Version 1.0

# 1. Data Entry and Storage

## Task 1.1: Design Database Schema

- Define the database schema for storing financial data.

- Identify the necessary tables, fields, and relationships.

## Task 1.2: Set Up SQLite Database

- Create a new SQLite database based on the designed schema.

- Implement basic database operations (create, read, update, delete).

## Task 1.3: Develop Data Entry Interface

- Design a simple user interface for manual data entry.

- Implement forms or input fields for capturing account balances and transactions.

## Task 1.4: Implement Data Validation

- Add data validation logic to ensure data integrity and prevent incorrect inputs.

- Handle error scenarios gracefully and provide user feedback.

# 2. Data Processing and Calculations

## Task 2.1: Develop Basic Calculation Functions

- Implement functions to calculate total net worth, total debt, total stock investment, total bank balances, and total monthly bills.

- Test the functions with sample data to ensure accuracy.

## Task 2.2: Integrate Calculation Functions

- Integrate the calculation functions into the data entry process.

- Update the user interface to display the calculated metrics.

# 3. Graphical User Interface (GUI)

## Task 3.1: Set Up BeeWare Development Environment

- Install and configure the BeeWare framework for cross-platform GUI development.

- Ensure the development environment is properly set up.

## Task 3.2: Design Basic GUI Layout

- Create a basic GUI layout using BeeWare’s tools and libraries.

- Define the main screens, navigation flow, and component placement.

## Task 3.3: Implement GUI Functionality

- Connect the GUI components to the backend functionality.

- Implement event handlers and callbacks for user interactions.

# 4. Data Security and Privacy

## Task 4.1: Implement Basic Encryption

- Research and select a suitable encryption library for Python.

- Implement basic encryption for sensitive data stored in the database.

## Task 4.2: Secure Local Data Storage

- Ensure that the SQLite database file is stored securely on the user’s device.

- Implement file-level encryption or access controls if necessary.

# 5. Testing and Quality Assurance

## Task 5.1: Develop Test Cases

- Create a set of test cases to cover the core functionalities of the software.

- Include test cases for data entry, calculations, and GUI interactions.

## Task 5.2: Perform Manual Testing

- Execute the test cases manually and record the results.

- Identify and fix any bugs or issues discovered during testing.

## Task 5.3: Conduct Cross-Platform Testing

- Test the software on different platforms (Windows, Mac, iOS, Android) to ensure compatibility and consistency.

- Address any platform-specific issues or quirks.

# 6. Documentation

## Task 6.1: Create User Manual

- Write a simple user manual that guides users through the basic functionalities of the software.

- Include instructions for installing, launching, and using the application.

## Task 6.2: Prepare README File

- Create a README file that provides an overview of the project and its features.

- Include instructions for setting up the development environment and running the application.