

NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

School of Computer Science and Engineering

SC/CE/CZ2006: Software Engineering

Assignment: *Tree Financial Tracker*

2022/2023 Semester 1

Tutorial/Lab Group	SE3
Team Number	3

Members:

KELVIN PANG	U2122086A
NG I-SHEN SAMUEL	U2121307G
LIM YAO XIAN	U2121572K
SIM SHI JIE, KEITH	U2121044B

TABLE OF CONTENTS

1. Application Overview	3
1.1. Purpose	3
1.2. Scope	3
1.3. Users and Stakeholders	3
1.4. Assumptions	3
1.5. Constraints	3
2. UI Mockup	4
3. Functional Requirements	8
4. Non-Functional Requirements	11
5. Data Dictionary	12
6. Use Case Diagram	13
7. Use Case Descriptions	14
8. Class Diagram	24
9. Sequence Diagram	25
10. Dialog Map	31
11. System Architecture	32
12. Black Box Testing	33
12.1. Login	33
12.2. Registration	34
12.3. Add Transactions	36
12.4. Add Financial Account	38
13. White Box Testing	40
13.1. Account Creation	40
13.2. Login	41
13.3. Financial Account	41
13.4. Budget	42
13.5. Transaction	42
APPENDIX	43

1. **Application Overview**

1.1. Purpose

Tree Financial Tracker is a web-based financial tracking application, targeted at money-minded individuals that live in Singapore. The application aims to aid users in being more frugal with their monetary expenses by tracking their financial habits. The budget calculator feature of the application also helps users ensure that they are spending within their limits in order to achieve their financial goals.

1.2. Scope

Tree Financial Tracker acts as a one-stop shop for users to collate all their financial information into a single application. Users can then add their daily transactions, ensuring that their financial information is always up to date. Users can also create budget goals for items that they want to purchase within a set period of time, and the application would recommend how much they should save per day/month and how much to set aside for other essentials, based on their historical transaction history.

1.3. Users and Stakeholders

Individuals who have an interest in saving, planning and keeping track of their finances are the intended users of this application.

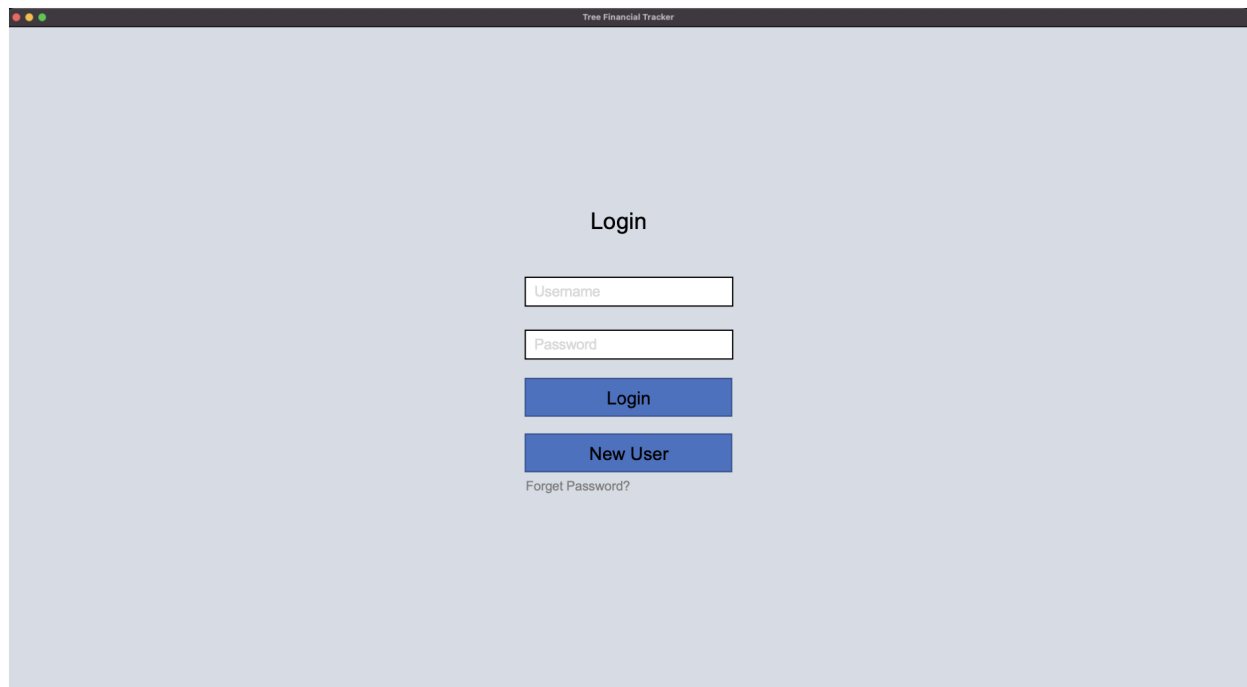
1.4. Assumptions

- Users must be connected to the internet.
- Users must use an internet browser that supports Tree Financial Tracker.

1.5. Constraints

- The application is only available in English.
- The application is primarily designed for desktop usage.

2. UI Mockup



The image shows a login page for a web application titled "Tree Financial Tracker". The page has a light blue background. In the center, there is a "Login" heading. Below the heading are two input fields: "Username" and "Password". Under the "Password" field is a blue "Login" button. Below the "Login" button is another blue button labeled "New User". At the bottom of the form area is a link that says "Forgot Password?".

Tree Financial Tracker

Login

Username

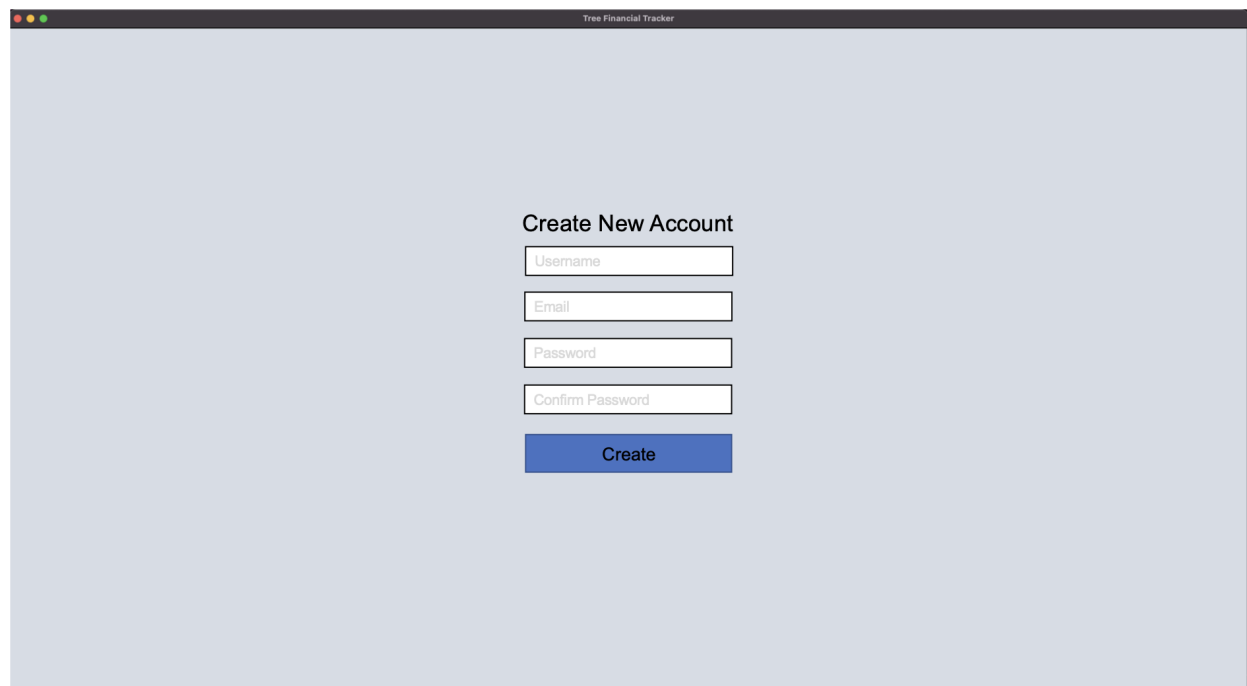
Password

Login

New User

[Forgot Password?](#)

Figure 1: Login Page



The image shows a registration page for a web application titled "Tree Financial Tracker". The page has a light blue background. In the center, there is a "Create New Account" heading. Below the heading are four input fields: "Username", "Email", "Password", and "Confirm Password". Below the "Confirm Password" field is a blue "Create" button.

Tree Financial Tracker

Create New Account

Username

Email

Password

Confirm Password

Create

Figure 2: Registration Page

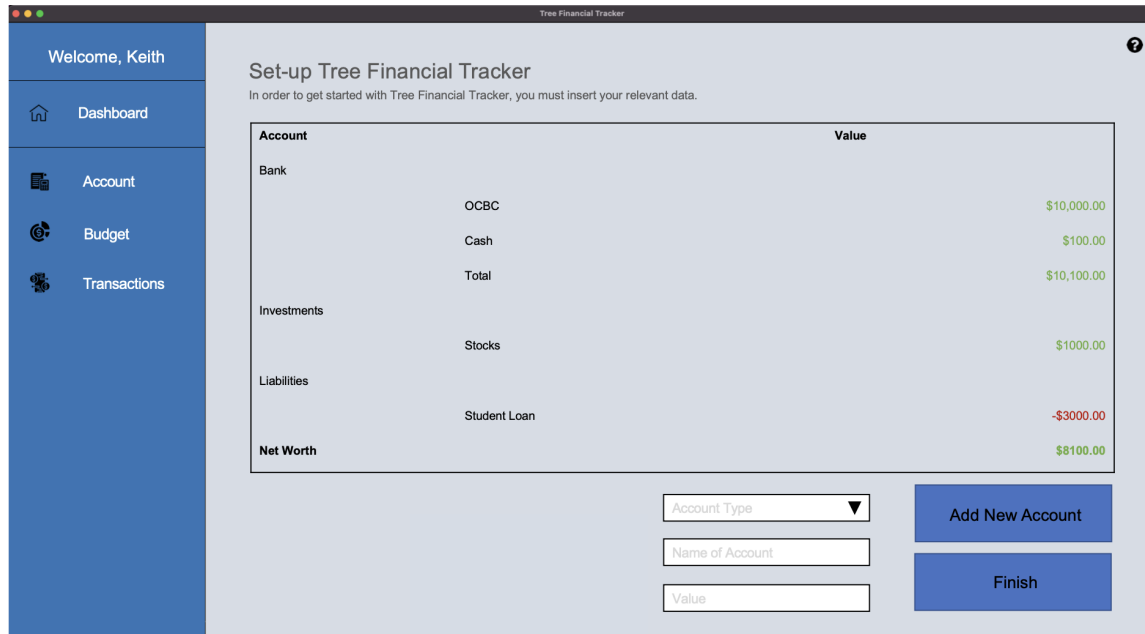


Figure 3: Initial Set-up Page

Figure 3 shows the user interface for the initial set-up page. The purpose of this page is for newly created users to add financial accounts such as banks, cash and liabilities such as student loans to calculate their current overall net worth.

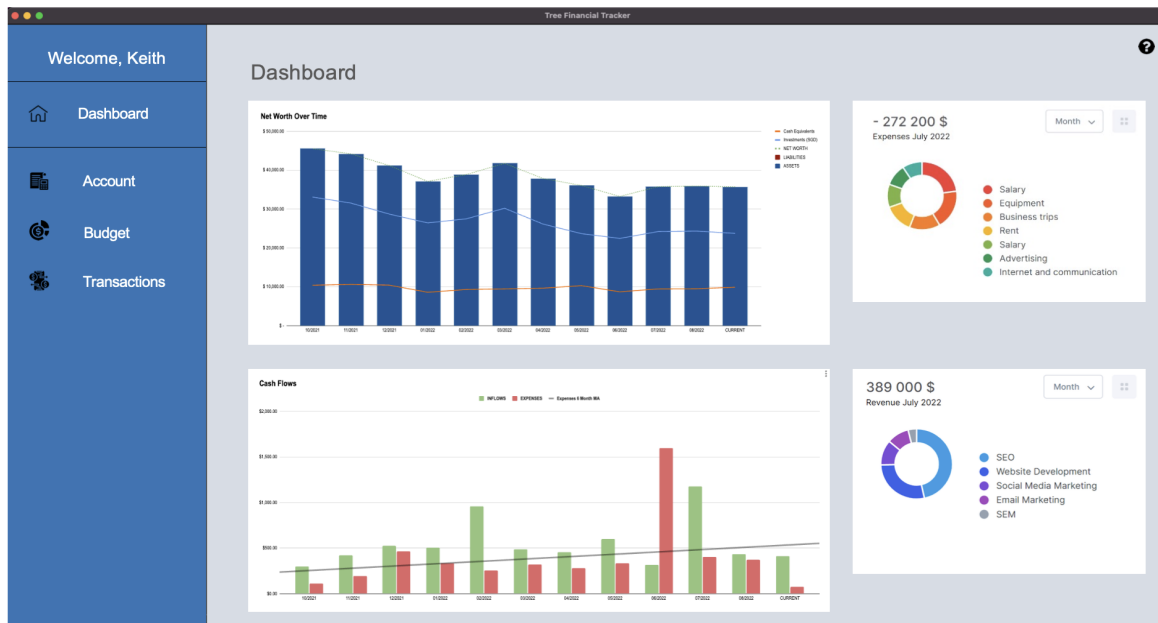


Figure 4: Dashboard Page

Figure 4 shows the user interface for the dashboard page. The purpose of this page is to illustrate to the user their overall financial standing at a glance. Users can easily tell their net worth and cash flow over a 12-month period and a breakdown of their expenditure and income.

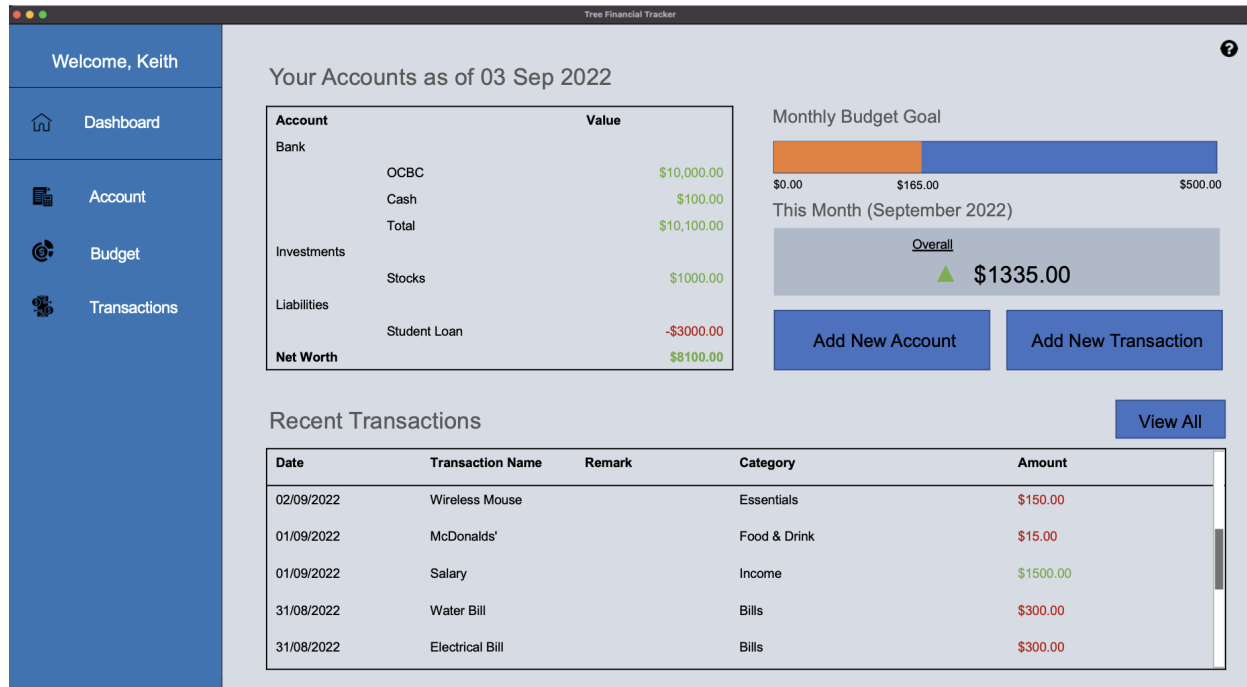


Figure 5: Account Page

Figure 5 shows the user interface for the account page. The purpose of this page is to show the user's current financial situation and the ten most recent transactions. It also shows the user's current month's cash flow and a progress bar to show if they are on track with their set budget goal. Users can also add new financial accounts and add new transactions from this page.

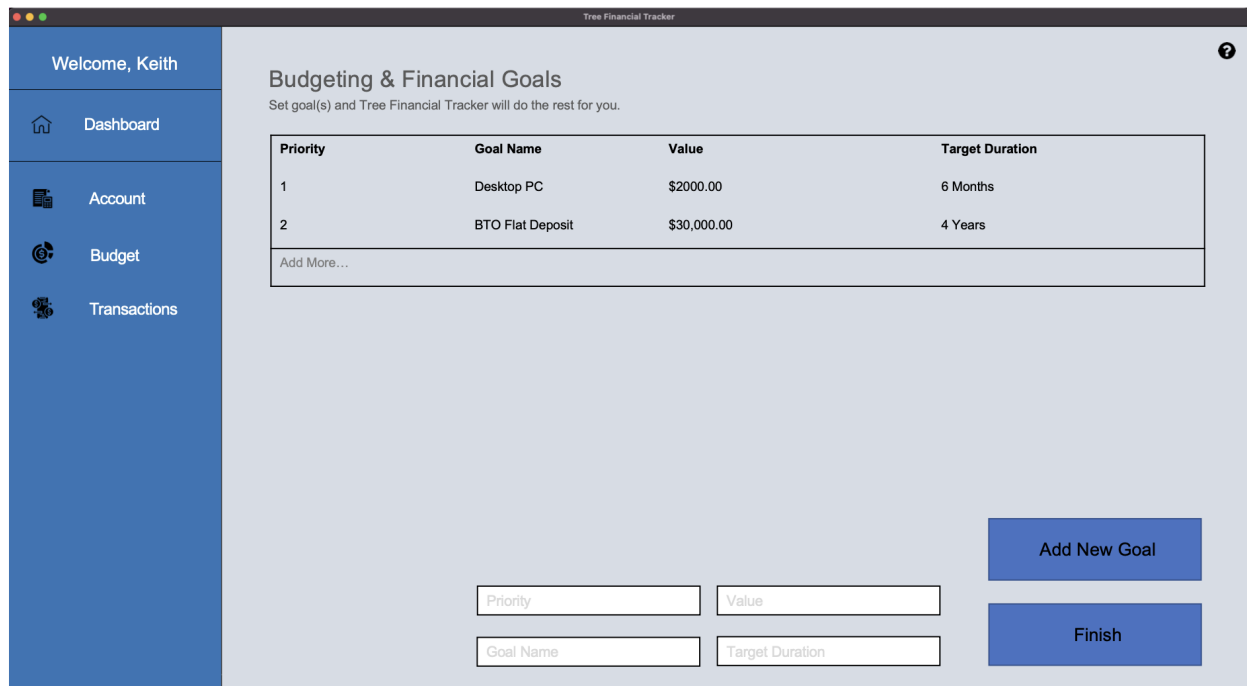


Figure 6: Budget Set-up Page

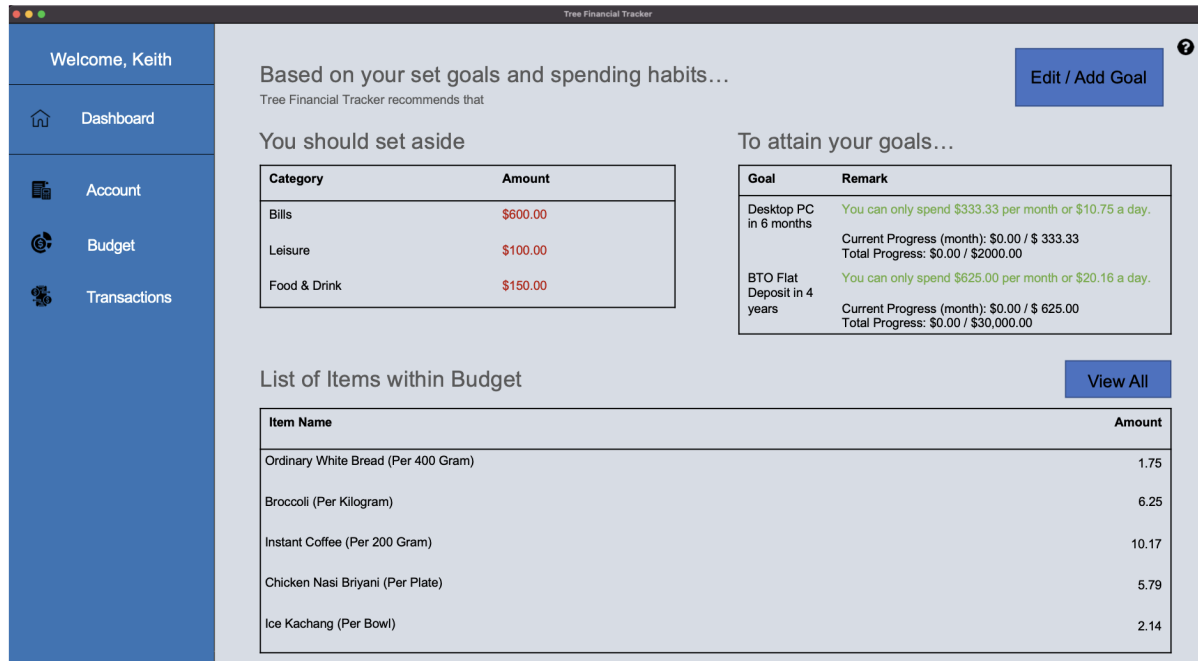


Figure 7: Budget Page

Figure 6 shows the user interface for the budget set-up page. The purpose of this page is for users to add new goals that they would like to achieve within a set period of time. Figure 7 shows the user interface for the budget page. The purpose of this page is to showcase the recommended budget calculated by the application based on the user's transaction history and set goal.

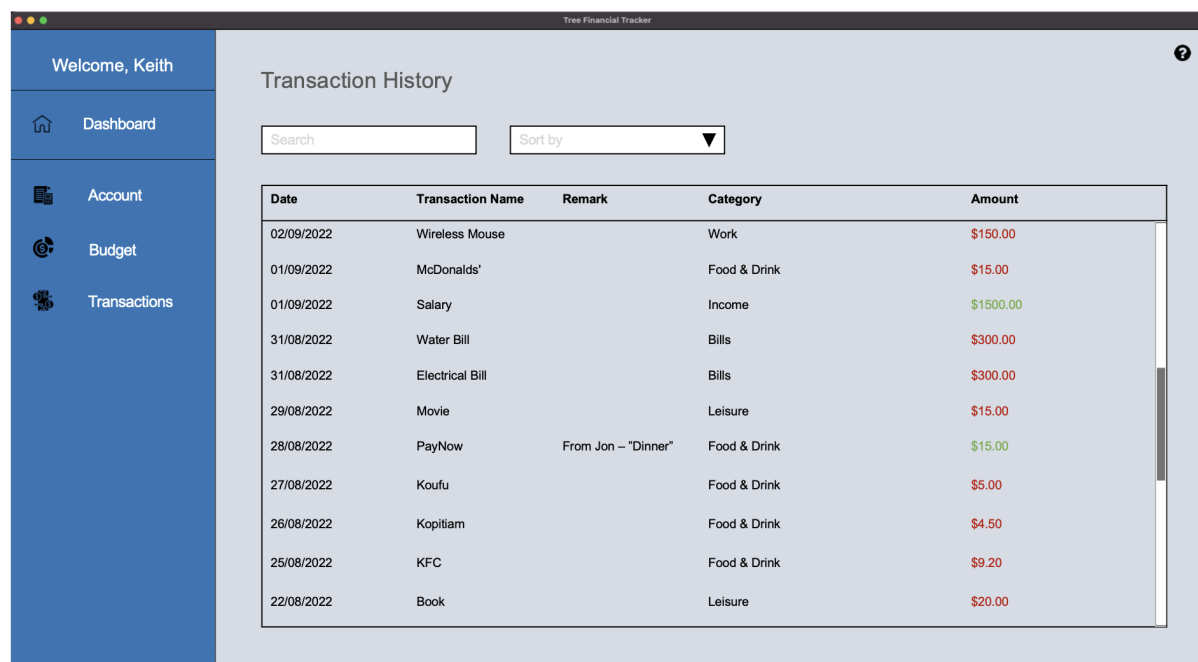


Figure 8: Transactions Page

3. **Functional Requirements**

1. Registration

- 1.1. The System must allow the user to register for a new user account.
 - 1.1.1. Information to include:
 - 1.1.1.1. Username
 - 1.1.1.2. First Name
 - 1.1.1.3. Last Name
 - 1.1.1.4. Valid Email Address
 - 1.1.1.5. Password & Confirm Password
 - 1.1.2. The system must validate that all required fields are filled up with valid values.
 - 1.1.3. The user account availability must be validated by the system.
- 1.2. The System must be able to validate their user account.
 - 1.2.1. Information to include:
 - 1.2.1.1. Verification Code
 - 1.2.2. The system must validate that the required field is filled up.
 - 1.2.3. The system must validate that the verification code is correct.
 - 1.2.4. The system will display an error message if the verification code is invalid.

2. Login

- 2.1. The user must be able to log in using their username and password.
- 2.2. The system must validate that all required fields are filled up with valid values.
- 2.3. The system must validate that the username and password are correct.
- 2.4. The system will display an error message if the login information is invalid.
- 2.5. The user must be able to change their password if they forgot their current password.
 - 2.5.1. The system must be able to email the user their verification code in order to reset their password.
- 2.6. The system will redirect the user to their dashboard upon successful login.

3. Dashboard

- 3.1. The system must allow new users to perform the initial set-up of their accounts.
 - 3.1.1. The user must be able to add new financial accounts.
 - 3.1.1.1. Information to include:
 - 3.1.1.1.1. Account Type
 - 3.1.1.1.2. Name of Account
 - 3.1.1.1.3. Value
- 3.2. The system must allow the user to add more financial accounts after the initial set-up.
- 3.3. The system must be able to display net worth as a graph.
- 3.4. The system must be able to display the user's cash flow over time.
- 3.5. The system must be able to display expenses by category in a pie chart.
- 3.6. The system must be able to display income by category in a pie chart.

4. Account

- 4.1. The system must be able to display the user's net worth, broken down into different financial accounts.
- 4.2. The user must be able to add a new transaction and its relevant details.
 - 4.2.1. Details include:
 - 4.2.1.1. Transaction Type
 - 4.2.1.2. Transaction Account
 - 4.2.1.3. Date
 - 4.2.1.4. Category
 - 4.2.1.5. Transaction Name
 - 4.2.1.6. Amount
 - 4.2.1.7. Remarks
- 4.3. The system must be able to display the monthly budget goal in the form of a progress bar.
- 4.4. The system must be able to display the cash flow for the current month.
- 4.5. The system must be able to display recent transactions made by the user.
- 4.6. The system must be able to redirect the user to the transaction page to view all transactions.
- 4.7. The user must be able to edit or delete their past transactions should the user choose to do so.

5. Budget

- 5.1. The user must be able to set financial goals by entering their goal information.
 - 5.1.1. Information Includes:
 - 5.1.1.1. Priority
 - 5.1.1.2. Goal Name
 - 5.1.1.3. Value
 - 5.1.1.4. Target Duration
- 5.2. The system must be able to display the estimated budget to the user
 - 5.2.1. The system must show how much money should be set aside by the user for essentials.
 - 5.2.1.1. Essentials Include:
 - 5.2.1.1.1. Food
 - 5.2.1.1.2. Housing
 - 5.2.1.1.3. Transportation
 - 5.2.1.1.4. Utilities
 - 5.2.1.1.5. Insurance
 - 5.2.1.1.6. Medical
 - 5.2.1.1.7. Personal
 - 5.2.1.1.8. Recreational
 - 5.2.1.1.9. Miscellaneous
 - 5.2.2. The system must display a suggested maximum amount the user can spend to attain their goals within the target duration.

- 5.2.2.1. The system must display the monthly progress and total progress towards the set goal.
 - 5.2.2.2. The system must display whether the user is on track to attain their goals.
 - 5.2.3. The system must be able to display the list of items that are within budget (e.g. Food, Dessert, Groceries, etc.) to the user.
 - 5.2.3.1. Information Includes:
 - 5.2.3.1.1. Item Name
 - 5.2.3.1.2. Amount
 - 5.2.3.2. The system must display a list of all items that are within the budget to the user via the table in the budget page.
- 6. Transaction
 - 6.1. The system must be able to show the entire transaction details and history of the user.
 - 6.1.1. Details include:
 - 6.1.1.1. Date
 - 6.1.1.2. Transaction Type
 - 6.1.1.3. Transaction Account
 - 6.1.1.4. Transaction Name
 - 6.1.1.5. Remarks
 - 6.1.1.6. Category
 - 6.1.1.7. Amount
 - 6.2. The user must be able to search by transaction name.
 - 6.3. The user must be able to sort by transaction details.
- 7. Interfacing with other systems
 - 7.1. The system must be able to retrieve the list of prices of common products and food items from Data.gov.sg API.
- 8. Data Format
 - 8.1. Date must be in "DD/MM/YYYY" format.
 - 8.2. Monetary values must be in "\$" and two decimal places of accuracy.

4. **Non-Functional Requirements**

1. Usability Requirements:

- 1.1. The system must be simple to use even for someone without much technical experience.
- 1.2. Font and text must be legible to all users.
- 1.3. The system must be consistent.
 - 1.3.1. Same processes must use the same actions (e.g. To show more details, right-click on the item.)
 - 1.3.2. Use the same visual cues for the same actions (e.g. The delete button option is shown whenever the user clicks on the name of the transaction).

2. Reliability Requirements:

- 2.1. The system must not crash when the user opens the application.
- 2.2. The system must be fully functional after a restart should a software crash occur.
- 2.3. The calculation made by the system must be accurate to **2 decimal places** (accuracy).

3. Performance Requirement:

- 3.1. The system must display the result to the user within **2 seconds**.
- 3.2. The system must take less than 3 seconds to load.
- 3.3. The system must be able to maintain without experiencing user latency.
- 3.4. The system must be able to retrieve information from external sources within 3 seconds.
- 3.5. The system must be able to provide an acceptable processing time of not more than 5 seconds.

4. Supportability Requirement:

- 4.1. The code used must be compatible with future updates of the code. For example, code that is going to be deprecated shall not be used.
- 4.2. Data collected will not be stored locally. It will be taken from the online database whenever needed. This ensures that the data extracted is the same regardless of platform or device.

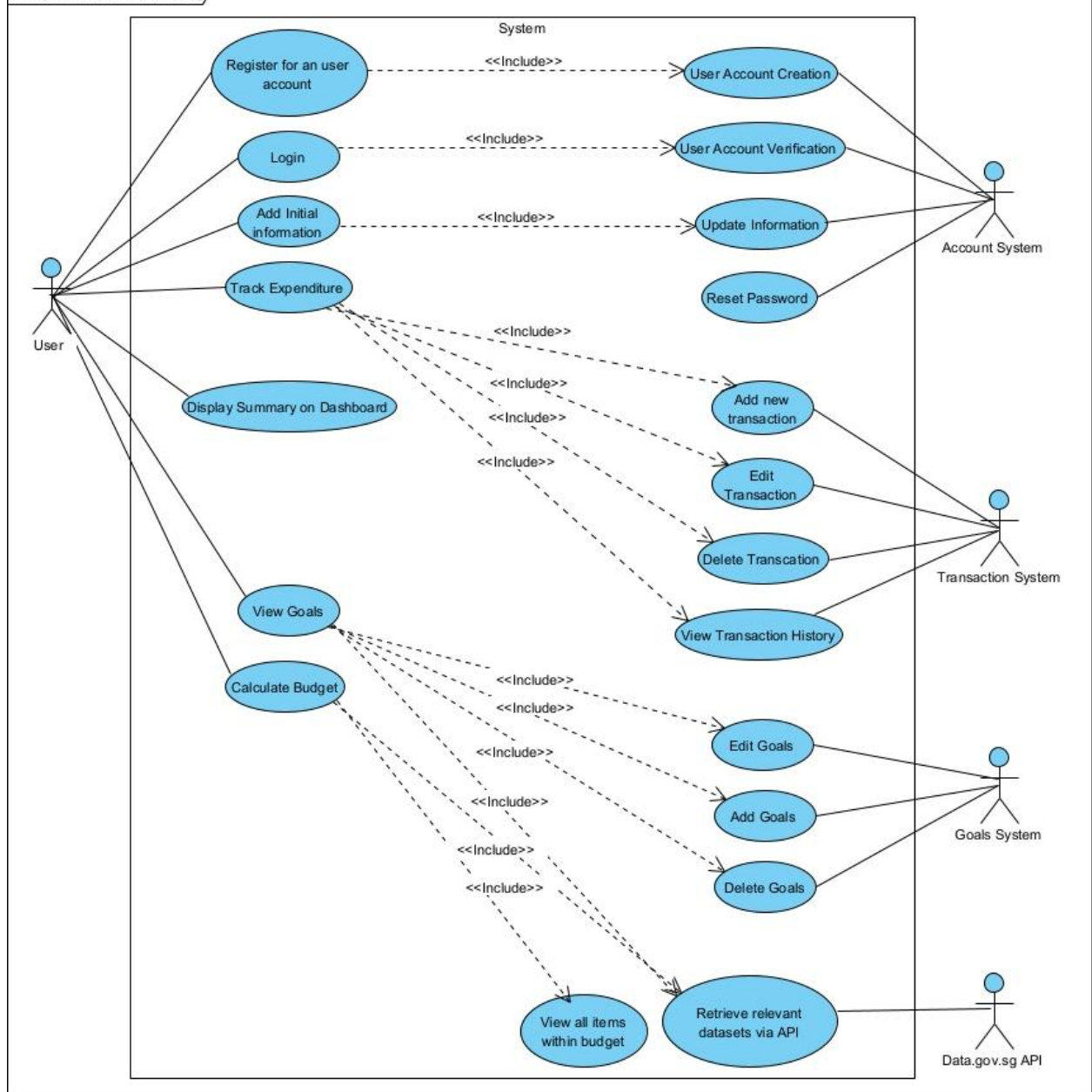
5. Data Dictionary

Term	Definition
User	A user is a person that uses the application to manage his/her budget in order to achieve his/her financial goal.
System	A system refers to the Tree Financial Tracker application.
Dashboard	A dashboard is a page where it displays key information in a visual manner to the user.
User Account	A user account is a user's personalized access to the system. It is used to keep track of daily expenses, cash flow over time, etc.
Financial Account	A financial account refers to the user's assets, liabilities or equity. (E.g. OCBC Bank or student loan)
Sort	Sort is a feature that will sort information in order. (E.g. Alphabetical order, Order by date, etc.)
Registration	Registration refers to a function that allows the user to create a user account in order for him/her to utilize the system.
Priority	A priority is a function that allows the user to set which goal they want to achieve first out of multiple goals set.
Target Duration	Target Duration is a function that allows the user to set a time frame to achieve a specific goal.
Search	Search is a feature that allows users to look for specific transactions based on the name of the transaction.
Set Aside	Set Aside is a function table that will recommend users on the amount to be set aside for each month based on the past transaction records.

6. Use Case Diagram

Visual Programming and Design (Department of Information Technology, University of the Pacific)

uc (Tree Financial Tracker)



7. Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Log in		
Created By:	Kelvin Pang	Last Updated By:	Keith
Date Created:	6/9/2022	Date Last Updated:	13/9/2022

Actor:	User, System
Description:	User logs in to the system.
Preconditions:	<ol style="list-style-type: none"> 1. The user must have an existing user account, 2. Devices must be connected to the internet.
Postconditions:	-
Priority:	High
Frequency of Use:	Less than 5 times per initial installation of application.
Flow of Events:	<ol style="list-style-type: none"> 1. Users provide their username and password to the respective input fields. 2. User clicks on the login button. 3. System ensures that the provided information is correct. 4. System redirects user to their account dashboard
Alternative Flows:	<p>AF-S1: User clicks on register instead of entering details.</p> <ol style="list-style-type: none"> 1. Redirect the user to the registration use case. <p>AF-S3: System detects invalid email/password.</p> <ol style="list-style-type: none"> 1. Prompt users to re-enter their credentials. 2. Return to step 1.. <p>AF-S1: Input fields are empty.</p> <ol style="list-style-type: none"> 1. Prompt users to re-enter their credentials.
Exceptions:	<p>EX-AF_S3: The device is not connected to the internet.</p> <ol style="list-style-type: none"> 1. Prompt users to connect to the internet.
Includes:	User account Verification
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Calculate Budget		
Created By:	Kelvin Pang	Last Updated By:	Yao Xian
Date Created:	6/9/2022	Date Last Updated:	8/11/2022

Actor:	User, Data.gov.sg
Description:	Calculation of budget
Preconditions:	1. User is already logged in.
Postconditions:	1. Users are able to view their monthly budget. 2. Users are able to view their goals. 3. Users are able to view the amount they have to save up. 4. Users are able to view a list of items recommended by the system.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User clicks on the budget button. 2. System will display the following: 1. Table of the list of items within the budget. 2. Table of a list of expenses to be set aside. 3. A Table to show the users how much they have to save up per month/day in order to achieve their goals 4. A table for the user to set/edit/delete their goals
Alternative Flows:	-
Exceptions:	-
Includes:	User account Verification
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Register for an user account		
Created By:	Samuel	Last Updated By:	Yao Xian
Date Created:	6/9/2022	Date Last Updated:	8/11/2022

Actor:	User, System
Description:	User creates an user account
Preconditions:	1. The user must not be logged in
Postconditions:	-
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks create an account button 2. System displays form for an User Account Creation - Username, First Name, Last Name, Email, Password, Confirm Password 3. User enters required information and submits form 4. System sends out a verification code to the user 5. User enters verification code on the required field 6. System stores information in database using the User Account Creation use case 7. System displays result of an User Account Creation Success 8. Return to login Page
Alternative Flows:	<p>AF-S3: System detects password and confirm password fields are different</p> <ol style="list-style-type: none"> 2. Prompt users to re-enter their credentials. 3. Return to step 1. <p>AF-S4: System detects invalid email is already in the database</p> <ol style="list-style-type: none"> 4. Display message "User account already exists" 5. Return to Login Page
Exceptions:	-
Includes:	User Account Creation
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Update Transactions		
Created By:	Samuel	Last Updated By:	Kelvin
Date Created:	6/9/2022	Date Last Updated:	17/9/2022

Actor:	User, System
Description:	User can view and add transactions
Preconditions:	<ol style="list-style-type: none"> 1. The user must be logged in 2. User is on accounts page.
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. User Clicks Add New Transaction 2. System displays form for Adding New Transaction - Transaction Type, Transaction Account, Date, Category, Transaction Name, Transaction Amount,Remarks 3. User enters transaction details and clicks Add New Transaction button
Alternative Flows:	<p>AF-S1: User clicks on the transaction name</p> <ol style="list-style-type: none"> 1. Display menu to edit or delete transaction 2. User clicks delete transaction 3. Transaction is deleted <p>AF-S1: User clicks on the transaction name</p> <ol style="list-style-type: none"> 1. Display menu to edit or delete transaction 2. User clicks edit transaction 3. Display menu for editing New Transaction - Transaction Type, Transaction Account, Date, Category, Transaction Name, Transaction Amount,Remarks 4. User updates transaction details 5. User clicks Edit Transaction button 6. System returns to detailed transaction page
Exceptions:	-
Includes:	Add new Transaction, Retrieve account information, Edit Transaction, Delete Transaction, View Transaction History
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Update Goals		
Created By:	Yao Xian	Last Updated By:	Keith
Date Created:	6/9/2022	Date Last Updated:	13/9/2022

Actor:	User
Description:	For the users to set/update their budget goals. Budget goals are calculated based on the list of items which they want to buy and prioritized according to their preferences.
Preconditions:	1. User is already logged in.
Postconditions:	1. Users are able to view how much they have saved up. 2. Users are able to view their progress through a progress bar.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. Users click on Budget. 2. Users enter required information and submit a form. 3. System verifies all required information is filled. 4. System sets/updates goals specific to the user's account 5. Goals are set/updated
Alternative Flows:	AF-S1: User had previously set up goals 1. User clicks on edit/add goal button 2. Return to step 2. AF-S2: System detects empty form or form filled up with invalid values 1. System displays error message "Please enter valid values into the form" 2. Prompt users to re-enter required information. 3. Return to step 3.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	View Goals		
Created By:	Yao Xian	Last Updated By:	Keith
Date Created:	6/9/2022	Date Last Updated:	17/9/2022

Actor:	User
Description:	For the user to view their goals
Preconditions:	1. The user must be logged in
Postconditions:	-
Priority:	-
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. Users click on Budget. 2. System must display a calculated budget based on previously set goals and list of items within the budget.
Alternative Flows:	AF-S2: User does not have a single goal <ol style="list-style-type: none"> 1. Redirect user to the Update Goals use case.
Exceptions:	-
Includes:	Retrieve relevant datasets via API , Delete Goals, Set Goals, Add Goals, Edit Goals
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Display Summary on Dashboard		
Created By:	Samuel	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	6/9/2022

Actor:	User
Description:	Display a summary of users information
Preconditions:	<ol style="list-style-type: none"> 1. The user must be logged in 2. The user must have some financial accounts added 3. The user must have some transactions added
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. User Clicks Dashboard 2. System displays user account information - Net worth over time, Cash Flow, Expenses, Revenue
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	8		
Use Case Name:	Add account Information		
Created By:	Samuel	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	8/11/2022

Actor:	User, System
Description:	Adds Bank account information
Preconditions:	1. User must be logged in
Postconditions:	-
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. User Clicks Account 2. Display Account Page 3. User Clicks Add New Account 4. Display Add new Account Page 5. User enters account details 6. User Clicks Add new Account 7. Account Information is updated
Alternative Flows:	-
Exceptions:	-
Includes:	Update Information, Retrieve account information
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

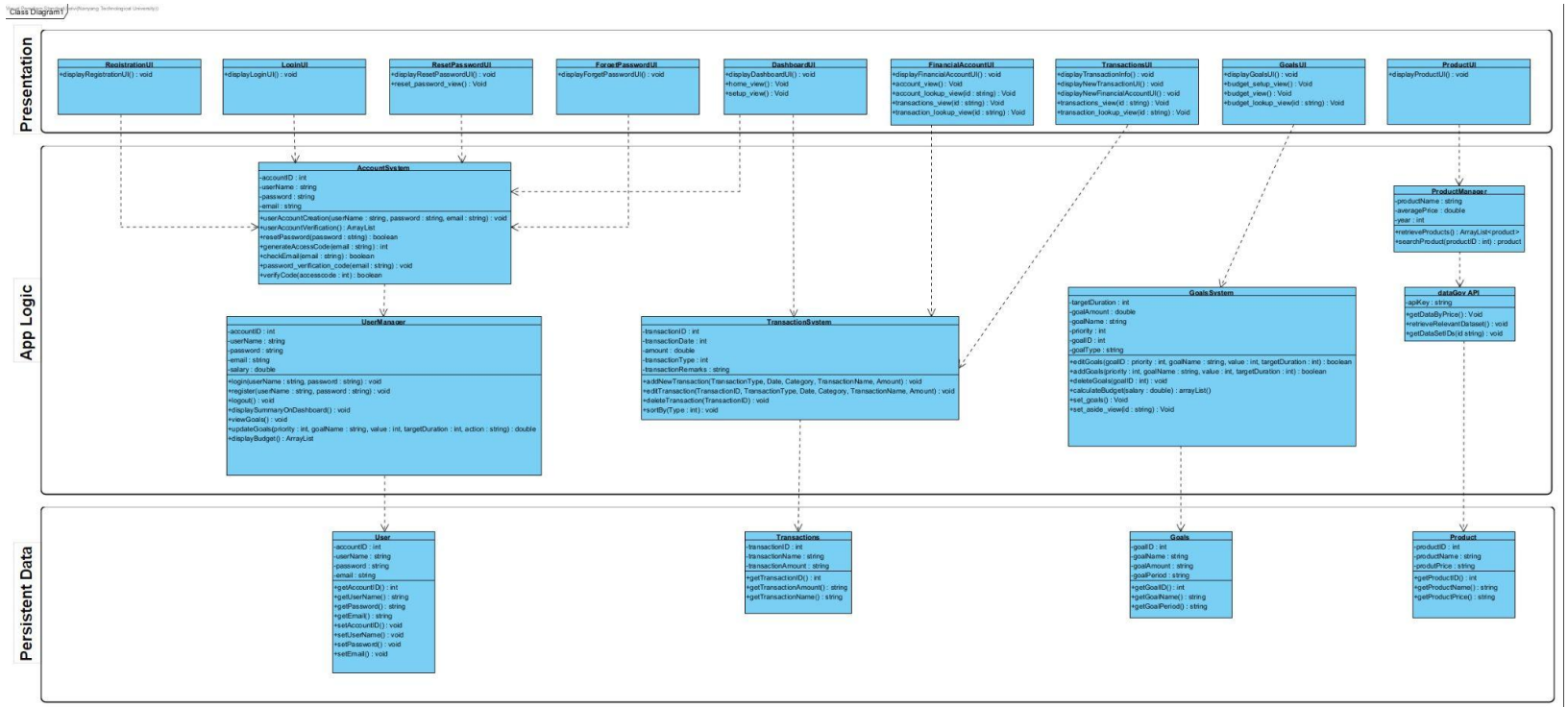
Use Case ID:	9		
Use Case Name:	Add initial Information		
Created By:	Samuel	Last Updated By:	Samuel
Date Created:	6/9/2022	Date Last Updated:	8/11/2022

Actor:	User, System
Description:	Adds Financial account information
Preconditions:	2. User must be logged in
Postconditions:	-
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. New User logs in for the first time 2. User enters account details 3. User Clicks Add new Account 4. Account Information is updated
Alternative Flows:	-
Exceptions:	-
Includes:	Update Information, Retrieve account information
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	10		
Use Case Name:	Forget Password		
Created By:	Kelvin Pang	Last Updated By:	Yao Xian
Date Created:	13/9/2022	Date Last Updated:	8/11/2022

Actor:	User, System
Description:	Allows the user to reset their passwords if the user needs to do so.
Preconditions:	<ol style="list-style-type: none"> 1. User must not be logged in. 2. Network must be available. 3. The user should know their email or user id.
Postconditions:	
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks "Forget password?" 2. System prompts user to enter their email. 3. User provide their email address to the respective input fields. 4. System ensures that the email is valid. 5. System emails a verification code to the user's registered email. 6. System prompts the user for the verification code. 7. User enters the received code. 8. System verifies the code. 9. System prompts user to enter a new password. 10. User enters their new password. 11. System checks that password meets the criteria. 12. System updates the user's password. 13. System redirect user to login page.
Alternative Flows:	<p>AF-S3: Entered email or id is invalid:</p> <ol style="list-style-type: none"> 1. Return to Step 2. <p>AF-S8: Entered the wrong verification code:</p> <ol style="list-style-type: none"> 1. Return to Step 6. <p>AF-S11: Entered password does not meet criteria:</p> <ol style="list-style-type: none"> 1. Return to Step 9.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

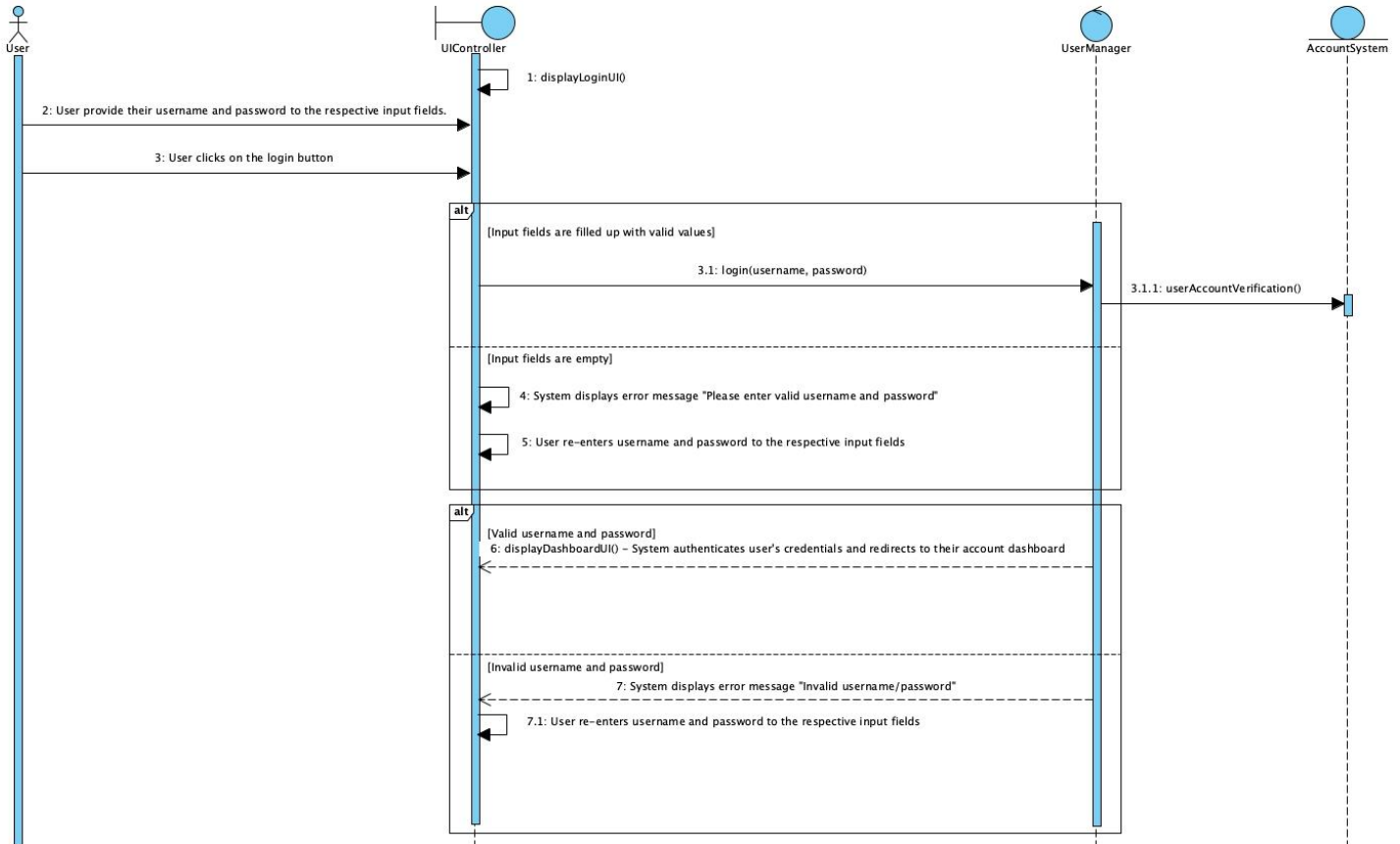
8. Class Diagram



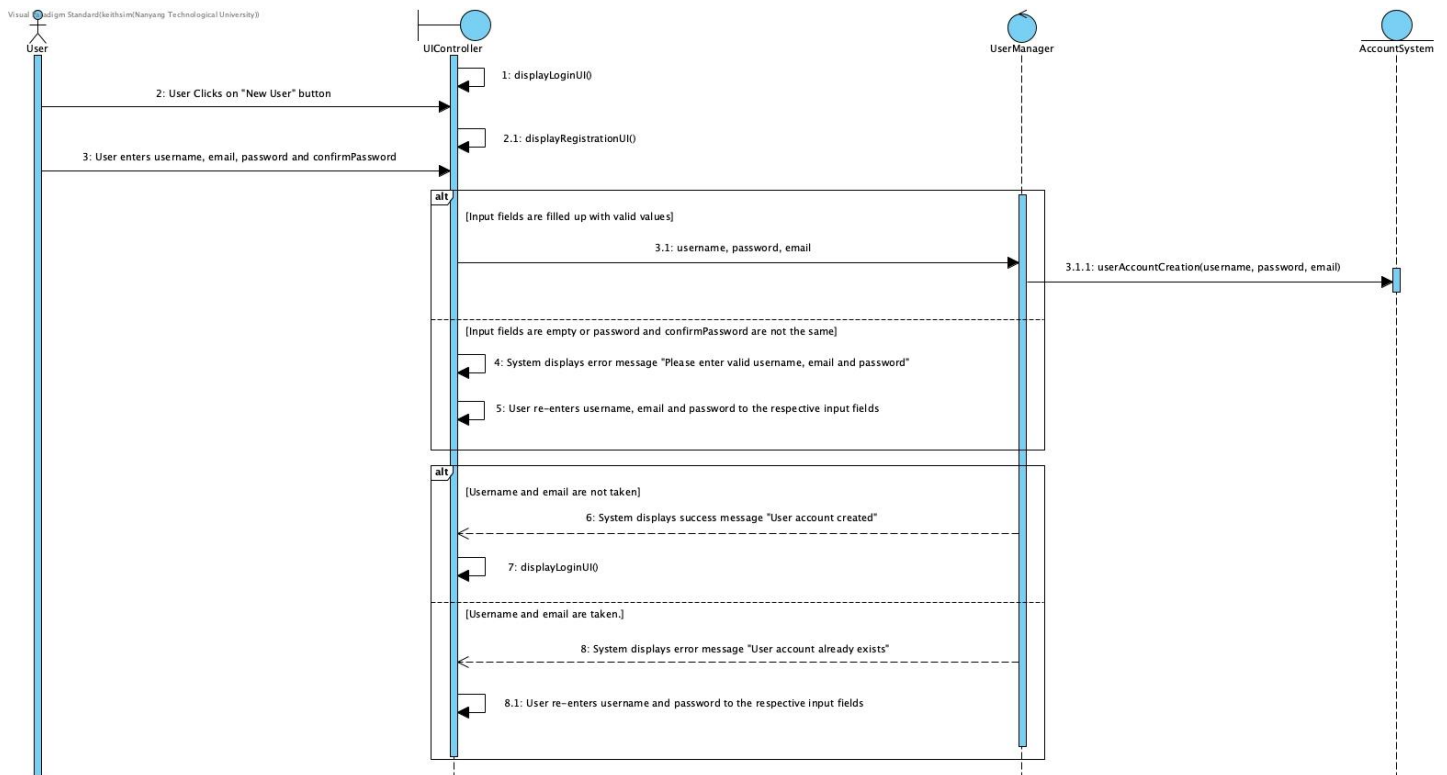
9. Sequence Diagram

Use Case 1 - Login:

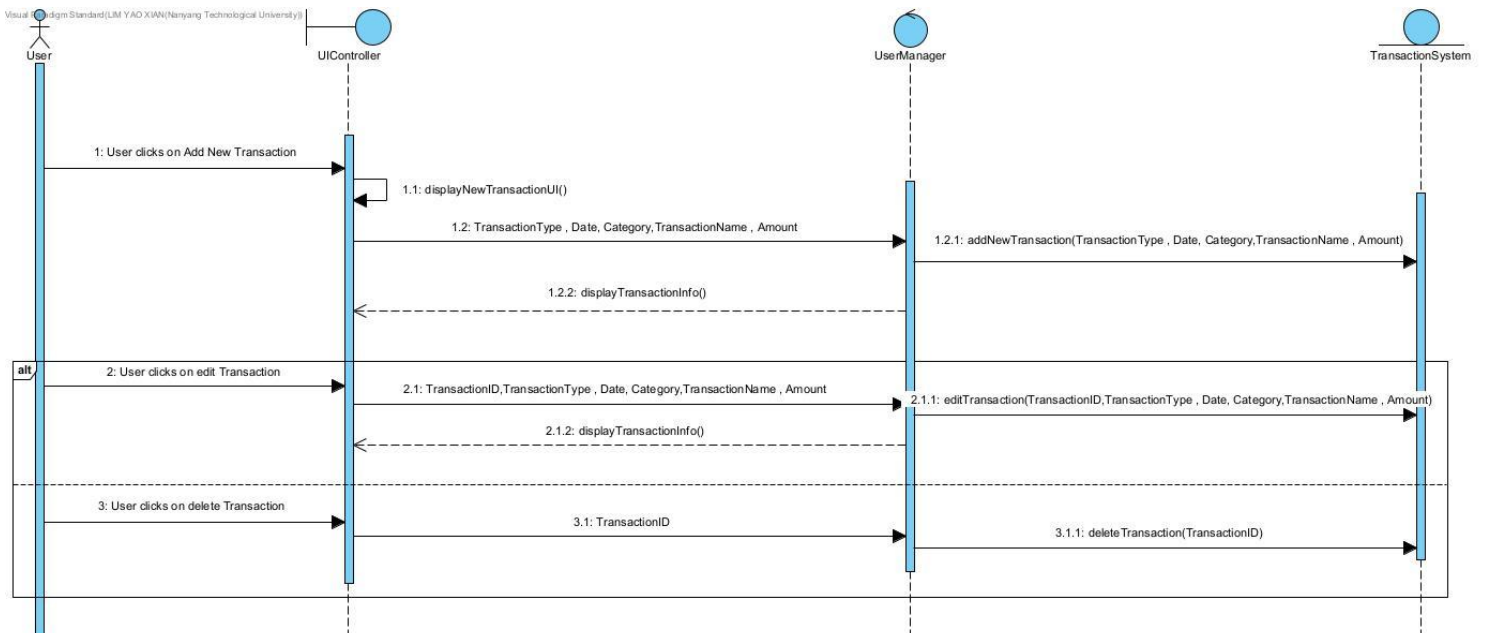
© Paradigm Standard (Nanyang Technological University)



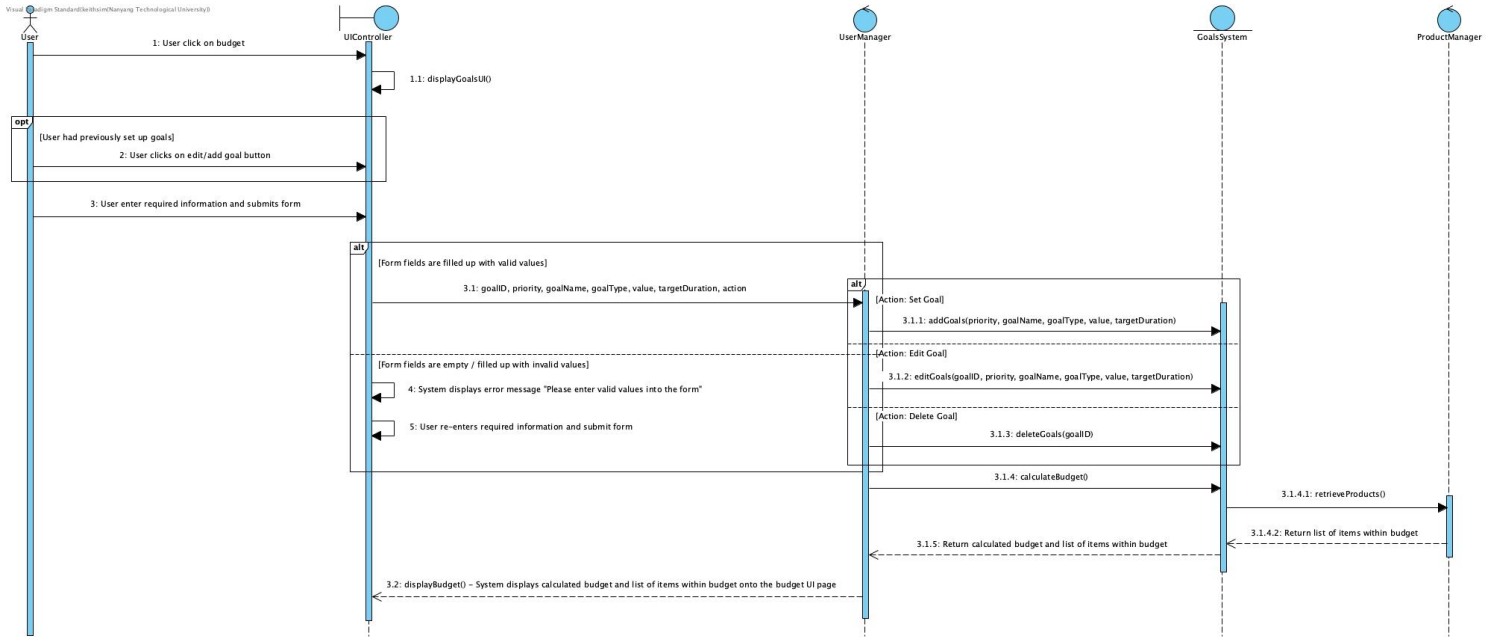
Use Case 3 - Register for an account:



Use Case 4 - Update Transaction:

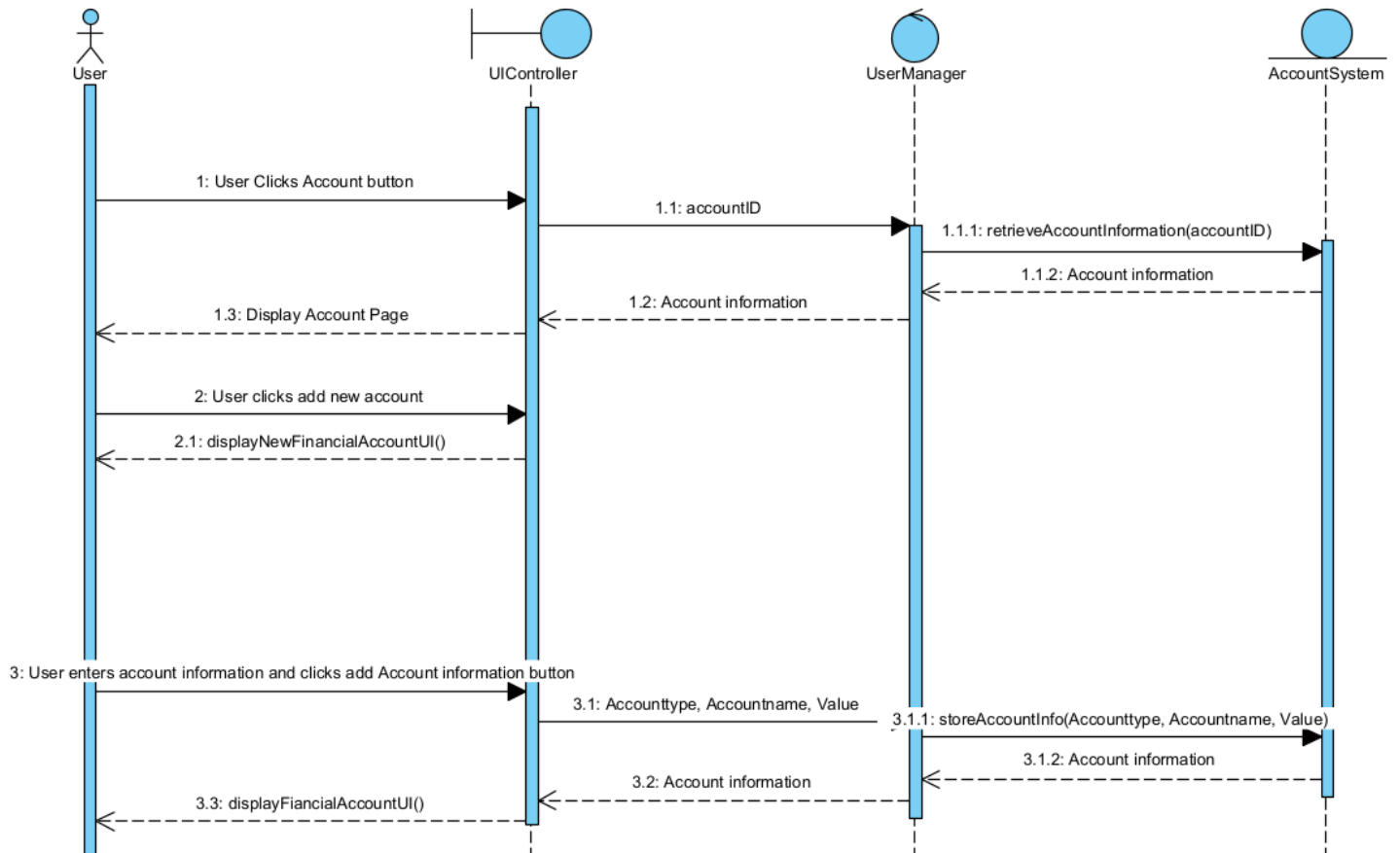


Use Case 5 - Update Goals:



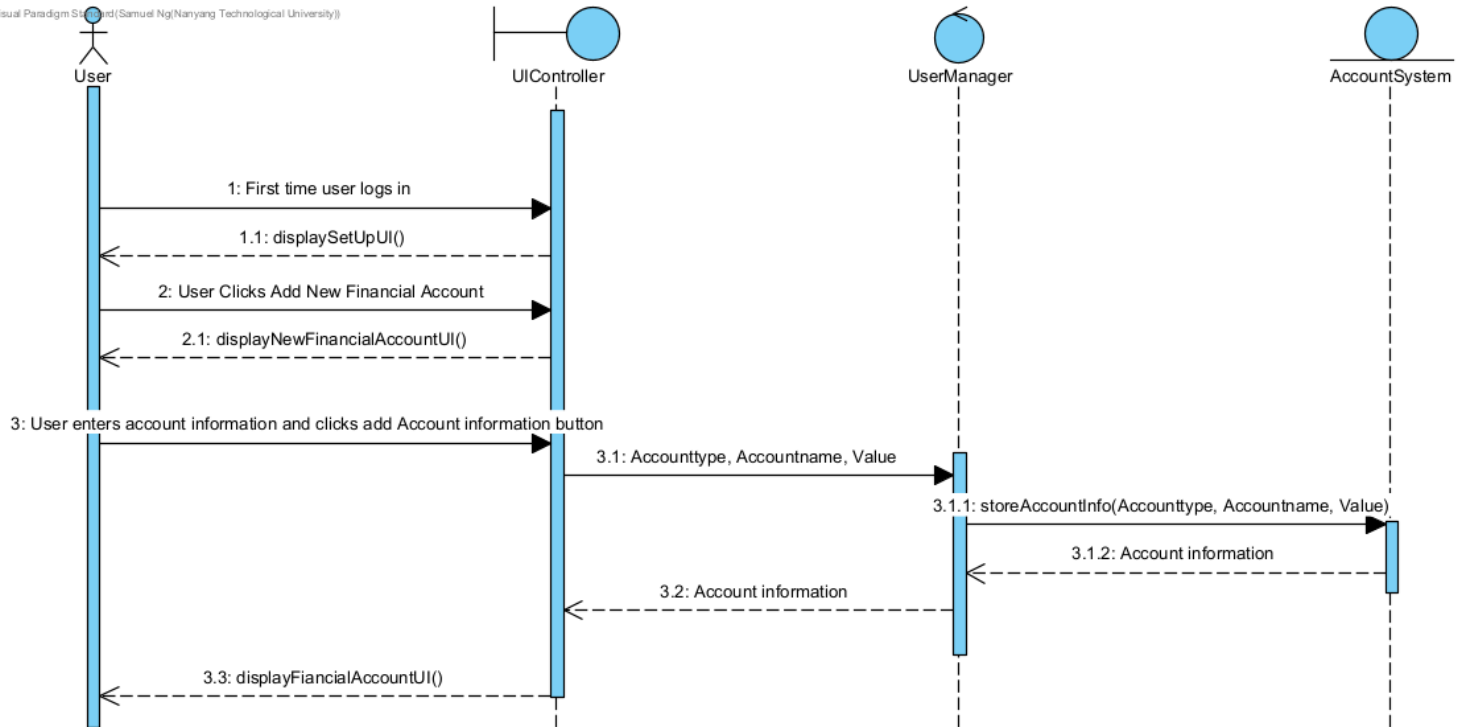
Use Case 8 - Add Account Information:

Small Paradigm Standard (Samuel Ng/Nanyang Technological University)

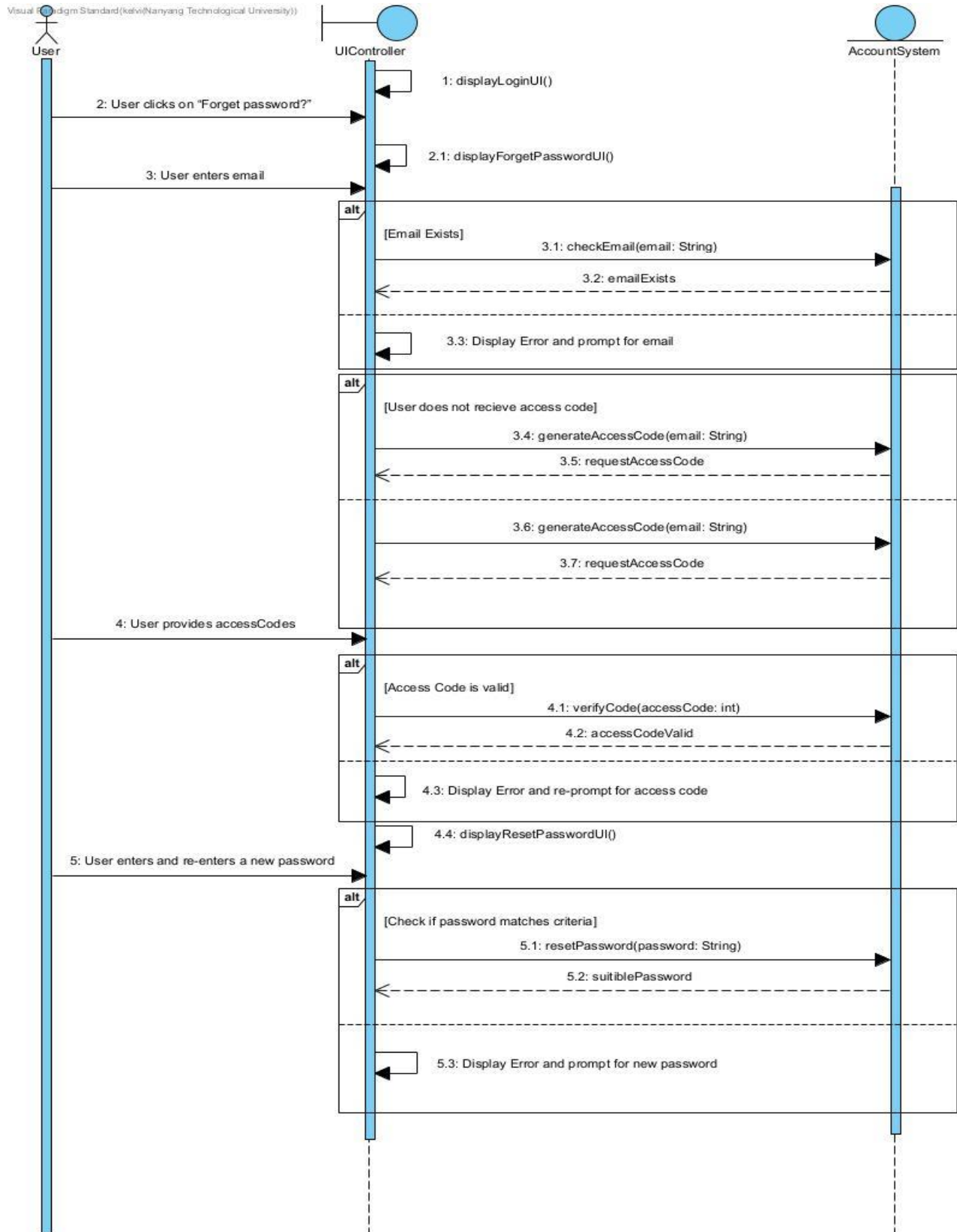


Use Case 9 - Add Initial Information:

Visual Paradigm Student (Samuel Ng/Nanyang Technological University)

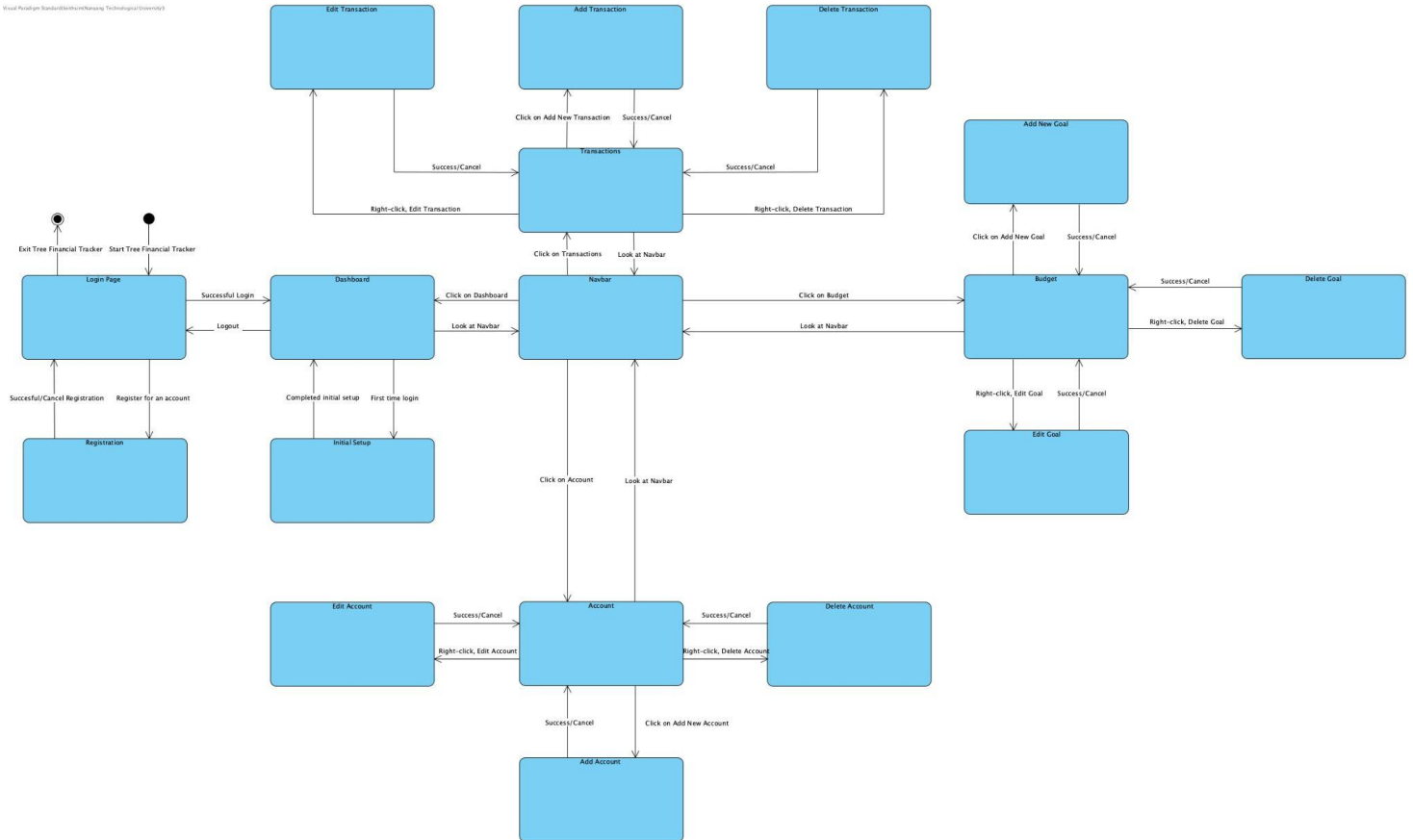


Use Case 12 - Forget Password



10. Dialog Map

Visual Programming Standard/Electronics Technology (Semester 3)



The diagram illustrates the layered architecture of a banking system, organized into three horizontal layers: Boundary, Control, and Entity.

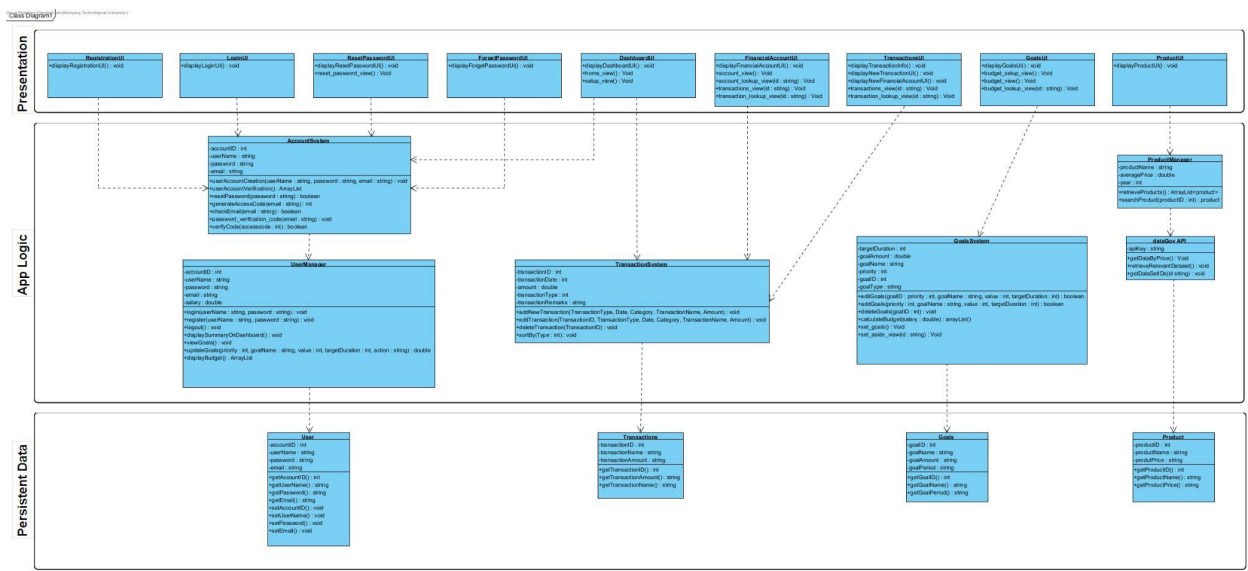
- Boundary Layer:** Contains user interface components: RegistrationUI, LoginUI, ResetPasswordUI, ForgetPasswordUI, DashboardUI, FinancialAccountUI, TransactionsUI, GoalsUI, and ProductUI.
- Control Layer:** Contains business logic components: AccountSystem, UserManager, BankingSystem, TransactionSystem, GoalSystem, and ProductManager.
- Entity Layer:** Contains data entities: User, Transactions, Goals, and Product.

Legend:

- Boundary:** Represented by a circle with a vertical line to its left.
- Control:** Represented by a circle.
- Entity:** Represented by a circle with a horizontal line to its left.

Relationships (indicated by dashed arrows):

- RegistrationUI** (Boundary) connects to **AccountSystem** (Control).
- LoginUI** (Boundary) connects to **AccountSystem** (Control).
- ResetPasswordUI** (Boundary) connects to **AccountSystem** (Control).
- ForgetPasswordUI** (Boundary) connects to **AccountSystem** (Control).
- DashboardUI** (Boundary) connects to **AccountSystem** (Control), **BankingSystem** (Control), **TransactionSystem** (Control), and **GoalSystem** (Control).
- FinancialAccountUI** (Boundary) connects to **AccountSystem** (Control) and **TransactionSystem** (Control).
- TransactionsUI** (Boundary) connects to **TransactionSystem** (Control) and **GoalSystem** (Control).
- GoalsUI** (Boundary) connects to **GoalSystem** (Control).
- ProductUI** (Boundary) connects to **ProductManager** (Control).
- AccountSystem** (Control) connects to **UserManager** (Control).
- UserManager** (Control) connects to **User** (Entity).
- BankingSystem** (Control) connects to **Transactions** (Entity).
- TransactionSystem** (Control) connects to **Transactions** (Entity).
- GoalSystem** (Control) connects to **Goals** (Entity).
- ProductManager** (Control) connects to **Product** (Entity).



12. **Black Box Testing**

12.1. Login

12.1.1. Generic Cases

Test Case ID	Scenario	Expected Result	Actual Result
1	User does not exist	System displays the error message "User does not exist."	System displays the error message "User does not exist."
2	User entered the wrong password	System displays the error message "Wrong Password Entered."	System displays the error message "Wrong Password Entered."
3	User entered the correct password	System redirects the user to the initial set-up page if it is their first time logging in, else system should redirect the user to the dashboard page.	System redirects the user to the initial set-up page if it is their first time logging in, else system should redirect the user to the dashboard page.
4	User presses login without filling in credentials	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."
5	User entered the correct password but the account is not verified	System redirects the user to the account verification page.	System redirects the user to the account verification page.

12.1.2. Specific Cases

Username	Password	Expected Result	Actual Result
user	asdf1234!	User does not exist.	User does not exist.
admin	wrongpass	Wrong Password Entered.	Wrong Password Entered.
admin	Password1!	Login successful	Login successful
(Empty)	(Empty)	Please fill in all fields.	Please fill in all fields.
admin	(Empty)	Please fill in all fields.	Wrong Password Entered.
(Empty)	Password1!	Please fill in all fields.	Please fill in all fields.

12.2. Registration

12.2.1. Generic Cases

Test Case ID	Scenario	Expected Result	Actual Result
1	Username already exists	System displays the error message "Username already exists."	System displays the error message "Username already exists."
2	Email already exists	System displays the error message "Email already exists."	System displays the error message "Email already exists."
3	Password & Confirm Password are not the same	System displays the error message "Passwords are different"	System displays the error message "Passwords are different"
4	All fields are filled in with valid values	System redirects the user to the account verification page.	System redirects the user to the account verification page.
5	Any of the fields are empty	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."

12.2.2. Specific Cases

Username	First Name	Last Name	Email	Password	Confirm Password	Expected Result	Actual Result
admin	admin	admin	admin@gmail.com	Password1!	Password1!	Username already exists.	Username already exists.
user	user	user	admin@gmail.com	Password1!	Password1!	Email already exists.	Email already exists.
user	user	user	user@gmail.com	Password1!	testpwd	Passwords are different	Passwords are different
user	user	user	user@gmail.com	Password1!	Password1!	Account created successfully, redirect to account verification page	Account created successfully, redirect to account verification page
(Empty)	user	user	user@gmail.com	Password1!	Password1!	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."

user	(Empty)	user	user@gmail.com	Password1!	Password1!	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."
user	user	(Empty)	user@gmail.com	Password1!	Password1!	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."
user	user	user	(Empty)	Password1!	Password1!	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."
user	user	user	user@gmail.com	(Empty)	Password1!	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."
user	user	user	user@gmail.com	Password1!	(Empty)	System displays the error message "Please fill in all fields."	System displays the error message "Please fill in all fields."

12.3. Add Transactions

12.3.1. Generic Cases

Test Case ID	Scenario	Expected Result	Actual Result
1	Any of the fields are empty	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."
2	Transaction amount is not an integer or decimal value	System displays the error message "Transaction amount should be a decimal number."	System displays the error message "Transaction amount should be a decimal number."
3	Transaction amount is a negative number	System displays the error message "Transaction amount should be greater than zero."	System displays the error message "Transaction amount should be greater than zero."
4	All fields are filled in with valid values	System displays the success message "Successfully added new transaction."	System displays the success message "Successfully added new transaction."

12.3.2. Specific Cases

Transaction Type	Transaction Account	Date	Category	Transaction Name	Transaction Amount	Remarks	Expected Result	Actual Result
(Empty)	OCBC	13/11/2022	Salary	Monthly Salary	3000	Nov 2022	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."
Income	(Empty)	13/11/2022	Salary	Monthly Salary	3000	Nov 2022	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."
Income	OCBC	(Empty)	Salary	Monthly Salary	3000	Nov 2022	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."
Income	OCBC	13/11/2022	(Empty)	Monthly Salary	3000	Nov 2022	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."

Income	OCBC	13/11/2022	Salary	(Empty)	3000	Nov 2022	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."
Income	OCBC	13/11/2022	Salary	Monthly Salary	(Empty)	Nov 2022	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."
Income	OCBC	13/11/2022	Salary	Monthly Salary	3000	(Empty)	System displays the error message "Please fill in this field."	System displays the error message "Please fill in this field."
Income	OCBC	13/11/2022	Salary	Monthly Salary	text	Nov 2022	System displays the error message "Transaction amount should be a decimal number."	System displays the error message "Transaction amount should be a decimal number."
Income	OCBC	13/11/2022	Salary	Monthly Salary	-3000	Nov 2022	System displays the error message "Transaction amount should be greater than zero."	System displays the error message "Transaction amount should be greater than zero."
Income	OCBC	13/11/2022	Salary	Monthly Salary	3000	Nov 2022	System displays the success message "Successfully added new transaction."	System displays the success message "Successfully added new transaction."

12.4. Add Financial Account

12.4.1. Generic Cases

Test Case ID	Scenario	Expected Result	Actual Result
1	Account Type field is empty	System displays the error message "Please fill in your account type."	System displays the error message "Please fill in your account type."
2	Account Name field is empty	System displays the error message "Please fill in your account name."	System displays the error message "Please fill in your account name."
3	Account Value field is empty	System displays the error message "Please fill in your account value."	System displays the error message "Please fill in your account value."
4	Account value is not an integer or decimal value	System displays the error message "Please fill in your account value in decimal number only."	System displays the error message "Please fill in your account value in decimal number only."
5	Account value is a negative number	System displays the error message "Account value should be greater than zero."	System displays the error message "Account value should be greater than zero."
6	All fields are filled in with valid values	System displays the success message "Successfully added new financial account."	System displays the success message "Successfully added new financial account."

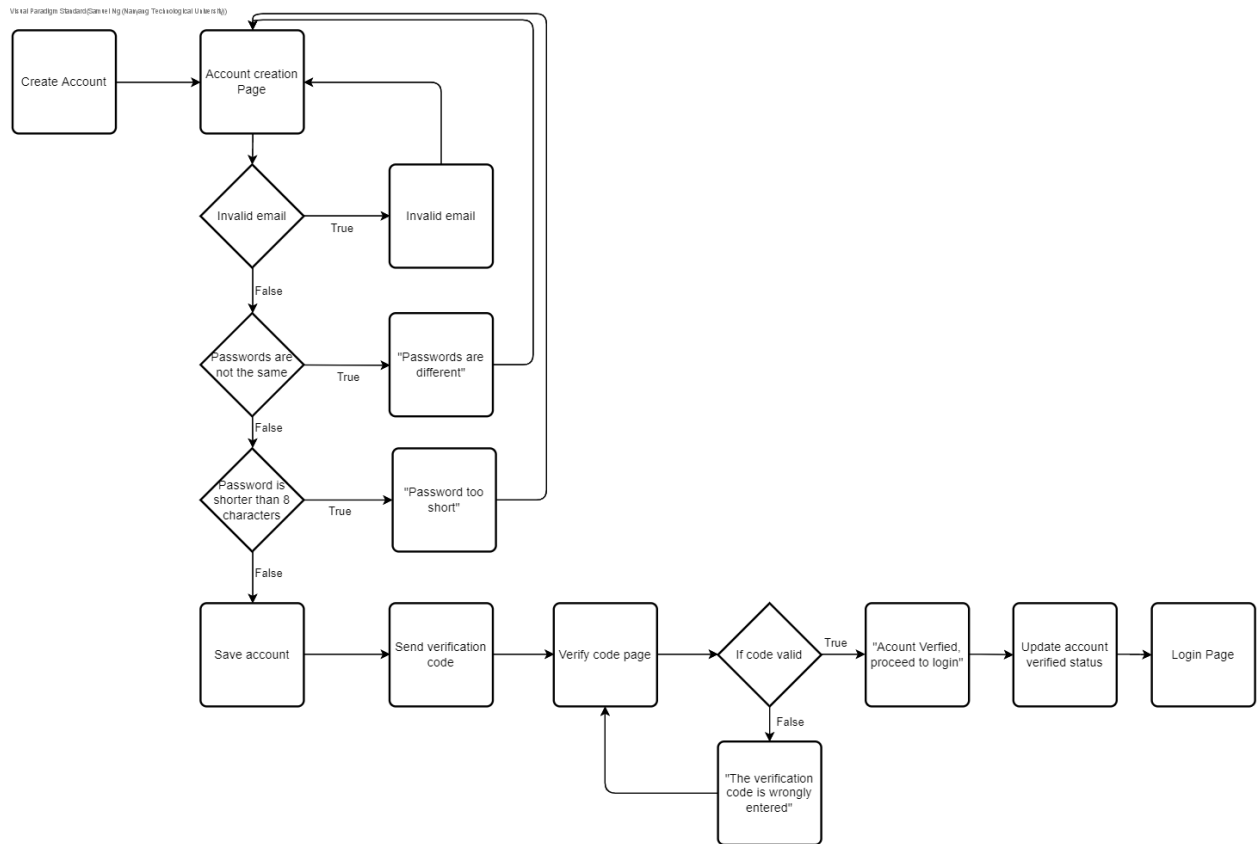
12.4.2. Specific Cases

Account Type	Account Name	Account Value	Expected Result	Actual Result
(Empty)	OCBC	10000	System displays the error message "Please fill in your account type."	System displays the error message "Please fill in your account type."
Assets	(Empty)	10000	System displays the error message "Please fill in your account name."	System displays the error message "Please fill in your account name."
Assets	OCBC	(Empty)	System displays the error message "Please fill in your account value."	System displays the error message "Please fill in your account value."
Assets	OCBC	test	System displays the error message "Please fill in your account value in decimal number only."	System displays the error message "Please fill in your account value in decimal number only."

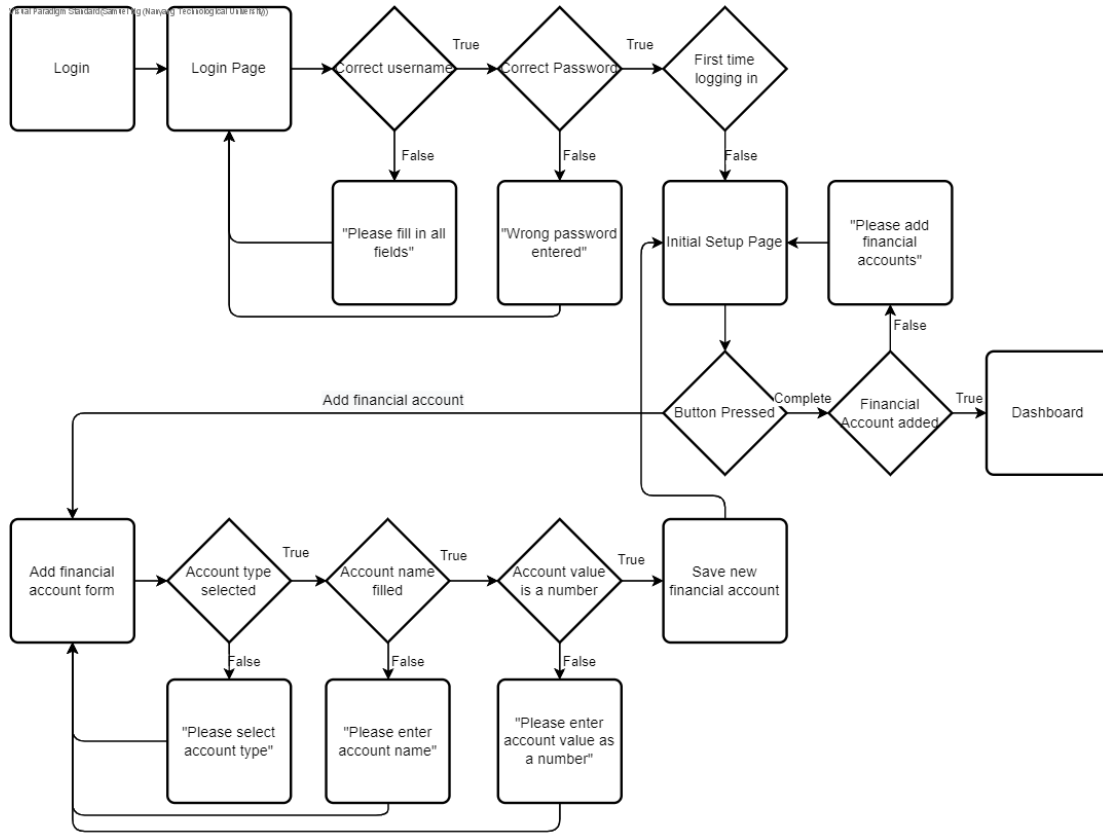
Assets	OCBC	-10000	System displays the error message "Account value should be greater than zero."	System displays the error message "Account value should be greater than zero."
Assets	OCBC	10000	System displays the success message "Successfully added new financial account."	System displays the success message "Successfully added new financial account."

13. White Box Testing

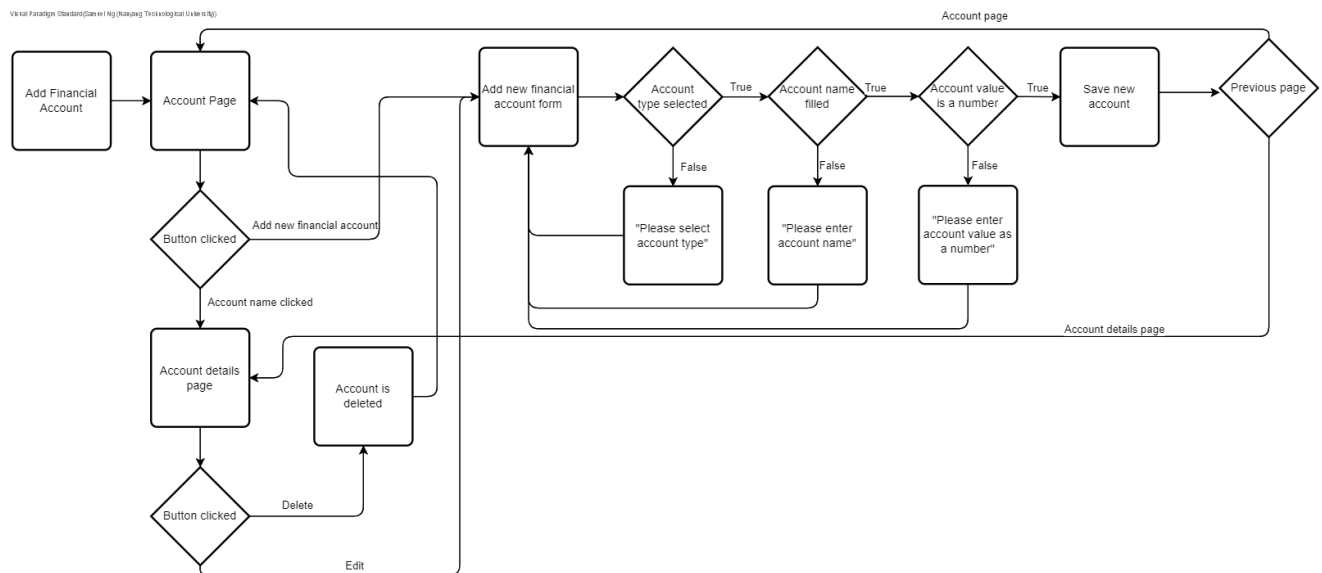
13.1. Account Creation



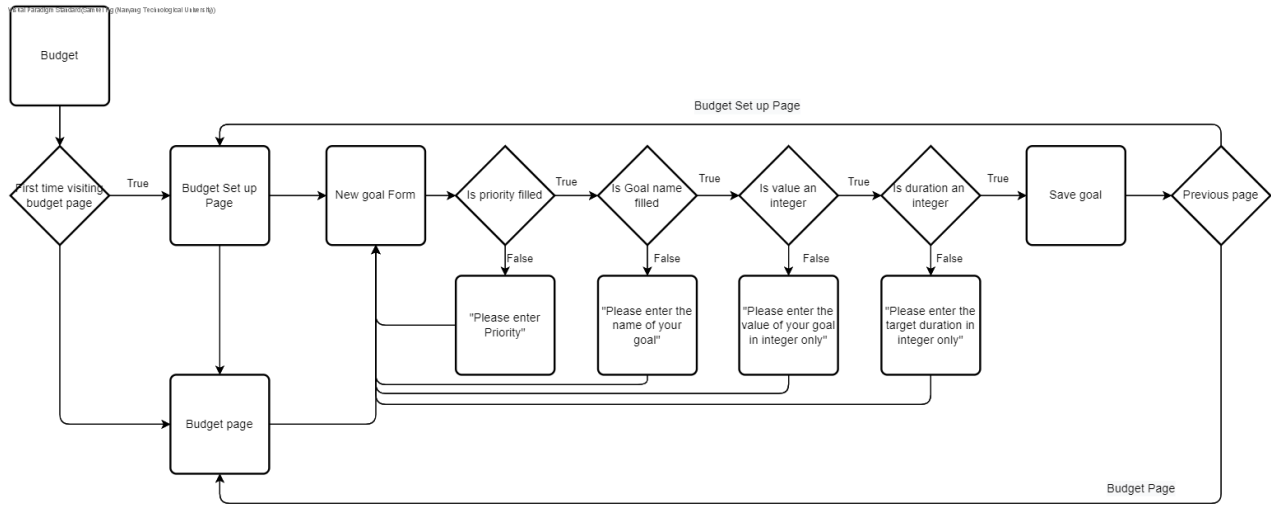
13.2. Login



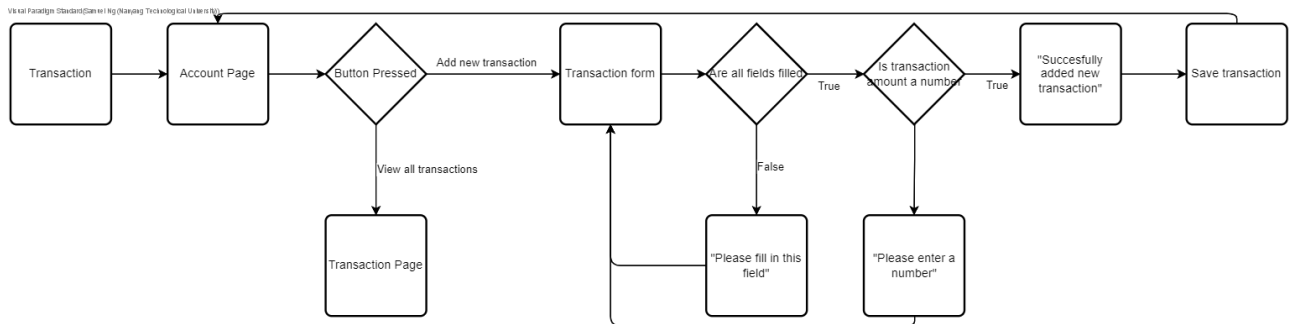
13.3. Financial Account



13.4. Budget



13.5. Transaction



APPENDIX

Frameworks used:

Django - 4.1.2

Template used:

Bootstrap 4 - Start Bootstrap Admin 2

<https://startbootstrap.com/theme/sb-admin-2>

Libraries used:

python-dateutil - 2.8.2

requests - 2.28.1

API used:

Data.gov.sg

Datasets used:

Average Retail Prices Of Selected Consumer Items, Annual

<https://data.gov.sg/dataset/average-retail-prices-of-selected-items-annual>

Tax Rates for Individual Income Tax

<https://data.gov.sg/dataset/tax-rates-for-individual-income-tax>

Public Transport Utilisation - Average Daily Public Transport Ridership

<https://data.gov.sg/dataset/public-transport-utilisation-average-public-transport-ridership>