

OLSKERCUPCOOKIES

Gruppe D | 31/10-2025



Andreas Rovelt

Cph-ar417@stud.ek.dk

rovelt123



Jonas Meinert Larsen

Cph-jl325@stud.ek.dk

Guacamoleboy



Ebou Ndow

Cph-en196@stud.ek.dk

Ebou

Indhold

Indledning	3
Baggrund	3
Kundekrav	3
Teknologi.....	4
Håb & Vision	5
Mockup & Aktuelt	7
Aktivitetsdiagram	9
Domænemodel	12
ER Diagram.....	13
Nagivationsdiagram	13
Særlige forhold.....	15
Unit Test.....	15
Status.....	17
Proces	17
Konklusion	18

Indledning

I løbet af de sidste to uger har vi arbejdet på et programmeringsprojekt med det overordnede tema "Cupcakes". Formålet med projektet har været at udvikle og levere en specifik opgave til en kunde – Olsker Cupcakes. I denne rapport beskrives vores arbejdsopgaver, diagrammer, planer, krav og anvendte teknologier, således at en 2. semester datamatiker-elev på tilsvarende niveau, kan opnå en fuldstændig forståelse af opgaven.

Herudover har vi udarbejdet forskellige diagrammer, samt udviklet et mockup i Figma, der illustrerer vores proces og tankegang gennem projektets forløb.

Baggrund

Olsker Cupcakes ønsker en komplet online løsning, for at fremme deres forretning, som er bosiddende på Bornholm. Olsker Cupcakes er en økologisk iværksætter cupcake forretning, som har fundet den *helt rigtige opskrift*. Dette betyder, at de ønsker fremdrift i virksomheden, hvorfor en online løsning er på plads. Specielt da de hører til på Bornholm, hvor indbyggertallet er minimalt. De har store planer, og vi skal hjælpe dem til at nå i mål.

Olsker Cupcakes har fået nogle hipstere fra København til at lave en Mockup, som vi kan bruge til videreudvikling af projektet. Undervejs i processen ville man normalt tjekke ind med kunden og få deres forventninger, farvekombinationer mm. Vi har derfor på baggrund af denne Mockup videreudviklet på temaet, og kommet frem til den helt rette løsning til Olsker Cupcakes. Den løsning som skal få Olsker Cupcakes frem i verden.

Kundekrav

Ifm. fremdriften af Olsker Cupcakes og deres fremtidige rejse, har vi fået nogle kundekrav som Olsker ønsker man skal kunne på deres platform.

Olsker Cupcakes ønsker at man som kunde kan navigere problemfrit via deres platform. Man skal ikke være i tvivl om noget som bruger. Herudover skal man kunne bestille specifikke cupcakes. Enten fra den faste menu, eller via "custom" afdelingen

hvor man som bruger kan lave sine helt egne cupcakes. Dvs. man kan tilføje en bund og en topping alt efter ønske. Når man er færdig med dette, kan man enten få det leveret, eller man kan komme fordi butikken og hente det selv. Ønsker man som kunde at gemme sine oplysninger har man mulighed for at oprette en bruger og bruge denne til fremtidige bestillinger, eller til blot at se hvilke favoritter man har.

Olsker Cupcakes har været i branchen længe, hvorfor de ved at kunder ofte ønsker returneringer eller afmeldinger af ordre. Derfor ønsker Olsker Cupcakes en løsning online såfremt de kan gøre netop dette. Altså en menu hvor de kan gøre alt fra at ændre priser, tilføje cupcakes, sende penge retur til kunder, afmelde ordre osv. Alt det en forretning kan i virkeligheden, ønsker Olsker Cupcakes at kunne online.

Olsker vil også gerne have at kunden har tilsvarende muligheder for at lette presset på kundeservicepersonalet. Derfor ønskes en menu som brugere kan benytte sig af til fx returneringer, retur, ændring af data og evt. Dette skal gerne sikre at e-mail og opbringninger bliver minimeret uden at miste kunder. Kunden har derfor mulighed for at navigere selv, og finde netop de løsninger, som de står og skal bruge.

Olsker Cupcakes vil gerne have at man som bruger undervejs i processen, med valg af cupcakes - får en menu - så man altid kan se hvad man har i sin kurv, og hvad prisen er (eks. levering). Der udarbejdes derfor en løsning som passer helt perfekt ind i Olskers vision. Olsker vil herudover gerne have at man som bruger altid kan se sin e-mail. Dette implementeres derfor. Olsker er informeret om, at det muligvis ikke giver super meget mening, men de ønsker til at starte med denne løsning, hvorfor vi løser det for dem.

Teknologi

Vi har udviklet projektet i Java 17, og benytter os af Maven som build. Herudover har vi benyttet diverse libraries til fx hashing af passwords, http routing osv. Den komplette liste er herunder:

- Javalin 5.6.1
- Jackson Databind 2.15.0
- BCrypt 0.4
- Javalin Rendering 5.5.0
- SLF4J Simple 2.0.7
- Thymeleaf 3.1.3

- PostgreSQL JDBC 42.6.0
- IntelliJ IDEA 2025.2.1 (Ultimate Edition)
- Visual Studio Code 1.104.3
- Javax Mail 1.6.2
- Junit-jupiter 5.8.1
- Docker 4.48.0

Der er benyttet flere teknologier, men disse har ingen betydning for projektet hvorfor de ikke inkluderes. Dette inkluderer bla Live Server, Code With Me, Photoshop, mm.

Håb & Vision

Som beskrevet tidligere, ønsker Olsker Cupcakes grundet deres placering på Bornholm at ekspandere deres virksomhed online, for at skabe nye kunder, og få deres navn ud i verden.

Olsker har derfor behov for en komplet løsning, som inkluderer ting som levering, kontakt og virtuel bestilling af ordre. Olsker ønsker ikke at ekspandere deres forretning uden en onlineløsning, da de ved at dette vil resultere i, at telefonen vil ringe mere end normalt, og det er ikke det Olsker Cupcakes ønsker. Olsker vil fokusere på det de er gode til – cupcakes. Derfor laves en løsning som netop hjælper Olsker Cupcakes, til at komme videre i verden, og få flere visninger.

Olsker skal derfor blot producere deres cupcakes - og pakke dem. En evt løsning kan være at de køber sig ind i en lagervirksomhed, og disse står for pakning. Dette ville gøre det endnu nemmere for Olsker at fokusere på deres vision, men indtil da står de selv for pakning af ordre.

De har sat sig nogle helt specifikke krav som de ønsker man skal kunne på deres platform. Såkaldte User Stories. De kan findes herunder:

User Story #1

Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.

User Story #2

Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en en ordre.

User Story #3

Som administrator kan jeg indsætte beløb på en kundes konto direkte i Postgres, så en kunde kan betale for sine ordrer.

User Story #4

Som kunde kan jeg se mine valgte ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.

User Story #5

Som kunde eller administrator kan jeg logge på systemet med email og kodeord. Når jeg er logget på, skal jeg kunne se min email på hver side (evt. i topmenuen, som vist på mockup'en).

User Story #6

Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

User Story #7

Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.

User Story #8

Som kunde kan jeg fjerne en ordrelinie fra min indkøbskurv, så jeg kan justere min ordre.

User Story #9

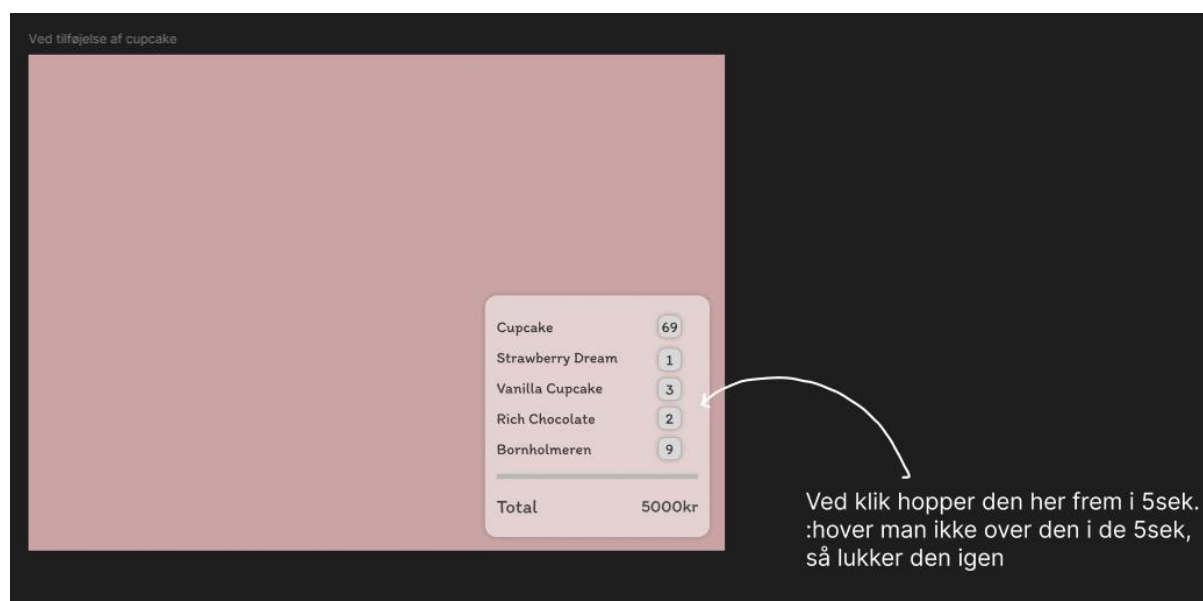
Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde udgyldige ordrer. F.eks. hvis kunden aldrig har betalt.

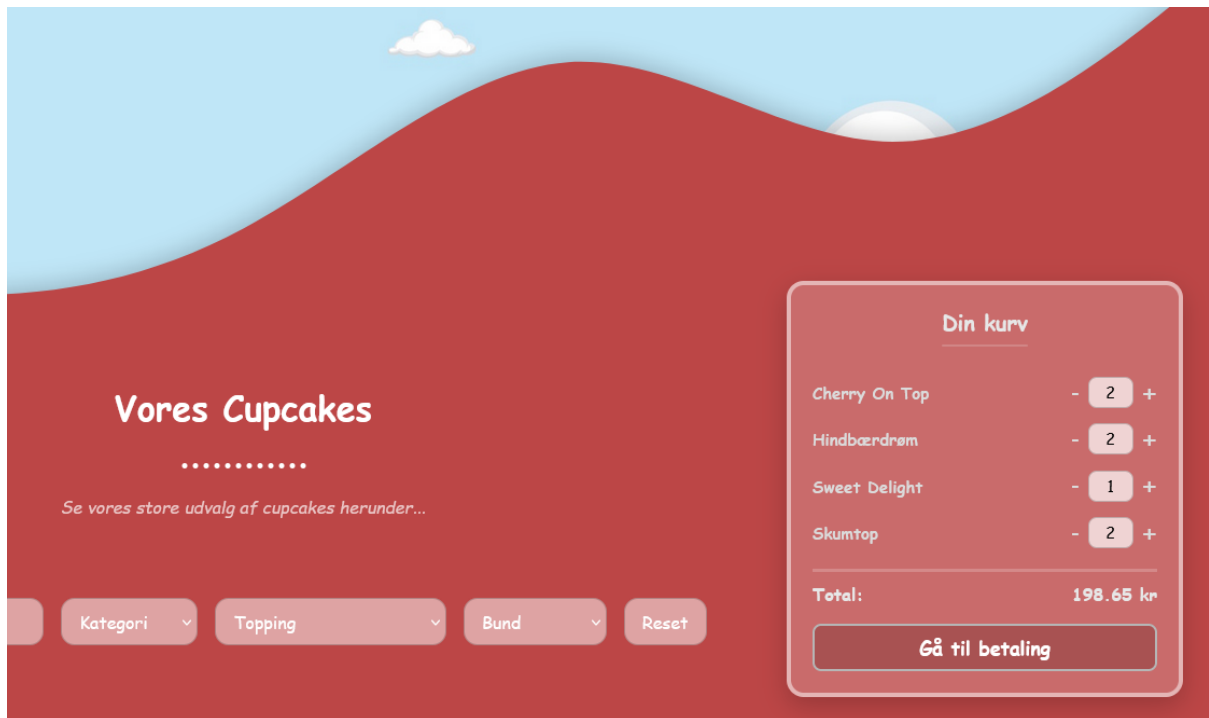
Mockup & Aktuelt

Som en del af vores projekt har vi leveret og lavet en mockup, som kan bruges til at vise kunden en idé på et produkt, inden projektet kodes. Vi har som gruppe, benyttet os af vores mockup som aftaledokument ift frontend delen. Alle ved hvilke farver vi benytter os af, og hvordan vi vil have at hjemmesiden ser ud i sidste ende. Personligt mener vi at et sådant aftaledokument, siger langt mere end fx en domænemodel, men begge har hver deres formål selvfølgelig.

Det vigtigste er dog, at alle - og specielt Lead Developer - ved hvordan slut resultatet er, og eller forventes at være. Dermed kan de andre developers guides korrekt ind på en path. Vi har ifm vores Mockup valgt nogle specifikke farver mm, som kendetegner vores hjemmeside. Disse er utrolig vigtige hvorfor vi har sat det op som side identitet i :root format. Dette sikrer at alle benytter samme farver og kombinationer. Ved at gøre det på den måde, og loade farverne med f.eks. `var(--cupcake-background)`, kan vi sikre at alle elementer som ift vores mockup og slutprodukt - altid har samme farve.

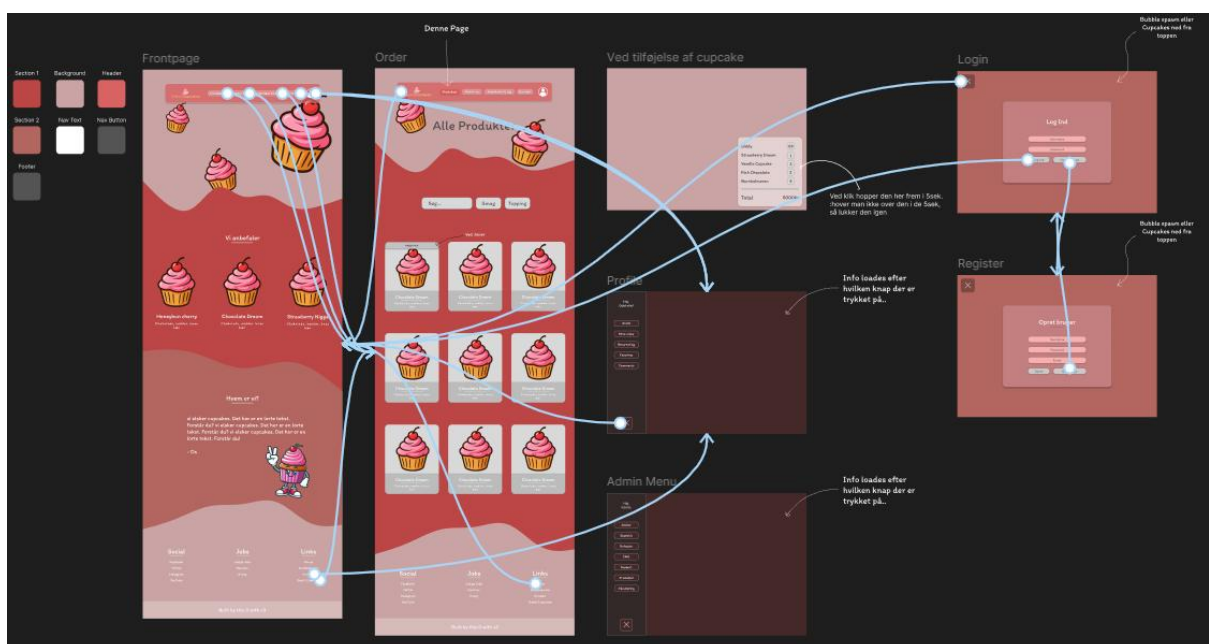
Herunder har vi vedhæftet en lille illustration af et specifikt element fra vores Mockup, for at vise Mockup vs slutprodukt. Hele vores Mockup kan findes under "Diagrammer" som mockup.png. Herudover har vi inkluderet et link til vores mockup preview som du kan finde under vores sammenligning.



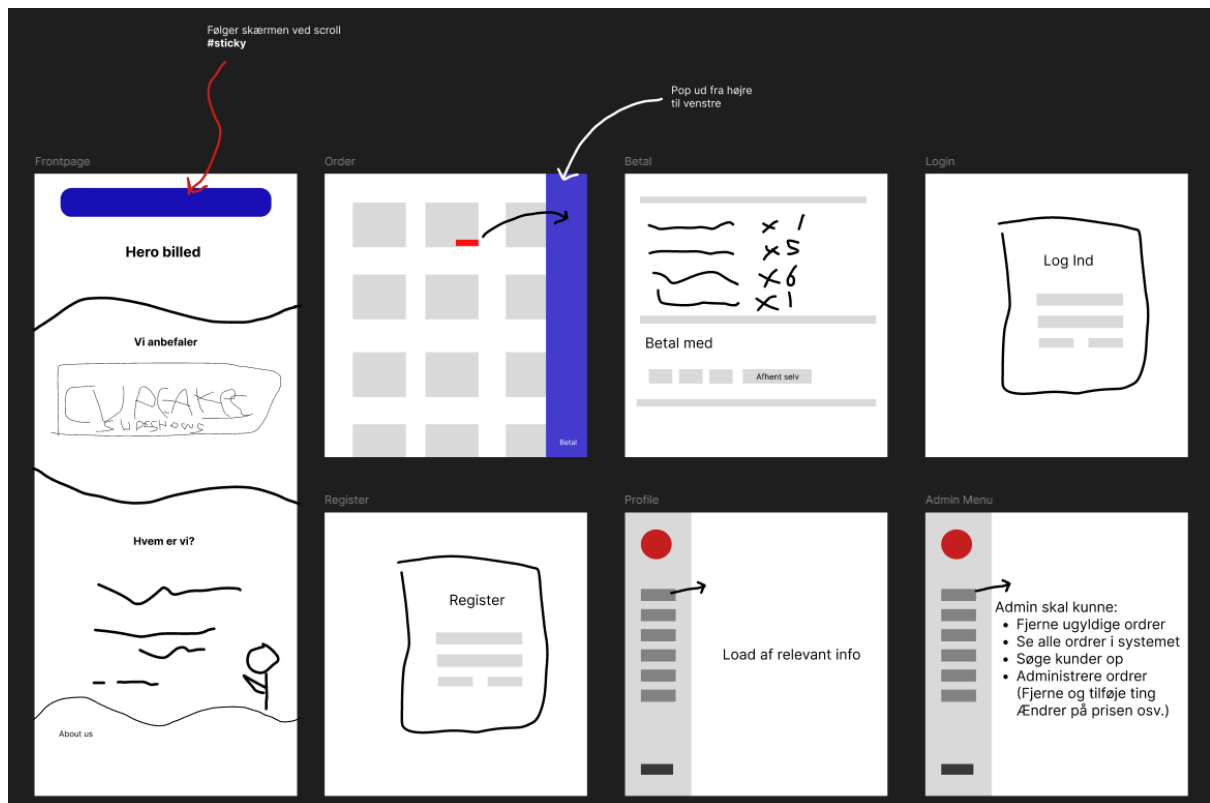


Som man kan se, har vores Mockup givet os et godt udgangspunkt, for hvordan vi mener vores færdige produkt skal se ud. Derfor er de næsten 1:1. Der er nogle små ændringer undervejs, da ting opdages som vi ikke havde haft i vores overvejelser, da vi lavede vores Mockup. Dette kan næsten ikke undgås.

Vi synes Mockup er yderst relevant, og noget som er meget vigtigt i et projekt, og bestemt noget som vi benytter os af fremover.



Herunder har vi inkluderet vores Paper Mockup som vi har lavet i Figma med tegne funktionen over Discord Voice.



Som det fremgår herover, var vores Paper Mockup ikke langt fra hverken vores Mockup, eller vores final produkt (hjemmesiden). Det har givet os et godt udgangspunkt til at lave vores Mockup, og noget vi kan se os selv bruge i fremtiden.

Preview link:

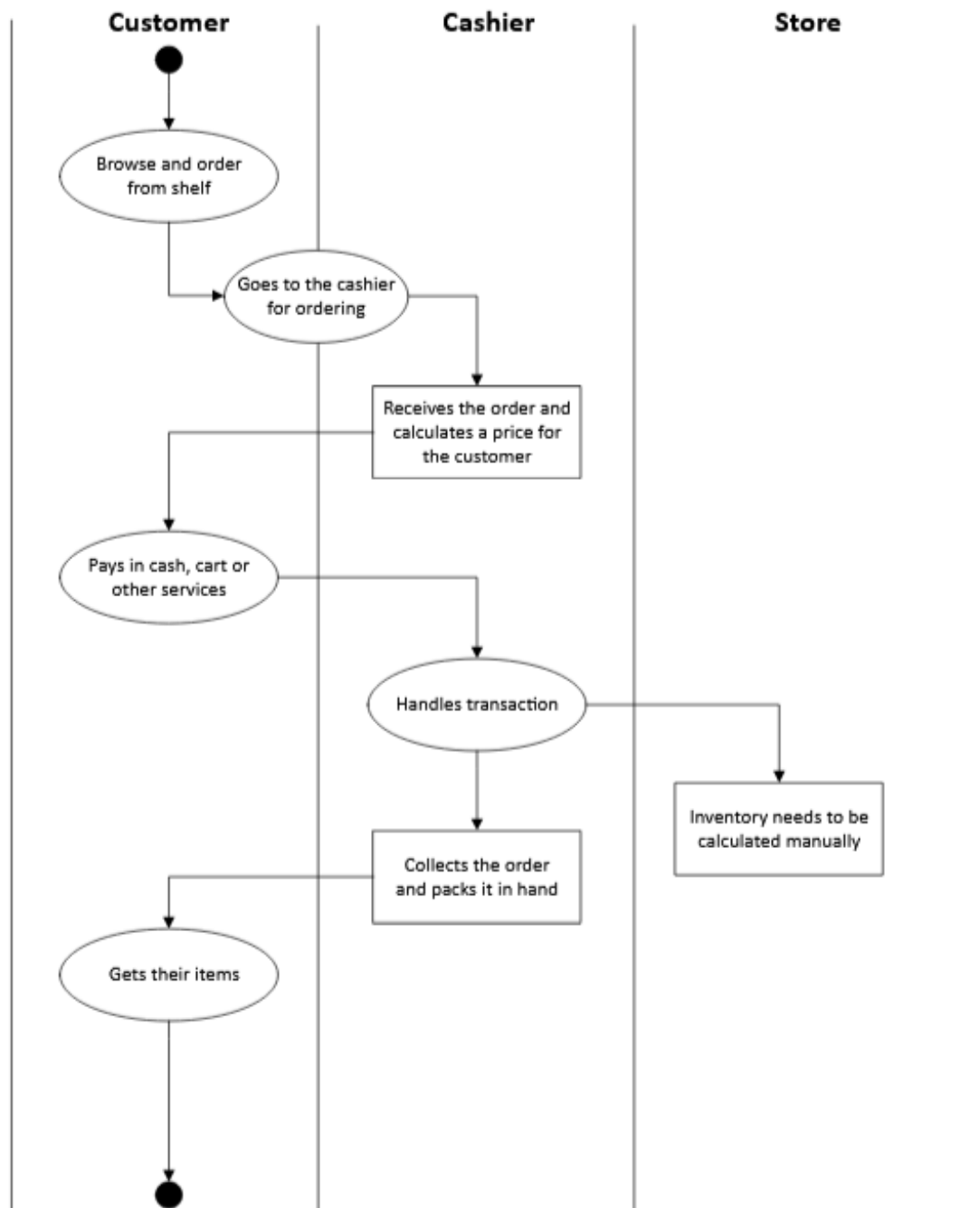
<https://www.figma.com/proto/42OztN095doLX1cTX0RmWr/CupCake-Mockup?node-id=1-3&t=RunOVWds22FMNYvp-1>

Aktivitetsdiagram

Vi har ifm. vores projekt udarbejdet et aktivitetsdiagram som hhv. AS-IS og TO-BE. Altså hvordan virksomhedens forretning ser ud lige nu, og hvordan det kommer til at se ud efter vores løsning er implementeret. Det hjælper både virksomheden, og os selv til at forstå om vi er på rette retning, og om det er den løsning som alle er enige i.

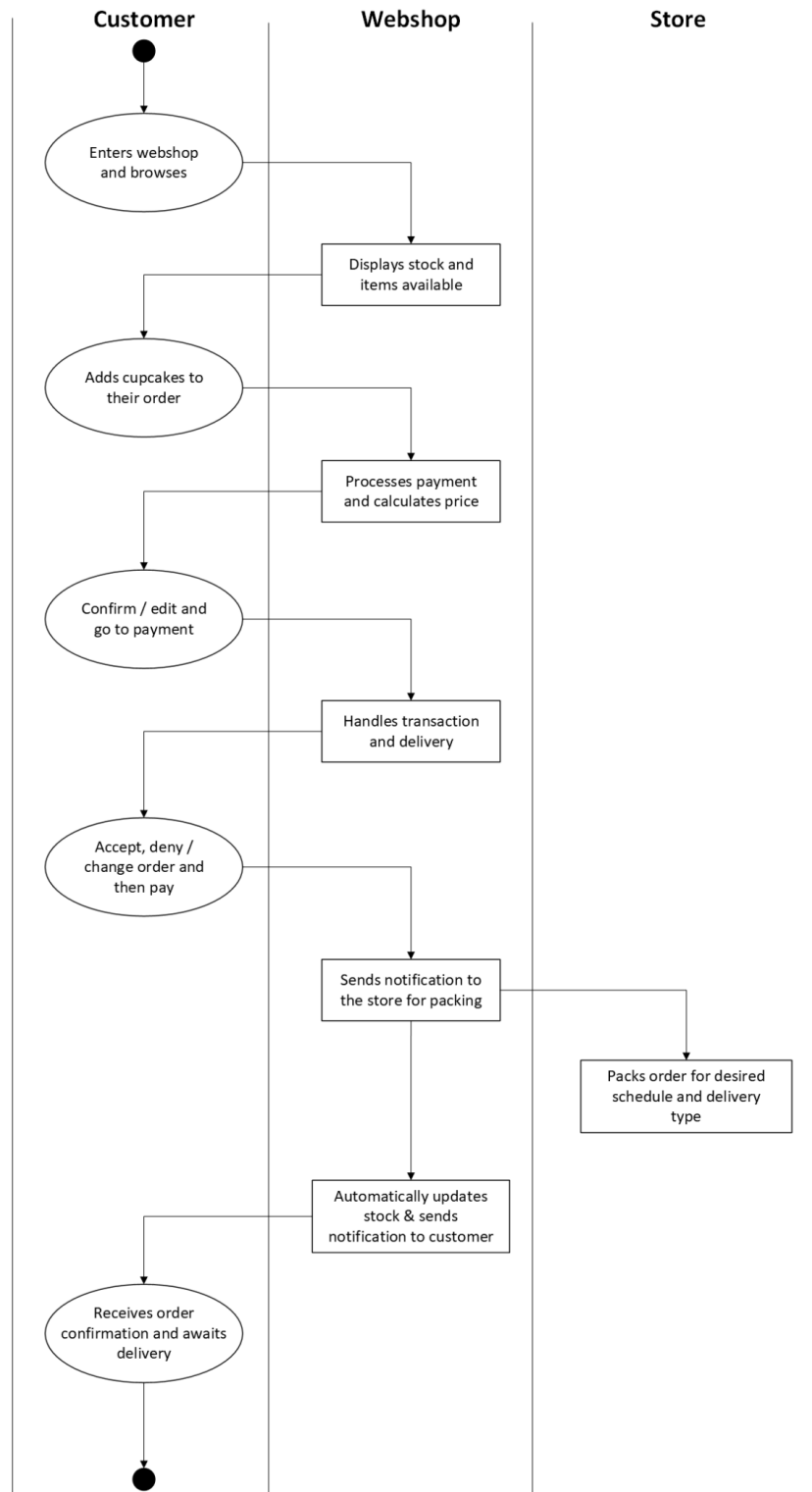
AS-IS:

En del af vores løsning har bl.a. været at udarbejde et AS-IS diagram til kunden, så de har mulighed for at se hvordan deres forretning fungerer lige nu. Som man kan se på diagrammet herunder er der meget mandkraft, forstået på den måde at der er mange ting som skal gøres manuelt, som ikke giver super meget mening ift. en onlineforretning. Olsker Cupcakes vision er at blive større og få bredere kundekreds, hvorfor denne løsning ikke er holdbar i længden, da arealet af forretningen ikke kan udvides.



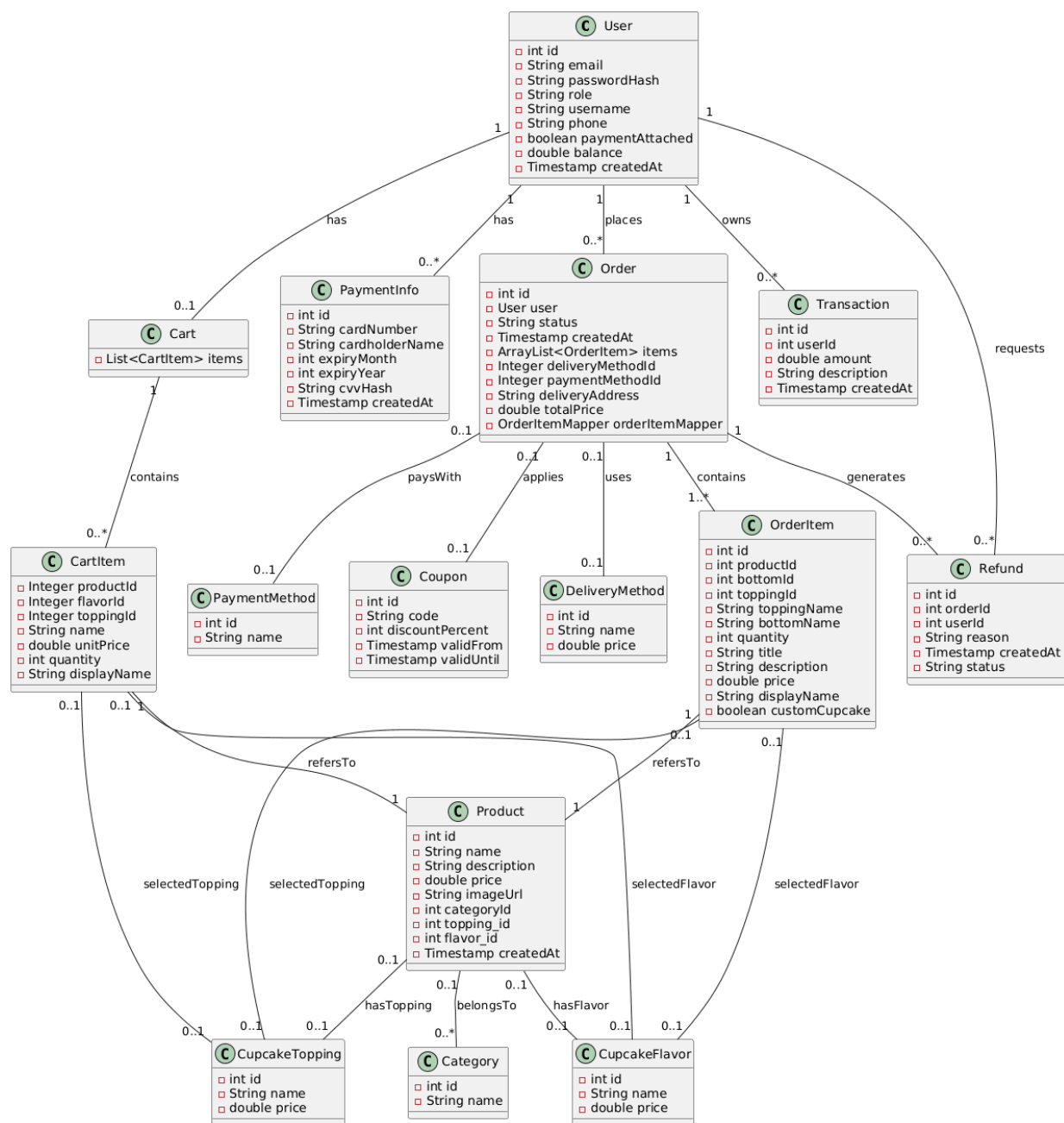
TO-BE:

Efter vores løsning er implementeret, håndterer vores webshop en stor del af de nye og muligvis eksisterende kunder, hvilket letter presset på Olsker Cupcakes, og deres medarbejdere. De har nu tid til at udvikle, og bruge tid på det de i virkeligheden har passion for – cupcakes.



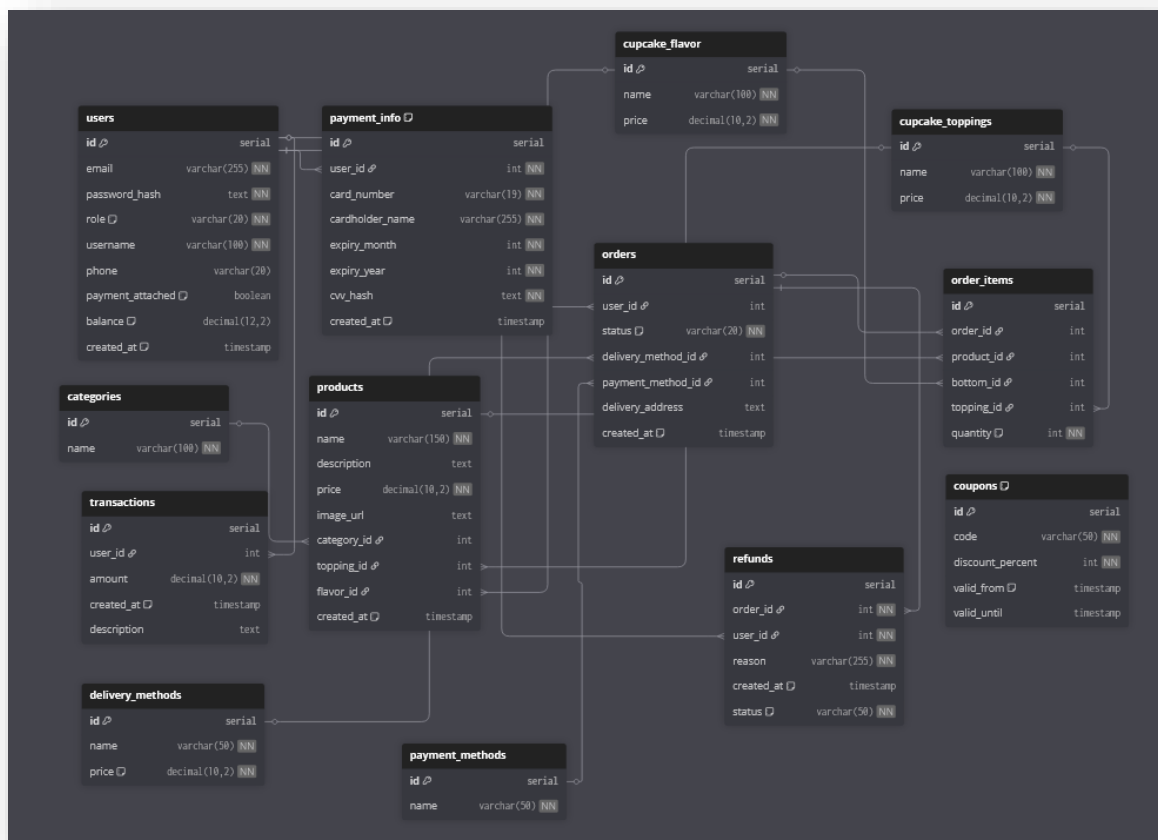
Domænemodel

Vi har i vores projekt lavet en domænemodel, som giver os et godt grundlag og fælles forståelse for hvad, og hvordan vores system fungerer. Sådan at alle er på samme kurs, og har samme vision. Denne sammen med User Stories og et ER Diagram, så er alle på samme plan, og klar til at arbejde.



ER Diagram

Vi har kigget på vores domænemodel, og denne har nu givet os en fælles forståelse for hvordan vores system fungerer, og håndterer data. Derfor kan vi gå videre til næste lag. Håndtering af data. Dette gøres gældende for et ER Diagram sådan at alle har samme forståelse for, hvordan data lagres, og hvilke nøgler der er vigtige. Herudover giver det anledning og bedre forståelse for hvordan databasen fungerer, uden at skulle tyde det fra en kode.

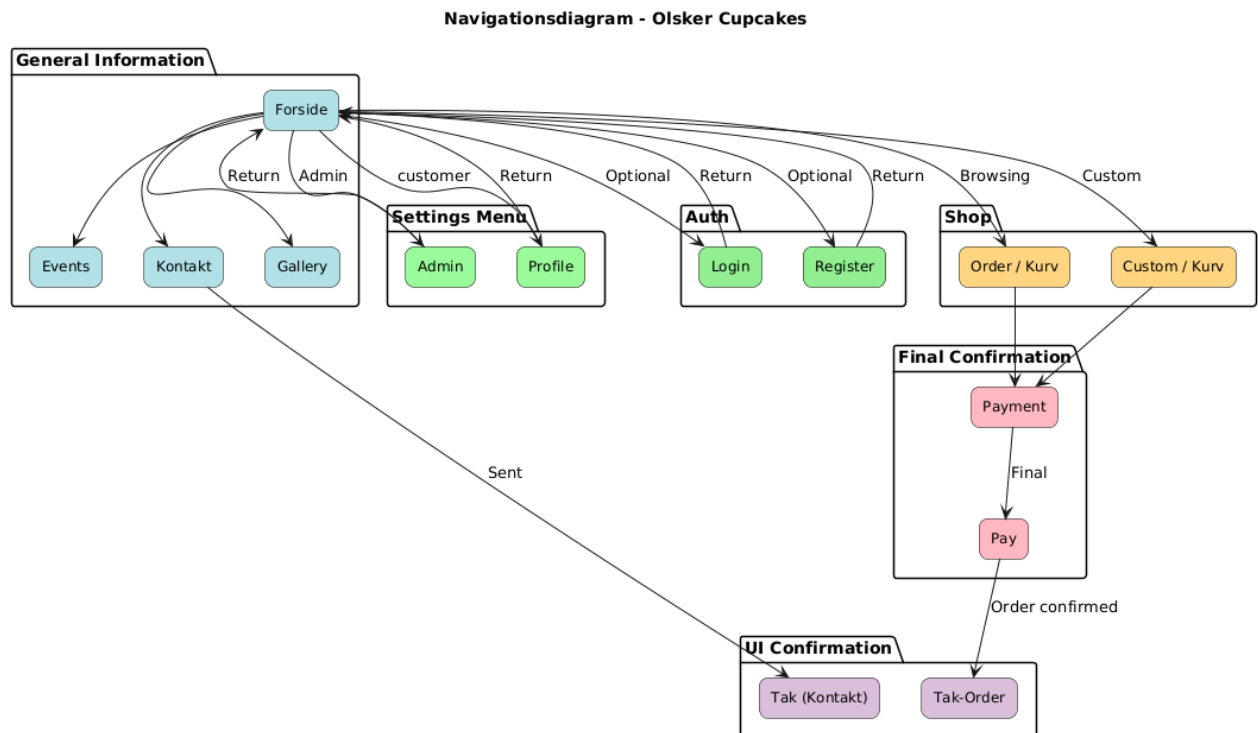


ER Diagram kan findes i folderen "Diagrammer".

Nagivationsdiagram

Navigationsdiagrammet skal sikre, at man via UI forstår netop hvordan man kommer fra de forskellige sider, og tilbage igen. Det er en yderst vigtig del af projektet, da Olsker Cupcakes netop ønsker en online løsning som fungerer upåklageligt. Derfor er det

vigtigt vi har helt styr på hvad de enkelte sider gør, og hvordan man kommer fra den ene til den anden



Vi har i vores projekt valgt at lave en navigations- bar som er fast på næsten alle sider. De sider hvor den ikke er gældende er der en exit knap som er tydelig, stor og nem at benytte sig af. Vores navigations- bar loades som component for at sikre at alle sider har samme version og ingen har en fejl version. Herudover kan vi via JavaScript ændre vores knap "Log Ind" til "Min Profil" og "Admin Menu" nemmere, da det kan opdateres via JavaScript, hvor HTML normalt er statisk. Man kunne også lave det i Thymeleaf, men det fungerer bedre i JavaScript - føler vi. Derfor har vi valgt den løsning.

Vi har herudover en footer og en footer-nav, som også er gældende pr side via components, for at sikre at de alle har samme version. Footer-nav har kundekravet med e-mail i sig. Footer har bare Quick Links til relevante sider og sociale medier.

Vi har som sagt lavet en "Admin Menu" og en "Min Profil". Disse kan kun tilgås af korrekte roller. De har `ctx.redirect safety` på deres routing hvorfor det gerne skulle være umuligt at nå siden uden at være logget ind. Såfremt man forsøger på det bliver man sendt til /login.

Særlige forhold

Vi har i vores løsning været nødt til at gemme relevant information i session attributter, såsom User Data og Ordre data. Dette er en nødvendighed for at kunne benytte sig af fx vores kurv på tværs af siderne. Vi har håndteret exceptions via vores egne exceptions, og dem som kommer med Java. Dette inkluderer ting som SQL, input mm. Som yderligere funktion, har vi i vores exceptions kaldt til en custom notification menu, som displayer til kunden / admin hvad de gør galt. Dette er for at man aldrig er i tvivl om, hvad der sker, og hvad man skal gøre anderledes. Vi har forsøgt at ramme alle fejlkoder som fx 404, 429, 500 osv.

Validering af data og opbevaring af data, er tænkt meget igennem. Vi benytter os af BCrypt, som giver os mulighed for at hashe sensitiv information såsom kortoplysninger, password mm. Dette gør at såfremt Olsker bliver hacket vil man ikke have adgang til sensitiv information, medmindre man dekrypterer informationen. Herudover selvfølgelig så Olsker ikke kan se informationen selv. Validering sker via forms som sender data til vores routing. Vores routing håndterer derefter denne data, og kigger om det er korrekt via en Mapper. Såfremt informationen ikke er korrekt eller brugeren ikke findes, displayer vi en custom notification til brugeren.

Vi har som sagt tidligere forskellige menuer og roller hvorfor det er relevant at kigge efter, og sikre at man ikke kan oprette en bruger og få "admin" rolle uden grund. Derfor sikrer vi os at alle nye brugere får "Customer" rollen, og Admin Menuen eller Developers kan derefter tilføje rolle, eller oprette Admin roller selv. Denne data hentes via vores Mapper og JDBC og gemmes i vores localStorage JSON til at load de korrekte menuer, footers mm.

Ifm opbevaring af sessionAttributes og data heraf har vi validering og safety på redirects i .post via Thymeleaf. På den måde sikrer vi os, at man ikke bare kan åbne vores "Admin Menu" hvis man skriver /admin i url, eller er logget ind som customer.

Unit Test

Som noget nyt i vores projekter som gruppe har vi valgt at lave Unit Tests til alle vores Mappers. Vi kan se idéen bag, og ønsker at udfordre os selv i noget, som vi måske synes mindre godt om at lave.

Vi har benyttet os af den AAA-model som vi lærte på 1. Semester fra Tine. Arrange, act & assert.

Det har givet os et fint indblik i, hvorfor man ville lave Unit Tests, og hvilken funktion de har i større projekter og gruppe sammenhænge. Os bekendt har Github også en funktion som gør, at man kan inkludere Unit Test eller lign ved PR. Det vil sige at når en Developer laver en PR, så kører den vigtige tests først og kigger om de fungerer uden problemer. Det giver Lead Developer (Scrum Master) mere tid til at fokusere på relevant kodning, og leading af gruppen og dens vej.

Måden vi har valgt at lave vores Unit Tests i dette projekt er, at indsætte en database med samme navn som den originale, men med _test efter. Dette sikrer, at man aldrig ved en fejl kommer til at ændre i den originale database. Herudover giver det en form for "local environment", som kan være effektivt for nogen developers, og giver mere frihed fremfor bekymring af at ændre i den originale database.

```
@BeforeAll
static void beforeAll() {
    Database.setDatabaseName("Cupcake_test");
}
```

```
// -----

public static Connection getConnection() throws SQLException {
    String url = String.format(URL, databaseName);
    return DriverManager.getConnection(url, USER, PASSWORD);
}

// -----

public static void setDatabaseName(String newDatabaseName) {
    databaseName = newDatabaseName;
}
```

Som det fremgår herover har vi ændret vores Database klasse, så den tillader dynamisk ændring af database navnet. Dvs når vi kører vores tests, så kører den altid på den

korrekte database, og når vi kører det originale program, så kører den uden _test efter.

Vi føler selv det har været den helt rigtige løsning for vores specifikke gruppe. Man kunne også have lavet det som et schema, men vi føler det her er en mere sikker approach.

Status

Vi har lavet og leveret en komplet løsning som gerne skulle være fuldt funktionel. Dette inkluderer fejlhåndtering, session, menuer, components mm.

Der er selvfølgelig ting som kunne være lavet bedre og eller anderledes, men i første omgang er løsningen komplet. Vi har ikke tilføjet payment da det normalt bliver håndteret af 3 part. Derfor har vi lavet en visuel approach i ren frontend. Vi har dog sat databasen op, så det bare er at connecte frontend skulle man ønske det i fremtiden.

Proces

Vi som gruppe er generelt meget ambitiøse, og ønsker at teste os selv. Derfor har vi forsøgt at lave en bedre version, og en større version end man normalt ville lave. Vi har forsøgt at lave en nyfortolkning af hvad en cupcake forretning kunne være, og hvordan den kunne se ud.

Vi har arbejdet hjemme, men med faste datoer og tidspunkter hvor man arbejder enten sammen over Discord, Code With Me, NGROK mm. Vi har herudover benyttet os af Kanban Github Project Overview, så alle ved hvad der skal laves, omfanget og har mulighed for at sætte sine egne forventninger og rammer til projektet. Vores Project Overview checkes og reguleres regelmæssigt af Lead Developer som i dette tilfælde var Jonas. Han stod derfor for at oprette bugs, issues mm så der altid var noget at tage sig til. Man havde altså selv ansvaret for sit arbejde.

Vi arbejdede primært med Pull Requests og reviews heraf fra Lead Developeren. Vi arbejdede i branches og havde en afleveringsbranch (Main) og en Development branch som vi brugte til at samle diverse løsninger, og teste dem af inden de blev sendt til aflevering. Det er her man evt finder bugs og tester dem. Vi valgte efter lidt tid at lave en

”review” branch yderligere da der var for mange fejl og bugs i de PR som blev sendt til Development. Når en Developer havde lavet et issue eller noget kode, så blev det sendt til review hvor Jonas som Lead Developer, stod for at kigge det igennem, og se om der var bugs eller fejl, som ville være et problem for koden i Development.

Generelt gik arbejdet fint nok. Vi kom i mål. Næste gang skal vi enten sætte ambitionerne ned, først starte på projektet, når ugen begynder eller få flere medarbejdere, da dette ikke er holdbart og eller belønnes mere end folk som leverer ”bare minimum”. Herudover har vi alle en forskellige idé om hvad det vil sige at have projektuge. Nogen arbejder meget. Andre arbejder ikke så meget. Det betyder så at nogen laver væsentlig mere end de resterende. Dette kan have væsentlig indvirkning på projektet og humøret i gruppen hos de enkelte individer hvorfor vi forsøger at finde en anden konstellation til næste projekt.

Konklusion

Vores løsning afspejler Olsker Cupcakes vision, og de har nu en komplet løsning som de kan bruge til at komme ud i verden og ikke begrænse sig af deres geologiske placering. Vi har testet det af, og sikret at man som Developer i fremtiden har mulighed for at forstå vores kode, og hvad der sker i vores løsning. Herudover har vi inkluderet diverse diagrammer og oversigter samt AS-IS såfremt der skiftes Developer. Dette sikrer, at de kan se hvordan virksomheden fungerede inden vores løsning kom til. Det er yderst relevant for dem at se hvordan og hvad der gjorde at Olsker Cupcakes opsøgte os for at få lavet en løsning. Altså hvad de manglede i sin tid. Sådan at man ikke kommer til at gå tilbage til udgangspunktet eller ”reverse”.