Project Plan: Expense Tracker Software

Contents

[Document Information 1](#_Toc191628185)

[List of Authors 1](#_Toc191628186)

[Revision History 1](#_Toc191628187)

[Introduction 2](#_Toc191628188)

[Glossary 2](#_Toc191628189)

[Milestones and Tasks 2](#_Toc191628190)

[Milestone 1: Core Functionality (Transaction Management) ✓ 2](#_Toc191628191)

[Week 1 (KW 49 – 2024) 2](#_Toc191628192)

[Milestone 2: Authentication and User Management ✓ 2](#_Toc191628193)

[Week 2 (KW 50 – 2024) 2](#_Toc191628194)

[Milestone 3: Transaction Viewing and Reporting ✓ 3](#_Toc191628195)

[Week 3 (KW 51 – 2024) 3](#_Toc191628196)

[Milestone 4: User Interface Integration ✓ 3](#_Toc191628197)

[Week 4 (KW 52 – 2024) 3](#_Toc191628198)

[Milestone 5: Settings and Data Persistence ✓ 3](#_Toc191628199)

[Week 5 (KW 8 – 2025) 3](#_Toc191628200)

[Milestone 6: Final Testing and Optimization ✓ 3](#_Toc191628201)

[Week 5 (KW 9 – 2025) 3](#_Toc191628202)

# Document Information

## List of Authors

|  |
| --- |
| Names |
| Charles Eboson |

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Notes |
| 0.1 | 01.12.2024 | Charles | First Draft |
| 0.2 | 20.12.2024 | Charles | * Changed the implementation of Milestone 5 from KW 1 – 2025 to KW 8 – 2025 due to upcoming exams * Changed the implementation of Milestone 6 from KW 2 – 2025 to KW 9 – 2025 due to upcoming exams |
| 0.3 | 28.02.2025 | Charles | Removes the documentation aspect in Milestone 6 |

# Introduction

This project plan outlines the tasks and milestones for coding the Expense Tracker Software. The tasks are divided into manageable milestones, ensuring a step-by-step implementation of the system's functionality.

## Glossary

The following abbreviations and domain-specific terms are used throughout the document.

|  |  |  |  |
| --- | --- | --- | --- |
| Abbreviation | Term | Description | Reference |
| KW | Calendar Week | Week count in a year | N/A |

# Milestones and Tasks

## Milestone 1: Core Functionality (Transaction Management) ✓

### Week 1 (KW 49 – 2024)

* Implement addTransaction(), editTransaction(), deleteTransaction(), getTotalBalance() and getRecentTransactions() methods in **TransactionManager**.
* Create basic transaction filtering logic using the Strategy Pattern (filter by date, amount, source and category.
* Unit test transaction-related methods to ensure correctness.

## Milestone 2: Authentication and User Management ✓

### Week 2 (KW 50 – 2024)

* Implement login(), logout(), and resetPassword() in **AuthManager**.
* Ensure secure password storage using hashing.
* Unit test all authentication processes.
* Integrate database functionality for storing, modifying and retrieving user account data through **UserAccountManager**.

## Milestone 3: Transaction Viewing and Reporting ✓

### Week 3 (KW 51 – 2024)

* Implement **TransactionHistory** to display all transactions and integrate Singleton and Observer patterns.
* Implement **ReportSummary** to generate reports and integrate Singleton, Observer and Strategy patterns.
* Implement **ExpenseSummary** to display categorized expenses and integrate Singleton and Observer patterns.
* Test various strategies for filtering and report generation.
* Test the Observer pattern.

## Milestone 4: User Interface Integration ✓

### Week 4 (KW 52 – 2024)

* Implement a console-based UI to test functionality.
* Add menu options for adding transactions, filtering, generating reports, and managing accounts.
* Display feedback for errors (e.g., failed login attempts, invalid input).
* Test UI interactions with the backend.

## Milestone 5: Settings and Data Persistence ✓

### Week 5 (KW 8 – 2025)

* Implement **Settings.**
* Implement saveData() and loadData() methods in **DataStorage**.
* Integrate the Singleton Database for managing transactions and user accounts.
* Test persistence functionality by saving and reloading transactions and user settings.

## Milestone 6: Final Testing and Optimization ✓

### Week 5 (KW 9 – 2025)

* Perform integration testing to verify that all components work together.
* Optimize database queries for efficiency.
* Identify and fix any bugs.
* Testing user experience and usability.