Rapport

OLSKER CUPCAKES

Datamatiker 2. semester
Flow 4 - WEBSTACK
Onsdag den 28 april 2021
Klasse A Gruppe 5

Lavet af:
Aleksander Storm, cph-as559@cphbusiness.dk, GITHUB: Alek952
Annika Rosenvinge Jespersen, cph-aj405@cphbusiness.dk,GITHUB: Annika-Rosenvinge
Jacob Volck Sinding, cph-js485@cphbusiness.dk, GITHUB: JacobSinding
Julius Krüger Madsen, cph-jm352@cphbusiness.dk, GITHUB: PsychedelicCoder
Maria Theresa Udengaard McNally, cph-mm620@cphbusiness.dk, GITHUB: cph-mm620

Indledning

Vi fik til opgave at skabe en hjemmeside for Olsker cupcakes. Vi har skrevet denne opgave i Java, HTML, CSS og SQL. Java delen har fungeret som bindeled mellem HTML delen og SQL delen af koden.

Flow 4 Projektet har primært handlet om hjemmesider, og hvordan man bygger dem op fra bunden. Vi har blandt andet benyttet os af Tomcat, og et Droplet på digitalocean, for at hoste vores servere, og bygget vores program i IntelliJ og brugt MySQL som database.

Baggrund

Vi modtog en vigtig opgave fra virksomheden Olsker Cupcakes, som er et dybde økologisk iværksætter firma fra Bornholm. Virksomheden specialiserer sig i salg af cupcakes både som takeaway og eat in.

Kunden stillede os til opgave, at designe og kode en hjemmeside til deres virksomhed, denne hjemmeside skulle have følgende krav:

1. Kunder skal kunne sammensætte sin egen cupcake ud fra de toppings og bunde som virksomheden tilbyder samt bestille deres cupcakes til afhentning ved butikken.
2. Kunder skal kunne oprette en konto på hjemmesiden for, at kunne betale og gemme en ordre.
3. Medarbejdere, skal kunne have en administrator bruger hvorpå der skal kunne indsættes “store credit” på kunders profiler som de kan bruge til køb på butikkens hjemmeside.
4. Kunder skal kunne se deres tilføjet cupcakes i deres indkøbskurv samt den total pris.
5. Som kunde/administrator skal man kunne logge ind med email og kodeord, efter man er logget ind skal man kunne se hvilken email man er logget ind med på hver side. Gerne som en del af topmenuen på hjemmesiden.
6. Administrator brugeren skal kunne se alle bestilte ordrer.
7. Administrator brugeren skal kunne se alle oprettet kunder i systemet samt deres ordrer. Så det er muligt, at holde styr på de forskellige kunder.
8. Som kunde skal man have mulighed for, at slette en ordrer.
9. Administrator skal kunne fjerne ordrer, såfremt de f.eks ikke er betalt.

Teknologivalg

Vi har kodet både backend og frontend til projektetet i IntelliJ Idea 2020.3.3 Ultimate Edition. Sproget der er blevet benyttet til backend, er java 8, og til frontend har vi både benyttet os af HTML, CSS og Bootstrap 5.

Vi har også haft brug for en database til vores projekt, og til det har vi brugt MySQL 8.0. For at få oprettet en forbindelse mellem vores kode og database har vi benyttet os af Maven, gennem IntelliJ Idea.

Til at hoste vores side har vi brugt en Droplet som bliver hostet på Digitalocean, hvorpå vi har installeret Tomcat 9, for at kunne hoste vores web applikation.

Krav

Virksomheden er en iværksættervirksomhed, som endnu ikke har en hjemmeside til deres online bestillinger af deres produkter.

De har en vision om, at benytte hjemmesiden og de data den bringer til, at optimere deres produktion samt indkøb af råvare for derigennem, at kunne mindske deres madspild.

Derudover ønsker de, at føre statistik over deres salg af cupcakes for, at kunne regne ud hvilken bunde/toppings der sælger bedst. Dette ville igen fremme optimeringen af salg i virksomheden.

Med vores system, så er kunderne fri for at vente i kø når de kommer til butikken og de kunder, der kommer til butikken uden at have bestilt i forvejen, skal ikke vente for længe. Systemet giver også kunderne mulighed for at se deres tidliger bestillinger, så de ikke glemmer hvilken cupcake, der er deres favorit.

Vi fik disse userstories at arbejde ud fra:

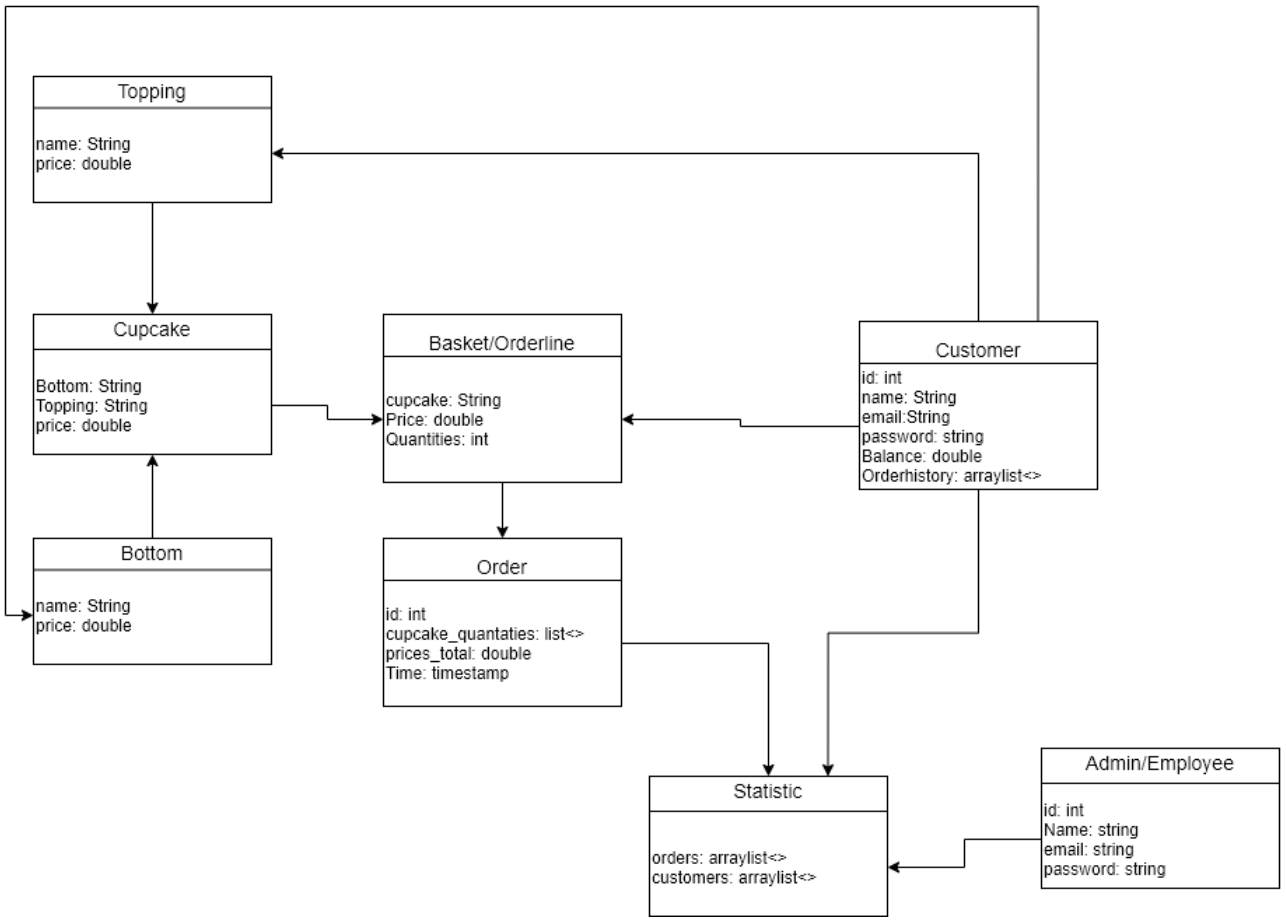
User story 1; Som kunde kan jeg bestille og betale cupcakes med en valgfri bund og top, sådan at jeg senere kan køre forbi butikken i Olsker og hente min ordre.
User story 2; Som kunde kan jeg oprette en profil/konto for kunne betale og gemme en ordre.
User story 3; Som administrator kan jeg indsætte beløb på en kundes ordre direkte i MySQL, så en kunde kan betale for sin ordre.
User story 4; Som kunde kan jeg se mine valgte ordrelinjer 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.

User story 6; Som administrator kan jeg se alle ordre 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 ordre, sådan at jeg kan følge op på ordre og holde styr på mine kunder.
User story 8; Som kunde kan jeg fjerne en ordre 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 ligegyldige ordre. F.eks. hvis kunden aldrig har betalt.

Diagrammer

Her ses alle vores diagrammer:

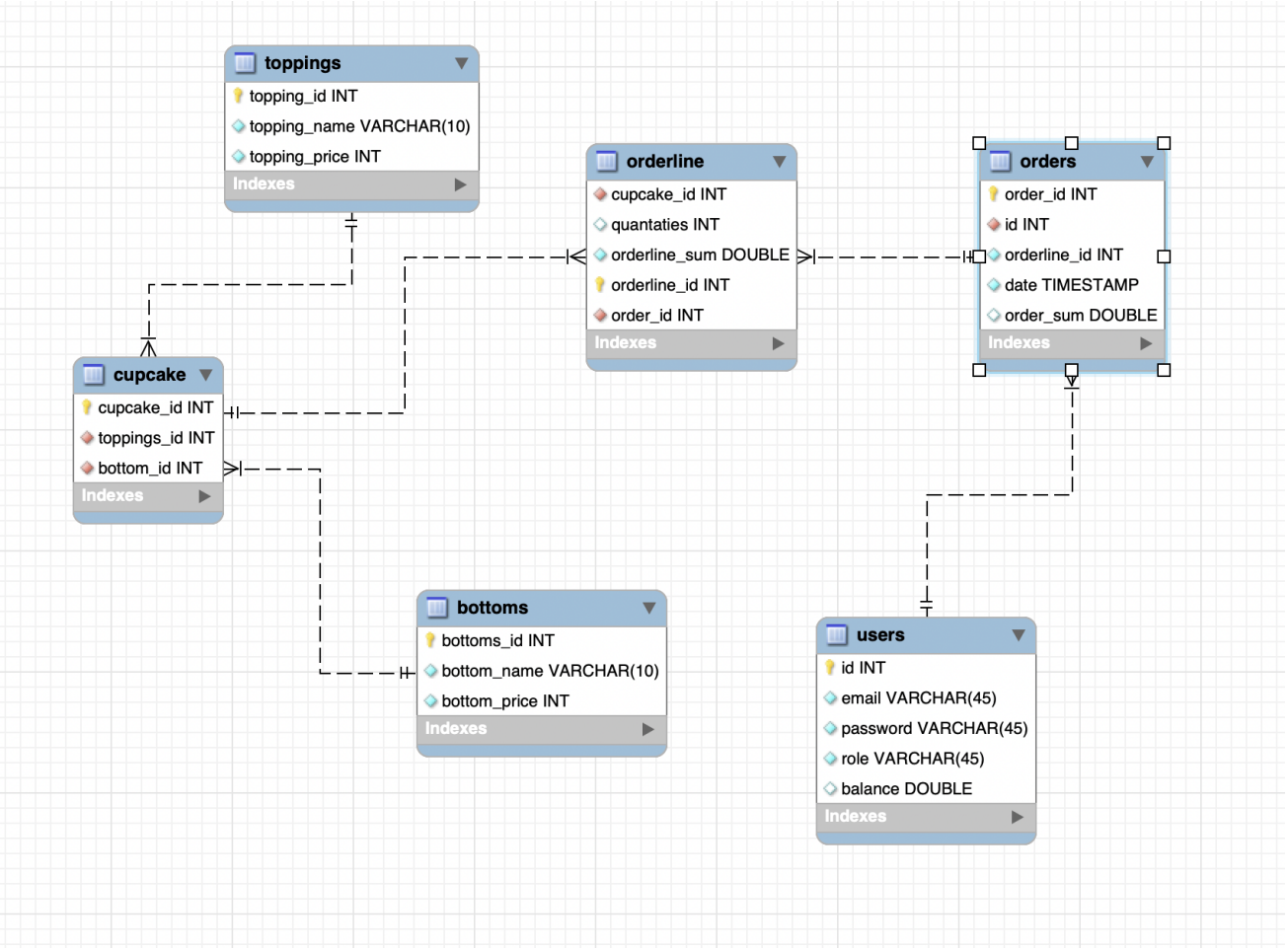
Domænemodel



Her ses domænemodellen for hjemmesiden. HTML/JSP siderne er ikke med og det er med vilje, da de ikke har nogle funktioner/metoder og de viser alt det der foregår bagved. Så hvis en kunde vælger en bund og en topping så bliver det til en cupcake som ryger over i kurven.

ERR diagram

Her ses EER diagrammet over databasen



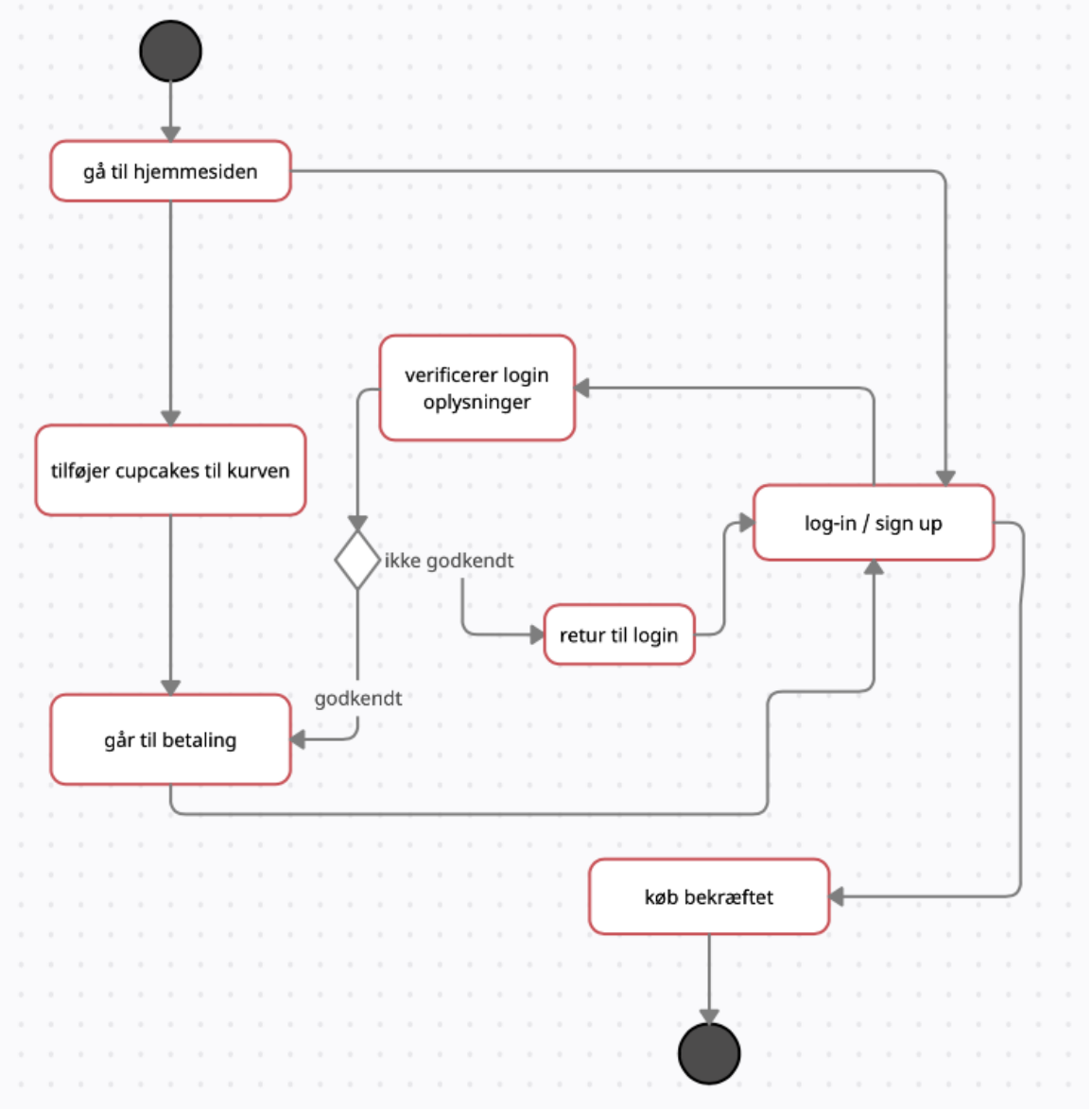
Databasen er normaliseret på 3. normalform. Der er ikke flere informationer end nødvendigt i hver tabel, f eks hvis man kigger på cupcake, kan man se den indeholder 3 ting, et id der auto incrementer, og så indeholder den *topping_id* og *bottom_id*. De kommer hver fra deres egen tabel, der henholdsvis hedder toppings og bottoms. De indeholder hver et id, der bliver sendt videre, et navn og så en pris.

Der kan være flere *orderlines* med hver ordre. Så hvis man f eks skal bruge 4 cupcakes, behøver man ikke gå ind og lave 4 forskellige bestillinger. Grunden til at der er et timestamp på orders er så kunden kan se hvornår de har købt deres cupcakes, så hvis de en anden dag gerne vil have den eller de samme cupcakes, så kan de finde dem. Timestampet bruges også til medarbejderne, så de kan se hvornår de sælger hvad og hvornår de har mest travlt.

Vi har valgt at lave users/employee under samme tabel. Hvis man vil ind på websitet skal man tilgå det, med en email og kodeord, og alle kontoer er tilknyttet en rolle, enten customer, eller employee. En user kan enten være medarbejder, der kan indsætte penge på folks konto, se hvilke cupcakes der er blevet bestilt og så videre, eller en user kan være kunde, så kan se deres ordre og se deres konto balance. Begge brugere kan bestille cupcakes, da begge har en balance, som det kan ses i tabellen.

Aktivitetsdiagram

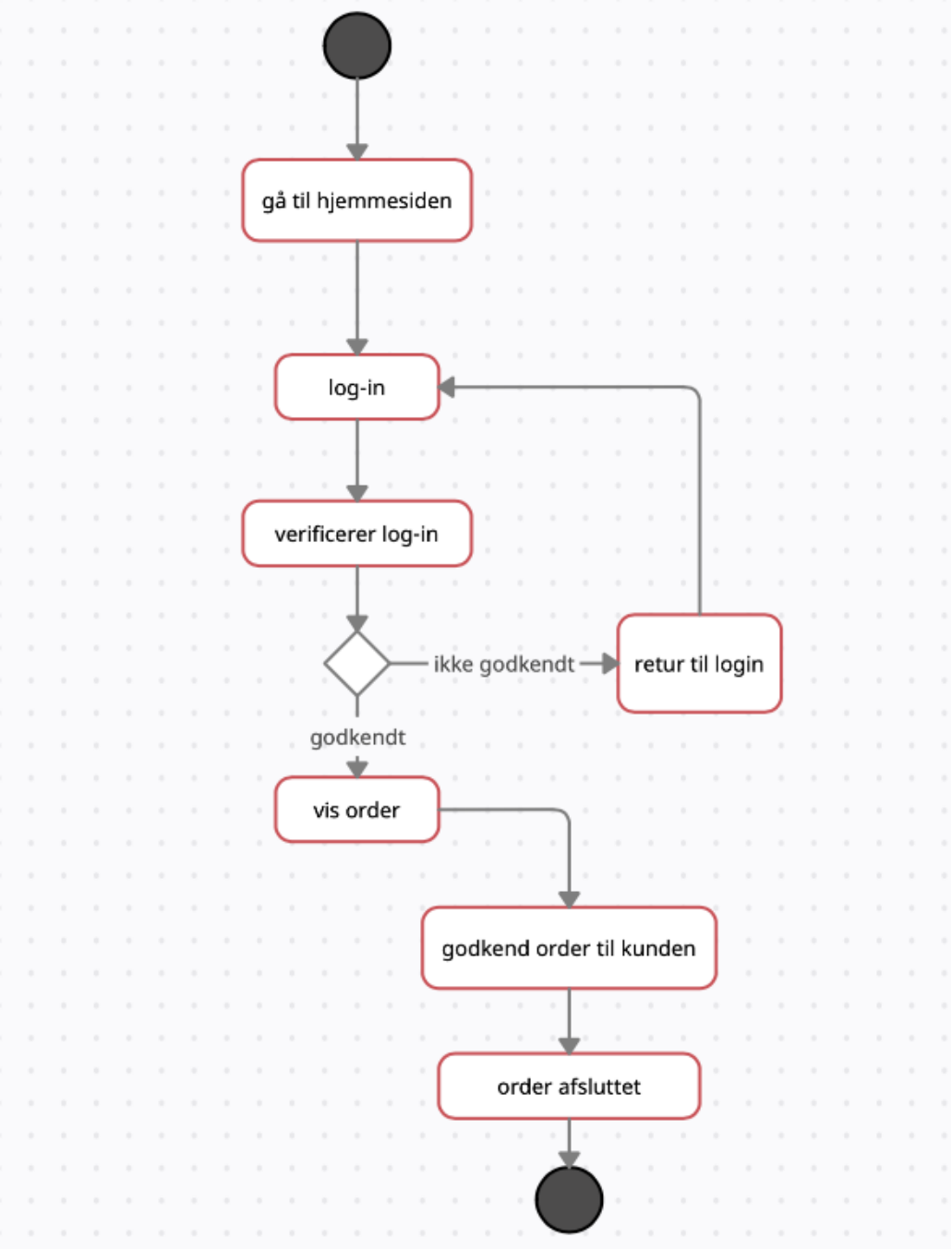
Kunde diagram



På diagrammet kan man se at som kunde/customer, som kan vælge enten at starte med at logge ind eller gå direkte til menuen og tilføje cupcakes til kurven, ved at gøre dette skal man først ved betaling oprette sig eller logge ind med tidligere oprettet log-in for at gennemfører sin order. Når man har oprettet sig eller logget ind kan man dernæst bekræfte købet og kundens del er afsluttet. Hvis man ikke har skrevet sine oplysninger rigtigt vil man blive sendt tilbage til log-in siden for at forsøge igen.

Dette er hvad vi havde tænkt os at lave, så hjemmesiden fungere måske ikke helt på samme måde, da den ikke er færdig.

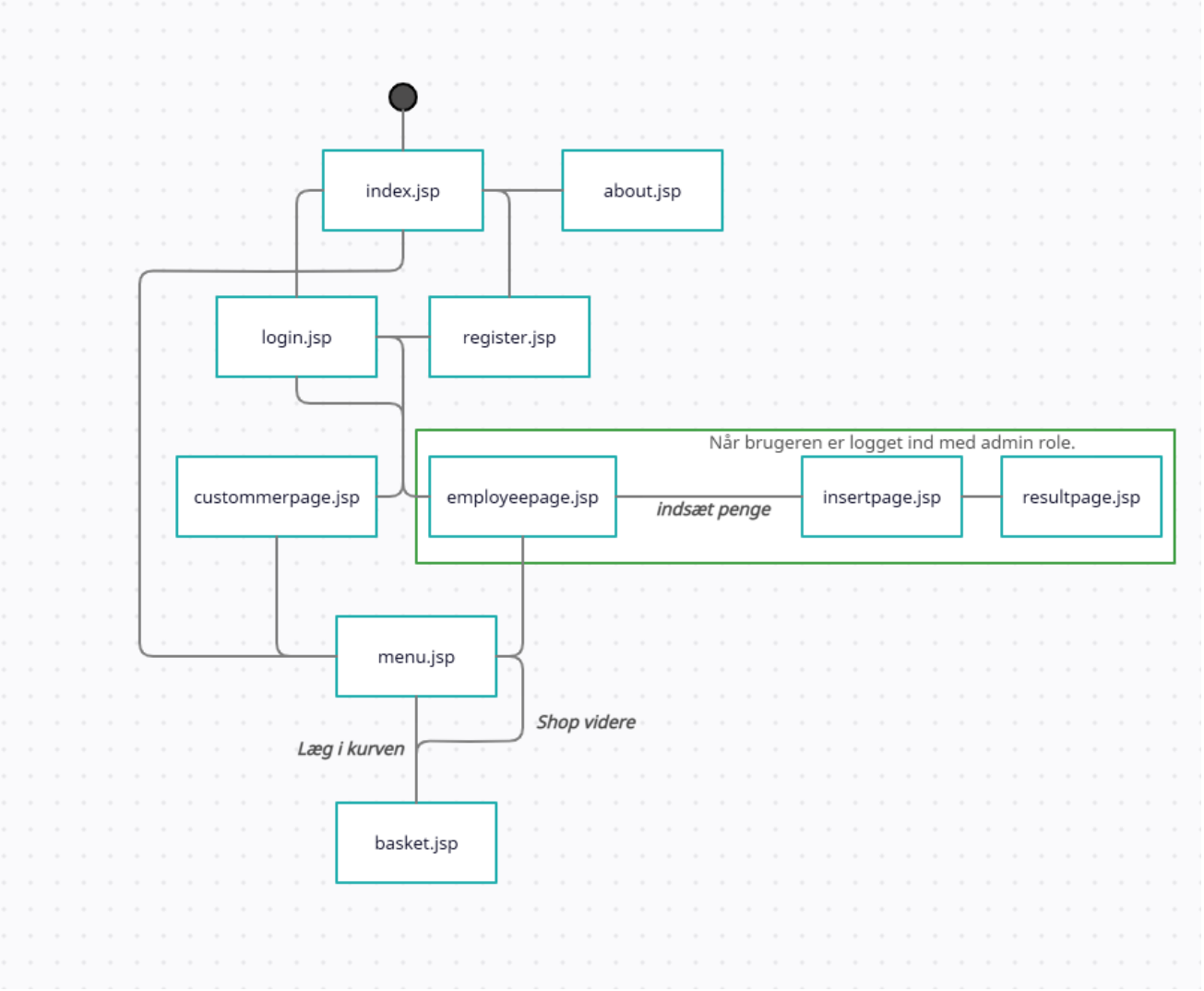
Employee diagram



For de ansatte skal man logge ind med det samme for at kunne se kundens ordre, igen hvis kode eller email er skrevet forkert, bliver man sendt tilbage til log-in siden for at prøve igen. Når man er blevet verificeret kan man nu få vist kundens eller kundernes ordre, her kan man så bekræfte ordrer og give kunden besked på hvornår den kan hentes i butikken. Når kunden har hentet sin ordrer bliver ordren afsluttet og den ansatte kan nu gå videre til næste kunde.

Igen er det hvad vi havde i tankerne og ikke hvad man faktisk kan på hjemmesiden.

Navigationsdiagram



Det første man rammer på vores hjemmeside er *"index"* siden, her kan man vælge at enten *"login"*, *"register"* eller gå til *"menu"*. Der ligger også en *"about"* side i vores navigations bar som brugeren også altid kan tilgå, ligesom at der også der ligger en vej tilbage til *"index"* siden der.

Hvis man ikke har en profil og skal registrer dig som kunde så indtaster man sin email og ønsket kodeord på *"register"* siden, hvor man herefter vil blive videresendt hen til *"Customer page"* siden, og så er man registreret som bruger i databasen med customer role.

Når du er logget ind er der 2 scenarier enten har du rollen admin ellers er du customer. Hvis du har rollen admin bliver du sendt hen til *"employee page"* siden, her vil der være en knap der hedder *"indsæt penge"* som vil tage dig hen til *"insert page"* siden, hvorefter du kan indtaste det ønsket antal penge der skal sættes ind og emailen til den konto hvor pengene skal sættes ind på, når admin har indtastet de ønskede værdier er der en *"submit"* knap der vil tage en hen til vores *"result page"* side, her modtager admin en kvittering på at pengene er indsat på den ønskede konto hvis alt er gået godt.

Hvis brugeren bevæger sig ind på vores *"menu"* side, der vil være 2 dropdown menuer en til

topping og en til bund, når brugeren har valgt sin ønsket bund og topping vil der være en *"læg i kurv"* knap som tager brugeren hen til vores *"basket"* side. på *"basket"* siden er der en oversigt over den valgte sammensætning af cupcake, og her har brugeren 2 valg, enten kan man trykke *"køb"*, hvilket lige nu bare rydder siden. ellers kan man trykke *"shop videre"* knappen og der vil man blive sendt tilbage til "menu" siden.

Særlige forhold

I vores program har vi ikke fået implementeret så mange informationer, som gemmes i sessions. Dog bliver vores cupcakes gemt i en session, når de bliver føjet til kurven, så vi kan få dem med videre, når man skal til betaling.
For at håndtere exceptions, har vi oprettet en mappe, hvori vi har vores klasse som bliver refereret til. Den returnere en String, hvis der er nogle krav som ikke bliver opfyldt.

I forhold til sikkerhed, har vi en database som indeholder alle vores brugere. De bliver inddelt i to forskellige roller *(customer, employee)*, som har forskellige rettigheder. Når vi skal validere at det er et gyldigt login som bliver brugt, tjekker vores program at det passer med vores database, i tilfælde af oplysningerne ikke passer, smider den en *UserException* som siger man har tastet forkerte oplysninger ind.
Som sikkerhed for at folk ikke kan tilgå sider med de forkerte adgangsrettigheder, har vi lavet en *CommandProtectedPage*, som tjekker efter brugerens rettigheder.

Vores database er tilsluttet vores kode gennem en JDBC. Denne JDBC gør det muligt at hente data fra vores database, og sende data dertil gennem vores *DatabaseMapper* klasse.

Design valg

For vores egen, virksomheden og kundens skyld, så har vi valgt at gøre hjemmesiden så simpel som mulig. Det er med til at sikre man ikke mister fokus og bliver forvirret over hvor man skal trykke, for at komme derhen eller gøre hvad man gerne vil.
Vi har valgt blå og orange som de 2 primære farver, da de komplementerer hinanden og samtidige giver de modspil, da orange er en varm farve og blå er en kold farve. Begge farver er også klare farver, der nemme at se. Vi har også brugt hvid og grå, men det er som sekundære farver. De er brugt som baggrundsfarver, både på siden, men også knapper.

Logoet er lagt i headeren, så hvis man tilgår hjemmesiden fra f.eks en telefon eller tablet, så skalere headeren, så logoet stadig er først, og knapperne til de andre sider, f.eks menu, falder ind som de bliver til en dropdown menu.

Login og sign up knapperne er helt ude i højre side, det er for at gøre det tydeligt at det er der man logger ind eller kan tilmelde sig, for at købe cupcakes.

Status på implementation

Vi er desværre ikke nået så langt med vores implementation af use cases som vi gerne ville på trods af vi egentlig følte vi var godt med fra starten af. Desværre blev gruppen ramt af noget sygdom og andre forhindringer der gjorde arbejdsprocessen blev forsinket.

U1 - Vi har ikke nået, at lave en funktionel kurv, hvor der kan tilføjes mere end en cupcake. Ydermere bliver denne cupcake tilføjet på alle tabellens linjer. Prisen har vi heller ikke fået nået, at implementere.

U2 - Det er muligt, at oprette en konto/profil og vælge en cupcake. Men cupcaken bliver kun gemt i sessions scopet og altså ikke i databasen. Vi har desværre heller ikke betalingen på plads til dette endnu.

U3 - Vi har fået færdig implementeret hele funktionen til denne use case. Den virker efter hensigten.

U4 - Kunden kan her vælge en enkelt cupcake og gå videre til kurven hvor han kan se den valgt cupcake. Dog optræder den samme cupcake flere gange i kurven og det er ikke muligt, at se prisen på cupcaken.

U5 - Vi har fået implementeret denne use case efter ønskerne fra kunde. Den virker efter hensigten.

De resterende use cases har vi ikke fået arbejdet på. Vi mangler stadigvæk, at implementerer vores database i koden. Vi har på nuværende tidspunkt kun implementeret vores user table fra database i vores kode. Det er her login informationer bliver lageret. Det betyder, at vi altså ikke har mulighed for, at gemme f.eks ordrerne endnu.

Med hensyn til Jsp siderne er alle oprettet og klar, og selve designet af siderne er klar. Dog mangler noget af funktionalitet der er forbundet med f.eks knapperne på siderne. I forbindelse med vi ikke har færdiggjort alle de use cases der er blevet stillet.

Process

Vi startede mandagen ud med, at mødes klokken 09:00. Herfra fortalte vi hver især hvad vi forventede, at ligge i denne opgave med hensyn til arbejdsmoral og arbejdsindsats. Da vi tænkte det var et godt udgangspunkt at arbejde udfra.

Herefter talte vi sammen om selve opgaven for, at finde vores styrker og svagheder. Vi valgte derefter, at hoppe på de opgaver som den enkelte følte de var klar til, at gå i gang med. Undervejs kommunikerede vi igennem en oprettet discord server, så vi kunne opdatere hinanden på arbejdet der blev lavet.

På daværende tidspunkt mente vi dette var den bedste fremgangsmåde. Dog er vi nu blevet klogere, da arbejdsprocessen ikke gik som planlagt.

Nogle i gruppen har haft troen på, at andre i gruppen havde lavet en del af opgaven. Hvor man så senere har fundet ud af denne del slet ikke var blevet løst. Dette har ført til en masse uafsluttede use cases.

Dette har vi dog nu lært på den hårde måde. Så når vi på mandag starter vores semesterprojekt skal vi sørger for, at alle tit melder ind med deres fremgang, så vi kan være sikker på alle er med og får færdiggjort opgaverne.

Derudover skal vi næste gang benytte os af github issues og/eller et kanban board så vi kan få et bedre overblik over hvilke opgaver der skal løses, der er løst og hvilke der er undervejs. Det overblik man får gennem disse løsninger vil fremme vores arbejdsprocess.