

# **Summer Training Report**

**On**

## **MULTIPLEX TICKET BOOKING**

**(Subject Code: BCA-508P)**

*Submitted in partial fulfilment of the  
requirement for the award of the degree*

**of**

## **BACHELOR OF COMPUTER APPLICATION**

*Submitted by*

**PRASHANT SRIVASTAV**

**2112040580036**

*Under the guidance of*

**Dr. Santosh Kumar Dwivedi**



**DEPARTMENT OF COMPUTER APPLICATION,  
SHRI RAMSWAROOP MEMORIAL COLLEGE OF MANAGEMENT, LUCKNOW**

**Affiliated to  
LUCKNOW UNIVERSITY, LUCKNOW**



**September - 2023**

## CERTIFICATS



Analyze Infotech, Janpriya Complex Phase-II,  
Tedhi Pulia, Aliganj, Lucknow,(U.P.)  
Phone :-9453193268, 9839434821,  
Web :-[www.analyzeinfotech.in](http://www.analyzeinfotech.in)  
E-mail :-[admin@analyzeinfotech.in](mailto:admin@analyzeinfotech.in)

Ref. No. ai/23/211

### TO WHOM SO EVER IT MAY CONCERN

This is to certify that **Mr. Prashant Srivastav** S/O Mr. Paramatma Prasad Srivastav student of BCA from Shri Ramswaroop Memorial College of Management, Lucknow in recognition of successful completion of Summer Training on "**Java Software Development**" at Analyze InfoTech. The duration of his project was from 01 August 2023 to 30 September 2023.

He has done a good job during his engagement with the software Development & Testing Division of Company. He has completed his project during the Training tenure. His performance was good and satisfactory.

I would like to take this opportunity to express my appreciation to Mr. **Prashant Srivastav** for his work and wish him All the best for future endeavors.

Yours Sincerely



Date: **01 October 2023**

\_\_\_\_\_  
**Pankaj Mishra**

**(Project Manager)**

**For Analyze InfoTech, Lucknow.**

[www.analyzeinfotech.in](http://www.analyzeinfotech.in)

## **ABSTRACT**

In an era of rapid digital transformation, the movie industry has witnessed significant advancements in how patrons book tickets to their favourite films. This abstract introduces a cutting-edge solution known as the "Multiplex Movie Booking" system. This innovative platform revolutionizes the way users reserve seats for their preferred movies, enhancing the overall cinema-going experience.

The Multiplex Movie Booking system is a user-friendly, web-based application designed to streamline the ticket booking process for multiplex cinemas. This platform offers a plethora of features to ensure a seamless and convenient booking experience for both cinema-goers and cinema operators.

## ACKNOWLEDGEMENT

Fifth Semester Project is a major component of Academic Schedule of B.C.A. Hence, I worked on **Multiplex Ticket Booking**. The conceptual Knowledge acquired by Management/ Computer student is best manifested in the project they undergo. The present project gives a perfect vent to my understanding of the practicalities of information of different educational areas. I express my whole hearted gratitude towards **Shri Ramswaroop College Of Management** for having given me the opportunity to undergo my project in the field of Website development of great report and allowing me to gain invaluable experience. I express my heart-felt gratitude to **Dr. Santosh Kumar Dwivedi** for supervising me during the project period. I also express my special thanks to all the staff member who gave me their precious time and help me whenever required. I am also grateful to my parents who have always been supportive in giving me correct decision and advice.

I also express my sincere thanks to all the Respondents, without whose help the completion of this project report was not possible

Prashant Srivastav  
5<sup>th</sup> Semester

## DECLARATION

I, the undersigned, solemnly affirm that the project report presented herein is the result of my own work conducted during the course of my studies, under the diligent guidance of Dr. Santosh Kumar Dwivedi. I assert that the statements and conclusions within this report are the direct outcomes of my research endeavours. Furthermore, I hereby certify the following:

1. The content of this report is entirely original and has been diligently undertaken by me, with the general oversight and guidance of my supervisor.
2. This work has not been submitted to any other institution for the purpose of obtaining any degree, diploma, or certificate, either within this university or at any other institution in India or abroad.
3. We have adhered to the university's prescribed guidelines for the composition and structuring of this report.
4. In cases where we have incorporated material, including data, theories, analyses, and textual content, from external sources, we have duly acknowledged and credited them within the body of the report and provided comprehensive references for further verification.

Signature of the Student

[Prashant Srivastav]

## TABLE OF CONTENTS

TOPIC	PAGES
1. COVER PAGE -----	1
2. CERITIFICATE -----	2
3. ABSTRACT -----	3
4. ACKNOWLEDGEMENT -----	4
5. DECLARATION -----	5
6. CHAPTER 1	
i. INTRODUCTION -----	8
• BACKGROUND -----	8
• OBJECTIVE -----	8-9
• PURPOSE AND SCOPE -----	9
7. CHAPTER 2	
i. SURVEY OF TECHNOLOGY -----	10
• JAVA PROGRAMMING LANGUAGE -----	10
• MySQL -----	10-11
8. CHAPTER 3	
i. SOFTWARE ENGINEERING -----	12
• SOGFTWARE DEVELOPMENT LIFECYCLE---	12
• THE STAGES OF SDLC-----	12-13
• WATERFALL MODEL-----	13-14
9. CHAPTER 4	
i. REQUIRMENT & ANALYSIS -----	15
• PROBLEM DEFINITION -----	15
• PLANNING AND SCHEDULE-----	15-16
• PERT CHART -----	16
• REQUIREMENT SPECIFICATION -----	17
• PRELIMINARY PRODUCT -----	17
• CONCEPTUAL MODEL -----	18
• DATA FLOW DIAGRAM -----	19
• ER-DIAGRAM -----	20-21
• USE-CASE DIAGRAM -----	22

10.CHAPTER 5	
i. TESTING	-----23
• SYSTEM TESTING	-----23
• WHY ITS DONE	-----23
• CAUSES OF ERROR	-----23-24
• TESTING PRINCIPLES	-----24-25
• TESTING OBJECTIVE	-----25
• KINDS OF TESTING	-----25-26
• TESTING TECHNIQUES USED	-----26-27
11.CHAPTER 6	
i. SCREENSHOT OF DATABASE	-----28
ii. PROJECT SCREENSHOT	-----29-33
12.FUTURE SCOPE OF PROJECT	-----34
13.CONCLUSION	-----35
14.REFERENCE	-----36

# CHAPTER 1

## 1. INTRODUCTION

### 1.1 BACKGROUND

In our rapidly evolving world, the significance of information technology and information management cannot be overstated. We find ourselves in the midst of the computer age, a period that has seen a remarkable surge in the popularity of computers. The computer revolution has permeated virtually every facet of human existence.

Computers are exceptionally well-suited for handling information; they are essentially information processors. They possess the capability to receive data, execute fundamental operations on that data, and generate outcomes in accordance with pre-established instructions.

One notable application of this technology is in the realm of multiplex cinema halls, where it plays a pivotal role in streamlining operations. The multiplex ticket booking software is purposefully designed to alleviate the burdensome tasks faced by professionals working within these cinema halls. Its primary functionalities encompass ticket reservations, management of movie show details, and client information handling.

### 1.2 OBJECTIVE

Today's world is computer world because most of work is doing with the help of computer. Dependency on computer is behind the few reasons. We cannot easily manage to store large number of data or information single handle. If we will be needing some information or data in urgency then we cannot manage in manually these works are very difficult if we cannot use computer.

As this is generic software it can be used by a wide variety of multiplex cinema halls to automate the process of manually maintaining the records related to the subject of maintaining the movie details and customer data.

This software is basically updating the manual multiplex ticket booking to automated ticket booking. So that organization can manage their record in efficient and organize them.

- The main objective is to automate non computer environment
- To save manpower.
- It will speed the processing of data and transaction.
- It will provide best security features such as provisions of passwords.



- **System Objective:**

Today's world is computer world because most of work is doing with the help of computer. Dependency on computer is behind the few reasons. We cannot easily manage to store large number of data or information single handle. If we will be needing some information or data in urgency then we cannot manage in manually these works are very difficult if we cannot use computer.

- **System Context:**

This section clearly depicts the environment and boundaries of the Multiplex ticket booking system and the entities with which it interacts. It helps us see how the system fits into the existing scheme of things. What the system will do by itself.

- **Functional Requirement:**

This Software must request Username and Password for access to data, after authentication will allow access to the system. The Software must allow input of products data from administrator and secured access.

- **Non-Functional Requirement:**

In this Software Input error will be returned in red with appropriate message box. System should automatically update after every transaction

## **1.3 PURPOSE AND SCOPE**

- **PURPOSE:**

The purpose of this document is to specify requirements and to give guidelines for the development of above said project. In particular it gives guidelines on how to prepare the above said project.

This document is intended to be a practical guide for people who developing this software.

- **SCOPE:**

As this is generic software it can be used by a wide variety of multiplex cinema halls to automate the process of manually maintaining the records related to the subject of maintaining the movie details and customer data.

## CHAPTER 2

## 2 SURVEY OF TECHNOLOGIES

### 2.1 JAVA PROGRAMMING LANGUAGE

Java is a versatile computer programming language characterized by its concurrent, class-based, and object-oriented nature. It is specifically crafted to minimize implementation dependencies, allowing developers to write code that can be executed across various platforms without the need for recompilation. This concept is encapsulated in the mantra "write once, run anywhere" (WORA).

Java applications are typically compiled into bytecode, a platform-independent intermediate code that can be executed on any Java Virtual Machine (JVM), regardless of the underlying computer architecture. As of 2016, Java stands as one of the most widely used programming languages, particularly for developing client-server web applications, boasting a community of approximately 9 million developers.

The origins of Java trace back to James Gosling's development efforts at Sun Microsystems, which has since become part of Oracle Corporation. Java made its debut in 1995 as an integral component of Sun Microsystems' Java platform. While it draws inspiration from the syntax of C and C++, it provides fewer low-level features compared to these languages.

### 2.2 MY SQL

MySQL stands as a prominent open-source Relational Database Management System (RDBMS) with a name that blends personal significance and technical prowess. This report provides a comprehensive overview of MySQL, covering its origins, licensing, ownership history, and the availability of proprietary editions.

#### 1. Origin and Name

MySQL owes its unique name to a creative blend of "My," the first name of Michael Widenius's daughter (one of its co-founders), and "SQL," which stands for Structured Query Language. This moniker not only reflects a personal touch but also underscores its central function: managing structured data using SQL.

#### 2. Open-Source Foundation

At its core, MySQL is rooted in open-source principles. The MySQL development project is underpinned by a commitment to transparency and community-driven collaboration. This means that the source code for MySQL is readily available to the public. This open-source nature encourages developers from all around the world to contribute to its development and improvement. It has been released under the GNU

General Public License (GPL), ensuring that it remains open and accessible to anyone who wishes to use and modify it.

### **3. Evolution of Ownership**

Over the years, MySQL's ownership landscape has undergone significant changes. In its early days, MySQL was primarily owned and sponsored by a single for-profit company named MySQL AB, headquartered in Sweden. This company played a crucial role in nurturing and popularizing MySQL. However, one of the most noteworthy transitions in MySQL's history occurred when it was acquired by Oracle Corporation.

### **4. Licensing Flexibility**

MySQL's licensing flexibility is one of its defining features. While the core of MySQL is open source and licensed under the GPL, it also offers proprietary licensing agreements. These proprietary agreements cater to businesses and organizations seeking specific features and support options beyond what the open-source version provides. This duality allows MySQL to cater to a wide range of users, from individual developers and small startups to large enterprises.

### **5. Paid Editions for Enhanced Functionality**

For those looking for additional functionality and support, MySQL offers various paid editions. These editions are designed to meet the needs of businesses and organizations with more demanding requirements. They often include advanced features such as high availability, scalability, and security enhancements. The availability of these paid editions ensures that MySQL can address a broad spectrum of use cases effectively.

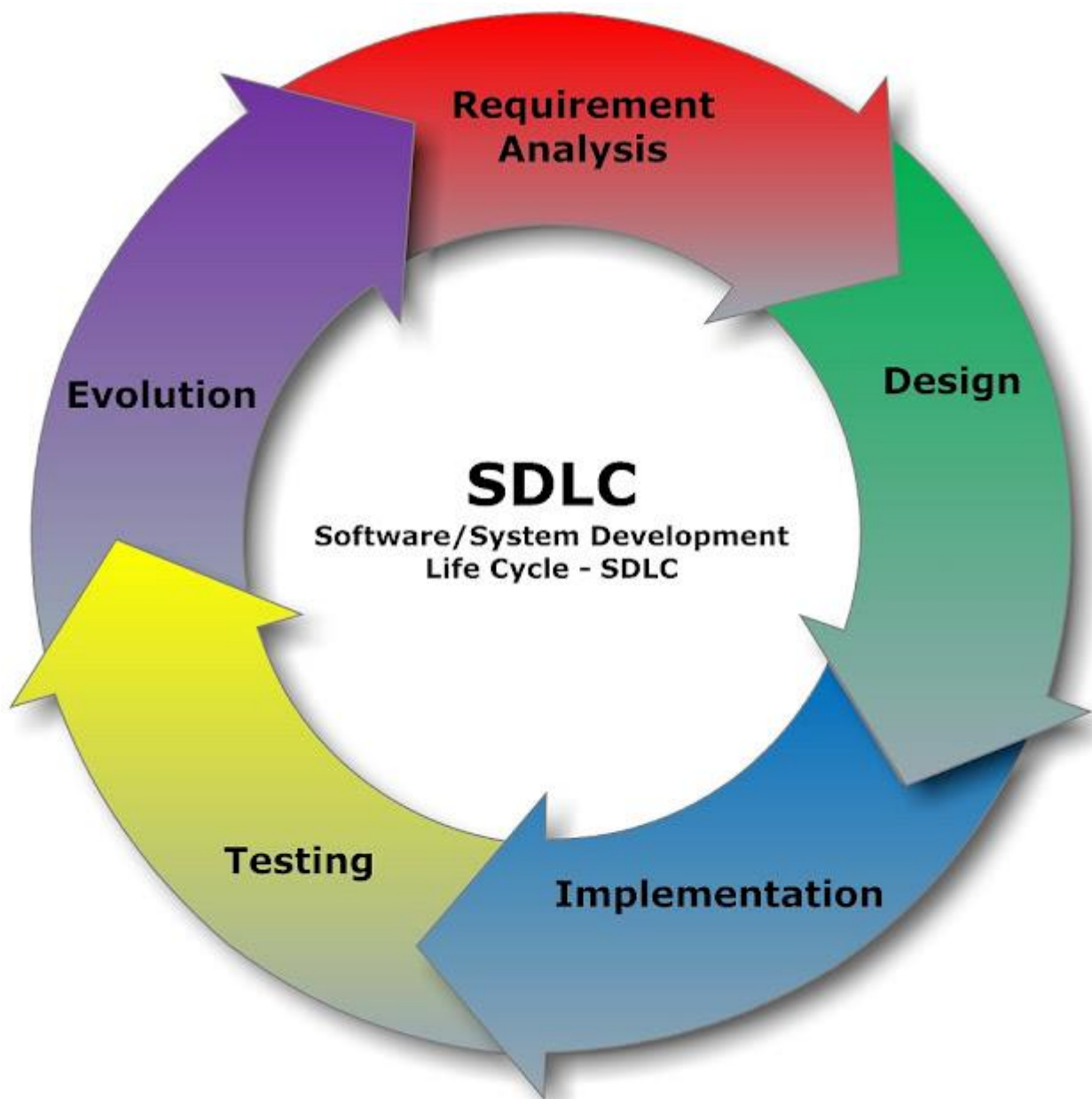
In conclusion, MySQL's journey from its inception as an open-source project to its current status as a significant part of Oracle Corporation's portfolio reflects the evolution of the database management landscape. Its name, a fusion of personal and technical elements, encapsulates its essence as an open, versatile, and widely adopted RDBMS. MySQL's dual approach to licensing, with both open-source and proprietary options, has allowed it to remain relevant and adaptable in a rapidly changing technological landscape. Its influence on data storage and management is substantial, and it continues to be a fundamental tool for developers, businesses, and organizations worldwide.

## CHAPTER 3

### 1. SOFTWARE ENGINEERING

Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

SDLC provides a series of steps to be followed to design and develop a software product efficiently. SDLC framework includes the following steps



## Methodologies to Consider

The system development life cycle (SDLC) is the overall process of developing software using a series of defined steps. This section discusses several SDLC models that work well for developing applications in Oracle Application Express.

- About Iterative vs. Planned Development
- About the Advantages of Creating Prototypes
- Waterfall
- Spiral
- Rapid Application Development

### About Iterative vs. Planned Development

When developing applications using Application Builder, you must find a balance between two dramatically different development methodologies:

- Iterative, rapid application development
- Planned, linear style development

Iterative, rapid application development offers so much flexibility that you run the risk of never completing your project. In contrast, planned, linear style development can yield applications that do not meet the needs of end users even if they meet the stated requirements on paper.

### About the Advantages of Creating Prototypes

The Oracle Application Express development environment enables developers to take a more iterative approach to development. Unlike many other development environments, creating prototypes is easy. With Oracle Application Express, developers can:

- Use built-in wizards to quickly design an application user interface.
- Make prototypes available to users and gather feedback.
- Implement changes in real time, creating new prototypes instantly.

Methodologies that work well with Oracle Application Express include Spiral and Rapid Application Development (RAD).

## SDLC models

Waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, agile, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next.

### Waterfall

The Waterfall is probably the best known SDLC model. In this methodology, the development process is broken down into the following stages:

#### 1. Project Planning

Establishes a high-level view of the intended project and determines its goals.

#### 2. Requirements Definition

Defines project goals into defined functions and operation of the intended application. Analyses end-user information needs.

#### 3. Design

Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.

#### 4. Development

The real code is written here.

## **5. Integration and Testing**

Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.

## **6. Installation and Acceptance**

The final stage of initial development, where the software is put into production and runs actual business.

## **7. Maintenance**

What happens during the rest of the software's life: changes, correction, additions, moves to a different computing platform and more. This, the least glamorous and perhaps most important step of all, goes on seemingly forever.

This methodology is referred to as a waterfall because the output from one stage is the input for the next stage. A primary problem with this approach is that it is assumed that all requirements can be established in advance. Unfortunately, requirements often change and evolve during the development process.

## **Spiral**

A Spiral methodology is actually a series of short waterfall cycles. Each waterfall cycle yields new requirements and enables the development team to create a robust series of prototypes. One advantage of this approach is that it accommodates changing requirements. Disadvantages include complex project management and the risk development goes on indefinitely.

## **Rapid Application Development**

A Rapid Application Development (RAD) methodology has a heavy emphasis on creating a prototype that closely resembles the final product. The prototype is an essential part of the requirements phase. Advantages of this model include the ability to accommodate changing requirements, rapid development cycles, and progress can be easily measured. The major disadvantage of this model is that the emphasis on prototyping can result in scope creep. As a result, developers can lose sight of their initial goals in the attempt to create the perfect application.

## CHAPTER 4

### 4. REQUIREMENTS & ANALYSIS

#### 1. PROBLEM DEFINITION

Cinema-going is one of the most popular out-of-home cultural activities, affecting a serious of social, economic and cultural phenomena in modern societies. Cinemas are considered to be an integral part of cities and they contribute to the definition of a local geography and identity. They also contribute to the preservation of the collective memory, since they constitute a significant social and cultural practice linked to a specific place, which acts as a common reference or landmark for many individuals. Through this project we present a comprehensive solution for ticket booking in multiplexes. Theater management system, and ticket selling software that is easy to understand, easy to use and offers the simplicity of fast point-and-click service to the employee and admin. This powerful software program is specifically designed for theater owners, to sell tickets. This intuitive visual interface makes day-to-day aspects of selling, exchanging, refunding, and reporting fast and easy for both the user and administrators. Theater Management controls all back-end and front-end functionalities like, movie details, ticket rate and show time, customer information and sales history saved in a database, etc. Theater admin Manages the report details like counter wise report, daily, weekly, monthly report and movie report etc.

#### 2. PLANNING AND SCHEDULE

##### PERT CHART:

A PERT (Program Evaluation and Review Technique) chart is a project management tool used to plan and analyse the tasks and dependencies within a project. It helps project managers and teams visualize and understand the flow of tasks, the sequence of activities, and the critical path—the longest sequence of tasks that determines the project's minimum duration.

The Work Breakdown Structure of our proposed system “E-Commerce” is shown below:

<b><u>ID</u></b>	<b><u>Task Name</u></b>	<b><u>Duration</u></b>	<b><u>Start</u></b>	<b><u>Finish</u></b>
1	Project Initiation	3 days	1/7/23	3/7/23
2	Draft Project Plan	3 days	4/7/23	6/7/23
3	Analysis Phase	5 days	6/7/23	10/7/23
4	Plan User Interviews	2 days	10/7/23	12/7/23
5	Schedule users Interviews	1 days	12/7/23	13/7/23
6	Conducting users Interviews	1 days	13/7/23	14/7/23
7	System Design	10 days	14/7/23	23/7/23
8	Modules Design	3 days	23/7/23	25/7/23
9	Data Structure Design	3 days	26/7/23	28/7/23
10	User Interface Design	3 days	31/7/23	2/8/23
11	Coding Phase	13 days	03/8/23	15/8/23
12	Testing Phase	13 days	03/8/23	15/8/23
13	Integration Testing	5 days	16/8/23	20/8/23
14	System Level Testing	4 days	21/8/23	24/8/23
15	Implementation	2 days	25/8/23	26/8/23
16	Post-Implementation Review	3 days	27/8/23	29/8/23



### **3. REQUIREMENTS SPECIFICATION**

#### **❖ Software Requirements:**

1. Java/JDK
2. NetBeans
3. MySQL
4. SQL YOG

#### **❖ Hardware Requirements:**

1. Pentium iv processor
2. 512 MB RAM
3. 40 GB HDD
4. Colour Monitor
5. Keyboard, Mouse

### **4. PRELIMINARY PRODUCT DESCRIPTION:**

#### **MODULE DISCRIPTION:**

This system will cover mainly two Modules, i.e,

1. User
2. Theatre (Admin)

#### **❖ Theatre Module will cover SEVEN Sub module**

- User Login
- User Registration
- Movie Selection
- Show Time Selection
- Theatre Selection
- Ticket Booking
- Display of Booked Tickets with Amount

## 5. CONCEPTUAL MODELS

### DATA FLOW DIAGRAM

The data flow diagram shows the flow of data within any system. It is an important tool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It's also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD.

In the DFD, four symbols are used and they are as follows.

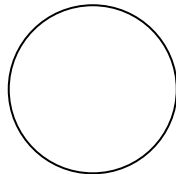
1. A square defines a source (originator) or destination of system data.



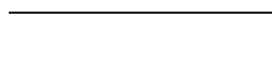
2. An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.



3. A circle or a "bubble" (Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.

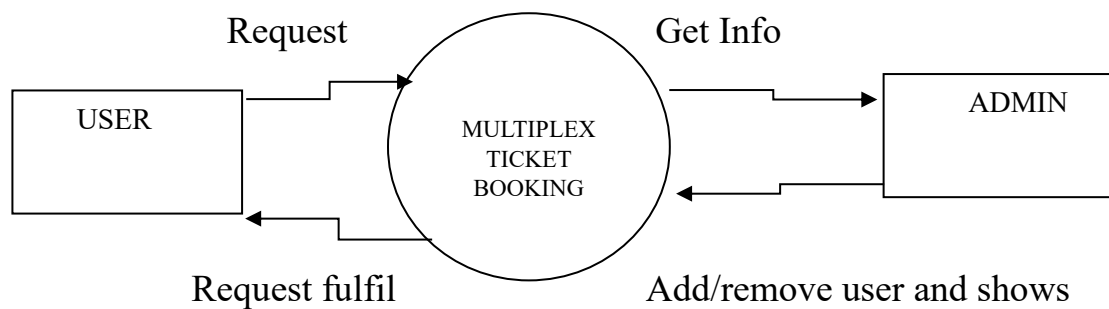


4. An open rectangle is a data store-data at rest, or a temporary Repository of data.



## **Context Level Data Flow Diagram:**

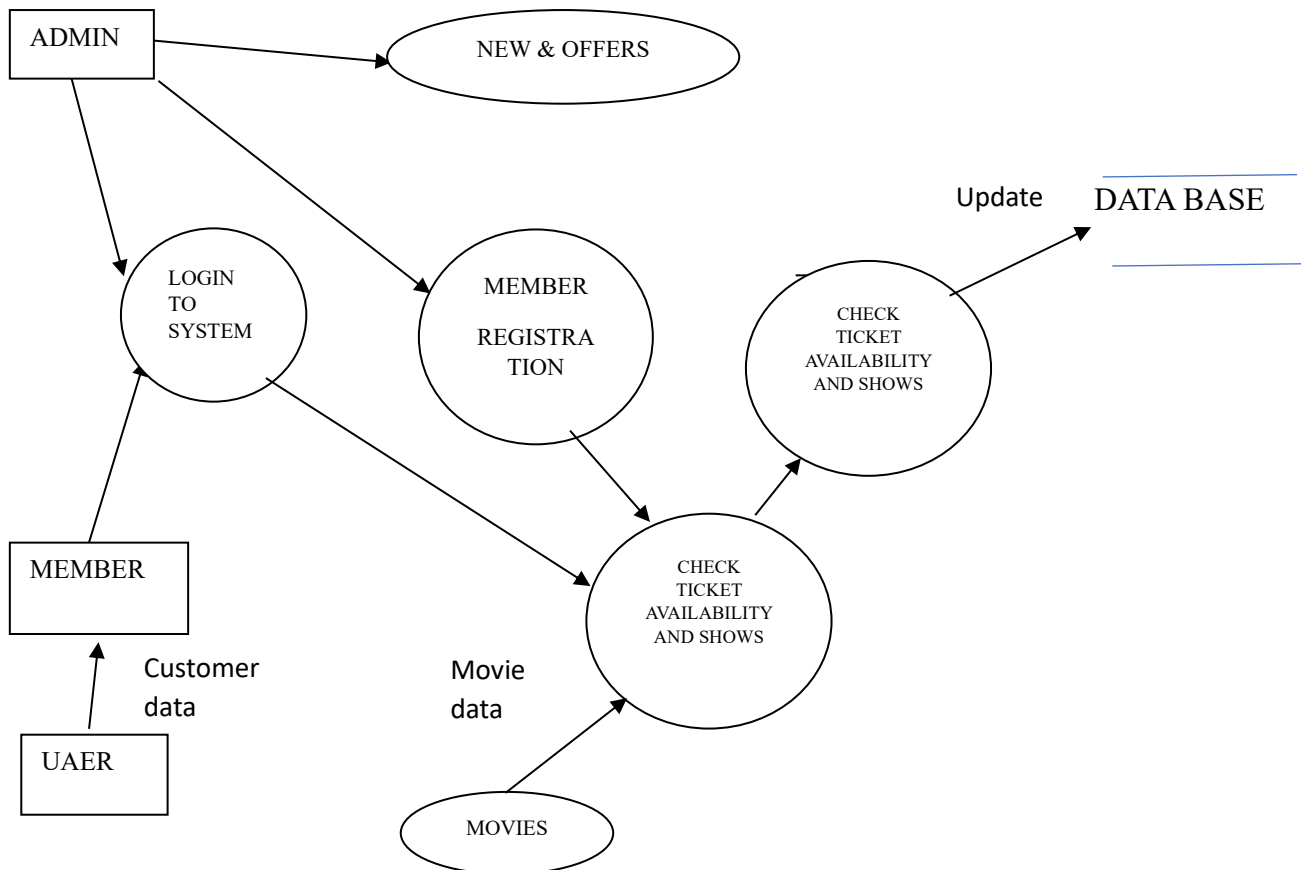
This level shows the overall context of the system and its operating environment and shows the whole system as just one process. Online book store is shown as one process in the context diagram; which is also known as zero level DFD, shown below. The context diagram plays important role in understanding the system and determining the boundaries. The main process can be broken into sub-processes and system can be studied with more detail; this is where 1<sup>st</sup> level DFD comes into play.



Zero Level Data Flow Diagram

## **First Level DFD:**

This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major high-level processes of the system and their interrelation. A process model will have one, and only one, level-1 diagram. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level 1.

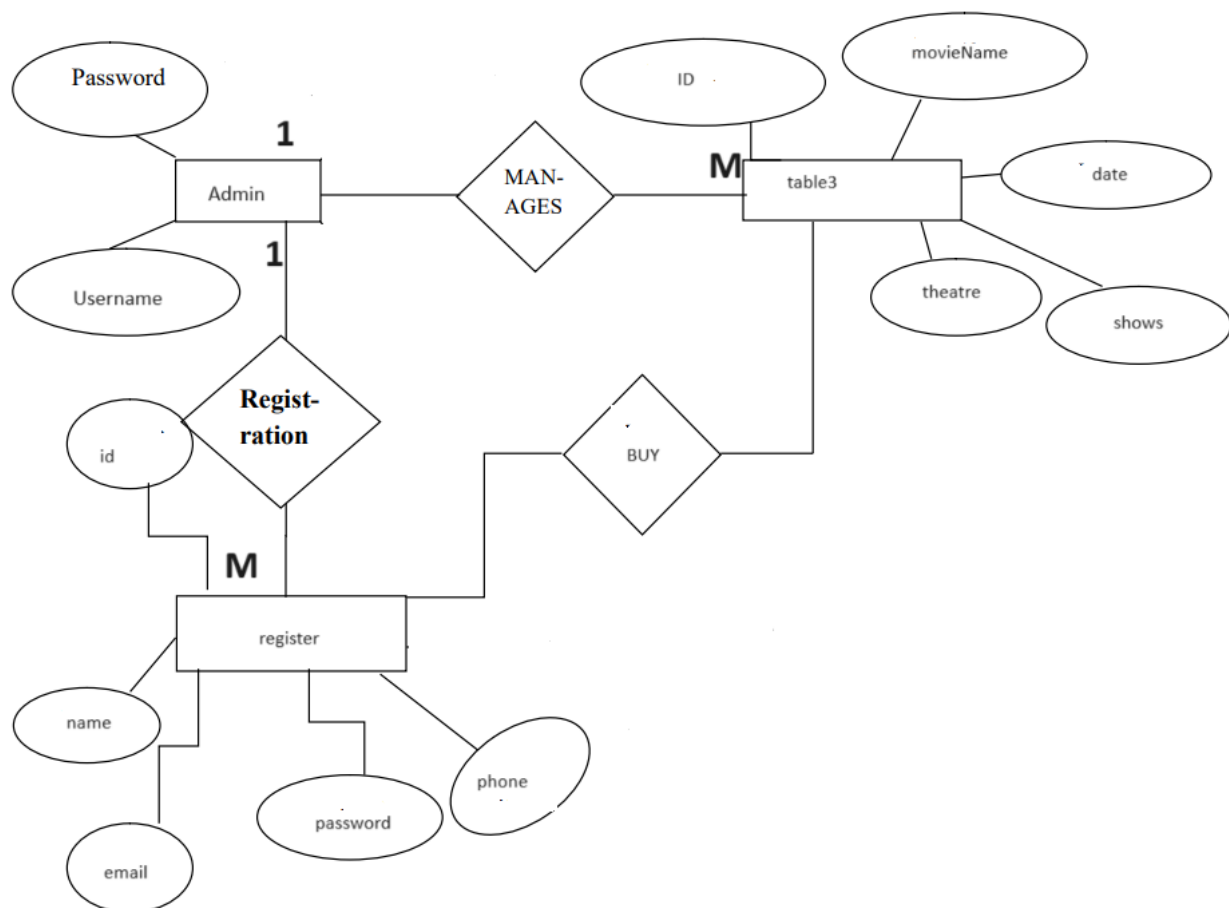


ONE LEVEL DATA FLOW DIAGRAM

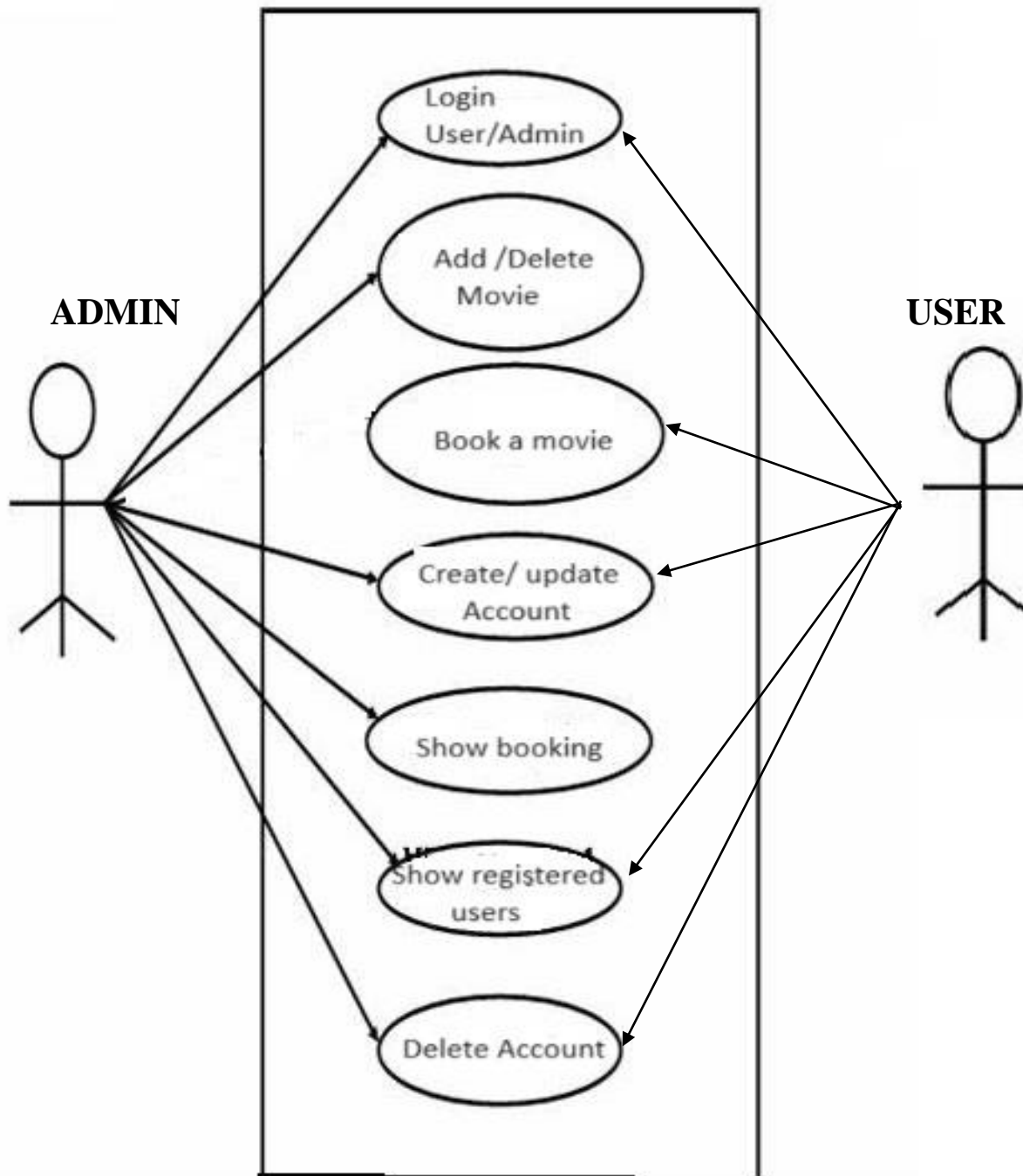
## ENTITY RELATION DIAGRAMS

The Entity Relation Model or Entity Relation Diagram (ERD) is a data model or diagram for high-level description of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity relationship diagrams. Such models are typically used in the first stage of Management information system design; they are used for example, to describe information needs and/ or the type of information that is to be stored in the Database during the requirement analysis. The data modeling technique, however, can be used to describe any ontology (i.e., an overview and classification of used term and their relationships) for a certain universe of discourse (i.e. area of interest).

In the case of design, a Management Information System that is based on a database, the conceptual data model is, a later stage (usually called logical design), mapped to a logical data model such as, relational data model; this is turn in mapped to a physical model during physical design. Note that sometimes, both of the phases are referred a “physical design”. There are number of conventions for entity-relation diagrams (ERDs). The classical notation is described in the remainder of this article, and mainly related to the conceptual modeling. There is a range of notation more typically employed in physical and logical database design.



## USE CASE DIAGRAM



## CHAPTER 5

### TESTING

#### 1. System testing

Black box testing method was used for system testing. The black box testing usually demonstrates that software functions are operational; that the input is properly accepted and the output is correctly produced; and that integrity of external information (databases) is maintained.

#### 2. Why testing is done

- Testing is the process of running a system with the intention of finding errors.
- Testing enhances the integrity of a system by detecting deviations in design and errors in the system.
- Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system.
- Testing also add value to the product by confirming to the user requirements.

#### 3. Causes of error

The most common causes of errors in a software system are:

- **Communication gap between the developer and the business decisionmaker:** A communication gap between the developer and the business decision maker is normally due to subtle differences between them. The differences can be classified into five broad areas: Thought process, Background and Experience, Interest, Priorities, Language.
- **Time provided to a developer to complete the project:** A common source of errors in projects comes from time constraints in delivering a product. To keep to the schedule, features can be cut. To keep the features, the schedule

can be slipped. Failing to adjust the feature set or schedule when problems are discovered can lead to rushed work and flawed systems.

- **Over Commitment by the developer:** High enthusiasm can lead to over commitment by the developer. In these situations, developers are usually unable to adhere to deadlines or quality due to lack of resources or required skills on the team.
- **Insufficient testing and quality control:** Insufficient testing is also a major source of breakdown of e-commerce systems during operations, as testing must be done during all phases of development.
- **Inadequate requirements gathering:** A short time to market results in developers starting work on the Web site development without truly understanding the business and technical requirements. Also, developers may create client-side scripts using language that may not work on some client browsers.
- **Keeping pace with the fast changing Technology:** New technologies are constantly introduced. There may not be adequate time to develop expertise in the new technologies. This is a problem for two reasons. First, the technology may not be properly implemented. Second, the technology may not integrate well with the existing environment.

#### 4. Testing principles

- To discover as yet undiscovered errors.
- All tests should be traceable to customer's requirement.
- Tests should be planned long before the testing actually begins.



- Testing should begin “in the small” & progress towards “testing in the large”.
- Exhaustive Testing is not possible.
- To be most effective training should be conducted by an Independent Third Party

## 5. Testing Objectives:-

- Testing is a process of executing a program with the intent of finding errors.
- A good test case is one that has a high probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

## 6. Kinds of Testing:-

**Black Box Testing-** Not based on any knowledge of internal designs or code. Tests are based on requirements and functionality.

**White Box Testing-** Based on the knowledge of the internal logic of an application’s code. Tests are based on coverage of code statements, branches, paths and statements.

**Unit Testing-** The most ‘micro’ scale of testing; to test particular functions and code modules. Typically done by the programmer and not by the testers, as it requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code; may require developing test driver modules or test harnesses.

**Integration Testing-** Testing of combined parts of an application to determine if they function together correctly. The ‘parts’ can be code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/ server and distributed systems.

**Functional Testing-** Black-box type testing geared to functional requirements of an application; testers should do this type of testing. This doesn't mean that the programmers shouldn't check that their code works before releasing it.

**Regression Testing-** Re-testing after fixes or modifications of the software or its environment. It is difficult to determine how much re testing is needed, especially near the end of the development cycle. Automated testing tools can be especially useful for this type of testing.

**Acceptance Testing-** Final testing based on the specifications of the end user or customer or based on use by end-users/ customers over some limited period of time.

**User Acceptance Testing-** Determining if software is satisfactory to an end user customer.

## 7. Testing Technique Used

We will continuously test our project to insure that it is fully functional. In order to perform testing test cases are designed with the intent of finding the errors in the project and help in removing those errors. Testing begins at the module level and is conducted systematically. It is generally conducted by independent test groups or third party.

Testing is done in our project Multiplex Ticket Booking System with help of black box testing that exercise all the functional requirement of the project test cases are designed using this approaches by providing set of input conditions to get the expected output.

### **Test Cases Designed**

#### **TEST CASES FOR CLIENT:-**

TEST CASE ID	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Password and key combination	Start	Operation started
2	No Input	Alert Message	Select a valid configuration setting

## **TEST CASES FOR SERVER:-**

TEST CASE ID	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	127.0.0.1	Connection establishes	Client Connected
2	Get Status	Alert Box	If no client selected “alert box” is displayed
3	Choose Path	Transfer files to valid path	Files transferred

## CHAPTER 6

### SCREENSHOT OF DATATABLE



Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> admin		1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> register		4	MyISAM	latin1_swedish_ci	2.2 KiB	-
<input type="checkbox"/> table3		7	MyISAM	latin1_swedish_ci	2.7 KiB	356 B
3 tables	Sum	12	InnoDB	utf8mb4_general_ci	20.9 KiB	356 B

### TABLE FOR ADMIN

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	AName	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/> 2	APassword	varchar(50)	utf8mb4_general_ci		No	None			Change  Drop  More

### TABLE FOR USERS

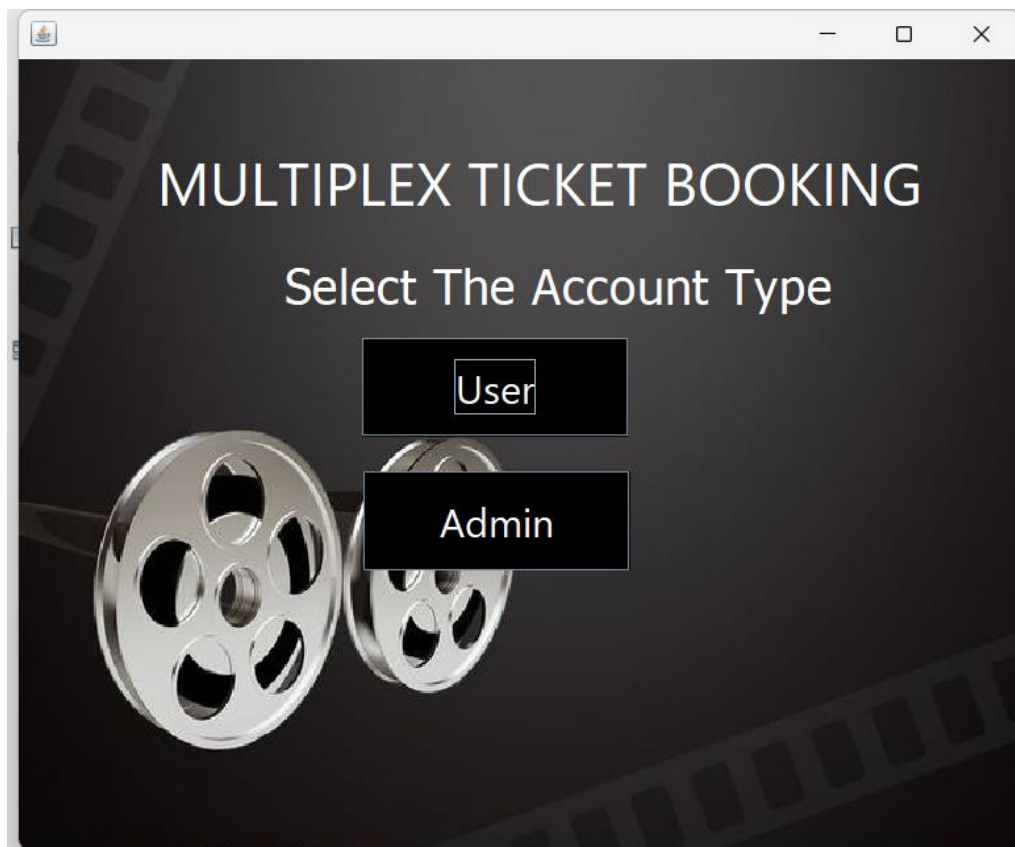
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	name	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	email	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	password	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 5	phone	varchar(50)	latin1_swedish_ci		No	None			Change  Drop  More

### TABLE FOR MOVIES

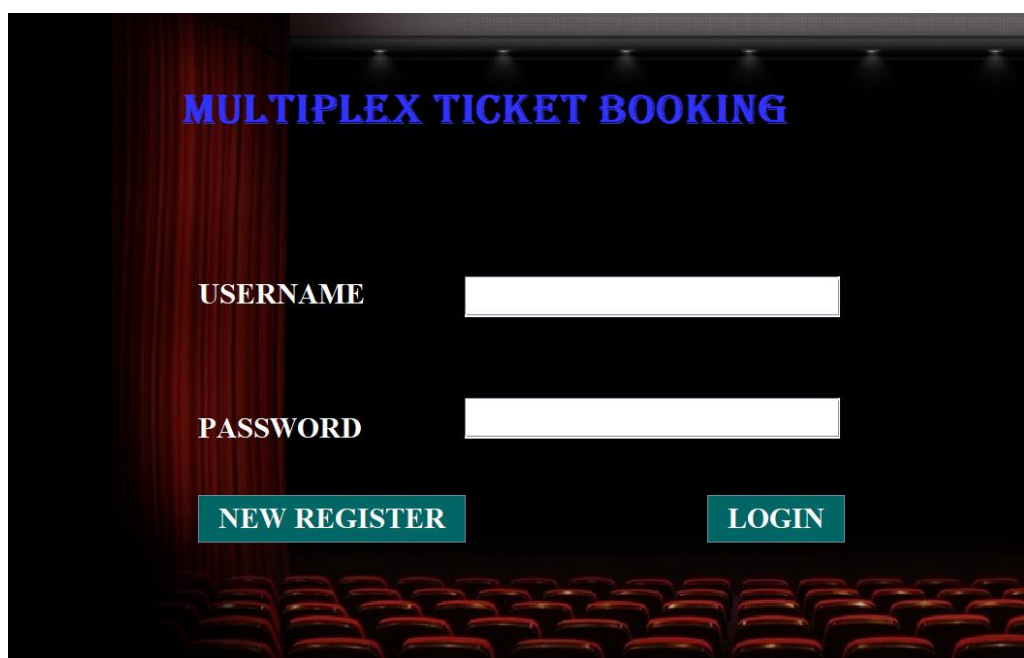
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	movieName	varchar(50)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	date	varchar(50)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 4	theatre	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 5	shows	varchar(30)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 6	tickets	int(5)			No	None			Change  Drop  More

## PROJECT SCREENSHOT

### Main page



### User login



## Movie selection

**MOVIE DETAILS**

SELECT MOVIE:	<input type="text" value="Select a movie"/>
SELECT THEATRE:	<input type="text" value="Select a theatre"/>
SELECT DATE:	<input type="text" value="Select a date"/>
SELECT TIME:	<input type="text" value="Select a show"/>
NO OF TICKETS:	<input type="text"/>
<input type="button" value="SUBMIT"/>	

## Booking details

**BOOKING DETAILS**

NO OF TICKETS:	3
THEATRE:	Inox
MOVIE:	raftaar
DATE:	30-Sept-2023
SHOW:	3:00AM
FARE:	300
<input type="button" value="BOOK"/> <input type="button" value="CANCEL"/>	

## Conformation details

### CONFIRM DETAILS

NO OF TICKETS: 3

THEATER: Inox

MOVIE: raftaar

DATE: 30-Sept-2023

FARE: 300

TIME: 3:00AM

### PAYMENT TYPE

- ☐ Credit card
- ☐ Debit card
- ☐ NET BANKING
- ☐ PAYTM WALLET

Make Payment

Go Back

## Payment

### CONFORM BOOKING

NO OF TICKETS: 3

THEATER: Inox

MOVIE: raftaar

DATE: 30-Sept-2023

FARE: 300

TIME: 3:00AM

### CARD DETAILS

Card number

Card holder name

CVV  EXP  01  2018

PROCEED

## Receipt

### RECEIPT

NO OF TICKETS: 3

THEATRE: Inox

MOVIE: 30-Sept-2023


DATE: 3:00AM

SHOW: raftaar

FARE: 300

THANKS FOR BOOKING

## Admin Login



Admin Name:

Admin Password:

## Admin Home


(a).

### Movies & Shows

id	Movie name	Date	Theater	Shows	Tickets
28	Nun 2	11-Oct-2023	PVR	12:00PM	194
22	raftaar	29-Sept-2023	chandi	2:00PM	1
27	Nun 2	03-Oct-2023	Inox	9:00PM	296
26	rasees	02-Oct-2023	Inox	12:00pm	500
23	race	29-Sept-2023	inox	12:00AM	8
24	raftaar	30-Sept-2023	Inox	3:00AM	430
25	raftaar	30-Sept-2023	galaxy	12:00AM	43

#### Add A Show

Movie Name

Date   Shows

Theater  Tickets

#### Delete A Show

Enter the movie id



(b).

## Registered User's

Id	Name	Email	Phone
5	sangam	sangam@gmail.com	1234234
6	prashant	prashant@gmail.com	1234234
7	soni	soni@gmail.com	2342345
4	siddharth	dfy6trfgh	345

MOVIES

LOGOUT

### Add User

Name

Email

Phone

Password

Submit

### Delete A User

Enter the User id:

Delete

## **Future Scope of Project**

The proposed system helps them in many ways. It helps them do billing very easily. Account maintenance also becomes easier. They can keep track of their all-movie details and customer account details. The software is provided with all the master entries to enter any new movies or to add or modify and delete the movie shows and timings.

As this is generic software it can be used by a wide variety of multiplex cinema halls to automate the process of manually maintaining the records related to the subject of maintaining the movie details and customer data.

In future it can be modify, so that it can be done online.

## **CONCLUSION**

This section discuss the result of work done in this project and also mentions the future scope improvement.

The software will be developed by implementing the concept of modularity which in turn reduces the complexity involved in maintaining it. The administrator should have a sound technical knowledge about maintaining the software and further enhancements will be undertaken by the developer.

The application is portable which ensure its adaptability for use on different computer terminals with different operating system and standards.

The factors guarantee the software's availability includes proper termination and correct input details. Also, the resources used for the project development are Microsoft certified which speaks of its high-quality standards.

Hence, we may conclude that the application system being developed helps a great deal in modifying the computerized MULTIPLEX TICKET BOOKING SYSTEM.

## REFERENCE

### **Book References**

The following books were referred during the analysis and execution phase of the project

**Common Language Runtime** -By Steven Pratschner

**SOFTWARE ENGINEERING**-By Roger S. Pressman

**UNIFIED MODELING LANGUAGE**-By Gradi Booch, Ivar Jacobson, James  
Rambaugh

**COMPLETE REFERENCE .NET**-By David S Platt

**MSDN 2003** -By Microsoft

**IMAGES**

Bing Search

Msn Search

Google Search

### **Designing and implementation phase: -**

1. Software engineering: a practitioners approach by roger s pressman.
2. System analysis and design by Elias m. Ewad.
3. DBMS : Bipin C Desai
4. Introduction to dbms – shailendrasharma
5. Modern database management -drkumar
6. Fundamentals of database systems- ramezelmsari
7. An intro to dbms- christopher j date
8. Database system concepts-abrahamsilbersch
9. Database management systems-raghuramkrishnan
10. Principles of database systems- jd Ullman
11. Foundation of databases- sergebabiteboul
12. Database systems-connolly

### **WEBSITES:-**

- [www.google.com](http://www.google.com)
- [www.netbeans.com](http://www.netbeans.com)
- [www.sql.com](http://www.sql.com)
- [www.wikipedia.com](http://www.wikipedia.com)
- [www.ask.com](http://www.ask.com)

THANKYOU