Software Project Management Plan

Recur

Team Merge Conflict

12/11/2025

Team Members

Curtis Lemke - Project manager

Logan George

Brendan Clouston

Viger Romo

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	9/22/2025	Initial Creation
V2.0	12/11/2025	Final Revisions

Document Storage

This document is stored in the project's Github repository at:
https://github.com/CS451-MergeConflict/ProjectDocuments

Document Owner

Curtis Lemke is responsible for developing and maintaining this document.

Table of Contents

1 Overview

1.1 Purpose and Scope

The purpose of the **Recur project** is to design and implement a web-based application that enables users to track, analyze, and manage recurring financial obligations such as subscriptions, memberships, and regularly billed services. Many users lack visibility into how much they spend on recurring charges over time. Recur addresses this problem by aggregating transaction data, detecting recurring expenses, and presenting actionable insights in a clear and intuitive interface.

The **scope** of the Recur project includes:

- User authentication and account management
- Ingestion of transaction data through third-party financial APIs

- Detection and management of recurring transactions
- Display of upcoming renewals and subscription costs
- Analytics and visualizations for historical spending
- Manual subscription entry and CSV import

Out of scope for this project are personalized financial advice, automated subscription cancellation, and real-time bank transaction processing guarantees.

1.2 Goals and Objectives

Project Goals

1. Provide users with clear visibility into their recurring financial commitments.
2. Reduce financial waste by helping users identify unused or forgotten subscriptions.
3. Deliver a modern, intuitive, and responsive user experience.

Project Objectives

1. Implement secure user authentication and persistent data storage.
2. Integrate third-party APIs to ingest transaction data.
3. Detect recurring expenses automatically and allow manual overrides.
4. Present upcoming renewals and historical analytics via dashboards.
5. Deliver a fully functional web application by the end of the project term.
6. Integrate LLM functionality to assist users with subscription optimization.

1.3 Project Deliverables

The following items will be delivered to stakeholders:

1. Source code for the Recur web application
2. Requirements Specification
3. Project Management Plan
4. Test Plan and test cases
5. Deployed application accessible via web browser
6. User documentation (help text and onboarding guidance
7. Deployed application accessible via web browser
8. Database documentation

1.4 Assumptions and Constraints

Assumptions

1. Third-party financial APIs will remain available during development.
2. Users will have access to modern web browsers.
3. Cloud hosting services will be available throughout the project.
4. Development team members will be available for the duration of the project.

Constraints

1. The application must be delivered as a web-based system.
2. Sensitive user and financial data must be securely stored and transmitted.
3. The project must be completed within the academic term.
4. Development will use the approved technology stack.
5. The project must be completed within the academic term.
6. Must use a database and hash user passwords

7. Cannot use a CMS and must use a CSS framework

8. Must include a login/register page, homepage/dashboard, user profile and settings, and two additional pages.

1.5 Schedule and Budget Summary

The project schedule consists of major milestones including requirements definition, design, implementation, testing, and deployment. Progress will be tracked using milestone completion and iteration reviews.

The project budget is constrained primarily by time and development effort, as infrastructure and tools rely largely on free or academic-tier services.

For this project we are allotted a **zero dollar budget** and must utilize free and open source tools.

1.6 Success Criteria

The Recur project will be considered successful if:

- All high-priority functional requirements are implemented
- The application successfully detects and displays recurring subscriptions
- Users can view analytics and upcoming renewals
- The system operates reliably in modern browsers
- The project is delivered within schedule constraints

1.7 Definitions

- **Recur** – The subscription management application described in this plan
- **Iteration** – A short development cycle delivering incremental functionality

- **Recurring Transaction** – A transaction that occurs on a regular schedule

---

1.8 Evolution of the Project Plan

The project plan will be reviewed and updated at the beginning and end of each iteration. Task estimates, schedules, and risks will be adjusted based on actual progress and newly identified issues.

---

2 Startup Plan

2.1 Team Organization

- **Project Manager - Curtis Lemke**
  Responsible for planning, coordination, risk management, and implementation.

- **Developers - Logan George, Brendan Clouston, Viger Romo**
  Responsible for feature implementation, testing, and documentation.

2.2 Project Communications

Project communication will occur through:

- Twice Weekly progress check-ins
  - Meet Monday and Thursday at 5:30PM over Discord

  Monday: Review outstanding issues and create sprint tickets

  Thursday: Review progress and issues in current sprint

- Version control commit history in Github

- Issue tracking and comments within the repository

- Discord messages and in-class meetings

2.3 Technical Process

The Recur project follows an **iterative and incremental development process**. Each iteration includes planning, implementation, testing, and review. Features are delivered incrementally, allowing early validation and feedback.

2.4 Tools

- **Programming Language:** TypeScript
- **Framework:** Next.js
- **Database:** Supabase (PostgreSQL)
- **Version Control:** GitHub
- **Charting:** Recharts
- **CI/CD:** GitHub Actions
- **Testing:** Unit and integration testing tools
- **Deployment:** Vercel

---

3 Work Plan

3.1 Activities and Tasks

Tasks will be identified using a work breakdown structure and tracked with:

- Task ownership
- Effort estimates
- Actual effort
- Dependencies

3.2 Release Plan

Major releases correspond to completed milestones:

1. Initial prototype
2. Subscription tracking functionality
3. Midsemester demo version
4. Analytics page and LLM integration
5. Final polished release

3.3 Iteration Plans

Each iteration focuses on a defined subset of features and concludes with a review of completed work and updated estimates.

Weekly sprints with the sprint starting and ending on Monday.

Two week sprints leading up the midsemester demo and final demo.

3.4 Budget

The project budget is tracked in terms of development effort rather than monetary cost. Cloud and API usage remains within free or academic limits.

Budget estimate: $0

---

4 Control Plan

4.1 Monitoring and Control

- Weekly status reviews
- Iteration reviews at milestone boundaries

- Continuous integration feedback

4.2 Project Measurements

Metrics collected include:

- Estimated vs. actual effort
- Number of completed features
- Defect counts during testing
- Performance and reliability indicators

---

5 Supporting Process Plans

5.1 Risk Management Plan

Key risks include:

- Third-party API changes
- Scope creep
- Schedule constraints

Risks are monitored and mitigated through early testing and incremental delivery.

5.2 Configuration Management Plan

- All artifacts stored in a Git repository
- Versioned documentation and source code
- Formal review required for major changes

5.3 Verification and Validation Plan

Verification and validation activities include:

- Unit testing
- Integration testing
- Manual acceptance testing

A separate test plan document details testing procedures.

5.4 Product Acceptance Plan

The product is accepted when:

- Core use cases function correctly
- Security requirements are met
- The application is stable and usable
- Acceptance criteria defined in the requirements document are satisfied