Expense Wise SRS Document

# Introduction

**1.1 Purpose**

address the need for users to efficiently track their expenses, manage expense categories, set budgets, and work towards their savings goals. Expense Wise provides users with an intuitive and user-friendly desktop application to help them gain better control over their finances whilst maintaining privacy.

**1.2 Scope**

The application will provide features for expense input, categorization, savings goals and budgeting. It will have a client-server architecture, with a web-based user interface for users to interact with the application.

**1.3 Definitions, Acronyms, and Abbreviations**

SRS: Software Requirements Specification

UI: User Interface

CRUD: Create, Read, Update, Delete

**1.4 References**

Expense Tracker Design Documents

Expense Tracker Wireframes (Figma Designs, Architecture Designs and UserFlow)

Overall Description

**2.1 Product Perspective**

The Expense Tracker application will consist of a client-side user interface and a server-side application that manages the business logic, data storage, and user authentication.

**2.2 Product Features**

User registration and login: Users can create accounts and log into the application to access their personal expense data.

Expense input: Users can enter their expenses, providing details such as the date, amount, description, and category.

Expense categorization: Users can categorize their expenses into predefined or custom categories for better organization.

Saving Goals: Users can add a savings goal, the amount, due date, and also add users to that goal for tracking savings commitment. Allows them to see how much have contributed to the goal and keep track.

Budgeting: Users can set budgets for their expenses and will appear red if it surpassed the budget assigned.

Visual Representation: Pie chart and calendar on the home page that displays all the users inputs is a easy to read format

**2.3 User Classes and Characteristics**

Regular Users: Users who register and utilize the application for personal expense tracking.

Administrators: System administrators responsible for managing user accounts and application settings.

Specific Requirements

**3.1 External Interfaces**

**3.1.1 User Interface**

The user interface should be web-based and provide the following functionalities:

- User registration and login screens

- Expense input form

- Category selection and category creation form options

- saving goal creation and ability to add funds to the goal

- Reports and analytics display done through the dashboard in the form of a pie chart and calendar

- Budget limits

**3.1.2 APIs**

The APIs are within the Server-side code that communicates with mongoDB to retrieve and post the data back and forth

**3.2 Functional Requirements**

**3.2.1 User Authentication**

- Users should be able to register accounts with unique usernames and passwords.

- Users should be able to log into the application using their credentials.

**3.2.2 Expense Management**

- Users should be able to add new expenses, providing details such as date, amount, description, and category.

- Users should be able to view and update existing expenses.

- Users should be able to delete expenses if needed.

**3.2.3 Expense Categorization**

- Users should be able to assign expenses to predefined or custom categories.

- Users should be able to create, update, and delete custom categories.

**3.2.4 Budgeting**

- Users should be able to set budget limits for different expense categories.

- Users should receive notifications when they exceed the defined budget limits.

- Users should be able to set reminders for recurring expenses and receive notifications accordingly.

**3.2.4 Saving Goal**

- User should be able to create a savings goal with total amount, due date, and invite people to that goal

- User should be able to add funds to that goal whenever they would like

**3.3 Non-Functional Requirements**

Performance: The application should respond quickly to user actions, even with a large amount of expense data.

Security: User data should be securely stored and transmitted, and user authentication should be implemented securely.

Usability: The user interface should be intuitive, visually appealing, and easy to navigate.

Reliability: The application should handle errors gracefully and ensure data integrity.

Display: displaying all the data on the front page from the retrieving of the data