



## **Echo: Personal Finance Tracker**

December, 2023

Echo Software Requirements and Design Specification Document

Ali Elbahaie 900213083

Amir Jemal 900233442

Hala Eissa 9001104347

Hana Eissa 900201222

Mazen Zaki 900211593

Omar Youssuf 900212166

CSCE 3701: Software Engineering

Dr. Sally ElGhamrawy

## Table of Contents

<b>Table of Contents.....</b>	<b>2</b>
<b>1. Introduction.....</b>	<b>5</b>
1.1 Purpose.....	5
1.2 Scope.....	5
1.3 Definitions, Acronyms, and Abbreviations.....	6
1.4 Overview.....	6
<b>2. Overall Description.....</b>	<b>7</b>
2.1 Product Perspective.....	7
2.2 User Classes and Characteristics.....	7
2.3 Motivation Behind The Software.....	8
2.4 Operating Environment.....	8
2.5 Design and Implementation Constraints.....	9
<b>3. System Features.....</b>	<b>9</b>
3.1 User Account Management.....	9
3.2 Income Tracking.....	10
3.3 Expense Tracking.....	10
3.4 Budgeting.....	10
3.5 Savings Goals.....	11
3.6 Investment Tracking.....	11
3.7 Reports and Analytics.....	11
3.8 Notifications.....	12
<b>4. Features to be Implemented.....</b>	<b>12</b>
Sprint 1.....	12
4.1 User Account Management.....	12
4.2. Income Tracking.....	12
4.3. Expense Tracking.....	12
Sprint 2.....	13
4.4. Budgeting.....	13
Sprint 3.....	13
4.5. Savings Goals.....	13
4.6. Reports and Analytics.....	13
<b>5. Diagrams.....</b>	<b>14</b>
5.1 New User Use-Case Diagram.....	14
5.2 New User Use-Case Diagram Description.....	15
5.3 Existing User Use-Case Diagram.....	17
5.4 Existing User Use-Case Diagram Description.....	18
5.5 Class Diagram.....	19
5.6 Sequence Diagram.....	20
5.6.1 Log In.....	20

5.6.2 Sign Up.....	20
5.6.3 Add Income or Expense.....	21
5.6.4 Add a Budget Or Saving Goal.....	22
5.6.5 Reports and Analytics.....	22
5.7 Activity Diagram.....	23
5.7.1 Existing User.....	23
5.7.2 New User Login.....	24
5.7.3 Add Income.....	24
5.7.4 User change income.....	25
5.7.5 Add expense.....	26
5.7.6 Change Expense Activity.....	27
<b>6. System Architecture and Database Design.....</b>	<b>29</b>
6.1 Monolithic Layered Architecture.....	29
6.2 Database Design.....	31
<b>7. Non-functional Requirements.....</b>	<b>31</b>
7.1 Performance Requirements.....	31
7.2 Security Requirements.....	31
7.3 Usability.....	32
7.4 Scalability and Reliability.....	32
<b>8. Potential Risks.....</b>	<b>33</b>
8.1 List of Potential Risks.....	33
8.2 Impact of each Risk.....	33
8.3 Mitigation Strategies.....	34
<b>9. Changes Since Milestone 1.....</b>	<b>35</b>
9.1 Technical Implementation Changes.....	35
9.2 Solution Design Changes.....	35
<b>10. Sprint 1 Report.....</b>	<b>36</b>
10.1 Sprint Logistics.....	36
10.1.1 Sprint Timeline.....	36
10.1.2 Team Composition.....	36
10.2 Sprint Goals.....	37
10.2.1 Planned versus Achieved.....	37
10.2.2 Story Completion.....	38
10.3 Scope and Deliverables.....	39
10.3.1 Scope Overview.....	39
10.3.2 Deliverables.....	39
10.4 Status Review.....	40
10.4.1 Completed Items.....	40
10.4.2 In-Progress Items.....	40
10.4.3 Delayed Items.....	40
10.5 Quality and Testing.....	41

10.5.1 Testing Overview.....	41
10.5.2 Defects and Issues.....	41
10.5.3 Code Quality Metrics.....	41
10.6 Sprint Retrospective.....	42
10.6.1 What Went Well.....	42
10.6.2 Areas for Improvement.....	42
10.6.3 Lessons Learned.....	42
10.6.4 Plans for Improvement.....	43
10.7 Next Sprint Planning.....	43
10.7.1 Objectives for Next Sprint.....	43
10.7.2 Backlog Prioritization.....	44
10.7.3 Resource Allocation.....	44
<b>11. Sprint 2 Report.....</b>	<b>45</b>
11.1 Sprint Logistics.....	45
11.2 Sprint Goals.....	45
11.3 Scope and Deliverables.....	45
11.4 Status Review.....	46
11.5 Quality and Testing.....	46
11.6 Sprint Retrospective.....	46
11.7 Next Sprint Planning.....	48
<b>12. Sprint 3 Report (Final Sprint).....</b>	<b>49</b>
12.1 Sprint Logistics.....	49
12.1.1 Sprint Timeline.....	49
12.1.2 Team Composition.....	49
12.2 Sprint Goals.....	50
12.3 Scope and Deliverables.....	51
12.3.1 Scope Overview.....	51
12.3.2 Deliverables.....	51
12.4 Status Review.....	52
12.4.1 Completed Items.....	52
12.4.2 In-Progress and Future Items.....	52
12.5 Quality and Testing.....	53
12.5.1 Testing Overview.....	53
12.5.2 Defects and Issues.....	54
12.6 Sprint Retrospective.....	55
12.6.1 What Went Well.....	55
12.6.2 Challenges Faced.....	56
10.6.3 Lessons Learned.....	56
10.6.4 Plans for Improvement.....	57
<b>13. Conclusion.....</b>	<b>58</b>
13.1 Reflective Summary.....	58

13.2 Achievements and Outcomes.....	58
13.3 Future Directions.....	58
13.4 Closing Note.....	58

# 1. Introduction

## 1.1 Purpose

This document is designed to serve as the cornerstone for the development of the Personal Finance Tracker software. It aims to detail the functional and design specifications, outline the system architecture, and establish the roadmap for future enhancements. Our goal is to provide a clear understanding of the product's intent, its capabilities, and the value it offers to users seeking to improve their financial management practices.

## 1.2 Scope

The scope of this document covers a comprehensive overview of the Personal Finance Tracker software, from its foundational user account management to the intricate details of financial tracking, reporting, and analysis. It encompasses the system features, design considerations, and the implementation plan across successive development sprints.

## 1.3 Definitions, Acronyms, and Abbreviations

- **Income Tracking:** The process of recording and monitoring all sources of income.
- **Expense Tracking:** The categorization and recording of all personal expenditures.
- **Budgeting:** The allocation of personal finances to various categories and monitoring of spending against these allocations.
- **Savings Goals:** Financial targets set by users, with tools to track progress towards achieving them.
- **Investment Tracking:** Monitoring of financial investments, including stocks, bonds, and other securities.

- **GUI:** Graphical user interface
- **CSV:** Comma-Separated Values

## **1.4 Overview**

The overview presents a snapshot of the Personal Finance Tracker software, highlighting its cross-platform utility, user-friendly interface, and the strategic approach to its development. The document further outlines the user classes and characteristics, the motivation behind the software's creation, and the environment it operates within.

## **2. Overall Description**

### **2.1 Product Perspective**

The Personal Finance Tracker software was initially designed as a standalone product, allowing users to independently manage their financials without relying on external integrations. While the current version operates autonomously, there is potential for future integration capabilities, such as linking to bank accounts and online investing platforms. This would further enhance the software's utility by providing real-time financial data and insights.

### **2.2 User Classes and Characteristics**

The primary target audience for the Personal Finance Tracker software encompasses individuals seeking clarity on their financial habits. These users span various age groups and professional backgrounds. Their primary objective when using the software is to gain a clear understanding of their spending patterns, ensuring they know the destination of every penny. By providing a comprehensive view of their finances, the software empowers users to make informed decisions and set realistic budgets.

### **2.3 Motivation Behind The Software**

In an increasingly complex financial landscape, individuals often find themselves juggling multiple income sources, expenses, investments, and financial goals. The motivation behind the Personal Finance Tracker software is to simplify this complexity. By consolidating all financial data into one intuitive platform, users can gain a holistic view of their financial health, make informed decisions, and set achievable financial goals. The software aims to promote financial literacy and empower users to take control of their financial future.



## **2.4 Operating Environment**

The Personal Finance Tracker software is engineered as a cross-platform desktop application utilizing QT Creator, a comprehensive development environment that enables seamless functionality across various operating systems. This design choice focuses on delivering a robust, responsive, and user-friendly interface tailored for desktop use. It is compatible with the most commonly used operating systems, including Windows, macOS, and Linux, ensuring users can manage their finances effectively regardless of their preferred platform.

By leveraging the power of QT Creator, the application offers a rich set of features that are optimized for performance and accessibility on desktop devices. Users can expect a consistent and intuitive experience with the added benefit of offline access to their financial data. Since the application does not rely on web connectivity, it provides enhanced security and reliability, allowing users to confidently track their income, expenses, and perform other financial management tasks with the assurance that their sensitive information remains secure on their personal computers.

## **2.5 Design and Implementation Constraints**

Certain constraints bind the development of the Personal Finance Tracker software. Time is a primary limitation, necessitating efficient development practices to ensure timely delivery of the product. Additionally, while the software aims to offer a comprehensive suite of features, certain advanced integrations, such as real-time bank data syncing, might be reserved for future versions due to current constraints.

### 3. System Features

#### 3.1 User Account Management

The User Account Management feature provides the foundational access controls for the Personal Finance Tracker software.

- **Registration:** New users can create an account by providing essential details such as name, email, and password.
- **Login:** Existing users can securely access their accounts using their registered email and password.
- **Password Recovery:** In case users forget their password, they can initiate a password recovery process, typically involving email verification.

#### 3.2 Income Tracking

Income Tracking allows users to have a clear picture of all their incoming finances.

- **Add Sources of Income:** Users can specify various sources of income, such as salary, freelance work, or other revenue streams.
- **Recurring Income:** Regular incomes, like monthly salaries, can be set as recurring, so they're automatically added.
- **One-time Bonuses:** Users can also add one-time income sources, like annual bonuses or gifts.

#### 3.3 Expense Tracking

This feature ensures users are aware of where their money is being spent.

- **Categorize Expenses:** Users can categorize their expenses into predefined categories like groceries, rent, and utilities, or create custom categories.

- **Recurring Expenses:** Regular expenses, such as monthly rent or subscription services, can be set as recurring.
- **One-time Purchases:** Sporadic expenses, like buying a gadget or furniture, can be added as one-time purchases.

### 3.4 Budgeting

Budgeting tools empower users to allocate their finances effectively.

- **Set Monthly Budgets:** Users can set monthly spending limits for different categories to ensure they don't overspend.
- **Visualize Overspending:** The software provides visual cues, like graphs or color indicators, to show when users are nearing or have exceeded their budget.

### 3.5 Savings Goals

This feature motivates users to save money for future needs or desires.

- **Set Savings Targets:** Users can specify financial goals, such as saving for a vacation or buying a car.
- **Track Progress:** The software visually displays the progress towards each savings goal, encouraging users to stay on track.

### 3.6 Investment Tracking

For users involved in investments, this feature provides essential insights.

- **Monitor Investment Portfolios:** Users can add details of their investments, such as stocks, bonds, or mutual funds, and track their performance.
- **Dividends:** Any dividends received from investments can be recorded.

- **Capital Gains/Losses:** The software calculates and displays any capital gains or losses from the sale of investments.

### 3.7 Reports and Analytics

Data-driven insights help users understand their financial habits.

- **Monthly Expense Breakdown:** Users can view a detailed breakdown of their monthly expenses by category.
- **Income vs. Expense Chart:** A comparative chart shows users their total income against their total expenses for a specified period.
- **Savings Progress:** Users can view reports on how well they're progressing toward their savings goals.

### 3.8 Notifications

Timely alerts ensure users are always informed.

- **Alerts for Overspending:** If users are nearing or have exceeded their budget for a category, they receive an alert.
- **Reminders for Bill Payments:** Users can set reminders for upcoming bill payments to ensure they never miss a due date.

## **4. Features to be Implemented**

### **Sprint 1**

#### **4.1 User Account Management**

User Account Management is fundamental as it enables users to create accounts, log in securely, and recover passwords if forgotten. Without a robust account management system, users cannot access the app and its features.

#### **4.2. Income Tracking**

Income tracking is crucial as it allows users to understand their financial inflow. It helps users plan their budget, savings, and investments effectively. Managing different sources of income and tracking them over time provides users with a clear financial overview.

#### **4.3. Expense Tracking**

Expense tracking is essential for financial management. Categorizing expenses, differentiating between recurring and one-time expenses, and monitoring spending habits are key aspects. Users need to know where their money is going to make informed decisions about budgeting and saving.

### **Sprint 2**

#### **4.4. Budgeting**

Budgeting tools empower users to control their finances. Setting spending limits and visualizing overspending provide immediate feedback, encouraging responsible financial behavior. Effective budgeting ensures that users live within their means and avoid unnecessary debts.

## **Sprint 3**

### **4.5. Savings Goals (Cancelled)**

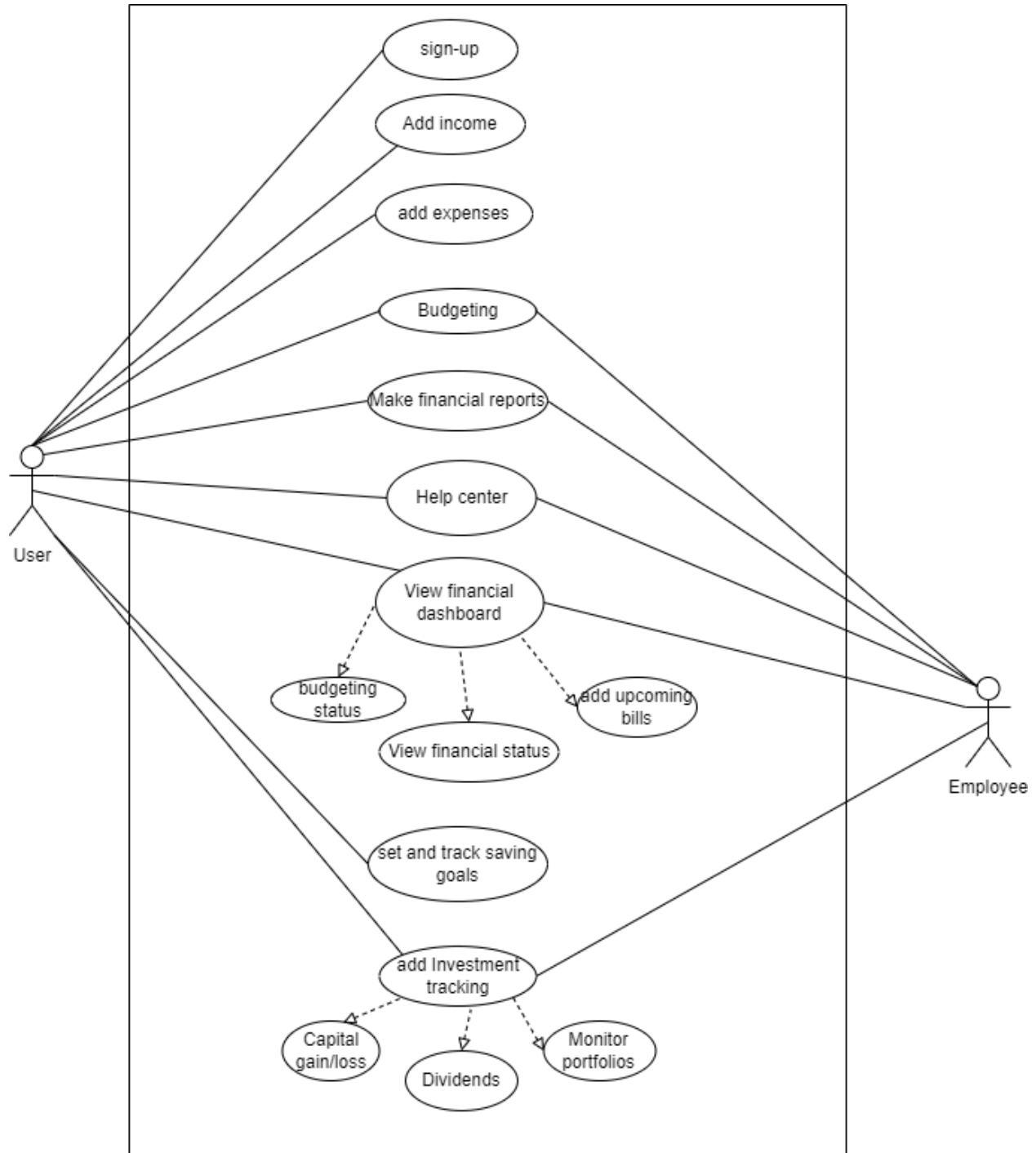
Savings goals motivate users to save for specific purposes. Setting targets and tracking progress visually encourages disciplined saving habits. Achieving savings goals gives users a sense of accomplishment and financial security for future needs or desires.

### **4.6. Reports and Analytics**

The implementation of a reports and analytics feature in our personal finance software during the upcoming sprint directly addresses user feedback for enhanced insight into their financial data. By providing users with visual reports and trend analysis, we aim to facilitate informed financial decision-making, improve user engagement, and maintain a competitive edge. This feature capitalizes on the inherent value of user data, transforming it into actionable intelligence that empowers our users to optimize their financial health and planning.

## 5. Diagrams

### 5.1 New User Use-Case Diagram



## 5.2 New User Use-Case Diagram Description

In this use case diagram, we have two actors which are the user who will use the system and the employee. The goal of this is that the user can add their income and check their financial dashboard, set saving goals, or make financial reports. In order for the user to accomplish these goals they would have to go through the following steps:

- The user access the website through their browser
- The user would sign up to the service in order for the system to be able to save all their information for later usage.
- The user would then be told to enter their income and their expenses.
- Then the user would be able to view their financial dashboard.
- From the financial dashboard the user can either choose to:
  - Add any upcoming bills to the system
  - View financial status
  - Edit budgeting status
- The user can also choose to set or track their saving goals
- OR the user can ask the system to make financial reports
- If the user faces any difficulties the user can interact with the help option where an employee of echo will be able to help with and guide them through the process.



Interaction:

- The user interacts with the website's interface in order to sign-up, add income and expenses, view their financial dashboard, or set saving goals, or make financial reports, and can ask for help from the system.
- The system interacts with the user by saving the information that they input, providing them with their financial dashboard when prompted to do so, saving their saving goals, or displaying financial reports when asked to. Moreover, when the user asks for help the system connects the user with an employee to help the user.

### 5.3 Existing User Use-Case Diagram



## 5.4 Existing User Use-Case Diagram Description

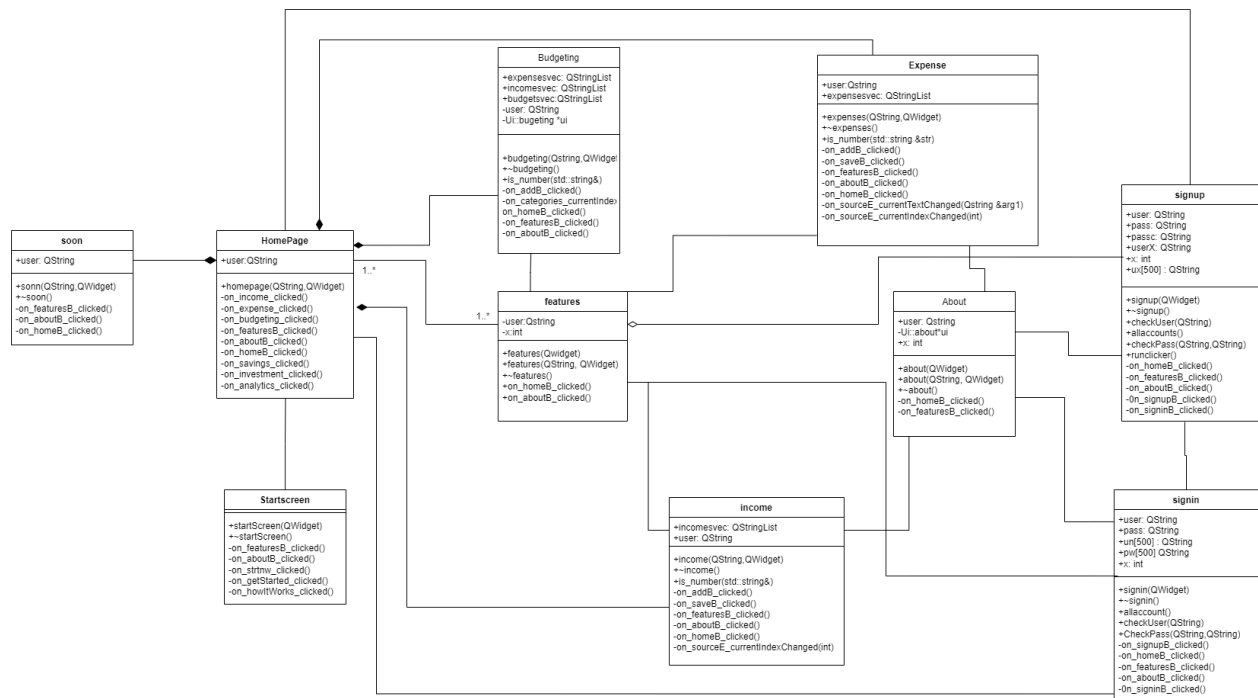
In this use-case diagram, we have three actors, which are the user who will use the system, the system itself, and the employee. The goal of this is that the user can add or adjust their income and check their financial dashboard, adjust their saving goals, or make financial reports. In order for the user to accomplish these goals, they would have to go through the following steps:

- The user access the app through his phone
- The user would log in to the service in order for the system to be able to view their previously saved information. .
- Then the user would be able to view their financial dashboard.
- The user would be able to set what their monthly budget would be.
- From the financial dashboard the user can either choose to:
  - Add any upcoming bills to the system
  - View financial status
  - Edit budgeting status
- The user can also choose to adjust or track their saving goals
- OR the user can ask the system to make financial reports.
- The user can add his investment portfolio and the system would help him monitor his investments, and show him the total amount of dividends and the capital loss or gain on this investment.
- If the user faces any difficulties the user can interact with the help option where an employee of Echo will be able to help with and guide them through the process.

Interaction:

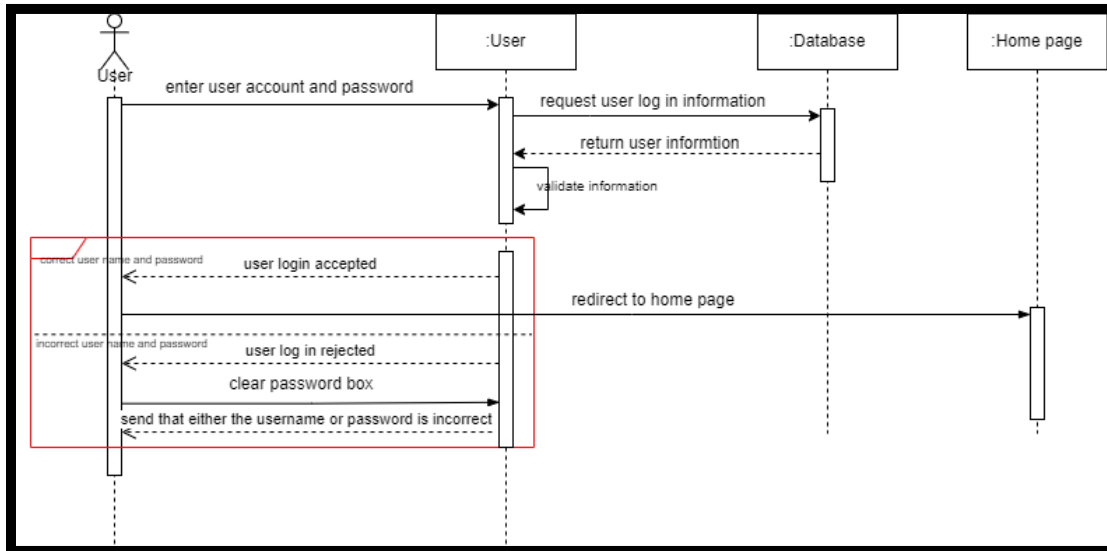
- The user interacts with the website's interface in order to log in, add or adjust income and expenses, view their financial dashboard, or change saving goals, or make financial reports, track their investments and how well it is performing and can ask for help from the system.
- The system interacts with the user by saving the information that they input, providing them with their financial dashboard when prompted to do so, saving their saving goals, or displaying financial reports when asked to. Moreover, when the user asks for help, the system connects the user with an employee to help the user.

## 5.5 Class Diagram

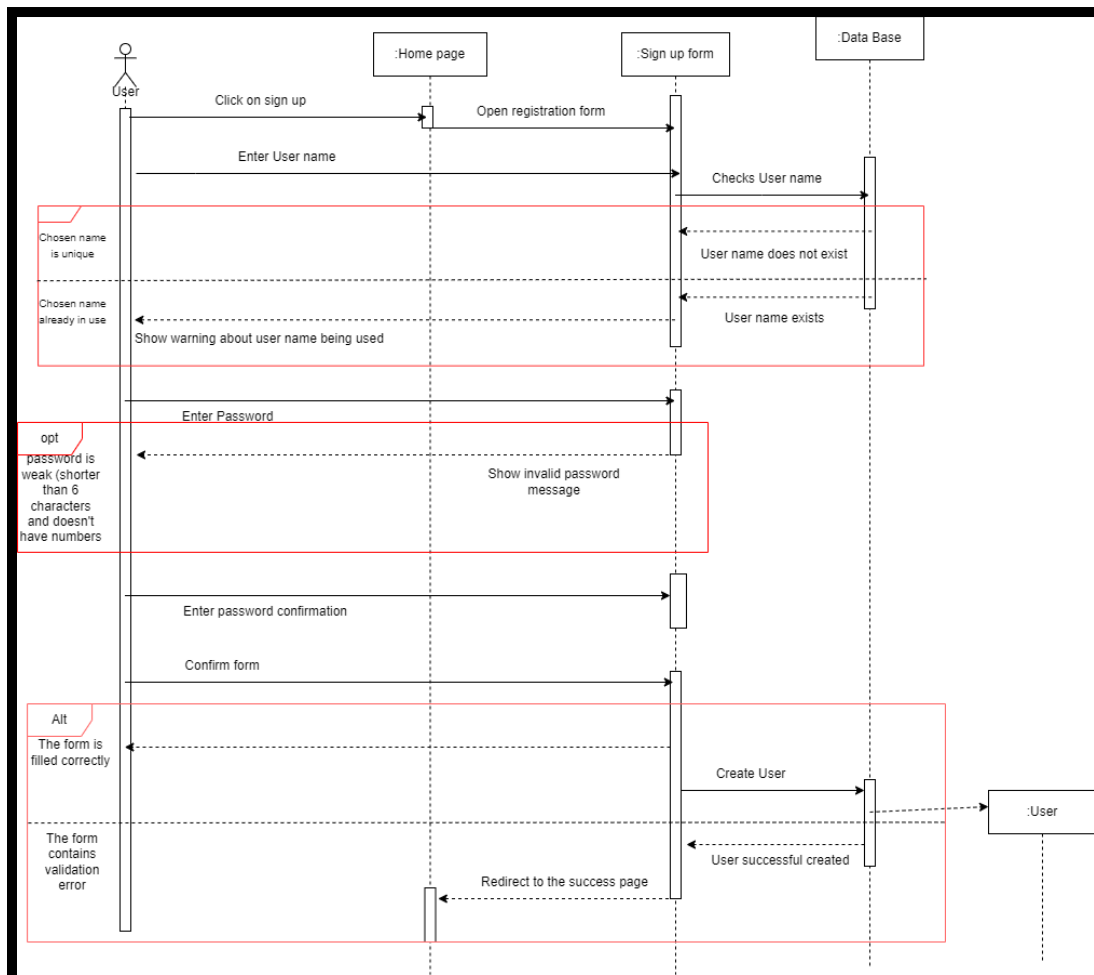


## 5.6 Sequence Diagram

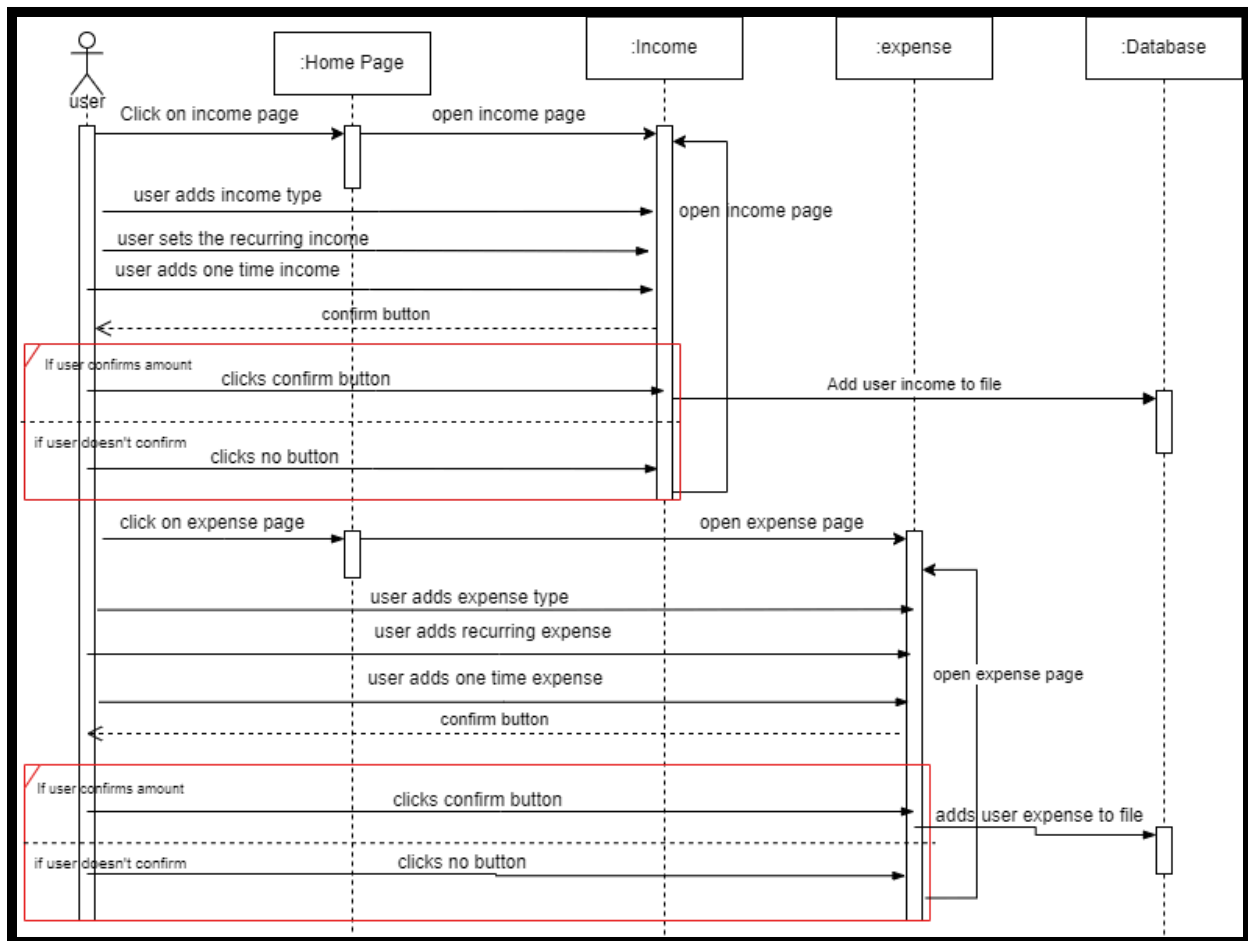
### 5.6.1 Log In



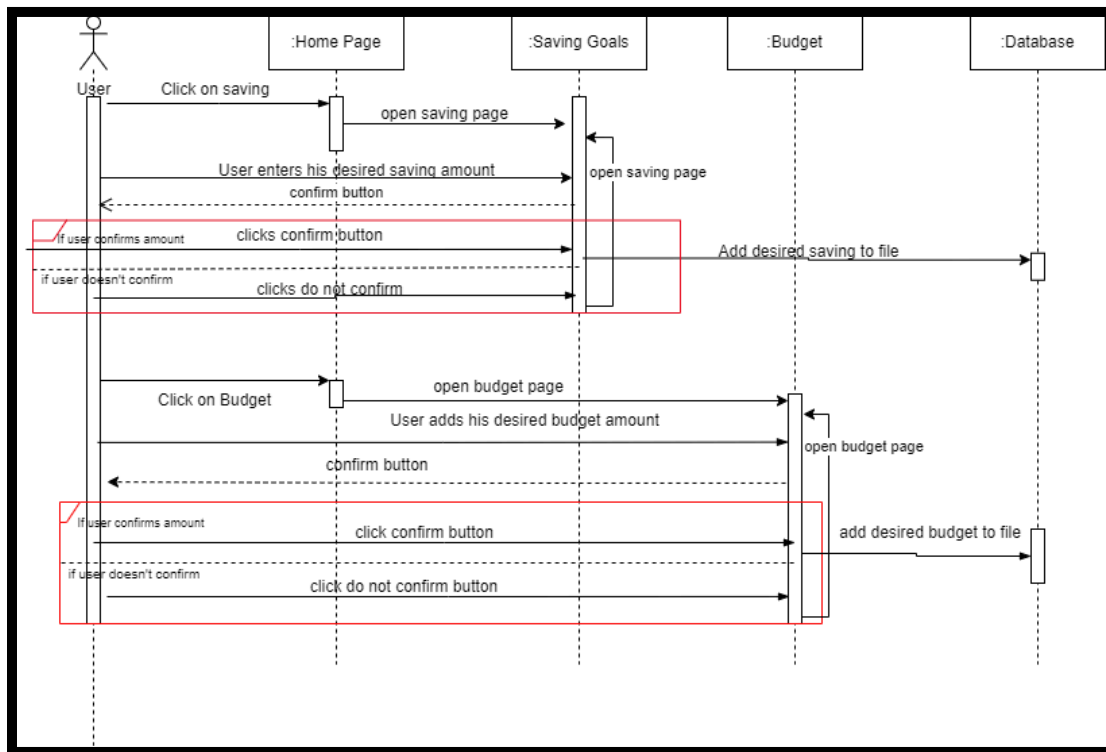
### 5.6.2 Sign Up



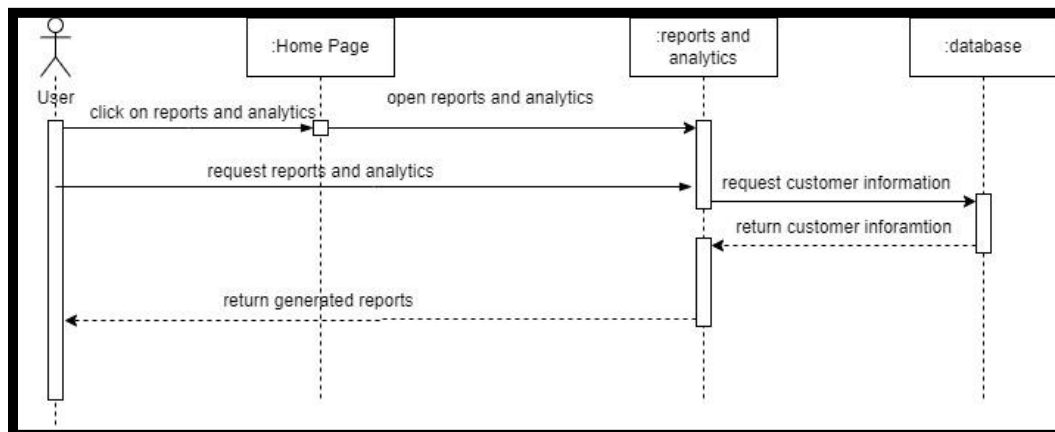
### 5.6.3 Add Income or Expense



### 5.6.4 Add a Budget Or Saving Goal

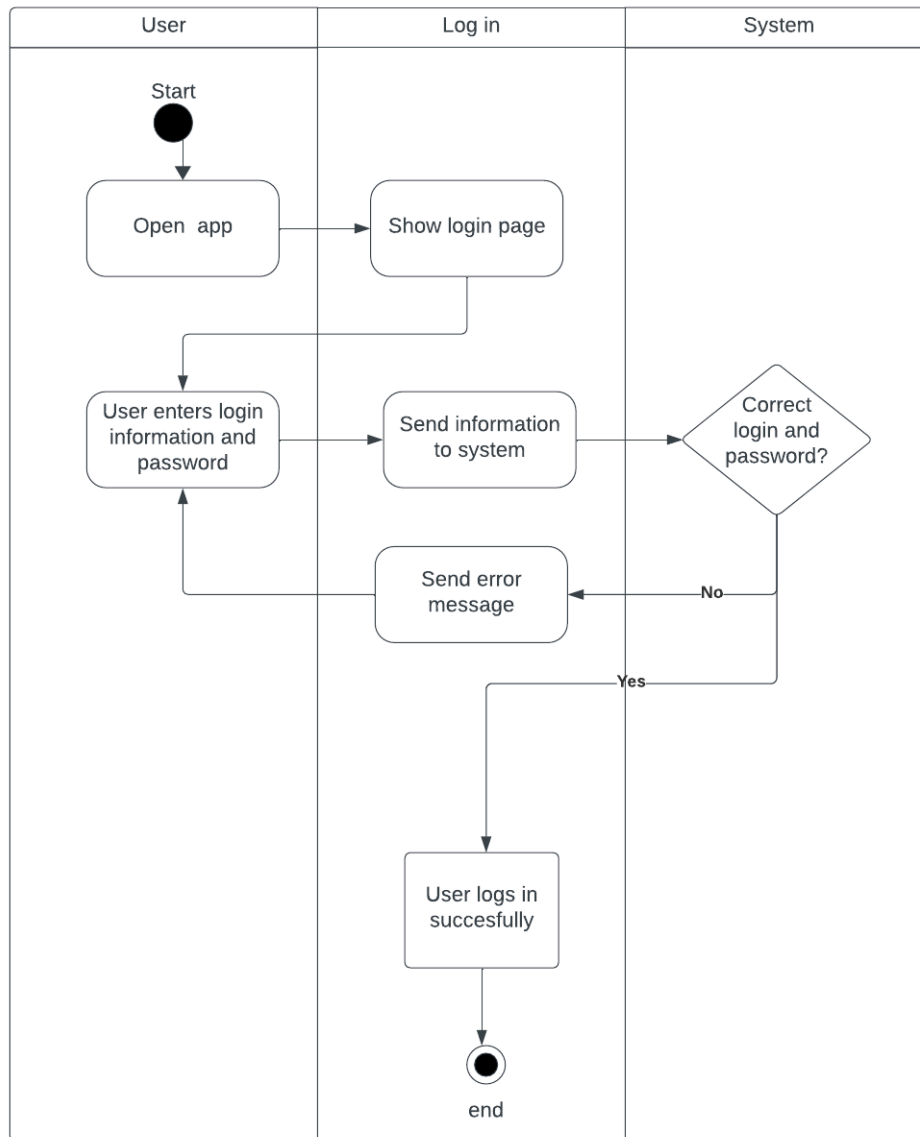


### 5.6.5 Reports and Analytics



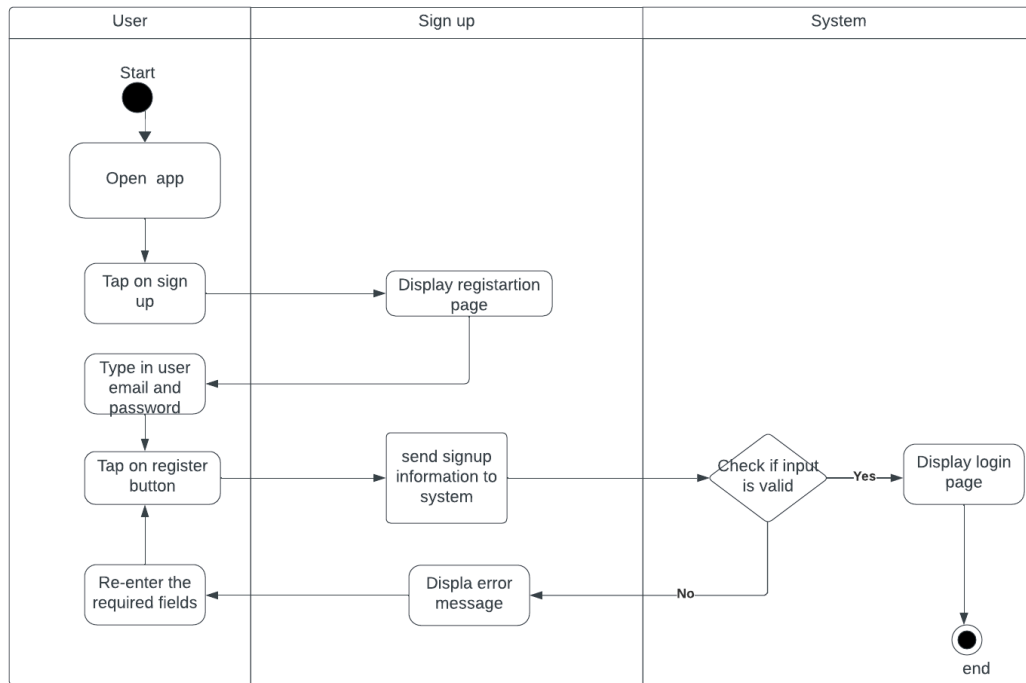
## 5.7 Activity Diagram

### 5.7.1 Existing User

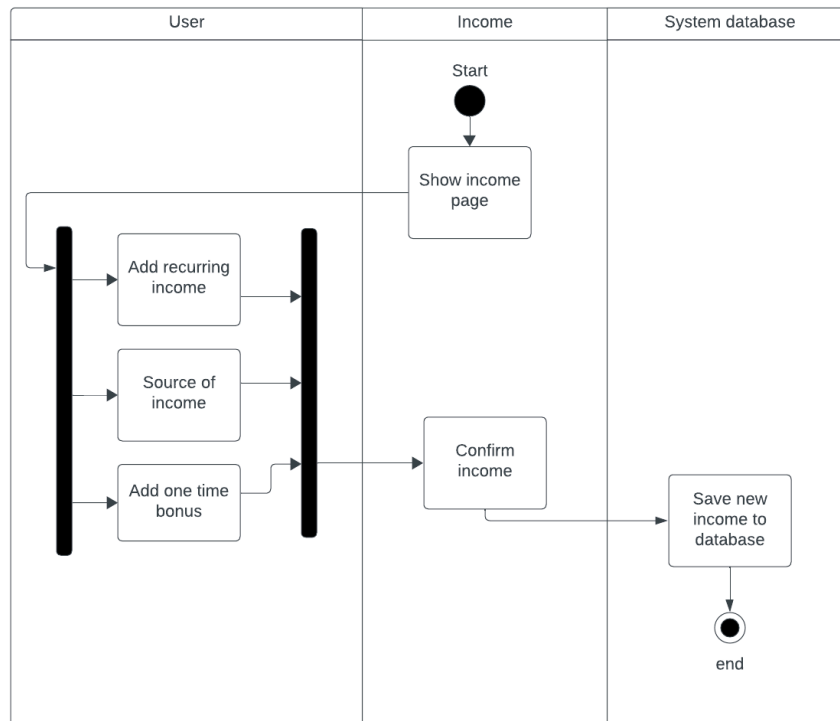




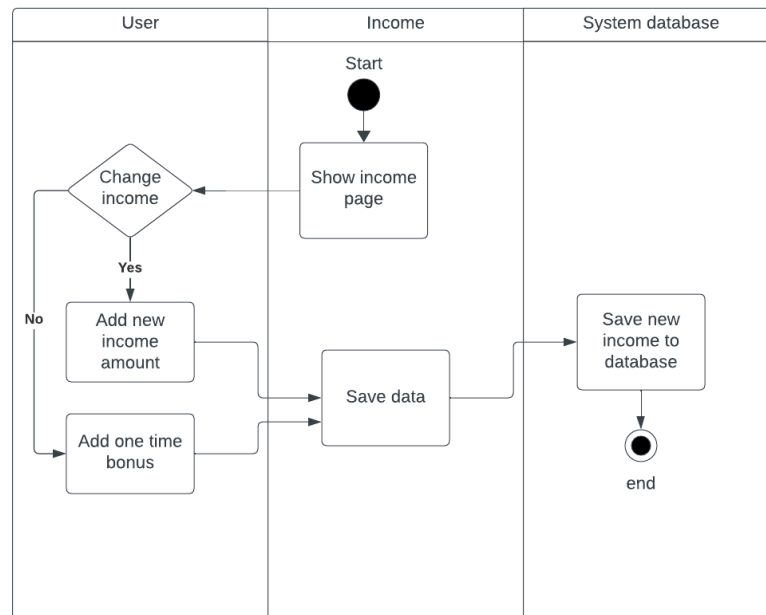
### 5.7.2 New User Login



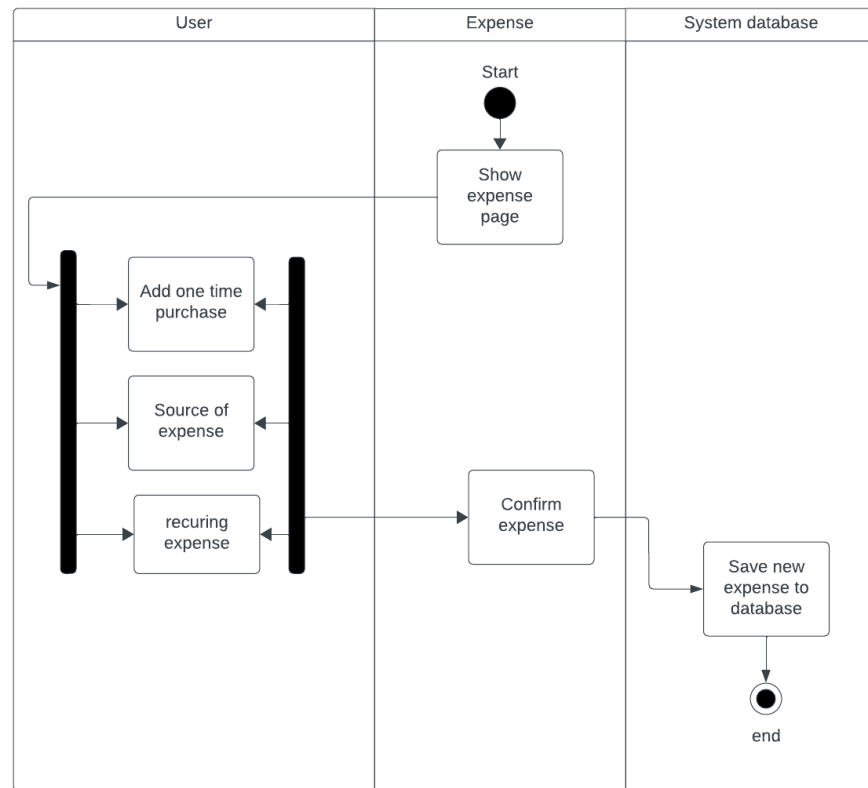
### 5.7.3 Add Income



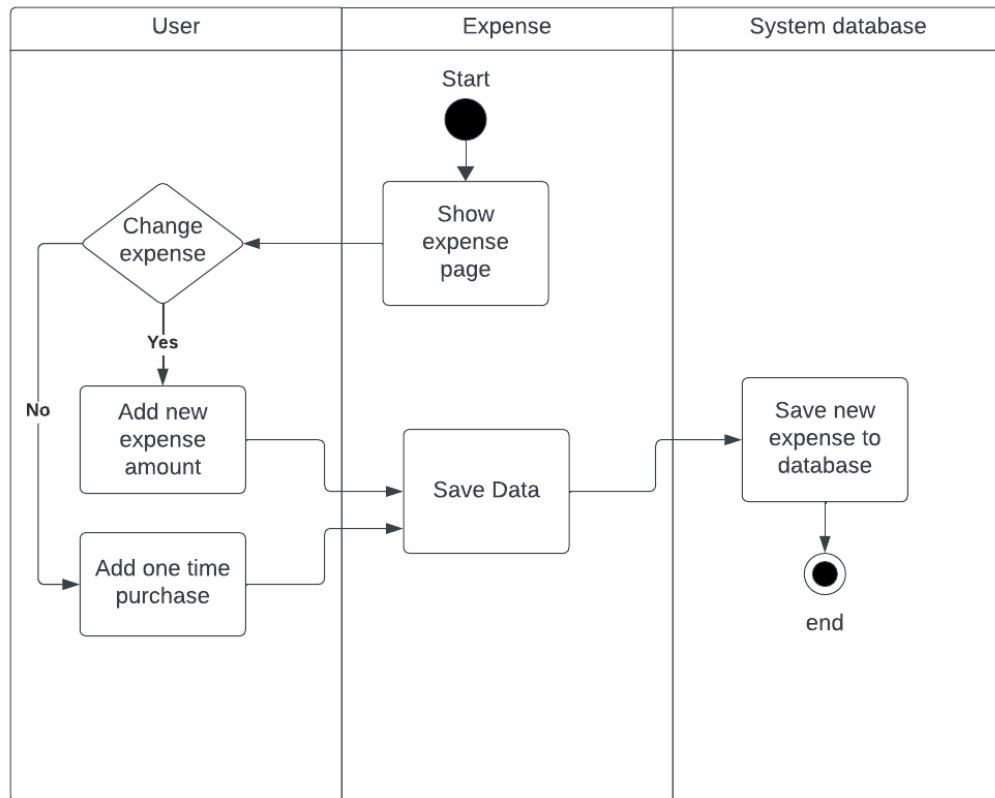
### 5.7.4 User change income



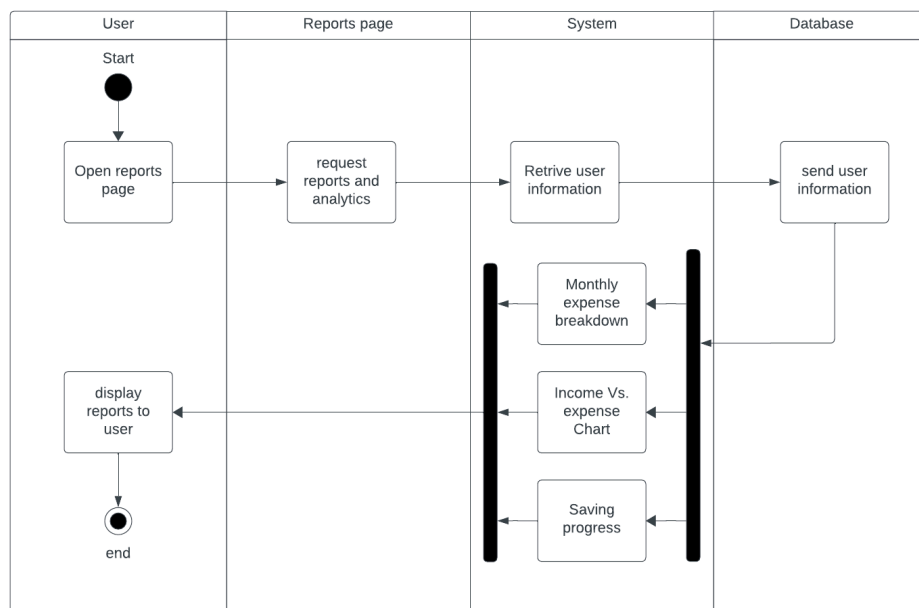
### 5.7.5 Add expense



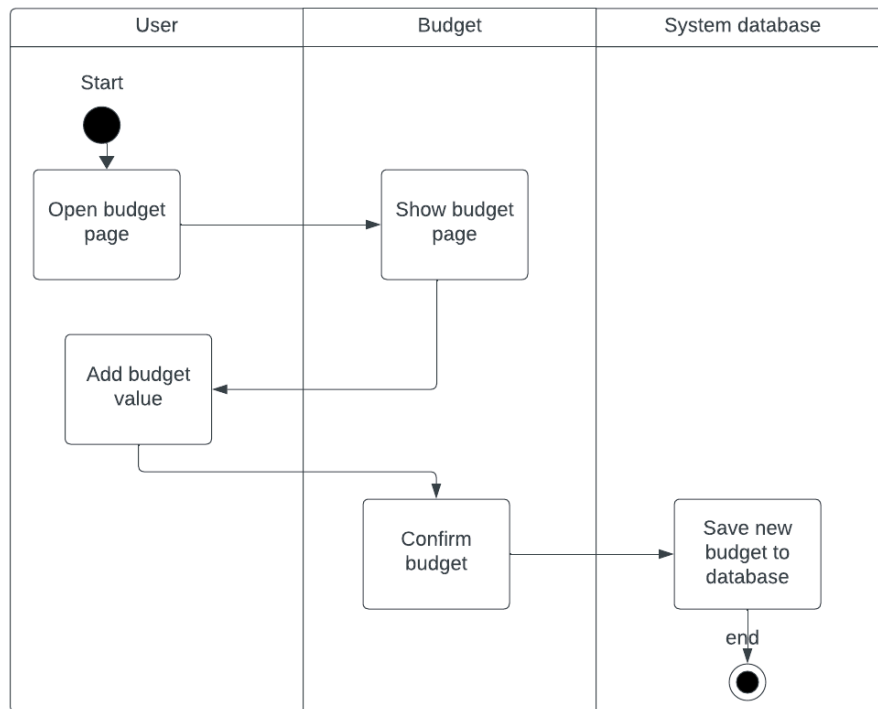
### 5.7.6 Change Expense Activity



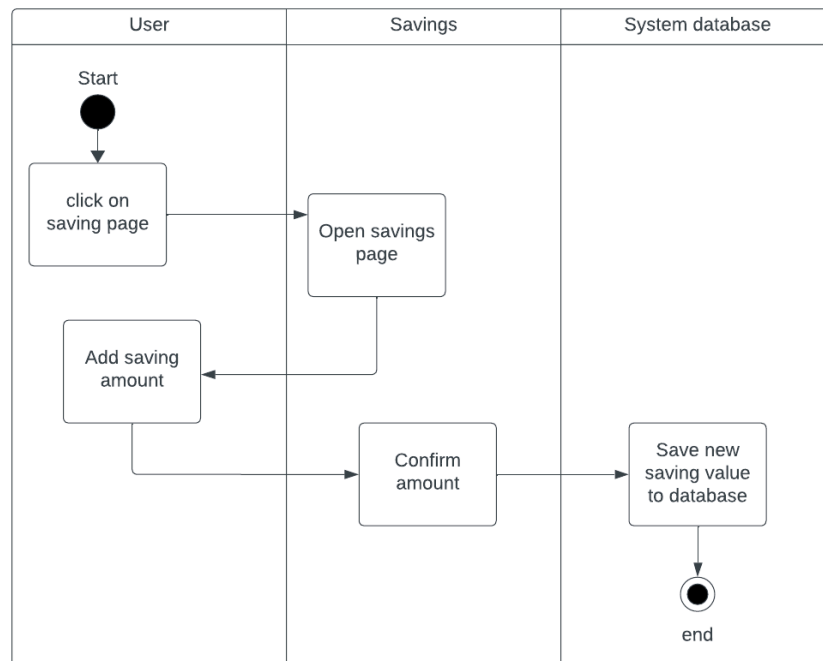
### 5.7.7 Reports and Analytics



### 5.7.8 Budgeting

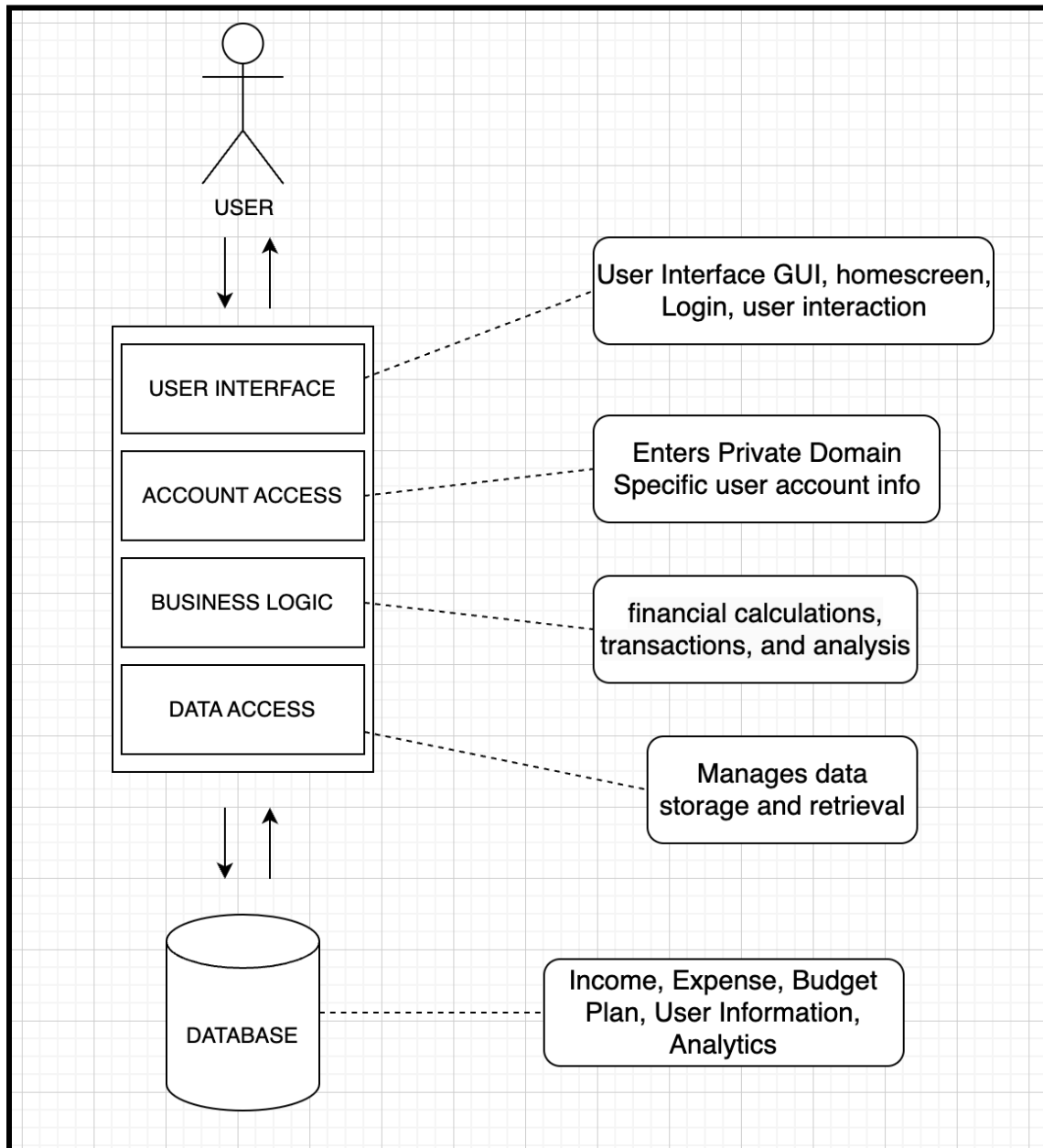


### 5.7.9 Saving goals



## 6. System Architecture and Database Design

### 6.1 Monolithic Layered Architecture



**Pros:** The architecture is straightforward and easier to understand since it follows a layered structure. Development teams can quickly grasp the overall structure of the application. The entire application is typically contained within a single codebase, making it easier to manage and deploy.

**Cons:** A failure in one part of the application can potentially affect the entire system. Increased development time to maintain and modify as business grows

## 7. Non-functional Requirements

### 7.1 Performance Requirements

The Personal Finance Tracker should be fast and responsive.

- **Quick Actions:** When you add expenses or check your reports, it should happen quickly.
- **Fast Start:** When you open the software, it should load up without making you wait too long.
- **Smooth Reports:** Even if you ask for a detailed report, it should be ready without any long waits.

### 7.2 Security Requirements

Your data's safety is our top priority.

- **Safe Data:** Your financial details will be kept private and safe.
- **Strong Passwords:** You'll need a good password to log in, and we might add extra safety steps in the future.
- **Regular Checks:** We'll often check to make sure everything is secure.

### 7.3 Usability

Using the software should be easy and clear for everyone.

- **Simple Design:** Everything should be easy to find and use.
- **Helpful Tips:** If you're not sure about something, there will be guides and tips to help you out.
- **Your Feedback Matters:** If you have ideas or face any issues, you can tell us, and we'll work on it.

## 7.4 Scalability and Reliability

The software is built to serve more people as it grows in popularity.

- **Ready for More Users:** Even if lots of people start using it, the software will work just as well.
- **Room for Your Data:** As you use it more and more, we've made sure there's space for all your financial details.
- **Always Available:** We aim to make sure you can use the software anytime you want.

And we have plans in place to keep your data safe and sound.



## 8. Potential Risks

### 8.1 List of Potential Risks

When working with very sensitive data, risks of all types especially user's data are something that are of importance in the development of the Personal Finance Tracker which is what we carefully considered throughout our thought process of each part of the software.

- **Data Breach:** Data breaches or unauthorized access to data could compromise the sensitive data our users entrust to us.
- **Technical Challenges:** Issues faced by users when using our software which are bugs or compatibility issues.
- **Performance:** As more users start using our software, performance could bottleneck resulting slow performance causing users to stop using our application.
- **Acceptance and Usage:** Users could find our application difficult to use leading to low user engagement.

### 8.2 Impact of each Risk

- **Data Breach:** Huge impact on our reputation and user trust. In addition, users could be using the same login credentials on our website as they do on others which could cause malicious actors attempting to exploit these credentials on other websites to gain access to their accounts on other platforms.
- **Technical Challenges:** It may cause a delay in the deployment of the project and impact the users negatively when using our application.

- **Performance:** Poor user experience due to slow performance which could lead to the loss of potential users.
- **Acceptance and Usage:** This could hinder the software's capabilities and features leading to loss of users.

### 8.3 Mitigation Strategies

- **Data Breach:** Updating our security measures regularly and always being on the lookout for new vulnerabilities and strategies made by black hats compromising other websites' data.
- **Technical Challenges:** Maintaining a clear and understandable documentation in addition to making a Bug Bounty system to solve them as fast as we can before other users are affected by these bugs.
- **Performance:** Making a feedback form for users to report any inconsistencies as well as always monitoring performance using graphs to track our crash rate before too many requests fail to complete.
- **Acceptance and Usage:** Focus on user experience design to allow our users a simple but interesting experience of tracking their finance and actively seeking user feedback on UI improvements.

## **9. Changes Since Milestone 1**

Since Milestone 1, there have been no shifts in the project goals, target users, or market strategy. The core objectives remain intact, with a steadfast focus on delivering a comprehensive personal finance tracker to our intended user base. While the overarching vision has stayed the same, there have been refinements in the project's use cases to better align with the user needs and technical capabilities.

### **9.1 Technical Implementation Changes**

The initial plan to develop the application as a web-based platform has been revised. The team explored the possibility of using React.js for this purpose; however, it was determined that the learning curve and time required to implement the software in React.js would not fit within our project timeline. Consequently, the development strategy shifted to creating a GUI application using Python. This approach was also later adapted; after further evaluation, the team found that C++ with QT Creator provided a more efficient path for implementation. This decision was made to leverage C++'s performance benefits and QT Creator's rich set of development tools and features for building cross-platform applications.

### **9.2 Solution Design Changes**

There has been a significant modification in the system's architecture since Milestone 1. After assessing various architectural options, a layered architecture was chosen for its modularity and ease of maintenance. This structure will allow for a clear separation of concerns, making the system more manageable and scalable. Furthermore, in regards to the database design, instead of relying on complex database systems, the team has decided to use CSV files for local data storage.

## **10. Sprint 1 Report**

### **10.1 Sprint Logistics**

#### **10.1.1 Sprint Timeline**

Sprint 1 commenced on November 3, 2023, and concluded on November 11, 2023. This sprint spanned over nine days, during which the team focused on the foundational development tasks and preliminary design necessary for the project's momentum.

#### **10.1.2 Team Composition**

- Ali took on the role of Scrum Master, facilitating sprint planning, stand-ups, reviews, and retrospectives to ensure the agile processes were adhered to. Ali also contributed as a Tester, verifying the functionality of features and ensuring they met the required quality standards.
- Hana and Hala were designated the task of creating and refining the Diagrams, which included use-case, class, sequence, and activity diagrams, crucial for the visual and technical representation of the system architecture and expected interactions.
- Amir was entrusted with the responsibility for the System Architecture, crafting the blueprint for the solution that aligns with the project's requirements and constraints.
- Mazen and Omar were the Implementers, focusing on coding the core functionalities as defined by the project's scope for this initial sprint.

## **10.2 Sprint Goals**

### **10.2.1 Planned versus Achieved**

The primary goals for Sprint 1 were to establish the foundational structures of the application by creating essential diagrams and developing core functionalities such as User Account Management, Income Tracking, Expense Tracking, and the initiation of the Reports and Analytics feature. The team successfully achieved the creation of all the planned diagrams, which laid out the system's architecture and user interaction flow.

In terms of functionality, the Account Management, Income Tracking, and Expense Tracking features were fully implemented. While these functionalities meet the current requirements, the team has identified opportunities for enhancing the user experience, which will be addressed in the upcoming sprint.

The development of the Reports and Analytics feature has begun, but it was decided to postpone its completion to Sprint 2 to ensure that it meets the high standards of quality and user experience that our application strives for.

### 10.2.2 Story Completion

During Sprint 1, two user stories were highlighted as the focus for our GUI personal finance tracking application:

1. Completed User Story: "As a user, I want to be able to register an account so that I can securely access my personal finance data."

This story was completed successfully. It included creating a secure registration system that allows users to create an account, secure it with a password, and retrieve their information through a login process.

2. In-Progress User Story: "As a financial enthusiast, I want to set a budget and savings goal within the app so that I can plan my finances according to my financial targets."

This story is currently in progress. The foundational features for tracking income and expenses have been developed, which form the necessary groundwork for implementing budget and savings goal functionalities. The completion of this story is contingent upon finalizing the comprehensive budgeting, savings goals, and analytics features, which will provide users with a holistic view of their financial health and planning capabilities.

## **10.3 Scope and Deliverables**

### **10.3.1 Scope Overview**

Sprint 1 of the Personal Finance Tracker project was dedicated to initiating core functionalities critical to the user experience. The team prioritized the development of the Account Management feature, enabling users to register, log in, and manage their profiles. Equally important was the implementation of the Income and Expense Tracking feature, allowing users to record and categorize their financial data efficiently. These features were complemented by a robust set of system diagrams, designed to provide clarity and direction for the application's architecture. An integral part of this sprint also involved rigorous testing protocols to ensure that these foundational features performed to specification and laid a solid basis for the application's future capabilities.

### **10.3.2 Deliverables**

The deliverables for this sprint consisted of comprehensive documentation and a functional codebase. Updated documentation, reflecting the evolution of the project's design and architecture, now includes new diagrams—such as use case narratives, activity diagrams, sequence diagrams, and class diagrams—that have been introduced following Milestone 1. These documents serve as a detailed reference that guides the development process and provides stakeholders with a clear understanding of the project's trajectory. Furthermore, the sprint's coding efforts have been consolidated into a GitHub repository, which now houses the initial implementation of the account management system and the transaction tracking features, as well as the associated test cases. This repository not only tracks the sprint's tangible outputs but also acts as a repository for collaborative development and version control moving forward.

## **10.4 Status Review**

### **10.4.1 Completed Items**

During Sprint 1, the team achieved a series of significant milestones that laid the groundwork for the Personal Finance Tracker application. All the planned diagrams were meticulously completed, ensuring a robust architectural blueprint for the system. These diagrams include comprehensive system and architecture visualizations, activity flows, sequence interactions, and class relationships. In addition to the diagrams, the team also successfully executed the feature development for account management, allowing users to register, log in, and manage their profiles securely. The income and expense tracking features were also fully implemented, providing users with the functionality to record and categorize their financial data accurately. Furthermore, rigorous testing was conducted to validate these features, confirming their operational integrity and readiness for user interaction.

### **10.4.2 In-Progress Items**

A focus on enhancing user experience has directed ongoing efforts, with several tasks earmarked for completion in the final sprint. Key improvements currently in progress include the implementation of personalized user interfaces, such as displaying the user's name upon login and refining the overall design of the software to ensure a more intuitive and engaging user experience. These enhancements are critical to user satisfaction and are being approached with a detailed and methodical strategy to ensure they meet the high standards set for the final product.

### **10.4.3 Delayed Items**

The Reports and Analytics feature, initially slated for completion in Sprint 1, has been deferred to the following sprint. This postponement was a strategic decision to allow for a more thorough development of the feature, ensuring it is fully functional and meets the project's



quality criteria. The delay will provide the team with the necessary time to focus on incorporating comprehensive reporting capabilities and detailed analytics, which are essential for users to gain insights into their financial health.

## **10.5 Quality and Testing**

### **10.5.1 Testing Overview**

Throughout Sprint 1, the team engaged in a rigorous testing process to ensure the quality and functionality of the newly developed features. This involved unit testing to validate individual components, integration testing to ensure that these components worked together seamlessly, and system testing to confirm that the application met all requirements as a whole. Test cases were derived from the requirements and designed to cover a wide range of scenarios, including both common and edge cases.

### **10.5.2 Defects and Issues**

The testing activities uncovered several defects that were immediately addressed by the development team. These included minor bugs related to user input validation and error handling in the account management feature, as well as formatting inconsistencies in the income and expense tracking modules. All identified issues were logged, prioritized according to severity, and most have been resolved. A small number of non-critical bugs remain outstanding, but plans are in place for their resolution in the upcoming sprint.

### **10.5.3 Code Quality Metrics**

To maintain high code quality, the team utilized static code analysis tools to detect potential vulnerabilities and maintain coding standards. Metrics such as cyclomatic complexity, code churn, and code coverage were monitored. The results indicate that the codebase maintains

a good level of quality, with most modules well within acceptable complexity thresholds. However, technical debt was identified in certain areas, primarily related to redundant code and suboptimal database interactions. Efforts to refactor these areas are planned for the next sprint to ensure the maintainability and scalability of the codebase.

## **10.6 Sprint Retrospective**

### **10.6.1 What Went Well**

The team excelled in task distribution, capitalizing on individual strengths to produce optimal outcomes. A significant milestone was the successful implementation and testing of the foundational features of the finance tracker, which was achieved on schedule. This accomplishment provided a tangible demonstration of the software's basic functionality and user experience. The team's division into two sub-teams, with the Scrum Master acting as a conduit and facilitator, proved effective in streamlining the work and ensuring the quality of deliverables. After overcoming initial communication hurdles, the team established a rhythm, with improved interactions towards the sprint's conclusion.

### **10.6.2 Areas for Improvement**

Early sprint communications posed a challenge but evolved into a strength through the adoption of regular meetings and consistent updates on work progress. The initial difficulty in understanding the requirements for deliverables highlighted the need for a clearer definition and internalization of project goals from the outset.

### **10.6.3 Lessons Learned**

The sprint underscored the importance of regular communication and setting clear deadlines for each task to ensure timely completion. A noteworthy revelation was the significance of detailed

planning before execution, which was more time-consuming yet critical for the success of the tasks. This approach has refined the team's workflow, emphasizing planning, peer review of strategies, and reporting on achievements at defined intervals.

#### **10.6.4 Plans for Improvement**

In light of the experiences from this sprint, our strategy for improvement entails a series of integrated measures designed to enhance team dynamics and project outcomes. We will initiate a structured communication protocol at the beginning of the upcoming sprint. This will include daily stand-ups for status updates and comprehensive weekly meetings to facilitate deeper discussions on progress and challenges. To ensure a consistent understanding of project goals and deliverables, we will dedicate a focused phase for thorough requirement gathering and analysis, complete with clear documentation.

### **10.7 Next Sprint Planning**

#### **10.7.1 Objectives for Next Sprint**

The forthcoming sprint is slated to be a pivotal phase in the project, with the team concentrating on completing the features outlined in the "4. Features to be Implemented" section of the documentation. The primary focus will be on the development of the Reports and Analytics feature, which is critical for providing users with insights into their financial health. This will be followed by the integration of the Budgeting and Savings features, which are essential to offering a comprehensive personal finance management tool. In tandem with these developments, there will be a concerted effort to enhance the application's design and user interface, ensuring an intuitive and engaging user experience.

### **10.7.2 Backlog Prioritization**

Given the sprint 1 outcomes, a strategic decision has been made to adjust the backlog prioritization. The Reports and Analytics feature has been given precedence, reflecting its delayed status from the previous sprint. Once this is underway, the team will shift their attention to the Budgeting and Savings Goals features, ensuring that these key components are implemented effectively.

### **10.7.3 Resource Allocation**

The next sprint will introduce a realignment of roles to capitalize on the team's strengths and streamline the development process. Mazen and Ali will apply their expertise to refine the Figma design of the application, aiming to elevate the visual and interactive aspects. Concurrently, Hana and Hala will focus on testing and quality assurance, ensuring that the application not only functions as intended but also adheres to the highest quality standards. Amir and Omar will continue their diligent work on the codebase, laying down the technical framework for the new features. In a collective effort, the entire team will contribute to the project documentation, promoting a holistic understanding and consistent progress reporting. This strategic redistribution of tasks is designed to bolster team synergy and enhance overall productivity, setting the stage for a successful sprint completion.

## **11. Sprint 2 Report**

### **11.1 Sprint Logistics**

Sprint 2, spanning from November 13. to November 26, 2023, marked a phase of strategic adaptation and focused execution within the team. Ali maintained his dual role as the Scrum Master and Tester, ensuring adherence to agile processes and quality standards. A significant shift was observed in task allocation, with Amir and Omar embracing a pair programming approach in QT Creator for more efficient coding practices. Mazen shifted his focus to the front-end design, utilizing Figma to conceptualize a more user-friendly interface for ECHO. Concurrently, Hana and Hala dedicated their efforts to refining the system diagrams, incorporating the feedback received from the previous sprint.

### **11.2 Sprint Goals**

The primary objectives for this sprint were centered around the development of the Budgeting feature, crafting an enhanced UI design in Figma, and laying the groundwork for future SQL database implementation. These goals were directly aligned with Echo's overarching mission to deliver a highly functional and intuitive personal finance tracker. The focus on UI and database schema design is particularly pivotal, as it underpins the software's usability and future scalability.

### **11.3 Scope and Deliverables**

The team's focus was two-fold: enhancing the Budgeting functionality and revising the Income and Expense tracking features. This involved altering the 'source' sub-functionality to allow users to select from general categories for their income and expenses, with an added option to include notes for personal reference. Another critical task was the initial design of the database schema, setting the stage for a more robust data management solution in the final sprint. The

tangible outputs from this sprint included updated source code, revised documentation, and the successful testing of the new Budgeting feature.

#### **11.4 Status Review**

This sprint witnessed the successful completion of the revised Budgeting feature and updates to the Income and Expense functionalities. However, the Saving Goal feature's development was deferred to the next sprint, underscoring the need for more focused time allocation for new feature implementations.

#### **11.5 Quality and Testing**

The quality assurance phase involved meticulous testing of the new Budgeting feature, where users interacted with the functionality to set budgets and allocate them to specific categories. A notable challenge surfaced in the form of a bug where budget totals failed to update correctly when categories were modified. This issue was swiftly resolved by correcting the update mechanism in the code, ensuring the feature's accuracy and reliability.

#### **11.6 Sprint Retrospective**

##### **What Went Well:**

- **Improved Team Communication:** The team successfully enhanced communication, which streamlined planning and coordination.
- **Forward-Looking Approach:** A proactive strategy was adopted for planning future sprints, particularly concerning the database setup.
- **Successful Task Execution:** The team completed the Budgeting functionality and updates to the Income and Expense features as planned.

### Challenges Faced:

- **Paired Programming Conflicts:** Scheduling conflicts arose during pair programming, slowing down the implementation process.
- **Task Distribution:** The sprint faced challenges in distributing tasks effectively due to a heavy skew towards technical tasks.

### Lessons Learned:

- **Importance of Independent Implementation:** The team realized the value of working independently on tasks to avoid bottlenecks caused by scheduling conflicts.
- **Core Feature Planning:** It was acknowledged that making changes to the core functionalities at later stages can be difficult, highlighting the need for thorough planning and execution from the outset.
- **Skill Development:** The sprint emphasized the need for continuous learning, particularly in areas where team members feel less confident.

### Plans for Improvement:

- **Enhanced Task Division:** Future sprints will focus on a more balanced distribution of tasks, considering both technical and non-technical aspects.
- **Collaborative Technical Support:** Team members plan to assist each other in technical tasks, fostering a collaborative learning environment.
- **Comprehensive Task Planning:** Emphasis will be placed on detailed task planning and setting aside time for skill development to ensure robust implementation of core features.
- **Scheduling Strategies:** The team will explore better scheduling methods for paired programming to avoid conflicts and ensure smooth progress.

## **11.7 Next Sprint Planning**

Looking ahead to the next sprint, the team plans to focus on implementing a SQL database, introducing the Saving Goal functionality, and developing the Reports feature. To enhance overall efficiency, there will be a more strategic division of implementation tasks and a collaborative approach to tackle technical challenges. This shift aims to maintain the project's momentum and ensure the successful realization of Echo's final features.



## 12. Sprint 3 Report (Final Sprint)

### 12.1 Sprint Logistics

#### 12.1.1 Sprint Timeline

The final sprint of Echo took place from November 26 to December 9, covering a period of two weeks. This sprint marked the culmination of the team's efforts in fully realizing the project's objectives and bringing the Personal Finance Tracker to a stage ready for deployment.

#### 12.1.2 Team Composition

- **Ali:** Continued in the role of Scrum Master, ensuring adherence to agile practices and efficient workflow.
- **Omar:** Served as the Lead Developer, focusing on implementing the remaining core functionalities of Echo.
- **Mazen:** Operated as the UI Developer, responsible for refining and implementing the user interface to enhance user experience.
- **Amir:** Assisted in the development of functionalities, collaborating closely with Omar to ensure seamless integration of features.
- **Hana and Hala:** Tasked with adjusting diagrams based on feedback from previous sprints, ensuring accuracy and coherence in the project's technical documentation.

## 12.2 Sprint Goals

The final sprint of Echo was strategically planned with definitive objectives that were crucial to the application's completion and operational readiness. These objectives, derived from the team's focus areas and project requirements, were as follows:

- **Developing Key Functionalities:** The team was dedicated to finalizing the report functionality. This feature is critical in providing users with comprehensive financial tracking and planning capabilities.
- **UI Implementation and Refinement:** The primary goal was to bring the Figma-based UI designs to life within the QT Creator environment, ensuring that the actual interface closely mirrored the planned design. This task was vital for maintaining a consistent and intuitive user experience.
- **Database Migration and Integration:** The transition of data storage from basic CSV files to a more robust QTSQL database.
- **Revising Diagrams and Documentation:** Updating the system diagrams and project documentation. Their task involved making adjustments based on feedback received in the previous sprint, ensuring that all technical materials accurately represented the final state of Echo.

Each of these goals was aligned with Echo's mission to provide a user-friendly and efficient personal finance management tool, and their successful completion was pivotal for the application's launch readiness.

## 12.3 Scope and Deliverables

### 12.3.1 Scope Overview

The scope of the final sprint was comprehensive, focusing on critical areas that would bring Echo to its full functionality and operational excellence.

- **Finalizing Reports:** The team dedicated significant efforts to develop and integrate the Reports functionality. These features are central to Echo's value proposition, offering users detailed insights into their financial habits and helping them achieve their saving objectives.
- **UI Implementation Based on Figma Designs:** A key focus was to ensure that the user interface, as designed in Figma, was accurately and effectively implemented in QT Creator. This task was essential to provide a seamless and engaging user experience.
- **Database Migration:** Transitioning from CSV file storage to a QTSQL database was a crucial part of this sprint. This migration aimed to enhance data management capabilities, paving the way for better performance, security, and scalability.

### 12.3.2 Deliverables

- **Developed Features:** Fully functional Reports features, ready for user interaction.
- **Implemented UI:** A polished and user-friendly interface, reflecting the designs conceptualized in Figma.
- **Migrated Database:** A successfully integrated QTSQL database, marking a significant improvement in data handling and storage.
- **Updated Documentation and Diagrams:** Revised technical documentation and system diagrams, incorporating feedback and showcasing the application's final architecture and functionalities.

## 12.4 Status Review

### 12.4.1 Completed Items

- **Reports Feature:** The Reports functionality, a key component of Echo, was successfully completed during this sprint. This feature now allows users to view and analyze their financial data, offering insights into their spending patterns and financial health.

### 12.4.2 In-Progress and Future Items

- **UI Enhancement:** While significant enhancements to the user interface were made, aligning closely with the Figma designs, there remains room for improvement. Future iterations of Echo will focus on refining the UI to further elevate the user experience.

## 12.5 Quality and Testing

### 12.5.1 Testing Overview

The final sprint placed a strong emphasis on ensuring the quality and stability of Echo. Rigorous testing protocols were implemented to validate the new features and the overall functionality of the application.

- **Comprehensive Testing of Reports Feature:** The newly developed Reports functionality underwent extensive testing. This included validating the accuracy of financial data representation and the reliability of different reporting features under various user scenarios.
- **UI Testing:** The enhanced user interface was rigorously tested for usability, responsiveness, and consistency with the Figma designs. User experience tests

were conducted to gather feedback on the interface's intuitiveness and ease of navigation.

### 12.5.2 Defects and Issues

During the intensive testing phase of the final sprint, the team encountered and addressed several key issues:

#### Reports Feature Bugs:

- **Data Accuracy Issue:** A bug was discovered where certain financial reports were not accurately reflecting the transaction data. This was traced back to a query error in the database.

#### User Interface Glitches:

- **Responsiveness Issue on Different Screen Sizes:** The UI did not consistently adapt to various screen sizes, affecting the layout and user experience. The team applied responsive design principles to ensure a uniform appearance across devices.
- **Inconsistent Button Behavior:** Some interactive elements, like buttons, were unresponsive under certain conditions. This was fixed by refining the event handling logic in the application's front-end code.

Each identified defect was systematically logged, analyzed, and addressed through a collaborative effort between developers and testers. This process not only resolved immediate issues but also provided insights into potential areas of improvement for future development cycles.

## 12.6 Sprint Retrospective

### 12.6.1 What Went Well

- **Successful Reports Feature Completion:** The team effectively finalized and integrated the Reports functionality, enhancing Echo's financial analysis capabilities.
- **UI Enhancements Achieved:** Significant improvements were made to the user interface, aligning it closely with the planned Figma designs and elevating the user experience.
- **Productive Testing and Debugging:** Rigorous testing led to the identification and resolution of several key bugs, improving the overall stability and reliability of the application.

### 12.6.2 Challenges Faced

- **Database Migration Plan Reassessment:** Initially planned migration to a QTSQL database was reassessed. Due to time constraints and further research, the team concluded that continuing with CSV files was the best approach for local data storage, considering the project's current scope and user needs.
- **Balancing Ambition with Practicality:** The team faced challenges in aligning ambitious technical goals with practical constraints, such as time and resources available.

### 10.6.3 Lessons Learned

- **Importance of Feasibility Analysis:** This sprint highlighted the need for thorough feasibility studies when considering significant technical shifts, especially concerning data management.
- **Adaptability in Project Plans:** The team learned to be more adaptable and make informed decisions based on current capabilities and project requirements.

#### 10.6.4 Plans for Improvement

- **Streamlined Integration Process:** The team aims to improve the integration of UI design and technical development, ensuring a more seamless and efficient workflow.
- **Revisiting Technical Strategies:** Future projects will include a more rigorous analysis of technical strategies from the outset, ensuring that decisions are both ambitious and practical.
- **Enhancing Project Management Practices:** The team plans to refine project management practices, focusing on better aligning project timelines with technical goals and ensuring more accurate feasibility assessments.
- **Robust Technical Training:** In response to the challenges faced with database integration, plans include investing in targeted training and knowledge-sharing sessions to better equip the team for handling similar situations in future projects.

## **13. Conclusion**

### **13.1 Reflective Summary**

As we conclude this document, we reflect on the journey of the Personal Finance Tracker software from concept to implementation. The detailed sections within have chronicled the thought process behind each feature, the careful consideration of the user's needs, and the technical rigor applied to ensure a reliable and secure financial management tool.

### **13.2 Achievements and Outcomes**

The successful completion of the initial sprints, as documented herein, signifies our commitment to delivering a product that not only meets but exceeds user expectations. The account management, income, and expense tracking functionalities are a testament to the team's ability to turn vision into reality.

### **13.3 Future Directions**

Looking ahead, the document has set the stage for the upcoming sprint's objectives, focusing on the integration of advanced features like budgeting, savings goals, and comprehensive analytics. These features are poised to elevate the user experience, offering a more profound insight into financial trends and forecasts.

### **13.4 Closing Note**

We trust that this document serves as a valuable resource for all stakeholders involved, providing guidance for current and future development efforts. As we continue to refine and enhance the Personal Finance Tracker software, we remain dedicated to our mission of empowering users with the tools they need to navigate their financial landscape confidently.