
Software Requirements Specification

for

Tours Management System

Prepared by Group-26

ACTS-CDAC Bengaluru

25/12/2023

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	3
1.1 Purpose.....	3
1.2 Intended Audience and Reading Suggestions	4
1.3 Product Scope	5
1.4 References.....	5
2. Overall Description.....	6
2.1 Product Perspective.....	6
2.2 Product Functions	6
2.3 User Classes and Characteristics	6
2.4 Operating Environment.....	7
2.5 Design and Implementation Constraints	7
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
2.8 Use Case Diagram.....	8
2.9 Data Flow Diagram	9
3. External Interface Requirements	9
3.1 User Interfaces	9
3.2 Hardware Interfaces	10
3.3 Software Interfaces	10
3.4 Communications Interfaces	12
4. System Features	13
4.1 System Feature 1	13
4.2 System Feature 2	13
4.3 System Feature 3.....	14
4.4 System Feature 4	16
5. Other Nonfunctional Requirements	17
5.1 Performance Requirements	17
5.2 Safety Requirements	17
5.3 Security Requirements	17
5.4 Software Quality Attributes	17
5.5 Business Rules	17
6. Other Requirements	18
Appendix A: Glossary.....	
Appendix B: Analysis Models	
Appendix C: To Be Determined List.....	

Revision History

Name	Date	Reason For Changes	Version

1 Introduction

1.1 Purpose

The purpose of the Tours Management System is to streamline operations related to sight-seeing and travel, catering to the widespread interest in traveling among people worldwide. It aims to clarify the system's functionalities, interfaces, operational constraints, and its responses to external stimuli. By offering a user-friendly interface, this system aims to provide a comprehensive solution that meets the needs of both stakeholders and developers involved in its creation and use.

1.2 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is tailored for a diverse audience involved in the development, management, and utilization of the Tour and Travel Management System.

1.2.1 Reading Sequence for Different Audience Types

1. For Developers:

- Start with the **Introduction (Section 1)** to understand the purpose and scope of the Tour Management System.
- Dive into **Overall Description (Section 2)** to get insights into the system's perspective, functions, and design constraints.
- Move to **External Interface Requirements (Section 3)** for details on user interfaces, hardware, software, and communication interfaces.
- Explore **System Features (Section 4)** to understand the functionalities to be implemented.

2. For Project Managers:

- Begin with the **Introduction (Section 1)** for an overview and understanding of the project's purpose.
- Delve into **Overall Description (Section 2)** to get a comprehensive view of the system's functions, user classes, and constraints.
- Focus on **External Interface Requirements (Section 3)** for insights into how users will interact with the system.
- Review **Other Nonfunctional Requirements (Section 5)** for performance, safety, and security considerations.

3. For Marketing Staff:

- Start with the **Introduction (Section 1)** to understand the purpose and scope of the Tour and Travel Management System.
- Review **Overall Description (Section 2)** for an overview of the system's functions, potential user classes, and constraints.
- Explore **System Features (Section 4)** to gain insights into the marketable aspects of the system.

4. For Users:

- Begin with the **Introduction (Section 1)** to understand the purpose and scope of the Tour and Travel Management System.

- Move to **External Interface Requirements (Section 3)** to get an understanding of how you will interact with the system.
 - Explore **System Features (Section 4)** to learn about the functionalities available to you.
5. **For Testers:**
- Start with the **Introduction (Section 1)** to understand the purpose and scope of the Tour and Travel Management System.
 - Dive into **Other Nonfunctional Requirements (Section 5)** for details on performance, safety, security, and software quality attributes.
 - Move to **System Features (Section 4)** to understand the features that need testing.
6. **For Documentation Writers:**
- Begin with the **Introduction (Section 1)** for an overview and understanding of the project's purpose.
 - Review the **Appendices (A, B, C)** for additional details that might be useful in creating user manuals and guides.

General Reading Sequence:

1. **Introduction (Section 1):** Provides an overview of the document.
2. **Overall Description (Section 2):** Offers a high-level view of the system.
3. **External Interface Requirements (Section 3):** Details how the system interfaces with users, hardware, software, and communications.
4. **System Features (Section 4):** Provides a breakdown of the system's features.
5. **Other Nonfunctional Requirements (Section 5):** Outlines performance, safety, security, and software quality considerations.
6. **Other Requirements (Section 6):** Addresses any additional requirements.
7. **Appendices (A, B, C):** Provide glossary, analysis models, and a list of to-be-determined items.

1.3 Product Scope

The Tour Management System project is an website aimed at efficiently managing travel-related activities. This platform empowers customers to explore, check availability, and book various travel destinations. The system covers essential features such as searching for tourist places, checking hotel room availability and prices, and selecting different travel packages.

1.3.1 Objectives and Features:

Destination Exploration: Users can search and explore various tourist destinations, facilitating informed travel decisions based on availability and preferences.

Package Reservations: Users can conveniently explore and book different travel packages, streamlining the reservation process for a hassle-free experience.

Online User Registration: The system supports online registration for users, ensuring a personalized experience and easy access to booking functionalities.

1.3.2 Business Benefits:

Increased Revenue: By providing an easy and accessible platform for customers to book travel services, the system is expected to contribute to increased revenue for the travel agency.

Improved Customer Satisfaction: The customizable tour packages and streamlined booking process aim to enhance customer satisfaction, fostering repeat business and positive word-of-mouth.

Operational Efficiency: Administrators will benefit from a centralized dashboard and reporting tools, leading to improved operational efficiency and strategic planning.

1.3.3 Alignment with Corporate Goals:

The Tour and Travel Management System aligns with the overarching corporate goals by:

Enhancing the company's competitiveness in the travel industry.

Contributing to revenue growth through increased bookings and customer satisfaction.

Streamlining internal operations, leading to improved efficiency and resource utilization.

1.4 References

1. Vision and Scope Document

- **Title:** Tour Management System Vision and Scope
- **Author:** Group 26 (ACTS-CDAC)
- **Version Number:** 1.0
- **Date:** 25/12/2023

2. IEEE-SWEBOK

- <https://www.computer.org/education/bodies-of-knowledge/software-engineering>

2. Overall Description

2.1 Product Perspective

The Tours and Travels Management System is a comprehensive software solution designed to streamline and automate various aspects of the travel management process. It acts as a centralized platform that facilitates the planning, booking, and management of travel-related services.

2.2 Product Functions

Booking Management:

- Facilitate the booking of accommodations, and transportation.
- Provide real-time availability and pricing information.

User Authentication and Authorization:

- Implement secure login mechanisms for customers, travel agents, and administrators.
- Define role-based access controls for different user classes.

2.3 User Classes and Characteristics

Booking Management:

- Facilitate the booking of accommodations, and transportation.
- Provide real-time availability and pricing information.

User Authentication and Authorization:

- Implement secure login mechanisms for customers, travel agents, and administrators.
- Define role-based access controls for different user classes.

2.4 Operating Environment

- The Tours Management System operates in a web-based environment, accessible through standard web browsers.
- It requires a reliable internet connection for real-time data updates and transactions.
- The system should be compatible with popular operating systems, including Windows, macOS, and Linux.

2.5 Design and Implementation Constraints

- The Tours Management System operates in a web-based environment, accessible through standard web browsers.
- It requires a reliable internet connection for real-time data updates and transactions.

- The system should be compatible with popular operating systems, including Windows, macOS, and Linux.

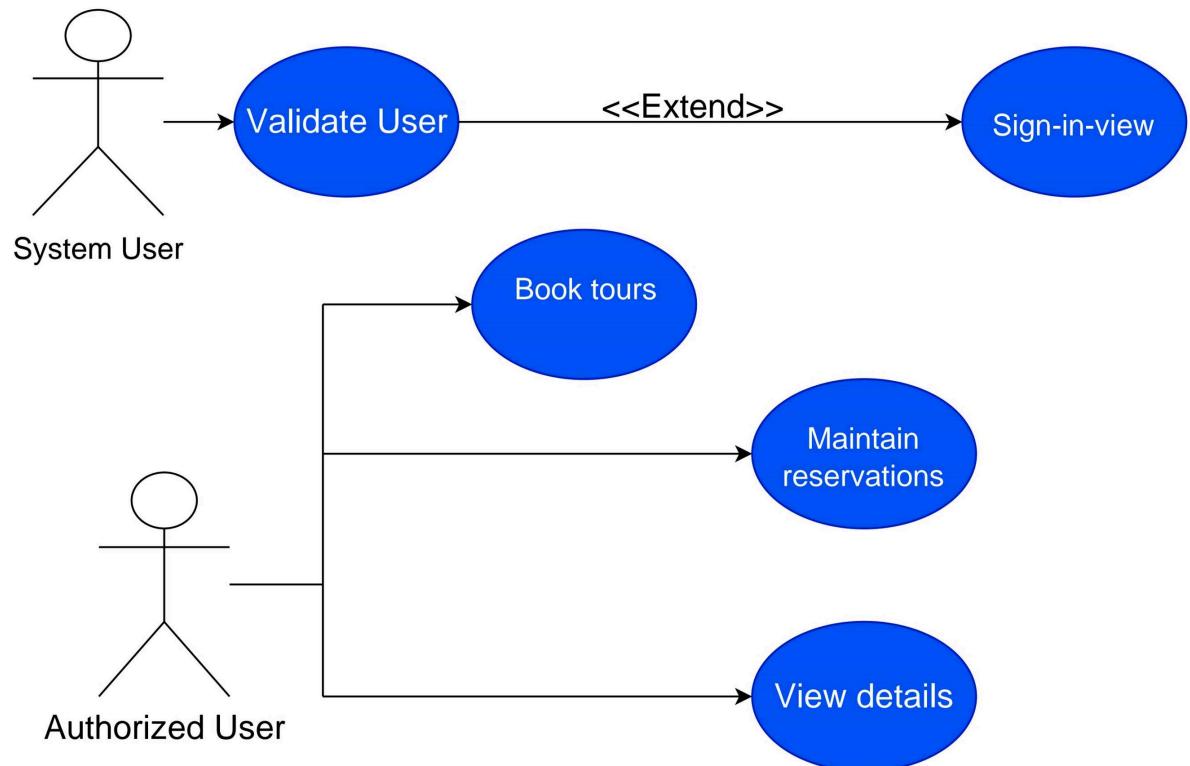
2.6 User Documentation

- The Tours Management System operates in a web-based environment, accessible through standard web browsers.
- It requires a reliable internet connection for real-time data updates and transactions.
- The system should be compatible with popular operating systems, including Windows, macOS, and Linux.

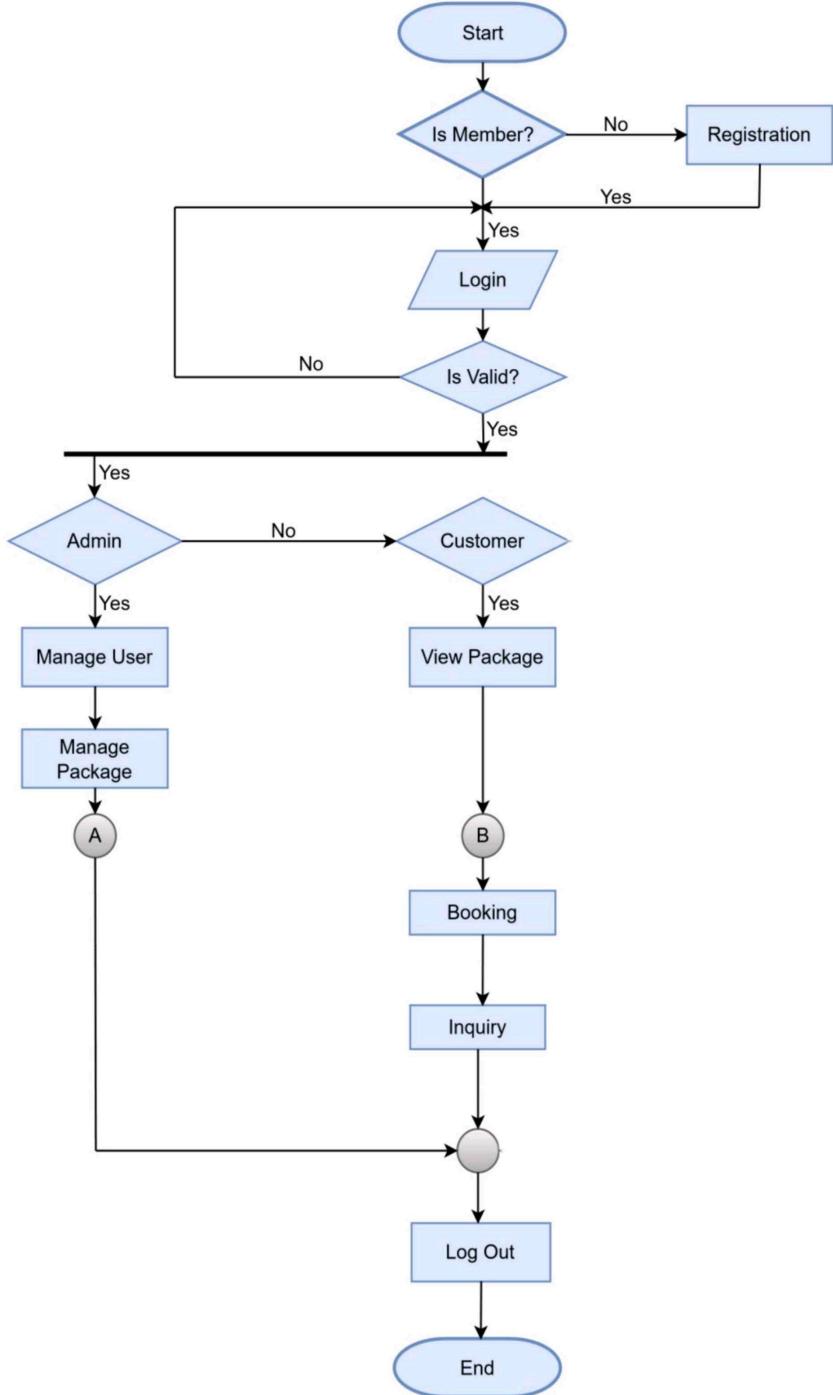
2.7 Assumptions and Dependencies

- The Tours Management System operates in a web-based environment, accessible through standard web browsers.
- It requires a reliable internet connection for real-time data updates and transactions.
- The system should be compatible with popular operating systems, including Windows, macOS, and Linux.

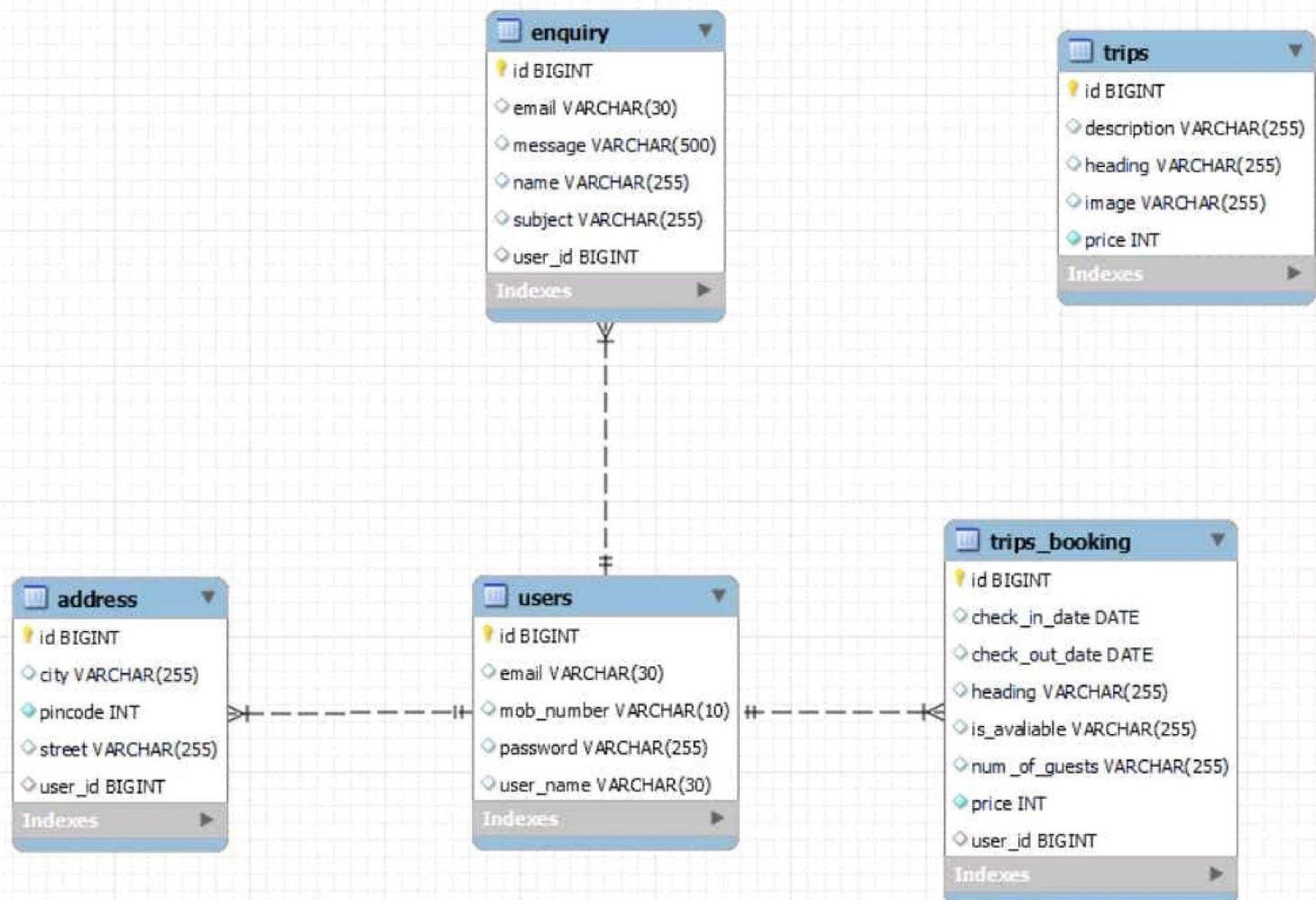
2.8 Use Case Diagram



2.9 Data Flow Diagram



2.10 ER Diagram



3. External Interface Requirements

3.1 User Interfaces

3.1.1 Customer Interface

Description: The customer interface provides users with a seamless experience for browsing and booking tour packages.

Characteristics:

- Intuitive design with easy navigation.
- Sample screen images attached (see UI specification document).
- Responsive layout for various devices.
- Consistent use of colours and branding.

Standard Buttons and Functions:

- Registration/Login
- Home
- Service
- Contact
- Book Now
- About

3.1.2 Admin Interface

- **Description:** The admin interface facilitates the management of tour packages, customer information, and system configurations.

• **Characteristics:**

- Accessible only to authorized personnel.
- Dashboard for quick insights.
- CRUD operations for tour packages and customer details.

• **Standard Buttons and Functions:**

- Add Package
- Edit Package
- Delete Package
- View Bookings
- View Users

3.2 Hardware Interfaces

3.2.1 Server

- **Description:** The system will be hosted on a web server.
- **Characteristics:**
 - Minimum server requirements:
 - Processor: Multi-core processor
 - RAM: 16 GB or higher
 - Storage: 100 GB SSD
 - Communication protocols: HTTPS.
 - Database server: [MySQL 8.0 or equivalent].

3.2.2 Client Devices

- **Description:** The system will be accessed through various client devices.
- **Characteristics:**
 - Supported devices: Desktop, tablet, smartphone.
 - Browser compatibility: Latest versions of Chrome, Firefox, Safari, and Edge.

3.3 Software Interfaces

3.3.1 Database

- **Description:** The system will interact with the database to store and retrieve data.
- **Software Components:**
 - Database Management System (DBMS): [MySQL 8.0].
- **Data Items:**
 - Customer information
 - Tour packages
 - Booking details

3.3.3 Application Frameworks

3.3.3.1 Backend (J2EE)

- **Description:** The backend of the Tours Management System will be developed using the SpringBoot framework.
- **Software Components:**

<ul style="list-style-type: none"> • Springboot Framework
<ul style="list-style-type: none"> • Services: <ul style="list-style-type: none"> • Business logic implementation. • Database interaction through Spring Data JPA.
<ul style="list-style-type: none"> • Communication Protocols: <ul style="list-style-type: none"> • HTTP for client-server communication.
<ul style="list-style-type: none"> • Data Sharing: <ul style="list-style-type: none"> • JSON format for data exchange.

3.3.3.2 Frontend (React.js)

<ul style="list-style-type: none"> • Description: The frontend of the system will be implemented using the React.js library.
<ul style="list-style-type: none"> • Software Components: <ul style="list-style-type: none"> • React.js • Redux for state management.
<ul style="list-style-type: none"> • User Interface: <ul style="list-style-type: none"> • Components for customer and admin interfaces. • Responsive design for various screen sizes.
<ul style="list-style-type: none"> • Communication Protocols: <ul style="list-style-type: none"> • RESTful APIs for backend communication.
<ul style="list-style-type: none"> • Data Sharing: <ul style="list-style-type: none"> • JSON format for data exchange.

3.3.4 Integrated Development Environment (IDE)

<ul style="list-style-type: none"> • Description: Development of the system will take place using integrated development environments.
<ul style="list-style-type: none"> • Software Components: <ul style="list-style-type: none"> • Eclipse for backend (SpringBoot) development. • Visual Studio Code for frontend (React.js) development.

3.4 Communications Interfaces.

3.4.1 Web Services

<ul style="list-style-type: none"> • Description: The system may utilize web services for real-time information retrieval.
<ul style="list-style-type: none"> • Communication Protocol: RESTful APIs.
<ul style="list-style-type: none"> • Security: API keys for authentication.

3.4.2 Data Transfer

- **Description:** Data transfer between the client and server.
- **Communication Protocol:** HTTP.
- **Security:** SSL/TLS encryption.

This section provides an overview of the external interfaces for the Tours Management System. For detailed design specifications, refer to the separate UI/UX design documents, API documentation, and other relevant specifications.

4. System Features

4.1 System Feature 1: User Authentication

4.1.1 Description and Priority

The User Authentication feature enables users to securely log in to the system, ensuring access control and data protection.

4.1.2 Stimulus/Response Sequences

1. User Login:

- User enters valid credentials.
- System validates credentials.
- If valid, system grants access.
- If invalid, system displays an error message.

4.1.3 Functional Requirements

REQ-1: User Authentication

- The system shall provide a login screen with fields for username and password.
- The system shall validate user credentials against stored data.
- If authentication fails, an error message shall be displayed.
- Successful authentication shall grant access to the user.

4.2 System Feature 2: Tour Search and Booking

4.2.1 Description and Priority: This feature allows users to search for available tours and book them.

4.2.2 Stimulus/Response Sequences:

- **Stimulus 1:** User selects a tour for booking.
 - **Response 1:** System provides details of the selected tour and prompts the user to confirm.
- **Stimulus 2:** User confirms the booking.
 - **Response 2:** System processes the booking, updates availability, and provides booking confirmation.

4.2.3 Functional Requirements:

REQ-1: Tour Details

- Users should be able to view detailed information about each tour, including itinerary, pricing, and available dates.
- High-quality images and descriptions should be provided for each tour.

REQ-2: Booking Process

- The system must support a user-friendly booking process.
- Users should be guided through steps, including selecting dates, providing participant information, and confirming the booking.

REQ-3: Pricing

- The system must accurately calculate and display the total cost of the selected tour.

REQ-4: Booking Confirmation

- Booking details should be stored in the user's account for future reference.

System Feature 3: Customer Panel

4.3.1 Description and Priority: The Customer Panel feature provides users with a to log in into account, view package, and access other customer-related functionalities.

4.3.2 Stimulus/Response Sequences:

- **Stimulus 1:** User logs in to their account.

- **Response 1:** System displays the customer dashboard with personalized information.
- **Stimulus 2:** User navigates to the service.
 - **Response 2:** System lists the package details.
- **Stimulus 3:** User Books package.
 - **Response 3:** System saves the changes and confirms the update.

4.4 System Feature 4: Admin Panel

4.4.1 Description and Priority: The Admin Panel feature provides authorized administrators with a centralized interface to manage and oversee system activities.

4.4.2 Stimulus/Response Sequences:

- **Stimulus 1:** Admin logs in to the admin panel.
 - **Response 1:** System displays the admin dashboard with relevant administrative tools.
- **Stimulus 2:** Admin updates product information.
 - **Response 2:** System saves the changes and reflects the updated information across the system.
- **Stimulus 3:** Admin reviews user.
 - **Response 3:** System presents a list of users.

4.4.3 Functional Requirements:

REQ-1: Admin Authentication

- The system must authenticate administrators securely before granting access to the admin panel.

REQ-2: Dashboard Overview

- The admin panel must provide a comprehensive dashboard displaying key system metrics, recent activities, and alerts.
- Dashboard components should be customizable based on admin preferences.

REQ-3: User Management

- Admins should be able to manage user accounts, including creation, modification, and suspension.
- The system must support role-based access control for different admin roles.

REQ-4: Product Management

- Admins should have the ability to add, edit, or remove products/services from the system.
- Changes to product information should be reflected across the platform.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Requirement: The system must be able to handle a minimum of 50 simultaneous user sessions. This requirement ensures that the system can maintain acceptable performance levels during periods of high demand, preventing degradation in user experience.

5.2 Safety Requirements

Requirement: The system must restrict access to sensitive user data to authorized personnel only, as defined by their roles and permissions. This requirement protects user privacy and prevents unauthorized access to sensitive information, ensuring compliance with data protection regulations.

5.3 Security Requirements

Requirement: The system must enforce password complexity rules, requiring users to create passwords with a minimum length of 8 characters, including at least one uppercase letter, one lowercase letter, one digit, and one special character. This requirement enhances the security of user accounts by ensuring that passwords are sufficiently complex and resistant to brute-force attacks.

5.4 Software Quality Attributes

Requirement: The system should be designed with high maintainability, allowing for easy updates and modifications without impacting the overall system stability.

Rationale: This requirement ensures that the system can adapt to changing requirements and evolve over time without incurring excessive maintenance costs.

5.5 Business Rules

Rule: Only users with the role of "Manager" can approve travel expense reimbursements exceeding 10000. This rule implies a functional requirement for expense reimbursement approval workflows, specifying the conditions under which approvals are required and by whom.

6. Other Requirements

Database Requirements: The system must use a relational database management system (RDBMS) to store and manage data. Specific requirements for database performance, scalability, and backup and recovery procedures should be outlined here.

Legal Requirements: The system must comply with all relevant laws and regulations related to the travel industry, including data protection laws (e.g., GDPR), consumer rights, and any industry-specific regulations.

Reuse Objectives: The project should aim to maximize code reuse and modular design to facilitate future enhancements and reduce development time for future projects.

Appendix A: Glossary

SRS (Software Requirements Specification): A document that describes the software system's intended purpose and functionality.

RDBMS (Relational Database Management System): A type of database management system that stores data in a structured format using rows and columns.