MOVIE TICKET BOOKING SYSTEM

PROJECT REPORT

18CSC202J- OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY

(2018 Regulation)

II Year/ III Semester

Academic Year: 2022 -2023

By

Parth Galhotra (RA2111032010029)

Rasesh Rajeev (RA2111032010019)

Under the guidance of

Dr. GOUTHAMAN. P

Assistant Professor

Department of Networking and Communications

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

NOVEMBER 2022

BONAFIDE

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY project report** titled "**MOVIE TICKET BOOKING SYSTEM**" is the bonafide work of **Parth Galhotra(RA2111032010029), Rasesh Rajeev (RA2111032010019)** who undertook the task of completing the project within the allotted time.

Signature of the Guide	**Signature of the II Year Academic Advisor**
Dr. Gouthaman. P	------------------------
Assistant Professor	**Professor and Head**
Department of NWC	Department of NWC
SRM Institute of Science and Technology	SRM Institute of Science and Technology

About the course:-

18CSC202J/ 8AIC203J - Object Oriented Design and Programming are 4 credit courses with **L T P C as 3-0-2-4** (Tutorial modified as Practical from 2018 Curriculum onwards)

Objectives:

The student should be made to:

- Learn the basics of OOP concepts in C++
- Learn the basics of OOP analysis and design skills.
- Be exposed to the UML design diagrams.
- Be familiar with the various testing techniques

Course Learning Rationale (CLR): The purpose of learning this course is to:

1. Utilize class and build domain model for real-time programs
2. Utilize method overloading and operator overloading for real-time application development programs
3. Utilize inline, friend and virtual functions and create application development programs
4. Utilize exceptional handling and collections for real-time object-oriented programming applications
5. Construct UML component diagram and deployment diagram for design of applications
6. Create programs using object-oriented approach and design methodologies for real-time application development

Course Learning Outcomes (CLO): At the end of this course, learners will be able to:

1. Identify the class and build domain model
2. Construct programs using method overloading and operator overloading
3. Create programs using inline, friend and virtual functions, construct programs using standard templates
4. Construct programs using exceptional handling and collections
5. Create UML component diagram and deployment diagram
6. Create programs using object oriented approach and design methodologies

Table 1: Rubrics for Laboratory Exercises

(Internal Mark Splitup:- As per Curriculum)

CLAP-1	5=(2(E-lab Completion) + 2(Simple Exercises)(from CodeZinger, and any other coding platform) + 1(HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-2	7.5=(2.0(E-lab Completion)+ 2.0 (Simple Exercises)(from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-3	7.5=(2.0(E-lab Completion(80 Pgms)+ 2.0 (Simple Exercises)(from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	**2 Mark -** E-lab Completion **80 Program** Completion from 10 Session (Each session min 8 program) **2 Mark -** Code to UML conversion GCR Exercises **3.5 Mark - Hacker Rank** Coding challenge completion
CLAP-4	5= 3 (Model Practical) + 2 (Oral Viva)	• **3 Mark** – Model Test • **2 Mark** – Oral Viva
Total	25	

COURSE ASSESSMENT PLAN FOR OODP LAB

S.No	List of Experiments	Course Learning Outcomes (CLO)	Blooms Level	PI	No of Programs in each session
1.	Implementation of I/O Operations in C++	CLO-1	Understand	2.8.1	10
2.	Implementation of Classes and Objects in C++	CLO-1	Apply	2.6.1	10
3,	To develop a problem statement. 1. From the problem statement, Identify Use Cases and develop the Use Case model. 2. From the problem statement, Identify the conceptual classes and develop a domain model with a UML Class diagram.	CLO-1	Analysis	4.6.1	Mini Project Given
4.	Implementation of Constructor Overloading and Method Overloading in C++	CLO-2	Apply	2.6.1	10
5.	Implementation of Operator Overloading in C++	CLO-2	Apply	2.6.1	10
6.	Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams and Collaboration diagrams	CLO-2	Analysis	4.6.1	Mini Project Given
7.	Implementation of Inheritance concepts in C++	CLO-3	Apply	2.6.1	10
8.	Implementation of Virtual function & interface concepts in C++	CLO-3	Apply	2.6.1	10
9.	Using the identified scenarios in your project, draw relevant state charts and activity diagrams.	CLO-3	Analysis	4.6.1	Mini Project Given
10.	Implementation of Templates in C++	CLO-3	Apply	2.6.1	10
11.	Implementation of Exception of Handling in C++	CLO-4	Apply	2.6.1	10
12.	Identify the User Interface, Domain objects, and Technical Services. Draw the partial layered, logical architecture diagram with UML package diagram notation such as Component Diagram, Deployment Diagram.	CLO-5	Analysis	4.6.1	Mini Project Given
13.	Implementation of STL Containers in C++	CLO-6	Apply	2.6.1	10
14.	Implementation of STL associate containers and algorithms in C++	CLO-6	Apply	2.6.1	10
15.	Implementation of Streams and File Handling in C++	CLO-6	Apply	2.6.1	10

LIST OF EXPERIMNENTS FOR UML DESIGN AND MODELLING:

To develop a mini-project by following the exercises listed below.

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant state charts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.

Suggested Software Tools for UML:

StarUML, Rational Suite, Argo UML (or) equivalent, Eclipse IDE and Junit

ABSTRACT

The project objective is to book cinema tickets in online. The Ticket Reservation System is an Internet based application that can be accessed throughout the Net and can be accessed by anyone who has a net connection. This application will reserve the tickets. This online ticket reservation system provides a website for a cinema hall where any user of internet can access it. User is required to login to the system and needs a credit card for booking the tickets. Tickets can be collected at the counter and Watching movies with family and friends in theatres is one of the best medium of entertainment after having a hectic schedule. But all this excitement vanishes after standing in hours in long queues to get tickets booked. The website provides complete information regarding currently running movies on all the screens with details of show timings, available seats. Ticket reservations are done using credit card and can be cancelled if needed. Our online tickets reservation system is one of the best opportunities for those who cannot afford enough time to get their tickets reserved standing in long queues. People can book tickets online at any time of day or night. Our reservation system also provides option to cancel the tickets which are reserved previously.

MODULE DESCRIPTION

INTRODUCTION:

THE MAIN PURPOSE OF OUR ONLINE TICKET BOOKING SYSTEM IS TO PROVIDE AN ALTERNATE AND CONVENIENT WAY FOR A CUSTOMER TO BUY CINEMA TICKETS. IT IS AN AUTOMATIC SYSTEM. AFTER THE DATA HAS BEEN FED INTO THE DATABASE, THE

STAFF DOES NOT NEED TO DO ANYTHING WITH THE ORDER ONCE IT IS RECEIVED THROUGH THE SYSTEM

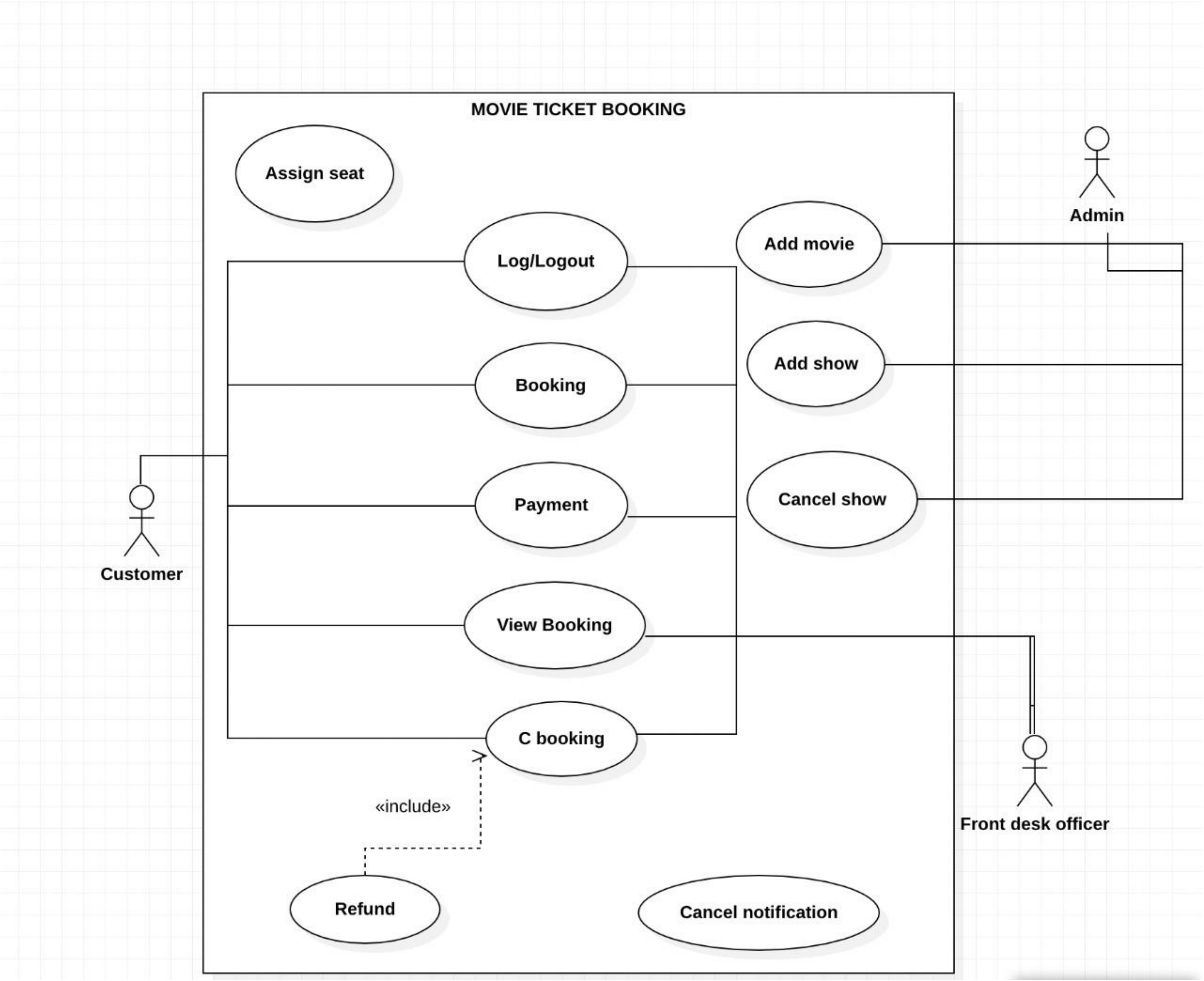
FEATURES OF MOVIE TICKET BOOKING SYSTEM:

- You can show the models the components of the Movie Ticket Booking System
- Model the Database Schema of Movie Ticket Booking System
- Model the Executables of an application of Movie Ticket Booking System.
- Model the system's source code of Movie Ticket Booking System.

SPECIFICATIONS:

- First it will take your login id and password or create an entry in the database if you are a first time user.
- Then it will check if the id and password that you have entered is correct
- Then it will check for permissions.
- It has many buttons for managing different categories like customer, shows, movie, seats, payment.
- It also has a logout function built in it for the next customer

Use Case Diagram with Explanation

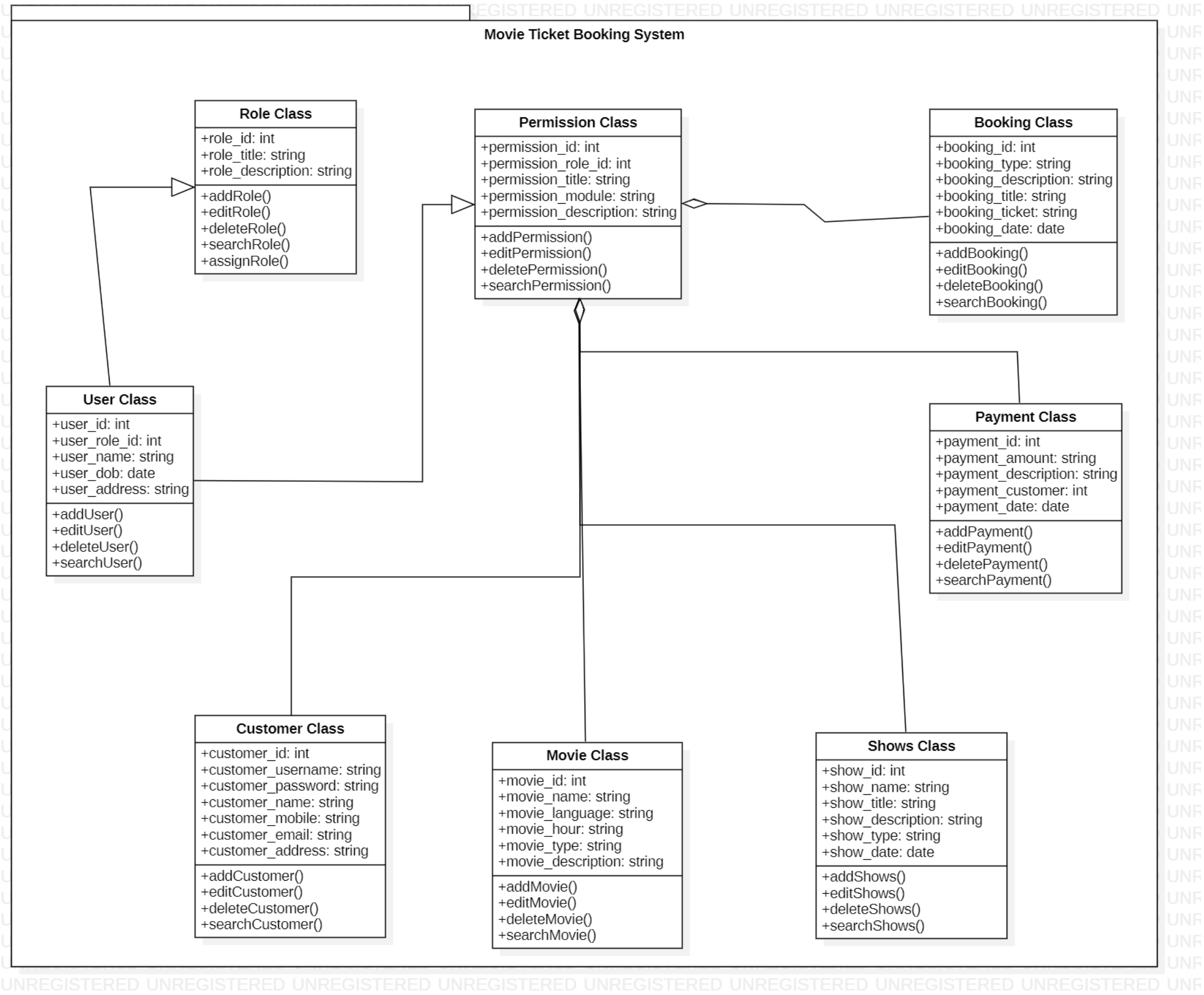


This Use Case Diagram is a graphic depiction of interactions among the elements of Movie Ticket Booking System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of the system. The main actors of the system in the Use Case Diagram are: Customer, Admin, Front Desk Officer, who perform different type of use cases such as Log/LogOut, Add Show, Cancel Show, View Booking, Payment

Relationships between and among the actors and the use cases of Movie Ticket Booking System:

- **Customer Entity:** Use cases are Log/Logout, Booking, Payment, View Booking, C Booking
- **Front Desk Officer Entity:** Use cases are View Booking, Log/Logout, Booking, Payment, C Booking
- **Admin Entity:** Use cases are Add Movie, Add Show, Cancel Show

Class Diagram with Explanation

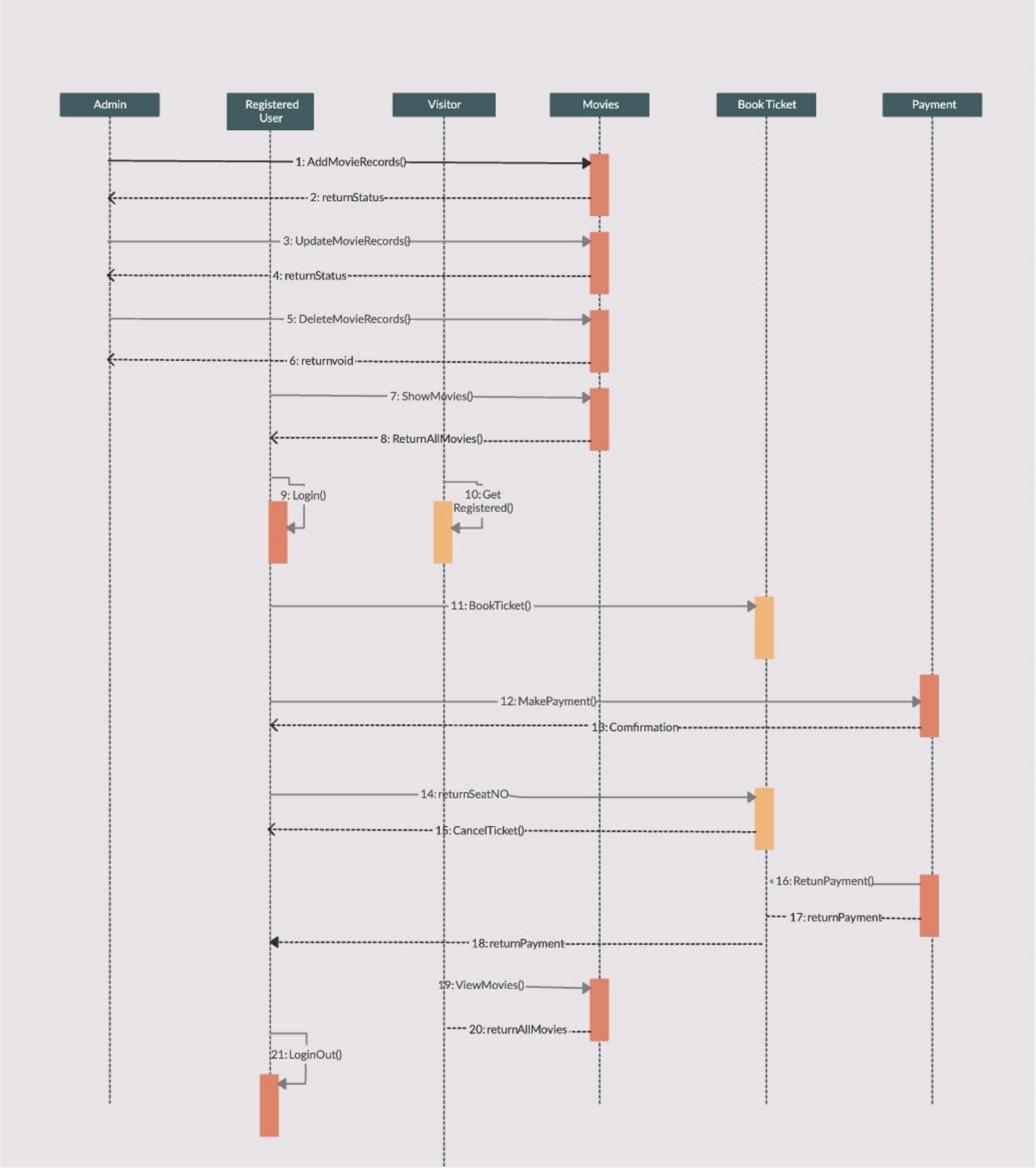


Movie Ticket Booking System Class Diagram describes the structure of a system's classes, their attributes, operations (or methods), and the relationships among objects. The main classes of the system are Movie, Customer Booking, Payment, Seats, Shows.

Classes of the Class Diagram:

- **Movie Class:** Manage all the operations of Movie
- **Customer Class:** Manage all the operations of Customer
- **Booking Class:** Manage all the operations of Booking
- **Payment Class:** Manage all the operations of Payment
- **Seats Class:** Manage all the operations of Seats
- **Shows Class:** Manage all the operations of Show

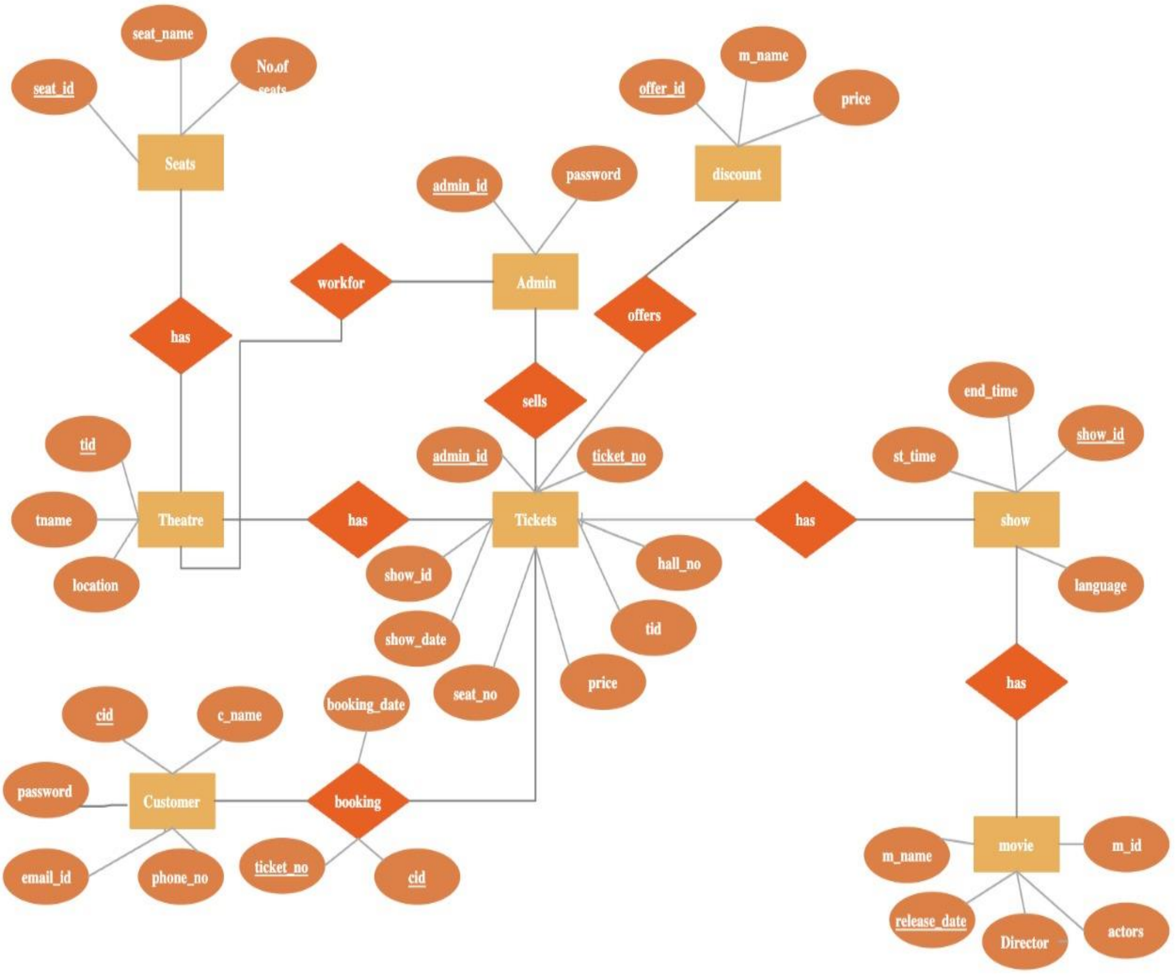
Sequence Diagram with Explanation



This diagram shows the interaction between the objects of the system. The instance of class objects involved in this Sequence Diagram are as follows:

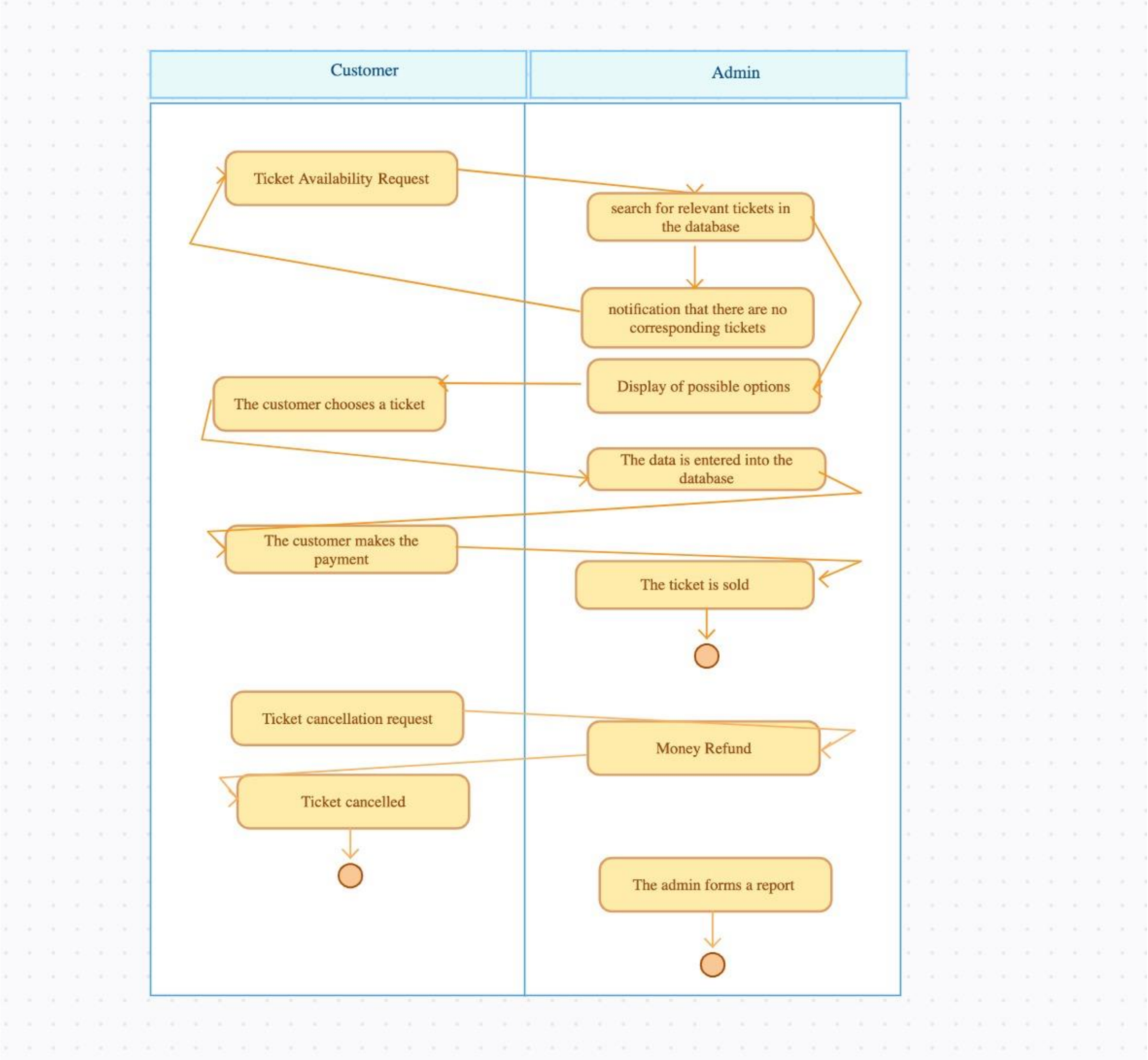
- Admin Object
- Registered User Object
- Visitor Object
- Movies Object
- Book Ticket Object
- Payment Object

Communication Diagram with Explanation



This diagram represents the model of the system entity. It shows all the visual instruments of the database table. It used structure data to define the relationships between structured data groups of the Movie Ticket Booking System functionalities.

State Chart Diagram with Explanation

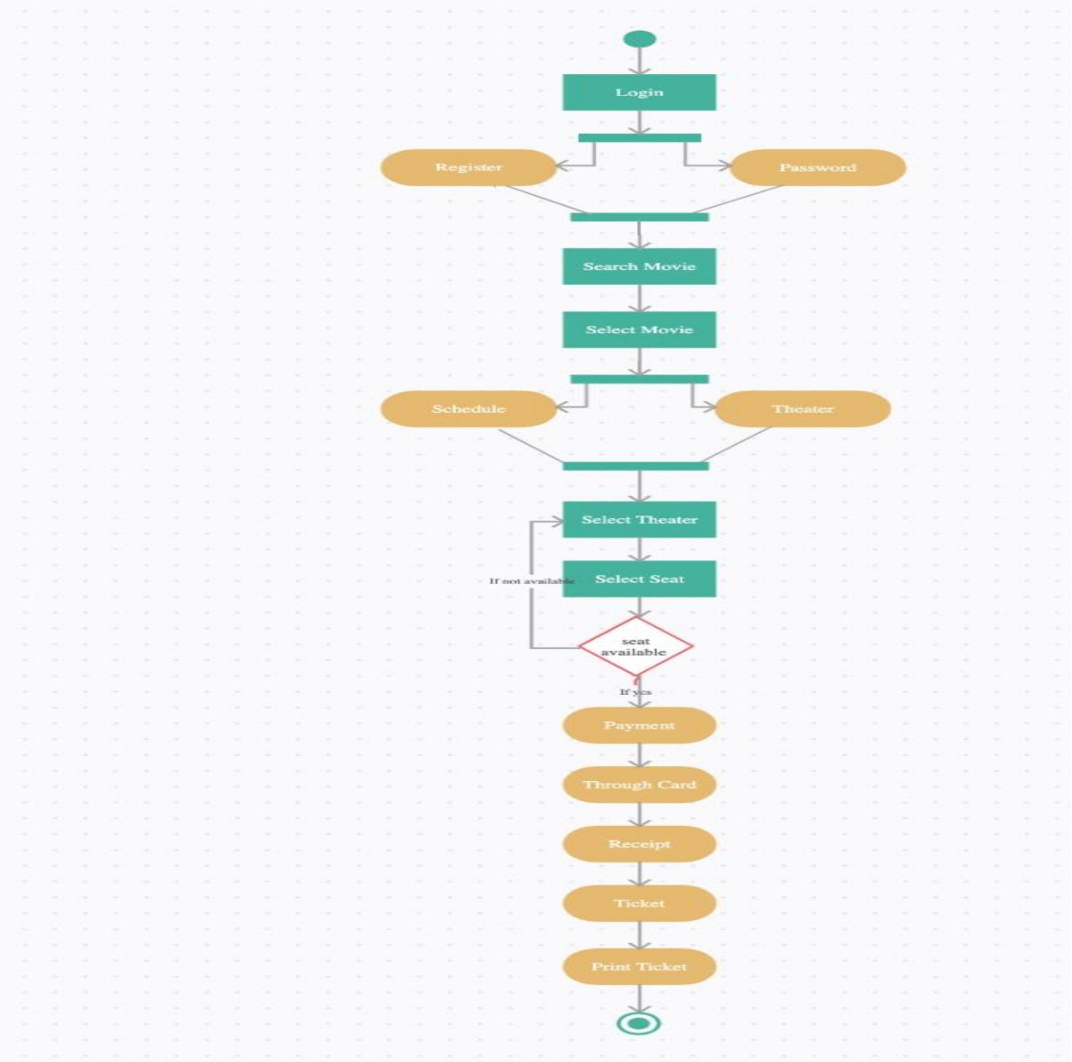

Customer
Admin
Ticket Availability Request
search for relevant tickets in the database
notification that there are no corresponding tickets
Display of possible options
The customer chooses a ticket
The data is entered into the database
The customer makes the payment
The ticket is sold
Ticket cancellation request
Money Refund
Ticket cancelled
The admin forms a report

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioral diagram and it represents the behavior using finite state transitions.

Uses of statechart diagram

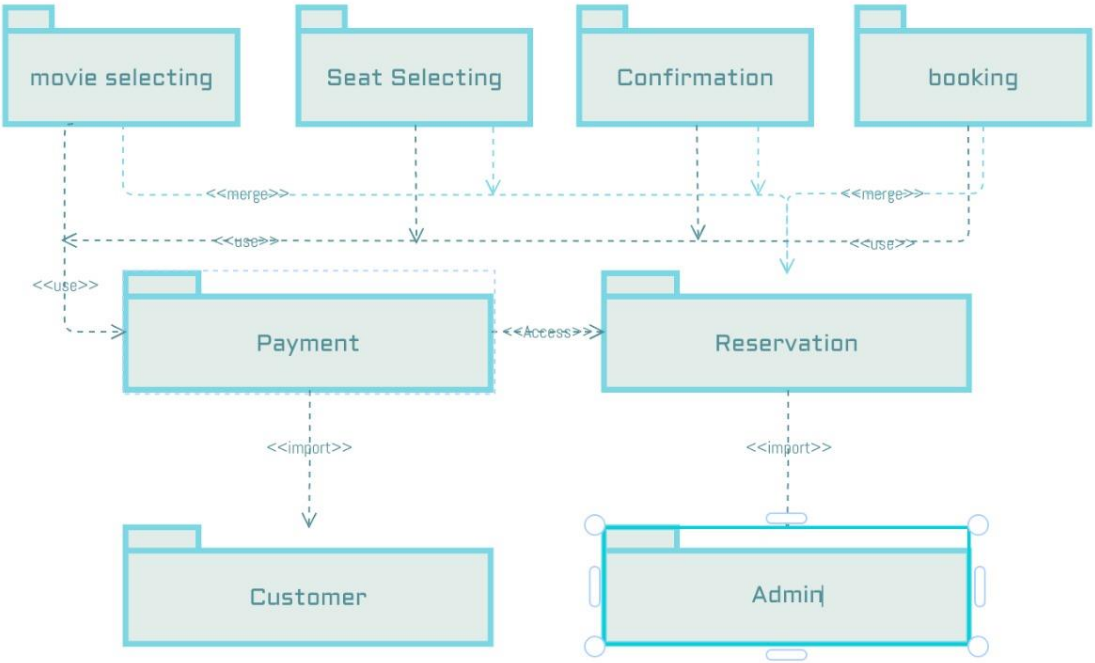
- We use it to state the events responsible for change in state (we do not show what processes cause those events).
- We use it to model the dynamic behavior of the system .
- To understand the reaction of objects/classes to internal or external stimuli.

Activity Diagram with Explanation



This is the Activity Diagram of the system which shows the flows of the systems Activity, where admin will be able to login using their username and password. After login user can manage all operations on Payments, Booking, Shows, Customer. All pages are secure and user can access these pages after login. The diagram helps to understand how the program will work. The various objects in these pages interacts over the course of the Activity, and user will not be able to access this page without verifying their identity.

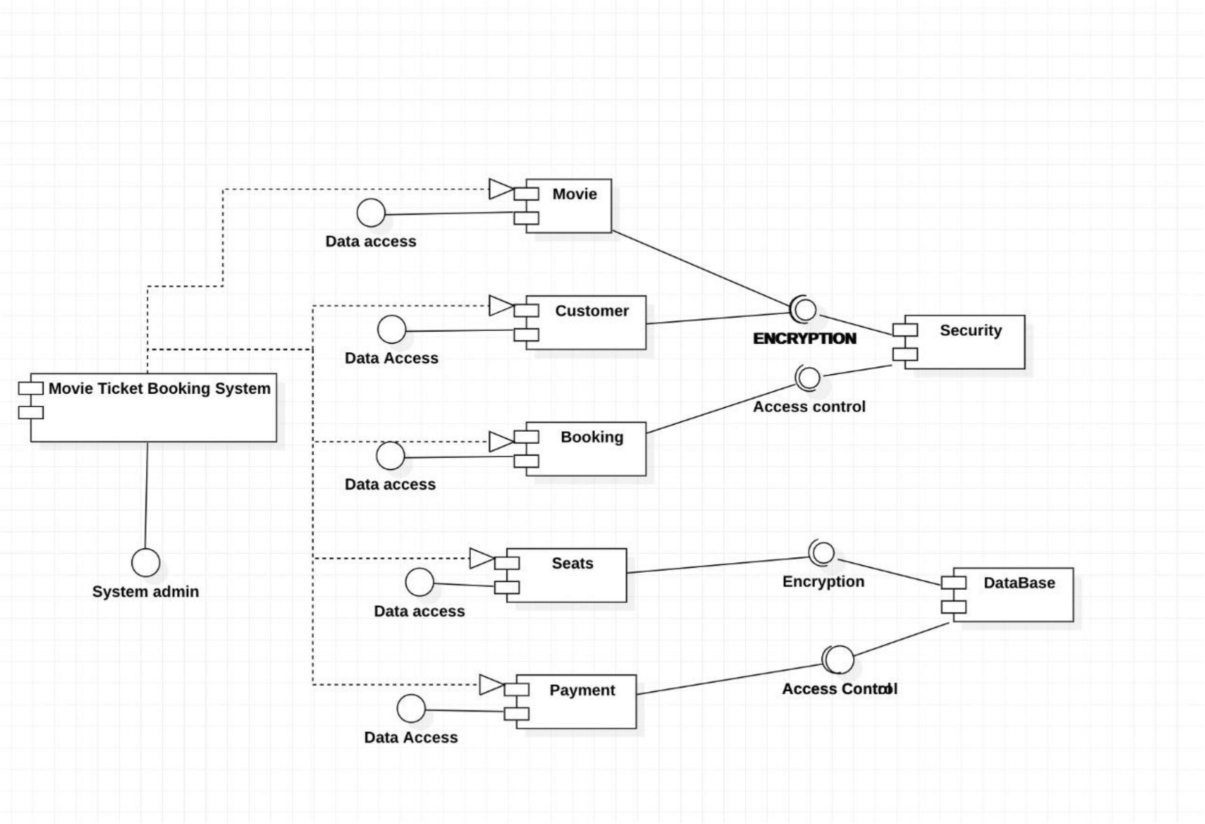
Package Diagram with Explanation



It shows the arrangement and organization of model elements in middle to large scale project. Package diagram can show both structure and dependencies between sub-systems or modules, showing different views of a system, for example, as multi-layered (aka multi-tiered) application - multi-layered application model. Package diagrams are used to structure high level system elements. Packages are used for organizing large system which contains diagrams, documents and other key deliverables.

Package Diagram can be used to simplify complex class diagrams, it can group classes into packages.

Component Diagram with Explanation



This shows components, provided and required interfaces,ports,and relationships between Shows, Seats, Customer, Payment and Movie. Tjis type of diagram is used in Component-Based Development(CBD) to describe systems with Service-Oriented Architecture(SOA).Movie Ticket Booking System UML Component Diagram describes the organization and wiring of the physical components in a system.

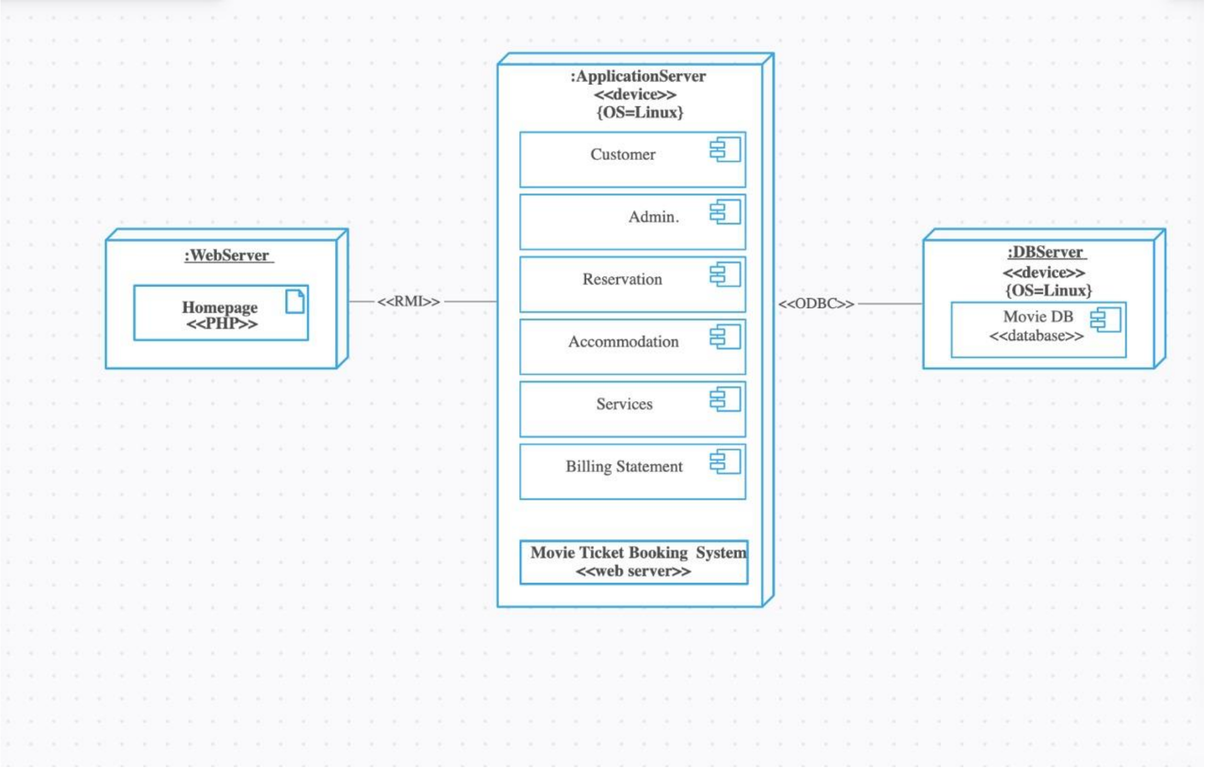
Components of Component Diagram:

- Shows Component
- Seats Component
- Customer Component
- Payment Component
- Movie Component

Features of Component Diagram:

- You can show the models the component of the system.
- Model the database schema of the system.
- Model the executables of an application of the system.
- Model the system’s source code

Deployment Diagram with Explanation



The Deployment Diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware). They show the structure of the run-time system.

- They capture the hardware that will be used to implement the system and the links between different items of hardware.
- They model physical hardware elements and the communication paths between them
- They can be used to plan the architecture of a system.
- They are also useful for Document the deployment of software components or nodes

Conclusion

This project is developed successfully and the performance is found to be satisfactory. This project is designed to meet the

requirements of assigning jobs

The user will be able to book the ticket using this website. The relationship between company manager, employee, and

customer satisfy a good communication to complete ticketing process.

We have designed the project to provide the user with easy retrieval of data, details of theatre and necessary feedback as much

as possible. In this project, the user is provided with a website that can be used to book movie tickets online

References

1. www.google.com
2. www.w3schools/html.com
3. www.udemy/webdevelopercourse.com
4. www.stackoverflow.com
5. www.reasearchgate.net
6. www.freeprojectz.com