisterende kvitteringsmaler.

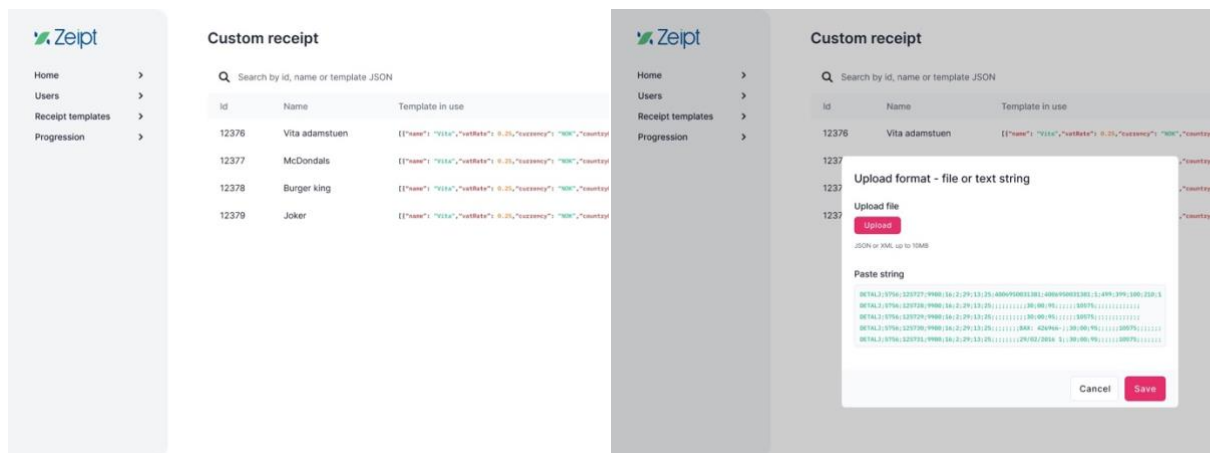


Illustrasjon 5: Prototypen brukt i oppgave 4

På bildet over ser vi testens utgangspunkt hvor man får valget om å enten laste opp en JSON eller XML fil, eller laste opp et annet format, som vi da i løsningen må gjøre om til et format som Zeipt forstår. Her er det tenkt at brukeren i førsteomgang skal laste opp en fil, og det er denne funksjonen som er fremhevet mest, ettersom vi i all hovedsak ønsker data i disse formatene.

### Oppgave 6: Legg til et nytt kvitteringsformat, denne gangen med en text string.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hvordan han/hun initialiserer opplastningsflyten når det allerede er eksisterende kvitteringsmaler.

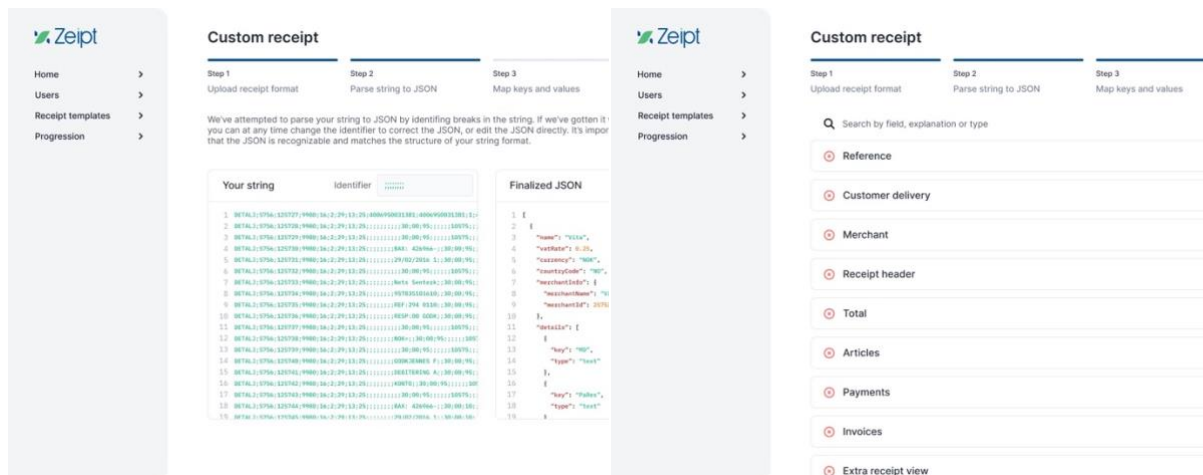


Illustrasjon 7: Prototypen brukt i oppgave 6

På bildet over ser vi dialogen som aktiveres dersom man trykker på "Upload text string instead" i bildet vist i oppgave 4. Her laster brukeren opp dataene sine i et annet format en foretrukket, og vi må da i løsningen gjøre om dette formatet, til et format som Zeipt foretrekker. Her er det tiltenkt at brukeren skal få opp dialogen, lime inn dataene sine i tekstfeltet, og deretter trykke "Save".

## Oppgave 7: Text stringen ble suksessfullt parset til JSON, fortsett til mapping av nøkler og verdier.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hva som må gjøres dersom systemet ikke automatisk klarer å kartlegge en eller flere verdier til Zeipt sine nøkler.



Illustrasjon 8: Prototype brukt i oppgave 7

På bildene over ser vi steget hvor brukeren ser sitt omgjorte resultat, og får beskjed om å dobbeltsjekke at strukturen og dataene ser logisk ut. Ved å fortsette fra dette steget vil brukeren bli tatt videre til kartleggingssteget hvor de kartlegger sine egne nøkler opp mot Zeipt sine.

### 3.4.3. Funn i brukertest

Resultatene fra testen er gjengitt i tabellen nedenfor i et samlet format. Tabellen er delt inn i oppgavenummer, observasjoner og tolkninger. En mer detaljert oversikt over resultatene kan finnes i Vedlegg G.

Oppgave	Observasjoner	Tolkninger
1	60% av deltakerne hadde erfaring og/eller var fagutdannet innenfor teknologi og systemutvikling.	60% er innenfor målgruppen, og disse svarene kommer derfor til å vektas tyngre enn svarene til de som ikke er det.
2	40% av deltakerne jobbet med teknologi / systemutvikling innenfor en bedrift som driver salg av varer og tjenester til privatpersoner. Av disse 40% var det kun halvparten (1/5) som hadde erfaring og/eller var	Svarene til denne deltakeren som i tillegg har erfaring og/eller som er fagutdannet innenfor teknologi og systemutvikling vil vektas tyngre enn de resterende 4/5.



	fagutdannet innenfor teknologi og systemutvikling.	
3	100% av deltakerne svarte at digitale kvitteringer er relevant for dem.	Digitale kvitteringer, og Zeipt sin forretningsmodell er relevant for folk både i, og utenfor, målgruppen til løsningen som skal utvikles.
4	Deltakerne brukte i snitt 21,9 sekunder på oppgaven, og hadde en misklikkrate på 40%. Alle deltakerne løste oppgaven.	Deltakerne som ikke er i målgruppen prøvde å klikke flere steder før de klarte å løse oppgaven, mens deltakerne i målgruppen skjønnte hvor de skulle klikke med en gang. Vi kan konkludere med at flyten er tilstrekkelig, så fremt man er kjent med terminologi og konvensjoner knyttet til faget.
5	Deltakerne brukte i snitt 6,1 sekunder på oppgaven, og hadde i snitt en misklikkrate på 10%. Alle deltakerne løste oppgaven.	Deltakerne klikket der de skulle og kan derfor konkludere med at oppsettet med kartlegging av nøkler er intuitiv nok.
6	Deltakerne brukte i snitt 5,4 sekunder på oppgaven, og hadde i snitt en misklikkrate på 50%. Alle deltakerne løste oppgaven til slutt.	Deltakerne som ikke er i målgruppen prøvde å klikke flere steder for de klarte å løse oppgaven, og at deltakerne som er i målgruppen skjønnte hvor de skulle klikke med en gang. Det kan virke som de utenfor målgruppen var forvirret av terminologi. Vi kan konkludere med at flyten er tilstrekkelig, så fremt man er kjent med terminologi og konvensjoner knyttet til faget.
7	Deltakerne brukte i snitt 6,2 sekunder på oppgaven, og hadde i snitt en misklikkrate på 40%. Alle deltakerne løste oppgaven.	Deltakerne som ikke var i målgruppen skjønnte med en gang hvordan de skulle gå videre for å løse oppgaven, mens deltakerne i målgruppen prøvde å klikke på andre elementer. Dette kan være fordi de som hadde kunnskap tilknyttet fagområdet prøvde å undersøke at alt var riktig før de fortsatte, mens de som ikke hadde

		kunnskap tilknyttet bare fortsatte. Vi kan konkludere med at flyten ikke er tilstrekkelig, og at vi må videreutvikle dette steget.
8	Deltakerne brukte i snitt 2,4 sekunder på oppgaven, og hadde i snitt en misklikkrate på 0%. Alle deltakerne løste oppgaven. Vi ser at 4/5 deltakere klikket der de skulle med en gang, mens 1/5 klikket på feil butikk i oversikten.	Misklikken fra 1/5 vektlegges ikke betraktelig, da vi ser at disse klikkene skjedde i løpet av det første sekundet deltakeren fikk tilgang til oppgaven, og kan derfor tyde på slurv og/eller at deltakeren ikke tok seg lang nok tid til å tilegne seg informasjonen på siden. Vi tar også i betraktning at selv da klikker deltakeren på en knapp med tilsvarende funksjon, men tilknyttet feil rad. Vi kan konkludere med at deltakerne skjønnte hvordan eksisterende kvitteringsmaler kan slettes.
9	Deltakerne brukte i snitt 1,4 sekunder på oppgaven, og hadde i snitt 0% misklikkrate. Alle deltakerne løste oppgaven. Vi ser at 100% av deltakerne umiddelbart klikket der de skulle.	Vi kan konkludere med at det er intuitivt hvordan nylige slettede kvitteringsmaler kan gjenopprettes.
10	Tilbakemeldingene fra deltakerne oppsummert var designet var lett, intuitivt og brukervennlig. Samtidig reagerte noen av deltakerne på at det var terminologi i både spørsmål og oppgave som de ikke forstod, og 1/5 deltakere synes at spørsmålene kunne vært mer tydelige.	Oppgavene var tilstrekkelig utformet for å teste løsningen, og vi kan konkludere med at resultatene i testen er valide og kan brukes i videre utvikling.

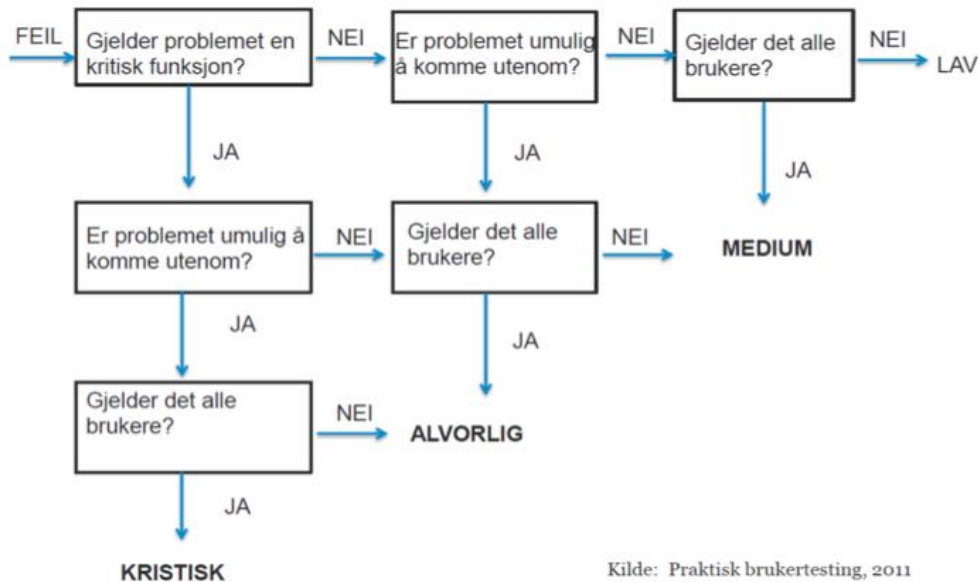
Tabell 3: Samlet oversikt over observasjoner og tolkninger i brukertesten

#### 3.4.4. Konklusjon

Ifølge funnene i brukertesten fungerer prototypen stort sett greit, og legger til rette for god brukervennlighet. Ved hjelp av et simpelt oppsett og en logisk sidestruktur kan vi se at løsningen er lett å lære selv for en som aldri har tatt den i bruk før.

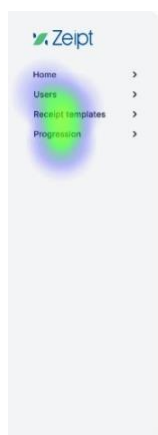
Brukertesten avdekker dog noen problemer rundt designprinsippene synlighet, sammenheng og hint.

For å ta stilling til alvorlighetsgraden av problemene avdekket har vi tatt utgangspunkt i problemkategoriseringsmodellen (Toftøy-Andersen og Wold 2011):

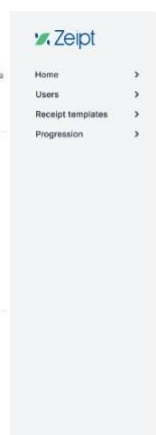


Illustrasjon 11: problemkategoriseringsmodell

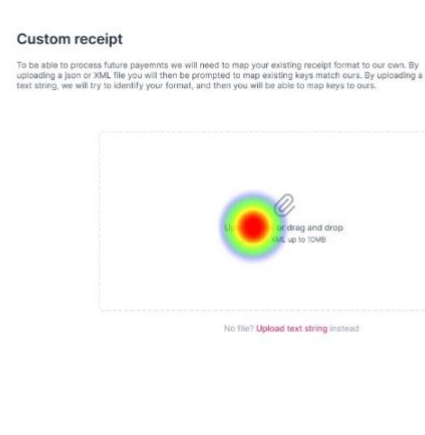
Vi ser først og fremst av de deltakerne som ikke er utdannet eller har erfaring innenfor faget sliter når det benyttes fagbegreper og terminologi. I oppgave 4 ser vi at deltakerne som ikke er i målgruppen prøver å trykke på veldig mye annet før de faktisk klarer å løse oppgaven, mens deltakerne i målgruppen skjønner med en gang hvor de skal trykke for å løse oppgaven. Dette illustreres med følgende illustrasjoner:



Illustrasjon 12: Oppgave 4 - Klikk fra deltakere utenfor målgruppen



Illustrasjon 13: Oppgave 4 - Klikk fra deltakere i målgruppen

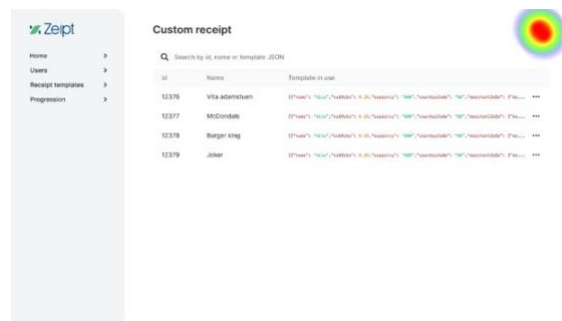


Man kan derfor konkludere med at flyten er tilstrekkelig, så fremt man er kjent med terminologi og konvensjoner knyttet til faget, men velger også å kategorisere problemet i kategorien “medium”. Vi vektlegger at målgruppen løste oppgaven slik den var tenkt, men problemet gjelder tross alt en kritisk funksjon som er umulig å komme utenom, men gjelder ikke alle brukerne.

Videre i oppgave 6 ser vi at deltakerne som ikke er en del av målgruppen igjen prøver å klikke flere steder før de klarte å løse oppgaven, og igjen spesielt hvor fagterminologi er benyttet. Dette illustreres av følgende illustrasjoner:



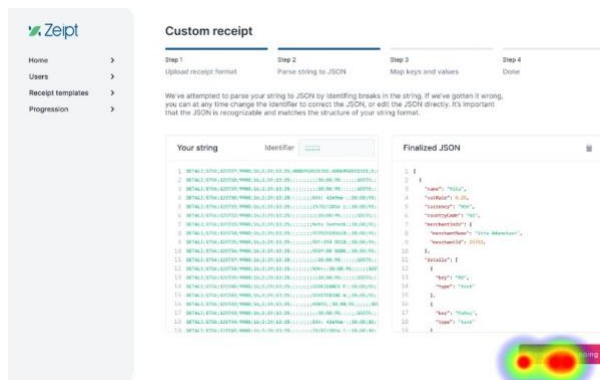
Illustrasjon 14: Oppgave 6 - Klikk fra deltakere utenfor målgruppen



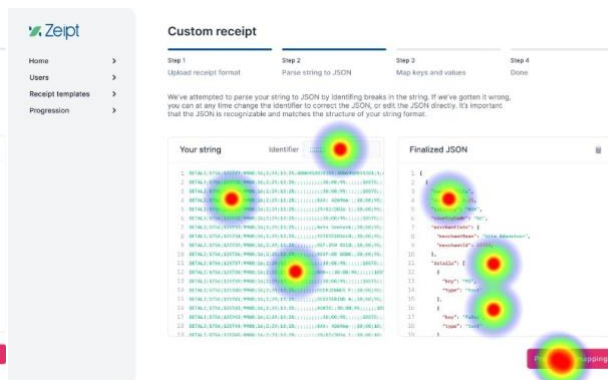
Illustrasjon 15: Oppgave 6 - Klikk fra deltakere i målgruppen

Ettersom målgruppen løser oppgaven slik den var tiltenkt, uten problemer, kan vi konkludere med at flyten i utgangspunktet er tilstrekkelig, men at det finnes forbedringspotensiale for å inkludere en større gruppe mennesker. Vi definerer dette problemet i kategorien medium. Dette fordi det gjelder en kritisk prosess, problemet er umulig å komme utenom, men det gjelder ikke alle deltakerne. Når det er sagt vektlegger vi at målgruppen løser oppgaven som tiltenkt.

Når det kommer til oppgave 7 ser vi at deltakerne som ikke var i målgruppen med en gang skjønnte hvordan de skulle gå videre for å løse oppgaven, mens deltakerne i målgruppen prøver å klikke på forskjellige elementer. Dette illustreres ved følgende illustrasjoner:



Illustrasjon 16: Oppgave 7 - Klikk fra



Illustrasjon 17: Oppgave 7 - Klikk fra

*deltakere utenfor målgruppen*

*deltakere i målgruppen*

Dette kan ha noe med at de som hadde kunnskap tilknyttet fagområdet prøvde å undersøke at alt var riktig før de fortsatte, mens se som ikke hadde kunnskap tilknyttet bare fortsatte. Vi kan derfor konkludere med at flyten ikke er tilstrekkelig, og at vi må videreutvikle dette steget, ettersom deler av målgruppen hadde problemer. Vi definerer dette problemet i kategorien alvorlig, fordi det gjelder en prosess kritisk funksjon som det er umulig å komme utenom, men gjelder ikke alle deltakerne.

Vi tar med oss disse funnene videre inn i designprosessen, og vil utbedre de problemene avdekket før vi starter med utvikling av løsningen. Eksempelvis vil vi endre litt på flyten i modalen som dukker når man laster opp en tekst string (oppgave 6) for å gjøre steget i oppgave 7 litt enklere, og med mindre informasjon enn det som er der per dags dato.

## 4. Teknisk løsning

### 4.1. Utvikling

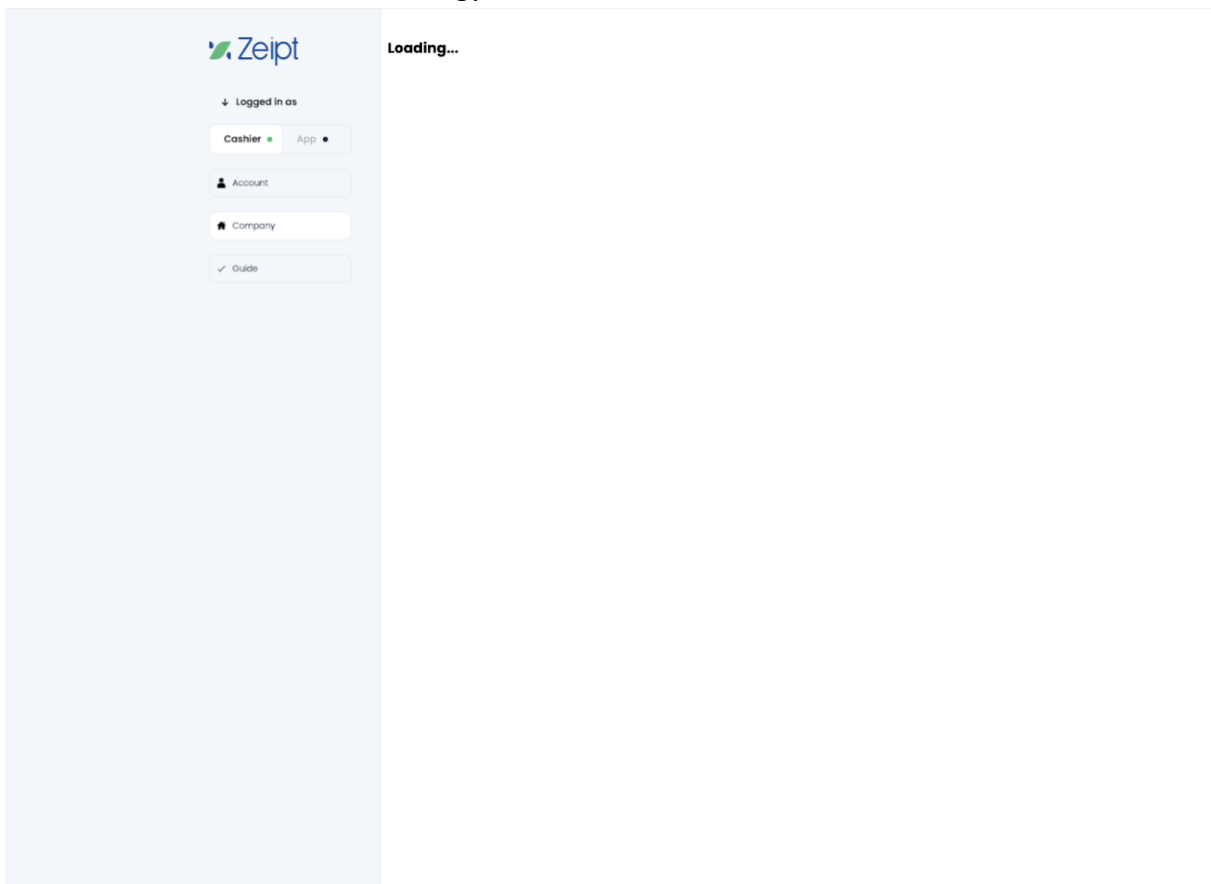
All koding i løsningen er skrevet i form av JavaScript-rammeverket React. Zeipt benytter seg eksklusivt av React i deres kodebase, noe som gjorde det til et åpenbart valg. For styling av komponenter bruker Zeipt pakken “Styled Components”, som komplementerer bruken av React. For lagring brukte vi Redux, som er et bibliotek for håndtering av state i JavaScript applikasjoner. State er med på å påvirke hvordan applikasjonen oppfører seg, og er en slags data-bank for applikasjonen. Måten man bruker Redux på er å ha en “store”, hvor all state lagres og opprettholdes.

#### 4.1.1. Utvikling av GUI

I dette punktet vil vi vise frem den ferdige løsningen og gå i detalj rundt hvordan ulike funksjoner fungerer og hvordan vi kom frem til et ferdig resultat. Zeipt satt ikke krav for at løsningen skulle være responsiv, fordi løsningen kun skal benyttes av brukere med datamaskin.

### Utgangspunkt

Opgaven vi valgte hadde ingen utgangspunkt. All design og kode måtte vi produsere selv, på en måte som komplementerte Zeipts design og kodelstil. Under er en illustrasjon over hvordan dashbordet så ut før vi begynte å kode:

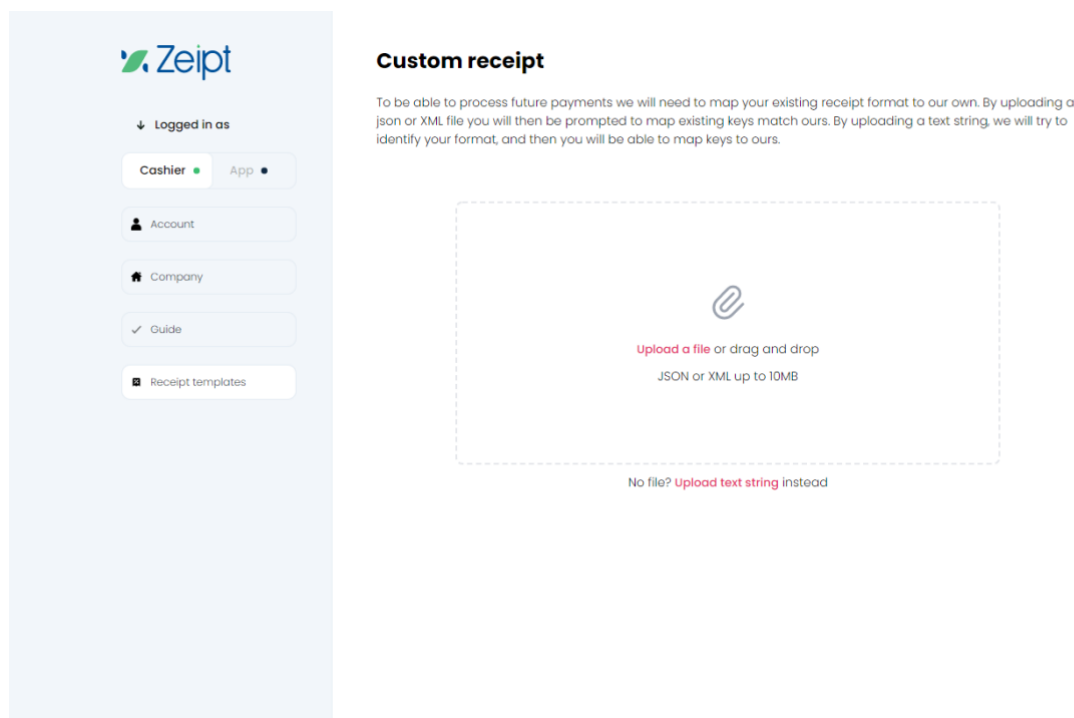


Illustrasjon 18: Utgangspunkt før design og kode

### Steg 1 - opplasting av kvitteringsformat

I steg 1 har brukeren to valg, enten laste opp en fil, eller en string. Velger bruker å laste opp en fil kontrolleres dette i en egen komponent som validerer den opplastede filen. Er ikke filen av typen JSON eller XML, vil brukeren få en feilmelding som ber de om å laste opp korrekt filformat. Hvis filen er av riktig format, vil brukeren få mulighet til å laste opp, og deretter bevege seg til neste steg. Filopplastnings-komponenten vil da lese innholdet og sende det til Redux store, slik at innholdet er tilgjengelig i de neste stegene.

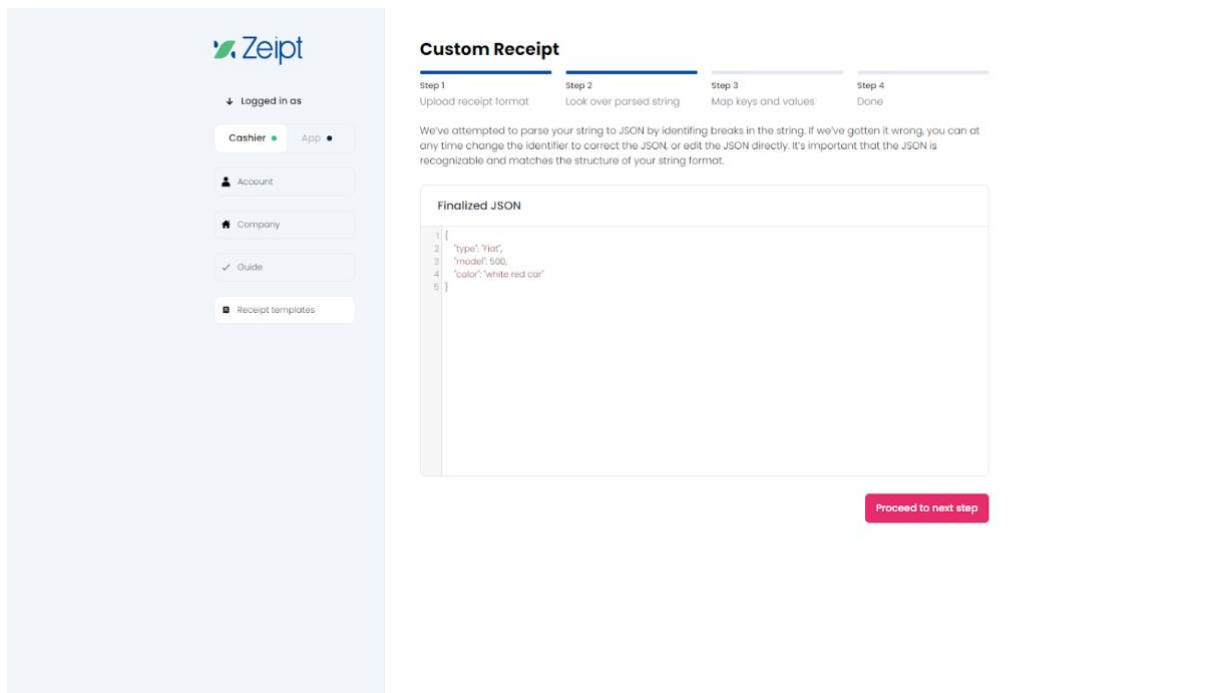
Velger brukeren å laste opp en string i motsetning til en fil, vil løsningen forsøke å "oversette" stringen om til JSON, som er Zeipts foretrukne format (se punkt 4.1.2. for detaljert gjennomgang av oversetter). Blir stringen suksessfullt omgjort til JSON, lagres dataene i Redux store og brukeren blir sendt til neste steg.



Illustrasjon 19: startside for løsningen (opplasting av kvitteringsformat)

## Steg 2 - se over opplastet fil eller string

Steg 2 i løsningen er steget hvor brukeren kan dobbeltsjekke at innsendt data er presentert riktig. Her blir dataene vist i JSON format, og brukeren kan gå inn og redigere hvis noe skulle være feil. Rent funksjonelt henter steget dataene som ligger i Redux store, og presenterer dem i en IDE (Integrated Development Environment). IDEen er en egen pakke som heter react-codemirror-2, som lar oss presentere data på denne måten.



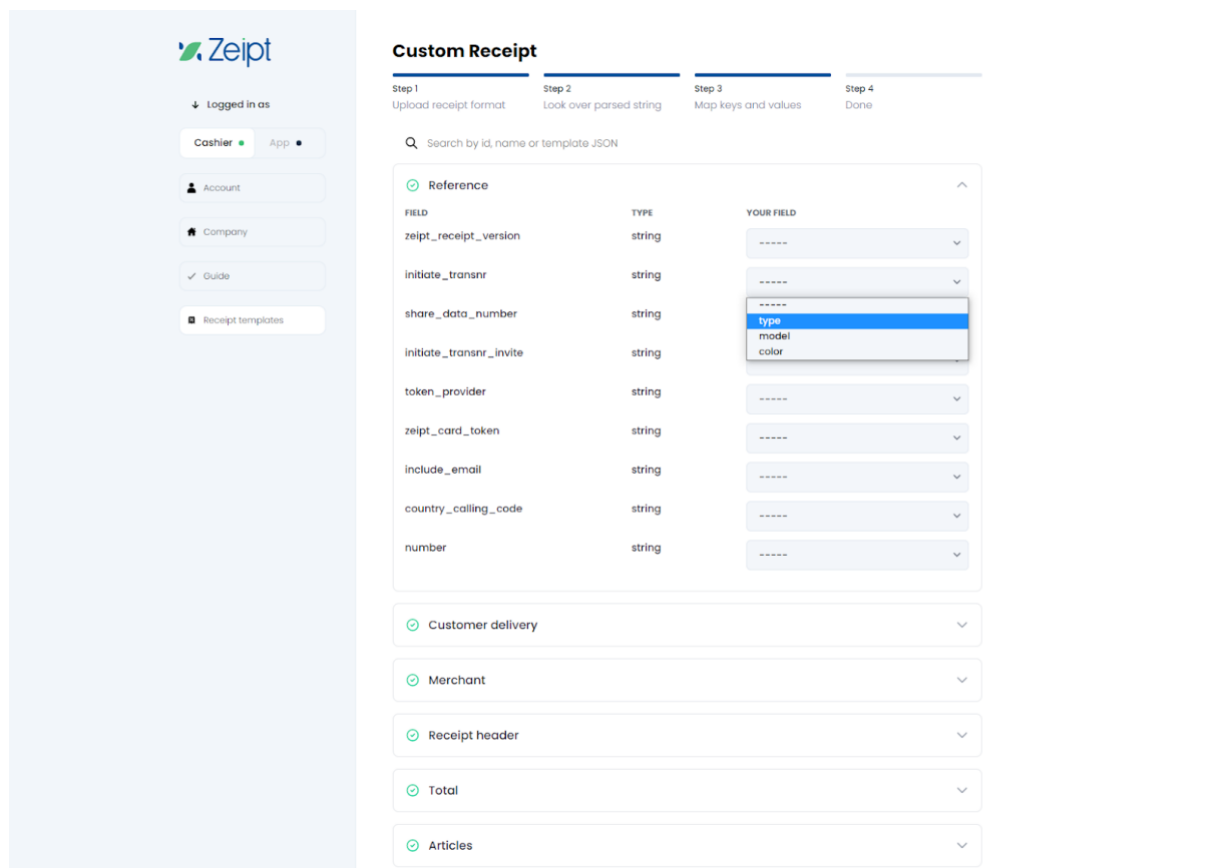
Illustrasjon 20: overse opplastet fil/string



### Steg 3 - mappe nøkler og verdier

I dette steget skal brukeren kartlegge verdier fra den opplastede dataen til Zeipts nøkler. Måten vi valgte å løse dette på var å utvikle dropdowns som presenterer Zeipt sine nøkler på en oversiktlig måte. Verdiene fra dataene i Redux state blir så presentert som alternativer i nok en dropdown hvor bruker velger alternativet som passer med Zeipt sin nøkkel. Nøkkel i denne sammenhengen er altså Zeipt sine predefinerte felt vi ser i illustrasjonen ovenfor.

Kartleggingen av felter og dropdowns er dynamisk, så hvis Zeipt noen gang skal oppdatere sitt eget kvitteringsformat, vil de kunne gjøre det bare ved å erstatte det gamle formatet. Funksjonen i koden vil kunne dynamisk legge til felter i enhver JSON man legger inn.



The screenshot displays the 'Custom Receipt' configuration interface. On the left is a sidebar with the Zeipt logo and user information: 'Logged in as Cashier App', 'Account', 'Company', 'Guide', and 'Receipt templates'. The main area has a progress bar with four steps: Step 1 (Upload receipt format), Step 2 (Look over parsed string), Step 3 (Map keys and values), and Step 4 (Done). Below the progress bar is a search bar and a table for mapping fields. The 'Reference' section is expanded, showing a table with columns for FIELD, TYPE, and YOUR FIELD. A dropdown menu is open for the 'share\_data\_number' field, showing options: type, model, and color.

FIELD	TYPE	YOUR FIELD
zeipt_receipt_version	string	-----
initiate_transnr	string	-----
share_data_number	string	type model color
initiate_transnr_invite	string	-----
token_provider	string	-----
zeipt_card_token	string	-----
include_email	string	-----
country_calling_code	string	-----
number	string	-----

Illustrasjon 21: Mapping

#### 4.1.2. Utvikling av oversetter

En annen viktig del av oppgaven vi fikk utdelt var å utvikle en oversetter som har som jobb å ta andre data i form av andre formater enn det som er ønsket, og deretter gjøre om dataene sendt inn til data som Zeipt forstår.

Kjernen av denne oppgaven sentrerer seg rundt å finne gjentakende tegn, eller sekvens av tegn, som vi identifiserer og kan bruke til å separere et par med nøkler og verdier (separator). Et par med nøkler og verdier består av to relaterte data elementer, hvor man da har en nøkkel som er en konstant som definerer dataen, for eksempel kan dette være kjønn, farge, pris, osv., og en verdi, som er en variabel som tilhører den konstanten, for eksempel kan dette være mann/kvinne, grønn, 100, osv.

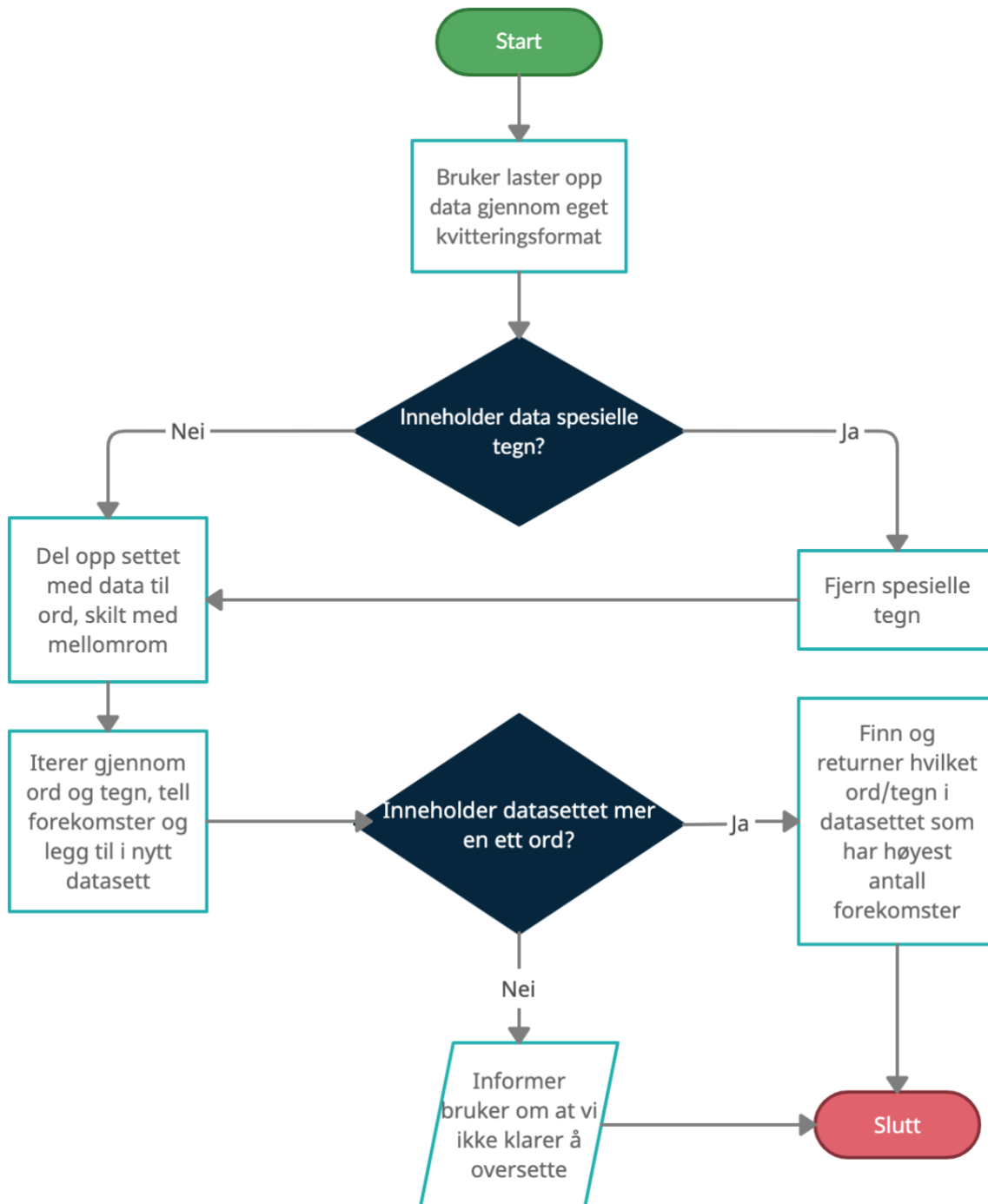


*Illustrasjon 22: Forklaring av sentrale begreper ved et nøkkel-verdi par*

Videre når vi har kartlagt hvilket tegn, eller sekvens av tegn, det opplastede formatet bruker for å separere nøkkel-verdi par, kan vi bruke denne til å bearbeide den opplastede dataen. Jobben består altså i all hovedsak av to prosesser: finne separator og oversette basert på separatoren.

### Finne separator

Først og fremst må vi finne ut hvilket tegn, ord, eller sekvens av tegn, som brukes til å separere par med nøkkel og verdier. Under er et flytdiagram som viser selve prosessen:



Illustrasjon 23: Flytdiagram over prosessen av å finne separator

Det første vi sjekker når brukeren laster opp dataene sine, er om det første og siste tegnet er spesielle tegn. Med spesielle tegn mener vi tegn som ikke er alfabetiske eller numeriske. Dette kan eksempelvis være punktum, spørsmålstejn, osv. Vi gjør denne sjekken fordi vi

ønsker kun å forholde oss til eventuelle nøkkel-verdi par, og de aller fleste standard formatene for presentering av datasett benytter seg av spesielle tegn for å vise hvor datasettet starter og slutter. Hvis dataene inneholder spesielle tegn som første og siste tegn fjerner vi disse.

Videre forsøker vi å dele opp dataene i segmenter basert på mellomrom i teksten. Det vil si at teksten: "Dette er et eksempel" vil bli delt opp i fire segmenter: "Dette" "er" "et" "eksempel". Dette gjør vi rett og slett for å telle hvor ofte hvert enkelt ord/tegn forekommer i de gitte dataene ved å iterere gjennom hele settet med segmenter.

Når vi har telt opp alle ordene, og hvor ofte de forekommer, gjør vi en ny sjekk hvor vi sjekker om datasettet med opptalte ord har flere enn ett ord. Hvis vi kun har klart å finne et ord, er det mye som tyder på at vi ikke har klart å separere ordene på rett måte, og vi gir da brukeren beskjed om at vi ikke klarer å oversette dataene de har lastet opp. Hvis det er flere enn et ord i datasettet, finner vi det segmentet med høyest antall forekomster, og returnerer dette. Et eksempel på hvordan setningen "Hvordan fungerer det å dele opp setninger? Jo, det fungerer slik", ser ut etter at vi har delt den opp, og telt forekomster:

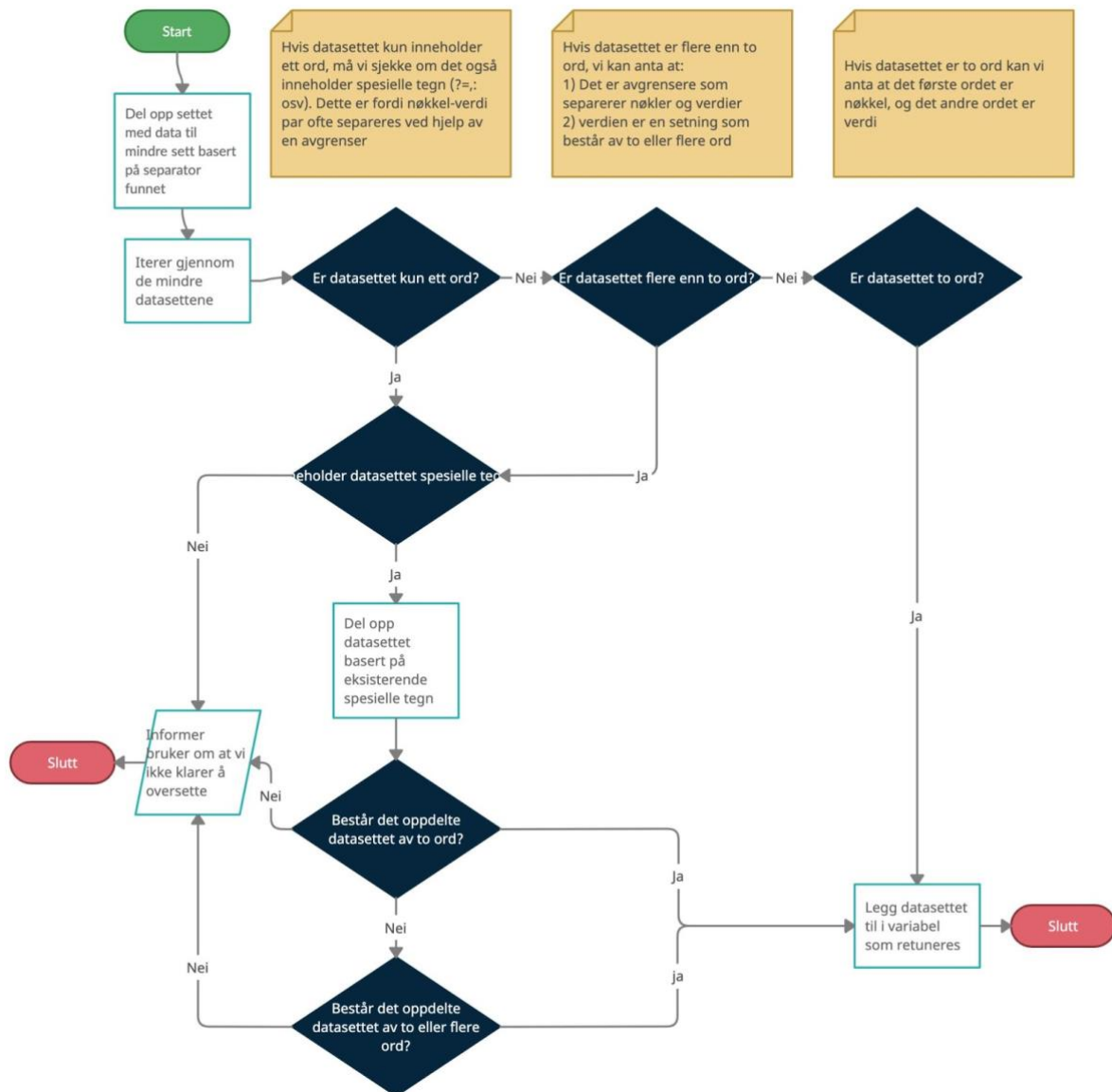
Ord	Forekomster
Hvordan	1
fungerer	2
det	1
å	1
dele	1
opp	1
setninger	1
Jo,	1
det	1
slik	1

Tabell 4: Oversikt over hvordan systemet strukturerer opptellingen av forekomster

I eksempelet over vil ordet "fungerer" returneres som det segmentet eller tegnet som forekommer mest.

## Oversette basert på funnet separator

Når vi har funnet hvilken separator det brukes til å separere nøkkel-verdi par, kan vi fortsette med å prøve og bryte opp det opprinnelige datasettet ytterligere, basert på den identifiserte separatoren. Under er et flytdiagram som viser selve prosessen:



Illustrasjon 24: Flytdiagram over prosessen å oversette dataene til ønsket format

Det første som skjer i denne prosessen er at vi tar separatoren funnet i forrige steg, og bruker denne til å dele opp de opplastede dataene på nytt. Da vil vi i teorien kun sitte igjen med nøkkel-verdi par, hvis separatoren er kartlagt riktig. Deretter itererer vi nok en gang gjennom hver enkelt nøkkel-verdi par og sjekker et par ting:

Vi tar utgangspunkt i at det opplastede datasettet ser slik ut, og vil bruke dette eksempelvis nedover:

```
[1] {type:"Fiat", model:"500", color:"white and red"}
```

**Antall ord:** Hvor mange ord er det hvis vi deler opp nøkkel-verdi paret om vi deler opp basert på mellomrom? Her er det et par ting som kan skje, og som vi må ta hensyn til. Det er ikke alltid at datasett benytter seg av mellomrom, og veldig ofte er dette tilfelle fordi det går raskere for en datamaskin å behandle dataene sendt inn hvis datasettet er så kort som mulig. Videre kan det skje at vi sitter igjen med et sett med to ord. I dette tilfellet antar vi at det første ordet er nøkkel, og det andre ordet er verdi. Vi kan også ende opp med flere enn to ord, i hvilket tilfelle vi er nødt til å bryte det ned enda mer for å se om vi klarer å kartlegge hva som er nøkkel, og hva som er verdi.

Ved å bruke datasettet over, som allerede er delt opp basert på separatoren ",", og vi da prøver å dele det opp på nytt basert på mellomrom vil det se slik ut (legg merke til at separatoren er fjernet):

```
[1] type:"Fiat"  
[2] model:"500"  
[3] color:"white and red"
```

Her vil det første tilfellet inntreffe, ettersom det ikke er noe mellomrom mellom nøkkel, avgrensener og verdi.

**Spesielle tegn:** Dersom det splittede nøkkel-verdi paret består av ett ord, eller flere enn to, sjekker vi om det inneholder spesielle tegn. Veldig ofte deles nøkler og verdier ved hjelp av en avgrensener, og denne avgrenseneren er som oftest et spesialtegn. Hvis det finnes spesielle tegn, tar vi da det gitte nøkkel-verdi paret og deler det opp basert på det spesielle tegnet. Vi vil da sitte igjen med to eller flere ord. Hvis vi bruker eksempelet over vil resultatet nå se slik ut (legg merke til at avgrenseneren er fjernet):

```
[1] type "Fiat"  
[2] model "500"  
[3] color "white and red"
```

Dersom vi sitter igjen med to ord antar vi at det første er nøkkel, og det andre er verdi. Hvis vi sitter igjen med flere enn to ord etter vi har delt opp basert på spesialtegn, antar vi at det første ordet er nøkkel, og de resterende ordene er verdi, som en lengre setning.

Ved å hele tiden bearbeide dataene stegvis vil vi til slutt klare å identifisere de faktorene vi trenger, og kan på den måten bruke de dataene på den måten, og i det formatet, som vi selv ønsker.

Det er noen tilfeller hvor vi rett og slett ikke klarer å bearbeide dataene videre, og vil da gi beskjed til brukeren om at vi ikke klarte det, og at de må laste opp en litt mer raffinert versjon av dataene sine. Hadde vi hatt mer tid, hadde vi videreutviklet å involvere bruker mer for å løse de problemene som oppstår, samt utviklet en løsning for å håndtere tilfeller hvor det er nestede datastrukturer.

#### **4.1.3. Dokumentasjon av kode**

Et viktig aspekt i programvareutvikling er kommenteringen og dokumentasjonen av den koden man skriver, for at det både skal være lett for en selv, og andre, å raskt sette seg inn i koden. Kommentering i kode-verdenen er å legge inn setninger og notater gjennom koden som forklarer hvordan programmet virker og intensjonen av den spesifikke kodeblokken. Ettersom målet med prosjektet var at Zeipt skal kunne videreutvikle løsningen ved overtagelse, er dokumentasjon av koden og hva som skjer noe vi har vektlagt i stor grad gjennom hele prosjektet. Et eksempel på dette er kommenteringen av koden som kjører i oversettelsesprosessen, illustrert under:

```

splitStrByBreakpoint
  .filter((item) => item)
  .forEach((element) => {
    // if the element only consists of one word we have to check
    // if it contains a special character. This is because key-value
    // pairs often is spilt by a special character, without any space
    // between the key, separator and value
    if (splitString(element.trim()).length === 1) {
      // check for special characters, and split on the
      // special character
      if (element.match(/^[^a-zA-Z0-9 ]/g)) {
        let splittedElement = utils.checkForAndSplitOnSpecialChars(element)
        // check if there are only two elements in the array.
        // If it is, we can assume that the first is key and
        // the second is value
        if (splittedElement.length === 2) {
          utils.extendObject(splittedElement[0], splittedElement[1])
          setCurrentlyParsing(false)
          setParsingSuccess(true)
        } else {
          // if we split the string on a spacial character, and there is
          // anything but two values, we're not able to identify which is
          // the key, and which is the value. Most likely, if this else is fired
          // we are only able to find the key.
          setCurrentlyParsing(false)
          setParsingSuccess(false)
          alert('We are not able to parse the string. Please upload a refined version.')
        }
      } else {
        // if we split the string on special characters, and the element
        // is only one word, we are not able to parse the string
        // e.g. the result would look like ['key'], and not ['key', 'value']
        setCurrentlyParsing(false)
        setParsingSuccess(false)
        alert('We are not able to parse the string. Please upload a refined version.')
      }
    }

    // if the amount of words in the element is above 2,
    // we can assume either that
    // 1) there are delimiters that separates keys and values,
    //    e.g. :, =, ? etc
    // 2) the value is a longer string, consisting of more than 1 word
  } else if (splitString(element.trim()).length > 2) {
    // check if special characters are present in string
    if (element.match(/^[^a-zA-Z0-9 ]/g)) {
      const splittedElement = utils.checkForAndSplitOnSpecialChars(
        element
      )

      // since we now have removed special characters, we can
      // assume that the first word is the key, and any following
      // words/numbers/characters are values.
      // however, we need to handle cases where the value is a longer string
      // of words.
      if (splittedElement.length >= 2) {
        utils.extendObjectWithLongerValue(splittedElement)
        setCurrentlyParsing(false)
        setParsingSuccess(true)
      } else {
        // if we split the string on special characters, and the element
        // is only one word, we are not able to parse the string
        setCurrentlyParsing(false)
        setParsingSuccess(false)
        alert('We are not able to parse the string. Please upload a refined version.')
      }
    } else {
      // if the string is longer than two words, but does not contain
      // special characters, we can assume that the value is a longer string.
      // in that case we take the first element as key, removes it, and joins the

```

Illustrasjon 25: Grundig dokumentert kode



#### 4.1.4. utfordringer

I React har man noe man kaller state og hook. Staten brukes til å påvirke hvordan koden oppfører seg, og hvilke parametere den benytter seg av når man trenger input fra brukeren som man skal bruke i koden som kjøres. En hook er en metode som lar deg "hekte" deg inn på React sin kjernefunksjonalitet, og lar deg enkelt bruke denne i din egen kode. I oversetteren (parser) har vi benyttet oss av "useState"-hooken til React for å hente ut og deklare en lokal state i komponenten. Når man benytter seg av denne hooken definerer man variabler for å hente ut, og sette ny verdi i state.

Problemet med useState, og en stor utfordring underveis i utviklingen, er at hooken re-deklarerer verdier asynkront. For å forklare nærmere, så vil all synkron kode kjøre nedover som programmet kjøres. Det betyr at hvis du trenger en variabel i en funksjon kalt X, som du henter gjennom en annen funksjon kalt Y, må funksjon Y kjøre før funksjon X. Dette løser man enkelt ved å bare plassere funksjon Y før X, og programmet vil da vente på at Y blir ferdig før den starter på X. Ved asynkron kode er det derimot ikke så lett, fordi koden vil fortsette å kjøre X uten at Y er ferdig. Det som skjer da er at X plutselig ikke har den variabelen den trenger og vil da resultere i en feil. Normalt løses dette ved at asynkrone funksjoner returnerer et løfte om en ny verdi i fremtiden, og som også gir beskjed til koden at den må vente på den nye verdien, men det gjør ikke useState hooken til React.

Måten vi løste dette på er at vi sjekker før koden starter om state eksisterer. Hvis den ikke gjør det kjører vi en synkron funksjon først som returnerer den verdien vi er ute etter. State vil bare være satt om vi har bedt om input fra brukeren, og han/hun har endret på hvilken separator vi skal bryte opp datasettet deres med.

Styled Components, CSS-rammeverket Zeipt bruker for styling i deres kodebase var noe vi ikke hadde mye erfaring med fra før, og viste seg å være en utfordring å benytte seg av i starten av kodingen. Dette var fordi rammeverket er såpass ulikt noe vi har vært borte i før, som førte til at arbeidsflyten lugget litt i starten. Etter hvert som vi ble mer vant med rammeverket, ble det enklere å opprettholde en god arbeidsflyt.

#### 4.2. Forslag til videre utvikling

I funksjonen som har som jobb å finne separatoren er det for øyeblikket ikke noe validering i forhold til om en nøkkel finnes flere steder i det opplastede datasettet (duplikater). Av tidshensyn ble dette bortprioritert. Det er veldig sjeldent at like nøkler forekommer, da de som nevnt tidligere identifiserer en spesifikk verdi. Det gir derfor ikke mening å gi to forskjellige verdier samme identifikator. Ut fra de testene vi har gjort ser vi dersom vi har to like nøkler vil funksjonen overskrive disse med den seneste verdien. I videre utvikling kan dette enkelt håndteres med forhåndsvalidering av det datasettet som sendes inn.

Videre håndterer ikke oversetteren nestede datastrukturer. Nestede datastrukturer er datastrukturer inni datastrukturer. Vi hadde dessverre ikke tid til å eksplisitt håndtere dette, men har skrevet en del liknende kode flere steder i løsningen som kan brukes som utgangspunkt for å implementere dette.

Som illustrert i flytdiagrammet i punkt 4.1.2. er det et par edge cases hvor vi “gir opp” å prøve og oversette datastrukturen videre. Hadde vi hatt mer tid hadde vi videreutviklet en løsning for å involvere bruker mer for å løse de problemene som oppstår.

Opprinnelig hadde vi planlagt å ha en side i løsningen hvor brukeren hadde oversikt over alle kvitteringsformatene de hadde oversatt med Zeipt sitt. Denne siden ble både designet og kodet, men mangel på tid gjorde at vi måtte legge fokus på andre ting. Veien videre ville vært å ferdigstille siden, hvor bruker kunne ha mulighet til å slette eller redigere sine kvitteringsformat.

## 5. Vurdering av løsning og prosjekt

### 5.1. Teknisk resultat

Vi har i dette prosjektet tatt på oss en bred og omfattende oppgave med å utvikle et verktøy som lar oppdragsgivers kunder kartlegge og koble deres eget kvitteringsformat, uavhengig av filtype og innhold, sammen med Zeipt sitt format. Dette gjøres ved hjelp av et grafisk grensesnitt, og en oversettelsesfunksjon der det trengs. Utviklingen av en slik løsning vil kraftig senke onboarding tiden for Zeipt sine fremtidige kunder da de slipper å oversette deres kvitteringsformat lokalt før de sender det til endepunktene. Dette har vi brukt mye tid og ressurser på for at resultatet skal være av den kvaliteten og nytten vi strekker oss etter. Dette, sett sammen med gruppens medlemsantall, har gjort at det ikke har vært mulig, innenfor prosjektperioden, å skrive gode unit-tester til koden.

Store deler av løsningen er svært visuell, og det er derfor lite hensiktsmessig å skrive unit-tester til kode som stort sett gir visuelle resultater. Unit-tester blir kun kjørt i en emulator, og testresultatet gjenspeiler ikke alltid virkeligheten. Når det er sagt kunne vi dratt nytte av en slik type testing i selve oversettelsesfunksjonen, for å sikre at vi har tatt hensyn til alle "edge casene".

Selv om vi ikke har skrevet unit-tester, har vi hele veien testet programvaren fysisk med mange forskjellige parametere. Et eksempel på dette er at vi har testet med mange forskjellige eksisterende formater i oversettelsesfunksjonen, samt formater som ikke er tradisjonelt brukt for å sikre at den fungerer optimalt.

### 5.2. Nytte for oppdragsgiver

Et av de viktigste kriteriene fra Zeipt sin side var at løsningen hadde en nytteverdi og noe som de kunne bruke i videre utvikling. I løpet av prosjektperioden har vi absolutt hatt en fordel av å komme inn som utenforstående. Vi har kommet med tanker og innspill som oppdragsgiver selv ikke hadde tatt stilling til, og vi har hatt en åpen dialog rundt hele prosjektperioden for å sammen utvikle det beste resultatet mulig.

Ettersom utgangspunktet vi ble tildelt var ganske magert, har vi kunne benyttet oss av frie tøyler i forhold til hvordan vi ønsker å løse oppgaven både design- og funksjonsmessig, noe vi har dratt nytte av. Vi har hatt mye fokus på design og layout for å vise løsningen fra en innbydende og spennende side.

I tillegg til selve løsningen har vi gjort et grundig innsiktsarbeid gjennom den eksterne brukertesten som vil være nyttig for Zeipt når løsningen skal utvikles videre. Gjennom arbeidet vi har gjort har vi lagt til rette for at oppdragsgiver kan lage et produkt som har tydelig identitet, og som treffer målgruppen løsningen er ment for.

### 5.3. Vurdering av metode, prosjektgjennomføring og verktøy

#### Prosjektgjennomføring

Vi valgte å benytte oss av Scrum som utviklingsmetode, som tidligere beskrevet i 2.2. Valg av utviklingsmetode. Tanken bak valget var å benytte en utviklingsmetode som la til rette for en iterativ utviklingsprosess, med forholdsvis satte rammer. For oss som bare var 2 på gruppen, fulgte vi ikke rammeverket til punkt og prikke, men tok heller nytte av mange av metodene innenfor Scrum vi følte var til hjelp for utviklingsprosessen. Rammeverket har hjulpet oss med å hele tiden forbedre løsningen i sprintene, etterfulgt av møte med oppdragsgiver for diskusjon rundt ytterligere forbedringer. Det at Scrum gjorde det mulig å inkludere produkteier i beslutningsprosessene, har gjort det enklere å utvikle et produkt som gjenspeiler det oppdragsgiver ønsker.

Noen utfordringer med vår bruk av Scrum var at daglige møter med daily standups var vanskelig å opprettholde slik vi ønsket, med tanke på gruppens størrelse var det større sannsynlighet for at halve gruppen kunne være utilgjengelig.

Prosjektet har vært en lang læringsprosess. Vi valgte en krevende oppgave for å utfordre oss selv, som har hjulpet oss ut av komfortsonen. Zeipt har gjort gjennomførelsen av prosjektet til en morsom oppgave og ga oss stor frihet til å være kreative.

#### Verktøy

I dette punktet vil vi se nærmere på de som hadde størst innvirkning på prosjektets resultat.

De viktigste verktøyene vi benyttet oss av var Figma og Visual Studio Code. Uten disse ville vi aldri endt opp med det vi i dag kan si oss fornøyd med. I designprosessen ble Figma avgjørende for utforming av designet, og lot oss jobbe om hverandre mot et resultat som oppdragsgiver kunne si seg fornøyd med. Visual Studio Code lot oss ha en god arbeidsflyt i kodedelen av prosjektet, og ga oss en strukturert måte å jobbe på.

Også verdt å nevne er Discord som kommunikasjonsplattform for oss innad i gruppen. Med tanke på Covid-19 pandemien gjorde Discord det mulig for oss å effektivt jobbe sammen, når det var utfordrende å møtes.

### 5.4. Vurdering av resultat i forhold til teori

Gjennom hele prosjektet har det vært viktig å utvikle en løsning med høy grad av brukskvalitet gjennom å forankre arbeidet vårt i teori tilknyttet faget. Vi har hovedsakelig lent oss på teori som Donald Norman og Jakob Nielsen. Samtidig har brukerens behov blitt ivaretatt på bakgrunn av litteratur fra Preece, Rogers og Sharp (2019) og Norbø (2017). Selv om forskningen bak designprinsippene, kognitive prosesser, og sammenhengen mellom psykologi og design gjort av Norman i 1988, er å anse som gamle, står den fortsatt støtt den

dag i dag. Teoriene bak forskningen, som er hvordan brukeren måte å oppfatter verden på, endrer seg ikke. Det som endrer seg derimot, er brukerens erfaringer, forutsetninger og behov, og det er disse aspektene vi er nødt til å ha i bakhodet når vi designer og utvikler løsninger. Derav er en brukersentrert tilnærming viktig for å skape noe av verdi, og som leverer en høy grad av brukskvalitet.

Vi har gjennom prosjektets første faser jobbet utfra en brukersentrert tilnæringsprosess. Dette har hjulpet oss mye for å legge til rette for at brukeropplevelse, interaksjon og informasjonsarkitektur ble god. Gjennom brukertesting og mål for brukskvalitet har vi hvilt oss mye på Preece, Rogers og Sharp (2019), kombinert med tilbakemeldinger, teknikker og metoder oppdragsgiver har erfaring med fra tidligere gjennomførte tester.

Alle disse kildene har vært til stor hjelp gjennom prosjektperioden, da spesielt med tanke på gjennomføring brukertesting.

## 6. Konklusjon

*Hvordan utvikle en web basert kvitteringsmal-bygger som er brukervennlig for sluttbruker?*

Vi har utviklet en løsning som lar sluttbruker kartlegge og koble deres eget kvitteringsformat, uavhengig av filtype og innhold, sammen med Zeipt sitt format som viser hvordan man ved hjelp av en brukersentrert tilnærming kan bygge en enkel og brukervennlig løsning som tilfredsstillende sluttbrukers ønsker og behov. Ved å lage denne løsningen som et utgangspunkt Zeipt lett kan videreutvikle, effektiviseres og reduseres onboarding tiden for fremtidige kunder, da de slipper å oversette deres kvitteringsformat lokalt før de sender det til de aktuelle endepunktene, og kan istedenfor benytte seg av et GUI som veileder dem gjennom prosessen.

Løsningen vår er et utgangspunkt på hvordan formidlingen av informasjon og brukervennlighet gjennomføres. Gjennom brukertesting av prototypen har vi opparbeidet oss en verdifull innsikt som vi har benyttet oss av underveis. Innsikten, og løsningen utviklet, er et produkt som oppdragsgiver kan bygge videre på. Med lengre tid og flere ressurser kunne en forbedring av punktene beskrevet i 4.2. vært gjennomført. Når det er sagt er vi veldig fornøyd med produktet vi leverer.

Ettersom vi hadde valgt en veldig åpen problemstilling, så har vi ut fra oppgaven presentert, klart å scope prosjektet ned, og tatt vurderinger på hva som er nødvendig for at løsningen vi leverer fra oss skal ha en viss nytteverdi, slik at ikke omfanget på prosjektet ikke skulle bli for stor. Det har vært både gøy og lærerikt å gjennomføre en såpass åpen oppgave. Vi sitter igjen med en forståelse for hvordan man angriper reelle problemstillinger i arbeidslivet, samt hvordan man må jobbe strukturert for å holde seg innenfor oppgavens scope. Samtidig har vi lært mye når det kommer til gjennomføring av en ekstern brukertest.

På grunnlag av dette kan vi konkludere med at innsikten samlet inn og løsningen vi overleverer svarer på problemstillingen. Vi har bevist gjennom innsikt, samarbeid med oppdragsgiver og brukertester at løsningen som er utviklet både har en nytteverdi og oppnår en høy grad av brukervennlighet for Zeipts fremtidige kunder.

Dette gjenspeiles også i oppdragsgivers vurdering av løsningen og innsiktsarbeidet vi har utført. For å sitere Sebastian, så *"[...] Det märks också att studenterna inte bara har följt de specifikationerna som lades fram till dem, utan att de även har reflekterat gott över hur upplevelsen ifrån användarens ögon skulle kunna se ut, och att de korrigerat sin tillnärmning under projektets gång för att bibehålla fokus på "smartnes" för kund"*.

Vi er derfor fornøyd med produktet og innsatsen levert, og at vi oppnår målsettingen vår, beskrevet i punkt 1.4., om at løsningen skulle ha en viss nytteverdi for oppdragsgiver.

## Referanseliste

- Cadle, James og Yeates, Donald. 2006. Project Management for Information Systems. Pearson Education Limited.
- Carr, Steven. 2020. "5 benefits of on-demand customer feedback you can enjoy from wherever you work best". Hentet 12. mai 2021.  
<https://www.usertesting.com/blog/on-demand-customer-feedback>.
- Cresswell, John W. 2014. "Research Design: Qualitative, Quantitative and Mixed Methods Approaches". Sage Publications Inc.
- Discord.com 2021. "We built Discord to help you create belonging". Hentet 11. mai 2021.  
<https://discord.com/company>.
- Enginess. 2014. "The 6 Principles Of Design, a la Donald Norman". Hentet 16. mai 2021.  
<https://www.enginess.io/insights/6-principles-design-la-donald-norman>.
- European Union. 2021. "Data protection under GDPR". Oppdatert 26. mars 2021.  
<https://europa.eu/youreurope/business/dealing-with-customers/data-protection/data-protection-gdpr/>.
- Fagerstrøm, Asle. Eg, Ragnhild. Johannessen, Magne og Vogt, Nina. 2020. Forbrukeratferd. Oslo: Gyldendal Akademisk. E-bok.
- Garrett, Jesse James. 2010. The Elements of User Experience: User-Centered Design for the Web and Beyond. Pearson Education.
- Geekbot.com. 2020. "Daily Standup Meetings: Everything You Need To Know (Standup Agenda, Purpose, Common Pitfalls, and More!)". Hentet 12 mai 2021.  
<https://geekbot.com/blog/daily-standup-meeting/>.
- Gube, Jacob. 2009. "Call to Action Buttons: Examples and Best Practices". Hentet 14. mai 2021.  
<https://www.smashingmagazine.com/2009/10/call-to-action-buttons-examples-and-best-practices/>.
- Nes, Solveig Maria. 2019. "En kort introduksjon til Scrum". Hentet 12. mai 2021.  
<https://www.visma.no/blogg/en-kort-introduksjon-til-scrum/>.
- Nielsen, Jakob. 2000. "Why You Only Need to Test with 5 Users". Hentet 7. april 2021.  
<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.

- Nielsen, Jakob. 2012. "How Many Test Users in a Usability Study". Hentet 7. april 2021. <https://www.nngroup.com/articles/how-many-test-users/>.
- Nordbø, Tone. 2017. Introduksjon til interaksjonsdesign. Oslo: Universitetsforlaget.
- Norman, Donald A. 2013. The Design of Everyday Things: Revised and Expanded Edition. Basic Books.
- NSD. 2021. "Norsk senter for forskningsdata". Hentet 5. april 2021. <https://www.nsd.no/>.
- NSD. 2021. "Informasjon til deltakerne". Hentet 5. april 2021. <https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/sjekkliste-for-informasjon-til-deltakerne/>.
- NSD. 2021. "Fylle ut meldeskjema for personopplysninger". Hentet 5. april 2021. <https://www.nsd.no/personverntjenester/fylle-ut-meldeskjema-for-personopplysninger/>
- NSD. 2021. "Hvordan gjennomføre et prosjekt uten å behandle personopplysninger?". Hentet 5. april 2021. <https://www.nsd.no/personverntjenester/oppslagsverk-for-personvern-i-forskning/hvordan-gjennomfore-et-prosjekt-uten-a-behandle-personopplysninger/>
- Oates, Briony J. 2006. Researching Information Systems and Computing. Sage Publications Ltd.
- Preece, Jenny, Yvonne Rogers, og Helen Sharp. 2019. Interaction Design: Beyond Human-Computer Interaction. John Wiley & Sons.
- Proff.no 2021. "Zeipt AS". Hentet 9. mai 2021. <https://www.proff.no/selskap/zeipt-as/oslo/internettdesign-og-programmering/IF5UEDEOC2C/>.
- Scrumdictionary.org 2021. "Definiton of Product Review". Hentet 16 mai 2021 <https://scrumdictionary.com/term/product-review/>.
- Scrum.org 2021. "What is a Daily Scrum?". Hentet 16 mai 2021 <https://www.scrum.org/resources/what-is-a-daily-scrum>.
- Toftøy-Andersen, Eli og Wold, Jon Gunnar. 2011. Praktisk brukertesting. Cappelen Damm Akademisk.



Visual Studio Code. 2021. "Why did we build Visual Studio Code?". Hentet 13 mai 2021.  
<https://code.visualstudio.com/docs/editor/whyvscode>.

## Vedlegg

### Vedlegg A: Begrepsliste

Begrep	Betydning
GUI	Graphical User Interface (Grensesnitt). En visuelt grensesnitt hvor bruker kan interagerer med elektroniske komponenter.
State	Objekt data lagres i. Brukes i de tilfeller man ønsker å hente input fra bruker i React.
Hook	En metode som lar deg "hekte" deg inn på React sin kjernefunksjonalitet.
Mapping	Kartlegging.
Input	Et aspekt som lar brukeren kommunisere med programmet som kjøres, og gi den rettningslinjer hvis antagelser er feil.
JSON	JavaScript Object Notation. En enkel tekstbasert standard for datautveksling.
XML	Extensible Markup Language. Markup standard for datautveksling.
String	Tekststreng. En av flere datatyper i programmering. Brukes for å beskrive data av typen tekst.
Edge case	Et problem som oppstår ved ekstreme (maks eller minimum) operasjonsparametere. Brukt for å beskrive situasjoner det er vanskelig å ta hensyn til.
IDE	Integrated Development Environment. Et program som legger til rette for programvareutvikling for utviklere.

**Vedlegg B: Gruppekontrakt**

Mål, visjon og ambisjonsnivå

Vi skal gjennomføre et prosjekt og utarbeide en rapport med tilhørende prosessdokumentasjon innen 20.05.2021.

Målet for prosjektet er at oppdragsgiver skal være fornøyd med resultatet og vi som gruppe skal være fornøyd med innsatsen lagt inn i prosjektet. Vårt karaktermål innenfor faget er A.

Prosjektet skal gjennomføres med like stort fokus på prosessen som rapporten, men viktigst er det å ha kundens behov i fokus.

Vi ønsker at alle gruppemedlemmene skal trives i prosjektet og i gruppen. Viktigste kriterium for suksess er at kunden er tilfreds med resultatet og at akseptansetesten resulterer i aksept fra kundens side.

Roller og ansvarsområder

Prosjektleder: Daniel Wendt Kjellid

Prosjektleder (vara): Ludvik Schanke Blunck

Øvrige ansvarsområder delegeres og rulleres underveis.

Prosjektleders overordnede ansvarsområder innebærer:

- Ansvar for å ha oversikt over prosjektets fremgang sett i forhold til prosjektplanen.
- Ansvar for å lede arbeidsdager og interne møter.
- Ansvar for utforming av prosjektplan.
- Ansvar for å ha en oversikt over hva gruppens medlemmer arbeider med til enhver tid.

Prosedyrer

Kommunikasjon og møter

Kommunikasjon vil hovedsakelig foregå over facebook (prosjektets facebookgruppe), mail, og telefon.

Problemstillinger diskuteres og løses av gruppen i felleskap. Avgjørelser blir tatt i konsensus.

Mail

Mail ut fra gruppen skal gå via gruppens avtalte mailadresse.

Kalender

Google Calendar skal benyttes for å gi medlemmene en enkel oversikt over møter og lignende aktiviteter.

#### Møter

Det skal skrives referat av alle møter, der det er relevant. Referat skal benytte seg av utformet mal. Rollen som møtereferent rulleres blant gruppens medlemmer.

Interne møter skal avholdes ved behov. Møtets omfang og struktur avgjøres av prosjektleder.

Gruppens medlemmer tar en daglig status med hva som skal jobbes med.

#### Fravær og sykdom

Planlagt fravær, som for eksempel arbeid eller ferie, skal meldes fra om i god tid.

Sykdom skal meldes fra om snarlig via en melding til prosjektleder. Ved sykdom forventes det at det syke medlemmet bidrar så langt det lar seg gjøre

#### Ansvar og forventninger

Gruppens medlemmer er sammen ansvarlige for prosjektets fremgang i henhold til prosjektplan.

Gruppens medlemmer forplikter seg til å utføre sine arbeidsoppgaver og innenfor gitte tidsfrister.

Det forventes at alle skal bidra til et godt og lærerikt arbeidsmiljø.

Det forventes aktiv deltakelse ved møter og i arbeid.

Det forventes at gruppe-medlemmer melder fra om mistriksel og lignende forhold til prosjektleder.

#### Plagiat

Plagiat aksepteres ikke. Plagiat defineres som det å kopiere andres arbeid, i sin helhet eller i deler, uten tillatelse og/eller å referere til arbeidets opphav.

#### Kontraktsbrudd

Kontraktsbrudd vil resultere i følgende prosedyre:

1. Det vil bli avholdt et møte for å diskutere bruddets omfang, og hva som kan gjøres for å forhindre lignende brudd i fremtiden.
2. Ved gjentakende brudd vil den interne veilederen bli involvert og rådført.
3. Skjer lignende brudd på nytt etter intern veileder har blitt involvert, vil det bli gitt en skriftlig advarsel. Veileder vil bli informert om situasjonen.

4. Ved enda flere brudd på kontrakten og avhengig av bruddenes alvorlighetsgrad, vil veileder bli ombedt om å vurdere konsekvenser. Disse konsekvensene kan i verste fall innebære separat vurdering eller ekskludering fra prosjektet.

#### Revidering

Revidering av kontrakt skal diskuteres og gjennomgås på møte hvor samtlige av gruppens medlemmer er tilstede. Endringer skal godkjennes enstemmig.

#### Interpersonlige spørsmål

Vi skal respektere andres meninger, men også si vår egen. Taushet er enighet.

Problemer diskuteres og løses av gruppen i fellesskap (konsensus) så langt det lar seg gjøre. Ros og konstruktiv kritikk er ønskelig både på møter og ellers i prosjektet.

**Vedlegg C: Prosjektplan**

Aktivitet	Uker																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Oppstart	█																			
Analyse og forarbeid			█	█																
Utvikling				█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Akseptansetester																			█	█
Levering																				█
Rapportering	█																			

Tabellen viser den overordnede planen for prosjektet, som vil løpe i 20 uker. Fasene i prosjektet er fordelt i aktivitetene oppstart, analyse og forarbeid, konfigurering, utvikling, akseptansetester, levering og rapportering. Hver aktivitet har hver sin farge, og de tykke vertikale linjene markerer avhengigheter. Ukenumrene som er mørke markerer viktige datoer i løpet av prosjektperioden:

- Uke 4: Oppstartsfasen ferdig
- Uke 5: Analyse- og forarbeidsfasen ferdig
- Uke 13: Innlevering av første utkast av rapport
- Uke 19: Konfigurerings- og utviklingsfasen ferdig
- Uke 20: Akseptansetest ferdig, innlevering av resultat og ferdig rapport

**Oppstart**

Fasen består av å definere og konkretisere oppgaven, konsept og beskrivelsen av hva som skal utføres og lages. I oppstartsfasen vil det også bli laget en overordnet, generell, plan og prosjektbeskrivelse. Det vil også bli utarbeidet en risikoplan og gruppekontrakt.

**Analyse og forarbeid**

Fasen består i all hovedsak av å legge grunnlaget for videre arbeid i prosjektet. Her vil vi analysere og researche problemstillingen, og finne ut hvordan denne kan løses på best mulig måte, samt hvilket verktøy og teknologi som kan brukes til å løse problemstillingen. Videre skal vi avdekke behov for eventuell opplæring i verktøy og/eller teknologien som skal brukes. Med dette i utgangspunkt vil det lages en mer detaljert plan. Denne fasen bestod også av prototypeutvikling og brukertesting.

**Utvikling**

Utviklingsfasen vil ha fokus på videreutvikling av Zeipt sin dashboard løsning, med fokus på å lage en Web basert kvitteringsmal-bygger. Denne fasen vil bli brutt ned i mindre bestander i en mer detaljert plan. Fasen vil gjennomføres i iterasjoner.

#### Akseptansetester

Når prosjektet nærmer seg slutten vil det bli gjennomført akseptansetester for å sikre at kvaliteten og oppdragsgivers forventninger er oppfylte på en tilfredsstillende måte.

#### Levering

Denne fasen vil bestå av klargjøring for levering av produkt og rapport.

#### Rapportering

Denne fasen vil løpe parallelt under hele prosjektet, og vil i leveringsfasen resultere i en sluttrapport. Det vil også bli levert et første utkast i midten av siste halvdel av prosjektet.

**Vedlegg D: Risikoanalyse**

Sans = Sannsynlighet, Kons = Konsekvens, Total = Totalvurdering

<b>Faktor og beskrivelse</b>	<b>Sans (1-10)</b>	<b>Kons (1-10)</b>	<b>Total</b>	<b>Ansv.</b>	<b>Forebyggende tiltak</b>	<b>Tiltak om risiko slår inn</b>
<b>Lengre fravær</b> Uforutsett fravær over lengre tid (mer enn 3 dager) grunnet sykdom og/eller andre hendelser.	3	10	30	Gruppen	Prosjektleder har en oversikt over hvilke arbeidsoppgaver gruppemedlemmene til enhver tid gjør, for å enkelt kunne kartlegge hvilke oppgaver som ikke utføres.	Det andre gruppemedlemmet overtar oppgaven.
<b>Kortere fravær</b> Uforutsett fravær over kortere tid (mindre enn 3 dager) grunnet sykdom og/eller andre hendelser.	5	7	35	Gruppen	Prosjektleder har en oversikt over hvilke arbeidsoppgaver gruppemedlemmene til enhver tid gjør, for å enkelt kunne kartlegge hvilke oppgaver som ikke utføres.	Fraværende gruppemedlem gjør arbeidet på egenhånd.
<b>Forholdsstyring</b> En faktor som kan inntreffe om oppdragsgiver ikke ser nytten av prosjektet, at vi ikke tar hensyn til oppdragsgivers behov, og/eller at løsning ikke møter de krav som oppdragsgiver/kunde har.	2	10	20	Gruppen	Sette en felles forståelse rundt forventninger, oppgaven og behov ovenfor oppdragsgiver.	Definere oppgaven sammen med oppdragsgiver, og sette forventninger og behov deretter.
<b>Planlegging</b> Oppsett av tidsfrister som både er realistiske og tilpasset det scopet satt for oppgaven.	4	10	40	Gruppen	Sørge for at oppgavens størrelsesomfang er gjennomførbart.	Tiltak for scope vil iverksettes.
<b>Bedriftsmiljø</b> Endringer av interne og eksterne faktorer i bedriften. Eksempelvis utskiftning i ledelsen og andre økonomiske	2	6	12	Gruppen, Zeipt	Sikre tidlig at gruppen får tilgang på de verktøyene og ressursene vi har behov for, fra oppdragsgiver. Ellers er en slik endring lite sannsynlig med	Om risikoen skulle inntreffe, må gruppen raskt få klarhet i hvordan de ser for seg videre samarbeid, og sørge for at vi får samme oppfølging som er definert i kontrakten.

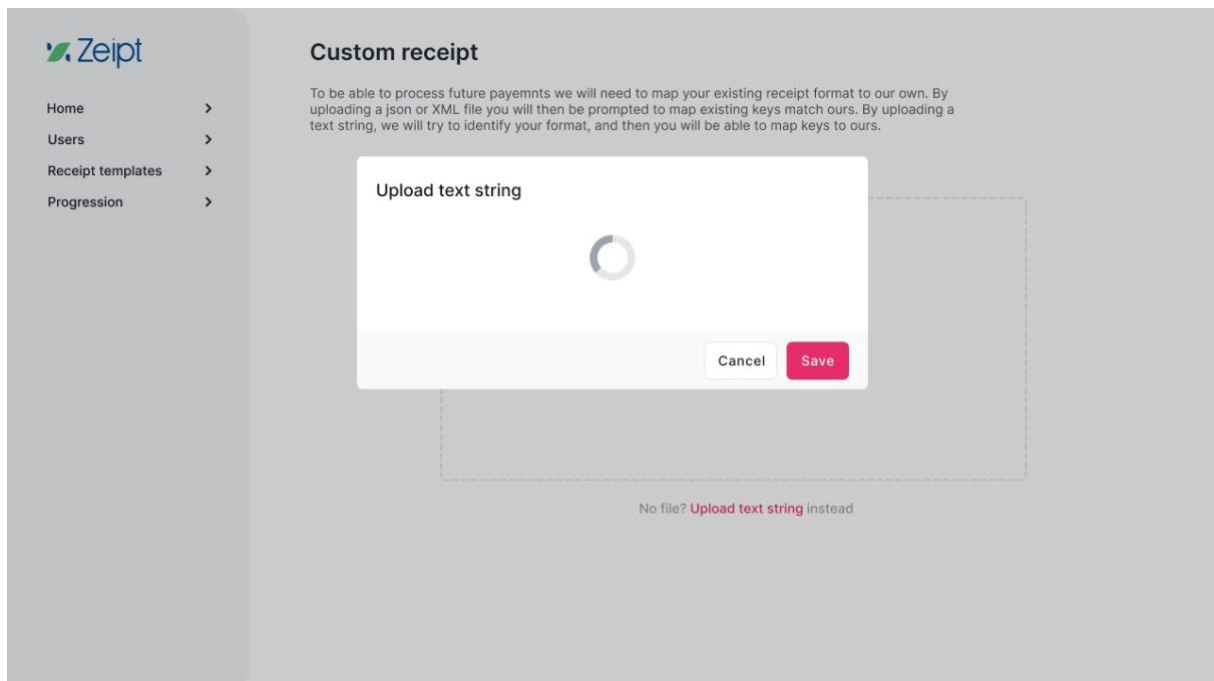
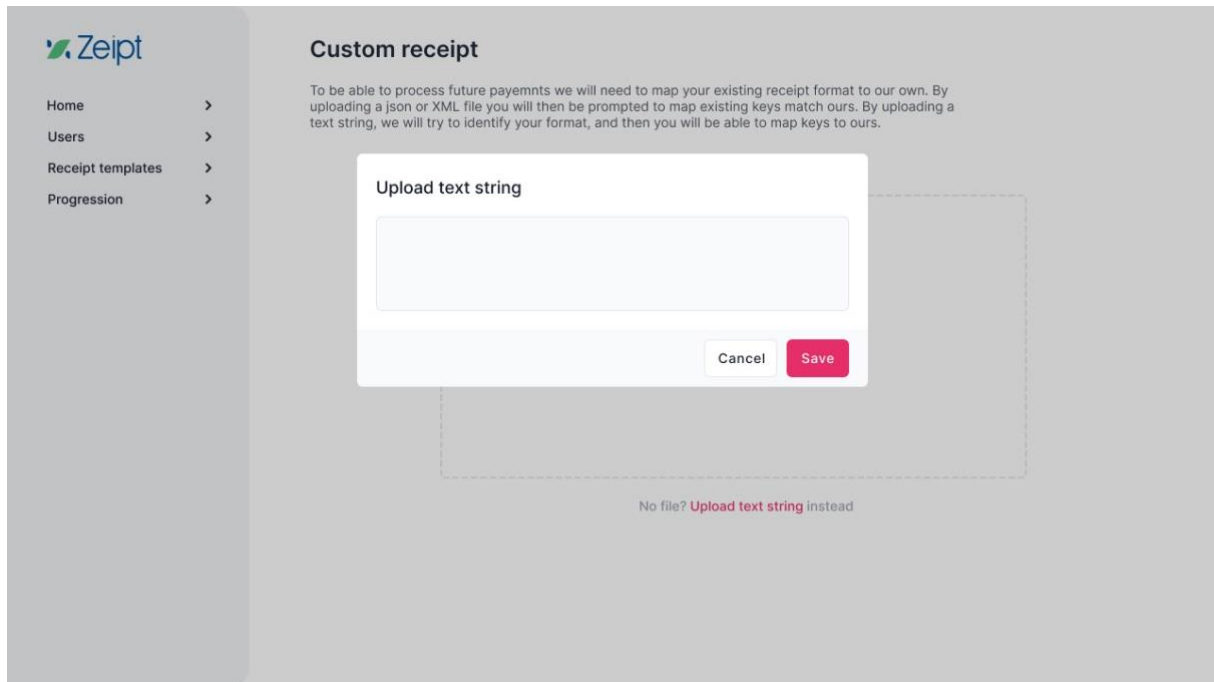



faktorer som fører til redusert kapasitet.				bedriftens nåværende situasjon.		
<b>Utviklingsprosess</b> Manglende plan og metode. Kan resultere i dårlig effektivitet, dokumentasjon og testing av produkt.	6	6	36	Gruppen	Sette fokus på effektivitet og unytte den korte prosjektperioden. Tidlig sette oss inn i eksisterende kodebase, arkitektur og programmeringsspråk. Benytte oss av en iterativ prosess slik at oppdragsgiver er involvert i prosessen hele veien.	Et internt gruppemøte det diskuteres ulike friksjonspunkter som setter stopper for effektiviteten. Finne ut hvordan vi kan overkomme disse, og iverksette tiltak.
<b>Prosjektstyring</b> Ledelse, styring og administrering av gruppen. Kartlegging av tilgjengelige og nødvendige ressurser, samt generell kontroll og tidsstyring av prosjektet.	6	6	36	Gruppen	Sørge for nok kunnskap om oppgaven. Gruppen har definert en overordnet leder som skal ha kontroll på prosjektet. Videre må vi bestemme oss for en prosjektstyringsmetode som passer til oppgaven.	Dersom valgt gruppeleder mangler ferdigheter til å styre og administrere gruppen overtar et annet medlem denne rollen. Dersom problemet vedvarer kontaktes intern veileder for råd.
<b>Kompetanse</b> Manglende kompetanse til å fullføre oppgaver.	3	7	21	Gruppen	Sørge for nok kunnskap om oppgaven, arkitektur, struktur og programmeringsspråk.	Ved manglende ferdigheter til å utføre oppgaven må gruppen sette seg inn i de manglende kompetanseområdene. Eksempelvis kurs og litteratur.
<b>Scope</b> Kartlegging og definisjon av oppgavens størrelse og krav.	10	10	100	Gruppen, Veileder	Fordele oppgaven i "need" og "nice", hvor "need" blir prioritert og fullført først.	Om "need" segmentene er større en først antatt vil vi kommunisere dette ovenfor veileder, før segmentene eventuelt deles inn i ytterlige "need" og "nice".
<b>Eksterne avhengigheter</b> Ressurser vi trenger fra oppdragsgiver og andre eksterne leverandører.	3	10	30	Gruppen, Veileder	Sørge for at Zeipt leverer verktøyene og tilgangene vi trenger, og forholder seg til avtalen definert i kontrakt.	Dersom det skulle oppstå en situasjon med manglende ressurser kontaktes intern veileder for råd.

Eksempelvis at Zeipt ikke leverer verktøy/tilganger man trenger.

<b>Gruppestyring</b> Dårlig kommunikasjon og lite eierskap til prosjektet. Det er viktig at man er flink til å holde en god dialog, og jobbe med oppgaver som er tilpasset for å unngå konflikter i gruppen.	3	10	30	Gruppen, Zeipt	Tydlig kommunikasjon, planlegging og gruppekontrakt.	Fordele arbeid og oppgaver hensiktsmessig. Forsøke å løse eventuelle konflikter internt, og dersom det ikke fungerer, kontaktes intern veileder for råd.
<b>Utstrysproblemer</b> Problemer med utstyr som kan skape unødvendig friksjon ovenfor prosjektet og dets tidsfrister.	3	8	24	Gruppen	Behandle nødvendig utstyr varsomt. Sørg for å ha backup av inn dokumentasjon og kode.	Laste ned backup.
<b>Problemer med intern veileder</b> Holde en kontinuerlig dialog med veileder under prosjektperioden.	1	9	9	Gruppen, Intern veileder	Sørg for god kommunikasjon med veileder under hele prosjektperioden.	Prøve å løse problemer direkte med personen. Om det ikke fungerer, tas det kontakt med andre ansatte på skolen.
<b>Problemer med ekstern veileder</b> Holde en kontinuerlig dialog med veileder under prosjektperioden.	1	9	9	Gruppen, Veileder	Sørg for god kommunikasjon med veileder under hele prosjektperioden.	Prøve å løse problemer direkte med personen. Om det ikke fungerer, tas det kontakt med intern veileder for råd.

## Vedlegg E: Utviklet prototype






- Home >
- Users >
- Receipt templates >
- Progression >

## Custom receipt

To be able to process future payemnts we will need to map your existing receipt format to our own. By uploading a json or XML file you will then be prompted to map existing keys match ours. By uploading a text string, we will try to identify your format, and then you will be able to map keys to ours.


**Upload text string**



**String upload successful**  
You will be redirected shortly.

[Proceed](#)

No file? [Upload text string](#) instead



- Home >
- Users >
- Receipt templates >
- Progression >

## Custom receipt

Step 1  
Upload receipt format

Step 2  
Parse string to JSON

Step 3  
Map keys and values

Step 4  
Done

We've attempted to parse your string to JSON by identifying breaks in the string. If we've gotten it wrong, you can at any time change the identifier to correct the JSON, or edit the JSON directly. It's important that the JSON is recognizable and matches the structure of your string format.

Your string

Identifier

```

1 DETAL3;5756;125727;9980;16;2;29;13;25;4006950031381;4006950031381;1;4
2 DETAL3;5756;125728;9980;16;2;29;13;25;::::::::::30;00;95;:::::10575;::
3 DETAL3;5756;125729;9980;16;2;29;13;25;::::::::::30;00;95;:::::10575;::
4 DETAL3;5756;125730;9980;16;2;29;13;25;::::::::::BAX; 426966-;30;00;95;::
5 DETAL3;5756;125731;9980;16;2;29;13;25;::::::::::29/02/2016 1;30;00;95;::
6 DETAL3;5756;125732;9980;16;2;29;13;25;::::::::::30;00;95;:::::10575;::
7 DETAL3;5756;125733;9980;16;2;29;13;25;::::::::::Mets Senteak;30;00;95;::
8 DETAL3;5756;125734;9980;16;2;29;13;25;::::::::::9570351010;30;00;95;::
9 DETAL3;5756;125735;9980;16;2;29;13;25;::::::::::REF:294 0110;30;00;95;::
10 DETAL3;5756;125736;9980;16;2;29;13;25;::::::::::RESP:00 GODK;30;00;95;::
11 DETAL3;5756;125737;9980;16;2;29;13;25;::::::::::30;00;95;:::::10575;::
12 DETAL3;5756;125738;9980;16;2;29;13;25;::::::::::NOK-;30;00;95;:::::10575;::
13 DETAL3;5756;125739;9980;16;2;29;13;25;::::::::::30;00;95;:::::10575;::
14 DETAL3;5756;125740;9980;16;2;29;13;25;::::::::::GODKJENNES F;30;00;95;::
15 DETAL3;5756;125741;9980;16;2;29;13;25;::::::::::DEBITERING A;30;00;95;::
16 DETAL3;5756;125742;9980;16;2;29;13;25;::::::::::KONTO;30;00;95;:::::10575;::
17 DETAL3;5756;125743;9980;16;2;29;13;25;::::::::::30;00;95;:::::10575;::
18 DETAL3;5756;125744;9980;16;2;29;13;25;::::::::::BAX; 426966-;30;00;10;::
19 DETAL3;5756;125745;9980;16;2;29;13;25;::::::::::29/02/2016 1;30;00;10;::
                    
```

Finalized JSON

```

1 [
2 {
3   "name": "Vita",
4   "vatRate": 0.25,
5   "currency": "NOK",
6   "countryCode": "NO",
7   "merchantInfo": {
8     "merchantName": "Vita Adamtuen",
9     "merchantId": 29753,
10  },
11   "details": [
12     {
13       "key": "RO",
14       "type": "text"
15     },
16     {
17       "key": "PaRes",
18       "type": "text"
19     }
20   ]
21 }
                    
```

[Proceed to mapping](#)

**Zeipt**

- Home >
- Users >
- Receipt templates >
- Progression >

### Custom receipt

Step 1 Upload receipt format    Step 2 Parse string to JSON    Step 3 Map keys and values    Step 4 Done

🔍 Search by field, explanation or type

- Reference
- Customer delivery
- Merchant
- Receipt header
- Total
- Articles
- Payments
- Invoices
- Extra receipt view

**Zeipt**

- Home >
- Users >
- Receipt templates >
- Progression >

### Custom receipt

Step 1 Upload receipt format    Step 2 Parse string to JSON    Step 3 Map keys and values    Step 4 Done

🔍 Search by field, explanation or type

- Reference
- Customer delivery
- Merchant**

FIELD	TYPE	YOUR FIELD
<b>merchant_gcld</b> Your merchant customer client id	string Required	
<b>org_number</b> The organization / VAT identification number of the provider VAT ID	string Required	org_number
<b>merchant_country_code</b> The organization / VAT identification number of the provider VAT ID	string Required	country_code
<b>purchase_country_code</b> The organization / VAT identification number of the provider VAT ID	string Required	purchase_country_code
<b>city</b> The organization / VAT identification number of the provider VAT ID	string Required	city
<b>zip_code</b> The organization / VAT identification number of the provider VAT ID	string Required	zip_code
<b>address</b>	string	address

**Zeipt**

- Home >
- Users >
- Receipt templates >
- Progression >

### Custom receipt

- Step 1 Upload receipt format
- Step 2 Parse string to JSON
- Step 3 Map keys and values
- Step 4 Done

Search by field, explanation or type

Reference

Customer delivery

FIELD	TYPE	YOUR FIELD
<b>merchant_gcld</b> Your merchant customer client id	string Required	
<b>org_number</b> The organization / VAT identification number of the provider VAT ID	string Required	merchant_id
<b>merchant_country_code</b> The organization / VAT identification number of the provider VAT ID	string Required	address
<b>purchase_country_code</b> The organization / VAT identification number of the provider VAT ID	string Required	zip_code
<b>city</b> The organization / VAT identification number of the provider VAT ID	string Required	purchase_country_code
<b>zip_code</b> The organization / VAT identification number of the provider VAT ID	string Required	city
<b>address</b>	string	zip_code
		address

**Zeipt**

- Home >
- Users >
- Receipt templates >
- Progression >

### Custom receipt


- Step 1 Upload receipt format
- Step 2 Parse string to JSON
- Step 3 Map keys and values
- Step 4 Done

Search by field, explanation or type

Reference

Customer delivery

FIELD	TYPE	YOUR FIELD
<b>merchant_gcld</b> Your merchant customer client id	string Required	merchant_id
<b>org_number</b> The organization / VAT identification number of the provider VAT ID	string Required	org_number
<b>merchant_country_code</b> The organization / VAT identification number of the provider VAT ID	string Required	country_code
<b>purchase_country_code</b> The organization / VAT identification number of the provider VAT ID	string Required	purchase_country_code
<b>city</b> The organization / VAT identification number of the provider VAT ID	string Required	city
<b>zip_code</b> The organization / VAT identification number of the provider VAT ID	string Required	zip_code
<b>address</b>	string	address




- Home >
- Users >
- Receipt templates >
- Progression >

### Custom receipt



🔍 Search by id, name or template JSON

id	Name	Template in use
12376	Vita adamstuen	<pre>[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me...</pre>
12377	McDondals	<pre>[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me...</pre>
12378	Burger king	<pre>[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me...</pre>
12379	Joker	<pre>[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me...</pre>



- Home >
- Users >
- Receipt templates >
- Progression >


### Custom receipt



🔍 Search by id, name or template JSON

id	Name	Template in use
12377	McDondals	<pre>[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me...</pre>
12378	Burger king	<pre>[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me...</pre>
12379	Joker	<pre>[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me...</pre>





- Home >
- Users >
- Receipt templates >
- Progression >

### Custom receipt



🔍 Search by id, name or template JSON

id	Name	Template in use
12377	McDondals	[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me... ***
12378	Burger king	[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me... ***
12379	Joker	[{"name": "Vita", "vatRate": 0.25, "currency": "NOK", "countryCode": "NO", "merchantInfo": {"me... ***

Recently deleted

12376	Vita adamstuen		
-------	----------------	---	---





## Vedlegg F: Spørsmål og oppgaver i brukertest

### Oppgave 1: Har du erfaring og/eller er du fagutdannet innenfor teknologi og systemutvikling?

Måleskala på denne oppgaven er Ja/Nei, og målet er å samle data for å se hvordan applikasjonen oppleves like intuitiv for alle, selv om den hviler tungt på språk og terminologi knyttet til fagfeltet. Videre ønsker vi også å kunne vektlegge svar fra målgruppen.

### Oppgave 2: Jobber du med teknologi og systemutvikling, eller har jobbet med teknologi og systemutvikling, innenfor en bedrift som driver salg av varer eller tjenester til privatpersoner?

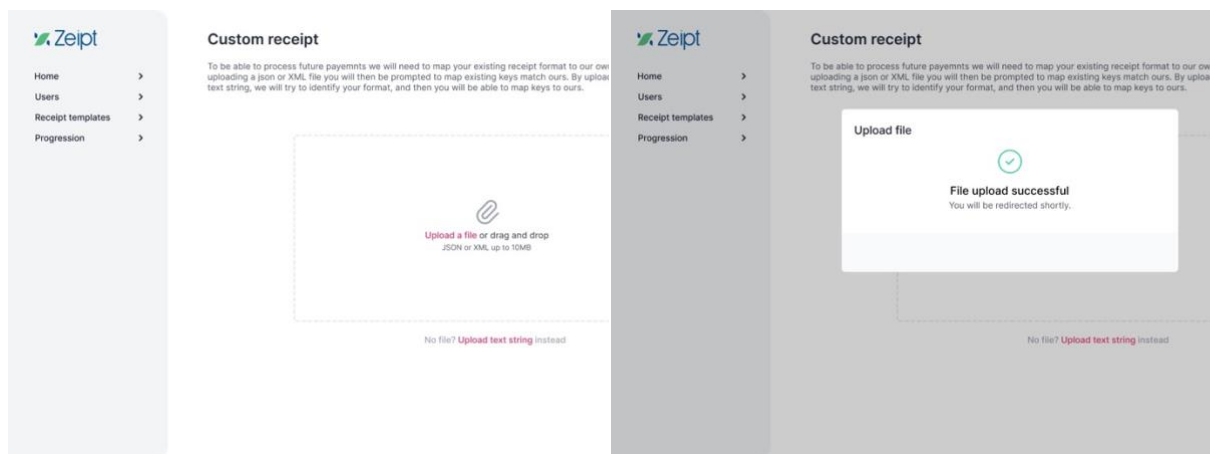
Måleskala på denne oppgaven er Ja/Nei, og målet er å tilspisse data som er enda mer relatert til målgruppen, for så å kunne vektlegge disse svarene.

### Oppgave 3: Er digitale kvitteringer noe som er relevant for deg?

Måleskala på denne oppgaven er Ja/Nei, og målet er å tilspisse data som er enda mer relatert til målgruppen, for så å kunne vektlegge disse svarene.

### Oppgave 4: Du vil mappe ditt kvitteringsformat til Zeipt sitt, last opp formatet som JSON/XML.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hvordan han/hun initialiserer opplastningsflyten når det ikke er noen eksisterende kvitteringsmalere.

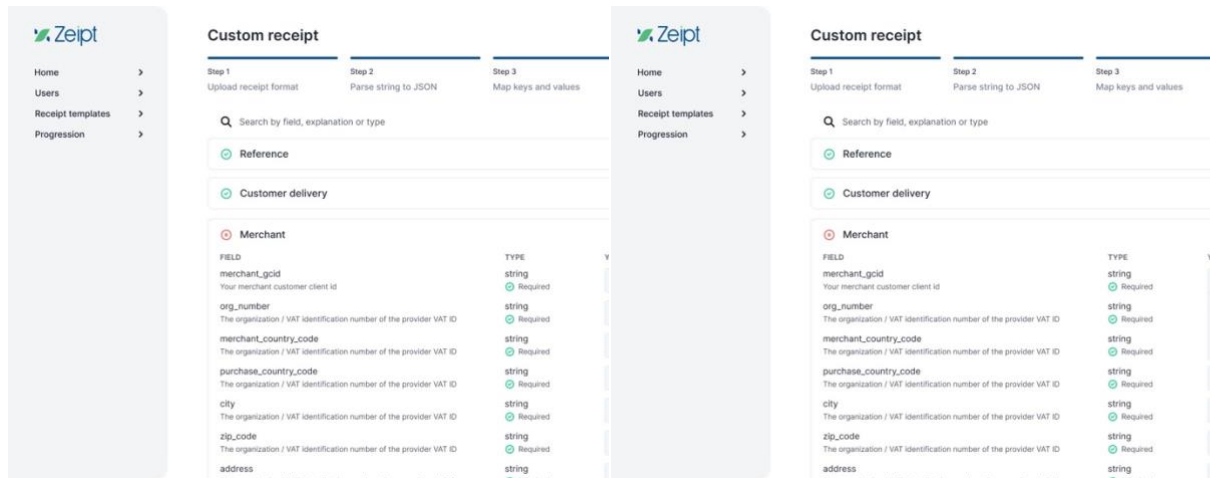


Illustrasjon 5: Prototypen brukt i oppgave 4

På bildet over ser vi testens utgangspunkt hvor man får valget om å enten laste opp en JSON eller XML fil, eller laste opp et annet format, som vi da i løsningen må gjøre om til et format som Zeipt forstår. Her er det tenkt at brukeren i førsteomgang skal laste opp en fil, og det er denne funksjonen som er fremhevet mest, ettersom vi i all hovedsak ønsker data i disse formatene.

### Oppgave 5: Opplastingen var vellykket, men noen nøkler og verdier må mappes manuelt. Finn ut hvilke.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se hvorvidt oppsettet med kartlegging av nøkler er intuitiv nok.

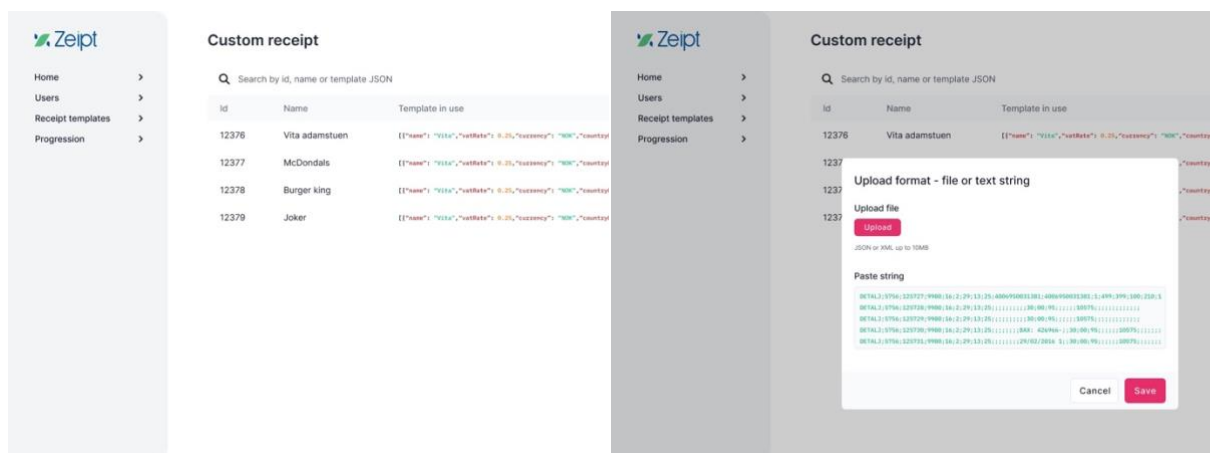


Illustrasjon 6: Prototype brukt i oppgave 5

På bildet over ser vi steget hvor man gjør den faktiske kartleggingen av nøkler og verdier, og er den funksjonen selve oppgaven sentrerer seg rundt. I bildet til høyre ser vi at en rullegardinmeny med verdier fra brukerens eget format er aktiv. Her er det tenkt at brukeren skal legge inn passende verdi fra sitt eget format til Zeipt sine nøkler Videre opplyses brukeren om hvilke datatype feltet er, og om det er pålagt å fylle ut.

### Oppgave 6: Legg til et nytt kvitteringsformat, denne gangen med en text string.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hvordan han/hun initialiserer opplastningsflyten når det allerede er eksisterende kvitteringsmaler.

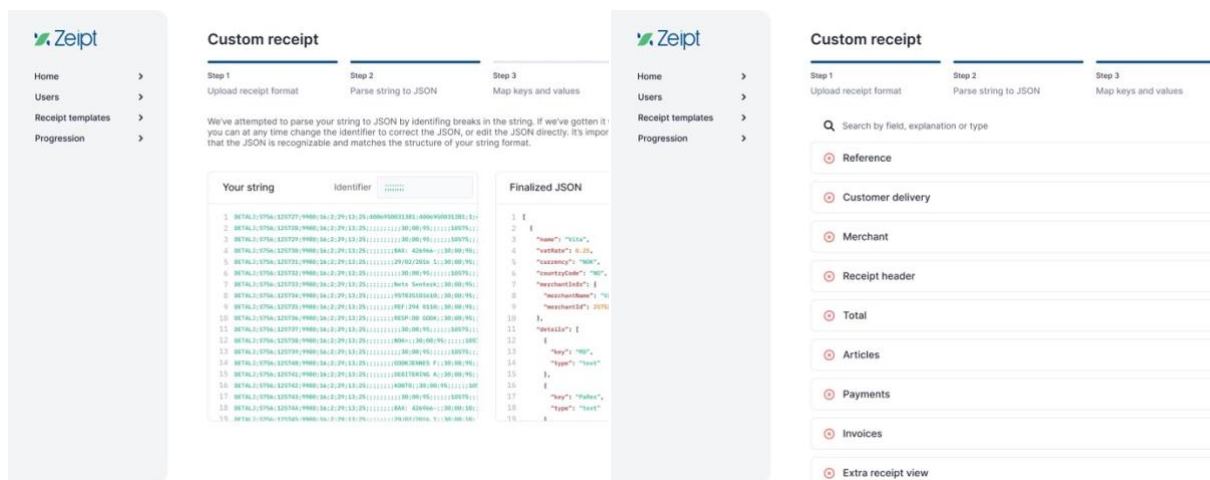


Illustrasjon 7: Prototype brukt i oppgave 6

På bildet over ser vi dialogen som aktiveres dersom man trykker på “Upload text string instead” i bildet vist i oppgave 4. Her laster brukeren opp dataene sine i et annet format enn foretrukket, og vi må da i løsningen gjøre om dette formatet, til et format som Zeipt foretrekker. Her er det tiltenkt at brukeren skal få opp dialogen, lime inn dataene sine i tekstfeltet, og deretter trykke “Save”.

### Oppgave 7: Text stringen ble suksessfullt parset til JSON, fortsett til mapping av nøkler og verdier.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hva som må gjøres dersom systemet ikke automatisk klarer å kartlegge en eller flere verdier til Zeipt sine nøkler.

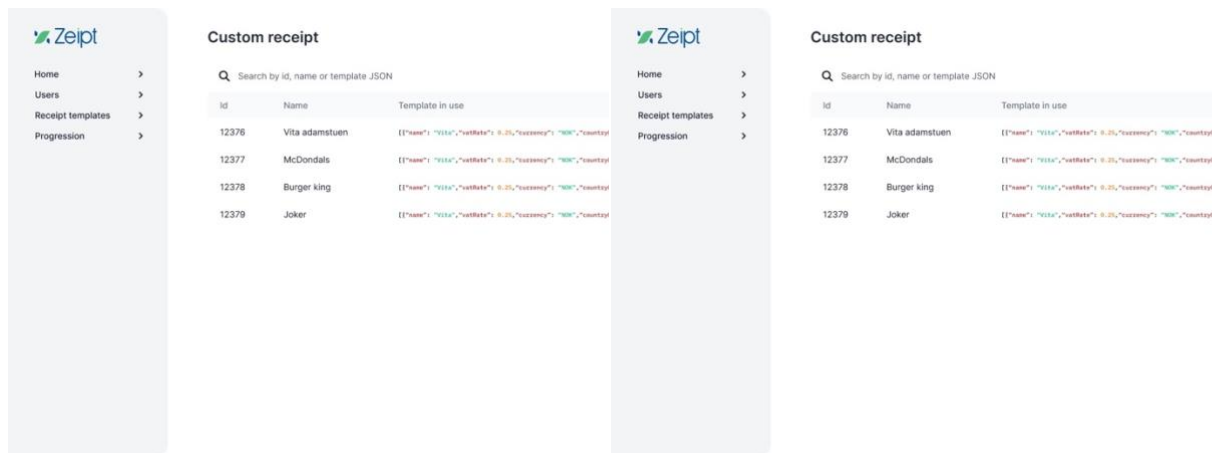


Illustrasjon 8: Prototype brukt i oppgave 7

På bildene over ser vi steget hvor brukeren ser sitt omgjorte resultat, og får beskjed om å dobbeltsjekke at strukturen og dataene ser logisk ut. Ved å fortsette fra dette steget vil brukeren bli tatt videre til kartleggingssteget hvor de kartlegger sine egne nøkler opp mot Zeipt sine.

### Oppgave 8: Slett kvitteringen fra Vita Adamstuen.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hvordan eksisterende kvitteringsmaler kan slettes.

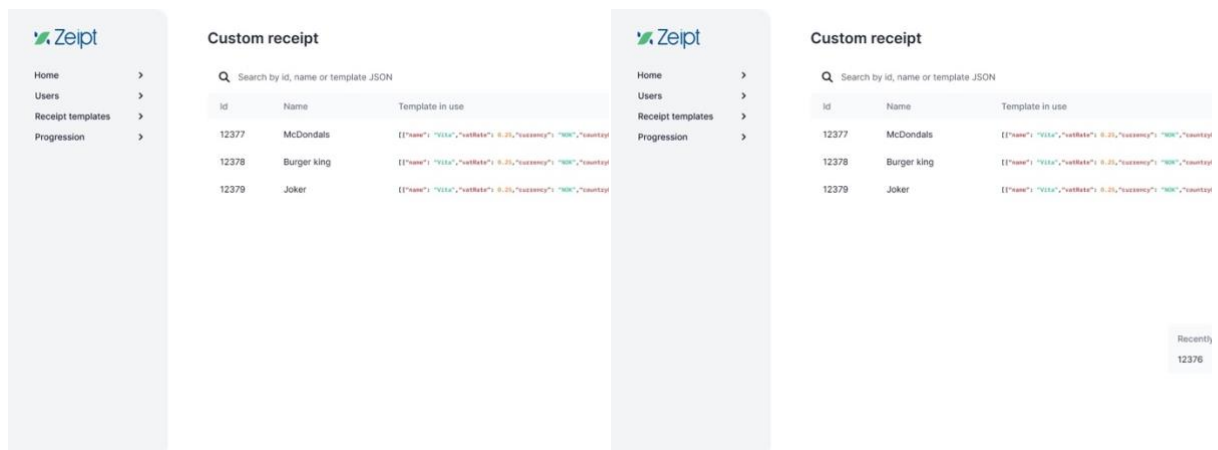


Illustrasjon 9: Prototypen brukt i oppgave 8

På bildene over ser vi en situasjon hvor det finnes aktive kvitteringsmalere fra før av. Her får brukeren beskjed om at kvitteringsmalen fra Vita Adamstuen skal slettes. Tanken er da at de skal klikke på de tre horisontale prikkene for å få opp ytterligere alternativer. I bildet til høyre ser vi at en meny med handlinger er aktiv.

### Oppgave 9: Du angret på å ha slettet kvitteringen fra Vita Adamstuen, finn en måte å legge den tilbake igjen.

Måleskala for denne oppgaven er oppdragsbasert for å se hvordan brukere interagerer med prototypen, og målet er å se om deltakeren skjønner hvordan nylig slettede kvitteringsmalere kan gjenopprettes.



Illustrasjon 10: Prototypen brukt i oppgave 9

På bildene over ser vi en situasjon hvor en kvitteringsmal nylig er slettet, og brukeren får beskjed om å gjenopprette den. Helt nederst til høyre har det dukket opp et papirkurv ikon, og tanken er at brukeren skal klikke på denne for å få opp en meny med ytterligere alternativer. I bildet til høyre ser vi denne menyen aktiv, og tanken er da at brukeren skal trykke på det grønne "pluss"-ikonet.

### Oppgave 10: Har du noen tilbakemeldinger?

Måleskalaen brukt i oppgaven er åpent spørsmål, og skal la deltakeren fritt komme med tilbakemeldinger og feedback og tjenesten og prosessen av å bli brukertestet. Målet er å få ærlige, konstruktive, tilbakemeldinger.

### Vedlegg G: Resultat og analyse fra ekstern brukertest

#### Resultat og analyse

##### Oppgave 1:

60% av deltakerne hadde erfaring og/eller var fagutdannet innenfor teknologi og systemutvikling. Dette betyr at 60% er innenfor målgruppen, og disse svarene kommer derfor til å vektas tyngre enn svarene til de som ikke er det.

##### Oppgave 2:

40% av deltakerne jobbet med teknologi/systemutvikling innenfor en bedrift som driver salg av varer og tjenester til privatpersoner. Av disse 40% var det kun halvparten (1/5) som hadde erfaring og/eller var fagutdannet innenfor teknologi og systemutvikling. Det betyr at svarene til denne deltakeren vil vektas tyngre enn de resterende 4/5.

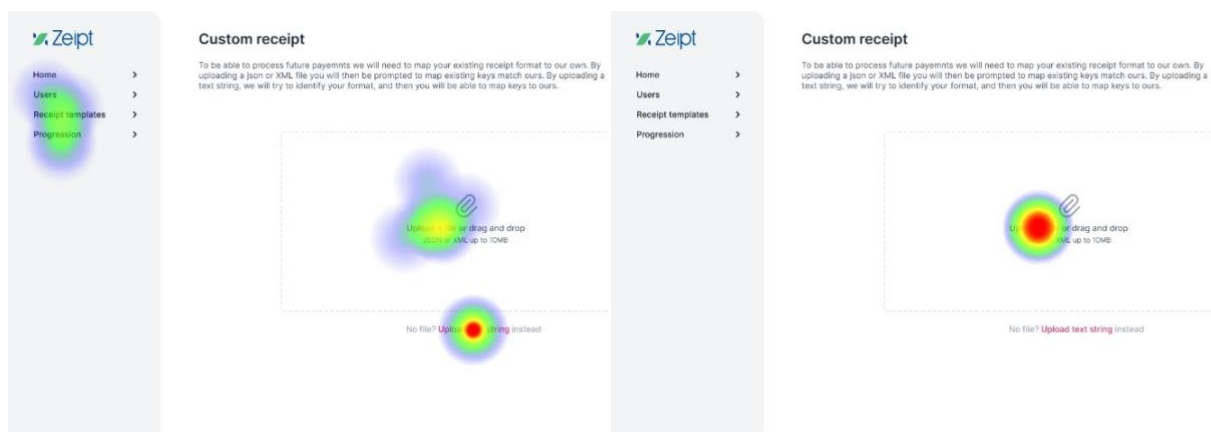
##### Oppgave 3:

Digitale kvitteringer var relevant for 100% av deltakerne.

##### Oppgave 4:

Deltakerne brukte i snitt 21,9 sekunder på oppgaven, og hadde en misklikkrate på 40%. Alle deltakerne løste oppgaven.

Målet med oppgaven var å se om deltakerne skjønner hvordan han/hun initialiserer opplastningsflyten når det ikke er noen eksisterende kvitteringsmaler. Vi ser at deltakerne som ikke er i målgruppen prøvde å klikke flere steder før de klarte å løse oppgaven, mens deltakerne i målgruppen skjønnte hvor de skulle klikke med en gang, og løste oppgaven uten å klikke feil. Dette illustreres ved følgende heatmaps:



*Illustrasjon A: Oppgave 4 - Klikk fra deltakere i målgruppe* *Illustrasjon B: Oppgave 4 - Klikk fra deltakere utenfor målgruppe*

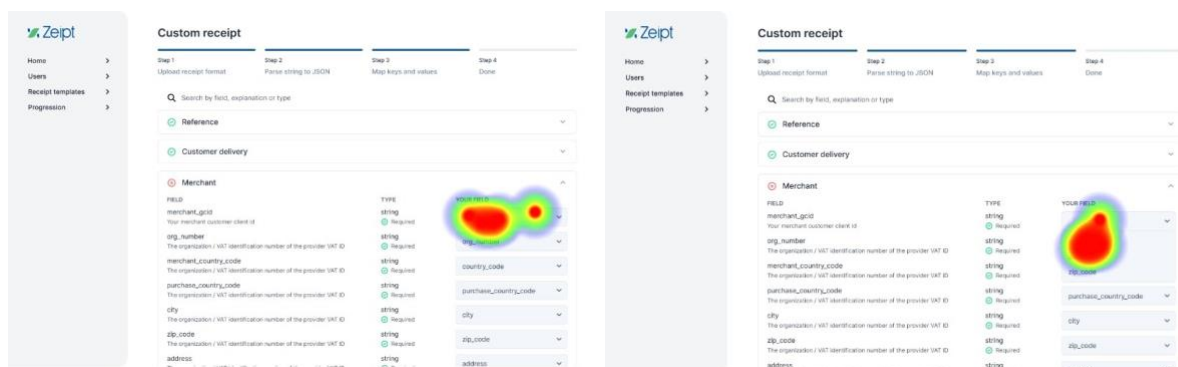
Vi kan derfor konkludere med at flyten er tilstrekkelig, så fremt man er kjent med terminologi og konvensjoner knyttet til faget.

For deltakerne som ikke er i målgruppen kan vi konstatere at problemene bunner i designprinsippene synlighet, sammenheng og hint. Vi definerer dette problemet i kategorien medium. Dette er fordi det gjelder en kritisk prosess, problemet er umulig å komme utenom, men det gjelder ikke alle deltakerne, men vi vektlegger at målgruppen løser oppgaven som tiltenkt.

### Oppgave 5:

Deltakerne brukte i snitt 6,1 sekunder på oppgaven, og hadde i snitt en misklikkrate på 10%. Alle deltakerne løste oppgaven.

Målet med oppgaven var å se hvorvidt oppsettet med kartlegging av nøkler er intuitiv nok. Vi ser at alle deltakerne klikket der de skulle. Dette illustreres ved følgende heatmap:



*Illustrasjon C: Oppgave 5 - Klikk fra alle deltakere*

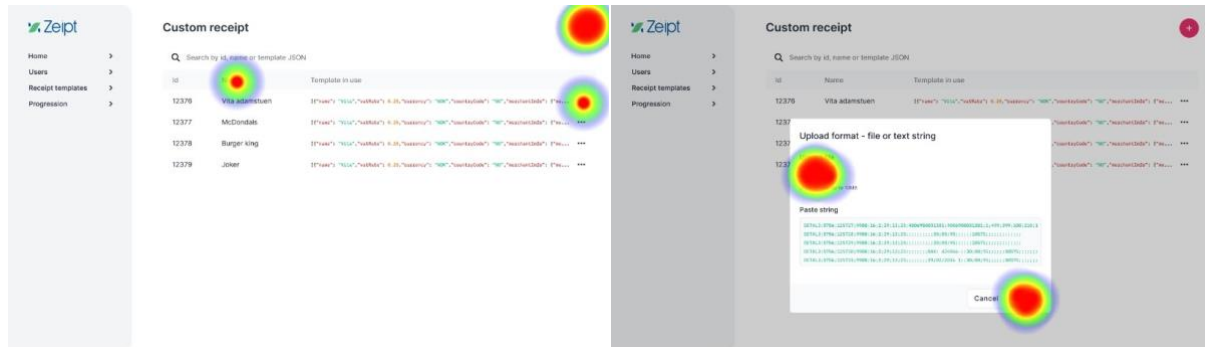
Vi kan derfor konkludere med at oppsettet med kartlegging av nøkler er intuitiv nok.

### Oppgave 6:

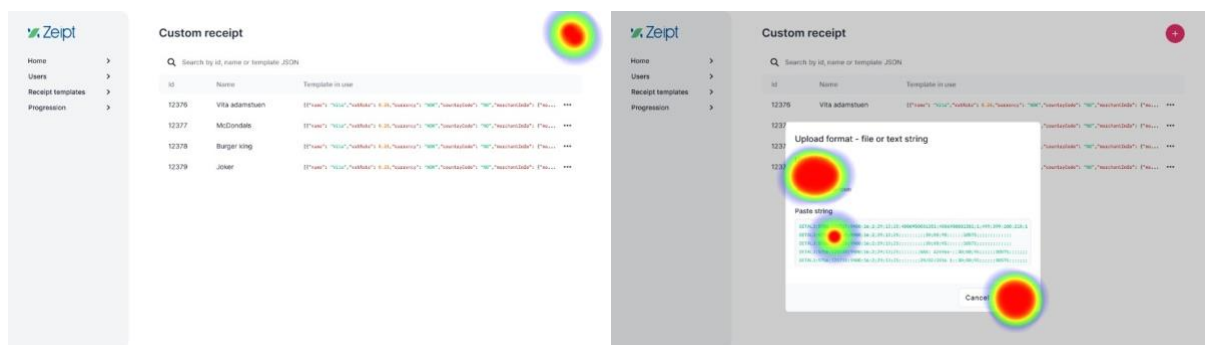
Deltakerne brukte i snitt 5,4 sekunder på oppgaven, og hadde i snitt en misklikkrate på 50%. Alle deltakerne løste oppgaven til slutt.

Målet med oppgaven var å se om deltakeren skjønner hvordan han/hun initialiserer opplastningsflyten når det allerede er eksisterende kvitteringsmaler. Vi ser igjen at deltakerne som ikke er i målgruppen prøvde å klikke flere steder for de klarte å løse oppgaven, og at deltakerne som er i målgruppen skjønte hvor de skulle klikke med en gang. Det kan se ut som deltakerne som ikke var i målgruppen ble forvirret av terminologi, både i

oppgaveteksten og i elementer på siden. Dette fordi vi ser at de prøvde å klikke på forskjellige steder hvor terminologi var involvert. Vi kan derfor konkludere med at flyten er tilstrekkelig, så fremt man er kjent med terminologi og konvensjoner knyttet til faget. Dette illustreres ved følgende heatmaps:



Illustrasjon D: Oppgave 6 - Klikk fra deltakere utenfor målgruppe



Illustrasjon E: Oppgave 6 - Klikk fra deltakere i målgruppen

For deltakerne som ikke er i målgruppen kan vi, igjen, konstatere at problemene bunner i designprinsippene synlighet, sammenheng og hint. Vi definerer dette problemet i kategorien alvorlig. Dette er fordi det gjelder en kritisk prosess, problemet er umulig å komme utenom, men det gjelder ikke alle deltakerne. Når det er sagt vektlegger vi at målgruppen løser oppgaven som tiltenkt.

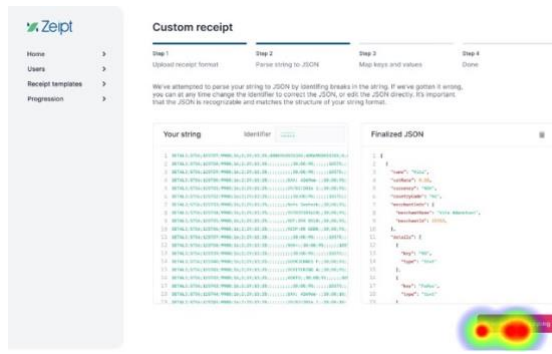
### Oppgave 7:

Deltakerne brukte i snitt 6,2 sekunder på oppgaven, og hadde i snitt en misklikkrate på 40%. Alle deltakerne løste oppgaven.

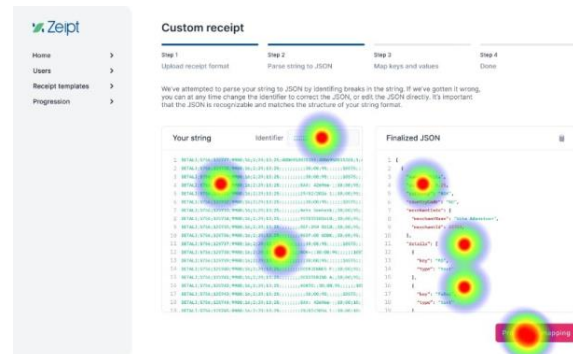
Målet med oppgaven var å se om deltakeren skjønner hva som må gjøres dersom systemet ikke automatisk klarer å kartlegge en eller flere verdier til Zeipt sine nøkler. Her ser vi faktisk at deltakerne som ikke var i målgruppen skjønnte med en gang hvordan de skulle gå videre for å løse oppgaven, mens deltakerne i målgruppen prøvde å klikke på andre elementer.



Dette kan ha noe med at de som hadde kunnskap tilknyttet fagområdet prøvde å undersøke at alt var riktig før de fortsatte, mens se som ikke hadde kunnskap tilknyttet bare fortsatte. Vi kan derfor konkludere med at flyten ikke er tilstrekkelig, og at vi må videreutvikle dette steget, ettersom deler av målgruppen hadde problemer. Dette illustreres ved følgende heatmaps:



Illustrasjon F: Oppgave 7 - Klikk fra deltakere utenfor målgruppen



Illustrasjon G: Oppgave 7 - Klikk fra deltakere i målgruppen

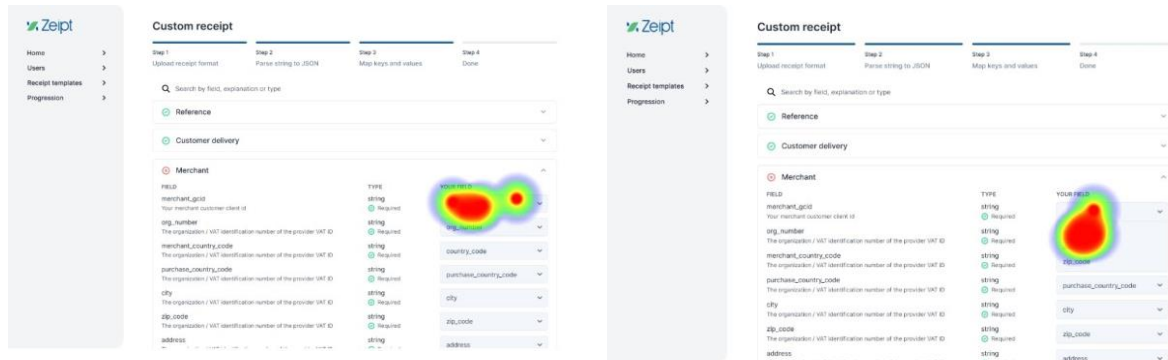
Vii kan konstatere at problemene bunner i designprinsippene synlighet, sammenheng og hint. Vi definerer dette problemet i kategorien alvorlig. Dette er fordi det gjelder en prosess som er kritisk, problemet er mulig å komme utenom, men det gjelder ikke alle deltakerne.

### Oppgave 8:

Deltakerne brukte i snitt 2,4 sekunder på oppgaven, og hadde i snitt en misklikkrate på 0%. Alle deltakerne løste oppgaven.

Målet med oppgaven var å se om deltakeren skjønner hvordan eksisterende kvitteringsmaler kan slettes. Vi ser at 4/5 deltakere klikket der de skulle med en gang, mens 1/5 klikket på feil butikk i oversikten. Dette vektlegges ikke betraktelig, da vi ser at disse klikkene skjedde i løpet av det første sekundet deltakeren fikk tilgang til oppgaven, og kan derfor tyde på slurv og/eller at deltakeren ikke tok seg lang nok tid til å tilegne seg informasjonen på siden. Vi tar også i betraktning at selv da klikker deltakeren på en knapp med tilsvarende funksjon, men tilknyttet feil rad. Vi kan derfor konkludere med at deltakerne skjønte hvordan eksisterende kvitteringsmaler kan slettes. Dette illustreres ved følgende heatmaps:





Illustrasjon H: Oppgave 8 - Klikk fra alle deltakere

### Oppgave 9:

Deltakerne brukte i snitt 1,4 sekunder på oppgaven, og hadde i snitt 0% misklikkrate. Alle deltakerne løste oppgaven.

Målet med oppgaven var å se om deltakeren skjønner hvordan nylig slettede kvitteringsmaler kan gjenopprettes. Vi ser at 100% av deltakerne umiddelbart klikket der de skulle. Vi kan derfor konkludere med at det er intuitivt hvordan nylige slettede kvitteringsmaler kan gjenopprettes.

### Oppgave 10:

Tilbakemeldingene fra deltakerne oppsummert var designet var lett, intuitivt og brukervennlig. Samtidig reagerte noen av deltakerne på at det var terminologi i både spørsmål og oppgave som de ikke forstod, og 1/5 deltakere synes at spørsmålene kunne vært mer tydelige.

**Individuelle svar**

Svar fra deltaker 34370590


**6.0** Har du erfaring og/eller er du fagutdannet innenfor teknologi og systemutvikling?  
Yes/No  
Yes

**6.0** Jobber du med teknologi og systemutvikling, eller har jobbet med teknologi og systemutvikling, innenfor en bedrift som driver salg av varer eller tjenester til privatpersoner?  
Yes/No  
Yes

**6.0** Er digitale kvitteringer noe som er relevant for deg?  
Yes/No  
Yes


**6.0** Du vil mappe ditt kvitteringsformat til Zeipt sitt, last opp formatet som JSON/XML.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
10.8 seconds  
MISCLICKS  
0




**6.0** Oppplastningen var vellykket, men noen nøkler og verdier må mappes manuelt. Finn ut hvilke.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
6.2 seconds  
MISCLICKS  
0




**6.0** Legg til et nytt kvitteringsformat, denne gangen med en text string.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
5.5 seconds  
MISCLICKS  
0




**6.0** Text stringen ble suksessfullt parsert til JSON, fortsett til mapping av nøkler og verdier.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
9.6 seconds  
MISCLICKS  
3




**6.0** Slett kvitteringen fra Vita Adamstuen  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
3.9 seconds  
MISCLICKS  
0



**6.0** Du angrep på å ha slettet kvitteringen fra Vita Adamstuen, finn en måte å legge den tilbake igjen.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
3.1 seconds  
MISCLICKS  
0



**6.0** Har du noen tilbakemeldinger?  
Open Question  
Lett og intuitivt. God flyt.

## Svar fra deltaker 34385140


**6.0** Har du erfaring og/eller er du fagutdannet innenfor teknologi og systemutvikling?  
Yes/No  
Yes

**6.0** Jobber du med teknologi og systemutvikling, eller har jobbet med teknologi og systemutvikling, innenfor en bedrift som driver salg av varer eller tjenester til privatpersoner?  
Yes/No  
Yes

**6.0** Er digitale kvitteringer noe som er relevant for deg?  
Yes/No  
Yes

**6.0** Du vil mappe ditt kvitteringsformat til Zeipt sitt, last opp formatet som JSON/XML.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
3.6 seconds  
MISCLICKS  
0



**6.0** Opplastningen var vellykket, men noen nøkler og verdier må mappes manuelt. Finn ut hvilke.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
3.4 seconds  
MISCLICKS  
0



**6.0** Legg til et nytt kvitteringsformat, denne gangen med en text string.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
4.2 seconds  
MISCLICKS  
0



**6.0** Text stringen ble suksessfullt parsert til JSON, fortsett til mapping av nøkler og verdier.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
6.2 seconds  
MISCLICKS  
3



**6.0** Slett kvitteringen fra Vita Adamstuen  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
3.0 seconds  
MISCLICKS  
0



**6.0** Du angrep på å ha slettet kvitteringen fra Vita Adamstuen, finn en måte å legge den tilbake igjen.  
Mission


OUTCOME  
• Direct  
AVG. DURATION  
2.4 seconds  
MISCLICKS  
0



**6.0** Har du noen tilbakemeldinger?  
Open Question


Intuitivt. God flyt.

## Svar fra deltaker 34334346

 Har du erfaring og/eller er du fagutdannet innenfor teknologi og systemutvikling?


Yes/No

No

 Jobber du med teknologi og systemutvikling, eller har jobbet med teknologi og systemutvikling, innenfor en bedrift som driver salg av varer eller tjenester til privatpersoner?


Yes/No

No

 Er digitale kvitteringer noe som er relevant for deg?

Yes/No

Yes

 Du vil mappe ditt kvitteringsformat til Zeipt sitt, last opp formatet som JSON/XML.

Mission

OUTCOME

• Direct

AVG. DURATION

46.6 seconds

MISCLICKS

7



 Oppplastningen var vellykket, men noen nøkler og verdier må mappes manuelt. Finn ut hvilke.

Mission

OUTCOME

• Direct

AVG. DURATION

8.0 seconds

MISCLICKS

0



 Legg til et nytt kvitteringsformat, denne gangen med en text string.

Mission

OUTCOME

• Direct

AVG. DURATION

17.9 seconds

MISCLICKS

2



 Text stringen ble suksessfullt parset til JSON, fortsett til mapping av nøkler og verdier.

Mission

OUTCOME

• Direct

AVG. DURATION

3.0 seconds

MISCLICKS

0



 Slett kvitteringen fra Vita Adamstuen

Mission

OUTCOME

• Direct

AVG. DURATION

10.0 seconds

MISCLICKS

0



 Du angrep på å ha slettet kvitteringen fra Vita Adamstuen, finn en måte å legge den tilbake igjen.

Mission

OUTCOME

• Direct


AVG. DURATION

2.9 seconds

MISCLICKS

0



 Har du noen tilbakemeldinger?

Open Question

Alt gikk greit for meg, men vanskelig å forstå når jeg ikke skjønner halvparten av hva det er snakk om. Det var lett å forstå hva jeg skulle trykke på og hva de forskjellige knappene kom til å gjøre


## Svar fra deltaker 34326779

**Har du erfaring og/eller er du fagutdannet innenfor teknologi og systemutvikling?**  
Yes/No  
Yes


**Jobber du med teknologi og systemutvikling, eller har jobbet med teknologi og systemutvikling, innenfor en bedrift som driver salg av varer eller tjenester til privatpersoner?**  
Yes/No  
No

**Er digitale kvitteringer noe som er relevant for deg?**  
Yes/No  
Yes


**Du vil mappe ditt kvitteringsformat til Zeipt sitt, last opp formatet som JSON/XML.**  
Mission  
OUTCOME  
• Direct  
AVG. DURATION  
40.7 seconds  
MISCLICKS  
6




**Opplætningen var vellykket, men noen nøkler og verdier må mappes manuelt. Finn ut hvilke.**  
Mission  
OUTCOME  
• Direct  
AVG. DURATION  
25.1 seconds  
MISCLICKS  
1




**Legg til et nytt kvitteringsformat, denne gangen med en text string.**  
Mission  
OUTCOME  
• Direct  
AVG. DURATION  
8.8 seconds  
MISCLICKS  
1




**Text stringen ble suksessfullt parsert til JSON, fortsett til mapping av nøkler og verdier.**  
Mission  
OUTCOME  
• Direct  
AVG. DURATION  
6.6 seconds  
MISCLICKS  
0



**Slett kvitteringen fra Vita Adamstuen**  
Mission  
OUTCOME  
• Direct  
AVG. DURATION  
3.5 seconds  
MISCLICKS  
0



**Du angrer på å ha slettet kvitteringen fra Vita Adamstuen, finn en måte å legge den tilbake igjen.**  
Mission  
OUTCOME  
• Direct  
AVG. DURATION  
2.4 seconds  
MISCLICKS  
0



**Har du noen tilbakemeldinger?**  
Open Question  
Fine oppgaver, og design man kjenner seg igjen i!

## Svar fra deltaker 34326459


**6b** Har du erfaring og/eller er du fagutdannet innenfor teknologi og systemutvikling?  
Yes/No  
No

**6b** Jobber du med teknologi og systemutvikling, eller har jobbet med teknologi og systemutvikling, innenfor en bedrift som driver salg av varer eller tjenester til privatpersoner?  
Yes/No  
No

**6b** Er digitale kvitteringer noe som er relevant for deg?  
Yes/No  
Yes


**6a** Du vil mappe ditt kvitteringsformat til Zeipt sitt, last opp formatet som JSON/XML.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
7.8 seconds  
MISCLICKS  
0




**6a** Oppplantningen var vellykket, men noen nøkler og verdier må mappes manuelt. Finn ut hvilke.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
18.1 seconds  
MISCLICKS  
0




**6a** Legg til et nytt kvitteringsformat, denne gangen med en text string.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
11.3 seconds  
MISCLICKS  
4




**6a** Text stringen ble suksessfullt parset til JSON, fortsett til mapping av nøkler og verdier.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
5.4 seconds  
MISCLICKS  
0




**6a** Slett kvitteringen fra Vita Adamstuen  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
3.5 seconds  
MISCLICKS  
0



**6a** Du angrep på å ha slettet kvitteringen fra Vita Adamstuen, finn en måte å legge den tilbake igjen.  
Mission

OUTCOME  
• Direct  
AVG. DURATION  
3.0 seconds  
MISCLICKS  
0



**6c** Har du noen tilbakemeldinger?  
Open Question  
Veldig rent og tydelig hva man skal trykke på :)

## Vedlegg H: Utdrag fra Git

Commit	User	Timestamp	Message	Changes
f0f689c	danielkjellid	Tue May 11 15:59:57 2021	fixed chevrons in mapping step	1 file changed, 9 insertions(+), 10 deletions(-)
a3b478f	Ludvik	Tue May 11 00:18:20 2021	dropdown fields styling	2 files changed, 4 insertions(+), 6 deletions(-)
b97755f	danielkjellid	Mon May 10 23:30:54 2021	fixed updating of value in nested objects when updating jsonReceiptData	2 files changed, 40 insertions(+), 1 deletion(-)
21cd3ae	danielkjellid	Mon May 10 23:02:39 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client- dashboard into receipt-templates	
cf8efa3	danielkjellid	Mon May 10 23:02:25 2021	added support for automatically identify keys in uploaded json to use as options when mapping	3 files changed, 60 insertions(+), 12 deletions(-)
45088a6	Ludvik	Mon May 10 21:35:14 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client- dashboard into receipt-templates	
1abdd7a	Ludvik	Mon May 10 21:35:07 2021	field select styling	2 files changed, 7 insertions(+), 19 deletions(-)
626d31c	danielkjellid	Mon May 10 18:17:47 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client- dashboard into receipt-templates	
5994486	danielkjellid	Mon May 10 18:17:23 2021	fixed formatting in Parser step	1 file changed, 1 insertion(+), 1 deletion(-)
0f11c66	Ludvik	Mon May 10 18:15:58 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client- dashboard into receipt-templates	
19809fd	Ludvik	Mon May 10 18:15:51 2021	console error fixes	3 files changed, 9 insertions(+), 11 deletions(-)
0525f4f	danielkjellid	Mon May 10 18:14:55 2021	fixed redirect after fileupload	2 files changed, 2 insertions(+)
1abf3ec	danielkjellid	Mon May 10 17:54:32 2021	fixed type check in field component	1 file changed, 1 insertion(+), 1 deletion(-)
c770214	danielkjellid	Mon May 10 17:52:56 2021	fixed case where renderOptions did not break down arrays containing objects	1 file changed, 13 insertions(+), 4 deletions(-)
d6f225f	Ludvik	Mon May 10 17:40:24 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client- dashboard into receipt-templates	
c8eb2ff	Ludvik	Mon May 10 17:40:11 2021	map values function added	3 files changed, 36 insertions(+), 129 deletions(-)
bc6b629	danielkjellid	Mon May 10 17:01:29 2021	renamed steps	1 file changed, 2 insertions(+), 2 deletions(-)
3b417fd	danielkjellid	Mon May 10 16:58:05 2021	added check and a way to proceed in Parser	3 files changed, 60 insertions(+), 29 deletions(-)
2efc568	Ludvik	Mon May 10 16:28:49 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client- dashboard into receipt-templates	
d09693e	Ludvik	Mon May 10 16:28:40 2021	nav option for receipt templates added, fileuploader fix	6 files changed, 174 insertions(+), 151 deletions(-)
e5296c2	danielkjellid	Mon May 10 16:27:22 2021	fixed rendering of options in mapping step	2 files changed, 28 insertions(+), 4 deletions(-)
e9cc0e6	Ludvik	Mon May 10 15:24:21 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client- dashboard into receipt-templates	
3ed2aac	Ludvik	Mon May 10 15:24:12 2021	jsonData added, mapping started	3 files changed, 22 insertions(+), 18 deletions(-)
6bf6729	danielkjellid	Mon May 10 15:23:11 2021	hooked parser into redux state	5 files changed, 24 insertions(+), 17 deletions(-)

62da398	danielkjellid	Sun May 9 22:10:41 2021	code cleanup	5 files changed, 5 insertions(+), 7 deletions(-)
a95661d	Ludvik	Sun May 9 22:04:24 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client-dashboard into receipt-templates	
d2f66d3	Ludvik	Sun May 9 22:04:16 2021	FileUploader connected to redux, Mapping.js view with components Dropdown.js and Field.js	14 files changed, 919 insertions(+), 61 deletions(-)
ceed426	danielkjellid	Sun May 9 21:59:58 2021	fixed error handling on parser	3 files changed, 25 insertions(+), 8 deletions(-)
9b1550c	danielkjellid	Sat May 8 21:16:11 2021	added local state to parsing action, changed props in ParsingStatusBar component	3 files changed, 68 insertions(+), 29 deletions(-)
7a3b0de	danielkjellid	Sat May 8 20:31:37 2021	parser now handles cases where there is a longer string in an object	1 file changed, 7 insertions(+), 8 deletions(-)
5e2480b	danielkjellid	Sat May 8 19:01:45 2021	modularized code used in parser	1 file changed, 76 insertions(+), 71 deletions(-)
ab7559d	danielkjellid	Sat May 8 18:06:58 2021	chore, commenting	1 file changed, 17 insertions(+), 14 deletions(-)
df722e5	danielkjellid	Sat May 8 18:00:31 2021	handle user input when identified break is wrong is now working, even with async changes to useState	3 files changed, 41 insertions(+), 47 deletions(-)
672e3c2	danielkjellid	Sat May 8 16:42:06 2021	changed parse to type, cheking string instead of boolean	1 file changed, 1 insertion(+), 1 deletion(-)
1e73457	danielkjellid	Thu May 6 23:38:53 2021	Added support for user-input when identifying breaks, as well as modifications to the Modal styles.	5 files changed, 217 insertions(+), 39 deletions(-)
8997935	Ludvik	Mon May 3 23:21:58 2021	fileuploader cleanup	2 files changed, 22 insertions(+), 5 deletions(-)
7bb3b2d	Ludvik	Wed Apr 28 16:03:03 2021	merge	
43bd2b3	Ludvik	Wed Apr 28 16:00:59 2021	File uploader component functional	10 files changed, 245 insertions(+), 88 deletions(-)
5945d42	danielkjellid	Mon Apr 26 21:35:54 2021	added correct parsing to type in parser	1 file changed, 29 insertions(+), 7 deletions(-)
49ced45	danielkjellid	Mon Apr 26 17:26:20 2021	added code for identifying breaks and structuring string to object on uploaded string	3 files changed, 153 insertions(+), 6 deletions(-)
a1a83a5	Ludvik Blunck	Thu Mar 11 20:40:55 2021	merge	
a906f59	Ludvik Blunck	Thu Mar 11 20:33:25 2021	merge fix	8 files changed, 114 insertions(+), 6 deletions(-)
be12f9c	Ludvik Blunck	Thu Mar 11 20:28:16 2021	modal fix and HasTemplates page added	10 files changed, 127 insertions(+), 9 deletions(-)
aa9acce	danielkjellid	Thu Mar 11 16:15:10 2021	started working on editor panels.	8 files changed, 114 insertions(+), 6 deletions(-)
5ff9ceb	Ludvik	Mon Mar 8 12:51:12 2021	Merge branch 'receipt-templates' of gitlab.com:zeipt/zeipt-frontend/web/client-dashboard into receipt-templates	
4483170	Ludvik	Mon Mar 8 12:50:40 2021	modal styling	3 files changed, 308 insertions(+), 285 deletions(-)
f743853	danielkjellid	Thu Mar 4 16:16:43 2021	added step counter component	4 files changed, 123 insertions(+), 2 deletions(-)
7a15f98	danielkjellid	Thu Mar 4 13:53:52 2021	removed unused code, fixed routes	7 files changed, 5 insertions(+), 190 deletions(-)
2a26387	danielkjellid	Thu Mar 4 13:48:39 2021	structure CustomReceipt folder in pages	9 files changed, 238 insertions(+)



7ea4731	Ludvik	Thu Feb 25 17:47:17 2021	Added NoTemplates version of landing page, and started on modal	10 files changed, 405 insertions(+), 121 deletions(-)
---------	--------	-----------------------------	---	---

## **Vedlegg I: Oppdragsgivers vurdering**

### **Innledning:**

Daniel Kjellid og Ludvik Blunck gjennomførte under våren 2021 sitt bachelorprosjekt hos oss på Zeipt, der den antagna oppgiften handla om å underletta for framtidige kunder å integrere seg med Zeipt sitt system og dermed senke inngangsbarriæren for å begynne konsumere Zeipt sine tjenester.

Zeipt er en B2B leverantør av digitale kvitton for kassaleverantører og har under en lang tid identifisert at kunder i visse tilfeller har lite for mye arbeid med å integrere seg med Zeipt sitt system. Oppgiften gikk dermed ut på å videreutvikle vårt dashboard med å skape et nytt webbasert interface i JavaScript som tilbyr kunden et GUI for mapping av eget output format av kvittot til Zeipt's input format, der den grafiske mappingen deretter blir oversatt til "instruksjoner" som blir lagret på serversiden. Disse "instruksjoner" kjøres deretter ved request av kunden for å kunne mappe deres format i realtid online og dermed minimere mengden av eget arbeid på sin klientside.

### **Utførelse:**

Studentene hadde innledende tilgang til eksisterende kodebase som gav en viss form av forståelse av designprofil og kodestruktur å bygge videre på. Det lå i studentenes egne ansvar å finne ut hvordan de ville strukturere selv arbeidet samt legge fram egne forslag på hvordan de såg for seg at brukarresan skulle utformes.

Studentene har under arbeidets gang valgt å følge "design first" metodikk der de på kontinuerlig basis viser fram design forslag til en åpen dialog der feedback har kunnet gis intuitivt og beslutningsgrunnlag kunnet formidles raskt og effektivt for videre progression i arbeidet. De har også introdusert egne tekniske løsninger på hvordan "free text" input fra klienten har kunnet leses inn og automatisk parses inn i et strukturert data format (json) der dette deretter er tagget brukarresan videre til et ferdig instruksjonssett for plattformen å kunne agere på.

### **Resultat:**

Målsætning med prosjektet var å skape en funksjonalitet som vi har som avsikt å lansere, så vi i Zeipt hadde satt høye krav på skapandet av brukarvenlighet, følgende av designprofil samt kvalitet, der dette skal kunne kombineres med allerede eksisterende kode til foretaket, noe som vi avser at studentene godt har oppnådd.

Det merkes også at studentene ikke bare har fulgt de spesifikasjonene som ble framstilt dem, utan at de også har reflektert godt over hvordan opplevelsen fra brukers perspektiv skulle kunne se ut, og at de korrigerer sin tilnærming under prosjektets gang for å bibeholde fokus på "smartness" for kunden.

Valet att använda sig av en "design first" metodik vid utformande och exekverande av projektet tyder också på en mogenhet i sin tankegång med att effektivt inkludera oss tidigt (och löpande) i utformningen av projektet. Som då i retur gav en starkare tillnärmning till existerande designprofil (Som viktas högt vid en ux upplevelse) och även skapade en mer dynamisk och omställningsduktig arbetsmiljö att genomföra projektet i.