Criterion B: Design

Table of Contents

Initial sketch of UI Design

The following images are the initial sketches of different screens of the application, which have been mailed to my client Ms. Aarya Timilsina and are approved by her as well[1].

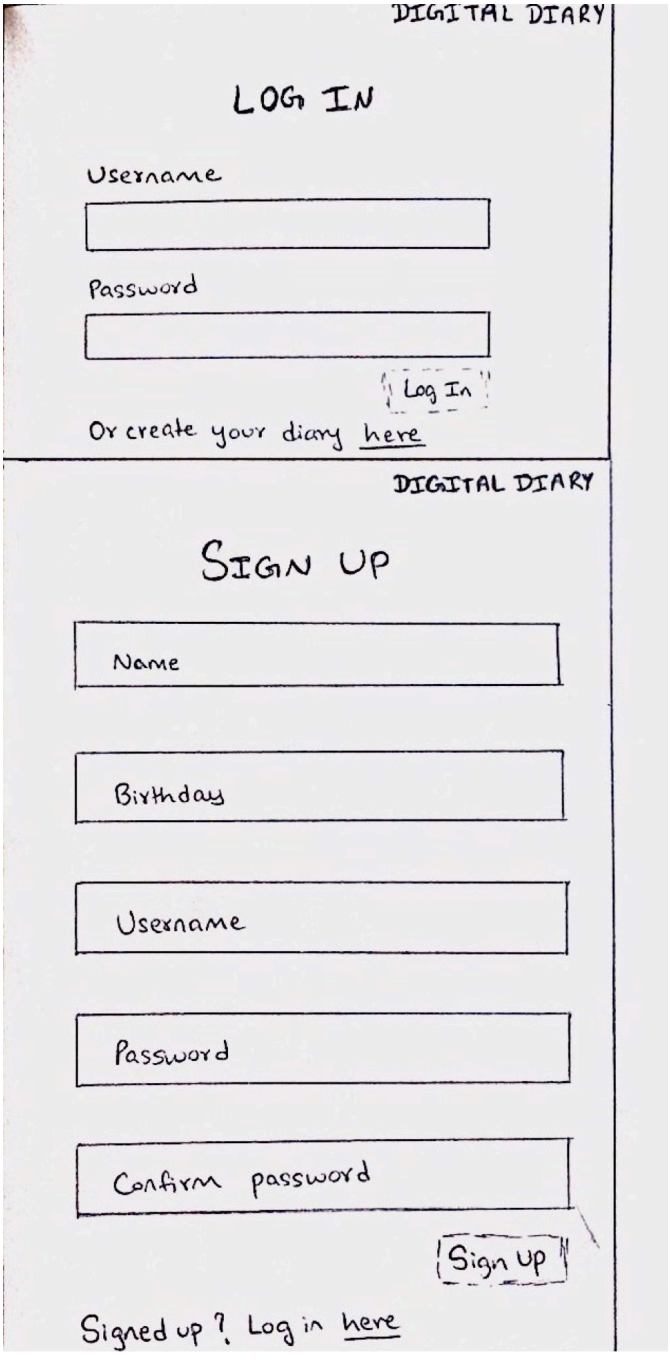


Figure 1: Log In and Sign Up pages

[1] See Appendix A.4 for reference

Figure 2: The main page and the "Entries" section

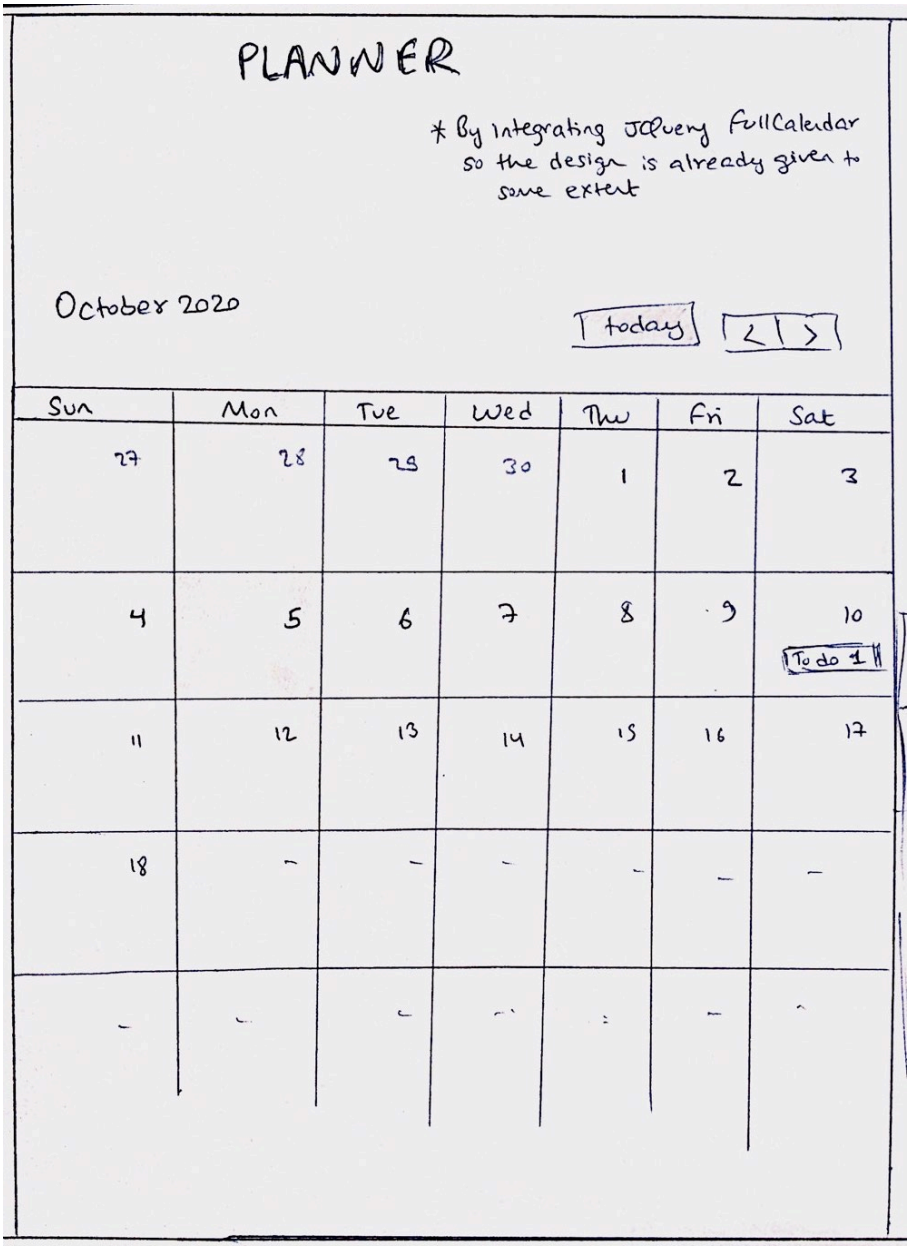


Figure 3: “Planner” section

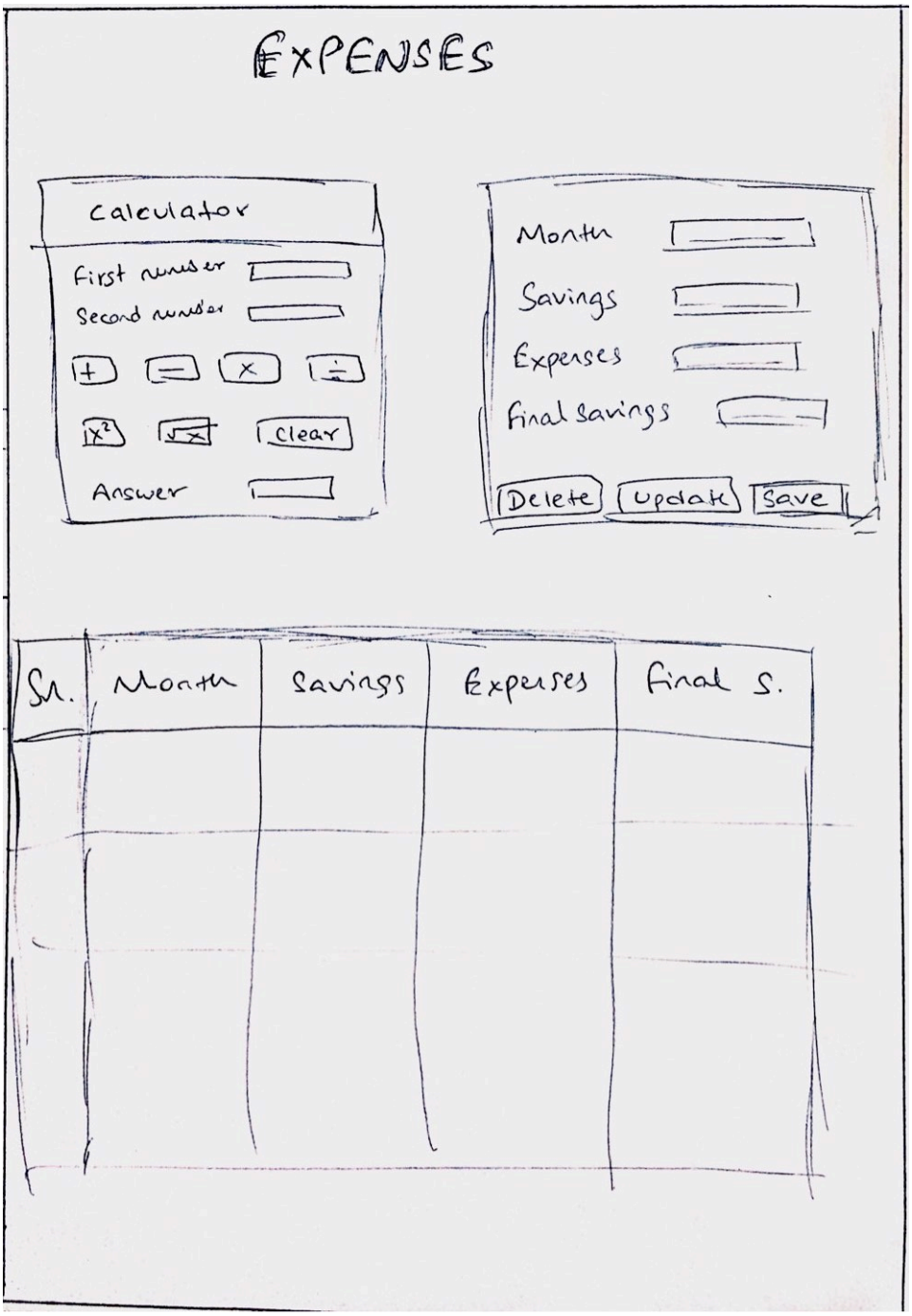


Figure 4: “Expenses” section

Data dictionary

The following tables display the data dictionary for the databases in the web-based application. I will be using MAMP localhost server to access phpMyAdmin of its database. All the tables present the field name, description, data type, default values, validation rules as well as the 'extra' column dedicated to holding any other information that has not been mentioned.

Table 1: User

This table will store data about the user for the purpose of logging in and initially signing up in order to access further records in the program.

Field Name	**Data Type**	**Description**	**Default Value**	**Validation Rule**	**Extra**
Id	INTEGER	Contains the auto incremented serial numbers	0	Not NULL	AUTO_INCREMENT Primary Key
Name	VARCHAR (255)	The full name of the user	<empty>	Not NULL	
Birthdate	DATE	The date of birth of the user	<empty>	Not NULL	
Username	VARCHAR (255)	Contains a unique username	<empty>	Not NULL	
Password	VARCHAR (255)	The password for the user	<empty>	Not NULL	

Table 2: Entries

This table will store the data about several journal entries done by the user. It will include the serial number, title, image, diary entry and date of an entry.

Field Name	Data Type	Description	Default Value	Validation Rule	Extra
Id	INTEGER	Contains the auto incremented serial numbers	0	Not NULL	AUTO_INCREMENT Primary Key
Title	VARCHAR (255)	Contains the different titles of the entries	<empty>	Not NULL	
Date	DATETIME	The date of entry	CURRENT_TIMESTAMP	Not NULL	
Image	VARCHAR (255)	Contains the name of the image uploaded by the user	<empty>	Not NULL	
Body	TEXT	Contains the body of the diary entry	<empty>	Not NULL	

Table 3: Events

This table will be containing elements such as the date and time of when the event is supposed to start and end, title, and a brief description of the plans made by the user.

Field Name	Data Type	Description	Default Value	Validation Rule	Extra
Id	INTEGER	Contains the auto incremented serial numbers	0	Not NULL	AUTO_INCREMENT Primary Key
Start	DATETIME	The date and time of when the event is supposed to start	<empty>	Not NULL	

End	DATETIME	The date and time of when the event is supposed to end	<empty>	Not NULL	
Title	VARCHAR (255)	The title of the event	<empty>	Not NULL	

Table 4: Expenses

This table will be holding data about the monthly expenses of the user that includes inputting the month, year, initial saving, expenses and final savings.

Field Name	**Data Type**	**Description**	**Default Value**	**Validation Rule**	**Extra**
Id	INTEGER	Contains the auto incremented serial numbers	0	Not NULL	AUTO_INCREMENT Primary Key
Month	VARCHAR	Contains the month that is considered	<empty>	Not NULL	
Year	INTEGER (4)	Contains the year that is considered	<empty>	Not NULL	
Savings	INTEGER	The initial savings of the user	<empty>	Not NULL	
Expenses	INTEGER	The expenses done by the user in the particular month	<empty>	Not NULL	
FinalSaving	INTEGER	The final saving of the user after the expenses are subtracted from the	<empty>	Not NULL	

		initial savings.			

Table 5: Help

This table will be holding the information that will be provided in the help section.

Field Name	Data Type	Description	Default Value	Validation Rule	Extra
Id	INTEGER	Contains the auto incremented serial numbers	0	Not NULL	AUTO_INCREMENT Primary Key
Title	VARCHAR (255)	Contains the title that summarizes the information	<empty>	Not NULL	
Info	TEXT	Contains the information/ help on how to use different functionalities	<empty>	Not NULL	

SQL Queries

The following SQL queries will be used while developing the Digital Diary System.

INSERT: This SQL query will be used to store the username, password, title, expenses, etc. in the database.
Example – INSERT INTO Entries (Title, Body, Image) VALUES (?, ?, ?);

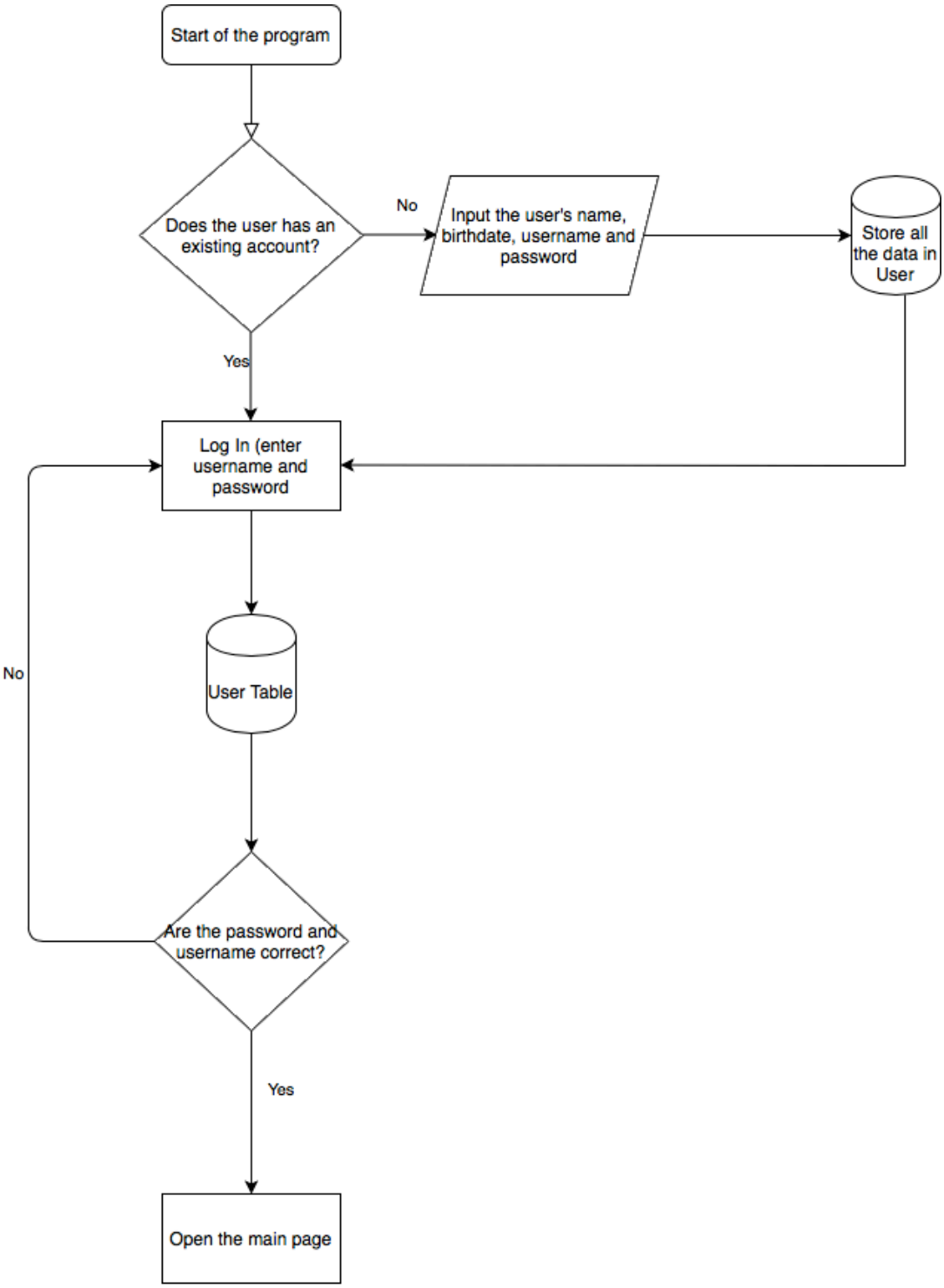
UPDATE: This SQL query will be used to modify and store the username, password, title, expenses, etc. in the database into an existing record.
Example – UPDATE Expenses SET Month=?, Year=?, Savings=?, Expenses=?, FinalSavings=? WHERE Id=?;

DELETE: This SQL query will be used to delete an existing record from the database.
Example – DELETE FROM Events WHERE Id=?;

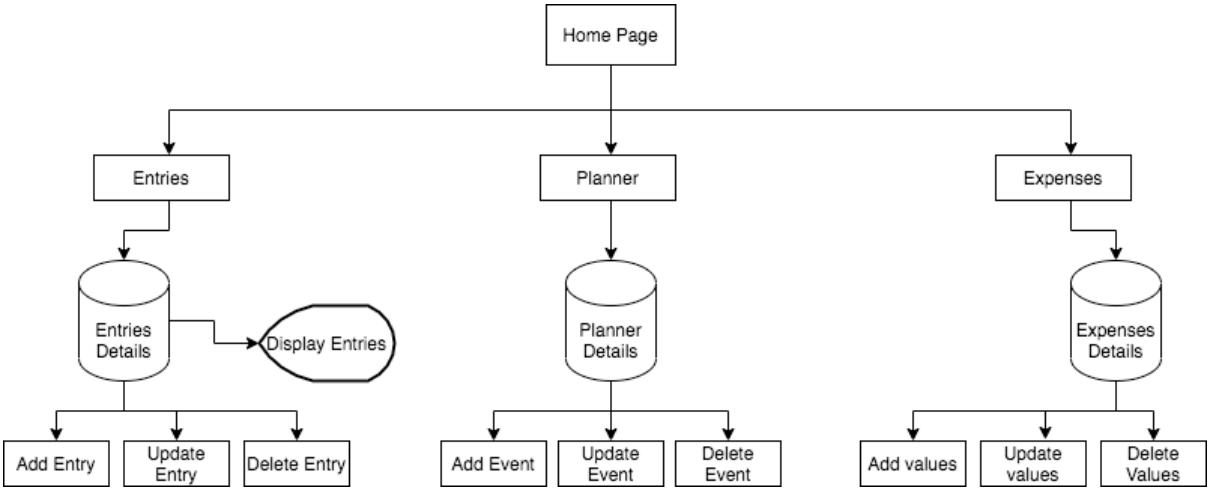
SELECT: This SQL query will be used to retrieve and display the data from the database in the GUI.
Example – SELECT Title, Info FROM Help;

Flow-chart

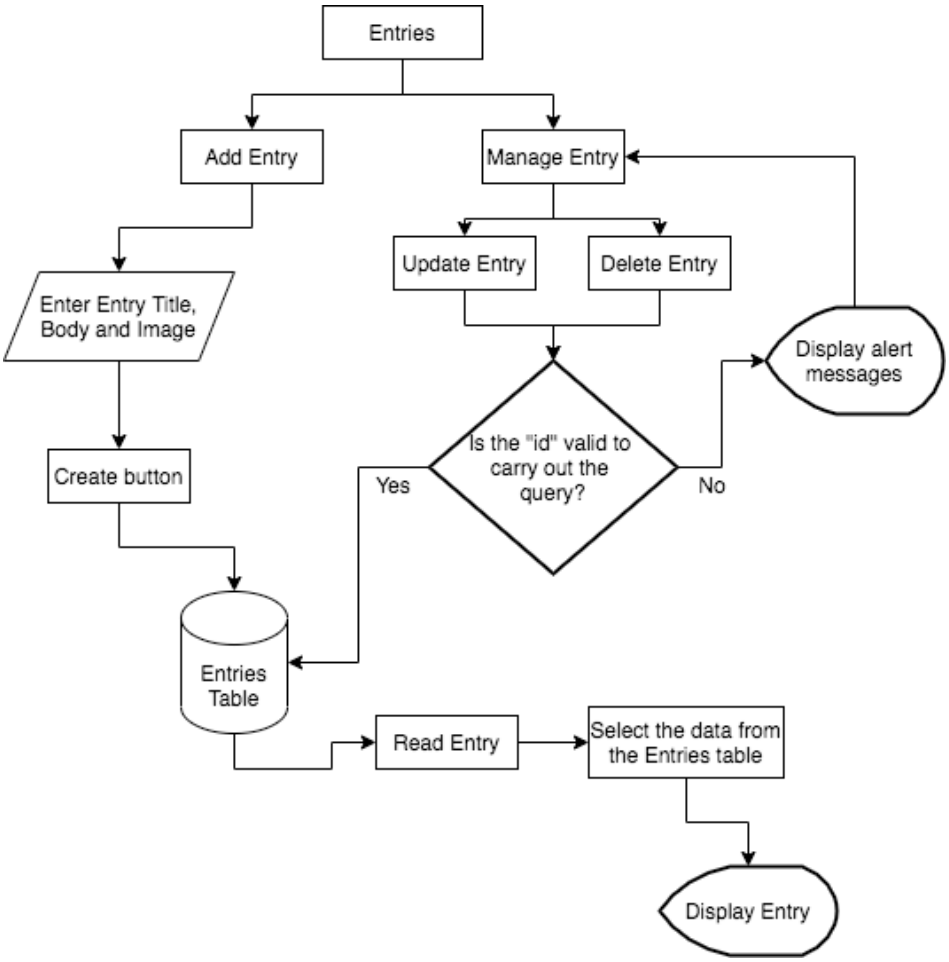
The flowchart here shows the process of either logging in or signing up for further access.

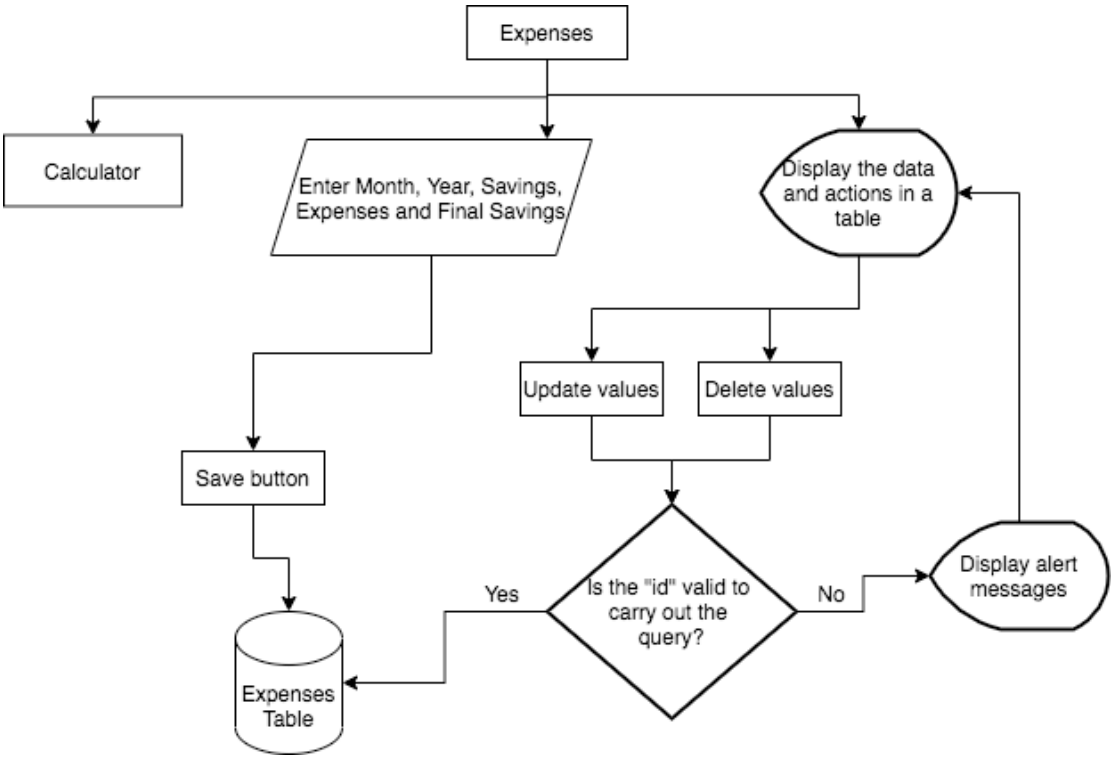


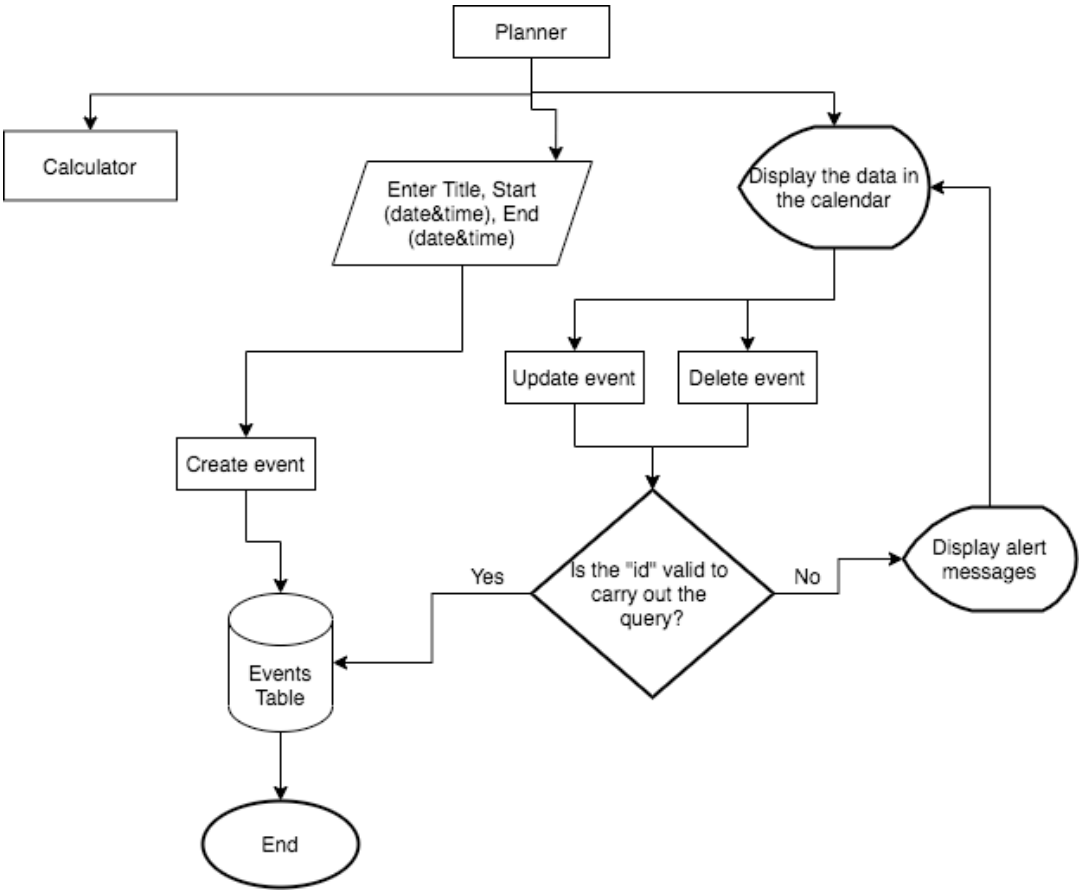
The following flowchart displays the general structure of my application.



Now, the flowcharts represent the structure of CRUD functionality in my all three sections.




Expenses
Calculator
Enter Month, Year, Savings, Expenses and Final Savings
Display the data and actions in a table
Update values
Delete values
Save button
Is the "id" valid to carry out the query?
Yes
No
Display alert messages
Expenses Table


Planner
Calculator
Enter Title, Start (date&time), End (date&time)
Display the data in the calendar
Update event
Delete event
Create event
Is the "id" valid to carry out the query?
Yes
No
Display alert messages
Events Table
End

Library

FullCalendar, the JavaScript Library, will be used to aid in the GUI as well as managing events in the Planner section.

Test plan

In order to test my program, I have listed the plans as shown in the following table.

Test Type	Description
Unit Testing	Test whether or not each PHP, CSS and JavaScript files and their respective elements are working. Ensures Success Criteria 1, 3, 6 and 7 are met
Database Testing	Test whether the database is connected to the PHP or not. Further test if using the 'add', 'update' and 'delete' functionalities causes the data in the database to change as well. Ensures Success Criteria 3, 4 and 8 are met
Calculation Testing	Check if the calculations done by the calculator in the program are correct. Ensures Success Criteria 7 is met
Functionality Testing	Test the functionality of buttons such as 'log in', 'sign in', 'search', 'save', 'update', 'view' and 'delete'. Ensures Success Criteria 3, 4 and 5 are met Check for error handling in the instances of confirming the password, username, etc. and whenever a user provides invalid input or abnormal and extreme data.
Dry run testing Testing	Manually testing for logical errors via a pen and paper to ensure that the program as a whole or parts of the program is working as intended.
Integration Testing	Check if the program works efficiently and smoothly when the units are integrated together.

	Test whether or not the transition from one page to another page is as intended and smooth, causing it to be as user friendly as possible. Ensures Success Criteria 2 is met
User acceptance Testing	Give the application prototype to the client to test if all of the requirements are met. Receive feedback from the client regarding the prototype and how it can be improved further. Ensures Success Criteria 2, 9 and 10 are met

Extensibility

In this web-application, a log in page and sign up page will be developed. However, it only serves the purpose of security to my client, as it is a single user application. In the future, this application can expand to being a multi-user application.