Deadline: Friday, May 27, 2022, until 2pm (14:00)

Yaşar University
Spring, 2021 - 2022
SE 2224 - Software System Analysis
Asst. Prof. Dr. Deniz Özsoyeller

Final Project

Notes:

- **Submit your Project to Sakai “Assignments”. Do not send your project via email!**
- **Late submissions (submissions after the deadline) will not be evaluated.**
- **This is an individual project (not a group project).**
- **The project requires the completion of three parts: 1- Report [50 points], 2- Implementation (Coding) [40 points], 3- Presentation [10 points].**

Requirements:

The project should be designed as a **Desktop Application (with GUI)** developed with **Java** programming language. Note that you are not designing a Web Application or a Mobile Application.

Introduction:

In this project, you will develop a software application called “**FavoriteMovies**”. Suppose that the application will run on your home computer (laptop or desktop), and be used by your family and friends. The main goal of the application is to keep a list of the movies that are watched together, also to rate them. Assume that the people who watch a movie together will each rate that movie.

There should be 3 lists named **Movies**, **Stars**, and **Ratings**. Each list is stored in a text document (.txt file). The data in each list is separated by a space character. There should be a new line for each new entry. Below, you can see the data names and types that should be stored in each list. The appropriate value should be inserted in place of each data during insertion to a list.

- Movies(**movieId**: int, **title**: string, **year**: int, **length**: int, **genre**: string, **studioName**: string, **producerName**: string, **shortDescription**: String, **avgRating**: double, **numWatched**: int)

  Movies list keeps information about the movie. avgRating can be initially set to a default value such as 0 and later be updated. numWatched shows the number of times that the same movie is watched.

- Stars(**movieId**: int, **movieTitle**: string, **movieYear**: int, **NameSurname**: string)

  Stars list keeps information about the leading stars in a movie. Assume that at most 2 stars per movie will be entered.

- Ratings(**movieId**: int, **NameSurname**: string, **relation**: string, **rating**: int)

  Ratings list keeps information about the ratings of your family and friends for each movie you watch together. In this list, you keep the information about the person who rates the movie. Rating value is from 1 to 10.

Below, you can see the detailed information about the required functions.

A login JFrame Form:

Add a Jframe form for a user login with only the username. The form will have a JTextField component for a username. It will also have a JButton component. When the button is clicked, check if the username in the text field is correct or not. If the username is correct, another JFrame that enables the **functions** explained below will be visible; otherwise, a message dialog box with the text "incorrect username" appears. Consider that your family and friends are the authorized users. The usernames can be stored in a data structure such as an array in your code or in a separate text document.

Functions:

1. **Add a movie to the Movies list.** This function gets the data from the user via JTextFields and adds it to the Movies list. The function should check the list before adding a new entry to avoid any duplicates. The function should also check the last moveId in the list and increment it by one for the new movieId entry. Do not create a JTextField to enter the movie id. When the insertion is completed successfully, the function shows the new bookid in a dialog box.
2. **Add a star to the Stars list.** This function gets the data from the user via JTextFields and adds it to the Stars list. The function should check if the movieId exists in the Movies list before inserting into the Star list. If the movieId does not exist, then the data should not be added to the Stars list.
3. **Add a person and the person's rating for a specific movie to the Ratings list.** This function gets the data from the user via JtextFields and adds it to the Rating list. The function should check if the movieId exists in the Movies list before inserting into the Rating list. After each entry, the function should find the movies with the same movieId, calculate the average rating of the movie and update the avgRating field in the Movies list for this movie.
4. **Show movie id by the given title.** This function shows the movieId for the given movie title from a JTextField.
5. **Increase the number of times that the movie is watched.** For the movie id and the new numWatched value input from JTextFields, this function updates the field numWatched in the Movies list.
6. **Show the image of a movie poster with the given movie id from a JTextField.** Assume that: The images will be named as "poster1.jpg, poster2.jpg, poster3.jpg, ..."; The number at the end of the "poster" string is same as the movieId.; The directory path of the poster images (location of the poster images in the computer) are the same. You do not need to find the movie's actual poster image. Instead, you can simply create temporary "poster" image files (in jpeg format) yourself as shown below:

7. **Show the title of the highest rated movie for a specific genre.** This function shows the highest rated movie according to the selected genre by the user via JComboBox.
8. **Show the movies of a star.** This function shows the movies (movie titles) that a star was in. The user inputs the star name from a JTextField and the function shows the list of the movies that the star was in.

Note: How you design the GUI (Graphical User Interface) of your project is optional.

Report:

Use the **project template** in Sakai to prepare your project report. Each section in the report should be completed.

Note that you need to use a **software tool** such as Visual Paradigm to draw the UML Diagrams required in the project.

Presentation:

- Prepare a voiced powerpoint presentation.
- Prepare and present each slide in English.
- Add slides for the introduction of your project and the outline of your presentation.
- Add the screenshots of your project's GUIs and show a sample output for each function and explain how they work.
- You do not need to add your UML diagrams to your presentation.
- Your presentation should not exceed 11 slides and 7 minutes.