# OBJECT ORIENTED ANALYSIS AND DESIGN WITH SOFTWARE ENGINEERING LABORATORY

## PROJECT REPORT

## ON

# MOVIE TICKET BOOKING SYSTEM


## ABSTRACT

Watching movies with family and friends has always been everybody's favourite hobby. This project is aimed at developing a movie ticket booking system for a Cinema Hall. The movie ticket booking system is an application that can be accessed throughout by all its users. This application will automate the reservation of movie tickets.

This ticket reservation system provides a command prompt for a cinema hall where any user can access it from the place where they are. The existing system has two ways of booking tickets for a movie, one is to book tickets at the ticket counter of the respective cinema hall and the other one is through computers with any kind of operating systems like windows/Linux/MacOs. Former is one of the hectic processes where one should stand in long queues for hours.All this becomes a highly tedious job.Automating everything remains our primary concern which led to the development of this website.This has both admin and user privileges giving more personalized experience.

**JANUARY – MAY 2021**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**EC CAMPUS,**

**BENGALURU – 560100, KARNATAKA, INDIA**

# TABLE OF CONTENTS

# SOFTWARE REQUIREMENTS SPECIFICATION

## 1. Introduction

### 1.1 Purpose

Product: Movies Ticket Booking System.

The purpose of this document is to build a system for managing the ticket bookings of Movies. The main purpose of this project is to provide a reliable , secure, efficient and user friendly environment to the customers and management authorities. Also benefits the customer with efficient and faster service. This project can save money and efforts in managing the record, just a mouse click can make the task easier and faster. This product solves the problem of finding out the available tickets for a Movie, without physically going to the venue.

### 1.2 Intended Audience

This project is a prototype for the Ticket Booking System and it is restricted within the college premises. This is being implemented under the guidance of college professors. This project is useful for the ticket booking team and as well as to their customers. Starting from the overview for all readers, the developers can go on to focus on the system features and the other non functional requirements. The users can look into the interfaces such as the user interface section and the software/ hardware interface.

### 1.3 Product Scope

The purpose of the Ticket Booking System is to ease ticket management and to create a convenient and easy-to-use application for customers, trying to buy tickets for various events. The system is based on a database with its ticket management functions. We will have a database file supporting ticketing information for multiple event organizers. Above all, we hope to provide a comfortable user experience along with the best pricing available**.**

### 1.4 References

https://krazytech.com/projects

https://dokumen.tips/download/link/online-movie

https://www.quora.com/What-are-the-functional-and-non-functional-requirements-for-online-movie -ticket-booking

https://www.researchgate.net/publication/342466860_A_PROJECT_ON_ONLINE_TICKET_BOO KING_SYSTEM

# 2. Overall Description

## 2.1 Product Perspective

This product is a prototype of the Ticket Booking System and is inspired by an already existing system known as BookMyShow. The below Block diagram shows the overall functionality of the online movie ticket booking system.

## 2.2 Product Functions

1. User functions:

- Searching for a particular movie.
- Booking a ticket for a movie.
- Cancellation of a movie.
- View the details of a movie.
- Allowing the event organizers to update the information about their movie.
- Allowing the administrative users to modify the details of event organizers in the database.
- Users should be able to select their seat as per their choice in the hall.
- User should have different options of payment.

## 2.3 User Classes and Characteristics

Classes of Users:

- Customers:
    - Allowed to book tickets
    - See the details of the movie
    - select the available seat types
- Manager/admin:
    - Decides the movie to be screened

- Number of shows presented in a day
- Timings of the movie
- Also reserve seats for the tickets booked online
- Can add  movies to the show halls

### 2.4 Operating Environment

The software will work under these environments:

- Operating system: Windows/Linux/MacOS2
- Platform: C++
- Any version is applicable.

### 2.5 Design and Implementation Constraints

- Regulatory policies: It is mandatory that the input data must be one among the given options.
- Control function: The software must be very user-friendly and display appropriate error messages.
- Reliability Requirements: Data redundancy and use of special/blank characters must be avoided.
- Higher Order Language Requirements: The server application should be available 365 days 24x7.
- Security and Safety Issues: Username, password both are considered as security issues.

### 2.6 Assumptions and Dependencies

- The system will assume that if a person is booking multiple tickets, then he/she is selecting the events which are organized at separate time intervals

## 3. External Interface Requirements

### 3.1 User Interfaces

command prompt.
User Interface Steps :

- Customer
    1: Sign In
    2: Login
    3: Choose Movie from list

4: Detail of the movie
5: Ticket Booking
6: Payment

- Manager/admin
  1.Login
  2.Add movies to the show halls
  3.Add movie details
  4.Block the seats for users

## 3.2 Software Interfaces

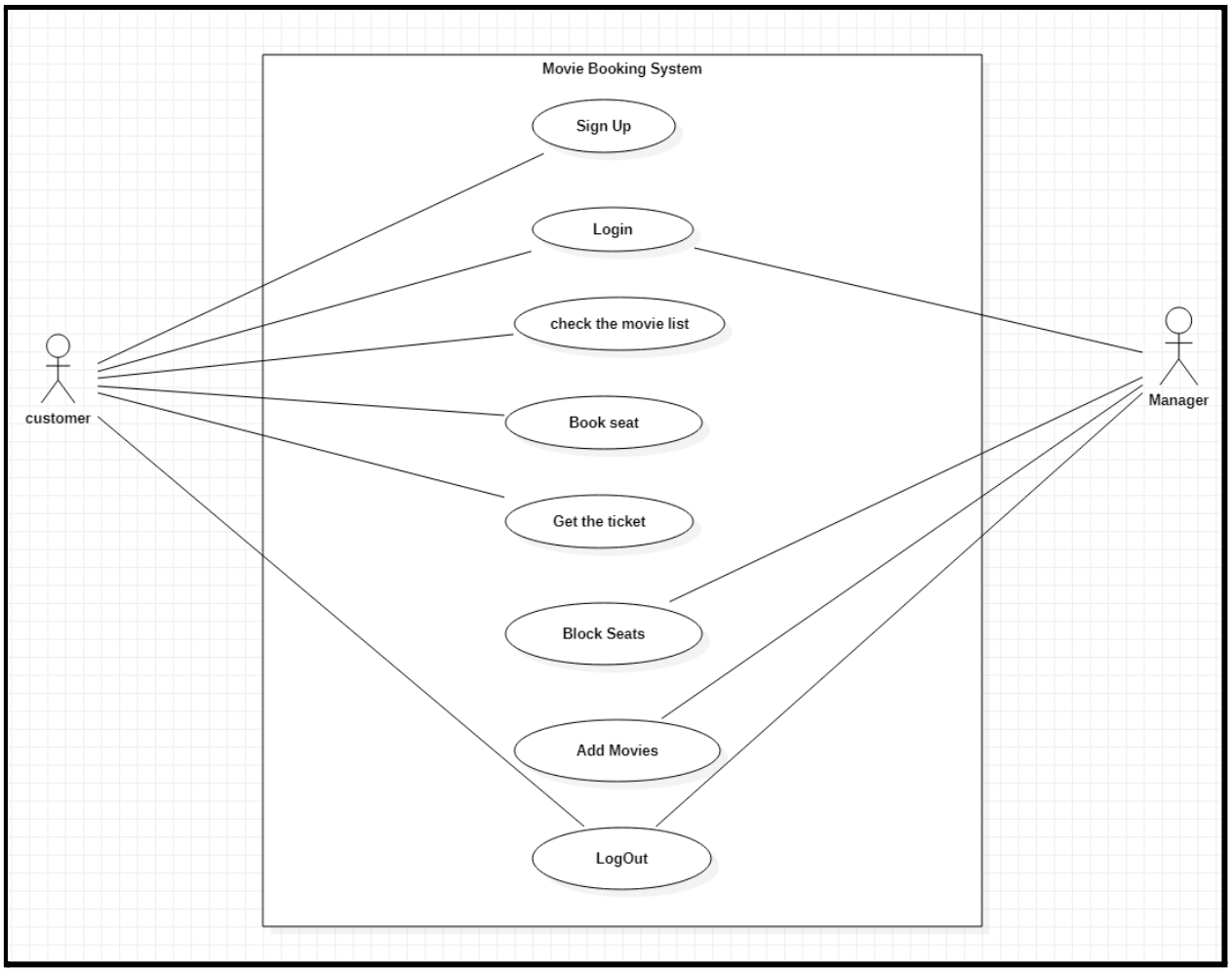
- Operating system - Windows/Linux/MacOS for their best support and user friendliness
- To implement the project we have chosen C++ language for it's OOPS concept

## 3.3 Communications Interfaces

1. The system should have installed C++ application to run the software.

# 4. Analysis Models

## Use Case Diagram




# 5. System Features

In this section we are going to discuss the System Features of the application based on the use case diagram of the application as shown above.

## 5.1 Booking Tickets

### 5.1.1 Description and Priority

This feature is for all the customers who want to book tickets for various events.(Can also be accessed by administrators)
Priority: HIGH

### 5.1.2 Stimulus/Response Sequences

As soon as the customer submits the information form for booking a ticket, the function will update the back-end database.

#### 5.1.3 Functional Requirements

REQ-1: Number of seats booked should be less than the total number of seats.

REQ-2: The seats to be booked should not belong to any other user.

REQ-3: The event to be booked should be scheduled for sometime in the future.

REQ-4: The seats booked should not be available for other users.

### 1.5 Adding a new movie:

**5.4.1** Description and Priority This feature available for the Manager only to add a new event to the database.

Priority: HIGH

**5.4.2** Stimulus/Response Sequences As soon as the event organizer submits the information form for adding an event, the function will update the back-end database.

### 1.6 Login and Sign up:

**5.7.1** Description and Priority This feature is for all types of consumers of the product. The consumers have to sign up as user, administrator, organiser and so on, so that they have the privileges of the respective consumer type.

Priority: HIGH

**5.7.2** Stimulus/Response Sequences Whenever a user logs in or signs up, the details have to be logged/updated to the back-end database

5.8.3 Functional Requirements

REQ-1: The consumer must specify the type of user.

REQ-2: One user should have only one account.

REQ-3: Users should be able to delete the account as per their will.

# 6. Other Nonfunctional Requirements

## 6.1 Performance Requirements

1. The system must be able to perform smoothly without any interruptions.
2. There should be minimum interruptions in connectivity.
3. The performance shall depend upon hardware components of the client/customer
4. The back-end server must be able to handle the amount of requests for no-lag interaction with the product.

5. The database design should have minimum latency.

## 6.2 Safety Requirements

If there is extensive damage to a wide portion of the database file due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

## 6.3 Security Requirements

1:No one should have access to the back-end database other than the application and the administrative users. 2:This system is completely secured, as each user will Require an authenticated user id and password. 3:Sensitive data will be encrypted before being sent over insecure connections like the internet. 4: Admin must not disclose passwords to anybody.

## 6.4 Software Quality Attributes

1. Maintainability: The administrators should maintain correct movie timings.
2. Usability: The number of bookings should satisfy the total number of seats available.
3. Availability: The event should be scheduled on the specified date and time.
4. Portability: These applications will be easily portable on a Window based System that has SQL installed.

## 6.5 Business Rules

1. Customers: Booking, Searching, etc.
2. Administrative users: They can perform operations to manage the database like adding the movies along with their details.

# PROJECT PLAN

## 1:The lifecycle to be followed to execute our project is as followed:

**ITERATIVE MODEL:**We can make use of the iterative model for the implementation of our project "MOVIE BOOKING SYSTEM"**.**

1: Some working functionality examples Booking tickets can be developed quickly and early in the life cycle.

2: Parallel development can be planned.

3: Testing and debugging during smaller iterations is easy.

4:It supports changing requirements.

5:Easier to manage risk - High risk part is done first.

6:Risks are identified and resolved during iteration; and each iteration is an easily managed milestone.

7:During the life cycle, software is produced early which facilitates customer evaluation and feedback.

8: Initial operating time is less.

## 2: The tools that we want to use throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

**Version Control**: Version control systems are software tools that help software teams manage changes to source code over time.

For our project we will be using **Github.**

**Development Tool:**

IDE: VS CODE

Source Control : Github

**Design Tool:**

StarUML for Use case model.

Creately for Work breakdown Structure.

**Bug tracking tool:**

VS CODE Debugger extension.

JIRA

**Planning Tool:**

Officetimeline for Gantt Chart

**Testing Tools:**

Selenium using Java(Database Testing)

## 3: Determine all the deliverables and categorise them as reuse/build components and justify the same.

### 3.1 Project Management

1. Project Charter : Reuse

   Since the project scope is similar to that of BookMyShow, the project charter is reusable deliverable.

2. Management Plan : Build

   The Management plan is different from other apps to provide ease of use.

### 3.2 Initiation

1. Business Case : Build

   Based on the scope of the project, a requirements document has been built.

2. Project Planning : Build

   Given the timeline and resources, a Project Plan has been made.

### 3.3 Requirement Analysis

1. Requirement gathering : Reuse

   Customer Requirements taken into consideration during the project build of BookMyShow are inherited by this project.

2. Requirement Analysis : Reuse

   The gathered requirements are analyzed for feasibility and are similar to that of BookMyShow.

3. Security Planning : Reuse

   Basic security in terms of data integrity is maintained.

### 3.4 Design

1. High Level Design : Build

A basic understanding of the features provided by the application is documented.

2. Proof of Concept : Reuse

BookMyShow serves as a proof of concept as it has been implemented and is feasible.

3. Detailed Design : Build

Detailed explanation of the project features are documented including it's action diagrams and use case diagrams.

### 3.5 Development

1. Front end : Reuse

Front end design can be inherited from the base design of BookMyShow.

2. Backend : Build

The database is built from scratch based on the scope of the application.

3. Integration Plan : Build

The frontend and backend will be combined using necessary frameworks.

### 3.6 Testing

1. Testing Tools : Reuse

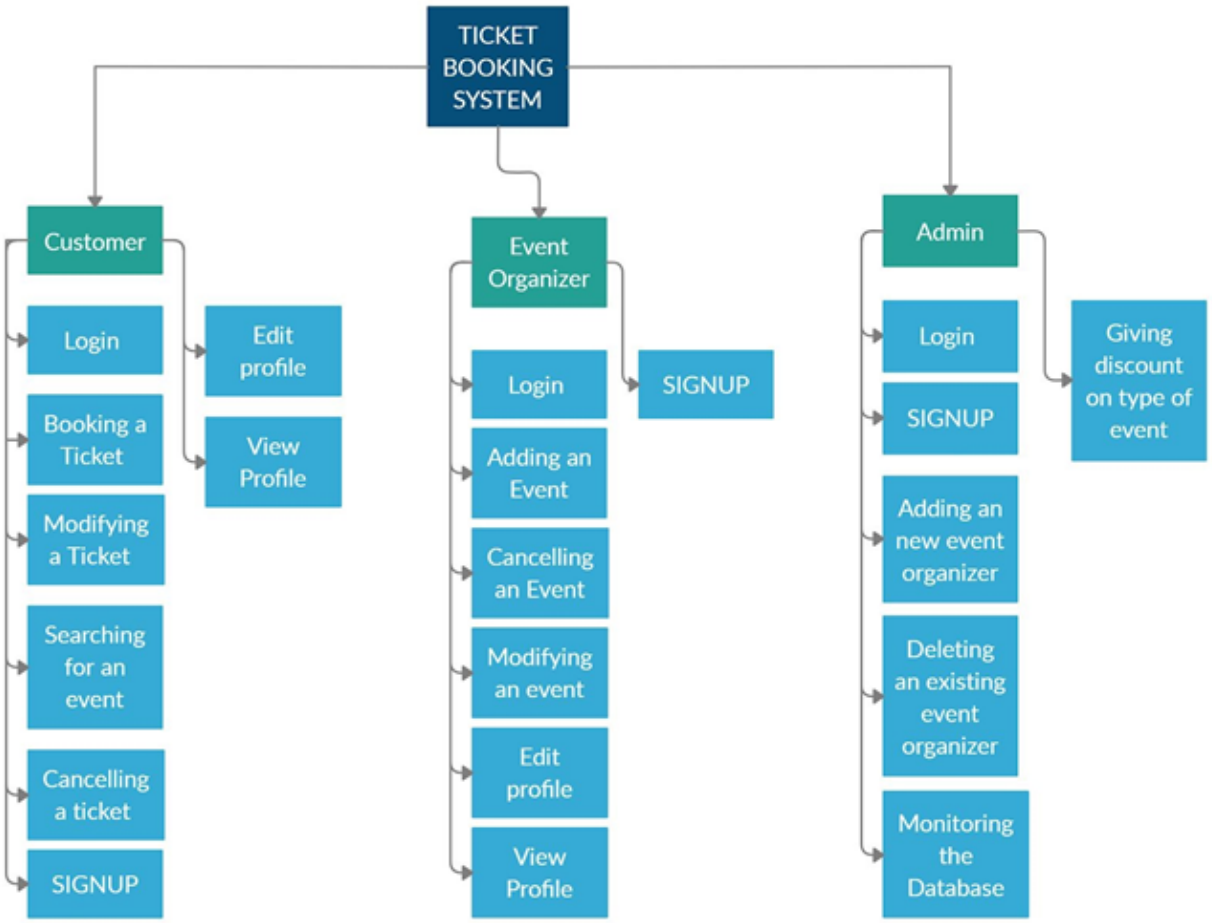
Open source and free to use testing tools will be used for testing the application and the database.

2. Test Report : Build

The final test report will be made based on the output produced by the final application

developed.

## 4: Create a WBS for the entire functionalities in detail.




## 5: Do a rough estimate of effort required to accomplish each task in terms of months.

**Cocomo (Constructive Cost Model)** is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

Our project falls under the **Organic Model.**
A software project is said to be an organic type if the team size required is small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

Formula:

Effort=a*(KLOC)^b

Time=c*(Effort)^d

For our project we will  assume Kilo lines of code = **1.5**

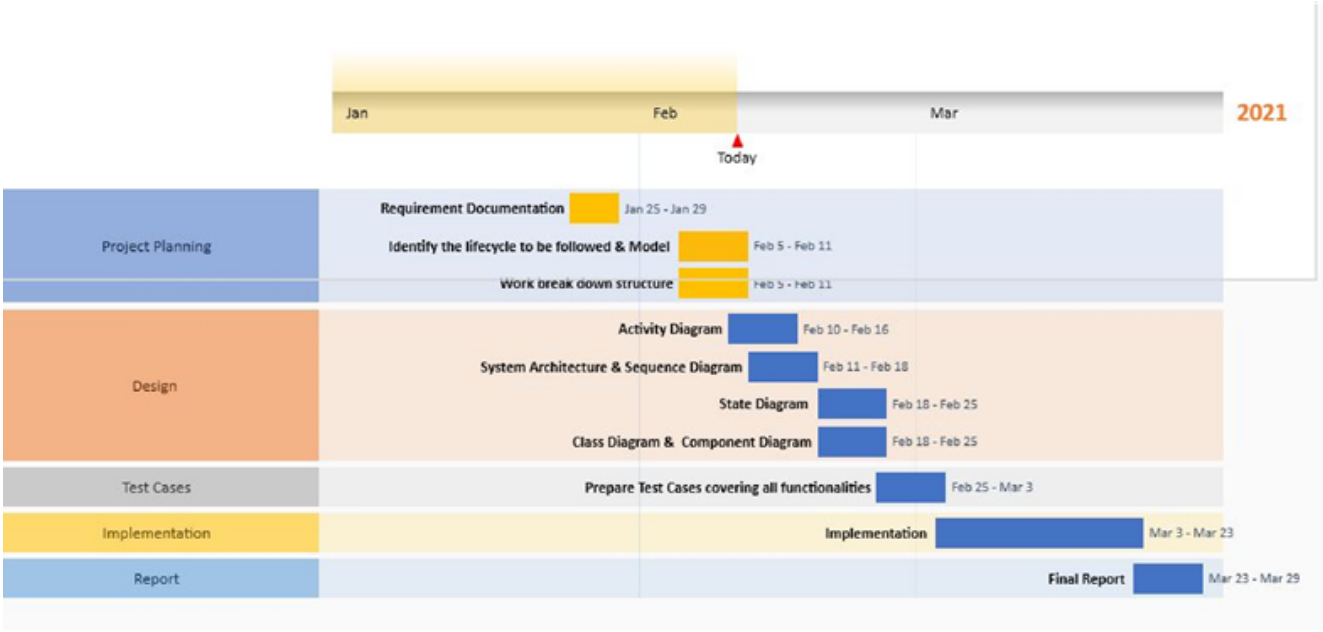
After substituting

For basic COCOMO model  **a=2.4 b=1.05 C=2.5 d=0.38**

**Effort =3.673 Person-Months.**

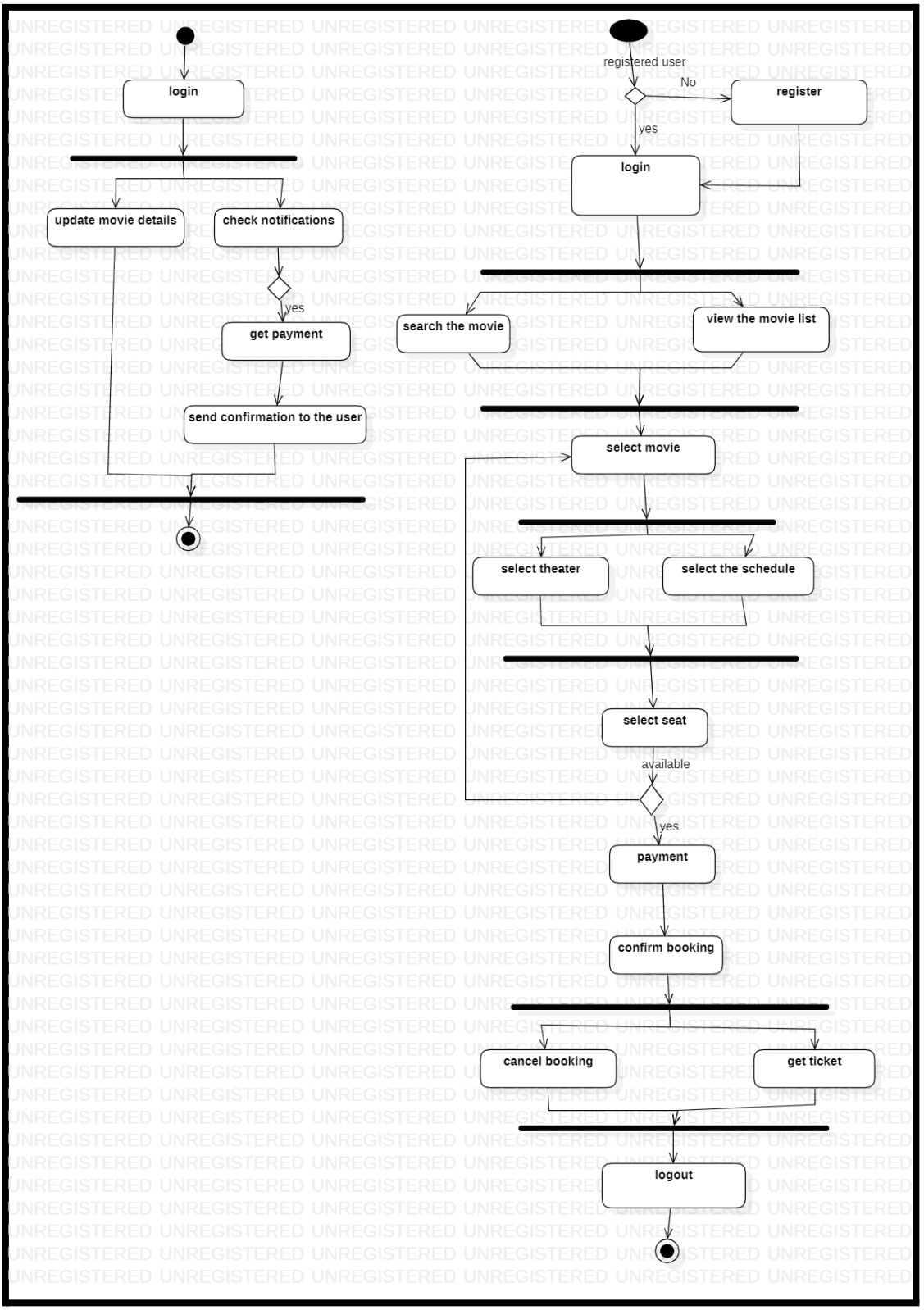
**Development Time=4.098 Months**

# 6: Create the Gantt Chart for scheduling.

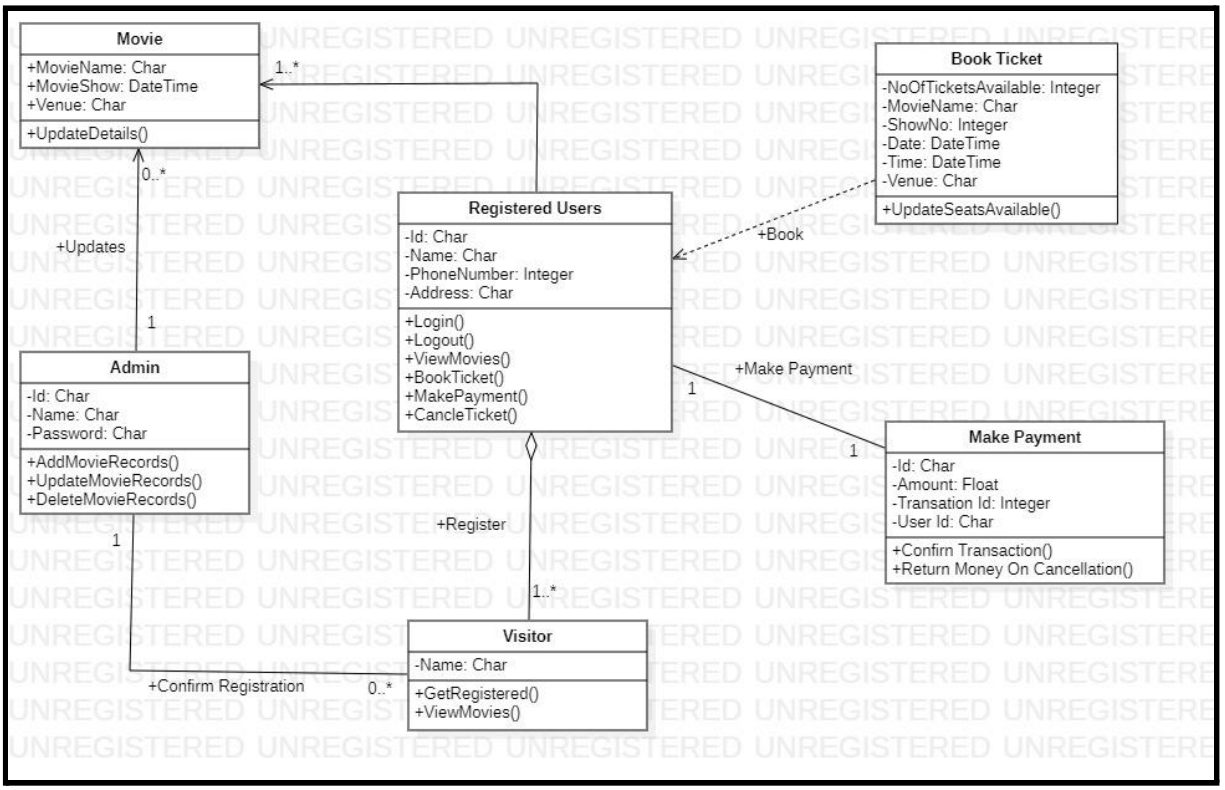


# 7. DESIGN DIAGRAMS

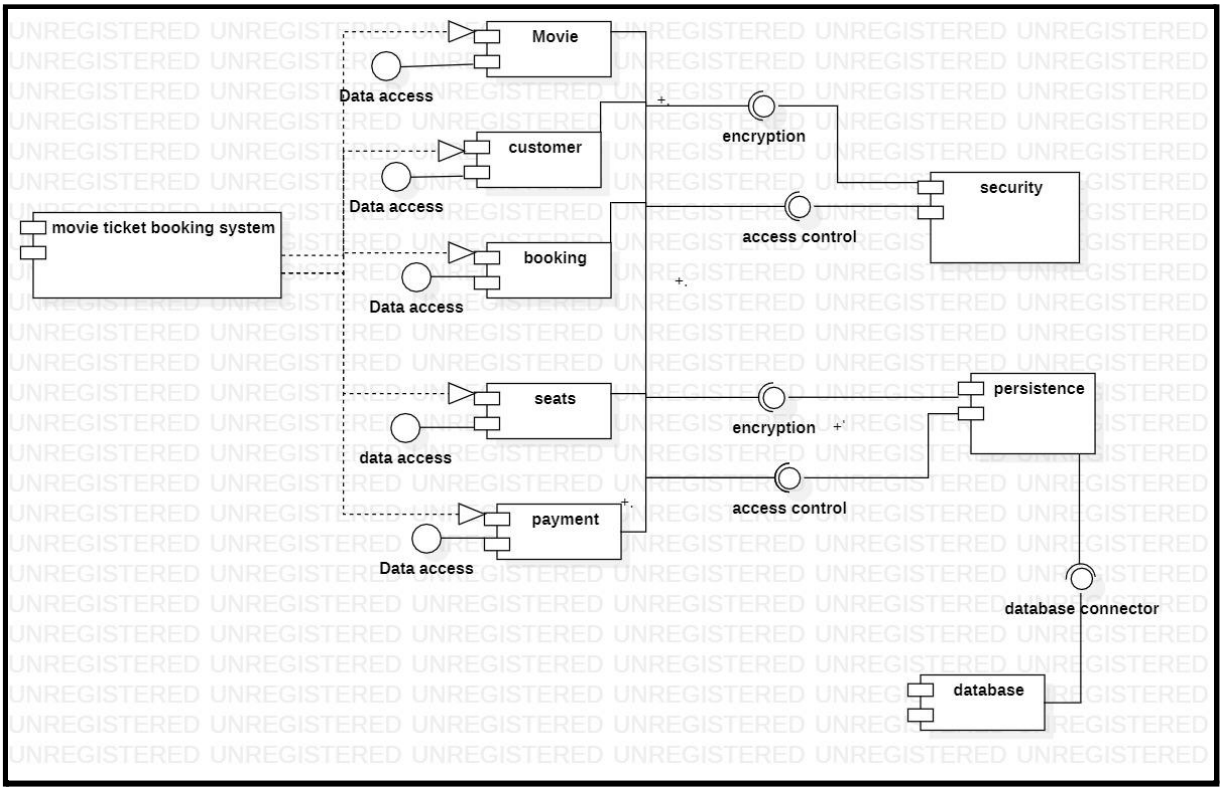
## 1. Activity diagram



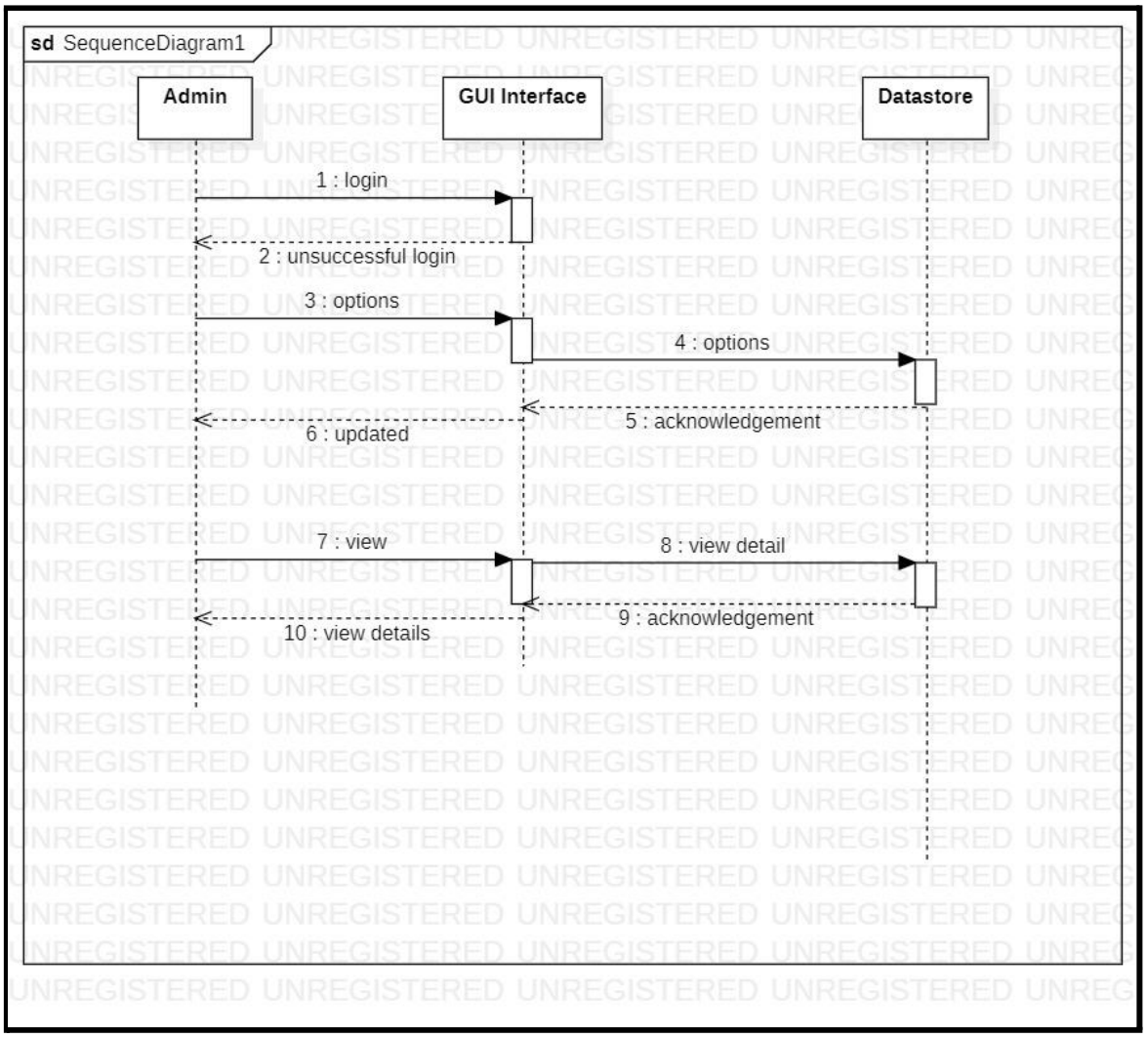
## 2. Class Diagram



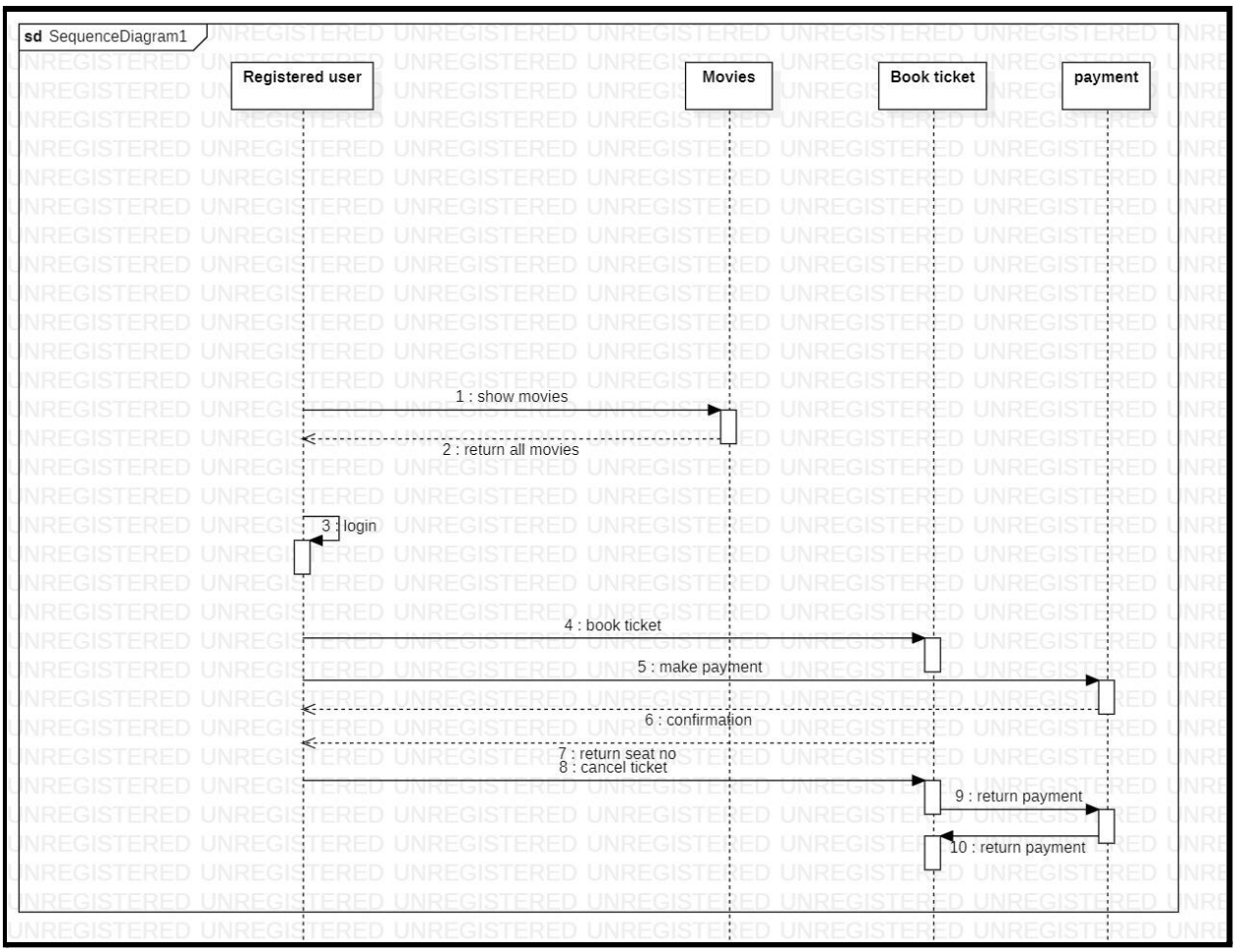

## 3. Component diagram



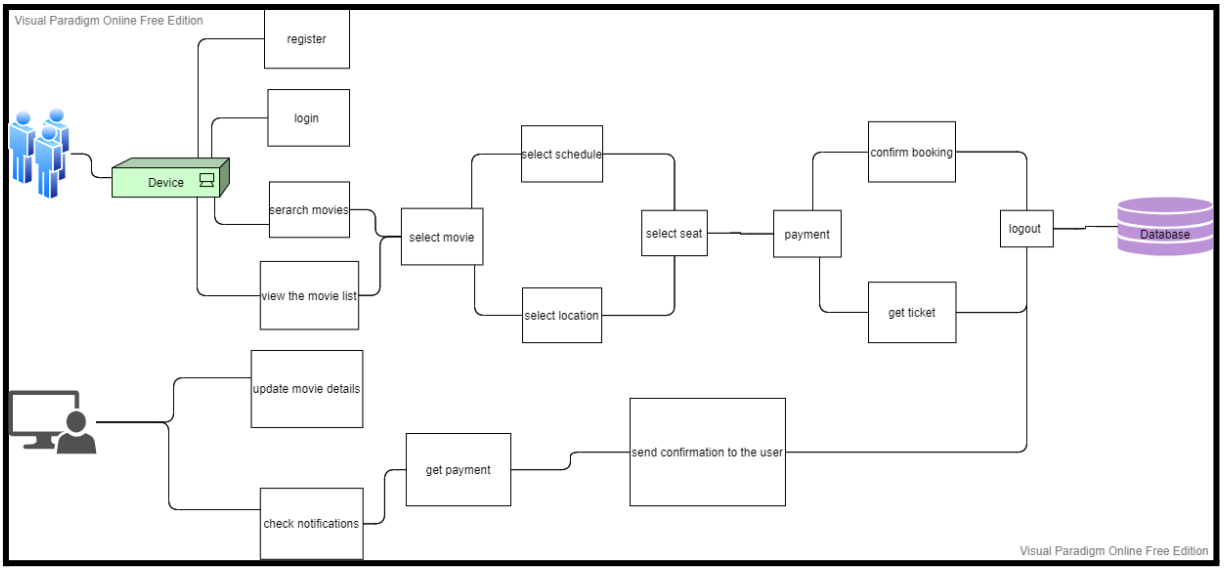
## 4. Admin Sequence Diagram



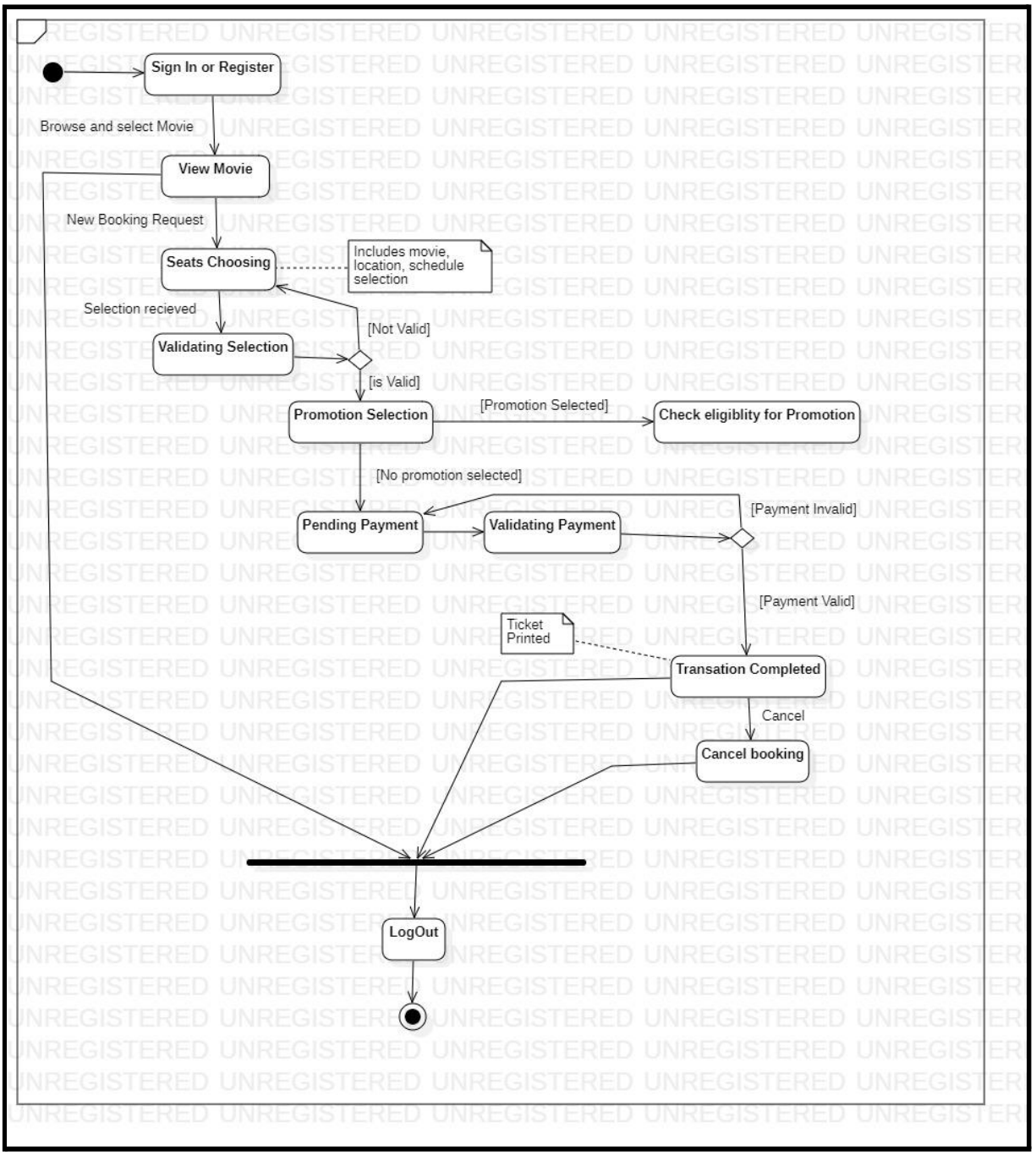

## 5. Customer sequence diagram




## 6. System architecture



# 7. State Diagram



# MODULE DESCRIPTION

## USE CASE 1: EVENT LISTING AND SELECTION (Movie)

Firstly after selecting one of the theaters the customer has to either SignIn or LogIn to get into this module. In this module a set of 13 movies is displayed and the customer has to select one from it. After selecting the movie he/she has to select one among the given time slots. After the time slot the user is directed to the next module to select the seats they want to book.

## USE CASE 2:SEAT SELECTION AND BOOKING

This module allows us to block the required seat.Once a particular seat is reserved it will be marked as X in the seat matrix hence not letting others to book that particular seat.After booking the seat the ticket is displayed along with the cost.

## USE CASE 3: EVENT MANAGEMENT

the manager has to login to the system using the login credentials given by the company then will be able to add the movies running in a particular show hall. Then the admin will be able to block the seats for the customer.

# TEST CASES

## USE CASE 1: EVENT LISTING AND SELECTION (Movie)

| Test Case_ID | Name Of The Module | Test Case Description | Pre-Conditions | Test Steps | Test Data | Expected Result | Actual Result | Test Result |
|---|---|---|---|---|---|---|---|---|
| UT_01 | Selection of Cinema theater | To test the functionality of cinema theatre selection. | Should be able to access any browser and theatre searched should be in the database | 1.Select the movie theatre where you want to book your ticket<br>2. Enter any key to start booking as customer<br>3. Start selecting<br>4. exit | Pressing any key other than 1-4 will exit the program. | Exit the program on pressing any other key. | The program is aborted. | PASS |
| UT_02 | Selection of Cinema theater | To test the functionality of cinema theatre | Should be able to access any browser and theatre | 1.Select the movie theatre where you want to book your ticket | Pressing a key from 1-4 will let you | After successful selection of the theater, | Selection successfully,disp | PASS |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | selection. | searched should be in the database | 2. Enter any key to start booking as customer<br>3. Start selecting<br>4. exit | book ticket in respective theatres | you should ask if the user is a manager or customer. | lays login options for manager or customer. | |
| UT_03 | Verifying if the user is a New user or an old user | To test the functionality of choosing between Sign up or Log in. | Should be able to access any browser and theatre searched should be in the database | 1.Enter N/n if you are a new customer and want to create a new account.<br>(Sign Up) | Type<br>N or n | Request for user details to create new account and update in the database | Get user name , user ID and password from user | PASS |
| UT_04 | Verifying if the user is a New user or an old user | To test the functionality of choosing between Sign up or Log in. | Should be able to access any browser and theatre searched should be in the database | 1. Enter any other letter if you are not a new customer<br>(Log in) | Any key other than N/n | Get registered user user details | Asks for User ID and Password | PASS |
| UT_05 | Creating an user account | To test the functionality of account creation. | Should be able to access any browser and theatre searched should be in the database | 1.Sign in with credentials<br>2. Exit | Sign In credentials:<br>First name :<br>Last name:<br>User ID:<br>Password : | Sign in successful , account created. The user account details are updated in the database | Sign in successful, new account created. Logged in as that particular user to select movie. | PASS |
| UT_06 | Login to the user account | To test the functionality of logging in to the user account | Should be able to access any browser and theatre searched should be in the database | 1. Login by entering your credentials l<br>2. exit | Login with credentials:<br>User id:<br>Password | Checks the details from the database and allows Login success. | Login successful,display movie list | PASS |
| UT_07 | Selection of the | To test the functionality | Should have an account in | 1. View the list of movies | By selecting 1-15 you can | Upon successful | Asks input | PASS |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | movie | of movie selection | the database | 2. exit | choose respective movies. | movie selection, Movie timings should be asked corresponding to the movie name | from the user regarding the time slot to book from the available time slots. | |
| UT_08 | Selection of the movie | To test the functionality of movie selection | Should have an account in the database | 1. View the list of movies<br>2. exit | Any key other that 1-15 will as you to re-enter the movie ID required | Display movie list again to choose the correct movie ID. | Display movie list again to choose the correct movie ID. | PASS |
| UT_09 | Selection of movie show time | To test the functionality of selecting movie show time | Should have an account in the database and should have selected a correct movie ID | 1. Refer the time slots available for respective movies and enter corresponding number<br>2. Choose one available time slot for the show | Entering number other than corresponding number to the required time slot for the movie. | Ask to enter the number corresponding to the available time slot AGAIN | Program getting into an infinite loop. | FAIL |
| UT_10 | Selection of movie show time | To test the functionality of selecting movie show time | Should have an account in the database and should have selected a correct movie ID | 1. Refer the time slots available for the respective movie and enter the corresponding number.<br>2. Choose one available time slot for the show | Number corresponding to the required time slot from the available time slots. | Ask for the number of seats to be booked. | Ask for the number of seats to be booked. | PASS |

# USE CASE 2:SEAT SELECTION AND BOOKING

| Test Case_ID | Name of the Module | Test Case Description | Pre-conditions | Test steps | Test data | Expected result | Actual result | Test Result |
|---|---|---|---|---|---|---|---|---|
| UT_01 | Choosing the number of seats to be booked. | To test the functionality of number of seats to be booked | Should have an account in the database and should have selected a correct movie ID and timing. | 1.Select the number of seats required<br>2. exit | Enter the number of seats . | Display the seat matrix | Display the seat matrix | PASS |
| UT_02 | Choosing the number of seats to be booked | To test the functionality of number of seats to be booked | Should have an account in the database and should have selected a correct movie ID and timing. | 1.Select the number of seats required<br>2. exit | Enter the number of seats in words | Asks to enter in numbers. | Asks to enter in numbers. | PASS |
| UT_03 | Block the required seat. | To test the functionality to block the seat. | Should have an account in the database and should have selected a correct movie ID and timing | 1.Select the required seats from the matrix<br>2. exit | Select the available seat from the seat matrix | Mark that seat as X indicating that its blocked. | Mark that seat as X indicating that its blocked.comments | PASS |
| UT_04 | Block the required seat. | To test the functionality to block the seats to the number of tickets we | Should have an account in the database and should have | 1.Select the required seats from the matrix<br>2. exit | 1.Select the available required seats from the matrix | Let you book the seat as many times as you have | Let you book the seat as many times as you have entered for | PASS |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | have chosen. | selected a correct movie ID and timing | | 2. exit | entered for the number of tickets. | the number of tickets. | |
| UT_05 | Selection of class of seat | To test the functionality of selection of class of seat. | Required number of seats should be selected. | 1.Make a choice among the given 3 classes. 2.Exit | 1.Enter the number 1,2 or 3 as an option. | let you make a choice and bill according to the class chosen | let you make a choice and bill according to the class chosen. | PASS |
| UT_06 | Selection of class of seats | To test the functionality of selection of class of seat. | Required number of seats should be selected. | 1.Make a choice among the given 3 classes. 2.Exit | 1.Enter number other than 1,2,3 as an option. | Let us enter the option again | Let us enter the option again. | PASS |
| UT_07 | Block the required seat. | To test the functionality to block the seat | Required number of seats should be entered previously | 1.Enter the seat number 2.Exit | Entering the seat number which exceeds the actual number of available rows. | Directs you to book as customer again | Directs you to book as customer again | PASS |
| UT_08 | Block the required seat. | To test the functionality to block the seat | Required number of seats should be entered previously | 1.Enter the seat number 2.Exit | Entering the seat number which exceeds the actual number of available columns. | Directs you to make choice again | Directs you to make choice again | PASS |
| UT_09 | Display the bill | To test the functionality to display the bill | Number of tickets ,seats and class should be chosen . | 1. choose the number of tickets. 2.Choose the type of seats. 3.Exit | Choose the seats according to the requirement and enter. | Displaying the bill with the total amount calculated according to the | Displaying the bill with the total amount calculated according to the choice. | PASS |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | choice. | | |
| UT_10 | Display the the bill | To test the functionality to display the bill | Number of tickets ,seats and class should be chosen | 1. choose the number of tickets.<br>2.Choose the type of seats.<br>3.Exit | Choose the seats according to the requirement and enter. | Display the cost with GST | Display the cost with GST | PASS |
| UT_11 | Block the required seat. | To test the functionality to display the bill | Required number of seats should be entered previously | 1.Enter the seat number<br>2.Exit | Entering the seat number in small letter | Ask to make a choice again | Runs infinite loop | FAIL |

## USE CASE 3: EVENT MANAGEMENT

| Test Case_ID | Name of the Module | Test Case Description | Pre-conditions | Test steps | Test data | Expected result | Actual result | Test Result |
|---|---|---|---|---|---|---|---|---|
| UT_01 | Login to the admin account | The admin will have to login using the login credentials which are already registered. | The admin should be having the login credentials provided by the company. | 1. The admin has to select an option to login as admin.<br>2. Have to provide the login credentials and should be able to access the account | Username and password | The admin should be able to logged into the app as admin | The admin was able to logged into the app as admin | PASS |
| UT_02 | Add movies | The admin has to add the movies which are going to be | The admin should be having the login credentials | The admin has to enter the new movies along with the | Details of the movie Like the starting and ending | The list of movies should get updated | .the entered movie will be added to the movie lists | PASS |

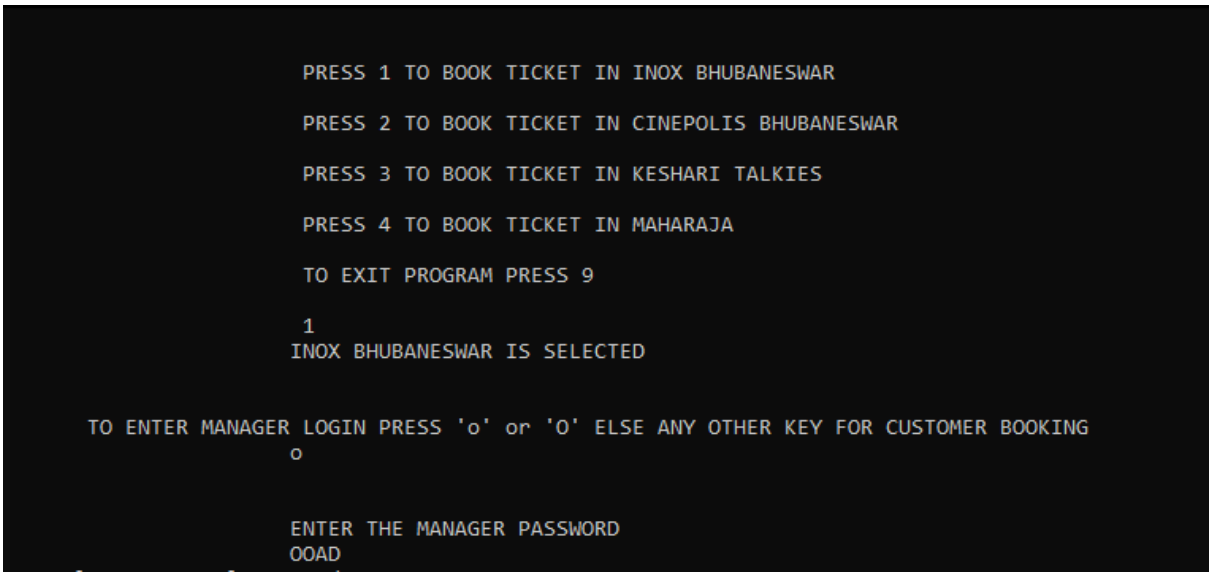
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | aired soon in the collabed theatres | provided by the company. | theatre address and the timing of the airing movie | time. | | | |
| UT_03 | Remove Movies | The admin has to remove the movies which are not airing anymore in the collabed theatres | 1. The admin should be having the login credentials provided by the company.<br>2. the movie to be removed, should be present in the movie list. | The admin has to enter the new movies along with the theatre address and the timing of the airing movie | Details of the movie Like the starting and ending time. | The removed movie shouldn’t be present in the movie list | The removed movie wasn't present in the movie list | FAIL |
| UT_04 | Add theatre | The admin will have add newly collabed theatre | Theatre details should be entered | The admin will have to enter the theatre name, opening and closing time of that theatre | Name, venue, opening and closing of a theatre | The added movie should appear in the list of theatre | The newly added movie will be seen in the theatre list | PASS |
| UT_05 | Remove theatre | The admin will have to remove some collabed theatre | 1. The admin should be having the login credentials provided by the company.<br>2. the movie to be removed, should be present in the movie | The admin will have to enter the theatre name, opening and closing time of that theatre | Name, venue, opening and closing time of a theatre | The removed movie shouldn't be there in the movie list | The removed movie wouldn't be seen in the theatre list | FAIL |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | list. | | | | | |
| UT_06 | Update the movies' details | The admin will have to update the airing time of a movie in a particular theatre | 1. The admin should be having the login credentials provided by the company.<br>2. the movie to be updated, should be present in the movie list. | The admin should be able to add the airing and ending timing of a particular movie | Details of the movie Like the airing time. | The new time of an airing movie should be updated | The new time of an airing movie will get updated | PASS |
| UT_07 | Reserve the seats for particular users | The admin will have to reserve the booked seats for the users | 1.The user details should be precise. | The admin should block the number of seats by name of the user. | The user name, number of seats and the type of class | Particular and seats should be reserved for a particular user by marking them as X | The booked number of seats and the class will be reserved for a particular user and are marked with X | PASS |
| UT_9 | Update the number of seats available | The admin should be able to update the number of seats available in each theatre for each movie in a particular time. | 1. The admin should be having the login credentials provided by the company.<br>2. Have the hold of reserved number of seats | The admin should block the reserved seats by marking them as X by which the remain seats can be found | The number of seats in each theater for a particular movie in a particular time | Should display the remaining available seats | The remaining seats will be displayed | PASS |
| UT_10 | Logout | The admin should able log out of | The admin should be having the | The user just has select the option to | User login credentials | The admin should be able to | The admin was able to come out of | PASS |

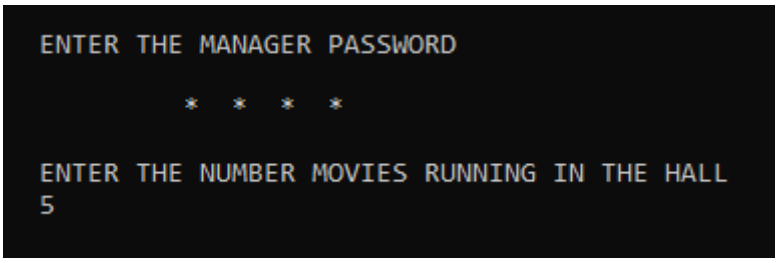
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | his/her account | login credentials provided by the company. | logout | | come out of his account | his account | |

# SCREENSHOTS OF OUTPUT

## Selecting a Theater and logging in as Manager



**Entering number of Running Halls in the Theater**




**Entering Movie Details like:**
**Movie Name, Timing**

```
ENTER THE NUMBER MOVIES RUNNING IN THE HALL
5


ENTER THE NAME OF MOVIE 1
Avengers End Game


ENTER THE NUMBER OF SHOWS OF MOVIE: Avengers End Game IN A DAY : 3


ENTER THE SHOW TIMINGS OF THE MOVIE : Avengers End Game
10:00AM
04:00PM
08:00PM


ENTER THE NAME OF MOVIE 2
Twilight


ENTER THE NUMBER OF SHOWS OF MOVIE: Twilight IN A DAY : 3


ENTER THE SHOW TIMINGS OF THE MOVIE : Twilight
10:00AM
04:00PM
08:00PM


ENTER THE NAME OF MOVIE 3
The Pursuit of Happyness


ENTER THE NUMBER OF SHOWS OF MOVIE: The Pursuit of Happyness IN A DAY : 3


ENTER THE SHOW TIMINGS OF THE MOVIE : The Pursuit of Happyness
10:00AM
04:00PM
08:00PM


ENTER THE NAME OF MOVIE 4
```

## Movie running in the theater list

```
THE MOVIES RUNNING IN HALL ARE:


MOVIE : 1       Avengers End Game

THE TIMINGS OF THE MOVIE : Avengers End Game :-
10:00AM,04:00PM,08:00PM,


MOVIE : 2       Twilight

THE TIMINGS OF THE MOVIE : Twilight :-
10:00AM,04:00PM,08:00PM,


MOVIE : 3       The Pursuit of Happyness

THE TIMINGS OF THE MOVIE : The Pursuit of Happyness :-
10:00AM,04:00PM,08:00PM,


MOVIE : 4       Zindagi Na melagi Dubara

THE TIMINGS OF THE MOVIE : Zindagi Na melagi Dubara :-
10:00AM,04:00AM,08:00PM,


MOVIE : 5       Inception

THE TIMINGS OF THE MOVIE : Inception :-
10:00AM,04:00PM,08:00PM,


MOVIE : 6

THE TIMINGS OF THE MOVIE :  :-


MOVIE : 7
```

## Manager Blocking the Seats

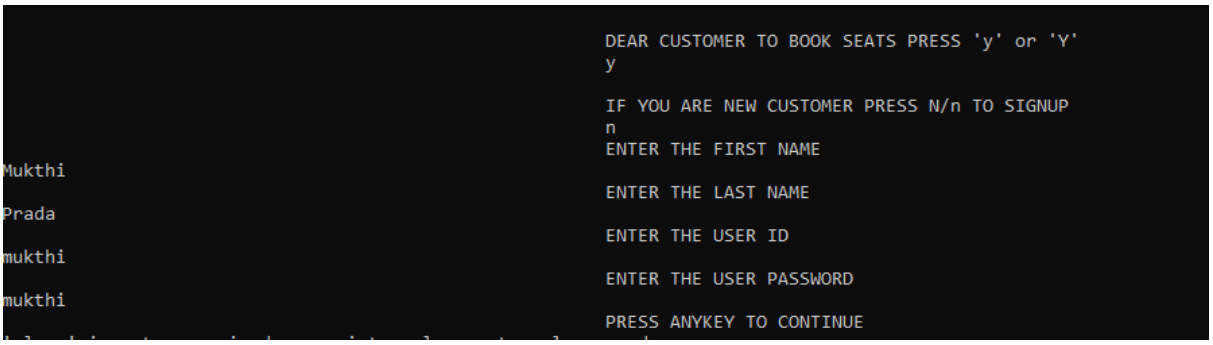
```
 ENTER THE SHOW NUMBER WHOSE SEATS U WANT TO ASSIGN FOR BOOKING
2


 ENTER THE NUMBER CORRESPONDING TO THE TIME TO BOOK TICKET
2


ADMIN ENTER THE SEATS WHICH WONT BE AVAILABLE FOR BOOKING
```

```
COST OF RECTILINEAR:Rs 500/-
COST OF SUPER DELUXE :Rs 200/-
COST OF  DELUXE :Rs 100/-

                                        BOOKED SEATS ARE MARKED WITH [x]


                                        THIS IS THE SEAT MATRIX FOR MOVIE: Twilight

      [A1] [A2] [A3] [A4] [A5]       [A6] [A7] [A8] [A9] [A10] [A11] [A12] [A13]     [A14] [A15] [A16] [A17] [A18]

      [B1] [B2] [B3] [B4] [B5]       [B6] [B7] [B8] [B9] [B10] [B11] [B12] [B13]     [B14] [B15] [B16] [B17] [B18]

      [C1] [C2] [C3] [C4] [C5]       [C6] [C7] [C8] [C9] [C10] [C11] [C12] [C13]     [C14] [C15] [C16] [C17] [C18]

      [D1] [D2] [D3] [D4] [D5]       [D6] [D7] [D8] [D9] [D10] [D11] [D12] [D13]     [D14] [D15] [D16] [D17] [D18]

      [E1] [E2] [E3] [E4] [E5]       [E6] [E7] [E8] [E9] [E10] [E11] [E12] [E13]     [E14] [E15] [E16] [E17] [E18]

      [F1] [F2] [F3] [F4] [F5]       [F6] [F7] [F8] [F9] [F10] [F11] [F12] [F13]     [F14] [F15] [F16] [F17] [F18]

      [G1] [G2] [G3] [G4] [G5]       [G6] [G7] [G8] [G9] [G10] [G11] [G12] [G13]     [G14] [G15] [G16] [G17] [G18]

      [H1] [H2] [H3] [H4] [H5]       [H6] [H7] [H8] [H9] [H10] [H11] [H12] [H13]     [H14] [H15] [H16] [H17] [H18]

      [I1] [I2] [I3] [I4] [I5]       [I6] [I7] [I8] [I9] [I10] [I11] [I12] [I13]     [I14] [I15] [I16] [I17] [I18]

      [J1] [J2] [J3] [J4] [J5]       [J6] [J7] [J8] [J9] [J10] [J11] [J12] [J13]     [J14] [J15] [J16] [J17] [J18]

      [K1] [K2] [K3] [K4] [K5]       [K6] [K7] [K8] [K9] [K10] [K11] [K12] [K13]     [K14] [K15] [K16] [K17] [K18]

      [L1] [L2] [L3] [L4] [L5]       [L6] [L7] [L8] [L9] [L10] [L11] [L12] [L13]     [L14] [L15] [L16] [L17] [L18]

      [M1] [M2] [M3] [M4] [M5]       [M6] [M7] [M8] [M9] [M10] [M11] [M12] [M13]     [M14] [M15] [M16] [M17] [M18]


                                        SCREEN THIS WAY
      ____________________________________________________________________________________________________________
```

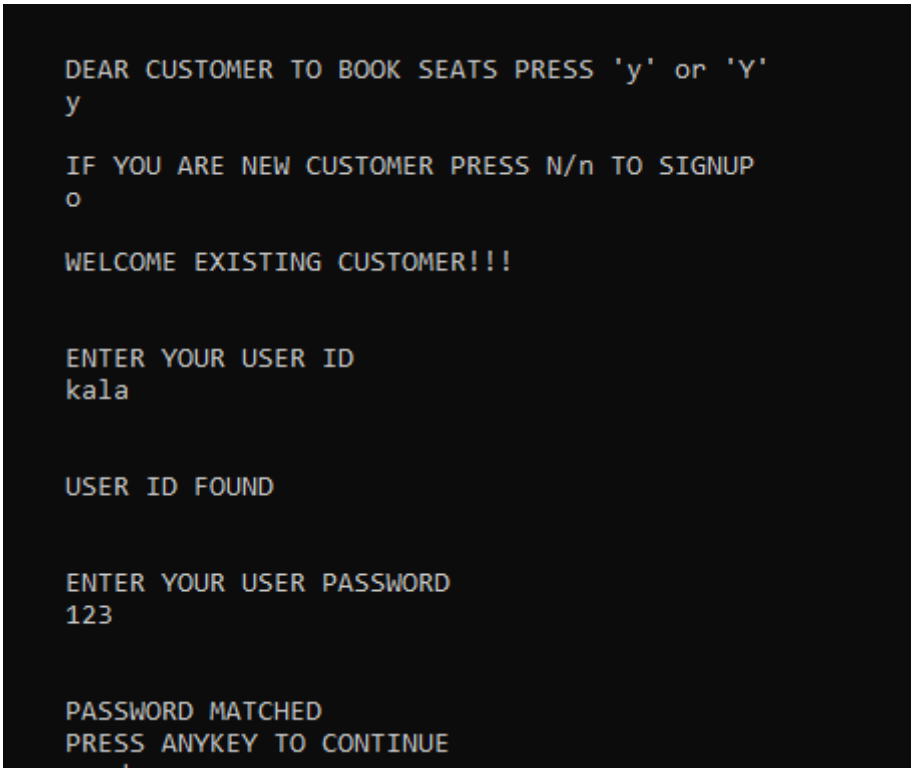
```
      [I1] [I2] [I3] [I4] [I5]       [I6] [I7] [I8] [I9] [I10] [I11] [I12] [I13]     [I14] [I15] [I16] [I17] [I18]

      [J1] [J2] [J3] [J4] [J5]       [J6] [J7] [J8] [J9] [J10] [J11] [J12] [J13]     [J14] [J15] [J16] [J17] [J18]

      [K1] [K2] [K3] [K4] [K5]       [K6] [K7] [K8] [K9] [K10] [K11] [K12] [K13]     [K14] [K15] [K16] [K17] [K18]

      [L1] [L2] [L3] [L4] [L5]       [L6] [L7] [L8] [L9] [L10] [L11] [L12] [L13]     [L14] [L15] [L16] [L17] [L18]

      [M1] [M2] [M3] [M4] [M5]       [M6] [M7] [M8] [M9] [M10] [M11] [M12] [M13]     [M14] [M15] [M16] [M17] [M18]


                                        SCREEN THIS WAY
      ____________________________________________________________________________________________________________

                                        ENTER THE SEAT NUMBER   A1
```

```
                                        BOOKED SEATS ARE MARKED WITH [x]


                                        THIS IS THE SEAT MATRIX FOR MOVIE: Twilight

      [ X] [A2] [A3] [A4] [A5]       [A6] [A7] [A8] [A9] [A10] [A11] [A12] [A13]     [A14] [A15] [A16] [A17] [A18]

      [B1] [B2] [B3] [B4] [B5]       [B6] [B7] [B8] [B9] [B10] [B11] [B12] [B13]     [B14] [B15] [B16] [B17] [B18]

      [C1] [C2] [C3] [C4] [C5]       [C6] [C7] [C8] [C9] [C10] [C11] [C12] [C13]     [C14] [C15] [C16] [C17] [C18]

      [D1] [D2] [D3] [D4] [D5]       [D6] [D7] [D8] [D9] [D10] [D11] [D12] [D13]     [D14] [D15] [D16] [D17] [D18]
```

## New Customer creating a new account (Sign In)




## Log in of old customer

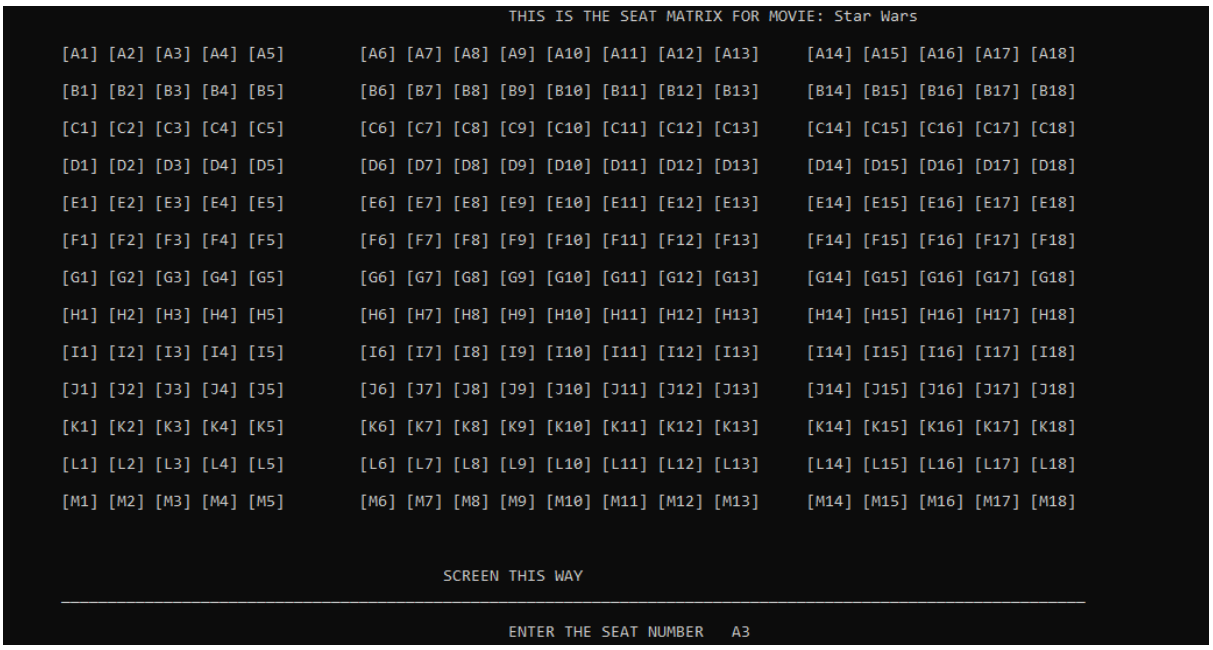


## Selecting movie and timing

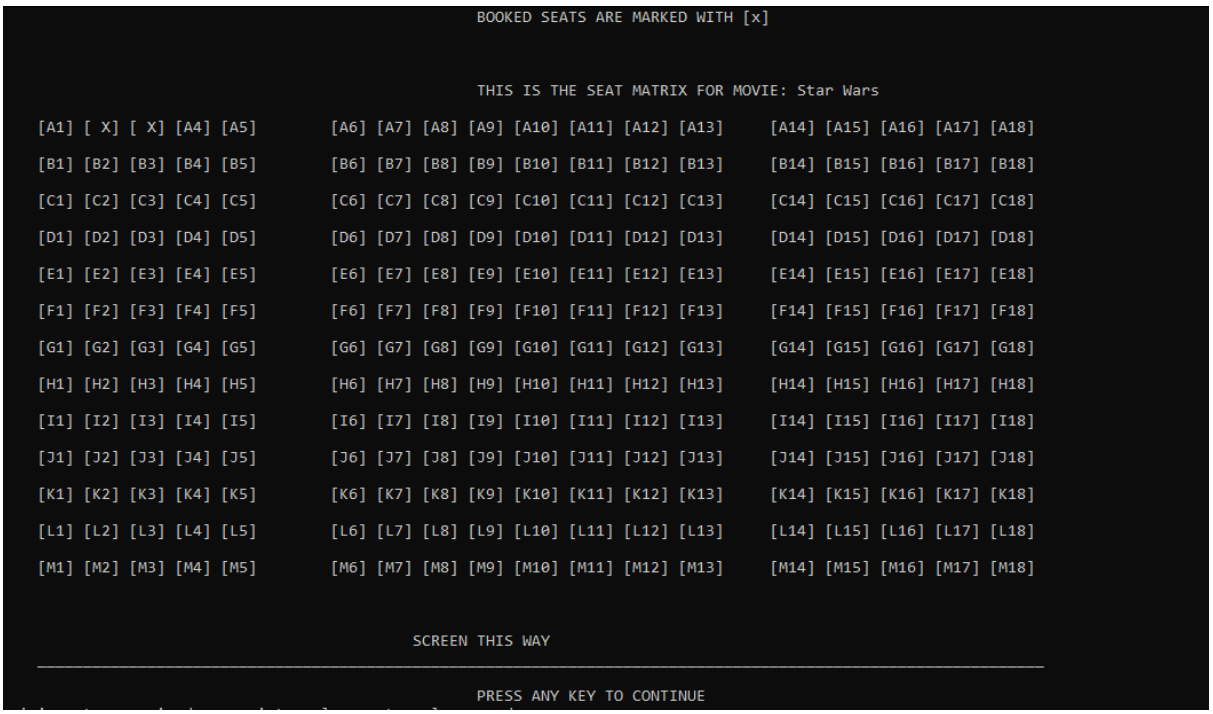
```
MOVIE : 10      Inception

THE TIMINGS OF THE MOVIE : Inception :-
10:00AM,04:00PM,08:00PM,


MOVIE : 11      Frozen II

THE TIMINGS OF THE MOVIE : Frozen II :-
10:00AM,04:00PM,08:00PM,


MOVIE : 12      Star Wars

THE TIMINGS OF THE MOVIE : Star Wars :-
10:00AM,04:00PM,08:00PM,


MOVIE : 13      Sky Is Pink

THE TIMINGS OF THE MOVIE : Sky Is Pink :-
10:00AM,04:00PM,08:00PM,



ENTER THE MOVIE NUMBER TO BOOK TICKET
12


CHOOSE THE NUMBER CORRENSPONDING TO TIMING TO BOOK TICKET FOR MOVIE :-
Star Wars                                                              2
```

## Booking Seats

```
[J1] [J2] [J3] [J4] [J5]      [J6] [J7] [J8] [J9] [J10] [J11] [J12] [J13]     [J14] [J15] [J16] [J17] [J18]
[K1] [K2] [K3] [K4] [K5]      [K6] [K7] [K8] [K9] [K10] [K11] [K12] [K13]     [K14] [K15] [K16] [K17] [K18]
[L1] [L2] [L3] [L4] [L5]      [L6] [L7] [L8] [L9] [L10] [L11] [L12] [L13]     [L14] [L15] [L16] [L17] [L18]
[M1] [M2] [M3] [M4] [M5]      [M6] [M7] [M8] [M9] [M10] [M11] [M12] [M13]     [M14] [M15] [M16] [M17] [M18]


                                SCREEN THIS WAY
_____________________________________________________________________________________________________________

                                ENTER THE NUMBER OF SEATS TO BE BOOKED


                                BOOKED SEATS ARE MARKED WITH [x]

                                 MOVIE NAME=Star Wars
                                 2


                                CUSTOMER BOOKING
```




# After booking two seats and being marked as ‘X’ automatically



## Billing Process

```
                        COST :
                               2 * 500
                              +0 * 100
GST 18% = 180




                        THE TOTAL TICKET COST = Rs 1180/-
```

## Ticket Display

```
 ____________________________________________________________________
|                                                                    |
|   Dear Customer,                                                   |
|                     Congratulation!! Your tickets has been booked. |
|                                                                    |
|                 THE DETAILS:                                       |
|                           MOVIE HALL: MAHARAJA                     |
|                           MOVIE NAME: Twilight                     |
|                           SHOW STARTS:04:00PM                      |
|                           NUMBER OF TICKETS BOOKED: 2              |
|                THE SEAT NUMBERS ARE: A1,A2                         |
|                                                    QR CODE         |
|                                                      ..++>>>       |
|                                                      ..__ ***      |
|                                                      &%^>>>>       |
|                                                      :::::::       |
|____________________________________|_______________________________|
                                     ""


                              PRESS 'C' or 'c' IF U WANT TO BOOK SEAT AGAIN?
```

**The seat blocked by manager is shown as booked when customer goes to book his/her seat**


# Appendix B: Field Layouts

**An Excel sheet containing field layouts and properties/attributes and report requirements.**

**Attributes of Customer:**

| FIELD | LENGTH | DATA TYPE | DESCRIPTION | MANDATORY |
|---|---|---|---|---|
| First Name | 256 | String | User name | Y |
| Last Name | 256 | String | User name | N |
| User ID | 10 | String | Unique username | Y |
| Password | 10 | String | | Y |

**Attributes for Manager:**

| FIELD | LENGTH | DATA TYPE | DESCRIPTION | MANDATORY |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| **User ID** | **10** | **String** | **Unique manager Id** | **Y** |
| **Password** | **10** | **String** | | **Y** |

**Attributes of Theater:**

| FIELD | LENGTH | DATA TYPE | DESCRIPTION | MANDATORY |
|---|---|---|---|---|
| **Theater Name** | **256** | **String** | **Unique Theater name** | **Y** |
| **Movie Names** | **256** | **String** | **Unique Movie name** | **Y** |
| **Movie Time Slot** | **10** | **String** | **Movie Timing** | **Y** |
| **Number of Seats** | **3** | **Integer** | **Available seats** | **N** |
| **Types of Seats** | **256** | **String** | **3 types of seats** | **Y** |
| **Cost** | **16** | **Integer** | **amount of each seat** | **Y** |

**Attributes of Ticket:**

| FIELD | LENGTH | DATA TYPE | DESCRIPTION | MANDATORY |
|---|---|---|---|---|
| **Ticket ID** | **10** | **Integer** | **Unique Id for each ticket** | **Y** |
| **User ID** | **10** | **String** | **Unique ID for user** | **Y** |
| **Ticker QR code** | **256** | **String** | **Unique QR code to represent each ticket** | **Y** |
| **Number of seats** | **3** | **Integer** | **Number of seats booked** | **Y** |
| **Total cost** | **10** | **Integer** | **Total amount to be payed** | **Y** |

**Sample Report Requirements: Include the fields to be included in the report**

| Customer Report | Theater Report | Manager Report | Ticket Report |
|---|---|---|---|
| First Name | Theater Name | User ID | Ticket ID |
| Last Name | Movie Name | Password | User ID |
| User ID | Time Slot | | Ticket QR code |
| Password | Number of Seats | | Number of seats |
| | Types of Seats | | Total cost |
| | Cost | | |