Programming Project / C++ – Watched Movies Management Sytem –

Student 1: David Cojocaru

Student 2: Victor-Daniel Luca

I. Task Description

Student 1 is responsible for managing:
-The movie database: printing, searching movies

Student 2 is responsible for managing user interactions:
-Registering a user
-Adding/removing movies to/from a list
-Viewing lists

II. Data Structures Used by the Team

The following classes will be used:

- Movie: string id, string title, string url, float rating, int runtime, int year, string genres, int numVotes, string directors
- User: string name, List toWatch, List watched
- List: string name, vector<string> items
- MovieManager: vector<Movie> movies

III. File Structure

The following files will be used:

Version 1(CSV files):

users.csv
A file where are stored the credentials for each user:
Username (primary key)

movies.csv
Stores the complete list of available movies. Each row represents a movie:
Position, Const, Title, Url, Runtime (mins), Year, Genres, Num Votes, Directors

<username>_toWatch.csv
Stores a list of movies for a specific user:
Title

<username>_watched.csv
Stores a list of movies for a specific user:
Title

IV. Interacting with Executables

Application 1 will offer the following options:

```
./main1.exe viewMovies
```

View all available movies

```
./main1.exe addMovieToCsv <id> <title> <url> <rating> <runtime> <year> <genres> <numVotes> <directors>
```

Add a new movie

```
./main1.exe search <field> <value>
```

Search movies by different fields

Application 2 will offer the following options:

```
./main2.exe help
```

Display all possible commands.

```
./main2.exe registerUser <userName>
```

Register a new user

```
./main2.exe viewAllUsers
```

View all users (names and lists).

```
./main2.exe viewUser  <userName>
```

For a given user view his lists.

```
./main2.exe addMovie <title> <user> <list_name>
```

Add a movie to a list.

```
./main2.exe deleteMovie <title> <user> <list_name>
```

Remove a movie from a list.