**Ex.No:02(b) Preparation of Software Requirement Specification (SRS) Document**

**Date:16.08.2024**

**Aim:**

To prepare the software requirement specification (SRS) document for privacy preserving data encryption strategy for cloud computing in mobile cloud computing**.**

**Concepts Involved:**

**Project Title:**

Trav Split – Online Money Split Up with Team

**Abstract:**

Splitting costs on group trips can be a pain often causing confusion and arguments about who needs to pay what. Our project, "Online Money Split Up with Team," aims to simplify this process and make it fair. The app allows you to group expenses as shared (split by everyone) or individual (split by specific people) ensuring costs are divided. Users can add, change, and remove expenses, with each entry including important info like amount, description, date, who paid, and who's involved. We built the backend with Java and Spring Boot to handle data, and used React for the frontend to create a smooth user experience. The app figures out each person's share and gives a clear breakdown of who owes what. It's built to work with different group sizes and multiple trips making it useful for any group activity. In short, "Online Money Split Up with Team" takes the headache out of managing group expenses. It makes sure everyone knows what they owe, so you can enjoy your trip without worrying about money.

**Process Models:**

**Prescriptive Process Model – Spiral model:**

Chosen for its focus on risk management, iterative development, and flexibility. This helps in refining the "Trav Split" project efficiently through continuous risk assessment and adjustments.

**Specialized Process Model - Component-Based Model:**

Selected because it builds on the Spiral Model's benefits:

* Modularity: Reuse of components for easier development.
* Flexibility: Simplifies updates and upgrades.
* Scalability: Allows the system to grow by adding or modifying components.

**Software Requirement Specification(SRS) Document**

**1. Introduction**

**(i) Purpose of this Document**

The purpose of this Software Requirement Specification (SRS) document is to provide a comprehensive description of the "Trav Split – Online Money Split Up with Team" application. This document outlines the functional and non-functional requirements, interface requirements, and constraints necessary for the development and implementation of the system. It serves as a reference for stakeholders, developers, and quality assurance teams involved in the project.

**(ii) Scope of this Document**

This document covers the requirements for a web-based application designed to facilitate the splitting of expenses among group members during trips or other shared activities. The application will allow users to manage expenses, calculate balances, and track payments within a user-friendly interface. The document details the system's features, operating environment, performance criteria, and design constraints.

**(iii) Overview**

This document provides an overview of the "Trav Split" system, including its intended use, user classes, and the environment in which it will operate. It outlines the key functional requirements, external interface requirements, and non-functional attributes essential for the system's success. The document also includes a preliminary schedule and budget, along with appendices containing supplementary information.

**2. General Description**

The "Trav Split" application is designed to simplify the management of shared expenses among groups, such as during trips or group activities. It integrates with backend services for data management and provides a responsive front-end for users. The system supports multiple platforms and is accessible via modern web browsers.

* **Expense Management:** Users can add, update, and delete expenses.
* **Shared vs. Individual Expenses:** Expenses can be categorized as shared or individual.
* **Expense Tracking:** The system tracks who paid for what and who is involved in each expense.
* **Balance Calculation:** Automatically calculates how much each person owes or is owed.
* **Multi-Trip Support:** Users can manage expenses across multiple trips.
* **Group Size Flexibility:** The system can handle varying group sizes.

**User Classes and Characteristics**

* **Regular User:** Individuals participating in group activities who need to split expenses.
* **Administrator:** Users who can manage trips, groups, and user permissions.

**Operating Environment**

* **Backend:** Java with Spring Boot running on a server.
* **Frontend:** React, compatible with modern web browsers (Chrome, Firefox, Safari, Edge).
* **Database:** Relational database (e.g., MySQL, PostgreSQL).
* **Operating System:** Platform-independent, accessible via any system supporting a web browser.

**Design and Implementation Constraints**

* The application must ensure data consistency and accuracy in calculations.
* The system should be designed to scale, accommodating varying group sizes and trips.
* The front-end should be responsive, supporting both mobile and desktop views.

**3. Functional Requirements**

* **Expense Management:**
  + Support for adding, editing, and deleting expenses with details like amount, description, date, payer, and participants.
* **Balance Calculation:**
  + Automatic calculation of balances after expenses are entered or updated.
  + Provide a clear breakdown of how balances are calculated.
* **Multi-Trip Support:**
  + Manage expenses across multiple trips, with separate calculations for each.

**4. Interface Requirements**

* **Mobile Application Interface:** Secure and user-friendly interface for managing expenses.
* **Admin Console:** Interface for managing trips, user groups, and permissions.
* **Cloud Storage APIs:** Interfaces for storing and retrieving data securely.

**5. Performance Requirements**

* The system should respond to user actions within 2 seconds under normal conditions.
* The backend should support up to 1,000 concurrent users without performance degradation.

**6. Design Constraints**

* Ensure compatibility with major web browsers and cloud platforms.
* The design must accommodate the limited processing power of mobile devices.
* The application should comply with relevant data protection regulations.

**7. Non-Functional Attributes**

* **Data Confidentiality:** Ensure the security and privacy of user data.
* **Usability:** Provide an intuitive and easy-to-navigate interface.
* **Scalability:** The system should scale to support increasing users and trips.
* **Availability:** The system should have a high availability with failover mechanisms in place.
* **Modularity:** The code should be modular and well-documented for ease of maintenance.

**8. Preliminary Schedule and Budget**

**Preliminary Schedule**

* Requirements Gathering: 1-2 weeks
* Design and Architecture: 2-3 weeks
* Implementation: 5-6 weeks
* Testing and Quality Assurance: 1 week
* Deployment and Maintenance: 2 weeks

**Preliminary Budget**

* The preliminary budget for this project is approximately $100.

**9. Appendices**

**References**

* Java and Spring Boot Documentation
* React Documentation
* REST API Design Guidelines

**List of Acronyms Used in the Document**

* **SRS:** Software Requirements Specification
* **UI:** User Interface
* **API:** Application Programming Interface

**Result:**

In this exercise, the software requirement specification document for Trav Split – Online Money Split Up with Team has been prepared successfully.