

Cupcake Hold A

Sidsel Lefmann Schack
cph-ss683@cphbusiness.dk
Skainsmate

Magnus Kongstad
cph-mk797@cphbusiness.dk
Magkon

Yusuf Al-Khafaji
cph-ya56@cphbusiness.dk
StylizedAce

Peter Andersen
cph-pa153@cphbusiness.dk
Peter537

29. marts 2023



Indhold

1	Indledning	2
2	Baggrund	2
3	Teknologi valg	3
3.1	IntelliJ	3
3.2	Server container	3
3.3	SQL	3
3.4	Java	3
4	Krav	3
4.1	User Stories	3
5	Domæne model og EER diagram	4
5.1	Domænemodel	4
5.2	EER diagram	4
5.3	Navigationsdiagram	5
5.4	Klassediagram	6
6	Særlige forhold	8
7	Status på implementation	8
8	Process	8
9	Videopræsentation	8

1 Indledning

Dette projekt består af en hjemmeside bygget i html, css og java. Til den er koblet en mySQL server. Vi har som vores 2. semester-projekt fået som opgave at lave en cupcake-butik man kan tilgå "online"(lokalt). Til dette har vi skulle uddeligere forskellige opgaver, og udarbejde en løsning til opgaven. Vi startede først med at fastlægge en plan med et navigations diagram og finde overensstemmelse med de funktionelle krav. Løbende har vi haft fokus på at planlægge, drøfte med hinanden, og holde hinanden opdateret på hvad der laves så vi trinvist får udbygget opgaven i 'pace' med hinanden. Vi sikrede os derved at udarbejde backend først med alle de funktionaliteter vi tidligere havde aftalt og udformede det til semi-færdigt produkt. Dertil byggede vi endelig en frontend, og havde alle fokus på det og løbende at interagere med det så det forbliver funktionelt. Der var nogle udfordringer hen ad vejen som vi knoklede med at få løst. Opgaven blev til sidst også testet og optimeret, og dokumenteret med modeller og en rapport.

2 Baggrund

Olsker Cupcakes er et bageri på Bornholm. Deres forretningsmodel består af at sælge cupcakes, hvor man helt selv kan sammensætte hvilken smag bund og hvilken smag top man vil have. For at facilitere dette, har de brug for en hjemmeside hvor deres kunder kan bestille cupcakes til afhentning og betale for dem.

Olsker cupcakes har også brug for at kunne se alle deres kunder og alle deres ordre i systemet. Til dette skal der laves en separat admin del af hjemmesiden.

3 Teknologi valg

3.1 IntelliJ

- IntelliJ 2021.2.4
- IntelliJ 2022.3.1
- IntelliJ 2022.3.2
- IntelliJ 2022.3.3

3.2 Server container

- JakartaEE 8
- Java EE 8
- Tomcat 9.0.73

Jakarta 8 og Java EE 8 er kompatible med hinanden.

3.3 SQL

- MySQL 8

3.4 Java

- Java JDK 19.0.2
- Amazon Corretto 11.0.18

Java JDK 19.0.2 og Amazon Corretto 11.0.18 er kompatible med hinanden.

4 Krav

Kravene til denne opgave blev alle givet i opgavebeskrivelsen, vi har ikke selv haft kontakt med virksomheden.

4.1 User Stories

Til dette projekt har vi fået udleveret de følgende user stories:

- **US-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.
- **US-2:** Som kunde kan jeg oprette en konto/profil for at kunne betale og gemme en ordre.
- **US-3:** Som administrator kan jeg indsætte beløb på en kundes konto direkte i MySql, så en kunde kan betale for sine ordrer.
- **US-4:** Som kunde kan jeg se mine valgte ordrelinier i en indkøbskurv, så jeg kan se den samlede pris.
- **US-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).
- **US-6:** Som administrator kan jeg se alle ordrer i systemet, så jeg kan se hvad der er blevet bestilt.

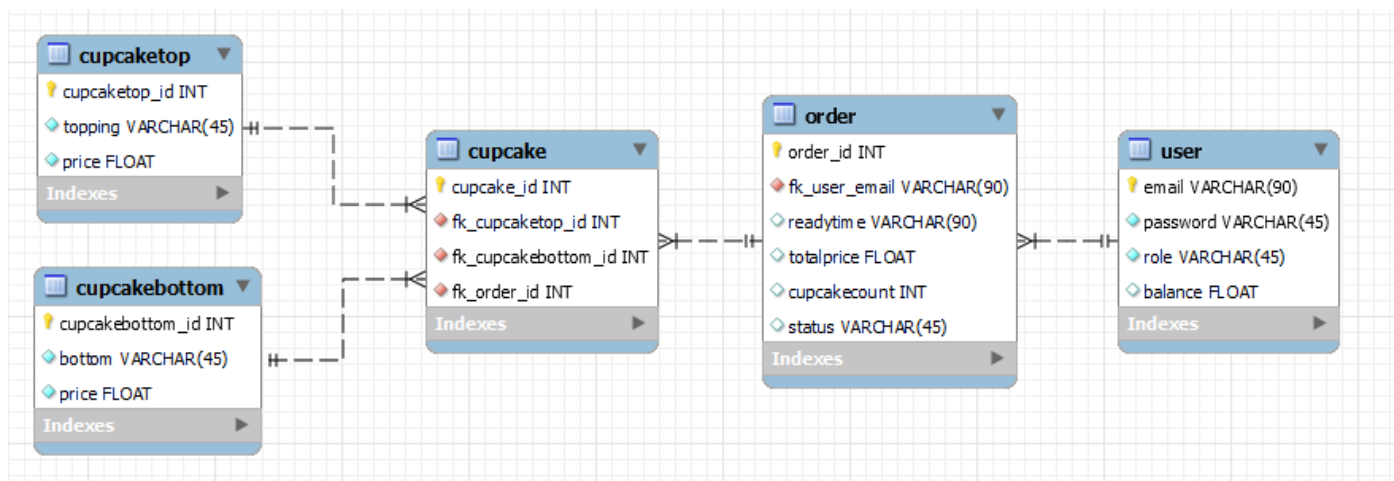
- **US-7:** Som administrator kan jeg se alle kunder i systemet og deres ordrer, sådan at jeg kan følge op på ordrer og holde styr på mine kunder.
- **US-8:** Som kunde kan jeg fjerne en ordre fra min indkøbskurv, så jeg kan justere min ordre.
- **US-9:** Som administrator kan jeg fjerne en ordre, så systemet ikke kommer til at indeholde ugyldige ordrer. F.eks. hvis kunden aldrig har betalt.

5 Domæne model og EER diagram

5.1 Domænemodel

Da projektet blev planlagt, blev der ikke designet en domænemodel. I stedet blev der fokuseret på at designe et EER-diagram og et navigations-diagram, da der var et ønske fra udviklerne om at få overblik over hvordan dataen var struktureret og hvordan siden kom til at se ud. Derfor blev et domænemodel anset som underordnet og indforstået fra de andre diagrammer.

5.2 EER diagram



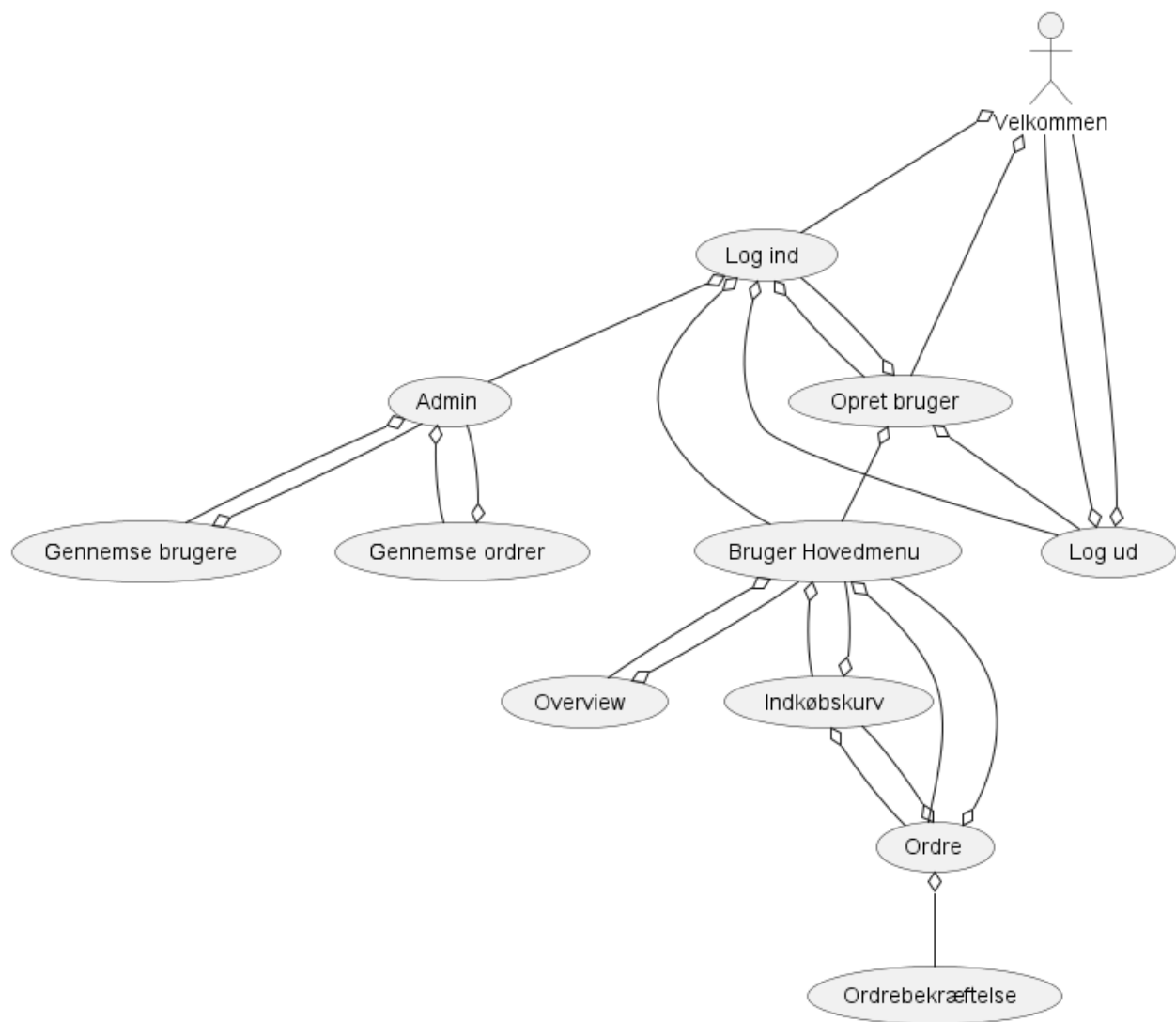
Figur 1: EER diagram

Vi har ikke direkte mange-til-mange tabeller, men vi har lavet en mellemtabel (cupcake) som samler cupcaketop og cupcakebottom med order.

De eneste steder vi kan argumentere for at vi bruger redundant data er i top og bottom. Siden både top og bottom har en ingrediens, ville man kunne lave en separat ingredient table. Dette anses dog som dårlig stil da de gerne vil kunne ses som individuelle elementer. Hvis de skal splittes ud ville de se ens ud, men de skal anses som individuelle elementer som bare har lignende data.

Det vil ikke give mening at lave en cupcake for hver type top bottom kombination, fordi så får vi redundant data. Hvis vi tilføjer en ny smag til listen så skal vi ikke til at rette flere steder i dataen, vi skal bare rette i en tabel. Derfor har vi lavet fremmednøgler som kan refereres til i cupcake.

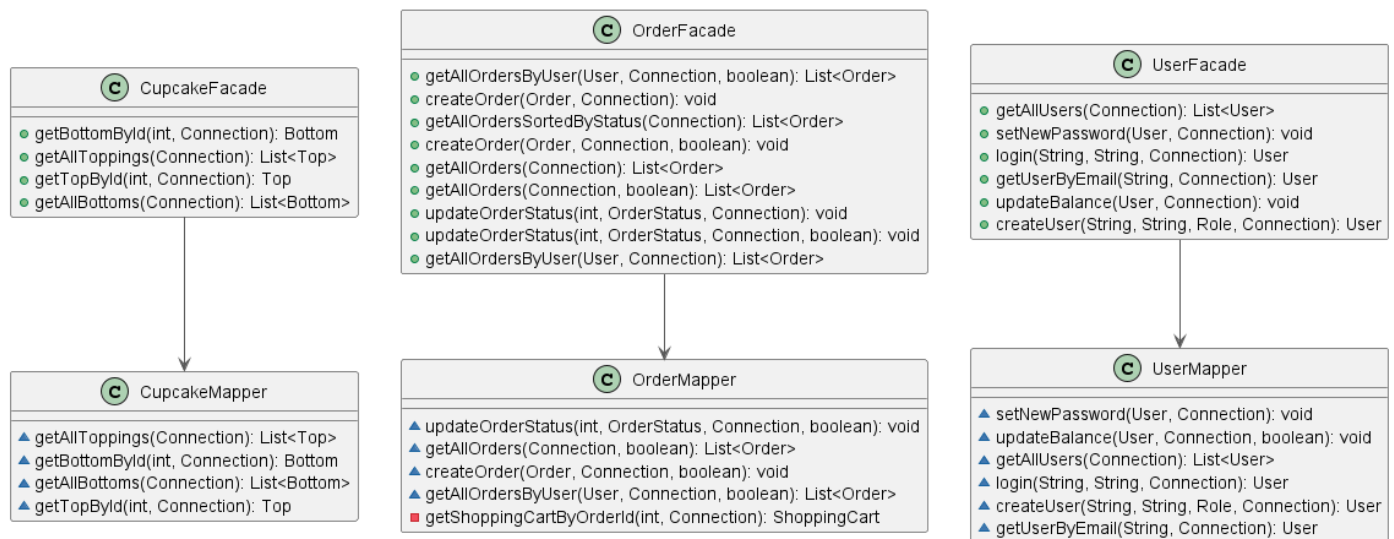
5.3 Navigationsdiagram



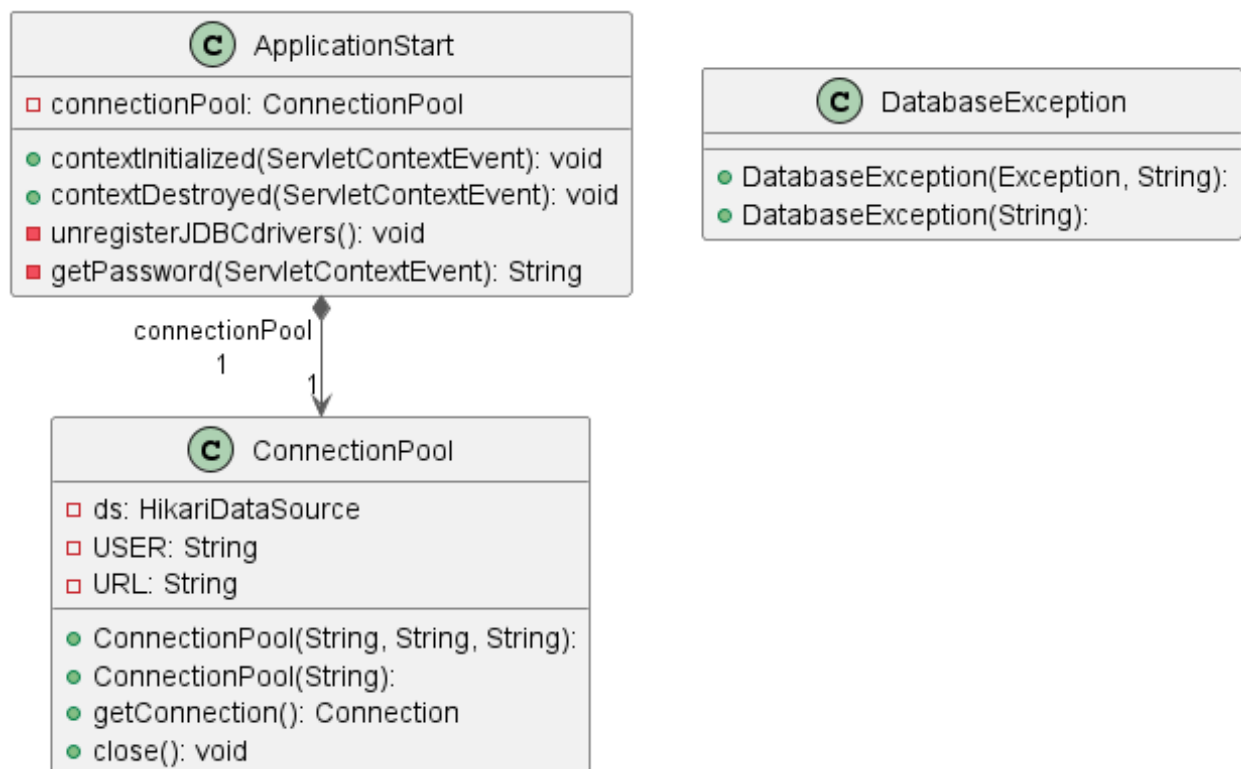
Figur 2: Navigationsdiagram over hjemmesiden

5.4 Klassediagram

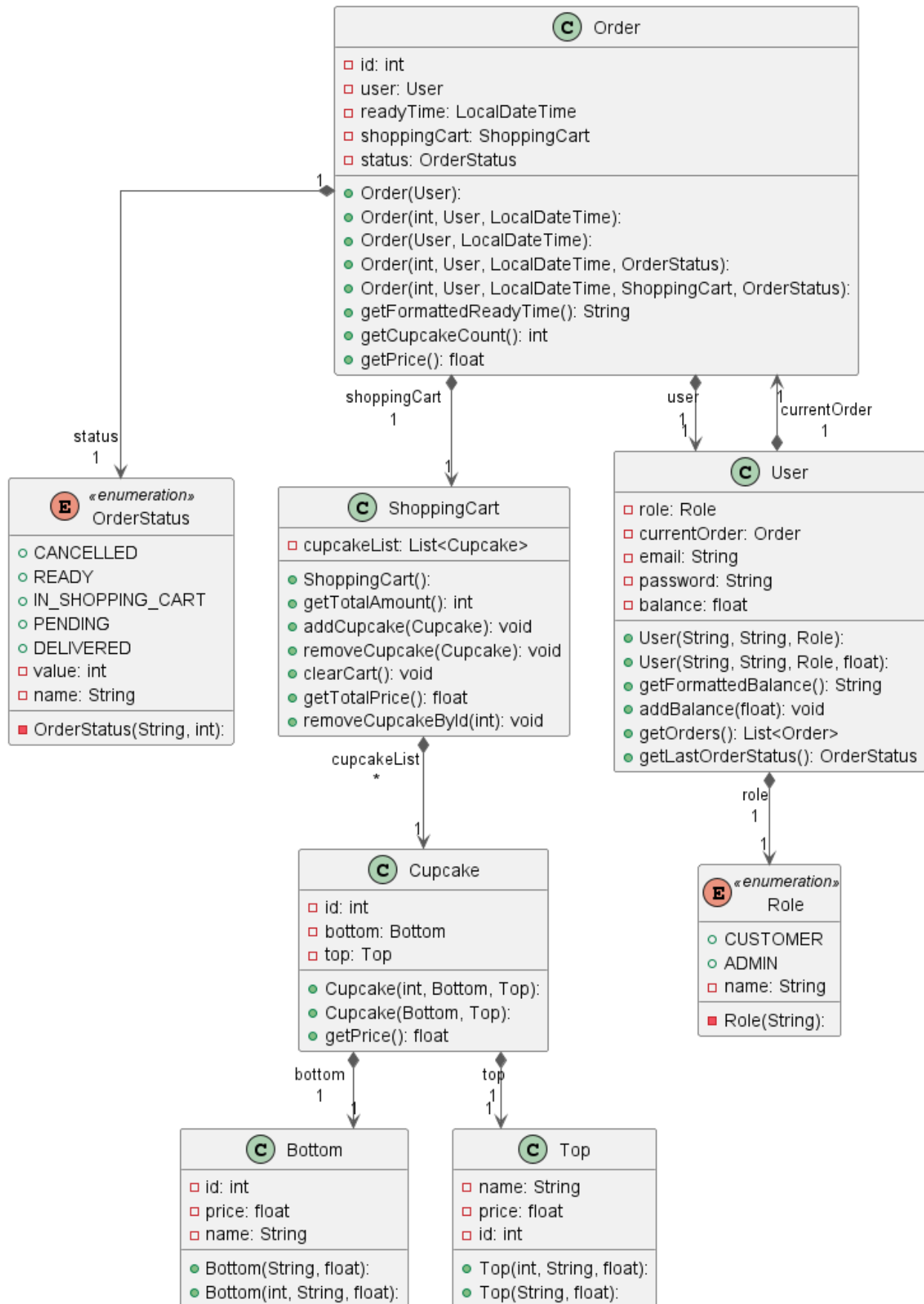
Vores kode består af en del klasser, med tilhørende mappers og facader. Vores klassediagrammer kan ses i figur 3, 4 og 5.



Figur 3: Klassediagram over Facade og tilsvarende Mappers



Figur 4: Klassediagram over Applicationsstart, ConnectionPool og Exceptions



Figur 5: Klassediagram over Entiteter

6 Særlige forhold

Til at starte med oprettede vi vores "readytime" i order-tabellen som datetime, men vi fik en hel masse problemer med at 'poste' dette til databasen, da java og sql ikke brugte samme datetime formater. I stedet oprettede vi "readytime" som en varchar og lavede eventuelle transformationer til datetime i java i stedet.

"order" er et reserveret keyword i MySQL, så det er vigtigt at specificere hver gang vi bruger order i et sql statement.

7 Status på implementation

Vi har færdiggjort alle user-stories og testet vores hjemmeside. Til næste projekt ville vi gerne arbejde lidt mere med design og lave SVG-generation så vi ville kunne lave billeder af kagerne når de bliver bygget.

8 Process

Vi benyttede Github Projects til at oprette tasks under de forskellige user stories. Vi lavede i fællesskab et navigationsdiagram og et EER diagram. Imens Sidsel lavede en mockup, oprettede Magnus databasen og Yusuf og Peter begyndte på back-end koden. Vi testede mockuppen, og da alle var tilfredse gik vi i gang med front-enden sammen med opdelt opgaver. I front-end koden forsøgte vi at fokusere på funktionaliteten først, så vi hurtigst muligt kunne teste hjemmesiden. Til at starte med, kom vi dog til at bruge lidt for meget tid på at gøre det "lækkert", så vi endte i en situation hvor vi måtte vente på front-enden for at kunne teste. Det var ikke optimalt og noget vi vil forbedre til næste projekt.

I vores gruppe fungerer det rigtig godt at fordele opgaverne ud, så 1-2 personer sidder og arbejder på én del. I forhold til vores SP4 projekt på første semester, har vi denne gang fokuseret rigtig meget på at præsentere vores arbejde for hinanden så alle ved hvordan koden fungerer. Det gør vi i form af møder, som vi altid drøfter før vi starter, halvvejs og afrundende til slut.

9 Videopræsentation

Her er et youtube-link til vores præsentation

<https://youtu.be/N5TwB1fWGKw>