

## Criterion B: Design

### File Provided by Client:

My client Sunny, has provided me with 2 google sheet files for my usage, which I will convert to CSV files and incorporate it into my program.

A	B	C	D	E	F
Entry	Year	Category	Transaction	Amount	
Spring Camp	2021	Camp Fee	E-transfer	1961.65	
Zoom Fee	2021	Fee	Cash	84	
Summer Camp	2021	Camp Fee	E-transfer	1718	
Spring Camp	2022	Camp Fee	E-transfer	920	
Summer Camp	2022	Camp Fee	E-transfer	680	

Figure 1: Revenue File, Spreadsheet that displays categories into how the revenue data is structured and distributed.

	A	B	C	D	E	F	G
1	Entry	Year	Transaction	Category	Amount	GST	
2	Stollery Charity	2021	E-transfer	Donation	686	0	
3	Spring Camp	2021	Debit	Payment	1161.37	55.3	
4	Yearly Slidesgo	2021	E-transfer	Subscription	31.96	1.52	
5	Summer Camp	2021	Debit	Payment	793.8	37.8	
6	Spring Camp	2022	Debit	Payment	1401.23	66.73	
7	Instructor Fee	2022	Cash	Payment	21	1	
8	Summer Camp	2022	Debit	Payment	612.05	29.15	
9							

Figure 2: Spendings File, Spreadsheet that displays categories into how the spendings data is structured and distributed.

### Categories of tables:

Spendings	Revenue
+ ID: Int (PK) (Not Null)	+ ID: Int (PK) (Not Null)
+ Year: Int (Not Null)	+ Year: Int (Not Null)
+ Transaction: Text (Not Null)	+ Category: Text (Not Null)
+ Category: Text (Not Null)	+ Transaction: Text (Not Null)
+ Amount: Float (Not Null)	+ Category: Text (Not Null)
+ GST: Float (Not Null)	

Figure 3: Tables displaying the 2 different databases, Spendings and Revenue

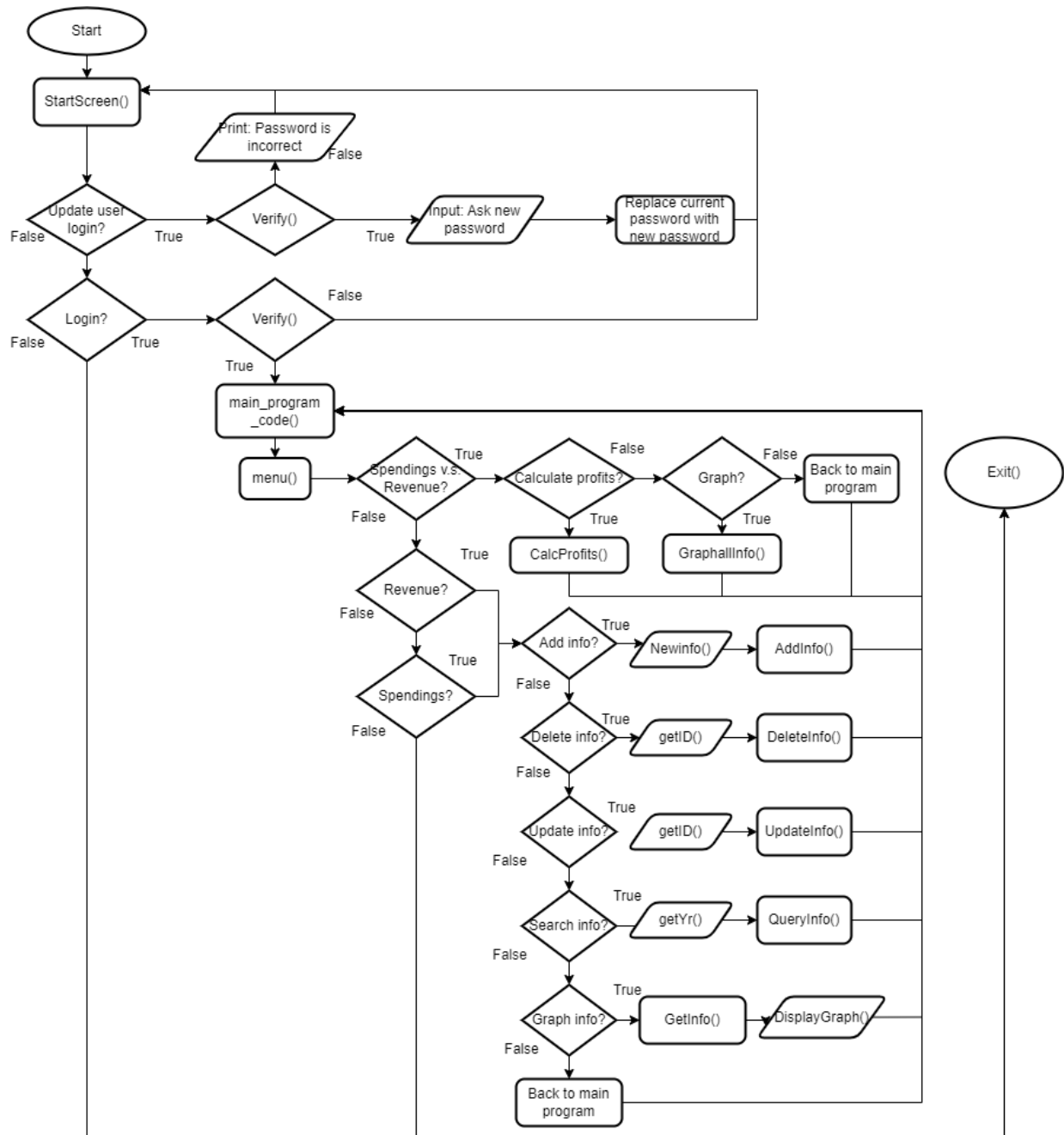


Figure 4: Flowchart demonstrating the general outline/logic of the program

## Subprogram and Important Algorithms

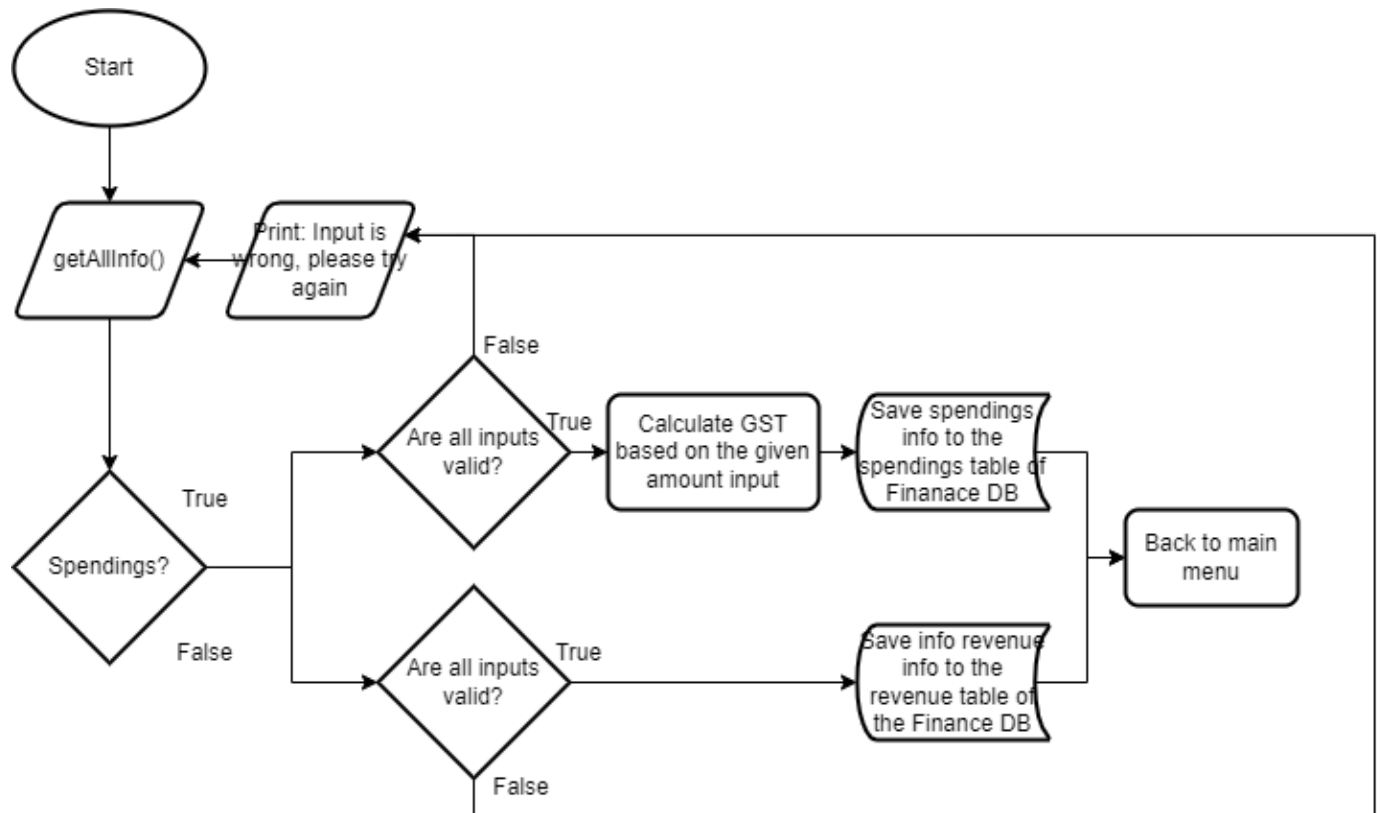


Figure 5: Adding new data into the database, into the spendings/revenue table

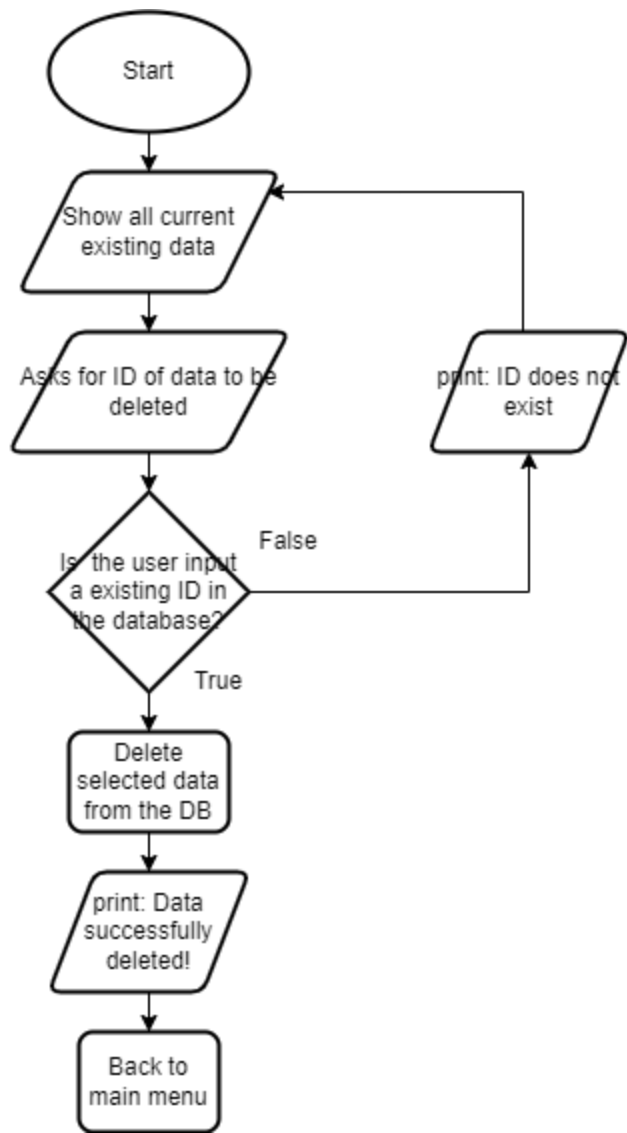


Figure 6: Removing the data from the database

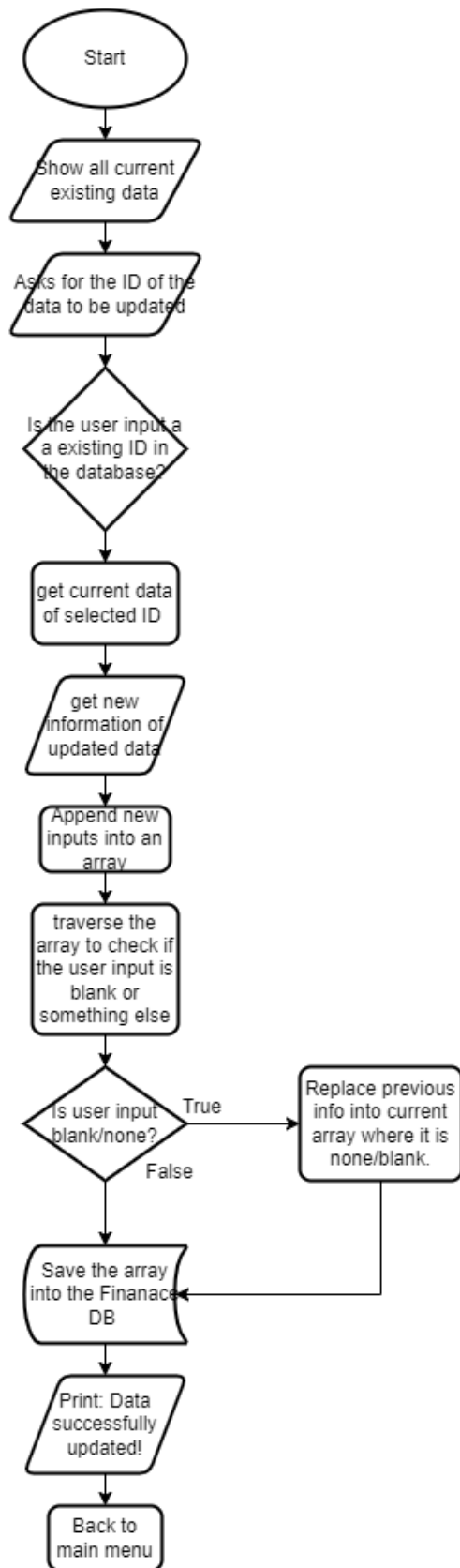


Figure 7: Updating data in the database

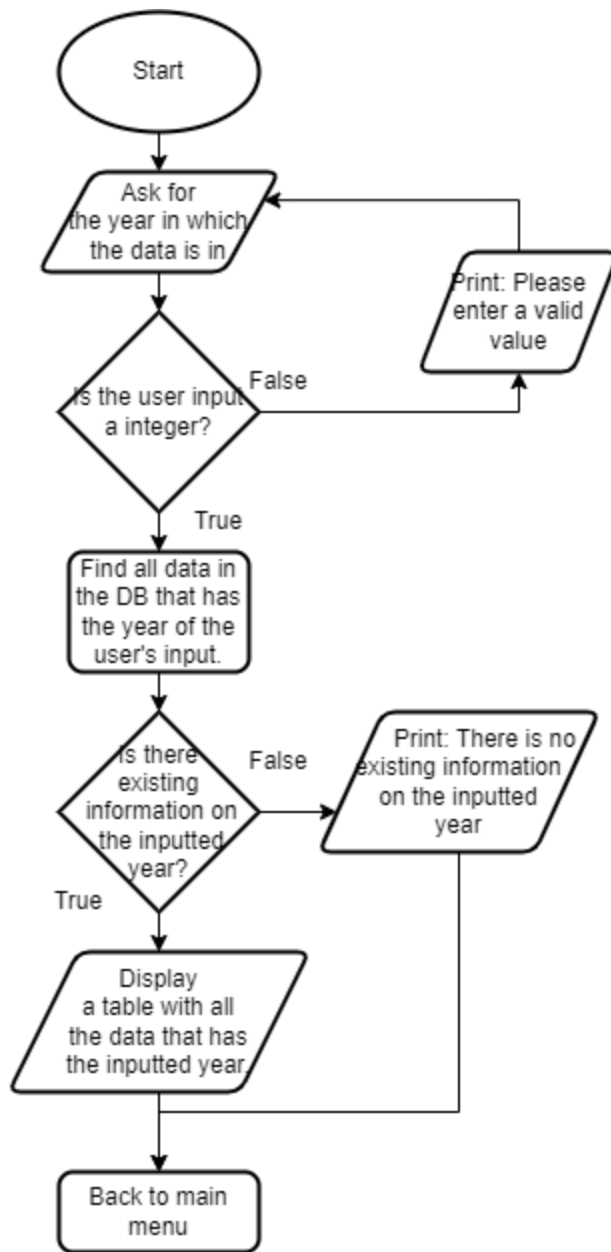


Figure 8: Searching for data in the database

## User-Interface Overview

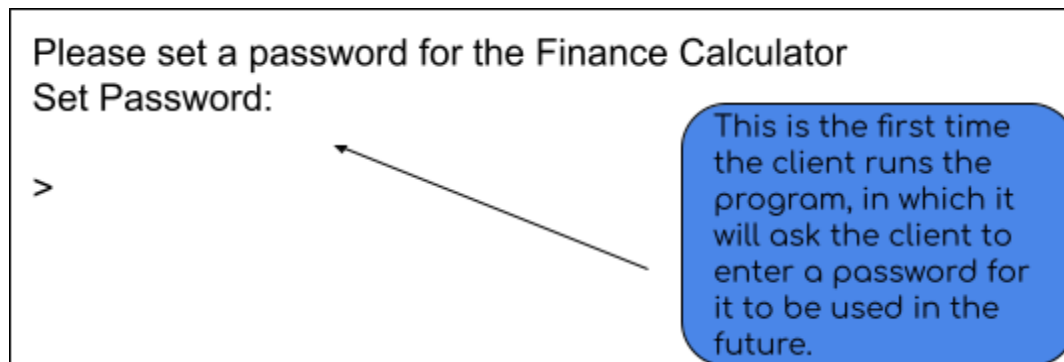


Figure 9: First time a user is running the program

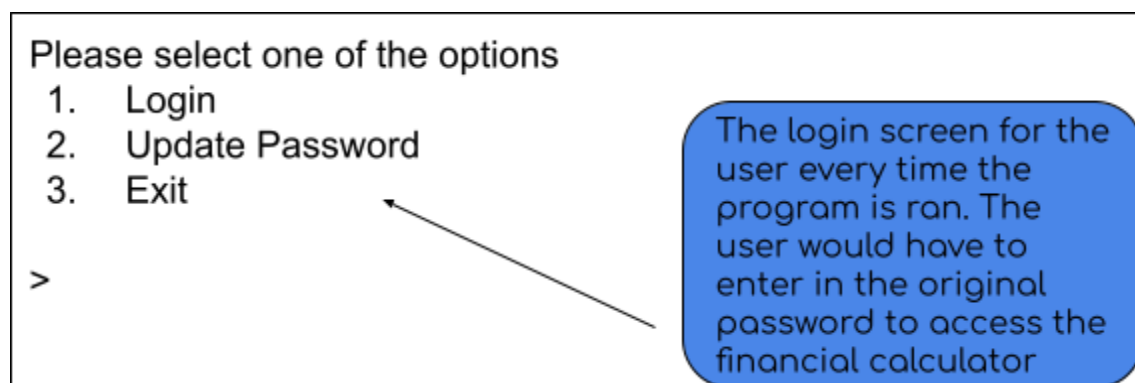


Figure 10: Login screen every time the user starts the program

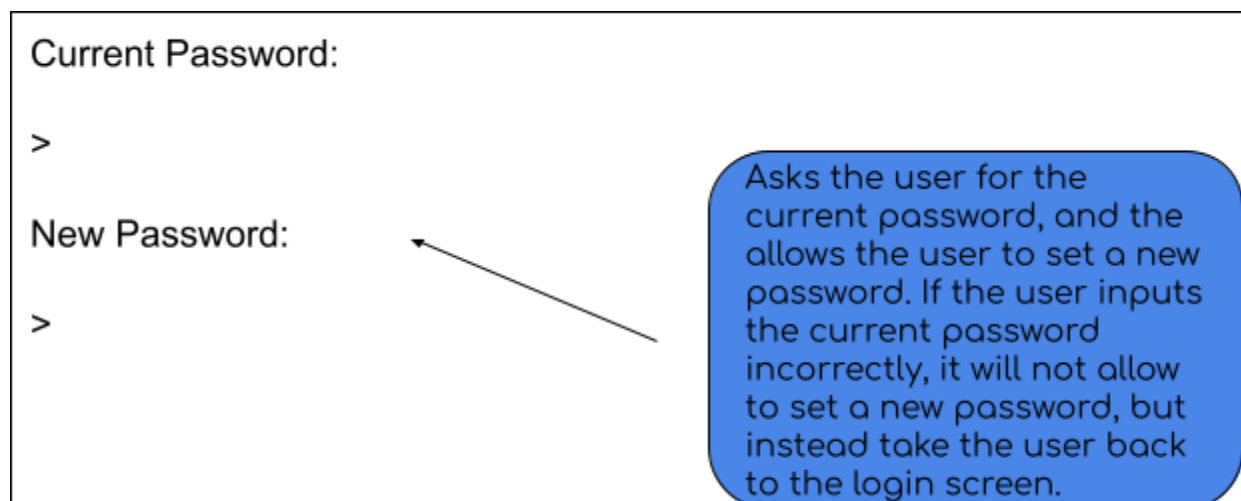


Figure 11: Set new password displayed if option 2 was picked from the login screen

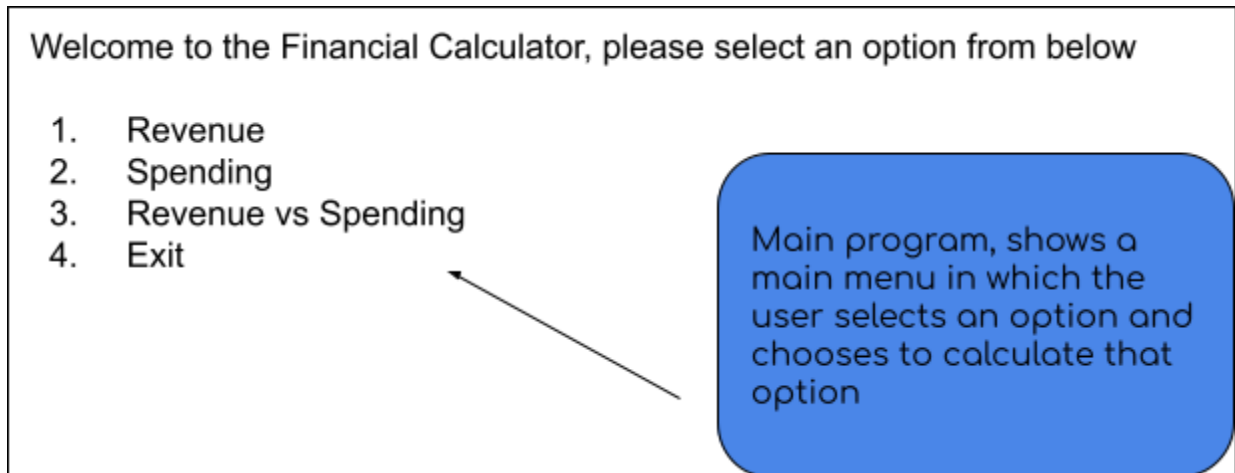


Figure 12: Shows a main menu after choosing the login option (1)

Recent Transactions						
ID	Entry	Year	Transaction	Category	Amount	GST
5	Spring Camp	2022	Debit	Payment	1401.23	66.73
6	Instructor Fee	2022	Cash	Payment	21	1
7	Summer Camp	2022	Debit	Payment	612.05	29.15

Please Choose a Integer

1. Add new data
2. Update data
3. Delete existing data
4. Search for a dataset
5. Graphing
6. Back

Displays 3 most recent transactions, displays a submenu for the user to choose.

Figure 13: Submenu displayed after choosing Revenue (1) or Spending (2), but revenue does not have the GST column



Entry/Context: **Summer Camp Fee**  
 Year: **2023**  
 Category Payment/Other: **Payment**  
 Transaction: **E-transfer**  
 Amount: **1000**

Data Successfully Added!

Asks the user information one by one and then displays confirmation after successfully adding new information into the database. User input is in red

Figure 14: Gets the info and adds new data if option 1 in the submenu is selected

Please select an id

ID	Entry	Year	Amount
1	Stollery Charity	2021	686
2	Spring Camp	2021	1161.37
3	Yearly Slidesgo	2021	31.96
4	Summer Camp	2021	793.8
5	Spring Camp	2022	1401.23
6	Instructor Fee	2022	21
7	Summer Camp	2022	612.05

>  
 Leave field blank for no changes  
 Entry (Random entry): **Random Entry**  
 Year (2021): **2022**  
 Category (Random Fee): **Fee**  
 Transaction (E-transfer):  
 Amount (920.0): **1023**  
 Information successfully updated!

Displays table of all information, user chooses one id. Asks one by one new information to be replaced with current information, current information is in the brackets. If a input is none, than it remains the current info.Red is user input. Confirmation message at end.

Figure 15: Updates info if option 2 is selected from submenu

Please select an id

ID	Entry	Year	Amount
1	Stollery Charity	2021	686
2	Spring Camp	2021	1161.37
3	Yearly Slidesgo	2021	31.96
4	Summer Camp	2021	793.8
5	Spring Camp	2022	1401.23
6	Instructor Fee	2022	21
7	Summer Camp	2022	612.05

> 4  
4 successfully deleted

Displays a table with the current information and asks the user which id to be deleted. Red is user input. At the bottom displays a message confirming the id that is deleted.

Figure 16: Deletes info if option 3 is chosen from submenu

Input the year of the data:  
> 2022

Available Info

4	Spring Camp	2022	Camp Fee	E-transfer	920
5	Summer Camp	2022	Camp Fee	E-transfer	680

To proceed, press any key

User inputs a year of the data that is desired, and then the program shows a table with the following data.

Figure 17: Queries info if option 4 is selected from the submenu

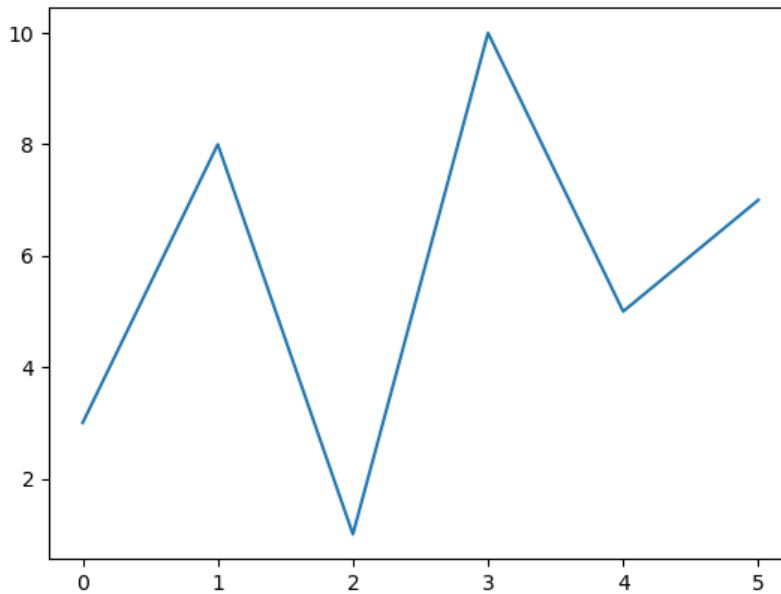


Figure 18: A graphical representation of the data appears if option 5 is selected on the submenu

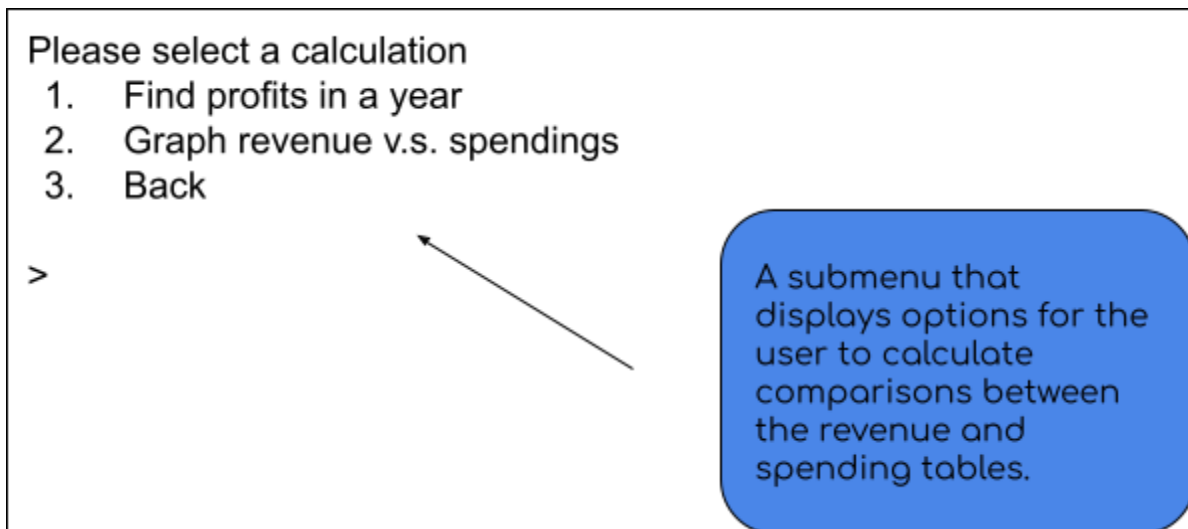


Figure 19: Submenu displays after selecting the revenue v.s spending (3) option

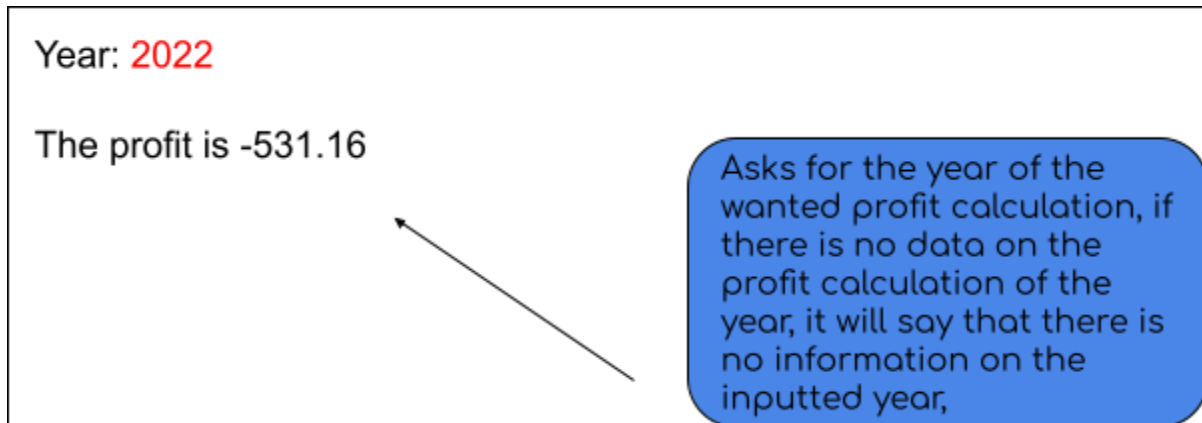


Figure 20: Finds the profits in a single year if option 1 is selected in the revenue v.s spending submenu

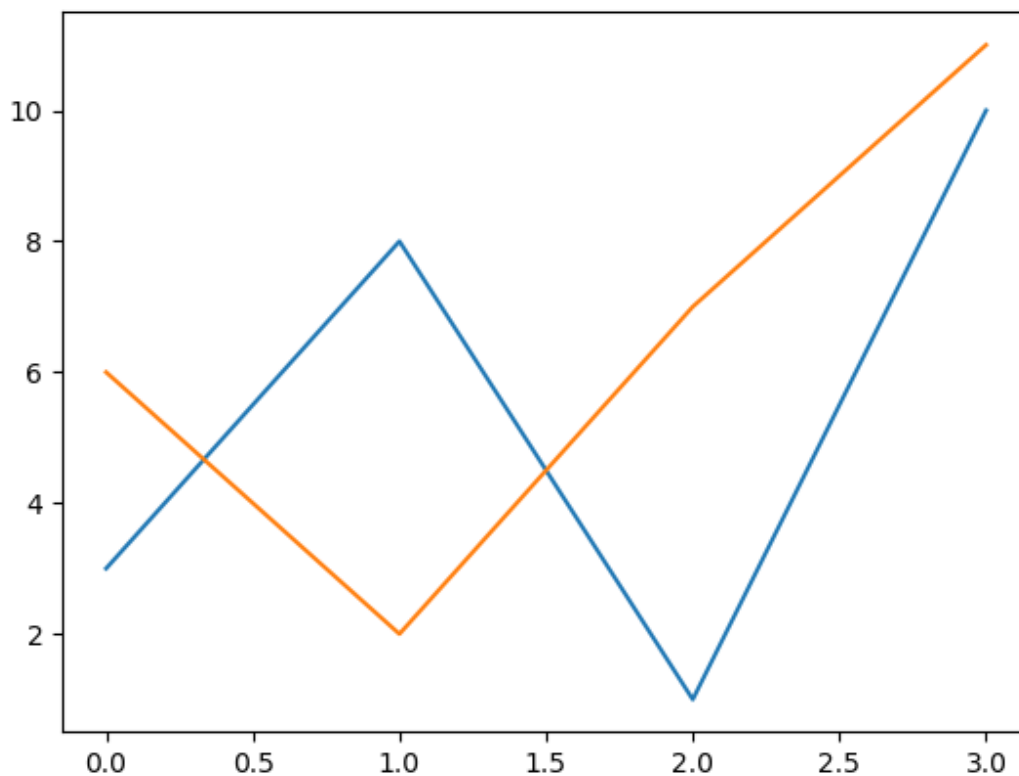


Figure 21: Displays a graph of both spendings and revenue in comparison if option 2 is chosen in the revenue v.s spending submenu

## **Test Plan**

Test Action	Plan to Test Action
Program will display a working login screen	<ul style="list-style-type: none"><li>- Run the program for the first time, set a password, and then try the login feature</li><li>- Try other random wrong password to check if it denies the inputs</li></ul>
Program will be able to update a new password	<ul style="list-style-type: none"><li>- Input the wrong password for the "current password", make sure that it doesn't let the user set a new password, since the input of the current password was wrong.</li><li>- Try implementing a new password, and then rerun the program to see if it updated</li></ul>
Program will display a menu for the choice of revenue, spendings, and revenue vs spendings	<ul style="list-style-type: none"><li>- Run the program, check if the main menu of options are arranged properly and in desired fashion</li><li>- Go into submenus and then head back, check if main menu of options are proper</li></ul>
Program is able to enter a submenu after selecting one the choices from the main menu	<ul style="list-style-type: none"><li>- Run the program and login, from there, choose one of the options, and see if the submenu displayed is proper and in desired condition</li></ul>
Program is able to add new data to each respective table	<ul style="list-style-type: none"><li>- Select the add info in a submenu, and input all information, see if all information is properly added into the DB</li><li>- Do not input all info into the inputs. Program should ask again for proper inputs</li><li>- Make sure program tells which inputs were misinputted</li></ul>
Program is able to delete data to each respective table	<ul style="list-style-type: none"><li>- Input a ID that does not exist in the tables, program should ask again for a proper input</li><li>- Input an existing data, see if the</li></ul>

	program deletes that data.
Program is able to edit/update data to each respective table	<ul style="list-style-type: none"> <li>- Select the update in the submenus, and input new information, check in DB if properly updated</li> <li>- Input none, and see if program takes the current data as the updated data</li> </ul>
Program is able to query & search for existing data	<ul style="list-style-type: none"> <li>- Input a year that does not exist, program should say that database does not exist</li> <li>- Input a year that does exist, program should display a table using tabulate that shows all the years existing in that particular table.</li> </ul>
Program is able to state "Data is non-existent" if inputted data is not available or does not exist within the relational database	<ul style="list-style-type: none"> <li>- Input random values into the programs, if it does not exist, program should inform the user that it does not exist</li> </ul>
Program is able to error check non-applicable inputs, and redirect the user again for a proper input	<ul style="list-style-type: none"> <li>- Input non-applicable values into the program, check if the program checks for the validity of the values.</li> <li>- Check if the program informs the user of their mistake</li> </ul>
Program is able restart automatically and loop continuously without the user having to run the program multiple times	<ul style="list-style-type: none"> <li>- Repeat the program by performing random selected tasks from sub-menus</li> <li>- Check if the program is in a while loop</li> </ul>
Program is able to backtrack from the submenu to the main menu	<ul style="list-style-type: none"> <li>- Check if the "back" option in the submenu works</li> </ul>
Program is able to exit	<ul style="list-style-type: none"> <li>- Check if the option "Exit" in the main menu and login screen works</li> </ul>

Word Count: 46