

SOFTWARE ENGINEERING PROJECT

Finanote

Student Expense Tracking Platform

Ahmed Khalil EL ATRI – Omar AIT SAID

November 2025



Project Overview



FinaNote

The Problem

Students struggle to manage expenses effectively, often relying on manual methods or generic apps that are too complex or raise privacy concerns.

Our Solution

A simple expense tracker designed specifically for students with intuitive interface, achievement system, and comprehensive financial analytics.

Key Features

- Track daily expenses with categories
- Set and monitor budget limits
- View spending summaries and alerts



Feasibility Study

Evaluation of technical capabilities, market needs, and
implementation timeline

Market research and validation

User Survey: 27 Students (October 2025)

Would you use a simple, free web app that helps you track expenses without registration or ads?

27 responses

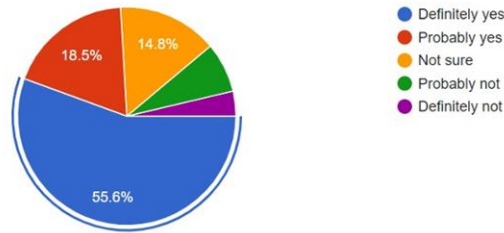


Figure 4: User interest in a simple, free, registration-free expense tracking app

✓ 55.6% would definitely use FinaNote

What are the main problems you face when managing your money? (Select all that apply)

27 responses

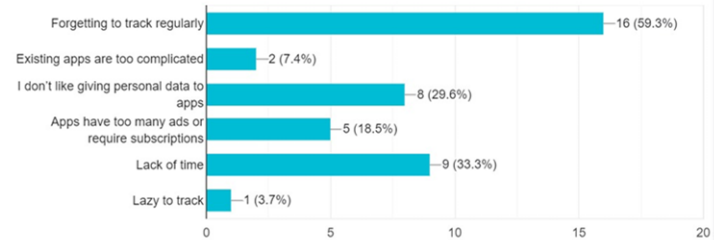


Figure 3: Main problems faced by students when managing money

⚠ 59.3% struggle with regular tracking

🔍 Clear market need identified

Technical Stack

Timeline: Oct 7 - Nov 31 (8 weeks)

Backend

Spring Boot 3.2.5

Java enterprise framework

Java 21

Programming language

Hibernate / JPA

Object-relational mapping

H2 Database

Embedded database

Frontend & Security

HTML5 / CSS3

Modern markup and styling

JavaScript

Dynamic client-side logic

Spring Security

Authentication framework

JWT Tokens

Stateless authentication

Requirements: MoSCoW Method

Must Have

- Add/view expenses
- Set budget limits
- Budget warnings
- User authentication

Should Have

- Edit/delete expenses
- Category breakdown
- Dashboard analytics

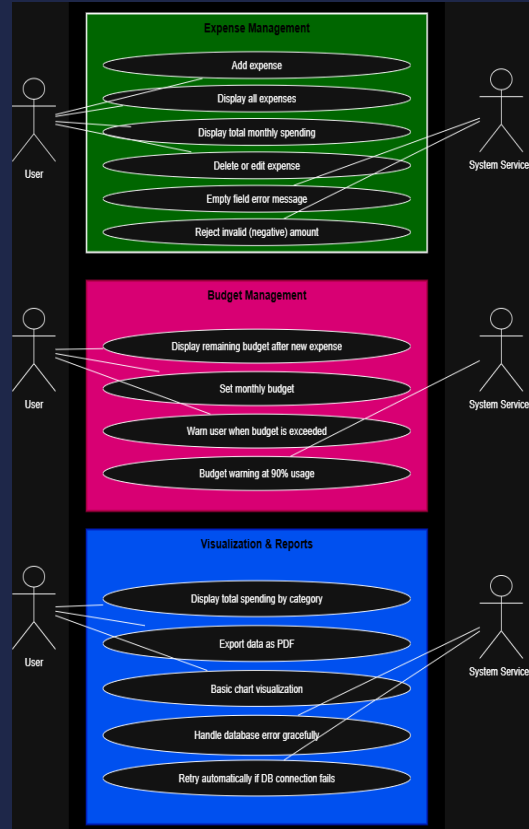
Could Have

- PDF export
- Chart visualization
- Savings goals

Won't Have (v1)

- Mobile app
- Social features
- Bank integration

Use Case Diagram



Actors & System Interactions



Student/User



Authentication

Register, Login, Logout, Password Reset



Expense Mgmt

Add, Edit, Delete, View Expenses



Reporting

Monthly Total, Category Breakdown, Biggest Expense



Budget

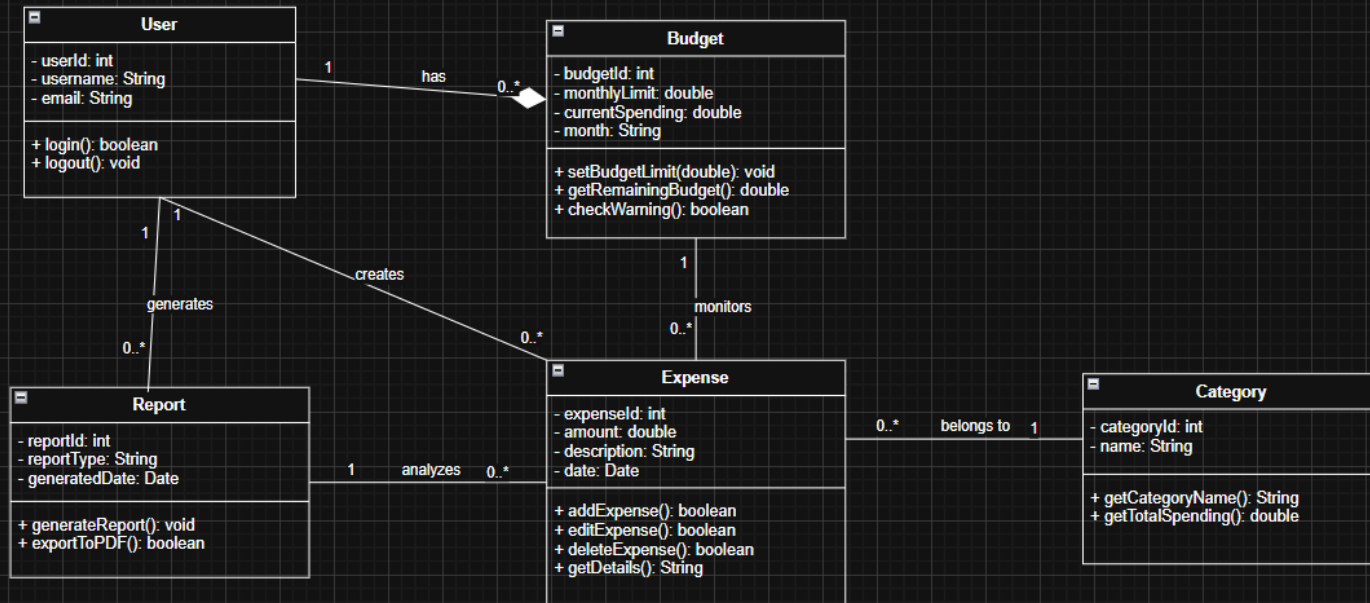
Set Budget, Monitor Spending, Receive Alerts



Data

Export CSV, Import CSV, Backup Data

Class Diagram



Core Classes: User • Expense • Category • Budget • Report



Live Demo

What we built :



Complete SE Lifecycle

Feasibility → Design → Implementation → Testing



User-Validated Solution

27 student survey, real pain points addressed



Secure Platform

JWT authentication, Spring Security integration



Core Features Delivered

Expense tracking, budgets, alerts, analytics



Scalable Architecture

Modular design, clean code, well-documented



Full-Stack Application

Spring Boot backend + responsive frontend

How we built it :

Our Development Journey

Learning Resources

- YouTube tutorials (Spring Boot crash course)
- Official Spring documentation
- Baeldung articles & guides

Research & Inspiration

- GitHub repositories analysis
- Similar project architectures
- Best practices exploration

AI Assistance

- Claude for debugging & optimization
- Code review and suggestions
- Architecture design validation

Problem Solving

- JWT token configuration challenges
- H2 database schema setup
- CORS and security configurations

Iteration & Testing

- Continuous integration testing
- Bug fixing and refinement
- User feedback incorporation

Collaboration

- Pair programming sessions
- Code reviews between team
- Professor guidance & feedback

Future Vision – v2.0 Roadmap



High

Mobile Apps

Native iOS & Android applications



High

Cloud Sync

Multi-device data synchronization



Medium

Receipt OCR

Automatic expense entry via photo



Medium

Shared Budgets

Roommate expense tracking



Low

Multi-Currency

International student support



Low

AI Predictions

Smart spending forecasts

Thank You

Questions & Discussion

Software Engineering Project - Fall 2025