
Software Requirements Specification

for

Personal Financial Goal and Budget Tracking Application

Version 1.0 approved

Prepared by Tunahan Oral

Software Engineering Course

2025-11-07

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) describes the functional and non-functional requirements for the Personal Financial Goal and Budget Tracking Application. The purpose of this document is to define the goals, features, interfaces, and constraints of the system that enables users to manage financial goals, track budgets, record transactions, and view financial progress.

1.2 Document Conventions

- Requirements are labeled as **REQ-#**.
- High-priority requirements are marked as (**High**), medium priority as (**Medium**).
- Terminology such as *goal*, *budget*, and *transaction* follows definitions in the Glossary.
- Bold text is used for section headers and requirement identifiers.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for:

- **Developers:** To understand functional and non-functional requirements.
- **Project Managers:** To plan development progress and milestones.
- **Testers:** To create test cases based on defined system behavior.
- **Instructors/Evaluators:** Reviewing the project based on standard SRS practices.

Readers should begin with “Overall Description” for system understanding, then proceed to “System Features” for detailed requirements.

1.4 Product Scope

The application helps users plan financial goals, manage monthly budgets, monitor income/expenses, and view progress toward savings targets.

It supports personal finance decision-making, encourages disciplined saving habits, and provides automated financial insights to help users achieve long-term goals.

1.5 References

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

UML Use Case & Class Diagrams (Prepared by Student)

Financial Management Design Principles Documentation

2. Overall Description

2.1 Product Perspective

This system is a **standalone personal finance management application**.

It includes the following subsystems:

- Goal Management
- Budget Tracking
- Transaction Recording
- Reporting & Analytics
- User Account Management

No legacy systems are replaced; this is a new product designed for individual users.

2.2 Product Functions

- Creating financial goals
- Creating budgets
- Recording income and expenses
- Viewing financial analytics and reports
- Calculating required savings
- User authentication

2.3 User Classes and Characteristics

Regular User: Manages personal finances using the app

Admin (Optional): Manages data integrity, oversees system logs

2.4 Operating Environment

Operating System: Windows, macOS, Android, iOS

Database: SQLite or any local relational database

2.5 Design and Implementation Constraints

Must comply with data protection guidelines (basic privacy principles).

Mobile UI must remain lightweight and responsive.

SQLite or equivalent must be used for local data persistence.

System must support storage of multiple financial goals per user.

2.6 User Documentation

User Guide (PDF)
In-app tooltips and help section

Quick start guide explaining basic operations

2.7 Assumptions and Dependencies

Users provide accurate financial data.

Application depends on stable local storage.

Cloud backup (if used) depends on third-party storage services.

3. External Interface Requirements

3.1 User Interfaces

Dashboard for goals, budgets, and financial summaries

Forms for adding/editing transactions

Visual charts (progress bars, pie charts)

Login and registration screens

Consistent layout with standard buttons (Save, Edit, Delete)

3.2 Hardware Interfaces

Local device storage.

3.3 Software Interfaces

SQLite database, optional cloud API.

3.4 Communications Interfaces

HTTPS (if cloud sync enabled)

4. System Features

4.1 Financial Goal Management

Allows users to create, edit, delete, and monitor personal financial goals.

Priority: High

Stimulus/Response Sequences

1. User selects "Create Goal" → System prompts for details
2. User enters target amount, deadline, description → System saves the goal

3. User views progress → System calculates % completed

Functional Requirements

REQ-1: The system shall allow users to create a financial goal.

REQ-2: The system shall calculate progress automatically.

REQ-3: The system shall allow users to edit or delete goals.

4.2 Budget Management

Users can create monthly budgets and set category-based spending limits.

Priority: High

Functional Requirements

REQ-4: The system shall allow users to create monthly budgets.

REQ-5: The system shall track expenditures and compare with limits.

REQ-6: The system shall warn users when budget limits are exceeded.

4.3 Transaction Recording

Supports recording income and expenses with categories.

Priority: High

Functional Requirements

REQ-7: The system shall allow users to record income.

REQ-8: The system shall allow users to record expenses.

REQ-9: The system shall categorize transactions.

REQ-10: The system shall maintain a transaction history.

4.4 Reporting & Analytics

Provides visual and tabular summaries of spending and goal progress.

Priority: Medium

Functional Requirements

REQ-11: The system shall display monthly spending by category.

REQ-12: The system shall display progress toward each goal.

REQ-13: The system shall generate historical financial reports.

4.5 User Account Management

Manages login, logout, and profile data.

Priority: High

Functional Requirements

REQ-14: The system shall authenticate users.

REQ-15: The system shall allow password updates.

REQ-16: The system shall secure sensitive user data.

5. Nonfunctional Requirements

5.1 Performance Requirements

Dashboard loads <2 seconds.

5.2 Safety Requirements

Confirmation dialogs before deletion.

5.3 Security Requirements

The system shall encrypt stored passwords.

Only authenticated users may access financial data

5.4 Software Quality Attributes

Usability: UI must be simple and intuitive.

Reliability: System should handle minor crashes without data loss.

Maintainability: Code should follow modular structure.

Portability: Should be deployable on both mobile and desktop platforms.

5.5 Business Rules

A user cannot create transactions without an existing profile.

Budget periods must not overlap.

Each transaction must belong to one category.

6. Other Requirements

Appendix A: Glossary

Goal: A financial target defined by the user.

Budget: A monthly spending plan with category limits.

Transaction: A recorded income or expense.

Progress: The percentage of the goal achieved.

Appendix B: Analysis Models

UML Use Case Diagram

UML Class Diagram

Appendix C: TBD List

TBD-1: Cloud provider

TBD-2: Notification services