Cinema booking system

Contents:

Software Design Document

1.introduction

The primary goal of this project is to provide a comprehensive solution for managing cinema bookings. This includes:

- Empowering users to browse movies, view showtimes, select seats, and make bookings with ease.
- Empowering administrators with powerful tools to efficiently manage movie listings, theater capacities, bookings, and revenue tracking.

2. Functional Requirements

1. **Register**: user shall register to use the website. The users should have a username and password.
2. **log in**: user, journalist and admin shall log in the system by filling username and password boxes.
3. **Book movie**: user can book movie and select date and time.
4. **log out**: user may need to log out the system.
5. **Add to favorite**: user shall add movies to favorite list.
6. **Remove from favorite**: user shall remove movies from favorite list.
7. **Select ticket quantity:** user can select any quantities of tickets.
8. **Choose booking details and ticket type**

3. Non- functional Requirements

1. Performance Requirements

-the page works immediately when the user open it.

-refresh process takes less than max 5 sec.

2. Usability & Humanity Requirements

-the system is easy to use and understand without training

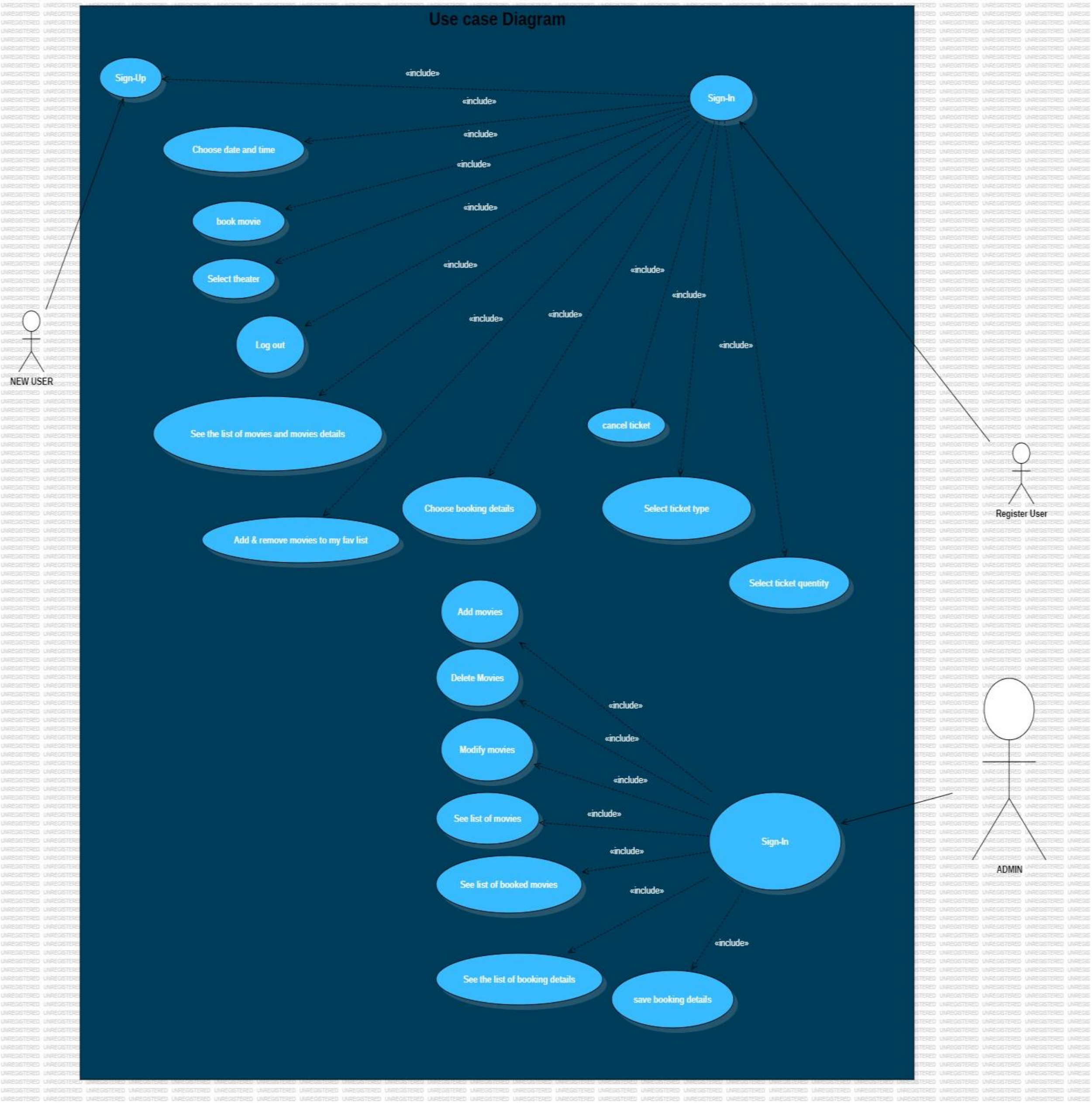
-the system is easy with payments.

3. Security Requirements

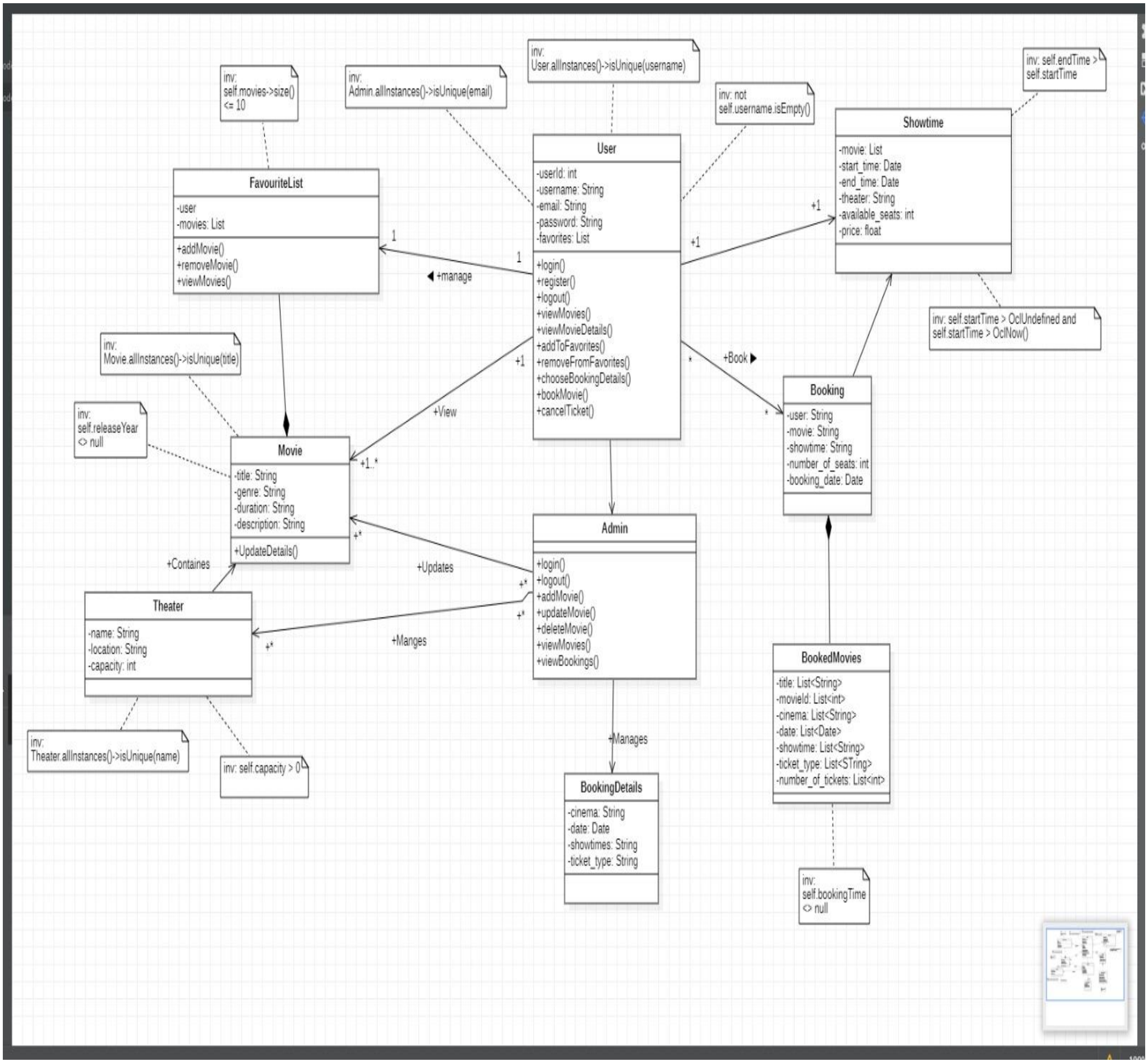
-the system saves the information of every user

-the system protects your payment information from theft

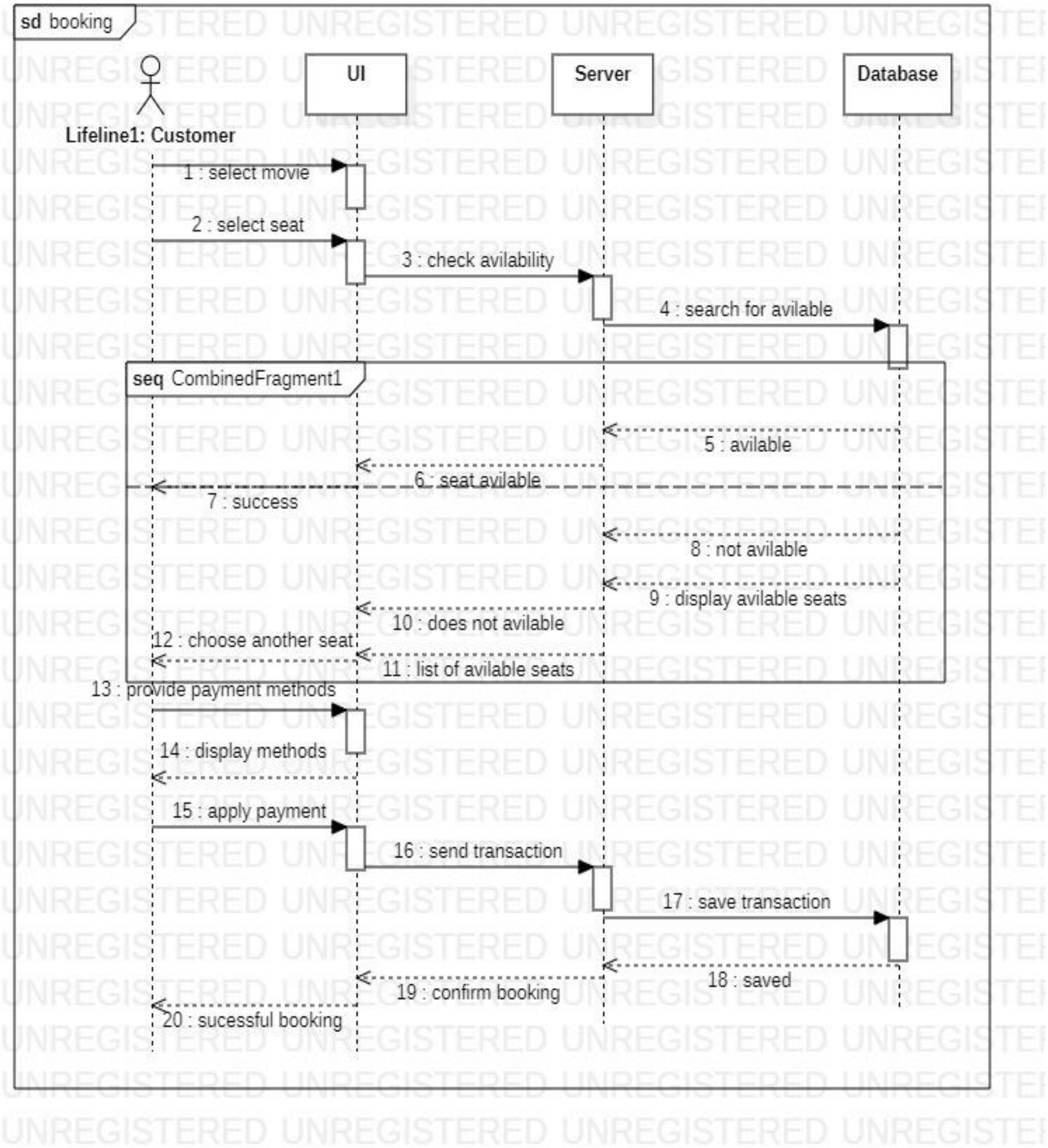
4. Use-Case Diagram



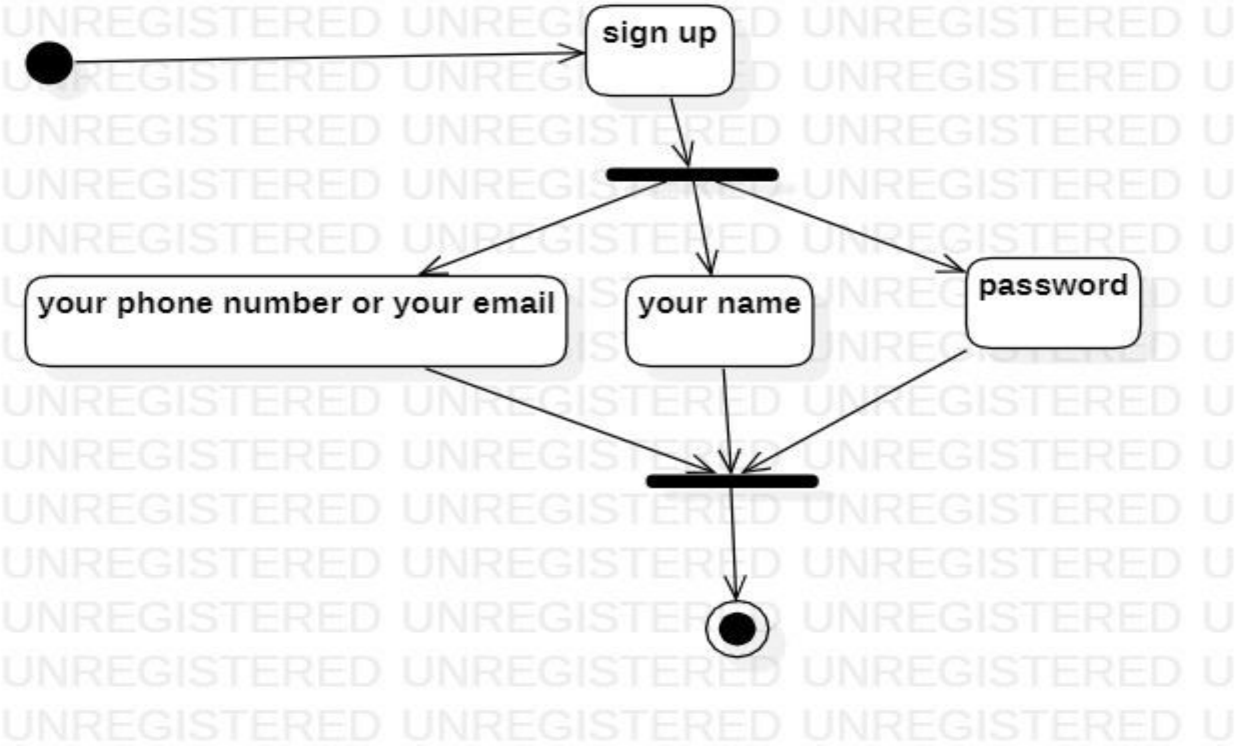
5. Class diagram & OCL

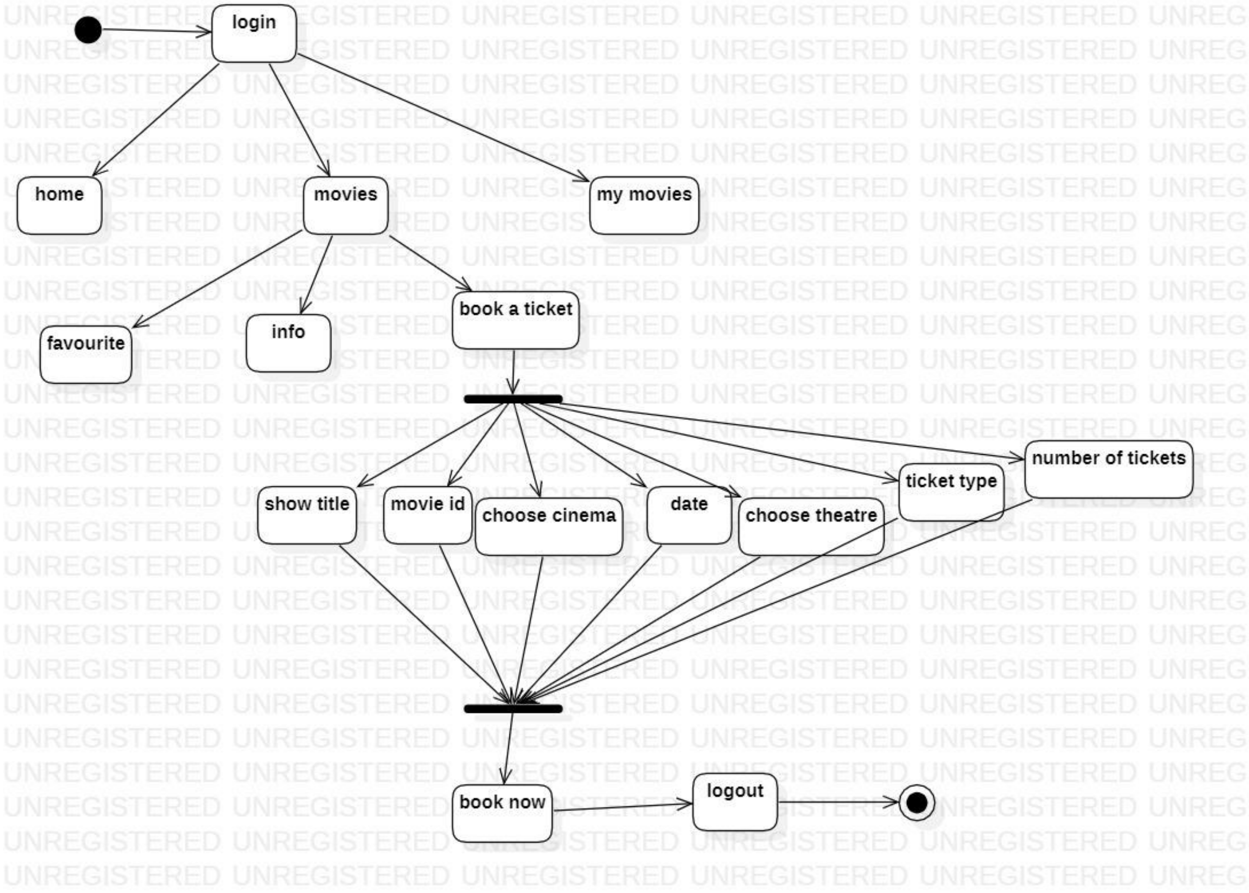


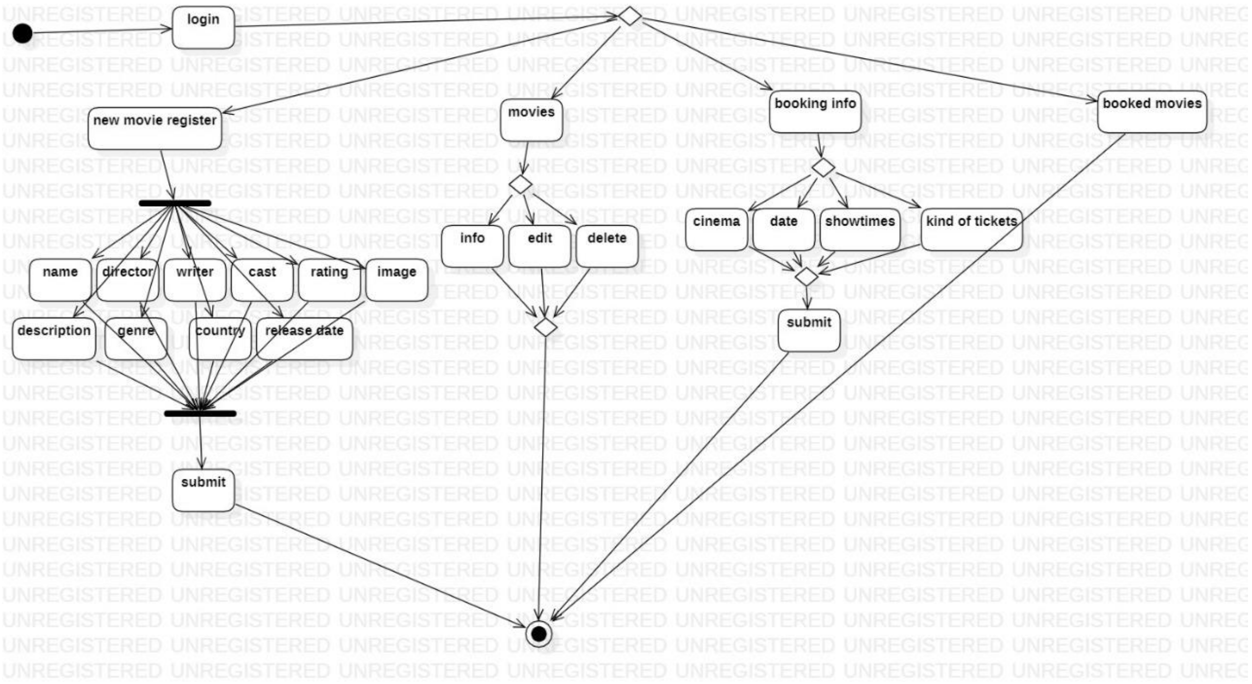
6. Sequence Diagram



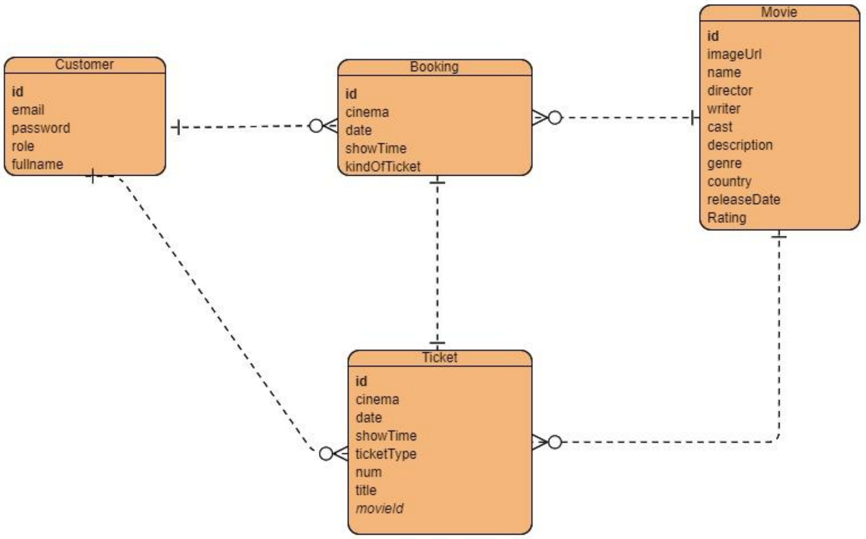
7.Activity Diagram







8.ERD Diagram



9.Database Schema

The database schema consists of tables to store information about users, movies, favorites, booking details, and booked movies.

5.1 User Table

- **Table Name: user**
- **Columns:**
  - **id (Primary Key, Auto-generated): Unique identifier for each user.**
  - **email: Email address of the user.**
  - **password: Password of the user.**
  - **role: Role of the user (e.g., admin, user).**
  - **fullname: Full name of the user.**

5.2 MyMovies Table

- **Table Name: mymovies**
- **Columns:**
  - **id (Primary Key): Unique identifier for each movie.**
  - **imageUrl: URL of the movie image.**
  - **name: Name of the movie.**
  - **director: Director of the movie.**
  - **writer: Writer of the movie.**

5.3 Movies Table

- **Table Name: movies**
- **Columns:**
  - **id (Primary Key, Auto-generated): Unique identifier for each movie.**

- **imageUrl: URL of the movie image.**
- **name: Name of the movie.**
- **director: Director of the movie.**
- **writer: Writer of the movie.**
- **cast: List of cast members (element collection).**
- **description: Description of the movie.**
- **genre: Genre of the movie.**
- **country: Country of origin of the movie.**
- **releaseDate: Release date of the movie.**
- **rating: Rating of the movie.**

<u>5.4 BookingDetails Table</u>

- **Table Name: bookingdetails**
- **Columns:**
  - **id (Primary Key, Auto-generated): Unique identifier for each booking detail.**
  - **capacity: Capacity of the cinema.**
  - **location: Location of the cinema.**
  - **cinema: Name of the cinema.**
  - **date: Date of the booking.**
  - **showTime: Time of the movie show.**
  - **kindOfTicket: Type of ticket booked.**

<u>5.5 BookedMovies Table</u>

- **Table Name: bookedmovies**
- **Columns:**

- **id (Primary Key, Auto-generated): Unique identifier for each booked movie.**
- **cinema: Name of the cinema where the movie is booked.**
- **date: Date of the booking.**
- **showTime: Time of the movie show.**
- **ticketType: Type of ticket booked.**
- **num: Number of tickets booked.**
- **title: Title of the booked movie.**
- **movieID: ID of the booked movie.**

10.Technologies Used

- **Backend Framework: Springboot**
- **Database: MySQL**