**Software Requirements Specification (SRS)**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to define the requirements for the Flight Cost Calculator project. This web application will calculate the cost of flights based on distance, departure date, service class, extra baggage, and the number of tickets.

**1.2 Scope**

This project includes developing a web-based application that allows users to input flight parameters and calculates the total cost based on predefined rules. The application will be accessible through standard web browsers and will provide a user-friendly interface for entering data and viewing results.

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* **UI**: User Interface

**1.4 References**

* None

**2. Overall Description**

**2.1 Product Perspective**

The Flight Cost Calculator is an independent web application designed to assist users in calculating the cost of flights. It will be a client-side application running in web browsers without the need for server-side components.

**2.2 Product Functions**

* Input fields for flight distance, departure date, service class, extra baggage, and number of tickets.
* Calculation of flight cost based on predefined rules.
* Display of the total flight cost.
* Validation of user inputs.

**2.3 User Characteristics**

* Users should have basic knowledge of using web browsers.
* No specific technical skills are required to use the application.

**2.4 Constraints**

* The application must run on standard web browsers (e.g., Chrome, Firefox, Safari, Edge).
* Maximum extra baggage allowed is 50 kg.
* Maximum number of tickets allowed per calculation is 4.

**2.5 Assumptions and Dependencies**

* Users will have access to a device with an internet connection and a web browser.
* The application will not store user data.

**3. Specific Requirements**

**3.1 Business Requirements**

**3.1.1 Cost Calculation**

* The application shall calculate the cost of flights accurately based on distance, departure date, service class, extra baggage, and number of tickets.
* The calculation rules shall be transparent to users and consistently applied.

**3.1.2 User Interface**

* The application shall have a clean and intuitive user interface.
* The interface shall provide clear instructions and error messages.

**3.2 Functional Requirements**

**3.2.1 Input Fields**

* The application shall provide an input field for the user to enter the flight distance in miles.
* The application shall provide an input field for the user to select the departure date.
* The application shall provide a dropdown menu for the user to select the service class (Economy, Business, First).
* The application shall provide an input field for the user to enter the extra baggage weight in kilograms.
* The application shall provide an input field for the user to enter the number of tickets.

**3.2.2 Cost Calculation**

* The application shall calculate the cost based on the following rules:
  + Flat rate for flights less than 500 miles: $100
  + For flights between 500 and 1000 miles:
    - $0.10 per mile if departing within 7 days
    - $0.08 per mile if departing within 30 days
    - $0.06 per mile if departing within 90 days
  + For flights over 1000 miles:
    - $0.30 per mile if departing within 7 days
    - $0.25 per mile if departing within 30 days
    - $0.20 per mile if departing within 90 days
  + Service class multipliers:
    - Economy: 1x cost
    - Business: 2x cost
    - First: 3x cost
  + Additional cost for extra baggage:
    - No additional cost for flights less than 500 miles
    - $25 per kg for flights between 500 and 1000 miles
    - $50 per kg for flights over 1000 miles
  + The total cost shall be multiplied by the number of tickets.

**3.2.3 Validation**

* The application shall validate that the flight distance is a positive number.
* The application shall validate that the departure date is a valid date.
* The application shall validate that the extra baggage weight is between 0 and 50 kg.
* The application shall validate that the number of tickets is between 1 and 4.

**3.2.4 Result Display**

* The application shall display the calculated total cost to the user.
* The application shall display error messages in red if any input validation fails.

**3.3 Data Validations**

**3.3.1 Input Data Integrity**

* The application shall ensure that user inputs are sanitized to prevent injection attacks.
* Data inputs shall be validated to ensure they conform to expected formats (e.g., date format for departure date).

**3.4 Technical Requirements**

**3.4.1 Compatibility**

* The application shall be compatible with the latest versions of major web browsers (Chrome, Firefox, Safari, Edge).
* The application shall be responsive and work effectively on both desktop and mobile devices.

**3.4.2 Performance**

* The application shall perform calculations and display results within 1 second.
* The application shall handle simultaneous user requests efficiently without significant latency.

**3.4.3 Security**

* The application shall not store or transmit any user data.
* The application shall implement HTTPS to ensure secure communication between the client and server.

**3.4.4 Maintainability**

* The application code shall be modular and well-documented to facilitate maintenance and updates.
* Code versioning shall be managed using a version control system (e.g., Git).

**4. Use Cases**

**4.1 Calculate Flight Cost**

**Actors:**

* User

**Description:**

* **Basic Flow**:
  1. User enters the flight distance, departure date, service class, extra baggage weight, and number of tickets.
  2. System validates the input data.
  3. System calculates the total cost of the flight based on predefined rules.
  4. System displays the calculated total cost to the user.
* **Alternate Flows**:
  1. If input validation fails (e.g., invalid date, negative baggage weight), the system displays an error message to the user.

**5. Additional Sections**

**5.1 Glossary of Terms**

* **Flight Distance**: The distance in miles between the departure and destination airports.
* **Service Class**: The class of service chosen by the passenger (Economy, Business, First).
* **Extra Baggage**: Additional weight in kilograms beyond the standard allowance for baggage.
* **Number of Tickets**: The quantity of flight tickets being purchased in a single transaction.

**5.2 User Interface Design**

* The user interface shall include a single-page layout with input fields for flight parameters and a results section.
* Error messages shall be displayed prominently in red text near the corresponding input fields when validation fails.

**5.3 Legal and Compliance Requirements**

* The application shall comply with data protection regulations regarding the handling of personal information, despite not storing any user data.
* Terms of use and privacy policy links shall be provided in the footer of the application, outlining user rights and responsibilities.

**5.4 Future Enhancements**

* Potential future enhancements could include integrating with airline APIs to retrieve real-time flight data and pricing information.
* Enhanced reporting features could be implemented to provide users with detailed breakdowns of their flight costs.