



ProjDat 2015

Delrapport 1

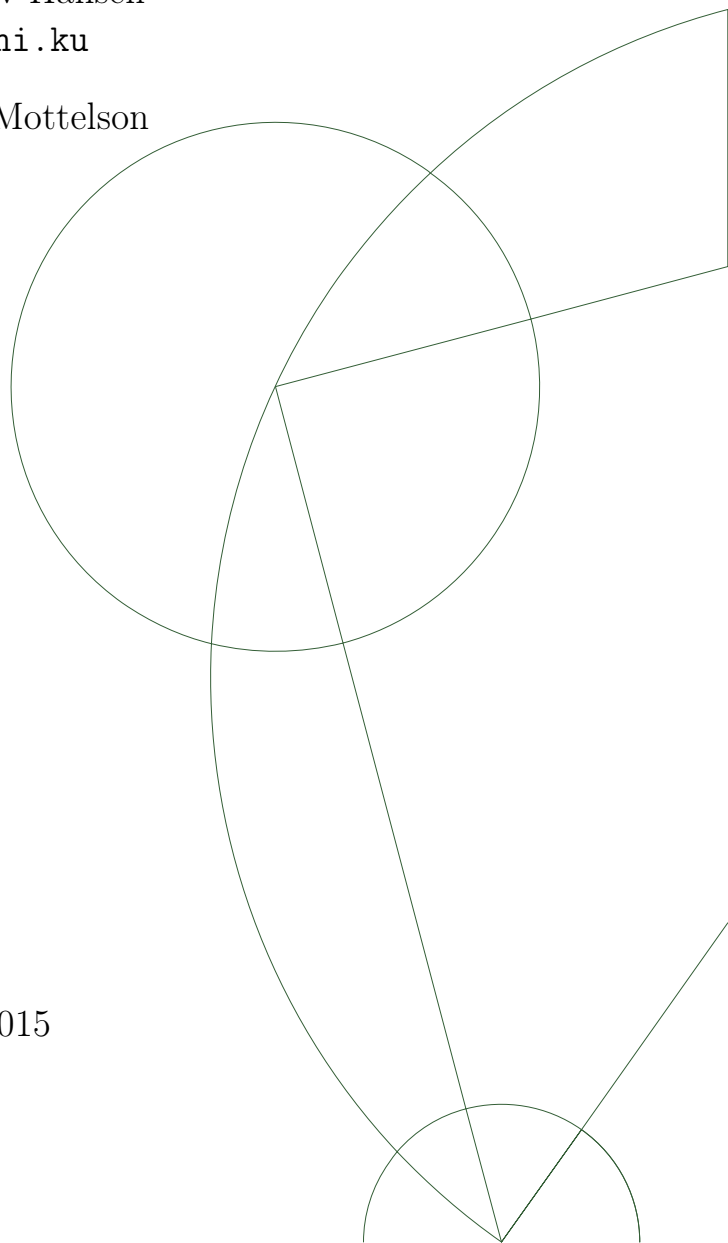
Christian Kjær Larsen
wkm839@alumni.ku

Lukas Svarre Engedal
xxx123@alumni.ku

Tobias Sønderskov Hansen
xxx123@alumni.ku

Instruktor: Aske Mottelson

18. marts 2015



Indhold

1	Problemformulering	1
1.1	Problemområde	1
1.2	Scenarier	1
1.3	Funktionelle krav	1
1.4	Ikke-funktionelle krav	2
2	Indledende projektplan	2
2.1	Projektoversigt	2
2.2	Resultatbeskrivelse	2
2.3	Arbejdsfordeling	2
2.4	Køreplan	2
2.5	Risikohåndtering	3
3	System og softwarearkitektur	3
3.1	Delsystemer	3
4	Projektaftalen	3
5	Intern projektetablering	3
6	Afleveringsopgaver	3
6.1	OOSE 1.6	3
6.2	OOSE 1.8	3
6.3	OOSE 2.(6, 7, 9, 10)	3
6.4	OOSE 5.3	3
6.5	OOSE 7.1	3

1 Problemformulering

1.1 Problemområde

Fra næste år af laver de markant om på datalogi uddannelsen, og i den forbindelse slår de bl.a. kurserne *Introduktion til Programmering* og *Objektorienteret programmering og design* sammen til et nyt kursus, hvor de i stedet for SML og Java vil bruge F#. I den forbindelse forudser kursuslederne at det muligvis kan blive svært at finde tilstrækkeligt med instruktører til at hjælpe de studerende med at lære F# og med at løse opgaver i F#, og kursuslederne vil derfor have automatiseret hele opgaveaspektet lidt.

I den forbindelse har de bedt os om at hjælpe dem med at lave et slags interaktivt læringssystem, hvor de studerende kan logge ind og løse en række opgaver der skal hjælpe dem med at lære F# og de kan få hints til opgaverne hvis de har brug for det, og kursuslederne kan tilføje nye og ændre eller fjerne eksisterende spørgsmål og få en slags statistik over hvordan de studerende klarer opgaverne.

1.2 Scenarier

Tilføjelse af spørgsmål En underviser på kurset går ind på hjemmesiden. Vedkommende logger ind på siden, og han tages til en speciel side for administratorer. Her skal han tilføje en ny opgave. Vedkommende giver spørgsmålet en kategori efter hvilket læringsmål det opfylder. Opgaveteksten og de rigtige svar skrives ind. Eventuelt tilføjes hints.

Løsning af opgaver En studerende på kurset logger ind på hjemmesiden, vedkommen- de tages til en side, hvor man ser alle læringsmålene, og hvad man allerede har opfyldt. Den studerende vælger et af de tilgængelige læringsmål, og en side åbnes hvor man kan besvare en ny opgave inden for emnet. Den studerende svarer på spørgsmålet, og får feedback med det samme, og der er mulighed for hints, hvis spørgsmålet er svært.

1.3 Funktionelle krav

- Systemet skal være en hjemmeside med opgaver, som kan løses individuelt af studerende.
- Der kræves login, så man kan følge med i den studerendes udvikling.
- Spørgsmålene skal være grupperede efter hvilke læringsmål de tester.
- Læringsmålene skal igen grupperes efter et *threshold*, som er et øvre læringsmål hvorefter der kommer sværere stof.
- Der skal være hints til opgaverne, og der skal kunne gives feedback til hintsne.
- Alle forsøg på opgavebesvarelser skal gemmes i en log.

Hvis der er tid i en af de senere iterationer, så er der en række ekstra krav som kan implementeres.

- Loggen skal bruges til at give underviseren information om hvilke spørgsmål der er svære.

- Der skal tilføjes en grad af *gamification*, så der gives badges og point for fremskridt, og det bliver muligt at følge med i andres fremskridt.
- Hvis man ikke har øvet sig i et emne i et stykke tid, så falder ens erfaring i området.

1.4 Ikke-funktionelle krav

De ikke-funktionelle krav til dette projekt er forholdsvis begrænsede. Det vigtigste krav er at det skal være nemt at tilføje nye opgaver til systemet, sådan at kursuslederne efterfølgende kan opbygge en tilstrækkelig samling af opgaver til de studerende.

Derudover er det også det mere overordnede krav at det endelige produkt gerne skal være så simpelt som muligt, så det er nemt at overtage og udvide og bygge videre på efter at vi overgiver projektet til kunden. Det er bl.a. derfor at vi har valgt at bruge Flask frameworket og at lave projektet i Python, idet at det var hvad kunden foreslog og foretrak.

2 Indledende projektplan

2.1 Projektoversigt

Projektet omhandler udvikling af en webapplikation. Den skal understøtte undervisningen på førsteårskurset i programmering på DIKU. Applikationen skal hjælpe studerende med at forstå kernestof, ved at stille opgaver tilhørende de forskellige læringsmål. Underviserne skal så kunne tilføje nye opgaver, og indhente statistik på allerede stillede opgaver.

2.2 Resultatbeskrivelse

Til sidst i projektet er målet, at vi afleverer et fungerende system til kunden. Til prototypen skal der også være en mængde dokumentation til kunden, så vedkommende ved, hvordan løsningen er sammensat. Til kurset skal vi aflevere en mængde delrapporter, som dokumenterer vores arbejde i løbet af projektet.

2.3 Arbejdsfordeling

På nuværende tidspunkt er det svært at sige en masse om arbejdsfordelingen, umiddelbart er vi alle tre interesserede i opgaven og dens forskellige aspekter og vil gerne deltage i dem alle, men hen af vejen finder vi måske ud af at opdele opgaven imellem os ud fra interesser og kompetencer.

2.4 Køreplan

Vores køreplan er at følge den plan der er fastlagt i kurset i forhold til hvornår vi skal have prototyper og rapporter klar, og derudover vil vi forsøge så vidt muligt at arbejde i iterationer af f.eks. 2 ugers varighed hvor vi tilføjer funktionalitet modulvis.

2.5 Risikohåndtering

Under udviklingsprocessen holder vi jævnligt møder, hvor vi snakker om hvor langt vi er nået. Dette kan bruges til at opdage, om noget af det der udvikles kan udgøre et problem i fremtiden. Det kan være, at to moduler, som to personer udvikler hver for sig, måske ikke er helt kompatible. Det kan også være at man har undervurderet tiden en opgave tager, og der så skal bruges mere tid, eller flere udviklere.

3 System og softwarearkitektur

3.1 Delsystemer

- Der skal være et delsystem, som tager sig af opbevaring af vedvarende data. Det er fx opgaver, brugere og en log over besvarelser. Det vil nok være en databaseserver.
- Der skal også være et delsystem, der tager sig af forretningslogikken. Der er i vores tilfælde nok en webserver, som tager imod forespørgsler fra brugere, og kommunikerer med databasen. Den vil nok også stå for at lave html, der vises i brugerens webbrowser.
- Brugerens webbrowser er også et undersystem. Det sørger for at sende forespørgsler til vores webserver, og præsentere den returnerede html. Det er også her at designprocessen foregår, altså den store treenighed, nemlig html, css og javascript.

4 Projektaftalen

5 Intern projektetablering

6 Afleveringsopgaver

6.1 OOSE 1.6

6.2 OOSE 1.8

6.3 OOSE 2.(6, 7, 9, 10)

6.4 OOSE 5.3

6.5 OOSE 7.1