

MULTIMEDIA DESIGN - 3.SEMESTER 1. PROJEKT

DATABASE MODELLERING



FAKTAARK

JONAS FALK

sniller27@hotmail.com

JONAS FALK

CHRISTIAN BRAMS

halkjaer-brams@hotmail.com

Christian

LINE RASMUSSEN

line-rasmussen@live.com

Line R.

AUGUSTA NAUNDRUP

augusta@naundrup.net

Augusta Naundrup

SNEZANA ILIC

snezana.ilic@outlook.com

Snezana

CLmul-b14e Gruppe 2
2. Database projekt

Projekt vejledere

Ivan Rosenvinge Frederiksen

Tue Becher

INDHOLDSFORTEGNELSE

INTRODUKTION	3
VIRKSOMHEDEN	4
PBS	4
WBS	5
GANTT SKEMA	5
ER DIAGRAM	6
USE CASE	8
USE CASE MODEL	10
CRUD MATRIX	11
ATTRIBUT SKEMA	13
ARBEJDSFORM	14

INTRODUKTION

I dette projekt skulle vi udvælge et velkendt firma, som skulle ligge til grundlag for vores projekt. Efterfølgende skulle vi udarbejde en e-shop til den valgte virksomhed, hvor vi, i dette projekt, udelukkende skulle lægge vægt på data-modeling.

Det vil sige, at vi ikke skulle lægge vægt på interface eller andre design-principper.

Derfor har vi i dette projekt arbejdet med ER-diagram, Use Case, Use Case-model, Attribut-skema og til sidst CRUD-matrix.

VIRKSOMHEDEN

Vi har valgt at arbejde med den danske tøjvirksomhed Buch Copenhagen. Tøjkæden har siden sin opstart i 2009 solgt tøj til kvinder med fokus på tøjstilen "bohemian".

Buch Copenhagen har 4 fysiske butikker i Danmark:

København Ø
Lyngby
Frederiksberg
Hørsholm

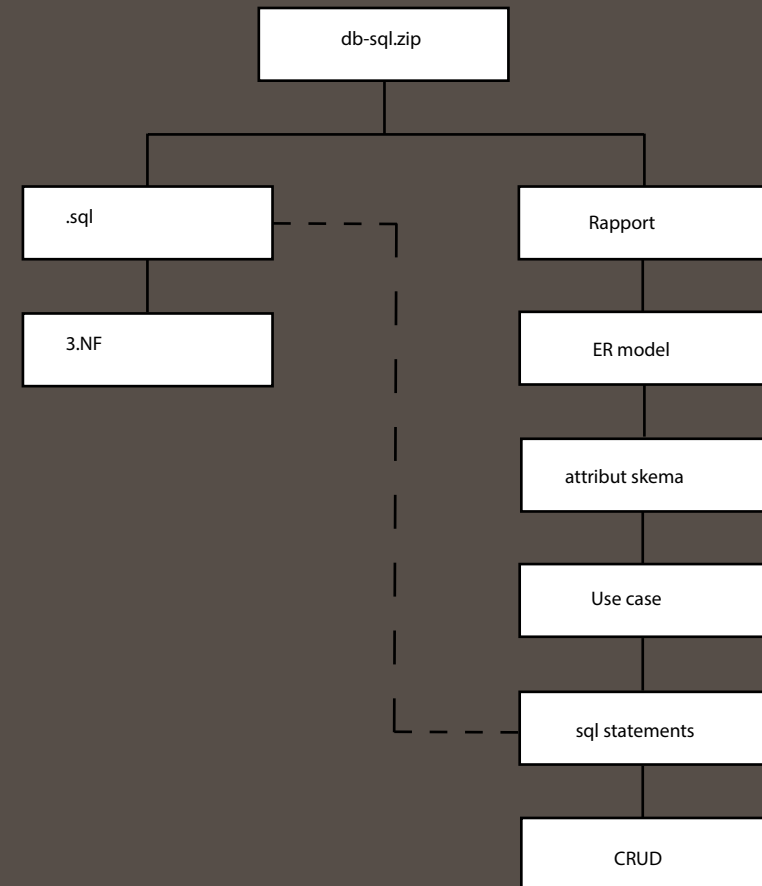
På Buch Copenhagen's Facebook-side fremgår det, at de har en hjemmeside - www.buchcopenhagen.dk. Denne side er dog blot et link tilbage til deres Facebook-side.

Vi undersøgte derfor, om vi kunne finde en reel hjemmeside for virksomheden, men uden held.

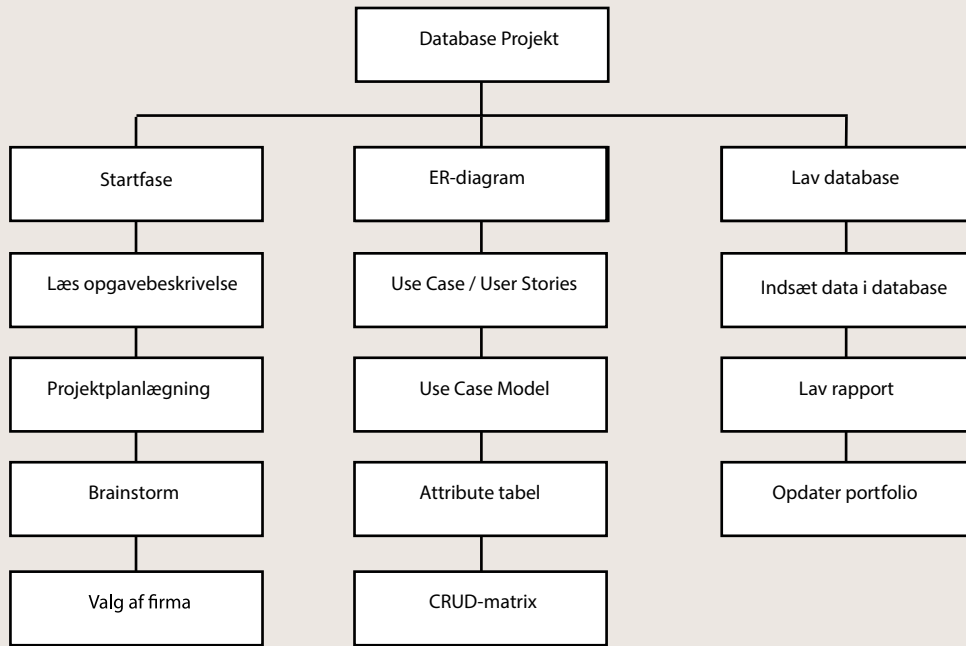
Vi kan konkludere, at Buch Copenhagen på nuværende tidspunkt ingen funktionel hjemmeside eller e-shop har. Dermed har virksomhedens kunder kun én mulighed for at købe deres tøj eller se hele deres kollektion - ved at besøge en af de 4 fysiske butikker. Derfor kunne man derfor forestille sig, at firmaets tøj ikke er særligt udbredt andre steder end i hovedstads området.

Ud fra de ovennævnte observationer, ser vi derfor virksomheden som et oplagt valg til dette projekt, da vi på et senere tidspunkt ville have stort set frie tøjler indenfor design af hjemmesiden, og ikke mindst deres e-shop.

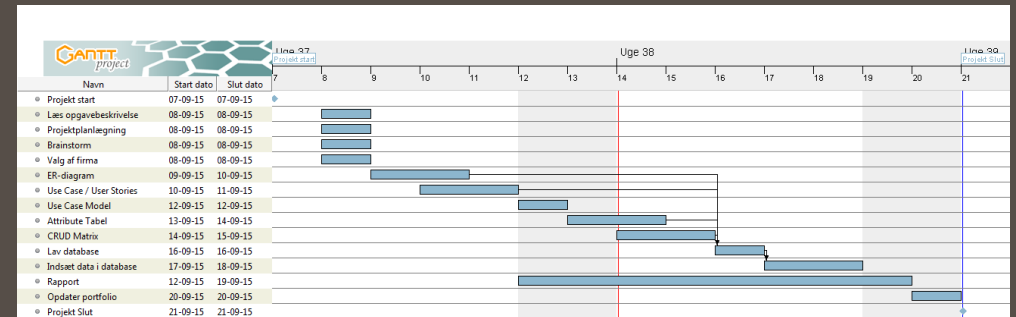
PBS



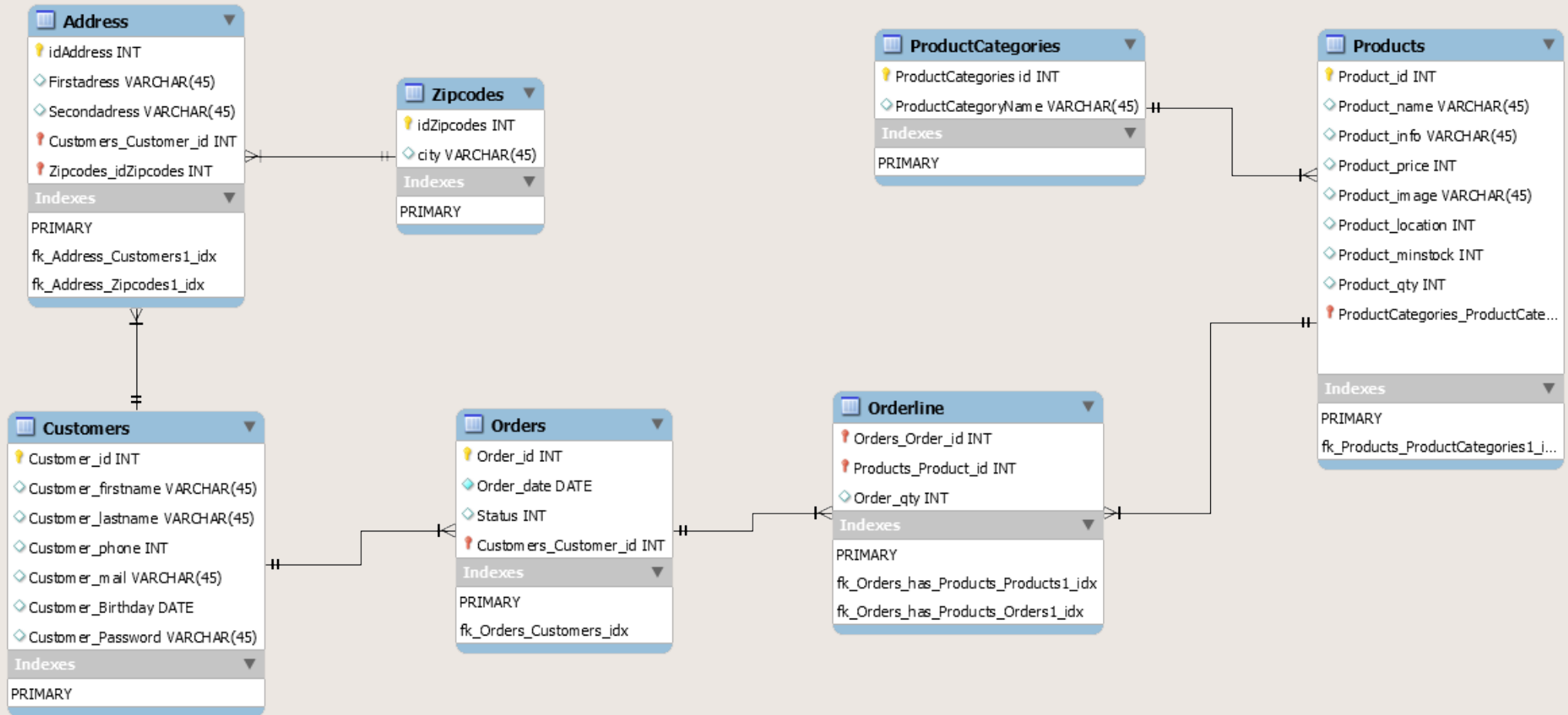
WBS



GANTT SKEMA



ER-DIAGRAM



ER-DIAGRAM

ER-diagrammet viser, hvordan vores database er opbygget. Kasserne i ER-diagrammet er entiteter, som er databasens tabeller. Inde i entiteterne kan vi se attributterne og deres datatyper, som indikerer hvilken data, tabellerne indeholder. Stregerne mellem entiteterne er relationer/cardinaliteter, som viser forholdet mellem entiteterne. Cardinaliteterne har vi valgt at illustrere med Crow's Foot notation. Hver tabel i databasen har en unik primærnøgle, som illustreres ved en gul nøgle. Primærnøglerne sørger for at alle rækkerne entydigt kan identificeres, og tabellerne er forskellige fra hinanden. Fremmednøglerne i tabellerne er illustreret ved en rød nøgle, som forbinder tabellerne, ved at fremmednøglen peger på en primærnøgle i en af de andre tabeller.

I databasen har vi følgende tabeller: Zipcodes, Address, Customers, Orders, Orderline, Products, ProductCategories.

I databasen kan en kunde oprettes med én eller flere adresser, afhængigt af hvor produktet skal sendes hen. Hver adresse kan have et postnummer (zipcode), mens et postnummer sagtens kan have flere adresser. Hver kunde kan have én eller flere ordrer, mens hver ordre kun er tilknyttet én kunde. Ordrene i Order-tabellen fungerer først og fremmest som indkøbskurv i e-shoppen, og bliver først til en egentlig ordre, når kunden gennemfører betalingen. Ved betaling ændres Status fra 0 til 1 i Order-tabellen. Umiddelbart er en eller flere produkter tilknyttet en eller flere ordrer. For at undgå en "mange-til-mange"-relation mellem Orders- og Products-tabellerne, har vi lavet en ordrelinje (Orderline-tabellen).

Denne binder to fremmednøgler sammen. De sammensatte fremmednøgler i Orderline-tabellen fungerer som én samlet primærnøgle. Produkterne i e-shoppen har også en produktkategori, som er tilknyttet til ProductCategory-tabellen.

Normalisering

For at kunne bruge databasen lettere og mere effektivt, har vi normaliseret databasen. Normalisering er en måde at organisere databaser på. Ved normalisering bliver databasen mere fleksibel, og det bliver lettere at indsætte, ændre og slette data. Man undgår også redundans i tabellerne, altså data der står flere steder. Dette sparer i sidste ende plads i databasen. Der ud over bliver data som regel også grupperet på en mere logisk måde. Vi har normaliseret databasen til 3. normalform, hvilket vi har gjort ved at opfylde følgende retningslinjer:

1NF: Der må ikke være information i rækker, der gentager sig (redundans).

2NF: For at 2NF kan opfyldes skal 1NF være opfyldt, og alle attributter, der ikke er en del af en sammensat primærnøgle, skal være afhængige af hele den sammensatte primærnøgle.

3NF: For at 3NF kan opfyldes, skal 2NF være opfyldt, og alle attributter skal direkte afhænge af primærnøglen.

USE CASE

Før vi laver databasen, har vi lavet use cases som en del af de funktionelle krav til systemet. Vores use cases viser, hvilke krav der er til interaktionen med Buch Copenhagen's shopping system. Vi har lavet forskellige use cases for både bruger og administrator, som er aktørerne.

Use Case	UC01
Use Case Navn	Registrer bruger og login
Beskrivelse	Vores use case viser, hvordan man registrerer sig på Buch Copenhagen's hjemmeside og logger ind.
Betingelser/antagelser	Målet for vores use case er at klargøre, hvordan man som bruger registrerer sig på Buch Copenhagen's hjemmeside, samt hvordan man efterfølgende logger ind.
Frekvens	Brugeren kan finde ud af at bruge en hjemmeside, og har kendskab til systemer og sider, hvor man kan registrere sig. Det kræves at brugeren har en mail, som vedkommende kan modtage informationer på.
Fremgangsmåde	Denne proces bliver gennemgået flere gange dagligt.

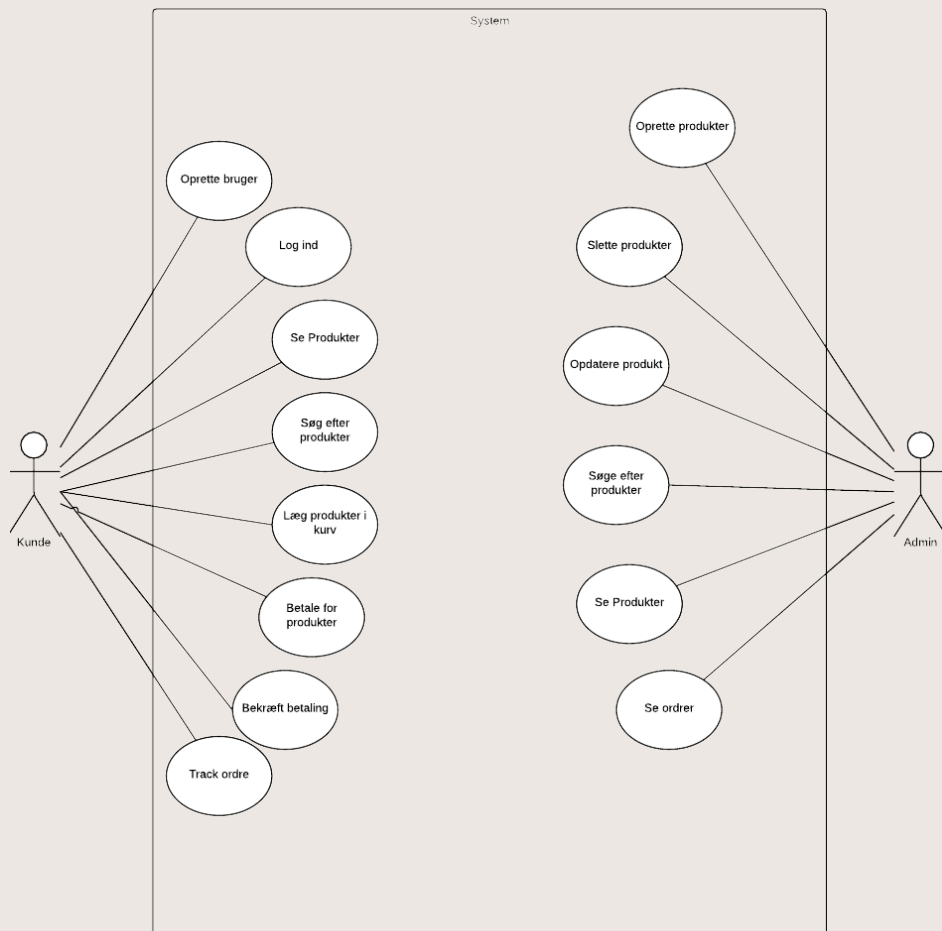
Alternativt flow	<ol style="list-style-type: none">1. Brugeren går til Buch Copenhagen's hjemmeside.2. (Brugeren går ind på registrerings-siden)3. Brugeren indtaster personinformationer, e-mail og password, og klikker på registrer.4. Brugeren får feedback på sin tilmelding og får tilsendt en mail.5. (Brugeren går til login-siden)6. Brugeren indtaster brugernavn og password og klikker login.7. Brugeren sendes til forsiden.
Udfald	<ol style="list-style-type: none">1. Brugeren forsøger at logge ind på hjemmesiden og indtaster forkert brugernavn/password.2. Brugeren får feedback om, at brugernavn/password er forkert.3. Brugeren klikker på "Glemt password?" og indtaster e-mail.4. Brugeren tjekker sin e-mail, hvor kodeordet står.5. Kunden logger ind.
Udfald	Vores use case slutter med, at kunden har fået en bekræftelsesmail vedr. sin tilmelding til Buch Copenhagen. Kunden er også blevet registreret i databasen.
Aktører	Bruger/kunde
Fortsættelse af use case	Vores use case fortsættes af "UC02"

Use Case	UC02
Use Case Navn	Køb i e-shop
Beskrivelse	Denne use case viser, hvordan brugeren køber et produkt i Buch Copenhagen's e-shop
Mål	Målet for vores use case er, at brugeren skal være i stand til at købe et produkt fra e-shoppen.
Betingelser/antagelser	Brugeren kan finde ud af at benytte en hjemmeside, og indkøbskurven er tom. Brugeren har også et kreditkort. Der er tilføjet produkter i e-shoppen. Det antages at brugeren er logget ind.
Frekvens	Denne proces bliver gennemgået flere gange dagligt.
Fremgangsmåde	<ol style="list-style-type: none"> 1. Brugeren går til shoppen 2. Brugeren vælger en kategori 3. Brugeren vælger produkt 4. Brugeren lægger produkt i indkøbskurven 5. Brugeren går til kassen 6. Brugeren betaler 7. Brugeren får respons på sit køb, samt en kvittering
Alternativt flow	1. Brugeren vælger produkt på forsiden i stedet for i kategorierne eller benytter sig af søgefunktion.
Udfald	Vores use case slutter med at kunden har købt et produkt, som er blevet registreret som en ordre i databasen.

Use Case	UC03
Use Case Navn	Indsætte produkt på website
Beskrivelse	Vores use case viser, hvilke trin man som administrator skal igennem, for at indsætte et produkt i databasen, og derefter på Buch Copenhagen's e-shop.
Mål	Målet er, at administratoren indsætter et produkt på Buch Copenhagen's hjemmeside, ved først at tilføje produktet samt informationer i databasen.
Betingelser/antagelser	Administratoren har Buch Copenhagen's database og webserver tilgængelig, samt deres login. Administratoren ved, hvilket produkt og produktinformation, der skal indsættes i databasen. Databasen indeholder ikke nogen nye produkter endnu, og kan ikke ses på Buch Copenhagen's e-shop.
Frekvens	Denne proces bliver gennemgået flere gange månedligt. (Når der er nye produkter)
Fremgangsmåde	<ol style="list-style-type: none"> 1. Administratoren logger ind i Buch Copenhagen's database. 2. Administratoren går til Product-tabellen og indtaster produktinformation via SQL. 3. Administratoren tjekker om produktet er tilføjet i tabellen. 4. Administratoren logger ind på webserveren og tilføjer evt. produktet via PHP-filer.

Alternativt flow	<ol style="list-style-type: none"> 1. Administratoren logger ind i Buch Copenhagen's database. 2. Administratoren går til 'Produkt' tabel, og indtaster produkt og produktinformation via SQL. 5. Administratoren tjekker om produktet er tilføjet i tabellen. 6. Administratoren opdager indtastningsfejl. 7. Administratoren ændrer tastefejlen vha. SQL Update Statement. 8. Administratoren logger ind på webserveren og tilføjer evt. produktet via PHP-filer.
Udfald	Vores use case slutter med, at et eller flere produkter er tilføjet til Buch Copenhagen's database og kan ses på Buch Copenhagen's e-shop.
Aktører	Administrator

USE CASE MODEL



En use case model beskriver krav til et system. Modellen kan indeholde flere scenarier, der viser interaktion mellem bruger, system og administrator, samt hvordan specifikke opgaver løses.

I modellen figurerer brugeren altid til venstre og administratoren til højre. Mellem dem er scenen, hvor de forskellige funktioner er vist i bobler.

I vores e-shop skal brugeren kunne se og søge efter produkter, lægge varer i kurven, samt have mulighed for at rette kurvens indhold, inden den bliver til en ordre og betalingen bekræftes.

Brugeren skal være logget på, før vedkommende kan gå videre til betaling, og derfor også have muligheden for at oprette sig som bruger før betalingen.

Der skal efterfølgende være mulighed for at tjekke ordrestatus.

Administrator skal have alle rettigheder, og kunne oprette, læse, ændre og slette alt i systemet.

CRUD-MATRIX

1 á 2

User Action	Produkt category	Products	Orderline	Orders	Customers	Adresse	Zipcodes
KØB							
- Forside, Buch Copenhagen	R	R					
- Vælger kategori	R	RU					
- Vælger produkt	R	R					
- Lægger produkt i indkøbskurven				CU			
BETALING - MEDLEM							
-Log in					R	R	R
- Gå til kassen			RUD				

CRUD-MATRIX

2 á 2

User Action	Produkt category	Products	Orderline	Orders	Customers	Adresse	Zipcodes
KØB							
- Forside, Buch Copenhagen	R	R					
- Vælger kategori	R	RU					
- Vælger produkt	R	R					
- Lægger produkt i indkøbskurven				CU			
BETALING - MEDLEM							
-Log in					R	R	R
- Gå til kassen			RUD				

I vores CRUD-matrix illustrerer vi, hvilke funktioner SQL bruger ud fra de forskellige 'user actions' der bliver foretaget - både af kunder, men også af administratoren.

ATTRIBUT SKEMA

I attribut-skemaet kan man få overblik over tabellerne, hvad de indeholder, samt hvad attributterne må og kan indeholde. Som eksempel kan vi se, at 'Customer Lastname'-attributten kan indeholde alle store og små bogstaver fra a-å, men må kun indeholde op til 30 tegn.

Entity	Attributes	Value	Notes	Datatype Num ./ Alfanum
Customer:	Customer_ID	1-Z	Unique no.	AN
	Custom. firstname	a-Å	Max. 30 char.	AN
	Custom. lastname	a-Å	Max.30 char	AN
	Mobil	0-9	Max. 10 char	N
	Custom. Email	a-Å	Max. 50 char.	AN
	Custom. Address	All char.	Max. 35 char.	AN
	Birthday Date	0-9	Max. 8 char.	N
	Password	0-9, a-Z	Max. 14 char.	AN
Adress:	Zip codes	1000-9999	Number	N
	Street	a-Å	Max. 30 char.	AN

Ordre:	Ordre_ID	1-Z		N
	Ordre_Data		Unique no.	N
	Status			
	Zip codes	1000-9999		N
Orderline:	Orders Order ID	1-Z		AN
	Products Product ID	1-Z	Unique no.	AN
	Order_qty	1-Z		AN
Produkt:	Produkt_ID	1-Z	Unique no.	AN
	Produkt_name	a-Å	Max 30 char	AN
	Produkt_info			
	Produkt_price	0-9	Number	N
	Produkt Image	Img		
	Produkt location			
	Produkt minstock			
	Produkt_Qtv			
	ProduktCategories	a-Å	Max. 20 char	AN
Category:	ProduktCategories ID	1-Z	Unique no.	AN
	ProduktCategories Name	a-Å	Max. 30 char.	AN

ARBEJDSFORM

Vi har valgt at arbejde med traditionel projektplanlægning, hvor vi starter med at lave Gantt-kort, WBS og PBS.

Dog er der dele af projektet, der er udarbejdet ud fra vandfaldsmodellen - ER-diagrammet skulle færdiggøres, inden vi kunne påbegynde arbejdet med CRUD-matrixen.

Vores vurdering var dog, at det ikke ville være optimalt at benytte vandfaldsmodellen fast igennem projektet, da vi vidste, at vi flere gange ville vende tilbage til allerede færdige arbejdsopgaver for at rette.

Derfor har vi primært arbejdet ud fra vores Gantt-kort gennem hele projektet.