

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment:2

Student Name: Arsh Jasiwal

UID:23BCS11682

Branch: B.E. CSE

Section/Group: 603(A)

Semester:6th

Date of Performance:10/01/2026

Subject Name: System Design

Subject Code:23CSH-314

1. Aim: To design and implement an Object-Oriented Movie Ticket Booking System using appropriate class relationships (association, aggregation, composition) to manage movies, shows, seats, and bookings.

2. Objective:

- To apply Object-Oriented Programming (OOP) principles in a real-world scenario.
- To demonstrate understanding of class relationships: association, aggregation, and composition.
- To implement a functional movie ticket booking system with core features.
- To practice creating UML diagrams before coding (optional but recommended).

3. Tools: Personal computer/laptop, [Draw.io](#), Lucidchart, or PlantUML

4. Procedure:

Identify Classes and Relationships

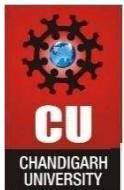
- List required classes: Movie, Theater, Show, Seat, Booking, Customer
- Define relationships between classes

Create UML Class Diagram (optional but helpful)

- Visualize associations, aggregations, and compositions

Implement Classes in Code

- Create classes with appropriate attributes and methods
- Implement relationships using object references



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Test the System

- Create sample data (movies, shows, seats)
- Simulate booking process
- Handle edge cases (seat already booked, invalid show)

Verify Relationships

- Ensure proper object lifecycle (composition vs aggregation)
- Test dependency between objects

5. Implementation:

Design the Class Structure

1. Movie
 - movieId, title, genre, duration, language
2. Theater
 - theaterId, name, location
 - List<Show> shows (Aggregation)
 - List<Seat> seats (Composition)
3. Show
 - showId, movie, startTime, endTime
 - Theater theater (Association)
 - List<Seat> availableSeats
4. Seat
 - seatId, row, seatNumber, seatType (Normal/Premium)
 - boolean isBooked
5. Booking
 - bookingId, customerName, bookingTime
 - Show show (Association)
 - List<Seat> bookedSeats (Aggregation)
 - double totalPrice
6. Customer (optional)
 - customerId, name, email, phone

6. Output:

Figure 1: State Diagram
Figure 2: Sequence Diagram

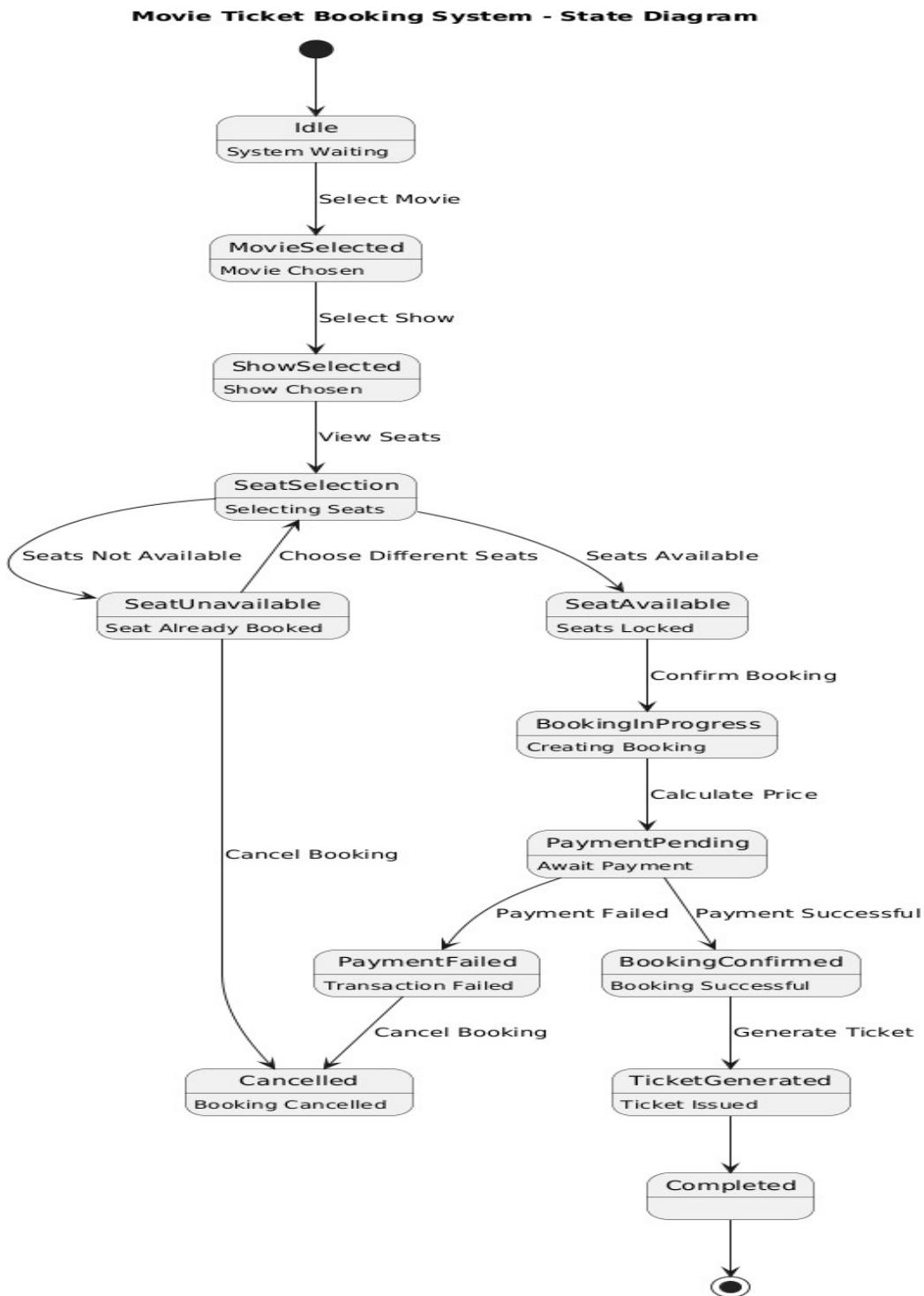
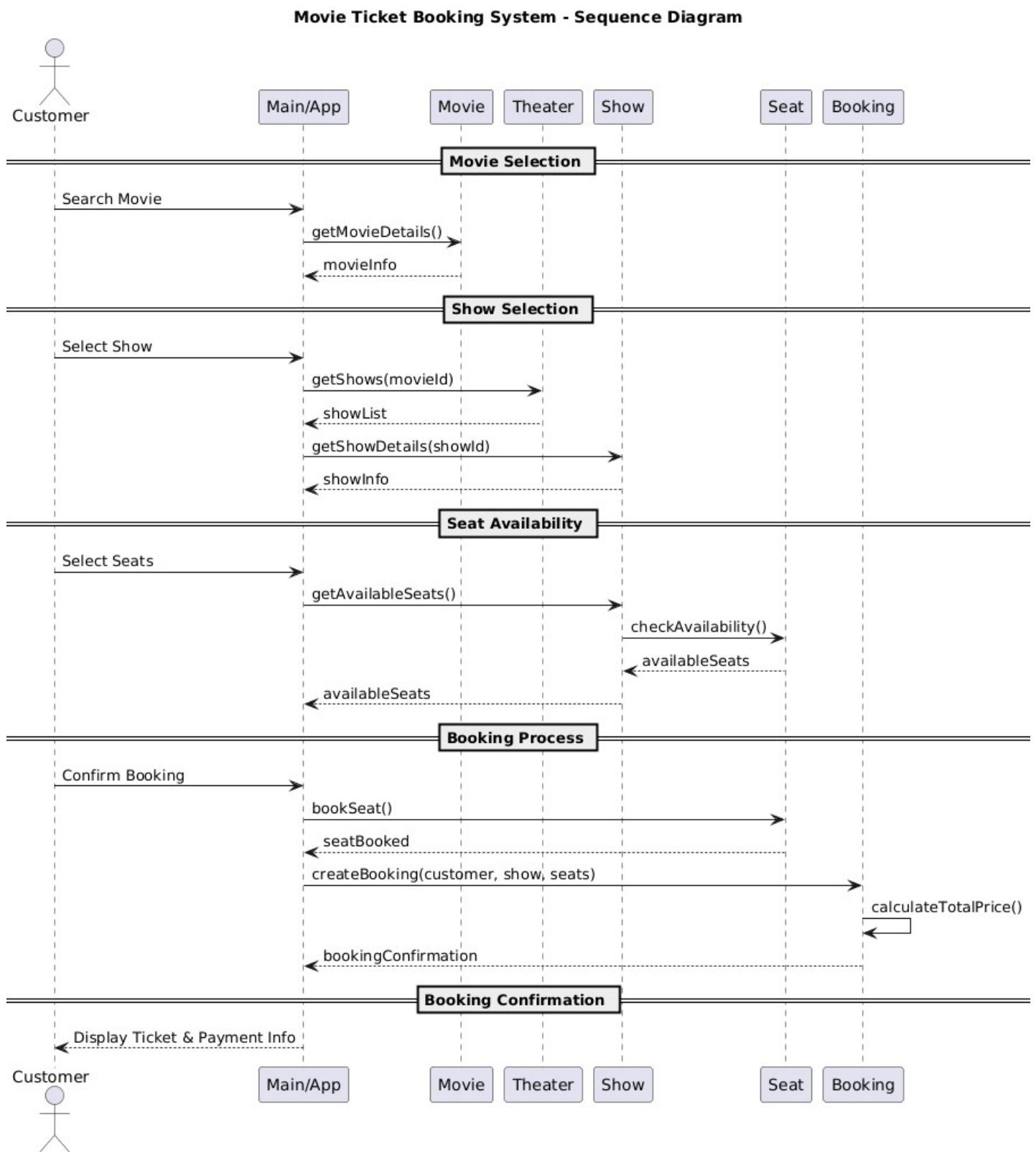
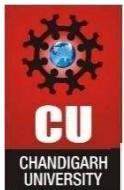


Figure 2





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

7. Learning Outcome:

- Learned how to design State Chart Diagram and Sequence Chart Diagram.
- Understood the concept of Unified Modeling language.