ระบบการจองตั๋วและจัดการโรงหนัง

Booking and Management System

โดย

นายกฤชพล สุวรรณวุฒิ	รหัส	1650705583
นายฐิรวัฒน์ แซ่ลิ้ม	รหัส	1600700535
นายกิตติกานต์ โคณบาล	รหัส	1650704271

หลักสูตรวิทยาศาสตรบัณฑิต

ภาคการศึกษาที่ 1 ปีการศึกษา 2566

ภาควิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศและนวัตกรรม

มหาวิทยาลัยกรุงเทพ

Purpose:

The Cinema Booking and Movie Management System is designed to streamline cinema operations, improve the booking process for customers, and provide admins with tools to manage movies, customer interactions, and bookings efficiently. The system offers a seamless way for customers to browse and book movies, while allowing administrators to manage the cinema's offerings and user data.

Key objectives include:

- **Automating seat bookings and movie reservations.**
- **Simplifying movie management for administrators.**
- **Enhancing customer experience by providing easy access to available movies and bookings.**

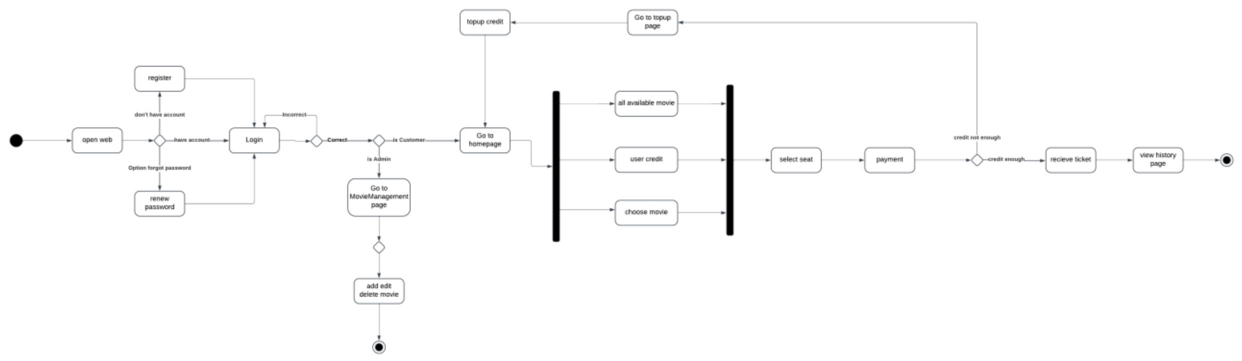
Users:

1. **Customer:**
    - **Role: Customers can browse movies and book seats**
    - **Responsibilities:**
        - **Browse available movies, view details such as title, genre, duration, and trailer.**
        - **Book seats for movies based on available showtimes.**
        - **View current and past bookings.**
2. **Administrator (Admin):**
    - **Role: Administrators manage movie listings.**
    - **Responsibilities:**
        - **Add, update, and remove movie listings, including titles, genres, trailers, and ratings.**

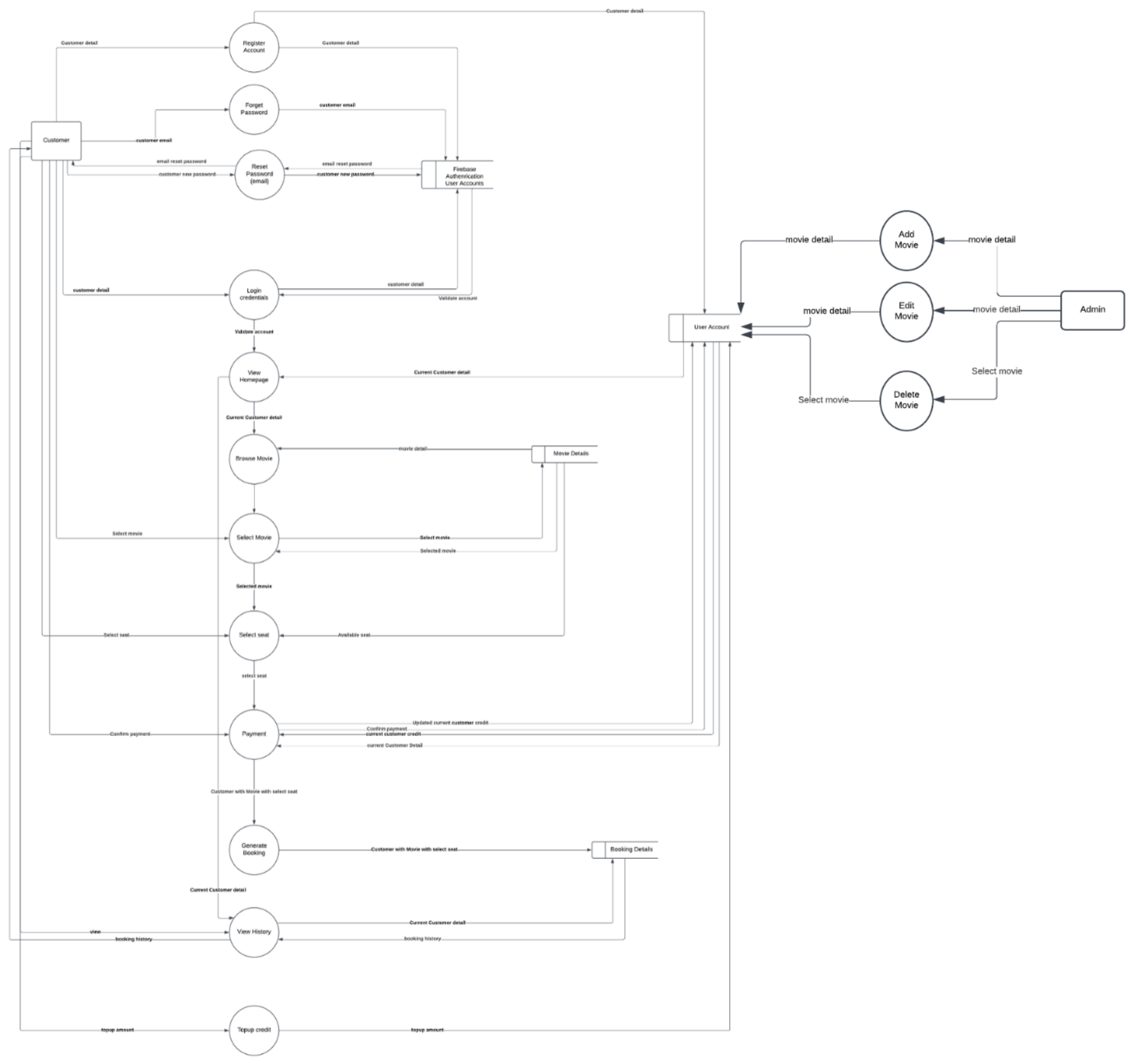
Features:

1. **For Customers:**
    - **Movie Browsing: Customers can view movie listings with comprehensive details (e.g., title, genre, trailer, rating).**
    - **Booking Seats: Customers can reserve seats for their preferred movie by selecting available showtimes.**
    - **Booking History: Customers can view a list of their current and past bookings, including details like movie title, showtime, and booked seats.**
2. **For Administrators:**
    - **Movie Management: Admins can add, update, and delete movie listings, including details like genre, trailer, and showtimes.**

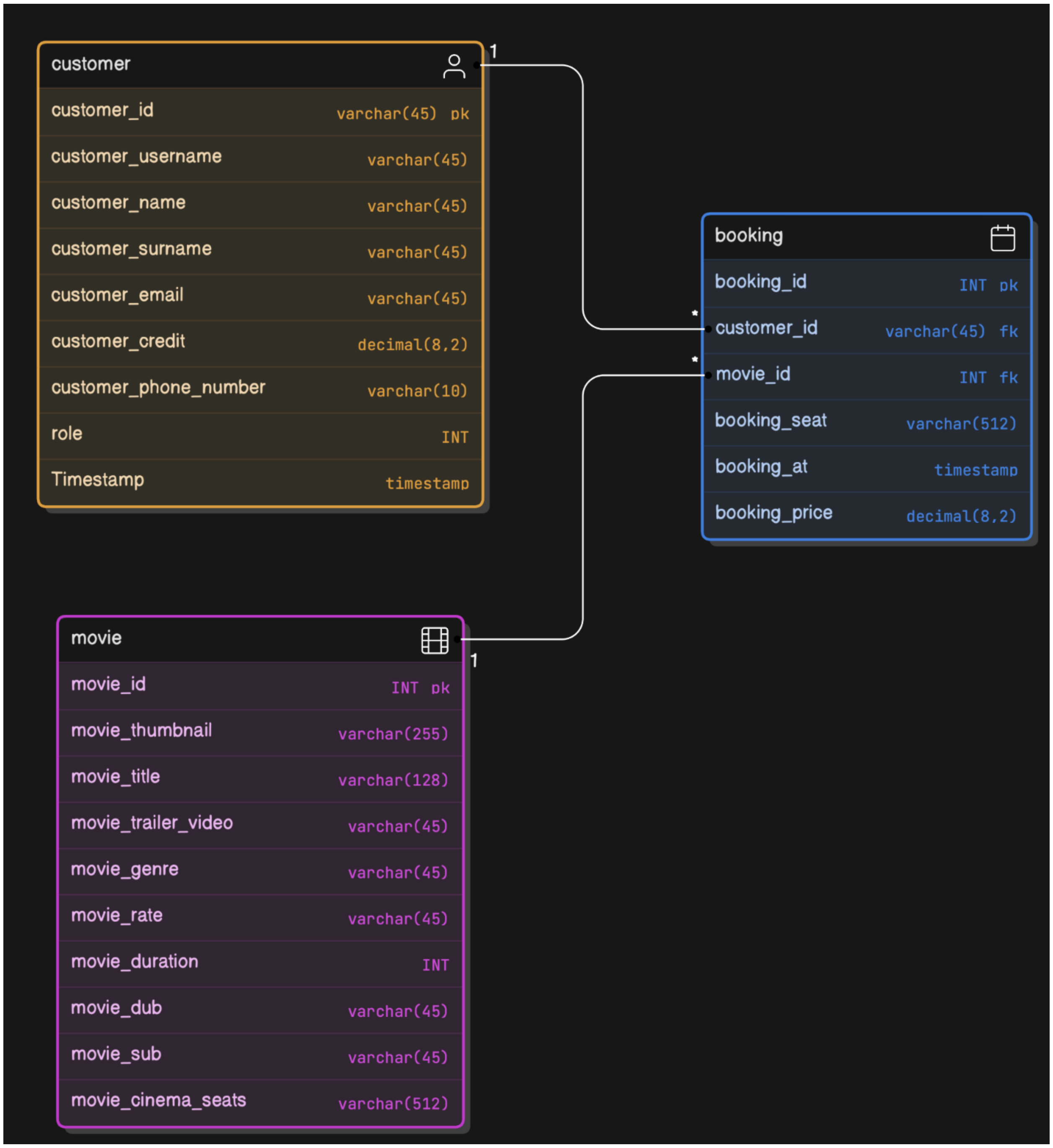
UML diagram for process



Dataflow diagram



ER Diagram


customer
customer_id varchar(45) pk
customer_username varchar(45)
customer_name varchar(45)
customer_surname varchar(45)
customer_email varchar(45)
customer_credit decimal(8,2)
customer_phone_number varchar(10)
role INT
Timestamp timestamp
1
booking
booking_id INT pk
*
customer_id varchar(45) fk
*
movie_id INT fk
booking_seat varchar(512)
booking_at timestamp
booking_price decimal(8,2)
movie
1
movie_id INT pk
movie_thumbnail varchar(255)
movie_title varchar(128)
movie_trailer_video varchar(45)
movie_genre varchar(45)
movie_rate varchar(45)
movie_duration INT
movie_dub varchar(45)
movie_sub varchar(45)
movie_cinema_seats varchar(512)

API Documentation

Admin

1. Fetch All Admins

- **Endpoint: /get**
- **Method: GET**
- **Description: Retrieves all admin users from the database.**
- **Response:**
  - **200 OK: Returns a list of all admin records.**
    - **Example: A list of admin objects.**
  - **500 Internal Server Error: Error fetching admin data.**

Booking

1. Fetch All Bookings

- **Endpoint: /get**
- **Method: GET**
- **Description: Retrieves all bookings from the database.**
- **Response:**
  - **200 OK: Returns a list of all booking records.**
    - **Example: A list of booking objects.**
  - **500 Internal Server Error: Error fetching booking data.**

---

2. Add a New Booking

- **Endpoint: /add**
- **Method: POST**
- **Description: Adds a new booking record for a customer and movie.**
- **Parameters:**
  - **Body:**
    - **customer_id: ID of the customer.**
    - **movie_id: ID of the movie.**
    - **booking_seat: The seat(s) booked (can be an array or string depending on implementation).**
- **Response:**
  - **201 Created: Booking successfully added to the database.**
  - **400 Bad Request: If required fields are missing or invalid.**
  - **500 Internal Server Error: Error adding the booking.**

Customer

1. Fetch All Customers

- **Endpoint: /get**
- **Method: GET**
- **Description: Retrieves all customers from the database.**
- **Response:**
    - **200 OK: Returns a list of all customers.**
        - **Example: A list of customer objects.**
    - **500 Internal Server Error: Error fetching customer data.**

---

2. Fetch Customer by ID

- **Endpoint: /getEachUser/:customer_id**
- **Method: GET**
- **Description: Fetch details of a specific customer by their ID.**
- **Parameters:**
    - **Path Parameter:**
        - **customer_id: ID of the customer.**
- **Response:**
    - **200 OK: Returns the customer details.**
        - **Example: A customer object with the details of the customer.**
    - **404 Not Found: If the customer is not found.**
    - **500 Internal Server Error: Error fetching customer data.**

---

3. Add a New Customer

- **Endpoint: /add**
- **Method: POST**
- **Description: Adds a new customer to the database.**
- **Response:**
    - **201 Created: Customer successfully added to the database.**
    - **500 Internal Server Error: Error adding the customer.**

---

4. Process Customer Credit

- **Endpoint: /processCredit/:id**
- **Method: PUT**
- **Description: Updates the credit amount of a specific customer.**
- **Parameters:**
    - **Path Parameter:**
        - **id: ID of the customer.**
    - **Body:**
        - **process_credit: The amount to update the customer's credit.**
- **Response:**
    - **200 OK: Credit processed successfully, returns the updated customer data.**
    - **400 Bad Request: If invalid or missing process_credit.**
    - **404 Not Found: If the customer is not found.**
    - **500 Internal Server Error: Error processing the credit.**

---

5. Edit Customer Details

- **Endpoint: /edit**
- **Method: POST**
- **Description: Edits the details of an existing customer.**
- **Response:**
    - **200 OK: Customer details updated successfully.**
    - **400 Bad Request: If required fields are missing or invalid.**
    - **404 Not Found: If the customer ID is not found.**
    - **500 Internal Server Error: Error updating the customer details.**

---

6. Delete Customer

- **Endpoint: /delete**
- **Method: POST**
- **Description: Deletes a customer by their ID.**
- **Response:**
    - **200 OK: Customer deleted successfully.**
    - **400 Bad Request: If the customer ID is missing or invalid.**
    - **404 Not Found: If the customer does not exist.**
    - **500 Internal Server Error: Error deleting the customer.**

---

7. Add New Customer (Alternate)

- **Endpoint: /addNewCustomer**
- **Method: POST**
- **Description: Adds a new customer to the database using query parameters.**
- **Response:**
    - **201 Created: Customer added successfully.**
    - **500 Internal Server Error: Error adding the new customer.**

Movie API

1. Fetch All Movies

- **Endpoint**: /get
- **Method**: GET
- **Description**: Retrieves all movies from the database.
- **Response**:
    - **200 OK**: List of movies.
    - **500 Internal Server Error**: Error fetching data.

---

2. Fetch Movie by ID

- **Endpoint**: /getEachMovie/:movie_id
- **Method**: GET
- **Description**: Fetch details of a movie by its ID.
- **Parameters**:
    - **Path Parameter**:
        - movie_id: ID of the movie.
- **Response**:
    - **200 OK**: Movie details.
    - **404 Not Found**: Movie not found.
    - **500 Internal Server Error**: Error fetching data.

---

3. Book Seats for a Movie

- **Endpoint**: /booking/:movie_id
- **Method**: POST
- **Description**: Book seats for a movie by updating the movie_cinema_seats field.
- **Parameters**:
    - **Path Parameter**:
        - movie_id: ID of the movie.
    - **Body**:
        - bookedSeats: Array of seat identifiers to book (e.g., ["A1", "A2"]).
- **Response**:
    - **200 OK**: Booking updated successfully.
    - **400 Bad Request**: No bookedSeats provided.
    - **404 Not Found**: Movie not found.
    - **500 Internal Server Error**: Error fetching or updating data.

---

4. Add a New Movie

- **Endpoint**: /add
- **Method**: POST
- **Description**: Adds a new movie to the database.
- **Body**:
  - movie_thumbnail: URL for the movie thumbnail.
  - movie_title: Title of the movie.
  - movie_trailer_video: URL for the movie trailer.
  - movie_genre: Genre of the movie.
  - movie_rate: Rating of the movie (e.g., 8.5).
  - movie_duration: Duration of the movie (e.g., 120 minutes).
  - movie_dub: Dub language of the movie.
  - movie_sub: Subtitle language of the movie.
- **Response**:
  - **201 Created**: Movie added successfully.
  - **500 Internal Server Error**: Error adding the movie.

---

5. Edit Movie Details

- **Endpoint**: /edit/:id
- **Method**: PUT
- **Description**: Edits details of an existing movie.
- **Parameters**:
    - **Path Parameter**:
        - id: ID of the movie to edit.
    - **Body**:
        - movie_thumbnail: Updated thumbnail URL.
        - movie_title: Updated title.
        - movie_trailer_video: Updated trailer URL.
        - movie_genre: Updated genre.
        - movie_rate: Updated rating.
        - movie_duration: Updated duration.
        - movie_dub: Updated dub language.
        - movie_sub: Updated subtitle language.
- **Response**:
    - **200 OK**: Movie updated successfully.
    - **500 Internal Server Error**: Error updating the movie.

---

6. Fetch Movie by ID

- **Endpoint**: /read/:id
- **Method**: GET
- **Description**: Fetch a specific movie's details by ID.
- **Parameters**:
    - **Path Parameter**:
        - id: ID of the movie to fetch.
- **Response**:
    - **200 OK**: Movie details.
    - **500 Internal Server Error**: Error fetching the movie.

---

7. Delete a Movie

- **Endpoint**: /delete/:id
- **Method**: DELETE
- **Description**: Deletes a movie by its ID.
- **Parameters**:
    - **Path Parameter**:
        - id: ID of the movie to delete.
- **Response**:
    - **200 OK**: Movie deleted successfully.
    - **500 Internal Server Error**: Error deleting the movie.

Role responsibility

นาย กฤชพล สุวรรณวุฒิ - Payment page, top up, home page(show user credit), movie management(delete movie)

นาย ฐิรวัฒน์ แซ่ลิ้ม - Movie management (add edit read delete) movie(show title (read)), goodjobhacker page

นาย กิตติกานต์ โคณบาล – Homepage, history, movie (select seat), sign in, sign up, forget password, movie management(delete movie)