Spending Tracker Java Implementation

Brian Brake & N'dia Thomas

IMPLEMENTATION PROCESS:

We wanted to create a simple program to keep track of financial expenses. Since we lost a few members, it is not as detailed as we hoped and planned it to be, but we made sure the most important functions were working.

We communicated a lot via text during the process to ensure that everything was going right and give each other ideas and suggestions for their code. We also met several times on-campus to plan out the project and discuss how we were going to code certain parts.

We created frames for each activity, including the start. The Setup frame would allow a user to enter their income and select the categories they want to track, the Add Expense frame would allow a user to add individual expenses to their selected categories, the Tracking frame would allow a user to visually view their expenses in 3 different ways: pie chart, expense vs. income chart, and a list chart; and the User Profile frame would let multiple users use the app by adding new users and letting others log in. The User Profile part was eventually dropped due to difficulty at the time.

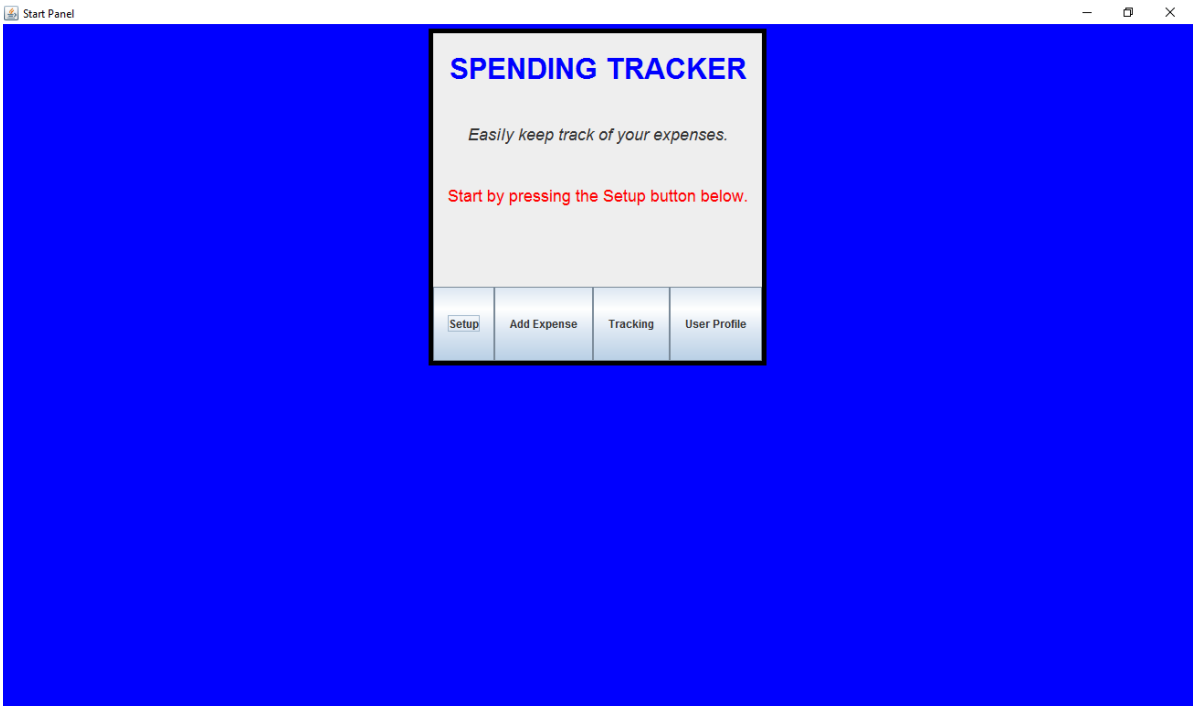
The list of active categories, probably the most important of the program, is an ArrayList that can be modified by adding and removing elements. The ArrayList, referred to in the program as catList, is initialized by the category checkboxes in Setup, and is heavily used in the AddExpense and Tracking frames.

There was only one major case where we had to borrow code, and it was with the pie chart (PieChartFinal) and converting the colors of the slices into Strings (ColorUtils) that could be used in the key. Both classes are cited at the top.

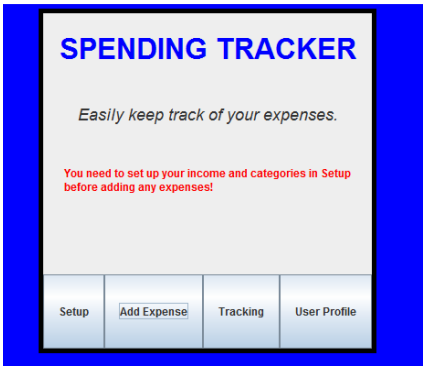
Overall, the implementation process was both enjoyable and frustrating as we progressed from creating the initial GUI (the Layouts were tedious, but we settled with GridBag) to actually connecting the code to the GUI, to seeing a functioning program.

DETAILED OVERVIEW OF PROGRAM:

As said above, this program functions as a basic spending tracker to keep track of expenses. This is done by setting up your income and categories you want to track, adding monetary expenses to the categories selected, and viewing your total expenses in three different methods.

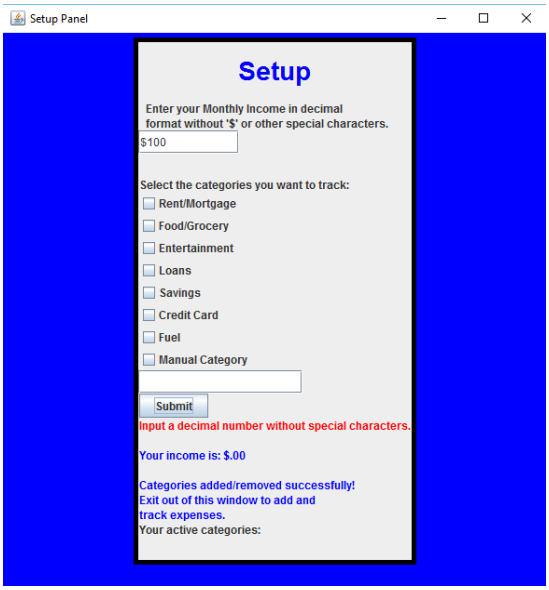


To use the app, start it (by running the StartFrame) and select Setup. If you try to click the Add Expense button before setting up your income or expenses, you will see an error message:



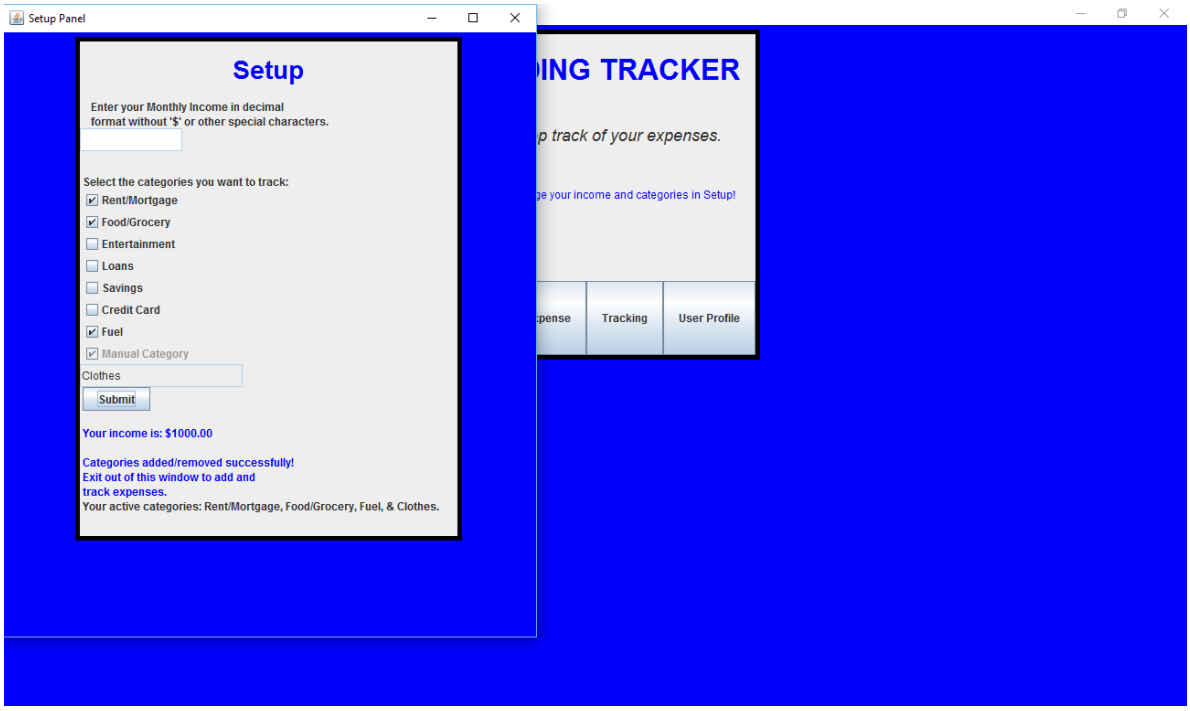
In the Setup frame, you can add your income, and check the categories that you want to track, and then press the Submit button to submit the information.

If you leave income box blank or enter a non-double string, you will see an error message:



You can also add **one** manual category by checking its box and entering the name you want to give it, which in this case is “Clothes”. We currently don’t have a way to add more than one category, but if we were going to do so, we would add a JButton called “Add another category” that would add (or set visible) an additional JCheckbox and JTextfield, up to a limited number of manual categories that are implemented the same way as the current one.

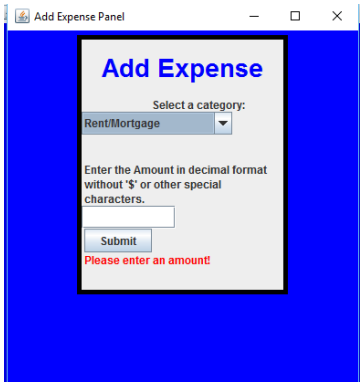
Once you click Submit after properly adding the income and selecting categories, this will appear:



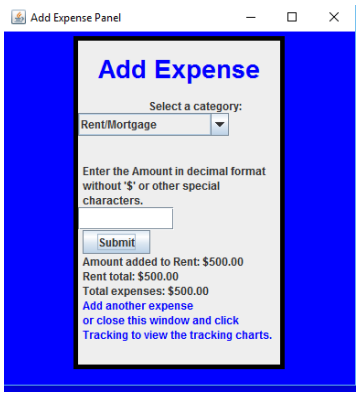
The text shows your income and your active categories, which in this example is $1000.00 and Rent/Mortgage, Food/Grocery, Fuel, and the manual category Clothes.

After this, you can access the Add Expense frame, so you can close out of Setup and click on the Add Expense button. Note that Start screen now states that you can change your income and categories in Setup (we will come back to this later).

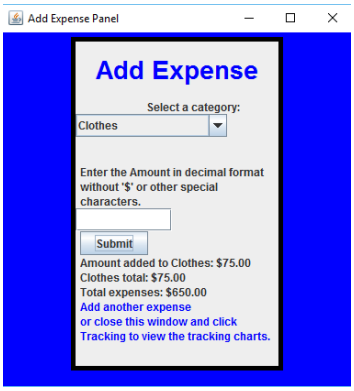
At the Add Expense frame, you can add expenses to an active category that you selected from the drop-down box. If you don't enter anything or don't type in a double, you will get an error message:



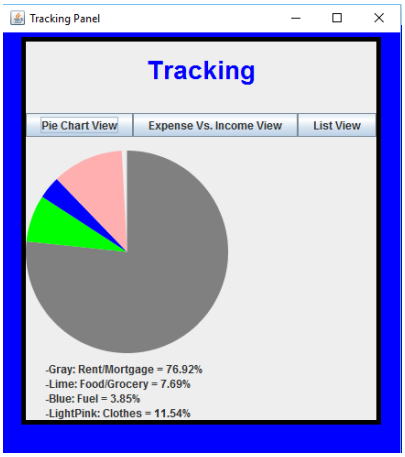
Once a number is added to a category and the Submit button is clicked, this text will appear:

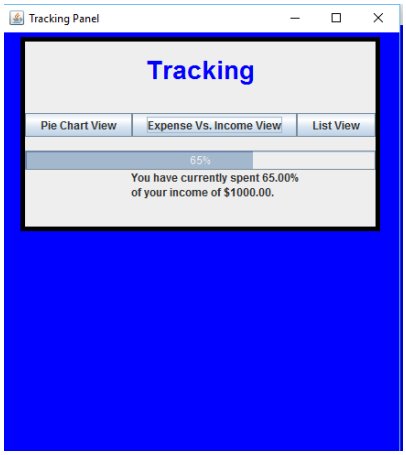


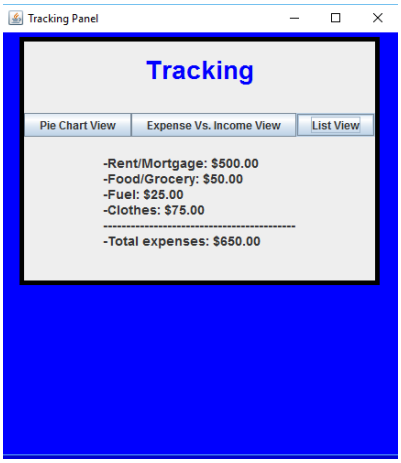
The information shows the amount you added, the current total for the category you added to, and your total expenses. You can continue to add to this category, or change the category and add to that one. For this example, say you've added $500 to Rent, $50 to Food/Grocery, $25 to Fuel, and $75 to Clothes for a total of $650.



You can exit out of the window, re-open it and continue to add expenses. This data will update in the Tracking frame, as seen here after clicking each of the buttons:





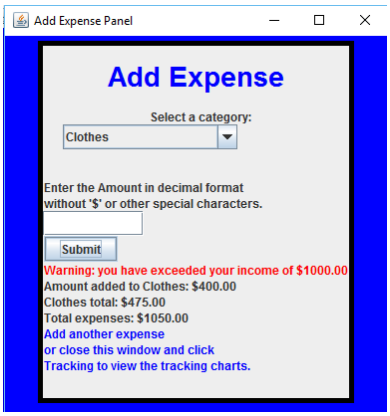


In **Pie Chart View**, you can see how each of your categories' expenses weigh compared to each other. The text shows the color and percentage of each category that has an expense of more than $0.

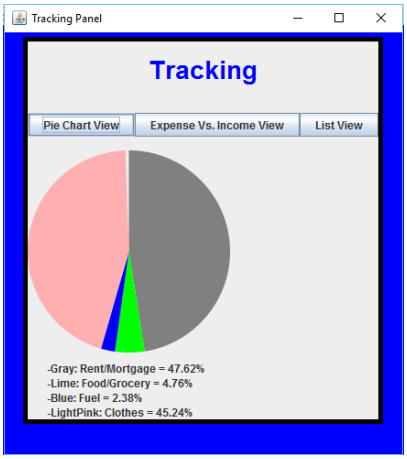
In **Expense Vs. Income View**, you can see how your total expenses compare to your income. In this case, you've spent 65% of our income of $1000.00

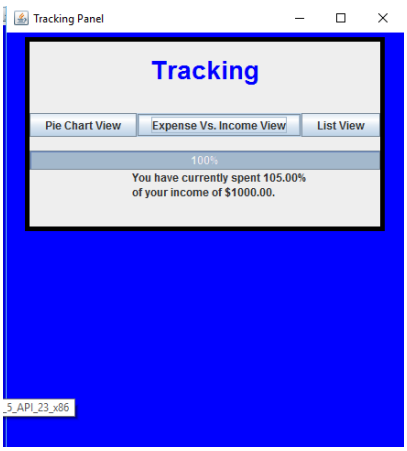
In **List View**, you can see the total expenses of each category that is greater than $0, and the overall expenses.

Now let's say that you have to buy a pricey $400 dress or tuxedo for a formal dinner. You can exit out of the Tracking window and go back to Add Expense and add this in:



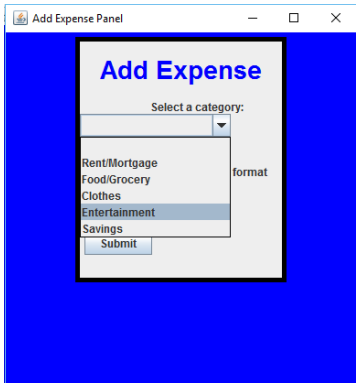
You get a warning saying that you've exceeded your income of $1000.00. This and future expenses can still be added, but this warning will stay. The updated expense will show in Tracking:



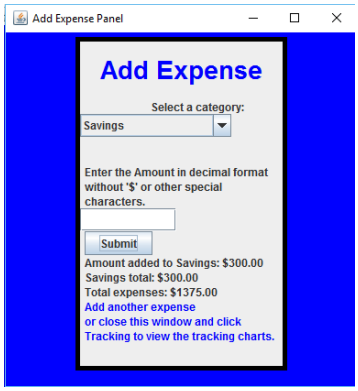


Say that you're given an extra $500 by your parents, and you want to put some it in savings and spend some for yourself. You can change your income to $1500 in add expense and select the Entertainment and Savings categories. You also decide that the Fuel expense is quite minimal so you want to remove it. You can simply deselect its checkbox:

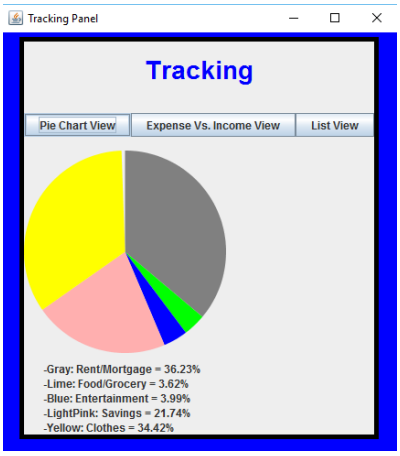
After clicking submit, your income is changed, the Entertainment and Savings categories are added, and the Fuel category is removed. You can go back to Add Expense to add money to Savings and Entertainment, which now appear in the drop down menu:

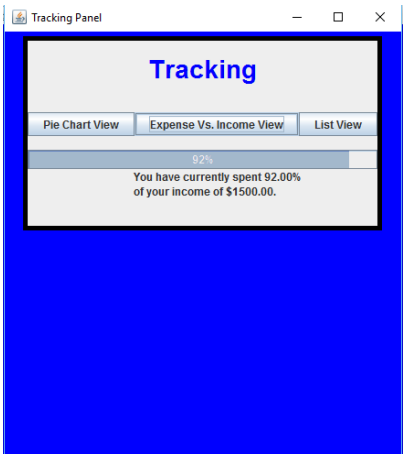


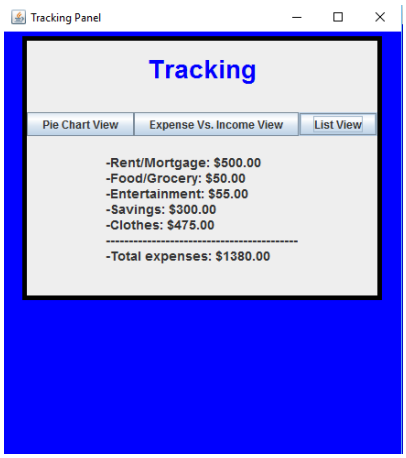
You can now add $55 to Entertainment and $300 to savings. Since you've increased the income, 'the exceeded income' warning goes away.



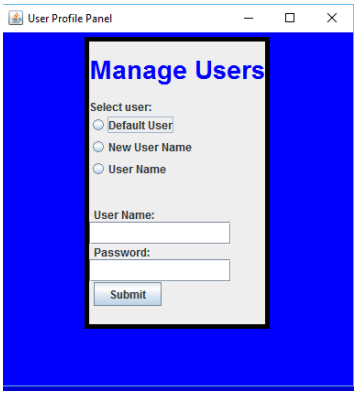
If you exit out of this window and go to Tracking, everything has been updated (namely removing the Fuel category information):







As for the User Profile, we were unable to completely implement this concept as it seems to require storing data in a file or database (which we attempted to do in the Database class) – which is also why this program is only functional when it is open; data is not retained. We did however design the User Profile window:



Clicking the Default User radio button would keep everything normal and allow for the current user to keep using the app on their own. Selecting the New User Name radio button would allow for a user to create a new user account to store and track their own expenses by inputting a username and password. The User Name radio button would allow for a user to login and access their expenses.

If were were to read and write data to a file, we would use the I/O to output individual expenses for each category. This way, we could potentially add a way to name and also remove individual expenses. We would also input data from the file to populate the income and category data for each user. We did not do this because it was not planned nor was it a focus in this project.