FinaNote: Simple Expense Tracker for Students

Feasibility Report

Omar Ait Said Ahmed Khalil El Atri

October 2025

Instructor: Pr. Imane Fouad

Contents

1 Technical Feasibility

1.1 Tools and Technologies

With the transition to a more robust and maintainable architecture, FinaNote now uses a modern Java-based web stack. Table 1 summarizes the updated technologies.

Component	Technology/Tool
Programming Language	Java 21
Backend Framework	Spring Boot 3.2.0
Security	Spring Security, JWT Authentication
Database	H2 (in-memory, fast, ideal for development/testing)
Frontend	HTML, CSS, JavaScript
Build Tool	Maven
IDE	IntelliJ IDEA / VS Code

Table 1: Updated Technology Stack for FinaNote

1.2 Skills Assessment

The team previously relied on Python and Flask, but after reassessment and based on project scope, the switch to Java/Spring Boot was made. Table 2 shows the updated skill evaluation.

Skill	Team Proficiency	Required?
Java Programming	Strong	Yes
Spring Boot Basics	Currently learning (but well-documented)	Yes
H2 Database & SQL	Some fundamentals	Yes
HTML/CSS/JS	Learning needed	Yes
JWT Authentication	Learning required	Yes (core to project security)

Table 2: Updated Skills Assessment

1.3 Technical Challenges

- Learning core Spring Boot concepts (controllers, services, repositories).
- Implementing secure JWT-based authentication with Spring Security.
- Designing a consistent REST API for expenses, users, and budgets.
- Connecting and configuring the H2 in-memory database.

Conclusion: The project remains technically feasible. Spring Boot provides a highly structured ecosystem with extensive documentation and examples, the tools are stable, modern, and suitable for a scalable student-oriented application. The team can rely on strong Java fundamentals to achieve the required features.

2 Market/User Feasibility

2.1 Target Users

FinaNote is aimed at **university students and young adults (ages 18–30)** who:

- Need a simple way to track daily expenses.
- Prefer lightweight tools without ads or forced registration.
- Care about privacy and security.
- Have limited or fluctuating budgets.

2.2 User Survey Results

The survey (27 respondents) validated strong interest in a simple, secure, and registration-free expense tracker. The data remains unchanged from the earlier analysis.

Survey Period: October 2025
Sample Size: 27 respondents
Method: Anonymous Google Forms

2.2.1 Key Findings

Current Tracking Behavior Figure 1 shows inconsistent tracking behaviors, highlighting the need for simple digital tools.

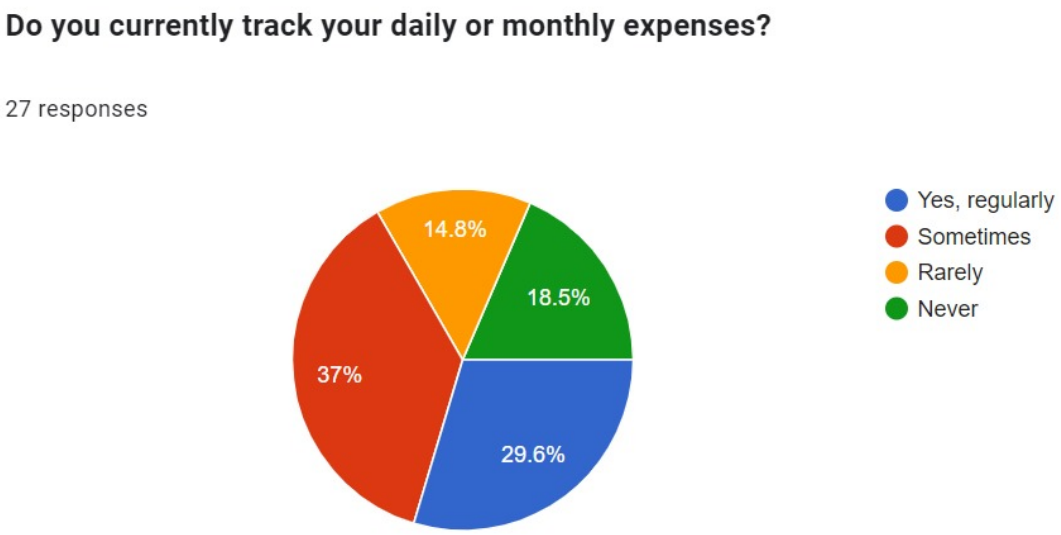

Figure 1: Current expense tracking behavior among surveyed students

Tracking Methods Figure 2 shows that most students rely on mental tracking or paper, confirming the lack of effective digital alternatives.

Pain Points Figure 3 indicates main challenges: forgetting to track, lack of time, and privacy concerns.

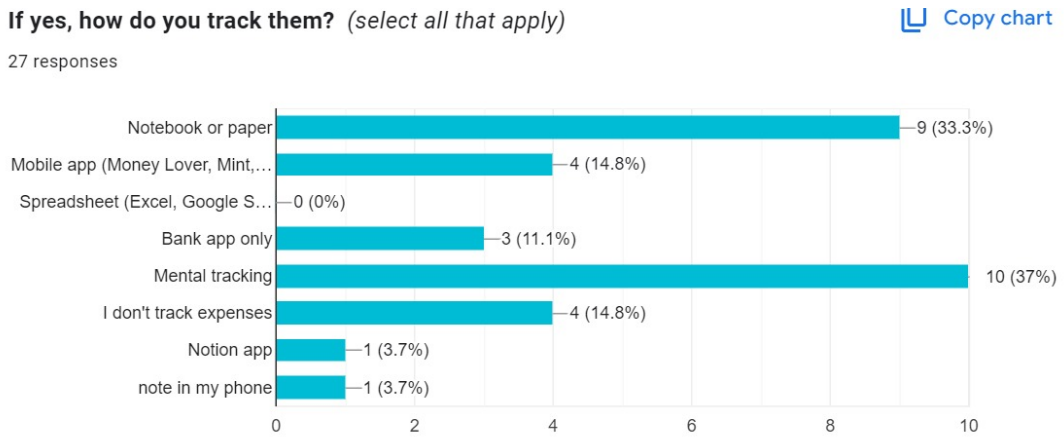

Figure 2: Current methods used by students to track expenses

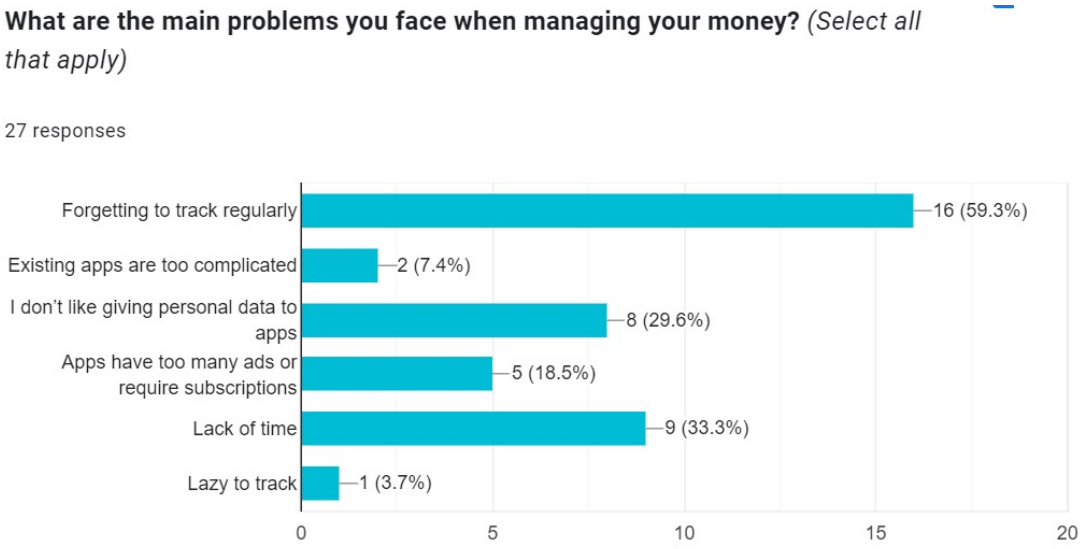

Figure 3: Main problems faced by students when managing money

Interest in FinaNote Figure 4 shows strong interest: more than half would definitely use the app.

Valued Features As shown in Figure 5, students want budgeting, simple expense entry, and category totals.

2.3 Competitive Analysis

Conclusion: Students want simple budgeting tools that avoid complexity, ads, and privacy risks. FinaNote meets all these expectations.

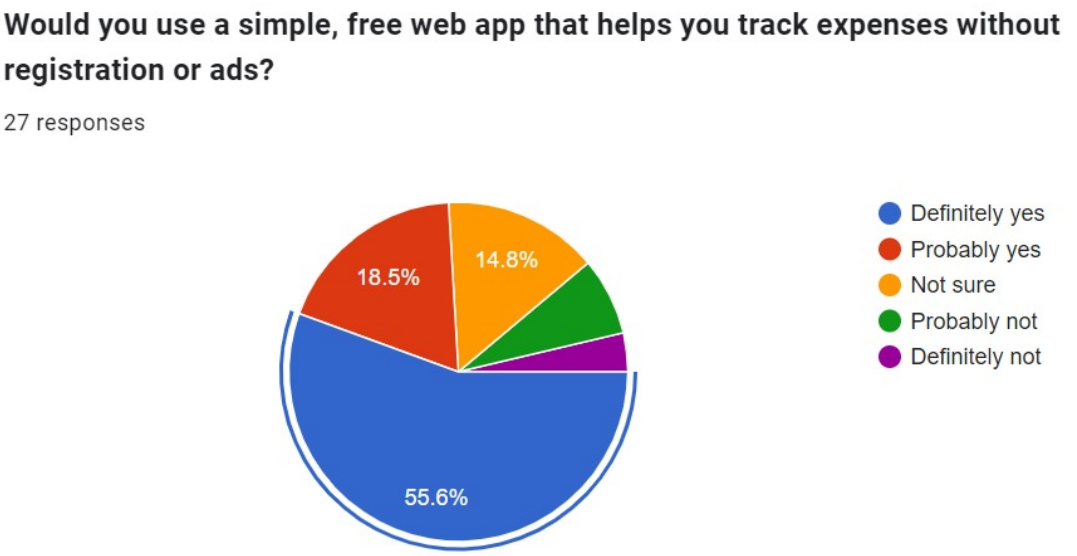

Figure 4: User interest in a simple, free, registration-free expense tracking app

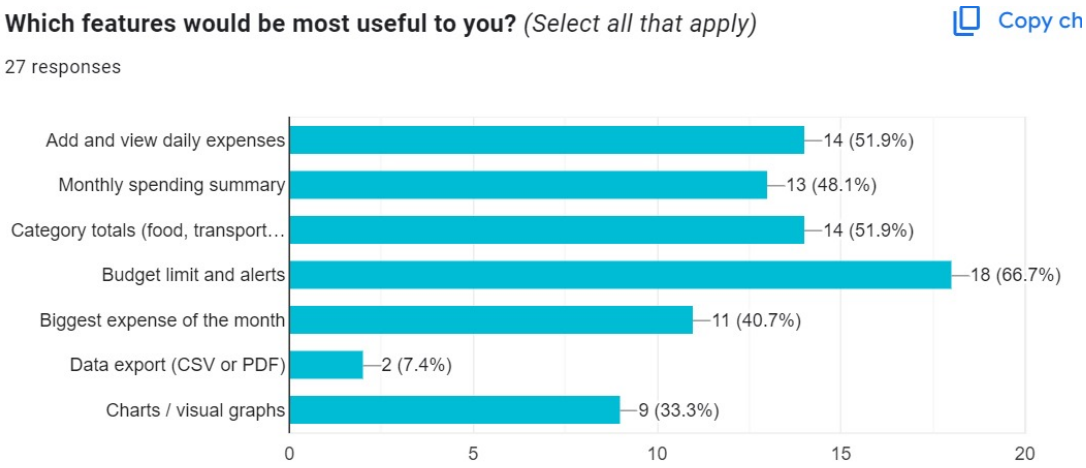

Figure 5: Most useful features according to surveyed students

3 Schedule Feasibility

Project Timeline: October 7, 2025 – December 31, 2025 (approximately 12 weeks)
Team: 2 members
Total Estimated Effort: ~140 hours

The extended deadline allows for a significantly more structured and less compressed development cycle. This provides room for deeper testing, security hardening, improved UI/UX design, and optional feature enhancements.

3.1 Key Milestones

- **October 7–18:** Spring Boot learning, project setup, security foundations
- **October 19–November 10:** Core backend development (expenses, categories, budgets)
- **November 11–20:** User authentication and JWT security integration

Competitor	What They Offer	Gap/Opportunity for FinaNote
Money Lover	Feature-rich, cloud sync, mobile app	Overly complex, contains ads, requires registration
Mint	Automatic tracking via bank linking	Privacy concerns, banking integration not suitable for students
Excel/Google Sheets	Flexible manual tracking	No automation, no budgeting tools
Paper notebook	Simple and private	No backup, no analytics or summaries

Table 3: Competitive Analysis

- **November 21–December 5:** Frontend UI with Thymeleaf + dashboards
- **December 6–20:** Testing, refinement, optimizations, additional features
- **December 21–31:** Final polishing, documentation, report, presentation

3.2 Task Breakdown

Phase	Period	Tasks	Hours
Setup & Learning	Oct 7–18	Spring Boot basics, project structure, H2 setup, Spring Security intro	25h
Core Backend Features	Oct 19–Nov 10	Expenses CRUD, budgets, categories, REST API	40h
User Authentication	Nov 11–20	JWT login, secured endpoints, roles	20h
Frontend Development	Nov 21–Dec 5	Thymeleaf pages, forms, dashboard, charts	30h
Testing & Optimization	Dec 6–20	Unit tests, integration tests, UI fixes, performance	20h
Final Preparation	Dec 21–31	Documentation, slides, rehearsal	10h

Table 4: Updated Development Task Breakdown (fits page)

3.3 Project Timeline (Gantt Chart)

Conclusion: With the extended schedule, the project becomes highly feasible. The team now has sufficient time for a complete and polished implementation, including authentication, UI refinement, testing, and thorough documentation. The new timeline significantly reduces technical risk and ensures a higher-quality final product.

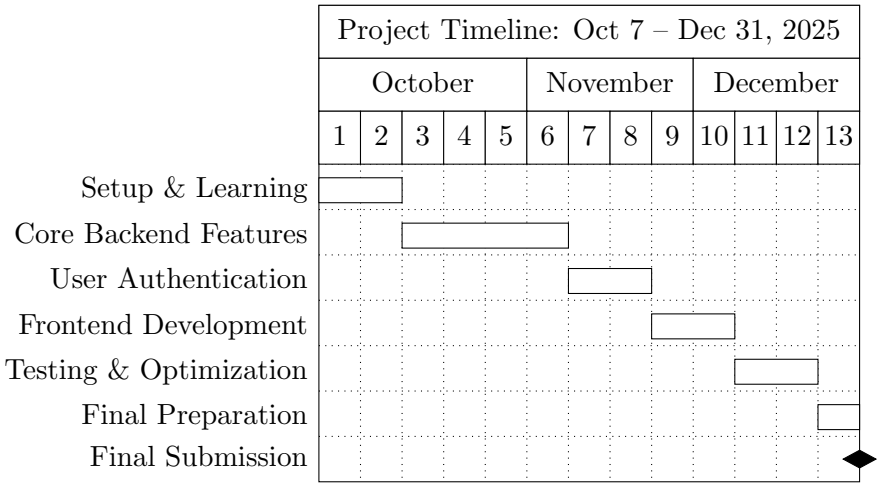

Figure 6: FinaNote Development Timeline (Perfect A4 Portrait Fit)