Software Requirements Specification

for

Ferry Ticketing System

Prepared by

Batch:B

Adwait Purao	**2021300101**	**adwait.purao@spit.ac.in**
Viraj Bhalerao	**2022301002**	**viraj.bhalerao22@spit.ac.in**

Instructor:	**Prasenjit Bhavthankar**
Course:	**Software Engineering**

Date:4/08/2023

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft 1.0	Adwait Purao, Viraj Bhalerao	Initial draft of the Ferry Ticketing System SRS document, including requirements gathering and analysis.	04/09/23

1 Introduction

Ferry operates passenger ferries (no vehicles) between the islands of Havelock and Andaman. This document outlines the high-level functionality that is desired for a new Ticketing System for the Ferry. The system will consist of a centralized database, along with several Ticketing terminals in various locations. Connectivity between terminals and the database server will be via the internet. We anticipate having a full time “live connection” at each location, and we do not anticipate the need for “batch” processing

1.1 Document Purpose

The system will consist of a centralized database, along with several Ticketing terminals in various locations. Connectivity between terminals and the database server will be via the internet. We anticipate having a full time “live connection” at each location, and we do not anticipate the need for “batch” processing.

1.2 Product Scope

The Ferry Ticketing System is a software application designed to streamline and enhance the process of booking and managing ferry tickets for passengers and operators. Its primary purpose is to provide a user-friendly platform for travelers to book ferry tickets, check schedules, and access real-time information about ferry routes and availability. Simultaneously, it offers ferry operators a comprehensive tool to efficiently manage reservations, track passenger data, and optimize their fleet operations.

The benefits of this system are multifaceted. For passengers, it offers the convenience of online booking, reducing the need to queue at ticket counters. They can easily compare schedules, choose preferred routes, and receive electronic tickets for a hassle-free boarding experience. Ferry operators benefit from improved operational efficiency as they can manage reservations, allocate seating, and monitor passenger loads more effectively. Additionally, the system provides valuable data insights, enabling operators to optimize their services based on historical booking trends and passenger preferences. Overall, the Ferry Ticketing System aims to simplify the ferry travel experience for passengers while helping operators run their services more efficiently and profitably.

Sell tickets online via secure Web Site
Send e-Tickets to passenger’s smart-phone (by email or text)

1.3 Intended Audience and Document Overview

The Intended Audience and Document Overview for the Ferry Ticketing System Software Requirements Specification (SRS) are as follows:

Clients and Stakeholders: This document is primarily intended for the clients or stakeholders who are funding, commissioning, or overseeing the development of the Ferry Ticketing System. It will provide them with a detailed understanding of the project's scope, requirements, and objectives.

Developers and Technical Team: Developers and the technical team involved in designing, building, and implementing the Ferry Ticketing System will find this document crucial. It outlines the technical specifications, system architecture, and detailed requirements necessary for system development.

Project Managers: Project managers can use this document to gain insights into the project's requirements, milestones, and dependencies. It will help them in planning and tracking the project's progress.

Marketing and Sales Staff: Marketing and sales staff can benefit from understanding the system's key features and benefits to effectively communicate and market the product to potential customers.

Users and Passengers: While not the primary audience for the SRS, users and passengers may refer to this document to understand the functionality of the Ferry Ticketing System and its benefits, especially if the system includes a user manual or help section.

Testers and Quality Assurance Teams: Testers can use this document to design test cases and ensure that the developed system meets the specified requirements.

Document Overview:

The rest of this SRS is organized as follows:

Introduction: Provides an overview of the document, its purpose, intended audience, and a brief description of the Ferry Ticketing System.

Scope: Describes the scope of the project, including the system's features and functionalities.

Functional Requirements: Details the functional requirements of the system, including user interactions, use cases, and system behavior.

Non-Functional Requirements: Outlines non-functional requirements, such as performance, security, and scalability.

System Architecture: Provides an overview of the system's architecture, including components and their interactions.

Data Requirements: Describes the data requirements, including data models and database specifications.

User Interface (UI) Design: If applicable, this section outlines the UI design guidelines and requirements.

Project Timeline: Offers a high-level project timeline, including milestones and deliverables.

Risks and Assumptions: Identifies potential risks and assumptions associated with the project.

Appendices: May include additional information, such as glossary, references, and supporting documents.

Sequence for Reading:

Readers are advised to start with the "Introduction" section to gain a broad understanding of the document's purpose. Then, the sequence can vary based on the reader's role:

Clients and Stakeholders: After the introduction, they may focus on the "Scope" and "Project Timeline" sections for an overview of what the system will accomplish and when it will be delivered.

Developers and Technical Team: They should delve into the "Functional Requirements," "Non-Functional Requirements," "System Architecture," and "Data Requirements" sections to understand the technical details and specifications of the system.

Project Managers: In addition to the sections for developers, project managers may pay close attention to the "Project Timeline" and "Risks and Assumptions" sections to ensure the project stays on track.

Marketing and Sales Staff: Their focus may be on the "Scope" and "User Interface (UI) Design" sections to understand the system's features and how they can be marketed.

Testers and Quality Assurance Teams: They should refer to the "Functional Requirements" section to design comprehensive test cases for validation.

Users and Passengers: Users may refer to the "Scope" and "User Interface (UI) Design" sections if available, as these sections provide insights into what the system offers and how to interact with it.

1.4 Definitions, Acronyms and Abbreviations

FMS: Ferry Management System

A software system designed to manage and streamline ferry operations, including ticketing, scheduling, passenger information, and vessel management.
API: Application Programming Interface

In the context of an FMS, an API could be used to integrate with other systems, such as online booking platforms or navigation systems.

DBMS: Database Management System

An FMS may use a DBMS to store and manage data related to ferry schedules, bookings, passenger information, and more.
Scheduling Algorithm

The algorithm used within the FMS to optimize ferry schedules and routes, ensuring efficient operations.
Ticketing System

A component of the FMS that handles ticket sales, reservations, and boarding passes for passengers.
Passenger Manifest

A document generated by the FMS that lists the names and details of passengers on a particular ferry trip.
Vessel Management

The aspect of the FMS that oversees the maintenance, availability, and allocation of ferry vessels.
RFID: Radio-Frequency Identification

RFID technology may be used in the FMS for tracking passengers and vehicles during boarding and disembarkation.
GPS: Global Positioning System

GPS integration can help in real-time tracking of ferry vessels for accurate scheduling and passenger information.
UI: User Interface

The interface through which ferry operators, employees, and passengers interact with the FMS.
Mobile App

A mobile application that passengers can use to book tickets, check schedules, and receive updates on ferry operations.
Maintenance Log

A record-keeping system within the FMS that tracks maintenance schedules and history for ferry vessels.

Online Booking Portal

A web-based platform that allows passengers to book ferry tickets and access information about ferry schedules.

Payment Gateway

The system that handles financial transactions for ticket purchases and other services within the FMS.

Ferry Route Optimization

Algorithms and tools used by the FMS to optimize ferry routes based on factors like passenger demand, weather conditions, and fuel efficiency.

Safety Protocols

Procedures and guidelines integrated into the FMS to ensure the safety of passengers and crew during ferry operations.

Weather Forecast Integration

The inclusion of weather forecast data in the FMS to help in decision-making regarding ferry operations during adverse weather conditions.

Crew Management

The component of the FMS that manages crew schedules, assignments, and certifications.

Passenger Information System

A part of the FMS that provides passengers with real-time updates on ferry schedules, delays, and other relevant information.

Security Measures

Features and protocols within the FMS to ensure the security of passengers and cargo during ferry journeys.

1.5 Document Conventions

The "Document Conventions" section for a Ferry Ticketing System Software Requirements Specification (SRS) typically includes information about the formatting and typographical conventions used in the document. Here's how it might be structured:

Formatting Conventions:

Font: The document adheres to IEEE formatting requirements, using Arial font size 11 or 12 for all text.
Text Spacing: Text throughout the document is single-spaced.
Margins: The document maintains 1-inch margins on all sides.
Typographical Conventions:

Italics: Italics are used for comments, providing additional context or explanations within the document. For example, notes about specific requirements or clarifications.
Section Titles: Section and subsection titles in the document follow the template provided.
Document Structure:

The document follows the IEEE standard format for Software Requirements Specifications.
It includes sections such as Introduction, Scope, Functional Requirements, Non-Functional Requirements, System Architecture, Data Requirements, User Interface Design (if applicable), Project Timeline, Risks and Assumptions, and Appendices.
Naming Conventions:

If there are specific naming conventions or standards relevant to the Ferry Ticketing System, they are outlined in this section. For example, naming conventions for database tables, variables, or user roles.
Acronyms and Abbreviations:

A list of acronyms and abbreviations used in the document may be provided for the reader's reference.
References:

If there are external references or standards used in the document (e.g., IEEE standards for software requirements), they are cited and listed appropriately in the References section.
Version Control:

If applicable, information about the document's version history and how changes are tracked and documented may be included.
Document Review and Approval:

This section may outline the process for document review and approval, including the names and roles of individuals or teams responsible for reviewing and signing off on the document.
Remember that the specific conventions and standards can vary depending on the organization and industry. It's essential to adhere to any specific documentation guidelines or templates provided by your organization when creating an SRS for a Ferry Ticketing System.References and Acknowledgments

2 Description

2.1 Product Perspective

The Ferry Ticketing System is envisioned as a pioneering, self-contained product that aims to revolutionize the ferry travel experience for both passengers and operators within the transportation industry. It is not designed as a replacement for any existing systems but rather as an innovative solution that addresses the current market's demand for efficient, user-friendly ferry ticket booking and management.

Independence and Comprehensive Platform:
This system stands independently, providing a comprehensive platform for ferry operators to manage their services and for passengers to conveniently plan and book their trips. It introduces a novel approach to handling ferry travel bookings and operations, delivering substantial benefits to both passengers and operators alike.

Passenger Interface:
The system's Passenger Interface encompasses all user-facing aspects, accessible through web and mobile applications. Passengers can efficiently browse available ferry routes, review schedules,

compare fares, and seamlessly make bookings. The system ensures real-time access to information and electronic tickets, significantly enhancing the overall passenger experience.

Operator Interface:
On the operator side, the system offers a dedicated interface, granting ferry operators the tools needed to manage reservations, assign seating, and closely monitor ferry capacities. Furthermore, it facilitates streamlined communication with passengers and provides valuable analytical insights into booking trends and operational efficiency.

External Interfaces:
The Ferry Ticketing System interacts with external entities through the following interfaces:

Payment Gateway: The system interfaces with external payment gateways to ensure secure online transactions for ticket purchases, fostering trust and convenience for passengers.

Ferry Fleet Information System: In some cases, the system might interface with the ferry fleet management system to obtain real-time data on ferry locations and availability, allowing for more accurate scheduling and passenger management.

User Interaction:
Passengers interact with the system through a highly intuitive and user-friendly interface, which enables them to select routes, choose travel dates, view available ferries, and make bookings with ease. On the other hand, operators interact with the system through a dedicated interface that empowers them to efficiently manage reservations, monitor passenger loads, and make informed operational decisions.

System Interaction with Environment:
The Ferry Ticketing System interacts harmoniously with passengers, ferry operators, and external payment gateways, seamlessly providing a superior booking and travel experience. It operates within the context of the ferry travel industry, enhancing the efficiency of ticketing processes and passenger management.

2.2 Product Functionality

Major Functions of the Ferry Ticketing System:

Sell ferry tickets for specific dates and times.
Track passengers by name for accurate passenger manifests.
Allow passenger check-in at the time of boarding through ticket scanning.
Provide real-time vessel occupancy information during ticket sales and boarding.
Enable ticket sales for cargo (both accompanied and unaccompanied).
Allow system configuration for route scheduling, vessel assignment, max seating capacity, ticket fares, etc.
Support the sale of ferry tickets exclusively, without food or beverage items.
Provide extensive reporting capabilities and the ability to create custom reports using industry-standard reporting tools.

2.3 Users and Characteristics

Users of the Ferry Ticketing System:

Passengers:

Characteristics: Passengers are the primary users of the system who book ferry tickets for their travel. They may have varying technical expertise, ranging from tech-savvy individuals to those less familiar with online booking systems.
Frequency of Use: Occasional users, booking tickets as needed.
Functions Used: Ticket booking, route selection, fare comparison, receiving electronic tickets.
Ferry Operators and Staff:

Characteristics: Ferry operators and their staff members utilize the system to manage ferry operations, including reservations, seating assignments, and monitoring passenger loads.
Frequency of Use: Frequent users, interacting with the system on a daily basis.
Functions Used: Managing reservations, assigning seating, checking in passengers, monitoring vessel occupancy.
Administrators:

Characteristics: Administrators oversee the system's configuration, including route scheduling, vessel assignments, ticket fares, and user privileges.
Frequency of Use: Regular users, managing system settings and ensuring smooth operations.
Functions Used: Configuring routes, setting fares, managing user roles and permissions.
Management and Reporting Staff:

Characteristics: These users focus on analyzing system data and generating reports for decision-making. They require a higher level of technical expertise.
Frequency of Use: Occasional users, accessing the system for reporting purposes.
Functions Used: Generating standard and custom reports using industry-standard reporting tools.
Priority Users:

The most important users for this product are passengers and ferry operators/staff. Passengers drive revenue by booking tickets, and ferry operators/staff ensure the smooth operation of ferry services. While administrators and reporting staff are essential for maintaining and optimizing the system, they have less direct impact on day-to-day operations and revenue generation compared to passengers and ferry operators/staff.

Less Important Users:

Management and reporting staff may be considered less critical users in terms of the immediate operational aspects of the ferry services. Their involvement is more focused on analyzing data and making strategic decisions based on the system's insights.

2.4 Operating Environment

The Ferry Ticketing System will operate in an environment that encompasses both the online interfaces used by passengers and the backend systems utilized by ferry operators and administrators. This environment includes the following components:

Hardware Platform: The system will run on standard hardware commonly used for web applications, including servers, databases, and user devices (such as laptops, desktops, tablets, and smartphones).

Operating System: The software will be compatible with various operating systems, including but not limited to Windows, macOS, and Linux distributions. It should also support mobile operating systems such as iOS and Android.

Web Browsers: The passenger interface will be accessible through modern web browsers such as Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge.

Server Infrastructure: The system's backend will operate on web servers, application servers, and database servers. These components will ensure smooth interaction between the different user interfaces and the database.

Database Management System: The system will interact with a relational database management system (RDBMS) to store and manage passenger data, booking information, routes, and schedules.

External Payment Gateways: The system will communicate with external payment gateways to facilitate secure online transactions for ticket purchases.

Reporting Tools: The system will support industry-standard reporting tools for generating various reports and analytics.

Minimum Platform Requirements:

Modern desktop or mobile device with an up-to-date operating system and web browser.
Internet connectivity for accessing the passenger interface and backend systems.

2.5 Design and Implementation Constraints

Hardware and Performance Constraints: The system's performance should meet passenger and operator expectations, necessitating hardware resources that ensure responsive interfaces and quick data processing. However, certain hardware limitations, such as server capacity and processing speed, may influence the system's scalability and response times.

Integration with Payment Gateways: The system's integration with external payment gateways requires compliance with the APIs and protocols provided by these gateways. This constraint limits the available options for handling online payment transactions, as the chosen gateways' technical requirements must be adhered to.

Database Management System: The use of a specific relational database management system (RDBMS) might be a requirement or constraint for data storage and management. This constraint limits the developers to working with the chosen RDBMS and may affect data structure and querying options.

Security and Privacy Considerations: The system must adhere to strict security and privacy regulations, which might limit certain design choices and technologies. The constraint ensures that the software development follows secure coding practices, encryption standards, and data protection measures.

Reporting Tools Compatibility: The requirement to use industry-standard reporting tools for generating various reports and analytics imposes a constraint on the developers. They must choose tools that are compatible with the system's architecture and data structure.

User Interface Design Guidelines: The need for a consistent and user-friendly interface across devices and platforms implies adherence to specific design conventions and user experience principles. Developers must work within these guidelines to ensure a seamless user interaction.

Language and Framework Selection: The choice of programming languages and development frameworks may be constrained by the technical expertise of the development team and the compatibility with the desired system functionalities.

Parallel Operations and Concurrency: The system needs to handle concurrent transactions, bookings, and reservations. Developers must consider concurrency control mechanisms and database transaction management to prevent conflicts and ensure data integrity.

Localization and Internationalization: If the system is intended for use in different regions or countries, it must accommodate multiple languages, currencies, and regional preferences. This constraint can impact the design and implementation of user interfaces and data handling.

Maintenance Ownership: If the customer's organization will be responsible for maintaining the delivered software, the developers need to ensure that the code is well-documented and adheres to coding standards for easy maintenance and troubleshooting.

These constraints guide the development process of the Ferry Ticketing System, influencing decisions about technologies, design, security measures, and system performance.

2.6 User Documentation

The user documentation for the Ferry Ticketing System will include comprehensive components designed to ensure a smooth and user-friendly experience for both passengers and operators. This documentation will consist of user manuals for passengers and ferry operators, on-line help resources accessible through the user interfaces, and interactive tutorials for new users. The user manuals will provide step-by-step instructions on how to navigate the system, make bookings, manage reservations, and utilize advanced features. On-line help resources will offer contextual assistance for specific tasks, guiding users through processes within the application. Interactive tutorials will serve as guided walkthroughs to help users familiarize themselves with the system's interface and functionalities. These user documentation components will be delivered in digital formats, accessible through the web and mobile applications, adhering to industry standards for usability and clarity.

2.7 Assumptions and Dependencies

Internet Connectivity: It is assumed that both passengers and operators will have consistent access to the internet during ticket booking and management processes. Any interruptions in internet connectivity may impact the user experience and the system's ability to update real-time data.

External Payment Gateway Availability: The system relies on third-party payment gateways for processing online transactions. Assumptions are made that these payment gateways will be available, reliable, and compatible with the system's requirements. Changes in the availability or functionality of these gateways could affect the payment process.

Hardware Compatibility: It is assumed that users' devices (computers, smartphones, tablets) meet basic hardware and software requirements for accessing the system. Incompatibility issues with older or unsupported devices could affect the user experience.

Regulatory Compliance: Assumptions are made regarding the system's compliance with industry and regulatory standards for ferry ticketing and online transactions. Any changes or updates to these regulations may necessitate adjustments to the system's functionality.

External Data Sources: If the system relies on external sources for vessel location data or route information, assumptions are made about the availability and accuracy of this data. Any discrepancies or changes in these external sources may affect real-time occupancy calculations and route scheduling.

Development Team Skills: The project assumes that the development team possesses the necessary skills and expertise to implement the required technologies, security measures, and reporting tools effectively. Any limitations in the team's capabilities could impact the development timeline and system functionality.

User Adoption and Training: It is assumed that passengers and operators will adapt to the system relatively quickly and without significant resistance. However, if users face difficulties in adopting the system, additional training or support resources may be required.

External Factors (Weather and Events): The system may need to consider external factors such as weather conditions and special events that can affect ferry schedules and occupancy. Assumptions are made about the predictability of these factors and the system's ability to adapt in real-time.

Integration with Existing Systems: If the system needs to integrate with existing ferry management or reservation systems, assumptions are made regarding the availability of suitable APIs or data exchange mechanisms. Any changes or limitations in these integrations could affect system functionality.

These assumptions and dependencies highlight potential areas of risk and uncertainty that could impact the development and operation of the Ferry Ticketing System. It's crucial to monitor these factors closely and be prepared to adapt to changes or unforeseen challenges during the project lifecycle.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The Ferry Ticketing System will consist of distinct user interfaces catering to both passengers and ferry operators. Below are the logical characteristics of each interface, along with descriptions of some key screens within each interface:

Passenger Interface:

Home Page: Upon launching the application, passengers will be greeted with a home page featuring an intuitive layout showcasing available routes, departure times, and fares. A search bar and filters will allow passengers to refine their search.

Route Selection: After initiating a search, passengers will be presented with a list of available routes. Each route card will display essential information such as departure and arrival locations, dates, times, and ticket fares.

Booking Process: Upon selecting a specific route, passengers will proceed to the booking process. They will provide details such as the number of passengers, preferred seating class, and any additional requirements. The system will calculate the total fare and present payment options.

Electronic Tickets: Once the booking is confirmed, passengers will receive electronic tickets on their device. These tickets will contain QR codes for easy boarding.

User Profile: Passengers can access and manage their user profiles, including saved payment methods, booking history, and preferences.

Ferry Operator Interface:

Dashboard: The dashboard will provide an overview of upcoming trips, occupancy rates, and any urgent notifications. Operators can quickly access critical information.

Reservation Management: Operators will manage reservations through an intuitive interface. They can search for specific bookings, assign seating, and check passengers in by scanning QR codes.

Vessel Assignment: Operators will assign specific vessels to routes based on availability and requirements.

Route Scheduling: The system will facilitate route scheduling, enabling operators to set departure times and frequencies.

Reporting and Analytics: The interface will offer reporting tools to track booking trends, occupancy rates, and revenue. Customizable reports can be generated using industry-standard reporting tools.

Shared Interface Characteristics:Consistent Navigation: Both interfaces will maintain a consistent navigation structure with a top menu or sidebar for easy access to different sections.

Responsive Design: The interfaces will be designed to be responsive, adapting seamlessly to different devices and screen sizes.

Standard Buttons and Functions: Standard buttons such as "Submit," "Cancel," and "Back" will be consistently placed across screens for ease of use.

Error Messages: Error messages will follow a standardized format, providing clear explanations and suggestions for resolution.

GUI Screenshots:
(For illustrative purposes, text descriptions are provided in lieu of actual screenshots.)

Passenger Interface Home Page: A visually appealing home page displaying routes, departure times, and fares. A search bar and filters are prominent.

Route Selection Screen: A list of routes with cards showing departure and arrival details, dates, and fares.

Booking Screen: Passenger inputs number of passengers, seating class, and preferences. The system calculates the fare.

Electronic Ticket: An electronic ticket displaying journey details, QR code for boarding, and booking reference.

Operator Dashboard: A dashboard summarizing upcoming trips, occupancy rates, and notifications.

Reservation Management: A screen showing reservations with QR code scanner for check-in.

Route Scheduling: Interface for setting route schedules and vessel assignments.

3.1.2 Hardware Interfaces

The Ferry Ticketing System interacts with various hardware components to ensure seamless operations and data flow. Below are the logical and physical characteristics of the interfaces between the software and the hardware components:

User Devices (Passenger and Operator Interfaces):

Logical Characteristics: The system supports a wide range of user devices, including desktop computers, laptops, tablets, and smartphones. It accommodates different screen sizes and orientations.

Physical Characteristics: The software interface is accessed via web browsers or dedicated mobile applications. The system's responsive design ensures optimal usability on various devices.
QR Code Scanning Devices (Operator Interface):

Logical Characteristics: The system interacts with QR code scanning devices used by ferry operators for passenger check-in. These devices provide real-time validation of electronic tickets.
Physical Characteristics: QR code scanning devices can be handheld or fixed installations at boarding points. They communicate with the software to validate tickets and update passenger statuses.
Payment Gateways (Passenger Interface):

Logical Characteristics: The software interfaces with external payment gateways for secure online transactions. It communicates payment details, receives payment confirmations, and updates booking statuses.

Physical Characteristics: Payment gateways are third-party services accessed through secure APIs. The system's interface facilitates the secure transfer of payment data.
Data Sources (Real-time Vessel Data and Route Information):

Logical Characteristics: The system interfaces with external data sources to obtain real-time vessel occupancy data and route information. These data sources help determine occupancy during ticket sales and route scheduling.

Physical Characteristics: Data sources could be APIs, databases, or data feeds provided by ferry operators or relevant authorities. The system integrates with these sources to ensure accurate information.
Reporting Tools and Libraries (Administrator and Reporting Staff Interfaces):

Logical Characteristics: The system interacts with industry-standard reporting tools and libraries to generate detailed reports and analytics.

Physical Characteristics: Reporting tools could be software libraries or web services that enable the creation of custom reports and visualizations based on system data.
Dependencies and Special Libraries:

Payment Gateway Integration Libraries: The software may utilize specific libraries or APIs provided by payment gateway providers to ensure secure payment processing.
QR Code Scanning Libraries: QR code scanning devices may use specialized libraries to decode QR codes and validate electronic tickets.
Data Source APIs: Interfaces with external data sources require the use of APIs provided by these sources to fetch real-time data.
The Ferry Ticketing System's hardware interfaces ensure smooth interactions between software components and external devices, contributing to the overall user experience and operational efficiency.

3.1.3 Software Interfaces

The Ferry Ticketing System interfaces with various operating systems to provide a seamless experience for users across different platforms. The specific interfaces include:

Supported Operating Systems:

Windows (versions 7, 8, 10)
macOS (versions High Sierra, Mojave, Catalina)
Linux distributions (Ubuntu, Fedora, CentOS)
Logical Characteristics:

Cross-Platform Compatibility: The software is designed to operate across multiple operating systems to ensure accessibility to a wide user base.
Responsive Design: The user interfaces are optimized for each operating system's conventions, ensuring a familiar and consistent user experience.
Physical Characteristics:

User Interfaces: The passenger and operator interfaces are accessible through web browsers or dedicated mobile applications compatible with the supported operating systems.
Data and Message Flow:

Incoming Data/Messages: User inputs, route information, payment details, and reservation data are entered by users through the software's interfaces. These inputs are processed by the system for various functionalities.
Outgoing Data/Messages: Electronic tickets, booking confirmations, reservation status updates, and real-time occupancy information are sent to users through the interfaces.
Shared Data Across Software Components:

Route and Schedule Data: Shared between the passenger interface, operator interface, and route scheduling system. This data ensures consistency in displayed routes, departure times, and booking availability.
User Profile Information: Shared between the passenger interface and operator interface. Users' profile data helps personalize the booking experience and assists operators in managing reservations.
Implementation Constraint:

Responsive Design Guidelines: The responsive design of the user interfaces must follow the conventions and guidelines of each supported operating system. This ensures that the interfaces seamlessly adapt to the different layouts and interactions of each platform.

The Ferry Ticketing System's interface with various operating systems is essential for accommodating diverse user preferences and device choices. The responsive design and cross-platform compatibility contribute to a user-friendly experience across different environments.

3.1.4 Communications Interfaces

The Ferry Ticketing System relies on several communication standards to ensure seamless interaction between users, external services, and data sources. These standards encompass data security, real-time updates, and reliable information exchange.

1. HTTPS for Web Communications:

The passenger and operator interfaces communicate with users through HTTPS, ensuring secure and encrypted data transfer between the system and user devices.
User inputs, bookings, and sensitive information are encrypted during transmission, enhancing data privacy and security.

2. API-Based Data Exchange:
The system interfaces with external data sources, such as real-time vessel occupancy data and route information, using industry-standard APIs.
APIs facilitate the exchange of data in a structured format, allowing the system to fetch real-time data for occupancy calculations and route scheduling.

3. Email Notifications:
The system sends email notifications to passengers and operators for booking confirmations, boarding instructions, and updates.
Email communication adheres to common email standards, ensuring messages are correctly formatted and delivered reliably.

4. QR Code Scanning:
QR code scanning devices interact with the software's electronic tickets through QR code recognition. The system generates QR codes for each booking.
The QR codes adhere to standard encoding formats, allowing scanners to extract and validate data efficiently.

5. Data Transfer Rates and Synchronization:
Real-time vessel occupancy data is retrieved at regular intervals to provide up-to-date information during the ticket sales process.
Data synchronization ensures that passenger information, reservations, and vessel occupancy data remain accurate and consistent across the system.

6. Data Security and Encryption:
The system employs encryption standards such as TLS/SSL for HTTPS communication to secure data in transit.
Sensitive user information, payment details, and electronic tickets are encrypted to prevent unauthorized access.

7. Error Handling and Feedback:
The communication standards include mechanisms for error handling and feedback to users. Clear error messages and status updates are provided in cases of communication failures or issues.
These communication standards ensure that the Ferry Ticketing System operates reliably, securely, and efficiently, fostering a seamless experience for passengers, operators, and other stakeholders.

3.2 Functional Requirements

The functional requirements of the Ferry Ticketing System are divided into several functional areas, each encompassing specific operations that the system is required to perform. The following subsections detail the product operations related to these functional areas:

1. Ticket Booking and Management:

1.1. Passenger Ticket Booking:

The system shall allow passengers to search for available routes based on departure and arrival locations, dates, and times.
Passengers shall be able to select their preferred route and specify the number of passengers and seating class.
The system shall calculate the total fare based on the selected route, number of passengers, and seating class.

1.2. Electronic Ticket Generation:

Upon successful booking, the system shall generate electronic tickets containing journey details, QR codes, and booking references.
The generated electronic tickets shall be sent to passengers via email and be available for download and printing.

1.3. Booking Confirmation and Payment:

Passengers shall receive booking confirmation emails containing payment instructions.
The system shall support secure online payment through external payment gateways.
After successful payment, the system shall update booking statuses and send payment confirmations to passengers.

1.4. Passenger Profile and Booking History:

Passengers shall have access to a user profile where they can view booking history, save payment methods, and manage preferences.
The system shall display upcoming trips, reservation details, and electronic tickets in the passenger's profile.

2. Reservation and Occupancy Management:

2.1. Operator Dashboard and Notifications:

Ferry operators shall have access to a dashboard displaying upcoming trips, occupancy rates, and urgent notifications.
The system shall notify operators of any critical updates or changes in schedules.

2.2. Reservation Management and Seating Assignment:

Operators shall manage reservations, assign seating, and view passenger details through an intuitive interface.
The system shall update reservation statuses in real-time upon seating assignments.

2.3. Real-Time Vessel Occupancy:

The system shall retrieve real-time vessel occupancy data from external sources.
Operators shall be able to view and monitor real-time occupancy rates during booking and boarding processes.

3. Route and Schedule Management:

3.1. Route Scheduling and Vessel Assignment:

Administrators shall have the ability to configure routes, set departure times, and assign specific vessels to routes.
The system shall support adjustments to schedules and vessel assignments based on demand and operational requirements.

3.2. Route Information and Fare Display:

Passengers shall be provided with accurate route information, including departure and arrival locations, dates, times, and vessel details.
The system shall display fare information for different seating classes along with any additional charges.

3.3. Special Events and Route Changes:

The system shall accommodate special events or route changes by allowing administrators to modify schedules, routes, and vessel assignments accordingly.
Passengers and operators shall be notified of any changes affecting their bookings or trips.

4. Reporting and Analytics:

4.1. Standard and Custom Reports:

The system shall provide predefined standard reports for route performance, booking trends, and revenue analysis.
Administrators and reporting staff shall be able to create custom reports using industry-standard reporting tools.

4.2. Data Export and Visualization:

Reporting staff shall have the ability to export report data in various formats (PDF, Excel, CSV) for further analysis.
Reports shall include visualizations such as graphs and charts to aid in understanding trends and patterns.

5. System Configuration and Security:

5.1. System Settings and Configurations:

Administrators shall be able to configure system settings, including route parameters, vessel assignments, and ticket fares.
The system shall ensure that changes made to configurations are reflected accurately across the application.

5.2. User Roles and Permissions:

The system shall define different user roles (administrators, operators, reporting staff, passengers) and assign appropriate permissions to each role.
User access shall be restricted based on assigned roles to ensure data security and integrity.
5.3. Data Privacy and Security:

The system shall encrypt sensitive user information and payment details during transmission using HTTPS.
Access to user profiles and booking data shall be protected based on assigned user roles.
These functional requirements outline the specific operations and behaviors that the Ferry Ticketing System is expected to perform within each functional area. The system's features, interactions, and capabilities are defined to ensure a comprehensive and user-friendly experience for passengers, operators, and administrators.

4 Other Non-functional Requirements

Performance Requirements

1. Response Time

1.1 The ferry ticketing system shall respond to user interactions within an average response time of 2 seconds for standard operations, such as searching for available tickets, making reservations, and processing payments.

1.2 The maximum allowable response time for any user interaction shall not exceed 5 seconds under normal system load.

2. Scalability

2.1 The system shall be designed to handle a minimum of 100 concurrent users during peak hours without degradation in performance.

2.2 The system's database shall be capable of storing and retrieving data for at least 1 million tickets without significant performance degradation.

3. Availability

3.1 The ferry ticketing system shall target an uptime of 99.9% over any calendar month, excluding planned maintenance periods.

3.2 Planned maintenance periods, which may impact availability, shall be communicated to users at least 72 hours in advance via system notifications.

4. Data Retrieval Time

4.1 The system shall retrieve passenger and ticket information from the database in less than 1 second for common queries.

4.2 Complex queries, such as generating sales reports, shall be completed within 5 seconds, even when dealing with large datasets.

5. Concurrent Transactions

5.1 The system shall support a minimum of 500 concurrent ticket reservation and payment transactions during peak load times.

5.2 Reservation and payment transactions shall be processed in parallel without affecting the overall system stability and performance.

6. Peak Load Handling

6.1 The system shall be stress-tested to ensure it can handle a peak load of 150% of the expected maximum concurrent users without critical failures.

7. Security Response Time

7.1 User authentication and authorization processes shall be completed in less than 1 second to minimize login and access delays.

7.2 Security-related operations, such as fraud detection and prevention, shall not introduce noticeable delays in regular ticketing operations.

8. Logging and Audit Trail

8.1 The system shall maintain an audit trail of all user interactions and system activities, with a minimal impact on system performance.

8.2 Audit logs shall be available for retrieval and analysis within 5 seconds.

These performance requirements ensure that the ferry ticketing system meets user expectations in terms of speed, availability, scalability, and security while also considering peak load conditions. Specific performance testing and measurement criteria should be established to validate that the system meets these requirements during development and ongoing operation.

Safety and Security Requirements

1. User Authentication

1.1 Users shall be required to authenticate themselves using a secure username and password combination.

1.2 Passwords must be stored securely using industry-standard encryption algorithms, and password policies shall enforce complexity and expiration rules.

2. Access Control

2.1 Role-based access control (RBAC) shall be implemented to restrict access to system features and data based on user roles and permissions.

2.2 Only authorized personnel shall have access to sensitive functions such as user data management, system configuration, and financial transactions.

3. Data Encryption

3.1 All communication between the user's browser and the server shall be encrypted using HTTPS to protect sensitive data during transmission.

3.2 Data at rest, including user information and payment details, shall be encrypted within the database to prevent unauthorized access.

4. Payment Security

4.1 Payment processing shall comply with Payment Card Industry Data Security Standard (PCI DSS) requirements to ensure the secure handling of credit card information.

4.2 Payment transactions shall use tokenization or encryption to protect sensitive payment data during storage and transmission.

5. Data Backup and Recovery

5.1 Regular automated backups of the database shall be performed to prevent data loss due to system failures, security breaches, or disasters.

5.2 A disaster recovery plan shall be in place to restore system functionality in the event of a catastrophic failure.

6. User Privacy

6.1 User privacy shall be protected, and explicit consent shall be obtained for collecting, storing, and using personal information in compliance with relevant data protection laws (e.g., GDPR, CCPA).

6.2 Users shall have the option to delete their accounts and associated data from the system.

4.1 Software Quality Attributes

Reliability:

The system should be available 24/7 with minimal downtime.It should accurately process and confirm ticket reservations and payments.Ensure data integrity to prevent loss of critical information.

Availability:

Specify the required uptime percentage (e.g., 99.9%) to ensure continuous service availability.

Include provisions for planned maintenance windows.

Performance:

Define response time requirements for various actions (e.g., ticket booking, seat selection, payment processing).Set throughput goals to handle a specific number of concurrent users or transactions.

Scalability:

Describe how the system should handle an increasing number of users and ferry routes.Specify scalability requirements for both vertical (scaling up) and horizontal (scaling out) scenarios.

Security:

Define authentication and authorization mechanisms for users, administrators, and staff.Outline encryption standards for sensitive data transmission and storage.pecify how the system will handle security vulnerabilities and inciden

5 Other Requirements

Database Requiremnets

The ferry ticketing system's database requirements encompass several critical aspects. Firstly, the choice of the database management system (DBMS) needs to be specified, including the selected DBMS type (e.g., MySQL, PostgreSQL, Oracle) and the version intended for deployment. This ensures consistency and compatibility with the chosen technology stack.
Data security is paramount, especially when dealing with sensitive data such as user payment information. Access control mechanisms should be detailed, including the definition of user roles and permissions, ensuring that only authorized users can access and modify data. Encryption measures to protect sensitive data should also be outlined.

Internationalization requirements

Firstly, the ferry ticketing system should support multiple languages, enabling users from different linguistic backgrounds to interact with the system comfortably. This multilingual support should be scalable to accommodate additional languages as the need arises, ensuring the system remains inclusive.

To address regional variations, localization efforts should extend beyond language translation. The system should adapt to different cultural norms and conventions, encompassing date and time formats, currency symbols, and address formats. This localization approach ensures that the user experience aligns with regional expectations.

Legal requirements

Firstly, the ferry ticketing system must comply with data protection laws, such as the General Data Protection Regulation (GDPR) in the European Union. This includes securing user data, obtaining informed consent for data processing, and providing users with mechanisms to access, rectify, or delete their personal information. The SRS document should specify how the system collects, stores, processes, and protects user data to meet these legal obligations.

In addition to data protection, the system must adhere to payment card industry standards, particularly if it handles payment transactions. Compliance with the Payment Card Industry Data Security Standard (PCI DSS) is crucial to ensure the secure handling of payment card information. The SRS should outline the system's security measures, including encryption, access controls, and regular security assessments, to align with PCI DSS requirements.

Reuse objectives

One of the primary reuse objectives for the ferry ticketing system is to maximize code reusability. This means developing software components, modules, and libraries in a modular and generic manner so that they can be easily reused across various parts of the system. By adhering to coding standards, creating libraries of common functions, and structuring the codebase with a focus on reusability, we aim to reduce development time and effort in the long run. This approach facilitates efficient updates and maintenance since changes made to reusable components automatically propagate to all instances where they are employed.

Appendix A – Data Dictionary

Field Name	Description	Type	Length	Constraints
TicketID	**Unique identifier for a ticket**	**Integer**		**Auto-generated, PK**

PassengerName	Name of the passenger	String	50 chars	
PassengerID	Unique identifier for the passenger	Integer		Auto-generated
Email	Email address of the passenger	String	100 chars	Valid email format
Phone	Phone number of the passenger	String	15 chars	Valid phone format
DeparturePort	Port from which the passenger departs	String	50 chars	
ArrivalPort	Port to which the passenger arrives	String	50 chars	
DepartureDateTime	Date and time of departure	DateTime		
ArrivalDateTime	Date and time of	DateTime		

	arrival			
FerryName	Name of the ferry	String	50 chars	
SeatClass	Class of the seat (e.g., Economy, Business)	String	20 chars	
SeatNumber	Seat number (e.g., A12)	String	10 chars	
TicketStatus	Status of the ticket (e.g., Booked, Cancelled)	String	20 chars	
Price	Price of the ticket	Decimal		
PaymentStatus	Status of payment for the ticket (e.g., Paid, Pending)	String	20 chars	
PaymentTransactionID	Unique identifier for the payment transaction	Integer		Auto-generated

PaymentDateTime	**Date and time of payment**	**DateTime**		
EmployeeID	**Unique identifier for the employee processing the ticket**	**Integer**		
EmployeeName	**Name of the employee processing the ticket**	**String**	**50 chars**	