

Cinema Ticket Reservation

Documentation



Prepared by : Aiymbubu Kulbaeva COM-18

Github repository :

<https://github.com/akulbaeva/CinemaTicketReservation.git>

Bishkek 2019

Introduction

This project utilizes Java and JavaFX/FXML using an OOP approach to storing the movie and theater data.

The main purpose of ticket reservation system is to provide an alternate and convenient way for a customer to reserve cinema tickets. It is an automatic system. After the data has been fed into the database, the staff does not need to do anything with the order once it is received through the system. The goals of the system are:

To provide an anytime anyplace service for the customer

To minimize the number of staff at the ticket box

To increase the profit

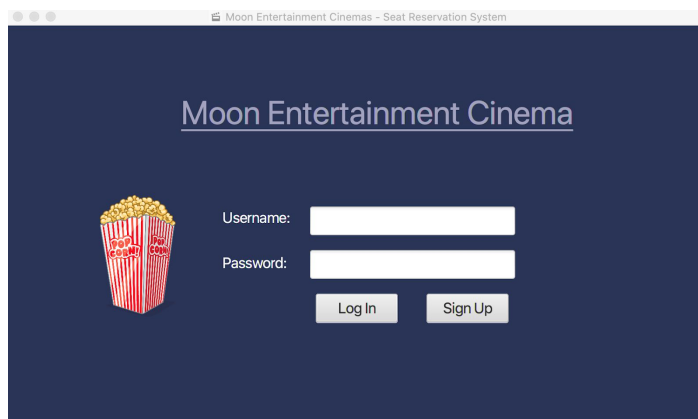
To obtain statistic information from the booking record.

Used technologies:

1. [Java 8](#)
2. [Scene Builder 8](#)
3. [MySQL Workbench](#)

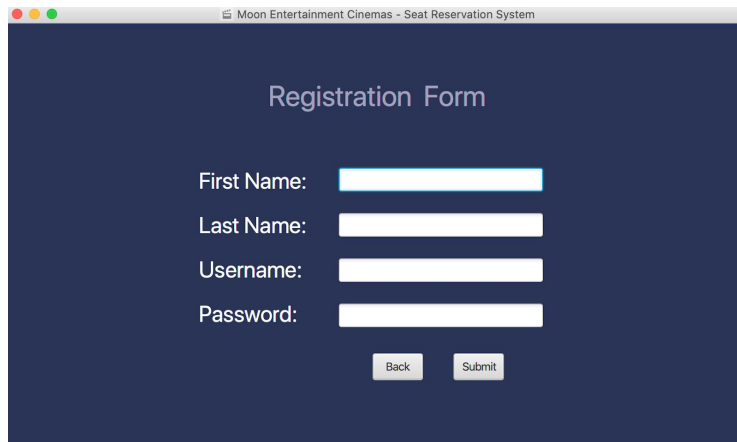
Execution Procedure:

When user executes this program it will show the startup GUI (Graphical User Interface) of this program.

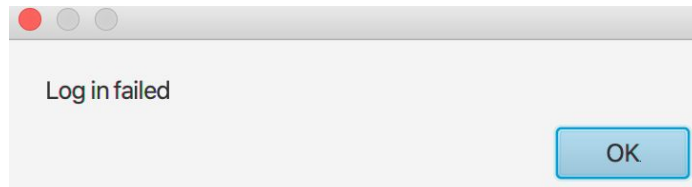


1. If user has already have the account they need to enter the username and password. Then press the “Log In” button.

If they do not have an account they need to press “Sign Up” button.

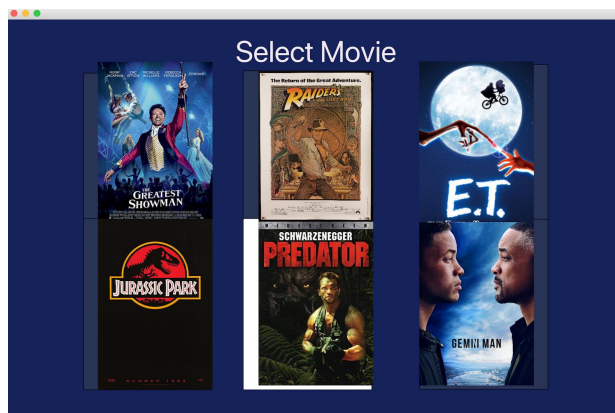
A screenshot of a web application window titled "Moon Entertainment Cinemas - Seat Reservation System". The main content area has a dark blue background and is titled "Registration Form" in white text. Below the title, there are four white text input fields labeled "First Name:", "Last Name:", "Username:", and "Password:". At the bottom of the form, there are two small white buttons labeled "Back" and "Submit".

2. After pressing the “SignUp” button the system will show “Registration Form”. After entering all Fields users need to press “Submit” or “Back” buttons.

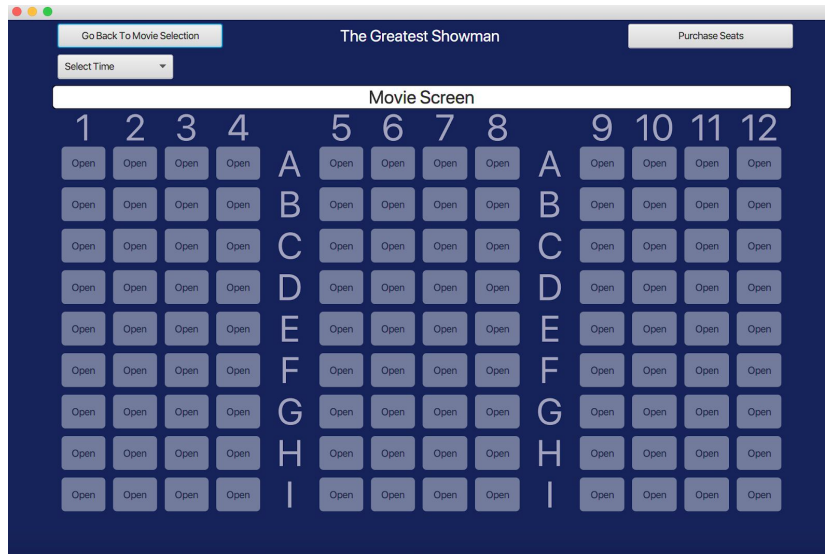


3.If they enter invalid username or password the TextField username and PasswordField will shake, then will show the warning

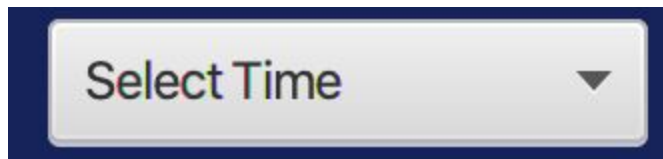
message.



4. If the user enters a valid username and password it will show the “Select Movie” on their screen. In this screen user need to choose the movie.



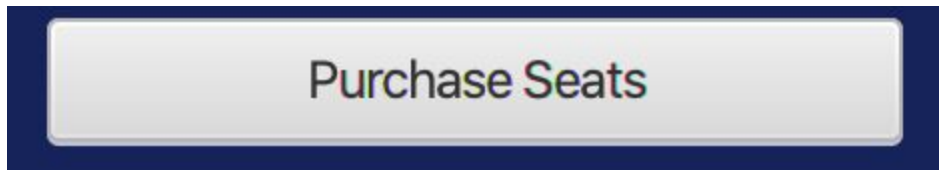
5. After selecting the movie the system will show “Seat Selection” on their screen.



6. To select the seats user need to first of all choose the time.

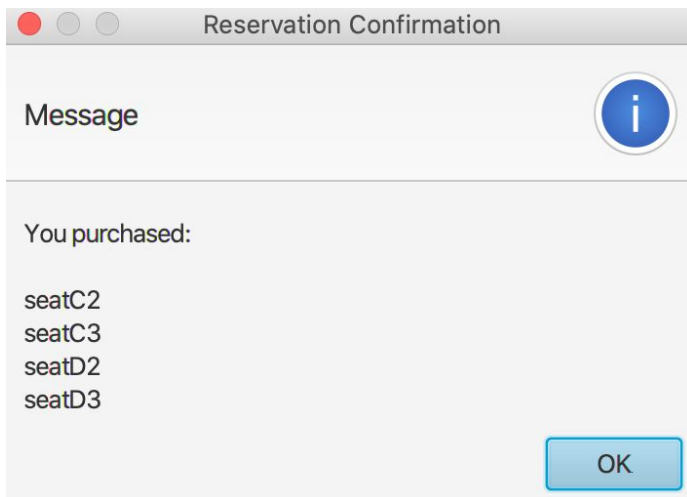


7. After choosing the time user can select the seats. When users select the seats Open Labels will turn into orange color and Open text will change into Selected text.

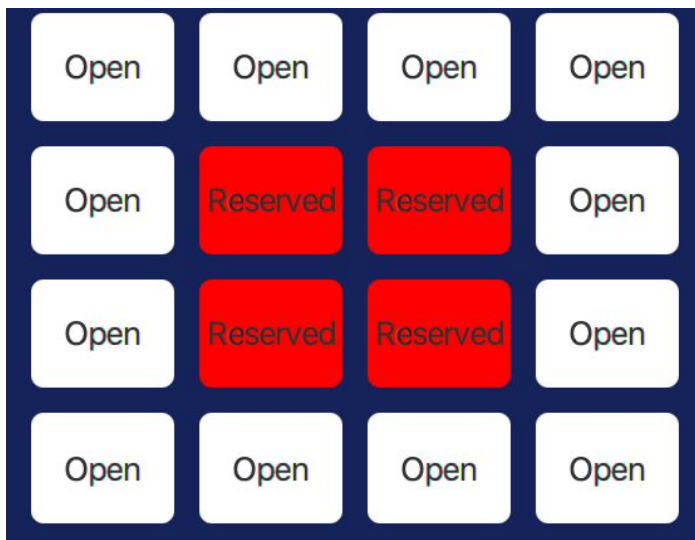


8. After select the seats user need to press the “Purchase Seats”

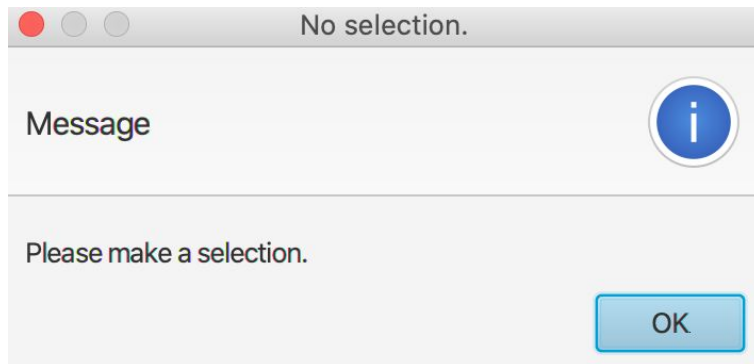
button.



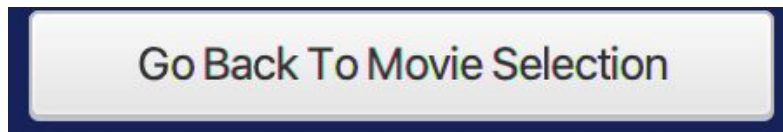
9. After pressing the “Purchase Seats” button the system will show the Message Alert where will shown which seats you choose. They need to press button “OK” to confirm the seats.



10. After confirming the seats which user choose the Selected text will turn into Reserved and color will change into red.



11. If user will not choose the time and press the “Purchase Seats” button the system will show a warning message.



12. If user accidently choose the movie or want to choose another movie they can

press “Go Back To Movie Selection” button and the system will show “Select Movie” on their screen.

Code part

```
private void loginUser(String loginText, String loginPass) {
    final Alert[] message = {null};
    DatabaseHandler dbHandler = new DatabaseHandler();
    User user = new User();
    user.setUserName(loginText);
    user.setPassword(loginPass);
    ResultSet result = dbHandler.getUser(user);
    int counter = 0;
    try {
        while (result.next()) {
            counter++;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    if (counter >= 1) {
        btnLogIn.getScene().getWindow().hide();

        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(getClass().getResource("name: \"SelectMovie.fxml\""));
        try {
            loader.load();
        } catch (IOException e) {
            e.printStackTrace();
        }
        Parent root = loader.getRoot();
        Stage stage = new Stage();
        stage.setScene(new Scene(root));
        stage.showAndWait();
    } else {
        Shake userLogInAnim = new Shake(tfUsername);
        Shake userPassAnim = new Shake(pfPassword);
        userLogInAnim.playAnimation();
        userPassAnim.playAnimation();
        message[0] = new Alert(Alert.AlertType.NONE, "Log in failed", ButtonType.OK);
        message[0].showAndWait();
    }
}
```

In loginUser() method, we are checking if the user exists in the database. If it exists, then the system logs in and goes to the “Select Movie” screen. If not, then the TextField and PasswordField are shaking and the warning message will show.

```
private void signUpNewUser() {  
    DatabaseHandler dbHandler = new DatabaseHandler();  
  
    String firstName = tfFirstName.getText();  
    String lastName = tfLastName.getText();  
    String userName = tfUserName.getText();  
    String Password = pfPassword.getText();  
  
    User user = new User(firstName, lastName, userName, Password);  
  
    dbHandler.signUpUser(user);  
}
```

We are calling `signUpNewUser()` method when we click “Submit” button in `SignUp` class. In this method, we are inserting all the details about users like `FirstName`, `LastName`, `UserName`, and `Password` and sending it into the database.


```

@FXML
private void clickMovie(MouseEvent event) {
    try {
        ImageView moviePoster = (ImageView) event.getSource();
        Stage stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        stage.close();
        stage.setOpacity(0.0);
        FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource( "name: \"SeatSelection.fxml\""));
        Parent root = (Parent) fxmlLoader.load();
        SeatSelection controller = fxmlLoader.getController();
        if (moviePoster.getId().equals("imageMovie1")) {
            controller.setMovieTitle("The Greatest Showman");
        } else if (moviePoster.getId().equals("imageMovie2")) {
            controller.setMovieTitle("Indiana Jones – Raiders of The " +
                " Lost Ark");
        } else if (moviePoster.getId().equals("imageMovie3")) {
            controller.setMovieTitle("E.T. The Extra-Terrestrial");
        } else if (moviePoster.getId().equals("imageMovie4")) {
            controller.setMovieTitle("Jurassic Park");
        } else if (moviePoster.getId().equals("imageMovie5")) {
            controller.setMovieTitle("Predator");
        } else if (moviePoster.getId().equals("imageMovie6")) {
            controller.setMovieTitle("Gemini");
        }
        controller.loadTheaters(moviePoster.getId(), theater1, theater2, theater3,
            theater4, theater5, theater6);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
        stage.setOpacity(1.0);
    } catch (IOException ex) {
        System.out.println(ex.toString());
    }
}

```

In clickMovie() method we select the movie and the system will show the “Seat Selection”.

```

private void loadSeats() {
    if (comboBoxSelectTime.getValue().toString().equals("9:00 AM")) {
        currentMovie = currentTheater.getFirstShow();
    } else if (comboBoxSelectTime.getValue().toString().equals("10:00 AM")) {
        currentMovie = currentTheater.getSecondShow();
    } else if (comboBoxSelectTime.getValue().toString().equals("5:00 PM")) {
        currentMovie = currentTheater.getThirdShow();
    } else if (comboBoxSelectTime.getValue().toString().equals("9:00 PM")) {
        currentMovie = currentTheater.getFourthShow();
    } else
        currentMovie = null;

    Label seatReference = new Label();
    String[] seatStats = currentMovie.getSeating();

    for (int i = 0; i < 108; i++) {
        seatReference = seatList.get(i);
        seatReference.setText(seatStats[i]);

        if (seatReference.getText().equals("Reserved"))
            seatReference.setStyle("-fx-background-color: red;");
        else if (seatReference.getText().toString().equals("Open"))
            seatReference.setStyle("-fx-background: white;");
        else if (seatReference.getText().toString().equals("Selected")) {
            seatReference.setText("Open");
            seatReference.setStyle("-fx-background-color: white;");
        }
        seatList.set(i, seatReference);
    }

    currentMovie.setTitle(movieTitleString);
    System.out.println(movieTitleString);
}

```

In the loadSeats() method, first of all, we choose the time and if the time selected we can choose the seats, after choosing the seats the Open text will change into Selected, and color will change into orange if we will press into “Purchase Seats” button the Selected text will change into reserved and color will change into red.

```

private void saveSeatingMap() {

    Label tempLabel = new Label();

    for (int i = 0; i < 108; i++) {
        tempLabel = seatList.get(i);
        currentMovie.setSeat(i, tempLabel.getText());
    }

    currentMovie.setTitle(movieTitleString);
    currentMovie.setID(currentTheaterID);
    System.out.println("SaveSeatingMap() - " + movieTitleString);

    if (comboBoxSelectTime.getValue().toString().equals("9:00 AM")) {
        currentTheater.setFirstShow(currentMovie);
    } else if (comboBoxSelectTime.getValue().toString().equals("10:00 AM")) {
        currentTheater.setSecondShow(currentMovie);
    } else if (comboBoxSelectTime.getValue().toString().equals("5:00 PM")) {
        currentTheater.setThirdShow(currentMovie);
    } else if (comboBoxSelectTime.getValue().toString().equals("9:00 PM")) {
        currentTheater.setFourthShow(currentMovie);
    }

    if (currentTheaterID.equals("imageMovie1"))
        theater1 = currentTheater;
    else if (currentTheaterID.equals("imageMovie2"))
        theater2 = currentTheater;
    else if (currentTheaterID.equals("imageMovie3"))
        theater3 = currentTheater;
    else if (currentTheaterID.equals("imageMovie4"))
        theater4 = currentTheater;
    else if (currentTheaterID.equals("imageMovie5"))
        theater5 = currentTheater;
    else if (currentTheaterID.equals("imageMovie6"))
        theater6 = currentTheater;
}

```

In the saveSeatingMap() method we getting the movie title and print it in the console.

```
Main x
/Library/Java/JavaVirtualMachines/jdk1.8.0_
The Greatest Showman
SaveSeatingMap() - The Greatest Showman
|
```

Database part:

```
public class DatabaseHandler extends Configs {
    Connection dbConnection;

    public Connection getDbConnection() throws ClassNotFoundException, SQLException {
        String connectionString = "jdbc:mysql://" + dbHost + ":"
            + dbPort + "/" + dbName;
        Class.forName("com.mysql.cj.jdbc.Driver");

        dbConnection = DriverManager.getConnection(connectionString, dbUser, dbPass);

        return dbConnection;
    }
}
```

Connection to mysql

```

public void signUpUser(User user) {
    String insert = "INSERT INTO " + Const.USER_TABLE + "(" +
        Const.USER_FIRSTNAME + "," + Const.USER_LASTNAME + "," +
        Const.USER_USERNAME + "," + Const.USER_PASSWORD + ")" +
        "VALUES (?, ?, ?, ?)";

    try {
        PreparedStatement prSt = getDbConnection().prepareStatement(insert);
        prSt.setString( parameterIndex: 1, user.getFirstName());
        prSt.setString( parameterIndex: 2, user.getLastName());
        prSt.setString( parameterIndex: 3, user.getUserName());
        prSt.setString( parameterIndex: 4, user.getPassword());
        prSt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}

```

```

public ResultSet getUser(User user) {
    ResultSet resSet = null;

    String select = "SELECT * FROM " + Const.USER_TABLE + " WHERE " +
        Const.USER_USERNAME + "=? AND " + Const.USER_PASSWORD + "=?";

    try {
        PreparedStatement prSt = getDbConnection().prepareStatement(select);
        prSt.setString( parameterIndex: 1, user.getUserName());
        prSt.setString( parameterIndex: 2, user.getPassword());

        resSet = prSt.executeQuery();
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    return resSet;
}

```