IT-DIPLOMUDDANNELSEN

Objektorienterede metoder

Eksamensprojekt

af

Jan Schrøder Hansen

Forår 2011

**INDHOLD**

[1. Indledning 3](#_Toc292533235)

[2. Opgavebeskrivelse 3](#_Toc292533236)

[3. Analyse 4](#_Toc292533237)

[4. UP Iterationer 4](#_Toc292533238)

[5. Konklusion 4](#_Toc292533239)

[6. Bilag 6](#_Toc292533240)

[6.1. Krav 6](#_Toc292533241)

[6.1.1. Funktionelle krav 6](#_Toc292533242)

[6.1.2. Ikke funktionelle krav 8](#_Toc292533243)

[6.2. Dansk/engelsk termer 8](#_Toc292533244)

[6.3. Administrtive use cases 8](#_Toc292533245)

[6.4. Indhold på vedlagte CD 8](#_Toc292533246)

# Indledning

Dette eksamensprojekt er lavet i forbindelse med faget Objektorienterede metoder på IT- Diplomuddannelsen, Ingeniørhøjskolen i København.

Faget har taget udgangspunkt i bogen ”UML 2 and the Unified Process” med undertitlen ” Practical Object-Oriented Analysis and Design” af Jim Arlow og Ila NeustadtRaghu Ramakrishnan. Samt materiale udarbejdet af underviser Jacob Nordfalk.

# 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 og som jeg videreudviklede i faget Databasesystemer og Web efteråret 2010. Begge rapporten kan findes på den vedlagte cd.

Under faget Web og Serverprogrammering 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. Under faget Databasesystemer og Web var mit fokus på database delen, hvor antallet af tabeller kom op på 11. Begge løsninger kan godt betragtes som prototyper.

I denne opgave vil jeg starte forfra med krav, analyse, design af ovenstående system. Selve UP processen vil ikke fremgå af denne rapport, da det ikke svært at beskrive denne proces i rapport form. Så rapporten vil følge den gamle vandfaldsmodel, med krav, analyse, design, etc.

Funktionelle og ikke funktionelle krav til ovenstående kan ses under bilag. Se under bilag afsnit 5.1. Dette skal ses som kravspecifikationer, samme med ovenstående opgave beskrivelse.

Jeg vil dog komme ind på hvordan opgaven kunne løses vha. af UP iterationer.

Da jeg altid skriver på engelsk i min kode, har jeg valgt at mine UML diagrammer også er på engelsk. Men da rapporter her er på dansk har jeg vedlagt en dansk/engelsk ordliste under bilag.

I de følgende afsnit gennemgås de forskellige udviklingsfaser, startende med analysen.

# Analyse

Jeg har valg at lave 2 use case diagrammer, selv om der kun er et system. Diagrammer er opdelt efter administrative use cases og use cases som giver værdi til løsningen. De administrative use cases kan ses under bilag. Disse use cases vil jeg ikke komme ind på, da de er ret trivielle CRUD[[2]](#footnote-3) use cases.

Figur - BaRI Use Cases

Som det fremgår af overstående, så er alle brugertyper, som også fremgår af kravene med på diagrammet. Der er fordi de enkelte brugerroller er vigtige for valgmulighederne i flere af ovenstående use cases.

Todo

# UP Iterationer

TODO

# Konklusion

TODO

# Bilag

## Krav

Følgende af en liste af krav til systemet, delt op efter funktionelle og ikke funktionelle krav. De enkelte krav prioriteres efter MoSCoW (Must have, Should Have, Could Have, Want to have).

### Funktionelle krav

|  |  |  |
| --- | --- | --- |
| Id | Krav | Prioritet |
| 1 | BaRI skal kunne oprette og vedligeholde ønsker til andre systemer | M |
| 2 | BaRI skal kunne oprette og vedligeholde fejl som er fundet i andre systemer | M |
| 3 | BaRI skal kunne knytte diskussionsindlæg til både ønsker og fejl. Et indlæg kan rettes af den bruger som har oprettet dette, så længe der ikke er nyere indlæg. | M |
| 4 | BaRI skal give mulighed for at uploade forskellige filer som knyttes til enten en fejl, et ønske eller et diskusionsindlæg. |  |
| 5 | BaRI skal kunne håndtere følgende sagsbehandlings statusser for både ønske og fejl: ”Oprettet”, ”Behandles”, ”I bero”, ”Godkendt til udv.”, ”Afvist”, ”Lukket”, ”Godkendt” | M |
| 6 | BaRI skal for godkendte fejl og ønsker kunne håndtere følgende udviklingsproces statusser: ”Ikke påbegyndt”, ”Begyndt”, ”Klar til test”, ”Testet” og ”Fejl fundet” | M |
| 7 | BaRI skal kunne prioritere ønsker med følgende prioriteter: ”Skal udvikles”, ”Bør udvikles”, ”Kunne være rart at have”, ”Kan vente” | M |
| 8 | BaRI skal kunne prioritere fejl med følgende prioriteter: ”Driftkritisk”, ”Kritisk”, ”Kosmetisk” | M |
| 9 | BaRI skal logge, alle ændringer på fejl og ønsker. Skal kunne vises i brugerfladen. | C |
| 10 | BaRI skal give mulighed for at slutbrugere kan stemme hvilke ønsker der skal prioteres | C |
| 11 | BaRI skal efter kunne vise en grafisk overside pr. system. Som viser antallet af ønsker med en given status, og på samme måde for fejl. | M |
| 11 | BaRI skal kunne operere med følgende brugerroller: Administrator, Slutbruger, Udvikler, tester, projektleder. | M |
| 12 | En bruger kan have en eller flere roller | M |
| 13 | En BaRI administrator kan som den eneste oprette og rette systemer, nye brugere, bruger/system grupper | M |
| 14 | BaRI skal kunne sende E-mail notifikationer, når der er ændringer på en fejl eller et ønske. De enkelte brugere skal for ønske eller sag selv tilslutte sig notifikation | C |
| 15 | BaRI skal kunne henvise produktion versioner og kode revisioner for fejl og ønsker | C |
| 16 | BaRi skal tildele alle fejl og ønsker et entydigt ID, som kan bruges til at fremsøge sagen, samt skal kunne bruges som henvisning/link fra andre sager/diskussionsindlæg. | M |
| 17 | BaRI skal give mulighed for at opdele systemer i moduler, så fejl og ønsker kan henvise til et modul inden for et system. Dette skal dog være valgfrit. | C |
| 18 | BaRI skal give mulighed for at afgive estimater på fejl og ønsker | M |
| 19 | BaRI skal give alle ikke slut-brugere mulighed for at mærke fejl, ønsker og diskussionsindlæg som ”ikke synlige for slut-brugere” | M |
| 20 | BaRI skal give en projektleder mulighed for at tildele en udvikler og en tester en fejl og et ønske. | M |

Følgende er funktionelle krav, som er opstilles i rolle matrixer. Rolle matrix (C=create, R=read, U=Update, D=Delete)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Krav id 100 | Fejl | Ønske | Diskussions-indlæg | Uploads | Bruger | Gruppe | System |
| Administrator | CRUD | CRUD | CRUD | CRUD | CRUD | CRUD | CRUD |
| Projektleder | CRU | CRU | CRU | CRU |  |  |  |
| Udvikler | CRU | CRU | CRU | CRU |  |  |  |
| Tester | CRU | CRU | CRU | CRU |  |  |  |
| Slut-bruger | CRU | CRU | CRU |  |  |  |  |

De enkelte rollers mulighed for at rette fejl og ønske statusser.

Sagsbehandlingsstatusser, brugere kan ændre status til følgende:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Krav id 101 | Oprettet | Overvejes | I bero | Godkendt til udv. | Godkendt | Afvist | Lukket |
| Administrator | J | J | J | J | J | J | J |
| Projektleder | J | J | J | J | J | J | J |
| Udvikler | J | N | N | N | N | N | N |
| Tester | J | N | N | N | N | N | N |
| Slut-bruger | J | N | N | N | J | N | N |

Udviklingsproces status:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Krav id 102 | Ikke pågyndt | Begyndt | Klar til test | Testet | Fejl fundet |
| Administrator | J | J | J | J | J |
| Projektleder | J | J | J | J | J |
| Udvikler | J | J | J | N | N |
| Tester | N | N | N | J | J |
| Slut-bruger | N | N | N | J | J |

Ønske prioriteter:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Krav id 103** | Skal udvikles | Bør udvikles | Kunne være rart | Kan vente |
| Administrator | J | J | J | J |
| Projektleder | J | J | J | J |
| Udvikler | N | N | N | N |
| Tester | N | N | N | N |
| Slut-bruger | J | J | J | J |

Fejl prioriteter krav:

|  |  |  |  |
| --- | --- | --- | --- |
| **Krav id 104** | Driftkritisk | Kritisk | Kosmetisk |
| Administrator | J | J | J |
| Projektleder | J | J | J |
| Udvikler | J | J | J |
| Tester | N | N | N |
| Slut-bruger | J | J | J |

### Ikke funktionelle krav

|  |  |  |
| --- | --- | --- |
| Id | Krav | Prioritet (MoSCoW) |
| B1 | BaRI skal være en WEB-Løsning | M |
| B2 | BaRI skal kunne køre på en Apache Tomcat version 6 ellere nyere | M |
| B3 | BaRI skal benytte følgende 2 java frameworks skal benyttes: Apache Wicket and Hibernate | M |
| B4 | BaRI skal persistere data i en JavaDB | M |
| B5 | BaRI skal kunne benyttes sammen med HTTPS | M |
| B6 | BaRI skal i sin brugerflade tilbyde følgende 2 sprog: Dansk og engelsk | M |
| B7 | BaRI skal benytte optimistisk låsning | M |

## Dansk/engelsk termer

|  |  |  |
| --- | --- | --- |
| DK term | EN term | Beskrivelse |
| Ønske | Request | Et ønske til et eksisterede system |
| Fejl | Error | En Fejl som er fundet i et kørende system |
| Bruger | User | En BaRI burger |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Administrtive use cases

Figur - Administrative Use Cases

## Indhold på vedlagte CD

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

* Rapport – Indeholder denne rapport i Word 2007 og PDF format
* Databasesystemer og WEB – Indeholder rapporten til faget Databasesystemer og WEB i PDF format.
* WEB og Serverprogrammering – Indeholder rapporten til faget WEB og Serverprogrammering i PDF format.

1. Wicket – se <http://wicket.apache.org/> [↑](#footnote-ref-2)
2. CRUD – Create, Read, Update og Delete [↑](#footnote-ref-3)