

Project Plan: Expense Tracker

1. Project Overview

Goal: Build an application that lets users input their Income(s) and Expense(s), Categorize them and return the Surplus/Deficit

2. Timeline

Estimated Project Duration: 8 Weeks

3. Milestones and Tasks

Milestone 1: Project Set Up and Initial Planning

Task 1.1: Choose project and determine requirements

Time: 1 Day

Deliverables: Project Plan, Requirements Document, Gantt Chart

Task 1.2: Choose programming language and create Repository

Time: 1 Day

Deliverables: Working GitHub Repository with README file

Functional IDE chosen and installed for Java coding

Task 1.3: Design the basic UI layout

Time: 2 Days

Deliverable: Description of what the application will look like

Milestone 2: Backend Development and Functionality

Task 2.1: Design an algorithm that receives user inputs

Time: 1 Week

Deliverable: Modular algorithm that stores user inputs locally

Task 2.2: Implement data categorization and error handling

Time: 1 Week

Deliverable: Assigns data (User Inputs) to a specific type such as Income or Expense and their category (E.g. Rent Expense)

Task 2.3: Calculate the net income and net expense (Surplus/Deficit)

Time: 1 Week

Deliverable: Algorithm that returns a Positive (Surplus) or Negative (Deficit) number

Milestone 3: Frontend Development

Task 3.1: Build basic User Interface

Time: 3 Days

Deliverable: UI that prompts user to input data (Income or Expense)

Task 3.2: Connect UI to API and Backend functions

Time: 1 Week

Deliverable: Functional connection to API

Result and Category displayed in UI

Task 3.3: Implement add/edit/delete functionality to UI

Time: 4 Days

Deliverables: User can input additional Incomes/Expenses

Result is modified and displayed again

Milestone 4: Testing and Bug Fixing

Task 4.1: Write unit tests for Backend functionality

Time: 2 Days

Deliverable: Backend unit tests E.g. Positives, Negatives, Decimals, Integers

Task 4.2: Write tests for Frontend UI (Different User Inputs)

Time: 2 Days

Deliverable: Comprehensive Frontend tests

Task 4.3: Fix identified bugs and improve error handling

Time: 5 Days

Deliverable: Bug-free system with error handling messages

Milestone 5: Deployment, Consultation, Final Review

Task 5.1: Upload final program to GitHub (With documentation)

Time: 1 Days

Deliverable: Final application/documentation able for review on GitHub

Task 5.2: Consultation and feedback from lecturer

Time: 1 Week

Deliverable: Documented feedback with possible improvements

Task 5.3: Final modification, review, and documentation

Time: 1 Week

Deliverable: Error-free/Modified application with finalized documents

4. Deliverables

1. Project Requirements Document:

Details of the application such as the features, design and possible challenges

2. Source Code:

Functional, Error-free code that performs intended tasks

3. UI Design:

Visual features of the application

4. Working application:

Overall working application that combines Frontend and Backend functionality

5. Unit Tests:

Test cases for the Frontend and Backend (Base cases and Extreme cases)

6. GitHub Uploads:

README, Project Plan, Gantt Chart, Features, Test Cases, Presentation

7. Project Report:

Summary of the project, challenges faced and resources utilized

5. Risks and Mitigation Plans

1. Lack of Experience:

Mitigation: Continuous consultation/feedback from lecturer

2. Time Management:

Mitigation: Maintain and follow a rigid schedule

3. Potential Errors and Bugs:

Mitigation: Continuous testing and error handling

Construct and implement secure test cases

6. Project Schedule Overview

1. MS1: Project Set Up and Planning

Duration: 1 Week

2. MS2: Backend Development

Duration: 2 Weeks

3. MS3: Frontend Development

Duration: 2 Weeks

4. MS4: Testing and Bug Fixes

Duration: 2 Weeks

5. MS5: Project Deployment

Duration: 1 Week