Usability: Undo Functionality

This markdown file was originally a pull request message.

homework requirements

- User can remove any transaction altogether by selecting the row.
- The remove update is reflected in the Total Cost.

what this PR does:

Expect behavior:

1. User select a certain row in the transaction table.
2. User click delete transaction button.
3. The selected transaction gets removed
4. Changes also reflects in the the total cost field.
5. The transaction in list is deleted

view

- Add a delete button at the bottom of the panel https://github.com/xyw0025/hw2/pull/3/commits/f02d4df182a71e7f179c14a6c7b7546cd805c2cf https://github.com/xyw0025/hw2/pull/3/commits/9ad5bfc714c32f787e58eea43caf7cdea0f99f0f

```


...
public class ExpenseTrackerView extends JFrame {

  ...
  private JButton deleteTransactionBtn;
  ...

  public ExpenseTrackerView() {
    ...
    deleteTransactionBtn = new JButton("Delete");

    ...
    inputPanel.add(deleteTransactionBtn);

    ...
    buttonPanel.add(deleteTransactionBtn);
    ...

  }


  ....
  public JButton getDeleteTransactionBtn() {
    return deleteTransactionBtn;
  }

  public void deleteTransactionRow(int index_number) {
    this.getTableModel().removeRow(index_number);
  }
}
```

model

- add method removeTransaction: takes an integer `index_number` as parameter, remove the transaction at index `index_number` in list

https://github.com/xyw0025/hw2/pull/3/commits/4a3924fd87c67e61b42bbef75dc9889bda8bf053

```
...
  public void removeTransaction(int index) {
    transactions.remove(index);
  }
```

ExpenseTrackerApp.java

- add listener to delete transaction button
- when listener gets triggered
  - get the index of the selected row
  - call method `deleteTransaction` in controller

https://github.com/xyw0025/hw2/pull/3/commits/d6a35dd77dd6f2a20ac9ed78a0e10fb9ead0592f

```
.....
    view.getDeleteTransactionBtn().addActionListener(e -> {
      boolean is_deleted = deleteTransaction(controller, view);

      if (!is_deleted) {
        JOptionPane.showMessageDialog(view, "There's no such transaction
in the table!");
        view.toFront();
      }
    });
  }

  private static boolean deleteTransaction(ExpenseTrackerController
controller, ExpenseTrackerView view) {
    int row_index = view.getTransactionsTable().getSelectedRow();

    if (checkRowNum(view, row_index)) {
      boolean is_deleted = controller.deleteTransaction(row_index);
      return is_deleted;
    } else {
      return false;
    }
  }

  private static boolean checkRowNum(ExpenseTrackerView view, int
row_index) {
    int row_counts = view.getTransactionsTable().getRowCount();
    return (row_index < row_counts);
  }
```

Controller

when ExpenseTrackerApp.java calls method in controller, meaning that the delete button is clicked and the selected row is indeed a transaction: https://github.com/xyw0025/hw2/pull/3/commits/d6a35dd77dd6f2a20ac9ed78a0e10fb9ead0592f

- `controller` calls method in `model` -> delete the transaction in the list
- `controller` calls method in `view` -> remove the row

```
  // Other controller methods
  public boolean deleteTransaction(int index_number) {
    model.removeTransaction(index_number);
    view.deleteTransactionRow(index_number); // the row number also start
from 0
    refresh();
    return true;
  }
```