E-Ticketing System

1. Chodavarapu Srinidhi - 22BD1A6712

2. Darsh Agrawal - 22BD1A6714

3. Mohammed Areeb Akhter - 22BD1A6737

4. Pasham Ayush Reddy - 22BD1A6744

Problem Statement:

In the rapidly digitizing E-ticketing industry, users increasingly demand a seamless and efficient experience when booking tickets for events such as concerts and marriages. However, the current fragmented experience across different platforms often results in frustration due to inconsistent ticket tracking, missing payment receipts, and lack of real-time status updates. Our E-Ticketing application, accessible via a server and a webpage, is designed to address these challenges by offering a unified platform that enables users to book tickets effortlessly.

The application aims to provide a cohesive and secure experience for users, allowing them to browse and book tickets for events, make payments through an integrated payment gateway, and receive receipts immediately, which they can download and store for future reference. Additionally, users will be able to track the status of their tickets in real-time, ensuring they remain informed and updated throughout the booking process. Notifications regarding event status, such as confirmations or cancellations, will further enhance user satisfaction.

By simplifying the interface and focusing on key functionalities, the E-ticketing system ensures a streamlined, user-friendly experience, transforming the ticketing process into a hassle-free and enjoyable journey for all users.

Software Requirement Specification

For

E-Ticketing System

Version 3.0

Prepared by:

1. Chodavarapu Srinidhi - 22BD1A6712

2. Darsh Agrawal - 22BD1A6714

3. Mohammed Areeb Akhter - 22BD1A6737

4. Pasham Ayush Reddy - 22BD1A6744

Keshav Memorial Institute of Technology

25-10-2024.

Table of Contents

Revision History

Name	Date	Reason for changes	Version
Week-1	26-09-24	SRS Draft Template	1.0
Week-2	19-10-24	SRS Documentation - Use case, Class Diagram	2.0
Week-3	25-10-24	SRS Final Document – Component, Sequence Diagram	3.0

1. Introduction

The Software Requirement Specification is designed to document and describe the agreement between the customer and developer regarding the specification of the software product requested. This documentation is done to provide a clear idea of customer requirements. This document can be used as reference in further development of the software system.

1.1 Purpose

The purpose of this document is to specify the software requirements for an E-ticketing system that enables users to browse and book tickets for events like concerts and marriages. It describes the system's functionality, expected behaviour, and constraints, providing a reference for developers, testers, and stakeholders involved in the project.

1.2 Document Convention

Heading:

Font-Size: 16, Font-Style: Bold, Font: Times New Roman

Subheading:

Font-Size: 14, Font-Style: Bold, Font: Times New Roman

Content:

Font-Size: 12, Font: Times New Roman.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers responsible for system implementation, providing them with clear specifications for building the E-Ticketing platform. It is also designed for testers who will validate the functionalities outlines in the requirements, ensuring that the system performs as expected. Managers can use this document for project tracking, helping them monitor progress and ensure that the system is developed according to plan. Additionally, it serves event organizers

using the platform, allowing them to understand the system's capabilities for managing events, ticketing, and user interactions.

1.4 Product Scope

The E-ticketing system focuses on creating a unified platform for users to browse, book, and manage tickets for events like concerts and marriages. The system will operate on a server and be accessible through a webpage. It will support an integrated payment system and real-time status updates. Users can track the event status and download tickets and receipts via the website.

1.5 References

https://www.studocu.com/in/document/university-college-of-engineering/software-engineering/e-ticketting-software-engineering-projects/65486732

https://www.studocu.com/in/document/university-college-of-engineering/software-engineering/e-ticketting-software-engineering-projects/65486732

2. Overall Description

2.1 Product Perspective

The E-ticketing system will function as a single-server platform, accessible via a webpage. It will provide a seamless ticketing experience for events like concerts and marriages, offering an efficient booking system with real-time tracking, secure payment, and receipt generation.

2.2 Product functions

2.2.1 Recruiter Portal

- **Job Posting Management**: Recruiters can post event tickets for concerts and marriages, update event details, and specify ticketing requirements.
- **Ticket Management**: Recruiters can manage ticket sales, view bookings, and update available slots based on event capacity.
- **Notification and Alerts**: Recruiters will receive real-time notifications when a user books a ticket, or when an event update occurs.
- **Attendee Database**: Recruiters can access a searchable database of attendees, categorized by booking details and ticket type.

2.2.2 Attendee Portal

- **Profile Management**: Attendees can create and update their profiles, including adding personal details and past event bookings. Filter and search for specific events
- **Event Search and Booking**: Attendees can search for concerts or marriage events based on various filters like location, date, and event type, and book tickets directly from the platform.
- **Booking Tracking**: Attendees can track the status of their ticket bookings in real time, including confirmations or cancellations.

- **Event Recommendations**: The system will suggest upcoming events based on the attendee's profile and booking history.

2.3 Operating Environment

- Web browsers (Chrome, Firefox, Safari, Edge) This E - Ticketing server shall operate in all famous browsers, for a model we are taking ,Internet Explorer Versions 7.0,8.0 and 9.0,with flash player 9 and JavaScript.

- This e-commerce website shall operate in all famous browsers, for a model we are taking, Internet Explorer Versions 7.0, 8.0 and 9.0, with flash player 9 and JavaScript.

2.4 User Characteristics

2.4.1 Event Attendees:

- The primary users who will use the system to browse and book tickets for concerts and marriages.

- Must have basic computer literacy, including the ability to navigate websites and complete online payments.

- Will access the system through web browsers on PCs or laptops with a stable internet connection.

- Responsibilities include searching for events, booking tickets, and managing bookings (e.g., checking event updates, downloading receipts).

2.4.2 Event Organizers:

- The primary users who will use the system to browse and book tickets for concerts and marriages.

- Must have basic computer literacy, including the ability to navigate websites and complete online payments.

- Will access the system via web browsers on PCs or laptops with a stable internet connection.

- Responsibilities include creating and managing event listings, tracking ticket sales and attendees, updating event details, and handling booking inquiries.

2.5 Design and Implementation Constraints:

- The system will be hosted on a server and accessed via modern web browsers, supporting devise like PCs and laptops with a stable internet connection.

- All transactions should be secured using SSL encryption.

2.6 Assumptions and Dependencies

- MERN stack (MongoDB, Express, React, Node.js) for the web application development.
- Python Django for server-side processing and APIs.
- JSON for data exchange between the server rand the client – side interface.
- Reliable internet access for both event organizers and attendees.
- Users being comfortable with basic web navigation and online payments.

3. External Requirements

3.1 User Interfaces:

- The system will provide a web-based interface accessible via modern web browsers such as Chrome, Firefox, and Edge.
- The user interface will be intuitive and responsive, allowing both event attendees and event organizers to navigate easily.
- Attendees will access event listings, booking forms, and payment processing through a simple, mobile - friendly layout.
- Organizers will have access to an administrative dashboard where they can manage event details, track bookings, and view attendee information.

3.2 Hardware Interfaces:

- The system does not require specific hardware interfaces; it will be accessed via standard PCs or laptops with stable internet connections.
- The server hosting the system will require standard hardware to support the MERN stack and Python Django server, ensuring optimal performance under expected traffic loads.
- The system may also be optimized for mobile-friendly access but does not require specialized mobile hardware.

3.3 Software Interfaces:

- The system will integrate with MongoDB for data management and storage
- It will use third-party payment gateways for processing online transactions securely.
- API integration will allow external systems (e.g., notifications, mailing services) to connect for additional functionalities.

- React will be used for building the client-side web application, while Express.js and Node.js will handle server-side operations.

3.4 Communications Interfaces:

- The system will use **HTTP/HTTPS** protocols for secure communication between users' browsers and the server.
- All communication between the client and server, including login, event browsing, and payment, will be encrypted using **SSL/TLS.**
- The system will support **real-time notifications** and updates via **Web Sockets** or similar technology for delivering booking confirmations and event status changes to users.

4. System Features:

These requirements include the development of search tools, sorting, filtering, navigation, and visual components of the platform, which can be maintained by event organizers.

4.1 Event Organizer/Management Staff:

Requirement ID : RI.01.01
Title : Event Management
Description : Control the entire database containing records of events, bookings, and attendee details. Any issues related to accessing or updating the database should be resolved as quickly as possible.
Priority : 1

Requirement ID : RI.01.02
Title : Event Created and Update
Description : Event organizers can create new event listings (concerts and marriages), update event details (date, time, venue), and set ticket availability.
Priority : 1

Requirement ID : RI.01.03
Title : View all bookings
Description : Event organizers must be able to view all ticket bookings, track attendee details, and manage ticket availability in real time.
Priority : 1

Requirement ID : RI.01.04
Title : Event Notifications
Description : Organizers can send notifications (event updates, cancellations) to attendees based on their bookings.
Priority : 2

4.2 Event Attendee:

Requirement ID : R1.02.01
Title : Attendee Registration
Description : New attendees should sign up by creating an account with valid login credentials to access the platform.
Priority : 1

Requirement ID : R1.02.02
Title : Attendee Login
Description : Attendees must use their registered credentials to log into the system and access available events.
Priority : 1

Requirement ID : R1.02.03
Title : View and Edit Personal Details
Description : Attendees must be able to view and edit their personal details (e.g., name, contact information) in their profile.
Priority : 2

Requirement ID : R1.02.04
Title : Browse and Book Events
Description : Attendees must be able to browse available concerts and marriages, apply filters, and book tickets directly through the platform.
Priority : 2

5. Other Non-Functional Requirements

5.1 Performance Requirements

- **Response Time:** Transactions such as loading applications, allotting interview and must complete within 2-5 seconds. Account details and applications should load within 1 second under normal load conditions.

- **Concurrent Users**: The system must handle up to 5000 concurrent users wit without significant degradation in performance. The system should scale to accommodate peak traffic, especially during working hours and end-of-month transactions.

- **Database Transactions:** Each transaction should commit to the database within 1 second, ensuring data consistency and atomicity (ACID compliance) to avoid transactional errors.

- **Backup Speed:** Automated backups of sensitive data should occur within off-peak hours and must not impact system availability. The system must recover within 15 minutes in the event of failure.

5.2 System Requirements

- **Data Loss Prevention**: In case of a system crash or unexpected shutdown, all pending transactions must be either rolled back or stored securely to prevent any loss. The system must log any discrepancies and alert the administrators immediately.

- **Physical Safety**: The system must ensure that physical access to critical server components is restricted to authorized personnel only. No sensitive operations should be allowed unless the user has the correct level of access.

- **Transaction Safety:** To prevent incorrect transactions, the system must perform thorough checks, including balance verification and approval workflows for high-value transactions. Any failed transaction should trigger automated rollback mechanisms and alert the customer.

- **Fraud Detection:** The system should have a built-in fraud detection mechanism to alert and block suspicious or unauthorized activities based on user behaviour analysis and transaction patterns.

5.3 Security Requirements

- **User Authentication**: All users, whether customer, bank staff, or admins, must authenticate via a secure two-factor authentication (2FA) system before accessing the platform. Passwords should adhere to the latest encryption standards and be stored using cryptographic hashing (e.g., SHA-256).

- **Data Encryption**: Sensitive data like passwords, transaction details, and personal information must be encrypted in transit and at rest using at least 256-bit AES encryption.

- **Access Control**: Different levels of access should be enforced:

    - **User**: Can only access personal details, applications and search for events.

    - **Organiser**: Can only put forward a request to add an event to the admin.

    - **Admin**: Can modify system-wide configurations, grant permissions, access logs.

5.4 Software Quality Attributes

- **Availability**: The system must be available 99.99% of the time, ensuring minimal downtime, particularly during critical financial operations like payroll processing.

- **Reliability**: Transactions must be processed reliably, without any duplication or data loss. Fail-safe mechanisms should prevent incomplete transactions.

- **Scalability**: The system must be able to scale horizontally, supporting additional users and accounts without performance degradation as the number of customers grows.

- **Maintainability**: The code base should be modular and easy to maintain, allowing for quick fixes and updates. Routine maintenance must be conducted without affecting system availability.

- **Portability**: The system should support deployment across various platforms and cloud services to allow for distributed operations and disaster recovery.

- **Interoperability**: The system must be able to integrate with third-party services like payment gateways (UPI, credit cards, internet banking) and notification services (SMS, email).

- **Usability**: The user interface must be intuitive and easy to use, ensuring a smooth banking experience for all customer demographics.

- **Testability**: All features must be easily testable with automated scripts, especially for security vulnerabilities, performance benchmarks, and functional testing.

5.5 Business Rules

- **Applicant Permissions**: Applicants can only access their own account details, manage their application, and apply for jobs. No applicant can access another applicant's account details.

- **Recruiter Permissions**: Recruiter can approve or reject applications, allot interview slots, and oversee all the applications. Permissions are role-specific.

- **Administrator Permissions**: Admins can override any system settings, including enabling or disabling recruitment services, modifying job details, and managing staff permissions.

- **Alerts & notifications**: The applicant must receive alerts as soon as the recruiter allots an interview slot and as well as the status of his application.

6. Other Requirements

Appendix A: Glossary

- **Attendee**: An individual looking to book tickets for events such as concerts and marriages via the E-ticketing platform. They can browse events, book tickets, and manage their bookings.
- **Event Organizer**: A person or organization using the E-ticketing system to create and manage event listings, track ticket sales, and send notifications to attendees.
- **Booking System**: The feature that allows attendees to search for events, select tickets, make payments, and receive confirmation receipts.
- **Ticket**: A digital entry pass purchased by an attendee for an event. The system generates tickets after successful booking and payment.
- **Event Management**: A process managed by event organizers to create, update, and oversee event details such as time, location, and ticket availability.
- **Encryption**: The method of securing sensitive data, such as user credentials and payment information, ensuring that all data transmitted and stored is encrypted to maintain security and privacy.
- **Payment Gateway**: A third-party service integrated into the platform to process payments securely and generate receipts for successful ticket purchases.
- **Real-Time Updates**: Instant updates provided by the system regarding event status, booking confirmations, or cancellations, ensuring that users receive timely notifications.
- **Responsive User Interface (UI)**: A design feature that adjusts the layout of the platform to various device screen sizes, providing a smooth user experience on both desktop and mobile browsers.

- **MongoDB**: A NoSQL database used to store and manage event and user data in the E-ticketing system.

- **User Interface (UI)**: The visual part of the platform through which attendees and organizers interact with the system, including navigation menus, event listings, and booking forms.

- **Availability**: The system's capability to be accessible at all times, allowing users to browse events, book tickets, or manage events 24/7.

- **Cross-Platform**: A feature ensuring the E-ticketing system operates smoothly across multiple devices, such as desktops, tablets, and smartphones, through web browsers.

- **Event Notifications**: Alerts provided to attendees by organizers about event changes, confirmations, or cancellations.

- **Scheduled Maintenance**: Periodic downtime planned for system updates and improvements to ensure the platform remains secure and performs optimally.

- **User Assistance**: An integrated help system or FAQ section that guides attendees and organizers through using the platform efficiently, ensuring ease of use.

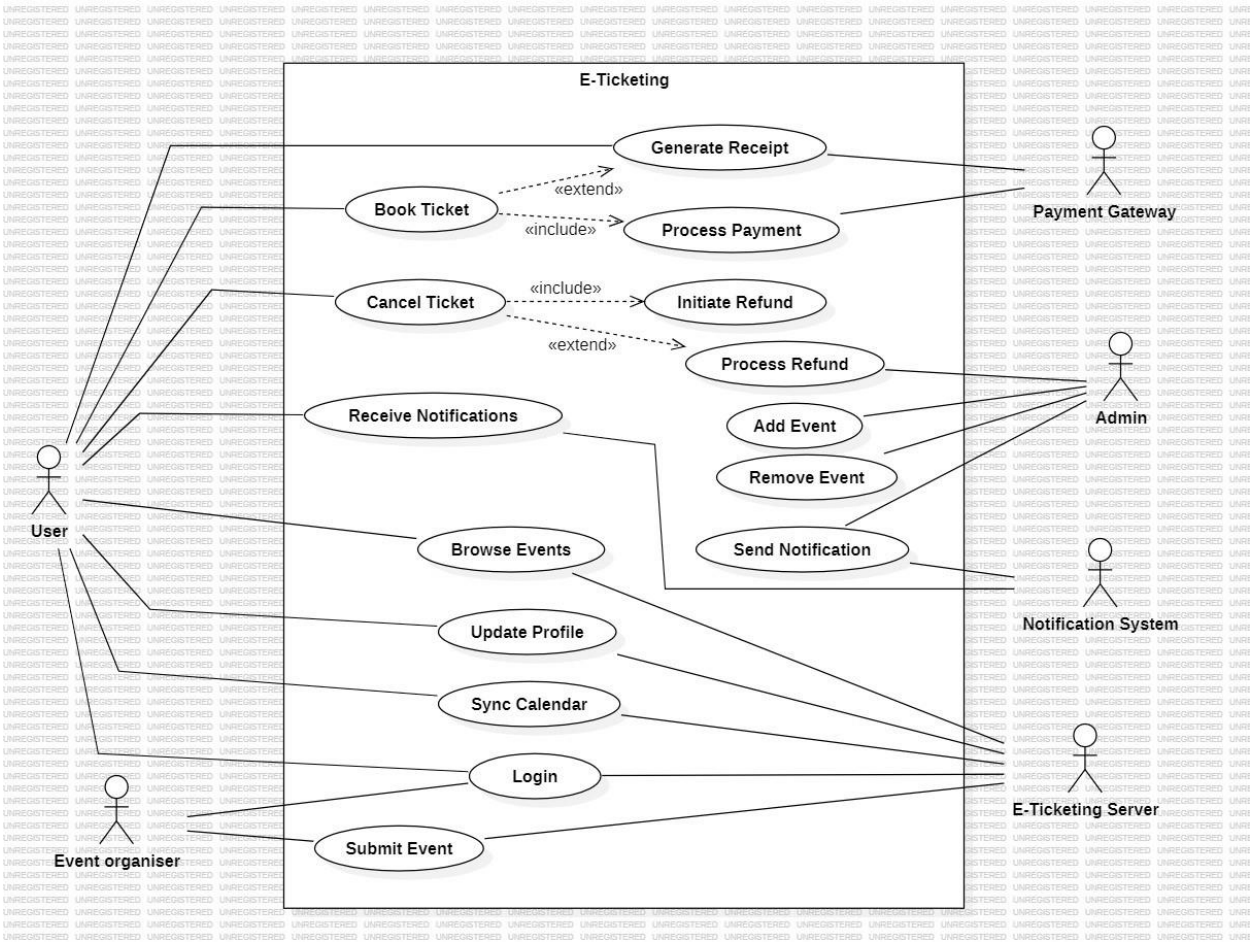
Appendix B: Analysis Models

Use Case Template:

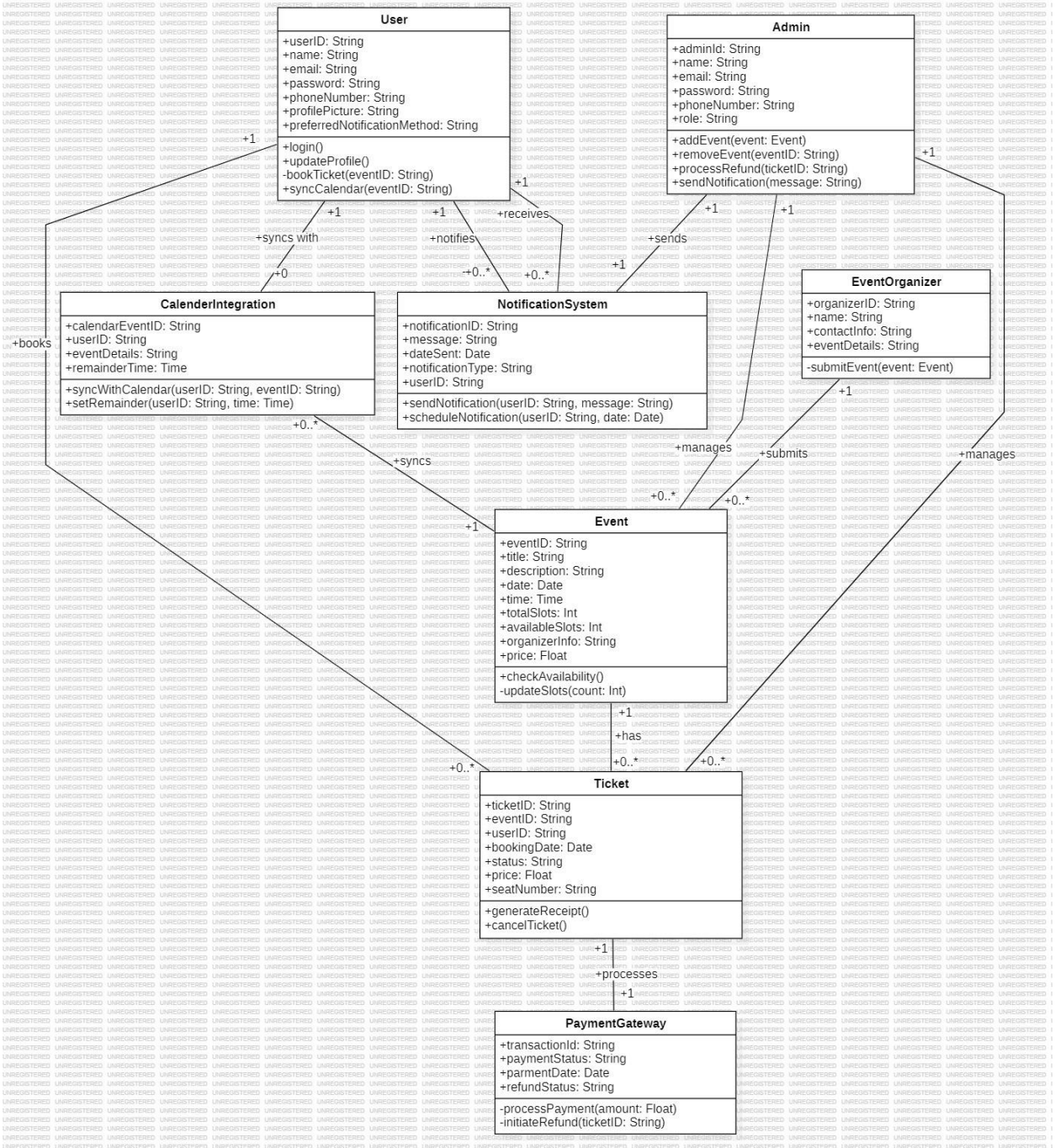
<table>
<tr><td colspan="2">Use Case ID:</td><td colspan="2">1234322</td></tr>
<tr><td colspan="2">Use Case Name:</td><td colspan="2">E-Ticketing</td></tr>
<tr><td colspan="2">End Objective:</td><td colspan="2">Make a E-ticketing platform for events</td></tr>
<tr><td rowspan="4">Created By:</td><td>1. Chodavarapu Srinidhi</td><td rowspan="4">On (date):</td><td rowspan="4">October 18, 2024</td></tr>
<tr><td>2. Darsh Agrawal</td></tr>
<tr><td>3. Areeb Akhtar</td></tr>
<tr><td>4. Pasham Ayush Reddy</td></tr>
<tr><td colspan="2">User / Actor</td><td colspan="2">User, Admin, Event organizer</td></tr>
<tr><td colspan="2">Trigger:</td><td colspan="2">User logging in the site to buy tickets for events</td></tr>
<tr><td colspan="4">Basic/Normal flows</td></tr>
<tr><td colspan="2">User Actions</td><td colspan="2">System Actions</td></tr>
<tr><td colspan="2">The user login to the site by entering details.</td><td colspan="2">Login page requests the user to provide a proper username and password.</td></tr>
<tr><td colspan="2">User views events present on the site.</td><td colspan="2">The home page suggests to the user some events or the user can search for the required event.</td></tr>
<tr><td colspan="2">Users make purchases of the tickets of the event with appropriate purchase options.</td><td colspan="2">Site will provide users with COD or other online payment methods.</td></tr>
<tr><td colspan="2">User wants to view and edit his/her own details.</td><td colspan="2">Site will provide the user to edit the changes</td></tr>
</table>

User completes the purchase and checkout.	Site processes the purchase and gives tickets.
Users can sync their calendar with the site.	Site syncs his calendar.
Users can receive notifications about the latest events.	Site notifies users with the latest events.
Users can cancel their tickets.	Admin processes the refund of the tickets
Exception Flows	
User Actions	**System Actions**
The user tries to login but doesn't have an account on the website.	The page requests the user to register an account in the registration page before committing the login.
The user tries to login by entering details.	The details entered are incorrect. Error message is displayed and the user needs to enter the correct details.
User tries to book tickets for a sold out event.	A “Sold out” message is displayed.
Payment by user fails.	A payment failed message is displayed.

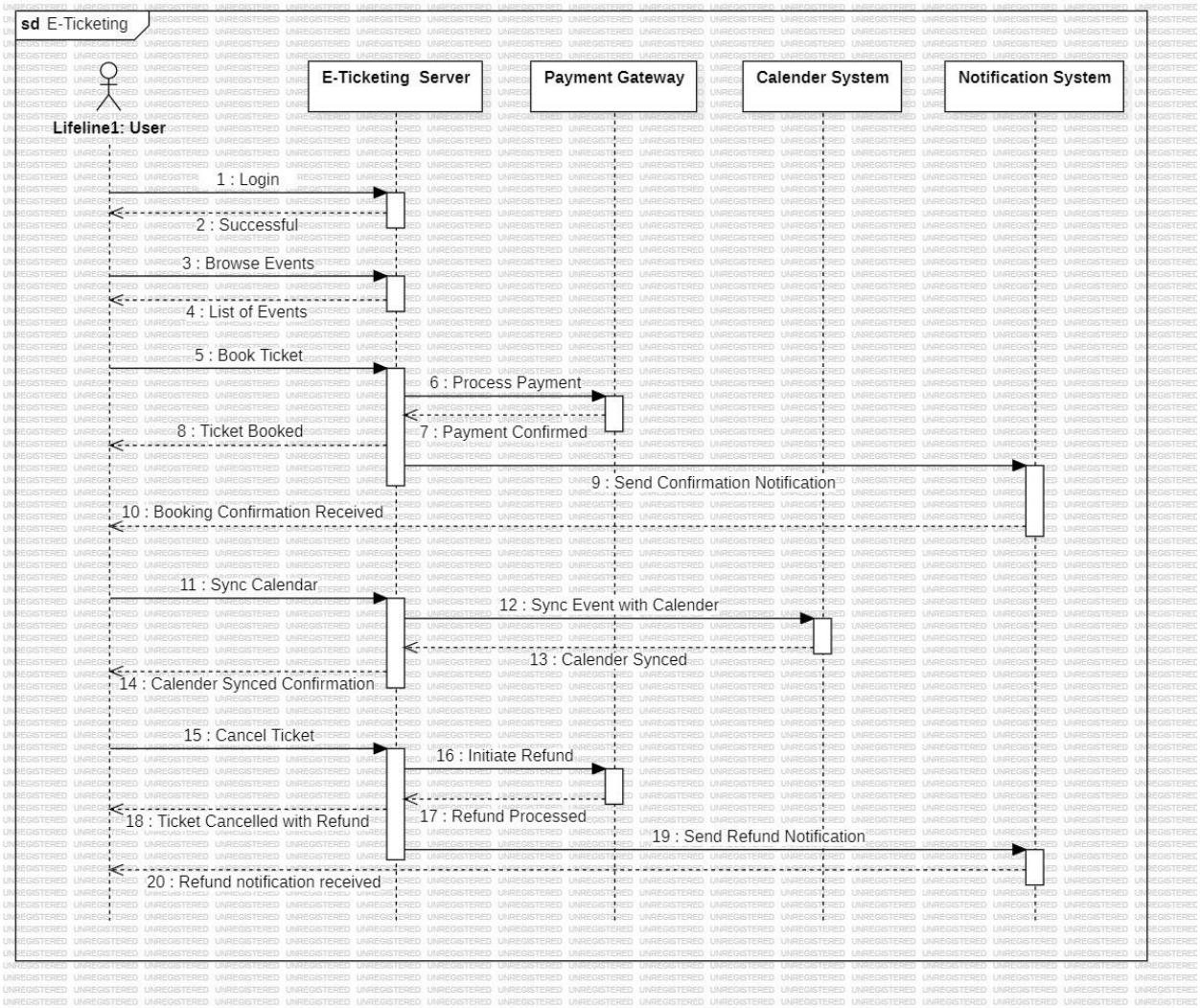
Use-Case Diagram:



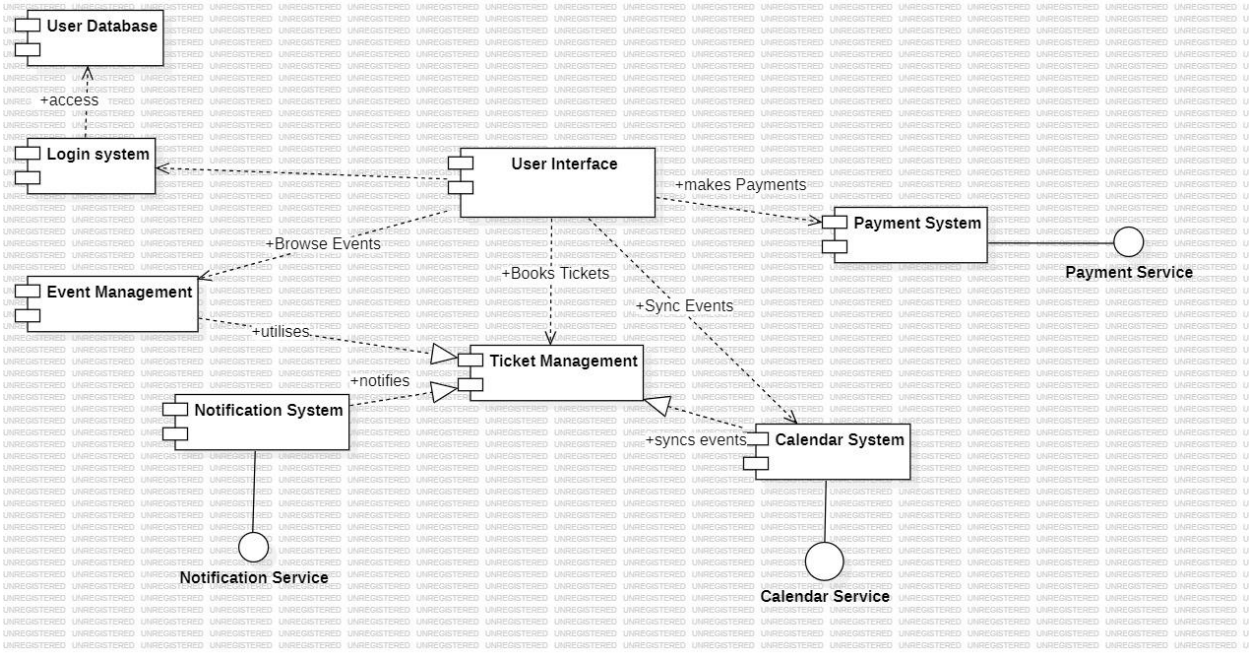
Class Diagram:



Sequence Diagram:



Component Diagram:



Appendix C: To Be Determined (TBD) List

This appendix lists items that need further clarification or finalization during the project's development. These items will be addressed as the project evolves, and final decisions will be made during subsequent stages.

- **Third-Party Payment Systems:** The exact payment gateways (e.g., PayPal, Stripe, Google Pay) to be integrated for processing payments securely and efficiently are still under evaluation.

- **Finalized User Roles and Permissions**: User roles and permission levels, especially for the **Admin** class, need further definition. This will include detailed permissions (e.g., VIP, standard, student discounts, managing users, and generating reports) and access control for sensitive actions.

- **Tickets Types and Categories**: Finalization of the specific ticket types (e.g., VIP, standard, student discounts) and categories (e.g., transportation, event types) the system will support.