

Welcome to our software project :

Eng:Mohammed Hammed

## Cinema Management System

### Team Members

حبيبہ رجب نصار سليم علي

رضوی عادل معروف الشوربجي

نورھان عبد المنعم محمود العدلي الخولي

## Functional Requirements

These are the specific behaviors or functions the system must exhibit.

---

### Register

The Register page in a Cinema Management System is designed for new Admins, to create an account.

The page features fields for entering essential information, including:

- Name
- Email
- Password

Additionally, there is a hyperlink labeled **"Already have an account?"** that allows existing admins to quickly navigate to the login page, facilitating easy access for those who have already registered.

---

### Login

The Login page of the Cinema Management System is a user-friendly interface designed for administrators to access the system securely.

The Login page includes:

- Two input fields where admins can enter their **email address** and **password**.
  - If an admin does not already have an account, the page includes a hyperlink titled **"Create a new account"** that directs them to the registration page to sign up for a new account.
-

## Admin Dashboard

The Dashboard page of the Cinema Management System serves as a central hub for administrators, providing essential insights and functionality for efficient management.

### Features of the Admin Dashboard:

- **Available Movies:** Displays the number of movies currently available in the system.
  - **Total Sold Tickets:** Highlights the total number of tickets sold.
  - **Total Income:** Displays the income generated by the current admin.
  - **Navigation Frames:** Allows admins to switch between the following options:
    - Add Movies
    - View Available Movies
    - Edit showtimes
    - View Sales Information
  - **Sign out:** A secure **Sign out** button enabling admins to exit the system when their tasks are complete.
  - **Popular Movie Poster:** Displays a poster of a popular movie currently available in the system.
- 

## Edit Showtime

In the **Edit Showtime Page**, the admin can use a **search bar** to quickly find and select any movie.

Once the desired movie is selected, the admin can update its current show status by selecting one of the following options:

- **Showing**
  - **End Showing**
-

## CRUD Operations

The system will support the necessary features for performing **CRUD operations** (Insert, Read, Update, Delete) on movies.

### Insert:

- **Image Upload:**
    - Label: **"Import"**
    - Input Type: File Upload (allows admin to upload an image)
  - **Movie Title:**
    - Label: **"Movie Title"**
    - Input the title of the movie.
  - **Genre:**
    - Label: **"Genre"**
    - Input the genre of the movie (e.g., Action, Comedy, Drama, etc.).
  - **Movie Duration:**
    - Label: **"Duration"**
    - Input Type: Text (e.g., "120 minutes").
  - **Published Date:**
    - Label: **"Published Date"**
    - Input Type: Date Picker (to select a date).
- 

### Read:

The admin can view available movies in a table that contains all the movies in the system. They can use a **Search Bar** to quickly find and access specific movies.

---

### Update:

After searching for a movie and selecting it from the table:

- The form fields will populate with the movie's existing data.

- The admin can modify the fields as needed.
  - Upon clicking **"Update Movie,"** the system will validate the input fields.
    - If validation passes, the movie details will be updated in the database.
    - The system will display a success or error message.
- 

#### Delete:

After searching for a movie and selecting it from the table:

- Clicking the **"Delete Movie"** button will prompt a confirmation dialog:
  - **"Are you sure you want to delete this movie?"**
- If confirmed, the movie will be removed from the database.
- The movie table will refresh, and a success or error message will appear.

#### Additional Feature:

- **Clear Button:** Clears all input fields in the form.
- 

## Booking Tickets

The **Booking Tickets Page** allows the admin to efficiently control ticket sales.

#### Features:

1. The admin can select a movie from a table displaying all available movies in the system.
  2. After choosing a movie, the admin specifies the type of tickets (Special or Normal).
  3. The admin enters the number of tickets to be booked.
  4. Upon confirmation, the system finalizes the booking process.
-

## Sales Information

The **Sales Information Page** provides a detailed overview of ticket sales attributed to the current admin.

### Features:

- Displays a table with the following columns:
    - Number of Tickets
    - Movie Title
    - Date Checked
    - Time Checked
  - The admin has the capability to delete any sale from the system.
- 

## Non-Functional Requirements

These include the qualities and constraints under which the system must operate.

---

### Reliability

The system must have an uptime of **99.9%**, ensuring continuous availability for admins.

#### Improving Reliability:

- **Redundancy:** Implementing backup systems (e.g., redundant servers, network paths).
  - **Testing and Validation:** Continuous testing to identify and correct potential failures.
-

## Availability

### Uptime:

The system should have **99.9% uptime** to ensure its availability for admins.

### Improving Availability:

- Use **cluster grouping** with multiple servers/nodes to ensure seamless failover in case of failures.
- 

## Performance

The pages must load within **3 milliseconds**, and booking actions should process in under **3 milliseconds**.

### Improve Performance:

- Implement **caching strategies** to store frequently accessed data.
  - Optimize database queries and use indexing for faster retrieval.
- 

## Usability

The user interface must be simple, intuitive, and easy to navigate. Error messages should be clear and helpful for administrators.

### Improving Usability:

- Design a clean and minimalistic UI optimized for desktop applications.
-

## Security

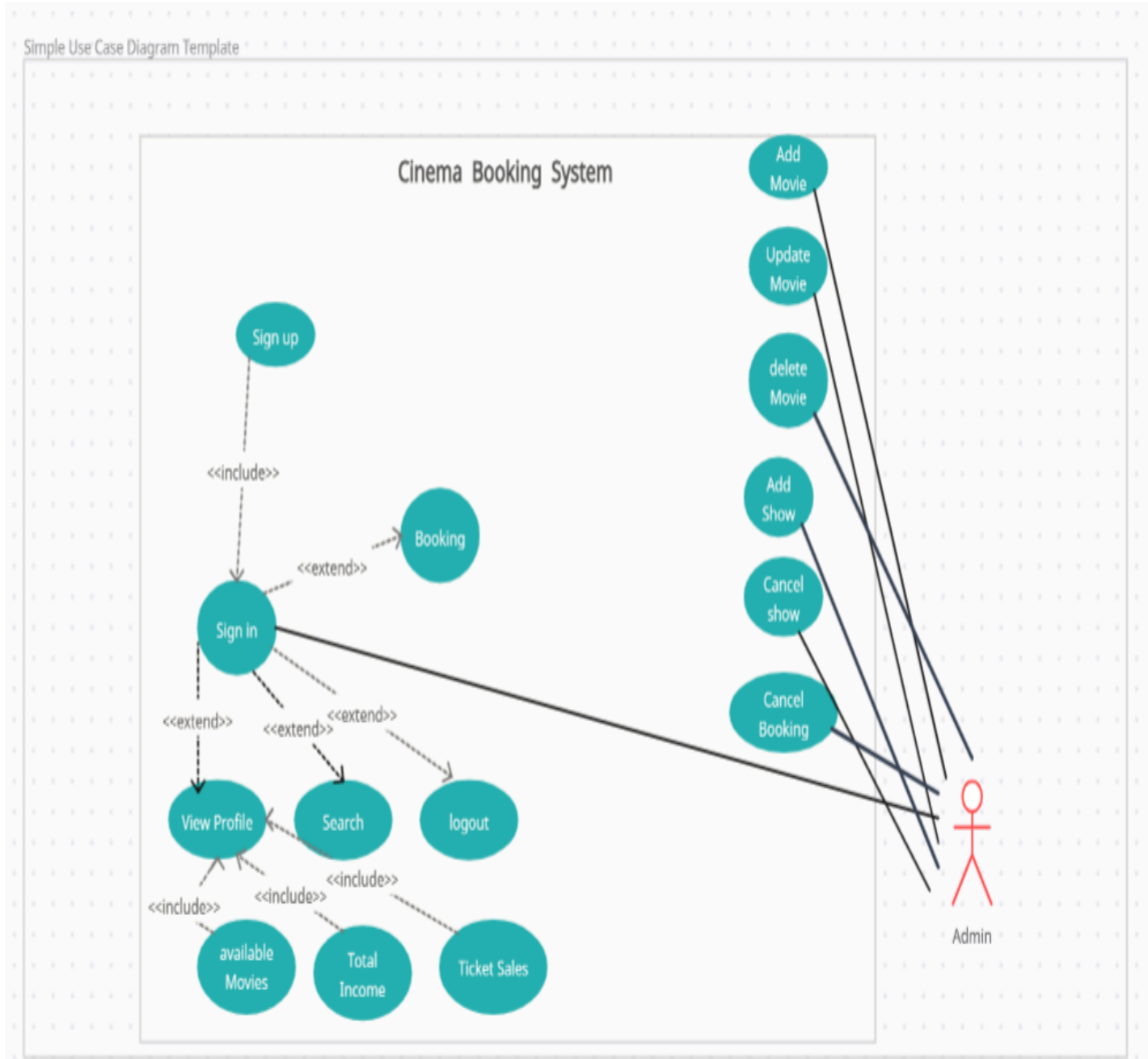
All admin data must be securely encrypted.

### Strategies for Security:

- Use **Web Application Firewalls (WAFs)** to protect the system from threats like SQL Injection and XSS (cross-site scripting).



## Use Case Diagram :



## Use Case Descriptions:

### Use Case 1: Register

**Actor:** New Admin

**Trigger:** Allows new Admin users to create an account in the Cinema Management System.

**Preconditions:**

The user must have access to the Register page.

The user must not have an existing account in the system.

**Flow of Events:**

1. **Access Register Page:** The new Admin navigates to the Register page.

2. **Enter Name:** The user inputs their name in the designated field.

3. **Enter Email:** The user inputs their email address in the designated field.

4. **Enter Password:** The user inputs their password in the designated field, ensuring it meets the required security criteria (e.g., minimum length, special characters).

5. **Submit Registration:** The user clicks the "Sign up" button to submit their information.

**6. Account Creation:** If validation is successful, the system creates a new Admin account and sends a confirmation message to the user: *“successful Add new admin!”*

**Postconditions:**

A new Admin account is created in the system.

**Alternative Flow (Error Handling):**

**Invalid Input:** If any of the input fields are invalid (e.g., email format is incorrect, password does not meet requirements), the system displays an error message indicating the specific issue. The user corrects the input and re-attempts registration.

**Email Already Exists:** If the email address is already associated with an existing account, the system displays a message indicating that the email is already in use. The user can choose to log in instead or provide a different email address.

---

## Use Case 2: Login

**Actor:** Admin

**Trigger:** Describes the process by which administrators securely log into the Cinema Management System through the Login Page.

**Preconditions:**

The administrator must have access to the Login Page.  
The user must have an existing account in the system.

**Flow of Events:**

1. **Access Login Page:** The administrator navigates to the Login Page.
2. **Input Credentials:**
  - The administrator enters their **email address** in the first input field.
  - The administrator enters their **password** in the second input field.
3. **Submit Login:** The administrator clicks on the "**Login**" button to submit their credentials.
4. **Validate Credentials:** The system validates the entered

credentials. If valid, the system displays the welcome message or the admin dashboard.

**5. Create New Account (Optional):** If the administrator does not have an account, they can click the "**Create a new account**" hyperlink. The system redirects the administrator to the registration page to sign up for a new account.

**Postconditions:**

If the login is successful, the administrator gains access to the Cinema Management System dashboard.

**Alternative Flow (Error Handling):**

**Invalid Credentials:**

1. After entering invalid credentials, the administrator is informed of the error and remains on the Login Page.
  2. The administrator can attempt to log in again or choose to create a new account.
-

## Use Case 3: Admin Dashboard

**Actor:** Administrator

**Trigger:** Allows administrators to manage reports, permissions, and customer information.

**Preconditions:**

The admin must be logged into the system.

**Flow of Events:**

1. The admin logs into the system and navigates to other pages.
2. The system displays ticket sale reports, total income, number of available movies, and a poster of a popular movie in the system.
3. The admin views and analyzes the provided data.

**Postconditions:**

Reports are successfully generated.

---

## Use Case 4: CRUD Operations

**Actor:** Administrator

**Trigger:** Allows the admin to add, edit, or remove movies.

**Preconditions:**

- The admin is logged into the Cinema Management System.
- The admin has the necessary permissions to perform CRUD operations on movies.

**Flow of Events:**

1. The admin navigates to the **Available Movie** section.
2. The admin selects an action: **Add, Update, or Delete**.
  - For **Add**: The admin inputs movie details, and the system saves them.
  - For **Update**: The admin modifies existing details, and the system updates the database.
  - For **Delete**: The admin selects a movie, and the system deletes it from the database.
  - For **Clear**: The admin clicks the "**Clear**" button to reset all input fields in the form.

**Postconditions:**

- Movies are successfully added, updated, or deleted from the system's database.
- The movie inventory reflects the latest changes made by the admin.

**Alternate Flow:**

-**Validation Errors:** If any input fields fail validation during Insert or Update, an error message is displayed, indicating which fields require correction.

**-Special Requirements:**

The system must ensure that movie images are of acceptable format and size. Input validation is required for all fields to ensure data integrity.

---

**Use Case 5: Edit Showtime**

**Actor:** Administrator

**Trigger:** Allows the admin to edit the show status of movies .

**Preconditions:**

- The admin is logged into the system.



- The admin has access to the Edit Showtime Page.
- The list of movies is available in the system.

### **Flow of Events:**

- 1- The admin navigates to the Edit Showtime Page.
- 2- The admin uses the search bar to enter the movie title or relevant keywords.
- 3- The system displays a list of movies matching the search criteria.
- 4- The admin selects the desired movie from the list.
- 5- The system presents the current show status of the selected movie.
- 6- The admin selects a new show status from the options: "Showing" or "End Showing."
- 7- The admin confirms the update. The system updates the show status of the movie.
- 8- The system displays a success message indicating that the update was successful.

### **Postconditions:**

- The selected movie's show status is updated in the system.

-The change is reflected in the system for users viewing the movie listings.

### **Alternate Flow:**

-**Validation Errors:** If any fail validation during edit showtime , an error message is displayed, indicating which fields require correction.

---

## **Use Case 6: Ticket Booking**

**Actor:** Administrator

**Trigger:** The system administrator manages ticket bookings.

### **Preconditions:**

- The admin is logged into the system.
- Movies are available for booking.

### **Flow of Events:**

1. **Access Booking Tickets Page:** The admin navigates to the **Booking Tickets** page.
2. **Select Movie:** The system displays a table of all

available movies. The admin selects a movie from the table.

3. **Choose Ticket Type:** The admin specifies the ticket type: **Special** or **Normal**.

4. **Enter Number of Tickets:** The admin enters the number of tickets to book.

5. **Finalize Booking:** The system processes the booking and updates the ticket inventory.

### **Postconditions:**

A ticket booking is successfully recorded in the system.

### **Alternate Flow :**

#### **Available Movie not selected for Booking:**

- Flow of Events: The system displays a message indicating that the “please select available movie”

#### **Ticket type number not selected :**

- Flow of Events: The system displays an error message indicating that the “please select ticket type number ”
- The admin is prompted to choose a valid ticket type (Special or Normal).

#### **Booking Process Failure :**

- The system displays an error message indicating that the booking could not be processed due to a technical issue.

---

## **Use Case 7: Sales Information**

**Actor:** Administrator

### **Triggers:**

Allows the admin to view detailed information about ticket sales attributed to them.

### **Preconditions:**

- The admin must be authenticated and logged into the system.
- The admin has access to the Sales Information Page.

### **Flow of Events:**

1. **Access Sales Information Page:** The admin navigates to the Sales Information Page.
2. **Display Sales Information:** The system retrieves and displays a table with the following columns:
  - Number of Tickets Movie
  - Movie title
  - DateChecked
  - Time Checked
3. **View Sales Records:** The admin reviews the sales records displayed in the table.

4. **Delete a Sale Record:** The admin selects a specific sale record they wish to delete.
5. **The system prompts** the admin to confirm the deletion. Upon confirmation, the system removes the selected sale from the database.
6. **The table** is refreshed to reflect the updated list of sales.

### **Postconditions:**

- The Sales Information Page displays an up-to-date list of ticket sales.
- Any selected sale can be deleted from the system if the admin chooses to do so.

### **Alternate Flow :**

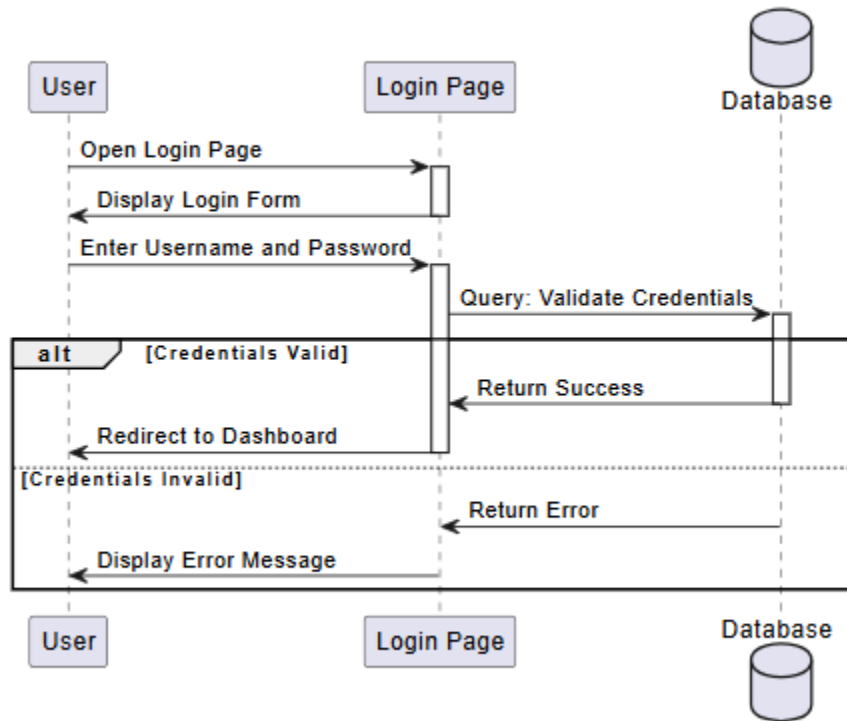
- If admin not selected a sale record for deletion, system prompts display message “please select record sale info!”
- The admin selects a specific sale record they wish to delete, system prompts the admin to confirm the deletion.

- Cancel Deletion: The admin decides not to proceed with the deletion and selects the "Cancel" option.

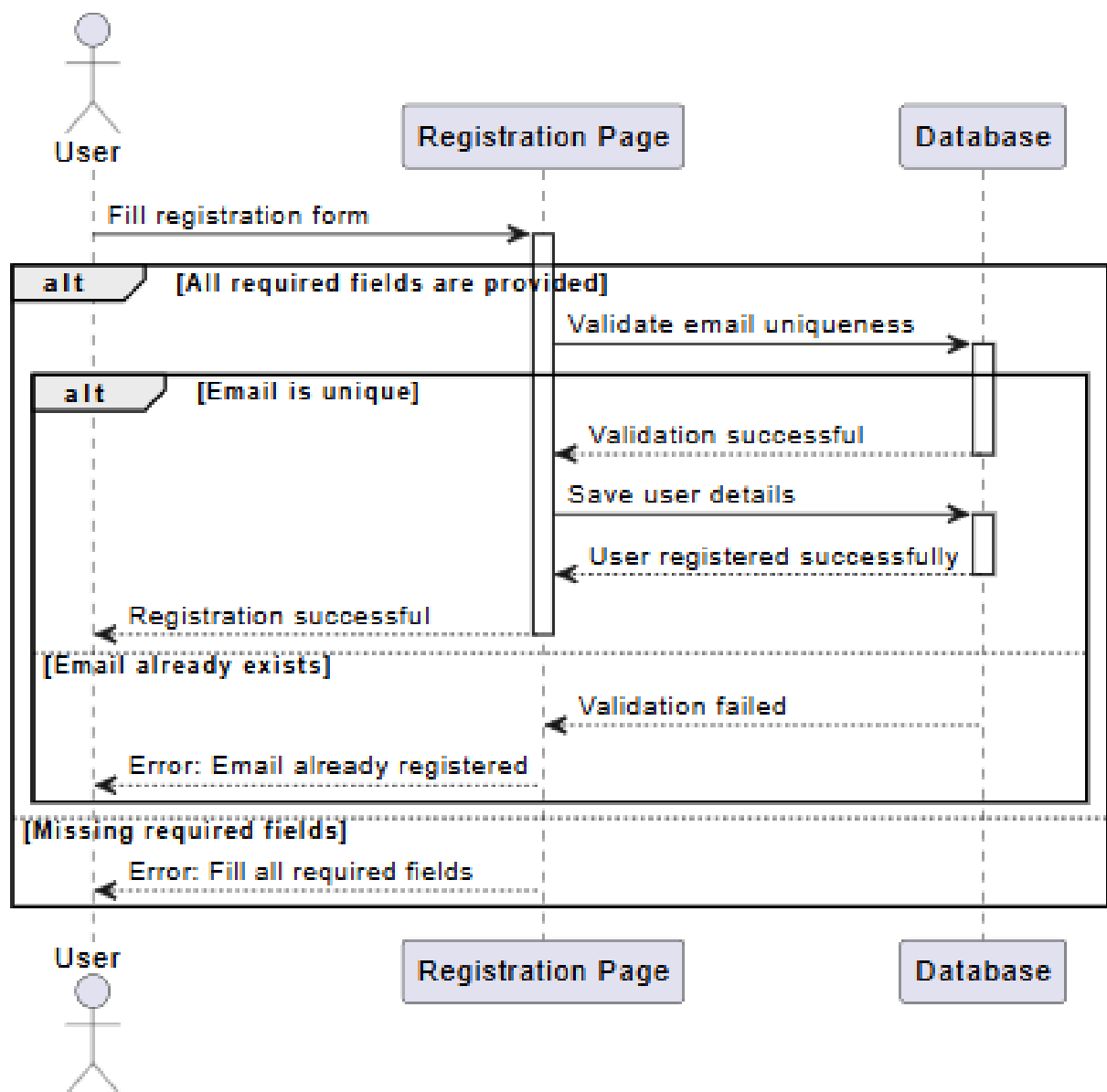
**End of Alternate Flow:**

- The system retains the selected sale record, and the sales table remains unchanged.
  - The admin can continue to view or interact with the sales records.
-

# Login:

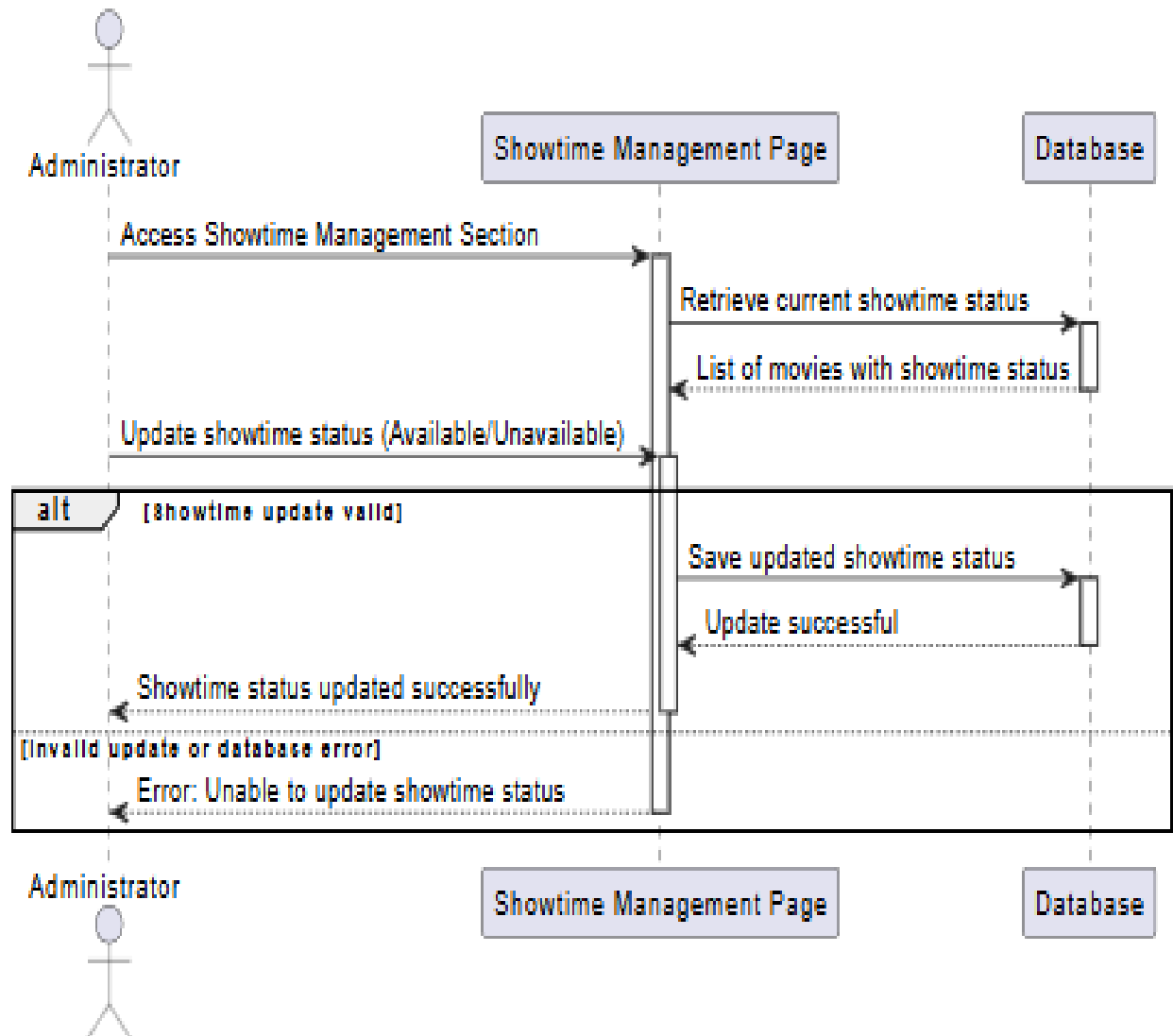


## Registration:

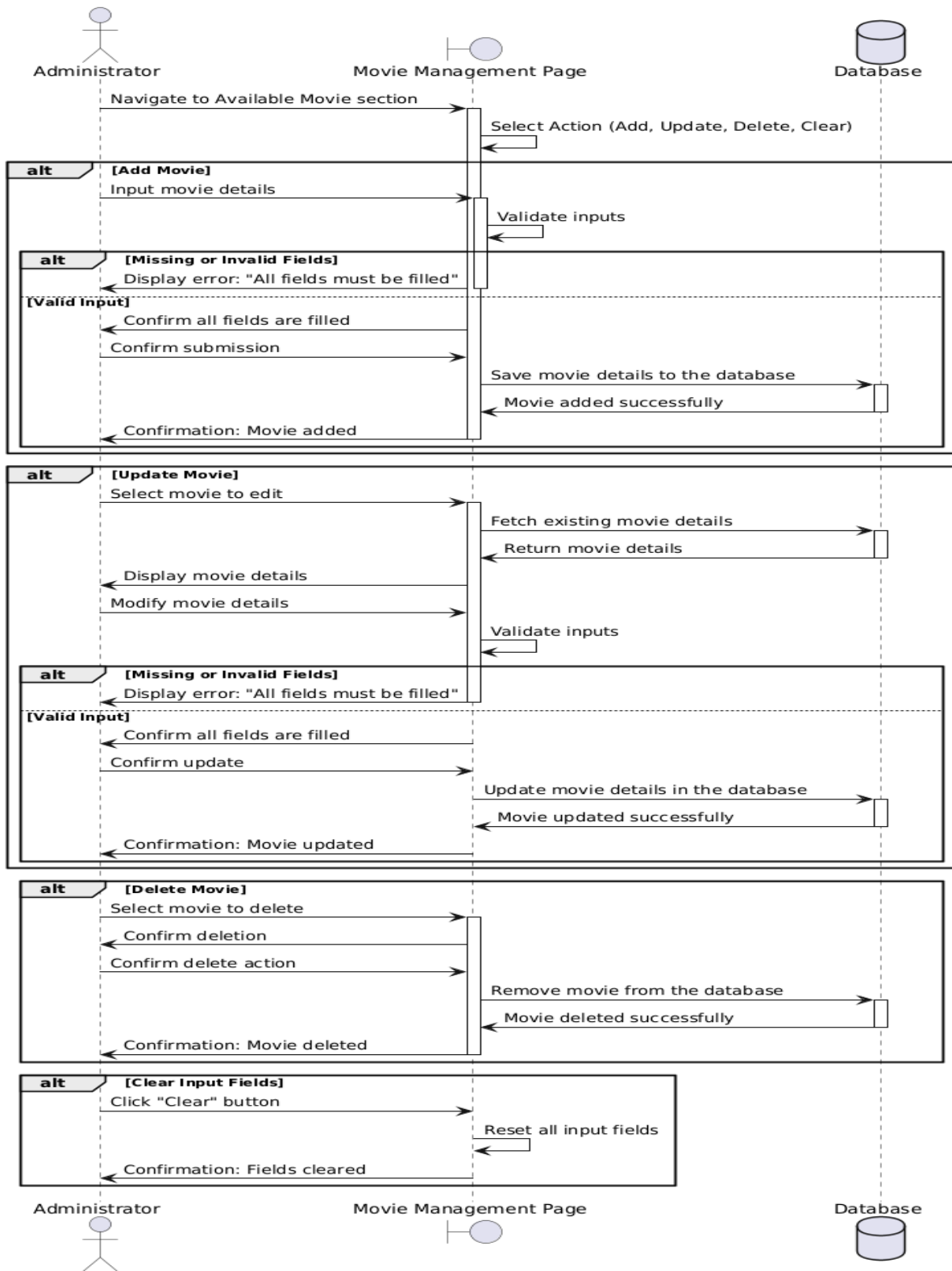




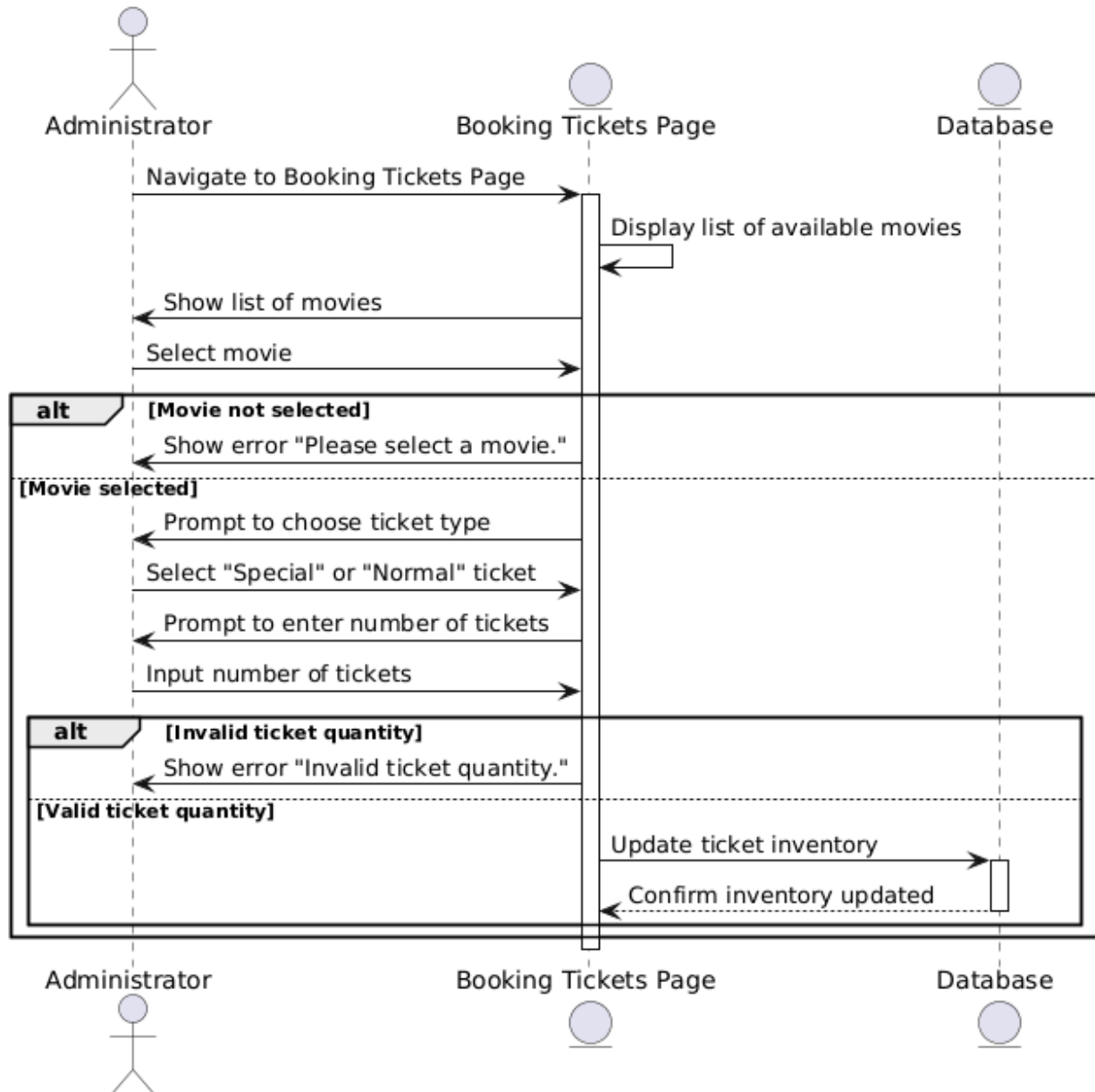
## Edit Showtimes :



# Available Movies(CRUD Operations ):

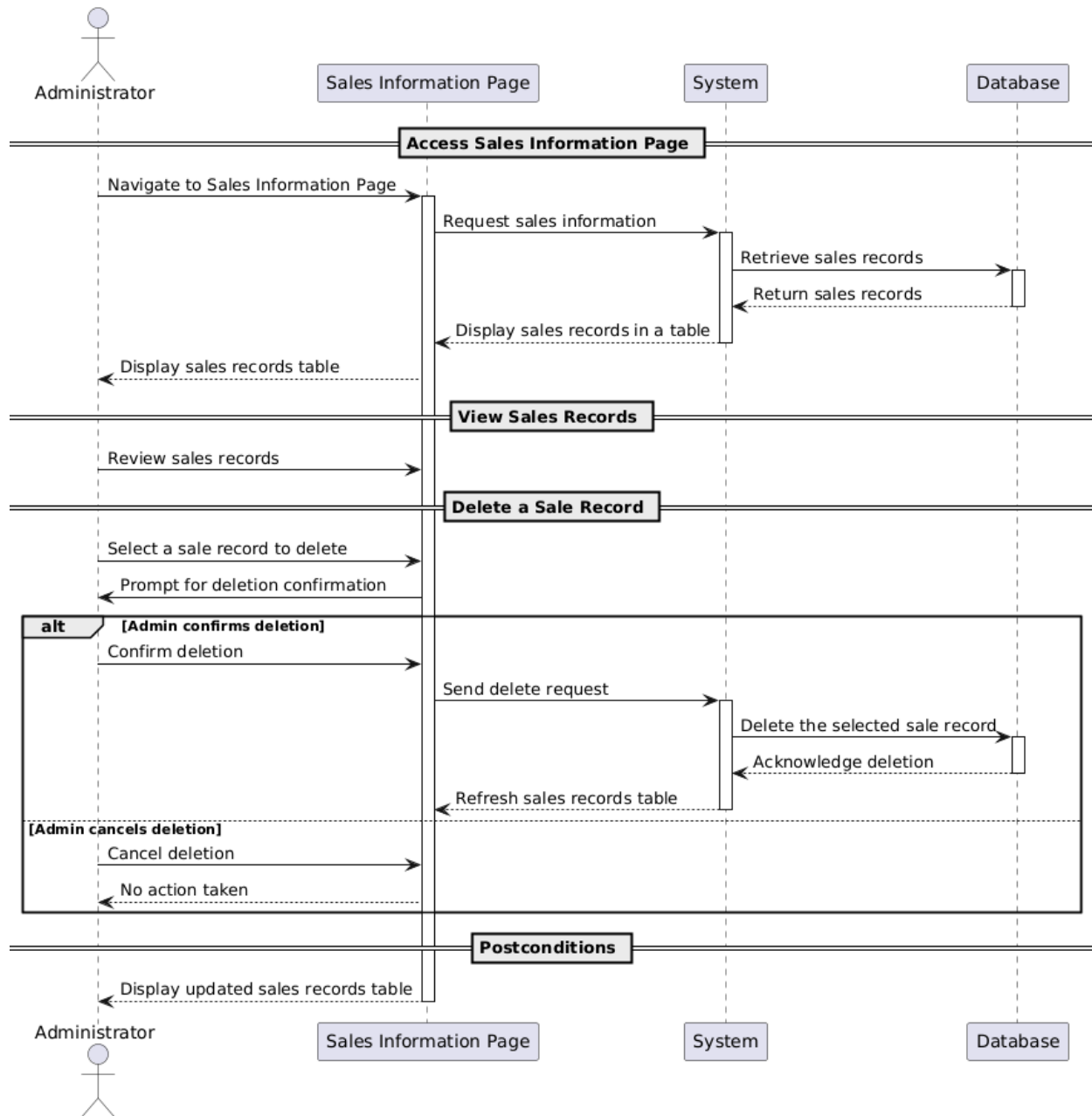


## Ticket Booking:

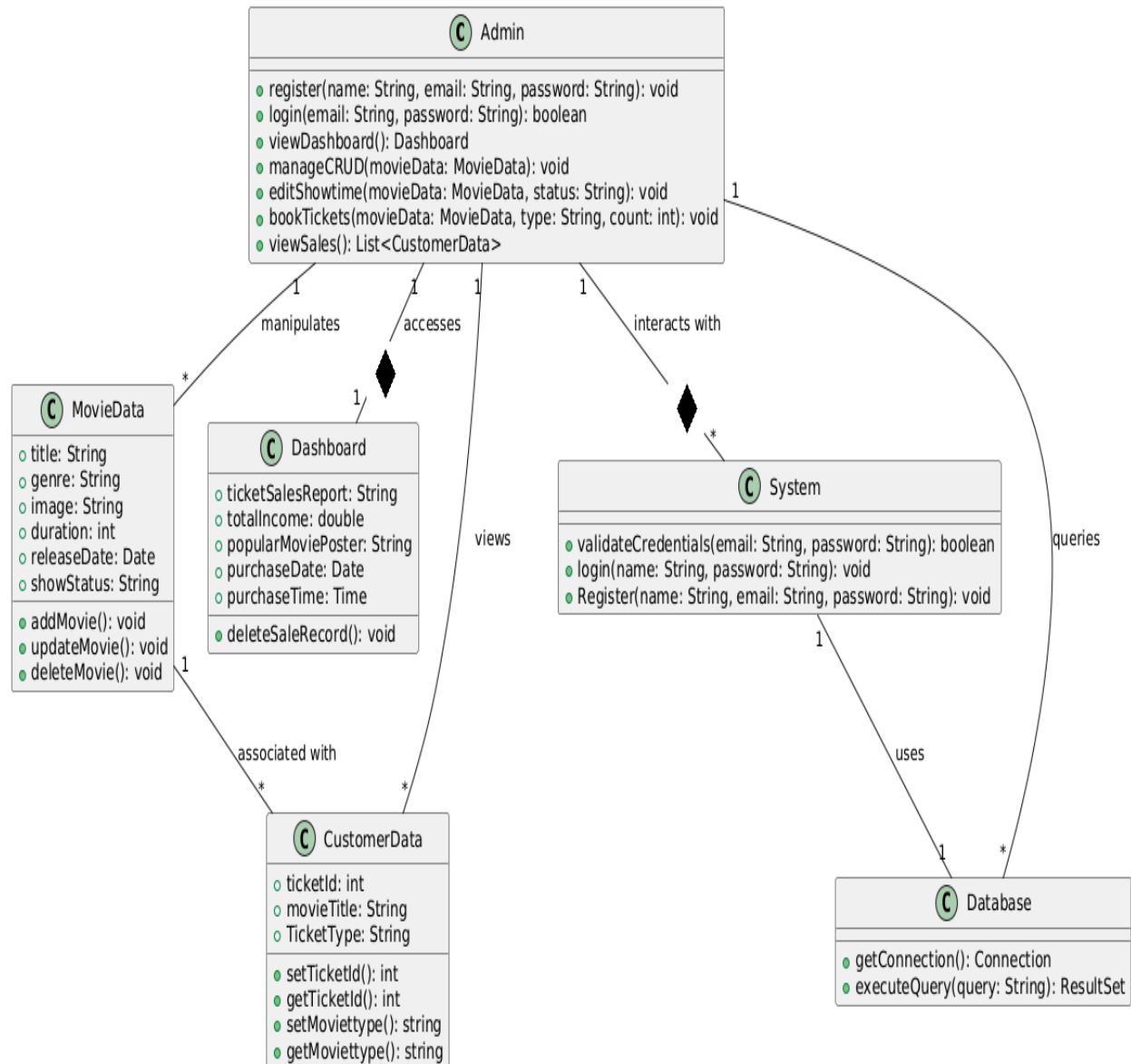


# Sales information:

Sequence Diagram for Sales Information Use Case



## Class Diagram:



## Object Diagram:

