CS 3110 MS3 Report

Team Members: Flynn Kelleher (fpk24), Harsh Patel (hhp26), Ilia Raiszadeh (ipr6), Joonseok Jung (jj575), Kayla Runkel (kmr227)
Team Name: Pixel Princesses
PM: Allen Wu (axw4)
PG: Rachael Close (rlc355)

Vision (one paragraph; current vision for system?; how has it evolved?)

The current vision for the system is to have a custom created user interface that hosts our expense tracker application. This application has multiple windows that emulate the experience of using a real application. The system not only allows users to load pre-existing expense information, but now also allows them to add expenses within the software and save them to a csv file on their computer. The system also takes in user information regarding budgeting and combines that with the expenses inputted to provide multiple levels of budgeting suggestions. The vision of the system evolved from just being a terminal based application to an actual graphical user interface. It has also evolved from being an analyzing application to one that actually applies the analysis of the expenses to provide intelligent feedback to the user regarding their spending.

Summary (one-two paragraphs; what have we accomplished?; what functionality?)

For MS3, we implemented 7 main categories of features. To begin, the user has the option to load an existing csv file or to create a new one. From there, the user has seven choices:

1. **View Expenses**: Each expense has a description, category, amount, and date. This window allows the user to scroll through their expenses (using the [w] and [s] keys).
2. **Total Expenses**: This window displays the total amount of dollars spent when considering all the expenses.
3. **Read Expenses**: Similar to the load screen, the user can reenter a CSV file.
4. **Save Expenses**: The user has the option to save their expenses to a new CSV file.
5. **Analyze**: When the Analyze button is clicked, the user can then choose between displaying a Pie Chart, a Bar Graph, or viewing Budget recommendations.
6. **Add Expenses**: Add expenses lets the user input a new expense to add to their list of expenses. The user enters a description, a category, an amount, and a date to describe their expense. The user should save their updated expense list to a new CSV if they want to save these added expenses.
7. **Exit**: Once the user has finished working with their expenses, they can exit the program and the window closes.

CS 3110 MS3 Report

Activity Breakdown (responsibilities, activities, features, hours)

Flynn:

- Primarily worked on the front end with Joon and Kayla. Began by creating the button infrastructure and moving most of the application out of the terminal
- Created the help buttons and helped consolidate some of the functions in main
- Implemented the test suite for budgeting
- Developed the budgeting system that allocates percentages based on income/goals/etc.
- Hours: 20

Harsh:

- Primarily worked on the backend with Illia to create the functions that were required to create expense types and provide the information for the pie and bar charts
- Worked on improving the pie and bar charts in the frontend
- Developed additional test cases to test the budgeting algorithm and also wrote tests for pie charts
- Hours: 20

Ilia:

- Worked particularly in the backend to form the backbone of the data structures we used to track expenses and work with users data.
- Implemented the compilation unit for the expense tracking data structure and a variety of relevant functions and operations our program uses
- Hours: 20

Joon:

- Primarily worked on the front end with Kayla and Flynn
- Implemented resizable window functionality and view expenses screen
- Debugged bar chart's axes
- Fixed main function to eliminate all hard-coded numbers and improve code readability
- Hours: 20

Kayla:

- Primarily worked on the front end with Flynn and Joon
- Worked on OUnit test suite
- Implemented initial pie and bar chart functionality
- Assist with button functions, layout and use
- Hours: 20

Productivity Analysis (one paragraph; how productive?; accomplish what we wanted?)

CS 3110 MS3 Report

In terms of productivity, there were many positive points to our operation and some areas where we could have improved on. One of the positive points that aided in our productivity was the level of communication between the group members. All the members of the group were always active on our slack and so there was always clear communication between everyone regarding tasks. Another aspect that really bolstered our productivity was the use of micro-deadlines. We would create multiple small deadlines for features and fixes in order to keep yourself accountable and to ensure that the project progresses. One of the areas we could have improved more on was the use of GitHub's additional features. We could have utilized branches more effectively to aid in simultaneous development. Oftentimes, we would have to wait until something is pushed in order to work on another part of the project which sometimes slowed things down. Ultimately, we did accomplish all that we set out for in this project. We successfully implemented all the features that we had shortlisted for the project.