DAT 1. semester - SP3

– en streamingtjeneste

1 Opgaven

I SP3 skal I lave en objektorienteret analyse af et domæne, og ud fra analysen lave et objektorienteret design af jeres løsning. Derefter skal I implementere løsningen i Java. I jeres løsning skal I anvende basale Java-konstruktioner som klasser, metoder, interfaces, løkker, betingelser, ArrayLists eller andre datastrukturer mv.

2 Domænet

I projektet skal I udvikle et softwaresystem til brug for streamingtjenester som Netflix, HBO, Viaplay etc.

I systemet er der medier, der kan afspilles. Medier kan være film eller serier. Både film og serier er inddelt i kategorier som fx crime, war, drama, family, romance og sci-fi. Serier består desuden af sæsoner og episoder. Alle medier har et udgivelsesår og en rating.

I systemet er der også brugere, som har lister over de film, de har set og de film, de gerne vil se.

Vi laver i første omgang ikke nogen grafisk brugergrænseflade til systemet, men laver det rent tekstbaseret.

3. Funktionelle krav

Der er følgende krav til systemet:

- Der skal være en startmenu (i tekst), hvor brugeren kan vælge mellem at
    - Oprette sig som bruger
    - Logge sig ind som bruger
- Der skal være en hovedmenu (i tekst), hvor brugeren kan vælge mellem at
    - Søge efter en bestemt film
    - Søge alle film i en kategori
    - Se sin liste over sete film
    - Se sin liste over gemte film
- Når brugeren får præsenteret en liste over alle film i en kategori, alle sete film eller alle gemte film, skal brugeren kunne vælge én af filmene
- Når brugeren har valgt en film, skal det være muligt at vælge mellem at afspille filmen (dette gøres naturligvis ikke "rigtigt" men blot ved at skrive "Titanic afspilles nu.." eller noget lignende), gemme filmen i sin liste over film, han eller hun ønsker at se senere eller slette filmen fra listen over film, han eller hun vil se senere. Vælger brugeren at afspille filmen, skal den gemmes i hans eller hendes liste over afspillede film.
- Brugernes login-oplysninger (brugernavn og password) skal gemmes i en fil, så de kan indlæses når programmet startes op igen.
- Der skal være fornuftig håndtering af fejl. Hvis der opstår fejl i programmet skal de håndteres passende steder og brugeren skal informeres, hvis det er relevant. Overvej hvem fejlen skal kommunikeres til. En IOException (som kan opstå, når man læser ind fra en fil) vil fx gøre, at programmet fra brugerens synspunkt ikke virker. Hvordan formidler vi til brugeren, at der er opstået en fejl?

Der *kan* tænkes en masse ekstra features ind, som fx (ikke prioriteret rækkefølge)

- En bruger vil gerne søge efter film, der har rating større end 8,5
- En bruger vil gerne søge efter film fra 80'erne
- Der skal kunne tilføjes og fjernes kategorier på en nem måde, fx "Christmas" eller "Halloween"
- Der skal kunne tilføjes film, serier, sæsoner og episoder hvis brugeren er administrator
- Data om brugeres lister skrives ned i en fil, når programmet lukkes.
- En bruger kan være klassificeret som "barn" og må således kun se film i kategorien "Family"
- Der skal kunne tilføjes nye typer medier, som fx e-bøger, lydbøger eller spil
- I kan lave en grafisk brugergrænseflade med Processing eller med Java Swing eller Java FX

4. Data

I skal i løsningen hente data om film og serier fra .txt-filer, som I får udleveret. Start med at lave en funktion, som kan indlæse data fra filerne og fylde nogle ArrayLists med film, serier, episoder, etc.

Der er også zip-filer med billeder, som I kan bruge hvis I vælger at lave grafisk brugergrænseflade.

5 Aflevering

Koden skal ligge i et repository på github og I skal aflevere et link til dette. Der skal kun afleveres ét link pr. gruppe og der skal ikke ligge andet i dette repository (ingen gamle ugeopgaver osv).

6 Gode Råd

Prioritering
Løs hellere nogle delproblemer godt end alle overfladisk. Det er *meget* bedre udtrykkeligt at beslutte, at et specielt delproblem (fx. "brugeren skal kunne se sin liste over film") slet ikke håndteres i systemet (og skrive det på en mangelliste!), end at programmere en uigennemtænkt, uafprøvet løsning til delproblemet. Sådan en delløsning kan komplicere den samlede softwareløsning, reducere dens brugbarhed og vanskeliggøre videreudvikling.

Divide and Conquer
Lad være med at lave alt på én gang.

Forsøg at se på hvert delproblem for sig, og løs hvert delproblem før I starter på noget nyt

- Eksempel: Lav funktionalitet, som kan fylde ArrayLists med film, serier og brugere
- Eksempel: Lav funktionalitet, som tillader brugeren at oprette sig som bruger

Start simpelt
Hvis noget virker meget uoverskueligt, så *Solve a Simpler Problem First*. Lav en løsning der er åbenlyst utilstrækkelig, men dog et skridt i den rigtige retning. Derefter er I meget klogere, og kan tage et nyt

Fx kan man starte med kun at have film og kun at have én bruger.

7 Gruppearbejde

En del af udfordringen er at I skal samarbejde i en gruppe. Det er derfor ikke muligt at arbejde solo med denne opgave. Før I starter arbejdet bør I snakke om i gruppen hvordan I vil arbejde, hvor ofte og hvor længe. Sørg for at organisere arbejdet så det bliver ligeligt fordelt, eller sådan at hver opgave passer til den enkeltes evner og interesser. Sørg for at jeres klassediagram hele tiden er i en tilstand hvor det fungerer som aftale for hvad der bliver kodet.

Vi ser hellere at alle i gruppen kan bevare overblik over koden end at enkelte gruppemedlemmer får implementeret af en masse features.