

Report – Movie Ticket Booking System

Course Code: <CODE>

**Document History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
| 1 | 25/09/2020 | Alen V George |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# **TABLE OF CONTENTS**

[**TABLE OF CONTENTS**](#_12xlf8elgb1u)3

[**TABLE OF FIGURES**](#_dwn22sxh6gl1)4

[**LIST OF TABLES**](#_hvoz4cpj7trb)4

[**1. INTRODUCTION**](#_stjvnayv28pk)5

[1.1 Problem Statement](#_euwm9b15qr85) 5

[1.2 Description](#_i4i773ittjoc) 5

[**2. REQUIREMENTS**](#_46nqz8zhpem2)6

[2.1 Software Requirements](#_t47twabno1fz) 6

[2.2 Functional Requirements](#_75iagyrhh40s) 6

[**3. DESIGN**](#_sd6s2j4219a8)7

[3.1 Activity Diagrams](#_t2pmt071hkpo) 7

[3.1.1 User Activity Diagram](#_mgfy3thjifag) 7

[3.1.2 Admin Activity Diagram](#_wur1mz4vsah3) 9

[3.2 Use Case Diagram](#_ha9qkhb4nta9) 10

[**4.TEST PLAN**](#_cdmtd9c8d4y1)11

[4.1 Introduction](#_82l14fqpy) 11

[4.2 Test Strategy](#_5568lrm0tcea) 11

[4.2.1 Scope of Testing](#_l9ttl5wy60e2) 11

[4.2.1.1 Features to be tested](#_18l1boowpr43) 11

[4.2.1.2 Features not to be tested](#_rtmhlmdhbqk1) 11

[4.2.2 Test Type](#_6lc60tdg5d0q) 12

[4.3 Test Objective](#_6dignpfaeu02) 12

[4.4 Test Criteria](#_bhofzegz82sn) 12

[4.4.1 Suspension Criteria](#_2odr0i13pwh1) 12

[4.4.2 Exit Criteria](#_ta189ifvevj2) 12

[4.5 Test Environment](#_rh4lymal62o1) 12

[**5. TEST CASES**](#_xq4dkgh69ral)13

[**6. EXPECTED RESULT**](#_fzcaawyutdx3)14

[**7. CONCLUSION**](#_qtmv5inm6rri)16

# **TABLE OF FIGURES**

Figure 1: User Activity Diagram 7

Figure 2: Admin Activity Diagram 9

Figure 3: Use Case Diagram 10

# **LIST OF TABLES**

Table 1: Functional Requirements 6

Table 2: Features to be tested 11

Table 3: Test Cases 13

Table 4: Expected Results 14

# **1. INTRODUCTION**

## **1.1 Problem Statement**

To develop a system which helps the users to book movie tickets for the desired show without visiting the theatre.

## **1.2 Description**

The purpose of the movie ticket booking system is to provide an alternate way for users to purchase movie tickets. The system offers separate login for users as well as the admin. After logging in, the users can view a list of movies, book tickets, search movies and view all the booked tickets under his/her name. New users need to register in the system in order to book tickets. Admin has the privileges to add movies and to view all bookings.

The project is done using the concept of structures and file handling. All the details entered by the admin and the users are stored in a file and so the details are not lost when the program is terminated. The system hence provides a reliable method for storing and accessing data.

# **2. REQUIREMENTS**

## **2.1 Software Requirements**

The software requirements of the project are:

* MinGW Compiler
* CodeBlocks IDE

## **2.2 Functional Requirements**

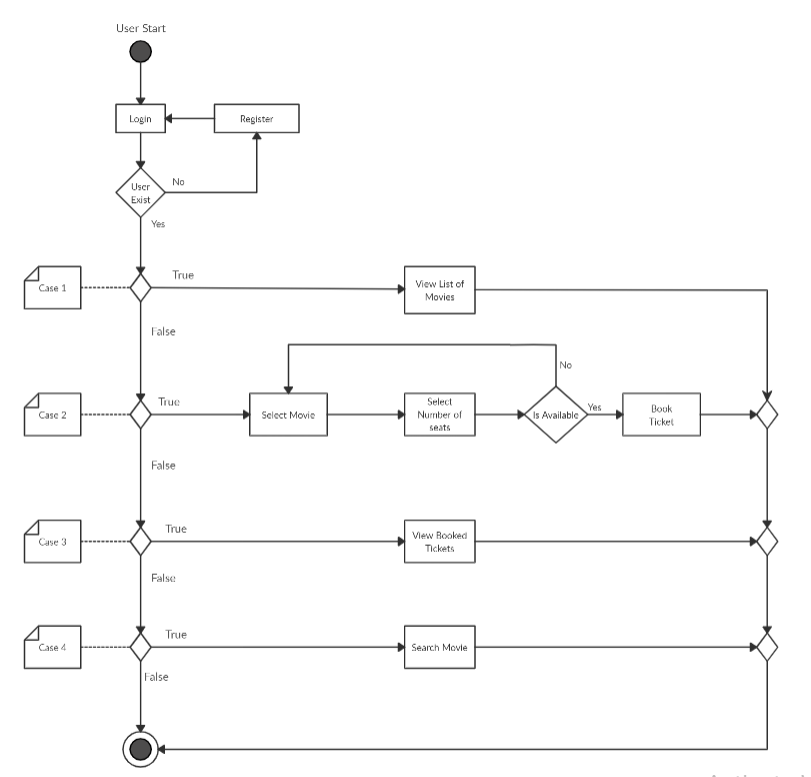
|  |  |
| --- | --- |
| **Functional Requirement Number** | **Description** |
| FR1 | The system shall allow the admin to login |
| FR2 | The admin shall be able to add movie details |
| FR3 | The admin shall be able to see the current reservations |
| FR4 | The system shall allow the registration of new users |
| FR5 | Existing users shall be able to login to the system |
| FR6 | The system shall allow logged in users to book tickets |
| FR7 | The system shall allow users to see the list of movies |
| FR8 | The system shall allow users to see their booked tickets |
| FR9 | The system shall allow user to search movie based on name |

*Table 1: Functional Requirements*

# **3. DESIGN**

## **3.1 Activity Diagrams**

### 3.1.1 User Activity Diagram



*Figure 1: User Activity Diagram*

The movie ticket gives various choices for a user. The functions available for users are:

**Register**

* This function allows users to register by providing their details.
* The system automatically pulls the details when the user books a ticket.

**Login**

* This function allows users to login after registration.
* Error message is given if login is unsuccessful.

**View Movies**

* This function allows users to see the list of available movies in the theatre.

**Book Tickets**

* This function allows users to book tickets for a desired movie.
* The user has to input the movie name and the number of seats to book.
* The system will show a ticket to the user.

**View Ticket**

* This function allows the user to see all the bookings under his/her name.
* The system will return an error message if no tickets are booked.

**Search Movies**

* This function allows users to search for movie details by entering a movie name.
* The system displays the details if the movie is available.
* Error message is shown if not available.

### 3.1.2 Admin Activity Diagram

### 

*Figure 2: Admin Activity Diagram*

The functions available for admin are:

**Login**

* The function allows admin to login with predefined username and password.
* The function returns an error message if invalid details are entered.

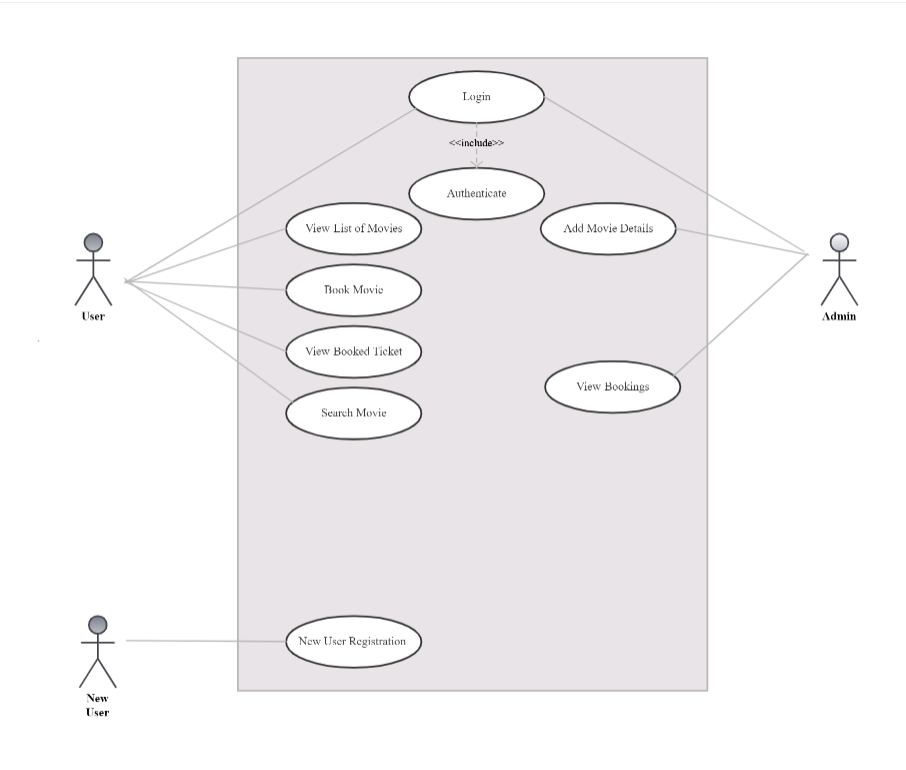
**Add Movie**

* This function allows admin to add details of the movies in the theatre.

**View Bookings**

* This function allows the admin to view all the current bookings.
* Error message is shown if there are no bookings.

## **3.2 Use Case Diagram**



*Figure 3: Use Case Diagram*

# **4. TEST PLAN**

## **4.1 Introduction**

The test plan is designed to identify the items to be tested, the features to be tested, type of testing to be done and to define the test criteria for the movie ticket booking system.

## **4.2 Test Strategy**

### 4.2.1 Scope of Testing

#### 4.2.1.1 Features to be tested

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Applicable Role** | **Description** |
| Register | User | A new user can register to system by providing details |
| Login | User  Admin | **User:** Existing users can login to the system  **Admin:** Admin can login to the system |
| Add Movie | Admin | The admin can add the list of movies |
| View Bookings | Admin | The admin can view the list of bookings |
| Movie List | User | User can see the list of movies |
| Book Ticket | User | User can book tickets for desired movie |
| View Ticket | User | User can view booked tickets |
| Search Movies | User | User can search for movies |

*Table 2: Features to be tested*

#### 4.2.1.2 Features not to be tested

The features that are not to be tested are:

* User Interfaces
* Hardware interfaces

### 4.2.2 Test Type

Three types of testing are to be done for the movie booking system:

1. **Unit Testing**: In this testing each individual module of the system is tested to ensure that they are fit to use.
2. **Integration Testing:** Here the modules are integrated together and tested as a group.
3. **System Testing:** Conducted on a complete, integrated system to evaluate systems compliance with its specific requirements.

## **4.3 Test Objective**

The test objectives are to verify the functionality of the movie ticket booking system, the focus should be on testing the functionalities such as Add Movie, Book Ticket, View Ticket..etc to guarantee that these operations can work normally.

## **4.4 Test Criteria**

### 4.4.1 Suspension Criteria

If more than 40% of the test cases fail, the test is suspended until all the failed cases are fixed.

### 4.4.2 Exit Criteria

Specifies the criteria that denotes a successful completion of a test phase

* Run rate is mandatory to be 100% until a clear reason is specified
* Pass rate is 90%

## **4.5 Test Environment**

All the tests are carried out in the CodeBlocks IDE.

# 

# **5. TEST CASES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** |
| TU01 | Check admin login with valid data | 1. Enter Username  2. Enter Password | Username= admin  Password= admin24 |
| TU02 | Check admin login with invalid data | 1. Enter Username  2. Enter Password | Username=admin  Password= admin123 |
| TU03 | Check new user registration | 1. Enter username  2. Enter password  3. Enter name  4. Enter Mobile Number | Username = alen  Password = alen123  Name = Alen V George  Mobile Number =9388094648 |
| TU04 | Check user login with valid data | 1. Enter username  2. Enter password | Username=alen  Password=alen123 |
| TU05 | Check user login with invalid data | 1. Enter username  2. Enter password | Username = alen  Password=alen234 |
| TU06 | Check if movie details can be added by admin | 1. Enter choice in admin panel  2. Enter movie code  2. Enter movie name  3. Enter show timing  4. Enter release date  5. Enter ticket price  6. Enter number of seats | Choice =1  Movie code= 1  Movie name = “onward”  Movie show timing= 7:45 PM  Movie release date= 24 September, 2020  Ticket price=200  Seats=60 |
| TU07 | Check if all reservations  can be seen by admin | 1. Enter Choice in admin panel | Choice = 2 |
| TU08 | Check if user can view list of movies | 1.Enter Choice in user panel | Choice =1 |
| TU09 | Check if the user can book a ticket with valid movie name | 1. Enter choice in user panel  3. Enter movie name  4. Enter number of seats | Choice = 2  Movie name = “onward”  Number of seats = 3 |
| TU10 | Check if the user can book a ticket with invalid movie name | 1. Enter choice in user panel  3. Enter movie name  4. Enter number of seats | Choice =2  Movie name = “avengers”  Number of seats = 3 |
| TU11 | Check if the user can book a ticket with valid number of seats | 1. Enter choice in user panel  3. Enter movie name  4. Enter number of seats | Choice = 2  Movie name = “onward”  Number of seats = 3 |
| TU12 | Check if the user can book a ticket with invalid number of seats | 1. Enter choice in user panel  3. Enter movie name  4. Enter number of seats | Choice = 2  Movie name = “onward”  Number of seats = 61 |
| TU13 | Check if user can view  booked tickets if user has already booked the ticket | 1. Enter Choice | Choice = 3 |
| TU14 | Check if user can view  booked tickets if user has not booked the ticket | 1. Enter Choice | Choice = 3 |
| TU15 | Check if user can search for movies with valid movie name | 1. Enter choice in user panel  2. Enter the movie name to search | Choice = 4  Movie name=”onward” |
| TU16 | Check if user can search for movies with invalid movie name | 1. Enter choice in user panel  2. Enter the movie name to search | Choice = 4  Movie name = “avengers” |

*Table 3: Test Cases*

# **6. EXPECTED RESULT**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Expected Results** |
| TU01 | Check admin login with valid data | Login should be successful |
| TU02 | Check admin login with invalid data | Admin should not be able to login and error message should be displayed |
| TU03 | Check new user registration | User should be able to register |
| TU04 | Check user login with valid data | User should be able to login |
| TU05 | Check user login with invalid data | User should not be able to login and error message should be displayed |
| TU06 | Check if movie details can be added by admin | Admin should be able to add movie details |
| TU07 | Check if all reservations  can be seen by admin | Admin should be able to see all reservations |
| TU08 | Check if user can view list of movies | Users should be able to see the list of movies |
| TU09 | Check if the user can book a ticket with valid movie name | User should be able to book ticket |
| TU10 | Check if the user can book a ticket with invalid movie name | User should not be able to book ticket and an error message should be shown |
| TU11 | Check if the user can book a ticket with valid number of seats | User should be able to book ticket |
| TU12 | Check if the user can book a ticket with invalid number of seats | User should not be able to book ticket and an error message should be shown |
| TU13 | Check if user can view  booked tickets if user has already booked the ticket | Users should be able to view booked ticket under his/her username |
| TU14 | Check if user can view  booked tickets if user has not booked the ticket | Error message should be displayed |
| TU15 | Check if user can search for movies with valid movie name | The details of the searched movie should be displayed |
| TU16 | Check if user can search for movies with invalid movie name | Error message should be shown |

*Table 4: Expected Results*

# 

# 

# 

# **7. CONCLUSION**

The movie ticket booking system is a simple but efficient way to book and manage tickets. The system offers a limited amount of functions at the given time but can be expanded to include more functionalities. The system helps in reducing long queues at movie theatres and easy management of tickets. The system is very user friendly, helping users to book tickets with ease.