Steven Guikal August 13, 2019

**Problem Definition**

After conducting and interview and spending some time thinking of the problem, I believe I have come up with a clear understanding of what is required. This budgeting application will start off with a simple menu from which all actions can be performed. After any main output occurs, such as a monthly breakdown, the user will have to press enter to bring up the menu so that its appearance doesn’t block the current output. The menu will have the following categories for ease of use: Add Expense, Add Income, Modify Categories, Plan, Search, Savings Goals, and View. Just like all of the previous assignments, the Add Expense, Add Income, Modify Categories, Plan, and Savings Goals will be simple, text based input menus to satisfy a user’s choice. The Search option will first have the user select a search key (Name, Category, or Date) and then choose a time period for their choice (Certain Month, All). Finally, they will enter their search string and the program will perform a sequential search (Name, Category) or binary search (Date) for the given string. The transactions matching this search will then be shown.

While only taking up one menu slot, the View option is perhaps the most important of all the functions. This menu option will be able to show the user breakdowns of current and past months, along with a planning and goals display. The current month display will show a running balance in the top left along with the total expenses beside that. A breakdown of the expenses will be placed underneath with a list of transactions under that. Viewing past months is nearly identical except instead of a running balance, the savings for that month are written. For a total breakdown of all past activity, a current balance, total expenses, and total savings (current balance – current month’s expenses) are shown in one row at the top. Underneath that information comes a table of the past several months with the month, its savings, and its expenses in rows. With all of the expenses taken care of, the planning and goals page is the only one left. The left side of the screen contains the planning section with current and planned breakdowns for every category and the right side contains the goals section with a list of goals sorted by priority along with a calculated completion amount where savings are split proportionally between priorities.

With the input and output sections covered, the last section of the IPO chart remains. To further ease of use in the application, manual saving and loading to and from files will not be required. Instead, the program will automatically save all arrays to files and load them on program start. This also has the added benefit that if the user accidentally closes the program or their computer shuts down, the most progress loss that could occur would be a single action (transaction modification, income creation, etc.). Because all transactions are stored, all displayed numbers can be recalculated on the fly, instead of having to store them and modify them. While this does make the program slightly less efficient, the amount of input we have is minimal so little to no effect will be visible. To counteract this, other areas, such as searching use more efficient methods. For example, when searching for a transaction with a specific date, a binary search is performed because transactions are stored in a sorted array by date. This binary search is much faster than a sequential search and is in a place that has the most objects compared to any other part of the program.