# Software Requirements Specification for Expense Tracking App

# Introduction

This project is dedicated to building an expense management app. This app will allow users to manage their expenses with their friends, colleagues and family. With the help of this app, users can track down their expenses easily with the help of graphs provided. This product also helps users to split their expenses in several ways. Additionally, users can create groups with other individuals to settle-down their expenses.

## 1.1 Purpose

This document specifies the requirements for a simple web and mobile application designed to facilitate basic shared expense tracking, such as trips, dining, or household bills. The system will provide accurate calculations and a user-friendly interface, prioritizing simplicity and ease of use.

## 1.2 Document Conventions

- \*\*Priority Levels\*\*: High, Medium, Low.  
- \*\*Formatting\*\*: Bold for critical requirements, italics for optional features.  
- \*\*Standards\*\*: Adapted from IEEE 830-1998.

## 1.3 Intended Audience and Reading Suggestions

- \*\*Developers\*\*: Focus on Sections 3 (Specific Requirements) and 4 (Non-Functional Requirements).  
- \*\*Testers\*\*: Review functional and performance requirements.  
- \*\*Instructors/Mentors\*\*: Evaluate Overall Description for scope and objectives.  
- \*\*End-Users\*\*: Focus on User Interfaces and Product Scope sections.

## 1.4 Product Scope

The app enables users to:  
- Create groups for shared expenses.  
- Add and manage expense details with basic split options.  
- Track balances among group members.  
- View a summary of expenses and balances.

## 1.5 References

- Simplified documentation on expense-sharing concepts.  
- Basic API documentation for integration.

# 2. Overall Description

## 2.1 Product Perspective

The app is a standalone product designed as an introductory project for computer science students. It will focus on implementing fundamental functionalities without advanced integrations or optimizations.

## 2.2 Product Functions

- \*\*Expense Management\*\*: Users can log expenses and assign equal or custom splits.  
- \*\*Group Management\*\*: Create and manage groups with limited members.  
- \*\*Balance Tracking\*\*: Display balances for each group member.  
- \*\*Expense Summary\*\*: Provide a basic overview of group expenses.

## 2.3 User Classes and Characteristics

1. \*\*Basic Users\*\*: Individuals splitting bills among friends or family.  
 - Require simple and intuitive features.

## 2.4 Operating Environment

- Platforms: Web.  
- Backend: Hosted on a free-tier cloud platform (e.g., Heroku, Firebase).  
- Database: Basic relational database (e.g., SQLite or MySQL).

## 2.5 Design and Implementation Constraints

- Minimal dependency on third-party APIs.  
- Limited to basic security and performance requirements.  
- Designed for small-scale usage (e.g., up to 50 users).

## 2.6 User Documentation

- Simple in-app guides and tooltips.  
- Basic FAQ section accessible via the webapp.

## 2.7 Assumptions and Dependencies

- Users have internet connectivity for syncing data.  
- Free-tier cloud resources are sufficient for hosting.  
- Dependencies on open-source libraries or frameworks.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

- Simple, responsive interface designed for ease of use.  
- Group creation and expense logging forms with basic validation.  
- Summary screen displaying key information.

### 3.1.2 Hardware Interfaces

- Supports Android devices (version 9+).  
- Web application optimized for Chrome and Firefox.

### 3.1.3 Software Interfaces

- Basic database integration for storing user and expense data.  
- Optional: Lightweight APIs for future scalability.

### 3.1.4 Communications Interfaces

- Secure HTTPs protocols for data transfer.

## 3.2 Functional Requirements

### Authentication Module

- Inputs: User email and password.  
- Process: Validate credentials using a simple authentication system.  
- Outputs: Successful login redirects to the dashboard.

### Expense Logging Module

- Inputs: Expense amount, description, payer, participants.  
- Process: Validate input, calculate splits, and store in the database.  
- Outputs: Updated balances for all participants.

### Group Management Module

- Inputs: Group name and participants.  
- Process: Create or edit group details.  
- Outputs: Updated group list.

# 4. Other Non-Functional Requirements

## 4.1 Performance Requirements

- Response time under 3 seconds for key operations.  
- Designed for small-scale usage (e.g., up to 50 active users).

## 4.2 Security Requirements

- Basic password protection and hashing for user credentials.  
- HTTPS for secure communication.

## 4.3 Software Quality Attributes

- \*\*Usability\*\*: Simple navigation with minimal setup.  
- \*\*Scalability\*\*: Basic design to support potential future expansions.  
- \*\*Maintainability\*\*: Clear code structure and documentation.