Software Requirements Specification (SRS)

Project Title: FinaNote — A Simple Expense Tracker for Students

Prepared for: Software Engineering Project

Prepared by: Omar Ait said & Ahmed Khalil El Atri

Instructor: Pr. Imane Fouad

Date: October 2025

Contents

1. Introduction

1.1. Purpose

This document defines the software requirements for **FinaNote**, a small web application that helps students manage and track their daily expenses. It describes what the system will do, how it will behave, and what technologies will be used to ensure the project is realistic.

1.2. Scope

FinaNote allows students to record their daily expenses, organize them by category, and monitor their monthly budget. It includes the following features:

- Add and view expenses.
- Organize expenses by category (food, transport, etc.).
- Display total spending by day and by month.
- Warn the user when they exceed a defined budget.

The goal is to make personal money management easier for students through a simple interface.

1.3. Definitions and Abbreviations

Term	Definition
SRS	Software Requirements Specification
FR	Functional Requirement
NFR	Non-Functional Requirement

1.4. References

No external references were required for this project as it is an original design based on our group choices.

2. Overall Description

2.1. Product Overview

FinaNote will be a web-based application built with:

- **Frontend:** HTML, CSS for the user interface
- **Backend:** Python for logic and server communication
- **Database:** SQLite to store expense records and user data

2.2. Product Functions

Main functions:

- Add new expense entries (amount, category, date, description).
- View all recorded expenses in a table format.
- View total spending by month and by category.
- Set a budget limit and receive a warning when it is exceeded.
- Edit or delete existing expense entries.

2.3. User Characteristics

The primary users are students and young adults who want a simple way to manage their money. The interface must therefore be simple and easy to understand without any tutorials.

2.4. Constraints

- Build our FinaNote application within our current technical abilities.
- Interface must remain simple and responsive.
- Must be completed within the academic semester.
- Respect submission deadlines and keep updating our project.

2.5. Assumptions:

1. The user has a phone or computer to use the FinaNote app.
2. The user has basic knowledge of how to use simple apps.
3. The device has enough storage to save expense data.
4. The internet connection is available when needed (for updates or backups).
5. The data entered by the user is correct and honest.

2.6. Dependencies:

1. The app depends on the device's operating system (Android, iOS, or PC).
2. It depends on some external libraries for charts, PDF export, and database.

3. It depends on internet access for online features and updates.
4. It depends on the development tools and frameworks used to build it.

3. Specific Requirements

3.1. Functional Requirements

ID	Requirement	Priority
F1	The system shall allow the user to add an expense with amount, category, date, and description.	Must Have
F2	The system shall display the remaining budget amount after each new expense is added.	Must Have
F3	The system shall display a list of all expenses.	Must Have
F4	The system shall calculate and display total monthly spending.	Must Have
F5	The system shall allow the user to set a monthly budget.	Must Have
F6	The system shall warn the user when the budget limit is exceeded.	Must Have
F7	The system shall allow the user to delete or edit an expense.	Should Have
F8	The system shall display total spending by category (food, transport, etc.).	Should Have
F9	The system shall allow data export as a PDF file.	Could Have
F10	The system shall allow basic chart visualization of spending.	Could Have

3.2. Exceptional Requirements

ID	Requirement Description
E1	If the user leaves a required field empty, the system shall display a clear error message.
E2	If a database error occurs, the system shall show a message and avoid crashing.

E3	If the connection to the database fails, the system shall retry automatically.
E4	If invalid data (e.g., negative amount) is entered, the system shall reject it.
E5	If the user exceeds 90% of the budget, the system shall display a budget warning message.

3.3. Non-Functional Requirement

ID	Requirement Description
N1	**Performance:** The system should load pages and respond within 2 seconds.
N2	**Usability:** The interface must be simple, readable, and intuitive.
N3	**Security:** Data must be stored locally and private.
N4	**Reliability:** Expense data should not be lost after the app closes.
N5	**Compatibility:** The web app must work on all modern browsers.
N6	**Maintainability:** Code should be modular and easy to update.
N7	**Scalability:** The system must handle hundreds of expense entries without slowing down.

4. Conclusion

This SRS defines the complete set of requirements for the **FinaNote** application. It ensures that the project is achievable with our current knowledge. The system will be developed to be efficient, and user-friendly, respecting all functional and non-functional expectations.