

Software Requirements Specification (SRS)

E-ticketing System

Team Members:

- Ankit Balyan 22BCE7599

- Mukul Singh 22BCE8235

- Chandsi Solanki 22BCE8571

- Subhadeep Singh 22BCE9006

- Piyush 22BCE8273

Date:

17/08/2024

|  |
| --- |
| **Revisions** |

| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| **First Draft** | **Team** | **V 0.1 – Description and Planning** | **17/08/24** |

|  |
| --- |
| **Review & Approval** |

**Requirements Document Approval History**

| **Approving Party** | **Version Approved** | **Signature** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Requirements Document Review History**

| **Reviewer** | **Version Reviewed** | **Signature** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Contents** |

1. Introduction ………………………………………………………………………………… (4)

2. General Description …………………………………………………………………………… (5)

3. Functional Requirements……………………………………………………………………… (7)

4. Interface Requirements ……………………………………………………………………… (8)

5. Non-Functional Requirements ……………….………………………………………….…. (11)

6. Operation Scenarios ……………….………………………………………….…. (12)

7.Preliminary UML Diagrams ……………….………………………………………….…. (13)

|  |
| --- |
| **1. Introduction** |

An Electronic Ticket, more efficient method of ticket entry, processing, and marketing for companies in transport, airlines, railways and other entertainment industries. Many venues/events such as a cricket match use electronic ticket or ‘E-Tickets’ for their events. We are implementing this system to movie theatres and the tickets are delivered are often delivered to us in the form of pdf, image or other downloadable formats that can be received through emails, SMS or through mobile applications which should be presented on the coming of the venue.

**1.1 Purpose**

The purpose of this document is to define and describe the requirements for the E-ticketing system in movie theatres, outlining the system’s functionality and its constraints. This document is intended for the following group of people:

* Developers for the purpose of maintenance and updates
* Documentation writers
* Management
* Testers

**1.2 Scope**

This document applies to the E-ticketing website of a movie theatre. This software is for designing tickets, managing reservations and creating unique codes for every ticket. It allows the user to book a ticket for a movie at their theatre and wished time (at user’s convenience).

The software takes as input the e-mail id and the phone number for primary verification to create an account. Payments should be done through online transaction like net banking, or through debit/credit cards or UPI Id’s.

After the payment, the software produces an e-ticket as an output. The user can download it in pdf format.

**1.3 Overview**

The E-ticketing system aims to streamline the process of booking, managing, and validating electronic tickets for movie theatres for simplifying the experience and providing ease of access from the couch at home. The system will include features for user registration, ticket purchasing, and real-time updates.

**1.4 Business Context**

The E-ticketing system will be used by Theatre staff and end-users to manage ticket sales and ensure a seamless ticketing experience. The system aims to reduce manual processes, improve efficiency, and enhance user satisfaction.

|  |
| --- |
| **2. General Description** |

This software allows the user to access various theatres and shows in them through a webpage. This software also helps the user to search for the liked movies and display available shows for them in different theatres. The user is supposed to select wished seats and book them by making an online transaction

An e-ticketing system is a digital platform used to issue, manage, and validate tickets electronically, replacing traditional paper tickets. Here's a general overview:

**2.1 Product Functions**

**2.1.1 Administration**

* Admin should be able to add/delete shows from the site
* Admins can reject the users if he/she does not accept terms and conditions.

**2.1.2 Users**

* Account: Members are given a provision to check their account’s information and change it.
* Availability: User can search presently available shows or upcoming shows in different theatres.
* Booking: User can book required number of seats by selecting empty seats and then pay for them.
* Transaction: To book the seats they user must pay the total amount including taxes and other charges through online transactions like net banking, or debit/credit cards, or through UPIs. A transaction id is created and given to the user for future purposes.
* Ticketing: A ticket is produced with seat numbers, time and place of show on it and also a QR code.
* Cancellation: Users may also unbook seats and get 50% cashback excluding taxes.

**2.2 User Characteristics**

Users of the system include:

- General public: purchasing tickets.

- Event organizers: managing events and ticket sales.

- Administrators: overseeing system operations.

There are different kinds of users that will be interacting with the system. The intended users of the software are as follows:

* USER 1: An inexperienced customer. This user has little to no experience with E-Ticketing systems or other means of electronic bookings a ticket or not a frequent user of the product. The users will find the portal friendly and easy to use and an easy way to book tickets.
* USER 2: An experienced customer. The user has used the portal several times and does most of his/her bookings through online platform. There is only little help session that too at the beginning of the session thus making the booking procedure easy and faster.
* Maintenance personnel: An administrator of the system should have more knowledge of the internal modules of the system and are able to rectify small problems that may arise due to disk crashes, power failures and other catastrophes. Friendly user interface, online help and user guide must be sufficient to educate the user how to use this product without any difficulties.

**2.3 User Problem Statement**

Current manual ticketing systems are inefficient, error-prone, and slow. The E-ticketing system aims to address these issues by automating ticket sales and validation processes.

**2.5 User Objectives**

Users seek a reliable, user-friendly system that allows for quick and secure ticket transactions, real-time updates, and easy event management.

**2.6 General Constraints**

The general constraints that the project has are as follows:

* The information of all the users and the shows must be stored in a database that is accessible by the website.
* The number of invalid pin entries must not exceed five. After five unsuccessful attempts, the user can create a new password by clicking ‘Forgot Password?’ and confirm the mail sent to their respected e-mail id’s.
* The password must be between 8 to 20 characters with one uppercase, one lower case, one number and one special character.
* The transaction must be successful for the seat to be booked.
* The user can access the portal from any device with good internet connection and browsing capacities.
* SQL used will be Oracle 11g.
* For Website we will use HTML, CSS and JavaScript
* The system must be compatible with modern web browsers and mobile devices.
* It must support secure payment processing.
* The system should be scalable to handle high traffic volumes during peak times.

|  |
| --- |
| **3. Functional Requirements** |

**3.1 User Registration and Authentication**

- Users must be able to create an account with a valid email address and password.

- The system shall provide secure authentication mechanisms.

- The user must be registered to book the tickets.

- Customer logins to the portal by entering valid email and password for the seats to be booked.

**3.2 Ticket Search and Purchase**

- Users shall be able to search for Movies, Theatres and shows.

- Users shall be able to select and purchase tickets using various payment methods.

- User is able to select a certain number of seats at the liked place.

- Users are able to check availability of the seats.

- Payment is done through net banking or debit/ credit cards or UP. The seats will be booked and blocked only after successful payments

- Receipt Generation: A unique transaction id is generated for future references.

**3.3 Ticket Generation and Validation**

- Ticket Generation: A ticket is produced with the seat numbers time and place of the show on it and also a QR

- The system shall generate electronic tickets with unique identifiers.

- Tickets must be validated at the entry point using QR codes or other validation methods.

**3.4 Administrative Functions**

- Administrators shall be able to create and manage events.

- The system shall provide reports and analytics on ticket sales.

|  |
| --- |
| **4. Interface Requirements** |

**4.1 User Interfaces**

- The system shall provide a graphical user interface (GUI) for both web and mobile platforms.

- The interface must be intuitive and easy to navigate.

The User can navigate as follows

1. A login screen is provided in the beginning for entering the required username/pin no. and account number.
2. An unsuccessful login leads to a reattempt (maximum five) screen for again entering the same information. The successful login leads to a screen displaying a list of supported languages from which a user can select anyone.
3. In case of administrator, a screen will be shown having options to reboot system, shut down system, block system, disable any service.
4. In case of reboot/shut down, a screen is displayed to confirm the user's will to reboot and also allow the user to take any backup if needed.
5. After the login, a screen with a number of options is then shown to the user. It contains all the options along with their brief description to enable the user to understand their functioning and select the proper option.
6. A screen will be provided for user to check his wallet balance.
7. A screen will be provided for the user to perform various transactions in his account like purchase, search or cancelling of the ticket.

**4.2 Hardware Interfaces**

- The Hardware interfaces will be implemented at the movie theatres

- The system shall interact with barcode scanners and printers for ticket validation and printing.

**1. Barcode Scanners**

**Objective:** To scan and validate electronic or printed tickets at the theatre.

**Required Interfaces:**

* **Connection Type:**
  + **USB:** Most barcode scanners connect via USB, providing a plug-and-play interface with the system. The scanner acts as a keyboard input device, sending scanned data directly to the computer or terminal.
  + **Bluetooth:** Wireless barcode scanners use Bluetooth to communicate with the system. This is useful for mobility and reducing cable clutter.
  + **Serial (RS232):** Older models or specialized scanners might use serial connections. These require a serial port or a USB-to-serial adapter.
* **Communication Protocols:**
  + **Data Transfer:** The scanner transmits scanned barcode data in real-time to the connected system. The data is typically formatted as text, which can be processed by the ticketing system.
  + **Configuration:** Some scanners allow configuration of settings such as barcode type (e.g., QR codes, Code 128), data formatting, and prefix/suffix settings.
* **Integration with Software:**
  + **Validation:** The scanned barcode data is sent to the e-ticket management system for validation against a database of issued tickets.
  + **Error Handling:** The system must handle errors such as invalid barcodes or failed scans, providing appropriate feedback to users.

**2. Other Possible Hardware Components**

* **Receipt Printers:** If the system provides receipts for purchases, integration with receipt printers (which may also use thermal printing technology) is necessary.
* **Self-Service Kiosks:** For theatres with self-service ticketing kiosks, these would integrate both barcode scanners and printers, and potentially include touch screens for user interactions.
* **POS (Point of sale) Terminals:** For box office or concessions sales, integration with POS terminals that include barcode scanners and receipt printers may be required.

**4.3 Communications Interfaces**

- The system shall support integration with email and SMS gateways for notifications.

- It shall provide APIs for third-party integration.

**Email Service Integration:**

* **Purpose:** Send email confirmations, receipts, and notifications to users.
* **Example Services:** Integrate with services like SendGrid or Mailgun to handle email communication.

**Authentication Providers:**

* **Purpose:** Allow users to log in or sign up using third-party accounts.
* **Example Services:** Integrate with OAuth providers like Google or Facebook for user authentication.

**Customer Support Systems:**

* **Purpose:** Provide users with support through various channels.
* **Example Systems:** Integrate with support platforms like Zendesk or Intercom for handling customer queries and issues.

**Analytics and Tracking:**

* **Purpose:** Track user interactions and analyse system performance.
* **Example Services:** Integrate with analytics platforms like Google Analytics to monitor user behaviour and system metrics.

**4.4 Software Interfaces**

The e-ticket management system for movie theatres requires integration with payment gateways like PayPal and Stripe for processing transactions, and with database management systems such as MySQL or PostgreSQL for data storage and retrieval. Additionally, interfaces with email services, authentication providers, customer support systems, and analytics tools may enhance the functionality and user experience of the system. Proper implementation of these interfaces ensures a seamless, secure, and efficient ticketing process.

**1. Payment Gateway Integration**

**Objective:** Enable users to make payments for movie tickets through secure and reliable payment processing services.

**Required Interfaces:**

* **PayPal Interface:**
  + **API Integration:** Utilize PayPal’s REST APIs(Representational State Transfer Application Programming Interface) to handle payment transactions. This involves sending payment requests to PayPal and receiving responses regarding transaction status.
  + **Checkout Integration:** Implement PayPal’s checkout experience, allowing users to pay using their PayPal accounts or credit/debit cards.
  + **Webhooks:** Set up webhooks to receive notifications from PayPal about payment status updates, such as payment completions or failures.
* **Stripe Interface:**
  + **API Integration:** Use Stripe’s APIs to manage payments, including creating payment intents, handling successful payments, and dealing with payment errors.
  + **Payment Forms:** Integrate Stripe’s pre-built payment forms or components to securely collect payment details from users.
  + **Webhooks:** Configure webhooks to get real-time updates about payment events, such as successful payments or disputes.

**2. Database Management System Integration**

**Objective:** Store and manage data related to movie showtimes, ticket sales, user information, and more.

**Required Interfaces:**

* **MySQL:**
  + **Database Connection:** Set up connections to MySQL databases using appropriate database drivers or connectors.
  + **Schema Design:** Design database schemas (database structure) to store information about movies, theatres, showtimes, users, and transactions.
  + **Query Execution:** Use SQL queries to interact with the database, including operations like inserting new records, updating existing records, and retrieving data.

|  |
| --- |
| **5. Non - Functional Requirements** |

**5.1 Security**

* **SSL Encryption for All Transactions**: All communications between users and the system, such as searching for movies, booking tickets, or making payments, must be encrypted using SSL (Secure Socket Layer). This ensures that sensitive data like credit card details and personal information are protected from interception by unauthorized parties.
* **Compliance with Data Protection Regulations**: User data, including personal information and payment details, must be securely stored in accordance with relevant data protection laws (such as GDPR in Europe or CCPA in California). This includes ensuring data is encrypted, access is restricted, and user privacy is maintained.

**5.2 Binary Compatibility**

* **Compatibility with Modern Operating Systems and Browsers**: The system should work seamlessly across all modern operating systems (like Windows, macOS, iOS, and Android) and web browsers (such as Chrome, Firefox, Safari, and Edge). This ensures users can access the ticketing service from any device without compatibility issues.

**5.3 Reliability**

* **Uptime of 99.9%**: The system must be highly reliable, with an uptime of at least 99.9%. This means that the system should only experience a maximum downtime of around 8.76 hours per year, ensuring users can book tickets almost all the time without disruption, which is critical during peak times such as weekends and holidays.

**5.4 Maintainability**

* **Designed for Easy Maintenance and Updates**: The system should be built in a modular and clear manner, making it straightforward for developers to perform updates, add new features, or fix bugs. This is important for quickly adapting to changes in movie offerings, ticket pricing, and user interface improvements.

**5.5 Portability**

* **Portable Across Different Platforms**: The system should be accessible from various platforms, including web browsers on desktop and mobile devices. This ensures that users can book tickets from anywhere, whether they're at home using a computer or on the go using their smartphone.

**5.6 Extensibility**

* **Designed to Allow for Future Enhancements**: The system should be built with an architecture that supports easy addition of new features. For instance, adding a loyalty program, implementing new payment methods, or integrating with third-party services like food delivery for movie snacks should be straightforward.

**5.7 Reusability**

* **Reusable Components for Other Projects**: Components of the system, such as the payment processing module, user authentication, or the ticket booking workflow, should be designed in a way that they can be reused in other similar applications, like booking systems for concerts or sporting events.

**5.8 Application Affinity/Compatibility**

* **Compatibility with Existing Ticketing Infrastructure**: The E-ticketing system must be able to integrate smoothly with existing systems used by movie theatres, such as point-of-sale (POS) systems, loyalty management systems, and auditorium management software. This allows for seamless operation and avoids the need for theatres to overhaul their existing systems.

**5.9 Resource Utilization**

* **Efficient Use of Server and Network Resources**: The system should be optimized to use server and network resources efficiently, ensuring fast response times and minimizing costs related to server maintenance and bandwidth. This is crucial during high-traffic periods, such as the release of a popular movie.

**5.10 Serviceability**

* **Easy Troubleshooting and Support**: The system should have built-in tools and logging capabilities that allow for easy monitoring, troubleshooting, and support. This includes error logs, system alerts, and diagnostic tools that help identify and resolve issues quickly, minimizing downtime and maintaining user satisfaction.

|  |
| --- |
| **6. Operation Scenarios** |

1. **User Registration and Login**
   * The user accesses the platform, creates an account with a valid email, and sets a password. After that, the user logs in to the system using those credentials.
2. **Ticket Search**
   * The user searches for available movies and theatres. They can filter results based on location, date, and time.
3. **Ticket Booking**
   * The user selects a show, chooses a number of seats, and proceeds to payment.
   * Payment can be made via credit card, debit card, or UPI.
   * Upon successful payment, an e-ticket is generated with a unique QR code.
4. **Ticket Cancellation**
   * The user can cancel the booking, receiving 50% cashback, excluding taxes.
5. **Ticket Validation**
   * Upon arriving at the theatre, the user presents their e-ticket, which is scanned and validated using a QR code.
6. **Admin Functions**
   * Administrators can add or remove shows, reject user registrations, and manage event data.

|  |
| --- |
| **7. Preliminary UML Diagrams** |

**Use Case Diagram**

**Use Case Scenarios**

**1. User Scenarios**

Scenario 1: User registers, logs in, and browses available movies.  
Scenario 2: User selects a movie, chooses seats, completes booking, and receives a QR-based e-ticket.  
Scenario 3: User cancels booking and receives a partial refund.

**2. Administrator Scenarios**

Scenario1: Admin adds a new movie schedule.

Scenario 2: Admin views monthly sales and downloads reports.  
**Actors**:

* **User**: Interacts with the system to perform tasks such as registration, searching movies, booking tickets, viewing bookings, and canceling tickets.
* **Admin**: Manages movies, showtimes, and user accounts, and views transaction reports.

**Use Cases**:

1. **User Use Cases**:
   * **Register/Login**: User registers a new account or logs in to the system.
   * **Search Movies**: User browses movies by title, genre, or showtime.
   * **Select Seats**: User selects available seats for a chosen movie.
   * **Book Ticket**: User completes the booking process, including payment.
   * **View E-Ticket**: User views or downloads the booked e-ticket.
   * **Cancel Ticket**: User cancels a booked ticket, possibly receiving a partial refund.
2. **Admin Use Cases**:
   * **Manage Movies**: Admin adds, updates, or removes movie listings.
   * **Manage Showtimes**: Admin manages scheduling of movie showtimes.
   * **View Reports**: Admin generates transaction and sales reports.

**Relationships**:

* **Association** (basic interaction) between **User** and actions like *Register/Login*, *Search Movies*, *Book Ticket*, and *Cancel Ticket*.
* **Association** between **Admin** and *Manage Movies*, *Manage Showtimes*, and *View Reports*.
* **Include** relationship within **Book Ticket**, which includes *Payment* as a part of the booking process.
* **Extend** relationship from **Cancel Ticket** to **Book Ticket**, as cancellation is only possible after booking.

**Class Diagram**

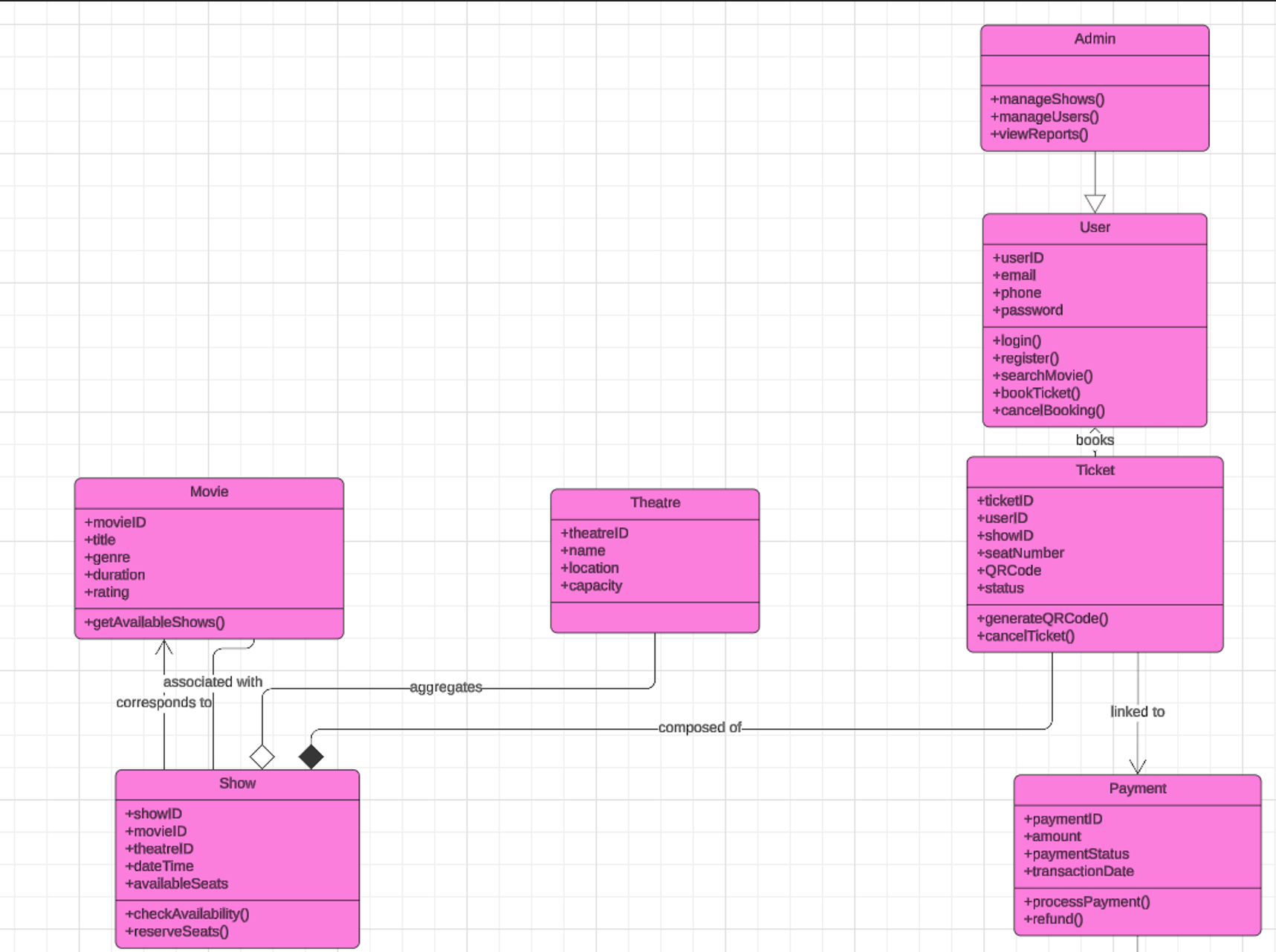
The Class Diagram will represent the key classes involved in the ETS with attributes and methods, showing relationships between classes.

**Classes:**

1. User
   * Attributes: userID, email, password
   * Methods: register(), login(), searchMovie(), bookTicket(), cancelTicket()
2. Admin (inherits from User)
   * Methods: addMovie(), updateShowtime(), viewReport()
3. Movie
   * Attributes: movieID, title, genre, duration, rating
   * Methods: getShowtimes()
4. Showtime
   * Attributes: showtimeID, movieID, theatre, dateTime, availableSeats
   * Methods: checkAvailability(), reserveSeats()
5. Ticket
   * Attributes: ticketID, userID, showtimeID, seatNumber, QRCode, status
   * Methods: generateQRCode(), cancelTicket()
6. Payment
   * Attributes: paymentID, amount, paymentStatus, transactionDate
   * Methods: processPayment(), refund()

**Relationships:**

1. Inheritance:
   * Admin inherits from User, meaning Admin is a specialized User with additional privileges.
2. Associations:
   * User associates with Ticket: A User can book one or more Tickets.
   * Admin associates with Movie and Showtime: Admin manages movies and showtimes.
   * User associates with Payment: Payments are linked to Users’ bookings.
   * Showtime aggregates Ticket: Each showtime consists of multiple tickets, but tickets depend on the showtime for validity.
   * Movie aggregates Showtime: A movie has multiple showtimes.
3. Composition:
   * Ticket composition with Showtime: A ticket is part of a specific showtime and cannot exist without it.
   * Payment composition with Ticket: Payments are essential for ticket bookings and are created alongside tickets.

****

**Sequence Diagram**

**Sequence Diagram for Movie Theatre E-Ticketing System**

**Actors:**

1. User: Performs actions like registration, login, movie search, ticket booking, and viewing bookings.

2. Admin: Inherits from User, with additional privileges for managing movies and generating reports.

3. System: Processes all requests from the User and Admin.

**Diagram Breakdown**

**1. User Registration and Login:**

**- User -> System: Sends registration request with email and password.**

**- System -> System: Validates email and password format, stores user data in the database.**

**- System -> User: Sends confirmation of successful registration.**

**- User -> System: Sends login request with email and password.**

**- System -> System: Authenticates user credentials.**

**- System -> User: Confirms successful login and displays user dashboard.**

**2. Movie Search:**

**- User -> System: Sends search request for movies based on criteria (e.g., title, genre).**

**- System -> System: Queries movie database and retrieves matching results.**

**- System -> User: Displays search results.**

**3. Ticket Booking:**

**- User -> System: Selects movie, showtime, and seats.**

**- System -> System: Temporarily reserves selected seats.**

**- User -> System: Proceeds with payment.**

**- System -> System: Processes payment via payment gateway.**

**- System -> System: Confirms payment and finalizes booking, generates e-ticket with QR code.**

**- System -> User: Sends booking confirmation and e-ticket details.**

**4. Ticket Cancellation:**

**- User -> System: Sends request to cancel ticket.**

**- System -> System: Validates cancellation policy (e.g., within allowed time).**

**- System -> System: Processes partial refund.**

**- System -> User: Confirms cancellation and refund status.**

**5. Admin Movie Management:**

**- Admin -> System: Sends request to add a new movie.**

**- System -> System: Validates movie details and adds to the database.**

**- System -> Admin: Confirms movie addition.**

**- Admin -> System: Requests update or deletion of an existing movie.**

**- System -> System: Validates request, updates or deletes movie data.**

**- System -> Admin: Confirms update or deletion.**

**6. Admin Report Generation:**

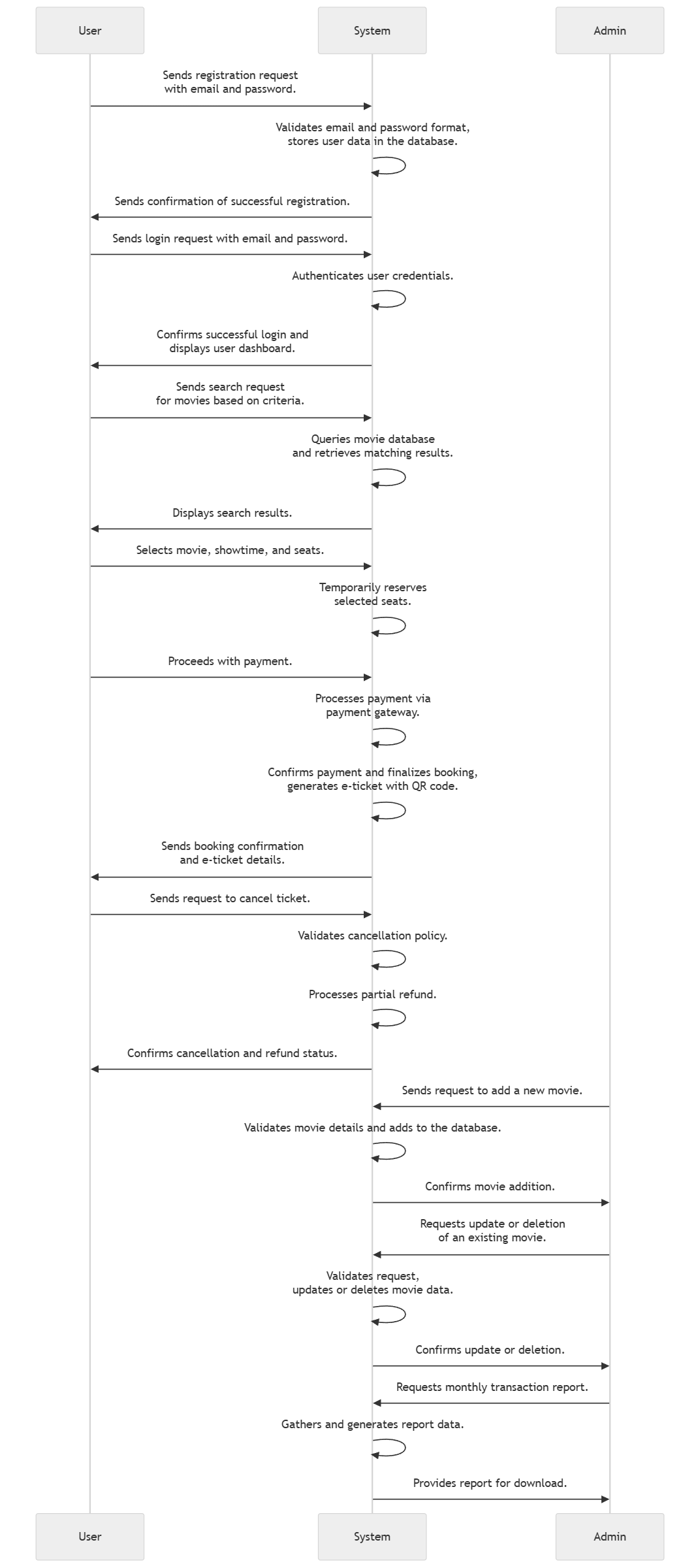
**- Admin -> System: Requests monthly transaction report.**

**- System -> System: Gathers and generates report data.**

**- System -> Admin: Provides report for download.**

**This sequence diagram demonstrates the primary flows for both User and Admin, showing their interactions with the System. The Admin role inherits from User and has additional privileges for managing movies and generating reports.**

**Visual (Conceptual):**



**Boundary Value Analysis  
1. User Registration and Login**

**Parameters to Test:**

* **Password Length**: Typically, passwords may have a minimum and maximum length.
* **Email Validation**: Email length and format can be checked as boundaries.
* **Maximum Login Attempts**: The system may allow a limited number of failed login attempts before locking an account.

**Boundary Conditions:**

* **Password Length**: Assume minimum length = 8 characters, maximum length = 20 characters.
  + Test cases:
    - Just below minimum: 7 characters (should fail).
    - At minimum: 8 characters (should pass).
    - Just above minimum: 9 characters (should pass).
    - Just below maximum: 19 characters (should pass).
    - At maximum: 20 characters (should pass).
    - Just above maximum: 21 characters (should fail).
* **Email Length**: Assuming a typical maximum email length of 64 characters.
  + Test cases:
    - Minimum: 1 character before "@" (should fail as it’s invalid).
    - Just below maximum: 63 characters (should pass).
    - At maximum: 64 characters (should pass).
    - Above maximum: 65 characters (should fail).
* **Login Attempts**: Assume max attempts = 5.
  + Test cases:
    - 4 attempts (should not lock the account).
    - 5 attempts (should lock the account).
    - 6 attempts (should remain locked).

**2. Ticket Booking and Seat Selection**

**Parameters to Test:**

* **Seat Selection**: Number of seats that a user can select for a booking.
* **Show Time Slots**: Available showtimes within operating hours.

**Boundary Conditions:**

* **Seat Selection**: Assume maximum seats per booking is 10.
  + Test cases:
    - Minimum: 0 seats (should fail as it’s invalid).
    - Just below maximum: 9 seats (should pass).
    - At maximum: 10 seats (should pass).
    - Just above maximum: 11 seats (should fail).
* **Show Time Slots**: Assume the system operates from 10:00 AM to 11:00 PM.
  + Test cases:
    - Just before opening: 9:59 AM (should fail).
    - At opening: 10:00 AM (should pass).
    - Midday slot: e.g., 3:00 PM (should pass).
    - Just before closing: 10:59 PM (should pass).
    - At closing: 11:00 PM (should pass).
    - Just after closing: 11:01 PM (should fail).

**3. Payment Processing**

**Parameters to Test:**

* **Transaction Amount**: Range of acceptable transaction amounts.
* **Card Expiry Date**: Validity of the card’s expiration date.

**Boundary Conditions:**

* **Transaction Amount**: Assume minimum transaction = $1, maximum = $500.
  + Test cases:
    - Just below minimum: $0 (should fail).
    - At minimum: $1 (should pass).
    - Just above minimum: $2 (should pass).
    - Just below maximum: $499 (should pass).
    - At maximum: $500 (should pass).
    - Just above maximum: $501 (should fail).
* **Card Expiry Date**: Assume transactions only allowed for future dates.
  + Test cases:
    - Expired date (last month): should fail.
    - Current month: should fail.
    - Next month: should pass.
    - Several years in the future: should pass.

**4. Ticket Cancellation and Refunds**

**Parameters to Test:**

* **Cancellation Window**: Maximum time before the show when cancellations are allowed.
* **Refund Percentage**: Boundary for the percentage refunded based on cancellation time.

**Boundary Conditions:**

* **Cancellation Window**: Assume cancellation allowed up to 2 hours before the show.
  + Test cases:
    - Just outside window: 1 hour 59 minutes before showtime (should fail).
    - At window limit: 2 hours before showtime (should pass).
    - Just inside window: 2 hours 1 minute before showtime (should pass).
* **Refund Percentage**: Assume 50% refund if canceled before the allowed time.
  + Test cases:
    - Cancellation within allowed time: Refund should be 50%.
    - Cancellation after allowed time: Refund should be 0%.

**5. User Profile and Password Management**

**Parameters to Test:**

* **Profile Update Fields**: Minimum and maximum allowed characters for fields like name or address.
* **Password Reset Token Validity**: Time limit for reset token validity.

**Boundary Conditions:**

* **Name Field**: Assuming minimum 1 character, maximum 50 characters.
  + Test cases:
    - Just below minimum: 0 characters (should fail).
    - At minimum: 1 character (should pass).
    - Just above minimum: 2 characters (should pass).
    - Just below maximum: 49 characters (should pass).
    - At maximum: 50 characters (should pass).
    - Just above maximum: 51 characters (should fail).
* **Password Reset Token Validity**: Assume tokens are valid for 15 minutes.
  + Test cases:
    - Just before expiration: 14 minutes 59 seconds (should pass).
    - At expiration: 15 minutes (should pass).
    - Just after expiration: 15 minutes 1 second (should fail).

**Summary Table for Boundary Value Test Cases**

| **Parameter** | **Boundary Value** | **Expected Outcome** |
| --- | --- | --- |
| **Password Length** | 7 characters | Fail |
|  | 8 characters | Pass |
|  | 20 characters | Pass |
|  | 21 characters | Fail |
| **Login Attempts** | 4 attempts | Pass |
|  | 5 attempts | Lock |
| **Seat Selection** | 0 seats | Fail |
|  | 10 seats | Pass |
|  | 11 seats | Fail |
| **Show Time** | 9:59 AM | Fail |
|  | 10:00 AM | Pass |
|  | 11:00 PM | Pass |
| **Transaction Amount** | $0 | Fail |
|  | $1 | Pass |
|  | $500 | Pass |
|  | $501 | Fail |
| **Cancellation Window** | 1 hr 59 mins before show | Fail |
|  | 2 hrs before show | Pass |
| **Refund Percentage** | Within allowed time | 50% refund |
|  | After allowed time | 0% refund |
| **Name Field** | 0 characters | Fail |
|  | 50 characters | Pass |
|  | 51 characters | Fail |
| **Password Reset Validity** | 14 mins 59 secs | Pass |
|  | 15 mins | Pass |
|  | 15 mins 1 sec | Fail |

**Requirements Based Testing:**

1. User Registration and Login Requirements

Requirement:

* Users must be able to register with an email and password.
* The system shall authenticate users securely on login.

Test Cases:

* TC1: Verify that a new user can register with a unique email and valid password.
  + *Expected Result*: User account is created, and a confirmation message is shown.
* TC2: Verify that the system rejects registration with an already registered email.
  + *Expected Result*: Registration fails with an error message.
* TC3: Verify that registered users can log in with the correct credentials.
  + *Expected Result*: User is redirected to their dashboard upon successful login.
* TC4: Verify that the system denies access on entering incorrect login credentials.
  + *Expected Result*: User is shown an error message for incorrect credentials.
* TC5: Verify that after five unsuccessful login attempts, the account is temporarily locked.
  + *Expected Result*: System locks the account and displays a message for account recovery.

2. Movie Search and Booking Requirements

Requirement:

* Users should be able to search for movies by title, genre, or showtime.
* Users should be able to select and book seats.

Test Cases:

* TC6: Verify that users can search for movies by title.
  + *Expected Result*: Search results show relevant movies based on the title entered.
* TC7: Verify that users can search for movies by genre.
  + *Expected Result*: Search results filter movies based on selected genre.
* TC8: Verify that users can search for movies by available showtime.
  + *Expected Result*: Only movies showing at the specified time are displayed.
* TC9: Verify that users can select seats from the available options.
  + *Expected Result*: Selected seats are reserved temporarily until booking is completed.
* TC10: Verify that once payment is completed, the selected seats are booked.
  + *Expected Result*: Seat status changes to "Booked" and cannot be selected by other users.

3. Payment Processing Requirements

Requirement:

* Users must be able to make secure payments through integrated payment gateways.

Test Cases:

* TC11: Verify that users can proceed to payment after selecting seats.
  + *Expected Result*: Payment options are shown, and user can choose a preferred method.
* TC12: Verify that payment is successful with valid payment details.
  + *Expected Result*: Payment completes, and a success message with booking details is shown.
* TC13: Verify that payment fails with invalid card details.
  + *Expected Result*: Payment does not proceed, and user is shown an error message.
* TC14: Verify that users receive a booking confirmation email/SMS after successful payment.
  + *Expected Result*: Booking details are sent to the user’s email or phone.

4. Ticket Generation and Validation Requirements

Requirement:

* The system must generate an e-ticket with a QR code upon successful booking.
* Tickets must be validated at the venue using QR codes.

Test Cases:

* TC15: Verify that an e-ticket with a unique QR code is generated after successful booking.
  + *Expected Result*: E-ticket with a QR code is available for download or sent via email.
* TC16: Verify that the QR code on the e-ticket is scannable and valid at the theatre.
  + *Expected Result*: QR code scanner recognizes the ticket as valid, and user is allowed entry.
* TC17: Verify that invalid or duplicate QR codes are rejected at the venue.
  + *Expected Result*: QR code scanner shows an error message for invalid or duplicate tickets.

5. Ticket Cancellation Requirements

Requirement:

* Users should be able to cancel a booking within a specified time frame and receive a partial refund.

Test Cases:

* TC18: Verify that users can cancel their ticket within the allowed time frame (e.g., 2 hours before showtime).
  + *Expected Result*: Ticket is canceled, and user receives a partial refund notification.
* TC19: Verify that ticket cancellations are not allowed within the restricted period (e.g., less than 2 hours before showtime).
  + *Expected Result*: System shows a message indicating that cancellation is not allowed.
* TC20: Verify that the partial refund amount matches the cancellation policy.
  + *Expected Result*: Refund amount is correctly calculated and issued to the user.

6. Administrative Requirements

Requirement:

* Administrators must be able to add, update, or delete movies and showtimes.
* Administrators should be able to view and generate reports.

Test Cases:

* TC21: Verify that the admin can add new movie listings.
  + *Expected Result*: New movie appears in the search results for users.
* TC22: Verify that the admin can update showtimes for an existing movie.
  + *Expected Result*: Updated showtimes are reflected in user searches.
* TC23: Verify that the admin can delete movies from the system.
  + *Expected Result*: Deleted movie is no longer available for booking.
* TC24: Verify that the admin can view and download sales and transaction reports.
  + *Expected Result*: Reports are generated accurately and available for download.

7. Non-Functional Requirements

Requirement:

* System must handle up to 1000 concurrent users with minimal latency.
* System must provide 99.9% uptime.
* Data must be encrypted to ensure security and comply with GDPR.

Test Cases:

* TC25: Perform load testing to ensure the system can handle 1000 concurrent users without performance issues.
  + *Expected Result*: Response times remain acceptable under maximum load.
* TC26: Verify that all sensitive data (e.g., passwords, payment information) is encrypted.
  + *Expected Result*: Data transmission is encrypted and secure.
* TC27: Monitor uptime over a period to ensure it meets the 99.9% requirement.
  + *Expected Result*: System downtime does not exceed acceptable limits.

**POSTIVE AND NEGATIVE TESTING   
1. User Registration and Login**

**Positive Testing:**

* **PT1: Register a new user with a valid email and password (e.g., "user@example.com" and "ValidPass123!").**
  + ***Expected Result*: Registration is successful, and a confirmation message is shown.**
* **PT2: Log in with a registered email and the correct password.**
  + ***Expected Result*: User logs in successfully and is redirected to their dashboard.**

**Negative Testing:**

* **NT1: Try registering with an already existing email.**
  + ***Expected Result*: Registration fails with an error message indicating the email is already in use.**
* **NT2: Attempt to register with an invalid email format (e.g., "userexample.com").**
  + ***Expected Result*: Registration fails, and an error message prompts the user to enter a valid email.**
* **NT3: Log in with a registered email but an incorrect password.**
  + ***Expected Result*: Login fails, displaying an error message about incorrect credentials.**
* **NT4: Enter incorrect login details five times to lock the account.**
  + ***Expected Result*: Account is temporarily locked, and the user is advised to recover their account.**

**2. Movie Search and Booking**

**Positive Testing:**

* **PT3: Search for movies by title, genre, and available showtimes.**
  + ***Expected Result*: The system displays relevant search results based on criteria.**
* **PT4: Select available seats, proceed to payment, and complete the booking.**
  + ***Expected Result*: Seats are booked, and a booking confirmation is sent.**

**Negative Testing:**

* **NT5: Search with invalid or misspelled movie title (e.g., "XYZ" or "Unknwon").**
  + ***Expected Result*: No results found message, prompting the user to adjust the search.**
* **NT6: Attempt to select more seats than available.**
  + ***Expected Result*: System prevents overbooking and notifies the user that selected seats are unavailable.**
* **NT7: Try to book without selecting any seats.**
  + ***Expected Result*: System blocks the booking process and displays an error prompting seat selection.**

**3. Payment Processing**

**Positive Testing:**

* **PT5: Complete a payment with valid credit card details.**
  + ***Expected Result*: Payment succeeds, and a success message with booking details is displayed.**
* **PT6: Complete a payment using a valid debit card or UPI.**
  + ***Expected Result*: Payment is successful, and a confirmation message is sent.**

**Negative Testing:**

* **NT8: Enter an invalid credit card number (e.g., incorrect format or non-existing card).**
  + ***Expected Result*: Payment fails, showing an error message that card details are invalid.**
* **NT9: Use an expired credit card.**
  + ***Expected Result*: Payment fails, and the system alerts the user about the expired card.**
* **NT10: Attempt payment with insufficient funds.**
  + ***Expected Result*: Payment fails, and the user is notified of insufficient balance.**

**4. Ticket Generation and Validation**

**Positive Testing:**

* **PT7: Successfully book a ticket and check for the e-ticket with a unique QR code.**
  + ***Expected Result*: E-ticket with QR code is generated and sent to the user’s email.**
* **PT8: Scan the QR code at the theatre.**
  + ***Expected Result*: QR code is validated, and the user is granted access.**

**Negative Testing:**

* **NT11: Present an invalid or duplicate QR code at the theatre.**
  + ***Expected Result*: QR code validation fails, and entry is denied with an error message.**
* **NT12: Attempt to scan a QR code for a different showtime or day.**
  + ***Expected Result*: QR code validation fails with a message indicating the ticket is invalid for the current show.**

**5. Ticket Cancellation**

**Positive Testing:**

* **PT9: Cancel a ticket within the allowed timeframe (e.g., 2 hours before showtime).**
  + ***Expected Result*: Cancellation is successful, and a partial refund is issued.**
* **PT10: Confirm that refund calculation aligns with cancellation policy.**
  + ***Expected Result*: Refund matches the stated percentage, and confirmation is provided to the user.**

**Negative Testing:**

* **NT13: Attempt to cancel a ticket within 1 hour of the showtime (outside of the allowed timeframe).**
  + ***Expected Result*: Cancellation fails, and the user is notified that it’s too late to cancel.**
* **NT14: Try to cancel a non-existent or already-canceled ticket.**
  + ***Expected Result*: System prevents duplicate cancellation, and an error message is shown.**

**6. Administrative Functions**

**Positive Testing:**

* **PT11: Admin adds a new movie with valid details.**
  + ***Expected Result*: Movie is successfully added and visible to users.**
* **PT12: Admin generates a transaction report for the current month.**
  + ***Expected Result*: Report is generated with accurate data, ready for download.**

**Negative Testing:**

* **NT15: Admin attempts to add a movie with missing or invalid information.**
  + ***Expected Result*: System blocks addition, showing an error message about required fields.**
* **NT16: Admin tries to delete a movie that has ongoing bookings.**
  + ***Expected Result*: Deletion is prevented, with a message indicating that existing bookings prevent deletion.**

**Summary of Positive and Negative Testing**

| **Functionality** | **Positive Testing Scenario** | **Expected Result** |
| --- | --- | --- |
| **User Registration** | **Register with valid email and password** | **Account is created successfully** |
| **User Login** | **Login with correct credentials** | **User is logged in** |
| **Movie Search** | **Search by valid title or genre** | **Relevant results are displayed** |
| **Booking** | **Select seats and complete payment** | **Booking is successful** |
| **Payment** | **Complete payment with valid card** | **Payment is processed and confirmed** |
| **Ticket Generation** | **Generate e-ticket after booking** | **E-ticket with QR code is sent to the user** |
| **Ticket Validation** | **Scan valid QR code** | **Ticket is validated, and entry is granted** |
| **Ticket Cancellation** | **Cancel ticket within allowed timeframe** | **Ticket is canceled, and a partial refund is issued** |
| **Admin Management** | **Add movie with valid details** | **Movie is added and visible to users** |

| **Functionality** | **Negative Testing Scenario** | **Expected Result** |
| --- | --- | --- |
| **User Registration** | **Register with invalid email format** | **Registration fails, error message is shown** |
| **User Login** | **Login with incorrect password** | **Login fails, showing an error message** |
| **Movie Search** | **Search by misspelled title** | **No results found message** |
| **Booking** | **Attempt booking without seat selection** | **Booking blocked, error message shown** |
| **Payment** | **Use expired card for payment** | **Payment fails, expired card message displayed** |
| **Ticket Validation** | **Scan invalid QR code** | **Validation fails, entry is denied** |
| **Ticket Cancellation** | **Cancel ticket within restricted timeframe** | **Cancellation blocked, error message shown** |
| **Admin Management** | **Attempt to delete movie with active bookings** | **Deletion blocked, error message shown** |

**This combination of positive and negative testing helps to ensure the ETS behaves correctly under normal conditions and handles edge cases and error conditions gracefully, resulting in a more robust and user-friendly system.**