**Online Movies Booking System**



Ashish Pokhrel

00172912

Computing Project

Level 5 in Computing

Softwarica College of IT and E-Commerce

Kathmandu, Nepal

04/01/2019

Table of Contents

[Acknowledgement 4](#_Toc5647089)

[Abstract 4](#_Toc5647090)

[CHAPTER 1 - INTRODUCTION 4](#_Toc5647091)

[1.1 Project Introduction 4](#_Toc5647092)

[1.2 Justification of Project 4](#_Toc5647093)

[1.2.1 Background of Project 4](#_Toc5647094)

[1.3 Problem Statement 5](#_Toc5647095)

[1.4 Features 5](#_Toc5647096)

[1.5 Aims and Objectives 6](#_Toc5647097)

[1.5.1 Aims 6](#_Toc5647098)

[1.5.2 Objectives 6](#_Toc5647099)

[1.6 Scope 7](#_Toc5647100)

[CHAPTER 2 – Analysis 7](#_Toc5647101)

[2.1 Development Methodology 7](#_Toc5647102)

[2.2 Merits and Pitfalls of Project 9](#_Toc5647103)

[2.2.1 Merits 9](#_Toc5647104)

[2.2.2 Pitfalls or Disadvantages 10](#_Toc5647105)

[2.3 Requirement Analysis 10](#_Toc5647106)

[2.4 Prioritization 10](#_Toc5647107)

[2.5 Feasibility study 13](#_Toc5647108)

[2.6 Natural Language Analysis 13](#_Toc5647109)

[2.6.1 Steps for Natural Language Analysis: 14](#_Toc5647110)

[2.7 Use-Case Diagram 14](#_Toc5647111)

[Justification 15](#_Toc5647112)

[2.8 Initial Class Diagram 16](#_Toc5647113)

[2.9 Architecture 17](#_Toc5647114)

[Chapter 3: Design 19](#_Toc5647115)

[3.1 Structural Design 19](#_Toc5647116)

[3.1.1 Class Diagram 19](#_Toc5647117)

[Justification 20](#_Toc5647118)

[3.2 Behavioral Diagram 21](#_Toc5647119)

[3.2.1 Activity Diagram 21](#_Toc5647120)

[3.3.2 Sequence Diagram 25](#_Toc5647121)

[3.3 Database Design 28](#_Toc5647122)

[3.1 Entity Relationship Diagram 28](#_Toc5647123)

[3.2 Meta-Data 29](#_Toc5647124)

[Chapter 4: Implementation 31](#_Toc5647125)

[4.1 Programming Language 31](#_Toc5647126)

[4.2 Development Environment 31](#_Toc5647127)

[Chapter 5: Testing 32](#_Toc5647128)

[5.1 Black Box testing 32](#_Toc5647129)

[5.2 White Box Testing 44](#_Toc5647130)

[Chapter 6: Other Project Issues 52](#_Toc5647131)

[6.1 Risk Management 52](#_Toc5647132)

[6.2 Configuration Management 54](#_Toc5647133)

[6.2.1Version Control 55](#_Toc5647134)

[Future Work 56](#_Toc5647135)

[User Manual 56](#_Toc5647136)

[Limitation 56](#_Toc5647137)

[Chapter 7: Conclusion of the project 57](#_Toc5647138)

[Chapter 8: Reference and Bibliography 58](#_Toc5647139)

[Chapter 9: Appendix 59](#_Toc5647140)

[Figure 1:Waterfall model 11](#_Toc5651853)

[Figure 2:Use-case Diagram for Online Movies Booking 16](#_Toc5651854)

[Figure 3: Initial Class Diagram 17](#_Toc5651855)

[Figure 4: MVC Design Pattern 18](#_Toc5651856)

[Figure 5: Final Class Diagram 21](#_Toc5651857)

[Figure 6: Activity Diagram for movies Booking 23](#_Toc5651858)

[Figure 7: Activity diagram for search movies 24](#_Toc5651859)

[Figure 8: activity diagram for admin 25](#_Toc5651860)

[Figure 9: Sequence Diagram for User activity 26](#_Toc5651861)

[Figure 10:Sequence Diagram for Cancel Booking Movies 28](#_Toc5651862)

[Figure 11: ER-Diagram for Online movies Booking System 29](#_Toc5651863)

[Figure 12: metadata of user 30](#_Toc5651864)

[Figure 13: metadata for ticket 30](#_Toc5651865)

[Figure 14: metadata for shows 31](#_Toc5651866)

[Figure 15: metadata for screen 31](#_Toc5651867)

[Figure 16: metadata for movies 31](#_Toc5651868)

[Figure 17: metadata for hall 31](#_Toc5651869)

[Figure 18: (i.) Black Box Testing for login 35](#_Toc5651870)

[Figure 19: (ii.) redirect to home after login 36](#_Toc5651871)

[Figure 20: White Box Testing for Login 36](#_Toc5651872)

[Figure 21: i) Black Box testing for User Register 37](#_Toc5651873)

[Figure 22: ii.) Redirecting to Login after User Register 37](#_Toc5651874)

[Figure 23: i) Black Box for Add Movie 38](#_Toc5651875)

[Figure 24: ii.) view Movie in home page 38](#_Toc5651876)

[Figure 25: i) Blackbox testing for Choose Seat 39](#_Toc5651877)

[Figure 26: ii) View on my ticket page of book ticket 39](#_Toc5651878)

[Figure 27: i.) testing for add show 40](#_Toc5651879)

[Figure 28: View in showtime after add show 41](#_Toc5651880)

[Figure 29: Black Box Testing for Cancel Booking 41](#_Toc5651881)

[Figure 30: Black Box Testing for Admin Dashboard 42](#_Toc5651882)

[Figure 31: redirecting to admin dashboard after admin Login 43](#_Toc5651883)

[Figure 32 i.) Black Boxing testing for delete movie in database 44](#_Toc5651884)

[Figure 33 ii.) Error during deleting movies 44](#_Toc5651885)

[Figure 34: White Box Testing for Login 45](#_Toc5651886)

[Figure 35: White Box testing for Register 46](#_Toc5651887)

[Figure 36: Unit Testing for Add Movies 47](#_Toc5651888)

[Figure 37: Unit Testing for Book seat 48](#_Toc5651889)

[Figure 38:Unit Testing for cancel Booking 48](#_Toc5651890)

[Figure 39: Unit Testing for Add showtime 49](#_Toc5651891)

[Figure 40: Unit Testing for Delete movie 49](#_Toc5651892)

[Figure 41: Unit Testing for Add screen 50](#_Toc5651893)

[Figure 42: White Box Testing for edit Movie 51](#_Toc5651894)

[Figure 43: White Box Testing fpr add hall 52](#_Toc5651895)

[Figure 44: GitHub for Movie Booking 56](#_Toc5651896)

[Figure 45: Tree structure 57](#_Toc5651897)

[Figure 46: User Model 61](#_Toc5651898)

[Figure 47: Movie Model 62](#_Toc5651899)

[Figure 48: Show Model 62](#_Toc5651900)

[Figure 49: Booking Model 63](#_Toc5651901)

[Figure 50: Screen Model 63](#_Toc5651902)

[Figure 51: Hall Model 64](#_Toc5651903)

[Figure 52: User Controller 65](#_Toc5651904)

[Figure 53: (i) Movie Controller 66](#_Toc5651905)

[Figure 54: (ii) Movie Controller 67](#_Toc5651906)

[Figure 55:)(i) Show Controller 68](#_Toc5651907)

[Figure 56: (ii). Show Controller 69](#_Toc5651908)

[Figure 57: Seat Controller 69](#_Toc5651909)

[Figure 58: (i) Ticket Controller 70](#_Toc5651910)

[Figure 59: Hall Controller 70](#_Toc5651911)

# Acknowledgement

The Project Online Movies Booking is a web application that helps to customer to book seats for movies through web application.

# Abstract

Online Movie Booking System is a web application which is developed for customer to book movie ticket using web application through internet from anywhere anytime from globe. This site help customers to know about running movies in cinema hall along with showtime and price. Customers can book movies ticket without visiting to cinema hall. The main aim of this organization called as City Movie is to provide all information related to cinema hall seat, running movies showtime, etc. The organization (city movie) want to develop a website show that customer can easily view and book movie through the sites. By using this site, customer can book movie ticket for home or anywhere.

Online Movie Booking System has three users i.e. Register User who can book movie, read showtime, movies details, etc. Other one is Admin who add, edit, and delete Movie, Show, screen, hall, etc. to the database.

I have chosen Waterfall methodology for development methodology and for implementation I have used Laravel 5.7 version framework of php with XAMPP server with MySQL database with MVC design Pattern for architectural design.

# CHAPTER 1 - INTRODUCTION

## 1.1 Project Introduction

Nowadays, people are very modern and technology based. They want easy and stress less life. They do not want to stand on long queue. They used technology like mobile phone, Laptop and internet for buying products, ordering food and booking seat for travel or watch movie in theaters on online. The proposed project “Online Movies Booking System” whereby customers can book ticket for movies in online from a multiplex web app in real time.

## 1.2 Justification of Project

### 1.2.1 Background of Project

Online Movie booking system is web portal where you can book movies for specific date or choose timing for movies show, watch movies trailer and read reviews. You can also view Upcoming movies.

You can also know the rate for Movies and timing just browsing for anywhere and anytime on online. You just need to register in the app and login to browse to all movies which are currently running in theaters and many more facilities. Customers can book 24 hours a day from anywhere for the around globe.

Online movies booking system is very user-friendly application where you can interact with system easily. You do not need to go for theater for booking for movies. You can track everything about movies, timing of movies just clicking it of web app on online. I have used PHP for programming and My SQL for manage database of movies.

## 1.3 Problem Statement

Customer or staff of cinema hall encounter many problems by booking movie ticket on paper-based system. There is chances of duplication of ticket or seats, customer cannot get seat which they want to sit. Nowadays, customer want to do any task work quickly and shortly, so they don’t want to stand on queues for booking movies ticket in font of theaters booking counter. So that admin or manager of movies hall get difficult to manage these problems. The main problem of current running system is time consuming and complex system for both customer and staff or manager of cinema hall.

The main motive of this project is to reduce complexity of the running system in cinema hall. The proposed project will overcome all the problem related to booking process of movie ticket. This new proposed system shows all show time scheduled for movies so that any customer can see details of movies with respective show time from using web browser from anywhere anytime.

## 1.4 Features

* User can Login and Register to the system.
* User can Book movies.
* Admin can do crud operation.
* User can watch details of movies with trailer and showtime.
* User can Cancel Booking.
* User can change their profile details.
* Lists of Upcoming Movie with details.

## 1.5 Aims and Objectives

### 1.5.1 Aims

* Main aims of the proposed system are to book the movie ticket for cinemas.
* Replace to Manual Booking i.e. paper based booking.
* Promote Movie through internet.
* User can know showtime with details of movies through web application by surfing internet.

### 1.5.2 Objectives

* Booking Movies ticket from online.
* User can know details of movies with scheduled showtime by surfing internet with visiting cinema hall.
* User friendly.
* Helpful for management for decision making.
* Provide 24 hors service to customer through online.
* For make secured and reliable booking system for cinema.

## 1.6 Scope

Online movie Booking system is a web portal which helps for organization for storing and retrieving information about movies, upcoming movies, seats reservation, ticket sales etc. UI is user friendly so that user can easily use system.

# CHAPTER 2 – Analysis

Analysis is the process of identification and documentation of requirement of the proposed system. In the analysis phrase, first step is to feasibility study after that we model system in use case diagram and class diagram which is also called system modelling. (SDC, 2011)

Analysis is done to examine the requirement which is given by the user so that it is easy to find out features and importance of the system. Analysis is important parts before starting developing any software. There are many methods for analysis project like a problem identification, gathering a requirement from a user, and also develop prototype of the system. There are many techniques for analysis for project like by interview with user, questionnaire with user, observation in the workplace, etc.

For this project I prefer to used soft approach methodology for analysis. Soft approach is mainly user center focused. By using soft approach, system id design according to user requirement so that system will be user friendly.

## 2.1 Development Methodology

Waterfall model is the first introduced in the software development. It is simple and easy to understand. Each phrase needs to finished before next phrase start. In this methodology, whole process is divided into separate tasks. Outputs of one phrase is input for other phrase int this methoolody.it is also called sequential process which is look like a downward waterfall, (SHARMA, 2018)

Different Phrase of Waterfalls Methodology are:

1. Requirement Analysis:

First phrase of waterfall model where analyst understand the user requirement so that they clear on what need to be design, its function and its purpose. Inputs and output are studied.

1. System Design:

Gather requirement in phrase first studied here and according to that system is design. It also helps for overall architecture of the system.

1. Implementation:

From inputs of system design, system is developed in small program called as units. After all units developed system is integrated. Each developed small unit is test in next phrase.

1. Testing

Small program developed in implementation phrase is tested in this phrase. In this phrase each unit test it functionality which is called as unit testing. Testing is so important because after deployment user should not face any error when running a system. Both functional and non-functional testing is done here.

1. Deployment:

After Testing both functional and non-functional testing in testing phrase, now system is release in user environments.

1. Maintenance:

This phrase is done after installation system to the customer environment. In this phrase modification can be make from request by customers or when some defect is available in the system. User get can full support and regular maintenance for developed system.

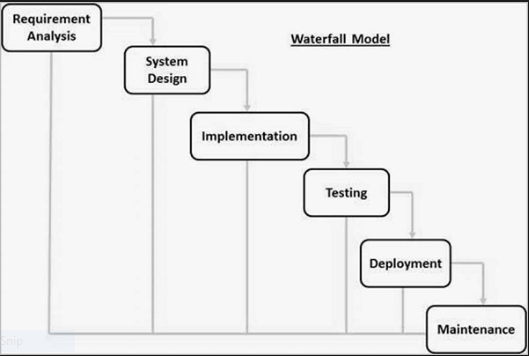


Figure :Waterfall model

## 2.2 Merits and Pitfalls of Project

### 2.2.1 Merits

* Always available for Reservation Seats.
* Reduce no-shows.
* Less in Phone Time.
* Get valuable information of business

### 2.2.2 Pitfalls or Disadvantages

* Online Payment system not available.
* Require internet Access.
* Virus and Trojans can steal information.

## 2.3 Requirement Analysis

Requirement analysis is the process of identification of user requirement for system developed. It is one important parts of system development. For success of any product, requirement analysis is important for analyze any project requirement. (ReQst, 2019)

There are two types of requirement:

1. Functional Requirement

Functional Requirement is describing of how system can do.

1. Non-Functional Requirement

Non-function Requirement is describing of how system behave.

## 2.4 Prioritization

Prioritization is the process of determine candidate requirement for the system which should be included for certain release. It can ensure easily that which requirement should be first and which should be last. (requirement.com, 2004).

I have used MOSCOW prioritization for requirement analysis for this project.

MOSCOW stand for

M=**Must Have**

S=**Should Have**

C=**Could Have**

W=**Would have**

|  |  |  |  |
| --- | --- | --- | --- |
| Functional and non-functional requirement | Requirement | Justification | Moscow |
| F(R1) | User Registration to system | User must registration to system. | Must have |
| F(R2) | User Login to system | User must Login to System for booking movies. | Must have |
| F(R3) | Add Movies | Admin can add movies | Must have |
| F(R4) | View Movies | User can view details of movies | Must have |
| F(R5) | View show time | User can view details of show time of movie | Must have |
| F(R6) | View show date | User can view details show date of movie | Must have |
| F(R7) | Update User profile | User can update their profile | should have |
| F(R8) | Update Movies | Admin can update movies | Must have |
| F(R9) | Ticket generate after booking Movies | Ticket must generate after user book particular movie | Must have |
| F(R10) | Seat selection | User can select seat | Must have |
| F(R11) | Show name | System must view show name | Must have |
| F(R12) | Show type | System must view show type and show time | Must have |
| F(R13) | Delete Movie | Admin can delete movie | Must have |
| F(R14) | Admin Login | Admin can login to system | Must have |
| F(R15) | Update Admin Profile | Admin can update his/her profile | Should have |
| NF(R16) | Security of system | Make sure that System must secure and protected from unauthorized access. | should have |
| NF(R17) | Portability | System should be run in every platform. | Should have |
| NF(R18) | usability | System should navigation easily and easy to use. | Should have |
| NF(R19) | Scalability | System should store more and more information easily | Could have |
| NF(R20) | Maintainability | Maintenance should be done in regular basic | Must have |
| F(R21) | Check seat availability | Admin and user can check available seat | Must have |
| NF(R22) | Privacy | System should keep user information private | should have |
| NF(R23) | Reliable | Precise and accurate information should given to system | Should have |
| F(R24) | Payment System | Online payment system for paying fee for movies | Could have |
| F(R25) | Rating and Reviewing Movie | User can rate and Review Movie | Could have |

## 2.5 Feasibility study

Feasibility study is used to describe of idea which is ensuing a project which is legally and technically feasible and economically just able. It can also identify project is potentially success. For this project feasibility study is undertaken which are as below:

Advantages of feasibility study:

* It can provide valuable information for the project.
* By evaluating multiple factor, it has enhanced success rates.

Disadvantages of feasibility study:

* Costly
* Wrong information

## 2.6 Natural Language Analysis

Natural Language Analysis is the process of identifying verbs, adjectives and noun in the description of project Scenario. NLA helps to get candidate class with their relationship and attributes. Nouns is related to potential class; adjectives are a potential attributes and verbs is potential functionality of the system.

### 2.6.1 Steps for Natural Language Analysis:

* Find out all nouns, verbs and adjectives from scenario.
* Remove all having synchronism.
* Remove all ambugies words.
* Remove technical details like database, network etc.
* Remove project out of scope.

## 2.7 Use-Case Diagram

A Use-case is a software and system engineering term that describe how user can interact with system to perform a particular operation. A use case acts as a software modelling technique that implies function to be implemented. (Techopedia, 2018)

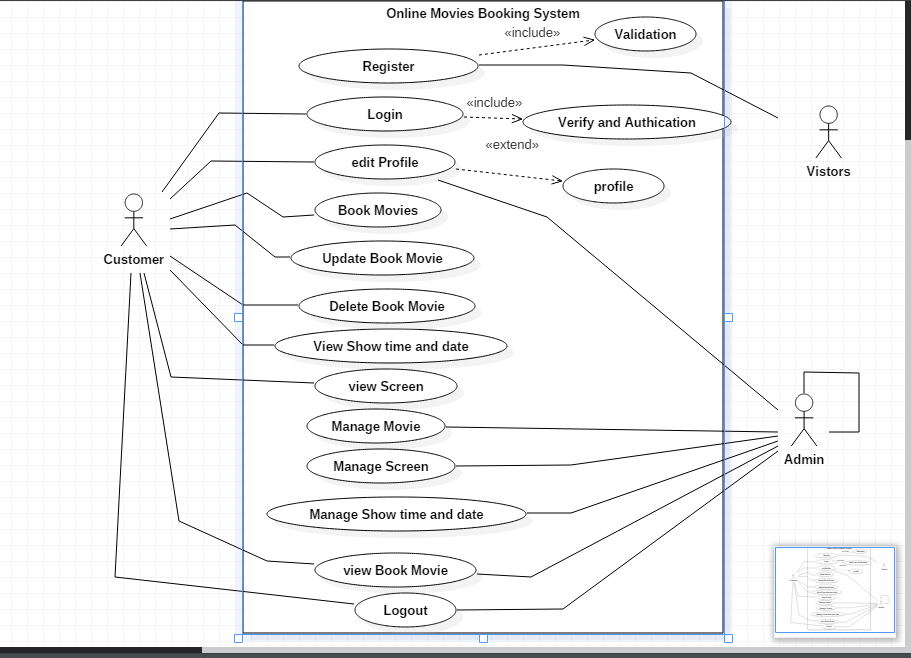


Figure :Use-case Diagram for Online Movies Booking

### Justification

In above diagram, there are three types of actors i.e. first one is visitor’s who is unregister customer for the system, second one is registered customer and last one is admin who manages whole system.

First actor i.e. visitor who can register to the system via registration script and he/she can become the registered customer. Now, he/she can do all things that registered customer can do to the system.

Second actor i.e. registered customer who is mainly focus by system. He/she can book movies, edit their profile, view different show time and date of movies running in the theaters and also update and delete book movie.

Lat actor i.e. admin who is system administration. He/she can manage movies, different shows times and dates, manages screen view book movies manages etc.

# 2.8 Initial Class Diagram

A class diagram is a description of relationship between different classes. It is a static diagram which represent static view of system. It describes the attributes and operations of class. It also shows a collection of classes, interface, association. (Guru, 2019)

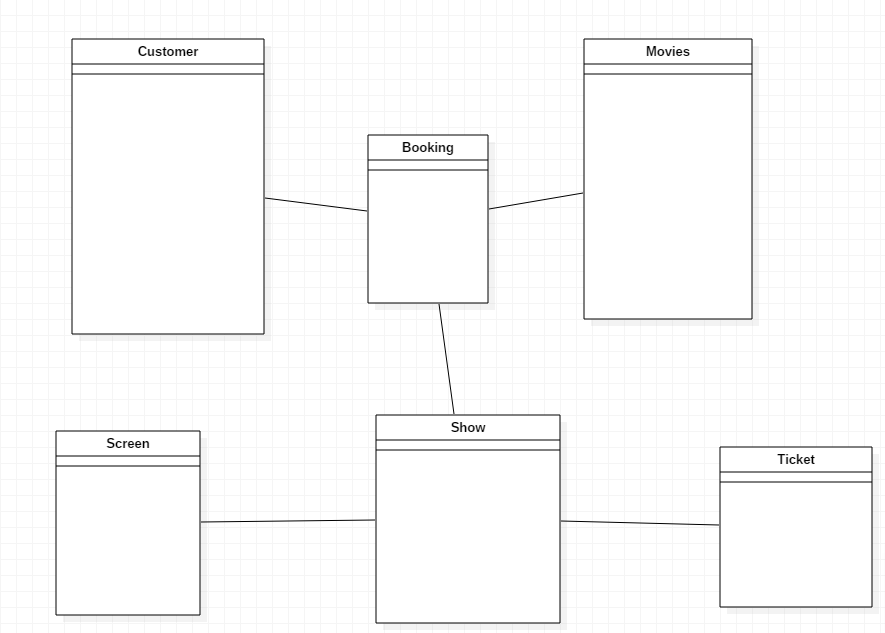


Figure : Initial Class Diagram

## 2.9 Architecture

**Design Pattern**

Design pattern is a reusable solution for commonly occurring problem is software development. It is not a finished product. It is only template for how to solve any problem during development of software. (Sourcemaking, 2019)

I have used MVC design pattern for my project movies booking. MVC stands for Model view Controller. It is a structural design pattern. It broken down application into three parts i.e. model, view and controller. It is frequently used design pattern in software development industry to develop project scalable and extensible.

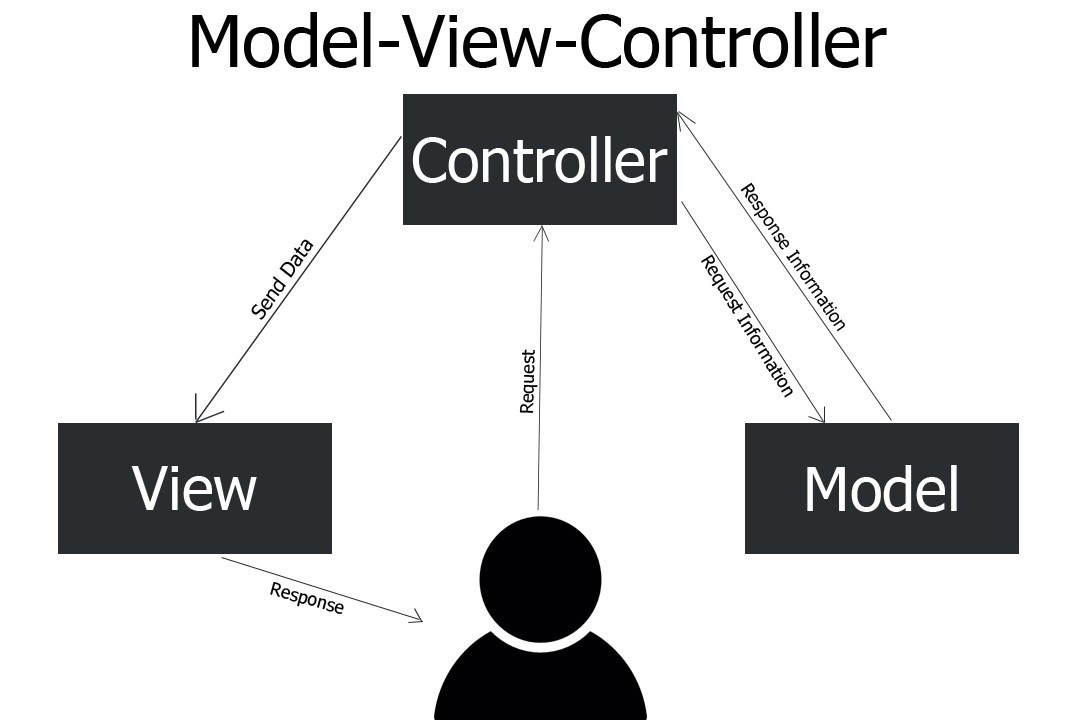
****

Figure : MVC Design Pattern

**Model:**

Model represents all business logic of the system

**View:**

View represents all UI parts of the system.

**Controller:**

Controller acts as mediator for view and model for communication.

# Chapter 3: Design

A design phrase is detailed document providing information about a going to developed product of process. Its main aims to ensure that development product should meet users’ requirements. There are three types of design phrase i.e. structural design, behavioral design and database design.

## 3.1 Structural Design

Structural diagram illustrates static structure of the system. It shows how the different component or module is connected and interact with each other. It cannot change unit the system development completed. It also shows the relationship of class in the system. Class diagram, ER-diagram, object diagram, Package diagram, model diagram, collaboration diagram are examples of structural diagram. For this project, I have used Class Diagram, ER-diagram which are illustrated briefly with diagram below.

## 3.1.1 Class Diagram

Class Diagram is UML, a type of static view of the system that describes the whole structure of the system by show classes, attributes, operation or method and relationship between classes. (Visual-Paradigm, 2019)

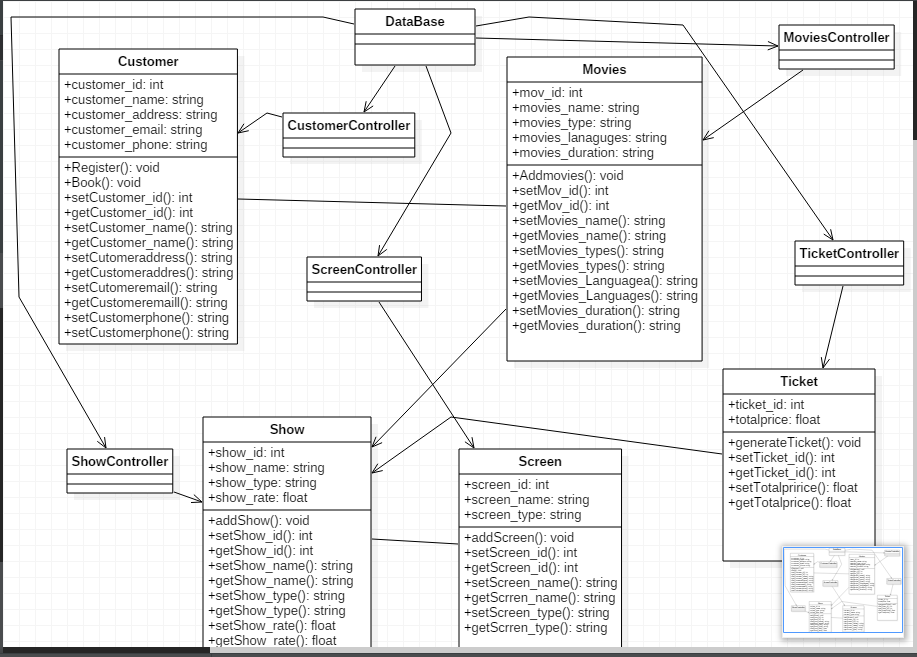


Figure : Final Class Diagram

### Justification

The above class diagram illustrated that each class has each controller and controller are direct associated with Database. Ticket class is dependent with show class and ticket controller. Booking is dependent on Movies and Customer class.

## 3.2 Behavioral Diagram

Behavioral diagram depicts the dynamic behavior of object with a system which can describes the changes of system over run time. Changing of requirement might change to the structure of the system. System diagram is design according to the requirement of the user. Activity diagram, sequence diagram, use-case diagram, state machine diagram, communication diagram, timing diagram, etc. are examples of behavioral diagram. (UML, 2019). For this project, I have used three behavioral diagrams i.e. Use-Case Diagram, Activity diagram, and Sequence Diagram which are illustrate as below:

### 3.2.1 Activity Diagram

Activity diagram is UML diagram which describes the dynamic aspect of the system. It is advanced version of flow chart which show logic of flow of one activity to another activity.

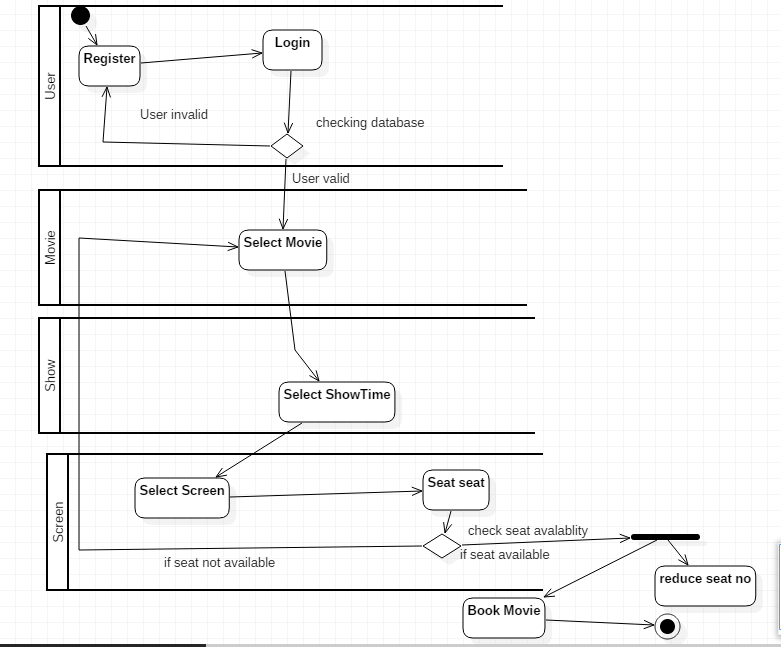


Figure : Activity Diagram for movies Booking

**Justification**

The above activity diagram is for booking movies for user. At first user can register and after that login in user class. If user is valid system will redirect to movies class display with movies name where user can select movies or if user is invalid system will redirect to register form. After selecting movies, now user will select show time in show class after that again select screen and seat in screen class. If seat is unavailable then system will redirect to movies list again or if seat is available two operation is performed simultaneously one is book movies and another one is reducing no of seats from particular screen.

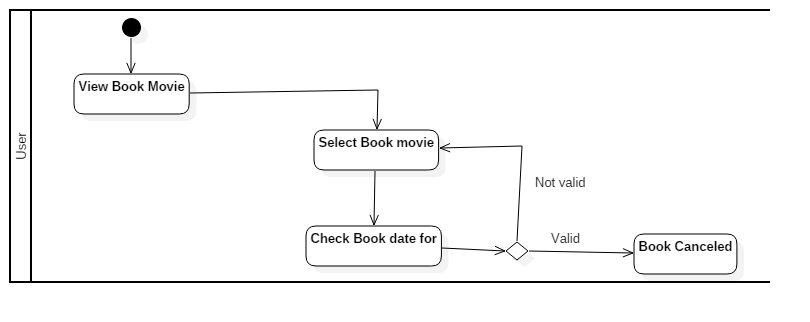


Figure : Activity diagram for search movies

**Justification**

Above activity diagram is for cancel book movie by the users. At first user can view movies in his/her booking lists. After that customer select book movie and if movie date is already gone then, customer will choose upcoming date where he/she has book movies. If date is not gone than user will cancel book from the database.

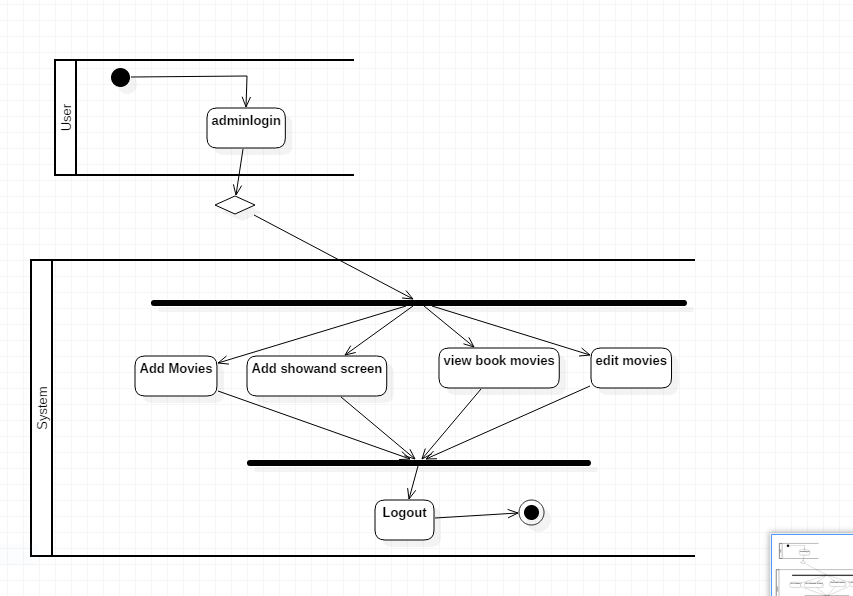


Figure : activity diagram for admin

**Justification**

Above activity diagram illustrated the activity done by the admin. Admin can add edit delete movies, showtime and screen. In the diagram at first admin login to system after that admin will add movies add showtime and screen view booked movie by user and edit movies.

### 3.3.2 Sequence Diagram

Sequence diagram can illustrate how object can interacts with message system in sequential order. It describes the runtime system. It also describes how object can pass within system. It is used by developer team to understand requirement a new and existing system. Sequence diagram for my project are as follows:

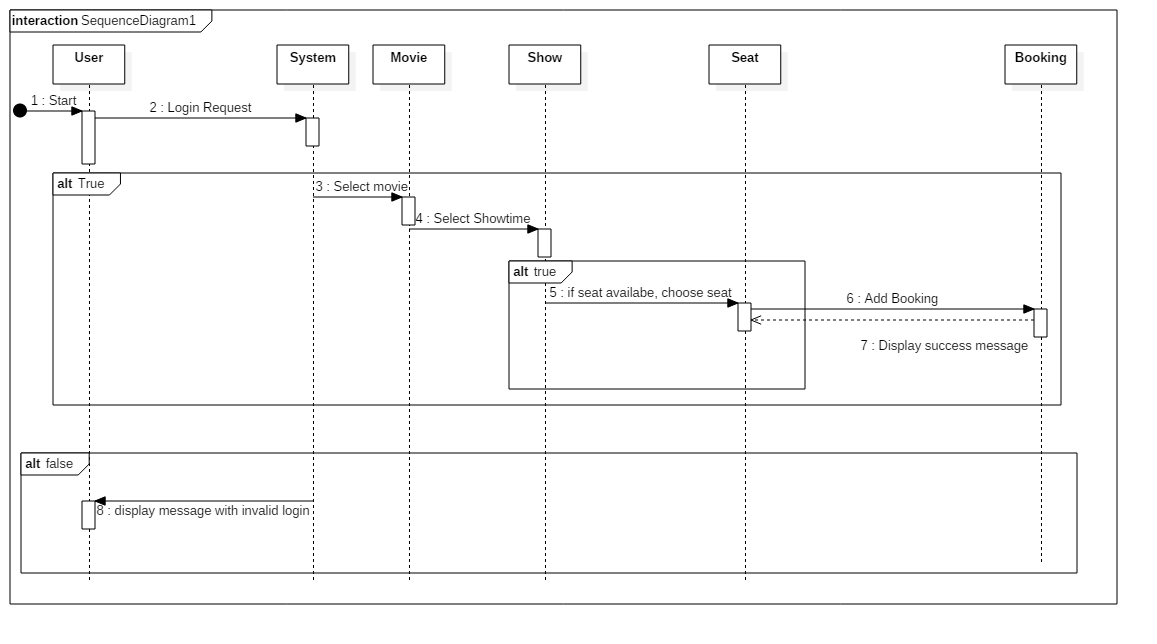


Figure : Sequence Diagram for User activity

**Justification**

As above diagram, it is sequence diagram for User activity in the system. It shows in diagram that User sends a Login request message with username and password to the system and system will send message to the database, and check if username and password valid then in combined fragment if else user send object to movie class after that to showtime object and again it check to database whether seat is available or not if seat available user sends object to seat and movie is book if not available user get redirect to show time again. while database send response to the system of and again system send success message to User interface.

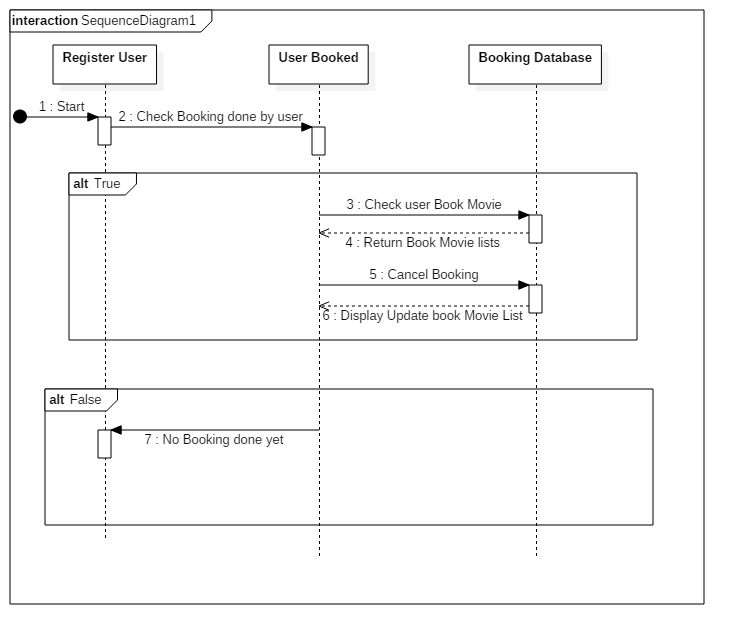


Figure :Sequence Diagram for Cancel Booking Movies

Above figure, it is sequence diagram for user cancel book movie from the database. At first, user check whether they have made booking or not. If booking is made, then select book movies and cancel it and send to booking database and booking database send response with new updated booking lists in UI interface. If booking is not made already, then it response message with movie not book yet.

## 3.3 Database Design

Database design is the process of designing, development and implementation of data for proposed system. The main aims of database design are to produce logical and physical model for any system. (Guru, 2019)

### 3.1 Entity Relationship Diagram

An Entity Relationship Diagram also called as ERD is a type of flowchart that illustrates how entities relate with each other within system.

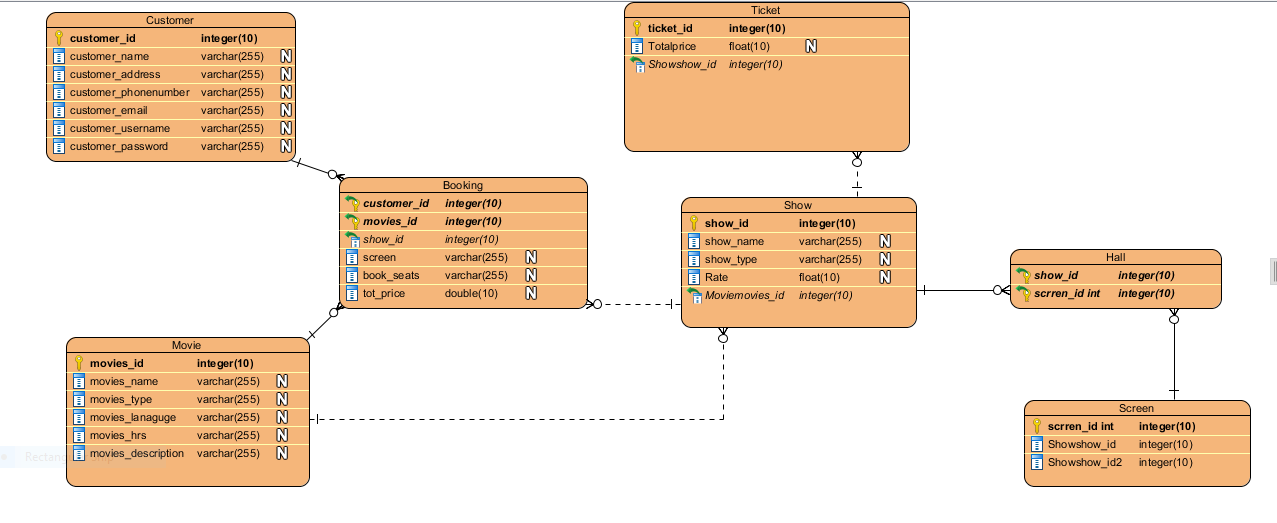


Figure : ER-Diagram for Online movies Booking System

In above diagram, it is ER-diagram for online movie booking system. It shows the relationship between entities of database. Above ER-diagram is developed for real working database for movies booking. In above ER-Diagram, customer and movies has many to many relationships has linking table called as Booking table between customer and movies because one Customer can book multiple movies whereas one movie can book by multiple customers. Shows and booking has one to many relationships where showed is foreign key in booking table because one show has multiple booking but one booking has for only one show only. Movies and Show table have one to many relationships because one movie has multiple show but one show has only one movie at time. Show and Screen table has many to many relationships which both have linking table called as Hall table because one show has multiple screens also one screen has multiple show. And, show and ticket has one to many relationships.

### 3.2 Meta-Data

Meta-data is data of data, which describe the context of data stores in columns helps for organize, find and understand data in Relational database management system.

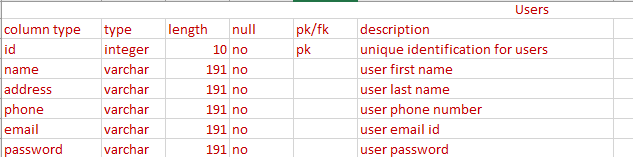


Figure : metadata of user

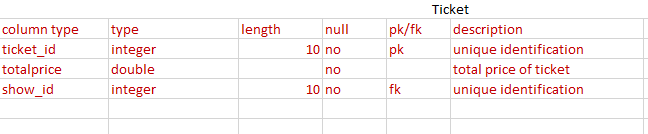


Figure : metadata for ticket

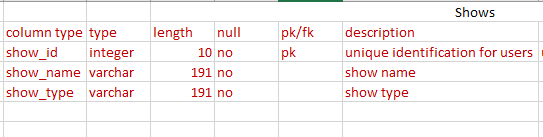


Figure : metadata for shows

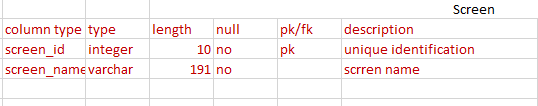


Figure : metadata for screen



Figure : metadata for movies

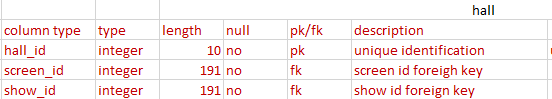
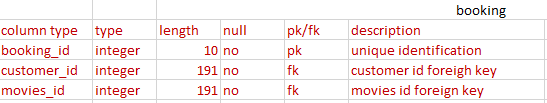


Figure : metadata for hall



# Chapter 4: Implementation

Implementation is the process of converting logical implementation into the real working system. It also ensures that how a new system is built and how systems functions work. It is 3rd phrase of the software development lifecycle. Code and UI is in appendix.

## 4.1 Programming Language

Programming Language is the language which can give instructions for the computer to perform a specific task with specific output. It is used for creating a program or a system by implementation algorithm. The high-level language such as Java, C#, Python, C++, etc. are more easily to use but require compiler or interpreter to convert into machine code because computer understand only machine language like 0 and 1.

For My project, I have used php with Laravel Framework for converting logical into real implementation. Php stands for Hypertext Preprocessor which is a server-side scripting language used for development of the web application. Laravel is most used framework php for developing web systems. Laravel use MVC design Pattern. Php as follows object-oriented program. For store data in database, I have used MYSQL database. I have also php with JavaScript and ajax for more user friendly.

## 4.2 Development Environment

Development Environment is the processes and programming tools which is used for development or creating any software products. It is also called as physical environment. An integrated development environment is the set of processes and tools which are integrated to provide developers for viewing Interface and development processes like code, testing and other packages. The examples of integrated development environment are Microsoft Visual Studio. There are many development environments like Microsoft Visual studio, Visual Code, Php Storm, sublime, etc. (TechTarget, 2019)

For my project, I have used Visual Code for development environment. It is a free of cost and flexible so that it is easy to use. For my project, I have installed Laravel Framework with MVC design pattern in visual code. MVC is architectural design pattern which can reuse code and also maintains the code standards.

# Chapter 5: Testing

Software Testing is the process of check whether actual result matches expected result to ensure that software is defect fee. It also helps to finds out error, missing requirement. (Guru, 2019)

## 5.1 Black Box testing

Black Box Testing is software testing that focuses on the system functionality rather than internal structure. Black box testing ensures that system output whether it is giving correct output or not. It is also called functionality testing.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testcase No. | Testcase | Expected result | Actual result | Pass or fail | Action | Conclusion |
| k1 | Login | User can access system by entering username and password. | System accessed. | Pass | No action | Pass |
| 2 | Register | User can redirect to login after filling details on register page | Login form view | Pass | No action | Pass |
| 3 | Book Ticket | User can book tickets. | User book ticket Successfully | Pass | No action | Pass |
| 4 | Add Movies | Movies should add in database | Movies added to database successfully | Pass | No action | Pass |
| 5 | Update Movies | Movies should be update in database | Movies Updated successfully | Pass | No action | Pass |
| 6 | Delete Movies | Movies should be deleted from database | Movies deleted from database | Pass | No action | Pass |
| 7 | Add show | Show should be added to database | Show added successfully | Pass | No action | Pass |
| 8 | Add screen | Screen should add to database | Screen added successfully | Pass | No action | Pass |
| 9 | Add hall | Hall should be add in database | display error message | fail | fix error message | fail |
| 10 | Add Upcoming Movies | Add upcoming movies in database | Display error message with no route found | fail | Fix route error | fail |
| 11 | Update Upcoming Movies | Update upcoming movies in database | Updated successfully | pass | No action | Pass |
| 12 | Cancel Booking | Book ticket should delete from database | Book ticket deleted successfully | pass | No action | Pass |

**Test No:** 1

**Purpose of test:** get username and password and redirect to home page

**Test name:** test Login

**Expected Result:** Login and redirect to home page

**Actual Result:** Login Successfully and redirect to home page

**Test Status**: Pass

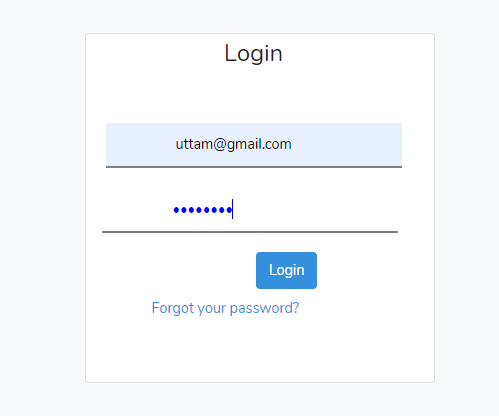


Figure : (i.) Black Box Testing for login

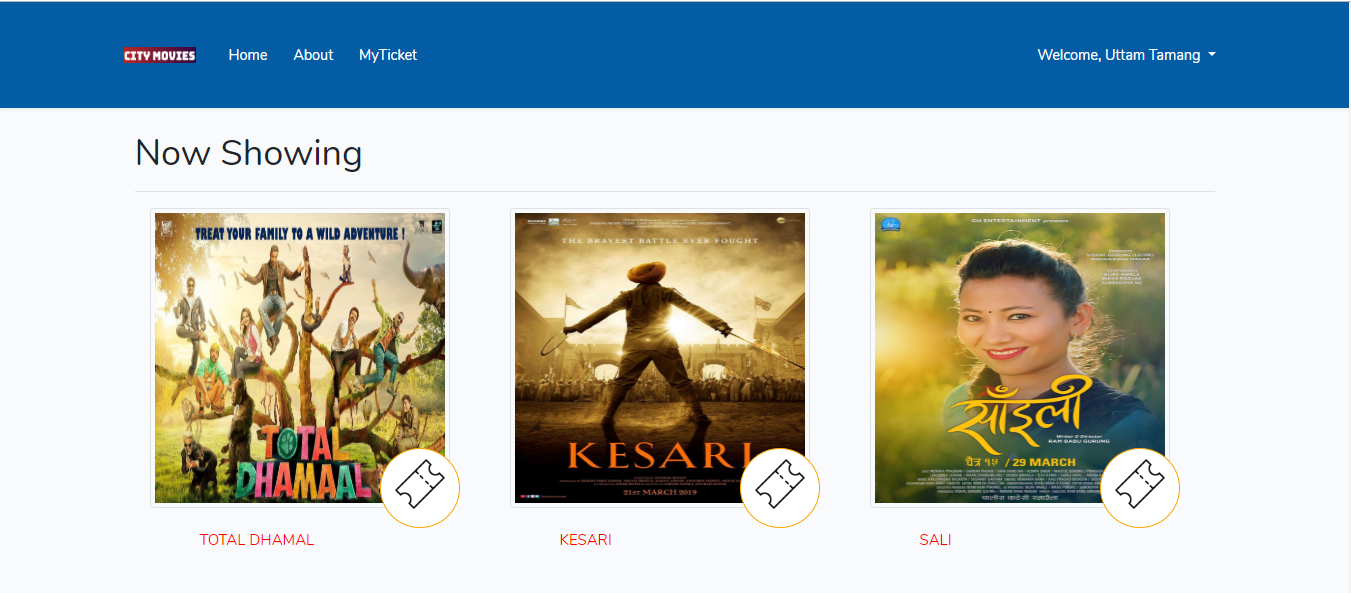


Figure : (ii.) redirect to home after login

Figure : White Box Testing for Login

**Test No:** 2

**Purpose of test:** register to system and redirect to home page

**Test name:** test User

**Expected Result:** register and redirect to Login page

**Actual Result:** register Successfully and redirect to Login Page

**Test Status**: Pass

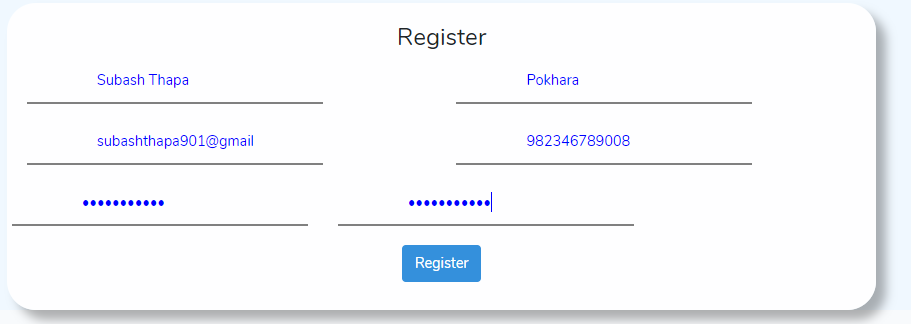


Figure : i) Black Box testing for User Register

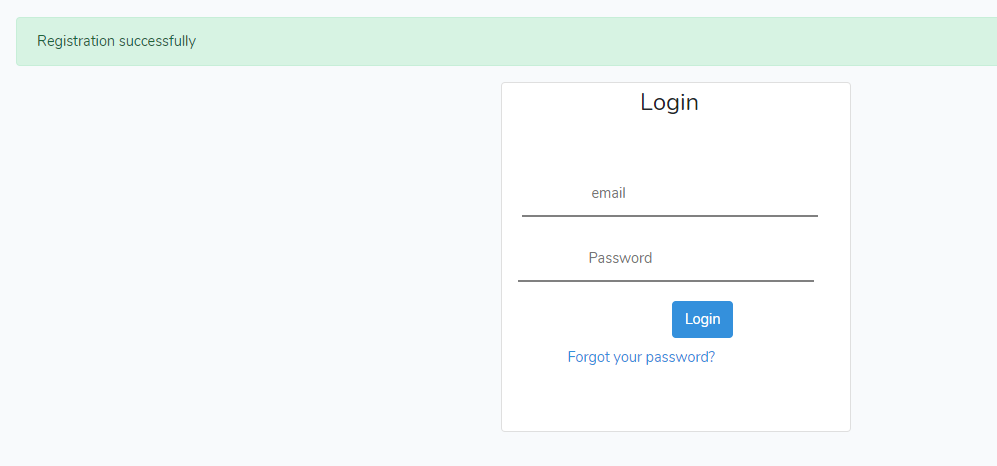


Figure : ii.) Redirecting to Login after User Register

**Test No:** 3

**Purpose of test:** Add Movie to the system

**Test name:** test Movie

**Expected Result:** add Movie and view on home page

**Actual Result:** Movie added Successfully and View to home page

**Test Status**: Pass

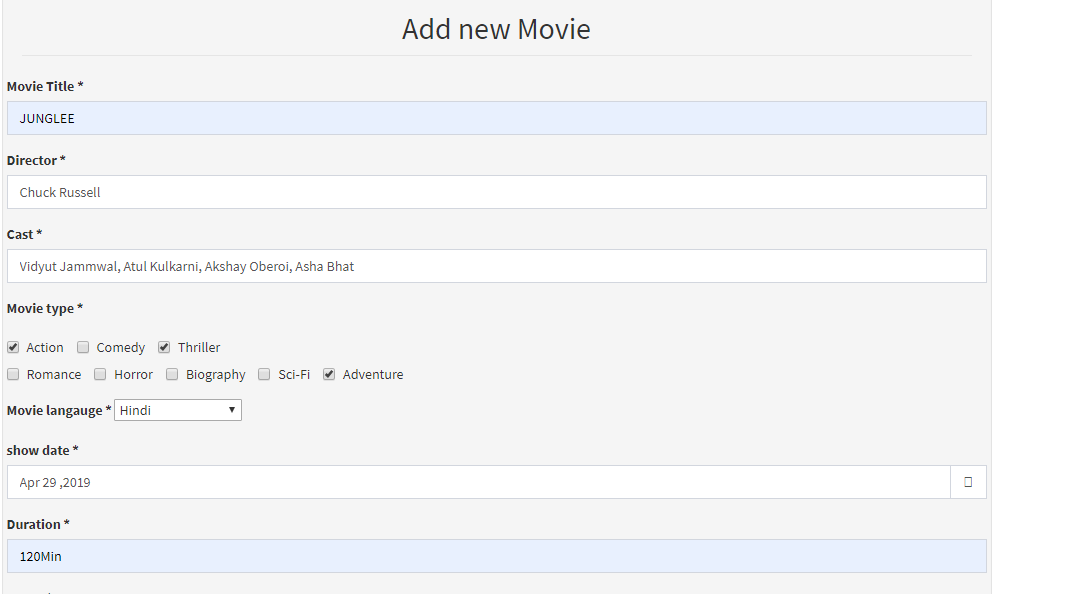


Figure : i) Black Box for Add Movie

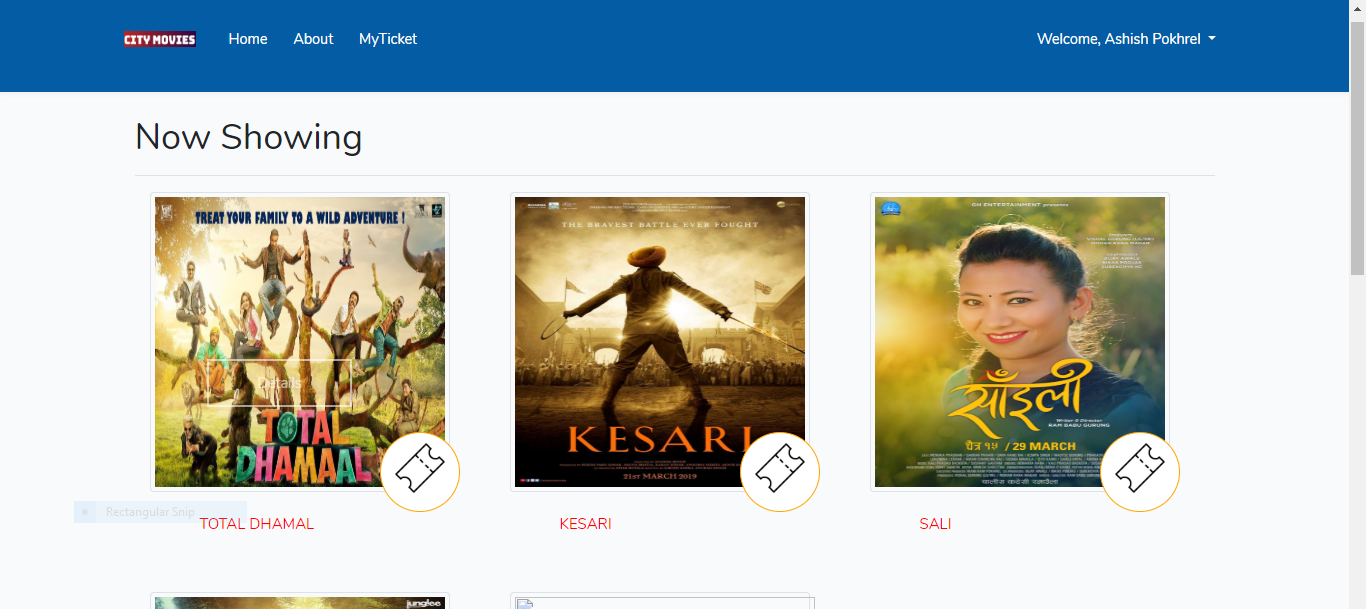


Figure : ii.) view Movie in home page

**Test No:** 4

**Purpose of test:** Book Movie ticket to

**Test name:** test book Movie ticket

**Expected Result:** book Movie Ticket and view on my ticket page

**Actual Result:** Book ticket Successfully and View to my ticket page

**Test Status**: Pass

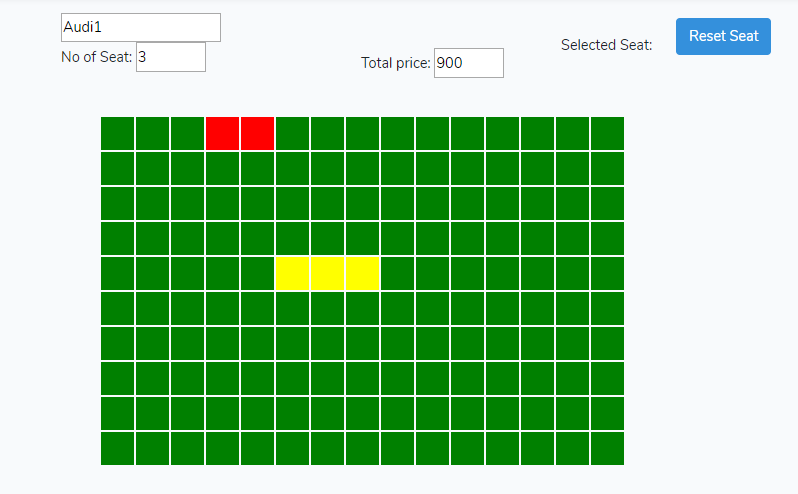


Figure : i) Blackbox testing for Choose Seat

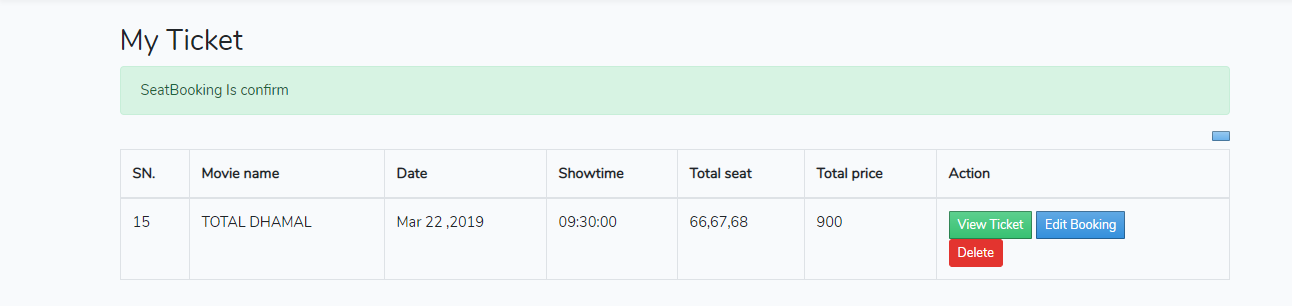


Figure : ii) View on my ticket page of book ticket

**Test No:** 5

**Purpose of test:** Add showtime

**Test name:** test showtime

**Expected Result:** add showtime and view on showtime page

**Actual Result:** show added Successfully and View to my showtime page

**Test Status**: Pass

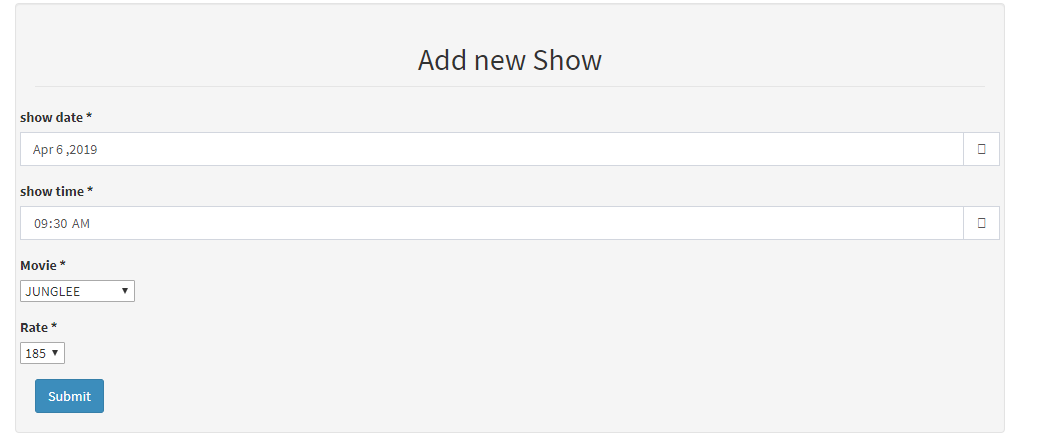


Figure : i.) testing for add show

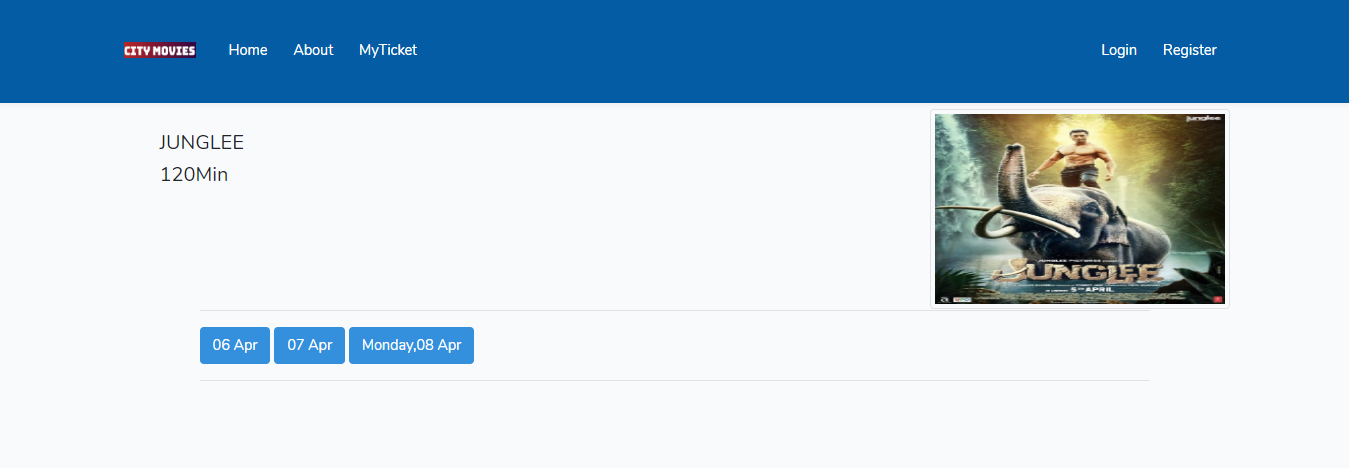


Figure : View in showtime after add show

**Test No:** 6

**Purpose of test:** Cancel Booking

**Test name:** test book

**Expected Result:** cancel booking

**Actual Result:** cancel booking Successfully

**Test Status**: Pass

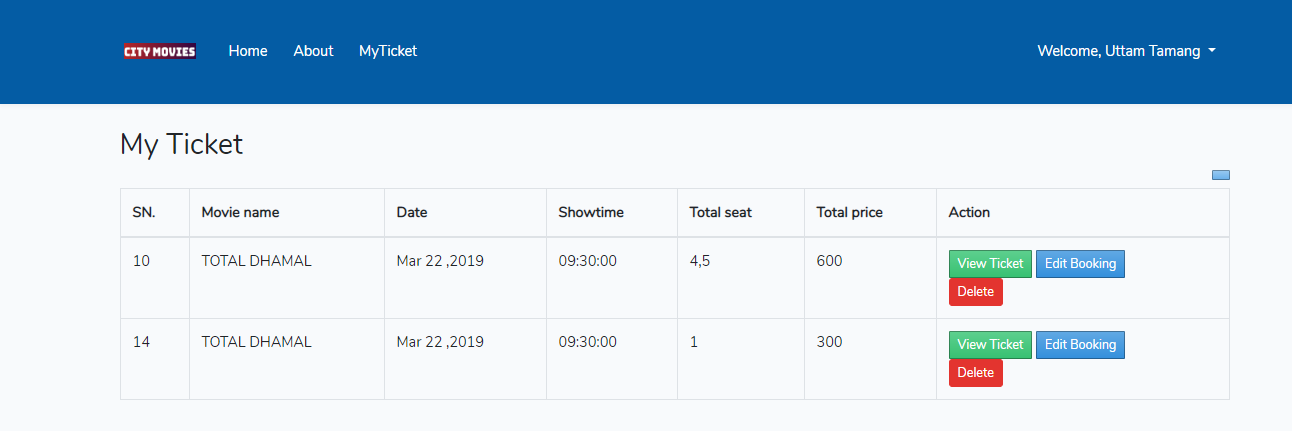


Figure : Black Box Testing for Cancel Booking

**Test No:** 7

**Purpose of test:** Access by Admin

**Test name:** test Admin Login

**Expected Result:** retrieve username and password and redirect admin

dashboard

**Actual Result:** Login Successfully and redirect to Admin dashboard

**Test Status**: Pass

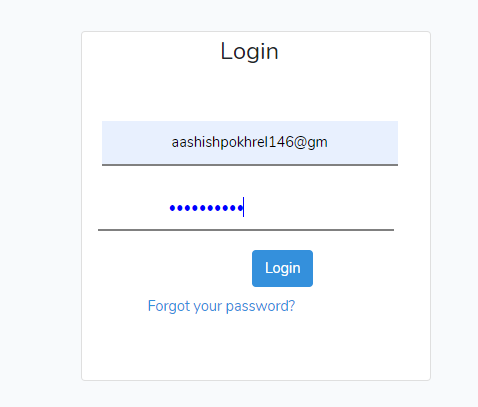


Figure : Black Box Testing for Admin Dashboard

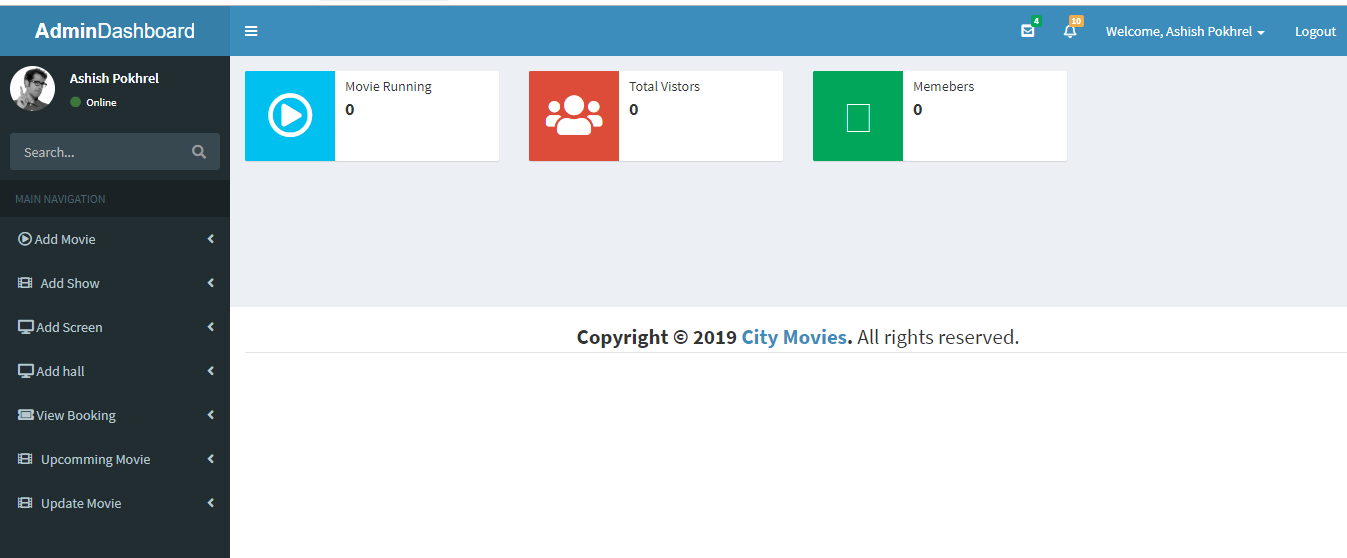


Figure : redirecting to admin dashboard after admin Login

**Test No:** 8

**Purpose of test:** Delete Movie from database

**Test name:** test Delete Movie

**Expected Result:** Movie should be deleted

**Actual Result:** Display with error message

**Test Status**: Fail

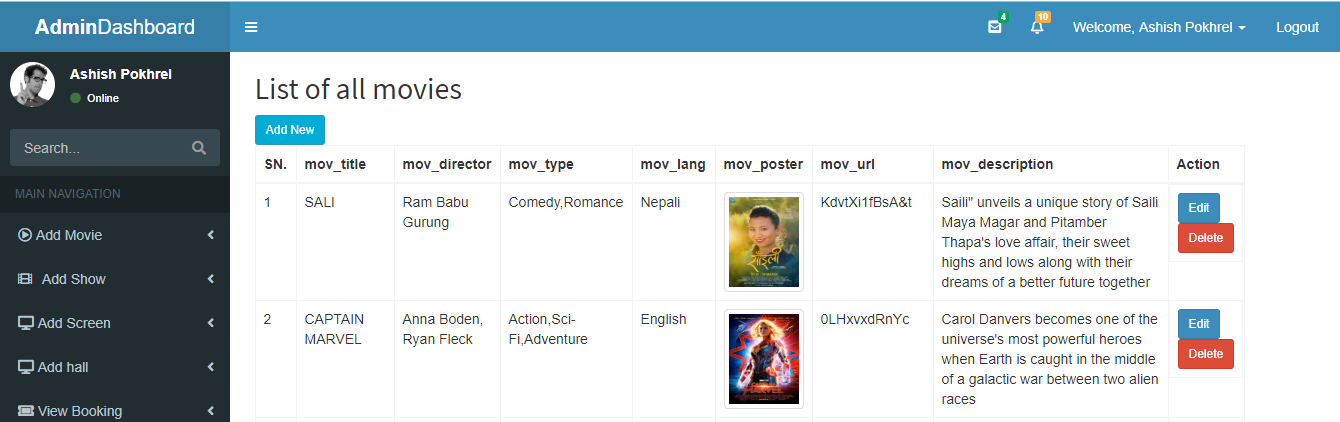


Figure i.) Black Boxing testing for delete movie in database

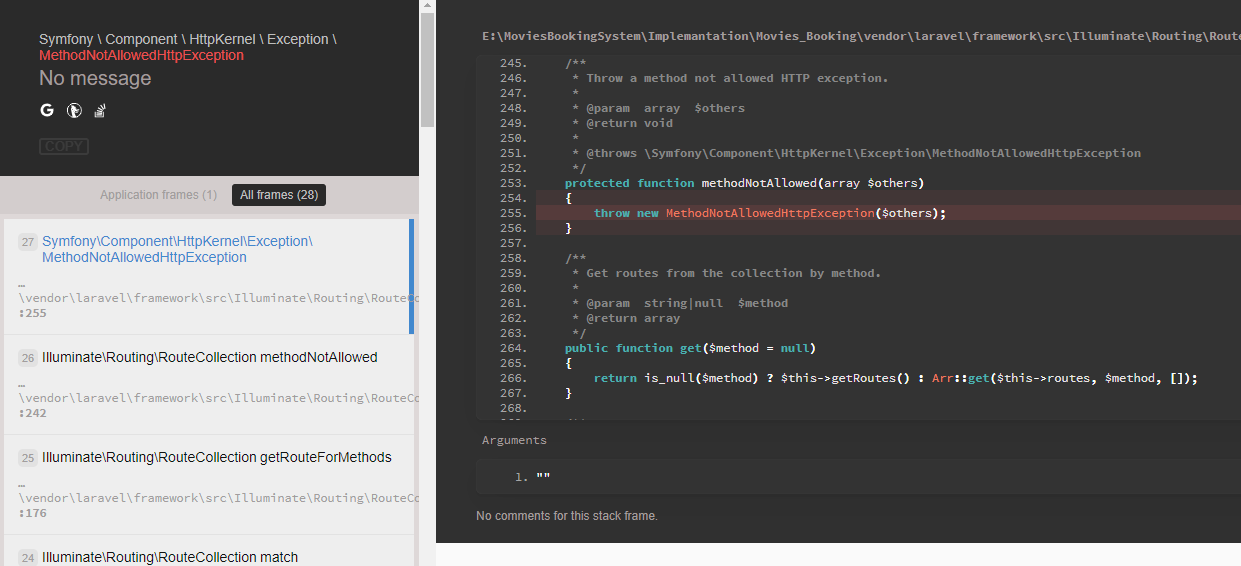


Figure ii.) Error during deleting movies

## 5.2 White Box Testing

White box testing is software testing technique, which checks the code and internal structural of code. In this type of testing, tester focus on the flows of input and output through the application so that tester can easily improve the either function is working or not and also check is performance too. It is also called as unit testing, structural testing , etc.

1. **Login**



Figure : White Box Testing for Login

**2.Register**

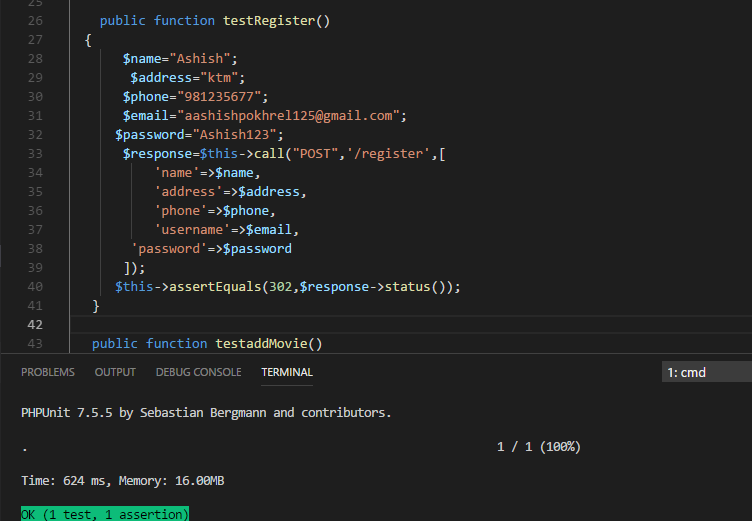


Figure : White Box testing for Register

1. **Add Movie**

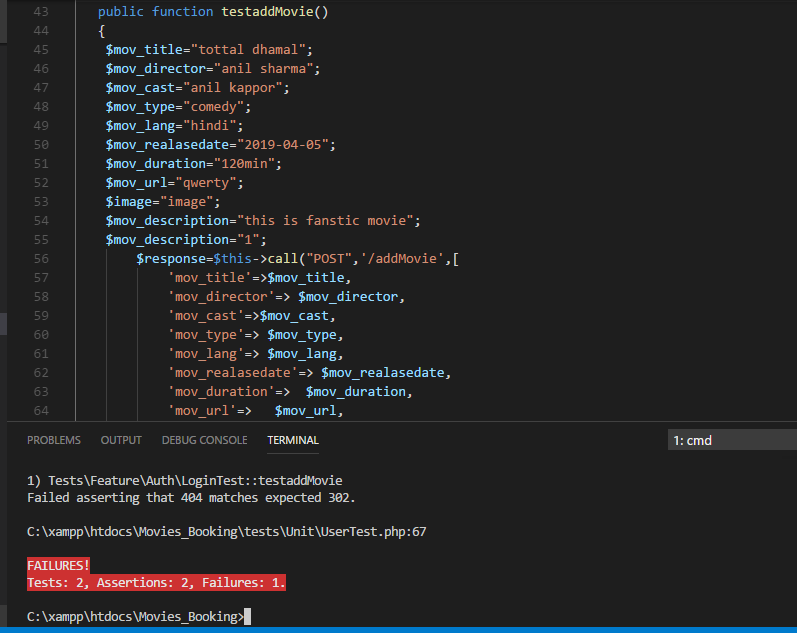


Figure : Unit Testing for Add Movies

**4 . Book seat**

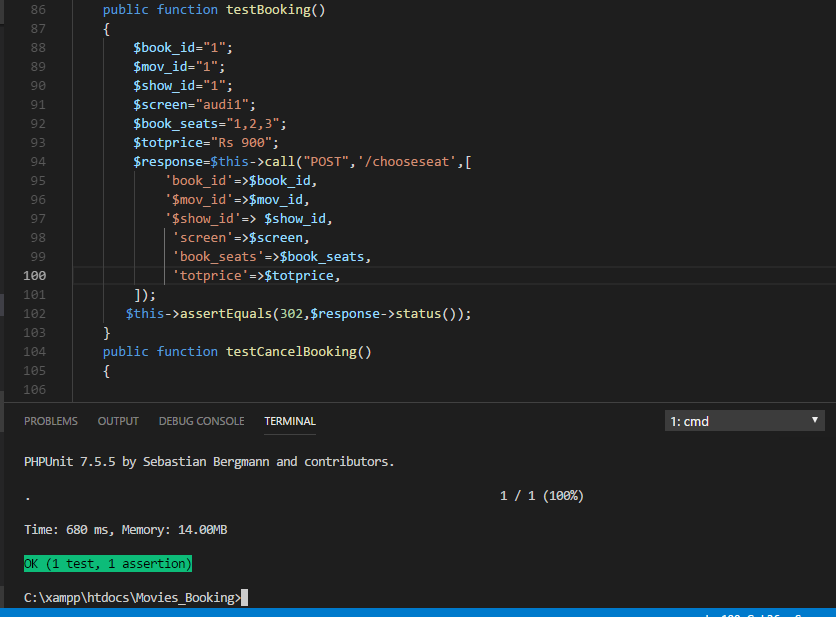


Figure : Unit Testing for Book seat

1. **Cancel Booking**

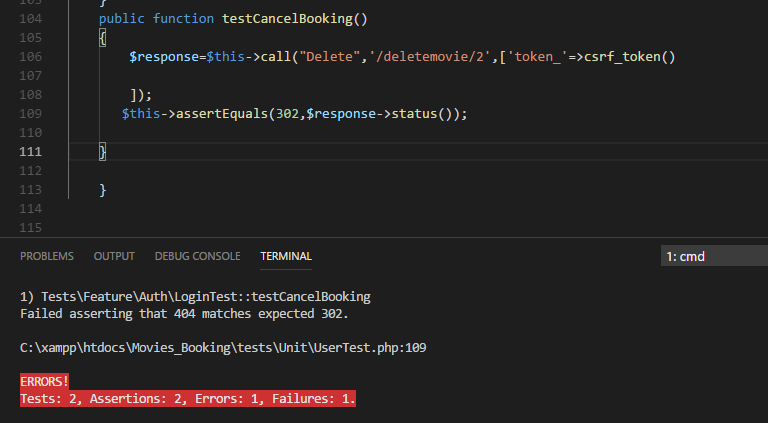


Figure :Unit Testing for cancel Booking

1. **Add Showtime**

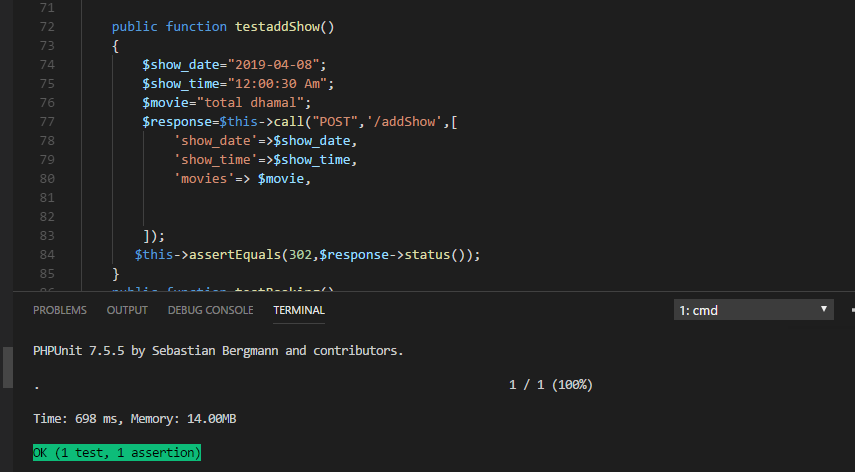


Figure : Unit Testing for Add showtime

1. **Delete Movie**

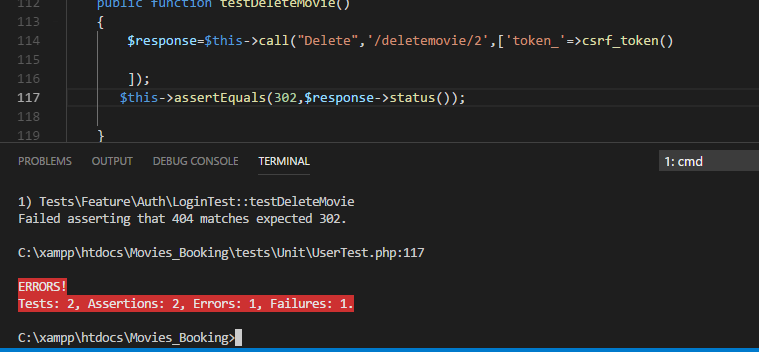


Figure : Unit Testing for Delete movie

1. **Add Screen**

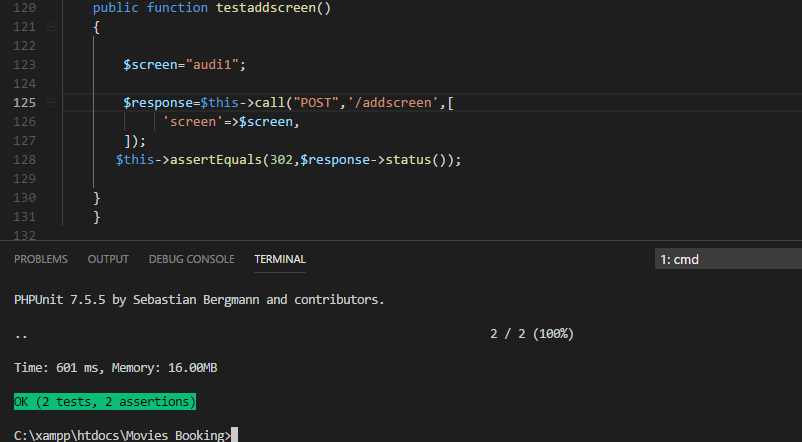


Figure : Unit Testing for Add screen

1. **Test for Edit Movies**

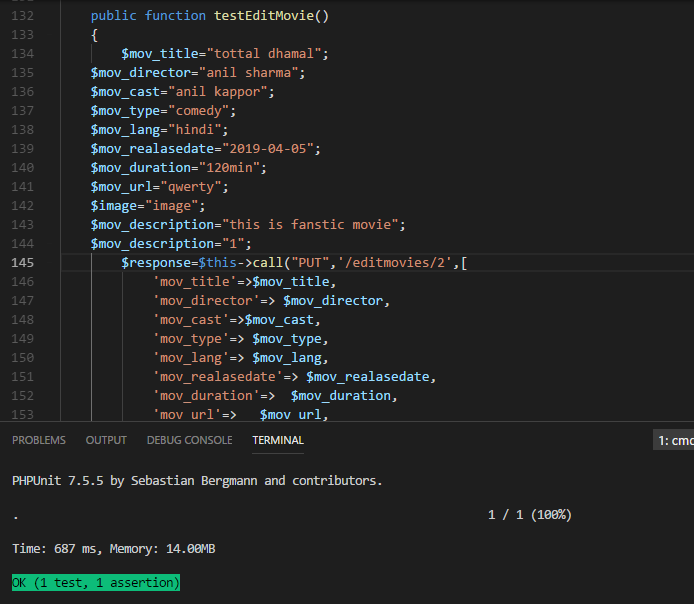


Figure : White Box Testing for edit Movie

1. **Test for Edit ShowTime**

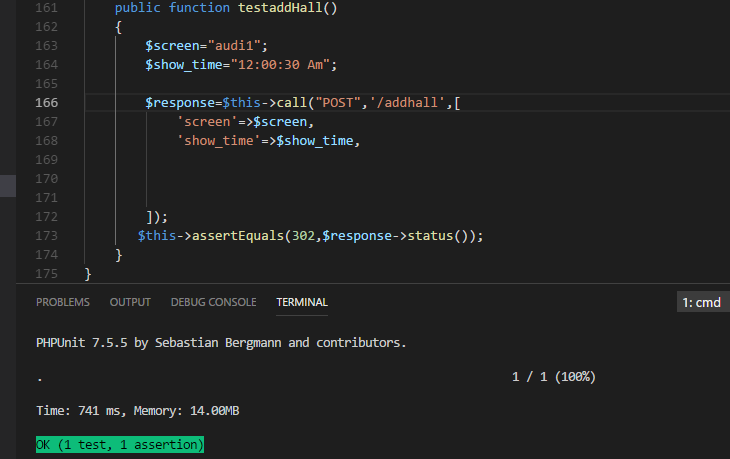


Figure : White Box Testing fpr add hall

# Chapter 6: Other Project Issues

## 6.1 Risk Management

Software Development uses different technological advancements and also required high levels of knowledge for developing any software. Due to these factors, every software development has some uncertainly which is called as Risk. (CAST, 2019)

For success of any software project, the first and foremost things is that to identify risk that will create obstacles during software development. Project Risk is also known as Potential problem for any project. Risk is also the activity which can compromise the success of any software project. It can lose cost, time of software project.

In software development, Risk Management is the process of identifying, controlling and managing risks which are evolved before or during development process. Generally, there are three process for risk management i.e. are as follows: (Proffesionalqa, 2019)

1. Risk identification

This is the first steps of risk management where Potential risks are identified which may affect software during software development.

1. Risk Analysis

This is the second step of risk management where identified risks in first steps is analysis. Each risk is analysis to know their possible outcomes and prioritization.

1. Risk Control

This is third steps of risks management where risks are managed, control and remove for get success result.

**Impact = Likelihood \* Consequence**

Risk Likelihood values are shown as follows

|  |  |
| --- | --- |
| Likelihood | Value |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Risk Consequence values are shown below

|  |  |
| --- | --- |
| Consequence | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

Risk Consequence values are shown below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | Risks | Likelihood | Consequences | Impact | Solution |
| 1 | Insufficient resources | 2 | 3 | 6 | All required resources for the project should be collected. |
| 2 | Hard Disk Failure | 1 | 5 | 5 | Data must back up. |
| 3 | Requirement does not meet | 2 | 5 | 10 | Proper planning should be done. |
| 4 | Server Failure | 1 | 4 | 4 | Online Backup should be done. |
| 5 | Lack of skill | 1 | 3 | 3 | Training should provide for skills employees. |
| 6 | Scheduling problem | 2 | 4 | 8 | Divide tasks in different schedule |
| 7 | Lack of cost | 2 | 4 | 8 | Cost estimation should be done properly. |

## 6.2 Configuration Management

The term refers to the system which track hardware, software and related information of the system. Configuration management is involving practices of processing system changes systematically with updating system while maintain the system integrity. To achieve of goal of the system, configuration management should be implemented with details policies, procedures to manage to version. (Guru, 2019)

### 6.2.1Version Control

Version Control is a software tools that helps to manage different version of source code of developer’s teams. It can allow us to re-back whole project into previous state. It can easily find problem when someone modifies source code lastly. There are three types of version control i.e. Local version control, centralized version control and distributed version control. (Git, 2019)

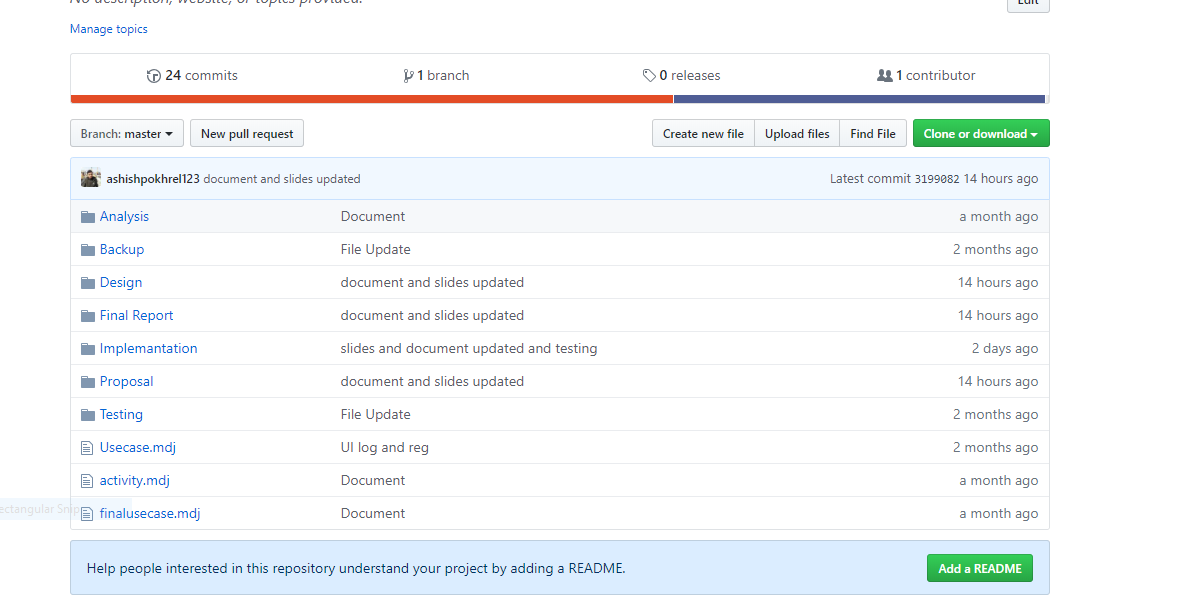


Figure : GitHub for Movie Booking

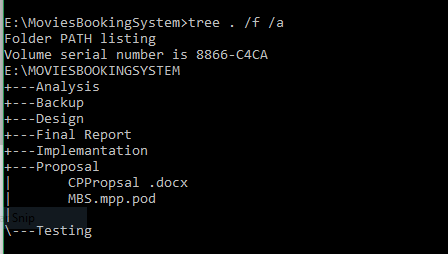


Figure : Tree structure

## Future Work

Given Project is academic so that I have provided with limited time, so that I can not include all features which are told in proposal. These features will come in future update.

* Online Payment System
* Rating and Reviewing Movie
* Security Features
* Live Chat Features

## User Manual

## Limitation

Following are the limitation of project:

* Online Payment System is not available.
* User can only cancel booking not to update booking.

# Chapter 7: Conclusion of the project

Online Movies Booking system is a customized and user friendly a web application where you can book seat for movies in theaters from anywhere anytime. It has facilities of booking movies, read reviews of movies, watch trailer, see lists of upcoming movies. Admin can add lists of movies. Suitable breakdown and scheduling are done properly. Design pattern MVC and waterfall methodology is used for the project. The only one limitation of the system is online payment system is not available.

# Chapter 8: Reference and Bibliography

CAST. (2019). Retrieved from https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects

Geeks. (2019). Retrieved from https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/

Git. (2019). Retrieved from https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Guru. (2019). Retrieved from https://www.guru99.com/database-design.html

Guru. (2019). Retrieved from https://www.guru99.com/software-testing-introduction-importance.html

Guru. (2019). Retrieved from https://www.guru99.com/software-configuration-management-tutorial.html

Proffesionalqa. (2019). Retrieved from http://www.professionalqa.com/risk-management-activity

ReQst. (2019). Retrieved from https://reqtest.com/requirements-blog/requirements-analysis/

requirement.com. (2004). Retrieved from http://www.requirements.com/Glossary/RequirementsPrioritization/tabid/121/Default.aspx

SDC. (2011). Retrieved from http://sdc.net.au/services/application-development/analysis-and-specification.aspx

SHARMA, L. (2018). Retrieved from https://www.toolsqa.com/software-testing/waterfall-model/

Sourcemaking. (2019). Retrieved from https://sourcemaking.com/design\_patterns

Techopedia. (2018). Retrieved from https://www.techopedia.com/definition/25813/use-case

TechTarget. (2019). Retrieved from https://searchsoftwarequality.techtarget.com/definition/development-environment

UML. (2019). Retrieved from https://www.uml-diagrams.org/uml-24-diagrams.html

Visual-Paradigm. (2019). Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

# Chapter 9: Appendix

Implementation of Laravel MVC design pattern

**Model**

1. **User Model**

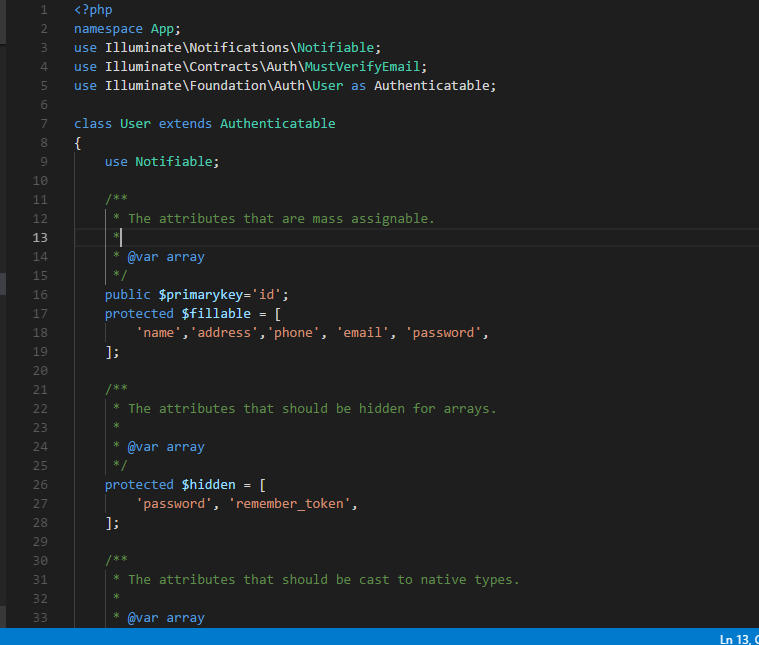


Figure : User Model

1. Movie Model

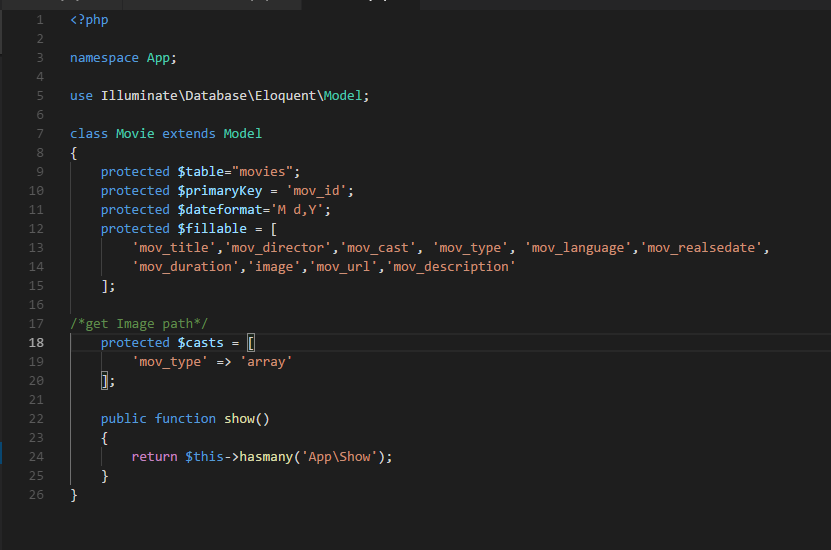


Figure : Movie Model

1. Show Model

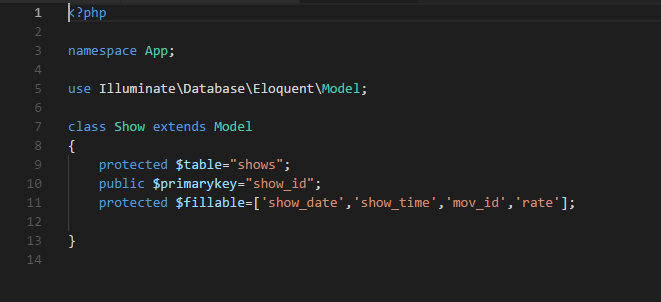


Figure : Show Model

Iv) Booking Model



Figure : Booking Model

V) Screen Model

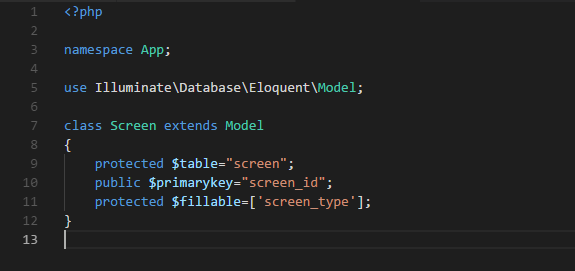


Figure : Screen Model

Vi) Hall Model

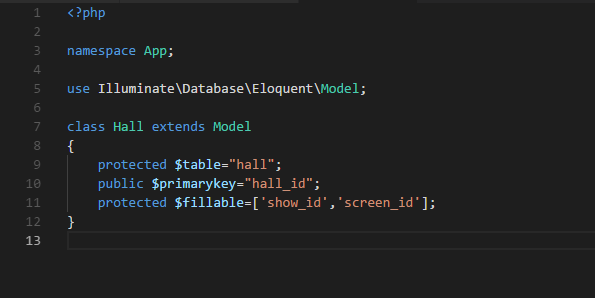


Figure : Hall Model

**Controller**

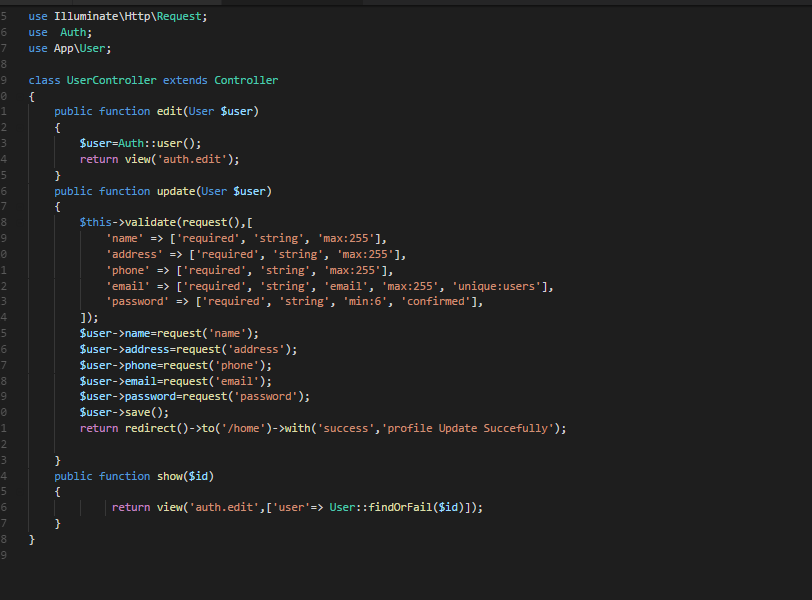
1. **User Controller**

Figure : User Controller

Ii) Movie Controller

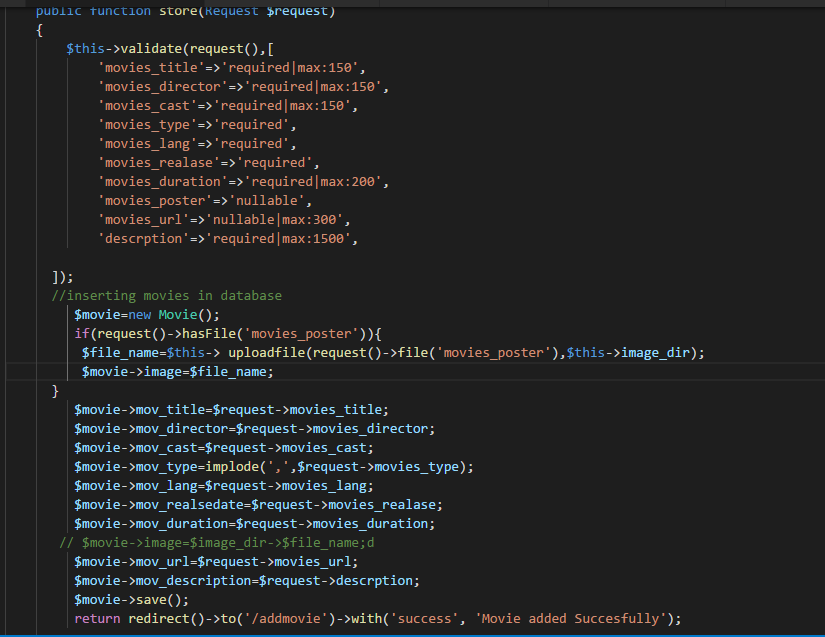


Figure : (i) Movie Controller

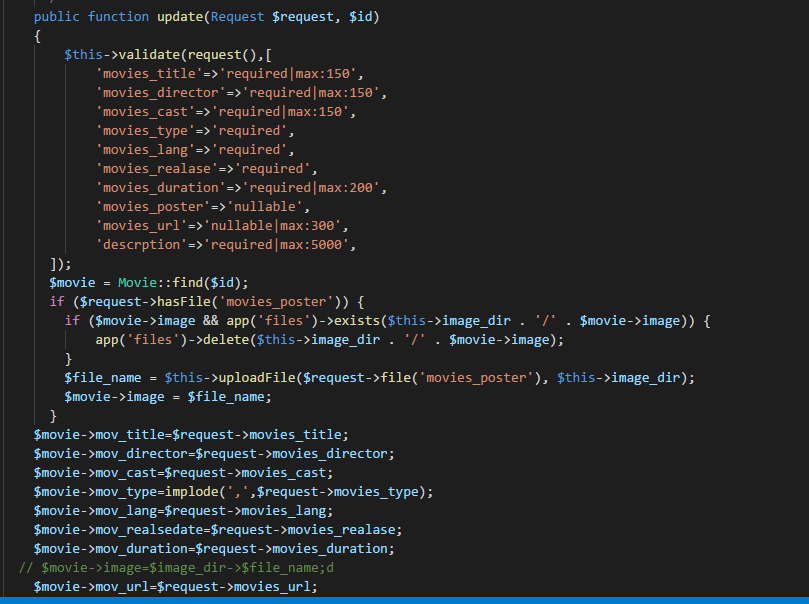


Figure : (ii) Movie Controller

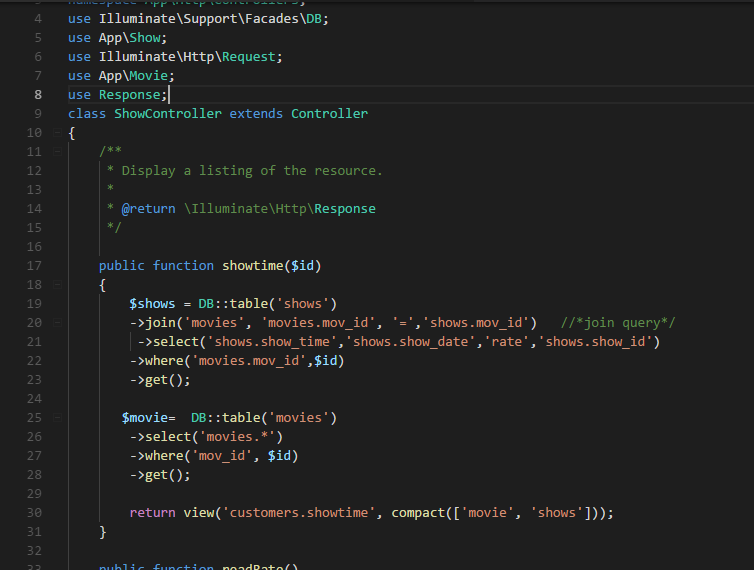
Iii) Show Controller

Figure :)(i) Show Controller



Figure : (ii). Show Controller

Iii) Seat Controller



Figure : Seat Controller

1. Ticket Controller

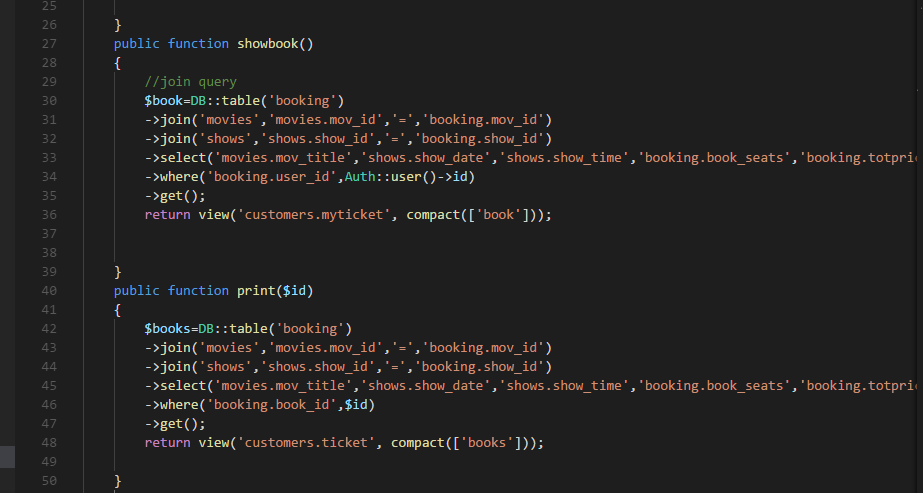


Figure : (i) Ticket Controller

V) Hall Controller

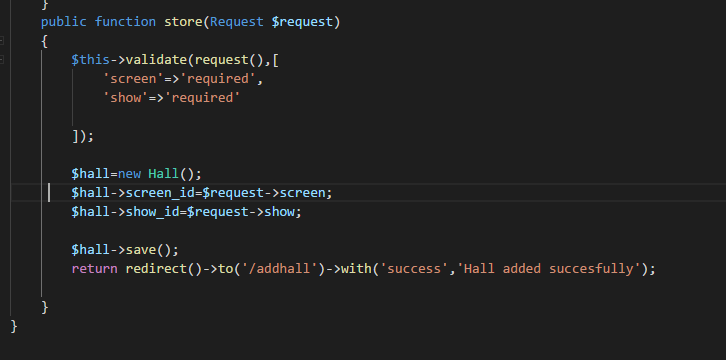


Figure : Hall Controller