Design Document: Undo Functionality for Expense Tracker App

Team name: CodeCrafters
Team members: Hemangani Nagarajan, Harshita Loganathan
Git: https://github.com/HEMANGANI/hw2

1. Overview

The Undo functionality is a crucial feature in modern user interfaces, allowing users to easily revert actions they might have taken mistakenly. In our app, users have the ability to remove any transaction, and this removal is immediately reflected in the Total Cost.

2. Features

Transaction Removal:
Users can select any transaction row they wish to remove. A clear visual indicator (e.g., a highlighted row) will show which transaction is currently selected.

Immediate Update:
Once a transaction is removed, the Total Cost is instantly updated to reflect this change. The app ensures that all dependent calculations or displays are also updated in real-time.

3. Design Overview

The design of the Undo functionality is based on the Model-View-Controller (MVC) architectural pattern. This pattern divides the application into three interconnected components:

Model: Represents the data and the business logic of the application.
View: Displays the data to the user and represents the UI of the application.
Controller: Manages user input and updates the Model and View accordingly.

4. MVC Architecture for Undo Functionality

Model
ExpenseTracker: Represents an individual transaction. It has methods to remove a transaction and get the total cost.
Total Cost: Represents the cumulative cost of all transactions.

View
ExpenseTrackerView: Displays individual transactions to the user.
Total Cost View: Displays the cumulative cost to the user.
Undo Button: A UI element that users can click to trigger the undo action.

Controller

ExpenseTrackerController: Manages the interaction between the Model and View. It handles the undo action, updates the Model, and refreshes the View.

5. How to Use the Undo Functionality

Selecting a Transaction:

Navigate to the transaction list/table. Click on the transaction row you wish to remove. The row will be highlighted to indicate it's selected.

Removing the Transaction:

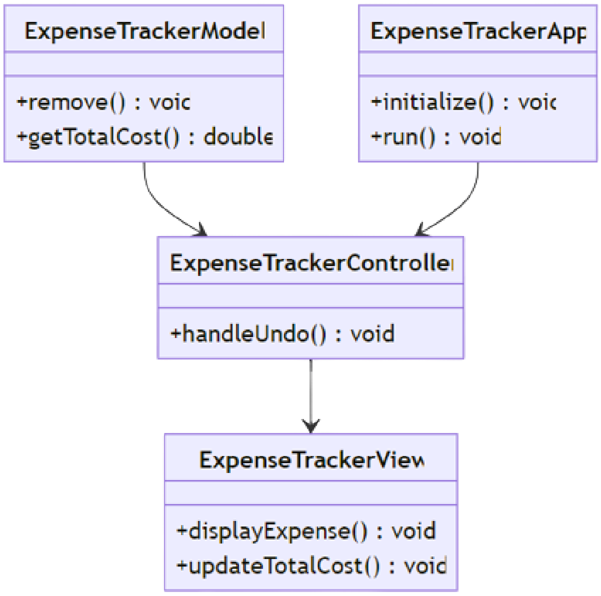
After selecting the desired transaction, click on the "Undo" button. A confirmation prompt will appear, asking if you're sure about removing the transaction. Confirm the removal.

Viewing the Updated Total Cost:

Once the transaction is removed, the Total Cost displayed at the bottom (or a designated area) of the transaction list will be updated.

6. UML Diagram for Undo Functionality

Using the MVC architecture pattern, creating a class diagram to represent the relationships and interactions for the undo functionality.



Description

ExpenseTrackerModel Class:

- Represents the data and business logic of the application.
- Provides methods to remove an expense and retrieve the total cost.

ExpenseTrackerView Class:

- Represents the visual display of the application.
- Has methods to display an expense and update the total cost.

ExpenseTrackerController Class:

- Manages the interaction between the Model and View.
- Contains a method to handle the undo action.

ExpenseTrackerApp Class:

- Represents the main application class.
- Provides methods to initialize and run the application.

The arrows in the diagram represent the relationships and interactions between the classes. The ExpenseTrackerModel class interacts with the ExpenseTrackerController, which in turn updates the ExpenseTrackerView. The ExpenseTrackerApp class initializes and runs the application by interacting with the ExpenseTrackerController.