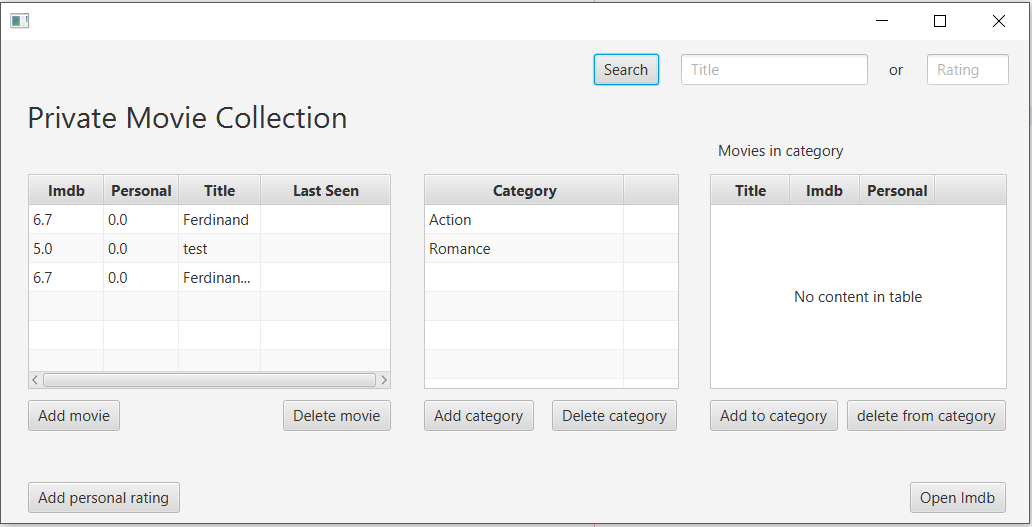
**Semester eksamen - Private Movie Collection**

Af Ásvør Rasmussen, Michael Lemmiche & Theis Kvist Kristensen



Liste af funktionaliteter som programmet skal indeholde.

1. Tilføje/fjerne film
2. Tilføje/fjerne genrer
3. Tilføje flere genrer per film
4. Søge efter specifikke filmtitler eller dele af filmtitler, En eller flere genrer/og eller specifikt minimum imdb anmeldelse.
5. Klikke på en film og afspille den i enten programmet eller i computerens standard videoafspiller.
6. Tilføje/ændre en personlig pointanmeldelse af hver film.
7. Sortere filmene efter score, title eller kategori mens søgefiltrene er aktive.
8. Kun .mp4 og mpeg4 filer skal kunne tilføjes til programmet
9. Når programmet starter skal der komme en besked om at man skal huske at slette film med en personlig rating under 6 og film der ikke er blevet set i mere end 2år.
10. programmet skal så vidtmuligt kun kunne tilføje en film 1 gang.

Brugergrænsefladen viser tre tableviews, film, kategori og film i kategorier.

Under hver af de tre tableviews er der to knapper, den ene er til at tilføje til det tableview og den anden knap er til at fjerne fra det tableview. I nederste venstre hjørne har vi en knap med funktionen til at tilføje en personlig rating, i nederste højre hjørne har vi en knap som hvis man trykker på den så åbner den IMDb’s hjemmeside.

I øverste højre hjørne har vi et søgefelt hvortil knappen Search tilhører.

Man kan søge på IMDb og Titlen på en film i det tableview der indeholder filmene, hvor man kun kan søge på filmens titel i det tableview som indeholder de film som er i den valgte kategori.

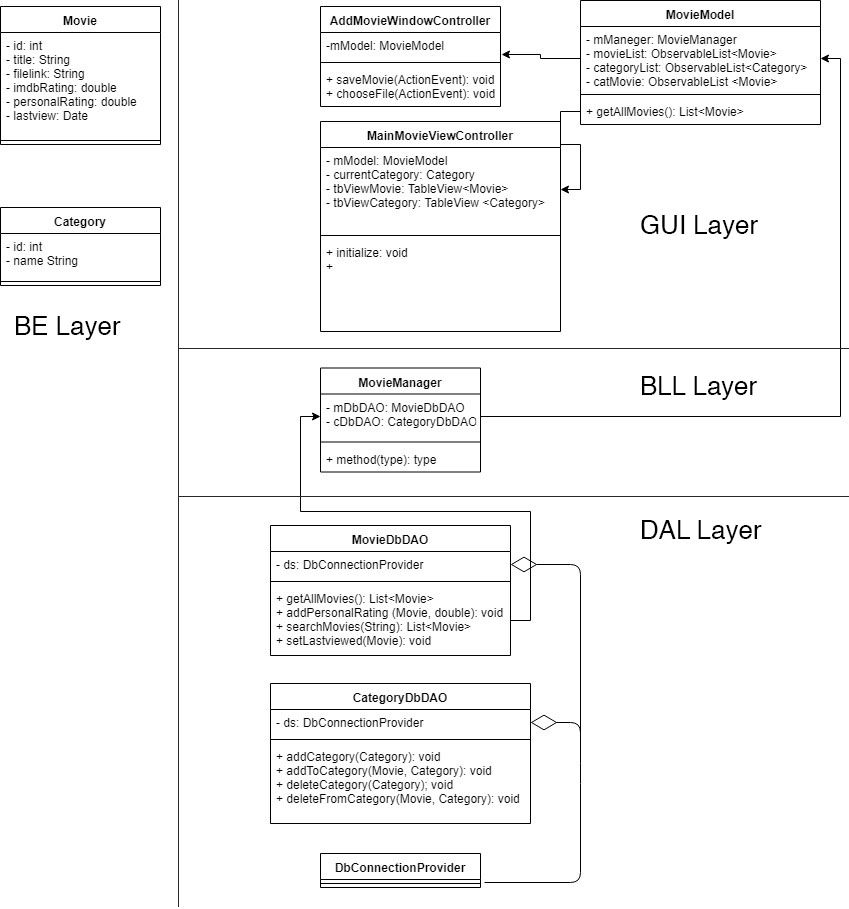
Vi har Testet programmet med mp4 filer og programmet fungerer som ønsket.

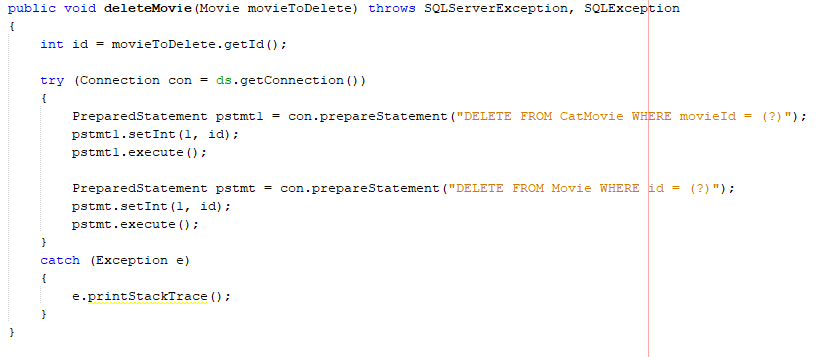
Vores projekt er delt ind i følgende mapper GUI.View, GUI.Controller, GUI.Model, BLL, BLL.Exception, DAL.Database, DAL.Exception og BE.

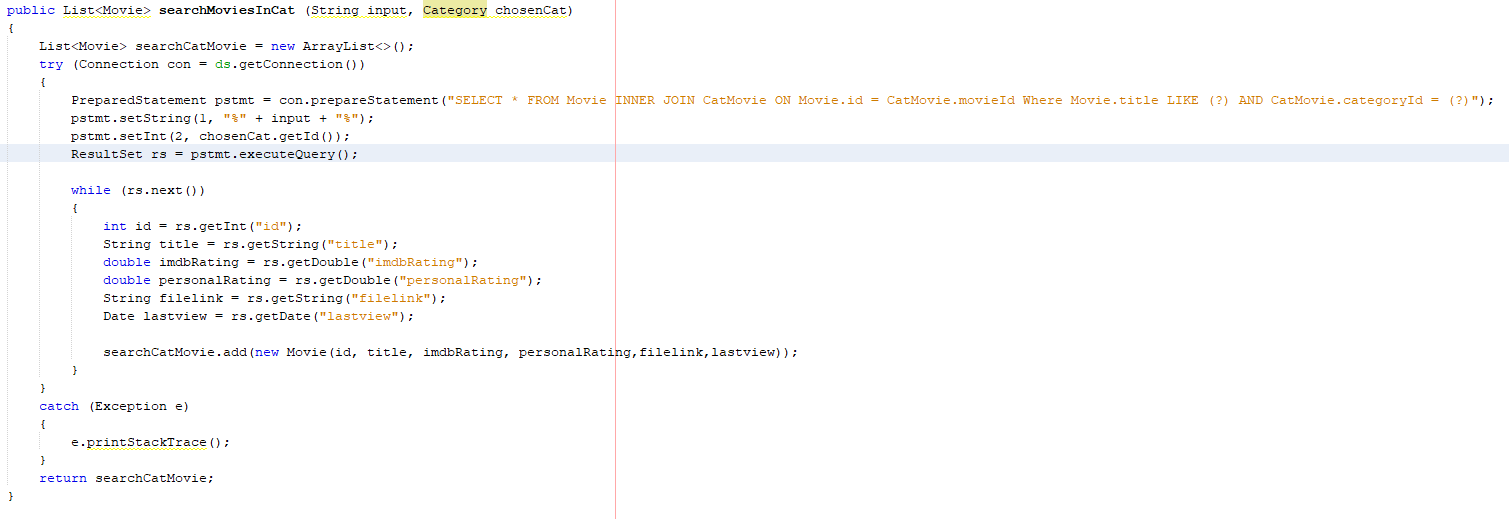
Inde i mappen GUI.View har vi to FXML filer hvor den ene er hovedevinduet og det andet er tilføj en film vinduet. GUI.Controller mappen indeholder to controllere, den ene styrer hovedevinduet og den anden styrer tilføj en film vinduet. GUI.Model mappen indeholder vores model og funktion som vores model har er at den omdanner vores kode og sender den til vores controller så den kan håndterer vores kode. BLL laget indeholder vores MovieManager som har til opgave at sende informationerne fra DAL laget til vores Model. BLL.Exception laver Exceptions som man kan bruge til at håndterer de Exceptions der sker i vores BLL lag men vi bruger JOptionPane til at stop så mange fejl som muligt i GUI laget. DAL.Database Her etablere vi forbindelse til vores database, hvor at vi så kan sende SQL-statements ned til databasen og få informationer tilbage, Vores MovieDbDAO klasse sende største delen af den information som den sender ned i movie tabellen og CategoryDbDAO klassen sende største delen af den information som den sender ned i category tabellen. DAL.Exception laver Exceptions som man kan bruge til at håndterer de Exceptions der sker i vores DALL lag men vi bruger JOptionPane og forskellige if statements til at stoppe så mange fejl som muligt. BE indeholder Beskrivelsen af vores Business entiteter.

Github Link:

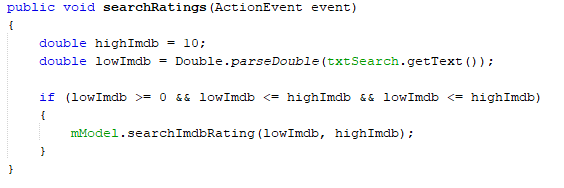
<https://github.com/mich363r/PrivateMovieCollection>

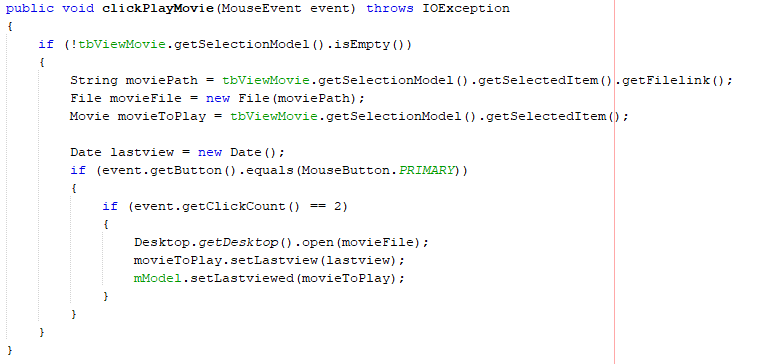
Vores UML diagram viser vores klasser og nogler metoder der går igennem alle lagene

Vi laver to prepared statements for at vi kan fjerne filmen både i movie og category i databasen.



Til at søge på film i en kategori bruger vi en inner join, og i den inner join gør vi det at vi tager alle film og sammenligner deres id med CatMovie.movieId hvor titlen ligner input og categoriId er lig med den valgte category, for at undgå at vi finder alle film inde i catMovie ligegyldig hvilken kategori filmen tilhører.

Da vi aldrig kan få en IMDb rating over ti, sætter vi den øvre grænse til at være 10 og det er den der hedder highImdb. Vores lowImdb tager det input som brugeren giver, den tager så string værdien og returnerer en tilsvarende double værdi. Den returnerede værdi kan vi bruge til at finde film hvis IMDb rating ligger mellem den værdi og vores max værdi.



Vi trykker på en film i vores tableView<Movie> og så henter vi placeringen af filen fra vores database og gemmer denne i en lokal variable som vi bruger til at oprette et nyt movie objekt. Dette objekt bliver efterfølgende åbnet i systemets default mp4 afspiller.

Endvidere, så sætter vi datoen på hvornår filmen sidst blev set, både i vores liste og i vores database.

Det første billede blev det brugt JOptionPane men da swing er forældet har vi brugt dialogs istedet for. og i vores deleteMovie der bruger vi 2 typer af Alerts, Error og den anden er Confirmation.

