

Hon Hao Chen - Project Portfolio

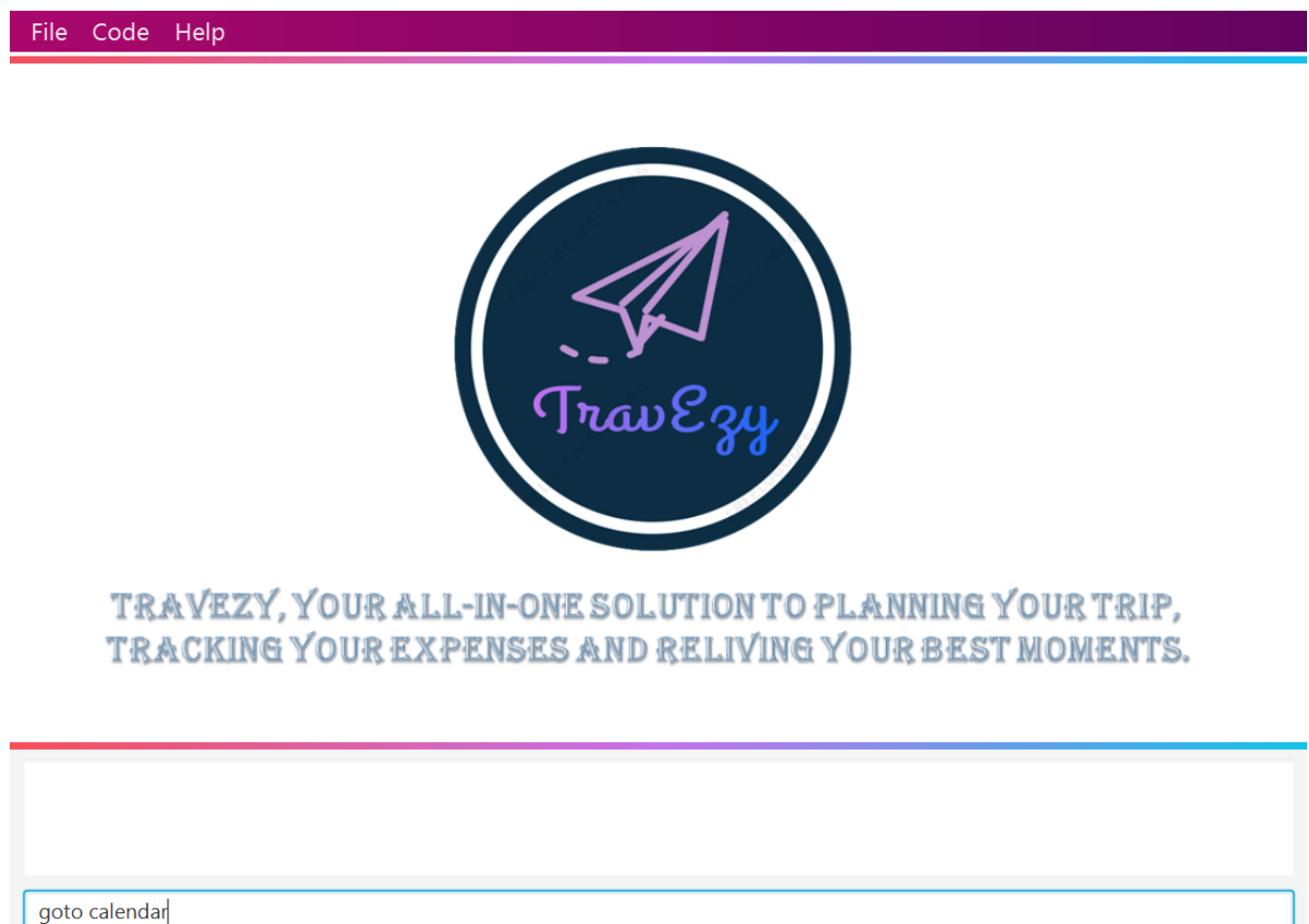
PROJECT: TravEzy

About the project

My team of 4 software engineering students and I were tasked with enhancing a basic [Command Line Interface \(CLI\)](#) desktop *Addressbook* application for our Software Engineering project. Our constraint is that we have to target users who are able to type fast and prefer typing over other means of input.

With the stated constraint in mind, we choose to morph the given *Addressbook* application into an travelling companion application which synthesis various kinds of features including financial tracker, diary, itinerary planner, contacts and achievement. This enhanced application allows users who love to travel to plan their trip beforehand, record down their expenses and write down their precious memories during travel. We ensured that the users have to type commands and instructions for all of the added features.

This is what our project looks like:



My Role

My role was to design and write the codes for the *Financial Tracker* feature. The following sections illustrate my contributions in detail, as well as the relevant documentation I have added to the user and developer guides in relation to these contributions.

Summary of contributions

This section shows a summary of my contributed feature and other relevant contributions to the team project.

Major contributions

Financial Tracker:

- What is it?
 - The *Financial Tracker* is an expense recorder whereby users can keep track of the records of their spending during their trips to different countries.
- Justification
 - Our target users which we have chosen are NUS students who love to travel. During our users analysis, we realised that students nowadays are very likely to overspend their budget during travelling. Therefore, this *Financial Tracker* feature allows our target users to keep a record of their spending in order to remind them of their budget.
- More to it
 - Users have the expense list filtered countries by countries and can switch between each expense list seamlessly.
 - Users are able to sort their expense list according to their needs and undo any accident actions that has been done.
 - Users can generate an overview of their expenses in an easy-to-look pie chart form!

Code contributions: Click [here](#)

Minor contributions

Project management:

- Contributed to the idea of using **Page Manager** to handle pages switching among the six different features.
- Started implementing user interface(Ui) components and link them to FXML. This provides skeletal codes for the rest of my teammates to adapt with.
Code for reference: [36](#)
- Modified common **Command** abstract class to be generic so that the other **Command** classes in other features can take in their own **Model** classes.
Code for reference: [70](#)
- Helped on modifying *Main* feature and QnA section in the UserGuide.

Code for reference: [206](#)

Enhancements to existing features:

- Updated the *Main* page's Ui.
Code for reference: [103](#)

Community:

- PRs reviewed (with non-trivial review comments): [34](#)
- Reported bugs and suggestions for other teams in the class: [14](#)

Contributions to the User Guide

Given below are sections which I contributed to the User Guide. They showcase my ability to write documentation targeting end-users. I shall focus on 3 relevant commands here which are the `switch`, `sort` and `summary` commands.

Switching between countries: `switch`

Currently in somewhere else? Switch to that expense list instead!

Format:

`switch COUNTRY`

TIP

You can always use the *Countries dropdown box* directly instead!

WARNING

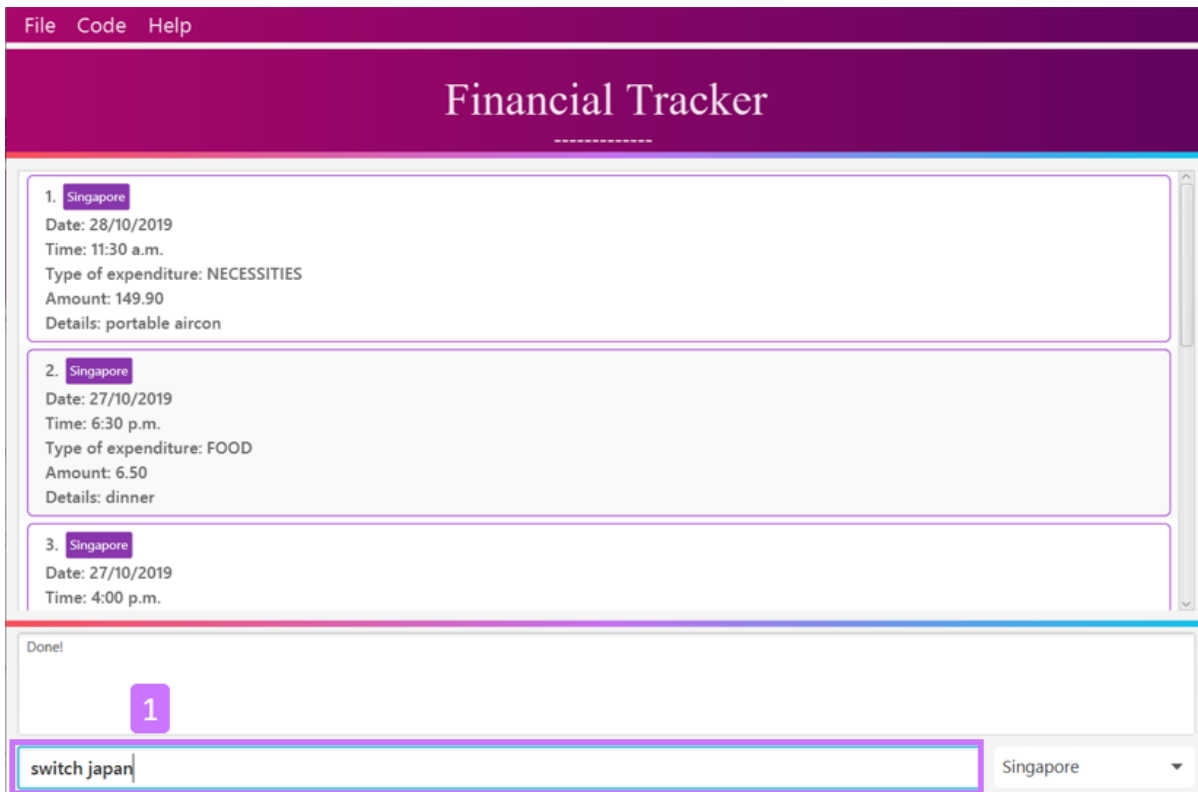
You can only type in countries which are only listed from the *Countries dropdown box*

Example:

```
switch Japan
```

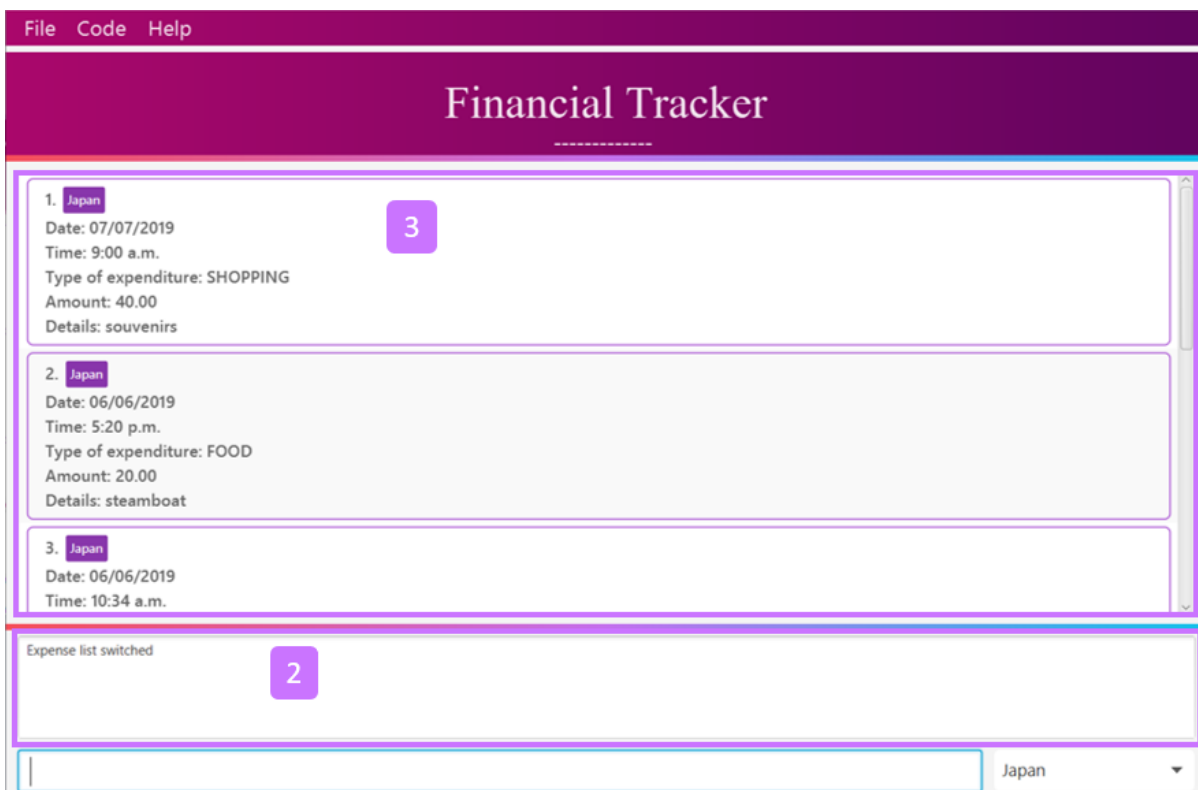
Step by step:

Step 1. Type `switch Japan` in the *Command box* and press *Enter*.



Step 2. The *Result box* will display the message "Expense list switched".

Step 3. Now your expense list inside the *Expense list panel* has been switched to that which is in Japan!



Sort out your expense list: **sort**

The default sorting way of the expense list is not your thing? Just sort your expense list according

your needs!

TIP | All of the sorting is done in reversed order :)

Format:

sort CRITERIA

Where CRITERIA can be amount, date, time, type and default

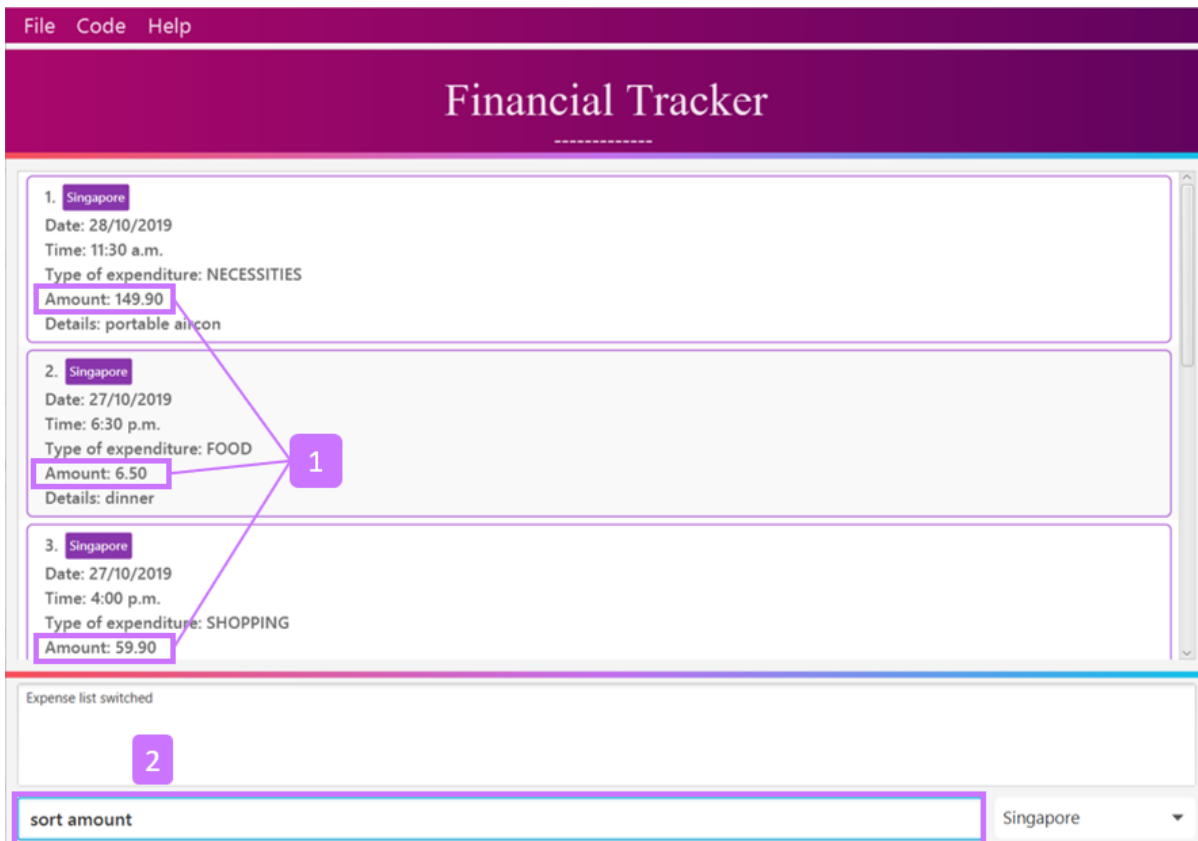
Example: to sort by amount

```
sort amount
```

Step by step:

Step 1. Notice the amount field in each expenses are currently not in order.

Step 2. Now, type `sort amount` in the *Command box* and press *Enter*.



Step 3. The *Result box* will display the message "Expense List sorted!".

Step 4. Now all your expenses are sorted in descending order of your amount!

The screenshot shows the 'Financial Tracker' application interface. At the top is a purple header bar with the title 'Financial Tracker' and a menu with 'File', 'Code', and 'Help'. Below the header is a scrollable list of three expense entries. Each entry contains the following information: a location (Singapore), a date, a time, a type of expenditure, an amount, and details. The first entry is dated 28/10/2019, the second and third are dated 27/10/2019. Annotations with purple boxes and numbers are present: a box around the first entry's amount (149.90) is connected by a line to a box around the second entry's amount (59.90), which is labeled with a purple box containing the number '4'; a box around the third entry's amount (6.50) is connected by a line to a box containing the number '3'. Below the list, there is a section labeled 'Expense List sorted!' with a large empty box below it. At the bottom, there is a search bar and a dropdown menu currently set to 'Singapore'.

Location	Date	Time	Type of expenditure	Amount	Details
Singapore	28/10/2019	11:30 a.m.	NECESSITIES	149.90	portable aircon
Singapore	27/10/2019	4:00 p.m.	SHOPPING	59.90	New wireless keyboard
Singapore	27/10/2019	6:30 p.m.	FOOD	6.50	

Generate an overview of your spending: **summary**

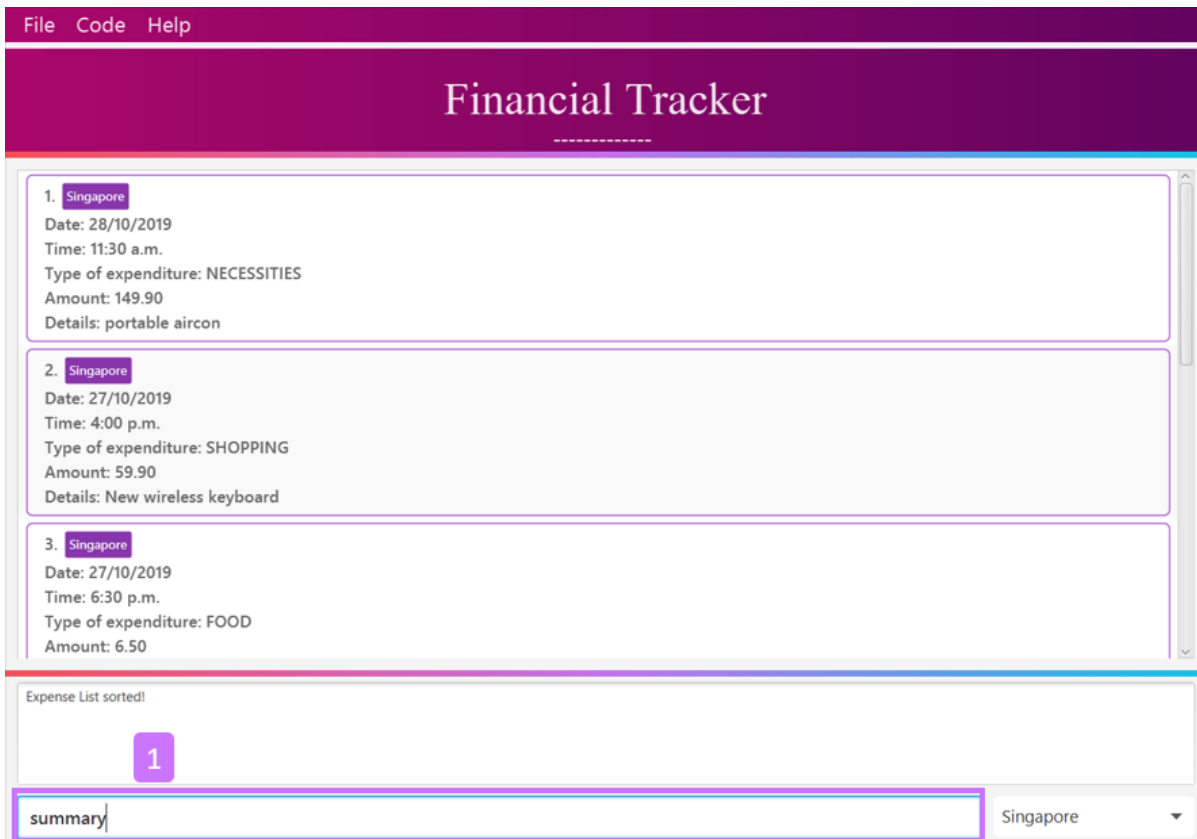
Do you ever have difficulty summarising your spending? Afraid not! You can view your expenses statistics easily!

Format:

summary

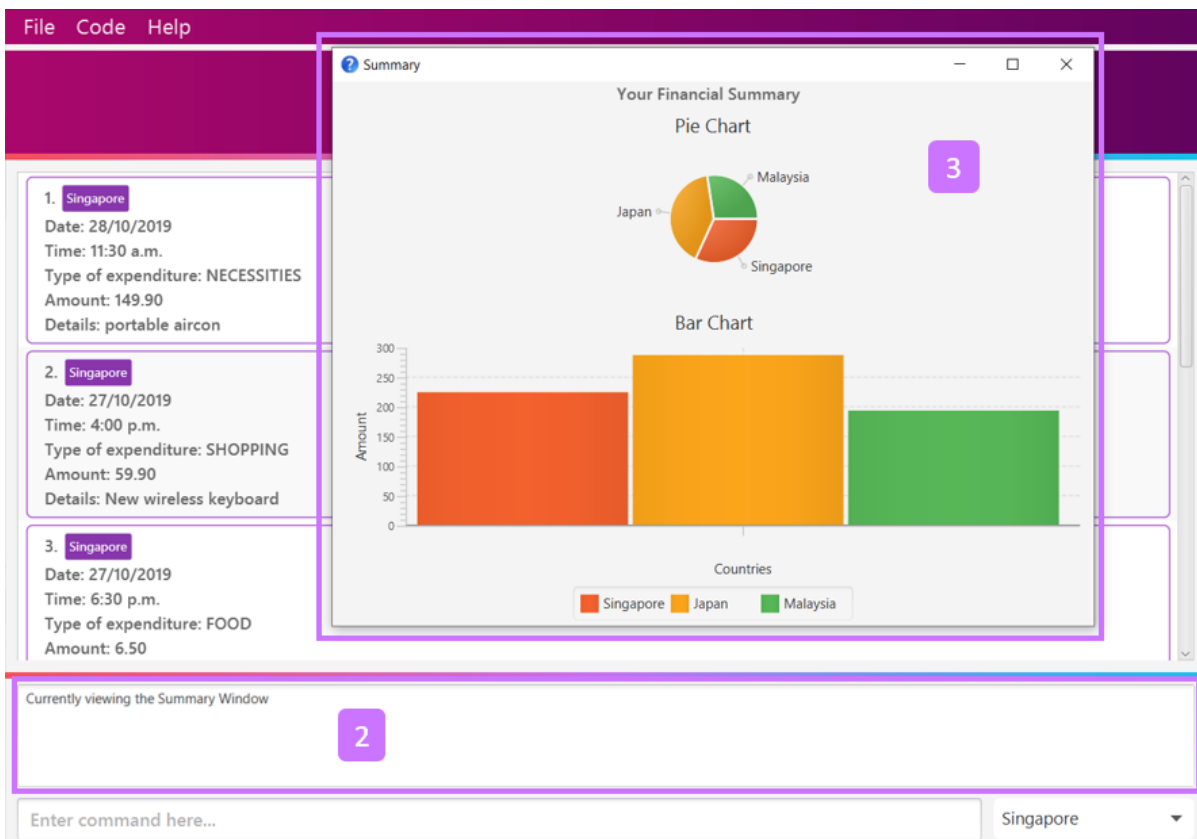
Step by step:

Step 1. Type **summary** in the *Command box* and press *Enter*.



Step 2. The result box will display the message "Currently viewing the Summary Window".

Step 3. As you should have noticed, the Summary Window has been popped out showing you statistics of your expenses in a nice-looking pie chart and bar chart form!



Contributions to the Developer Guide

_Given below are sections I contributed to the Developer Guide. They showcase my ability to write technical documentation and the technical depth of my contributions to the project.

Financial Tracker

The Financial Tracker feature in TravEzy allows users to keep track of their financial expenses with appropriate categorized expenditure and country.

Architecture:

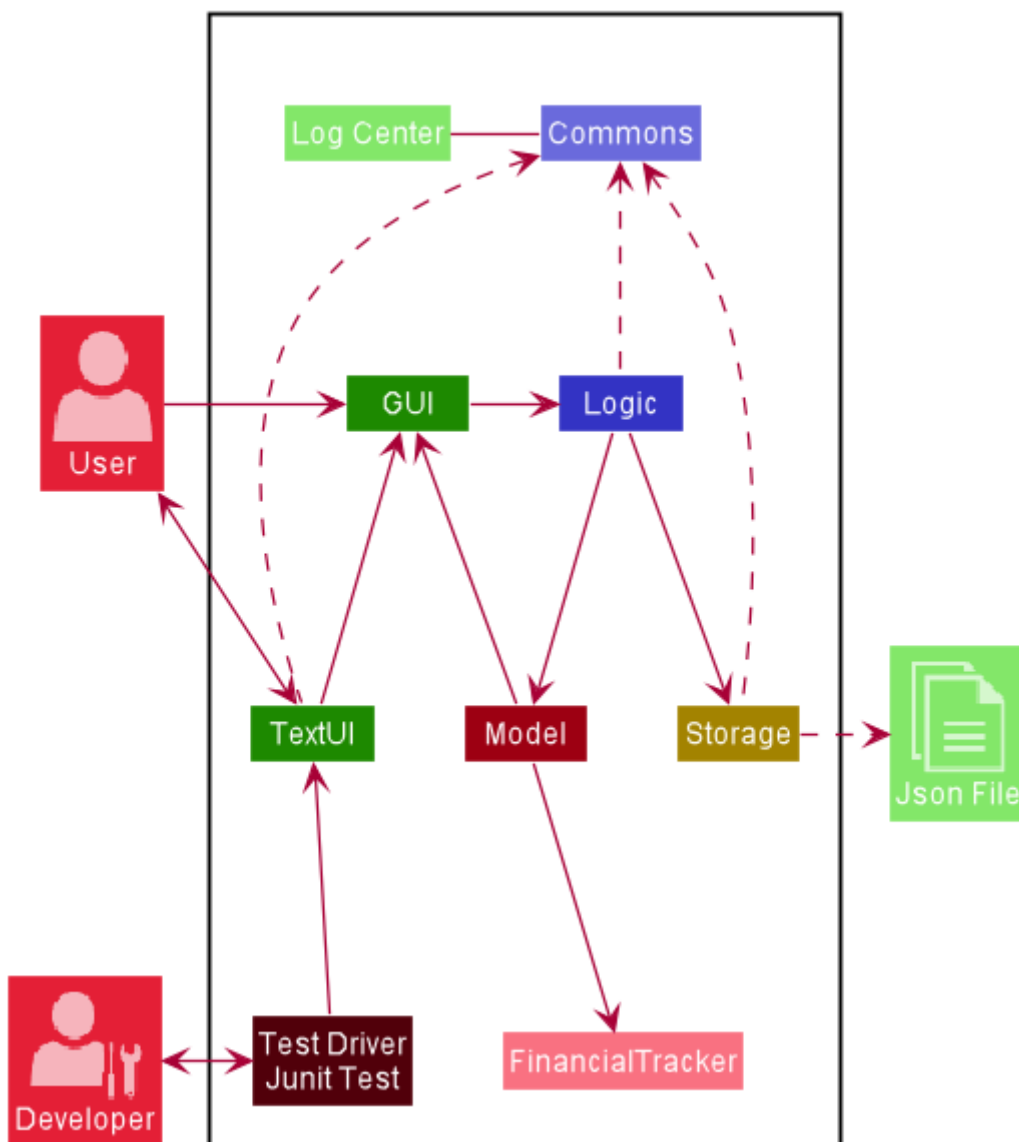


Figure 1. Architecture Diagram of the Financial Tracker feature

The Architecture Diagram given above explains the high-level design of the financial tracker feature. Inputs given by the user are channeled from the **textUI** and parsed by **CommandParser** before different commands are formed which generates the **Model** and updates the **FinancialTracker** object which contains the expense list.

The `textUI`, `Log Centre` and `Storage` all stem from the common package of the main TravEzy application. This is to ensure that each command in financial tracker have only one parser validating the command. The GUI also collaborate with `Model` Class directly, which will be changed to only communicate with `Logic` class in later time.

Financial Tracker Class Diagram

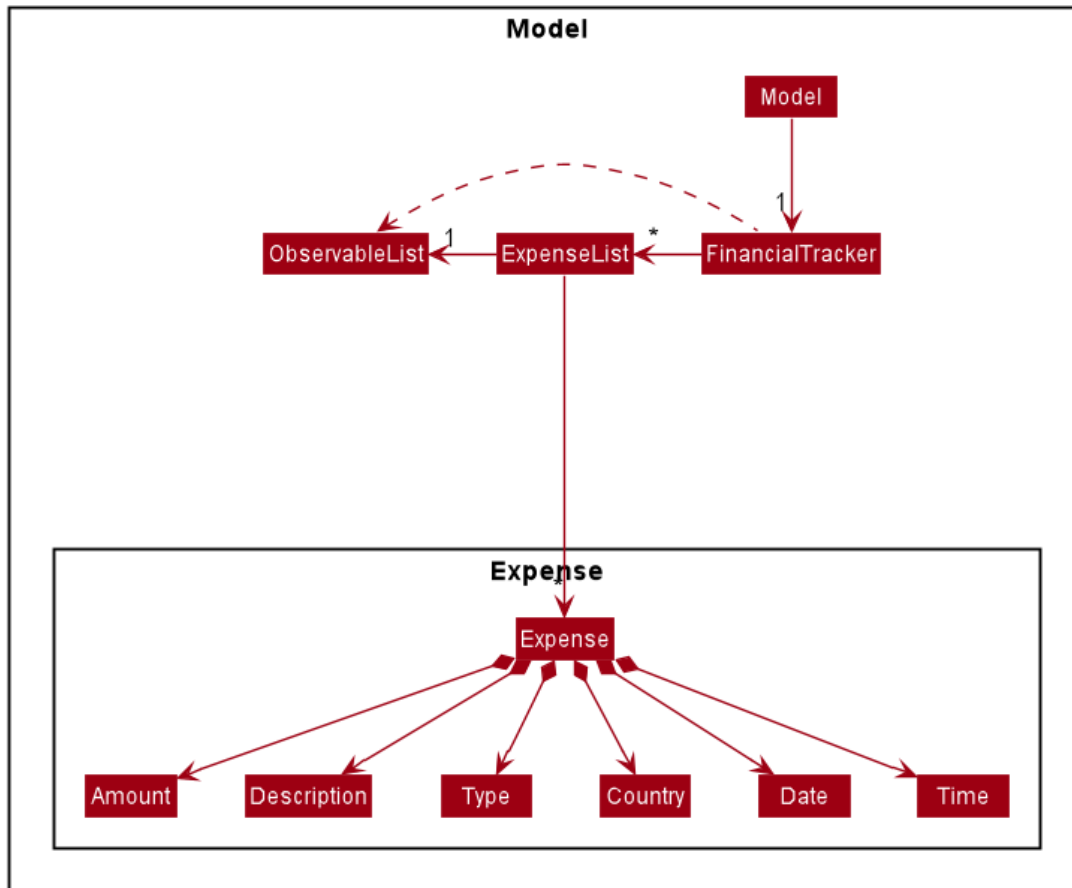


Figure 2. Class Diagram of the Financial Tracker feature

The `Model`, stores the `FinancialTracker` data which includes the `ExpenseList` which keeps track of all the expenses that are included by the user and stores it into the storage in a JSON file.

It exposes an unmodifiable `ObservableList<Expense>` that can be 'observed' e.g. the UI can be bound to this list so that the UI automatically updates when the data in the list change.

NOTE

Each time a new `Country` is selected from the javafx drop down menu, an expense list will be generated for that specific country, this implementation is to ease categorising and sorting.

Sequence Diagram example

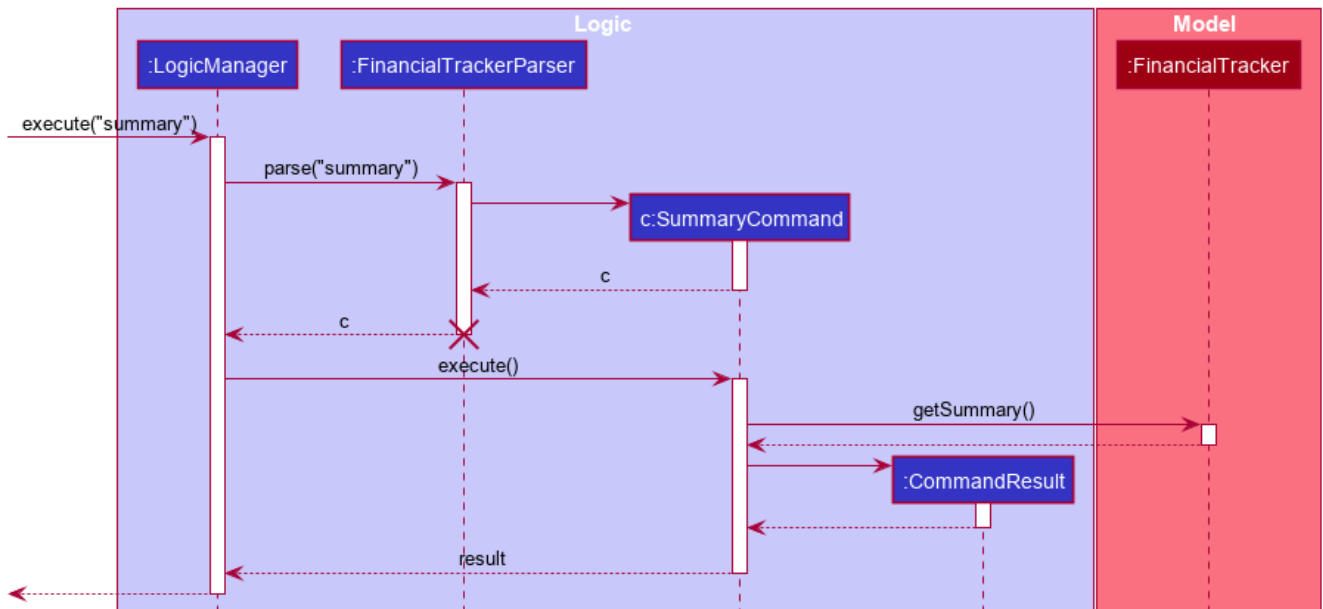


Figure 3. Sequence Diagram of the Financial Tracker feature

The above sequence diagram shows the sequence of events that will take place when a user calls **Summary** command. Note that that above sequence of events are very likely to those for the other commands in Financial Tracker feature.

Activity Diagram example

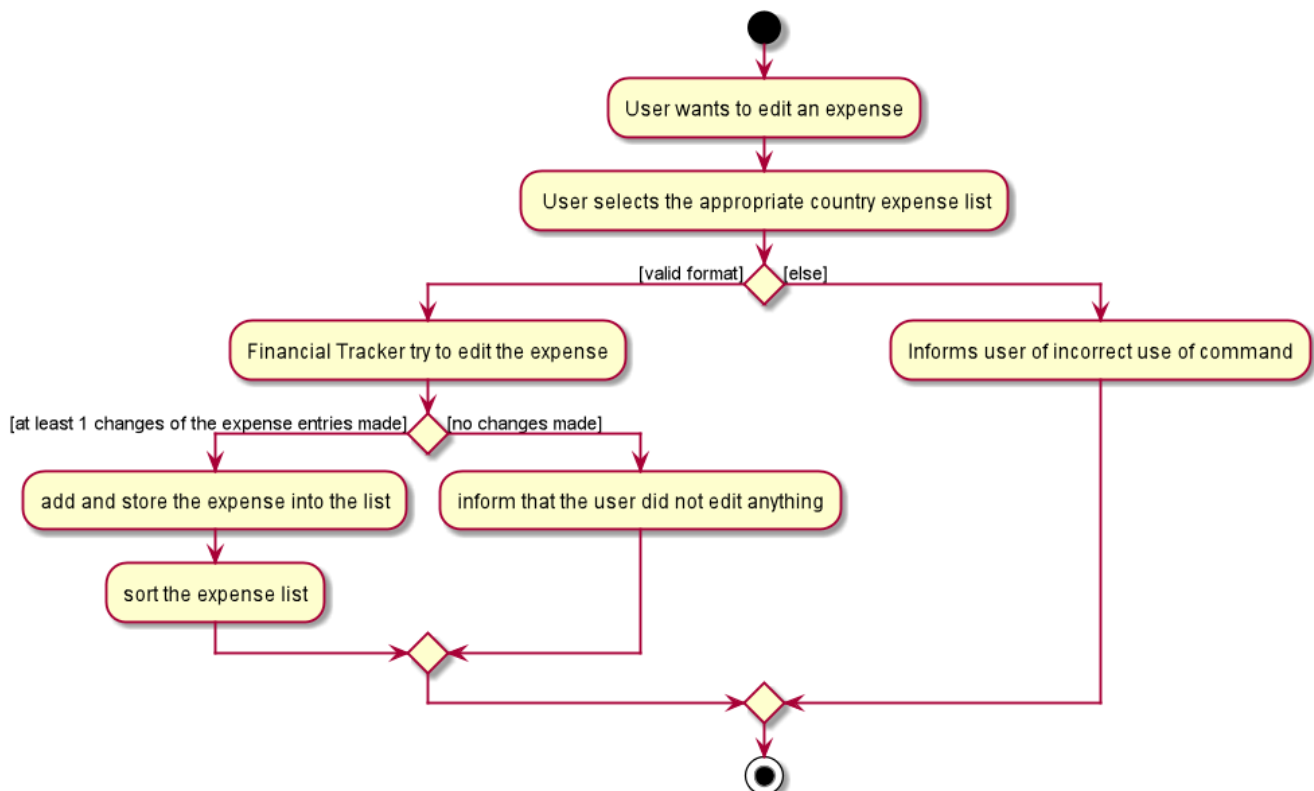


Figure 4. Activity Diagram of the Financial Tracker feature

The above activity shows how **edit** command works.