

Criterion B: Design

Table of Contents

1. Rough sketch of the products GUI	2
2. Functions.....	6
3. FlowCharts.....	7
3.1 FlowChart of Functions related to Trackerdata database.....	9
4. UML diagrams.....	12
5. SQLite Implementation.....	15
5.1 SQLite Queries.....	15
5.1.1 SELECT.....	15
5.1.2 CREATE.....	15
5.1.3 INSERT.....	15
5.1.4 UPDATE.....	15
5.1.5 DELETE.....	15
5.2 SQLite Tables.....	15
5.2.1 Trackertable in Trackerdata database.....	15
5.2.2 tasks table in tasks database.....	16
5.2.3 sqlite_sequence table in tasks database.....	16
6. Pseudocodes.....	17
6.1 Pseudocode for visualize() function.....	17
6.1.1 def visualize_pie_chart().....	17
6.1.2 def visualize_bar_graph().....	17

6.2 Pseudocode for totalsum() function.....	18
6.3 Pseudocode for printdata() function.....	18
6.4 Pseudocode for complete_task() function.....	19
7. Test Plan.....	20
8. References.....	22

1.Rough sketch of the products GUI

LOGIN	
Username	Admin
Password	*****
Exit Window	Login

Sketch 1: Loginfrontpage

Expense Tracker
Visualize
Buying List
Close

Sketch 2: Mywindow

EXPENSE TRACKER			
Name	<input type="text"/>		
Date	<input type="text"/>		
Cost(\$)	<input type="text"/>		
	<input type="button" value="Add"/>		
	<input type="button" value="Search"/>		
	<input type="button" value="Delete selected"/>	<input type="button" value="Delete all"/>	
	<input type="button" value="View all"/>		
	<input type="button" value="update"/>		
	<input type="button" value="Back"/>	<input type="button" value="Report"/>	<input type="button" value="close"/>

Sketch 3: EXPENSE

Expense Report	
<input type="button" value="Total Spent"/>	
<input type="button" value="Back"/>	

Sketch 4: Report

TOTAL SPENT \$	
<input type="button" value="Close"/>	

Sketch 5: Amount Spent

1
1 apple 12-02-2023 12
2 banana 12-01-2023 119
3 car 12-05-2023 10000
4 Pen 1-07-2023 10
5 book 1-20-2023 50
6 bag 2-21-2023 80

Sketch 6: Print Data

EXPENSE TRACKER	
Piechart	BarGraph
<input type="button" value="close"/>	
<input type="button" value="Back"/>	

Sketch 7: Visualize

<input type="button" value="Add"/>	
2 Pen 5 Apple	
<input type="button" value="Delete"/>	<input type="button" value="Complete"/>
<input type="button" value="close"/>	
<input type="button" value="Back"/>	

Sketch 8: Buying List

2. Functions

- loginpage(): contains code for login page logic
- trackerwindow(): contains code for tracker window logic, including database management, data input, data view, data search, data deletion, and data visualization
- connectdata(): creates the Trackertable SQLite database if it doesn't exist
- datainput(itemname, date, cost): inserts data into the Trackertable database
- view(): retrieves all data from the Trackertable database
- search(): searches the Trackertable database based on provided parameters
- delete(): deletes a row from the Trackertable database based on the selected item/number
- deletealldata(): deletes all data from the Trackertable database
- visualize(): generates a pie chart and bar graph visualizing the data in the Trackertable database
- insertitems(): inserts data into the Trackertable database, called when "Add" button is clicked
- totalsum(): calculates the total amount spent and shows the result in a new window, called when "Total spent" button is clicked
- print(): prints the data from the listbox which is in the Trackerdata database
- report(): opens a new window to run print() and totalsum()

3. FlowChart

Chart 1 loginpage()

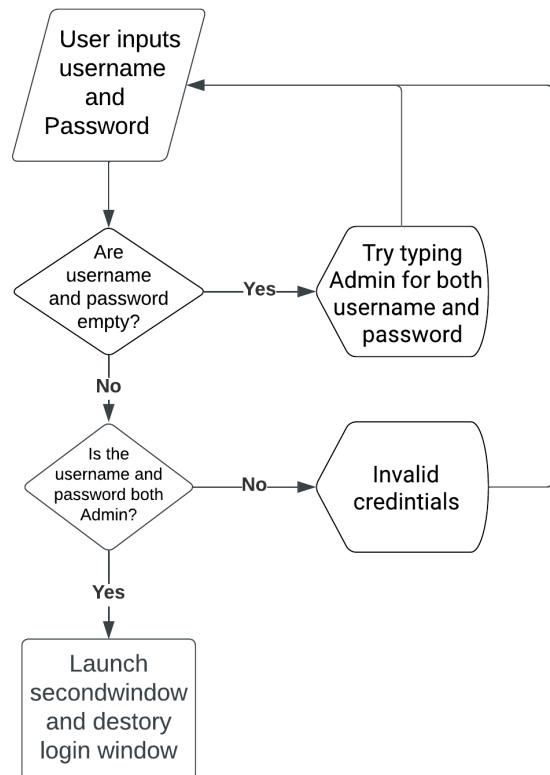


Chart 2 overview of secondwindow()

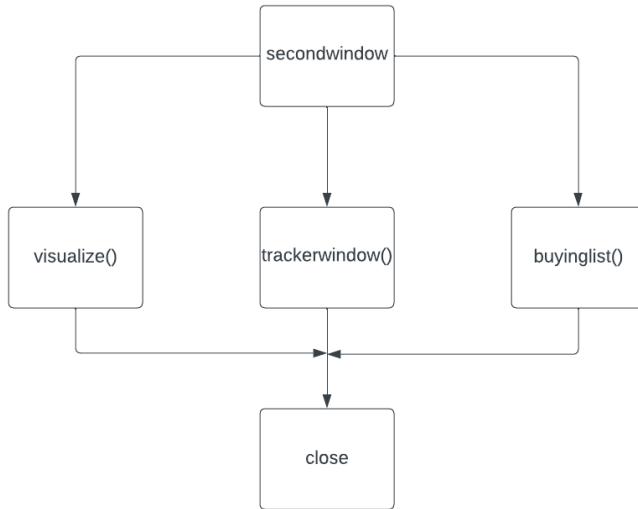
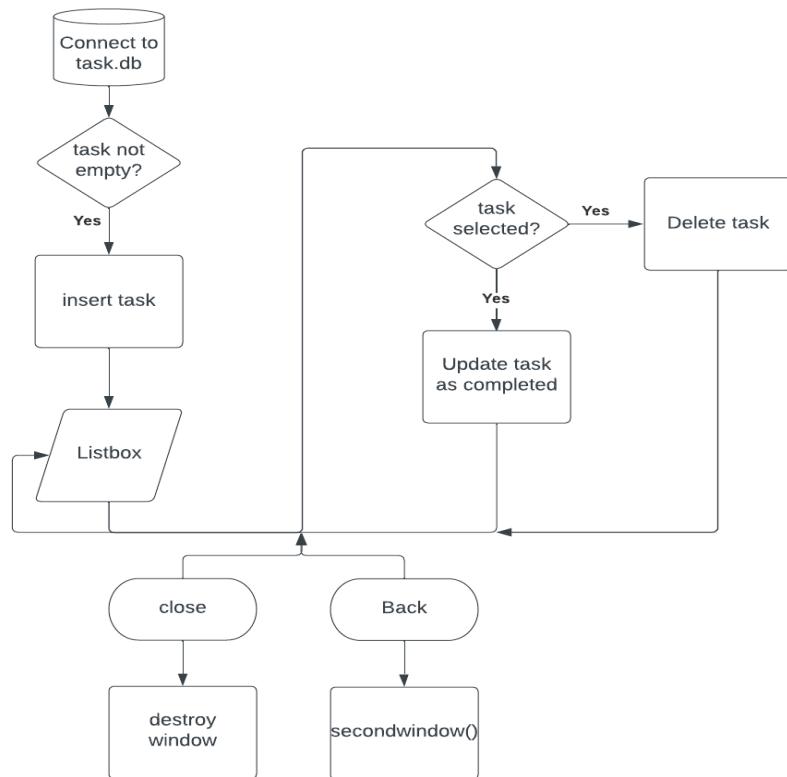


Chart 3 buyinglist()



3.1 FlowChart of Functions related to Trackerdata database

Chart 4 insertitem()

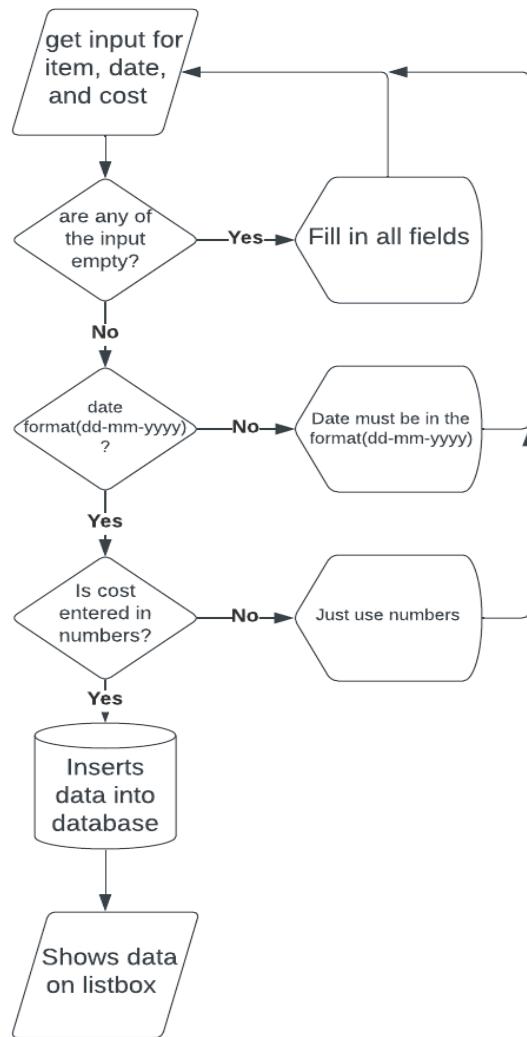


Chart 5 update()

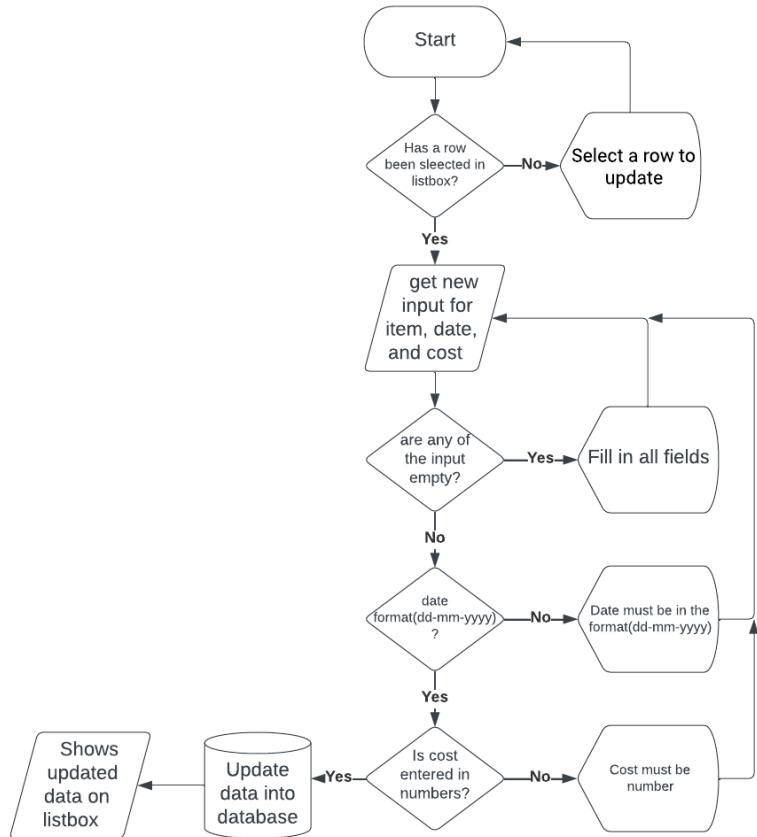


Chart 6 overview of report()

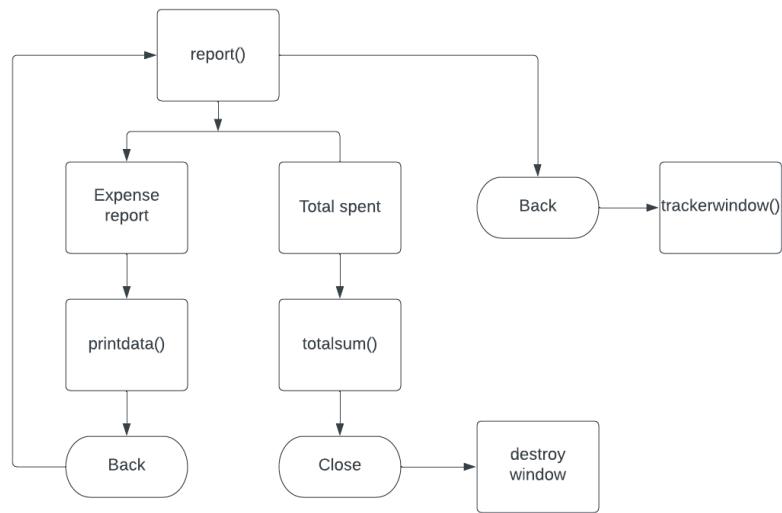
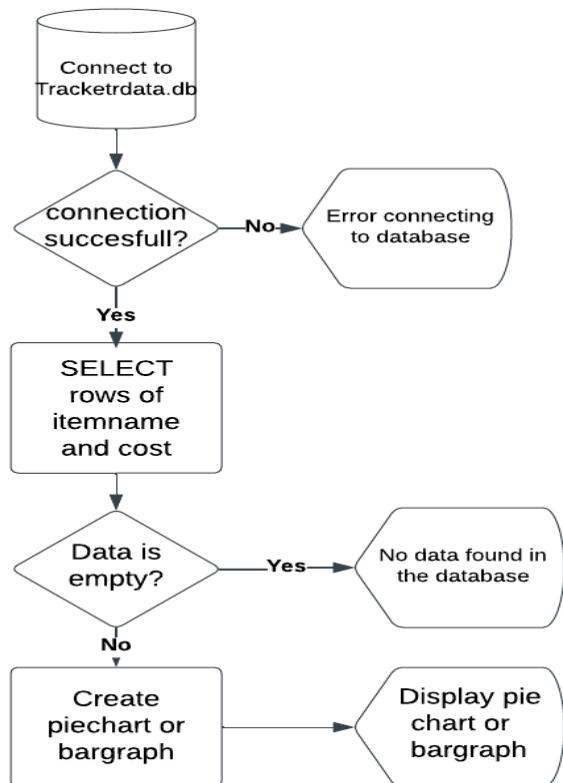
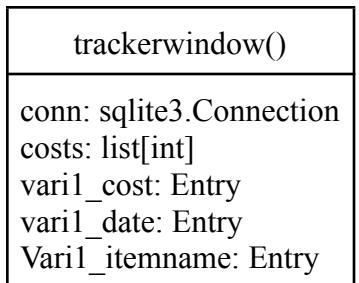
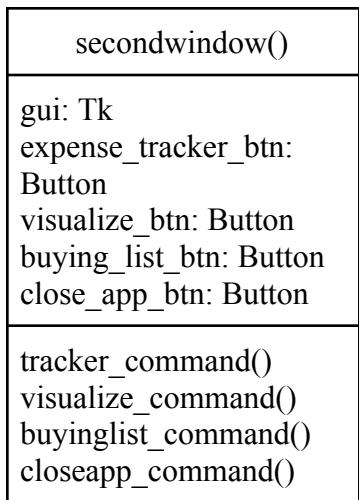
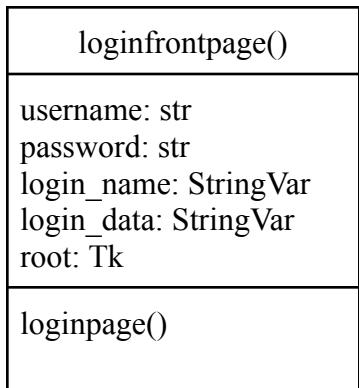


Chart 7 visualizing using pie chart and bar graph



4. UML diagrams

The UML diagrams presented below will show the functions and their relationships, as well as the methods and their parameters, of the program.



```
exp_itemname: Entry  
labels: list[str]  
list1: Listbox  
gui: Tk  
data_listbox  
print_button  
back_button
```

```
connectdata()  
datainput()  
view()  
search()  
delete()  
deletealldata()  
insertitems()  
totalsum()  
update()  
report()  
printdata()  
close()
```

```
visualize()
```

```
gui: tk.Tk()  
pie_chart_button:  
tk.Button()  
bar_graph_button:  
tk.Button()  
back_button: tk.Button()  
close_button: tk.Button()
```

```
visualize_pie_chart()  
visualize_bar_graph()
```

```
buyinglist()
```

```
task_entry: Entry
```

```
add_button: Button  
task_listbox: Listbox  
delete_button: Button  
complete_button: Button  
back_button: Button  
close_button: Button  
conn: Connection
```

```
add_task()  
delete_task()  
complete_task()
```

5. SQLite Implementation

5.1 SQLite Queries

5.1.1 SELECT:

SELECT queries will be used to retrieve the data from the database to later add them to the listbox, to visualize, to find total sum and to view all.

5.1.2 CREATE:

CREATE queries will be used to create tables for two database names tasks and Trackerdata.

5.1.3 INSERT:

INSERT queries will be used to insert data into the table of the database tasks and Trackerdata.

5.1.4 UPDATE:

UPDATE queries will be used to update data to mark as completed in database tasks and to update the data of selected items in database Trackerdata.

5.1.5 DELETE:

DELETE queries will be used to delete selected and all items from database and listbox from both database tasks and Trackerdata.

5.2 SQLite Tables

5.2.1 Trackertable in Trackerdata database:

Name	Type	Descriptions
Number(Primary Key)	INTEGER	Used to store number for the item which is also a primary key
itemname	TEXT	Used to store the name of the item
date	TEXT	Used to store the date of the item
cost	TEXT	Used to store the cost of the item

5.2.2 tasks table in tasks database

Name	Type	Descriptions
ID(Primary Key)(AUTINCREMENT)	INTEGER	Used to store ID for the tasks which is autoincremented and also a primary key
task	TEXT	Used to store tasks/list of items user wants to buy
completed	INTEGER	Used to mark buying task as completed

5.2.3 sqlite_sequence table in tasks database

name	tasks
seq	AUTOINCREMENT

AUTOINCREMENT: The term AUTOINCREMENT is used to automatically increase the value of a field in a table. Only INTEGER fields can be used with the keyword AUTOINCREMENT (tutorialspoint, n.d.). SQLite will automatically assign a unique integer value to the seq column.

6. Pseudocodes:

6.1 Pseudocode for visualize() function

6.1.1 def visualize_pie_chart():

 connect to the database
 get data from database

 if no data:
 show error message and return
 else:

 create empty lists for products and costs
 for each row in the data:
 append product name to products list
 append cost to costs list

 create pie chart with costs and products
 show the chart

6.1.2 def visualize_bar_graph():

 connect to the database
 get data from database
 if no data:

 show error message and return
 else:

 create empty lists for products and costs
 for each row in the data:
 append product name to products list
 append cost to costs list

 create bar graph with products and costs
 label the axes and title the graph
 show the graph

6.2: Pseudocode for totalsum() function

```
def totalsum():
    connect to database
    execute query to get sum of costs
    get the total sum from the query results
    create a label with the total sum and add it to the GUI window
    create a "Close" button to close the GUI window
    return the total sum
```

6.3 Pseudocode for printdata() function

```
def printdata():
    create a listbox and pack it into the window
    get data from database
    for each row in the data:
        add row to the listbox
    create button that calls print_to_printer()
```

```
def print_to_printer():
    create a temporary file
    for each item in the listbox:
        write item to temporary file
    close the temporary file
    if platform is Windows:
```

```

Print data
else:
    display an error message

```

6.4 Pseudocode for complete_task() function

```

def complete_task():

    get selected item from task_listbox
    if a task is selected:

        update tasks table with completed=1 for selected task
        commit the changes to the database

        change the color of the selected item to gray.

```

7. Test Plan

Action to Test	Success Criteria Tested	Method
Logging in Test	1	-Check if the user is only allowed to enter the application if the username== “Admin” and password== “Admin”
Database connectivity Test	3,4,5,6,10	-Check whether a connection has been established between the program and the database. -Check that functions like add, delete, search, and update are functioning and the database tables

		are updated appropriately also it is updated in the program
Window functionality Test	2,9,10,7	-Check if user is taken into the appropriate window with appropriate functions and button
Error Test	11	-Check if a message is displayed to the user if any type of error has occurred using data validation and try and except statements
Viewing item Test	6	-Check if the inserted item can be viewed in a listbox.
Printing Test	8	-Check if the user can print their expenses and view them before printing
Visualize Test	9	-Check if the user can see their expenses visually on a pie chart and bar graph both.
Buying list function Test	10	-Check if the user can mark their buying list as completed , add item and delete item.

8. References

tutorialspoint. (n.d.). *SQLite - AUTOINCREMENT*. Tutorialspoint. Retrieved March 3, 2023, from https://www.tutorialspoint.com/sqlite/sqlite_using_autoincrement.htm