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

1.1. Purpose of the System

MovApp is an application for passengers, who travel via buses or planes. The main goal of this application is, users, view and watch different categories of movies. The first interface of the system is a guest user menu, which has a limited action such as selecting and then watching a movie from category part, seeing and watching the top-rated movies by other users. However, if the guest users logged in to the application, they can rate the movies, can see their own top-rated movies according to their previous ratings and last but not least add movies to their watch later lists. Therefore, our main aim is to design an application that assists the user to choose, watch and rate the movies.

1.2. Design Goals

1.2.1. Adaptability

Java is one of the programming languages which provide cross-platform portability. Java has an easy way to connect with database systems. Because of these reasons, MovApp was developed by Java, Version 8 Update 111 (build 1.8.0_111-b14).

1.2.2. Efficiency

The system must response to the orders with high performance for the purpose of providing smoothness while choosing the movie and watching the desired movie. Moreover, users can search movies efficiently to find the desired movie. In other words, being fast and smooth are one of the important design goals. Thanks to “ProjectActionListener” class, almost all objects in the program created together in the same time. After that, all these classes were controlled in the same class during the whole working time off the application. Therefore, application works efficiently.

1.2.3. Usability

Another important and main design goal in the program is the user-friendly interface and comprehensible graphical contents because MovApp is developed in order to provide a tidy and helpful view for the users. Therefore, the operability of the program is designed attentively.

1.2.4. Reliability

When an unwanted behaviour occurs, such as user is entering a wrong ID while logging up to the system or leaving the ID field blank, program will not crash and handle this exceptional behaviour. Furthermore, while user is signing up to the system, program will not crash if user leaves one of the fields blank. Also, ID's of the users must be their identity number which was given by Turkish government. So, the program will check if the ID is valid or not. There will be different solutions for different exceptions and none of them will result in crashes or data loses.

During the project process, after completing every subsection, the program was tested to detect any crash situation or unexpected outputs. Boundary condition inputs were tried to execute, and the problems were fixed if there are any.

1.2.5. Security

There is no security check for Guest User. Guest users can directly connect to the system. Registered user menu required a security check. Every registered user should have an ID which was given by the Turkish government. All the IDs will be stored in the database of the program. If a user enters an ID, validation will be done with the help of the programs database. If the ID is valid, user can see the registered user home page.

1.2.6. Extensibility

Thanks to Object Oriented Design of the project new parts can easily insert into the program. To exemplify, in order to create new movie category, all the program needs the movies that was related with that category. Afterwards, very few line of code implementation is sufficient.

Moreover, the program may be extended later on, allowing registered users to make comments on movies and also see other registered users' commands. Also, a simple search history function may be added, which shows The last searches of the user. Lastly, movies may continue from the point where user was stopped previously.

1.3. Trade-offs

1.3.1. Efficiency and Reusability

Reusability can be the applications concern, any data which have different subcategories and common features can be viewed and searched via our program. For instance, this application can be converted into a program that contains books, games or music rather than movies.

1.3.2. Functionality and Usability

Since the main purpose of the program is entertainment and informing the user about movies, easy usage of the program was the main focus. The functionality of the program is basic in order to make a user-friendly and a comprehensible application. The interface of the program has useful directions on it with the help of the labels and universal images. With the help of this, the program can be used without having any previous knowledge.

2. SOFTWARE ARCHITECTURE

2.1. Subsystem Decomposition

3-tier Architecture is one of the most popular design. MovApp's system is also in 3-tier architecture with the data, logic and presentation tiers.

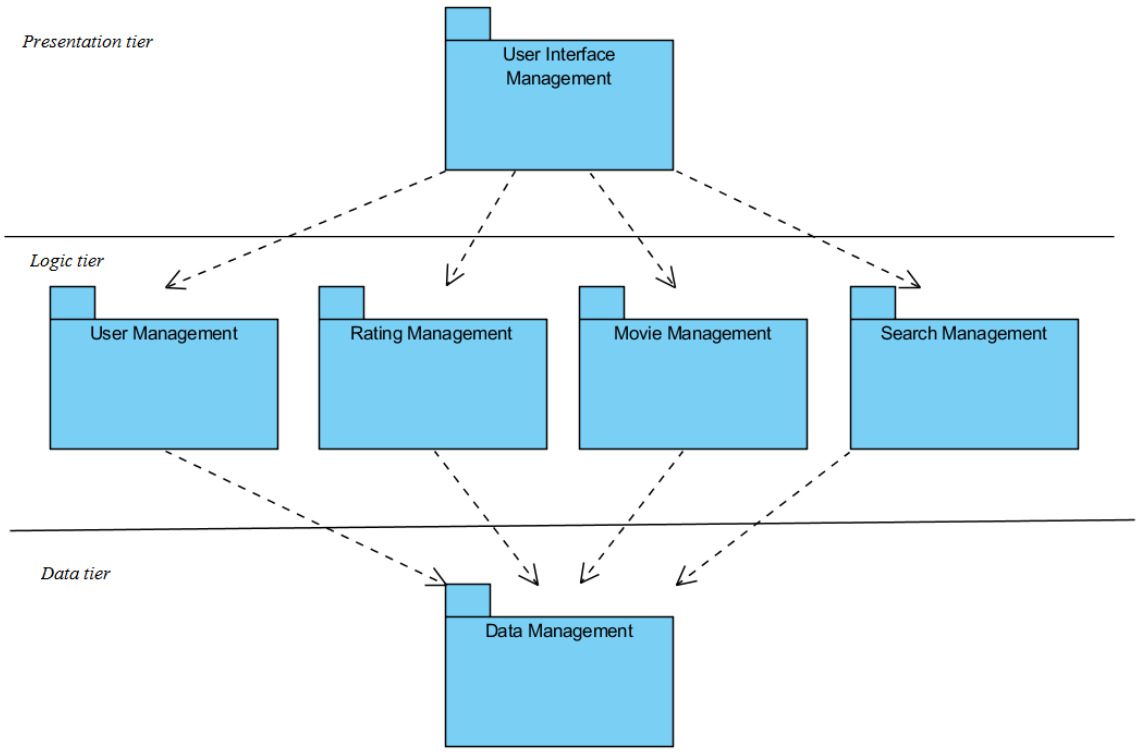

Figure 1 - Subsystem Decomposition

Presentation Tier: The most top level of the program is the user interface. The purpose of the user interface is to translate abstract tasks and result in a visual display. Therefore, the user can understands and operates according to his or her needs. User Interface Layer is the only member of this tier. This layer is the monitor where users can operate.

Logic Tier: This layout is the core of the program. It coordinates the application; process commands, make logical decisions and evaluations. Also, calculations are performed by this layout. Logic Tier is a kind of bridge in between Presentation and Data tier. It moves and processes data between the two surrounding layers. This tier is composed of four layers that are User Management Layer, Rating Management Layer, Movie Management Layer and Search Management Layer.

Data Tier: This layout is the memory of the program. Data tier stores movies, movie's descriptions, movie's rates, movie's casts, user's lists and so on. Also, it retrieves information from the database to the Logic Tier. Data Tier includes Data Management Layer which has all classes about the database.

2.2. Hardware/Software Mapping

MovApp has been developed in Java on NetBeans integrated development environment 8.2. Also, our database has been constructed by MySQL Work Bench 6.3. Since Java has an own virtual machine, switching among different platforms is not a big problem for MovApp. In fact, we are developing it for desktop now, but the actual usage of MovApp will be on the touchable screen platforms. That is why the application have screen keyboard on some pages.

For now, the application's hardware requirement is mouse connected to the computer. While selecting movies, movie categories, or any button (login, signup, logout, etc.), a mouse will be used for presentation purposes. After MovApp is implemented on the touchable-screen platforms, application will not need a mouse.

Since the database is local, application do not need an internet connection.

2.3. Persistent Data Management

Most media items (like movies) occupy a significant amount of memory. MovApp is not only a media player, but it also stores movies and related items (like icons, descriptions, movie rates, etc). We create our data tier with MySQL which is secure, compatible, and easy to use.

MovApp access the database when following scenarios happen.

- When the program started first, default movie icons, titles, and some other movie information is retrieved from the database.
- While using search field, the program also uses the database in order to compare entered string with titles of movies.
- While selecting any category or list, the database is also used to retrieve related movies.
- After selecting any movie, description will appear. Also, when the user clicks watch button, the media player will be displayed. These events work with the database too.
- When a guest wants to log in, entered ID will be compared with stored IDs from the database.
- Also, movie rates are stored in our database.

2.4. Access Control and Security

There two types of user correspondingly there two different access ways. There is no security check system for being guest users, but they have limited access while using the program. Guests can access the sub-categories and top movie list but they cannot create a list for themselves or they can see the rate of movies but cannot rate the movies. The second user type is registered user, which is needed to be login the system thus there is a security check for users. Obviously, these users have more extended access area while experiencing the program. They can access sub-categories, their individual lists, top list and can rate the movies. If any registered user rate any film rate will change automatically, every user can access the rate of movies, but they cannot access the information that who did rate the movies.

2.5. Boundary Conditions

2.5.1. Initialisation

- What data need to be accessed at startup time?

Start-up time allowed the user to see sub-categories, top lists and search button for movies. Also at the start time, all movies can be watched.

- What services have to be registered?

To reach registered user page, the system requires a login from users. Registered user page additionally allows the user to create UserTopList and WillWatchList and rate the movies.

- What does the user interface at start up time?

Start-up time begins with guest user menu which has restricted usage of the system, also login and signup page buttons exist on it.

2.5.2. Termination

- Are single subsystems allowed to terminate?

While the movies are displayed on the page, the display screen can be terminated without terminating all system.

Closing any sub-list does not cause termination of the whole system.

- How are updates communicated to the database?

When users rate any movie, this movie's mean rate (which composed of different users who give a rate for this movies) should be updated. We have used MySQL for a store our database of the program. First, the code reaches old rate and number of user who rated the film before and then takes the new rate of the user. Finally, calculate the new rate of the movie and update it.

2.5.3. Error

- How does the system behave when a node or communication link fails?

When guest users want to rate any movies system will warn the users their boundaries, and want them to register to the system.

If the user enters the wrong ID while registering the system, the program will warn the user.

2.6. Design Patterns

In the software development process, the system is called the Design Pattern, which is the result of reusing the codes, solving the emerging software problems and coding these solutions as standards and reusable. Design templates are the standard and the most appropriate solution that is developed by considering the solutions that software developers have produced over time in response to the problems they encounter. These methods can be used by all software developers. As we mentioned before, our design is in 3-tier architecture which is an appropriate pattern for a long time. In addition, we have used some other patterns too. These are 'search algorithm' and 'screen keyboard'. We have added 'search feature' to find a specific movie directly. We will need screen keyboard when the application starts to be used on a mobile platform. Both of these are being used most mobile applications.

3. SUBSYSTEM SERVICES

3.1. Detailed Object Design

The diagram that was shown below, is the UML Class Diagram of the MovApp application.

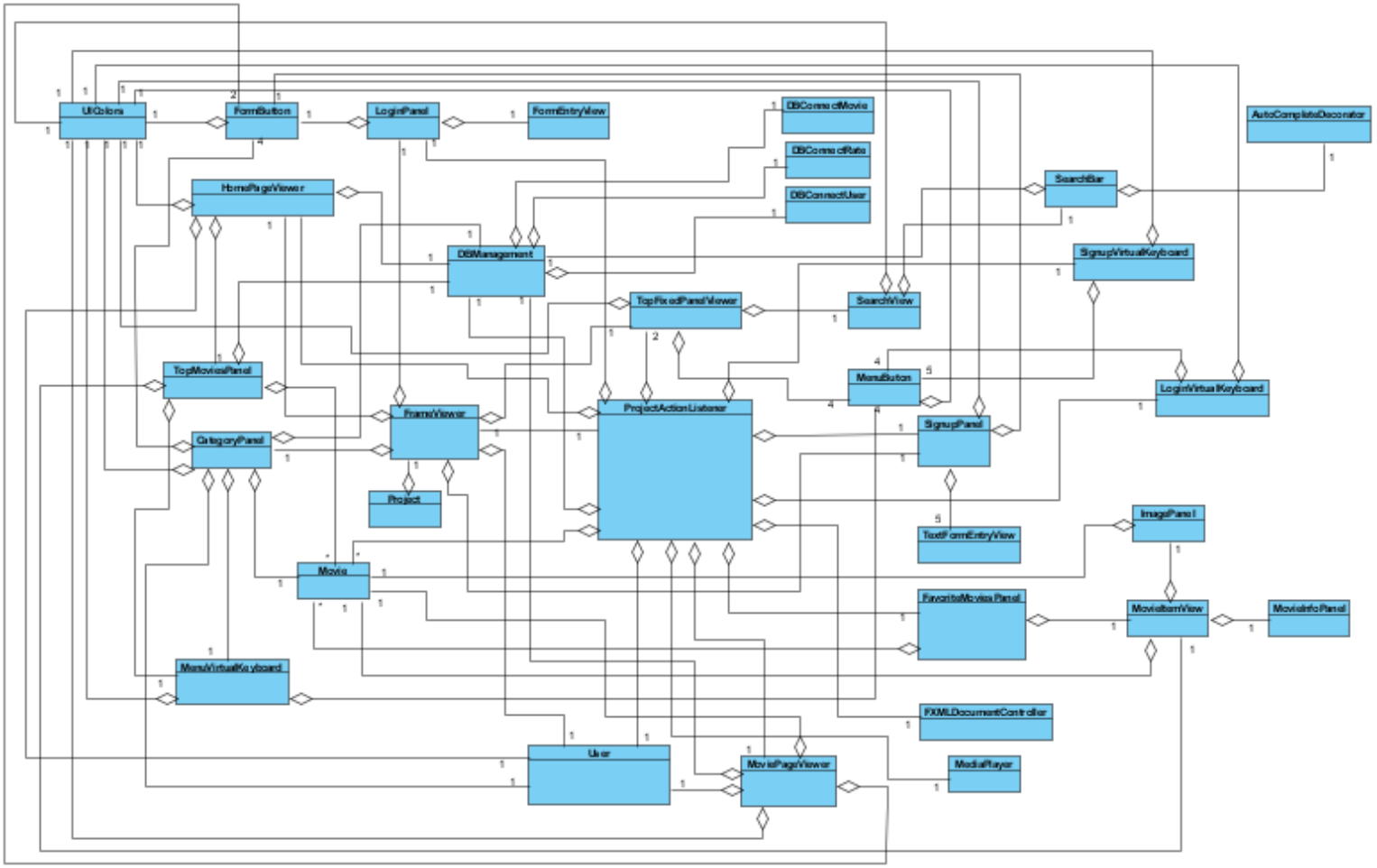

Figure 2 - UML Class Diagram of the MovApp application

For the purpose of reducing the confusion and make the class diagrams understandable, all classes that interact with other classes will be examined carefully.

The figure that was shown below is the UML Class diagram of the classes that are directly related to the controller class called “ProjectActionListener”.

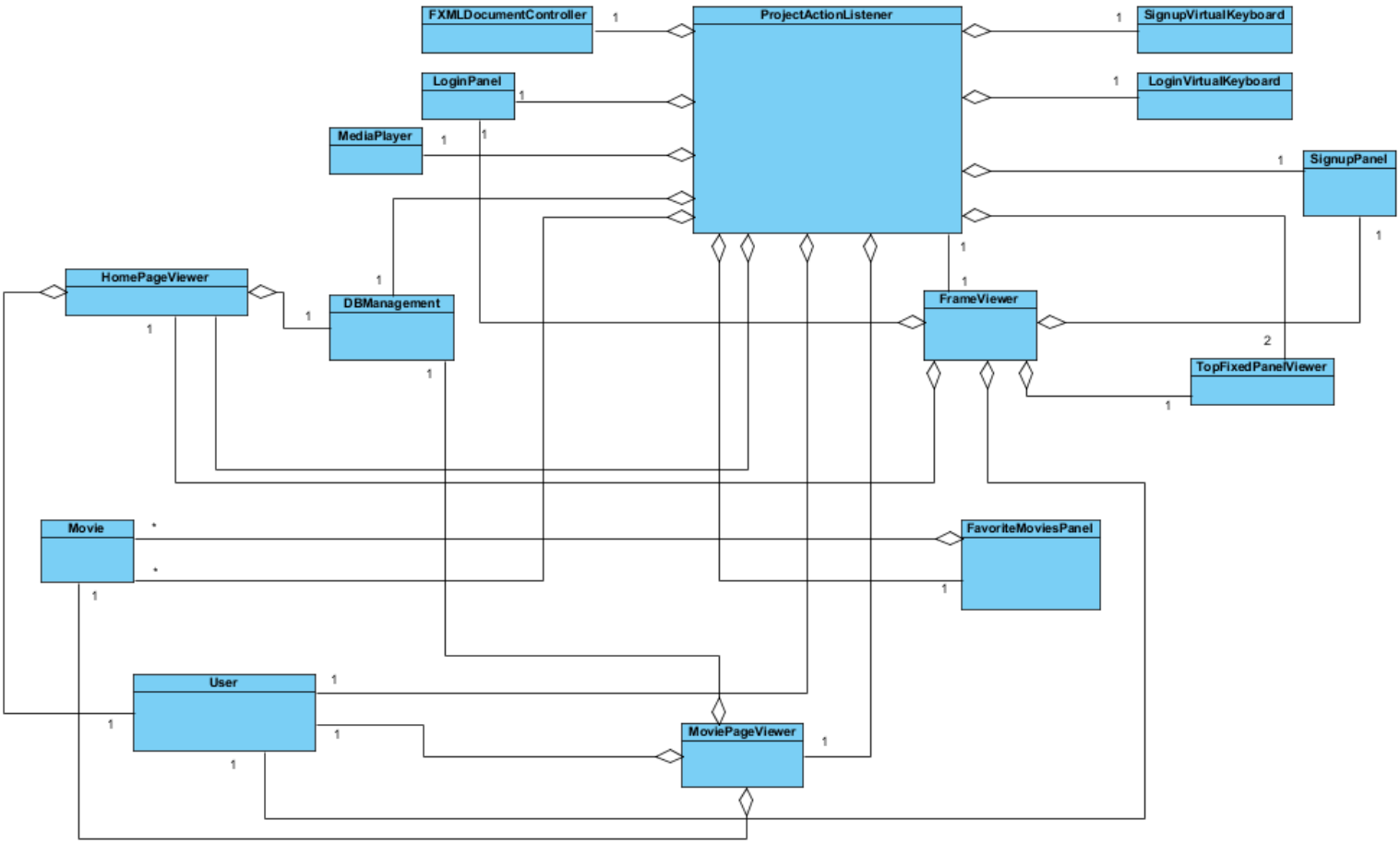

Figure 3 - Class Diagram

“ProjectActionListener” is the only controller class inside the application. This class interacts with the main user interface classes to change the user interface according to users actions.

The figure that was shown below is the Class diagram of the “FrameViewer” class.

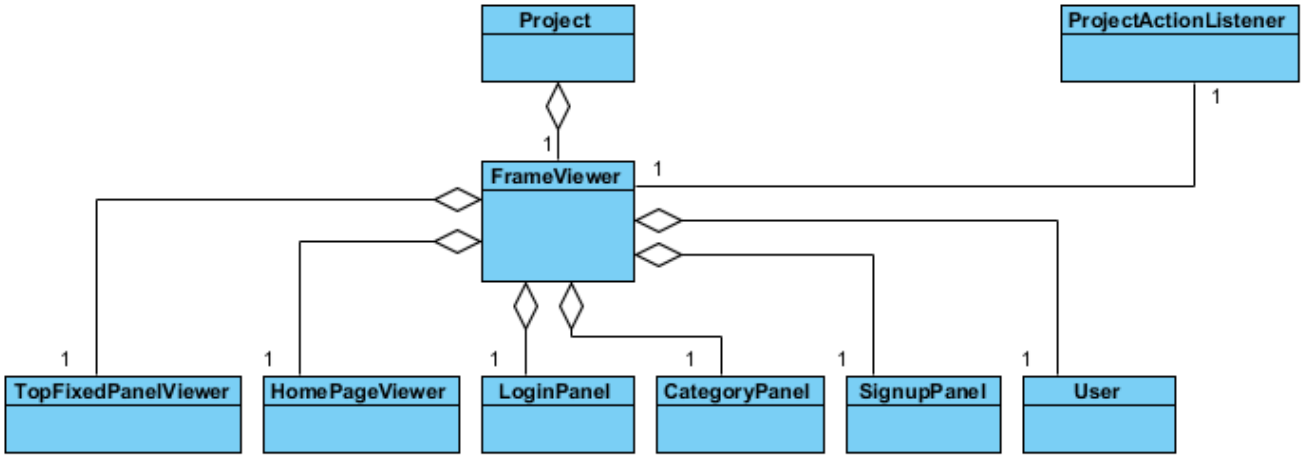

Figure 4 - Class Diagram

“FrameViewer” class is the starter class of the application. This class is directly related to the project class, which has the main method, called “Project”. After the application starts, “FrameViewer” class, creates the view classes that will be used during the application. After that, “ProjectActionListener” class called in order to handle user actions that are received from view classes.

The figure that was shown below is all the classes related to the “FrameViewer” class.

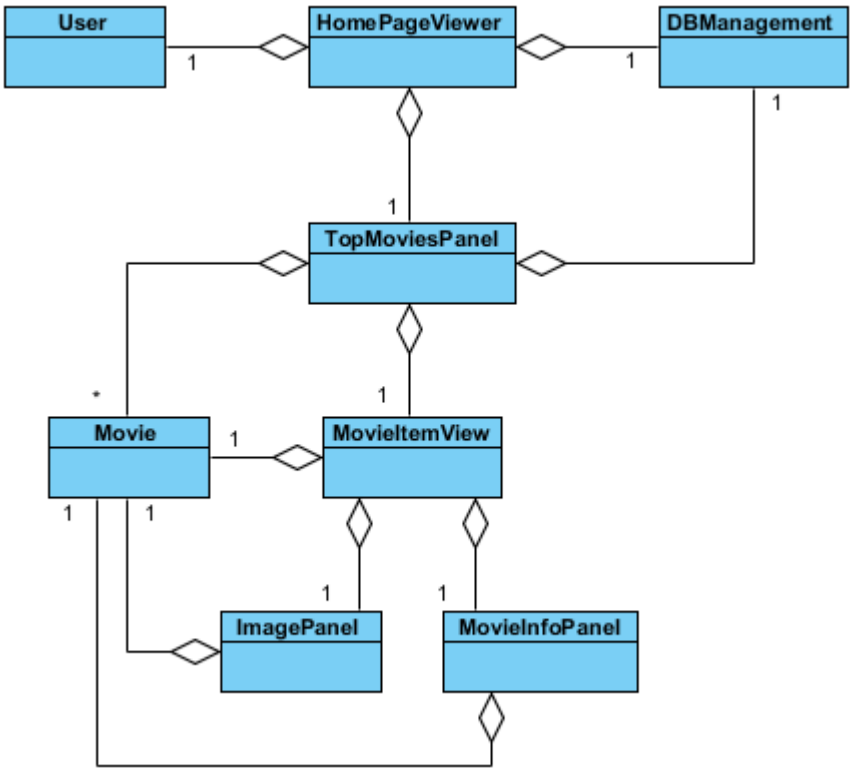

Figure 5 - Class Diagram

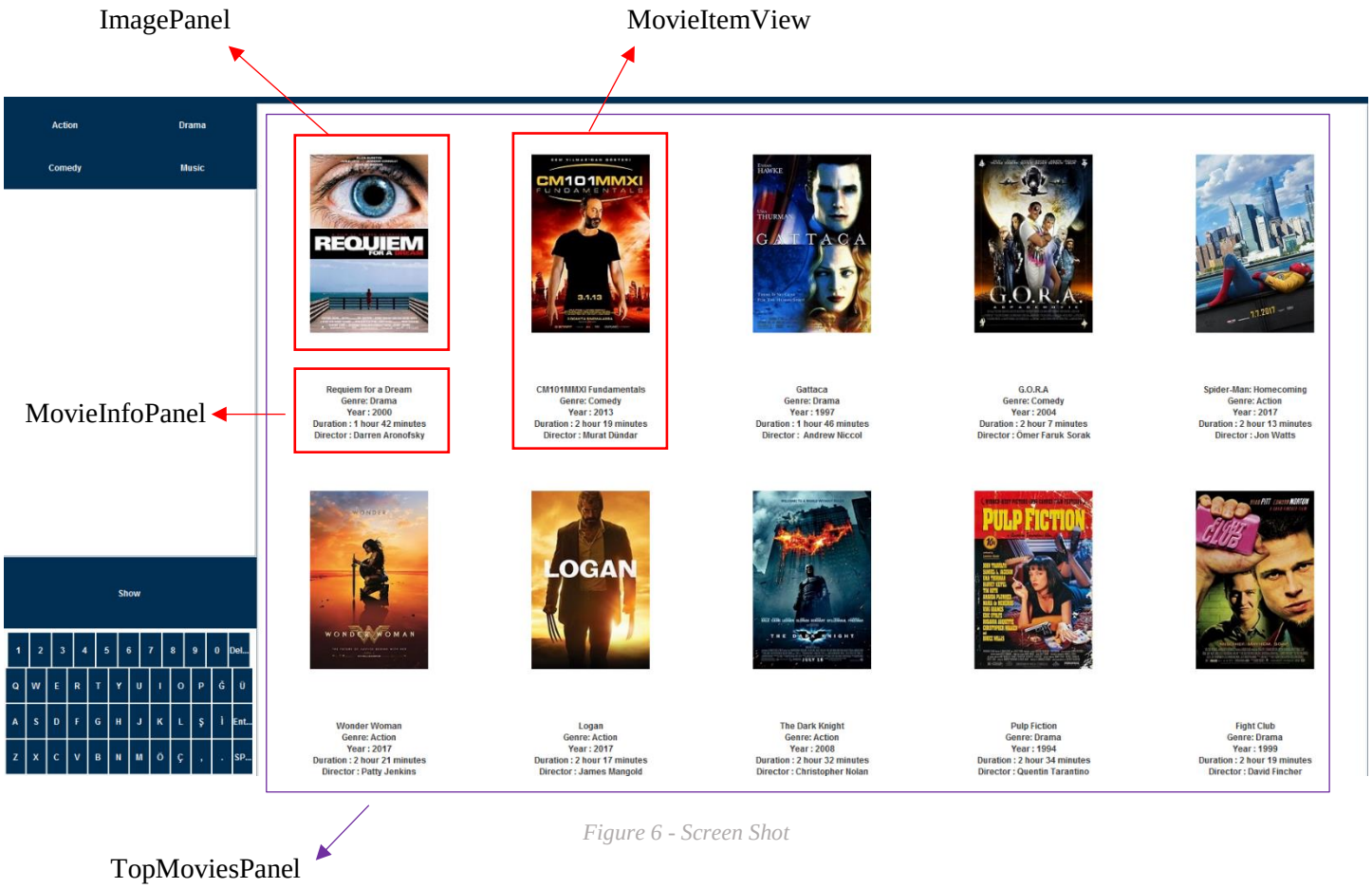

Figure 6 - Screen Shot

"HomePageViewer" class is one of the main pages for both registered and guest users. As shown in page 11, every class inside the "HomePageViewer" class, creates a small part. By combining these small parts, the application first generates a class called "TopMoviesPanel". After that, merging a scroll panel and "TopMoviesPanel", the program generates the class called "HomePageViewer".

The figure that was shown below is all the classes that were directly connected with the "CategoryPanel" class.

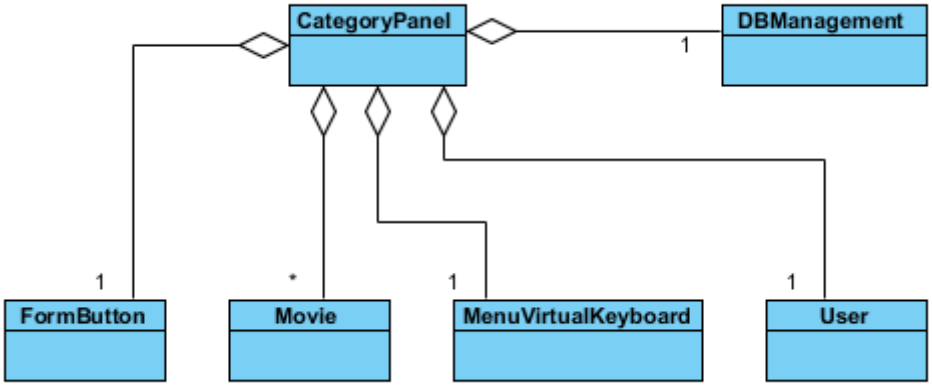

Figure 7 - Class Diagram

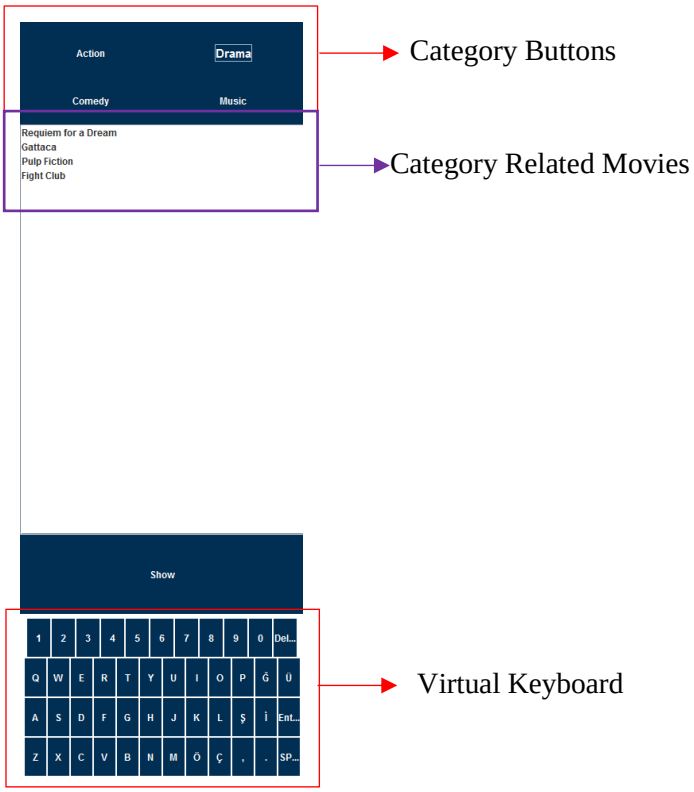

Figure 8 - Screen Shot

As can be seen on page 9, category panel is directly connected with "ProjectActionListener" class. With the help of this class, users can easily pick the moives according the categories. Moreover, user's personal watch later lists will be shown in "CategoryPanel" as well.

Moreover, Virtual Keyboard for searching movies is located on category panel for simplicity and user-friendly user interface purposes.

The figure that was shown below is all the classes that was strictly related with the class called "TopFixedPanelViewer".

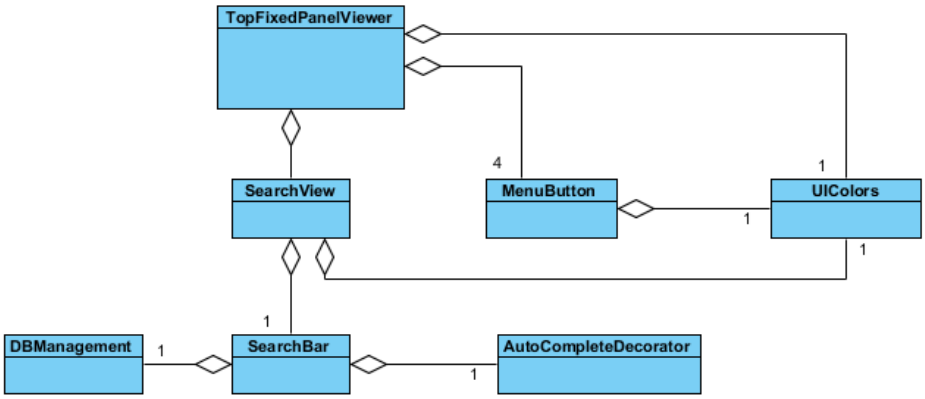

Figure 9 - Class Diagram

Guest's "TopFixedPanelViewer":

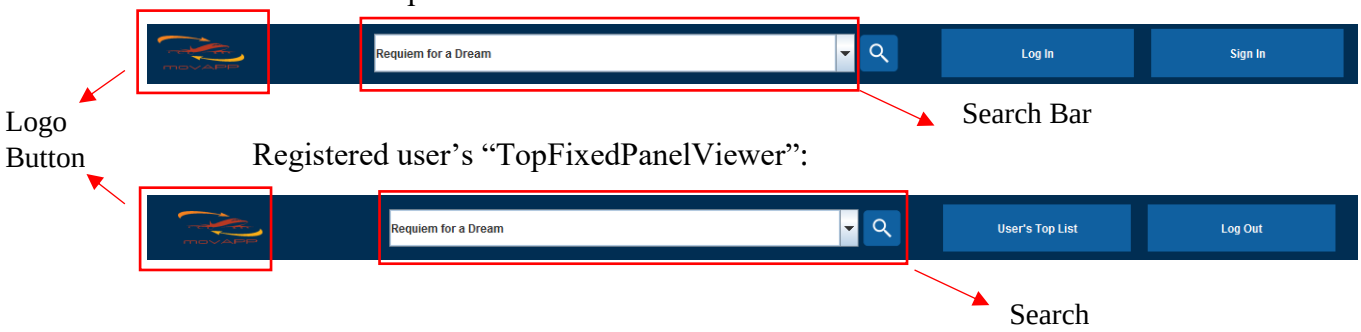

Figure 10 - Screen Shot

As can be seen from page 9, "TopFixedPanelViewer" class are directly connected with the "ProjectActionListener" class.

Logo Button helps to user to navigate directly to the home page from every page.

The figure that was shown below is all the classes that was strictly related with the class called “LoginPanel”.

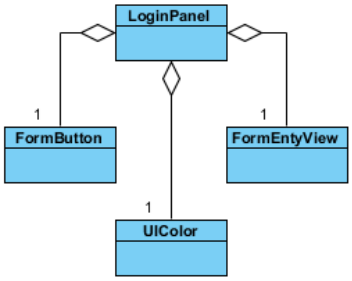

Figure 11 - Class Diagram

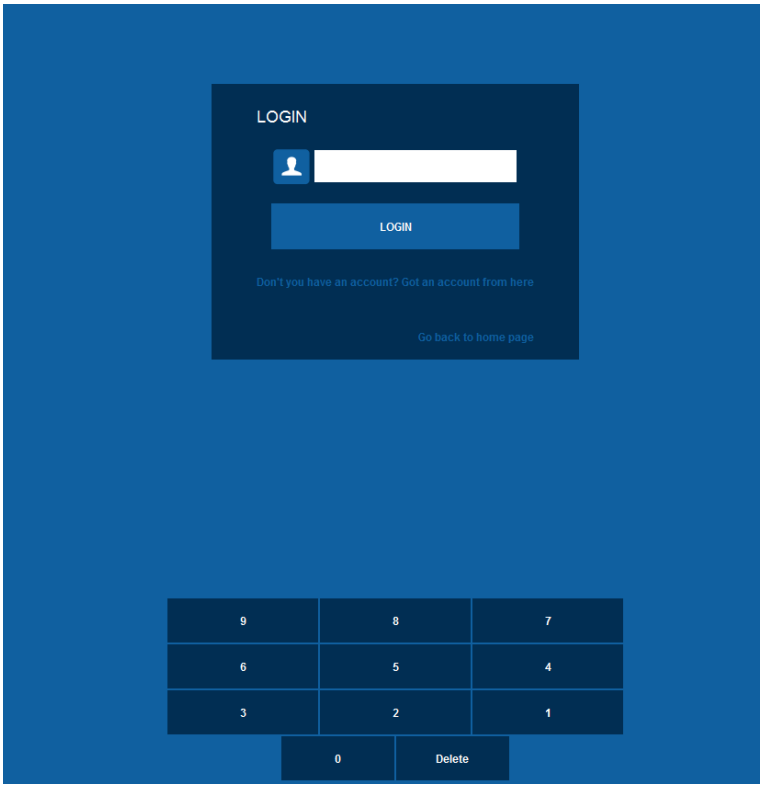

Figure 12 - Screen Shot

“LoginPanelClass” is directly connected with the “ProgramActionListener” class. Moreover, “LoginVirtualKeyboard” class is not added directly inside the “LoginPanel” class. “LoginVirtualKeyboard” class is added as another panel under the “LoginPanel” class inside the “ProjectActionListener” class.

The figure that was shown below is all the classes that was strictly related with the class called “MoviePageViewer”.

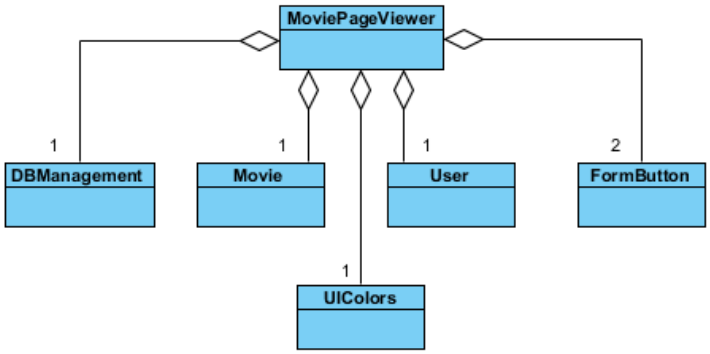

Figure 13 - Class Diagram

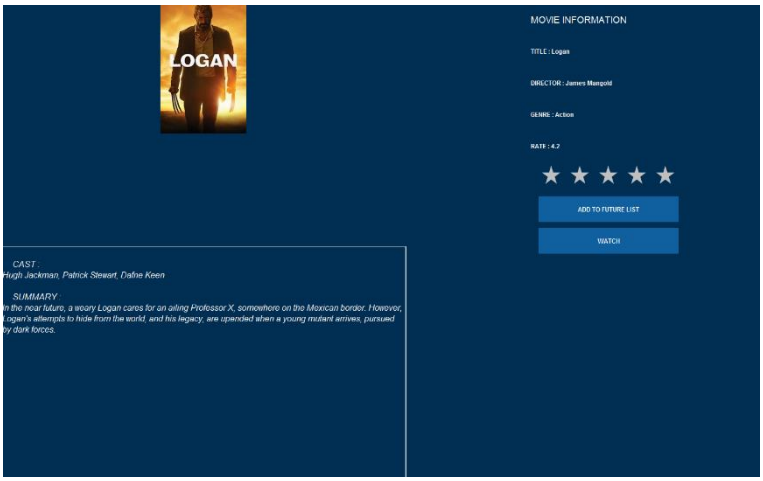

Figure 14 - Screen Shot

3.2. User Interface Management Subsystem

3.2.1. Guest Home Page / Logged in Home Page

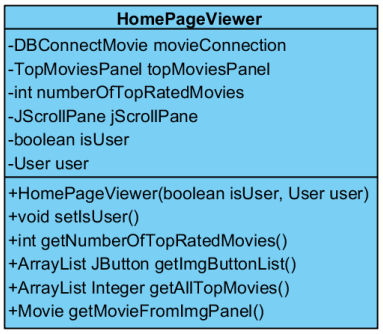

Figure 15 - HomePageViewer Class

Common for Guest and Logged in User:

The homepage is the first screen that will be instantiated when the MovApp is first executed. It displays categories, show button for choosing a movie from categories, top movies, virtual-touchable keyboard, movies and the top fixed panel where the logo, search bar, and other user related buttons such as log in button, sign up button, user's top movies panel and logout button. Categories will be shown at the left middle corner and users can click the desired category, and it shows movies in the just below the category panel.

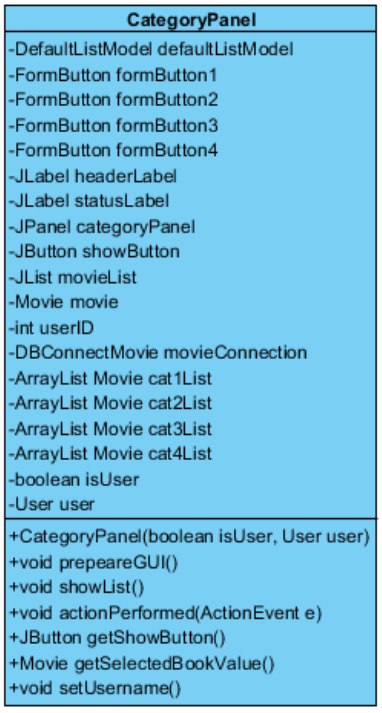

Figure 16 - CategoryPanel Class

Show button will be used when the user chooses the category in order to demonstrate sub section movies of that category.

Virtual - touchable keyboard is the one of the essential requirement of the screen. It will be shown just below on the show button. In order to search movies, the user needs to access keyboard on the screen.

Top movies panel shows the well-liked twenty-five movies inside the home page according to their total rates which were given by the logged in users. Top movies panel is just under the top fixed panel.

The search bar will be shown at the top of the screen with the search box in order to find desired movies.

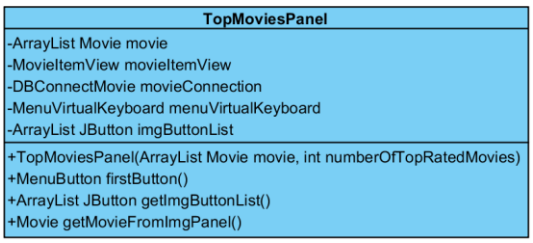

Figure 17 - TopMoviesPanel Class

On the top of the left part of the Top Fixed Panel, the application has its own logo. Thanks to this logo, users can directly go back to the home page whenever they click this logo.

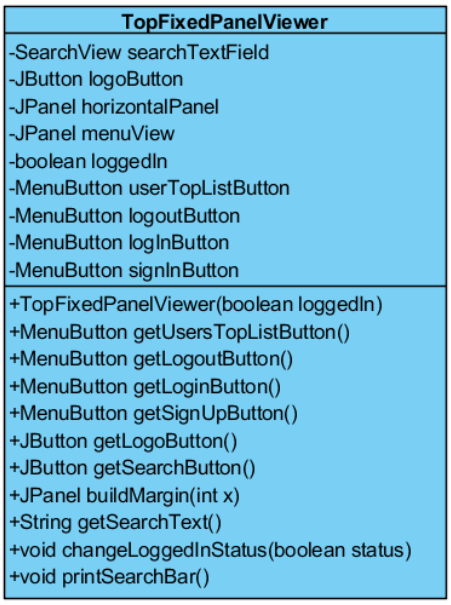

Figure 18 - TopFixedPanelViewer Class

Guest User:

Login button is placed at the top right corner of the HomePageViewer. With the help of this button, guest users can easily log in to the application called MovApp. This button clears the whole panel and creates a new login panel.

Signup button helps to users who do not have an account to sign up via webpage, SMS or directly inside the application.

Logged in User:

After user logged in to the application, the user will see his or her personal top movies button, will watch list button and logout button rather that login and signup button.

Personal Top Movies shows the user's highly rated top 10 movies. Moreover, will watch list button shows the user's watch list.

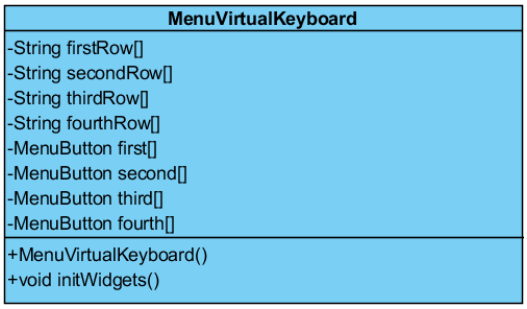

Figure 19 - MenuVirtualKeyboard Class

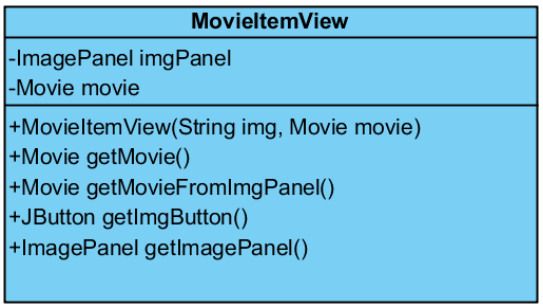

Figure 20 - MovieItemView Class

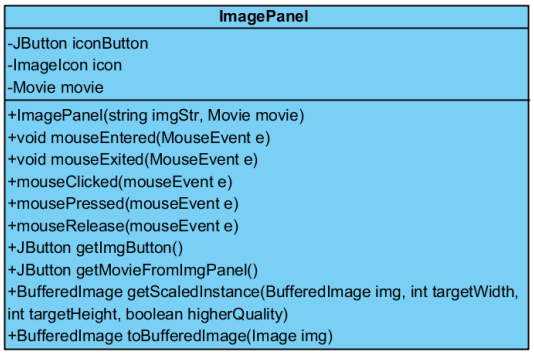

Figure 21 - ImagePanel Class

3.2.2. Search Results

SearchBar:

Search Results is initiated when the results should be displayed. With the help of AutoCompleteDecorator class, this application auto completes according to desired text entry.

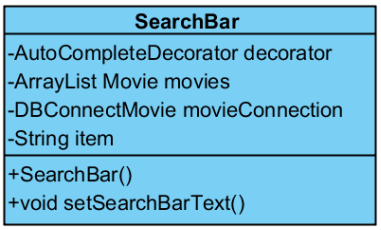

Figure 22 - SearchBar Class

SearchView:

SearchView includes the SearchBar class and some user-friendly images.

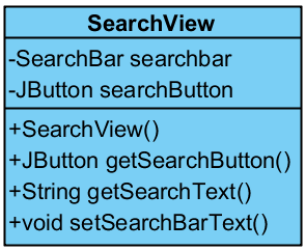

Figure 23 - SearchView Class

3.2.3. Login Screen

LoginPanel:

"LoginScreen" is initiated when the user is directed from "MainMenu" to login or when the user is directed from "SignUpScreen". In order to login or sign up, the virtual-touchable keyboard will be accessible in these screens. In this class, there are two text-fields for username and password, and there are two buttons: login button and a button that links the user to "SignUpScreen" view.

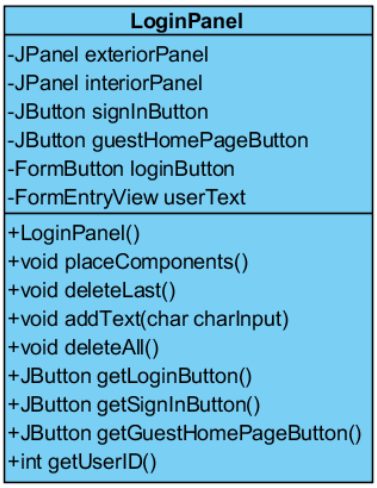

Figure 24 - LoginPanel Class

FormEntryView:

When the user wants to login, “FormEntryView” will be initiated and the program will ask the user to give login information which is ID, the must because every transportation in Turkey, when transporters buy a ticket, they have to give their ID information to the sellers. Hence, ID is a core requirement for transportation. As a matter of fact, in the MovApp, the obligation for login is ID.

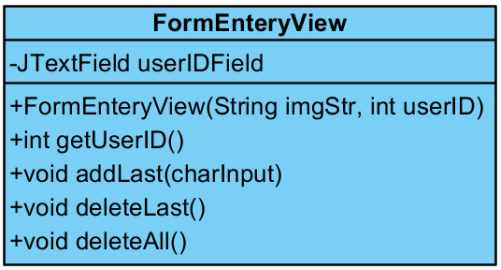

Figure 25 - FormEntryView Class

LoginVirtualKeyboard:

LoginVirtualKeyboard is a virtual keyboard that helps the guest user to login the movAPP application. Because of the idea that this application will work on touchable screens, movAPP needs a virtual keyboard for users who want to log in to the program.

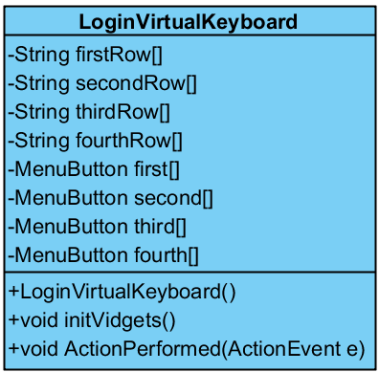

Figure 26 - LoginVirtualKeyboard Class

3.2.4. Sign Up Screen

SignupVirtualKeyboard:

SignupVirtualKeyboard is a virtual keyboard. Thanks to this, guest users can sign up to the movAPP application. Because of the idea that this application will work on touchable screens, movAPP needs a virtual keyboard for users who want to sign up to the program.

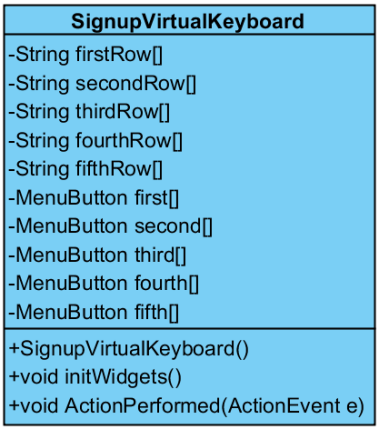

Figure 27 - SignupVirtualKeyboard Class

SignUpPanel:

The SignupPanel screen is initiated when the user is linked from home page screen.

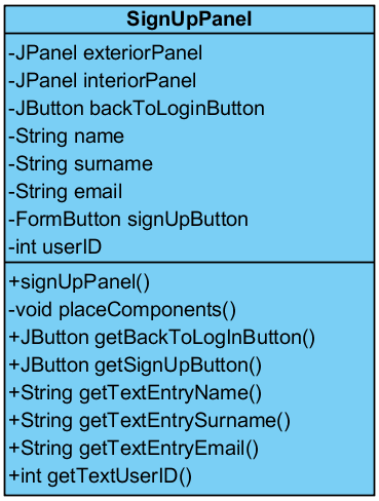

Figure 28 - SignupPanel Class

3.2.5. Movie Screen

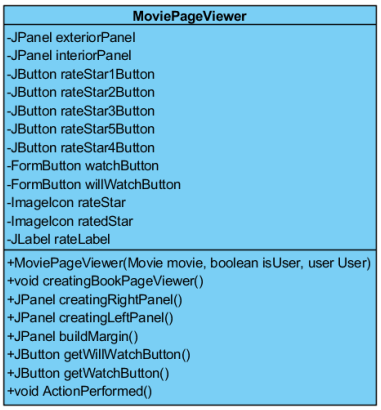

Figure 29 - MoviePageViewer Class

This page will be replaced inside the homepage with the Top Movies Page regardless of the user's logged in the situation.

Users can reach this page from category panel, top movies panel and search button.

With the help of this panel, users can see a small description of a movie like a director, year, duration, category and rate. Furthermore, users can see the cast and the summary about the desired movie. Only the users that were logged in can rate or add the movies to their will watch list. Logged in users can see and change the rate which was given by themselves before.

Users can watch the selected movies regardless of their status of logged in.

3.3. User Management Subsystem

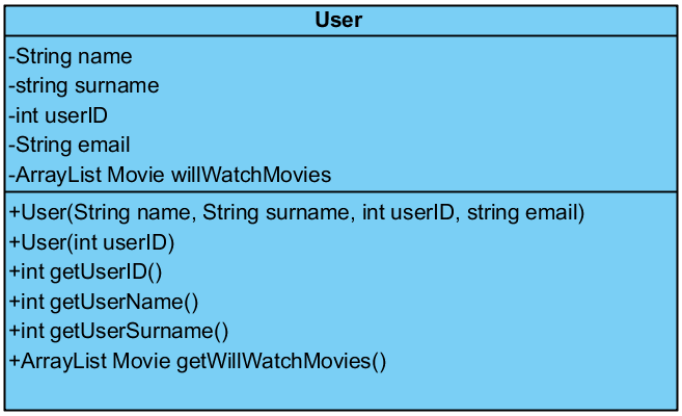

Figure 30 - User Class

Because the application has only one type of user, User management is far easier than the other management subsystems. The user is a general class that holds all the information about the user. When guest user wants to login to the application

3.4. Movie Management Subsystem

Finding and watching movies is available for both guest and logged in users. Users can only watch movies via "MoviePageViewer" class. When users click watch button inside the "MoviePageViewer" class, program will automatically start a media

player program which was coded with JavaFX. The program will find the desired movie according to movie's special ID which is now shown to the user.

3.5. Rating Management Subsystem

The rating process is only available for the logged in users. Hence, this system needs to reach users login information whether let them rate the movies or not. This is a subsystem which uses one of the database classes in order to access movies' rates. Moreover, logged in users can rate the movies. After the user's rating process, rating management system will be updated in order to demonstrate the new rate of the selected movie. Lastly, after user rates the movie, he/she can see his/her own rate for this movie when they re-enter the movie information page.

3.6. Search Management Subsystem

The MovApp is created crucially for being a user-friendly program. One of the most desired features from MovApp is to use it or access it in a fast and easy way. Search Management Subsystem is an example of it. Search management Subsystem is a simple subsystem where only one class handles the necessary task. AutoCompleteDecorator is a critical task which is basically a word completion. MovApp's search button pop ups the strings when they become pairs and compare them within database's pairs. Hence, it is connected to the database system in order to access the movies' name.

3.7. Homepage Management Subsystem

Applications Homepage is managed with the help of the class called "ProgramActionListener". All the view classes are used inside the controller class. According to user's actions, controller class changes the frame by combining different type of view classes.

When the program first starts, the application uses "TopFixedPanelViewer", "HomePageViewer" and "CategoryPanel" classes for forming the home page. When the user presses one of the movies inside the "HomePageViewer" panel or "CategoryPanel", controller class removes the "HomePageViewer" and add the desired panels information page which is called "MoviePageViewer". Furthermore, when the user clicks on logo button, controller class removes the "MoviePageViewer" panel from the screen and adds "HomePageViewer" panel back. Furthermore, when the user presses the login button, all panels inside the frame is removed, then two new panels, "LoginPanel" and "LoginVirtualKeyboard" panel is added to the frame.

3.8. Data Management Subsystem

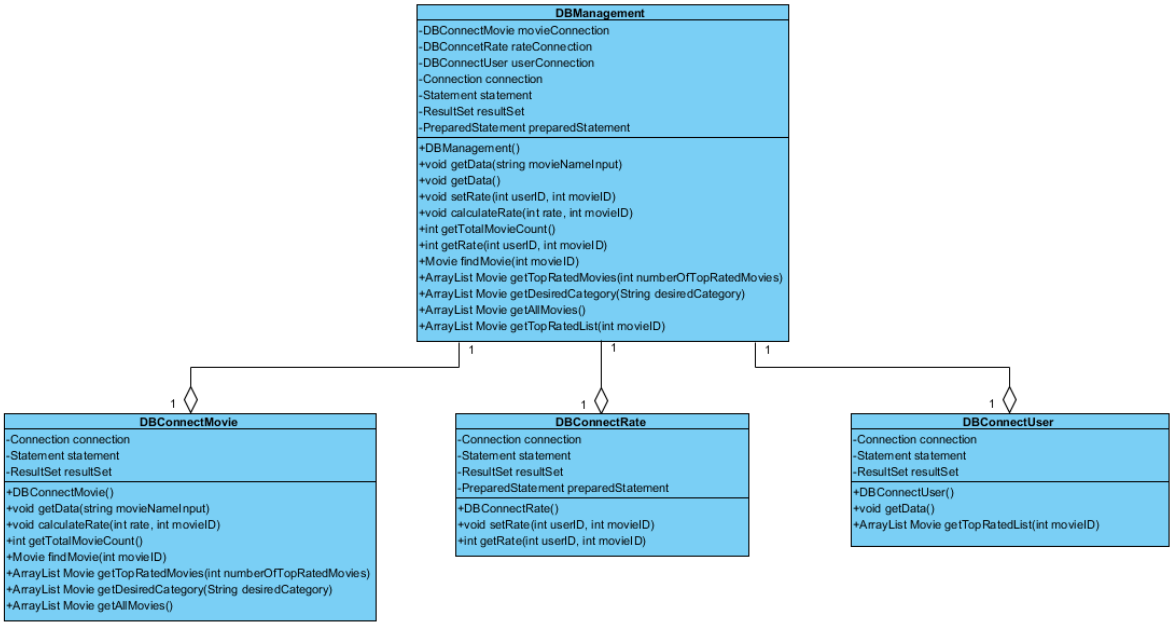

Figure 31 - Database Class Diagram

The figure that was shown above is the applications Data Management Subsystem. This system has four total classes. One class is related to all the other three classes. Every database class inside the subsystem is directly related to one the specific topics such as rating, getting the movie.

3.8.1. DBManagement

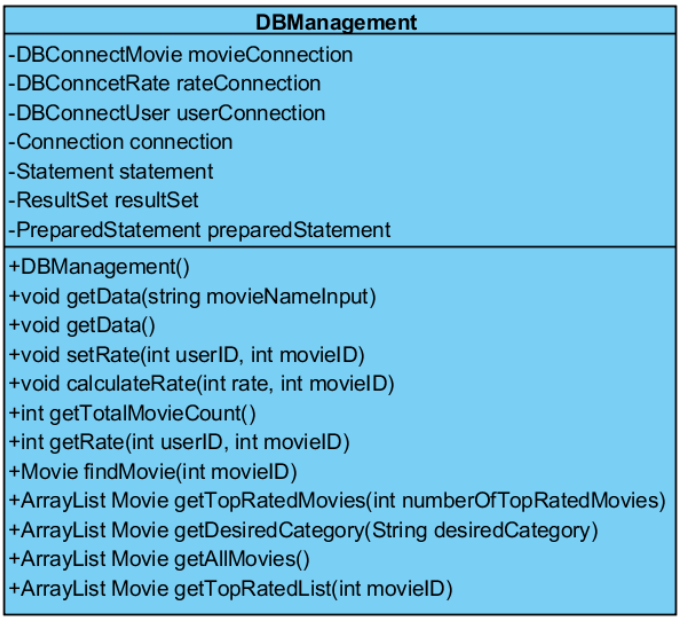

Figure 32 - DBManagement Class

DBManagement class is like a roof class for the other database classes. This class connects all database classes in one class. By using this class, database classes are connected themselves with a class that behaves like a controller. Moreover, rather than calling all database classes in project classes, this database class will be created automatically by the application.

3.8.2. DBConnectMovie

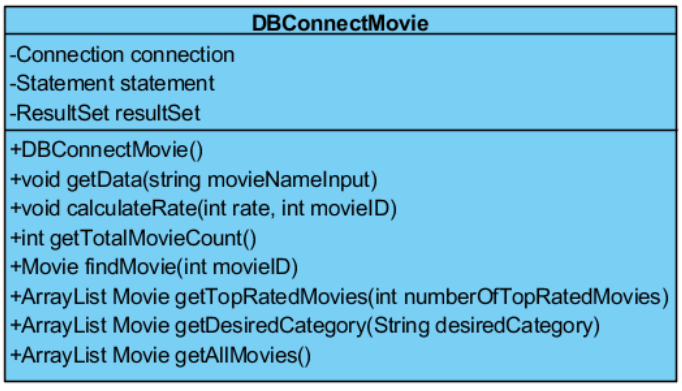

Figure 33 - DBConnectMovie Class

"DBConnectMovie" class is one of the sub database classes in MovApp desktop application. This class is used by the main database class which called "DBManagement". With the help of this class, the application creates a connection with the database which holds movie information such as name, director, description.

3.8.3. DBConnectUser

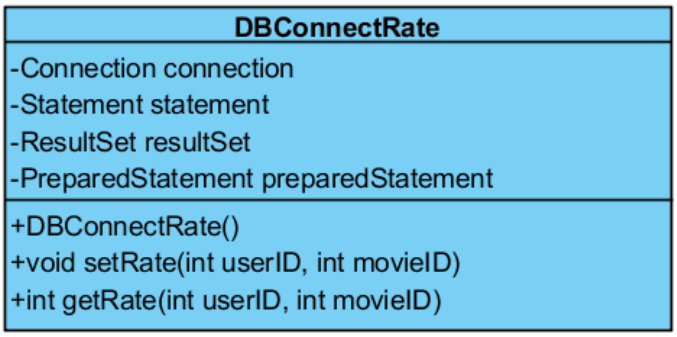

Figure 34 - DBConnectRate Class

DBConnectRate class is used for the purpose of reaching movie rates which were given by the logged in users. This database class holds only which users give which grade to which film. By using this class, applications main database class DBManagement generates the total rate for each movie. After the rates generated individually, DBManagement class writes the data to the related classes.

3.8.4. DBConnectRate

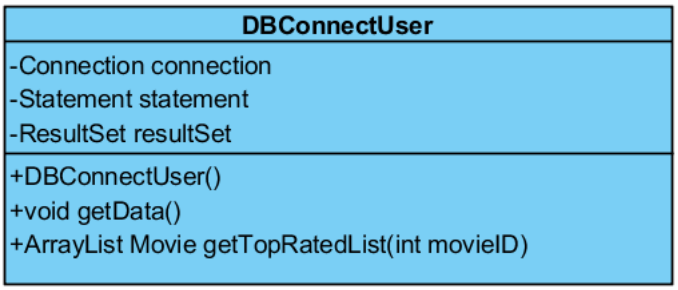

Figure 35 - DBConnectUser Class

"DBConnectUser" is a database class that checks the user id if it exists in the database or not. This class contains other data information such as user's name and surname. Moreover, this class will be used for the new users as well. If the guest user wants to open an account for this application, all his or her information will be written to the database with the help of this class.

3.9. Integration Between Classes in the Context of Use Cases

3.9.1. Guest User Login

This is the essential function of MovApp because the program offers some additional features for users and in a deeply way it is aimed that program will be used by a user who travels in regular periods. It is initiated directly on the main screen. When the user wants to login, she or he enters ID. In this level, the program accesses "ConnectUserDB" database and checks if there is a user

with that specific ID. With the help of "ConnectUserDB", users can enter the MovApp or see a warning "ID is NOT found".

3.9.2. Search Movie

The MovApp gives the option of search button for any kind of users. You can search for any movies via search bar with the help of touchable- virtual keyboard whether you are a guest or logged-in user. The search management is a user-friendly task for MovApp. The user can find a movie whatever they desire to watch with not writing the complete name of the movie. Its AutoCompleteDecorator's job to figure out what is coming next to that string. It can understand and shows the user the prediction. That happens through accessing "DBConnectMovie" in order to find out movie's name meanwhile user is writing. Hence, the accessing and responding phases have to be as fast as possible.

3.9.3. Rate Movie

Rating operation is essential for logged-in users. Users can rate the desired movies and see the rate of the movies. Also with the specific colour design, they can identify what they rated before for the specified movie. When users click the desired movie, they can reach the movie description page. Accessing this page will be done through "DBConnectUser" because database needs to get rated points and shows them in description part when user go to movie description page. It also needs to update the rate when users rate for not-rated movie or desire to re-rate.

3.9.4. User's Top List Panel

User's Top List Panel is a panel that shows top rated movies for that specific user. The application connects to the database and selects top 10 movies that user rated highly. For doing that, with the help of the controller class, everything inside the frame except "TopFixedPanelViewer" is removed. Then, "TopMoviesPanel" is added to the frame's centre. "TopFixedPanelViewer" is not removed because navigating back to the home page could only be done with the help of "TopFixedPanelViewer".