<Expense Tracker Web Application>

Requirements Analysis and Design

Version 1.0

Nov-10-2021

<T05>

Farah Sheherin

Allan John Valiente

Hyunsoo Cho

Yoonho Hwang

Table of Contents

Revision History

Date	Description	Author	Comments
Nov.07.21	Version 1.0	T05	First Revision

1.0 Introduction

The Introduction section provides an overview of the Expense Tracker web application using software requirements analysis and design for the scope of the system.

1.1 Purpose

This document describes the high-level software requirements for the Expense Tracker web application. It describes what, not how, of the capabilities of the Expense Tracker for the intended audiences.

1.2 Scope

The scope of this project covers site design and wireframe diagramming, coding to the approved wireframe, a web application for all the web browsers, Establishment of a test bed, testing and debugging prior to making the site public. Connection to bank accounts and application for mobile device lie outside of the scope.

2.0 System Overview

The System Overview section introduces the system context and design.

2.1 Project Perspective

The Expense Tracker web application provides service for users to record and track expenses. The following are the main features that are included in Expense Tracker:

- User account: The application allows the user to create their accounts in the system and provide features of updating and viewing profiles.
- Budget vs. Actual Expenditure: The application showcases the difference between expenditures and income and have total details of the budget and how much will be invested in real-time. This helps follow the budget and make solutions like reducing unwanted costs.
- Create reports: The application creates reports of expenses, loss and profits, balance sheets, and revenues.

2.2 System Context

Expense Tracker is a web application designed for individual and family users. Users can open a household user account, in addition to the usual individual account. For a household account, a primary user will be responsible for creating budgets for the household. A secondary user will have a very simple interface to enter expenses.

2.3 General Constraints

- Must be completed by April 2022
- Security considerations

2.4 Assumptions and Dependencies

This project makes the following assumptions:

- All the team members can complete their respective tasks within the schedule planned efficiently.
- The expected project's timeline can be met, and the project will complete within the expected time.
- The scope and specifications of the project will not undergo changes when the project takes place. However, when conducting the project, there might be cases where the scope and specifications need to be altered to cater to the requirements and needs of the project.

The following are the internal and external dependencies that will have to be acknowledged and addressed:

- Development of a web application cannot be started unless the project plans are approved.
- Development depends on design being finished.
- Testing cannot be conducted unless the website is designed and built.

3.0 Functional Requirements

In this section, we discuss the functional requirements in detail. Section 3.1 describes the requirements and their step-by-step description. Section 3.2 captures the functional requirements in the form of use-case tables.

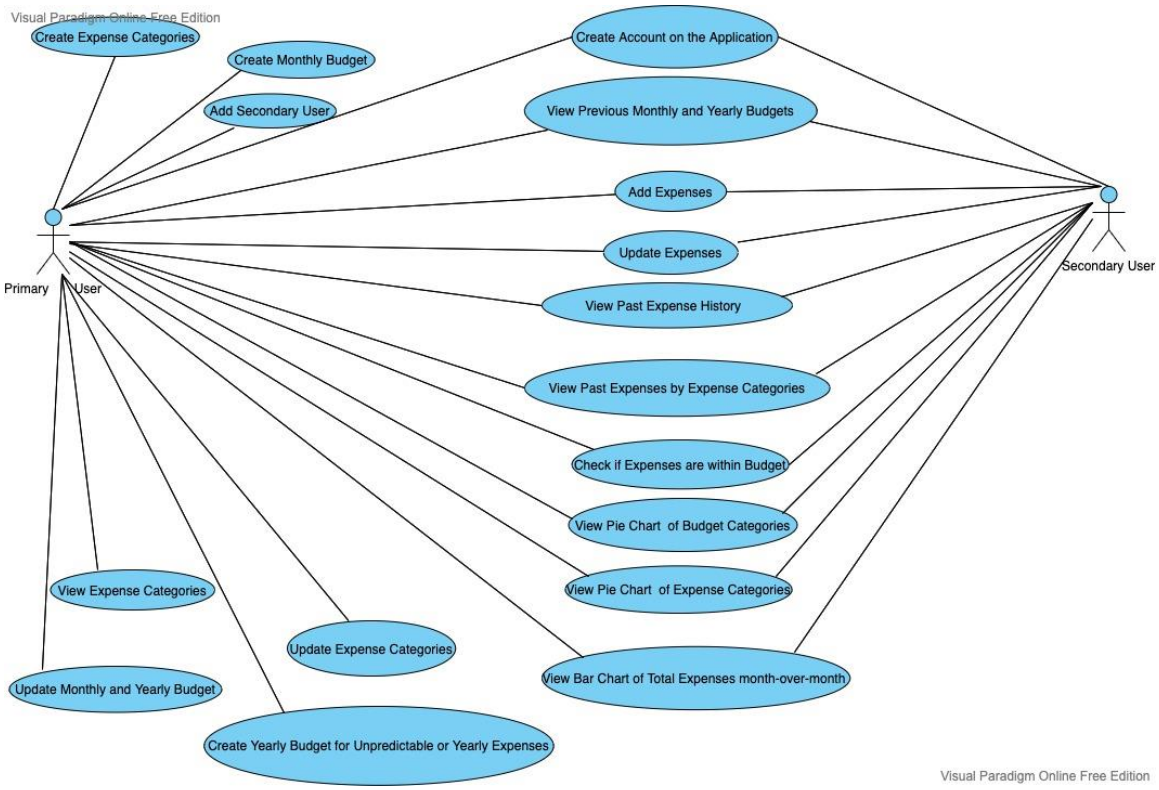


Figure: Use Case Diagram

3.1 Description of Functional Requirement

This section contains the functional requirements and their step-by-step descriptions.

Use case: **Create account on the application**

Brief Description

The user provides full name, email address and password to open an account with the application.

Initial Step-By-Step Description

Before this use case can be initiated, the user has already accessed the login page through the full URL or by clicking Sign Up button on home page.

1. The user enters her first name, then last name.
2. She enters her email address which will be used as the username for login later.
3. She enters a password.
4. The system checks if there's any existing users with the same email address.
5. If email address is new, the system encrypts the user's password and store all the given information in the database.
6. The user is successfully registered, and a success message is shown.

Use case: **Add secondary user**

Brief Description
The primary user adds secondary users to her account.

Initial Step-By-Step Description
Before this use case can be initiated, both the primary user and the secondary user must be registered with the application.

1. The primary user clicks “Add secondary user” button in settings page.
2. She enters secondary user’s email address (username).
3. If there’s an account associated with the given email address, the system links between primary and secondary users.
4. The system saves this associated in the database.
5. The primary user is shown a success message.

Use case: **Create expense categories**

Brief Description
Before allocating money to an expense category in the budgets, the user needs to create expense categories.

Initial Step-By-Step Description

1. User clicks “Expense Category” to go the expense category page, then click “New” button on the top of the page.
2. User enters category name, then click “Save” button.
3. The system will check if there’s any duplicate expense category.

Use case: **View expense categories**

Brief Description
User wants to view all expense categories.

Initial Step-By-Step Description
1. User clicks “Expense Category” to go the expense category page.
2. The “Expense Category” page lists all currently available expense categories.
3. User can check the existing categories.

Use case: **Update expense categories**

Brief Description
Sometimes users want to update category names for various reasons – better naming to reflect the category, misspelling etc.

Initial Step-By-Step Description

1. User clicks “Expense Category” to go the expense category page.
2. User clicks the edit button (“Pencil” icon) beside the category she wants to update.
3. User enters a new name, clicks the “Save” button.
4. The system checks if the expense category is not duplicated for this user’s account.

Use case: **Create monthly budget**

Brief Description

The primary user needs to set up a monthly budget to track expenses against a planned budget.

Initial Step-By-Step Description

1. User clicks “Budgets” menu and then clicks “New” button on the top of the page in order to create a new budget.
2. User enters the budget’s name, starting month and year, then clicks “Save” to start selecting expense categories and allocate money for each of them.
3. The system checks for budget name’s duplication. It also checks if the starting month and year is not in the past.
4. The system notifies the user of success.
5. The user then enters the amount of total monthly budget.
6. Then she adds multiple entries for expense category and allocate amount of money for that category. She clicks “Save” when finished for each expense category.
7. When user clicks “Save” button as she populates the budget with expense categories and allocated budget for each of them, the system checks if the total allocated money exceeds the budget amount set initially.

Use case: **Create yearly budget for unpredictable or yearly expenses**

Brief Description

Some expenses are unpredictable such as medical expenses, eyecare. But we need to budget for that. Some expenses are recurring but not on a monthly basis. Examples are yearly life insurance or renter insurance payment, clothes, shoes etc. The application allows a user to create a yearly budget to accommodate all these recurring and unpredictable expenses.

Initial Step-By-Step Description

1. User clicks “Yearly Budgets” menu and then clicks “New” button on the top of the page in order to create a new budget.
2. User enters the budget’s name, starting year, then clicks “Save” to start selecting expense categories and allocate money for each of them.
3. The system checks for budget name’s duplication. It also checks if the starting year is not in the past.
4. The system notifies the user of success.
5. The user then enters the amount of total yearly budget.

6. Then she adds multiple entries for expense categories and allocate amount of money for the categories. She clicks “Save” when finished for each expense category.
7. When user clicks “Save” button as she populates the budget with expense categories and allocated budget for each of them, the system checks if the total allocated money exceeds the budget amount set initially.

Use case: **Update monthly and yearly budgets**

Brief Description
Budgets are not set in stone. Because of changes to user’s circumstances, a user can update monthly and yearly budgets.

Initial Step-By-Step Description
1. User clicks “Monthly Budgets” or “Yearly Budgets” menu, whichever she wants to update.
2. The application shows the list of available budgets.
3. User clicks the edit button (“Pencil” icon) beside the budget she wants to update.
4. The application opens the selected budget in detail.
5. User can perform the following actions as part of update operation:
    a. Update total budget amount.
    b. Add expense category and allocate money.
    c. Remove expense category.
    d. Update allocated amount to expense categories.

Use case: **View previous monthly and yearly budgets**

Brief Description
Users may want to review the monthly and yearly budgets that they did in the past in order to gain insights into how their spending has evolved over time.

Initial Step-By-Step Description
1. User clicks “View past monthly budgets” menu or “View past yearly budgets” menu.
2. The application shows a side-by-side comparison of previous budgets. By default, the application shows comparison for the last 3 months.
3. User can select the range of months or years up to 12 months or 12 years in the past. That helps to minimize load on database.

Use case: **Add expenses**

Brief Description
Now that user has created monthly and/or yearly budgets, she wants to enter her day-to-day expenses in monthly budget and incidental expenses in a yearly budget.

Initial Step-By-Step Description
1. User clicks “Expenses” menu and “Add Monthly Expenses” or “Add Yearly Expenses” submenu.
2. User selects a date when the expense occurred, category of the expense, amount of expense. The expense is associated with the current month’s budget. The user can select a previous month if she wants.

Use case: **Update expenses**

Brief Description
To err is human. User can make mistakes while entering an expense. User may want to update an expense she added earlier.

Initial Step-By-Step Description
1. User clicks “Expenses” menu and then “Update Monthly Expenses” or “Update Yearly Expenses” submenu.
2. User selects the edit button (“Pencil” icon) beside the expense she wants to update.
3. User updates the expense item. Then she clicks “Save” button.
4. If user wants to update expense item of a previous budget, she needs to select a previous month or year from the dropdown list shown at the top of the page.

Use case: **View past expense history**

Brief Description
User wants to be able to view past expense history. This allows her to identify potential opportunities to reduce expenses in future.

Initial Step-By-Step Description
1. User clicks “Expenses” menu and “View Past Monthly Expenses” or “View Past Yearly Expenses” submenu.
2. The application shows expenses of previous months or years’ expenses. By default, the application shows expenses for the last 3 months or 3 years.
3. User can select a range of months or years up to a certain number of months or years in the past. This restriction is to minimize load on database.

Use case: **View past expenses by expense categories**

User wants to be able to view past expenses by expense categories month over month. This allows her to identify potential expense categories to eliminate in future.

Initial Step-By-Step Description

1. User clicks “Expenses” menu and “Compare Monthly Expense Categories” or “Compare Yearly Expense Categories” submenu.
2. The application shows a side-by-side comparison of previous months or years’ expenses by categories. By default, the application shows comparison for the last 3 months or 3 years.
3. User can select a range of months or years up to a certain number of months or years in the past. This restriction is to minimize load on database.

Use case: **Check if expenses are within budget**

Brief Description

User wants to know if her expenses are within the budgets she set earlier. This will allow her to know if the budgets need adjustments. This will also encourage users to modify their spending for rest of the month or year.

Initial Step-By-Step Description

1. User clicks “Expenses” menu and “View Budget Expenses” submenu.
2. The application shows expense categories from current month’s budget, amount of money spent in each category.
3. If user wants to see a previous month or year’s budget and total expense, she can select the month or year from the dropdown list at the top of the page.

Use case: **View pie chart of budget categories**

Brief Description

A picture is worth thousand words. The application shows user a pie chart of the budget categories. It enables the user to intuitively understand what percentage of her total budget is allocated to what expense categories.

Initial Step-By-Step Description

1. User clicks “Budgets” menu and “Budget at a Glance” submenu.
2. The application shows a pie chart with each expense category in the budget in a different color. The chart also includes data such as percentage of each expense category in relation to the total *budgeted* amount.
3. If user wants to see pie chart for a previous month or year, she can select the month or year from a dropdown lists at the top of the page.

Use case: **View pie chart of expense categories**

Brief Description
This is similar but not the same as the pie chart for budget categories. It is important that user can get insight into her spending by looking at a pie chart in order to know what percentage of her expenses so far belong to what categories.

Initial Step-By-Step Description
1. User clicks “Expenses” menu and “Expenses at a Glance” submenu.
2. The application shows a pie chart with each expense category in a different color. The chart also includes data such as percentage of each expense category in relation to the total *spent* amount.
3. If user wants to see pie chart for a previous month or year, she can select the month or year from a dropdown lists at the top of the page.

Use case: **View bar chart of total expenses month-over-month**

Brief Description
User wants to gain insight at a macro level – month over month. This will allow her to answer questions such as – which months are most expensive? Is there any pattern?

Initial Step-By-Step Description
1. User clicks “Expenses” menu and “Expenses month over month” submenu.
2. The application shows a bar chart with monthly expenses for the last 12 months.

3.2 Use Cases

This section structures the functional requirements in the form of use-cases.

3.2.1 Use Case # 1

Use Case Title #1: Create account on the application
Primary Actor: Any user (primary or secondary)
Level: Beginner
Stakeholders: User who’s going to register.
Precondition: User is on the sign-up page.
Minimal Guarantee: User has to sign up with an email address not registered yet.
Success Guarantees: Create an account successfully.
Trigger: User accesses sign-up page and enters full name, email address and password.

Main Success Scenario:

1. The user enters her first name, then last name.
2. She enters her email address which will be used as the username for login later.
3. She enters a password.
4. The system checks if there's any existing users with the same email address.
5. If email address is new, the system encrypts the user's password and store all the given information in the database.
6. The user is successfully registered, and a success message is shown.

Extensions:

1a. Given email address has already been used to open another account.

1a1. User is asked to provide another email address.

3.2.2 Use Case # 2

Use Case Title #2: Add secondary user
Primary Actor: Primary user
Level: Beginner
Stakeholders: Primary and secondary users.
Precondition: Both primary user and secondary user have accounts on the application.
Minimal Guarantee: Secondary user is successfully added.
Success Guarantees: Secondary user will be able to function with the primary user.
Trigger: Primary user clicks "Add secondary user" button in settings page.

Main Success Scenario:

1. On the "Add secondary user" page, the primary user enters the email address of the secondary user she wants to add.
2. The system checks if an account exists with the given email.
3. On success, the system shows success message to the primary user.

Extensions:

1a. The given email address of the secondary user is unavailable in the system. 1a1. The system asks the primary user to have the secondary user registered with the application first. 1b. Number of secondary users in the primary user's account exceed the limit. 1b1. The application notifies the primary user that she has reached the limit on secondary users.

3.2.3 Use Case # 3

Use Case Title #3: Create expense categories
Primary Actor: Primary user
Stakeholders: Both primary and secondary users
Precondition: User is logged in.
Success Guarantee: User will be able to create a unique expense category for her account.
Trigger: User clicks "Expense Category" to go the expense category page, then click "New" button on the top of the page.
Main Success Scenario: 1. User enters category name, then click "Save" button. 2. The system will check if there's any duplicate expense category. 3. If there's no duplication, the system will notify user of success.
Extensions: 1a. User already has an existing expense category with the same name. 1a1. The system notifies the user and does not create a duplicate. 1b. User reaches the maximum number of expense categories. 1b1. The system notifies the user and asks to check if any existing categories can be removed to make room for the new category.

3.2.4 Use Case # 4

Use Case Title #4: View expense categories
Primary Actor: Any user (primary or secondary)
Stakeholders: Any user (primary or secondary)
Precondition: User is logged in.
Success Guarantees: User is able to see all expense categories.
Main Success Scenario: 1. User clicks “Expense Category” to go the expense category page. 2. The “Expense Category” page lists all currently available expense categories. 3. User can check the existing categories.
Extensions: 1a. User has more expense categories than a single page can show. 1a1. The UI will have pagination in order for the user to go to the next page.

3.2.5 Use Case # 5

Use Case Title #5: Update expense categories
Primary Actor: Primary user of an account
Stakeholders: Both primary and secondary users of an account
Precondition: User is logged in.
Success Guarantees: Update an expense category successfully.
Trigger: User identifies an expense category that needs renaming.
Main Success Scenario: 1. User clicks “Expense Category” to go the expense category page. 2. User clicks the edit button (“Pencil” icon) beside the category she wants to update. 3. User enters a new name, clicks the “Save” button. 4. The system checks if the expense category is not duplicated for this user’s account.

3.2.6 Use Case # 6

Use Case Title #6: Create monthly budget
Primary Actor: Primary user of an account
Stakeholders: Both primary and secondary users of an account
Precondition: User is logged in.
Success Guarantees: Create a monthly budget successfully.
Trigger: User finds out that her current budget needs an overhaul.
Main Success Scenario: 1. User clicks “Budgets” menu and then clicks “New” button on the top of the page in order to create a new budget. 2. User enters the budget’s name, starting month and year, then clicks “Save” to start selecting expense categories and allocate money for each of them. 3. The system checks for budget name’s duplication. It also checks if the starting month and year is not in the past. 4. The system notifies the user of success. 5. The user then enters the amount of budget. 6. Then she adds multiple entries for expense category and allocate amount of money for that category. She clicks “Save” when finished for each expense category. 7. When user clicks “Save” button as she populates the budget with expense categories and allocated budget for each of them, the system checks if the total allocated money exceeds the budget amount set initially.

3.2.7 Use Case # 7

Use Case Title #7: Create yearly budget for unpredictable or yearly expenses
Primary Actor: Primary user of an account
Stakeholders: Both primary and secondary users of an account
Precondition: User is logged in.
Success Guarantees: A yearly budget is successfully created.
Main Success Scenario: 1. User clicks “Yearly Budgets” menu and then clicks “New” button on the top of the page in order to create a new budget. 2. User enters the budget’s name, starting year, then clicks “Save” to start selecting expense categories and allocate money for each of them. 3. The system checks for budget name’s duplication. It also checks if the

starting year is not in the past.
4. The system notifies the user of success.
5. The user then enters the amount of total yearly budget.
6. Then she adds multiple entries for expense categories and allocate amount of money for the categories. She clicks “Save” when finished for each expense category.
7. When user clicks “Save” button as she populates the budget with expense categories and allocated budget for each of them, the system checks if the total allocated money exceeds the budget amount set initially.

3.2.8 Use Case # 8

Use Case Title #8: Update monthly and yearly budgets
Primary Actor: Both primary and secondary users of an account
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in and the account has budgets.
Success Guarantees: Budget is updated successfully.
Main Success Scenario: 1. User clicks “Monthly Budgets” or “Yearly Budgets” menu, whichever she wants to update. 2. The application shows the list of available budgets. 3. User clicks the edit button (“Pencil” icon) beside the budget she wants to update. 4. The application opens the selected budget in detail. 5. User can perform the following actions as part of update operation: a. Update total budget amount. b. Add expense category and allocate money. c. Remove expense category. d. Update allocated amount to expense categories.

3.2.9 Use Case # 9

Use Case Title #9: View previous monthly and yearly budgets
Primary Actor: Both primary and secondary users of an account.
Stakeholders: Both primary and secondary users of an account.
Precondition: User is on the sign-up page.
Success Guarantees: The application shows previous budgets successfully.

Main Success Scenario:

1. User clicks “View past monthly budgets” menu or “View past yearly budgets” menu.
2. The application shows a side-by-side comparison of previous budgets. By default, the application shows comparison for the last 3 months.
3. User can select the range of months or years up to 12 months or 12 years in the past. That helps to minimize load on database.

3.2.10 Use Case # 10

Use Case Title #10: Add expenses
Primary Actor: Both primary and secondary users of an account
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in.
Success Guarantees: Expense item is successfully recorded in the system.
Main Success Scenario: 1. User clicks “Expenses” menu and “Add Monthly Expenses” submenu. 2. User selects a date when the expense occurred, category of the expense, amount of expense. The expense is associated with the current month’s budget. The user can select a previous month if she wants.

3.2.11 Use Case # 11

Use Case Title #11: Update expenses
Primary Actor: Both primary and secondary users of an account.
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in.
Success Guarantees: Expense items are successfully updated in the system.
Main Success Scenario: 1. User clicks “Expenses” menu and then “Update Monthly Expenses” or “Update Yearly Expenses” submenu. 2. User selects the edit button (“Pencil” icon) beside the expense she wants to update. 3. User updates the expense item. Then she clicks “Save” button. 4. If user wants to update expense item of a previous budget, she needs to

select a previous month or year from the dropdown list shown at the top of the page.

3.2.12 Use Case # 12

Use Case Title #12: View past expense history
Primary Actor: Both primary and secondary users of an account
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in.
Success Guarantees: User is able to see past expense history.
Main Success Scenario: 1. User clicks “Expenses” menu and “View Past Monthly Expenses” or “View Past Yearly Expenses” submenu. 2. The application shows expenses of previous months or years’ expenses. By default, the application shows expenses for the last 3 months or 3 years. 3. User can select a range of months or years up to a certain number of months or years in the past. This restriction is to minimize load on database.

3.2.13 Use Case # 13

Use Case Title #13: View past expenses by expense categories
Primary Actor: Both primary and secondary users of an account
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in and added expenses in the past.
Success Guarantees: User is able to see past expenses by expense categories.
Main Success Scenario: 1. User clicks “Expenses” menu and “Compare Monthly Expense Categories” or “Compare Yearly Expense Categories” submenu. 2. The application shows a side-by-side comparison of previous months or years’ expenses by categories. By default, the application shows comparison for the last 3 months or 3 years. 3. User can select a range of months or years up to a certain number of months or years in the past. This restriction is to minimize load on database.

3.2.14 Use Case # 14

Use Case Title #14: Check if expenses are within budget
Primary Actor: Both primary and secondary users of an account
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in.
Success Guarantees: User is able to see if expenses so far are within budget.
Main Success Scenario: 1. User clicks “Expenses” menu and “View Budget Expenses” submenu. 2. The application shows expense categories from current month’s budget, amount of money spent in each category. 3. If user wants to see a previous month or year’s budget and total expense, she can select the month or year from the dropdown list at the top of the page.

3.2.15 Use Case # 15

Use Case Title #15: View pie chart of budget categories
Primary Actor: Both primary and secondary users of an account.
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in and created budget.
Success Guarantees: User is able to see a pie chart of the budget categories.
Main Success Scenario: 1. User clicks “Budgets” menu and “Budget at a Glance” submenu. 2. The application shows a pie chart with each expense category in a different colour. The chart also includes data such as percentage of each expense category in relation to the total budgeted amount. 3. If user wants to see pie chart for a previous month or year, she can select the month or year from a dropdown lists at the top of the page.

3.2.16 Use Case # 16

Use Case Title #16: View pie chart of expense categories
Primary Actor: Both primary and secondary users of an account.
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in and added expenses.
Success Guarantees: User is able to see a pie chart expense categories for the

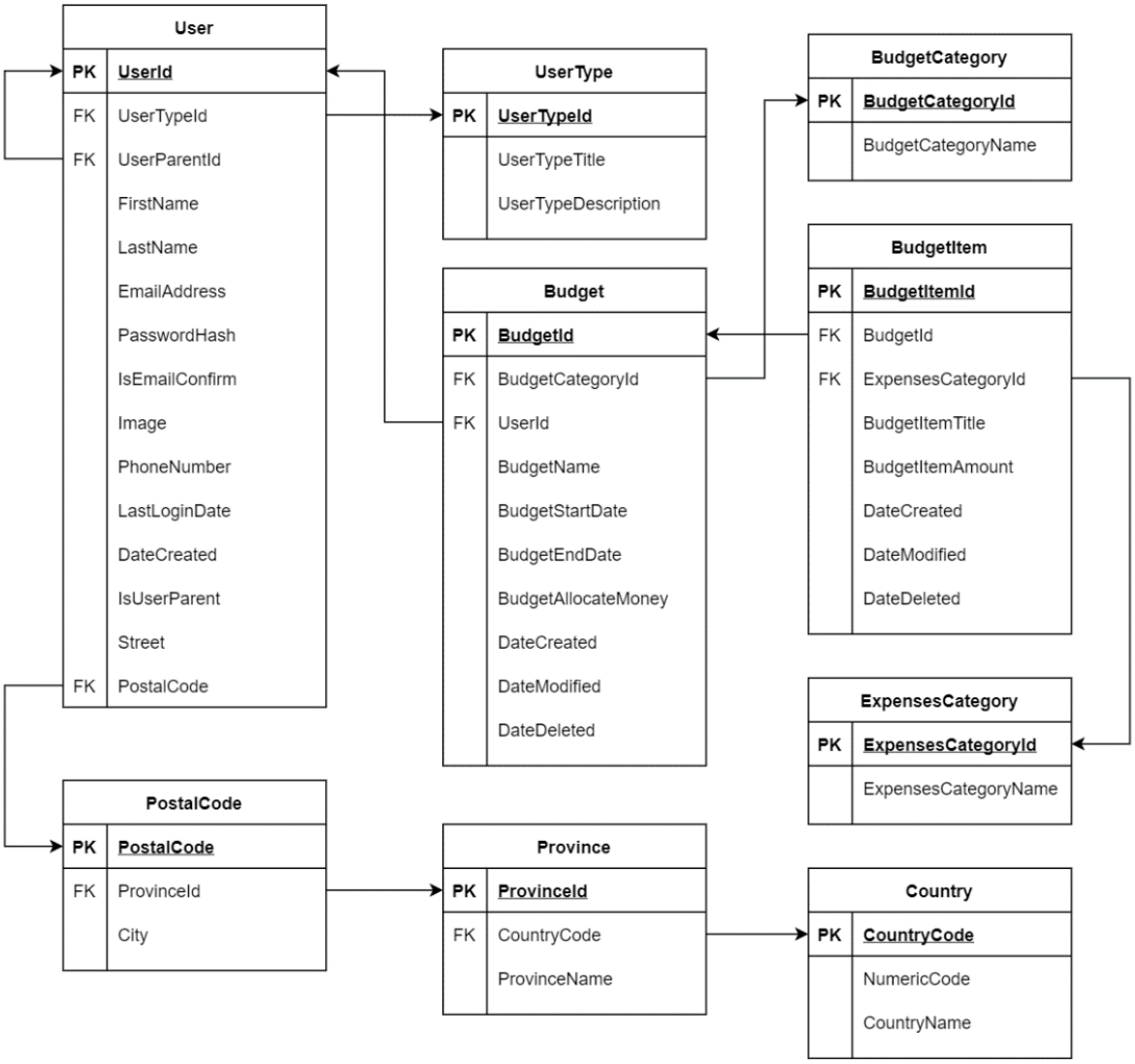
selected month.
Main Success Scenario: 1. User clicks “Expenses” menu and “Expenses at a Glance” submenu. 2. The application shows a pie chart with each expense category in a different colour. The chart also includes data such as percentage of each expense category in relation to the total spent amount. 3. If user wants to see pie chart for a previous month or year, she can select the month or year from a dropdown list at the top of the page.

3.2.17 Use Case # 17

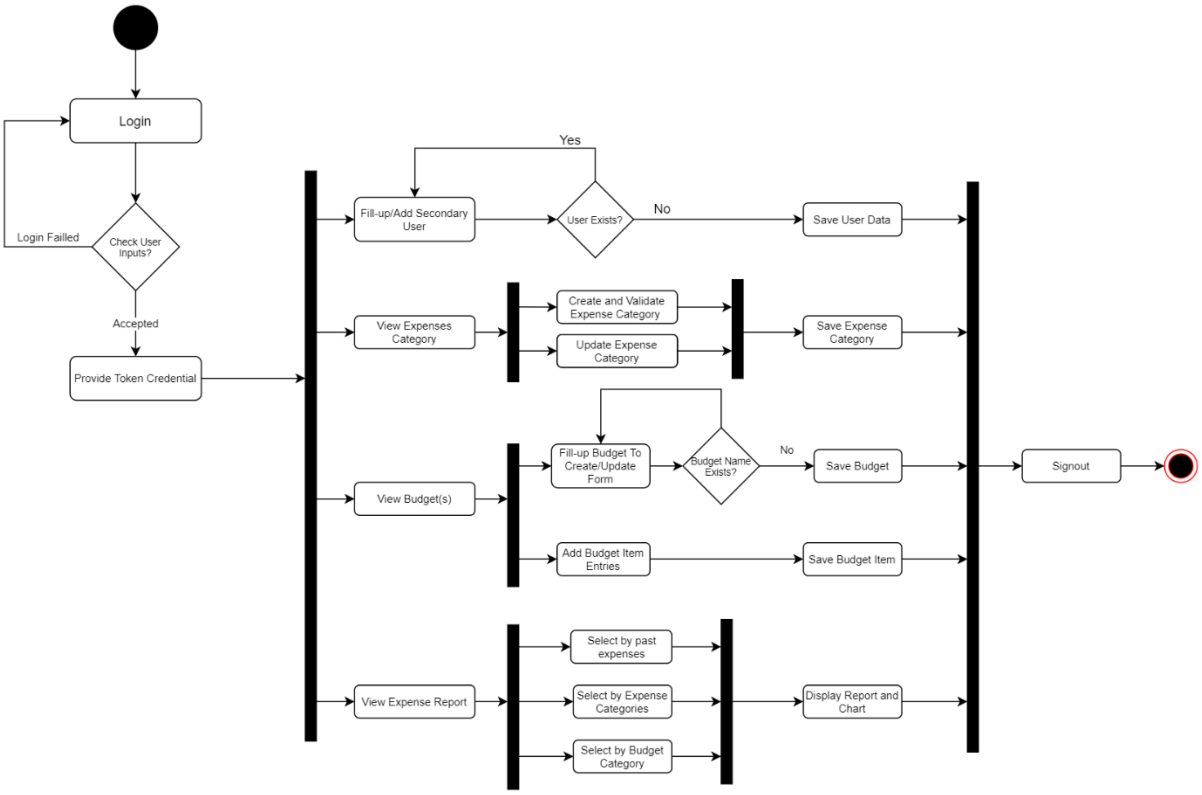
Use Case Title #17: View bar chart of total expenses month-over-month
Primary Actor: Both primary and secondary users of an account.
Stakeholders: Both primary and secondary users of an account.
Precondition: User is logged in and added expenses in the past.
Success Guarantees: User is able to see expenses month-over-month.
Main Success Scenario: 1. User clicks “Expenses” menu and “Expenses month over month” submenu. 2. The application shows a bar chart with monthly expenses for the last 12 months.

3.3 Data Modelling and Analysis

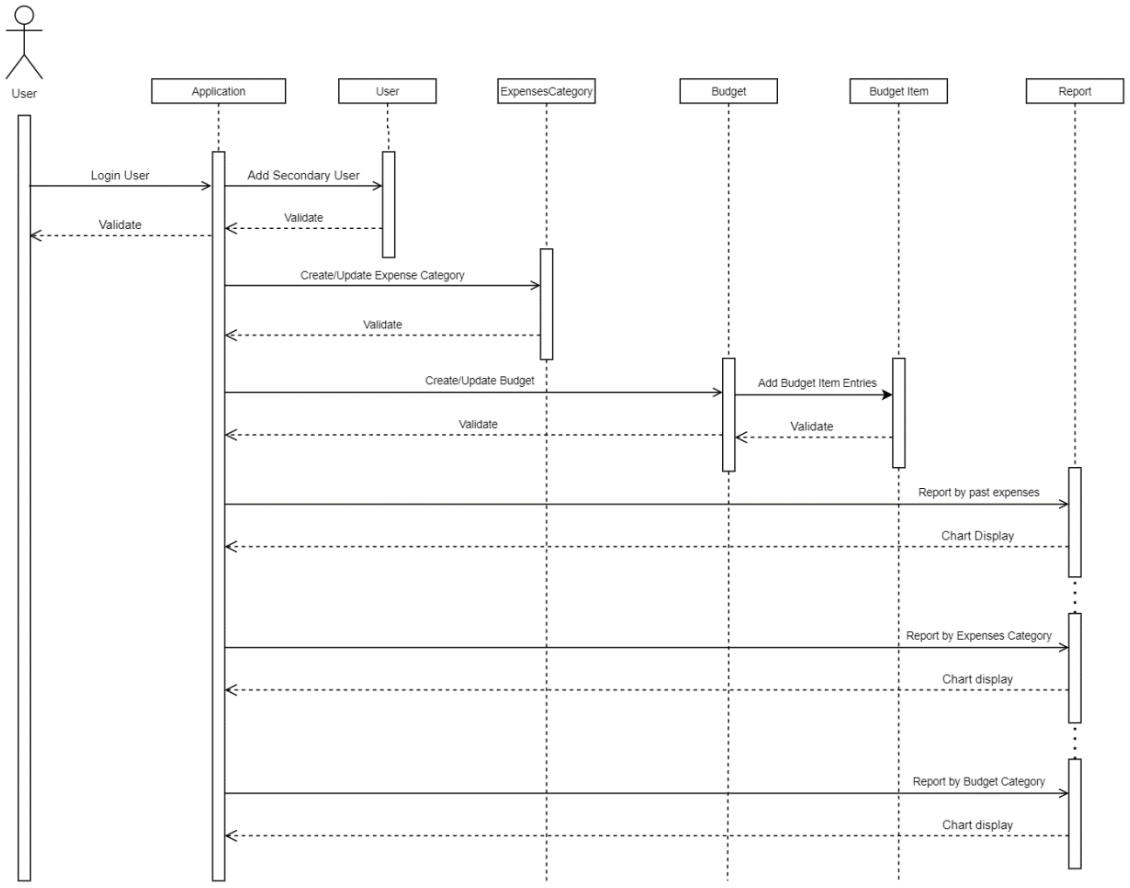
3.3.1 Normalized Data Model Diagram



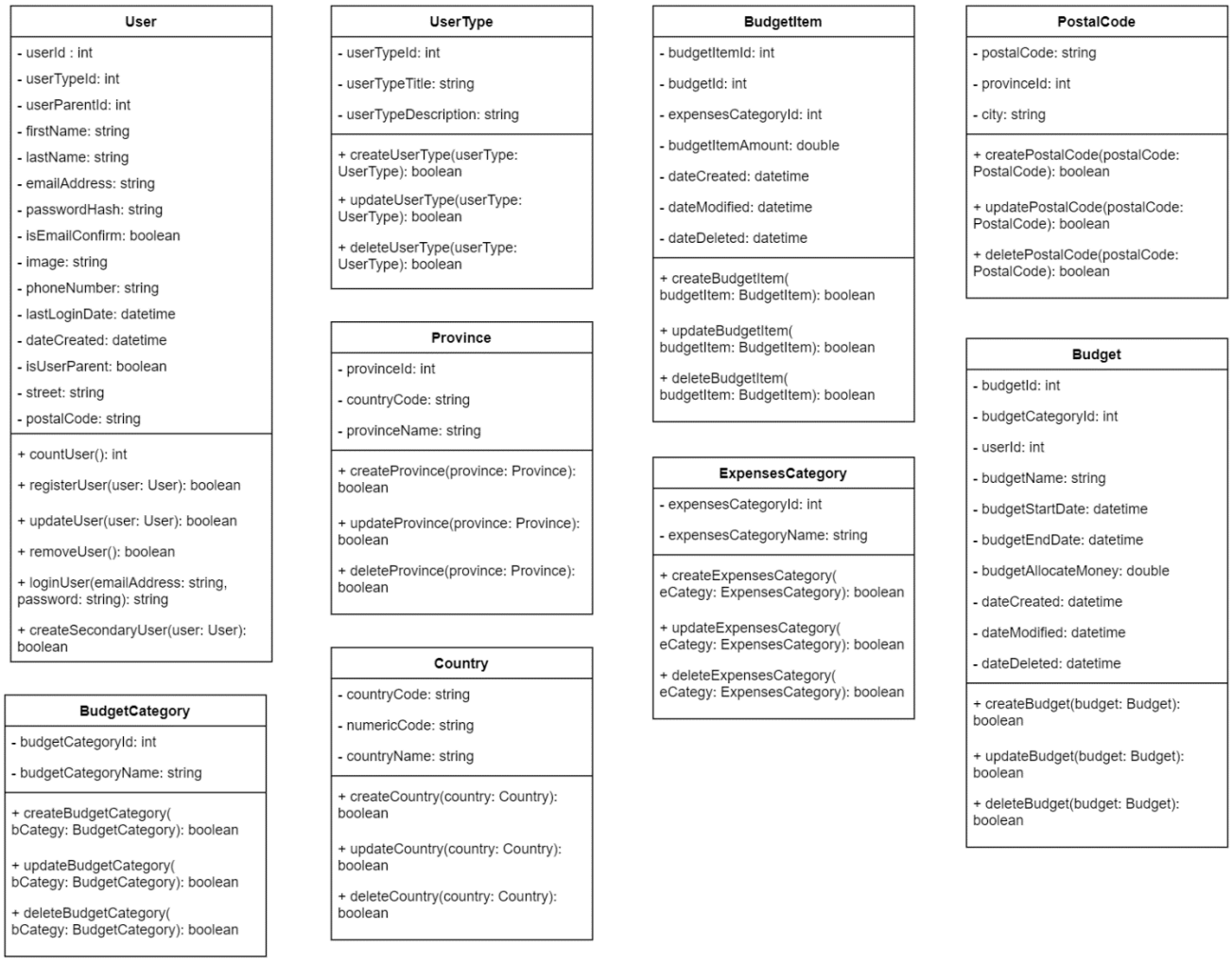
3.3.2 Activity Diagrams



3.3.3 Sequence Diagrams

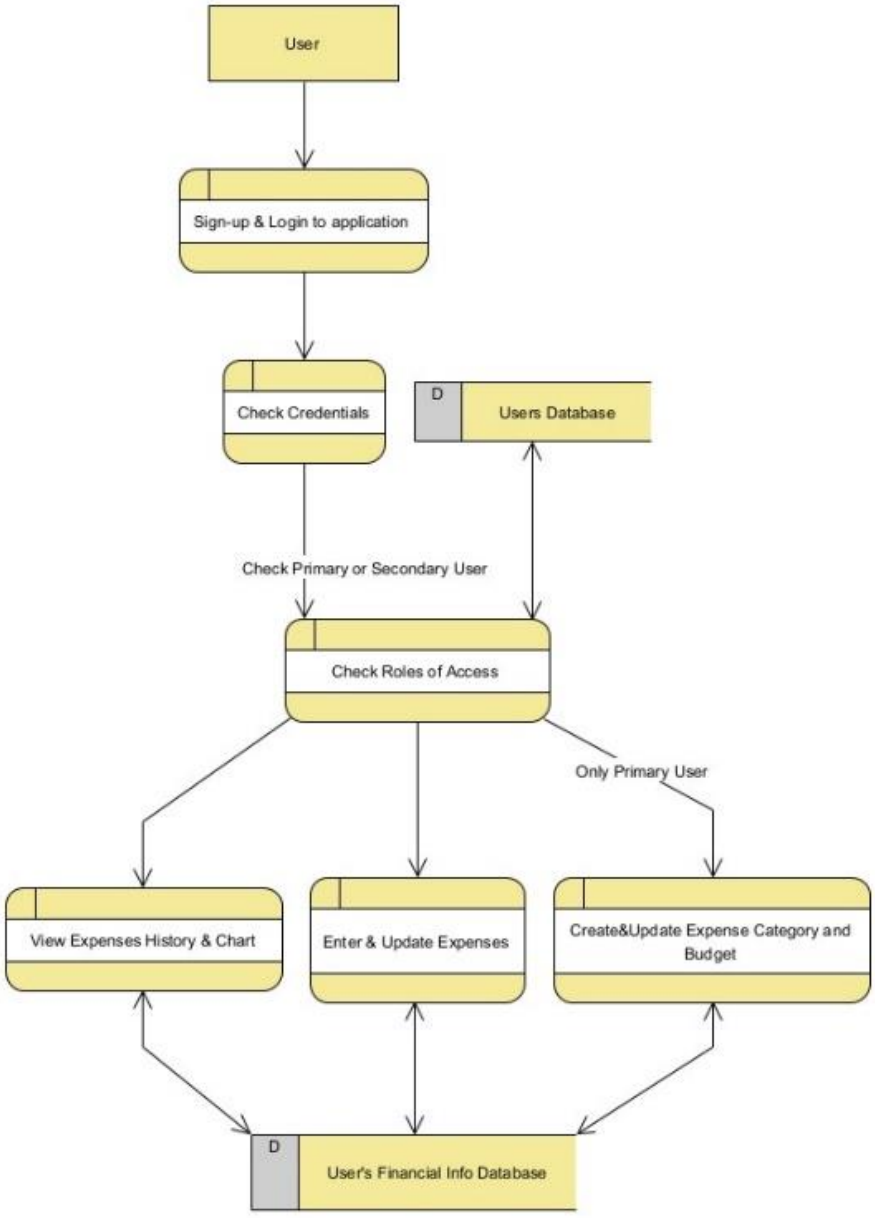


3.3.4 UML Class Diagram



3.4 Process Modelling

3.4.1 Data Flow Diagram



4.0 Non-Functional Requirements

4.1 Performance

- All the interface between a user and the Expense Tracker Web application shall have a maximum response time of two seconds.
- At least 20 percent of the CPU capacity shall be used in standard workload.
- Production of a simple report shall take less than 10 seconds for 99% of the cases.

4.2 Reliability

- The mean time to failure (MTTF) shall be no more than 1/100000.
- The user account update process shall roll back all related updates when any update fails to commit.

4.3 Availability

- The Expense Tracker Web application shall meet 90% uptime. (Downtime 36.5 days/year)
- Unless the web app is non-operational, the web app shall present a user with notification informing them that the web app is unavailable.

4.4 Security

- The access permissions for application data may only be changed by the application's data administrator.
- Passwords shall never be viewable at the point of entry or at any other time.
- The Expense Tracker Web application ensure that the user's all the data can be accessed only by authorized users.

4.5 Maintainability

- A development programmer who has at least three years of experience supporting this web application shall be able to add a new product feature, including source code modifications and testing.
- The Expense Tracker Web app shall not be shut down for maintenance more than once in a 24-hour period.

4.6 Portability

- The web application shall be developed for Windows, Mac, Linux operating systems.
- The meantime needed to replace the current Relational Database System with another Relational Database System shall not exceed 24 hours. No data loss should ensue.

5.0 Logical Database Requirements

The Expense Tracker Web application must store all the user account information as well as the user's financial records. The database allows concurrent access and will be always kept consistent, requiring a good database design. In the database, integer, double, string, boolean, datetime type of information will be held. Stored information will be used for messages, events. The email address gives the user option to receive any further information or update about the application.

6.0 Other Requirements

Additional requirements, if any.

7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Project Manager	Farah Sheherin		Nov-07-2021
Software Architect	Allan John Valiente		Nov-07-2021
Software Programmer	Hyunsoo Cho		Nov-07-2021
Software Programmer	Yoonho Hwang		Nov-07-2021