Software Project Management Plan

Financial Budgeting App: ClariFI

9/26/2025

Team Members

Abdul Mohammed
Haley Nilsen
Kaleb Flores Lopez
Sid Nsude

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	9/26/2025	Initial release
V2.0	12/10/2025	Final Release

Document Storage

This document is stored in the project’s repository at:
https://github.com/snsude/Commerce-Bank-Financial-App/tree/main

Document Owner

Abdul Mohammed is responsible for developing and maintaining this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

ClariFi is a financial banking application that empowers users to take control of their financial future through goal-oriented tracking and AI assistance. This web-based application is being developed as part of UMKC's Software Engineering Capstone class in partnership with Commerce Bank. ClarFi will serve as a modern financial management platform that bridges traditional banking services and personalized financial planning, offering users an intuitive interface to monitor their progress toward achieving their financial aspirations. The project leverages artificial intelligence integration and responsive web design to deliver a secure, user-friendly experience that meets the evolving needs of today's digital budgeting customers.

The purpose of this project is to streamline personal financial management and goal achievement by providing users with a centralized platform that transforms complex financial data into actionable insights and clear progress indicators. ClariFi actively supports users in reaching their financial objectives through intelligent monitoring and AI-powered assistance. This project aims to enhance user engagement with their financial health while reducing the complexity and friction typically associated with personal finance management.

The scope of this project compasses the complete design, development, testing, and deployment of a financial web application for Commerce Bank that will serve as a comprehensive financial goal-tracking platform. The application will provide secure user authentication and account management capabilities, robust budget tracking functionality, and sophisticated data visualization tools that present financial information in clear, actionable formats. A key component of the scope includes the integration of an AI-powered chatbot that will offer personalized assistance, financial insights, and recommendations to enhance the user experience. Additionally, the application will feature a comprehensive tutorial system designed to onboard new users and ensure they can effectively utilize all available features. The project scope explicitly does not include mobile app development for iOS or Android platforms, integration of third-party plugins beyond pre-approved libraries and frameworks, or support for international customers and their specific regulatory requirements at this initial stage of development.

1.2 Goals and Objectives

Goals and objectives define expected project outcomes. Goals are broad and inspirational. Objectives are narrow and measurable.

Project goals generally related project outcomes to business objectives (reduced cost, increased revenue, improved quality, etc).

A well-worded objective is SMART: Specific, Measurable, Attainable/Achievable, Realistic and Time-bound.

Partial Example

Project goals:

1. To deliver a secure, intuitive, and AI-enhanced web application that modernizes personal financial management for Commerce Bank customers.

2. To increase user engagement with financial planning by providing clear visualization of goals and progress.

3. To reduce the perceived complexity of budgeting and financial tracking through an assisted, goal-oriented experience.

Project objectives:

1. By 12/12/2025, successfully deploy a fully functional web application prototype incorporating all core features: user authentication, dashboard, budget tracker, goal tracker, AI chatbot, and data visualizations.

2. Achieve a successful user acceptance test (UAT) with the project sponsor, with 100% of critical functional requirements met and no high-priority bugs outstanding.
3. Complete and submit all required project deliverables (code, documentation, presentation) by the final project deadline.

1.3 Project Deliverables

This section lists the outputs of the project that are delivered to the customer.

Partial Example

The following items will be delivered to the customer on or before 12/12/2025:

1. Source Code – Completed and well documented codebase, for frontend, backend and database all hosted on a GitHub repository.
2. Technical Documentation – Documents like the architecture/design, test report, requirements, test plan, and project charter all available in the GitHub repository.
3. User Documentation & Tutorial – The user guide document along with a mp4 file showcasing the project and it's features.
4. Final Report & Presentation – A detailed report about the project's design, implementations, challenges and outcomes along with a presentation for a live demo.

1.4 Assumptions and Constraints

Assumptions are conditions, usually outside the control of the project team, that are taken for granted. Project plans (i.e. estimates) typically depend on certain assumptions being true. Assumptions that turn out to be false, may jeopardize project success. In order to reduce project risk, the project manager may elect to validate certain assumptions as part of the risk management process.

This is also a good place to document verbal promises or assurances given to you.

Constraints are limits or restrictions on freedom. Projects may have technical as well as non-technical constraints. Priorities for schedule and budget can impose non-technical constraints on a project. Restrictions on programming language or delivery platform are examples of technical constraints that limit design and implementation options.

Partial Example

Assumptions:

1. The project sponsors at Commerce Bank will be available for scheduled reviews and provide timely feedback.
2. All team members will maintain consistent communication and participation throughout the project timeline.
3. Necessary software tools and frameworks (e.g., React, FastAPI, LLM API access) will remain available and functional for the project duration.
4. The scope and primary requirements for the project will remain stable after the initial planning and mid-semester review.

Constraints:

1. Must complete the project within the academic semester timeline (by 12/12/2025).
2. Must operate with a $0 budget for software, services, and personnel.
3. Must implement an LLM (Large Language Model) integration within the web application.
4. Must use approved modern web technologies (e.g., React, FastAPI, PostgreSQL).
5. Must adhere to the University's academic and project submission deadlines

Note, the following is not a reasonable assumption for inclusion in this section: "We assume that our group has the necessary skills and knowledge needed to complete the project." This might be something you are taking for granted, but it is not something worth documenting in the project plan. The assumptions you want to list here are those that are outside your control. Once the development team is established, it is their responsibility to possess or develop the skills and knowledge needed to complete the project. If there is a concern that the existing team doesn't have the skills and knowledge needed to complete the project successfully, add it as a risk and develop a plan for mitigating the risk.

1.5 Schedule and Budget Summary

The schedule summary shows start and end dates for high-level activities ending in major milestones or deliverables. Milestones are major events in the project life cycle that are used to measure progress.

A Gantt chart is an excellent tool for visualizing the start and stop dates of major scheduled activities.

Schedule Summary

- **Weeks 1–2: Project Planning & Design** – Requirement analysis, design wireframes, draft documents, database schema, research.

- **Weeks 3–5: Core Development – Sprint 1** – Frontend/backend setup, user authentication, basic dashboard, AI agent foundation.

- **Weeks 6–8: Core Development – Sprint 2** – Implement goal tracking, integrate AI agent, connect core features, testing.

- **Weeks 9–10: Core Development – Sprint 3** – Polish features, implement data visualizations, comprehensive testing & bug fixing.

- **Weeks 11-12: Polish & Present** – Final deployment, documentation finalization, presentation preparation

The budget summary shows total project cost, possibly broken down into separate categories for such things as salaries, equipment, travel, overhead, etc.

Budget Summary

- **Software & Tools = $7**
- **Personnel = $0**
- **Total Project Cost = $7**

1.6 Success Criteria

Success criteria spell out what has to happen before the project can be considered a success. Having explicit success criteria serves two purposes. First, during a project success criteria help to focus attention on what is important. Second, at the conclusion of a project (project closure) success criteria are used to assess whether or not the goals and objectives of the project have been achieved.
To be effective in both of these endeavors, success criteria must be defined in a way that is both quantifiable and verifiable.

- **Project Completion:** All final deliverables are submitted and the final presentation is delivered on or before the scheduled date of 12/12/2025.
- **Functional Requirements:** 100% of the agreed-upon critical/core functional requirements are implemented and operational in the final prototype.
- **Sponsor Acceptance:** The project sponsor (Commerce Bank representative) formally accepts the final deliverables after the demonstration and review.

- **AI Agent Performance:** The integrated AI chatbot successfully responds to user queries related to budgeting and goals with relevant, accurate, and helpful information.
- **Usability:** The application is judged to be intuitive and easy to navigate during the final user demonstration, with a completed onboarding tutorial.

1.7 Definitions

- This section should define potentially unfamiliar or ambiguous words, acronyms and abbreviations.
- **AI (Artificial Intelligence):** The simulation of human intelligence processes by machines, especially computer systems.
- **LLM – Ollama 3 :** A type of artificial intelligence trained on vast amounts of text data to generate human-like language and answer questions.
- **FastAPI:** A high-performance web framework for building APIs with Python.
- **PostgreSQL:** An open-source object-relational database system.
- **React:** A JavaScript library for building user interfaces, specifically single-page applications.
- **Tailwind CSS:** A CSS framework for rapidly building custom user designs.
-

1.8 Evolution of the Project Plan

This section describes plans for updating the project plan throughout the project.

~~This project plan is a living document. It will be reviewed and updated at the beginning of each major project sprint (approximately every 3 weeks) to reflect the current project status, revised estimates, and any newly identified risks or scope changes. Major changes following the mid-semester review with the sponsor will be incorporated immediately. Version history will be maintained within the document.~~

Final Update: This project plan was actively maintained and updated throughout the project lifecycle, with versions archived at key sprint boundaries and following the mid-semester review. All planned evolution of the document is now complete. This final version serves as the archival record of the planned and executed management approach for the ClariFi project.

2 Startup Plan

2.1 *Team Organization*

This section explains project roles and the authorities and responsibilities associated with these roles. Lines of communication, authority and reporting relationships are often shown with an org chart. If development team is known, actual names can be associated with roles.

Partial Example

Project Manager:	The project manager is responsible for creating the project plan (with input from those doing the work), managing risks, running the weekly team meeting and providing monthly status reports to senior management.
Programmers (3):	Programmers are primary responsible for coding and unit testing modules. They are also expected to take part in architecture planning and review meetings.

2.2 *Project Communications*

This section contains the project communications plan. The communications plan describes how information is gathered and distributed.

- **Internal Team Communcation:** communication will take place through Microsoft Teams, we will have a weekly meeting to discuss /provide updates on progress
- **Jira Dashboard:** will have tasks assigned to each team member

2.3 *Technical Process*

This section describes the software development methodology or conventions the team agrees to live by. When following an organization standard process, this section will refer to the standard process and state any deviations that are planned for this project. In the absence of an organization standard process, this section will define planned phases, entry and exit criteria for each phase, major milestones, workflows, and other aspects of the proposed development process.

- **Sprints**
- **Sprint Reviews**
- **Version Control**
- **Weekly Meetings**

2.4 *Tools*

This section specifies the development tools the team will be using to perform their work.

Partial Example

- **Version Control**: GitHub
- **Frontend**: React, Tailwind CSS
- **Backend**: FastAPI, Python
- **Database**: PostgreSQL
- **LLM** : Ollama llama3
- **Communication:** Microsoft Teams
- **Design & Prototyping**: Figma
- **Automated testing** – unit tests will be implemented with the JUnit testing framework.

3 Work Plan

3.1 Activities and Tasks

- The ClariFi project follows a structured work breakdown approach to identify, assign, and track all development and testing tasks. Tasks are grouped by major system components, including authentication, database integration, dashboard functionality, and testing. Each task records ownership, effort, and scheduling details to ensure progress can be monitored throughout the project lifecycle.

- Depending on the task, the following attributes are recorded: task name, description, owner, estimated effort, actual effort, and planned versus actual start and completion dates.

3.2 Release Plan

3.2.1.1 Milestone 1: Requirements & Planning

- **Description:** Define system requirements, user roles, and core features for ClariFi.
- **Key Deliverables:**
    - Project scope definition
    - User roles (personal, business, sub-business)
    - Initial test specifications
- **Planned Completion:** Early Implementation Phase

3.2.1.2 Milestone 2: Architecture & Environment Setup

- **Description:** Establish backend, frontend, and database architecture.

- **Key Deliverables:**
    - Backend API structure
    - Database schema and connectivity
    - Frontend routing framework
- **Planned Completion:** Mid Implementation Phase

3.2.1.3 Milestone 3: Core Feature Implementation

- **Description:** Implement primary system functionality.
- **Key Deliverables:**
    - User authentication and login
    - Role-based dashboard routing
    - Dashboard data retrieval (summary, purchases, goals)
- **Planned Completion:** Late Implementation Phase

3.2.1.4 Milestone 4: Testing & Issue Resolution

- **Description:** Validate system behavior and resolve identified defects.
- **Key Deliverables:**
    - Automated authorization tests
    - Database connectivity tests
    - Incident resolution (login and dashboard routing issues)
- **Planned Completion:** Testing Phase

3.2.1.5 Milestone 5: Final Release

- **Description:** Deliver stable and tested version of ClariFi.
- **Key Deliverables:**
    - Final test report
    - Resolved incident documentation
    - Ready-to-deploy application
- **Planned Completion:** Product Release Phase

3.3 Iteration Plans

3.3.1 Iteration 1: Environment Setup & Authentication

- **Tasks Completed:**
  - Configure backend and frontend project structure
  - Establish PostgreSQL database connection
  - Implement user login authentication
- **Deliverables:**
  - Working login page
  - Verified database connectivity
- **Outcome:**
  Core system access established

3.3.2 Iteration 2: Role-Based Dashboards

- **Tasks Completed:**
  - Implement role-based routing logic
  - Create dashboards for personal, business, and sub-business users
  - Validate dashboard navigation after login
- **Deliverables:**
  - Role-specific dashboards
  - Correct post-login routing
- **Outcome:**
  Users are directed to the appropriate dashboard based on role

3.3.3 Iteration 3: Data Retrieval & Testing

- **Tasks Completed:**
  - Implement dashboard data endpoints (summary, recent purchases, goals)
  - Add unauthorized access tests
  - Add database connectivity tests
- **Deliverables:**
  - Secured API endpoints
  - Automated test cases

- **Outcome:**
  System behavior verified under authorized and unauthorized conditions

3.3.4 Iteration 4: Bug Fixes & Final Validation

- **Tasks Completed:**
  - Fix login and dashboard routing issues
  - Re-run failed test cases
  - Validate incident resolution
- **Deliverables:**
  - Updated incident report
  - Final test report
- **Outcome:**
  Stable system ready for release

3.4 Budget

The project budget represents the projected and actual cost of developing ClariFi over time. Since this project was primarily developed using open-source tools and student labor, monetary costs were minimal.

- **Cloud Database Hosting (Render):** $7
- **Development Tools:** $0 (open-source)
- **Testing Tools:** $0 (open-source)

Total Project Cost to Date: $7

At the current stage of the project, no additional costs are anticipated to complete development.

4 Control Plan

4.1 Monitoring and Control Schedule
The following reviews and reports will be used to track progress, identify issues early, and ensure alignment with project objectives.

Weekly (60 mins) – Team Sync Meeting. Led by the Project Manager, all project participants report on task status, progress against the sprint plan, and surface any blockers or resource needs.

Bi-Weekly (30 mins) – Managerial Review. Project Manager Abdul Mahammed presents a concise update on budget, timeline, risks, and deliverables to the project sponsor and key stakeholders.

[Date, e.g., Mid-August] – Requirements Review. A formal meeting to walk through and gain stakeholder sign-off on all functional and non-functional requirements documents, ensuring a shared understanding before design begins.

[Date, e.g., Early September] – Architecture Sign-off. A technical review where the solution design and system architecture are presented and approved by the lead architect and technical stakeholders.

[Date, e.g., Late November] – User Acceptance Test (UAT) Readiness Review. A gatekeeper meeting to confirm that all exit criteria for development and system testing are met, and the product is ready for formal validation by the end users.

[Date, e.g., Mid-December] – Project Closure & Retrospective. A final review to confirm all deliverables are accepted, archive project data, and document lessons learned.

A written Summary Status Report will be distributed by the Project Manager every Sunday evening. This report will include accomplishments from the past week, planned activities for the coming week, an updated milestone tracker, and any active risks or issues.

4.2 Project Measurements

Key product and process measures will be tracked to support objective decision-making and provide a basis for estimating future projects. The Project Manager is responsible for the collection and reporting of these metrics.

Phase	**Measurement**	**Source**
Planning & Design	Record estimated effort (in hours/days) for all high-level features and design tasks.	Project Manager
Implementation	Record Actual Effort for completed development tasks vs. original estimates.	Project Manager

Testing	Record the Number of Defects Found per test cycle (Unit, Integration, System).	Project Manager
Testing	Track Defect Status: Count of defects by severity (Critical, Major, Minor) and state (Open, In Progress, Resolved, Closed).	Project Manager
UAT	Track UAT Completion Rate: Percentage of test cases passed/failed by end-users.	Project Manager
Project Closeout	Archive final project data, including total actual effort, final schedule performance, and defect metrics, to the organizational repository.	Project Manager

5 Supporting Process Plans

5.1 Risk Management Plan

Clarifi faces several technical and managerial risks. The highest technical risks include **security vulnerabilities in the self-hosted authentication system**, **incorrect financial calculations**, and **integration issues between the React frontend, PostgreSQL backend, and Ollama3 LLM**. Managerial risks include **schedule slippage** due to coordination across four roles (project manager/full-stack developer, frontend developer, backend developer, and LLM developer) and **scope creep** driven by additional feature ideas for young adult and business users. Authentication and data accuracy risks are considered high priority due to their potential impact on user trust.

To mitigate these risks, the team will use proven libraries for authentication, conduct peer code reviews, and implement unit tests for budgeting and expense logic. Clear task ownership will be maintained by the project manager to reduce coordination issues. If a high-priority risk becomes a problem (e.g., a security flaw or incorrect calculations), development will pause on new features while the issue is corrected and retested. Risks will be reviewed at the end of each development iteration and updated as needed.

5.2 Configuration Management Plan

1. All Clarifi work products, including source code and documentation, will be stored in a centralized version control repository. Documents will follow the naming convention **XXXX.pdf (X's being the file name)**. Source code will be organized by component (frontend, backend, LLM integration).

2. The system requirements, project plan, and core source code will be baselined after group review at the end of each development phase. Once baselined, changes must be proposed to the group with a description, justification, and expected impact. If no objections are raised within two days, the change is approved; otherwise, it is discussed and resolved by group consensus. Baselined documents will include a change history section recording author, description, reason, and date of changes.

5.3 Verification and Validation Plan

Verification activities focus on ensuring the system is built correctly and include code reviews, unit testing of budgeting and expense calculations, and integration testing between the React frontend, PostgreSQL backend, and Ollama3 LLM in a localhost environment. Authentication and role-based access control will be specifically verified to ensure proper permissions for personal users, business users, and subusers.

Validation activities ensure the system meets user needs. This includes functional testing based on user scenarios such as creating budgets, tracking expenses, managing subusers, and viewing financial goals. End-to-end testing will confirm that Clarifi supports the intended workflows for young adults and businesses managing projects and finances.

5.4 Product Acceptance Plan

Clarifi will be considered acceptable when all core features operate as specified and meet defined quality standards. Acceptance criteria include successful user authentication, accurate expense and budget calculations, correct enforcement of user roles, and reliable interaction with the LLM features. No high-severity defects may remain open at the time of acceptance.

Acceptance will be determined through completion of documented test cases and final system demonstrations. The application must function correctly in a localhost deployment environment and meet the functional needs of both personal and business users before being approved as complete.