CS 319 - Object-Oriented Software Engineering
Analysis Report

MovApp

GROUP 6

Ahmet Batu ORHAN

Elif ÖZDABAK

Nergiz ÜNAL

Rıdvan ÇELİK

INSTRUCTOR BORA GÜNGÖREN

Bilkent University

July, 8

TABLE OF CONTENTS

1. INTRODUCTION

This report is prepared by Group 6 in the CS319 course. As an object-oriented software engineering project, a Java desktop application called MovApp will be developed. Although this application will be developed for desktop, the main area of usage of this program is touch operated screens inside the planes and busses. Because of the inability of reaching this equipment, implementation will be done on computers for demonstration purposes.

1.1. Purpose

By using the principles of object-oriented software techniques, an application for buses and planes will be developed. This application helps users to find movies and music to watch and listen.

Every day, a massive number of individuals travels to different places thanks to newest technological transportation opportunities. Although in some cases travelling is taking less than a couple of hours, sometimes individuals spend much more than a couple of hours. In other words, there can be a situation that travelers need to spend eight straight hours on busses or maybe on planes. This application's core aim is to help travelers to spend their times entertaining and gratifying on busses and planes.

All the Object-Oriented programming principles and Unified Modeling Language (UML) Techniques, which were taught in this course, will be used.

1.2. Scope

"MovApp" is a movie and music application that aims to help travelers to spend more enjoyable hours while travelling. This application enables everybody to search and watch movies, listening to music and more. The wide range of users is targeted with the help of basic and user-friendly design of MovApp. Moreover, reaching all the sources inside the application without registering the application, will enhance the number of users.

1.3. Overview

"MovApp" is an entertainment application for travelers. It allows the users to search and watch movies, rate movies and lastly bookmark movies for the purpose of watching them on another travel. With the aim of helping the users, movies that were rated highly by the other users will be stored and shown as well. Moreover, "MovApp" gives detailed information about the selected movie. Furthermore, music from different categories with their clips also included in this entertainment application.

Because of the inability of the required equipment, the demonstration will be held on the desktop. The application will be developed by using Java as language. Moreover, this application will have a database for storage purposes. However, it is not clear that which database system will be used inside the application. All the movies, descriptions about movies, rates and user related issues such as user identities will be stored in the database tables. Besides, every user in the database has its own top-rated movies and the movies the traveler will watch later. Therefore, database tables will have close relations.

The application will have a convenient and user-friendly graphical user interface. Even though a huge number of individuals a highly related with the technology, not every individual knows how to use technology. As a consequence, the application must be clear and understandable as much as it can for the targeting wide range of users.

2. REQUIREMENTS ANALYSIS

2.1. Functional Requirements

- Users can use the program as both ‘registered user’ or ‘guest user.'
- If a guest user has an account, she/he can log in via ID and become registered user, then she/he is able to rate movies, edit her/her ‘will watch’ and own ‘top list’.
- If the traveller wants to log in or sign up, there will be buttons on top right of the homepage. In the sign-up page, there will be such information about how to sign up.
- When the program starts, the homepage will be displayed at first. There will be a panel which shows top movies in the homepage. Users also will be able to choose any movie from the homepage.
- The left side of the homepage there is a category list. Users can look at movies as their categories such as action, drama, romantic, etc.
- In order to find a specific movie, there is a search button also on the homepage.
- Registered user will be able to rate a movie from 0 to 5. After the movie is rated, it will be automatically added to user’s top list.
- User’s top list includes all rated movies with descending order as their rates.
- Registered users can add any movie to their Will-Watch list.
- Will-Watch list includes movies which registered user think to watch.
- Users can select and watch movies in the Will-Watch list or User's Top list.
- When the user selects any movie from anywhere, description page will be displayed at first.

- Description page includes movie cover image, summary and explanation, Watch button, add to Will Watch button and rate movie option.

2.2. Non-Functional Requirements

2.2.1. Usability

The program is going to provide users with the ideal user interface. To do this, we are planning not to add any unnecessary features. For example, users may have own accounts to creates an own favourite or will-watch movie list, but it is not obligatory. The program can be used without an account. Also, we will categorise all movies according to their types, in order user to choose which kind of movie she/he wants.

2.2.2. Reliability

Users have to trust the system, even after using it for a long time. Since our database will work locally, it is hard the program to break down or to not respond. If any failure occurs, it will take a little time to handle with the problem since the program well organised and analytic classification.

2.2.3. Performance

Any response time in the program should not take more than 1 or 2 seconds. Our only problematic point is that after pressing "watch" button how much time the beginning of the movie takes. This will be the key issue to be resolved.

2.2.4. Supportability

The program does not require any special needs to maintain except for updating database. Secondly, we are developing the program on Java platform, so that it will have a wide range of use, and support.

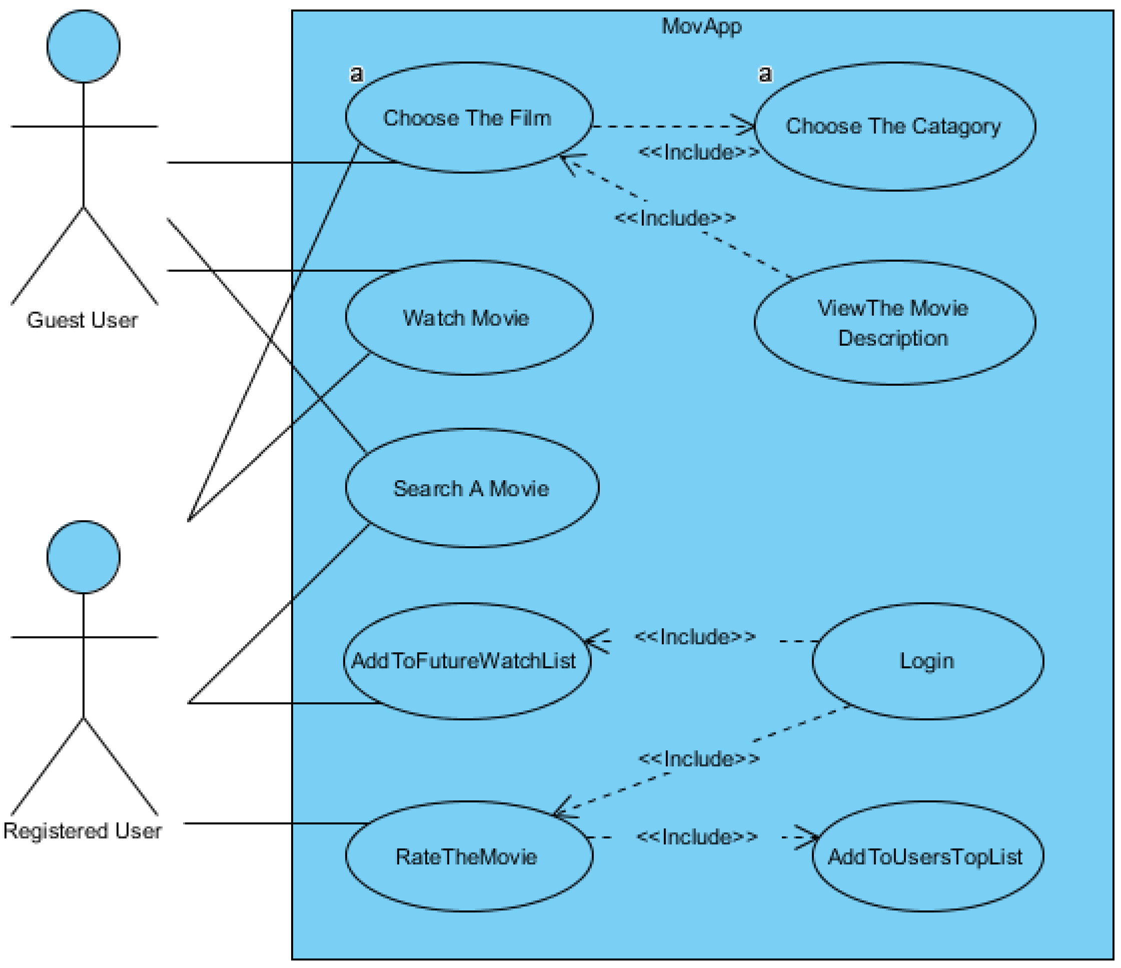
2.3. Constraints

- The program is being developed on Java 8, Net Beans IDE 8.2.
- For now, the program will work on desktop PC.
- The user interface will be in English.

2.4. Organising the Specific Requirements

2.4.1. System Models

2.4.1.1. Use Case Model



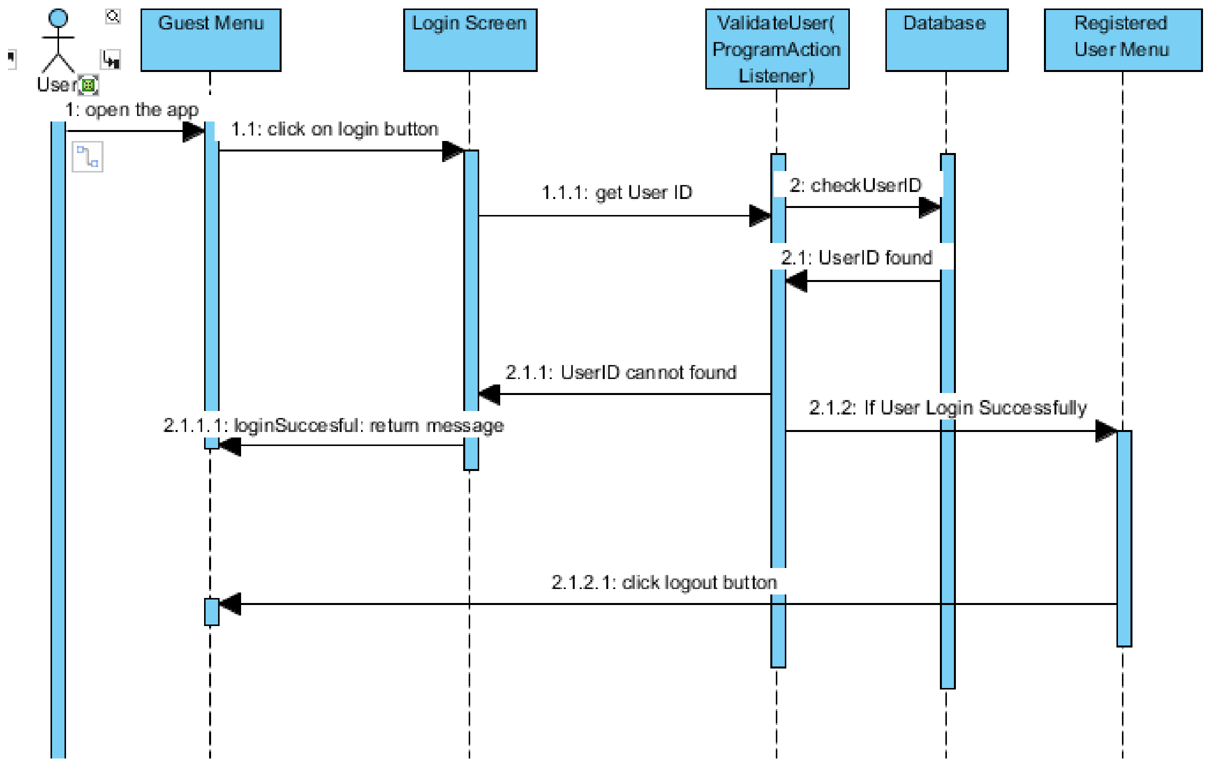
Main use case diagram of our application is above. Through this system, the guest user and registered user have common use cases. The both type of users can search any specific movie with the help of search button at the top of the interface by the way both of them can view the movie description of chosen movie and then watch the chosen movie. Additionally both type of user can select the movie category and then list the selected category's movies. The registered user also have use cases distinct from guest user. The registered user chose the movie by using UsersTopList or WillWatchList. UsersTopList contains movies that are rated by user before, WillWatchList contains the movies that are added this list by user in order to watch later. Thus the registered user can add the movies to own list or can give rate to those movies as an use case.

2.4.1.2. Dynamic Models

2.4.1.2.1. Sequence Diagrams & Scenarios

Sequence Diagram-1

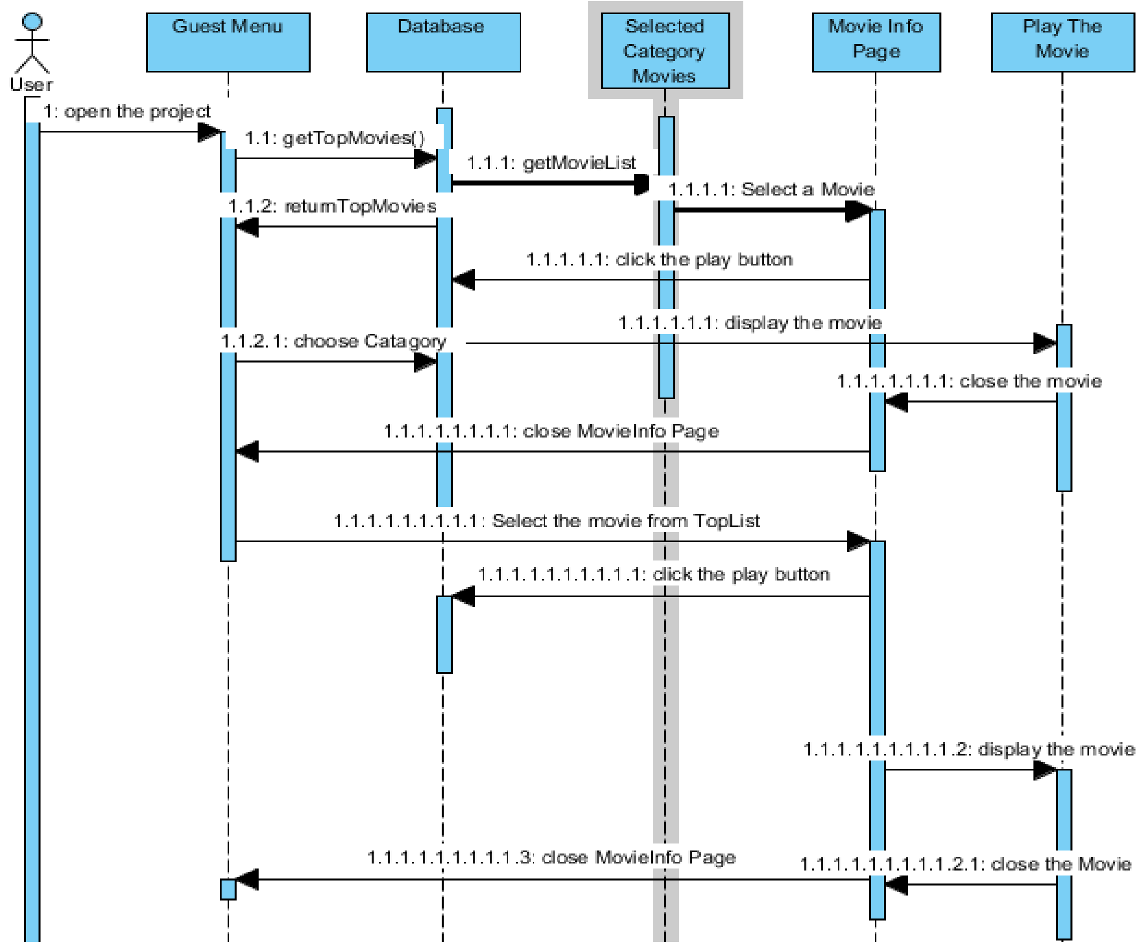
In this scenario, when the users open the application they will see the guest menu of program at the beginning. If user who have registration in the system database can login as a registered user and then reach the registered user menu which provide more sophisticated usage of program. There is an login button at the top of the GuestMenu, after users click on the login button system direct the user login screen, this screen ask for user's id. When user enter their id, system connect to the database system which is store the user's id who have registration in the system. If the user's id match any id in the system(User id found) , the user id will be validated and then user will login successfully.



User
Guest Menu
Login Screen
ValidateUser(ProgramAction Listener)
Database
Registered User Menu
1: open the app
1.1: click on login button
1.1.1: get User ID
2: checkUserID
2.1: UserID found
2.1.1: UserID cannot found
2.1.2: If User Login Successfully
2.1.1.1: loginSuccesful: return message
2.1.2.1: click logout button

Sequence Diagram-2

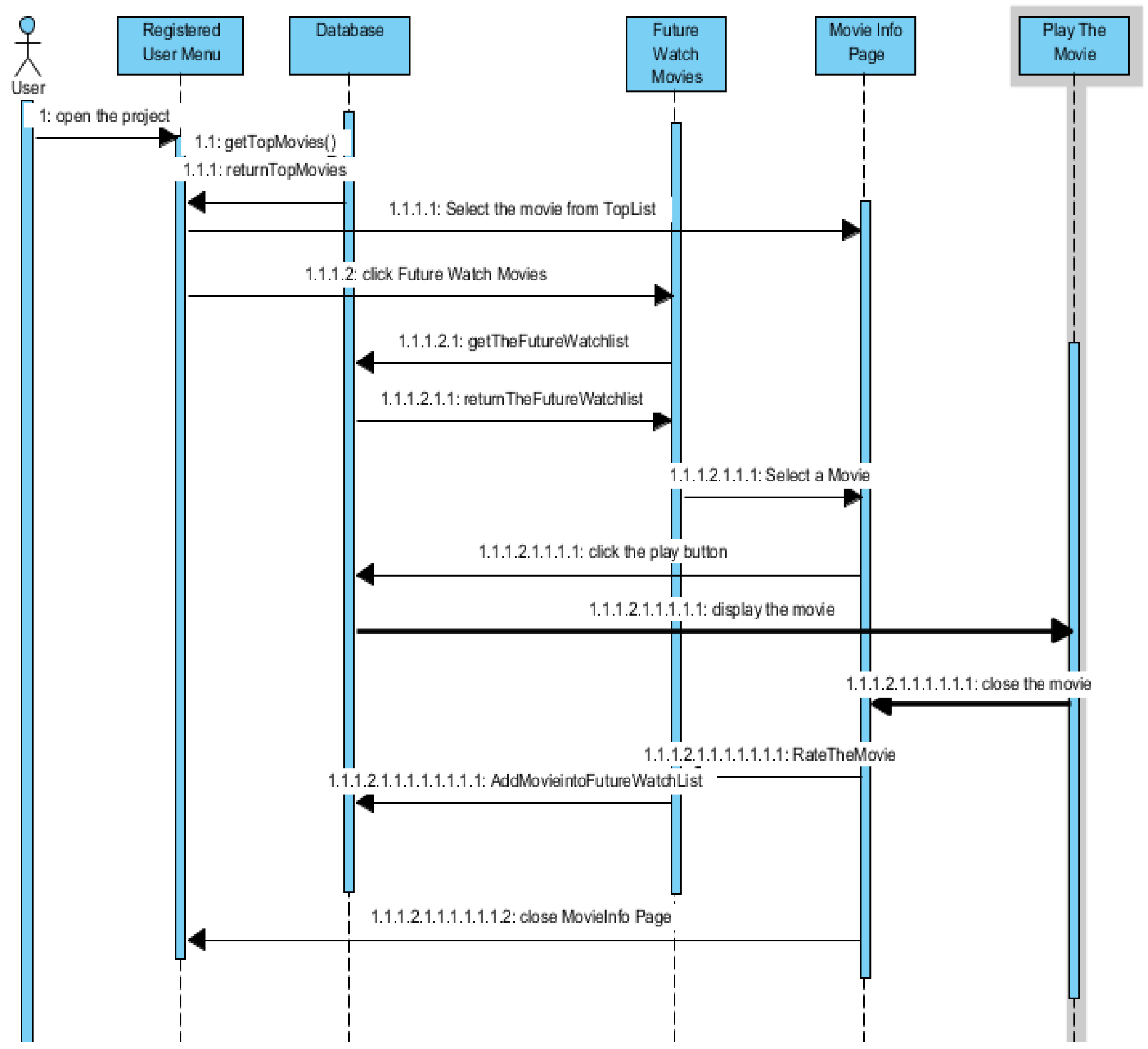
There are two options to use the program, the user can visit the MovApp as a guest user. Guest user has more limited usage of the program than registered user. Guest user can search any movie help of search button and can select movie categories by presented categories on the menu. Login is not needed to being guest user. When users enter the program, at the beginning of the program provide the list of Top Movies by reaching database of the program, (this list ranked according to different user's votes) the list is viewed on the menu and user can chose any movie from this tab. Another scenario in order to chose any movie is, using category list on the menu. Chosen category's movies listed on the menu, to reach chosen list the program connects to the database and then list them. If the user follow either these two path (choosing a movie), Movie information page will displayed on the screen, to get this page database will be used also. User can play the film by the "play" button on the movie info page, the movie will taken from our database . After user click the play button film will be displayed and the main menu cannot be viewed no more. As a last step of scenario when user exit from movie screen, movie info page will be viewed on the screen again.


User
Guest Menu
Database
Selected Category Movies
Movie Info Page
Play The Movie
1: open the project
1.1: getTopMovies()
1.1.1: getMovieList
1.1.1.1: Select a Movie
1.1.2: returnTopMovies
1.1.1.1.1: click the play button
1.1.1.1.1.1: display the movie
1.1.2.1: choose Catagory
1.1.1.1.1.1.1: close the movie
1.1.1.1.1.1.1.1: close MovieInfo Page
1.1.1.1.1.1.1.1.1: Select the movie from TopList
1.1.1.1.1.1.1.1.1.1: click the play button
1.1.1.1.1.1.1.1.1.2: display the movie
1.1.1.1.1.1.1.1.1.3: close MovieInfo Page
1.1.1.1.1.1.1.1.1.2.1: close the Movie

Sequence Diagram-3

This scenario has more sophisticated specifications than the guest user scenario. This time users have a registration on the system. Registered user can do anything guest user do, and can do some additional actions in the program. When user enter as registered user the program takes the usersTopList and WillWatchList from database, the diagram below explains watching a movie thanks to WillWatchList(FutureWatch in the diagram). Due to the both paths are too similar, we draw only the WillWatchList scenario.

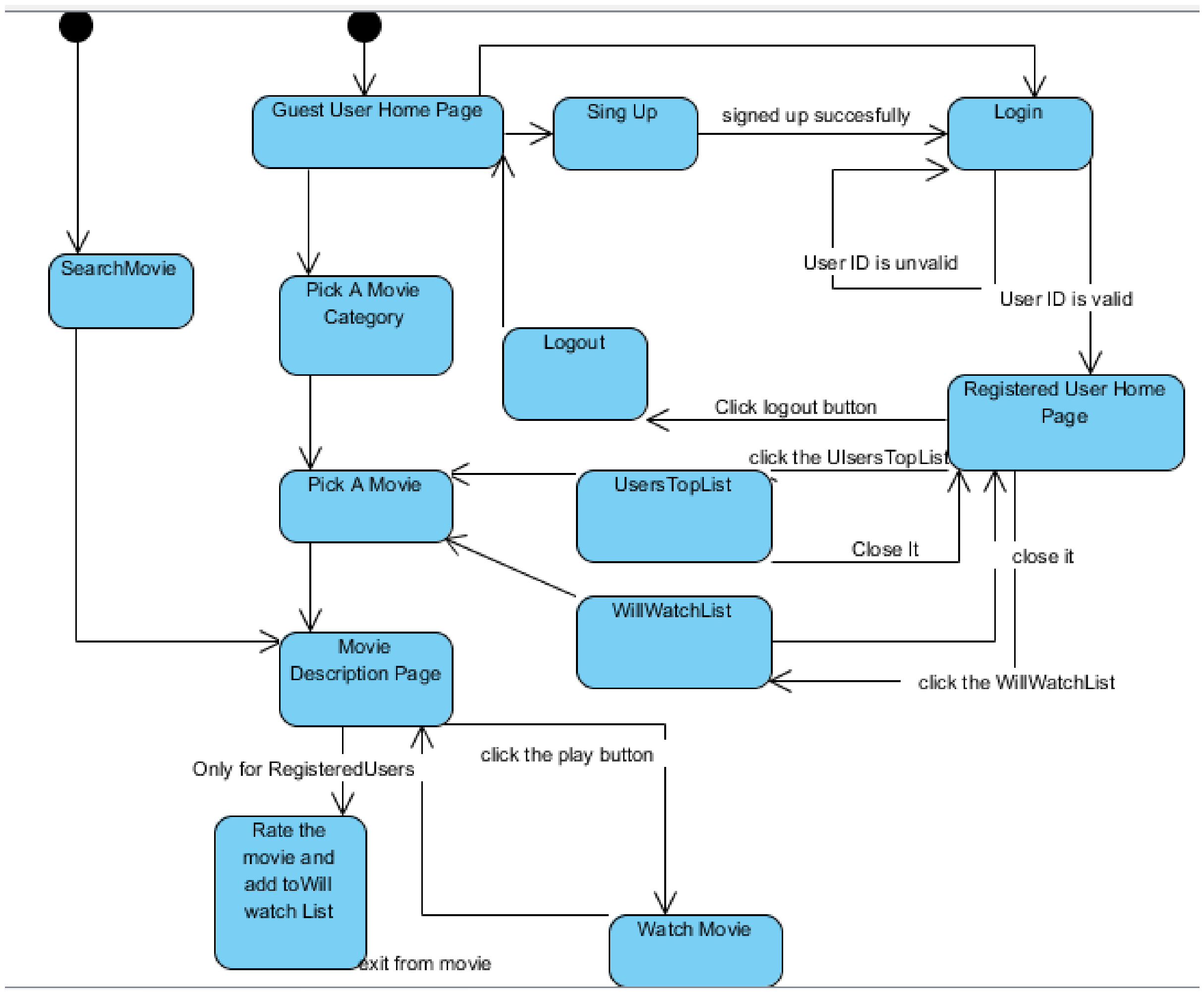
When user click the WillWatch button, it will be listed on the screen and then users can select a movie from this list. Movie information page will displayed on the screen, to get this page database will be used also. User can play the film by the “play” button on the movie info page, the movie will taken from our database. Additionally The Registered User can give ranks to the movies.


User
Registered User Menu
Database
Future Watch Movies
Movie Info Page
Play The Movie
1: open the project
1.1: getTopMovies()
1.1.1: returnTopMovies
1.1.1.1: Select the movie from TopList
1.1.1.2: click Future Watch Movies
1.1.1.2.1: getTheFutureWatchlist
1.1.1.2.1.1: returnTheFutureWatchlist
1.1.1.2.1.1.1: Select a Movie
1.1.1.2.1.1.1.1: click the play button
1.1.1.2.1.1.1.1.1: display the movie
1.1.1.2.1.1.1.1.1.1: close the movie
1.1.1.2.1.1.1.1.1.1.1: RateTheMovie
1.1.1.2.1.1.1.1.1.1.1.1: AddMovieintoFutureWatchList
1.1.1.2.1.1.1.1.1.1.2: close MovieInfo Page

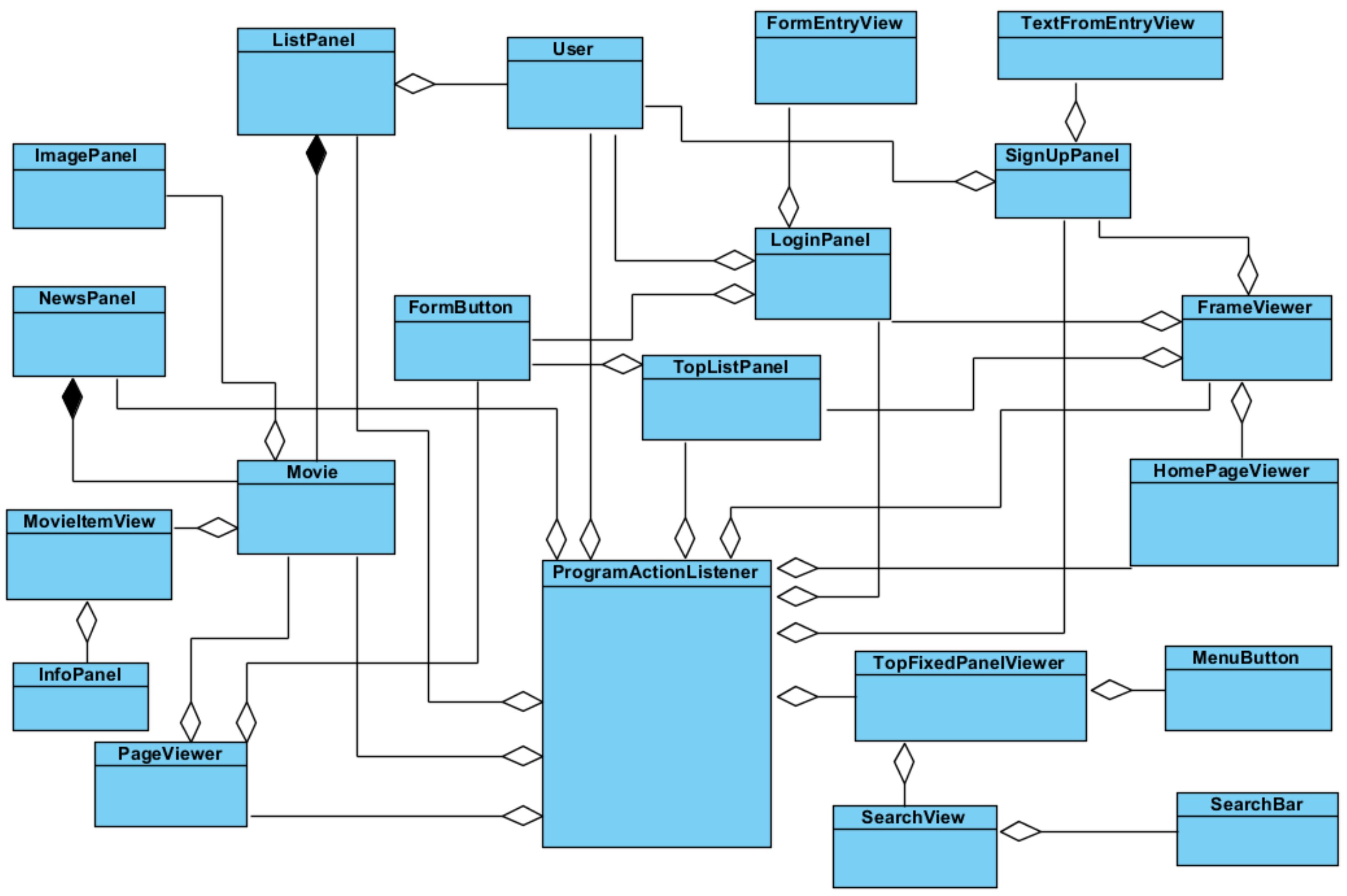
2.4.1.2.2. State Diagram

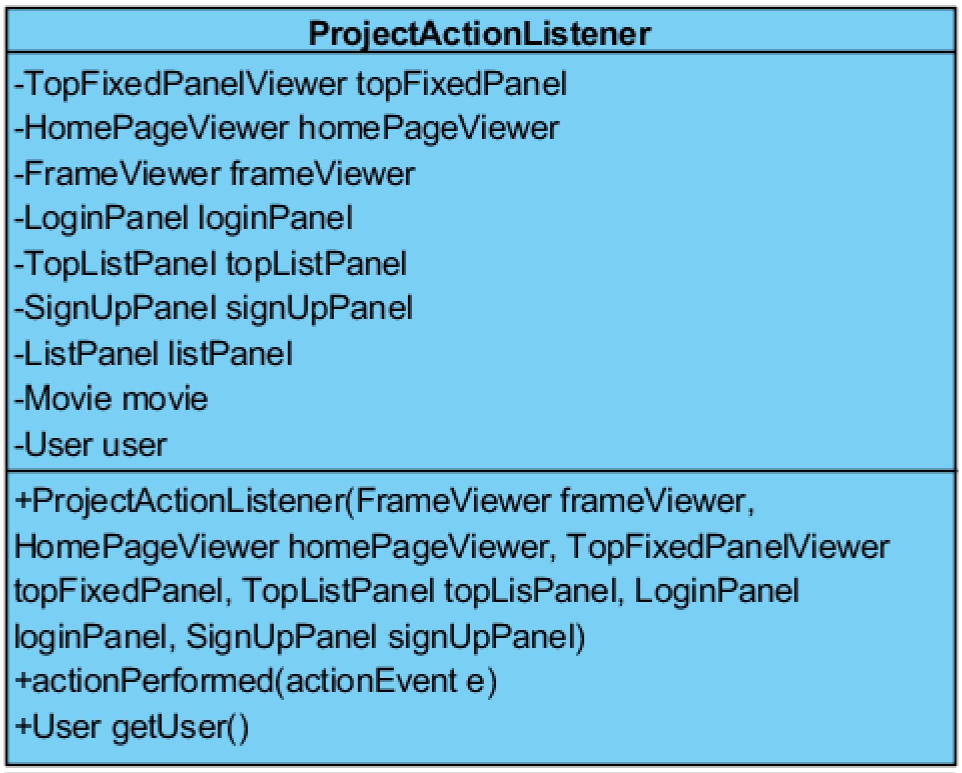
State diagram of MovApp is below, activities and states differentiate for different user types. While Guest user have login and signup activity, Registered Users do not have them because they have already logged in the system.

Movie Description page can be view by many ways : by using category list, search button, from top movie list, from UsersTopList and WillWatchList(Future Watch). Those cases reunion at this state. After this state watching movie process, registered user have give rank activity additional to guest user.


Guest User Home Page
Sing Up
signed up succesfully
Login
User ID is unvalid
User ID is valid
SearchMovie
Pick A Movie Category
Logout
Registered User Home Page
Click logout button
click the UIsersTopList
Pick A Movie
UsersTopList
Close It
close it
WillWatchList
click the WillWatchList
Movie Description Page
Only for RegisteredUsers
click the play button
Rate the movie and add to Will watch List
Watch Movie
exit from movie

2.4.1.3. Object and Class Model



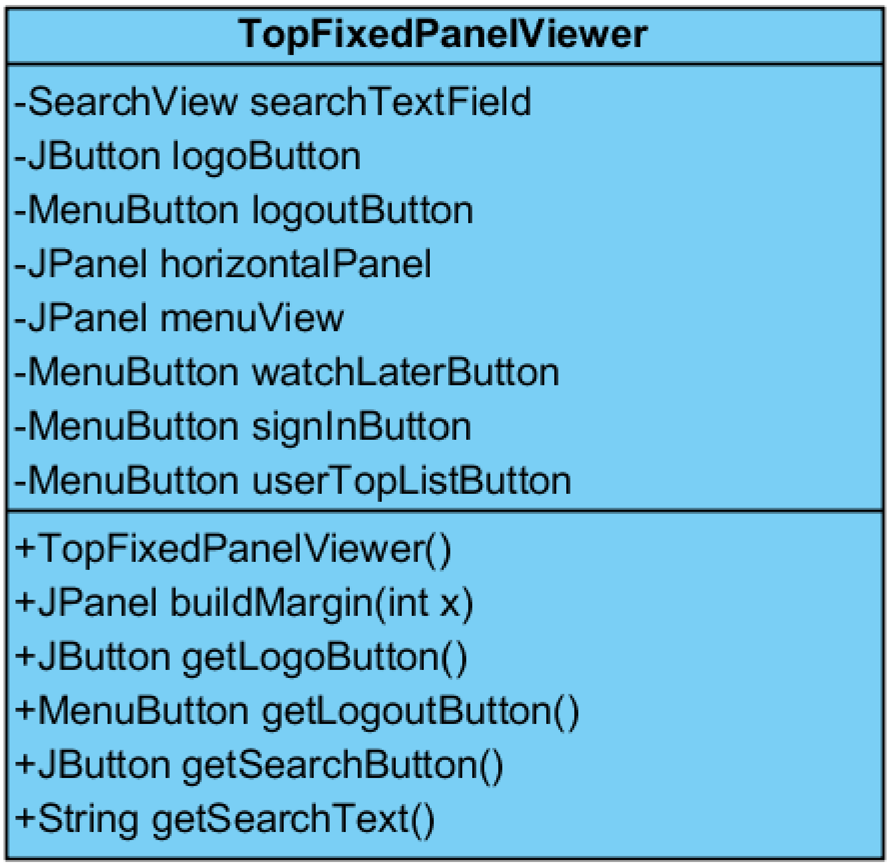


ProjectActionListener

-TopFixedPanelViewer topFixedPanel
-HomePageViewer homePageViewer
-FrameViewer frameViewer
-LoginPanel loginPanel
-TopListPanel topListPanel
-SignUpPanel signUpPanel
-ListPanel listPanel
-Movie movie
-User user

+ProjectActionListener(FrameViewer frameViewer, HomePageViewer homePageViewer, TopFixedPanelViewer topFixedPanel, TopListPanel topLisPanel, LoginPanel loginPanel, SignUpPanel signUpPanel)
+actionPerformed(actionEvent e)
+User getUser()

“ProjectActionListener” is the core class of the MovApp project. This class detects all the movements of the user. Then, changes the frames and opens pop-up windows according to user’s movements and demands.

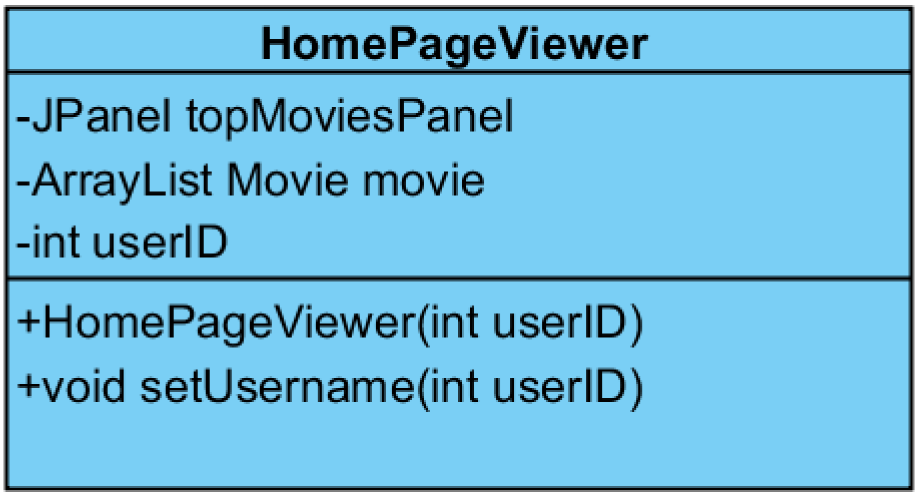


TopFixedPanelViewer

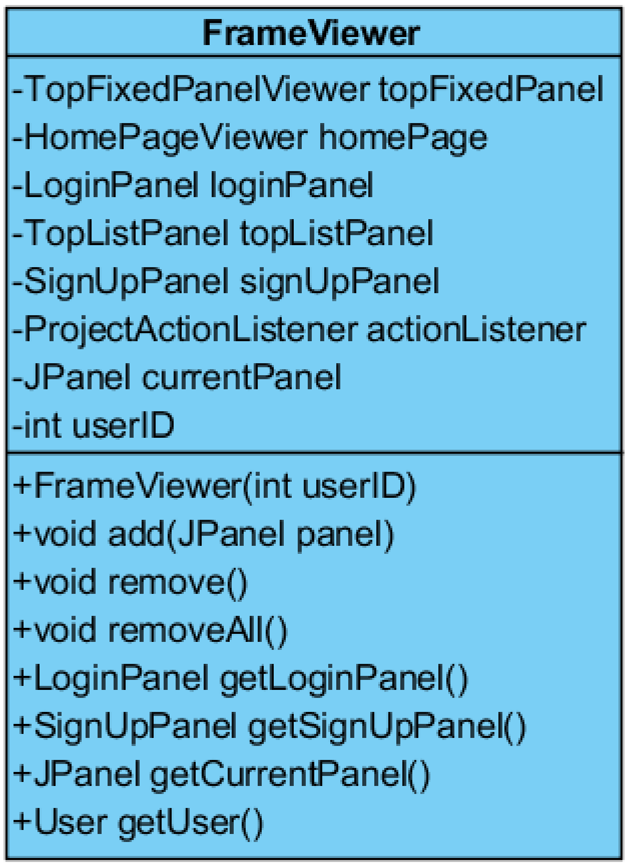
-SearchView searchTextField
-JButton logoButton
-MenuButton logoutButton
-JPanel horizontalPanel
-JPanel menuView
-MenuButton watchLaterButton
-MenuButton signInButton
-MenuButton userTopListButton

+TopFixedPanelViewer()
+JPanel buildMargin(int x)
+JButton getLogoButton()
+MenuButton getLogoutButton()
+JButton getSearchButton()
+String getSearchText()

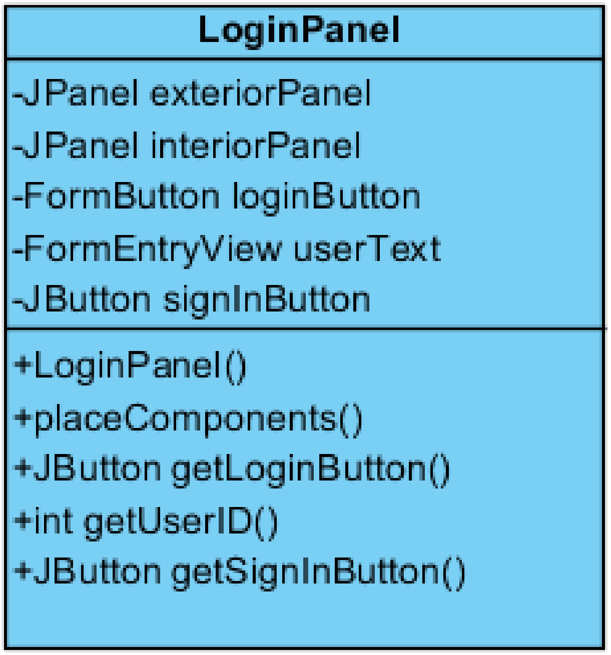
“TopFixedPanelViewer” is the main class that the user’s see on top of the “HomePageViewer”. These two classes are combined and create the Home Page that user’s see when they open the program. Thanks to this class, users can login to the application so they can rate the movies or add the movies to their future lists. Thus they can watch these movies later. Moreover, users can search movies regardless if they are registered or not by using the “Search Box” that autocompletes itself.



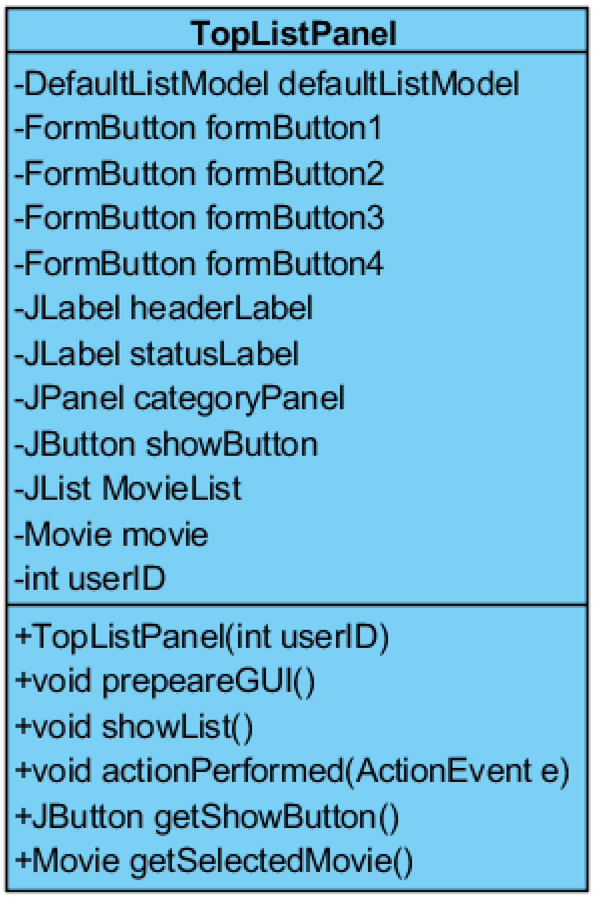
"HomePageViewer" is a huge part of our main page. With the help of this class, user's (registered or not) can see top movies and all categories such as action, comedy, romantic, etc. without using the search function



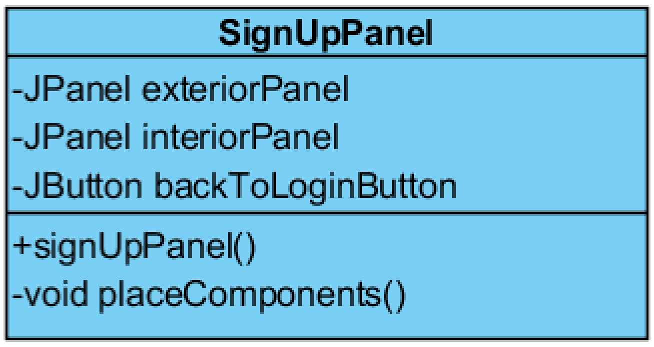
This class is used for the purpose of showing the application on the screen. Moreover, this class is used inside the class called "ProgramActionListener". According to the user's clicks, program changes the main frame or pop-ups some new frames for the user. Long story short, this class changes the frame according to the input that was coming from the "ProgramActionListener.".



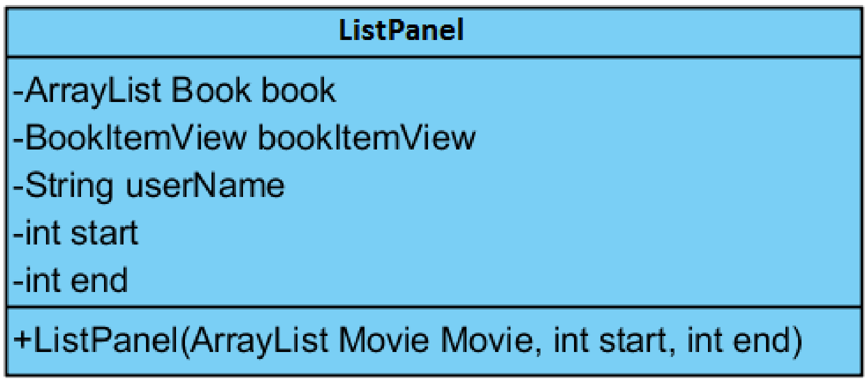
"LoginPanel" creates a panel for the users who want to login their accounts.



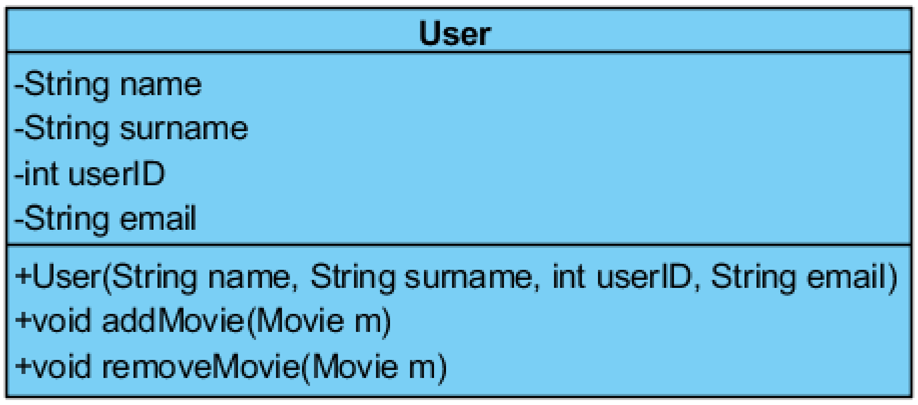
"TopListPanel" class was used inside the "HomePageViewer" class. With the help of this class, users can see the categories. After selecting the categories, movies with the same categories will show above the category buttons. Then, users can pick the movies and then see the descriptions about the movie thanks to the class called "PageViewer".



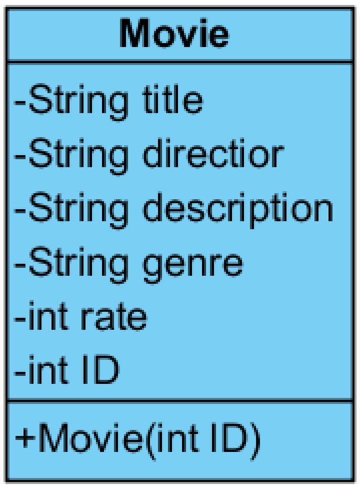
“SignUpPanel” class, creates a panel that shows the user how to sign up “MovApp” via their phones.


ListPanel
-ArrayList Book book
-BookItemView bookItemView
-String userName
-int start
-int end
+ListPanel(ArrayList Movie Movie, int start, int end)

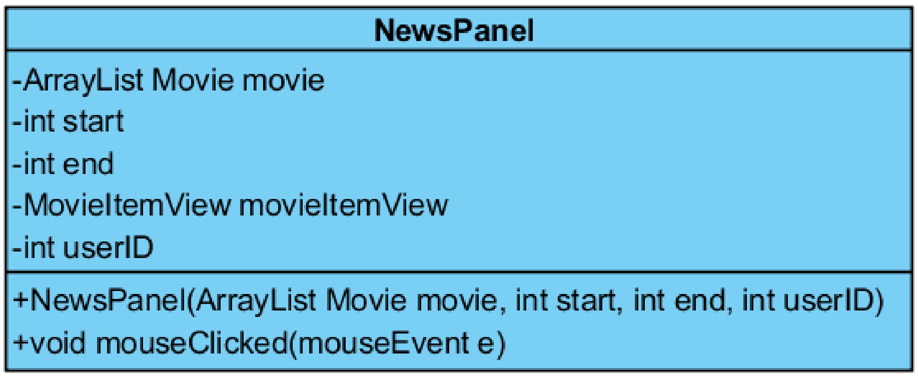
“ListPanel” is only for the registered users to access the movies which they want to watch when they travel another time.


User
-String name
-String surname
-int userID
-String email
+User(String name, String surname, int userID, String email)
+void addMovie(Movie m)
+void removeMovie(Movie m)

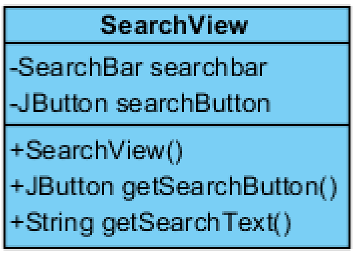
“User” class is the Model Class of the User regardless of they were registered or not(guest).



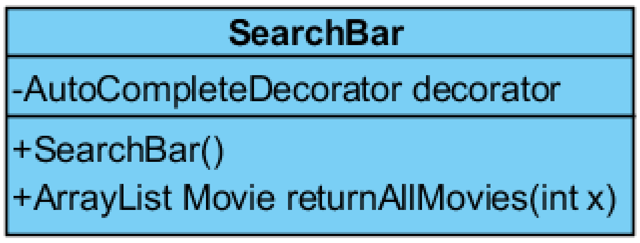
“Movie” class is the Model Class of the Movies


NewsPanel
-ArrayList Movie movie
-int start
-int end
-MovieItemView movieItemView
-int userID
+NewsPanel(ArrayList Movie movie, int start, int end, int userID)
+void mouseClicked(mouseEvent e)

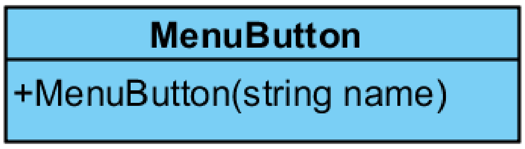
"NewsPanel" shows the users highly rated movies which were rated by the registered users. The purpose of this is the following: program recommends movies to users, so they easily access the highly rated movies.



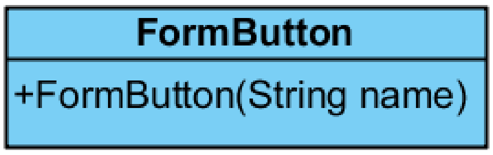
"SearchView" class uses the "SearchBar" class and combine it with the searchButton.


SearchBar
-AutoCompleteDecorator decorator
+SearchBar()
+ArrayList Movie returnAllMovies(int x)

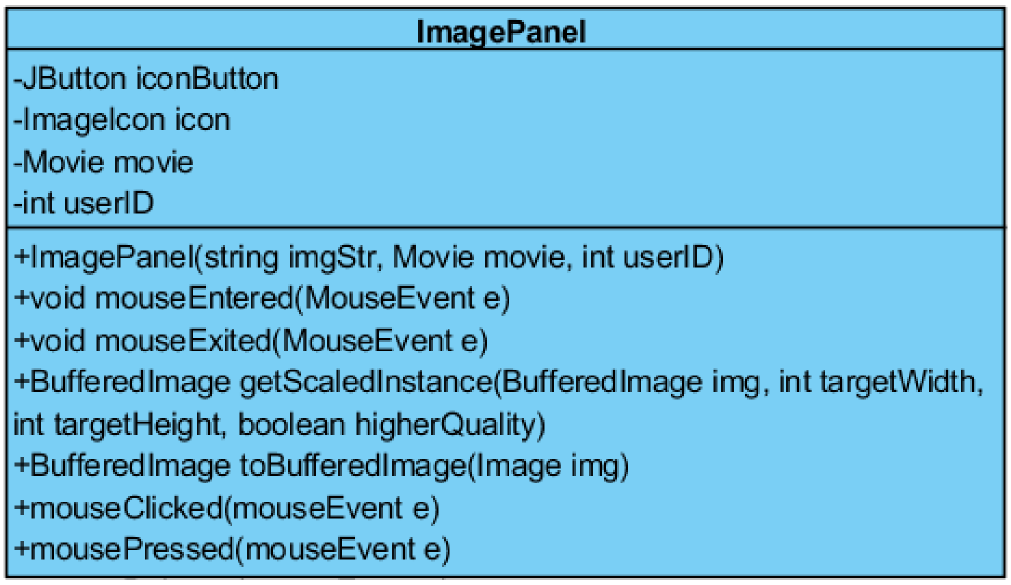
"SearchBar" helps users to search the desired movie. In order to make program user-friendly, AutoComplete search will be implemented.



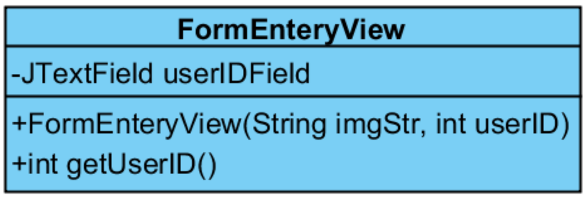
"MenuButton" class creates a button which was used inside the program. By using this class, our application maintains it is consistency situation.



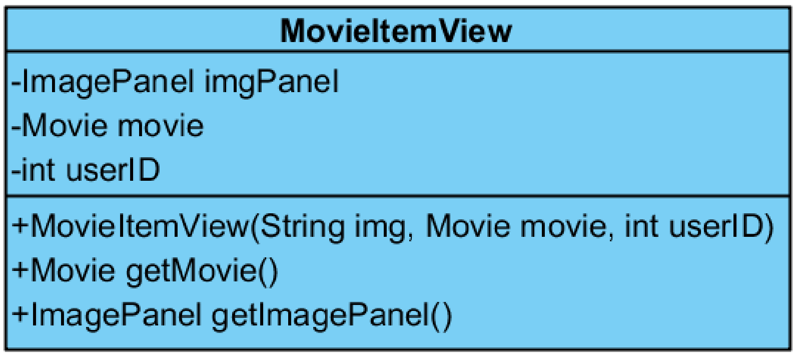
"FormButton" class creates a button which was used inside the program. By using this class, our application maintains it is consistency situation.


ImagePanel
-JButton iconButton
-ImageIcon icon
-Movie movie
-int userID
+ImagePanel(string imgStr, Movie movie, int userID)
+void mouseEntered(MouseEvent e)
+void mouseExited(MouseEvent e)
+BufferedImage getScaledInstance(BufferedImage img, int targetWidth, int targetHeight, boolean higherQuality)
+BufferedImage toBufferedImage(Image img)
+mouseClicked(mouseEvent e)
+mousePressed(mouseEvent e)

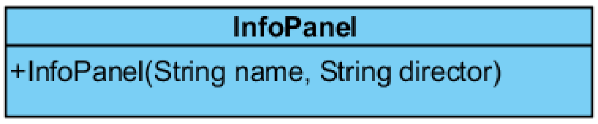
"ImagePanel" is used inside the class which is called "MovieItemView". This class' main purpose is to make the application more user-friendly. With the help of this class, the program will enlarge the image of the movie poster when the user first touches the movie.

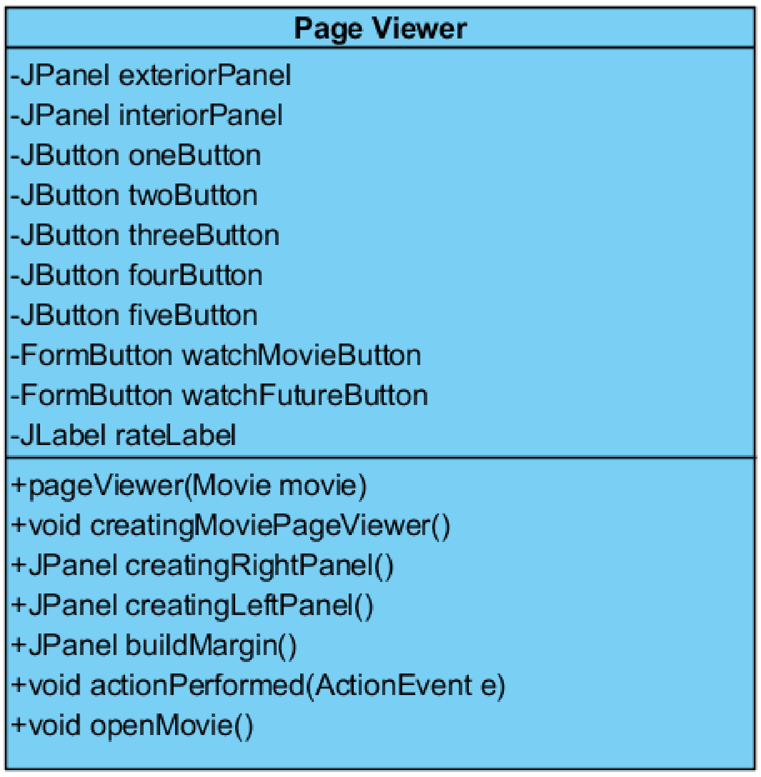


"FormEntryView" creates a login Panel that has boxes for getting the user ID from the user which wants to register to the program.


MovieItemView
-ImagePanel imgPanel
-Movie movie
-int userID
+MovieItemView(String img, Movie movie, int userID)
+Movie getMovie()
+ImagePanel getImagePanel()

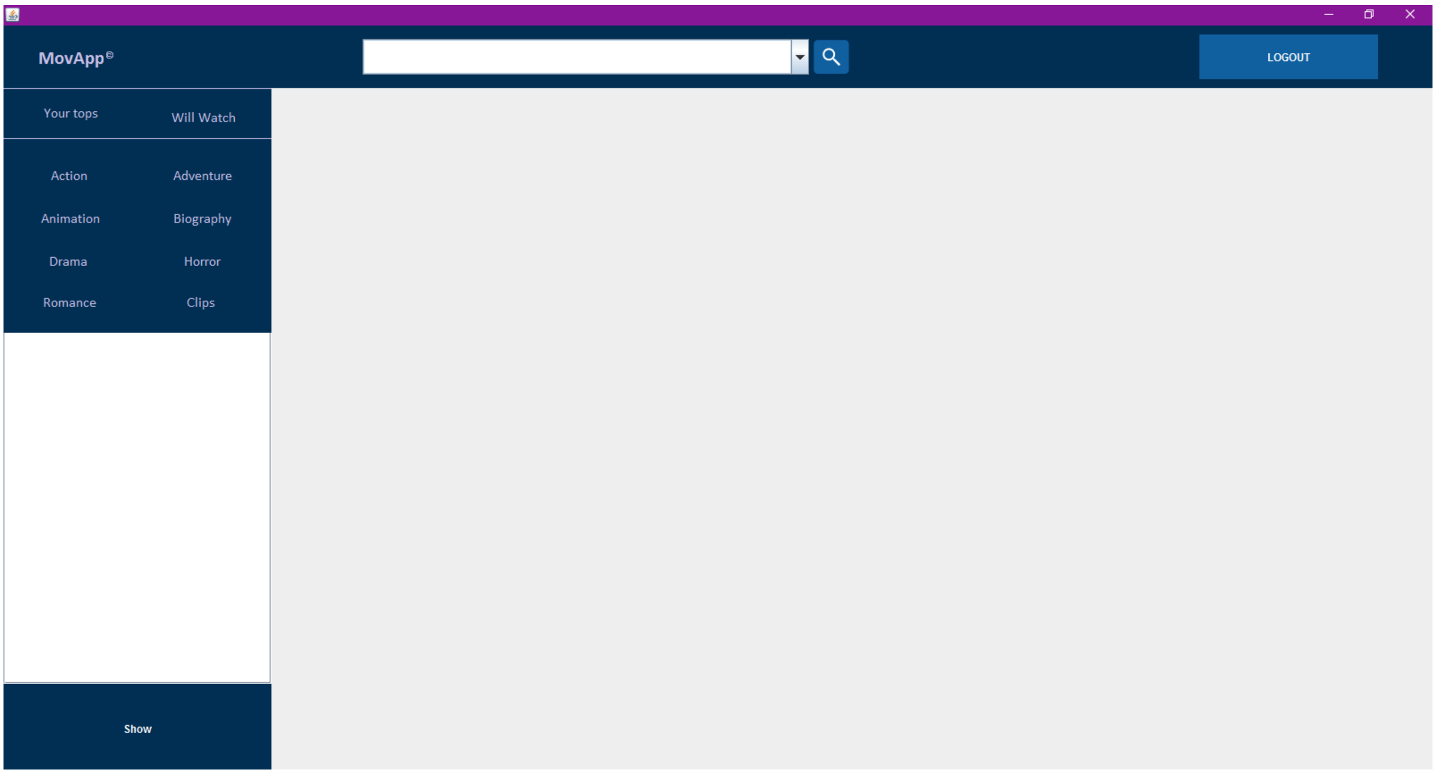
By using "MovieItemView", the program creates small panels for each movie. This class used in "NewsPanel" class for the purpose of showing the posters' of movies. Movies which was located on the home page determined by the registered users' ratings. Thanks to "ImagePanel" when the user touches the movie image for the first time, the poster enlarges. For the second touches, another pop-up window will appear that shows the information about the selected movie.



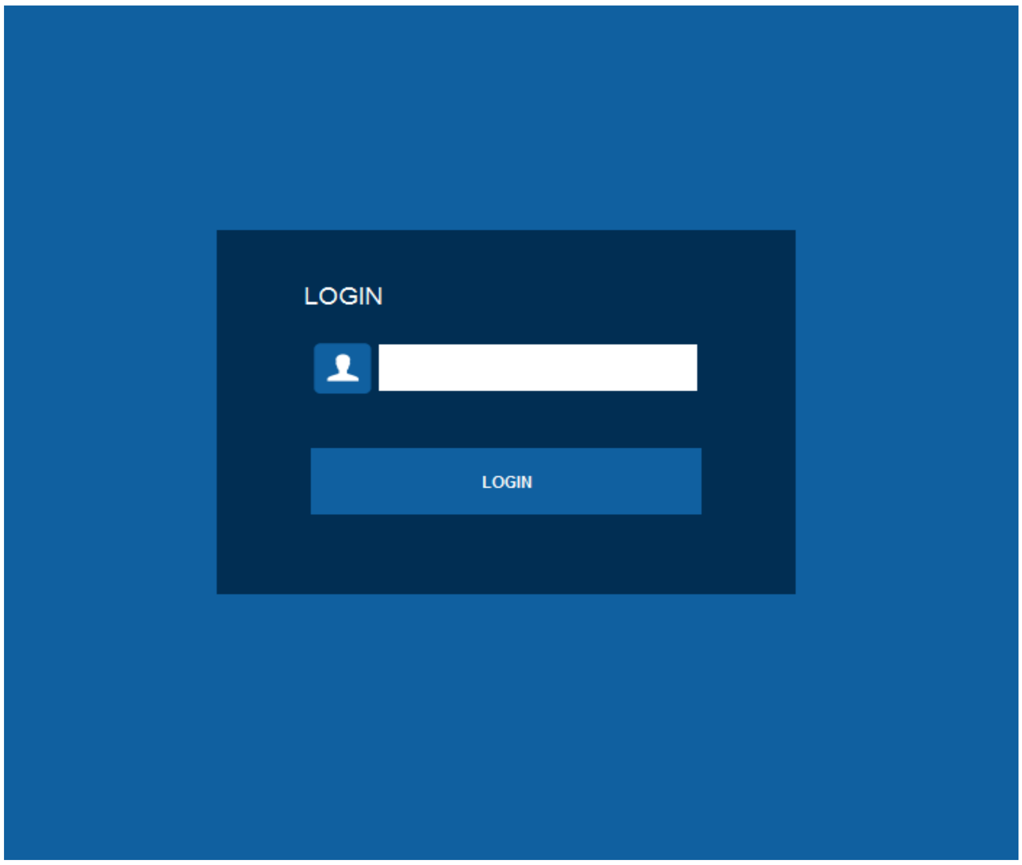


This class is activated after the user(guest or registered) picks the movie. With the help of this class, another pop-up window will appear that shows the user the poster of the film, some details about the movie such as director, cast, etc., a small description about the movie and the general idea. Furthermore, the user can see the rating of the movie: If the user is logged in, he/she can rate the movie between 1 and 5. Lastly, the user can watch the film by clicking the watch button.

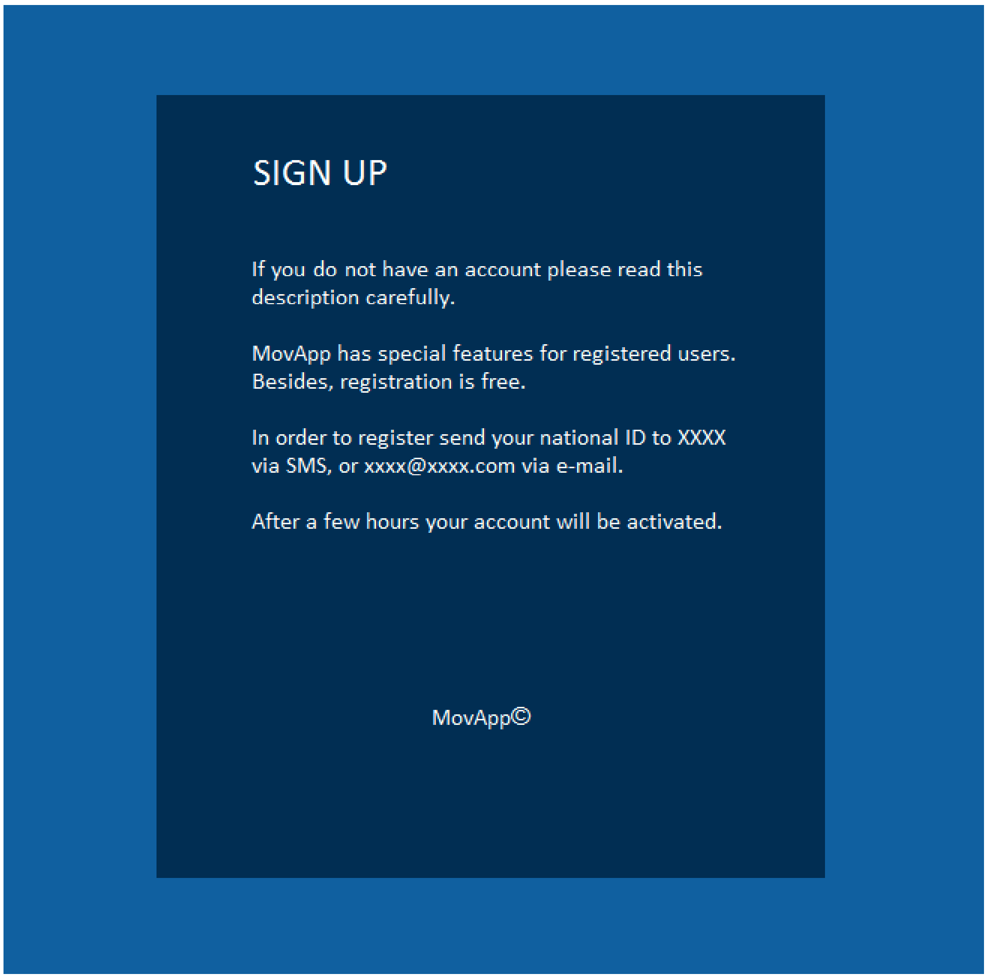
2.4.1.4. User Interface



User's Home Page



Login Page



SIGN UP

If you do not have an account please read this description carefully.

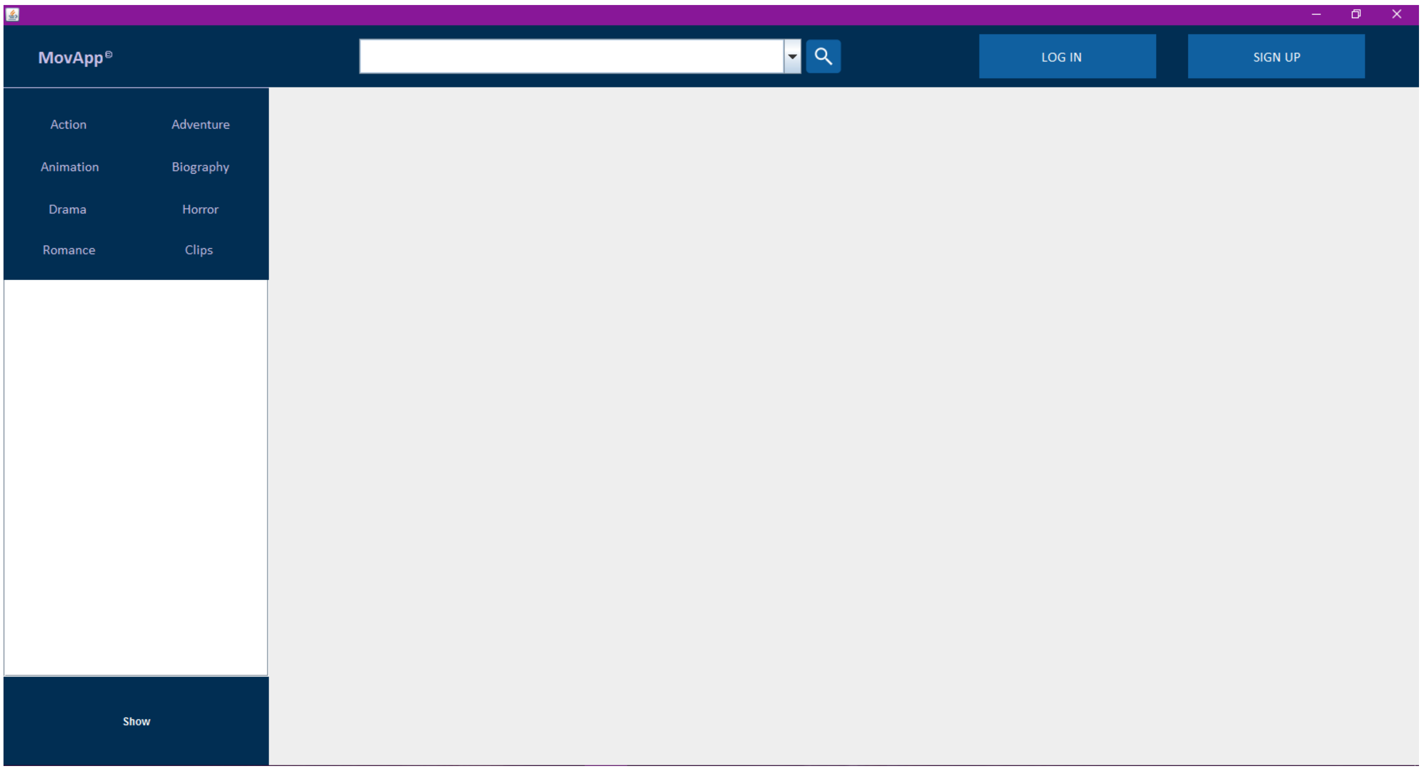
MovApp has special features for registered users. Besides, registration is free.

In order to register send your national ID to XXXX via SMS, or xxxx@xxxx.com via e-mail.

After a few hours your account will be activated.

MovApp©

Sign Up Page



guest's main page

3. Conclusion

In the analysis report, we investigate specific requirements, system models and user interface of MovApp application.

Overview includes all essential knowledge, detailed description and mechanism of application. Answers of several questions, how to login, how to sign-up, how to create future list, how to rate.

Requirements specification has three sub-sections that illustrate functional, non-functional and database-functional requirements. In both performance and implementation perspective requirements must be well thought out, balanced and clearly understood. We specify our requirements for utilise in system modelling.

In System model which is second main part of our analysis report, consist of four sections which are following:

1. Use Case Model
2. Dynamic Models
3. Class Model
4. User Interface

To decide what use cases we have, we spent some time on thinking about them to identify which use cases we should indicate for our project. At this point, we tried to specify which requirements, which actions are actual use cases. Our Dynamic model includes sequence diagrams, state diagrams and activity diagrams. We tried to show possible actions that will constitute the crucial parts of our application and interactions between the user and the System by sequence diagrams. In our class diagram, we tried to form a perfect class model since we know that in implementation and design process this model would constitute a crucial part. We spent so much time about thinking possible classes and connections between them.

For the user interface section, we prepared mock-ups and navigation path to show how MovApp will look like. We created a navigational path with using our use cases. Putting effort to take considerations of users' needs so that application will be much closer to be user-friendly. Screens are easy to understand, as far as possible they are minimalist.

To sum up, we tried to create a detailed wider scope for analysis report since it will guide us in our design and implementation process.