IT-DIPLOMUDDANNELSEN

DATABASESYSTEMER OG WEB

Eksamensprojekt

af

Jan Schrøder Hansen

eFTERÅR 2010

**INDHOLD**

[1. Indledning 3](#_Toc277421985)

[2. Opgavebeskrivelse 3](#_Toc277421986)

[3. ER Diagram 3](#_Toc277421987)

[4. Konklusion 4](#_Toc277421988)

[5. Bilag 5](#_Toc277421989)

[5.1. Kode 5](#_Toc277421990)

[5.2. Indhold på vedlagte CD 5](#_Toc277421991)

# Indledning

Dette eksamensprojekt er lavet i forbindelse med faget Web og serverprogrammering på IT- Diplomuddannelsen, Ingeniørhøjskolen i København.

Faget har taget udgangspunkt i bøgerne ”Database Management Systems” af Raghu Ramakrishnan og Johannes Gehrke og ”SQL i praksis” af Ben Forta. Samt materiale udarbejdet af underviser Torben Lund.

# Opgavebeskrivelse

For at komme igennem så meget af materialet i faget som muligt, har jeg valgt at lave en webløsning til håndtering af ønsker og fejlreporter, til et eller flere softwareprodukter.

Der skal være mulighed for at følge status på fejl og ønsker. Er fejlen eller ønsket godkendt, afvist, er det under udvikling, under test etc. Derudover skal der være mulighed for at uploade forskellige filer såsom skærmdumps af fejl, eller prototyper på ny skærmlayouts og andre filer som kan hjælpe til at belyse en sag.

Programmets navn er BaRI, som står for ”Bugs and Request Interceptor”.

Det er en løsning jeg startede på under faget Web og Serverprogrammering, efteråret 2009, rapporten kan findes på den vedlagte cd. I denne opgave var mit fokus på et web framework kaldet Wicket[[1]](#footnote-2) . En lille del af opgaven skulle også demonstrere at løsningen kunne persistere data i en database. Så der blev lavet 2 tabeller. I denne opgave vil jeg forsætte på denne løsning, men med fokus på database delen.

Programmet udvikles i sproget Java[[2]](#footnote-3). Med Java følger databasen JavaDB[[3]](#footnote-4), derudover benyttes Hibernate[[4]](#footnote-5) som bro mellem den relationelle database verden og den objekt orienteret verden.

I de følgende afsnit, gennemgås de forskellige designe faser, som benyttes i forbindelse med design af en database. Startende med et *entity-relationship* digram[[5]](#footnote-6), herefter kaldet et ER diagram.

# ER Diagram

Følgende figur er et ER diagram bestående af entiteter (kasser), som er identificerbare selvstændige objekter, navngives ofte med et navneord. Samt relationer (kasser på spidsen) som beskriver relationer mellem entiteter, navngives ofte med udsagnsord. Og sidst men ikke mindst af attributter (cirkler), som kan knyttes til både entiteter og relationer. Attributter beskriver enkelte data som navn, adresse etc. Den eller de attributter som entydig identificere en relation er navnet på attributten understreget.

ER Diagram.wmf

Figur - ER Diagram

Som det fremgår af ovenstående figur så har alle entiteter en nøgle kaldet id (alle de gule attributter). Dette skyldes at jeg har valgt at benytte frameworket Hibernate, som er nemmes at arbejde med når der bare benyttes et fortløbende tal som nøgle.

Som det fremgår af diagrammet er der et Produkt som indeholder navn på de produkter som skal systemet skal kunne modtage fejl og ønsker til. Så er der en BariUser, som indeholder de brugere som har adgang til systemet. Mellem Produkt er der en mange til mange relation mellem Produkt of BariUser kaldet userGroup som beskriver hvilke produkter de enkelte brugere har adgang til. En bruger har også en rolle, se userRole, som beskriver om en bruger en en udvikler, administrator eller en alm. bruger. En administrator kan alt i systemet, og som den eneste kan denne oprette nye brugere og produkter. En alm. bruger kan oprette fejl og ønsker, samt oprette og diskutionsindlæg og uploade skærmdumps m.m. En udvikler kan det samme som en alm. bruger, og en udvikler kan ændre status på hvor langt et ønske/fejl er mht. udvikling og test.

Centralt i systemet er der en BariCase som beskriver en fejl/ønske, til hvert BariCase kan der knyttes flere diskutionsindælg (DiscussionMessage) og igen til hvert diskustioninlæg kan der igen knytte flere filer som kan være skærmdumps eller andet (Attachement). En BariCase kan være af type fejl eller ønske (Type), samt have en sags status (CaseStatus), som kan være (todo…) samt en udvikler status (DevStatus )som kan være (todo)

# Konklusion

TODO

# Bilag

## Kode

Koden er delt op de enkelte javapakker.

## Indhold på vedlagte CD

Indholdet på den vedlagte CD er inddelt i følgende 3 kataloger:

* Løsning – Indeholder java kode, htmlfiler m.m samt NetBeans projektfil.
* Program – Indeholder en bari.war samt JavaDB skemafil til oprettelse af databasen.
* Rapport – Indeholder denne rapport i Word 2007 og PDF format og diagrammer i Dia-format. Samt rapporten til faget Web og serverprogrammering, som denne opgave bygger videre på.

1. Wicket – se <http://wicket.apache.org/> [↑](#footnote-ref-2)
2. Java – se <http://www.java.com/en/> [↑](#footnote-ref-3)
3. JavaDB – se <http://www.oracle.com/technetwork/java/javadb/overview/index.html> [↑](#footnote-ref-4)
4. Hibernate – se <http://www.hibernate.org/> [↑](#footnote-ref-5)
5. ER diagram – se <http://en.wikipedia.org/wiki/Er_diagram>, samt kap. 2 i Database Management Systems. [↑](#footnote-ref-6)