Software Design Specification

Movie Theatre Ticketing System (MTTS)

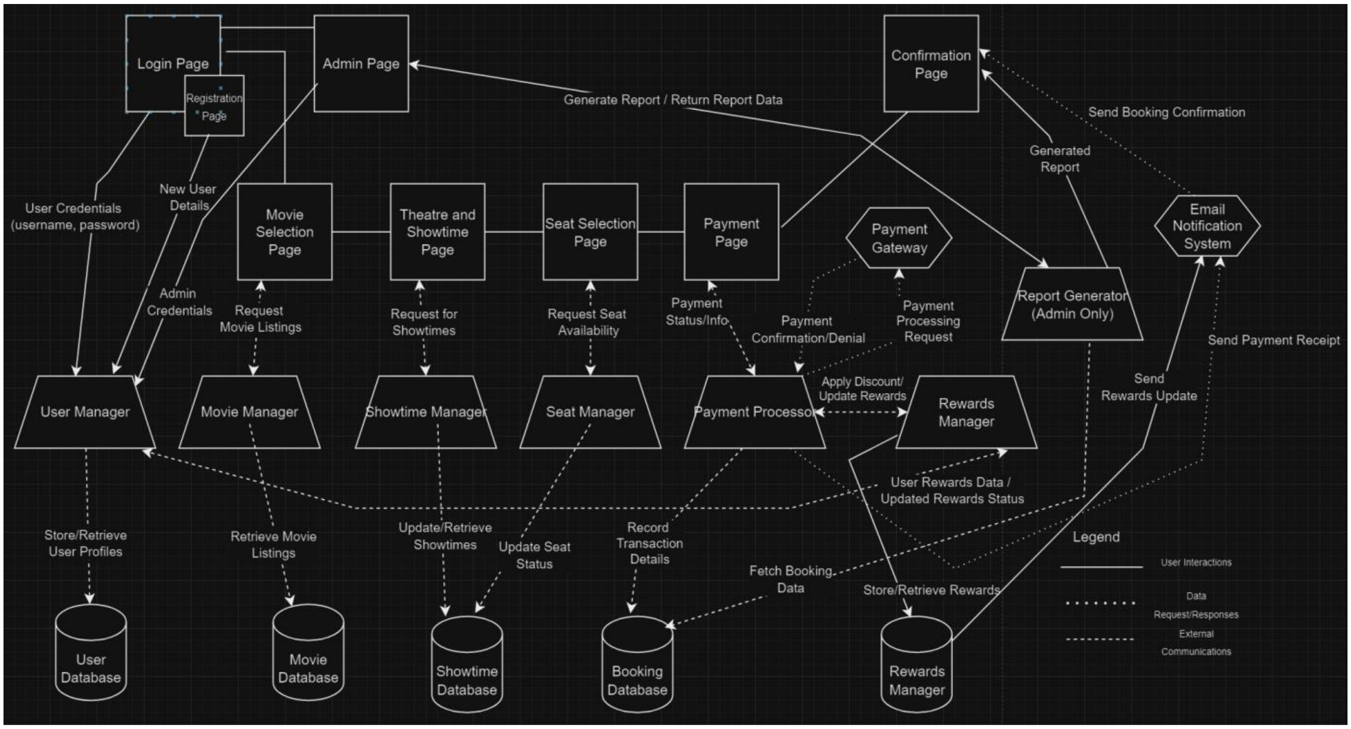
Team Members:
Julio Nevarez
Matthew Kloth
Doan Quoc Tien Nguyen
Jingyi Chen

System Description

The Movie Theatre Ticketing System (MTTS) is designed to optimize the management and sale of movie tickets, offer an intuitive interface for customers to acquire tickets, and incorporate administrative features for theatre personnel to oversee showtimes, access transaction history, and produce sales reports. The system seeks to provide a seamless experience for customers and theatre personnel, incorporating features like real-time seat availability, adaptable payment methods, and comprehensive show details.

Software Architecture Overview

SWA Diagram



Software Architecture Diagram Description

The software architecture for the **Movie Theatre Ticketing System (MTTS)** is structured into distinct layers that manage different functionalities, interactions, and data flow. The architecture can be divided into the following main components:

1. User Interface Layer

The User Interface Layer consists of various pages that provide an interactive experience for users and administrators. These pages capture user input, display system information, and initiate requests to the backend for data and operations. Pages include:

- **Login Page**: Handles user authentication, allowing existing users to log in and new users to navigate to the Registration Page.

- **Registration Page**: Manages new user registration by capturing details such as username, password, and contact information.
- **Movie Selection Page**: Displays available movies for users to choose from and requests movie listings from the backend.
- **Theatre and Showtime Page**: Shows detailed information about showtimes and theatre locations for selected movies.
- **Seat Selection Page**: Allows users to select available seats for a chosen showtime and sends seat selection data to the backend.
- **Payment Page**: Collects payment details, applies rewards or discounts, and initiates payment processing.
- **Confirmation Page**: Displays a summary of the completed booking and sends confirmation details to the user.

The Admin Page serves as a specialized interface for administrative users, providing access to generate reports, update showtimes, and manage the theatre's backend data.

2. Backend Application Layer

The Backend Application Layer is responsible for processing the logic, managing data transactions, and coordinating operations across different modules. This layer includes various **managers** that handle specific parts of the system:

- **User Manager**: Manages user authentication, registration, profile updates, and rewards status. It interacts with the User Database and the Rewards Manager to update user rewards and apply discounts.
- **Movie Manager**: Handles requests for movie listings and communicates with the Movie Database to fetch or update movie details.
- **Showtime Manager**: Manages showtimes for movies and updates theatre schedules. Interacts with the Showtime Database.
- **Seat Manager**: Manages seat availability for different showtimes, updating status when seats are reserved or released. Communicates with the Showtime Database.
- **Payment Processor**: Facilitates payment processing through an external Payment Gateway and updates the Booking Database with transaction details. It also works with the Rewards Manager to apply any rewards-based discounts or point updates during a transaction.

- **Rewards Manager**: Manages rewards information, including point accumulation, discount application, and user-specific reward updates. It communicates with the User Manager to display reward status and with the Payment Processor to apply discounts and update points after successful transactions.
- **Report Generator (Admin Only)**: Allows administrators to generate booking, sales, and performance reports by fetching data from the Booking Database.

The **Email Notification System** is integrated into this layer as a notification handler, responsible for sending automated emails triggered by various backend managers, such as the Payment Processor and Rewards Manager.

3. Database Layer

The Database Layer stores all persistent data and ensures data integrity across the system. It includes:

- **User Database**: Contains user profiles, credentials, purchase history, and rewards status.
- **Movie Database**: Stores movie details, including titles, genres, and ratings.
- **Showtime Database**: Manages data related to showtimes, theatre locations, and seat availability.
- **Booking Database**: Keeps track of confirmed bookings, seat reservations, and transaction details.
- **Rewards Database**: Maintains information on user rewards, accumulated points, and available discounts.

Each manager in the backend layer communicates with the corresponding database using data requests and responses to perform operations such as Create, Read, Update, Delete.

The Rewards Database is accessed by the Rewards Manager to store and retrieve user-specific rewards data, such as points earned and discounts applied.

4. External Systems Layer

The External Systems Layer integrates the MTTS with third-party services to enable additional functionality:

- **Payment Gateway**: An external service that validates and processes user payments. The Payment Processor communicates with the gateway to send payment requests and receive confirmations.

- **Email Notification System**: An external component responsible for sending transactional emails, such as booking confirmations, payment receipts, and rewards updates. The Payment Processor, Rewards Manager, and Confirmation Page send triggers to the Email Notification System to notify users of completed transactions, rewards status changes, and booking confirmations.

Each external system is represented as a separate component and uses secure communication channels to ensure data integrity and privacy.

5. Data Flow and Communication

The data flow in the system is facilitated using a combination of user interactions, data requests/responses, and external communications:

1. **User Interactions (Solid Lines)**: These represent direct interactions between the user and the UI pages as they navigate through the system.
2. **Data Requests/Responses (Dashed Lines)**: Backend managers communicate with databases and the UI components using these lines to handle data operations, including rewards management and seat availability checks.
3. **External Communications (Dotted Lines)**: Represent interactions with external systems, such as the Payment Gateway and Email Notification System, for secure payment processing and email notifications.

6. Architectural Considerations

- **Modularity**: Each component operates independently, allowing for easy updates or replacements.
- **Scalability**: The architecture can accommodate additional features or modifications without disrupting existing functionalities.
- **Security**: Sensitive data like user credentials and payment information is handled through secure communications with the User Manager and Payment Gateway.
- **Maintainability**: The clear separation of responsibilities between managers and UI components simplifies debugging and future extensions.

This architecture provides a robust foundation for the Movie Theatre Ticketing System, ensuring a seamless experience for both users and administrators while supporting advanced features like rewards and email notifications.

2. UML Class Diagram: Provide detailed class structures, attributes, and operations.

3. Class Descriptions:

- Class Name: Description of its purpose and role in the system.

- Attributes: List attributes and their data types.

- Operations: Specify all major functions, interfaces, and parameters in detail.

Development Plan and Timeline

1. Partitioning of Tasks: List of tasks and their estimated timeline.

2. Team Member Responsibilities: Which team member is responsible for each task.