CS 520
Homework 2
Design & Implementation

Group: Shreeya-Apoorva-hw
Shreeya Patil   Apoorva Rajan

Usability: Undo Functionality [20 points]

For the Usability section related to adding Undo functionality to the Expense Tracker App, the goal is to allow users to remove transactions and reflect the removal in the Total Cost, following the MVC architecture pattern. Here's how we could design this Undo functionality:

Model Changes:

- Create a data structure or a list in the `ExpenseTrackerModel` to keep track of removed transactions. This structure will store the removed transactions and their details (e.g., amount, category, timestamp).
- Create a method in the `ExpenseTrackerModel` to handle the removal of a transaction. This method should remove a transaction from the main list and add it to the list of removed transactions.
- To support undo, create a method to undo the last removal, which takes the last transaction from the list of removed transactions and adds it back to the main list.

View Changes:

- In the `ExpenseTrackerView`, add a way for the user to select and remove a transaction, e.g., a right-click context menu option on a row.
- Update the UI to display the Total Cost and reflect any changes made to it. Make sure that the Total Cost is recalculated after a transaction is removed.

Controller Changes:

- Implement the controller logic for the remove and undo functionality. When a user selects a transaction to remove, call the method in the `ExpenseTrackerModel` to remove it and update the Total Cost. Keep track of the removed transaction in the model.
- When a user invokes an undo action, call the appropriate method in the `ExpenseTrackerModel` to add the last removed transaction back to the main list and update the Total Cost accordingly.

User Interaction:

- Provide clear and intuitive options for users to remove transactions (e.g., via a right-click context menu).
- Clearly communicate the impact of removal on the Total Cost.
- Make the undo action easily accessible, such as through a menu item or button.

Testing:

- Ensure that the undo functionality is thoroughly tested to confirm that it correctly removes and restores transactions and updates the Total Cost.
- Test different scenarios, including multiple removals and undos.

Documentation:

- Update the README file to explain the newly added Undo functionality and how users can interact with it.