Jaron Somers

CS292 – Final Project

04/27/2016

The original requirements were:

*Favorite Movie Application*

*This application will consist of multiple classes, forms, and a database. I don’t have the UML diagram yet since I just thought of this idea and I sort of struggled coming up with an idea anyways. The idea of the application is for a user to permanently store movies with a name, rating, year released and their own description of the movie and why they liked it. I personally have a huge list of movies that I know I would love to watch over and over again but I just can’t think of them. If I hear the movie name, then I remember that I liked that movie a lot! So having this list that a user can maintain is beneficial in that way. Granted, this can be done with a pen and paper, but this helps you nicely format entries and sort based on highest rating.*

*Class Movie*

*-yearReleased int*

*-description String (set number of characters)*

*-name String*

*-score double (0.0-10.0)*

*::Movie-Constructor for a movie object*

*::getDescription*

*::getYearReleased*

*::getName*

*::getScore*

*MainForm*

*ArrayList<Movie> -List of the movies from the database*

*A Table loaded from SQL database listing movie objects.*

*::addMovie*

*::removeMovie*

*::refreshMovieTable (aka read)*

*::editMovie*

*MovieDetailsForm*

*Form for editing the details of the movie.*

*Changes can only be done in this form. It is accessible through an Edit Button while selecting an existing movie from the table.*

*Save and Cancel options to either save changes if there are any, cancel to exit the new form.*

*Requirements:*

*The application shall list the movies in ascending or descending order by score.*

*The application shall create, edit, or delete movies.*

*The application shall allow a user to specify a description of the movie*

*The application shall list the name of the movie*

*The application shall list the year released (inputted by the user) of the movie.*

*The application shall connect to SQL database for its ability for persistent storage*.

\*No changes were made to the original requirements. I still ended up using a regular service based database instead of sqlite.

The following requirements were implemented:

*The application shall list the movies in ascending or descending order by score.*

*The application shall create, edit, or delete movies.*

*The application shall allow a user to specify a description of the movie*

*The application shall list the name of the movie*

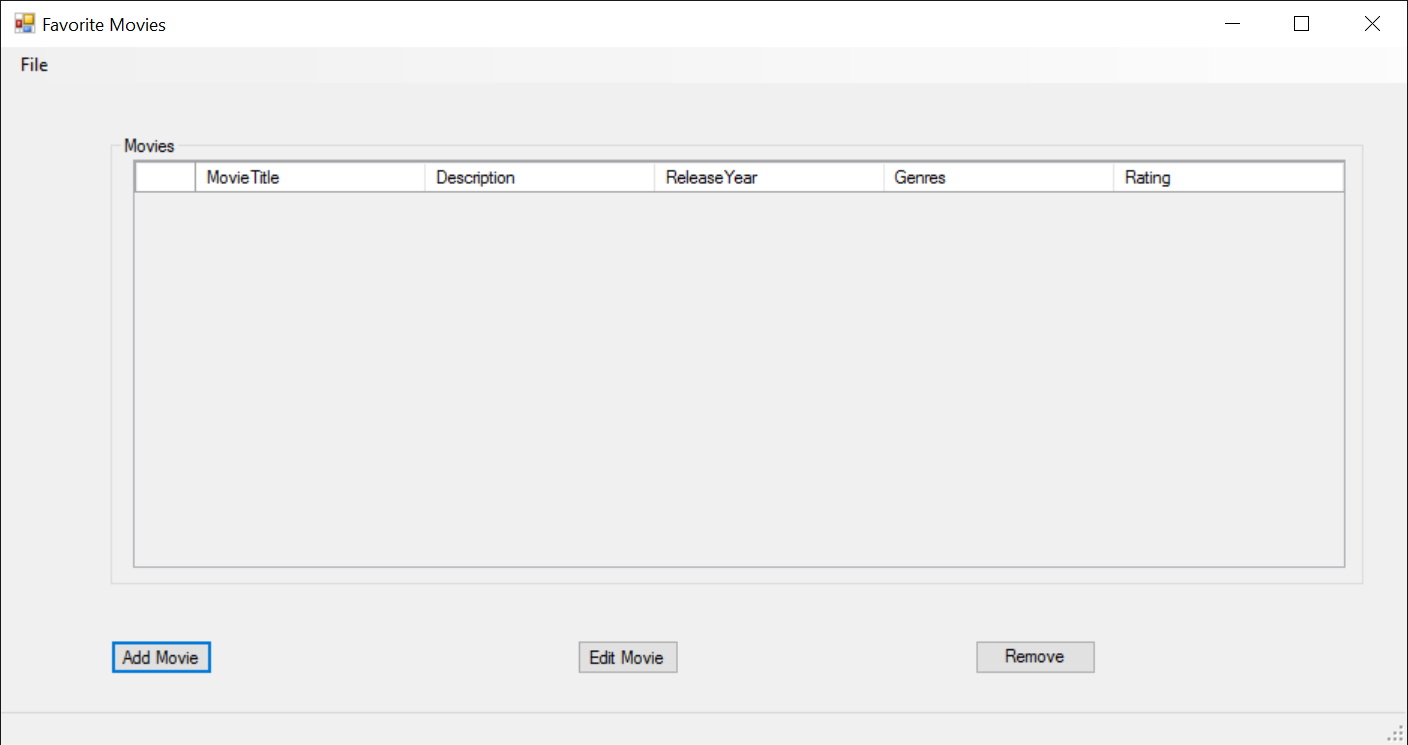
*The application shall list the year released (inputted by the user) of the movie.*

*The application shall connect to SQL database for its ability for persistent storage*

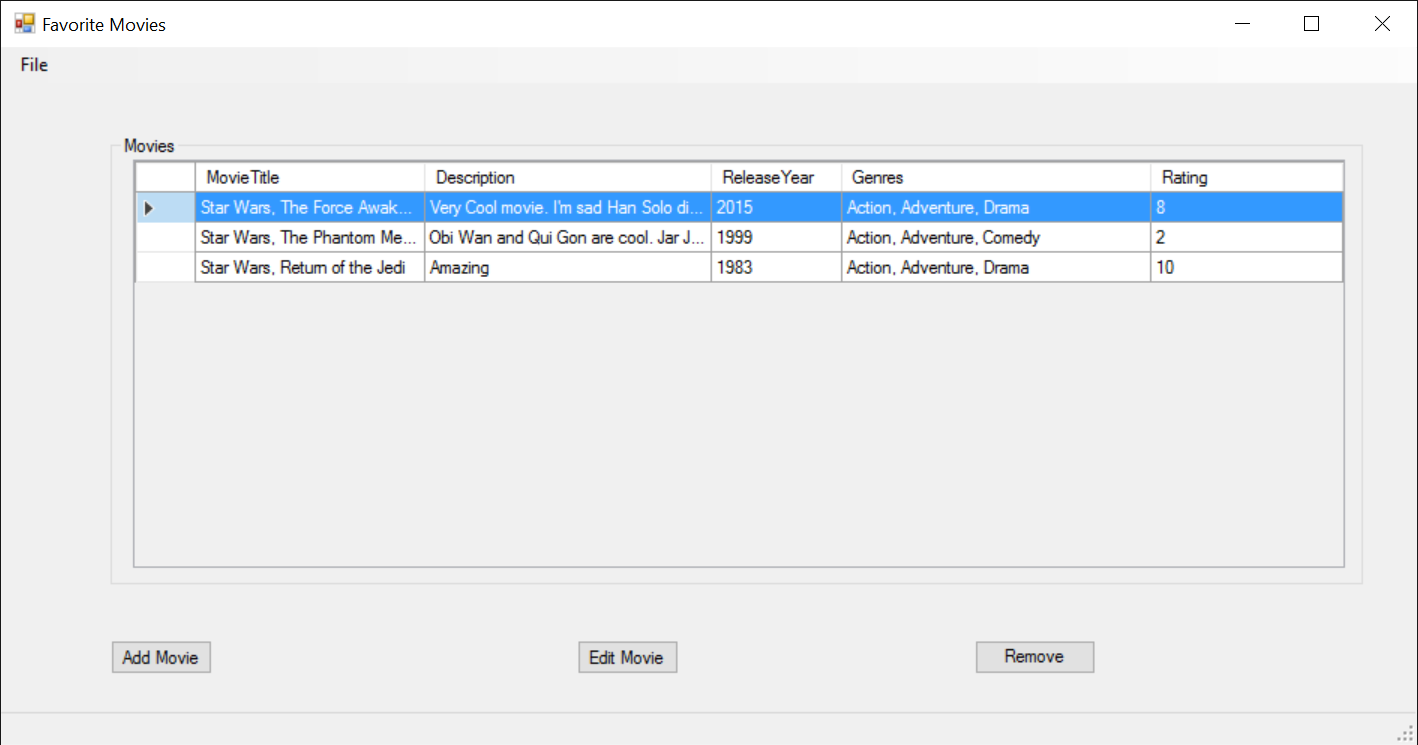
\*Every requirement was met with this project.

Screenshots:

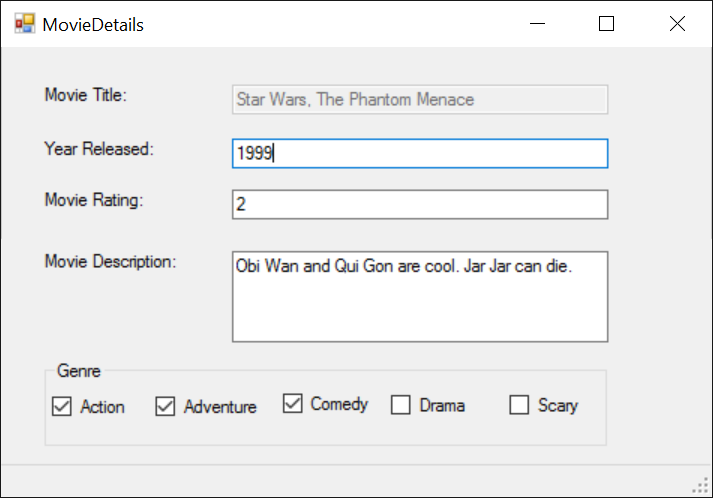
Below is the start of a fresh application with no movies in the database:



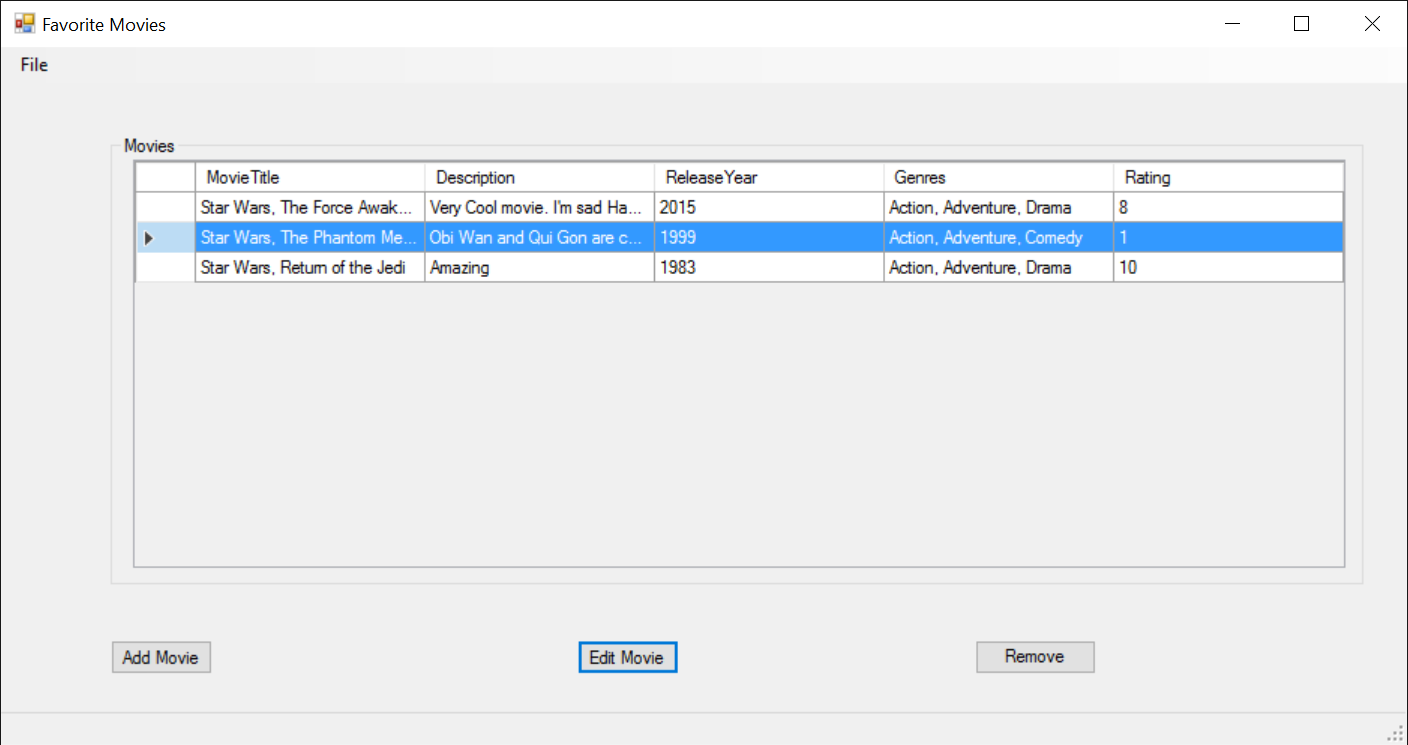
Below, I added a few test movies.



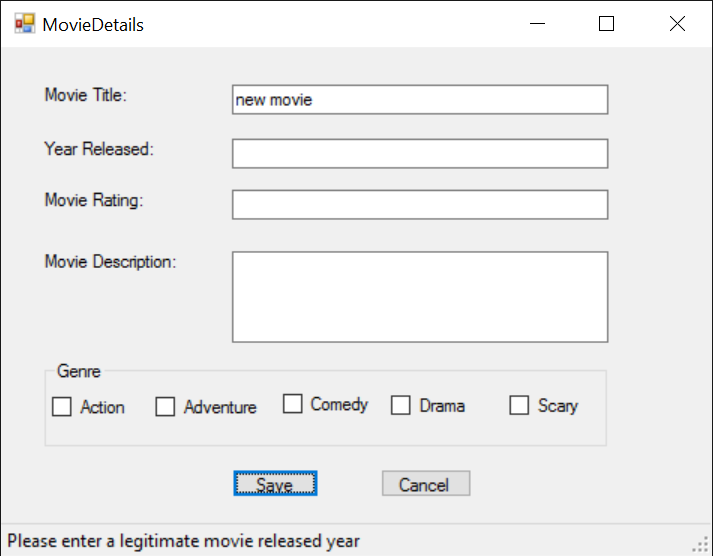
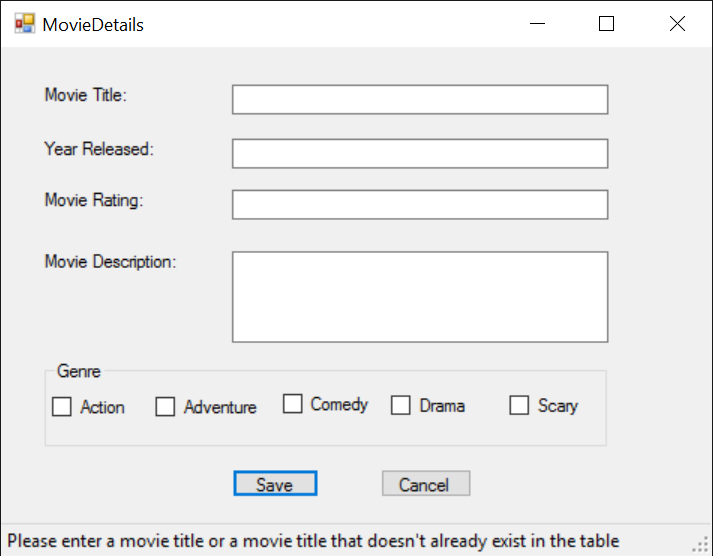
Here is the Movie Details view, this is where you can edit the selected movie or add a new movie by clicking the add movie button.

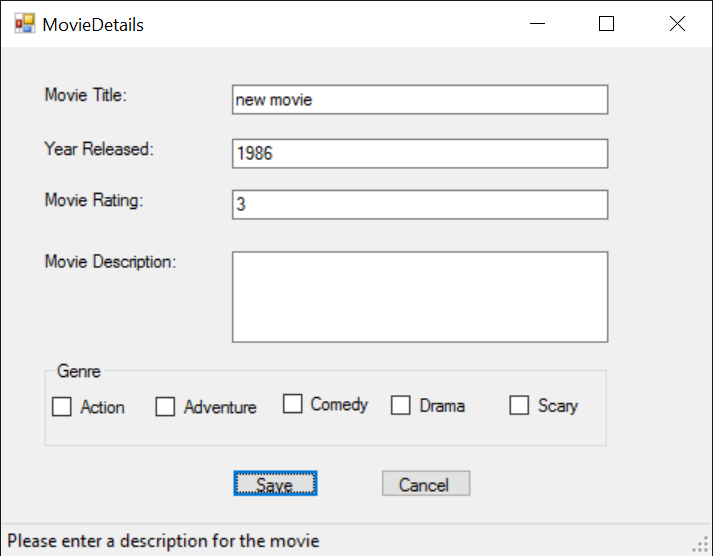
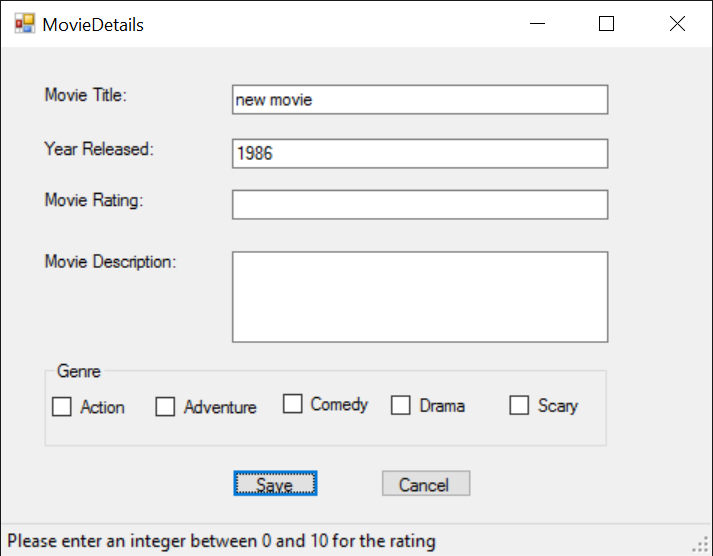


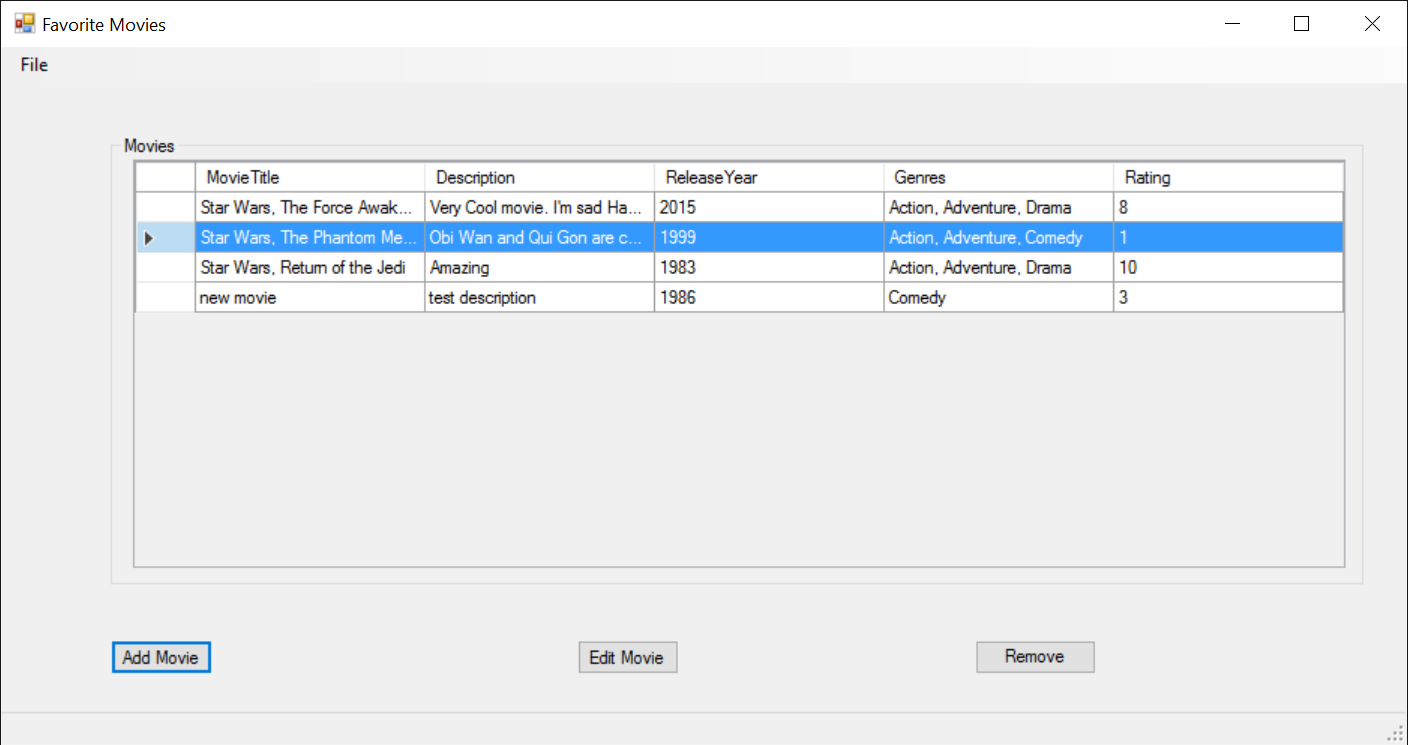
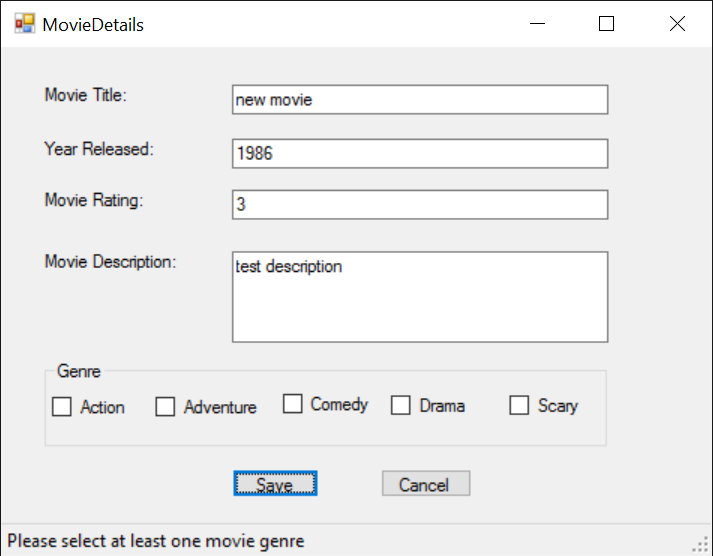
I then changed the movie rating to 1



When adding a movie, there is input validation for every field.

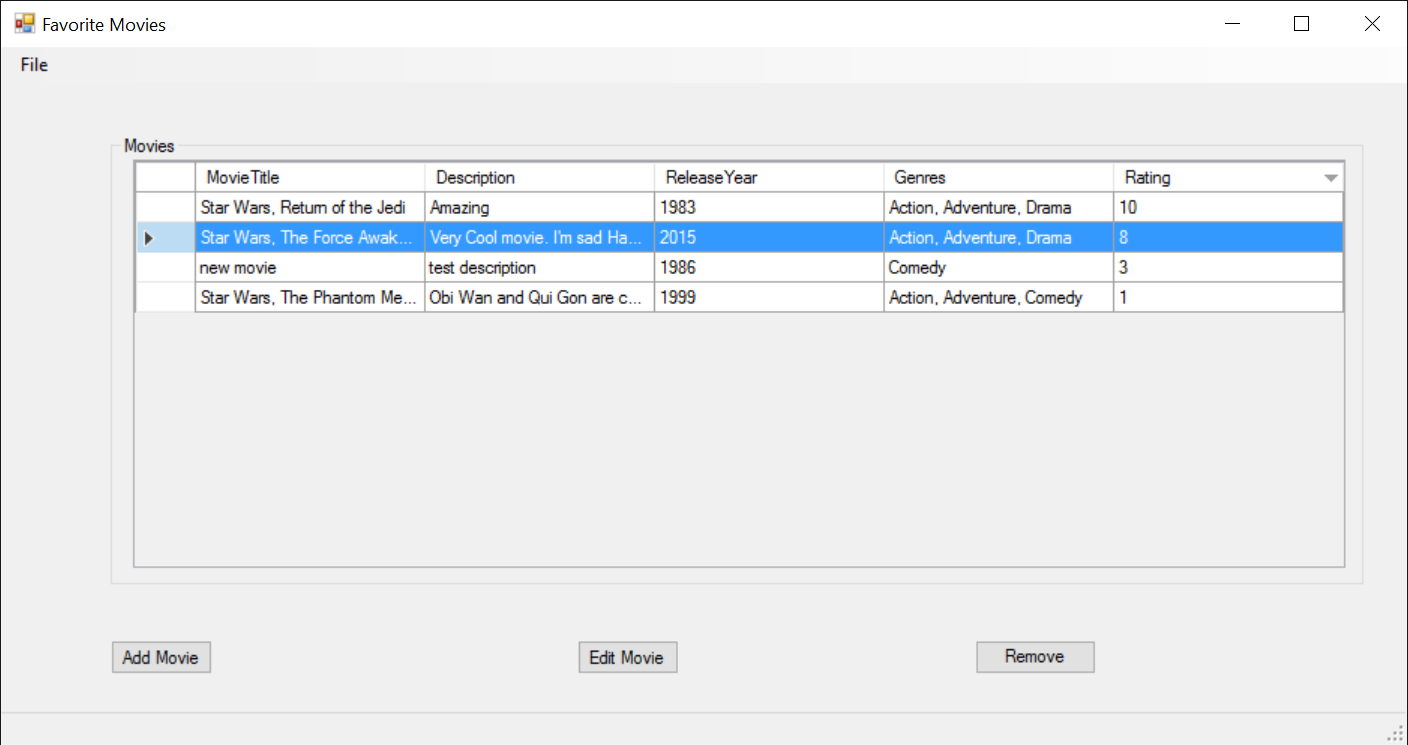






There is SQL Exception handling for all queries made on the database.

You can then sort the table by rating, title, description, release year, genres (using built in sort for the columns)



You can also remove movies

