**MOVIE TICKET BOOKING MANAGEMENT SYSTEM USING C**

**Name : Sriram R M**

**SFid : 104679**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SI.NO** | **Topic** | **Page number** |
| 1 | Problem statement | 3 |
| 2 | Project description | 3 |
| 3 | Activity diagram | 4 |
| 4 | Requirements gathering | 5 |
| 5 | Test plan | 6 |
| 6 | Test cases | 7 |
| 7 | Expected results | 9 |

**PROBLEM STATEMENT**

To manage movie ticket booking using software and handle all the related functions. To store all the records of transactions in a file and handle processes from the administrator side.

**PROJECT DESCRIPTION**

The software provides various functions for movie ticket management.

**Module 1 :** The admin and the user are provided with separate log-in modules. The admin/user shall login first by providing a username and a password. The software validates the password and if it is valid, the admin/user is logged in. Otherwise an error message is generated. The admin/user can try up to several times before the program terminates.

**Module 2 :** Next, an option is provided for adding a new movie to the system. This is done using the concept of file handling in C. After choosing this option, the details of the movie are to be filled and the movie gets added to the list of movies.

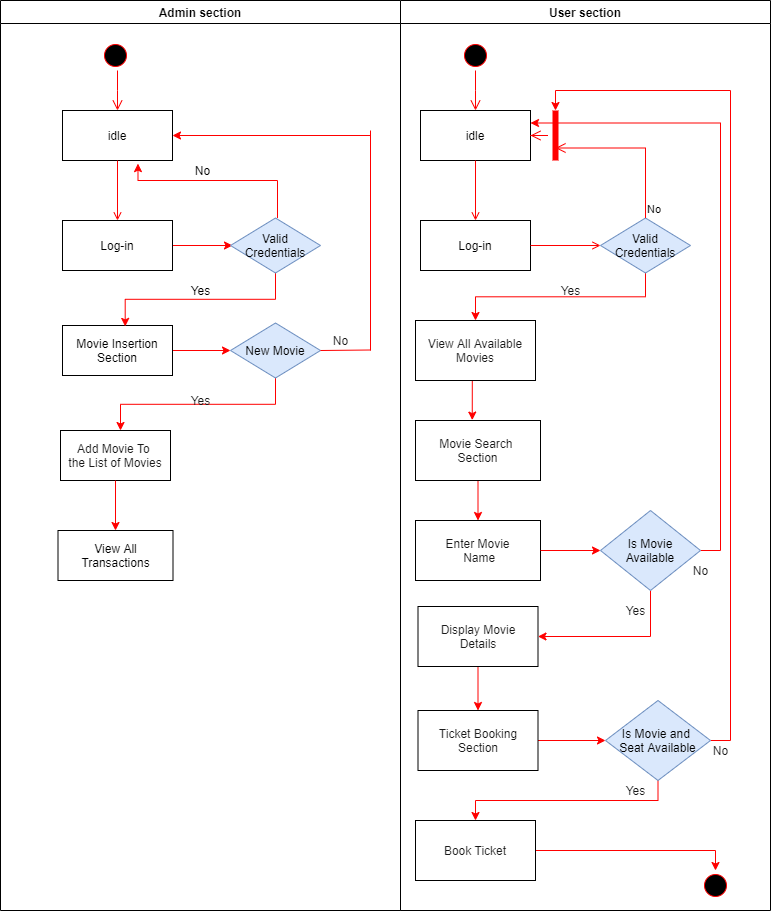
**Module 3 :** Second, an option is available to view the list of movies. After choosing this option, the list of movies gets displayed. Next, the user can search for a particular movie using the third option. This is done by providing the movie name. If the movie is available, the details of the movie gets displayed.

**Module 4 :** Next, a module is provided to book tickets. The user is asked about the movie name. After choosing a movie, details like name, contact number, total number of tickets and the seat numbers are to be filled. If a particular seat number is unavailable, the user is asked to re-enter the seat number. After filling the necessary details, the record gets stored and the ticket booking is done.

**Module 5 :** This module deals with handling all the booked tickets. List of reserved tickets are displayed.

Finally, the admin/user can log out of the system.

**ACTIVITY DIAGRAM :**

****

**REQUIREMENTS GATHERING**

**HARDWARE REQUIREMENTS:**

A PC/laptop with windows installed in it.

Core : i3 or higher

**SOFTWARE REQUIREMENTS:**

IDE : CodeBlocks 20.3

Compiler : GNU GCC Compiler (in-built with CodeBlocks)

**TEST PLAN**

A test plan is devised to identify and rectify the errors. The scope of the test plan should cover each module and validate their working. The following are the steps to be followed while designing the test plan

**Step 1** : Test cases are generated for each module. The test case should cover all the functions provided in the module.

**Step 2** : The individual modules are checked against the test cases to ensure proper working. If not, then proper corrections are made.

**Step 3** : The modules are integrated according to the interrelation between them and checked as a whole. Thus, groups of integrated modules are checked in this step.

**Step 4** : Finally, all the modules are integrated and the software as a whole is validated against the test cases in this phase of the test plan.

The following shows the modules and their respective features to be tested.

|  |  |  |
| --- | --- | --- |
| **SI NO** | **MODULE** | **FEATURES TO BE TESTED** |
| 1 | Log-in module | Log-in of user/admin after providing a valid password |
| 2 | Movie insertion module | Inserting a new movie to the list of movies after providing all the details regarding the movie |
| 3 | Movie view module | To view all the available movies and release date, price etc., |
| 4 | Movie ticket booking module | Booking tickets and viewing the booked tickets |
| 5 | Ticket record module | Viewing all the booked tickets |

**TEST CASES**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI.NO** | **Test Module** | **Test Case** | **Expected Outcome** |
| 1 | Log-in module  (For user) | User name : user01  Password : userpw | Log-in successful |
| 2 | Log-in module  (For admin) | Admin name : sriram  Password : admin | Log-in successful |
| 3 | Log-in module  (For user) | User name : user01  Password : wrong | Log-in failed |
| 4 | Log-in module  (For admin) | Admin name : sriram  Password : wrong | Log-in failed |
| 5 | Movie insertion module | Movie name : Superman  Release date : 26-Nov-2020  Cost : 350 | Movie appended to the list |
| 6 | Movie search module(searching available movie) | Movie Name : Superman | Details of the movie Spiderman |
| 7 | Movie search module(searching unavailable movie) | Movie Name : Spiderman | Movie not available |
| 8 | Movie view module | Selecting this module | List of available movies are displayed |
| 9 | Movie booking module(Booking available movie) | Movie name : Superman  Total No.of Tickets : 2  Select your seats : 3, 4 | Enjoy the movie  Movie : Superman  Total No.of Tickets : 2  Selected Seats : 3 4  Cost per ticket : 350  Total cost : 700  Record is inserted to the transaction file |
| 10 | Movie booking module(Booking unavailable movie) | Movie Name : Spiderman | Movie not available |
| 11 | Movie booking module(Booking unavailable seats) | Movie Name : Superman  Total No.of Tickets : 2  Select your seats : 3,4 | Seats not available |
| 12 | Movie booking module(Booking available seats) | Movie name : Superman  Total No.of Tickets : 2  Select your seats : 6,8 | Enjoy the movie  Movie : Superman  Total No.of Tickets : 2  Selected Seats : 6 8  Cost per ticket : 350  Total cost : 700  Record is inserted to the transaction file |
| 13 | Ticket record module | Selecting this option | List of booked tickets are displayed |

**EXPECTED RESULTS**

The system should function properly when provided with valid inputs. Otherwise, error messages should be displayed. The system should be tested against a large number of test cases and show consistency in all the tests.