

Functional Requirements

1. Registration
 - 1.1. The System registration must allow the user to register for a new user account.
 - 1.1.1. Information to include:
 - 1.1.1.1. Name
 - 1.1.1.2. Valid Email Address
 - 1.1.1.3. 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.
2. Login
 - 2.1. The user must be able to login 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 system must redirect the user to the forget password page after 3 failed login attempts.
 - 2.5.1. The system must email the user after 3 failed login attempts.
 - 2.6. The user must be able to change their password if they forgot their current password.
 - 2.6.1. The system must be able to email the user their temporary password.
 - 2.7. The system will redirect the user to their dashboard upon successful login.
3. Dashboard
 - 3.1. The system must allow new users to perform initial set-up of application.
 - 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
 - 3.1.2. The user must be able to add and sync their OCBC account balance as a new financial account.
 - 3.2. The system must allow the user to add more financial accounts after 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 the 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. Date
 - 4.2.1.3. Category
 - 4.2.1.4. Transaction Name
 - 4.2.1.5. Amount
 - 4.2.1.6. Remarks
 - 4.2.2. The system should be capable of auto-completing the average price of common food items and/or products.
- 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 net income 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. Bills
 - 5.2.1.1.2. Meals
 - 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 budget to the user via the “View All” button.

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 Name
 - 6.1.1.3. Remarks
 - 6.1.1.4. Category
 - 6.1.1.5. 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.
- 7.2. The system must be able to retrieve the list of prices of housing from Data.gov.sg API.
- 7.3. The system must be able to retrieve the list of prices of taxes from Data.gov.sg API.
- 7.4. The system must be able to retrieve a user’s OCBC account balance via OCBC’s 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.

Non-Functional Requirements

1. Usability Requirements:

- 1.1. The software must allow the user to change the **display language**.
 - 1.1.1. The user must be able to change within 4 different languages (e.g. English, Chinese, Malay, and Tamil)
- 1.2. The system must be simple to use even for someone without much technical experience.
 - 1.2.1. To provide help messages to guide the user along.
- 1.3. Font and text must be legible to all users.
- 1.4. The system must be consistent.
 - 1.4.1. Same processes must use the same actions (e.g. To show more details, right click on the item.)
 - 1.4.2. Use the same visual cues for the same actions (e.g. The delete button option is shown whenever the user right clicks 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 not more than 5 seconds.

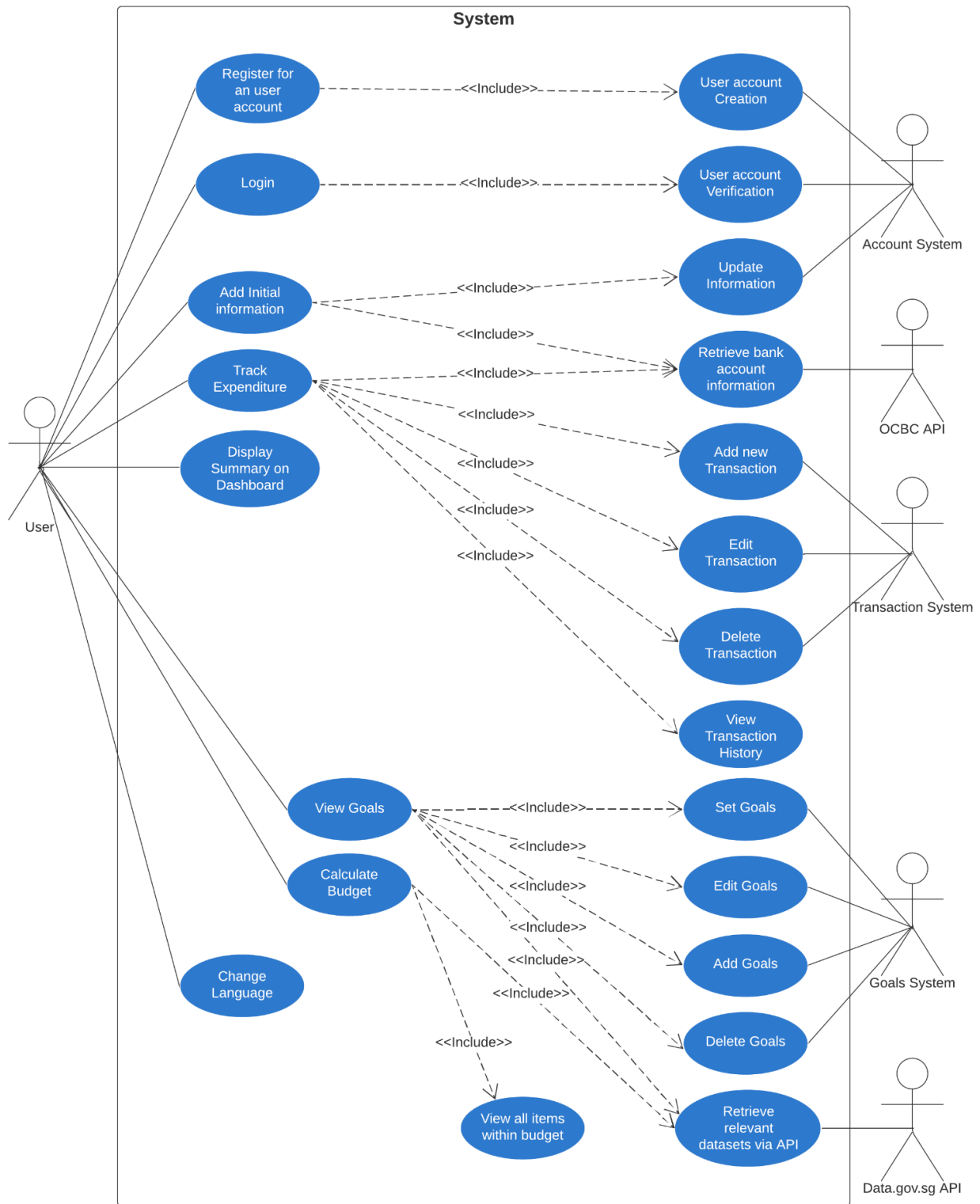
4. Supportability Requirement:

- 4.1. The code used must be compatible with the 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.

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.

Use Case Diagram



Use Case Descriptions

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

Actor:	User, System
Description:	User logs in to the system.
Preconditions:	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:	1. User 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 logs the user in.
Alternative Flows:	AF-S1: User clicks on register instead of entering details. 1. Redirect the user to the registration use case. AF-S3: System detects invalid email/password. 1. Prompt users to re-enter their credentials. 2. Return to step 1.
Exceptions:	EX-AF_S3: The device is not connected to the internet. 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:	
Date Created:	6/9/2022	Date Last Updated:	6/9/2022

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 recent transactions. 3. Users are able to view their assets and liabilities. 4. Users are able to view their progress towards their goals.
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 user's assets and liabilities. 2. The user's progression towards their goals. 3. A list of the user's recent transactions for the current month.
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:	
Date Created:	6/9/2022	Date Last Updated:	6/9/2022

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 New user button 2. System displays form for an User Account Creation - Username, Email, Password, Confirm Password 3. User enters required information and submits form 4. System stores information in database using the Account Creation use case 5. System displays result of an User Account Creation Success 6. 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"> 3. Prompt users to re-enter their credentials. 4. Return to step 1. <p>AF-S4: System detects invalid email is already in the database</p> <ol style="list-style-type: none"> 5. Display message "User account already exists" 6. Return to Login Page
Exceptions:	-
Includes:	User Account Creation
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Track expenditure		
Created By:	Samuel	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	6/9/2022

Actor:	User, System
Description:	User can view and add transactions
Preconditions:	1. The user must be logged in
Postconditions:	-
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. User Clicks Track expenditure 2. System displays financial account information - Account value, Recent Transactions, Monthly budget goal, current month saving/expenditure 3. User Clicks Add New Transaction 4. System displays form for Adding New Transaction - Transaction Type, Date, Category, Transaction Name, Amount 5. System returns to step 2
Alternative Flows:	<p>AF-S4: User right Clicks a transaction</p> <ol style="list-style-type: none"> 5. Display menu to edit or delete transaction <p>AF-S6: User Clicks delete transaction</p> <ol style="list-style-type: none"> 7. Transaction is deleted <p>AF-S6: User Clicks edit transaction</p> <ol style="list-style-type: none"> 7. Display menu for editing New Transaction - Transaction Type, Date, Category, Transaction Name, Amount
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:	Set Goals		
Created By:	Yao Xian	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	6/9/2022

Actor:	User
Description:	For the user to enter the list of items which they want to buy and prioritize them 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. Goals are set.
Alternative Flows:	AF-S2: User clicks on Add New Goal instead of entering details. 1. Redirect the user to the set goal use case. AF-S3: System detects invalid information. 1. Prompt users to re-enter their credentials. 2. Return to step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

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

Actor:	User
Description:	For the users to edit and delete 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. Users click on edit goals. 3. Users edit their information.
Alternative Flows:	<p>AF-S2: User right click and delete the goals.</p> <ol style="list-style-type: none"> 1. Remove the selected goal. <p>AF-S3: User clicks on submit with an empty form.</p> <ol style="list-style-type: none"> 1. Redirect the user to the set goal 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:	1. The user must be logged in
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 initial Information		
Created By:	Samuel	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	6/9/2022

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 User Account Information: Account, Value 3. User Clicks on OCBC 4. Display Form with field: username, password 5. User Clicks Sync 6. Account Information is updated
Alternative Flows:	AF-S1: On First Login <ol style="list-style-type: none"> 2. Display empty Account information AF-S3: User enters account details <ol style="list-style-type: none"> 4. User Clicks Add new Account 5. Account Information is updated
Exceptions:	-
Includes:	Update Information, Retrieve account information
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	Change Language		
Created By:	Yao Xian	Last Updated By:	
Date Created:	6/9/2022	Date Last Updated:	6/9/2022

Actor:	User, System
Description:	For the user to select their preferable language.
Preconditions:	1. The user must be logged in
Postconditions:	1. The system will change to the selected language.
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks on the change language button. 2. The system prompts the user on which language to choose. 3. The system converts to the selected language.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

UI Mock Up

Tree Financial Tracker

Login

Login


New User


[Forget Password?](#)



Create New Account

Welcome, Keith

 Dashboard

 Dashboard

Account

 Account

- Budget
- Transactions

- Budget
- Transactions

Transactions

Transactions

Set-up Tree Financial Tracker ?

In order to get started with Tree Financial Tracker, you must insert your relevant data.

Account	Value
Bank	
OCBC	\$10,000.00
Cash	\$100.00
Total	\$10,100.00
Investments	
Stocks	\$1000.00
Liabilities	
Student Loan	-\$3000.00
Net Worth	\$8100.00

Account Type ▼

Name of Account

Value

Add New Account



Finish

Sync with OCBC

Welcome, Keith

 Dashboard

 Dashboard



Account



Account

- Budget
- Transactions

- Budget
- Transactions

Transactions

Transactions

Set-up Tree Financial Tracker ?

In order to get started with Tree Financial Tracker, you must insert your relevant data.

Account		Value
Bank		
	OCBC	\$10,000.00
	Cash	\$100.00
	Total	\$10,100.00
Investments		
	Stocks	\$1000.00
Liabilities		
	Student Loan	-\$3000.00
Net Worth		\$8100.00

Account Type ▼

Add New Account

Name of Account

Value

Finish

Sync with OCBC

Welcome, Keith

 Dashboard

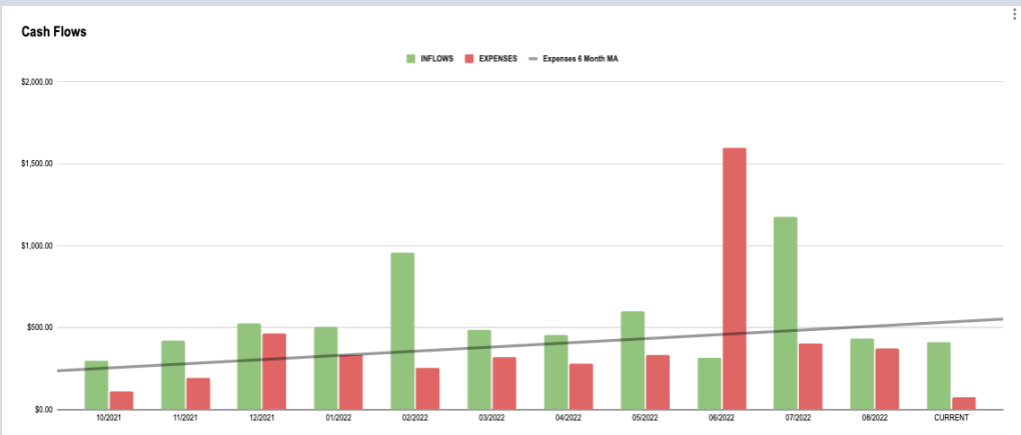
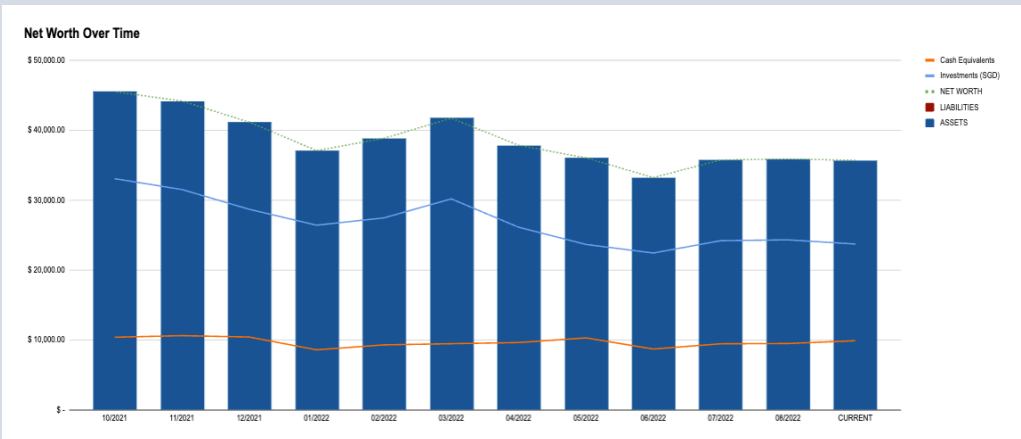
Account



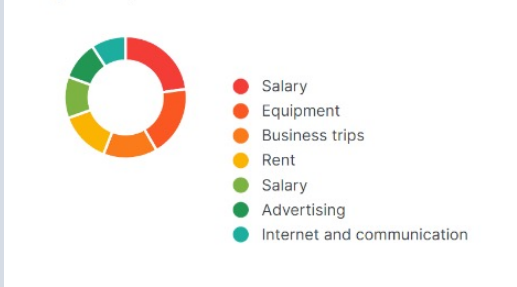
Budget

Transactions

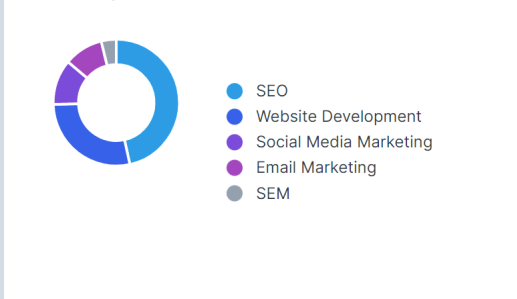
Dashboard

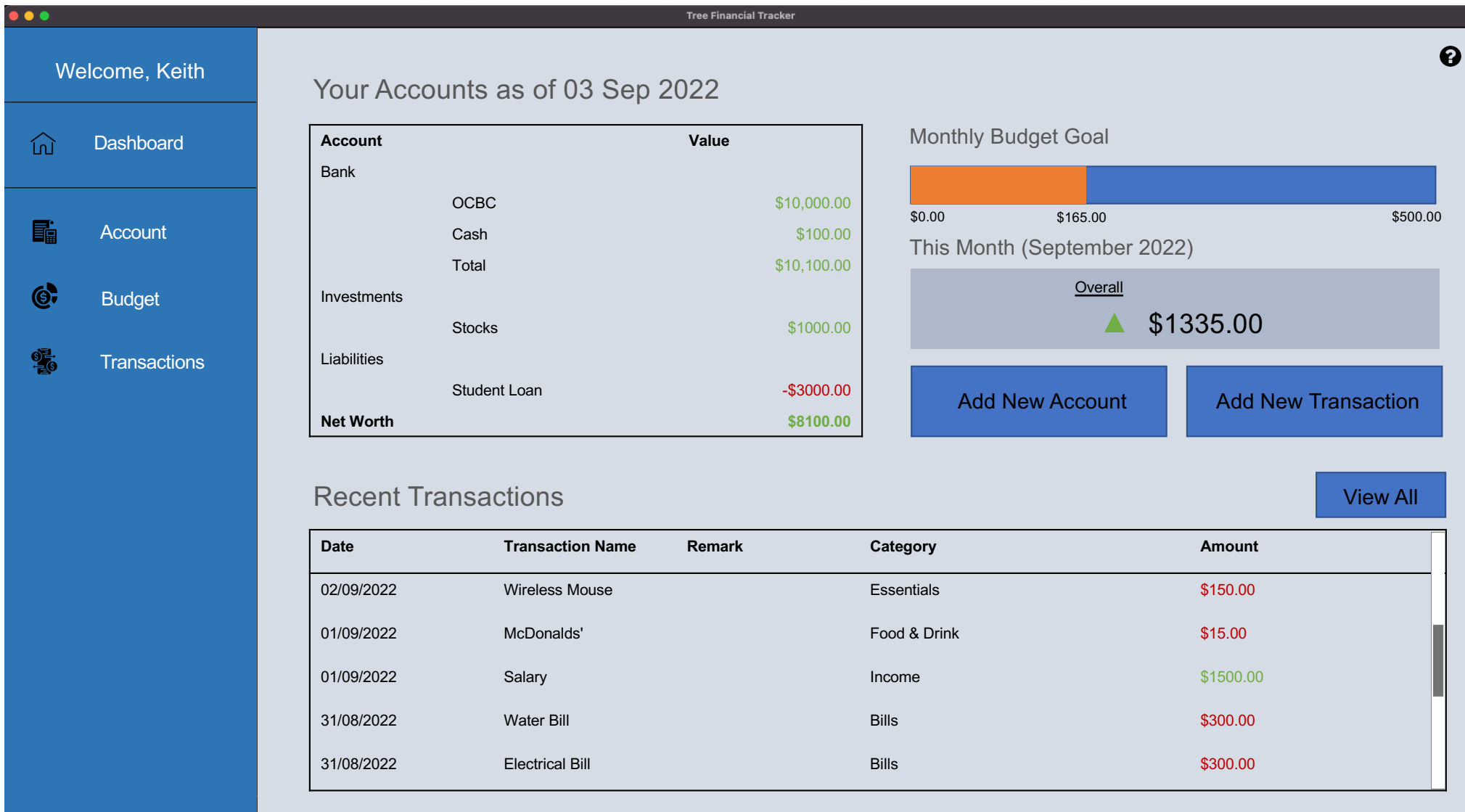


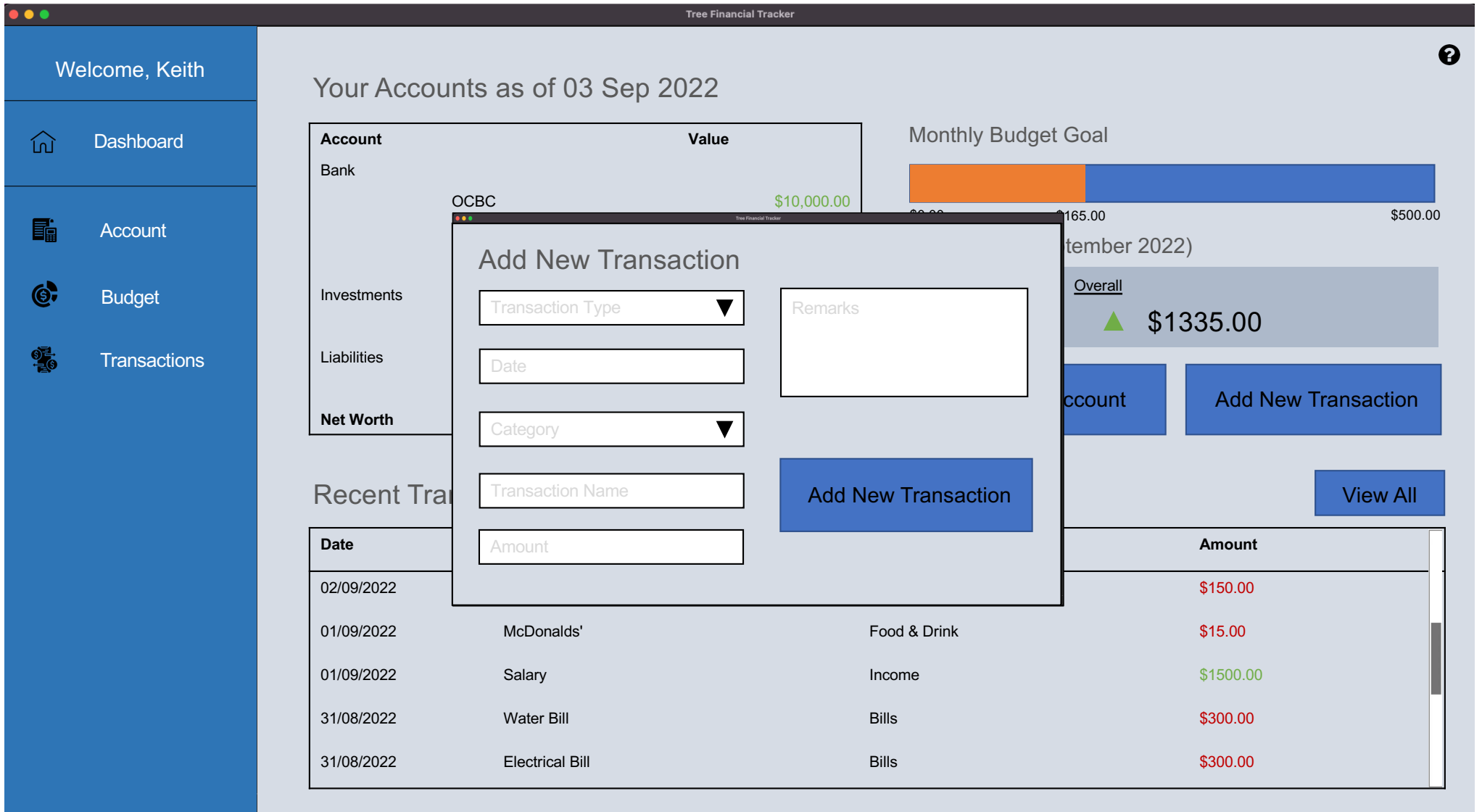
- 272 200 \$
Expenses July 2022

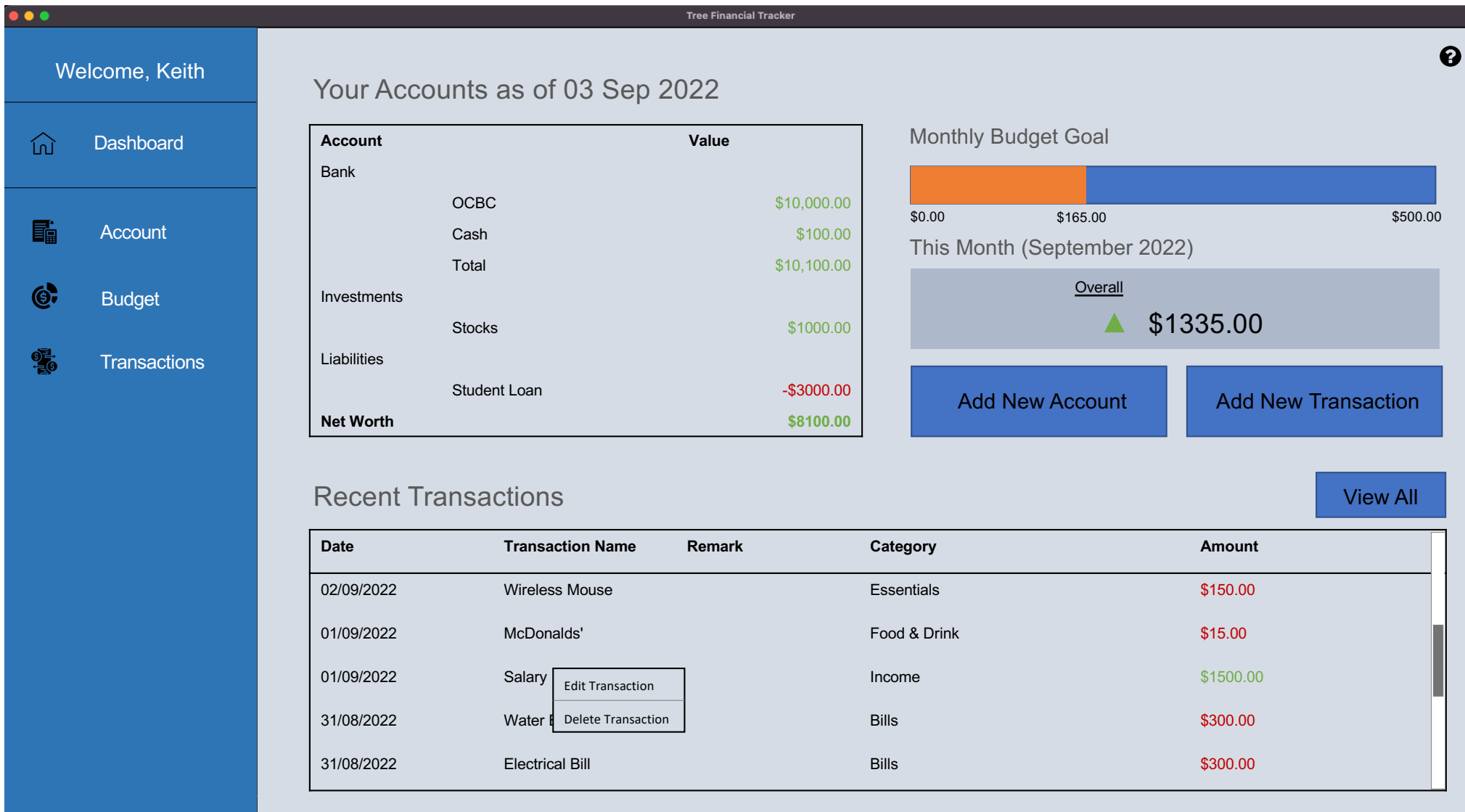


389 000 \$ Month









Welcome, Keith

Dashboard

Account

Budget

Transactions

Your Accounts as of 03 Sep 2022

Account	Value
Bank	OCBC\$10,000.00
Investments	
Liabilities	
Net Worth	

Monthly Budget Goal

165.00\$500.00

September 2022)

Overall

▲\$1335.00

AccountAdd New Transaction

View All

Recent Transactions

Date	Description	Category	Amount
02/09/2022			\$150.00
01/09/2022	McDonalds'	Food & Drink	\$15.00
01/09/2022	Salary	Income	\$1500.00
31/08/2022	Water Bill	Bills	\$300.00
31/08/2022	Electrical Bill	Bills	\$300.00

Edit Transaction

Income

01/09/2022

Income

Salary

\$1500.00

Remarks

Edit Transaction

Tree Financial Tracker

Welcome, Keith

Dashboard

Account

Budget

Transactions

Budgeting & Financial Goals

Set goal(s) and Tree Financial Tracker will do the rest for you.

Priority	Goal Name	Value	Target Duration
1	Desktop PC	\$2000.00	6 Months
2	BTO Flat Deposit	\$30,000.00	4 Years
Add More...			

Priority

Goal Name

Value

Target Duration

Add New Goal

Finish

Tree Financial Tracker

Welcome, Keith

Dashboard

Account

Budget

Transactions

Based on your set goals and spending habits...

Edit / Add Goal

Tree Financial Tracker recommends that

You should set aside

Category	Amount
Bills	\$600.00
Leisure	\$100.00
Food & Drink	\$150.00

To attain your goals...

Goal	Remark
Desktop PC in 6 months	<div>You can only spend \$333.33 per month or \$10.75 a day.</div> <div>Current Progress (month): \$0.00 / \$ 333.33</div> <div>Total Progress: \$0.00 / \$2000.00</div>
BTO Flat Deposit in 4 years	<div>You can only spend \$625.00 per month or \$20.16 a day.</div> <div>Current Progress (month): \$0.00 / \$ 625.00</div> <div>Total Progress: \$0.00 / \$30,000.00</div>

List of Items within Budget

View All

Item Name	Amount
Ordinary White Bread (Per 400 Gram)	1.75
Broccoli (Per Kilogram)	6.25
Instant Coffee (Per 200 Gram)	10.17
Chicken Nasi Briyani (Per Plate)	5.79
Ice Kachang (Per Bowl)	2.14

Tree Financial Tracker

Welcome, Keith

Dashboard

Account

Budget

Transactions

Budgeting & Financial Goals

Set goal(s) and Tree Financial Tracker will do the rest for you.

Priority	Goal Name	Value	Target Duration
1	Desktop PC	\$2000.00	6 Months
2	BTO Flat D	\$30,000.00	4 Years
Add More...			

Priority

Goal Name

Value

Target Duration

Add New Goal

Finish

Welcome, Keith

 Dashboard

 Dashboard



Account



Account

- Budget
- Transactions

- Budget
- Transactions

Transactions

Transactions

Budgeting & Financial Goals

Set goal(s) and Tree Financial Tracker will do the rest for you.

Priority	Goal Name	Value	Target Duration
1	Desktop PC		6 Months
2	BTO Flat		4 Years
Add More...			

1

Desktop PC

\$2000.00

6 Months

Edit Goal

Priority	Value
Goal Name	Target Duration

Add New Goal

Finish

Welcome, Keith

Dashboard

Account

Budget

Transactions

Transaction History

Search

Sort by

Date	Transaction Name	Remark	Category	Amount
02/09/2022	Wireless Mouse		Work	\$150.00
01/09/2022	McDonalds'		Food & Drink	\$15.00
01/09/2022	Salary		Income	\$1500.00
31/08/2022	Water Bill		Bills	\$300.00
31/08/2022	Electrical Bill		Bills	\$300.00
29/08/2022	Movie		Leisure	\$15.00
28/08/2022	PayNow	From Jon – "Dinner"	Food & Drink	\$15.00
27/08/2022	Koufu		Food & Drink	\$5.00
26/08/2022	Kopitiam		Food & Drink	\$4.50
25/08/2022	KFC		Food & Drink	\$9.20
22/08/2022	Book		Leisure	\$20.00

Tree Financial Tracker

Welcome, Keith

Dashboard

Account

Budget

Transactions

List of Items within Budget

Search

Item Name	Amount
Ordinary White Bread (Per 400 Gram)	1.75
Broccoli (Per Kilogram)	3.20
Instant Coffee (Per 200 Gram)	5.10
Chicken Nasi Briyani (Per Plate)	4.50
Ice Kachang (Per Bowl)	2.60
Lean Pork, Chilled (Per Kilogram)	2.30
Streaky Pork, Chilled (Per Kilogram)	11.20
Pork Rib Bones, Chilled (Per Kilogram)	2.40
Beef, Chilled (Per Kilogram)	9.57
Mutton, Chilled (Per Kilogram)	10.13
Whole Chicken, Chilled (Per Kilogram)	6.34