CSE 2102: Introduction to Software Engineering

Project – Video Inventory System

Assigned: March 23, 2023, Due: Deadline for each sprint is noted below

In this project, you will build an interactive system (command-line based) for managing the inventory for a video streaming service, such as Netflix. This system will be built incrementally following a process that is similar to the agile approach. Each sprint will be 1 week long, and will culminate with a demo in the lab.

Sprint 1 (Deadline: 03/30/2023) (100 pts.)

Browse the data in the file netflix.csv. In the first sprint, you will build a data model for the video inventory system in the following steps:

- Implement classes to model Netflix titles. These classes must be structured according to an inheritance hierarchy that is at least two levels deep. Discovering the relationships between the attributes of the titles that will allow you to organize these titles into an inheritance hierarchy is an important component of this sprint.
- Implement a database/data container class that can hold a collection of these titles.
- Implement the necessary accessor and mutator methods in each class.
- Implement test drivers to test the above classes.
- Draw a UML class diagram to represent your data model. The diagram should include all the details, including the attributes and methods.

Sprint 2 (Deadline: 04/06/2023) (200 pts.)

In this sprint, you will implement the following:

- Prompt the user for the name of the file that contains data on the titles.
- Read data from this file, and build a database of titles using the database/data container class implemented in Sprint 1.

  Implement a high-level, interactive user interface that prompts the user for the following options/use cases. The main menu should be displayed in a loop, that is, when one use case is completed, the user is prompted for another one until the user chooses to exit.
  - Add a title
  - Delete a title
  - Search for titles
  - Modify a title
  - Exit

```
Please enter the name of the input file: netflix_titles.txt
1. Add a title
2. Delete a title
3. Search for titles
4. Modify a title
Type Exit to end program
```

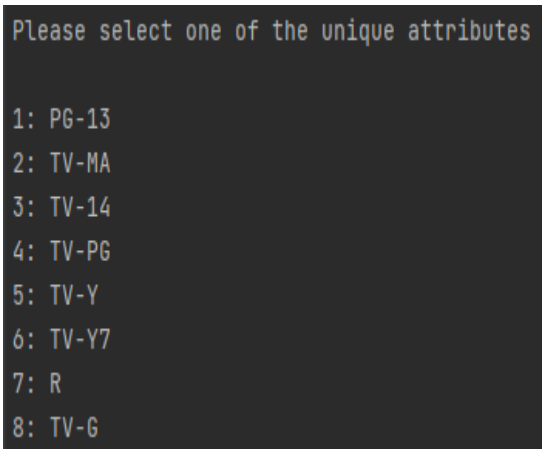
- In this sprint, you will also implement the Search use case.
- Once the user selects the search use case, the user should be further prompted to input whether the title of interest is a TV Show or a Movie.
- Once the user has selected the type of title, they should be prompted to select the specific attribute that would be the subject of their search. Users can search for titles based on their rating, director, genre, duration, country, and year.

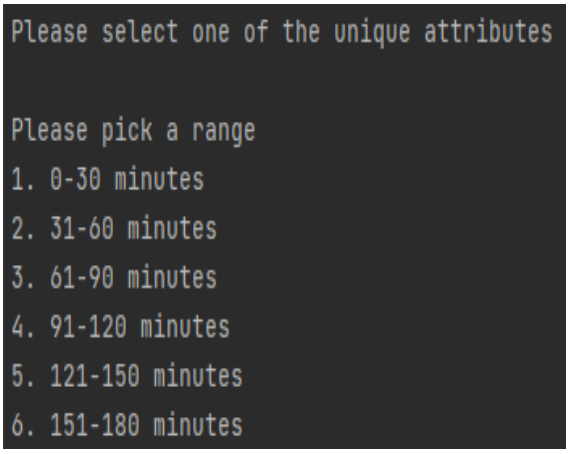
-

```
Are you looking for a Movie or Show?
Movie

Which attribute are you searching based on?
1. Rating
2. Director
3. Genre
4. Duration
5. Country
6. Year
```

Finally, the user should be prompted to input a specific value of the search attribute. The user should be shown all the unique values for the attribute selected in the previous question. All the unique values for each attribute are not pre-determined and should be collected on the fly while reading data from the input file. Duration is given by the number of seasons for TV shows, and in minutes for movies, increasing by 30 minute intervals. So, users can search for movies with duration between 0 and 30 minutes, 31 and 60 minutes and so on.

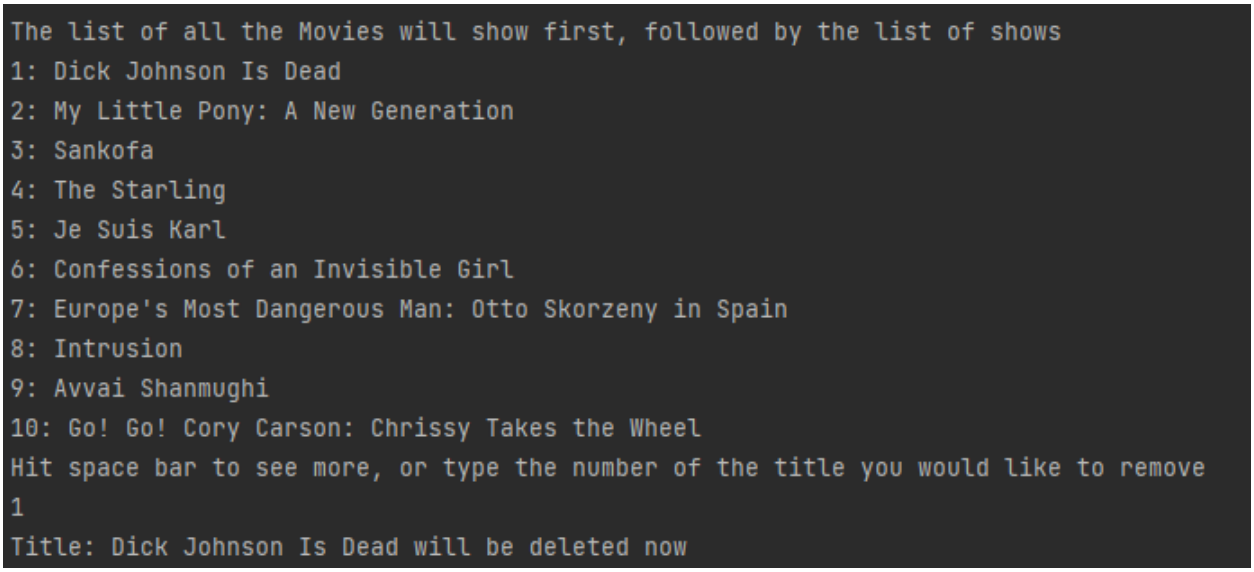




Use the data in the file netflix.csv to test your implementation.

Sprint 3 (Deadline: 04/13/2023) (100 pts.)

In this sprint, you will implement the Delete title use case. After the user selects the delete option, the system will display all the available titles and ask the user to enter the index number of the title that they wish to delete. The list of titles should be displayed one screen at a time, with the user hitting the space bar to proceed to the next batch.



Use the data in netflix.csv to test your implementation.

Sprint 4 (Deadline: 04/20/2023)

In this sprint, you will implement the Add title use case. After the user selects the add option, the system will prompt the user to enter all the attributes of the new title.

```
Enter the Type(Movie or Show): TV Show
Enter the title: Marquis's Adventure
Enter the director: Marquis Lewis
Enter the country: United States
Enter the release year: 2023
Enter the rating: TV-MA
Enter the duration(min if Movie or Seasons if Show): 1
Enter the genres(Seperated by ", "): Documentary
```

Sprint 5 (Deadline: 04/27/2023) (200 pts.)

In this sprint, you will implement the Modify title use. In this use case, the user will be prompted for the index of the title they would like to modify (similar to the Delete use case). Since most of the attributes of a title are determined at the time of release, the only attribute that can be changed is the rating of the title. Thus, the user should be prompted to enter the new rating.

```
The list of all the Movies will show first, followed by the list of shows
1: Dick Johnson Is Dead
2: My Little Pony: A New Generation
3: Sankofa
4: The Starling
5: Je Suis Karl
6: Confessions of an Invisible Girl
7: Europe's Most Dangerous Man: Otto Skorzeny in Spain
8: Intrusion
9: Avvai Shanmughi
10: Go! Go! Cory Carson: Chrissy Takes the Wheel
Hit space bar to see more, or type the number of the title you would like to change the rating of
1
Enter the new rating for the movie you choose
R
Rating for Dick Johnson Is Dead is now R
```

Delete, Add and Modify use cases should ultimately make permanent changes to the database. Another goal of this sprint is to make these changes persistent. That is, after the user has deleted or added a title, the database (for the sake of this project, flat file) should be updated in persistent storage. These new contents should be loaded the next time the application starts up. Use the data in netflix.csv to test your implementation.

Deliverables:

The following deliverables must be submitted on HuskyCT by midnight for each sprint:

a) Java and class files, Java files should be well-documented. Please include files from prior sprints as necessary to run the use case implemented in the current sprint. For Sprint #1 submit a UML class diagram of your data model.
b) You should test your code with netflix.csv file posted on HuskyCT. The expected output for each use case is also posted for comparison.
c) Please make sure that your code compiles, you will demo your code in the lab for each sprint.
d) Late submissions (without any legitimate excuse) will incur a penalty of 10% per day.