<Theater Ticketing System>

Software Design Specification

<2/27/2025>

<Group 7>
<Anthony Nguyen, Steven To, Charlie Pham>

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Fall 2023

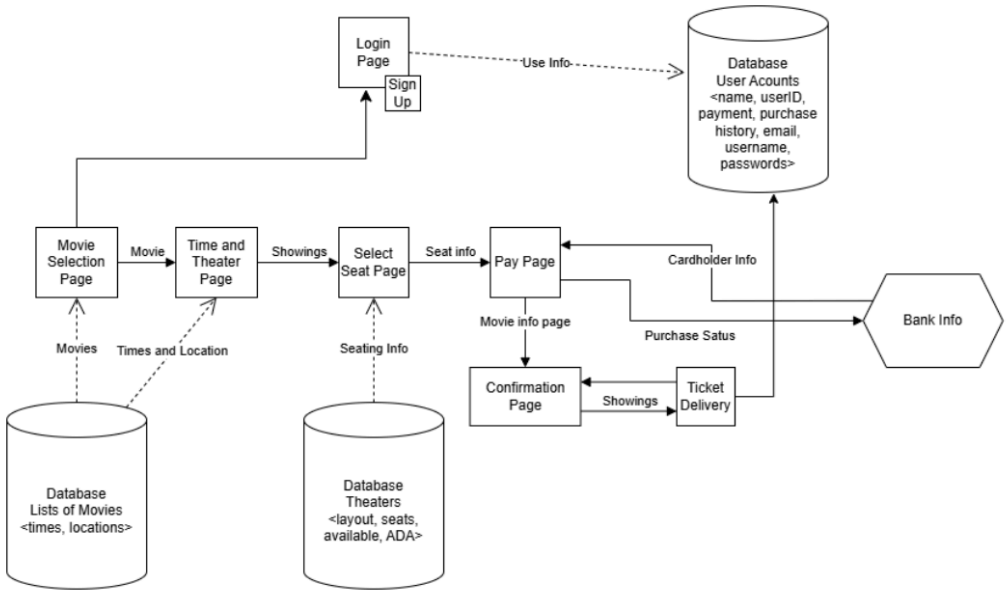
System Description

1. System Description

The theater ticketing system is a web-based application that allows users to browse moving screenings, select seats, and purchase digital tickets. The system offers real-time seat availability, secure payment integration, and automated notifications. It also includes administrative functionalities for managing movie schedules and ticket pricing. The system aims to enhance convenience for customers and optimize theater operations.

2. Software Architecture Overview

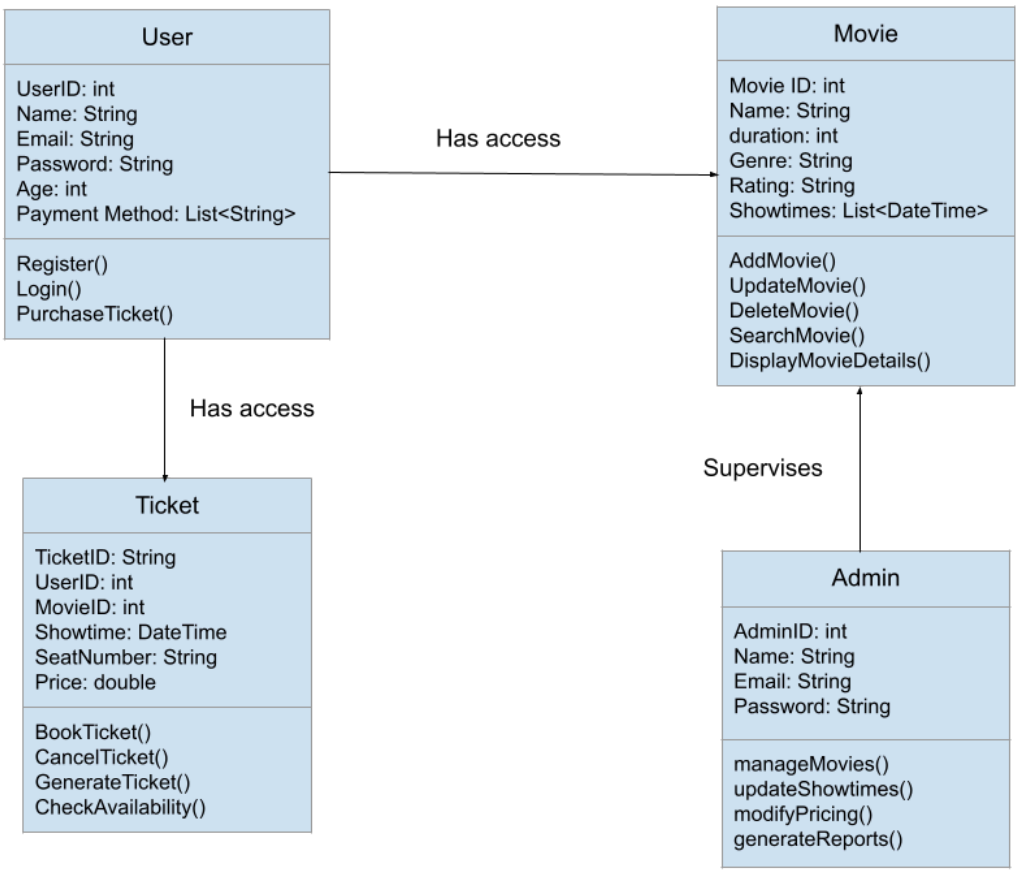
2.1 SWA Diagram



2.2 SWA Description

The user starts at the login page where they access their account, either regular customer or admin. This data from the database takes them to the movie selection page with certain attributes that correlate to the user type. Then at the movie selection page, the user can search for a movie, or look through the genre list. After that, they are taken to the time and theater arrangement page. After selecting the time you want to watch the movie, you are taken to the seating arrangement page. Then after selecting your seat, you are taken to the payment page. Once a payment type is selected and paid for, the system accesses the bank account and takes you to a confirmation page. From here you would confirm your purchase, and the receipt of the payment gets sent to the user account database. Your ticket gets delivered via email and also saved in the account database. With the email, you can then go in and enjoy your movie.

2.3 UML Class Diagram



2.4 Description of Classes

User

- Purpose: Represents an individual user who can register, log in, and purchase movie tickets.

Movie

- Purpose: Stores details of movies available in the system.

Ticket

- Purpose: Handles movie ticket information and seat reservations.

Admin

- Purpose: Manages the backend operations, including movie listings and reports.

2.5 Description of Attributes

- **User**

    - UserID: int
        - A unique identifier for each user.
    - Name: String
        - The full name of the user.
    - Email: String
        - The user's email address, and is used for account creation and communication.
    - Password: String
        - A secure password for account authentication.
    - Age: int
        - The user's age, and is used to enforce the minimum age requirement (14+).
    - PaymentMethods: (List<string>)
        - A list of payment methods associated with the user.
- **Movie**
    - MovieID: int
        - Unique identifier for the movie.
    - Title: String
        - Title of the movie.
    - Description: String
        - Brief synopsis of the movie.
    - Duration: int
        - Runtime of the movie in minutes.
    - Genre: String
        - Category of the movie (e.g., "Action", "Comedy").
    - Rating: String
        - Movie rating (e.g., "PG", "PG-13").
    - Showtimes: List<DateTime>
        - List of available showtimes.
- **Ticket**
    - TicketID: String
        - A unique identifier for each ticket.
    - UserID: int
        - The ID of the user who purchased the ticket.
    - MovieID: int
        - The ID of the movie associated with the ticket.
    - Showtime: DateTime
        - The date and time of the movie screening.
    - SeatNumber: String
        - The seat number assigned to the ticket.
    - Price: double
        - The price of the ticket is converted to the user's local currency.
- **Admin**
    - AdminID: int
        - Unique identifier for administrators.
    - Name: String

        - Name of the administrator.
    - Email: String
        - Email for authentication.
    - Password: int
        - A secure password for account authentication.

2.6 Description of Operations

- **User**
    - Register(name, email, password, age): void
        - Allows a new user to create an account.
    - Login(email, password): bool
        - Authenticates the user and grants access to the system.
    - PurchaseTicket(movieID, showtime, seatNumber): Ticket
        - Purchases a movie ticket.
- **Movie**
    - AddMovie(title, description, duration, genre, rating): void
        - Adds a new movie to the system.
    - UpdateMovie(movieID, title, description, duration, genre, rating): void
        - Updates movie details.
    - DeleteMovie(movieID): void
        - Removes a movie from the system.
    - SearchMovie(keyword): List<Movies>
        - Searches for movies by title, genre, or keyword.
    - DisplayMovieDetails(movieID): void
        - Movie Displays detailed information about a movie.
- **Ticket**
    - BookTicket(userID, movieID, showtime, seatNumber, price): Ticket
        - A user can book a ticket for a specific movie and showtime.
    - CancelTicket(ticketID): void
        - This enables a user to cancel a booked ticket.
    - GenerateTicket(ticketID): string
        - Generates a digital ticket for the user.
    - CheckAvailability(movieID, showtime): int
        - Check seat availability for a specific showtime.
- **Admin**
    - manageMovies(): void
        - Adds, edits, and removes movie listings.
    - updateShowtimes(): void
        - Modifies showtimes for movies.
    - modifyPricing(): void
        - Adjusts ticket pricing.
    - generateReports(): List<Report>
        - Produces sales and user activity reports.

3. Development Plan and Timeline

3.1 Partitioning of Tasks

Task	Estimated Completion Date
System Architecture and Design	Week 1-2
Testing and Debugging	Week 2-3
Payment Integration	Week 3-4
Frontend Development and UI Design	Week 4-5
Database and Backend Development	Week 5-6
Documentation and Finalization	Week 6-7
User Authentication and Role Management	Week 7-8
Seat Selection and Booking System Development	Week 8-9
Notification and Confirmation System	Week 9-10

3.2 Team Member Responsibilities

	Member
Title Page	Steven To
System Overview	Charlie Pham
Architectural Diagram	Anthony Nguyen
UML Diagram	Steven To
Classes Description	Charlie Pham
Attributes Description	Anthony Nguyen
Operations Description	Steven To