Ferry ticket Reservation System

Srs Report

****

**Group Members: -**

|  |  |  |
| --- | --- | --- |
| SR No. | Name | Roll No. |
| 1 | Piyush Thummar | 63 |
| 2 | Jaymin Panchal | 71 |
| 3 | Rishi Vora | 83 |
| 4 | HarshrajSingh Vaghela | 84 |

**Table of Contents**

1. **Introduction**
   1. Purpose
   2. Document Conventions
   3. Intended Audience
   4. Project Scope
   5. References
2. **Overall Description**
   1. Product Perspective
   2. Product Features
   3. User Classes
   4. Operating Environment
   5. Design and Implementation Constraints
   6. User Documentation
3. **System Features**
   1. Admin Feature
   2. Customer Feature
4. **External Interface Requirements**
   1. System Interfaces Requirements
   2. Hardware Interfaces
   3. Software Interfaces
   4. Communications Interfaces
5. **Non-functional Requirements**
   1. Non-functional Requirements
   2. Safety Requirements
   3. Security Requirements
   4. Software Quality Attributes
6. **Assumptions & Glossary**
7. **Introduction**
   1. **Purpose:**

The purpose of this Software Requirements Specification (SRS) is to define the functional, non-functional, and technical requirements for the Ferry Ticket Reservation System is created to enhance and simplify the ferry ticket booking process, making it more efficient for the passengers. Its primary purpose is to provide a user-friendly platform for customers to view ferry schedules and book tickets conveniently.

**1.2 Document Conventions:**

This document follows the IEEE Standards for Software Requirements Specification (SRS). Each section is numbered, with clear titles and headings for easy navigation. Specific programming concepts (like class names and method names) are highlighted in monospaced font, and technical jargon is defined in the glossary. For user clarity, the document divides requirements into functional and non-functional categories.

**1.3 Intended Audience:**

This document is intended for: -

**End-users (Passengers):** These are ferry passengers who will interact with the system to book, modify, cancel, and view ferry tickets.

* 1. **Project Scope:**

The **Ferry Ticket Reservation System** aims to provide a comprehensive and efficient solution for booking and managing ferry tickets.

**Features**: -

**Customer Functionality:** Search real-time ferry schedules and online ticket purchasing.

**1.5 References:**

* File I/O method taken from Chat GPT

1. **Overall Description**
   1. **Product Perspective:**

The Ferry Ticket Reservation System is important for sea travel, making it easier to book tickets. This system facilitates real-time seat availability checks, fare calculations and booking confirmations while ensuring data accuracy.

* 1. **Product Features:**
* **Login/Signup:** Users can log in or sign up to create an account and securely access the system.
* **Search & Browse:** Users can search for ferries by entering route details, dates, and times to view available options.
* **Ticket Booking**: Users to easily book a ticket by selecting a ferry, choosing seats, and making secure payments
* **Ticket Modification and Cancellation**: The system also supports flexibility by enabling users to modify their ticket, such as changing parking facility, etc.
* **Customer’s Feedback:** Passengers can provide feedback on their booking experience or the ferry services to help improve our quality.
  1. **User Classes:**

1. **Customer:**

* Can create an account or log in.
* Search available ferries.
* Book, view, modify, and cancel tickets.
* Provide feedback on the booking experience.
  1. **Operating Environment:**

1. **Operating Systems:**

* The system should run on Windows, Linux, or macOS, provided they have the necessary Java Development Kit (JDK) installed.

1. **External Software:**

* Java Development Kit (JDK) version 11 or later.
  1. **Design and Implementation Constraints:**
* **Data Storage:** The system uses local CSV files for storing data, which could pose scalability issues with large amounts of data.
* **No Graphical User Interface (GUI):** The system is limited to a CLI (Command Line Interface).
  1. **User Documentation:**

1. **User Manual:** A detailed document explaining how to use the system for booking tickets, logging in, providing feedback, and managing accounts.
2. **Admin Manual:** Instructions for administrators on managing ferry schedules, reviewing feedback, and maintaining user accounts.
3. **System Requirements Document:** Describes the necessary system specifications and software dependencies.
4. **System Features**
   1. **Customer Feature:**
5. **Registration and Login:** Users can sign up by providing a username, email, and password, or log in if they already have an account.
6. **Ferry Search:** Users can search for available ferries by selecting the source and destination terminals.
7. **Ticket Booking:** Once a ferry schedule is found, users can book tickets, which include seat selection and payment.
8. **Ticket Modification/Viewing:** Users can view their booked tickets and modify or cancel them if needed.
9. **Feedback Submission:** After a ferry trip, users can provide feedback about their experience. This is stored for future review.
10. **System Interface Requirements**
    1. **System Interface Requirements:**
11. **Database Interaction**: The system will interact with CSV files for data storage and retrieval.

* **Key files include:** -
* logins.csv: For storing user credentials.
* schedule.csv: For ferry schedules and availability.
* feedback.csv: For storing user feedback.

1. **Input Prompts:**

* Users are prompted for input, such as Departure and destination locations, Travel dates, Passenger details (e.g., name, age, and number of tickets), Payment confirmation.
* Input validation ensures invalid data (e.g., incorrect dates, unavailable routes) is not accepted.

1. **Output Display:**

* Results like available ferry schedules, ticket pricing, and booking confirmations are displayed in a structured format.
* Ferry details include attributes such as ferry name, route, departure time, arrival time, and ticket price, billing details, including passenger name, total cost, and ticket confirmation, are clearly shown.

1. **Error Handling:**

Clear error messages guide users to correct invalid inputs.

(e.g., "Invalid date. Please enter a valid format.").

* 1. **Hardware Interfaces:**

1. **Input Devices**: Standard keyboard for entering commands and data.
2. **Output Devices**: Monitor or terminal for displaying menu options, ferry schedules, booking details, and error messages.

* 1. **Software Interfaces:**
* Java SE Runtime Environment (JRE) 8+: Executes the compiled Java program.
* Java Utility Libraries: Includes Scanner for input handling and Date Time for managing schedules.
* Console Environment: Compatible with Command Prompt (Windows) and Terminal (macOS/Linux).
  1. **Communications Interfaces:**
* No network communication or multi-user interaction is required.
* The system operates offline in a standalone environment, managing ferry reservations locally.

1. **Non-functional Requirements**
   1. **Non-functional Requirements:**
2. **Performance:** The system should handle several user logins, ferry searches, and bookings simultaneously without noticeable delay.
   1. **Software Quality Attributes:**
3. **Reliability:**

The system should function reliably with correct ferry schedules, accurate ticket booking, and secure login.

1. **Usability:**

Clear menus and prompts make the system accessible even to non-technical users. Simple navigation reduces user errors.

1. **Maintainability:**

The code structure, with modular methods and classes, supports easy debugging and future enhancements.

1. **Scalability:**

The system can be extended to include persistent storage or additional features (e.g., advanced search filters).

1. **Assumptions & Glossary**
   1. **Assumptions:**

* Users have basic knowledge of using a text-based CLI.
* The system is used in a controlled environment where users have a Java runtime environment.
* The ferry schedules are accurate and up to date.
  1. **Glossary:**

1. **Customer:** User who browses and book a ticket.
2. **CSV:** Comma-Separated Values; a file format used for storing data in a tabular form.
3. **Ferry Schedule:** A list of available ferry departures, including source, destination, time, and price.
4. **Ticket:** A record of a customer’s ferry booking, including ferry details, departure time, and seat availability.
5. **Feedback:** A customer's review or comments about their ferry experience, which can influence service improvements.