

OODP PROJECT

UML – MOVIE TICKET BOOKING SYSTEM



Group Members

Name	Registration Number
Vaishali Vinayaga	RA2011026010100
Chereddy Sowmya Sri	RA2011026010113
Parkavi	RA2011026010119

INDEX

S.NO	CONTENTS
1	Abstract
2	Use Case Diagram
3	Class Diagram
4	Dynamic Diagram (Interaction Diagram & Behavior Diagram)
5	Sequence Diagram
6	Collaboration Diagram
7	Activity Diagram
8	State Chart Diagram
9	Component Diagram
10	Deployment Diagram
11	Conclusion

ABSTRACT

Movie Ticket Booking System is computerized system that provide the users' facility to book tickets for a show and to collect information regarding the films and theatres. Client has to register at the positioning to book tickets to the show. When choosing the show, the user is given a seating layout in order that he will choose seats of his selection.

The main purpose of the online ticket booking and purchasing system is to provide a convenient way of customers to buy cinema tickets. This software has two parts. First is user part and the other one is the administrator part. User part is used as a front end and administrator is the back end.

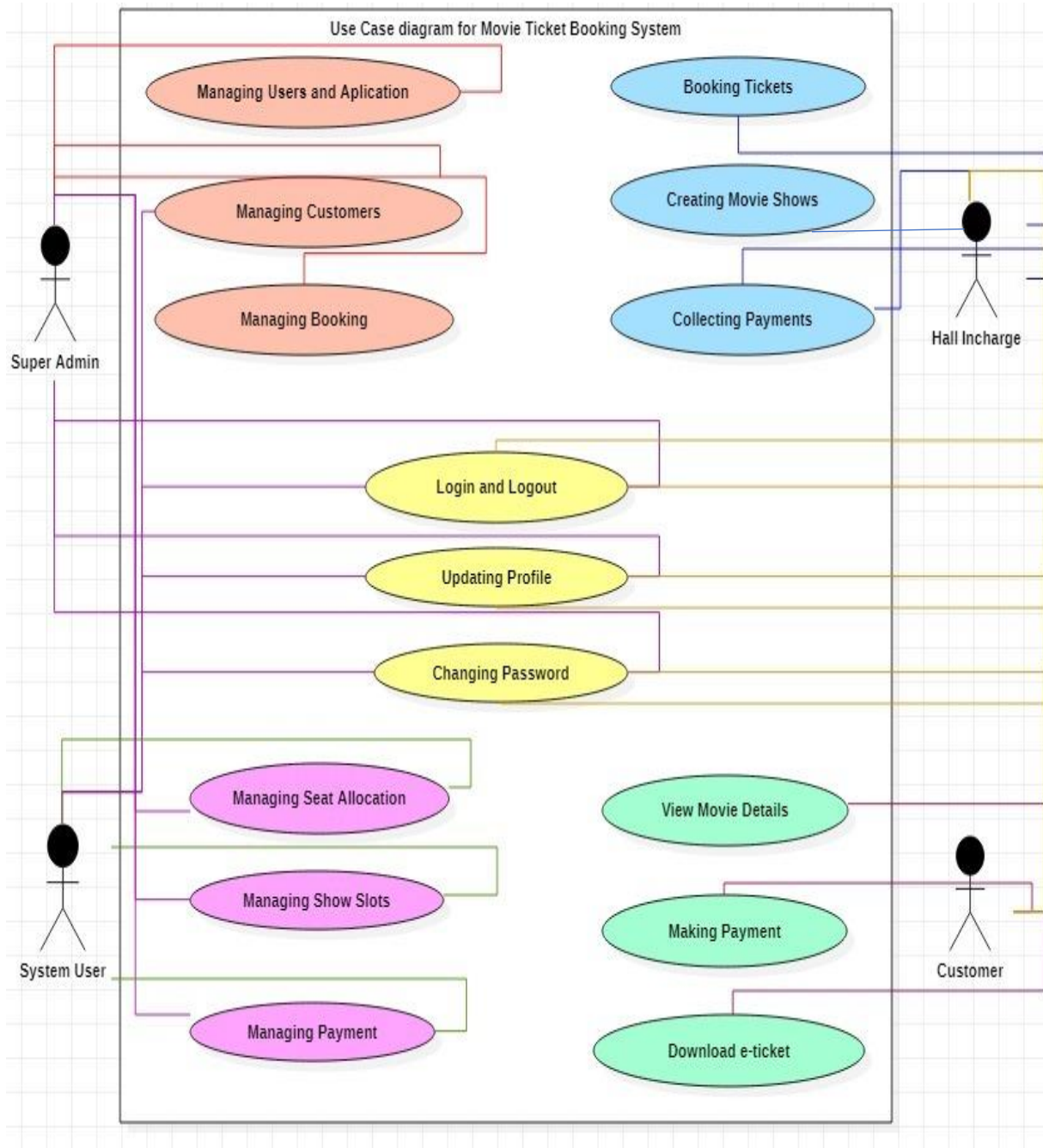
So, the system displays all the necessary details for the user such as the movie name, show timings, seat allocations and the mode of payments available. Once the user selects the movie, it checks for the availability of seats and proceed to the payment gateway. Otherwise, the user is asked to show a different show timing and check for availability. After the selection of the movie, the user is asked to create an account or login for booking the ticket.

Next, to book a ticket the user is asked for his personal details such as name, email id, city, pin code, phone number and mode of payment. Once the user gives all the details, its redirected to the payment gateway and the booking is successful if the payments credentials are valid. Finally, the e-ticket copy is sent to the mobiles and mails.

USE CASE DIAGRAM

1. Use case diagram is used to model the system of an application which captures a particular functionality and dynamic aspect of the system.
2. This Use Case Diagram is a graphic depiction of the 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 Movie Ticket Booking System.
3. The main actors of Movie Ticket Booking System in this Use Case Diagram are: Super Admin, System User, Hall Owners, Customer, who perform the different type of use cases such as Manage Movie, Managing Customer, Managing Booking, Managing Payment, Managing Seat Allocation and Managing show slots
4. Relationship between the actors and the use cases of Movie Ticket Booking System:
 - Super Admin Entity: Use cases of Super Admin are Managing users and application, Managing Customer and Managing Booking.
 - System User Entity: Use cases of System User are Managing seat allocation, managing show slots and managing payment.
 - Hall In-charge Entity: Use cases of Hall Owners are Booking tickets, creating movie shows and collecting payments.
 - Customer Entity: Use cases of Customer are View movie details, making payment and downloading e-ticket.

USE CASE DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



CLASS DIAGRAM

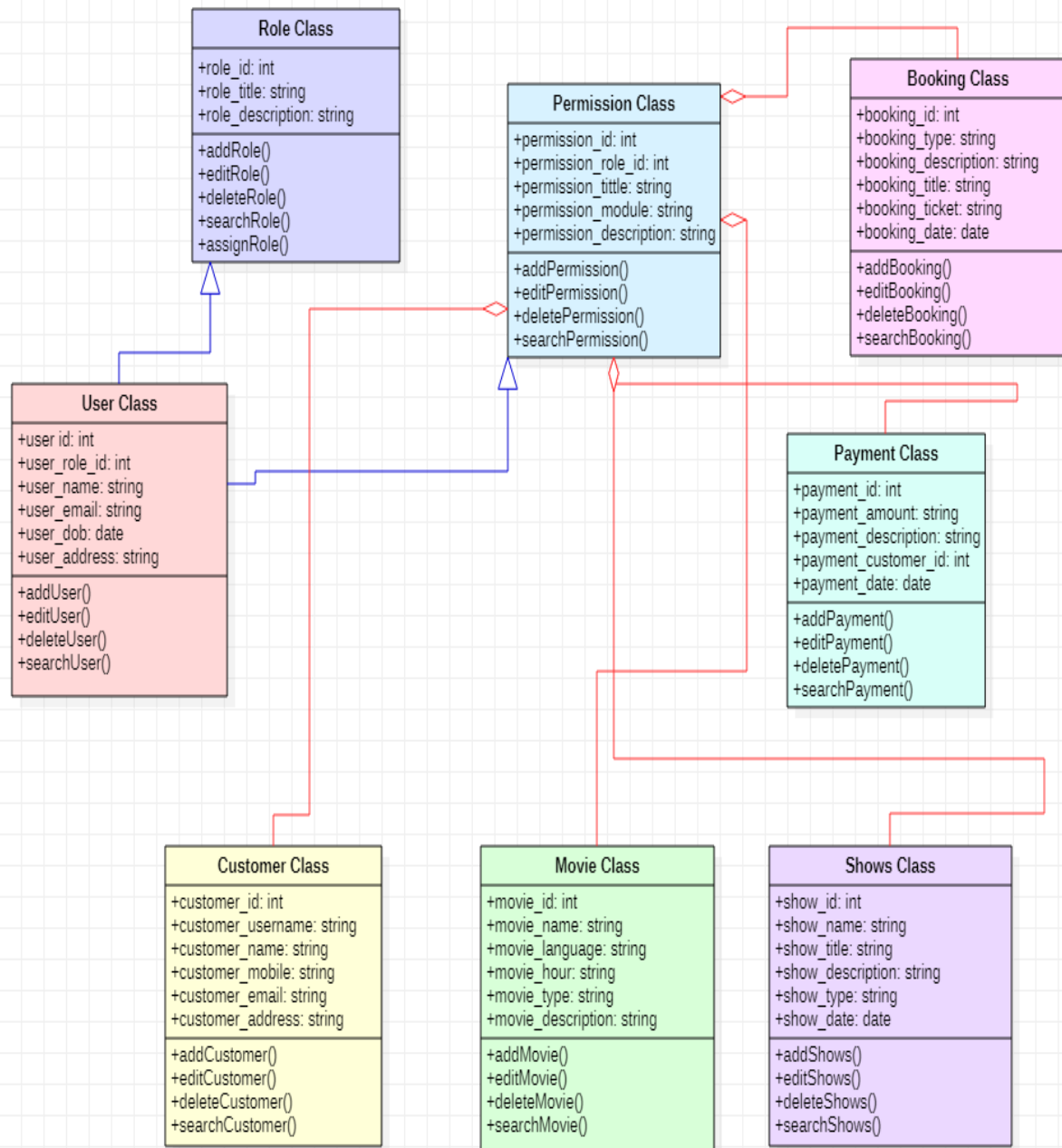
1. A class diagram describes the structure class, attributes, operations and relationship among various objects.
2. The main classes of the Movie Ticket Booking System are Movie, Customer, Booking, Payment, Seats, Shows.
3. Classes:
 - Movie Class : To manage movie operations
 - Customer Class : To manage customer services
 - Booking Class: To manage movie booking
 - Payment Class: To manage payment methods
 - Seat Class: To manage seat allocations
 - Show Class: To manage movie shows/slots
4. Classes and their Attributes:
 - Movie Attributes:
 - movie_id, movie_name, movie_language, movie_type, movie_hour, movie_description .
 - Customer Attributes: customer_id, customer_name, customer_mobile, customer_email, customer_username, customer_password, customer address
 - Booking Attributes:
 - booking_id, booking_title, booking type, booking ticket, booking_date, booking description
 - Payment Attributes:
 - payment_id, payment customer_id, payment_date, payment amount, payment_description
 - Seats Attributes:
 - seat_id, seat_movie id, seat_customer_id, seat_number, seat_type, seat_description

- Shows Attributes:
 - show_id, show_name, show type, show_time, show_date, show_description

5. Classes and their Methods:

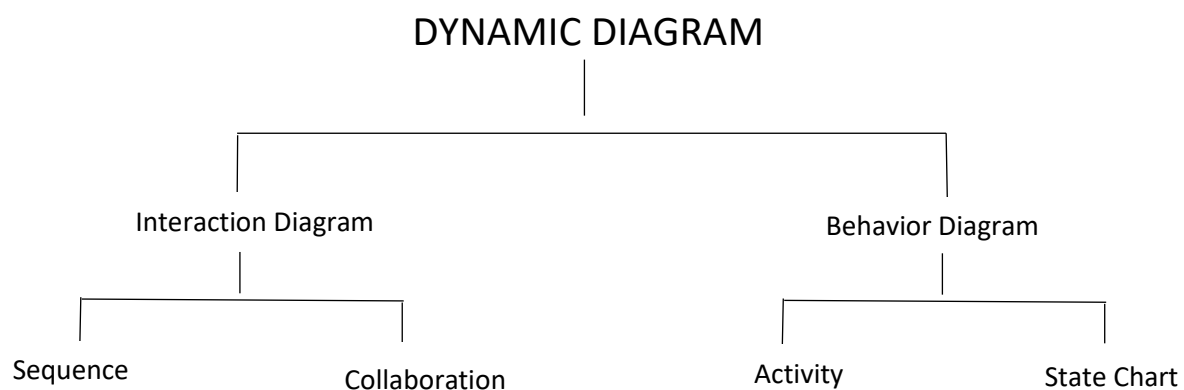
- Movie Methods:
 - addMovie(), editMovie(), delete Movie(), update Movie(), saveMovie(), searchMovie()
- Customer Methods:
 - addCustomer(), editCustomer(), deleteCustomer(), updateCustomer(), saveCustomer(), searchCustomer()
- Booking Methods:
 - addBooking(), editBooking(), deleteBooking(), updateBooking(), saveBooking(), searchBooking()
- Payment Methods:
 - addPayment(), editPayment(), deletePayment(), updatePayment(), savePayment(), searchPayment()
- Shows Methods:
 - addShows(),editShows(),deleteShows(), searchShow()

CLASS DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



DYNAMIC DIAGRAM

Dynamic UML diagrams describe the behavior of systems. A dynamic diagram describes the operations, actions, and changes that occur in a system over time. A static diagram, on the other hand, describes the characteristics of a system or part of a system.



INTERACTION DIAGRAM

Interaction Diagram is represented in UML by two diagrams known as Sequence diagram and Collaboration diagram. Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

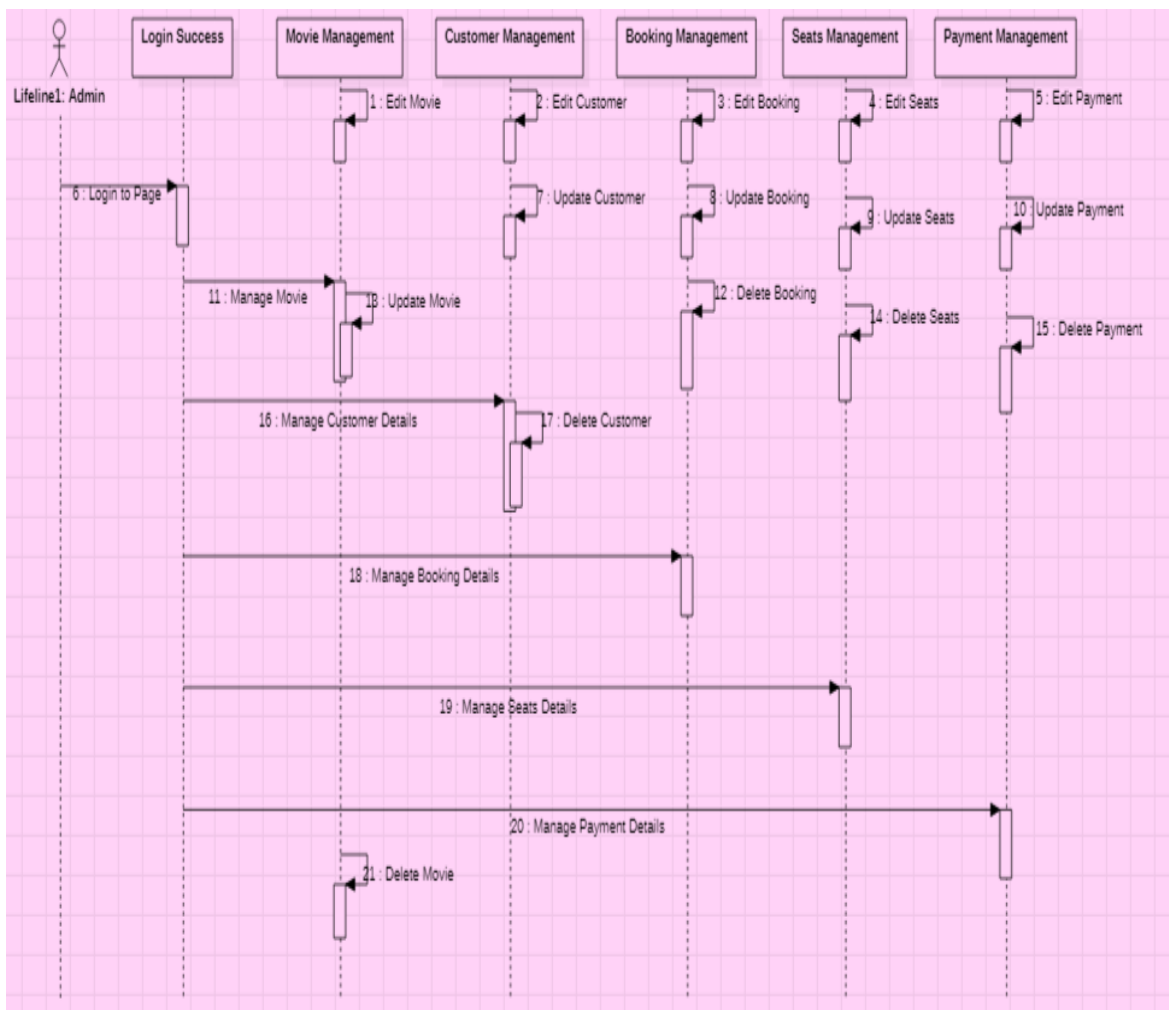
BEHAVIOUR DIAGRAM

Behavior Diagrams depict the elements of a system that are dependent on time and that convey the dynamic concepts of the system and how they relate to each other. The elements in these diagrams resemble the verbs in a natural language and the relationships that connect them typically convey the passage of time.

SEQUENCE DIAGRAM

1. This is the UML sequence diagram of Movie Ticket Booking System which shows the interaction between the objects of Shows, Booking, Movie Payment Seats. The instance of classes objects involved in this UML Sequence Diagram of Movie Ticket Booking System are as follows:
 - Shows Object
 - Booking Object
 - Movie Object
 - Payment Object
 - Seats Object
2. In the Login Sequence Diagram of Movie Ticket Booking System, admin will be able to login in their account using their credentials.
3. After login user can manage all the operations on Movie, Shows, Booking, Seats, Payment.
4. All the pages such as Booking, Seats, Payment are secure and user can access these page after login.
5. The various objects in the Seats Movie Shows, Booking, and Payment page interact over the course of the sequence, and user will not be able to access the page without verifying their identity.

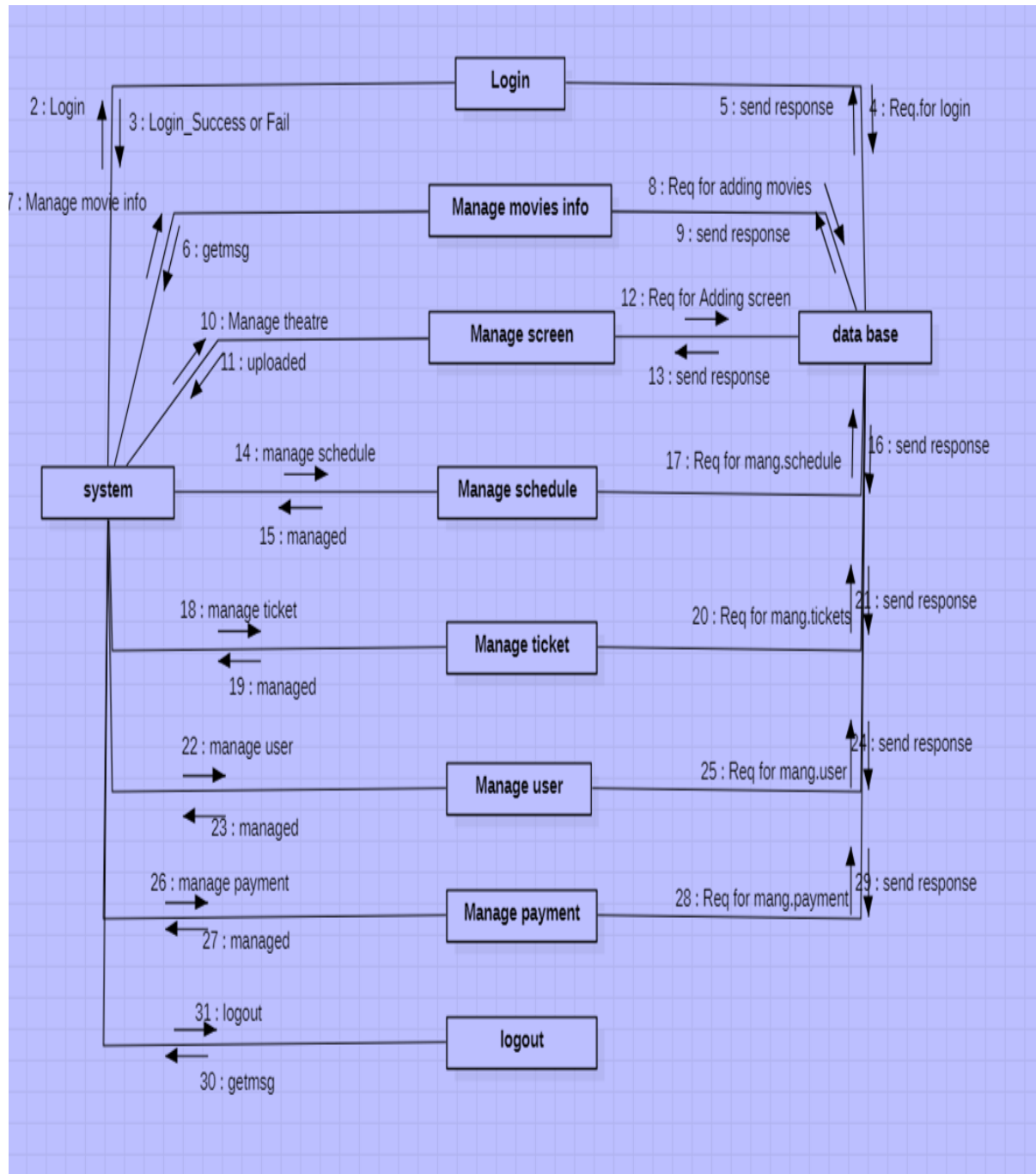
SEQUENCE DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



COLLABORATION DIAGRAM

1. The second interaction diagram is the collaboration diagram. It shows the object organization as seen in the following diagram. In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram.
2. Purpose of Communication Diagram:
 - Model message passing between objects or roles that deliver the functionalities of use cases and operations
 - Model mechanisms within the architectural design of the system
 - Capture interactions that show the past messages between objects and roles within the collaboration scenario
 - Model alternative scenarios within use cases or operations that involve the collaboration of different objects and interactions
 - Support the identification of objects (hence classes), and their attributes (parameters of message) and operations (messages) that participate in use cases.

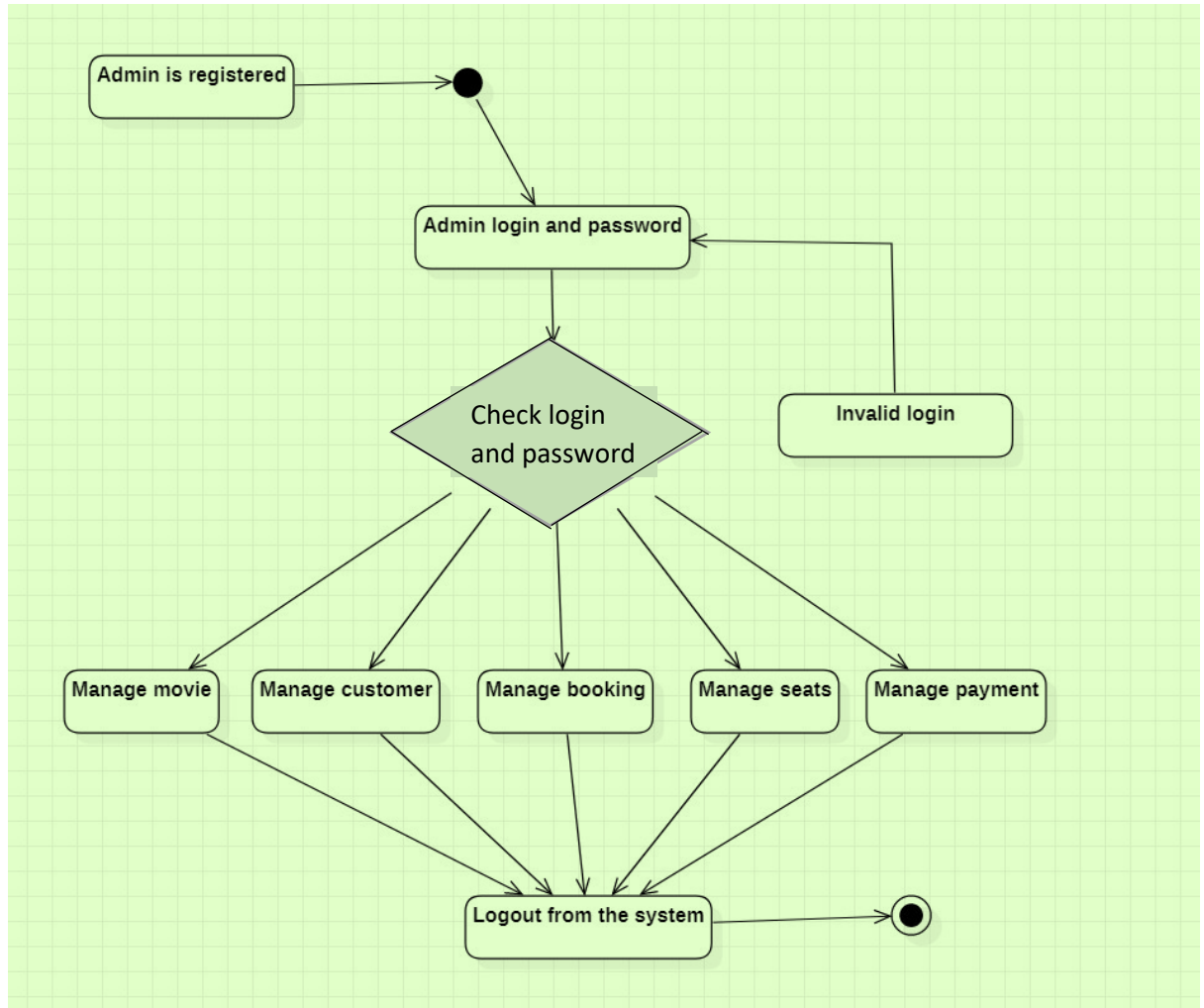
COLLABORATION DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



ACTIVITY DIAGRAM

1. Activity Diagram shows the flows of login activity, where admin will be able to login using their username and password.
2. After login user can manage all the operations on seats, Payment, Booking, Customer and Shows.
3. All the pages such as booking, customer, shows are secure and user can access these pages after login.
4. The various objects in the Customer, Seats, Payment, Booking, and Shows page interact over the course of the Activity, and user will not be able to access this page without verifying their identity.
5. The activities involved are:
 - Payment activity
 - Booking activity
 - Seats activity
 - Shows activity
 - Customer activity

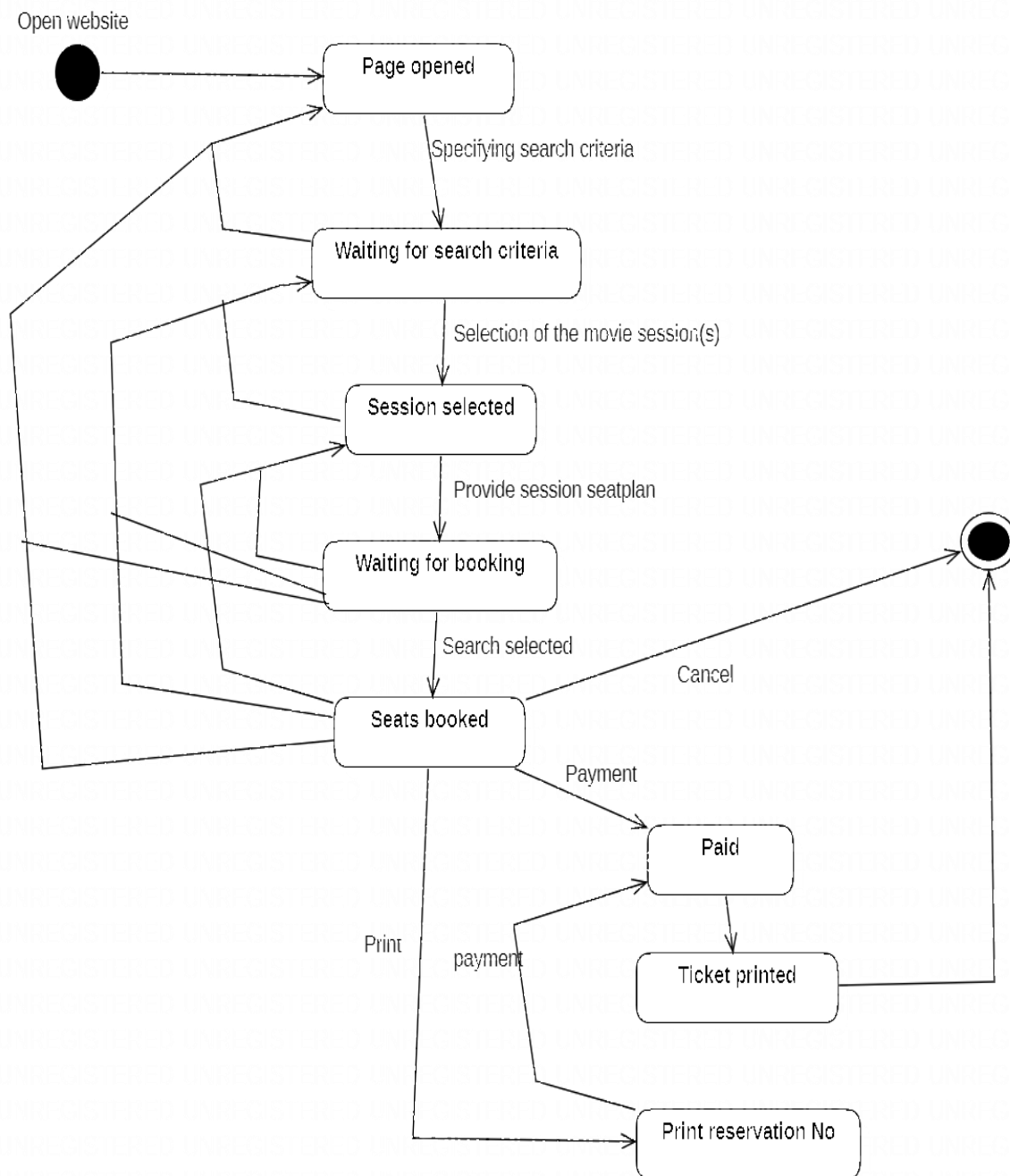
ACTIVITY DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



STATE CHART DIAGRAM

1. State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events.
2. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.
3. The diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered.
4. The most important purpose of State chart diagram is to model lifetime of an object from creation to termination.
5. They are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.
6. Purpose of State Chart Diagram:
 - To model the dynamic aspect of a system.
 - To model the life time of a reactive system.
 - To describe different states of an object during its life time.
 - Define a state machine to model the states of an object.

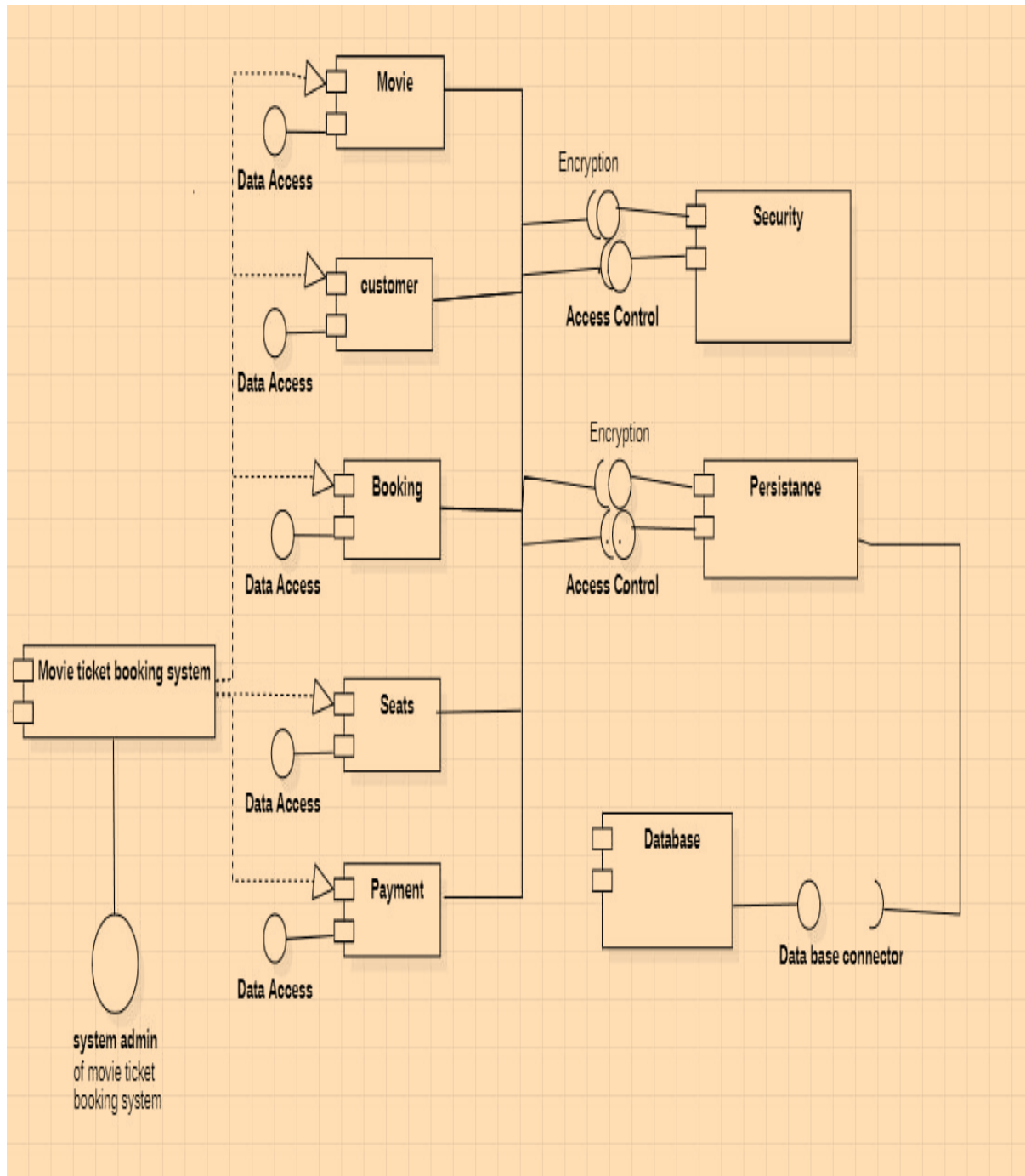
STATE CHART DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



COMPONENT DIAGRAM

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

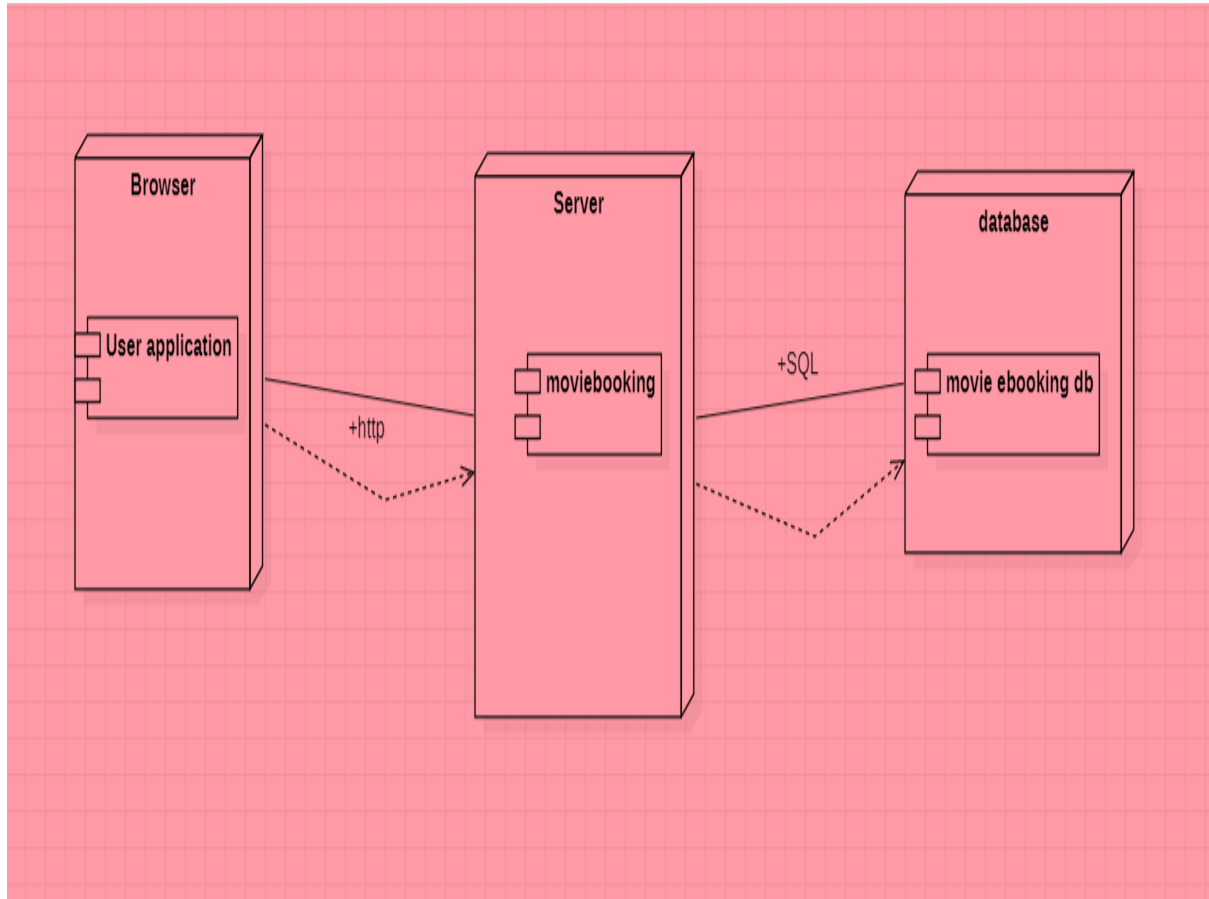
COMPONENT DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



DEPLOYMENT DIAGRAM

1. A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.
2. They are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.
3. Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.
4. They are used to model the configuration of run-time processing elements and the software components, processes, and objects that live on them. In the deployment diagram, we start by modelling the physical nodes and the communication associations that exist between them. For each node, we indicate what component instances live or run on the node.

DEPLOYMENT DIAGRAM FOR MOVIE TICKET BOOKING SYSTEM



CONCLUSION

In the modern world, the use of computers is becoming rampant. The recent developments in technology has revolutionized and consequently brought a paradigm shift in the way activities are accomplished.

UML helps to organize, plan and visualize a program. In addition, being a standard, it is widely used and accepted as the language for outlining programs. UML is used in a variety of purposes and its readability and re-usability make it an ideal choice for programmers.

This report has presented a simple, convenient, cost-effective, but efficient system with a user-friendly, sensitive and intelligible web interface. Whereby it can be accessed at any time provided there is internet. The diagram is meant to be understood by any type of programmer and helps to explain relationships in a program in a straightforward manner. Traditionally, to understand a program, a programmer would read the code directly. This could be thousands or millions of lines of code in very large programs. Having a UML diagram helps to quickly illustrate those relationships.

Software Used: Star UML