

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in

Information Technology

By

KUSHAL GOYAL M

(Reg. No. 38120043)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119 TAMILNADU.



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY



(DEEMED TO BE UNIVERSITY)

(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119
www.sathyabamauniversity.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **KUSHAL GOYAL M (Reg. No.38120043)** who underwent the professional training in “**ONLINE MOVIE TICKET BOOKING SYSTEM**” under our supervision from Jan 2019 to April 2019.

Internal Guide

Dr. MARY GLADENCE.L, Ph.D.,

Head of the Department

Dr. R. SUBHASHINI, Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, KUSHAL GOYAL M (Reg. No. 38120043) hereby declare that the Professional Training Report on “**ONLINE MOVIE TICKET BOOKING SYSTEM**” done by me under the guidance of Dr.Mary Gladence.L, Ph.D., at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor Technology degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. R. SUBHASHINI Head of the Department of Computing for providing me necessary support and details at the right time during the progressive views.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr. MARY GLADENCE. L for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-Teaching staff members of the Department of Computing who were helpful in many ways for the completion of my project.



Zend Technologies, Inc.



Kushal Goyal M

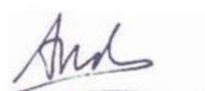
Has successfully completed the test requirements
and demonstrated the knowledge and skills
to be recognized as a

Zend Certified Engineer
PHP 5.3

Certificate ID- ZEND740686

Registration ID- 246195180


Zeev Suraski
Co-Founder


Andi Gutmans
Co-Founder

To Whomever It May Concern,

This is to certify that **Kushal Goyal M** S/O-Mr. Mukesh Goyal a student of Information Technology, **SATHYABAMA UNIVERSITY**, Chennai, India has successfully completed a 3 (Three) days (From 21st August to 23rd August, 2020) internship program at Mumbai. During the period of his internship with us he demonstrated good designing skills with a self motivated attitude to learn new things. He worked with us in briefing our clients encompassing the wide gamut of services.

The project topic that was undertaken by him during this internship is titled on;

1. **Custom Portals**
2. **Web Development**
3. **Internet Marketing**

As a part of the pitch, he also designed various banners, mailers by understanding the design briefs from our clients.

We wish him all the best for his future endeavors.

Warm regards,



Sanjeev Shakya

ABSTRACT

Today in this present era where technological advances are its vertex; there is not even a single sector which remains untouched by technology. Technology has not merely established simpler live, but also offers a high level of safety and security wherever required. Technological devices are available which provides refuge for the mankind.

Every organization, whether big or small, has challenges to overcome and managing the information of Tickets, Movie, Seats, Cinema Halls, Show timing. Every Online Movie Ticket Booking System has different Movie needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

The aim for developing this module is to manage the show and provide all the functionality related to movie. Movie Module is an important module in this project Online Movie Ticket Booking System which is developed on MySQL. It tracks all the information and all type of CRUD (Create, Read, Update and Delete) operations of the movie.

Highlights of Movie Module:

- Customer can see and book shows.
- Admin can see the list of movie details.
- Only admin can edit, update and delete the records of the movie.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	Abstract	vi
	List of Figures	ix
1	INTRODUCTION	1
	1.1 Outline Of The Project	
	1.2 Purpose Of The Project	
	1.3 Problem In Existing System	
	1.4 Proposed System	
2	PROJECT DESIGNS	4
	2.1 UML Diagram	
	2.1.1 Use Case Diagram	
	2.2 System Architecture	
	2.2.1 Block Diagram	
3	PROJECT IMPLENTATIONS	6
	3.1 Introduction	
	3.2 Hardware Requirement	
	3.3 Software Requirement	
	3.4 Working Explanation	
4	IMPLEMENTED SCREENSHOTS	27
5	CONCLUSION	32

Chapter No.	TITLE	Page No.
	APPENDIX	33
	A. SOURCE CODE	
	REFERENCES	48

LIST OF FIGURES

Figure		Page
No.	TITLE	No.
2.1.1	Manual Process Diagram	2
2.2.1	RDBMS Diagram	4
3.4.1	MVC Diagram	5
3.4.2	System Flow Chart	7
3.4.3	Case Diagram - Admin and Module	8
3.4.4	Case Diagram - Admin, User and Module	9
3.4.5	Class Diagram	10
3.4.6	Login Info – Explanatory Tabulation	11
3.4.7	Module Movie Info – Explanatory Tabulation	12
3.4.8	Module Booking Info – Explanatory Tabulation	13
3.4.9	Module Ticket Info – Explanatory Tabulation	14
3.4.10	PERT Chart	19
3.4.11	GANTT Chart	20

CHAPTER 1

INTRODUCTION

1.1 Outline of the Project

The Online Movie Ticket Booking System can automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

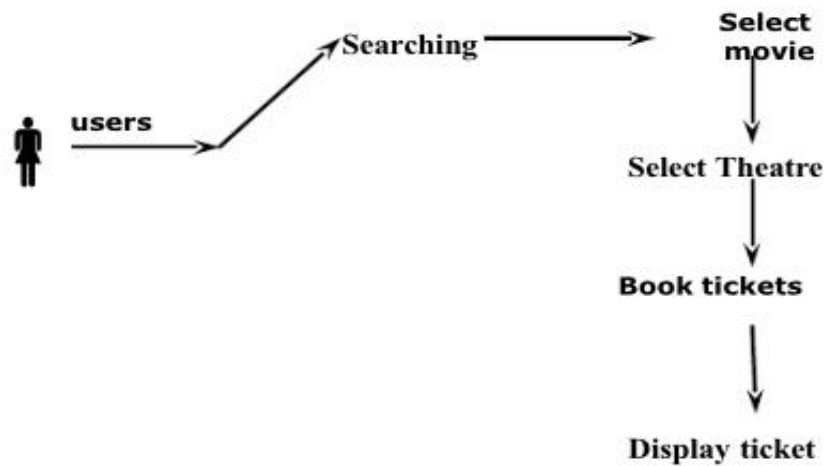
Online Movie Ticket Booking System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

1.2 Purpose of project

The main objective of the Project on Online Movie Ticket Booking System is to manage the details of Movie, Tickets, Booking Customer, Seats, Show timing. It manages all the information about Movie, Cinema Halls, Show timings, Movie. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Movie, Tickets, Cinema Halls, Booking Customer. It tracks all the details about the Booking Customer, Seats, Show timing.

1.3 Existing systems

Manual Process



Digital Process

Present Online Systems are developed in asp. In asp there are some limitations.

Disadvantage

- 1) It is scripting language.
- 2) It is very code complex.
- 3) There are server controls in asp
- 4) It doesn't support language interoperability

1.4 Proposed system

The computerization of this system would avoid the wrong interpretation and bad calculation of data. The system help the user to see any documents, source code, tasks, activities, team information with details at the click of a button. The record data is maintained and backed up such a way that data is not loss. The speed of the system could also increase.

Advantage

- Creating & Changing Issues at ease.
- User Accounts to control the access and maintain security.
- Multi-level Priorities & Severities.
- It contain better storage capacity.
- Accuracy in work.
- Well designed reports.
- Access of any information individually.
- Work becomes very speedy.
- User Accounts to control the access and maintain security.
- Query Issue List to any depth

CHAPTER 2

PROJECT DESIGN

2.1 UML DIAGRAM

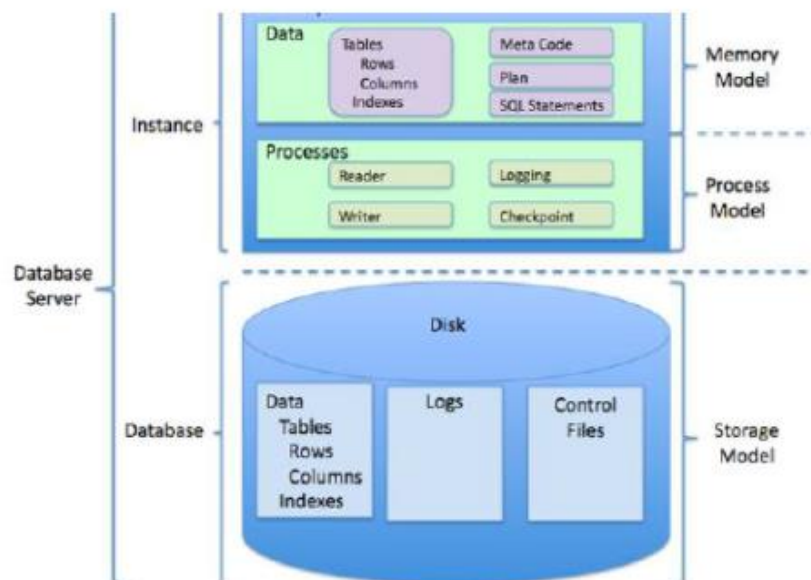
2.1.1 Related Database Management System

Relational Database Management System (RDBMS): This is an RDBMS based project which is System.

Brief Introduction about RDBMS:

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMS have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use. However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object-relational impedance mismatch in relational database, and XML databases.



2.2 Implementation Methodology:

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

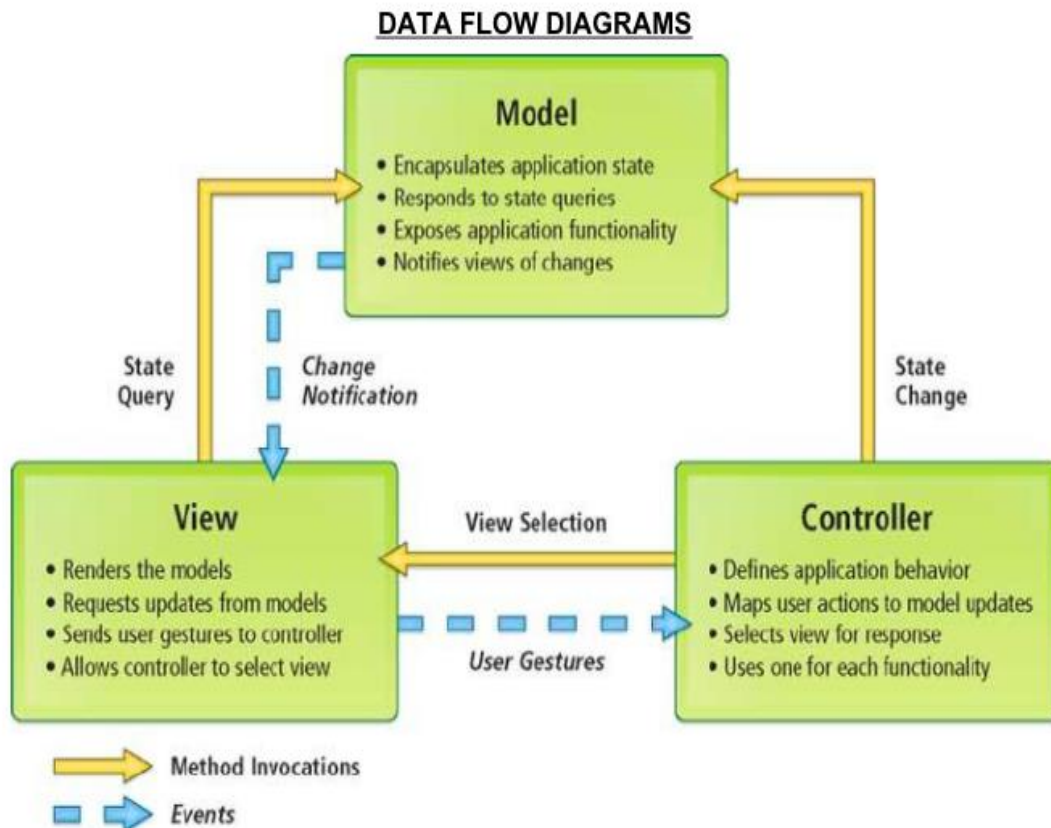
Model - The lowest level of the pattern which is responsible for maintaining data.

View - This is responsible for displaying all or a portion of the data to the user.

Controller - Software Code that controls the interaction between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response.

MVC (Model View Controller Flow) Diagram



CHAPTER 3

PROJECT IMPLEMENTATION

3.1 Introduction

A logical system is built which fulfils the given requirements. Implementational phase of project development deals with transforming the client's requirements into a logically working system. Normally, project is performed in the following two steps:

3.2 Hardware Requirements

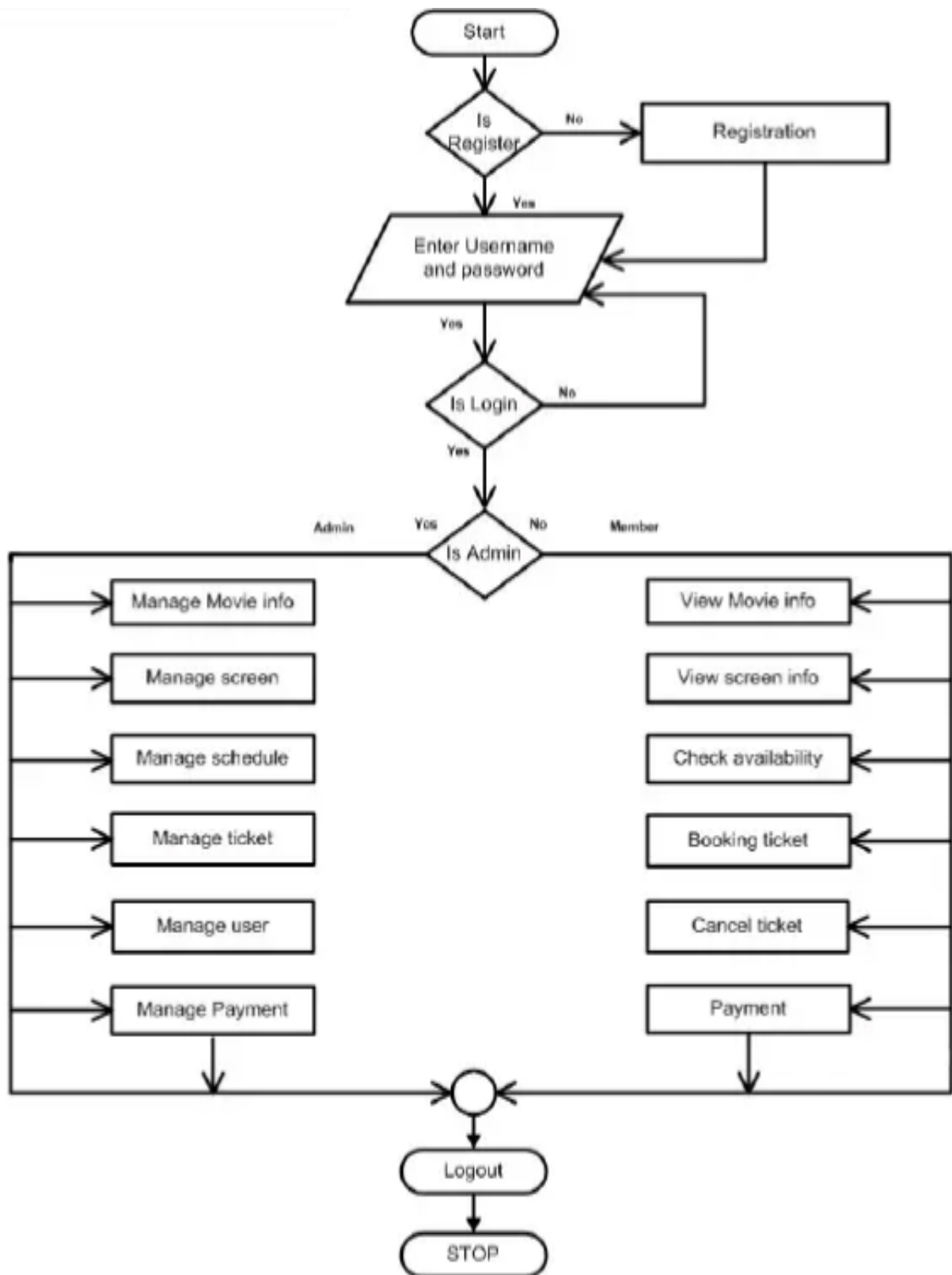
- RAM : 8GB
- HARD DISK : MINIMUM 20GB

3.3 Software Requirements

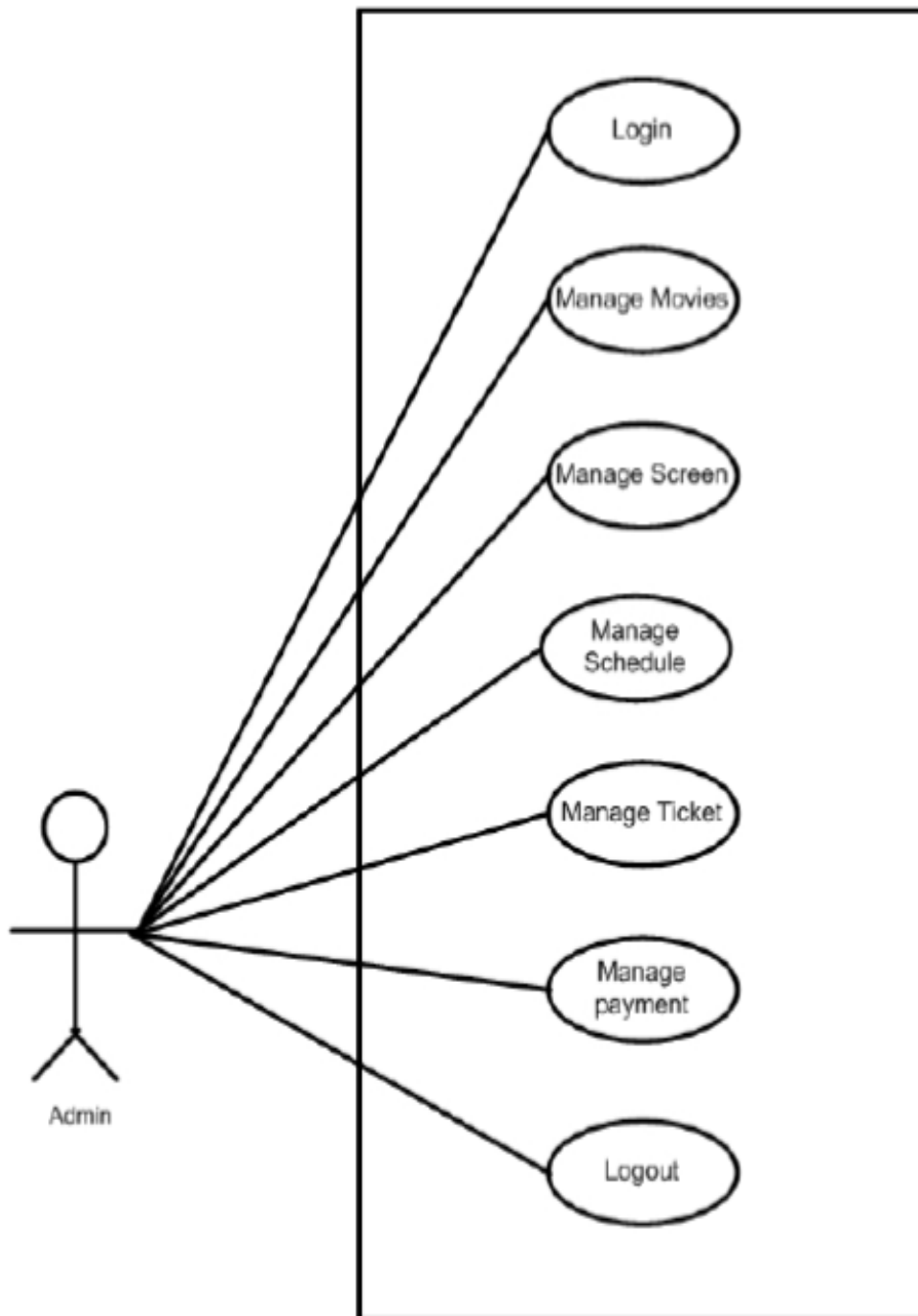
- OPERATING PLATFORM : macOS Mojave Version 10.14.2
- RDBMS : SQLSERVER 2000
- SOFTWARE : VS.NET 2003
- FRONT END TOOL : CSS
- BACK END TOOL : PHP
- PROGRAMMING TOOL : VISUAL STUDIO 2019

3.4 Working Explanation

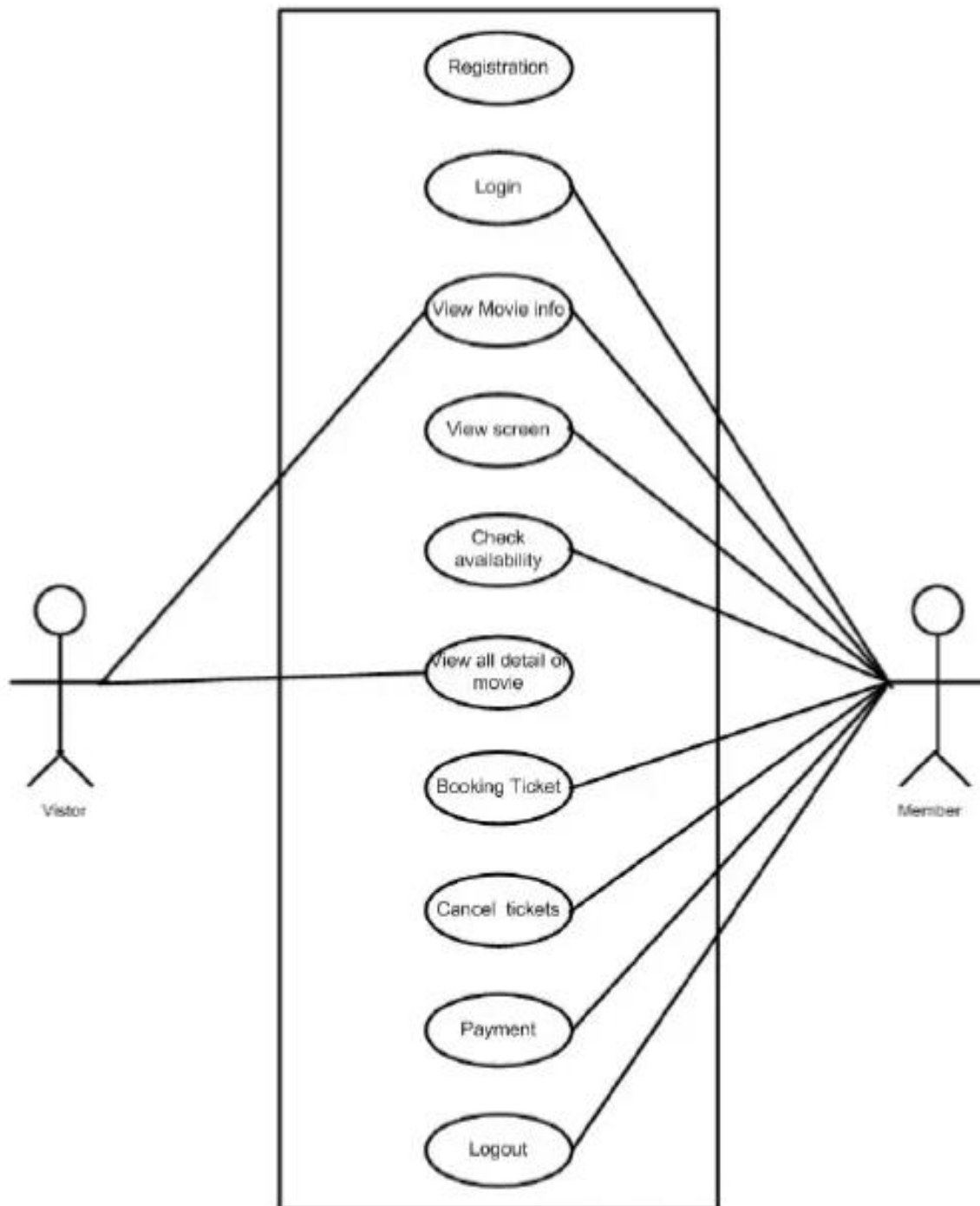
SYSTEM FLOW CHART



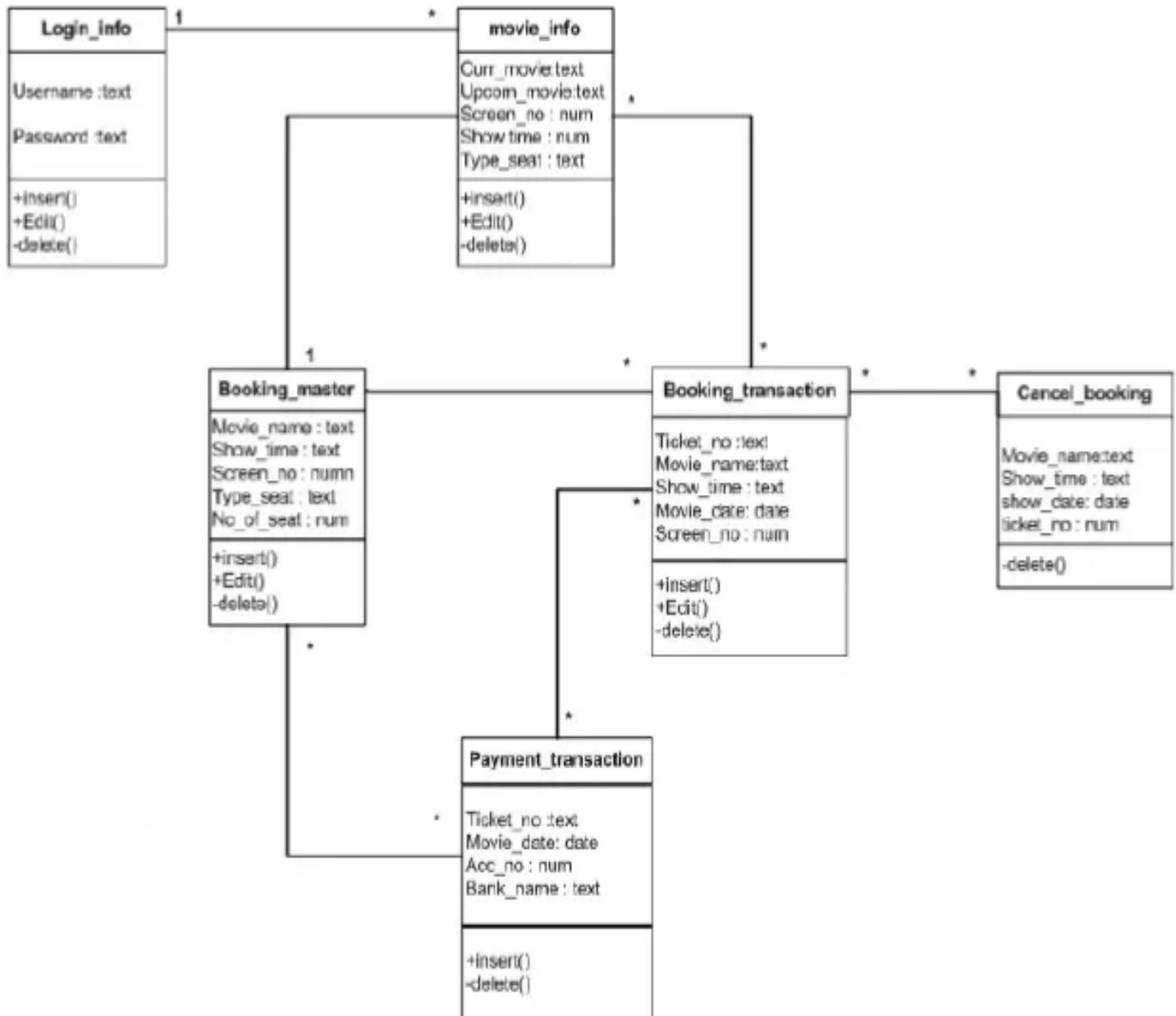
CASE DIAGRAM BETWEEN ADMIN AND MODULE



CASE DIAGRAM BETWEEN ADMIN, USER AND MODULE



CLASS DIAGRAM



DATA DICTIONARY

Module Admin and User Login Info:

Table Name :- login_info

Description :- To store login information.

Primary Key :-

Foreign Key :-

Index	Column Name	Data Type	Size	Constraints	Description
1	UserName	Varchar	(50)	Not Null	Used to store Username
2	Password	Varchar	(50)	Not Null	Used to store Password

Module Movie Info:

Table Name :- Movie

Description: - To store movie information.

Primary Key: - MovieId

Foreign Key: -

Index	Column Name	Data Type	Size	Constraints	Description
1	MovieId	Int	---	Primary key	Used to store Movie Id.
2	MovieName	Varchar	(50)	Not Null	Used to store Movie Name.
3	ImagePath	Varchar	(100)	Not Null	Used to store Image Path.
4	Director	Varchar	(100)	Not Null	Used to store Director.
5	Producer	Varchar	(50)	Not Null	Used to store Producer
6	Cast	Varchar	(200)	Not Null	Used to store Cast
7	Duration	Int	---	Not Null	Used to store Duration
8	Story	Varchar	(8000)	Allow Null	Used to store Story
9	Type	Varchar	(50)	Not Null	Used to store Type (running or upcoming)

Module Booking Info:

Table Name :- Book

Description :- To store booking information

Primary Key :- StatusId

Foreign Key :- SeatId , ShowId

Index	Column Name	Data Type	Size	Constraints	Description
1	StatusId	Int	---	Primary key	Used to store StatusId
2	SeatId	Int	---	Foreign Key	Used to store (Foreign key of Seats's column SeatId)
3	SeatName	Int	---	Allow Null	Used to store Seat Name
4	ShowId	Int	---	Foreign Key	Used to store (Foreign key of Show's column ShowId)
5	Status	Bit	---	Not Null	Used to store the status

Module Ticket Info:

TableName :- Ticket

Description :- To store ticket information

Primary Key :- TicketId

Foreign Key :- ShowId

Index	Column Name	Data Type	Size	Constraints	Description
1	TicketId	Bigint	---	Primary key	Used to store Ticket Id
2	ShowId	Int	---	Foreign Key	Used to store (Foreign key of Show's column ShowId)
3	Seats	Varchar	(50)	Not Null	Used to store seats
4	SeatType	Varchar	(50)	Not Null	Used to store Seat type
5	NoSeats	Int	---	Not Null	Used to store No. of seats
6	Amount	Int	---	Not Null	Used to store amount of ticket
7	IsPaid	Varchar	(50)	Not Null	Used to store paid to customer
8	Mobile	Bigint	---	Not Null	Used to store mobile no of customer
9	Email	Varchar	(50)	Allow Null	Used to store email id of customer

Project Profile

There has been continuous effort to develop tools, which can ease the process of software development. But, with the evolving trend of different programming paradigms today's software developers are really challenged to deal with the changing technology. Among other issues, software re-engineering is being regarded as an important process in the software development industry. One of the major tasks here is to understand software systems that are already developed and to transform them to different software environment. Generally, this requires a lot of manual effort in going through a program that might have been developed by another programmer. This project makes a novel attempt to address the issue of program analysis and generation of diagrams, which can depict the structure of a program in a better way. Today, UML is being considered as an industrial standard for software engineering design process. It essential provides several diagramming tools that can express different aspects/ characteristics of program such as

Use cases: Elicit requirement from users in meaningful chunks. Construction planning is built around delivering some use cases and each interaction basis for system testing.

Class diagrams: shows static structure of concepts, types and class. Concepts how users think about the world; type shows interfaces of software components; classes shows implementation of software components.

Interaction diagrams: shows how several objects collaborate in single use case.

Package diagram: show group of classes and dependencies among them.

State diagram: show how single object behaves across many use cases.

Activity diagram: shows behaviour with control structure. Can show many objects over many uses, many object in single use case, or implementation methods encourage parallel behaviour, etc.

The end-product of this project is a comprehensive tool that can parse any vb.net program and extract most of the object oriented features inherent in the program such as polymorphism, inheritance, encapsulation and abstraction.

What is UML?

UML stands for Unified Modelling Language is the successor to the wave of Object Oriented Analysis and Design (OOA&D) methods that appeared in the late 80's. It most directly unifies the methods of Booch, Rumbaugh (OMT) and Jacobson. The UML is called a modelling language, not a method. Most methods consist at least in principle, of both a modelling language and a process. The Modelling language is that notation that methods used to express design.

Notations and Meta-models:

The notation is the graphical stuff; it is the syntax of the modelling language. For instance, class diagram notation defines how items are concepts such as class, association, and multiplicity is represented. These are:

Class Diagram: The class diagram technique has become truly central within object oriented methods. Virtually every method has included some variation on this technique. Class diagram is also subject to the greatest range of modelling concept. Although the basic elements are needed by everyone, advanced concepts are used less often. A class diagram describes the types of objects in the system and the various kinds of static relationship that exist among them. There are two principal kinds of static relationship:

- Association
- Subtype

Class diagram also show the attributes and operations of a class and the constraints that apply to the way objects are connected.

Association: Association represent between instances of class. From the conceptual perspective, association represents conceptual relations between classes. Each association has two roles. Each role is a direction on the association. A role also has multiplicity, which is an indication of how many object may participate in the given relationship.

Generalization: A typical example of generalization evolves the personal and corporate customer of a business. They have differences but also many similarity. The similarities can be placed in generalization with personal customer and corporate customer sub type.

Aggregation: aggregation is the part of relationship. It is like saying a car has engine and wheels as its parts. This sounds good, but difficult thing is considering, what is the difference is aggregation and association.

Interaction: Interaction diagrams are models that describes how groups of objects collaboration in some behaviour.

Typically, an interaction diagram captures the behaviour a single use cases. The diagram shows a number of example objects and the messages that are passed between these objects in use cases. These are following approaches with simple use case that exhibits the following behaviour.

Package Diagram: One of the oldest questions in software methods is: how do you break down a large system into smaller systems? It becomes difficult to understand and the changes we make to them.

Structured methods used functional decomposition in which the overall system was mapped as a function broken down into sub function, which is further broken down into sub function and so forth. The separation of process data is gone, functional decomposition is gone, but the old question is still remains. One idea is to group the classes together into higher-level unit. This idea, applied very loosely, appears in many

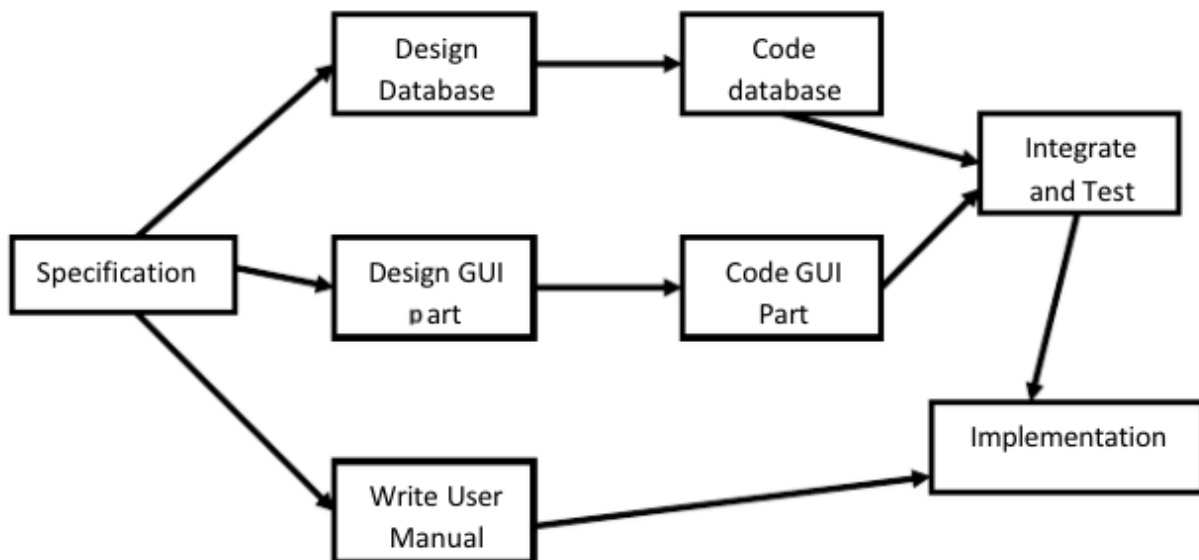
objects. In UML, this grouping mechanism is package. The term package diagram for a diagram that shows packages of classes and the dependencies among them.

A dependency exists between two elements if changes to the definition of one element may cause to other. With classes, dependencies exist for various reasons: one class sends a message to another, one class has another as part of its data; one class mentions another as a parameter to an operation. A dependency between two packages exists; and any dependencies exist between any two classes in the package.

State diagram: State diagram are a familiar technique to describe the behaviour of a system. They describe all the possible states a particular object can get into and how the objects state changes as a result of events that reach the objects. In most OO technique, state diagrams are drawn for a single class to show the lifetime behaviour of a single object. There are many form of state diagram, each with slightly different semantics. The most popular one used in OO technique is based on David Harel's state chart.

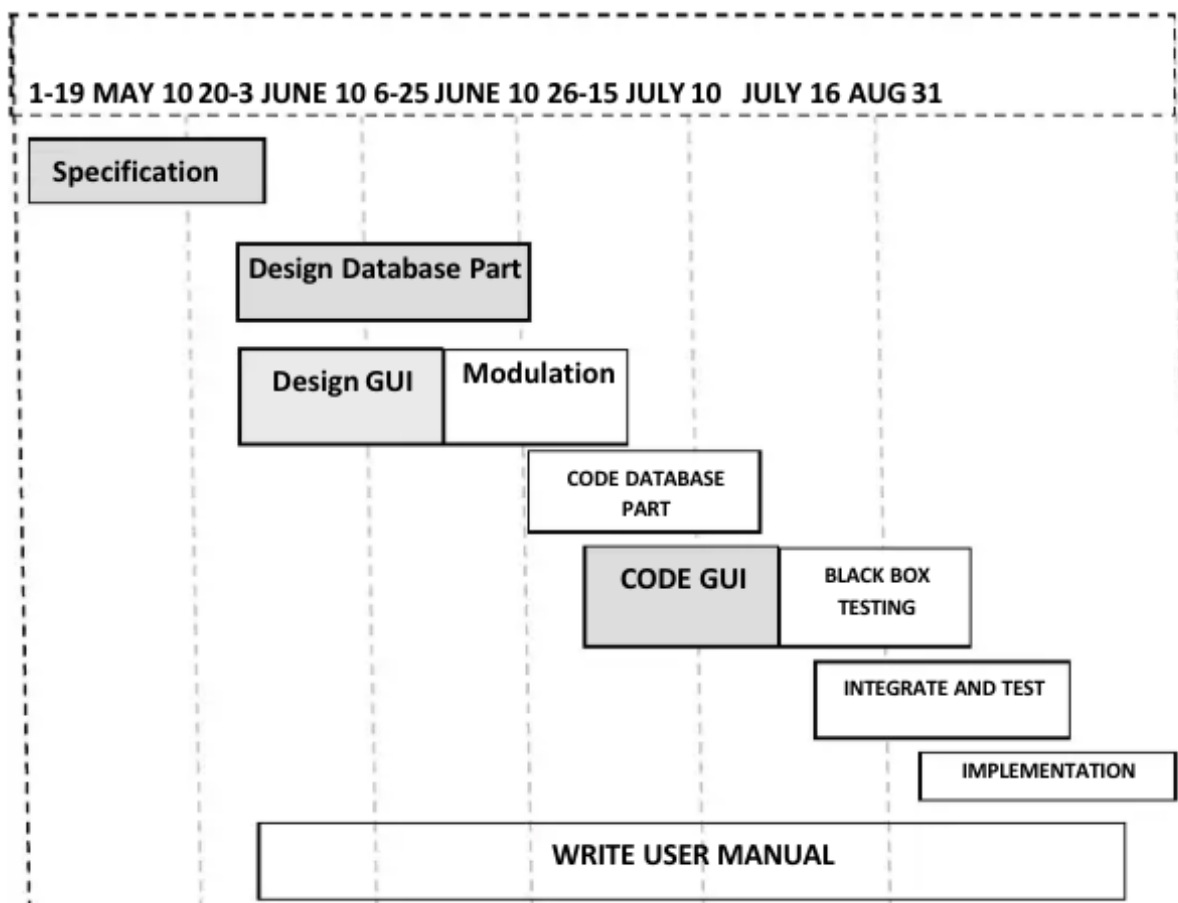
PERT CHART (Program Evaluation Review Technique)

PERT chart is organized for events, activities or tasks. It is a scheduling device that shows graphically the order of the tasks to be performed. It enables the calculation of the critical path. The time and cost associated along a path is calculated and the path requires the greatest amount of elapsed time in critical path.



GANTT CHART

It is also known as Bar chart is used exclusively for scheduling purpose. It is a project controlling technique. It is used for scheduling. Budgeting and resourcing planning. A Gantt is a bar chart with each bar representing activity. The bars are drawn against a time line. The length of time planned for the activity. The Gantt chart in the figure shows the Gray parts is slack time that is the latest by which a task has been finished.



Use Case Model of the Project:

The use case model for any system consists of "use cases". Use cases represent different ways in which the system can be used by the user. A simple way to find all the use case of a system is to ask the questions "What the user can do using the system?" The use cases partition the system behaviour into transactions such that each transaction performs some useful action from the users' point of view.

The purpose of the use case is to define a piece of coherent behaviour without revealing the internal structure of the system. An use case typically represents a sequence of interaction between the user and the system. These interactions consists of one main line sequence is represent the normal interaction between the user and the system. The use case model is an important analysis and design artifact (task). Use cases can be represented by drawing a use case diagram and writing an accompany text elaborating the drawing.

In the use case diagram each use case is represented by an ellipse with the name of use case written inside the ellipse. All the ellipses of the system are enclosed with in a rectangle which represents the system boundary. The name of the system being moduled appears inside the rectangle. The different users of the system are represented by using stick person icon. The stick person icon is normally referred to as an Actor. The line connecting the actor and the use cases is called the communication relationship. When a stick person icon represents an external system it is annotated by the stereo type<<external system>>

Project Testing

Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation. second phase is after the completion of software. this is system testing which verifies that the whole set of programs hanged together.

White Box Testing:

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. all logical parts of the software checked once. errors that can be corrected using this technique are typographical errors, logical within a module a logical decisions on their true and the false side are exercised, all loops and expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. When the box testing tests all the independent part bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

Black Box Testing:

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. black box testing tests the input, the output and the external data. it checks whether the input data is correct and whether we are getting the desired output.

Alpha Testing:

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

Beta Testing:

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

Unit Testing:

Each module is considered independently. it focuses on each unit of software as implemented in the source code. it is white box testing.

Integration Testing:

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. modules are integrated by using the top down approach.

Validation Testing:

Validation testing was performed to ensure that all the functional and performance requirements are met.

System Testing:

It is executing programs to check logical changes made in it with intention of finding errors, a system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

Implementation and Software Specification Testing

Detailed Design of Implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

Installation Phase

In this phase the new Computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

System Installation

The process of starting the actual use of a system and training user personnel in its operation.

System Analysis:

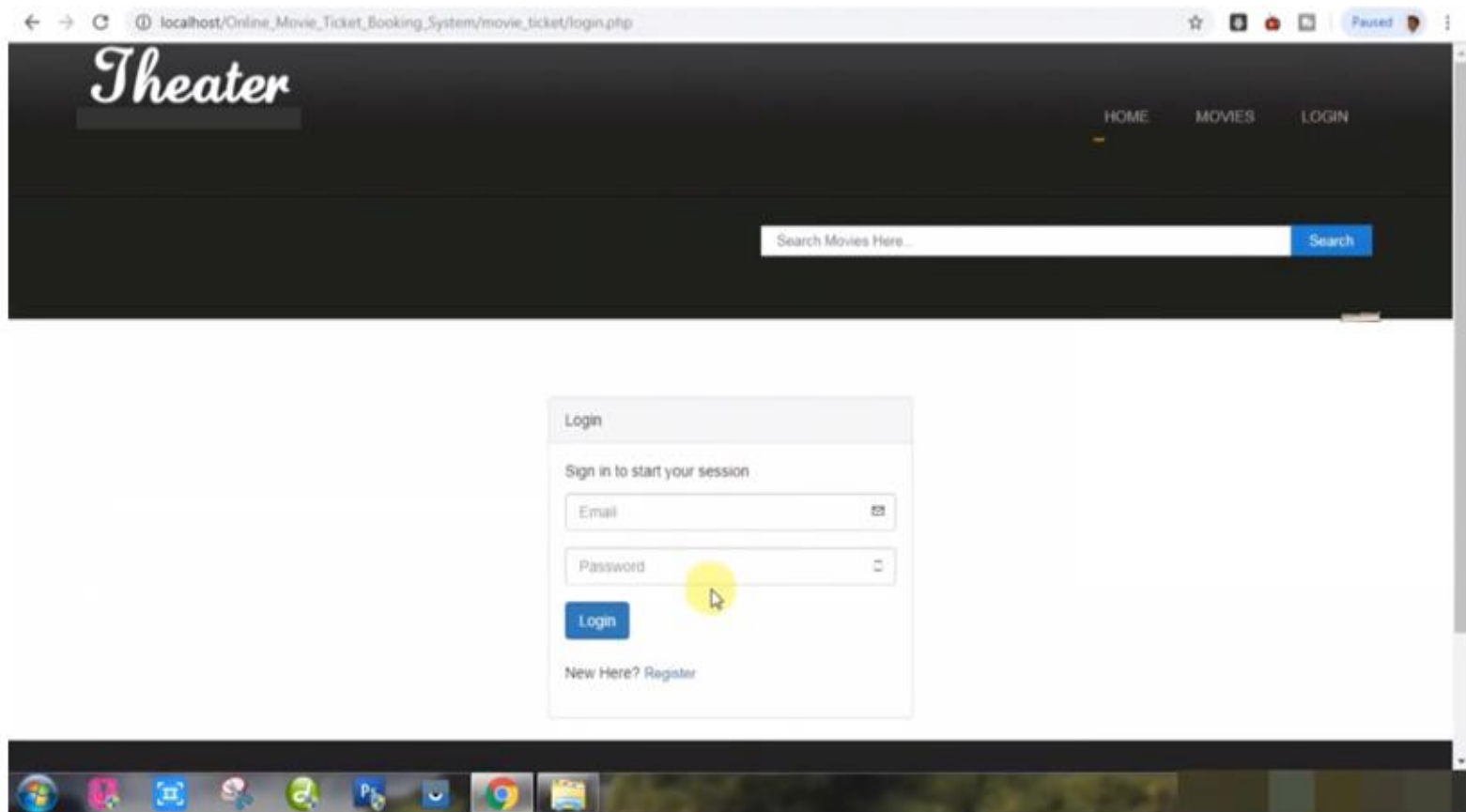
System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Online Movie Ticket Booking System to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process

of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

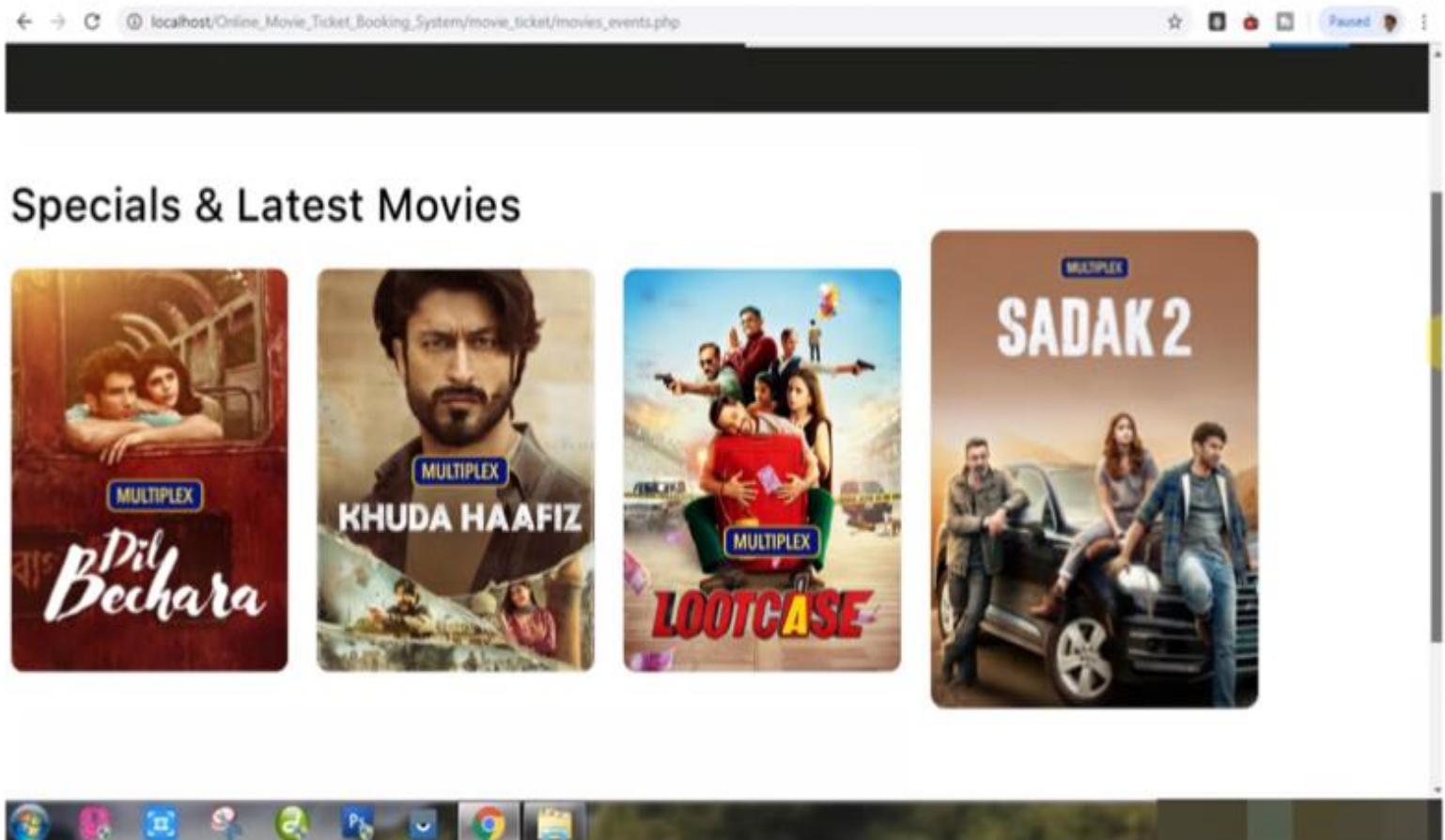
CHAPTER 4

IMPLEMENTED SCREENSHOTS

Login Page



Movie Page



Selection Page

Search Movies Here...

Sadak 2



Drama

2020 • 15+

Ravi, a suicidal taxi driver, plans to shut shop when a young girl, Aarya, compels him to take her on a life-altering journey to Kailash.

Starring Alia Bhatt, Aditya Roy Kapur and Sanjay Dutt

Watch Trailer

10:00 AM

02:00 PM

06:00 PM

09:00 PM

Booking Page

← → ↻

localhost/Online_Movie_Ticket_Booking_System/movie_ticket/booking.php



Drama

2020 • 15+

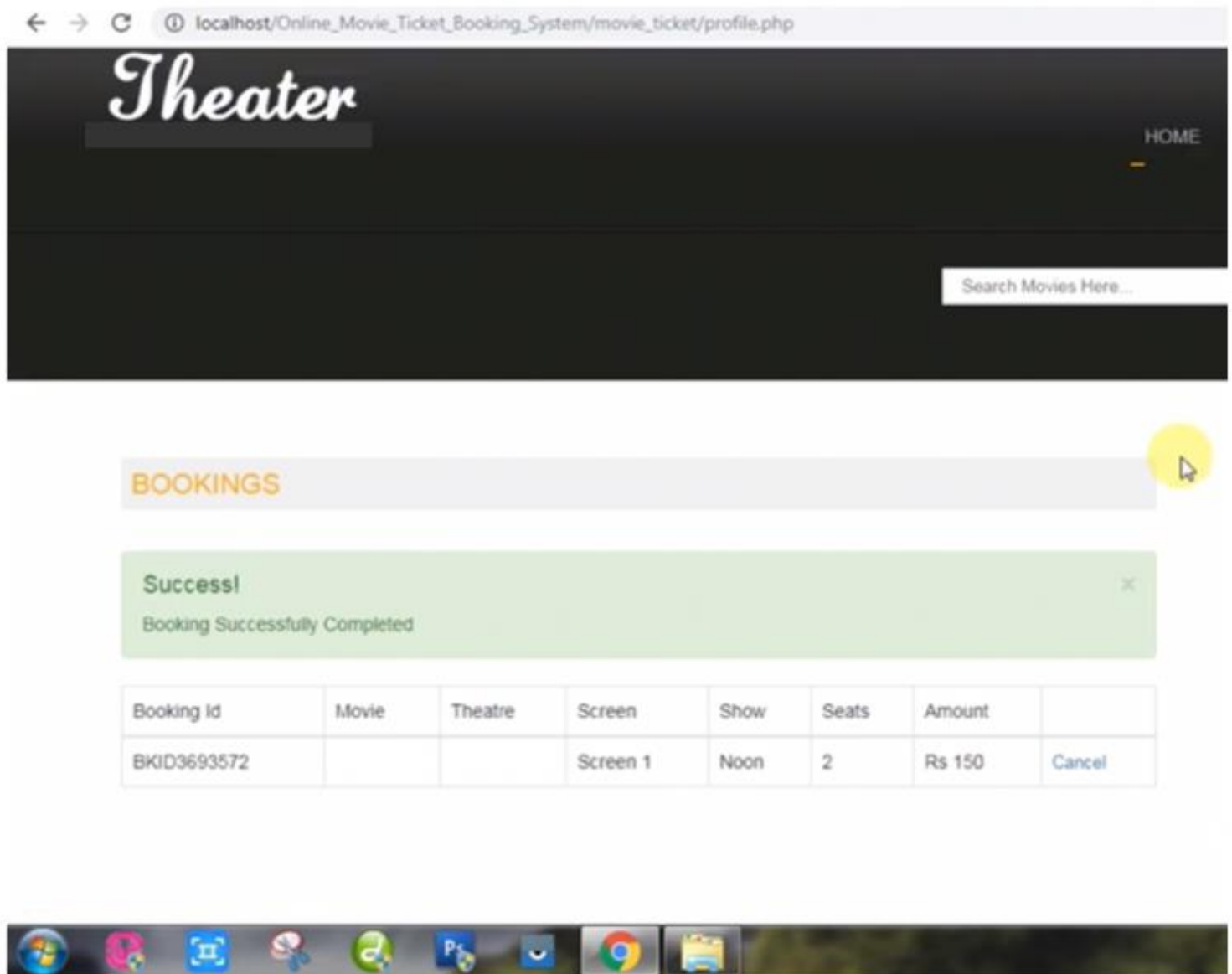
Ravi, a suicidal taxi driver, plans to shut shop when a young girl, Aarya, compels him to take her on a life-altering journey to Kailash.

Starring Alia Bhatt, Aditya Roy Kapur and Sanjay Dutt

Watch Trailer

Theatre	
Screen	Screen 1
Date	📅
Show Time	10:00 AM Noon Show
Number of Seats	1
Amount	Rs 75
Book Now	

Confirmation Page



CHAPTER 5

CONSLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points...

- A description of the background and context of the project and its relation to work already done in the area.
- The description of Purpose, Scope, and applicability. We define the problem on which we are working in the project
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system. Finally the system is implemented and tested according to test cases.

APPENDIX

SOURCE CODE:

Login Page:

```
<?php
if (!session_id()) {
    session_start();
}
include_once ('db.php');

$user=$_REQUEST['postName'];
$pass=$_REQUEST['postPassword'];

$sql="select userId,status from user where  userName ='$user' and password='$pass'";

$account_type="";

$res=$conn->query($sql);
if (($data=$res->fetch_object())) {
    $_SESSION['user']= $data->userId ;
    $account_type=$data->status;

    if ( $account_type ==="101") {
        echo "1";
    }else if ($account_type ==="202"){
        echo "2";
    }
}
}else{
    echo "3";
}

?>
```

Movie Selection Page:

```
<?php
if (!session_id()) {
# code...
    session_start();
}
include 'db.php';
?>

<!DOCTYPE html>
<html>
<head>

    <link href="https://bootswatch.com/flatly/bootstrap.css" rel="stylesheet">
    <link href="http://netdna.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css" rel="stylesheet">

    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
    <link href="css/ie10-viewport-bug-workaround.css" rel="stylesheet">

    <!-- Just for debugging purposes. Don't actually copy these 2 lines! -->
    <!--[if lt IE 9]><script src="../../assets/js/ie8-responsive-file-warning.js"></script><![endif]-->
    <script src="js/ie-emulation-modes-warning.js"></script>

    <!-- Custom styles for this template -->
    <link href="css/rotating-card.css" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
    <link href="css/anotherDefault.css" rel="stylesheet">
</head>
<body>

<div class="container">

<div class="panel with-nav-tabs panel-success">
    <div class="panel-heading">
        <ul class="nav nav-tabs">
            <li class="active"><a href="#nowshowing" data-toggle="tab">Showing Now</a></li>
        </ul>
    </div>
</div>
```

```
<div class="panel-body">
  <div class="tab-content">
    <div class="tab-pane fade in active" id="nowshowing">
```

```
<?php
```

```
$count=0;
```

```
$res=$conn->query("select * from movielist;");
```

```
while ($row=$res->fetch_object()) {
```

```
    $_SESSION['movie']=$row->movieId;
```

```
    if ($count==4) {
```

```
        echo "<div class='row'>";
```

```
        $count=0;
```

```
    }
```

```
    echo "
```

```
    <div class='col-md-3 col-sm-12'>
```

```
        <div class='card-container'>
```

```
            <div class='card'>
```

```
                <div class='front'>
```

```
                    <div class='cover'>
```

```
                        <img src='uploadimages/'.$row->image.'"/>
```

```
                    </div>
```

```
                <div class='content'>
```

```
                    <div class='main'>
```

```
                        <h3 class='name'>".$row->Name."</h3>
```

```
                        <p><b>IMDB: </b>".$row->imdb."</p>
```

```
                        <p class='profession'><b>Genre: </b>".$row->Genre."</p>
```

```
                        <p class='profession'><b>Director: </b> " ".$row->Director."</p>
```

```
                    </div>
```

```
                </div>
```

```
</div>
<!-- end front panel -->
<div class='back'>
    <div class='content'>
        <div class='main'>
            <h4 class='text-center'>"$row->Name ."</h4>
            <p class='text-center'>"$row->Description ." </p>
        </div>
        <div style='margin-top: 4vw;' class='buy_ticket'>
            <a href='javascript:void(0)' onclick='openLoginModal();' class='btn btn-primary btn-xs btn-block'>Click to book ticket</a>
        </div>
    </div>
</div> <!-- end card -->
</div> <!-- end card-container -->
</div>
</div>";

$count++;
} ?>

</div>
</div>
</div>
</div>
</div>
</div>
```

Booking Page:

```
<?php
if (!session_id()) {
    session_start();
}
include 'db.php';?>
<!DOCTYPE html>
<html>
<head>
    <title>Ticket</title>
    <link rel="stylesheet" type="text/css" href="css/customerPanel.css">
    <link href="bootstrap/bootstrap.min.css" rel='stylesheet' type='text/css' />

    <style type="text/css">
        .boxStyle{width: 100%;
            border: 1px solid #ccc;
            background: #FFF;
            margin: 0 0 5px;
            padding: 10px;
            font-style: normal;
            font-variant-ligatures: normal;
            font-variant-caps: normal;
            font-variant-numeric: normal;
            font-weight: 400;
            font-stretch: normal;
            font-size: 12px;
            line-height: 16px;
            font-family: Roboto, Helvetica, Arial, sans-serif;

        }
    </style>

</head>
<body>
    <?php include_once 'header.php'; ?>
    <div class="container">
        <div class="row">
            <div class="col-xs-12 toppad" >
                <div class="panel panel-info">
```

```

<div class="panel-heading">
    <h3 class="panel-title">
        <?php
//for time slot

        //$timeSlot=$conn->query("select time from timeslot");

//movie details

        $movieId=$_POST['movieId'];
        $_SESSION['movieId']=$movieId;
        $res=$conn->query("select * from movielist where movieId=$movieId;");
        $row=$res->fetch_object();

        echo "".$row->Name;?>
    </h3>
</div>
<div class="panel-body">
    <div class="row">
        <div class="col-md-4 col-lg-4 " align="center">
            <img alt="User Pic" src=<?php echo "uploadimages/".$row->image."";?> class="img-
responsive">
        </div>
        <div class=" col-md-8 col-lg-8 ">
            <table class="table table-user-information">
                <tbody>
                    <tr>
                        <td><strong>Movie Name </strong></td>
                        <td><?php echo "".$row->Name;?> </td>
                    </tr>
                    <tr>
                        <td><strong>Genre</strong></td>
                        <td><?php echo "".$row->Genre;?> </td>
                    </tr>
                    <tr>
                        <td><strong>Director</strong></td>
                        <td><?php echo "".$row->Director;?> </td>
                    </tr>
                    <tr>
                        <td><strong>IMDB</strong></td>
                        <td><?php echo "".$row->imdb;?> </td>

```

```

</tr>
<tr>
    <td><strong>Description</strong></td>
    <td><?php echo "". $row->Description; ?> </td>
</tr>

```

```

<form action="ticketConfirmation.php" method="post" >

```

```

<tr>
    <td><strong>Date</strong></td>
    <td><input class="boxStyle" type="date" name="date"></td>
</tr>

```

```

<tr>

```

```

    <td><strong>Show Time</strong></td>
    <td> <select name="timeSlot" class="boxStyle">
        <?php $timeSlot=$conn->query("select time from timeslot");
        while ($showTime=$timeSlot->fetch_object()) {

            echo " <option value='".$showTime->time."'>". $showTime->time."</option>";
        } ?>
    </select></td>

```

```

</tr>

```

```

<tr>

```

```

    <td><strong>Theater</strong></td>
    <td>
        <select name="theater" class="boxStyle">
            <?php $resource=$conn->query("select theaterName from theater");
            while ($theater=$resource->fetch_object()) {

                echo " <option value='".$theater->theaterName."'>". $theater-

```

```

>theaterName."</option>

```

```

            ";
        } ?>
    </td>

```

```

</tr>

```

```

<tr>

```

```
                <td colspan="2" width="100%">
                    <input class="btn btn-primary btn-xs btn-block" type="submit" name="submit"
value="Request For Ticket">
                </td>
            </tr>
        </form>

        </tbody>
    </table>

    </div>
</div>

</div>
</div>
</div>

</div>
</body>

</html>
```

Confirmation Page:

```
<?php
if (!session_id()) {
    session_start();
}

include 'db.php';
include_once 'header.php';

?>

<!DOCTYPE html>
<html>
<head>

    <title>Ticket</title>

    <link rel="stylesheet" type="text/css" href="css/customerPanel.css">
    <link href="bootstrap/bootstrap.min.css" rel='stylesheet' type='text/css' />

    <style type="text/css">
        .boxStyle{width: 100%;
            border: 1px solid #ccc;
            background: #FFF;
            margin: 0 0 5px;
            padding: 10px;
            font-style: normal;
            font-variant-ligatures: normal;
            font-variant-caps: normal;
            font-variant-numeric: normal;
            font-weight: 400;
            font-stretch: normal;
            font-size: 12px;
            line-height: 16px;
            font-family: Roboto, Helvetica, Arial, sans-serif;

        }
    </style>
```

```

</head>
<body>

<?php
$movieId=$_SESSION['movieId'];

$movieIdentity=$conn->query("select * from movielist where movieId=$movieId;");
$row=$movieIdentity->fetch_object();


$movieName=$row->Name;
$date=$_POST['date'];
$time=$_POST['timeSlot'];
$theater=$_POST['theater'];

$showOrder="";//id of reducing seat
$seatCount="";//seat counting

//Cheaking Date if it exist;
$resource=$conn->query("select date from showOrder;");
$dateExists=false;
while ($rw=$resource->fetch_object()) {
    if ($rw->date=== $date) {
        $dateExists=true;
        break;
    }
}

$seatCount="";
//First input in database table
if (!$dateExists) {
    $sql="INSERT INTO showOrder(showOrderId,date,timeslot,theater,movieName,seat) VALUES ('','$date','$time','$theater','$movieName','$50'.");
}

```

```

if ($conn->query($sql) === TRUE) {
    $result=$conn->query("select showOrderId from showOrder where date='".$date.'" and timeslot='".$time.'"
and theater='".$theater.'" and movieName='".$movieName.'"");

    $r=$result->fetch_object();

    $showOrder=$r->showOrderId;//we will use this id to reduce seat

    $result=$conn->query("select seat from showOrder where showOrderId='".$showOrder.'"");

    $r=$result->fetch_object();

    $seatCount=$r->seat;

}else{
    echo "Error: " . $sql . "<br>" . $conn->error;
}
}

//There is an input in the date field
//checking timeSlot
else{
    $resource=$conn->query("select timeslot from showOrder where date='".$date.'"");
    $timeExist=false;
    while ($rw=$resource->fetch_object()) {
        if ($rw->timeslot==$time) {
            $timeExist=true;
            break;
        }
    }
    if (!$timeExist) {
        $sql="INSERT INTO showOrder(showOrderId,date,timeslot,theater,movieName,seat) VALUES
('".$date."','".$time."','".$theater."','".$movieName."','".$50'.")";
    }
}

```

```

        if ($conn->query($sql) === TRUE) {
            $result=$conn->query("select showOrderId from showOrder where date='".$date.'" and timeslot='".$time.'"
and theater='".$theater.'" and movieName='".$movieName.'"");

            $r=$result->fetch_object();

            $showOrder=$r->showOrderId;//we will use this id to reduce seat

            $result=$conn->query("select seat from showOrder where showOrderId='".$showOrder.'"");

            $r=$result->fetch_object();

            $seatCount=$r->seat;
            //echo "".$seatCount;
        }else{
            echo "Error: " . $sql . "<br>" . $conn->error;
        }
    }else{

//there is an input date and timeSlot field
//theater Checking

        $resource=$conn->query("select theater from showOrder where date='".$date.'" and timeslot='".$time.'"");
        $theaterExist=false;
        while ($rw=$resource->fetch_object()) {
            if ($rw->theater==$theater) {
                $theaterExist=true;
                break;
            }
        }

        if (!$theaterExist) {
            $sql="INSERT INTO showOrder(showOrderId,date,timeslot,theater,movieName,seat) VALUES
('".$date."','".$time."','".$theater."','".$movieName."','".$50'.")";

            if ($conn->query($sql) === TRUE) {
                $result=$conn->query("select showOrderId from showOrder where date='".$date.'" and
timeslot='".$time.'" and theater='".$theater.'" and movieName='".$movieName.'"");

                $r=$result->fetch_object();

```

```

        $showOrder=$r->showOrderId;//we will use this id to reduce seat

        $result=$conn->query("select seat from showOrder where showOrderId='".$showOrder."'");

        $r=$result->fetch_object();

        $seatCount=$r->seat;
    }

}

else{
// database e date time theater sob e ase , ekhon movie ase kina check korbo.

        $resource=$conn->query("select movieName from showOrder where date='".$date.'" and timeslot='".$time.'"
and theater='".$theater."'");
        $movieExist=false;
        while ($rw=$resource->fetch_object()) {
            if ($rw->movieName=== $movieName) {
                $movieExist=true;
                break;
            }
        }
        if (!$movieExist) {
            $sql="INSERT INTO showOrder(showOrderId,date,timeslot,theater,movieName,seat) VALUES
('".$date."','".$time."','".$theater."','".$movieName."','".$50'.")";

            if ($conn->query($sql) === TRUE) {
                $result=$conn->query("select showOrderId from showOrder where date='".$date.'" and
timeslot='".$time.'" and theater='".$theater.'" and movieName='".$movieName."'");

                $r=$result->fetch_object();

                $showOrder=$r->showOrderId;//we will use this id to reduce seat

                $result=$conn->query("select seat from showOrder where showOrderId='".$showOrder."'");

                $r=$result->fetch_object();

```



```

        echo "".$movieName;

    ?>

</h3>
</div>
<div class="panel-body">
    <div class="row">
        <div class="col-md-4 col-lg-4 " align="center">
            <img alt="User Pic" src=?php echo "uploadimages/".$row->image."";?> class="img-responsive">
        </div>
        <div class="col-md-8 col-lg-8 ">
            <table class="table table-user-information">
                <tbody>

                    <tr>
                        <td><strong>Date </strong></td>
                        <td><?php echo "".$date ?> </td>
                    </tr>
                    <tr>
                        <td><strong>Movie Name </strong></td>
                        <td><?php echo "".$row->Name;?> </td>
                    </tr>
                    <tr>
                        <td><strong>Time </strong></td>
                        <td><?php echo "".$_POST["timeSlot"]?></td>
                    </tr>
                    <tr>
                        <td><strong>Movie Name </strong></td>
                        <td><?php echo "".$_POST["theater"]?></td>
                    </tr>
                    <tr>
                        <td><strong>Seat Available </strong></td>
                        <td><?php echo $seatCount;?> </td>
                    </tr>
                    <tr>
                        <td><strong>Ticket Price </strong></td>
                        <td> 300 </td>
                    </tr>

                <form action="seatReducing.php" method="post">

```

```
<input type="hidden" name="showOrderId" value=<?php echo ''. $showOrder.'''; ?>>

<input type="hidden" name="date" value=<?php echo ''.$date.'''; ?>>

<input type="hidden" name="time" value=<?php echo ''.$time.'''; ?>>

<input type="hidden" name="theater" value=<?php echo ''.$theater.'''; ?>>

<input type="hidden" name="seat" value=<?php echo ''.$seatCount.'''; ?>>


<tr>
    <td colspan="2" width="100%">
        <input class="btn btn-primary btn-xs btn-block" type="submit" name="submit"
value="Confirm Ticket">
    </td>
</td>
</form>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
```

REFERENCES

- <http://www.javaworld.com/javaworld/jw-01-1998/jw-01-Credentialreview.html>
- Head First Php 2nd Edition
- <http://www.jdbc-tutorial.com/>
- Portal Design Concepts by Apress
- <https://www.tutorialspoint.com/html/>
- <https://docs.oracle.com/javase/tutorial/>
- <http://www.wampserver.com/en/>
- <http://www.JSP.net/>
- <http://www.tutorialspoint.com/mysql/>
- <http://httpd.apache.org/docs/2.0/misc/tutorials.html>