

Flybooking

Flemming Mertz og Martin Høgedal

1.1 Formål og mål

Formålet med dette dokument er at beskrive de elementer vi planlægger at implementere i den opgave, der skal afleveres som en del af kursus. Beskrivelsen vil være på et højt abstraktionsniveau, da målet med dette dokument ikke er at beskrive et egentligt design, men blot at komme med en tilstrækkelig beskrivelse af opgaven til at få den godkendt som formulering.

Det specifikke emne for opgaven er ikke i sig selv vigtigt, men tjener blot til at vise en forståelse for de emner der er blevet gennemgået i løbet af kurset. Vi har derfor taget udgangspunkt i det eksempel der blev givet til en af de første undervisninger, med en lufthavn, eller nærmere betegnet et flyselskab.

Målet er at implementere et simpelt booking system, der skal gøre det muligt for en ekspedient at sælge flyrejser til kunder. Dette åbner mulighed for at bruge en masse af de principper og teknologier der gør sig gældende inden for OOP og specielt Java.

1.2 Relateret information

Dette dokument er beregnet til at svare til vejledningen til afleveringsopgaven, der findes på:

http://www.dat.ruc.dk/~keld/teaching/OOP_e03/Afleveringsopgaven/vejledning.pdf

2 Krav til opgaven

For at komme rundt i så mange principper og teknologier inden for Java, er der sat følgende krav til opgaven. (principper dækket af kravet i parentes):

Der skal være en grafisk brugergrænseflade, der gør det muligt at få vist, og oprette, kundeoplysninger, flyafgange og bookinger, og som samtidig gør det muligt at søge inden for disse data. (Swing)

Applikationen skal fungere som klient/server, samt give mulighed for at afvikle den lokalt. (netværk)

(Både server og klientdel skal implementeres)

Applikationen skal fungere med flere samtidige brugere. (trådsikkerhed)

Data skal gemmes i en database, og hvis ikke databasen er tilgængelig over netværket, skal data gemmes i en lokal fil, med mulighed for at synkronisere når databasen igen er tilgængelig. (database adgang)

Kravene til klientdelen vil fokusere på brugervenlighed og responstid, og for serverdelen vil sikkerhed, effektivitet og samtidighed være hovedfokus.

Alle dele skal dokumenteres med Javadoc, og overholde Sun's kode guidelines.

Samlet giver denne opgave mulighed for at bruge en masse af de delelementer der samlet gør Java stærkt, ligesom der er masser af mulighed for at udøve godt objektorienteret design. Der er gode muligheder for brug af patterns, og både Swing, netværksprogrammering, IO og samtidighed er inkluderet. Klassebegrebet, modularisering og fejlhåndtering bør være implicit i et Java program, og således er alle de punkter der er nævnt under "Indhold" på kursets hjemmeside indeholdt i denne opgave.