FlowCounts Project Proposal

1.2 Motivation

We are motivated to embark on this project because many accounting chores, such as tracking bona fide expenses, invoicing, and financial reporting, can sometimes get monotonous when done manually, more often with slips, omissions, or errors-and delays. We want to employ technology in streamlining the processes, minimizing human errors, and making financial management easier for small businesses and individuals. Our shared passion for both finance and software development pulled us toward exploring the project. More than just us working on this to fulfill our academic credit, this gives us a chance to gain hands-on experience in potentially what our future careers will be. We want to show how this kind of automation and data visualization can help small businesses and people to make better financial decisions in real time.

1.3 Statement of the Problem or Need

- The project hopes to face several issues in accounting:
- Errors and inefficiencies in manual bookkeeping.
- Lack of transparency and immediate access to financial records.
- Drawing up useful reports for persons-in-interest and decision-making.
- Difficulties in recognizing and analyzing expenses and revenues.
- Security when it comes to handling sensitive data with finance.

FlowCounts wants to create a simple-to-use platform that would automate recordkeeping, generate meaningful financial insights, and ensure the accuracy and security of data.

By programming real-time dashboards and logical tools, our web-based application FlowCounts will allow users to view their financial performance in an instant. This creates a less delayed, transparent environment for every transaction, expense, and invoice, that can be tracked efficiently.

1.4 Objectives

Other than resolving the aforementioned problems, FlowCounts should:

- **Objective 1:**Provide data visualization tools so users may hear cash flow trends.
- **Objective 2:**Implement access control with different roles (accountants, business owners, employees).
- **Objective 3:**Provide authentication and encrypt the data for security purposes.
- **Objective 4:**Allow scaling, whereby the platform will be available to scale up for individuals as well as companies.
- **Objective 5:**Offer a modern and intuitive web interface that requires minimal learning by users.
- **Objective 6:** Integrate automated report generation using charting libraries to provide instant summaries of income, expenses, and balance sheets.

1.5 Methodology

Our approach uses Agile development with weekly iterations.

Approach: We design and develop iteratively with React for the frontend and Django REST for the backend.

Task division:

- Caroline – Develop the backend API and design the database.
- Ali – Design the frontend interface and user experience.
- Brenden – Handle authentication, testing, and set up deployment.

Tracking progress: We use GitHub Projects (Kanban board), hold weekly stand-up meetings, and conduct milestone reviews.

Reports: We provide weekly progress updates, a mid-project status update, and a final presentation/demo.

Meetings: We have weekly team meetings and hold extra syncs when issues come up.

Milestones:

Phase	Due Date	Deliverable Focus	Results

Sprint 1- Login Module	October 2nd	build and test authentication system.	Functional Log-In / Log-Out module with role-based access and encrypted credentials.
Sprint 2- Chart of Accounts Module	October 14th	Create and manage chart-of-accounts records.	Account creation/editing forms, DB schema finalized, and account listing page.
Sprint 3 – Journalizing and Ledger module	October 22nd	Implement journal entry and ledger posting features.	Transaction entry interface, debit/credit validation, linked ledger view.
Sprint 4- Financial Reports	November 4th	Generate financial statements.	Income Statement, Balance Sheet, and export/download functions.
Sprint 5- Ratios and Dashboards	November 18th	Visual analytics and KPIs.	Ratio calculations, graphical dashboard, and final polish for all modules.
Final Project Presentation and Methodology Docs	December 7th	Compile and deliver final materials.	Group presentation, methodology documentation, and user/developer guides.

Tools: We use GitHub, VS Code, PostgreSQL or SQLite, Postman, Figma for UI mockups, and Slack or Teams for communication.

1.6 Knowledge Areas Needed for the Project

- Web development with React and Django REST.

- Database design using PostgreSQL or SQLite.
- Authentication and authorization with JWT tokens.
- Best practices in software engineering, including version control and testing.
- Basic accounting principles including expenses, revenue, and cash flow.
- Collaboration and project tracking using Agile methodologies and GitHub Projects.

1.7 Project Deliverables and Beneficiaries

Deliverables:

- A functional web application that includes both frontend and backend.
- Documentation such as a user manual, developer guide, and database schema.
- Test cases and evaluation reports.
- A final presentation/demo.

Beneficiaries:

- Small business owners.
- Freelancers managing their personal finances.
- Accounting students seeking a learning tool.
- Potentially larger organizations needing cash flow analysis.
- Small accounting firms or startups seeking lightweight financial management solutions.
- Academic institutions using it as a teaching tool in IT or Accounting courses.

1.8 History

There are similar accounting systems available, like QuickBooks and FreshBooks. However, these are often expensive, complicated, or too feature-rich for small teams. FlowCounts stands out by prioritizing simplicity, affordability, and features designed for small businesses and individuals.

1.9 Proprietary Information and Confidentiality Requirements

- User credentials and financial data must stay confidential.
- We require secure authentication and encrypted storage.
- No user financial data will be shared with third parties.

1.10 Required Facilities

- Development machines must have Python, Node.js, and necessary databases installed.
- We need access to GitHub for version control.
- Optional cloud services like Heroku, AWS, or Azure may be used for deployment.

1.11 Project Assumptions and Constraints

Assumptions:

- Users have internet access, basic accounting knowledge, and a secure hosting environment.

Constraints:

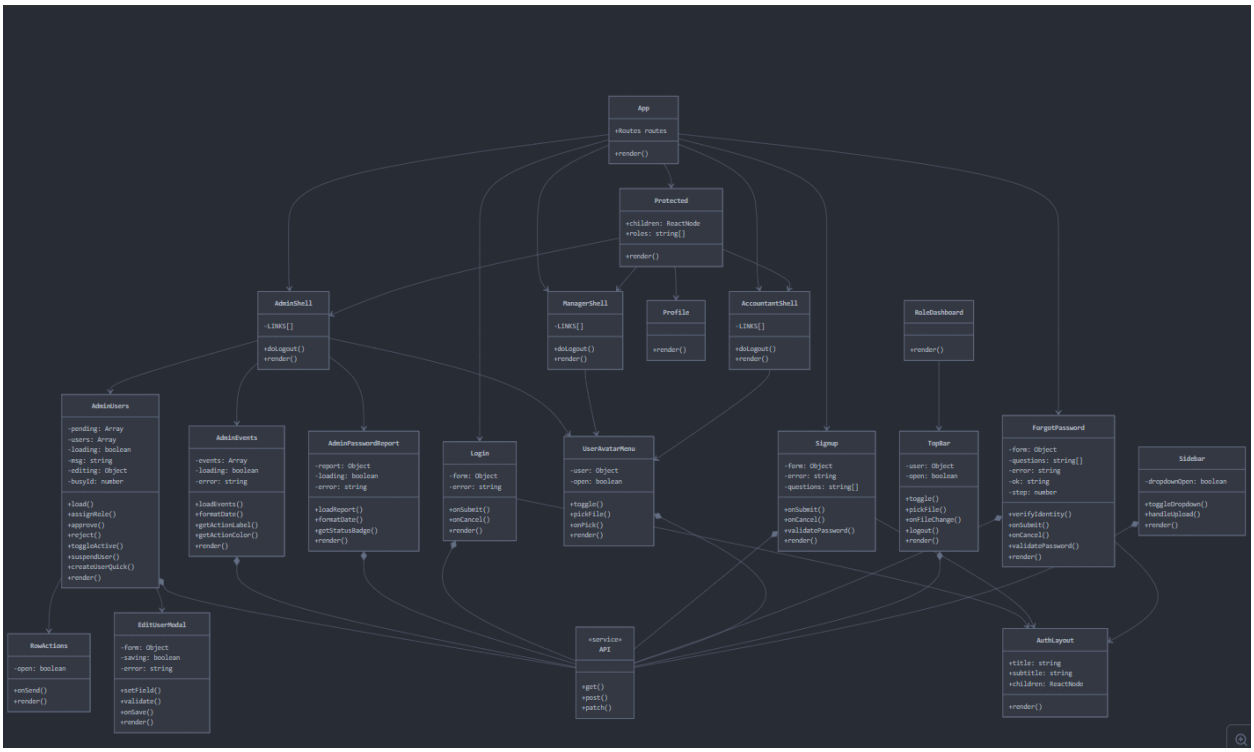
- We have limited time (one semester), a small team size, dependence on third-party libraries, and testing will be limited to sample data instead of real clients.
- The system's performance may vary depending on server hosting and internet speed.
- Integration with external APIs (e.g., email or payment gateways) will be limited by available free-tier services.

1.12 Major Stakholders

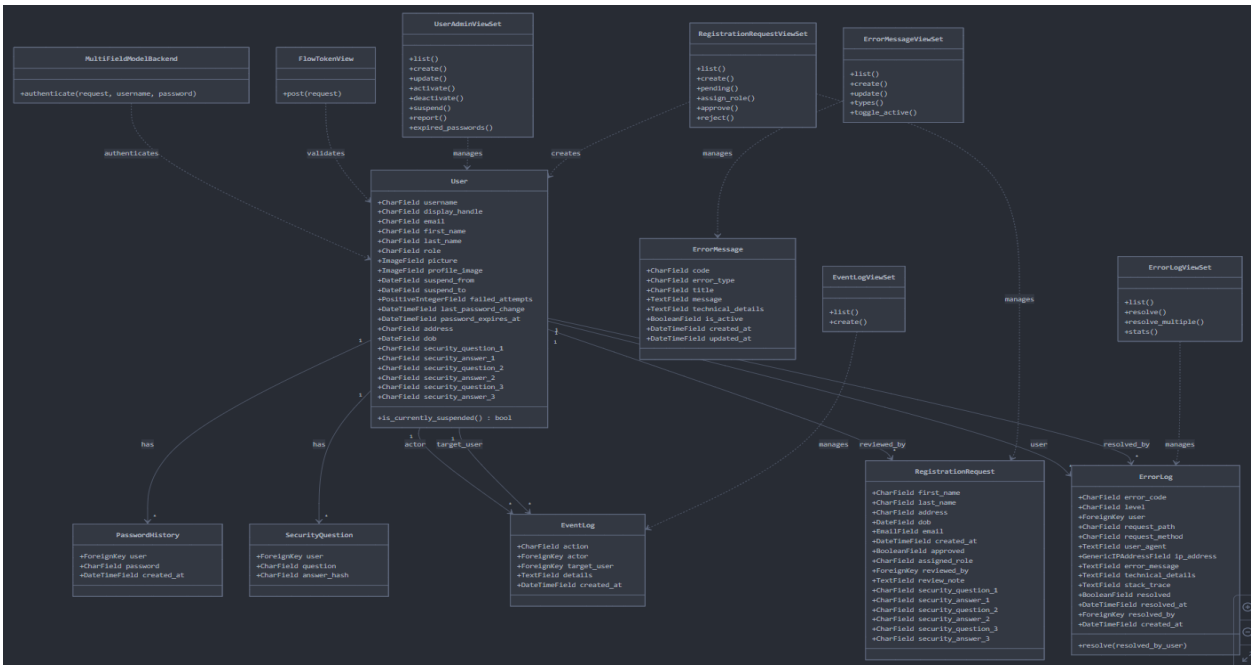
- Team members, who are developers.
- Instructor and advisors, providing academic oversight.
- End-users, including business owners and freelancers

1.13 Class Diagram

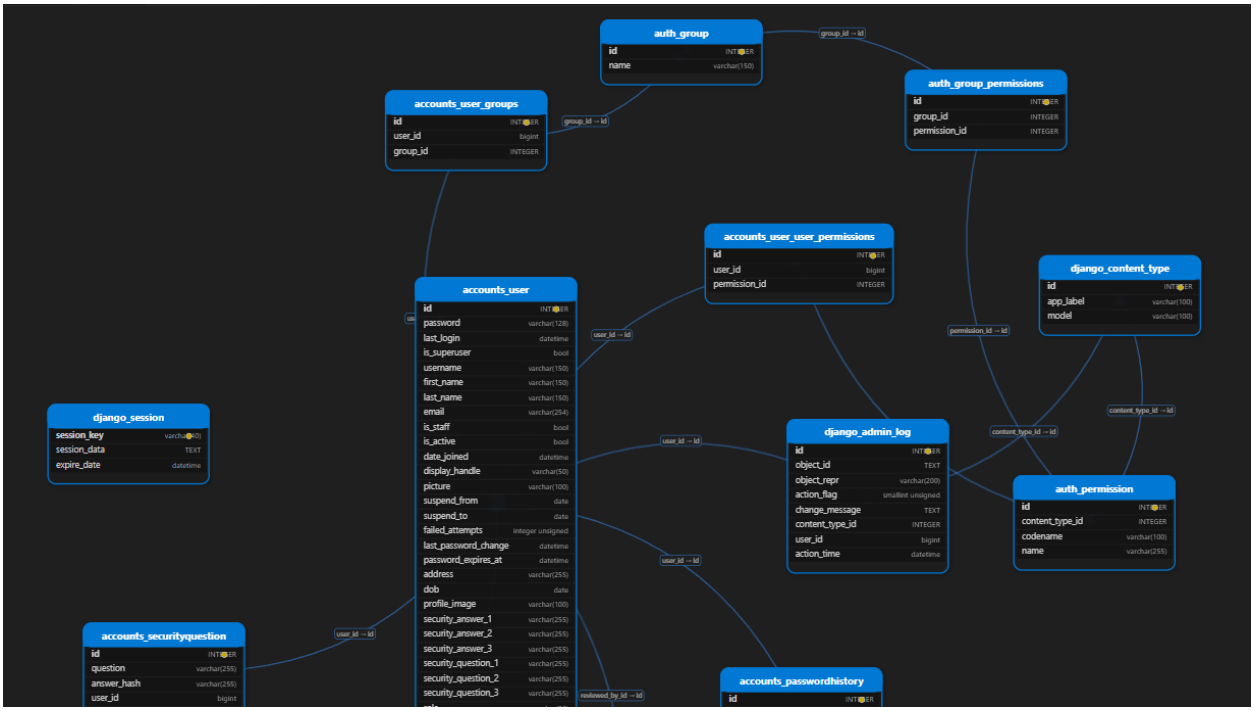
- Frontend classes UML Diagram:

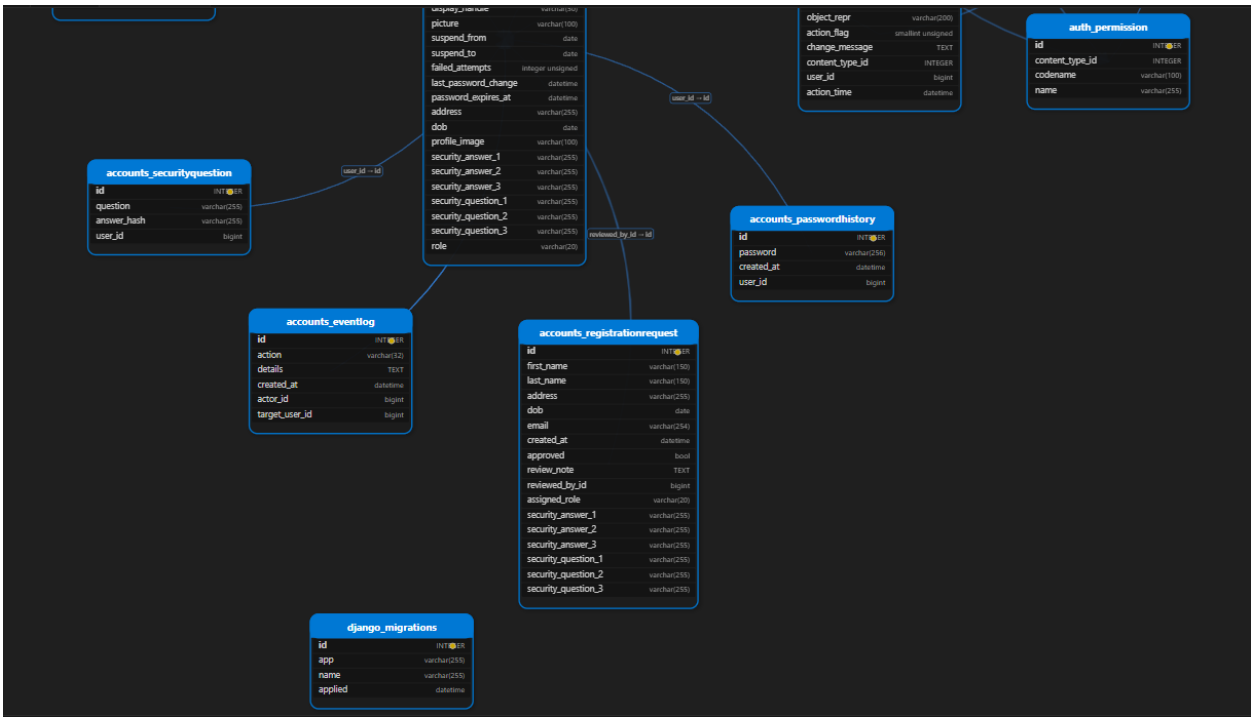


- Backend classes UML Diagram:



1.14 Database Diagram





1.15 Project schedule

This project is divided into 5 different sprints before the final submission. Each sprint includes defined tasks, start and end dates, responsible team members, and dependencies that connect one phase to the next. The schedule is presented below in Gantt chart form, followed by a detailed summary table.

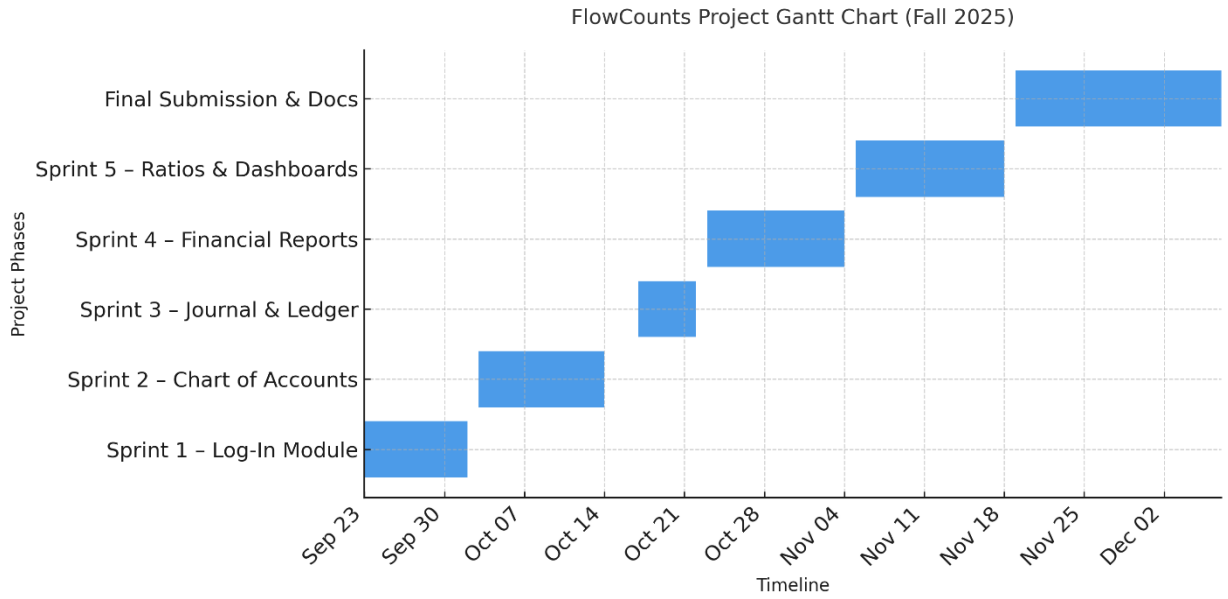


Figure 1. FlowCounts Project Gantt Chart (Fall 2025)

Phase / Task	***Description***	***Start Date***	***End Date***	***Dependencies / Relationships***	***Responsible Members***
Sprint 1 – Log-In Module	Develop and test the authentication system with role-based access.	Sept 23 2025	Oct 2 2025	Foundation for all later modules; must be completed before database integration.	Brenden (auth), Ali (API), Caroline (UI)
Sprint 2 – Chart of Accounts Module	Create and manage chart-of-accounts records and link them to the database schema.	Oct 3 2025	Oct 14 2025	Depends on successful log-in and database setup.	Ali (DB/API), Caroline (front-end)
Sprint 3 – Journalizing & Ledger Modules	Implement journal entry and ledger posting with debit/credit validation.	Oct 17 2025	Oct 22 2025	Requires Chart of Accounts Module to exist.	Ali (back-end), Brenden (validation/tests)
Sprint 4 – Financial Reports	Generate income statements and balance sheets	Oct 23 2025	Nov 4 2025	Requires journal and ledger data;	Brenden (tests), Caroline (UI)

	with export functions.			feeds into ratios and dashboards.	
Sprint 5 – Ratios & Dashboards	Build KPI ratio calculators and graphical dashboards.	Nov 5 2025	Nov 18 2025	Depends on finalized reports; prepares for final presentation.	Caroline (front-end), Ali (API data feeds)
Final Submission & Documentation	Prepare final video presentation, methodology documents, and user/developer guides.	Nov 19 2025	Dec 7 2025	Culminates all sprints; requires completion of every prior task.	All Members

The Gantt chart and table show how each sprint connects to the next as the project moves forward. Every module builds on the previous one, helping us stay organized and keep steady progress toward completing all deliverables.

1.16 Primary Contact

- Brenden Horne (Team Leader) - bhorne16@students.kennesaw.edu
- Caroline Varner - cvarner5@students.kennesaw.edu
- Ali Dabdoub - adabdoub6@students.kennesaw.edu