System Requirements Specification (SRS)

Project Name: Personal Expense Tracker
Prepared By: Tejashwini Vemavarapu
Date: 11/07/2024

Table of Contents

1. **Introduction**
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Definitions, Acronyms, and Abbreviations
    - 1.4 References
    - 1.5 Overview
2. **Overall Description**
    - 2.1 Product Perspective
    - 2.2 Product Functions
    - 2.3 User Characteristics
    - 2.4 Constraints
    - 2.5 Assumptions and Dependencies
3. **System Features and Functional Requirements**
    - 3.1 Expense Tracking
    - 3.2 Transaction Categorization
    - 3.3 Reporting and Visualization
    - 3.4 Predictive Modeling
4. **External Interface Requirements**
    - 4.1 User Interfaces
    - 4.2 Hardware Interfaces
    - 4.3 Software Interfaces
    - 4.4 Communication Interfaces
5. **Non-Functional Requirements**
    - 5.1 Performance Requirements
    - 5.2 Security Requirements
    - 5.3 Usability Requirements

- 5.4 Reliability and Availability
- 5.5 Maintainability and Supportability
- 5.6 Scalability

6. **Other Requirements**
    - 6.1 Legal and Regulatory Requirements
    - 6.2 Data Management Requirements

1. Introduction

1.1 Purpose

The purpose of this **System Requirements Specification (SRS)** is to define the system requirements for the **Personal Expense Tracker** project. This document specifies the functional and non-functional requirements for the application, outlining the system features, external interface requirements, and constraints. It serves as a foundation for development, testing, and validation.

1.2 Scope

The **Personal Expense Tracker** is a web-based application that allows users to monitor and manage their finances. The system will enable users to track daily expenses, categorize transactions, generate spending reports, visualize financial patterns, and predict future expenses based on historical data.

1.3 Definitions, Acronyms, and Abbreviations

- **EDA**: Exploratory Data Analysis
- **SRS**: System Requirements Specification
- **UI**: User Interface
- **API**: Application Programming Interface

1.4 References

- Business Requirements Document (BRD)
- Business Analysis Plan
- Industry standards for financial applications

1.5 Overview

This document provides detailed descriptions of the system's functional and non-functional requirements, along with interface requirements, constraints, and assumptions to guide the development of the **Personal Expense Tracker**.

2. Overall Description

2.1 Product Perspective

The **Personal Expense Tracker** is intended to be a standalone, user-centric application that helps individuals monitor and analyze their expenses. The product will leverage data visualization and predictive modeling to provide insights into spending habits and future expenses.

2.2 Product Functions

The key functions of the application include:

- **Expense Tracking**: Users can log daily expenses.
- **Transaction Categorization**: Automatically categorizes expenses.
- **Reporting and Visualization**: Generates spending reports and visual dashboards.
- **Predictive Modeling**: Forecasts future monthly expenses.

2.3 User Characteristics

The primary users are individuals with basic financial literacy who want to track and manage their expenses efficiently. Users are expected to have general familiarity with using web applications.

2.4 Constraints

- The application must comply with data privacy regulations.
- Limited to open-source tools for data analytics (e.g., Python, Tableau Public).

2.5 Assumptions and Dependencies

- Users will manually input transaction data or use batch upload if available.
- The system relies on historical data for predictive modeling accuracy.

3. System Features and Functional Requirements

3.1 Expense Tracking

- **FR-1**: The system shall allow users to log expenses, including the date, amount, and payment method (cash, credit, debit, etc.).
- **FR-2**: Users should be able to edit and delete logged expenses.

3.2 Transaction Categorization

- **FR-3**: The system shall categorize expenses based on predefined categories (e.g., Food, Travel, Utilities) using keywords.
- **FR-4**: Users can manually adjust the assigned category of any transaction.

3.3 Reporting and Visualization

- **FR-5**: The system shall generate monthly reports summarizing spending patterns by category and payment method.
- **FR-6**: Users shall be able to view interactive dashboards showing spending trends over time.
- **FR-7**: The system shall allow users to filter reports by category, date range, and payment type.

3.4 Predictive Modeling

- **FR-8**: The system shall use historical data to forecast monthly expenses.
- **FR-9**: Predictive models should include linear regression, random forest regressor, and gradient boosting for accuracy.
- **FR-10**: Users should be able to view projected expenses for the upcoming month.

4. External Interface Requirements

4.1 User Interfaces

- The UI shall provide easy navigation with intuitive menus and options.
- The dashboard shall display interactive charts and graphs for visualizing spending patterns.

4.2 Hardware Interfaces

- The system shall be accessible on standard devices (desktop, laptop, tablet, and mobile).

4.3 Software Interfaces

- The system shall integrate with **Tableau** for data visualization and **Python** for data analysis.

4.4 Communication Interfaces

- The system shall use HTTPS for secure communication between the client and server.

5. Non-Functional Requirements

5.1 Performance Requirements

- The system shall support up to 10,000 users simultaneously without performance degradation.
- Average page load time should be under 2 seconds.

5.2 Security Requirements

- All user data shall be encrypted both in transit and at rest.
- Users shall have secure login credentials, requiring strong passwords.

5.3 Usability Requirements

- The application interface shall be intuitive and easy to navigate for non-technical users.
- The UI should support screen readers for accessibility.

5.4 Reliability and Availability

- The application shall be available 99.9% of the time, with planned maintenance scheduled outside peak hours.
- Data integrity shall be maintained even in cases of system failure.

5.5 Maintainability and Supportability

- The system shall be modular to allow for easy updates and enhancements.
- Documentation for code and system features shall be maintained for support teams.

5.6 Scalability

- The application shall be designed to handle increasing user demand and data volume.
- Database and server resources should scale to accommodate additional users as needed.

6. Other Requirements

6.1 Legal and Regulatory Requirements

- The application must comply with **GDPR** for user data protection if users are based in the EU.
- It shall also comply with any relevant data privacy and financial regulations.

6.2 Data Management Requirements

- Data should be backed up daily to prevent loss.
- The system shall store a minimum of five years of transaction history for each user.