ProjDat2015 - Bibliotek udlånssystem

Oliver Sejling Kogut 010694
Thomas Nyegaard-Signori 141093
Casper Helms 260294

Instruktor: Markus Wittorf

April 22, 2015

Indholdsfortegnelse

1 Abstract

2 IT-projektets formål og rammer

Vores IT-projekts formål og rammer er beskrevet ved FACTOR-begrebet[s. 39-41, kilde 2]. FACTOR er et værktøj man kan bruge, til at beskrive/definere et IT-projekt kort og præcist.

2.1 FACTOR

Functionality:
Systemet har funktioner, således at man kan registrere brugere, bøger, udlån og udsende rykkere.

Application Domain:
Systemet administreres af ansatte på biblioteket, men kan ved udlån og returnering anvendes af brugerne selv.

Conditions:
Systemet vil blive udviklet i samarbejde med, og efter kundens ønsker.

Technology:
Systemet vil blive udviklet på henholdsvis mac og linux computere. Det bliver udformet som en hjemmeside, som bliver kodet i HTML, CSS og PHP.

Objects:
Bøger og brugere er hovedobjekterne i dette system.

Responsibility:
Systemet skal have ansvaret for at registrere og overskueliggøre brugere, bogbeholdning og udlån.

Det kan læses ud af ovenstående FACTOR beskrivelse af systemet, at formålet med projektet er at udvikle et system, som kan holde styr på brugere, bogbeholdning og de udlån der måtte være. Et andet formål med systemet, er at aflaste de ansatte eller med andre ord, simplificere arbejdet for dem.

Rammerne for projektet er i bund og grund ret fleksible. Vores kunde har naturligvis en række minimumskrav, som de gerne vil have opfyldt. Når dette er gjort vil der være mulighed for videreudvikling eller at foretage ændringer på det allerede udviklede system, hvis der er behov for dette.

3 Kravspecifikation for IT-løsningen

3.1 Funktionelle- og ikke-funktionelle krav

Funktionelle krav:
Funktionelle krav er de deciderede funktioner, som et it-system skal understøtte. Så som hvad programmet skal kunne[s. 12, kilde 1]. Både de funktionelle og ikke-funktionelle krav, er lavet for at skabe overblik over hvad systemet skal kunne.

- Systemet skal fungere således, at det skal kunne selvbetjenes af lånere både ved udlån og returnering af bøger.
- Bibliotekarer skal yderligere kunne registrere henholdsvis bøger og lånere i systemet.
- Systemet skal kunne tjekke for og udsende rykkere/påmindelser på email, til lånere der har overskredet returneringsdatoen.

Når minimumskravene er opfyldt, vil det være muligt for kunden at ønske mere funktionalitet, hvis tiden er til det og det ønskede krav ikke er for stort. Ovenstående er blot tænkt som værende det simpleste endelige system.

Ikke-funktionelle krav:
Ikke-funktionelle krav, fortæller noget om design af systemet, effektivitet, sikkerhed, sprog osv. altså krav, som ikke er direkte relateret til systemets funktioner[s. 12, kilde 1].

- Det skal være en hjemmeside, som let kan tilgås fra alle styresystemer via en browser.
- Hjemmesiden skal kun kunne tilgås, hvis man er koblet på bibliotekets netværk.
- Systemet skal generelt sikres mod misbrug udefra, hvad end det er rettet mod bruger oplysninger eller ved input af skadelig tekst i indtastningsfelter.
- Systemet skal være brugervenligt og simpelt. Dette vil vi gøre ved at udføre brugertest, og kvalitativ forståelse ved interview med brugere.

3.2 Use case model over system-funktionaliteten

En use case model bruges til at overskueliggøre et system, og bliver i dette tilfælde brugt til at illustrere hvilke funktionaliteter de forskellige aktørere har i systemet.

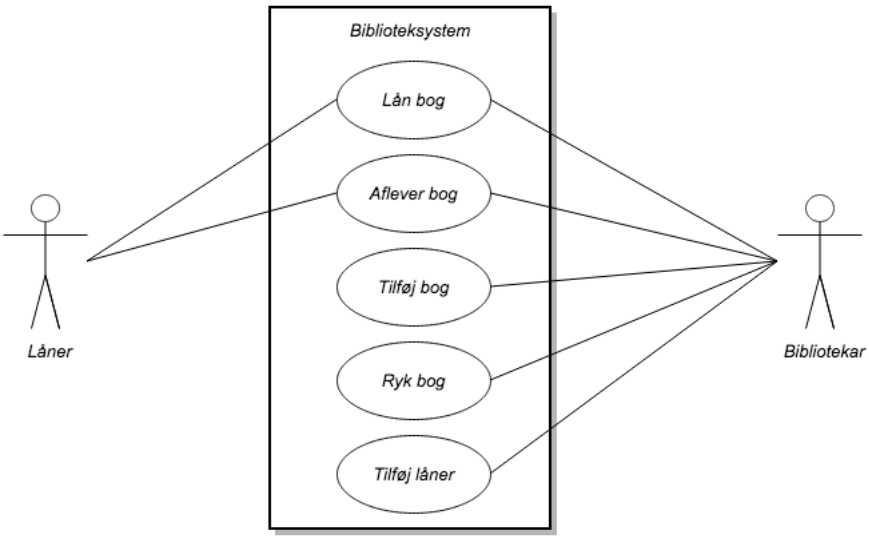

Fig. 3.1: Figuren herover er en illustration af hvilke muligheder, henholdsvis låner og bibliotekar har i systemet. Det skal ikke forstås således at de to aktører interagerer med hinanden, men blot at de har adgang til forskellige funktioner i samme system.

3.3 Use cases

Herunder er specificeret tre use cases, som har en vigtig rolle i systemet. Det er henholdsvis lån af bog, udsending af rykkere, og tilføjelse af bøger til systemet.

3.3.1 Lån af bog

Use case name	LånAfBogNy
Participating actor instances	bob: Låner
Flow of events	1. Bob finder en bog han vil låne og går hen til selvbetjenings computeren med den. 2. Bob vælger funktionen lån bog. 3. Bob indtaster sin mail(hvis han er registreret), og nummeret på bogen. 4. Bob har nu registreret lånet, og forlader biblioteket.

Fig. 3.2: Figuren herover viser, at ved brug af systemet kræver det kun låneren selv for at kunne låne en bog. Derudover forhindres menneskelige fejl, da en bog kun kan lånes, hvis den er registreret i systemet. Så lånet accepteres ikke, hvis der tastes forkert bognummer.

3.3.2 Udsending af rykker

Use case name	RykBogNy
Participating actor instances	alice: Bibliotekar
Flow of events	1. På sin computer vælger Alice funktionen der tjekker for udlån der har overskredet returneringsdatoen. 2. Alice får en liste over alle de udlån der har overskredet returneringsdatoen. 3. Alice klikker på knappen send rykkere. 4. Systemet sender automatisk en rykker ud på email, til hver af de mails der er registreret på udlån af bøger der endnu ikke er returneret.

Fig. 3.3: Figuren herover viser hvorledes automatiseringen af henholdsvis tjek og udsending af rykkere fungerer. Det er en klar forbedring af den tidligere metode, hvor de ansatte selv manuelt skulle søge en række bøger igennem, for derefter at ringe lånerne op for at rykke dem.

3.3.3 Tilføjelse af bog

Use case name	TilføjBog
Participating actor instances	alice: Bibliotekar
Flow of events	1. Alice klikker på funktionen tilføj bog 2. Alice indtaster oplysninger på bogen (nr. titel, forfatter, år osv.) 3. Alice bekræfter de indtastede oplysninger 4. Systemet fortæller Alice, om bogen blev succesfuldt registreret. 5. Alice kan hvis den blev registreret, sætte bogen på hylden, og fortsætte med den næste

Fig. 3.4: Figuren herover viser hvorledes en bibliotekar tilføjer en bog til systemets database. Hvilke oplysninger på bogen der bør kunne indtastes, er endnu ikke fastlagt, så ovenstående er derfor blot et eksempel.

3.4 Problemområde

Systemets problemområde, som består af de bøger, brugere, udlån og rykkere der skal administreres, registreres og lagres i databasen, og som systemet er bygget op omkring.

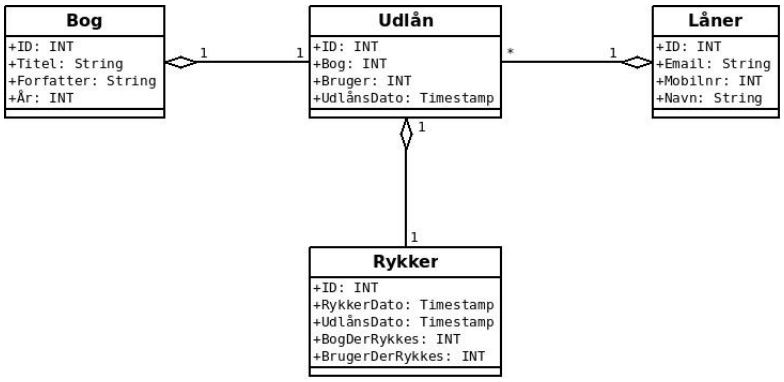

Fig. 3.5: Figuren herover er et klassediagram over de klasser, som vores problemområde, og vores system er bygget op omkring. De givne attributter er ikke fastlagt, men vi forventer, at det bliver noget lignende det, der er anført. Figuren hænger sammen således, at et udlån består af en bog og en låner, og en rykker består af et udlån(hvori bog og låner er implicit). Alle multipliciteter er 1 til 1, udover at en låner godt kan have flere udlån.

4 Systemdesign sammenfatning

5 Program- og systemtest

6 Brugergrænseflade og interaktionsdesign

7 Projektsamarbejdet

8 Bibliografi

1. Bruegge, Bernd og Dutoit, Allen H. 2014. "Object-Oriented Software Engineering Using UML, Patterns, and Java third edition"

2. Mathiassen L., Munk-Madsen A., Nielsen P. A., Stage J. 2000. "Object-Oriented Analysis and Design" (Uddraget fra ugeseddel 6)

9 Bilag 1: Versionsstyring

10 Bilag 2: Changelog for projektrapporten

14-04-2015 OSK Rapport påbegyndt
14-04-2015 OSK Punkt 2 og 3 påbegyndt

11 Bilag 3: Timeline

09-03-2015:
Møde med product owner Simon Shine, for at få de overordnede krav, og ønsker til IT-systemet.

13-03-2015:
Arbejde med første delrapport er begyndes.

23-03-2015:
Første delrapport færdig.