## Project Documentation

### Topic: GUI: Cinema Booking System

## Author: Myrzabek Darkhan

## Index

## 1.Introduction

## 2.Coding part with screen shots

## 3.Conclusion

### 

### 

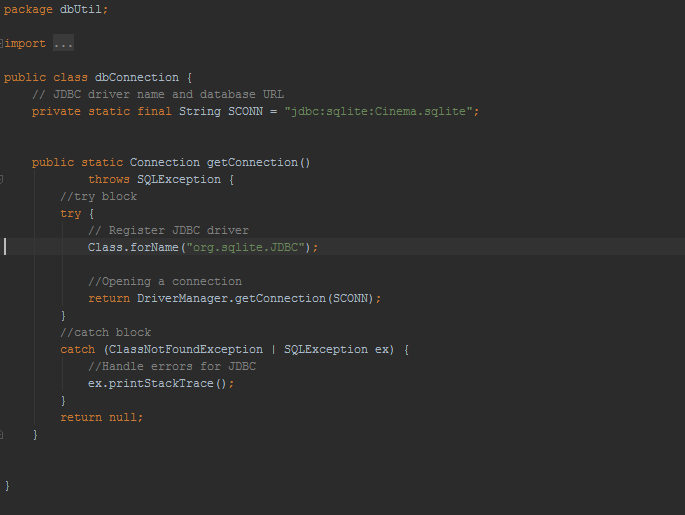
### INTRODUCTION

## This project is aimed at developing a Cinema Booking System for particular Cinema Halls. The Cinema Booking System is an booking application based on javaFX .This application will automate the reservation of tickets and enquiries about availability of the tickets and seats for particular cinema’s halls. This is one of the important features of our system. First customer buy reserve seats by this application and go to the cinema where he or she booked film. Then stuff in the cinema can easily check the reservation by this application. customer has to visit cinema hall for further process. So , the customer can pay from the cinema hall. Hence both the customer and the cinema can sort out further matter .For every cinema there will be one application.

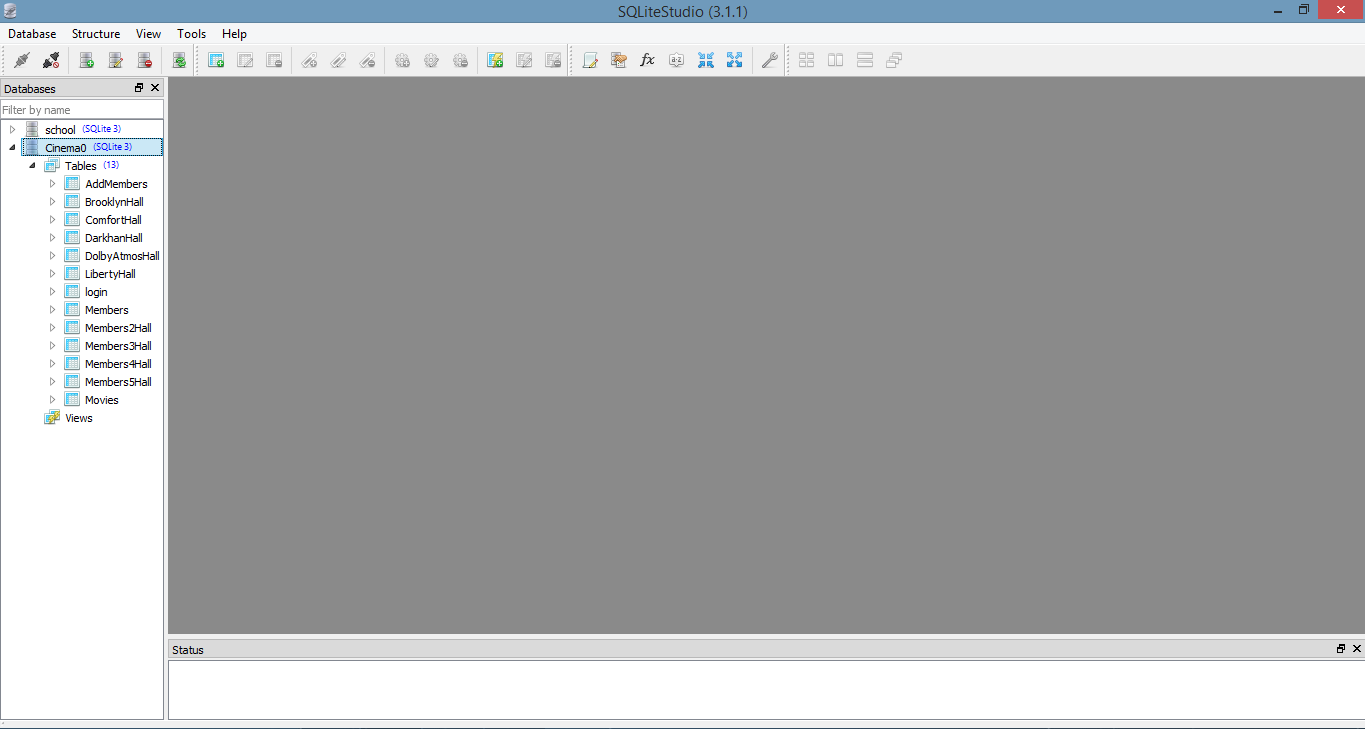
## 3.Coding part with screen shots

Firstly I am going to show database,how I created all this stuff.

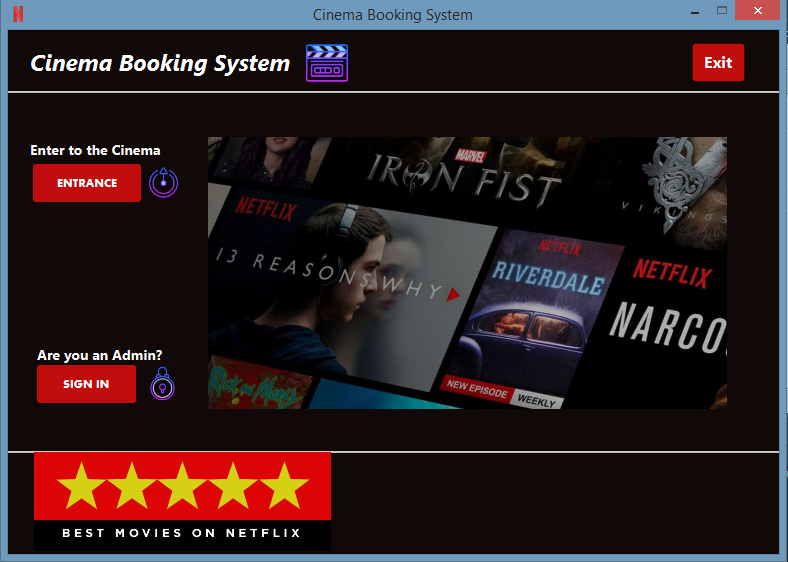
In this Cinema Booking System as database I used sqlite. First I am going to show how I connect it with java. For this I created class dbConnection . Here I showed path to my sqlite file (Cinema.sqlite).Then basic connection was made.1



The program which I used as a Database.



Let’s start from the main page. That’s how it looks when I run the project . Main things that there is 4 Buttons,2 textfields , and 5 labels ,one label shows wrong status when you have problems with username or password. Register button move to the Registration part, sign in button moves to the admin page.



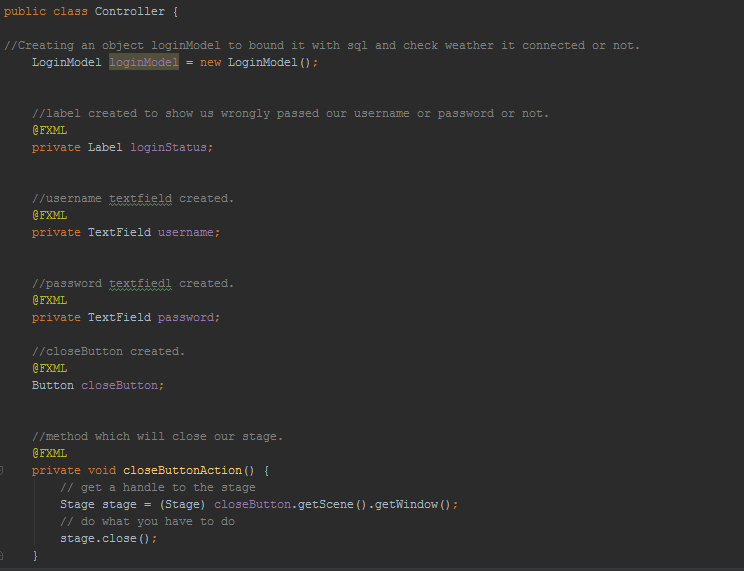
Here is the piece of code from above’s stage.



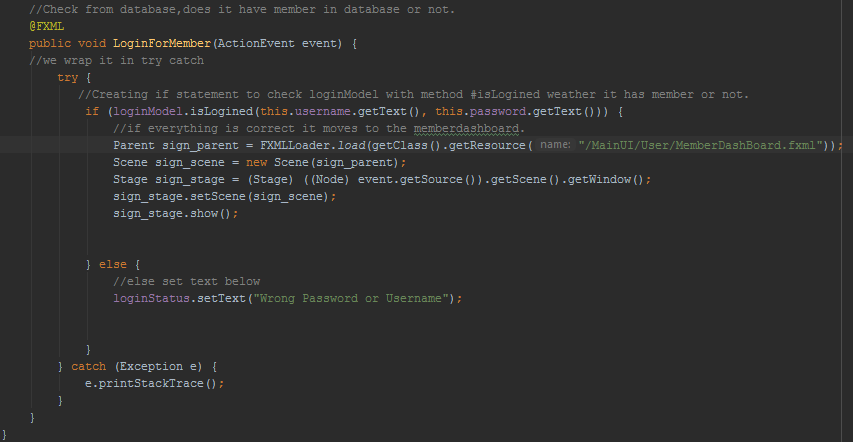
Main class ,from where we start our program, connect it with fxml (Login.fxml) setting scene,stage and with method getIcons setting the icons

Main controller

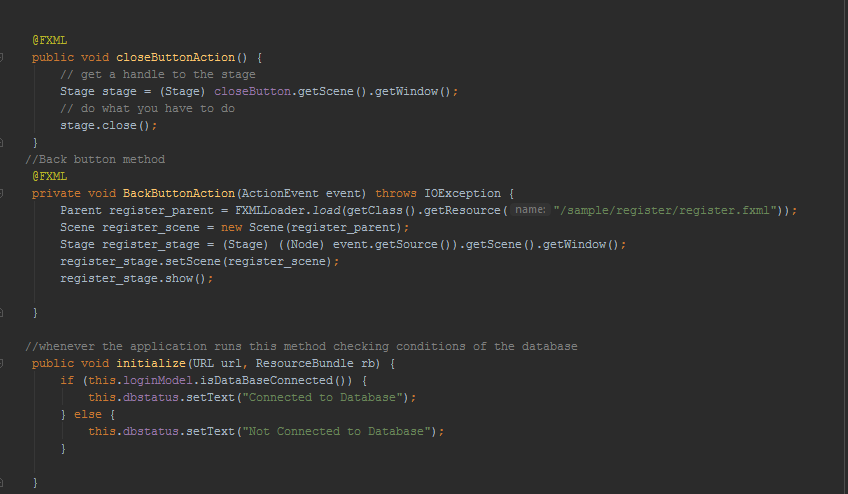
Main controller class where you move to the registration part, admin part , member scene. Below we are instantiating our attributes.



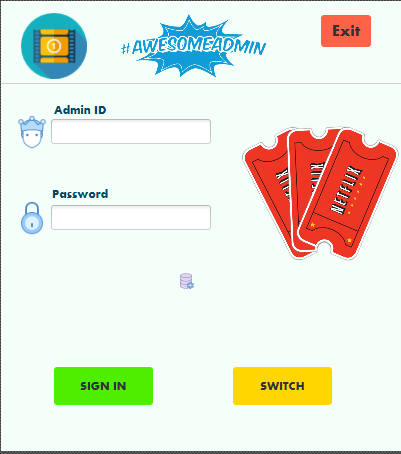
Method LoginForMember checks if there is exist registered member or not.I used here if condition to check, firstly it takes text from the textfield checks it with database,if it’s ok open memberdashboard. (about isLogin method we will talk about it later)



Method RegisterButtonAction creates new stage and setting it to scene which moves to the registration page. SignInButtonAction do to the same but moves to the admin page to sign in as an admin.



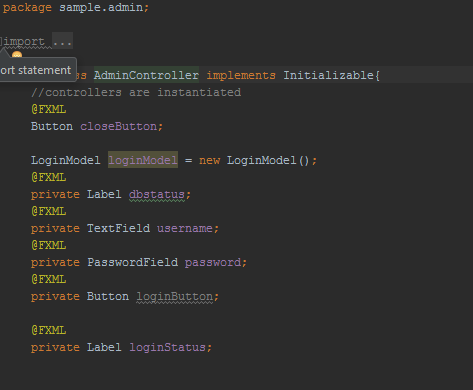
Here our Admin page.



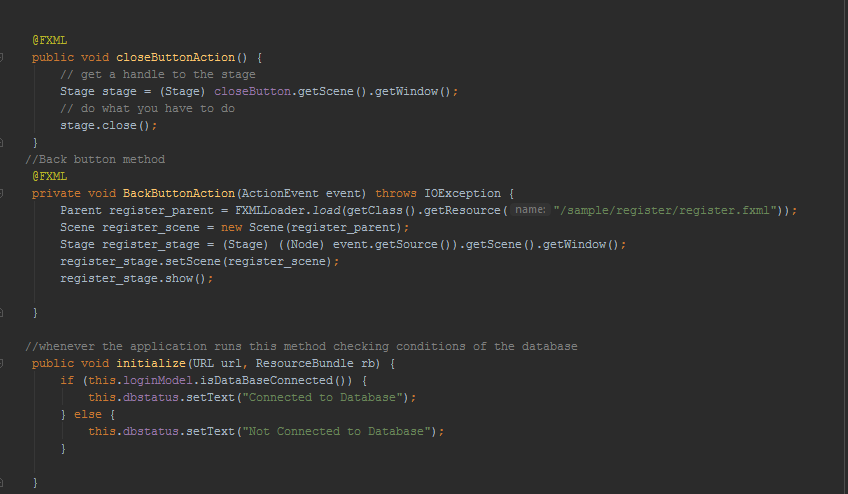
Admin part

Let’s move to the Admin part

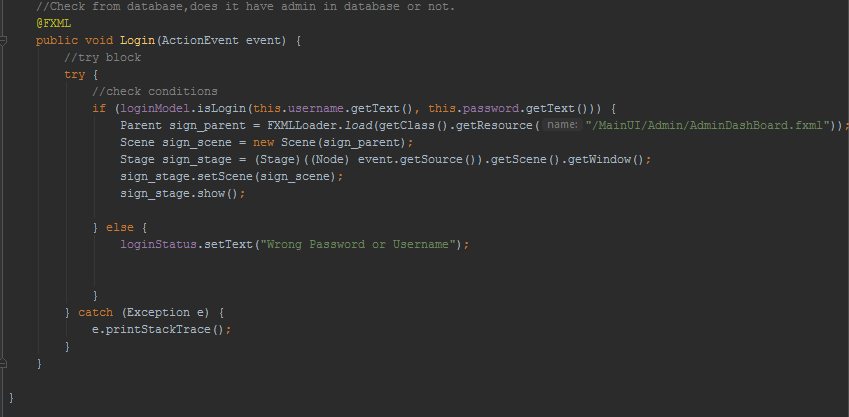
Here we are instantiating controllers



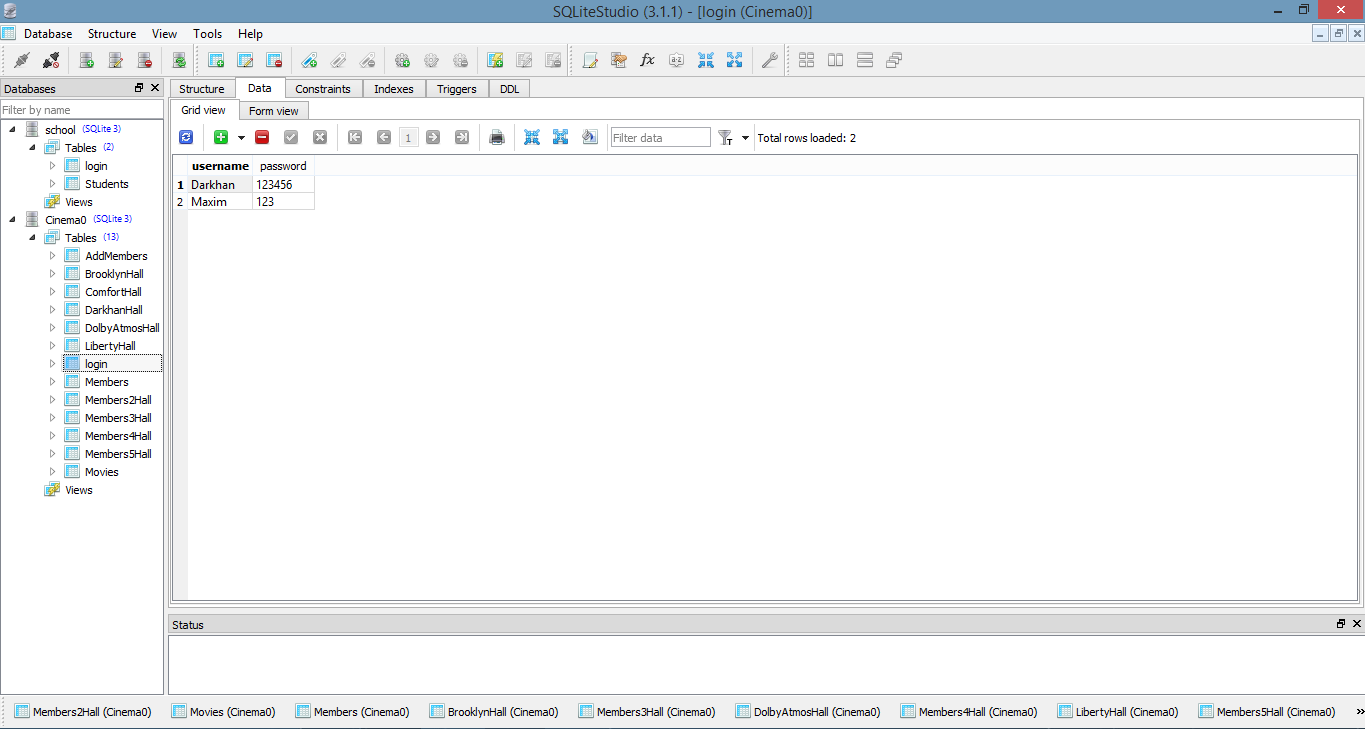
Methods close, back were explained above, The initialize method, which is needed to compare it with the constructor. In its essence, initialize performs the same function as setter, but is a different OOP block. It can be said that this was done so that the procedure for creating methods and designers was laid in the head.



Method Login same method as “LoginForMember” but it just checks admin’s input and further goes to the adminpage.

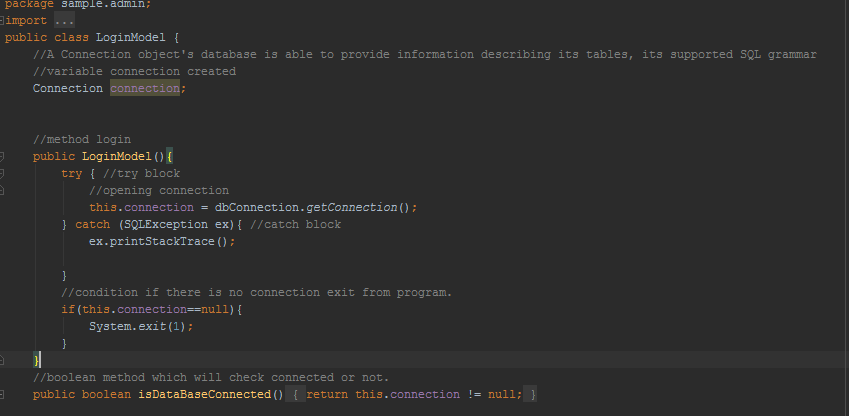


As you can see below the table with names and passwords, they are allowed to get all information about members. And can entrance as an admin. Program checks with input in the textfields and this table.

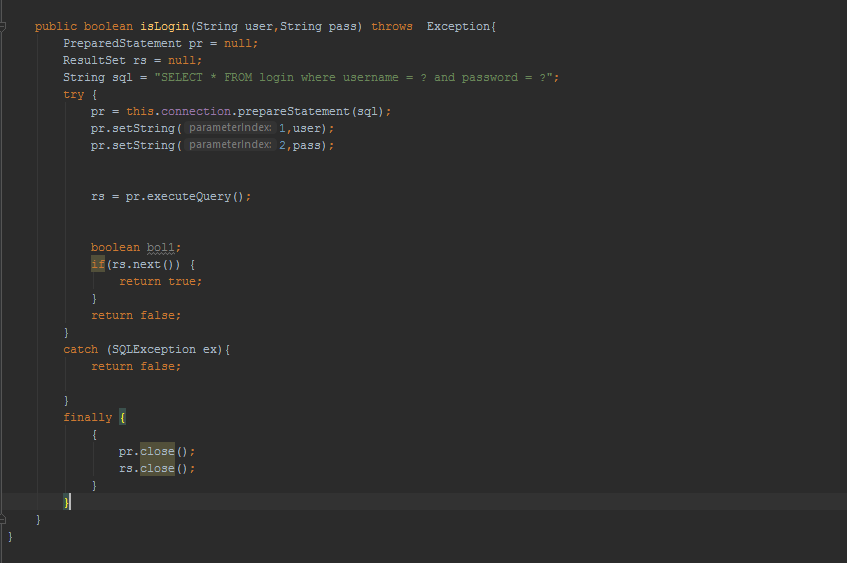


Let’s move to the LoginModel class.

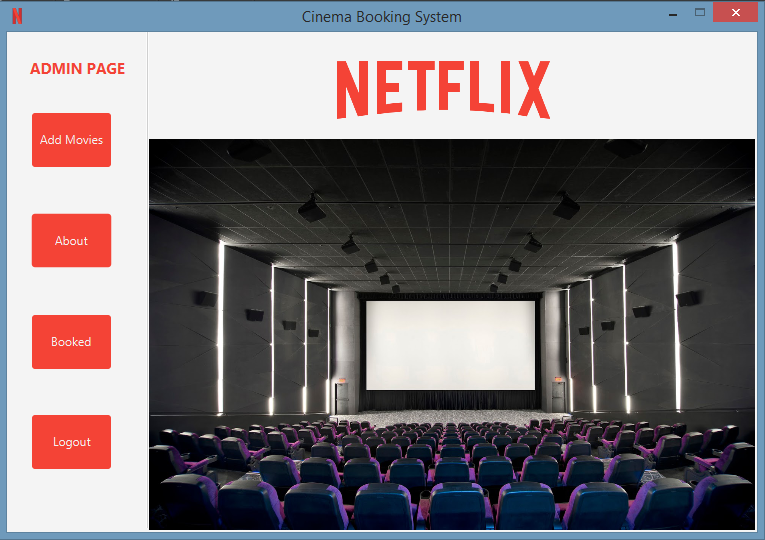
Here we are checking connection and admin’s input.



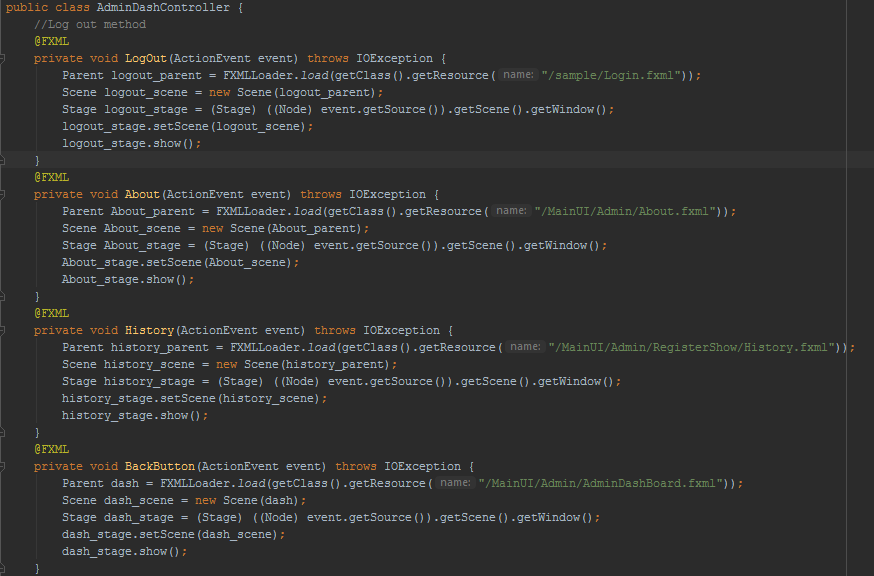
Method isLogin search from the table login and checks are you an admin or not. Firstly we store our request in String then prepare our statement with this string(sql).Then finally execute this request.



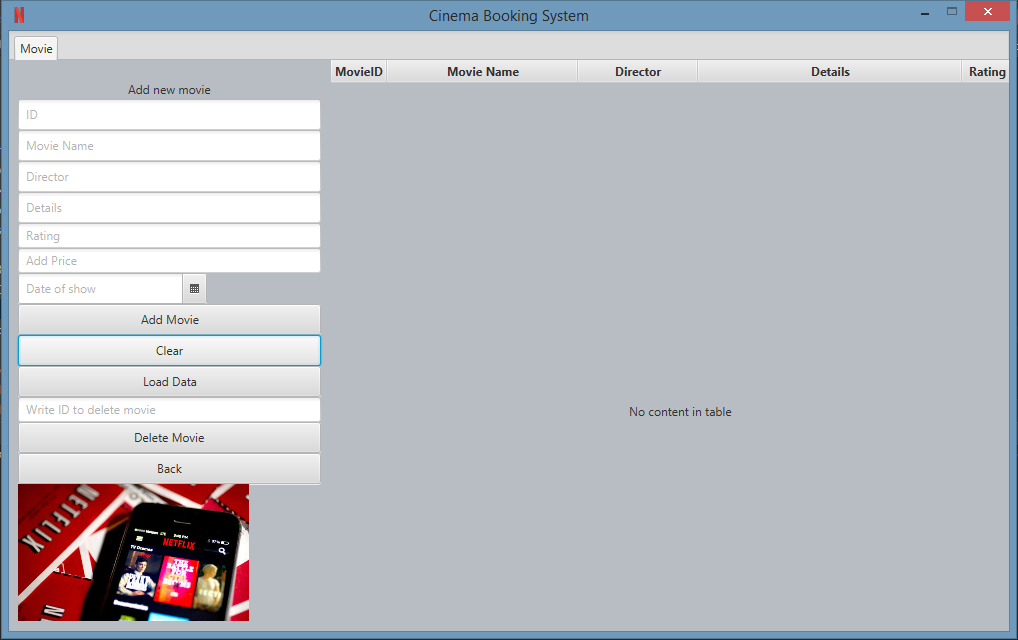
After you sign in as an admin you will see page below.Here 4 buttons addmovies,about,booked(history),logout.



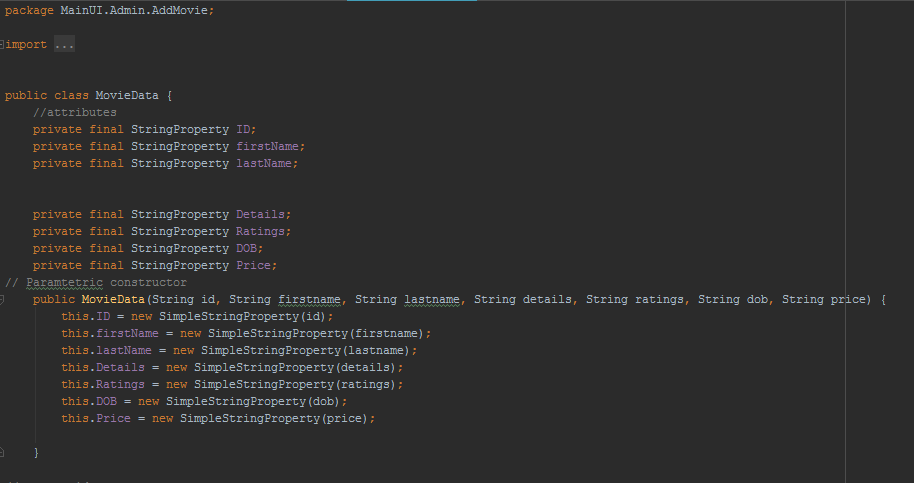
Here we have 4 main methods for every button. Method Logout left from the page .Method about provide information about admin. Method History shows us booked films from guests. Method Back button return to the MemberDashBoard.



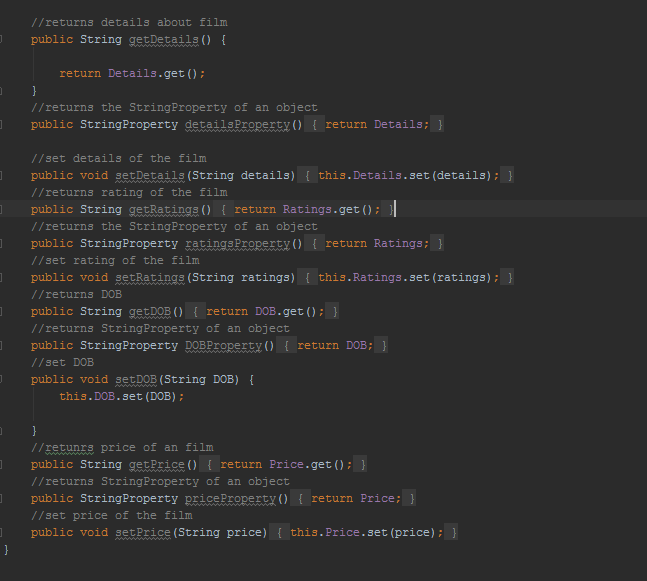
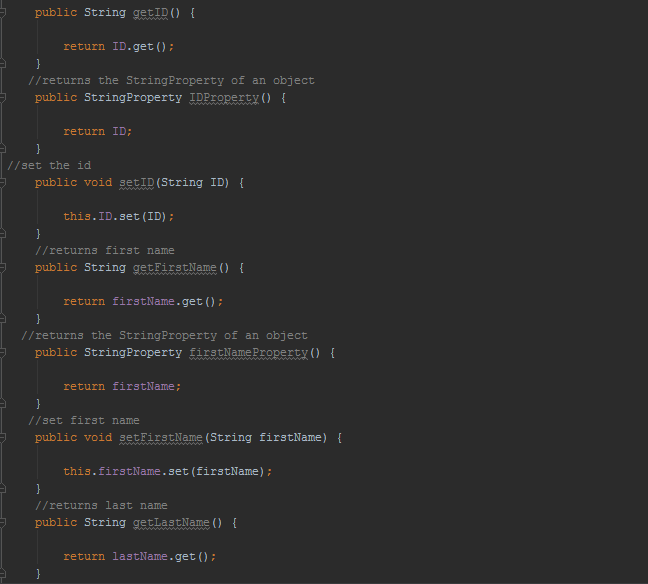
Whenever you press button addMovies you will move to the scene below. Here you can add, see or delete particular movie. For all this stuff respond 2 classes MovieData and MovieController.



Let’s start from the class movieData. Here we instantiated attributes and created constructor



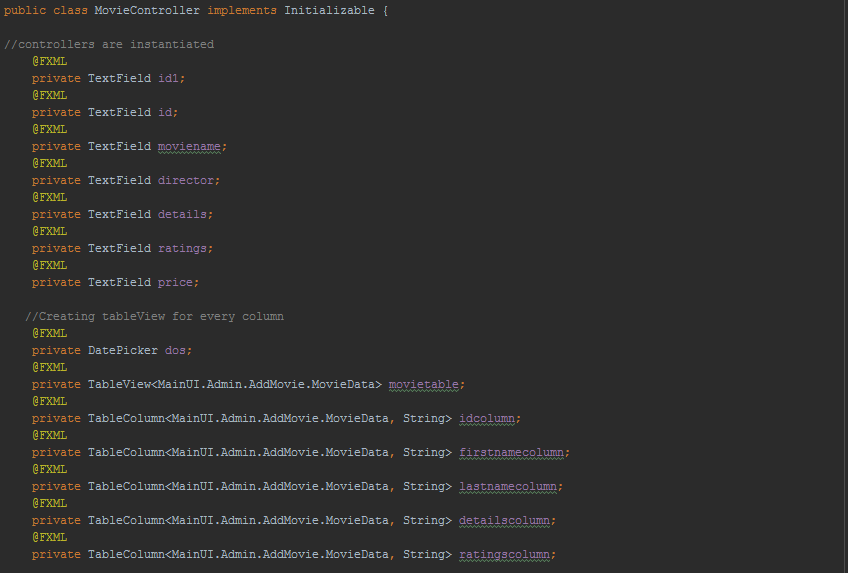
And for every column that we have we are creating getters and setters



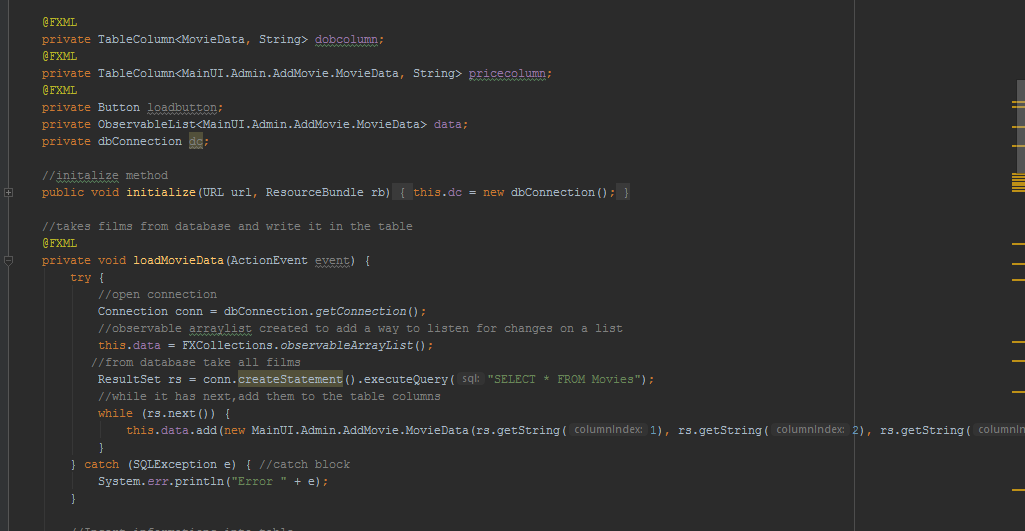
It will return a user with all its attributes loaded from the database. Hence, load all the attributes of the object when you first create it, don't try to lazily load each attribute one at a time. It will be simpler and more efficient.

Let’s move to the MovieController class

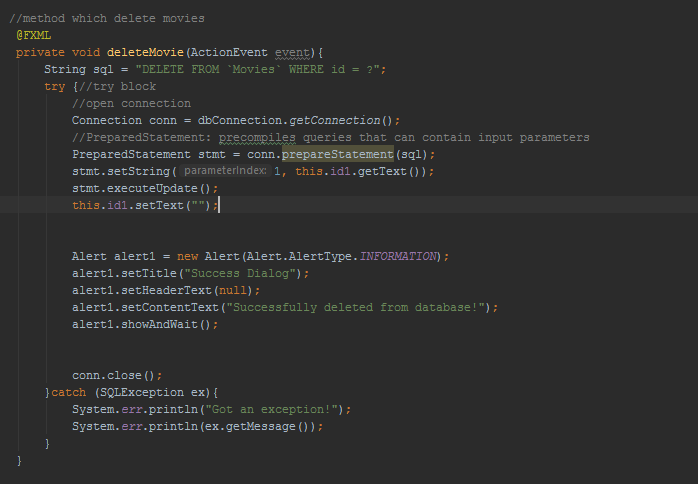
Here we are instantiating our controllers.And creating tableview for every column



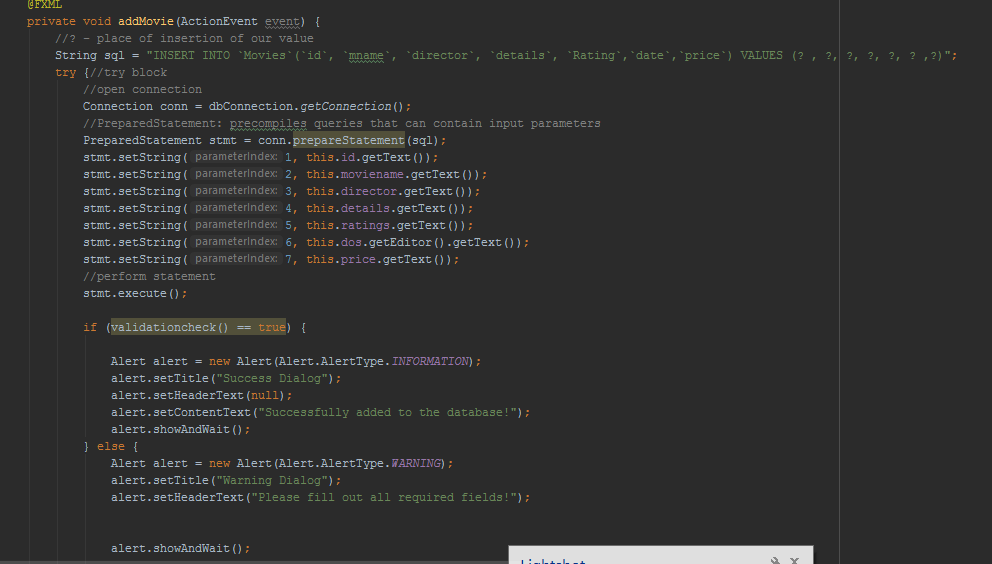
First method loadMovieData takes movies from the database and insert into table.Observable arraylist further will help us to store this informations. And we send a request to show all available films.



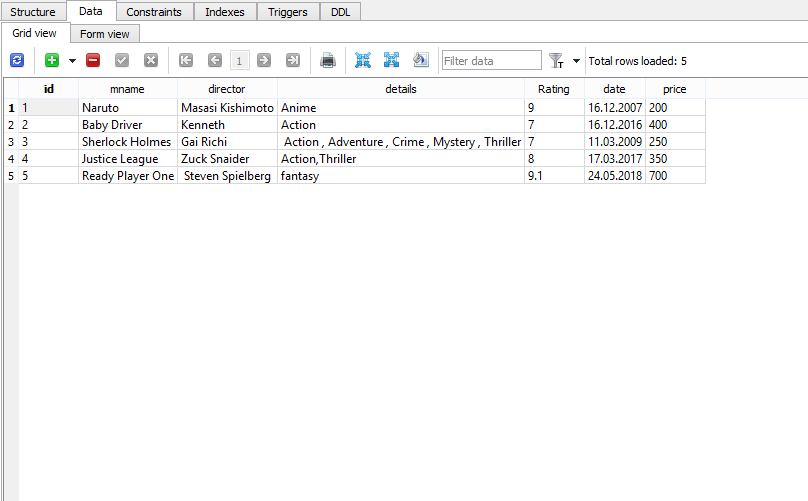
This method delete particular movie from database. You should just write id of the film that you are going to delete.



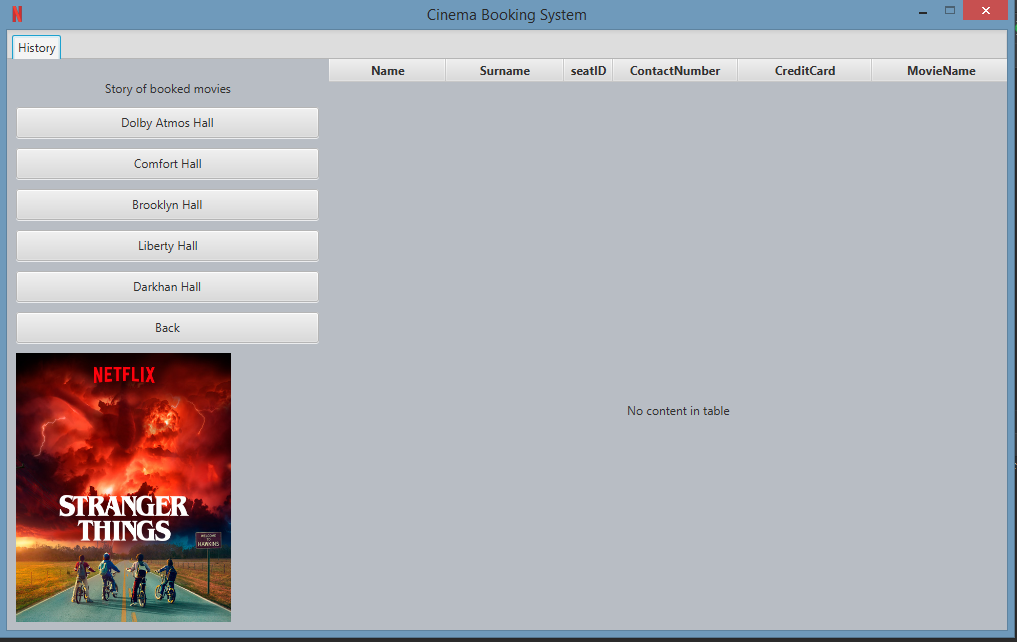
Method add movie insert into database new film that you write.Make a request which stored in string sql and all your input write it into the database, while checking conditions.



All movies will be stored and loaded from this table.(SQLite)

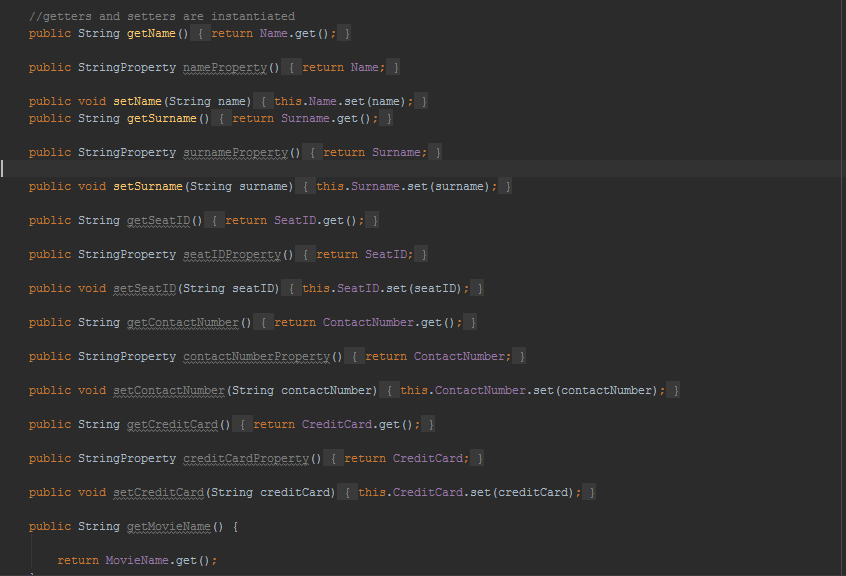
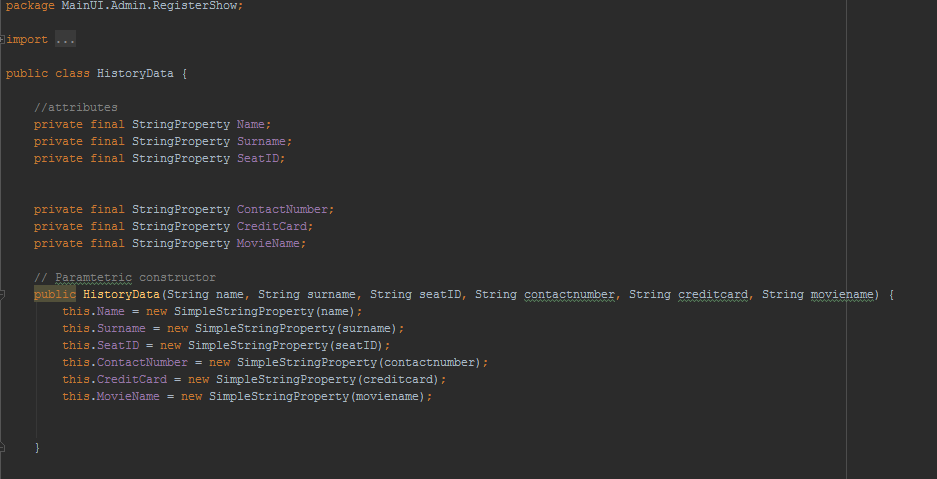


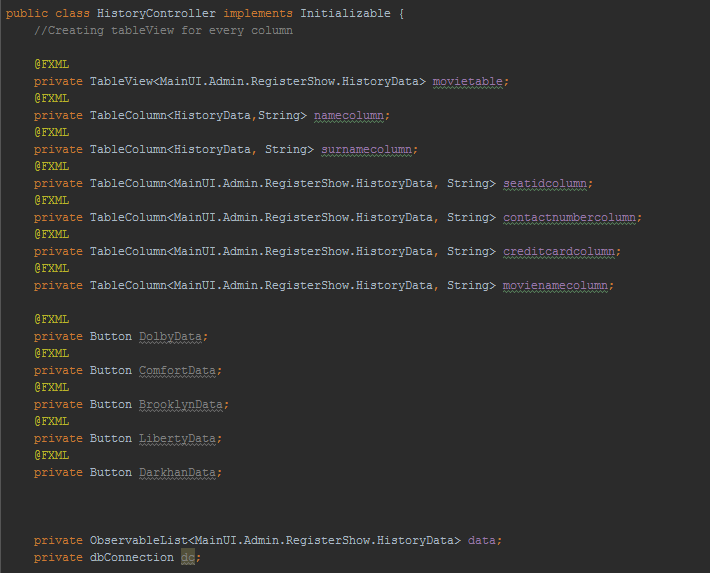
Whenever you press button History you will move to the scene below. Here you can see all booked movies by guests. For all this stuff respond 2 classes HistoryData and HistoryController.



Let’s move to the HistoryData class

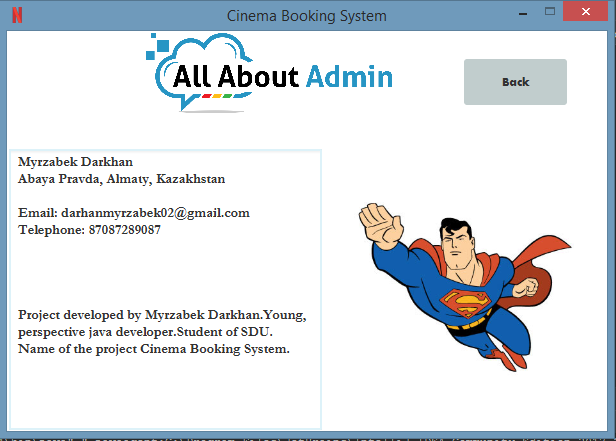
Here we also do the same,for every column that we have we are creating getters and setters







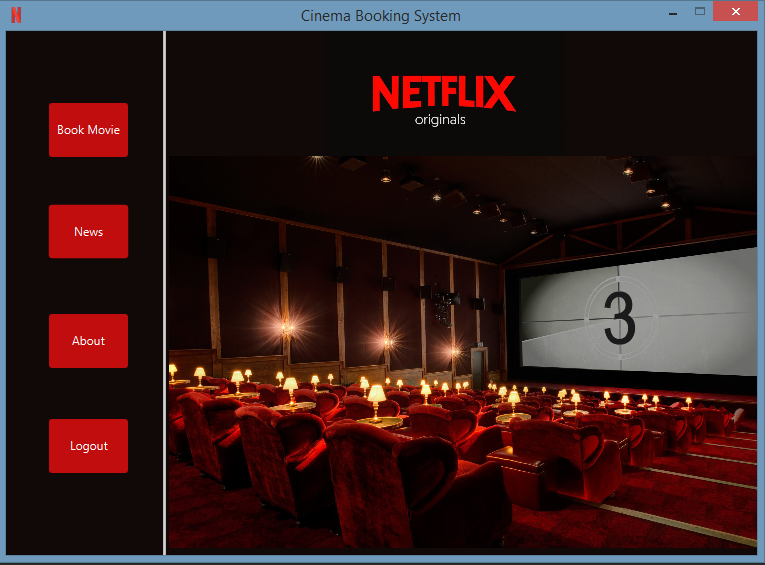
About button move to the scene where you find information about admin



Member part

Here we have 4 buttons ,book movie,news,about and logout.

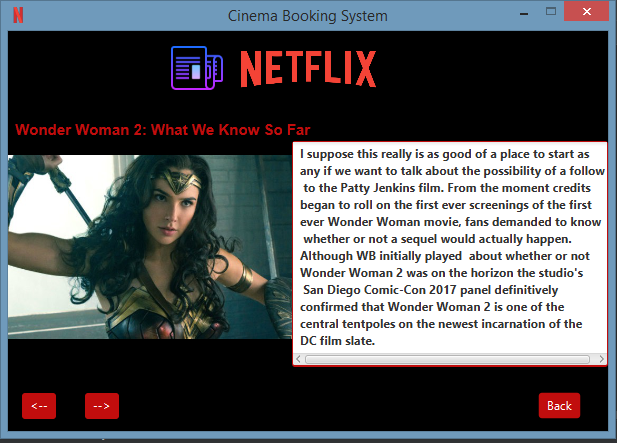
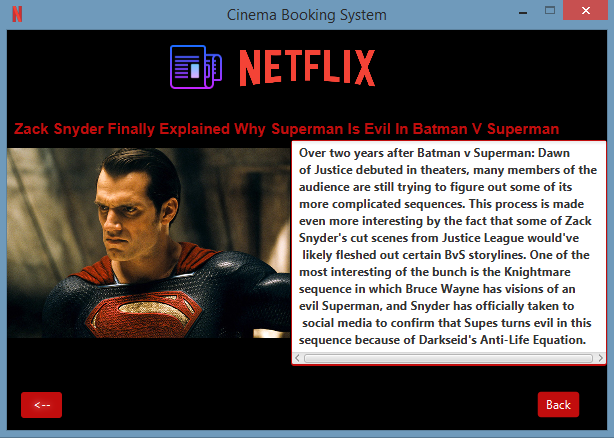
Furthermore I am going to explain precise details.



When about is pressed you move to the scene below, where you get information about this cinema.



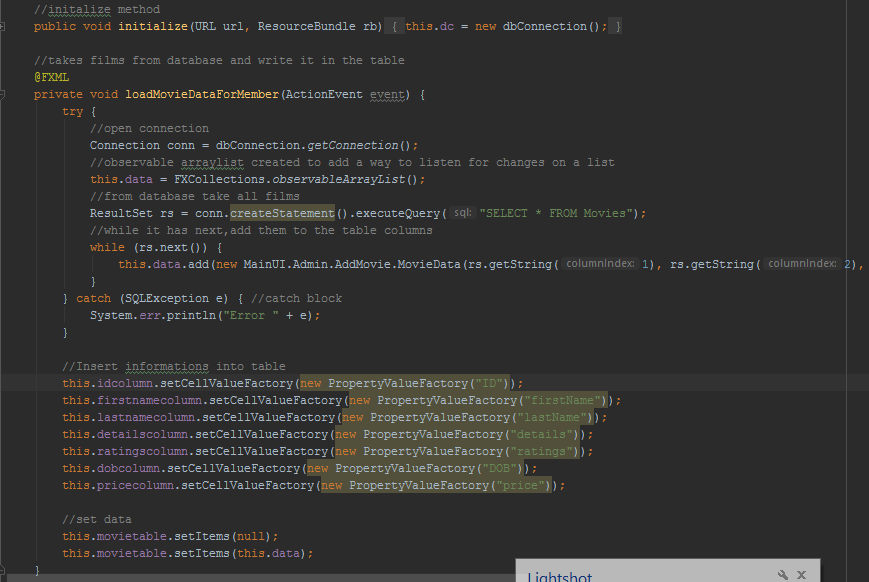
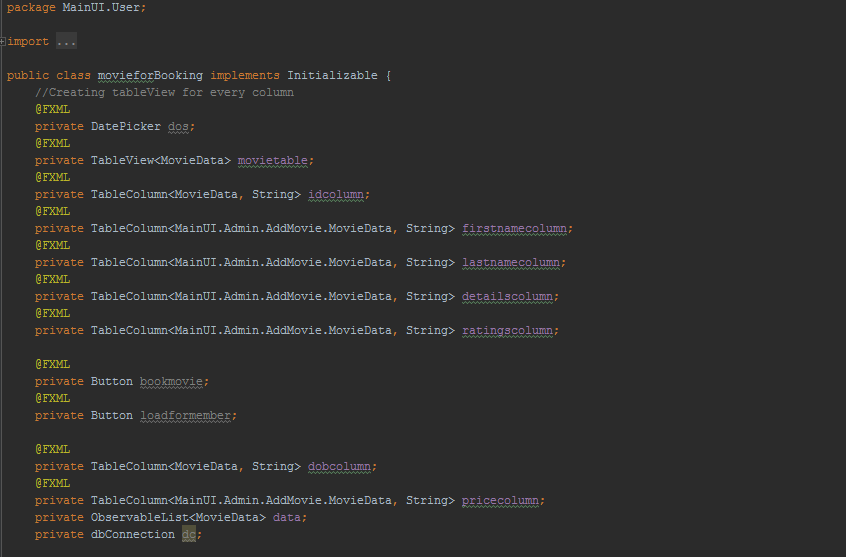
When news is pressed you will get latest news and all about cinema and cinematography.



The hardest part of my project was booking. And when button Book Movie is pressed you will move scene below. Button Show films load from database all available movies.



Here is the piece of code when I click button show movies.It creates TableView for every column and with method LoadMovieDataForMember it send request to get all information about films.It goes to the sql and find table movies.And all information there it write it to the TableColumns.

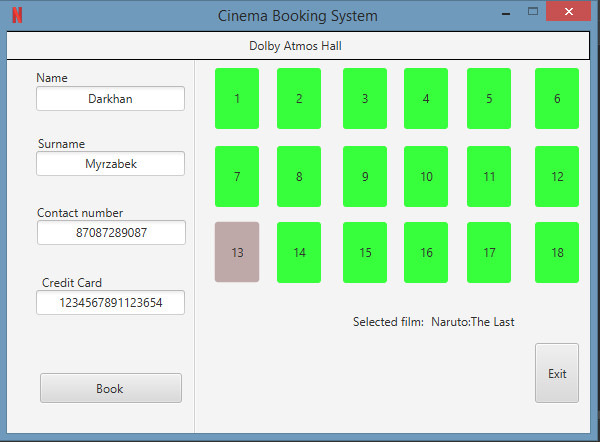


Then you will pick one of the halls and go for example to the Dolby Atmos Hall.As you can see

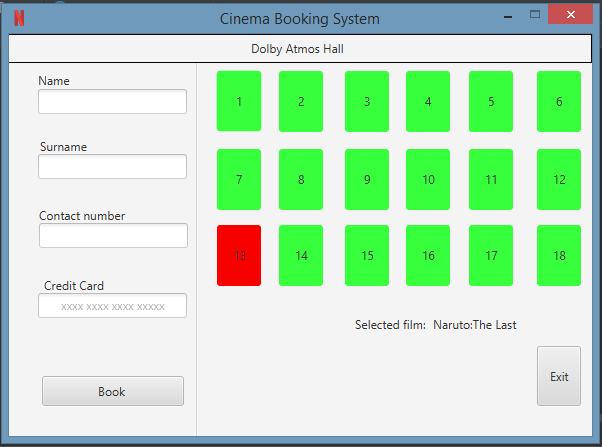
Now all seats are available.



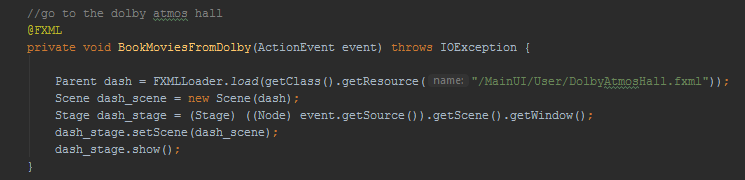
After we fill the fields and choose the seats, you can press book button.



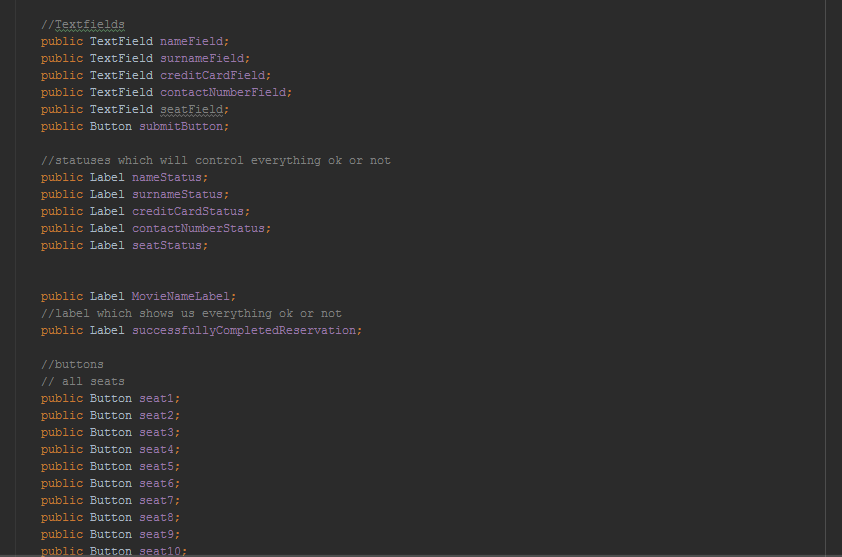
After booking you can see that seat is busy(red).



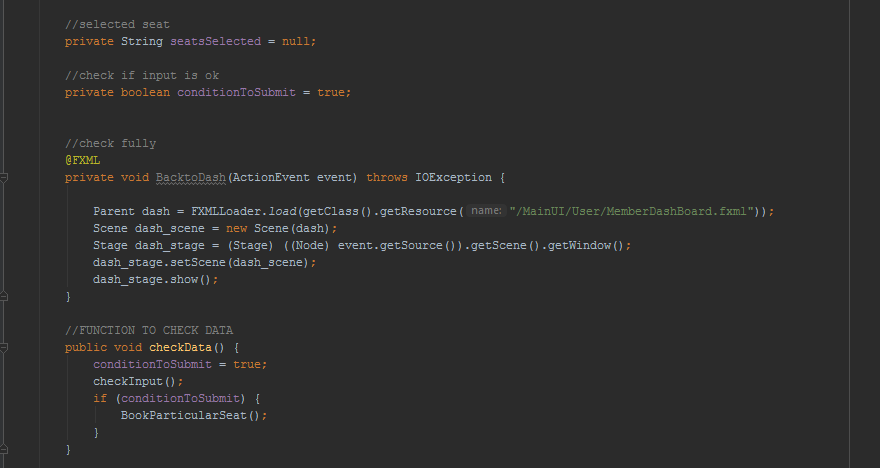
Path to the one of the halls.



Firstly we create all textfields and buttons.



For button book we instantiate method checkData it has two methods inside them, if both of them are sufficient ,book will be successful.



First method here, checkInput will check everything you write in the textfields with method getText. Every condition was very important.



Method bookParticularSeat insert into hall(for example Dolby Atmos) information that you wrote and seat that you choose. At this moment it will automatically insert into table MembersHall and into table DolbyAtmosHall it will set column available to “no” and number of the seat.

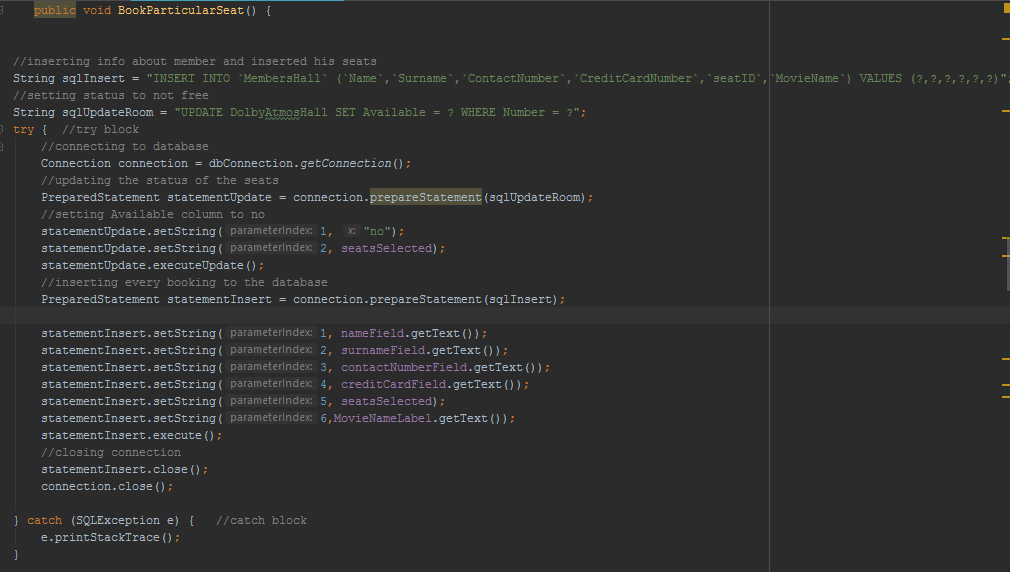


Table with your information

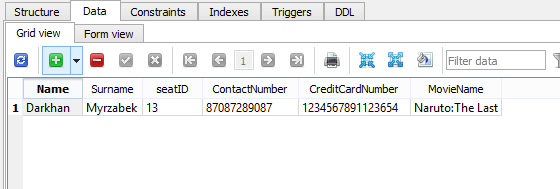
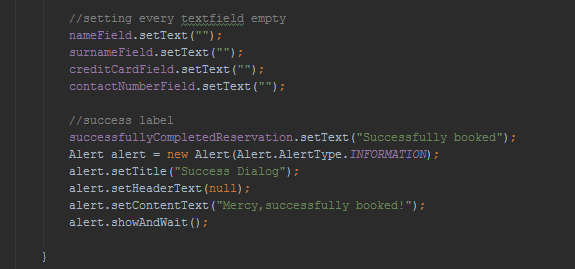
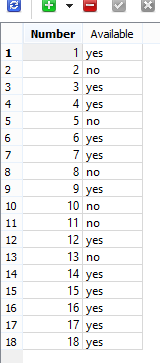
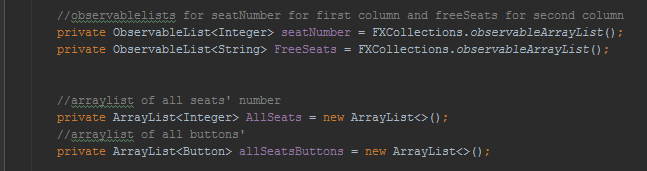


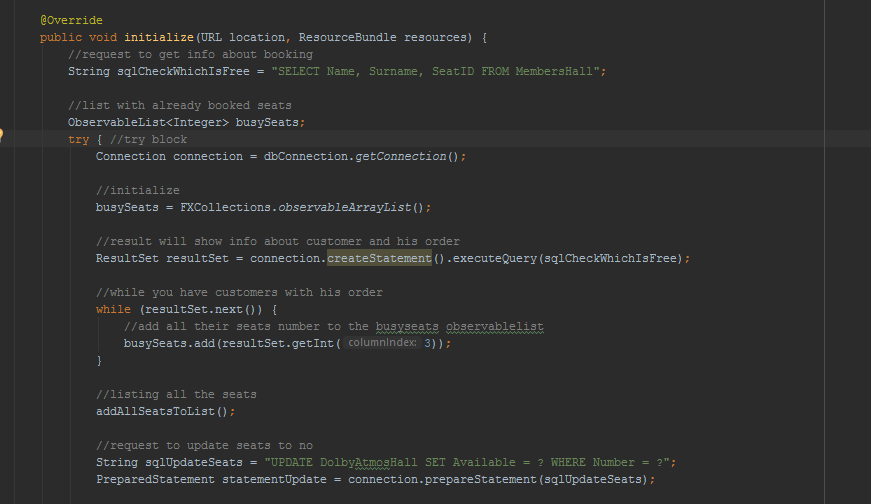
Table which checks condition of the seat.



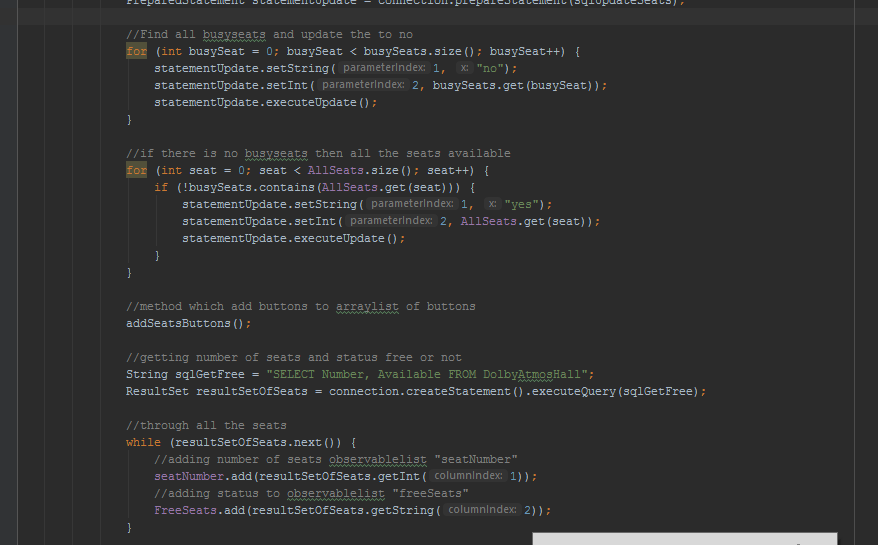
I created two observable arraylist for database where we have table (for example DolbyAtmosHall) to see their conditions.In this table I have 2 columns first shows availability (yes/no),second will insert the selected seat’s numbers.



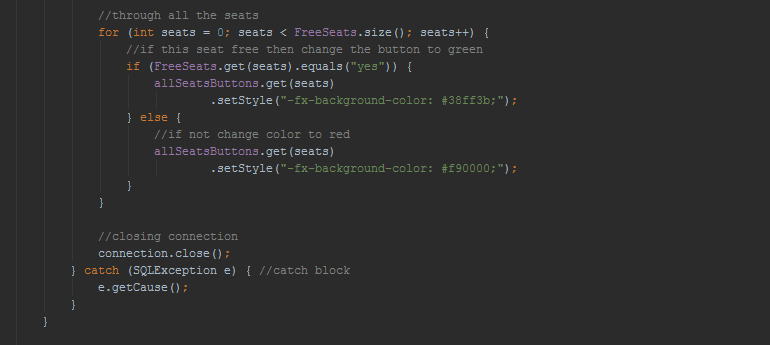
Here we start our request to the database for example in first hall (MembersHall) where we have to define which is free,all booked seats will be written in this table.And program will automatically save selected seats in observable arraylist (busyseats) which we created.At this time the program send also one request to update another table called(DolbyAtmosHall) where we will write condition and number of the seats.In the condition (Yes/no).



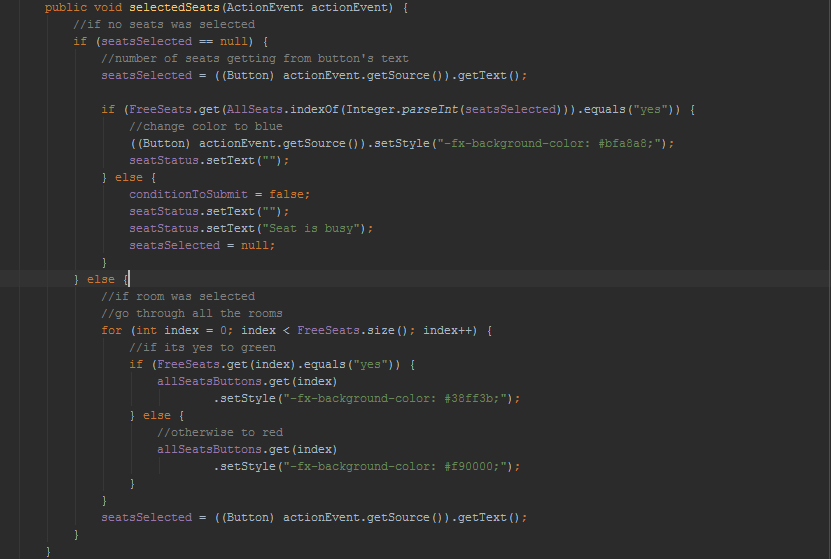
In this for cicle it go through busySeat list and in the database update them to no.Second for cicle go through list allseats and if there is no busyseats set them to the yes and in the database update them.addSeatsButtons method add all buttons that we instantiated to the list.And in while cicle we take our all availability of seats and number of seats insert into the table(both of them are observable arraylists).first column seatNumber,and second column freeSeats.



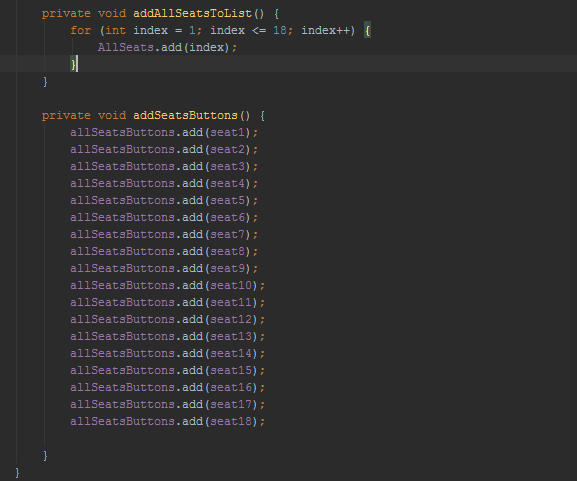
And finally go through all seats and if there is exist seat in FreeSeats arraylist then set their color to the green,otherwise to red.



When we select our seat(button) set color to the grey.



Through cycle for create seats’ numbers and add them to the allseats arraylist.Then created method which will add all buttons to the allseatsbuttons arraylist.



Conclusion.

This is my first project and hope you will appreciate it.

Link to the GITHUB:

https://github.com/MyrzabekDarkhan/CinemaBookingSystem