Software Requirement Specifications

Expense Tracker APP

Submitted by

Muhammad Anas F22BINFT1E02144
7th/5M

Submitted to

Mr. Faisal shahzad

Department of Information Technology

Faculty of Computing

The Islamia University of Bahawalpur

Summary

"SpendWise" is a modern application designed to help users efficiently track expenses and manage finances. This document outlines the goals, requirements, and design specifications of the project. The app will feature an intuitive interface, operate on both web and mobile platforms, and ensure data security with cloud-based synchronization. It aims to simplify financial management for personal and small business use.

Table of Contents

1. Introduction

Managing finances effectively has become essential in today's fast-paced world. From individuals struggling to track their daily expenses to small business owners trying to balance budgets, the need for a simple yet powerful solution is greater than ever. Traditional methods like pen and paper or spreadsheets are cumbersome and prone to errors. With the increasing availability of smartphones and internet connectivity, digital solutions offer the perfect alternative.

The Spend Wise Expense Tracker APP app and web platform aim to bridge this gap by providing a comprehensive,
user-friendly, and accessible solution for financial management. This project leverages modern technologies to ensure users can effortlessly log their expenses, monitor budgets, and gain actionable insights into their spending habits—all in real-time.

.

1.1. Purpose

The main goal of "Spend Wise Expense Tracker APP" is to help users keep track of their spending, manage their budgets, and get a better understanding of where their money goes. By offering insights into spending patterns, the app lets users set financial goals and alerts them when they're close to overspending. It works seamlessly across devices, making it easy to access your financial data anytime.

1.2. Scope

What "SpendWise" can do:

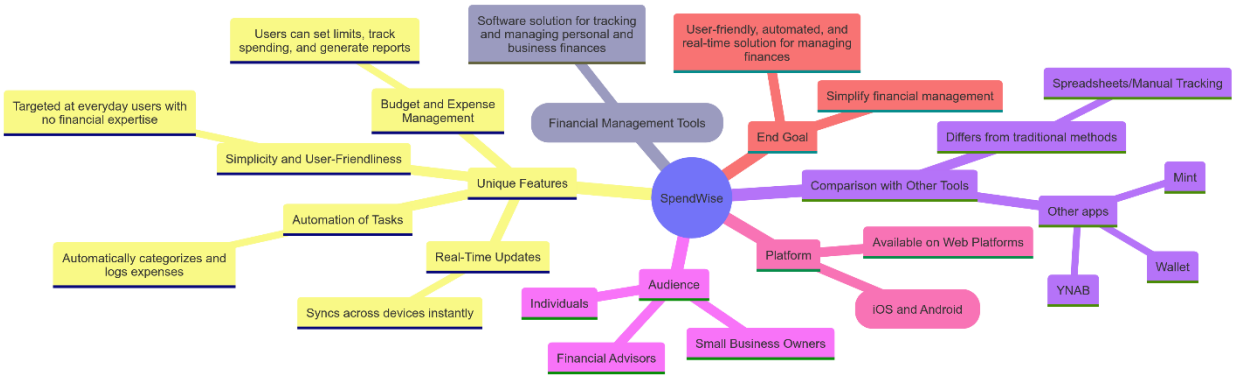
- **Track Expenses:** Log daily spending and organize it by category (like rent, groceries, etc.).
- **Manage Budgets:** Set limits for spending and get notifications when nearing them.
- **Data Analysis:** Provide clear reports and visualizations (charts and graphs) to see spending habits.
- **Cross-Device Sync:** Access and update data across all your devices.
- **Wide Accessibility:** Available on both mobile (iOS and Android) and web platforms.

What it won’t do:

- Advanced business-level accounting, like tax calculations or ERP system integration.
- AI-powered investment tracking or wealth management features.

1.3. Product Perspective

"Spend Wise Expense Tracker APP" is part of the financial management tools category but stands out by being simple, effective, and tailored for everyday users. Unlike spreadsheets or manual tracking, the app automates tasks and offers real-time updates, making it easier for people to stay on top of their finances.



1.4. User Characteristics

- **Who it’s for:**

1. Individuals managing personal finances.
2. Small business owners tracking their expenses.
3. Financial advisors working with clients’ data.

- **What they need:**

1. Basic knowledge of using apps or browsers.
2. No expertise in finance or accounting is required.
3. Language options to support users from different regions.

1.5. Similar apps and systems/Literature Review

Many tools like Mint, YNAB (You Need a Budget), and Wallet exist, but each has some

limitations.

- Mint is great at categorizing expenses but doesn’t support multiple currencies.
- YNAB focuses on budgeting but is complex to use.
- Wallet has excellent analytics but lacks flexibility for customization.

"SpendWise" combines the best of these while being easy to use and affordable.

1.6. Proposed Technologies

We’ll build "SpendWise" using:

- **Frontend:** React.js for the web interface and Flutter for mobile apps.
- **Backend:** Node.js and Express.js for server-side operations.
- **Database:** MongoDB for structured data and Firebase for live updates.
- **Hosting:** AWS for scalable cloud storage and processing.

2. Requirements

The **SpendWise** application is designed to meet a variety of user needs, ranging from basic expense tracking to advanced reporting and analytics. This section outlines the requirements that guide the development of the system. The requirements are divided into functional and non-functional categories to ensure clarity and comprehensiveness.

General Requirements

3. The system must provide an easy-to-use interface for both mobile and web users.
4. It should allow users to manage their financial data securely, ensuring data privacy and integrity.
5. The application must handle various currencies and offer multilingual support for global accessibility.
6. It must support role-based access, ensuring that admins and users have different permissions.
7. The app should seamlessly integrate with third-party services (e.g., Google authentication, cloud storage).

2.1. Functional Requirements

Authors will provide definite number of functional requirements in standard format. These requirements directly in the with functions which are already provided. The requirement format as following.

a) Sign-Up

Purpose: Allow new users to create accounts.
Details:

- **Input Fields:** Full Name, Email, Password (8+ characters, including numbers and special characters), Contact Number.
- **Process Flow:**
    - User enters details.
    - System validates input (e.g., email format, password strength).
    - Verification email sent; account activated upon confirmation.
    - Encrypted user details stored securely in the database.

Outcome:

- Successfully registered users can log in.
- Error messages displayed for invalid inputs (e.g., "Email already registered").

b) Expense Management

Purpose: Enable users to log and manage expenses.
Details:

- **Input Fields:** Amount, Date, Category, Description.
- **Process Flow:**
    - User selects "Add Expense" and fills in the required details.
    - System validates inputs.
    - Expense is stored and displayed on the dashboard.

Outcome:

- Users can view and edit logged expenses.

c) Budget Setting

Purpose: Allow users to set financial limits and track them.
Details:

- **Input Fields:** Budget Amount, Timeframe, Categories.
- **Process Flow:**
    - User defines a budget.

- System tracks spending and notifies users of progress.

Outcome:

- Notifications are sent when spending nears or exceeds the budget.

2.2. Non-Functional Requirements

How it will work:

- **Performance:** Ensure data synchronization within 5 seconds.
- **Security:** Employ encryption, two-factor authentication, and regular vulnerability assessments.
- **Scalability:** Support up to 50,000 concurrent users with optimized database queries.
- **Usability:** Maintain a clean, simple, and accessible interface with adaptive layouts for mobile and desktop.
- **Reliability:** Guarantee 99.9% uptime with automated server monitoring and failover mechanisms.
- **Compliance:** Adhere to GDPR and other global data privacy standards.

3. Use Cases and Flow of Processes

This section explains how different processes in "SpendWise" operate through user interactions. Each use case highlights a specific function of the system, detailing the steps users follow and how the app responds. These workflows are designed to ensure smooth functionality and meet user expectations. The processes include recording expenses, generating financial reports, managing budgets, and synchronizing data across devices.

The following use cases focus on two core functionalities of the app: adding an expense and generating reports. These demonstrate how users interact with the app and how the system ensures the desired outcomes.

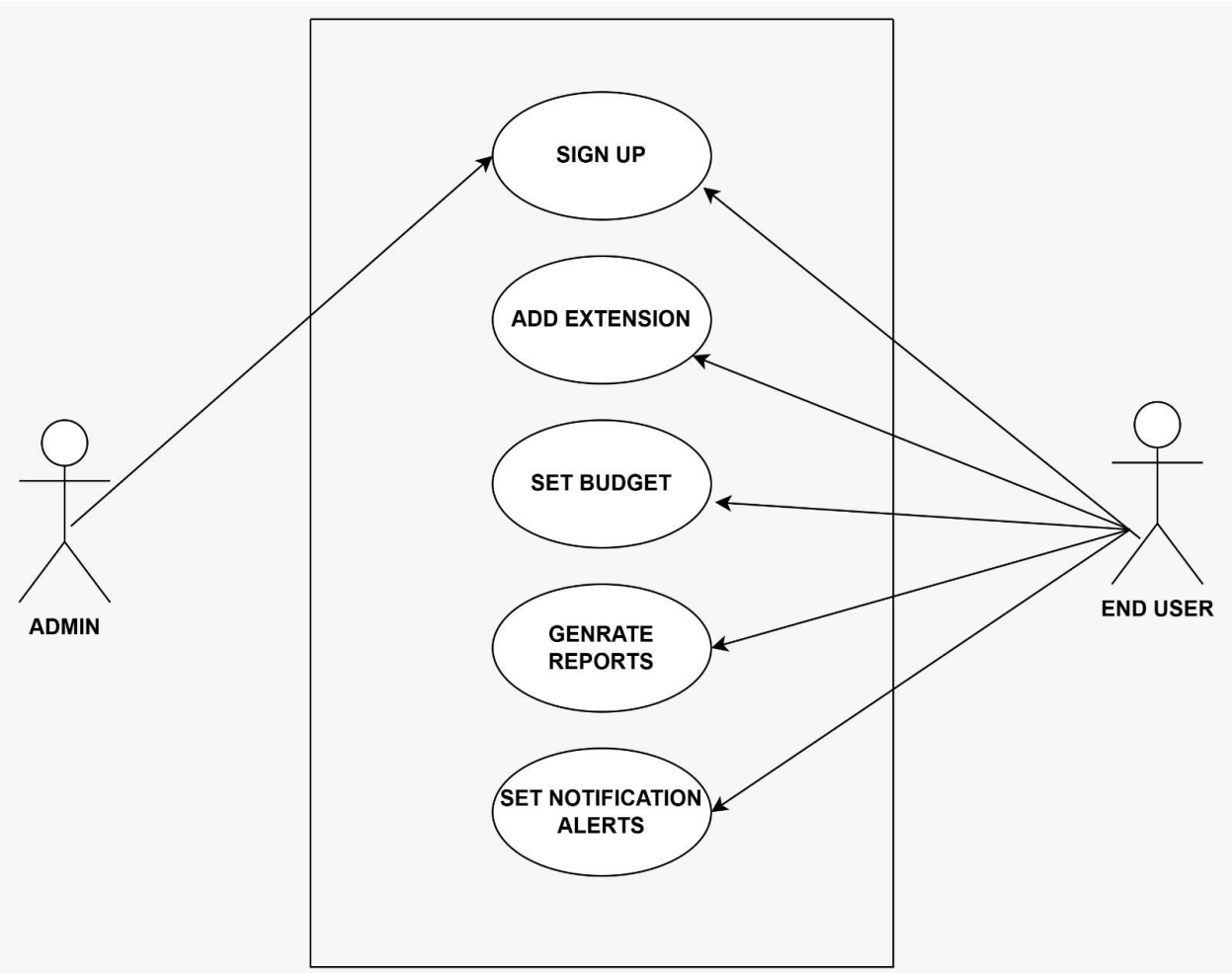


FIGURE 1: SYSTEM LEVEL USE CASE DIAGRAM

3.1 Use Cases 1

Use Case ID	UC001
Name	Sign Up
Description	This use case describes the process of signing up a new user for the SpendWise app.
Requirement(s)	FR001
Actor(s)	Admin, End User
Precondition	The user is not yet registered in the app.
Postcondition	The user is successfully registered and can log in to the app.
Basic Flow	1. Actor selects "Sign Up". 2. The system displays the registration form. 3. Actor enters required details (Name, Email, Password, Phone). 4. System validates inputs (email format, password strength, phone number). 5. On successful validation, the system creates a new user and sends a confirmation email. 6. The user activates the account from the email.
Outcome	Successfully registered users can log in. Error messages are shown for invalid inputs (e.g., "Email already registered").

3.2 Use Cases 2

Use Case ID	UC002
Name	Add Expense
Description	This use case describes the process of adding an expense to the SpendWise app.
Requirement(s)	FR002
Actor(s)	End User
Precondition	The user is logged into the app.
Postcondition	The expense is logged, categorized, and displayed on the dashboard.
Basic Flow	1. Actor selects "Add Expense". 2. The system displays an expense entry form. 3. Actor enters required details (Amount, Date, Category, Description). 4. The system validates inputs. 5. Once validated, the expense is stored and displayed on the dashboard.
Outcome	Expense is added to the system and visible on the dashboard. Error messages are shown for invalid inputs.

3.3 Use Cases 3

Use Case ID	UC003
Name	Set Budget
Description	This use case describes the process of setting a budget in the SpendWise app.
Requirement(s)	FR003
Actor(s)	End User
Precondition	The user is logged into the app.
Postcondition	A budget is set, and the user receives notifications if the budget is close to being exceeded.
Basic Flow	1. Actor selects "Set Budget". 2. System prompts for budget amount, categories, and timeframe. 3. Actor enters details. 4. System validates inputs. 5. Once validated, system tracks expenses in real-time. 6. Notifications are sent when the budget is nearing its limit.
Outcome	Budget is successfully set and tracked. Notifications are sent when nearing/exceeding the budget.

3.4 Use Cases 4

Use Case ID	UC004
Name	Generate Report
Description	This use case describes the process of generating a financial report in the SpendWise app.
Requirement(s)	FR004
Actor(s)	End User
Precondition	The user has logged expenses in the app.
Postcondition	A summary report is generated, displayed, and available for download.
Basic Flow	1. Actor selects "Generate Report". 2. The system prompts the user to select a time period. 3. Actor selects the time period. 4. The system generates trends and graphs. 5. Report is displayed on the screen. 6. Actor can export the report as PDF or CSV.
Outcome	Detailed report is generated and available for download. User can view and export the report.

3.5 Use Cases 5

Use Case ID	UC005
Name	Set Notification Alerts
Description	This use case describes the process of setting up budget alerts in the SpendWise app.
Requirement(s)	FR005
Actor(s)	End User
Precondition	The user is logged in and has set a budget.
Postcondition	The user will receive notifications when approaching or exceeding the budget limit.
Basic Flow	1. Actor selects "Set Notifications". 2. The system displays alert options for budget limits. 3. Actor configures alert levels (e.g., 75%, 90%, 100%). 4. The system saves the notification settings. 5. System tracks spending and sends alerts when nearing/exceeding the budget.
Outcome	Notifications are set and sent when the user approaches or exceeds the set budget.

4. References

Mint, YNAB, and Wallet Apps Documentation:

- **Mint**: https://www.mint.com/how-mint-works
- **YNAB (You Need a Budget)**: https://www.youneedabudget.com/support/
- **Wallet**: https://www.walletapp.com/

MongoDB and Firebase Architecture Guides:

- **MongoDB**: https://docs.mongodb.com/
- **Firebase**: https://firebase.google.com/docs

React.js Official Docs for Building Web Apps:

- React.js Documentation

AWS Cloud Hosting and Security Guidelines:

- **AWS**: https://aws.amazon.com/documentation/
- **AWS Security Best Practices**: https://aws.amazon.com/security/