**Team 1 Project 1A**

Project 1A – Movie Management System

**Luke Janis**

**Dawson Ploudre**

**Yeseon Heo**

**March 5th, 2023**

**Table of Contents**

1. System Design 3
2. UML Diagram 4
3. Test Cases 5
4. Team Member Contribution 7
5. Possible Improvements 7

1. **System Design**

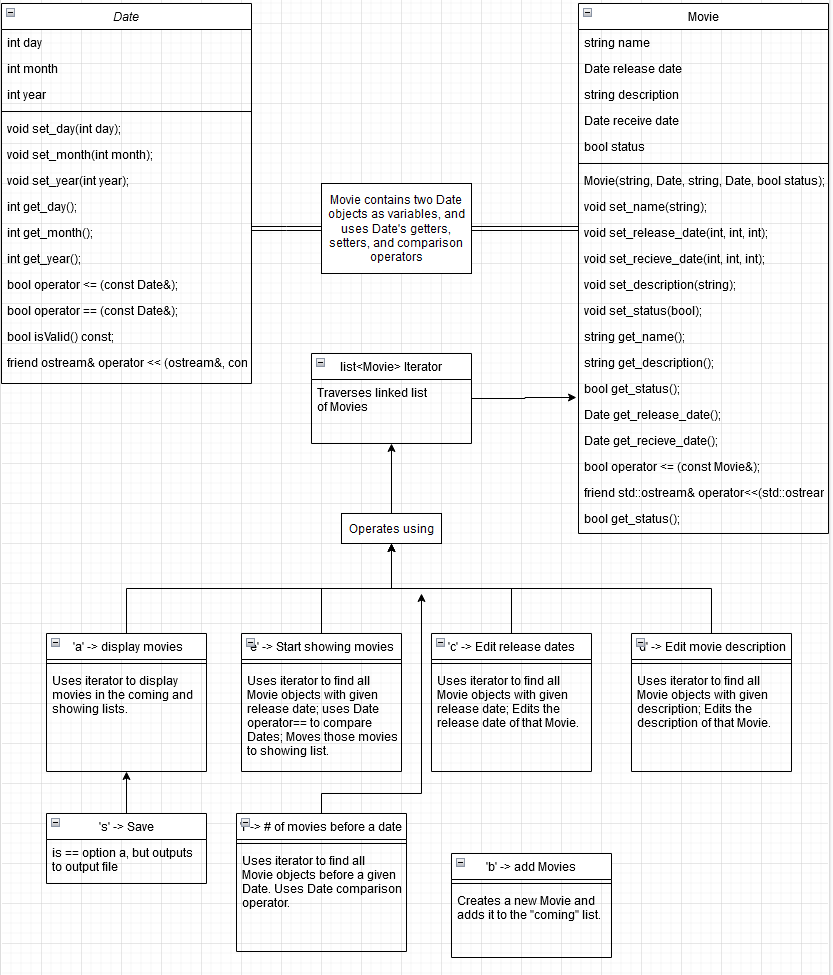
Our system implemented to create two lists (1. Showing and 2. Coming) to classify movies and modify information about movies. Through ifstream, all movies stored in the text file are stored in a double-linked list. Each movie includes name (string type), release\_date (date type), description (string type), receive\_date (date type), and status (bool type). The status is divided into released or received according to the status of the movies.

When the program starts, the menu is displayed in the console, and users can use the functions built into the menu by entering the desired options.

1. Display movies: Option a is a feature that shows the movie stored in the showingList and comingList. The comingList is sorted according to release\_date.
2. Display movies: Option a is a feature that shows the movie stored in the showingList and comingList. The comingList is sorted according to release\_date.
3. Edit release dates: Option c is a feature that allows you to change the release date of a movie stored in the comingList. If the movie matches the title of the movie you want, you can change the previously stored release\_date to a new release\_date.
4. Edit movie description: Option d can change the description of the movie stored in the comingList. If the title of the movie you want to modify is valid, the saved description can be newly changed.
5. Start showing movies in the theater: Option e is the ability to move movies stored in the comingList to the showingList when you enter a specific date. After selecting the movies that correspond to the date entered first, the movies are erased from the comingList. However, this function does not work if the movie on the date you entered is already in the showingList or the release\_date is not valid.
6. Number of movies before a date: Option f is a function that tells you how many movies are saved in the comingList when the release\_date is earlier than the date entered.

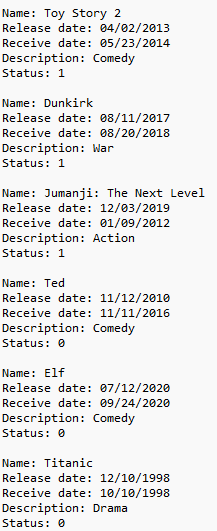
Lastly, s(save) function is a function that overwrites the values of newly added or modified movies and allows users to continue storing data through loop statements until q(quit) is entered. The q(quit) function terminates the program when entering q as an option.

1. **UML Diagram**

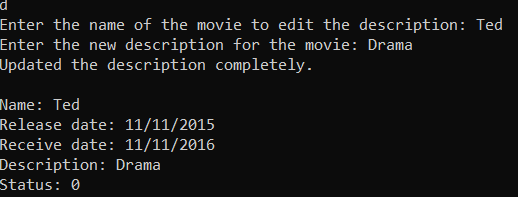
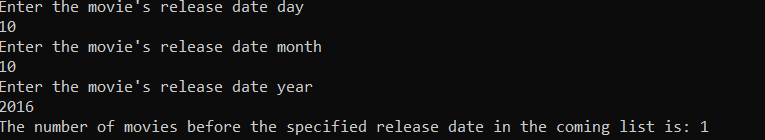


1. **Test Cases**

1) For the first test case, we wanted to test changing a release date and adding a movie. First, we changed the release date of Ted to 12/11/2010. Then we added a movie with name Titanic, release date 12/10/1998, receive date 10/10/1998, description: Drama, and status as 0. Then we saved the changes which puts our lists to the output file. We should expect to see that the lists should be updated with the new release date and movie.



As we can see, the program ran as expected.

2) For our second test case, we wanted to test out the editing description feature and the feature of showing movies in the coming list before a specified release date. I am going to change the description of Ted to Drama, we should expect to see the movie outputted with the new updated description. For showing the movies before a specific date, I will test with the date 10/10/2016, this should output 1 as there will be one movie in the coming list with a release date before this. 

As we can see, both tests were performed successfully.

1. **Team Member Contribution**

Luke Janis:

* Hosted Github repository and project report document
* Implemented the menu-based system in the console with stubs for each option, exiting with option q
* Worked with Dawson to read input file (before it was able to add in ascending release date order.
* Options a (display movies), b (add movies), and e (start showing movies)
* Added Date operator== to implement option e
* UML Diagram in project report

Dawson Ploudre:

* Wrote the file input system
* Wrote the output operators for both dates and movies
* Wrote the <= operator for both dates and movies
* Wrote the function that displays the number of movies in the coming list with a release date before the specified date as well as the save function
* Performed the test cases for the project report

Yeseon Heo:

* Write System Design for project report.
* Write the initial Date and Movie classes.
* Options c (edit release date of the movie) and Option d (edit the description of the movie).

1. **Possible Improvements**

* Variables could be named and described more clearly.
* Code for reading input file and adding the new Movie to coming or showing list needs to be more readable.
* The power of Github (branches, etc.) could be used for more of its potential. Github commits could be more descriptive.
* Writing cleaner codes by reviewing and searching to be more effective.