

Software Requirements Specification For E-Ticketing

Version 3.0 approved

**Prepared by:**

1. Aditya - 22BD1A6602
2. Harsha - 22BD1A660J
3. Armaan - 22BD1A6611
4. Anand - 22BD1A6615
5. Varshith - 22BD1A6618

# 

# Keshav Memorial Institute of Technology

**11-10-2024**.

# Table of Contents

# 1. Introduction……………………………………………………………

# 1.1 Purpose………………………………………………………………………………………………………. 1

# 1.2 Document Conventions……………………………………………………………………………………… 1

# 1.3 Intended Audience and Reading Suggestions……………………………………………………………….. 2

# 1.4 Product Scope……………………………………………………………………………………………….. 2

# 1.5 References…………………………………………………………………………………………………… 2

# 2. Overall description…………………………………………………….

# 2.1 Product Perspective…………………………………………………………………………………………... 3

# 2.2 Product Functions…………………………………………………………………………………………….. 3

# 2.3 Operating Environment……………………………………………………………………………….……… 4

# 2.4 User Characteristics………………………………………………………………………………….……….. 4

# 2.5 Design and Implementation Constraints……………………………………………………………………… 4

# 2.6 User documentation………………………………………………………………………………………….. 5

# 2.7 Assumptions and Dependencies……………………………………………………………………………… 5

# 3. Specific Requirements………………………………………………...

# 3.1 User Interfaces………………………………………………………………………………………………... 7

# 3.2 Hardware Interfaces ………………………………………………………………………………………….. 7

# 3.3 Software Interfaces…………………………………………………………………………………………… 7

# 3.4 Communications Interfaces…………………………………………………………………………………... 8

# 4. System Features……………………………………………………….. .

# 4.1 Global database………………………………………………………………………………………………. 8

# 4.2 Login interface……………………………………………………………………………………………….. 8

# 4.3 Payment Processing………………………………………………………………………………………….. 9

# 4.4 Admin Management………………………………………………………………………………………….. 10

# 4.5 Notifications & Alerts………………………………………………………………………………………. 10

# 5. Nonfunctional Requirements………………………………………….

# 5.1 Performance Requirements ……………………………………………………………………………….… 11

# 5.2 Safety Requirements ………………………………………………………………………………………. 11

# 5.3 Security Requirements ……………………………………………………………………………………… 11

# 5.4 Software Quality Attributes ………………………………………………………………………………… 11

# 5.5 Business Rules ……………………………………………………………………………………………… 12

# 6. Other Requirements………………………………………………………..…

# 6.1 Regulatory Compliance……………………………………………………………………………………… 13

# 6.2 Data Backup & Recovery……………………………………………………………………………………. 13

# 6.3 User Accessibility……………………………………………………………………………………………. 13

# 6.4 Localization & Internationalization………………………………………………………………………….. 14

# 6.5 Scalability……………………………………………………………………………………………………. 14

# 6.6 Session Management ……………………………………………………………………………………….. 14

# Appendix A: Glossary…………………………………………………………………………………………………. 14

# Appendix B: Analysis Models………………………………………………………………………………………… 15

# Appendix C: To Be Determined List………………………………………………………………………………….. 24

# Revision History

| **Name** | **Date** | **Reason for changes** | **Version** |
| --- | --- | --- | --- |
| Week-1 | 11-10-24 | SRS Template | 1.0 |
| Week-2 | 17-10-24 | SRS ( DIAGRAMS ) | 2.0 |
| Week-3 | 25-10-2024 | SRS Final document | 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.

# Purpose:

This is the Software Requirements Specification (SRS) for our **E-TICKETING** platform. The project focuses on simplifying ticket booking for events, travel, and other activities through an online system. It will allow users to purchase tickets conveniently without the need to visit physical counters. We will also provide event organizers and service providers with the ability to upload and manage their events or transport schedules for sale. The platform aims to streamline the entire process between the ticket buyer and the provider, making the transaction quick and easy.

# 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.

## Intended Audience and Reading Suggestions:

This document serves as a prototype for the **E-TICKETING** platform. It is designed for various audiences with different purposes in mind. Developers can use this document to design and implement the project; managers can use it for planning, managing costs, and tracking timelines. Event organizers and service providers can use it to understand the system's features, while users can assess if the platform meets their needs and suggest any improvements. Testers can refer to this document for testing functionalities. The functional and non-functional requirements outlined here are also beneficial for developers during the development process.

## Product Scope:

Our project is similar to existing e-ticketing systems but introduces several new features and improvements. The platform will be available 24/7, with periodic maintenance scheduled once a month. The primary goal is to enhance user convenience by allowing them to book tickets from anywhere, saving significant time compared to traditional ticketing methods. Service providers won't need to maintain physical stock and can manage bookings digitally. Our system will provide users with ticket recommendations based on their preferences and needs. Users will also have the ability to rate event organizers or service providers based on the quality of the service, enabling better customer experience and feedback management. User data will be secured against breaches and unauthorized access. Regular surveys will be conducted to gather user feedback and continuously improve the platform. A robust database will be maintained for both users and service providers.

User details will be kept safe and secure from data theft. We conduct surveys on how we can make our site better and what changes can be made. We also maintain a database for different users.

## References:

References were taken from various ticketing systems and platforms, including industry standards and best practices from ticketing websites like Ticketmaster, Eventbrite, IRCTC, BookMyShow, and other similar platforms.

1. **Overall Description**

## Product Perspective

Our project is an evolution of existing **e-ticketing** systems. It will include key functionalities found in ticketing platforms, such as allowing event organizers or service providers to set up digital listings, customers to browse through available events or services, and administrators to manage event categories, accept or reject service providers, and maintain platform policies. The system will be **mobile-friendly** with highly customizable themes. Additionally, it incorporates several new advancements such as an **advanced search engine** for easy ticket search, **high data security** to protect user information from breaches, and a **user-friendly interface** that ensures a smooth experience for users of all types.

## Product functions

* + 1. **Administrator**
* Administrators will have the ability to insert, modify, and delete event listings or service offerings.
* Can accept or reject event organizers or service providers based on platform policies and compliance with payment methods.
* Can add, edit, and arrange events or services into categories for easier navigation.
* Will be able to notify users about special discounts, promotional offers, and relevant updates.
* Administrators can record canceled or refunded tickets based on user actions.

## Customers/Users

* Users will receive up-to-date information about new events or services, as well as ongoing discounts or special offers.
* They will be able to manage and modify their account details, including payment information and preferences.
* Users can search for tickets by event, category, location, or any relevant search criteria to find exactly what they’re looking for.
* They will have the ability to cancel or modify their bookings according to the platform's terms and conditions.
* Users can suggest events or services to be added to the platform, and provide reviews or feedback on the event or service providers.

## Operating Environment

* This e-ticketing platform will function seamlessly across all major web browsers, including but not limited to Firefox, Chrome, Safari, and Edge. For a model reference, it is compatible with modern versions of these browsers with JavaScript and HTML5 capabilities.
* The platform will be accessible on Windows (7, 8, 10, 11), macOS, and Linux operating systems.
* Minimum system requirements include a Pentium 4 processor or higher.
* The platform requires at least 1GB of RAM or higher for optimal performance.

## User Characteristics

The users of this e-ticketing platform include customers, event organizers, and administrators who maintain the system.

* **Customers** are individuals booking tickets for events, transport, or other services. They are assumed to have a basic understanding of computers and internet browsing.
* **Event organizers** or service providers are responsible for uploading event details and managing ticket sales on the platform. They should have the necessary skills to manage their listings and handle bookings.
* **Administrators** are responsible for maintaining the overall system, ensuring smooth operations, and resolving any issues that arise. They should have a deeper understanding of the internal modules of the platform and be capable of troubleshooting problems as needed.

## Design and Implementation Constraints

* **Stable Internet Connection**: Users must have a device with a stable internet connection and internet browsing capabilities to access the e-ticketing platform.
* **Database Access**: The information about users, ticket availability, and event or transport listings will be stored in a database, which the platform can access in real-time.
* **Availability**: The platform will be available **24/7** to ensure that users can book tickets at any time.
* **Platform Compatibility**: The software is designed to be cross-platform and can run on any operating system, including mobile devices.
* **User Authentication**: Users must log in with the correct username and password to access their accounts and perform actions such as booking or managing tickets.

**2.6 User Documentation**

The user documentation for the e-ticketing system will provide comprehensive guides for various users, ensuring they can effectively navigate and utilize the platform. The documentation will include:

* **User Manuals for Customers:** Detailed step-by-step guides explaining how to create an account, search for available tickets, filter based on destination or price, book tickets, make payments, and manage bookings (e.g., cancellations or refunds).
* **Administrator Manuals**: Guides for system administrators on how to manage user accounts, view and edit ticket listings, update destinations, oversee payment systems, and generate detailed reports on user activity and transactions.
* **Training Guides for Support Staff**: These will help customer support teams assist users with ticket booking issues, payment queries, and troubleshooting common problems.
* **Online Help and FAQs**: A well-organized help section available on the platform itself, providing quick access to common issues, FAQs, and troubleshooting steps for users to solve problems without additional assistance.

Additionally, video tutorials and interactive demos will be developed for key features of the system, such as booking tickets and using advanced search filters.

## Assumptions and Dependencies

The website requires following third party products.

* **MongoDB**: For storing user data, ticket availability, and booking details in a scalable NoSQL database.
* **Express.js**: To handle the server-side application logic and API requests.
* **React.js**: For building an interactive and dynamic user interface that offers a seamless user experience.
* **Node.js**: To power the backend server, manage API routes, and handle requests efficiently.

## **External Interface Requirements**

### **3.1 User Interfaces**

The e-ticketing system will feature a user-friendly, responsive interface designed for seamless navigation. Users will have access to an intuitive dashboard, allowing them to search for tickets by location, date, price, and category. The interface will support filters for narrowing down ticket options and real-time updates on ticket availability. The ticket booking process will be clear, with options to review and confirm bookings, view payment history, and manage user profiles.

### **3.2 Hardware Interfaces**

The system will be compatible with multiple devices, including Windows, Mac, and Linux systems, as well as mobile platforms (iOS and Android). It will require a minimum hardware specification of a 1.7 GHz processor and 2 GB of RAM to ensure smooth operation. It will also support standard peripherals like monitors, keyboards, and touch-screen devices.

### **3.3 Software Interfaces**

The system will be developed using **JavaScript**, **Node.js**, and **React** for frontend, with **MySQL** for database management. It will also integrate with third-party payment systems such as **PayPal**, **Stripe**, and **Google Pay**. The backend will be deployed on cloud platforms like **AWS** or **Heroku** and will support RESTful APIs for seamless communication between client and server.

### **3.4 Communication Interfaces**

The system will use **HTTP/HTTPS** for secure data communication, ensuring encryption through SSL certificates. Web services will allow users to interact with the system, including receiving email or SMS notifications for ticket confirmations or cancellations. The platform will also support integration with external ticketing services through APIs.

**4. System Features**

**4.1 User Registration and Login**

**4.1.1 Description and Priority**

This feature allows users to create accounts and log in using email and password or via social login (Google, Facebook). High priority as it is essential for user interaction with the system.

**4.1.2 Stimulus/Response Sequences**

* **Stimulus**: A user submits registration information.
* **Response**: The system creates a new user account and logs them in, displaying a confirmation message.

**4.1.3 Functional Requirements**

* **REQ-1**: The system must validate unique emails during registration.
* **REQ-2**: Users should be able to log in using social media credentials.
* **REQ-3**: Users must receive a confirmation email after registration.

**4.2 Ticket Booking and Management**

**4.2.1 Description and Priority**

This feature allows users to search for tickets, view availability, and book tickets for events or travel. High priority, as this is the core function of the system.

**4.2.2 Stimulus/Response Sequences**

* **Stimulus**: A user selects ticket preferences (e.g., location, date).
* **Response**: The system displays available tickets, updates based on filters, and confirms the booking once payment is processed.

**4.2.3 Functional Requirements**

* **REQ-4**: The system must display real-time availability for tickets.
* **REQ-5**: Users should be able to add multiple tickets to a cart for bulk purchase.
* **REQ-6**: The system must provide a booking confirmation after payment.

**4.3 Payment Processing**

**4.3.1 Description and Priority**

This feature handles payment processing through various gateways, ensuring secure transactions. High priority for ensuring successful ticket purchases.

**4.3.2 Stimulus/Response Sequences**

* **Stimulus**: A user proceeds to payment after selecting tickets.
* **Response**: The system processes the payment, confirms it, and updates the user’s account with the transaction.

**4.3.3 Functional Requirements**

* **REQ-7**: The system must support multiple payment methods (credit card, PayPal, Google Pay).
* **REQ-8**: Payment information must be encrypted for security.
* **REQ-9**: The system should notify the user of payment success or failure.

**4.4 Admin Management**

**4.4.1 Description and Priority**

Admins will manage events, users, payments, and ticket availability. This feature is critical to maintaining system integrity and ensuring correct ticketing information.

**4.4.2 Stimulus/Response Sequences**

* **Stimulus**: An admin logs in and updates an event’s ticket availability.
* **Response**: The system reflects the changes in real-time for all users.

**4.4.3 Functional Requirements**

* **REQ-10**: Admins must be able to add, edit, or delete events and ticket listings.
* **REQ-11**: The system must generate reports on ticket sales and user activity.
* **REQ-12**: Admins should be able to view and manage user accounts.

**4.5 Notifications and Alerts**

**4.5.1 Description and Priority**

This feature sends real-time notifications about booking status, cancellations, or payment issues. High priority for keeping users informed.

**4.5.2 Stimulus/Response Sequences**

* **Stimulus**: A user books a ticket or an event is cancelled.
* **Response**: The system sends a notification via email or SMS.

**4.5.3 Functional Requirements**

* **REQ-13**: The system must send booking confirmations to users.
* **REQ-14:** The system should notify users of cancellations or changes in ticket availability.
* **REQ-15**: The system must ensure notifications are sent in real time.

# 5. Nonfunctional Requirements

# 5.1 Performance Requirements

# The e-ticketing system must be optimized for high performance, capable of handling large volumes of ticket searches, transactions, and user queries efficiently. Data retrieval for ticket availability, payment history, or booking details must occur within 3 seconds under normal operating conditions. The system should support up to 5000 concurrent users without performance degradation, and search results for available tickets should be generated in less than 5 seconds. Additionally, real-time ticket availability updates should reflect changes (e.g., sold-out events) within 2 seconds.

# 5.2 Safety Requirements

# Safety is critical to prevent loss of transaction data or incorrect bookings. The system must include a reliable backup mechanism, with automated backups scheduled hourly. A rollback feature must be available to allow administrators to undo erroneous actions, such as mistaken cancellations or payments. The system should ensure consistency in ticket inventory to avoid overselling or booking errors during high-demand periods.

# 5.3 Security Requirements

# The e-ticketing system must prioritize security by implementing encryption for both data at rest and data in transit. Role-based access control will be enforced to limit access to sensitive data (e.g., payment information) to authorized users only. Payment transactions will be secured via SSL and tokenization to prevent unauthorized access or fraud. An audit trail will record all access and modifications to user accounts, tickets, and payment details for accountability.

# 5.4 Software Quality Attributes

# Reliability: The system must maintain 99.9% uptime, ensuring users can access ticketing services without interruptions.

# Scalability: The system must be designed to scale horizontally, accommodating increased traffic during peak booking periods.

# Maintainability: The system must have a modular architecture, allowing for easy updates, bug fixes, and the integration of new features.

# Usability: The interface must be intuitive, with minimal training required for both users and administrators.

# Compatibility: The system will integrate smoothly with third-party payment processors and ticket distribution services, using industry-standard protocols (e.g., REST APIs).

# 5.5 Business Rules

# The system must be flexible enough to handle various ticketing models (prepaid, postpaid) and discount schemes (bulk tickets, early bird discounts). It must comply with local accounting standards, providing accurate reports for tax purposes and ensuring ticket availability and pricing are properly updated in real time. Refund and cancellation policies must be automated, issuing refunds based on predefined rules.

**6. Other Requirements**

**6.1 Regulatory Compliance**

The system must comply with data privacy regulations, such as the **General Data Protection Regulation (GDPR)** and **California Consumer Privacy Act (CCPA)**, ensuring the secure handling of user data. It must also adhere to **Payment Card Industry Data Security Standard (PCI DSS)** for processing payment information securely.

**6.2 Data Backup and Recovery**

Automated daily backups must be performed to prevent data loss. A **disaster recovery** plan must ensure that the system can be restored with minimal downtime in case of failure, including data replication across multiple locations to guarantee ticketing services remain available even during emergencies.

**6.3 User Accessibility**

The system must comply with **Web Content Accessibility Guidelines (WCAG) 2.1** to ensure accessibility for users with disabilities. This includes support for screen readers, keyboard navigation, and alternative text for images, making the system usable for all.

**6.4 Localization and Internationalization**

The system will support **multiple languages and currencies**, enabling a global audience to use the platform. Localization features will include the adaptation of date formats, currency symbols, and local ticketing standards.

**6.5 Scalability**

The system must be designed with scalability in mind, ensuring that it can efficiently handle increased traffic during peak times, such as high-demand ticket releases. This includes scalable cloud infrastructure to manage server loads dynamically.

**6.6 Session Management**

The system must implement **secure session management**, including automatic session expiration after a period of inactivity and **multi-factor authentication (MFA)** for additional security. The system will ensure secure logout protocols, especially on shared or public devices.

**Appendix A: Glossary**

* **REST API**: A set of rules allowing systems to communicate and exchange data securely over the web.
* **SSL Encryption**: Secure Sockets Layer, a standard security protocol for establishing encrypted links between web servers and browsers.
* **Tokenization**: The process of substituting sensitive data with unique identification symbols that retain essential information without compromising security.

**Appendix B:**

Analysis Models (Included as appropriate, such as use case diagrams, data flow diagrams, etc.)

**Use case template**



**Use Case ID:** 0123456789

**Use Case Name:** E-Ticketing

**End Objective:** Automate and facilitate the whole process of shopping



**Created by:**









October 16, 2024

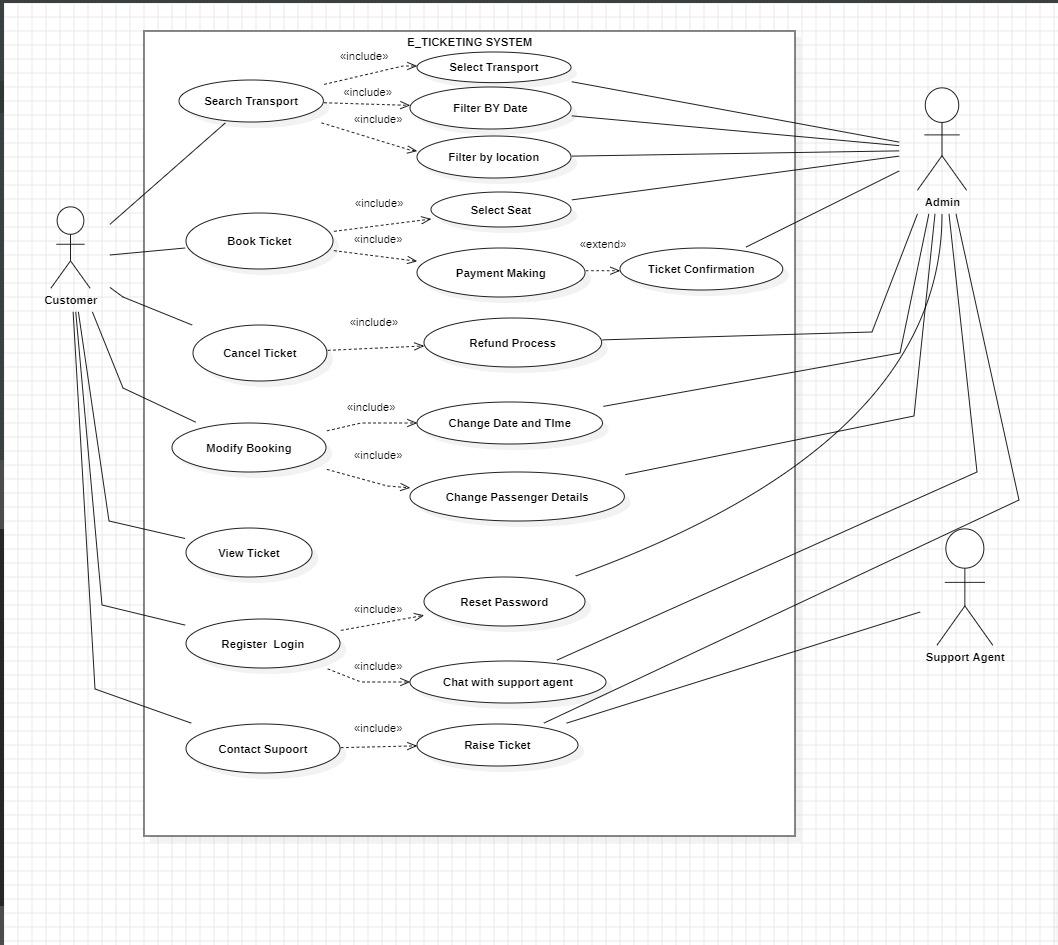


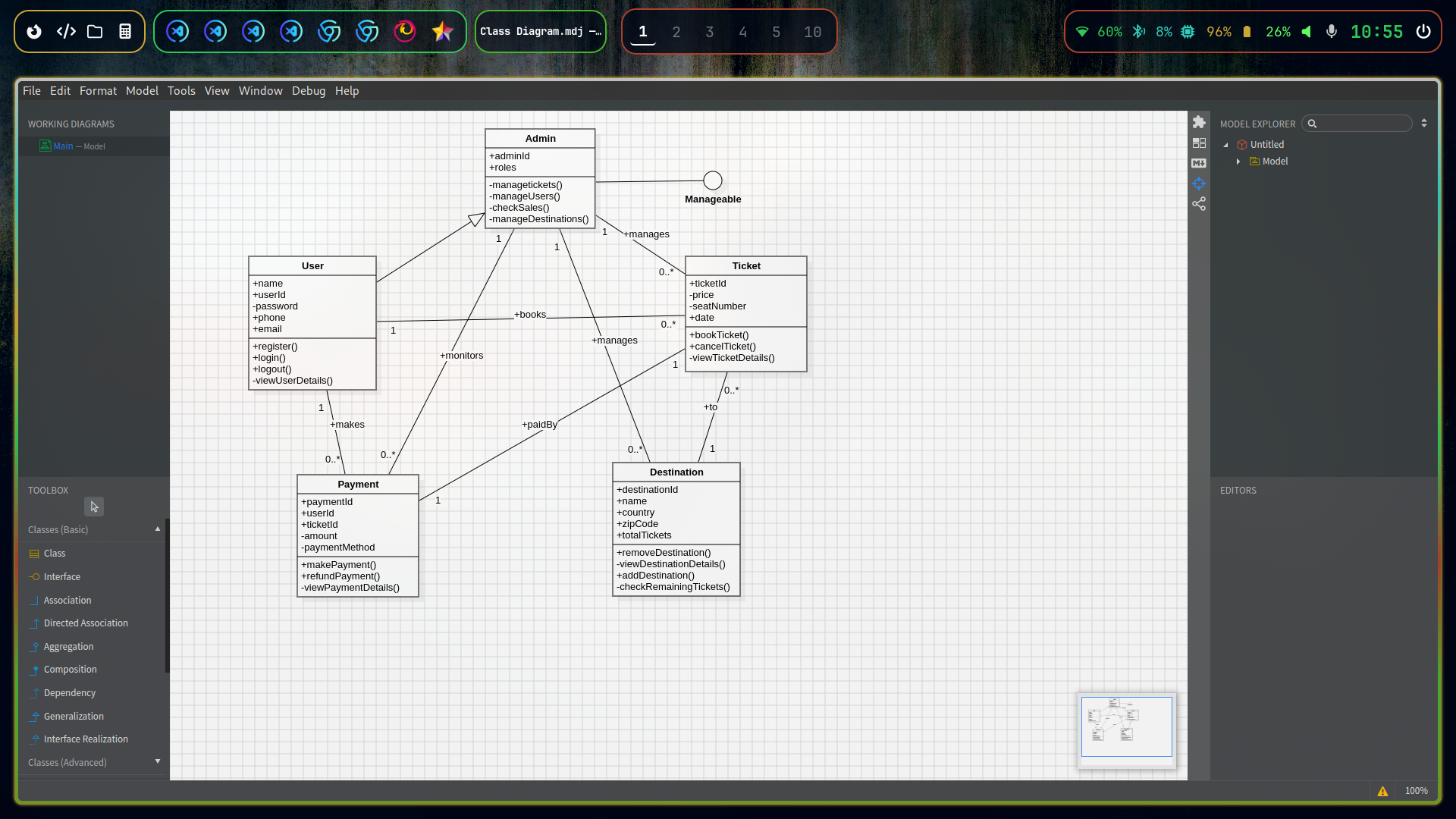


|  | |
| --- | --- |
| | The user tries to log in but doesn't have an account. | | --- |  |  | | --- | | | The page requests the user to register an account on the registration page before proceeding with the login. | | --- |  |  | | --- | |
| | The user tries to log in by entering details. | | --- |  |  | | --- | | | If the details entered are incorrect, the system displays a message: "Please check the username or password entered," prompting the user to re-enter the correct details. | | --- |  |  | | --- | |
| | The user tries to book a ticket. | | --- |  |  | | --- |   The user completes ticket booking and proceeds to payment. | | If the ticket is not available, the system displays a "Not available" message for that specific booking.  The system confirms the availability of the ticket and redirects the user to the payment page. | | --- |  |  | | --- | |

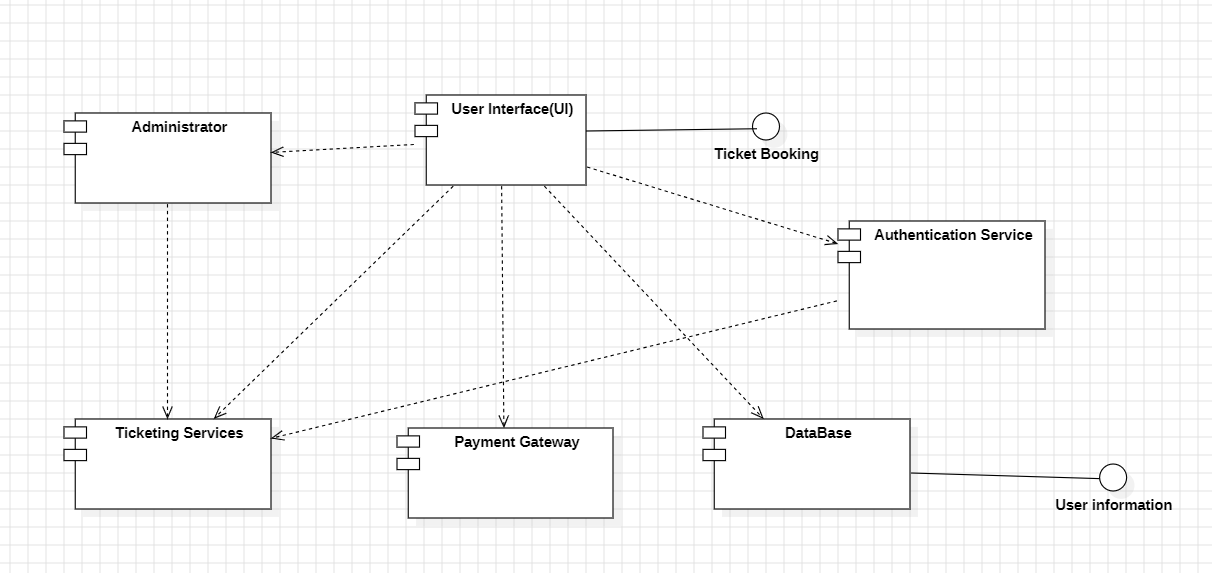
| **Exception Flows**  **User Actions System Actions** | |
| --- | --- |
| | The user tries to log in but doesn't have an account. | | --- |  |  | | --- | | | The system prompts the user to register an account on the registration page before proceeding with the login. | | --- |  |  | | --- | |
| | The user tries to log in by entering details. | | --- |  |  | | --- | | | If the login details entered are incorrect, the system displays a message: "Please check the username or password entered," prompting the user to re-enter the correct details. | | --- |  |  | | --- | |
| | The user tries to book a ticket. | | --- |  |  | | --- | | If the ticket is not available, the system displays a "Not available" message for that specific booking. |

**Use-Case Diagram**

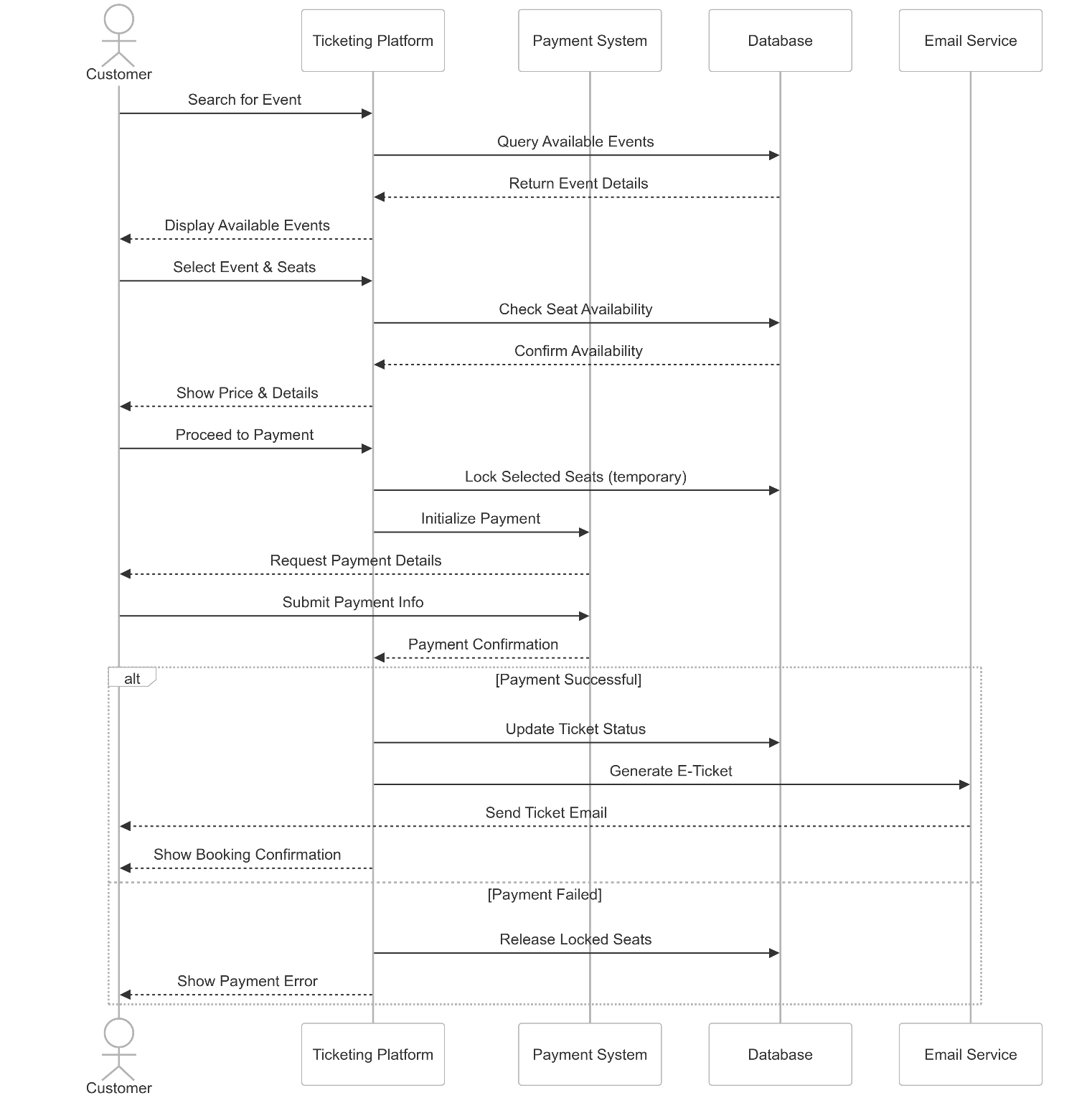
****

**Class Diagram**

**Component Diagram**

****

**Sequence 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.

1. **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.
2. **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., adding events, managing users, generating reports) and access control for sensitive actions.
3. **Ticket 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.