

# Objektorienteret programmering

## Opgaveformulering

Af Christian Franke (OB) og Ida Behrent (TIT)

Vi skriver begge projekter, der omfatter client/server-programmering i dette semester<sup>1</sup>. Derfor vil gerne benytte afleveringsopgaven til kurset, således, at vi kan inddrage både elementer fra undervisningen samt integrere elementer, der er relevant for vores projekt.

### Om Casen

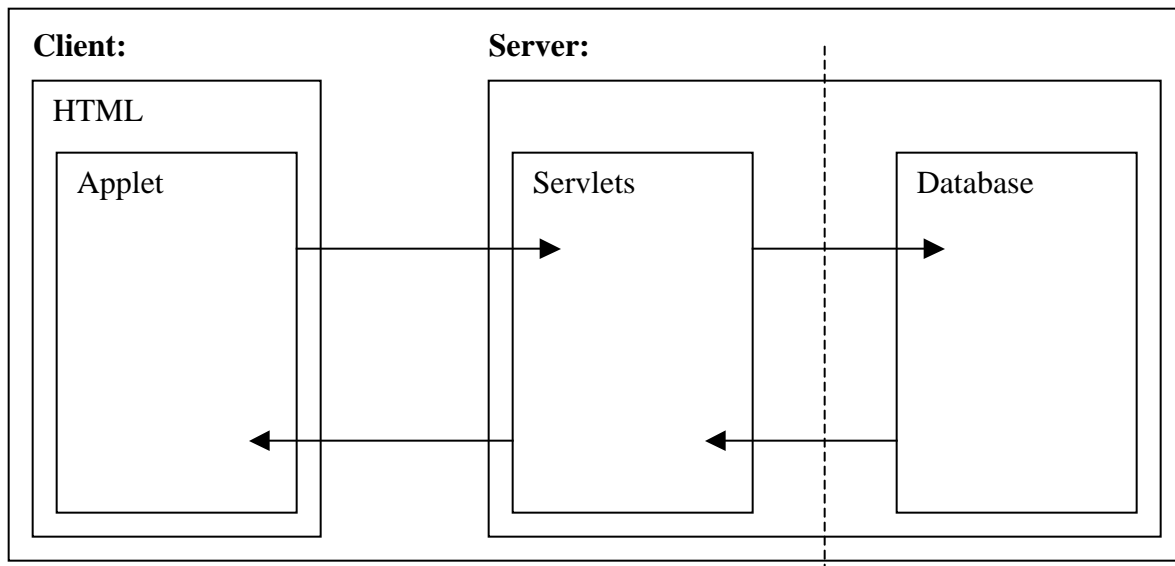
Eftersom vi ikke arbejder på det samme projekt, har vi fundet en case, som vi i stedet kan arbejde med og udvikle til fælles gavn for vores respektive projekter. Vi har valgt, at casen skal være et simpelt boghandler-søgesystem. Det vil sige, at det skal være muligt at få vist resultatet af en søgning på enten forfatter eller titel. Derudover skal brugerne kunne 'opdatere' databasen, i form af knytte kommentarer til en bog. Der er altså to overordnede typer af funktionalitet: 1) søgning/visning og 2) opdatering.

Grunden til at der ikke gås i detaljer med raffinerede søge- og opdateringsmetoder, er at vægten lægges på samspillet mellem applets og servlet-teknologien.

Figur 1 nedenfor er blot vejledende, men viser den overordnede idé i opgaven.

Den stiplede linie illustrerer her blot det forhold, der ville gøre sig gældende, hvis der havde været tale om en fuld Three-tier implementation. Fokus er imidlertid primært på client/server-delen frem for implementering af en hensigtsmæssig datastruktur i databasen. Derfor vil databasen være en flad fil.

Figur 1.



<sup>1</sup> Vi skriver ikke på samme projekt

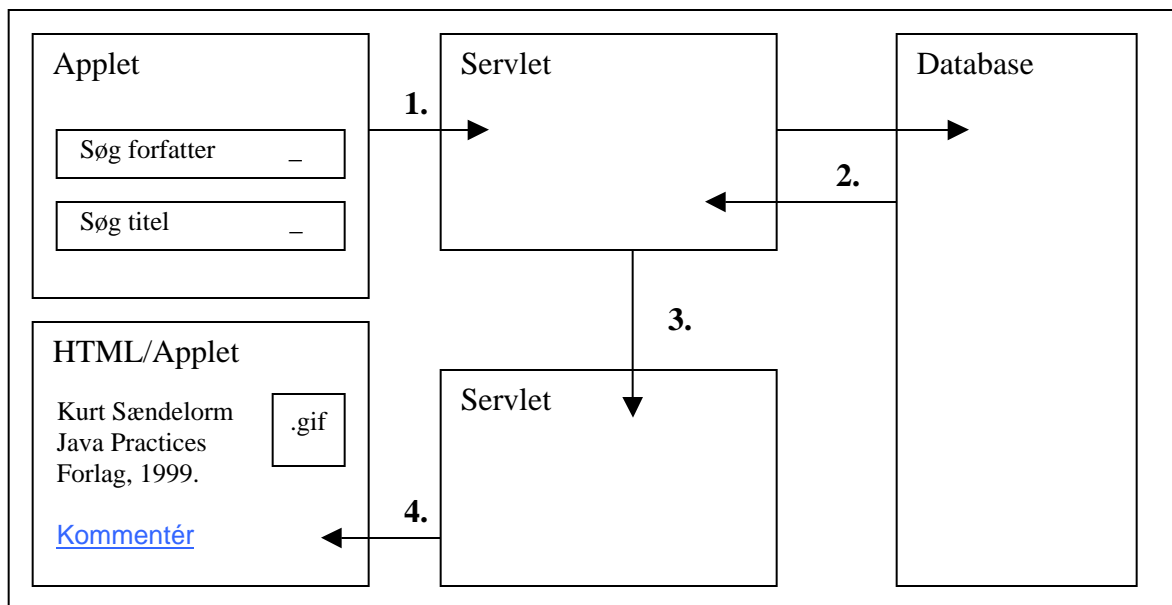
Afleveringsopgaven vil bl.a. indeholde følgende elementer:

- En grafisk brugergrænseflade vha. appletter og html
- RMI (Remote Method Invocation)
- Server-side programmering i form af servlets
- 'Database' i form af evt. en vector ell. arraylist i en fil.

### Søgning og visning

Via en applet kan man ved hjælp af scroll-downs vælge mellem forfattere/titler (som vi har lagt i databasen). Gennem RMI kaldes en servlet, der foretager søgningen i databasen. Resultaterne, der returneres fra databasen, vises via en servlet hos klienten, enten som HTML eller en applet. Det er endvidere tiltænkt, at visningen af en søgning også vil indeholde et billede af bogens cover. Dette er illustreret i figur 2.

Figur 2.



Figuren er ikke udtryk for den eksakte struktur, men blot overvejelser omkring designet.

1. Brugeren vælger fra en liste af forfattere/titler i applettens scroll-down menuer, og foretager et RMI til en servlet,
2. hvorefter en servletten henter forespørgslen fra databasen
3. og returnerer den til en anden servlet,
4. der skriver til browseren, enten som ren HTML eller gennem en applet.

### Opdater

Der kan tilføjes kommentarer til de enkelte bøger, hvis brugerne ønsker det fra et tekst-area enten i en applet eller HTML. Disse kommentarer tilknyttes pågældende bog/bøger, gemmes som tekst i databasen, og vil blive vist ved næste søgning/visning.

Afleveringsopgaven tænkes struktureret som følger:

- Usecases
- Kravsspecifikation
- Analyse/design
- Klassediagrammering
- Implementering
- Test

Da opgaven omfatter programmering af servlets, der ligger på en server, vil produktet blive afleveret som et link til en hjemmeside. Koden vil medfølge både som bilag til rapporten og på diskette eller CD-ROM.

Venlig hilsen

Christian Franke og Ida Behrent

Kurt Sændelorm  
Java Practices  
Forlag, 1999.

[Kommentér](#)