Rapport – Webshop med back end

Eksamensprojekt i Webprogrammering Server Side

Udarbejdet af Stine B. Holt og Marie B. S. Therkelsen

Hold: cs16auwssea3 – CPH business Academy

Afleveret d. 23. December 2016 på [www.wiseflow.dk](http://www.wiseflow.dk)



Indholdsfortegnelse

Projektbeskrivelse 3

Sidens funktioner 4

Arkitektur 4

MVC-model 8

Databasen 8

Sikkerhed og performance 9

Konklusion 10

Huskeliste til opsætning af webshop I Netbeans 10

# Projektbeskrivelse

Vi har fået til opgave at programmere en webshop udført i NetBeans, med backend i form af en database der er udført i MySQL.

Kravene til webshoppen er at den skal indeholde følgende funktioner:

* Vise de tilgængelige vare hvor der er flere detaljer omkring produktet
* Login
* Registrering af nye brugere med for/efternavn, e-mail, mobil og adresse
* Kurv side som vise hvilke varer, der er lagt i kurven
* Check ud funktionalitet hvor folk kan ”betale”, dog uden den sidste side med kreditoplysninger

De tekniske krav til webshoppen:

* Data skal persisteres i en database
* Der skal bruges MVC (model-view-controller design) lagdeling, hvor:
  + Præsentationslaget(view) laves med JSP
  + Kontrollaget laves med Serlets
  + Modellaget laves med javaklasser
* JSP siderne kan gøre brug af expression language og JSTL core biblioteket eller blot bruge scriptlets
* Login skal gemmes i en session således at brugeren er logget ind på tværs af sider

Indholdet af rapporten har vi valgt at holde forholdsvis komprimeret, da vi har valgt at lave en kort beskrivelse af alle vores lag (jsp-sider, servlets og java-klasser), der henvises derfor til kildekoden, hvis der er ønske om at se koden for den enkelte metode.

Vores webshop sælger plakater, i forskellige kategorier med levering. Her kan kunderne altså købe en flot plakat til hjemmet og få den leveret lige til døren.

Som kunde har man sin egen personlige konto på webshoppen, derfor kræver det et login med brugernavn og password, for at kunne handle på sitet.

## Sidens funktioner

Når man ønsker at tilgå webshoppen bliver man mødt af en login-side, her logger man enten ind med sit brugernavn og password eller man klikker på knappen ”Sign in”, som videresender dig til en side hvor det er muligt at oprette sin egen personlige bruger på webshoppen, når alle oplysninger er udfyldt får man adgang til webshoppen, ved at indtaste sit nylavet brugernavn og password, som så fører dig til webshoppen efter indtastning.

Her bliver man mødt af en pæn og overskuelig side, hvor der er fremvist billeder af plakater, navnet på plakaten, kategori og pris på plakaten. Hvis man har et ønske om at få flere detaljer om plakaten kan man klikke på knappen ”Se mere”, som findes under hvert produkt. Her sendes man til en side som indeholder yderligere information om plakaten. Herfra kan man tilføje sit produkt til indkøbskurven eller gå tilbage til oversigtssiden med alle plakaterne.

Hver gang man tilføjer et produkt til sin indkøbskurv, vises der en advisering der informere om at der nu er tilføjet et produkt til kurven. Når man så har tilføjet alle de plakater man ønsker at købe klikkes der på knappen ”Indkøbskurv”, her vises alle de plakater man har tilføjet til kurven, med navn, pris og ikke mindst summen af alle plakater i indkøbskurven. Herefter kan man så klikke på ”bestil ordre”, hvor man så sendes tilbage til webshoppens oversigtsside og der er en advisering i højre hjørne, om at ordren nu er bestilt.

Udover disse funktioner viser webshoppen også hvem man er logget ind som og har en log-out funktion.

## Arkitektur

Vores webshop er programmeret efter MVC-modellen, som er et ”design pattern” der er baseret på hændelser, hvilket vil sige at en indtastning i brugergrænsefalden f.eks. resulterer i en hændelse til controlleren, som videreformidler kaldet ned til model-laget, hvor efter kaldet bliver returneret med forespørgselen på view-laget. Kort sagt foregår kommunikationen mellem datamodellen og brugergrænsefalden via en controller der viderefører de forskellige kald og returnerer f.eks., data fra modellaget.   
Ved at anvende MVC-modellen er det let at vedligeholde koden, da tingene er opdelt i de forskellige lag og derudover opsat overskueligt i kode strukturen.

Opdelingen af MVC:

* **View’et** indeholder de visuelle elementer(brugergrænsefladen)
* **Controlleren** håndterer logikken via servlets i dette tilfælde
* **Modellen** håndterer data fra f.eks. en SQL databasen eller andre datakilder

View-laget indeholder brugergrænsefladerne for de forskellige views via

jps-sider:

* **Login.jsp** – Brugeren logger ind og bliver sendt videre til login servletten (loginservelt.java) og link til oprettelse af en ny bruger på CreateNewUser.jsp
* **CreateNewUser.jsp** –Brugergrænsefladen til oprettelse af en ny bruger hvori parametrene fornavn, efternavn, e-mail, mobil og adresse indtastes. Den nye bruger bliver oprettet via createNewUserServlet.java som også tilføjer de oplysninger, som brugeren indtaster på siden ned i databasen.
* **Produkt.jsp** – Brugergrænsefladen for visning af alle produkter og muligheden for at tilføje dem til indkøbskurven, når et produkt er valgt benyttes servletten BuyProdukt.java
* **ProduktDetails.jsp** – Viser listen med flere detaljer omkring den plakat man har klikket ”se mere” under, her benyttes productDetail.java til at hente de ekstra detaljer op fra databasen.
* **PurchaseKurv.jsp** – viser listen med de valgte varer og pris pr. plakat. Her vises summen af prisen på alle de valgte varer og ved klik på ”Bestil ordre” anvendes Placeorder.java.

Control-laget indeholder servlets og javaklasser:

* **Loginservlet.java** – Authenticate brugeren via UserMapper.java (metoden authentcateUser),

IF authenticated

Hentes listen med produkterne fra showProduct.java  
Else

Sendes en besked til brugeren om at de ikke er logget ind og der er mulighed for at oprette en ny bruger via createNewUser.jsp

* **CreateNewUserServlet.java** - Benytter UserMapper.java til at oprette nye brugere, og herefter sender brugeren til login.jsp efter oprettelse.
* **ProductDetail**.**java** –Benytter ProduktMapper.java til at fremvise det valgte produkts yderligere detaljer på ProduktDetails.jsp siden.
* **BuyProdukt.java** – Placerer varerne i indkøbskurven og trækker parametrene fra requestet(productID, UserID).
* **ShowPurchaseServlet.java** – Generere en liste for brugerens køb (getAllPurchaseByUser), når submit action fra indkøbskurven bliver sendt placeres ordren via placeorder.java
* **ShowProducts.java** – Generere en liste over alle produkterne på webshoppen via ProduktMapper.java
* **PlaceOrder.java** – Håndterer bestilling af ordren via OrderMapper.java

Model-laget indeholder vores mappers og dataacces funktioner:

* **DB.java** – Som er vores database connection til webshoppens databasen.
* **UserMapper** – Indeholder flere vores metoder omkring brugere og login;
  + authenticateUser – Funktion til at logge ind med eksisterende bruger.
  + createUser – Opretter nye brugere
  + getUser – Vælger den ind loggede bruger i forbindelse med køb af en vare.
* **ProduktMapper** – Indeholder vores metoder i forbindelse med visning af produkter/varer og detaljer omkring disse.
  + getAllProducts – Indeholder listen med alle vores produkter
  + getProduct – Indeholder en detaljeret side om alle produkterne
* **PurchaseMapper** – Indeholder metoder omkring indkøbskurven
  + placeProduktForPurchase– Placerer valgte varer i indkøbskurven
  + getAllPurchaseByUser – Viser indkøbskurven/listen med alle valgte varer, herunder produkt Id, produkt navn, antal og prisen
  + placeOrder – Submitter alle varer i indkøbskurven, (her ville man i andre tilfælde, blive sendt videre til en betalingsside, men den del skal vi ikke lave).
* **PasswordStroage.java –** Indeholder vores SALT og HASHING metoder til at gøre brugerens password krypteret i databsen.

## MVC-model

Purchasekurv.jsp

ProduktDetails.jsp

produktside.jsp

createNewUser.jsp

Login.jsp

View-laget

Model-laget

Control-laget

User.java

UserMapper.java

ProductByOrder.java

Product.java

Order.java

ProduktMapper.java

PasswordStorage.java

OrderMapper.java

DB.java

placeOrder.java

showPurchaseServlet.java

createNewUserServlet.java

BuyProduct.java

showProducts.java

ProductDeail.java

loginServlet.java

## Databasen

Vores database er udarbejdet i MySQL og opdelingen er:

I vores database har vi 5 tabeller;

* **Image –** Bruges ikke, men kunne indeholde billederne der vises i brugergrænsefladen ud fra hver vare.
* **Produkt –** Indeholder alle vores produkter(plakater) og detaljerne omkring disse;
  + Id
  + Name
  + Kategori
  + Description
  + Size
  + Pris
  + Leveringstid
  + Image
* **User –** Indeholder alle brugerprofiler med oplysninger om brugerne;
  + ID
  + Username
  + Password
  + FName
  + LName
  + Email
  + Mobile
  + Adresse
  + Postnummer
* **Ordertable –** Indeholder alle ordre der er lavet på webshoppen af brugerne;
  + OrdreID
  + Username
  + Ordredate
  + Ordrebestilt
* **produktONorder –** Indeholder de produkter der er lavet en odre på
  + ProduktID
  + OrdreID

Derudover har vi lavet nogle SQL-sætninger som viser ”data pr. ordre” og ”summen af ordrens pris for en bruger”, som vises på PurchaseKurv-siden.

# Sikkerhed og performance

I forhold til sikkerhed har vi saltet og hashet vores password i databasen, hvilke vil sige at man ikke kan læse de passwords der kommer ind når brugerne opretter sig på webshoppen, men i databasen vises de som en række tal og bokstaver. Ved at bruge salting og hashing undgår man at fremmedpersoner vil kunne skaffe sig adgang til databasen, hvor alle webshoppens brugeres brugernavne og passwords ligger gemt i klartekst, ved at hashe krypterer man passwords så de ligger i talkoder der ikke kan bruges som passwords.

Derudover kunne man sikre webshoppen for man-in-the-middel angreb, som ville kunne gøres ved at benytte HTTPS-kryptering, som er en kommunikation over HTTP, der er en protokol der bruges til kommunikation på World Wide Web (WWW), dog uden en krypteret forbindelse hvorimod HTTPS er en krypteret forbindelse der beskytter mod et angreb hvor en tredjeperson udgiver sig for at være afsender/modtager. Dermed kan tredjepersonen ændre på kommunikationen mellem afsender og modtager, f.eks. en bruger og en server, der tror de kommunikerer med hinanden og ved at tredjepersonen udgiver sig som en af parterne lokkes der ofte personfølsomme oplysninger ud af en af parterne der vil blive misbrugt af tredjepersonen.

# Konklusion

Alt i alt fungerer vores webshop efter hensigten, koden er letlæselig og ordentlig struktureret. Vi har opfyld alle de obligatoriske krav der var stillet til projektet og rapporten, vi er tilfredse med resultatet af projektet.

Med hensyn til udviklingen af projektet, startede vi med at lave en oversigt over hvad de forskellige lag skulle indeholde og derefter startede udviklingen. Det gav os en god start på projektet med overblikket over hvad vi skulle nå frem til kodemæssigt og hvilke metoder der var behov for.

## Huskeliste til opsætning af webshop I Netbeans

For at opsætte webshoppen på din egen PC er det vigtigt at huske følgende:

* Tilføj filmappen WEB.INF i Netbeans under Web pages i projektet.
* Skriv dit eget personlige password i DB.java i stedet for ”nice”