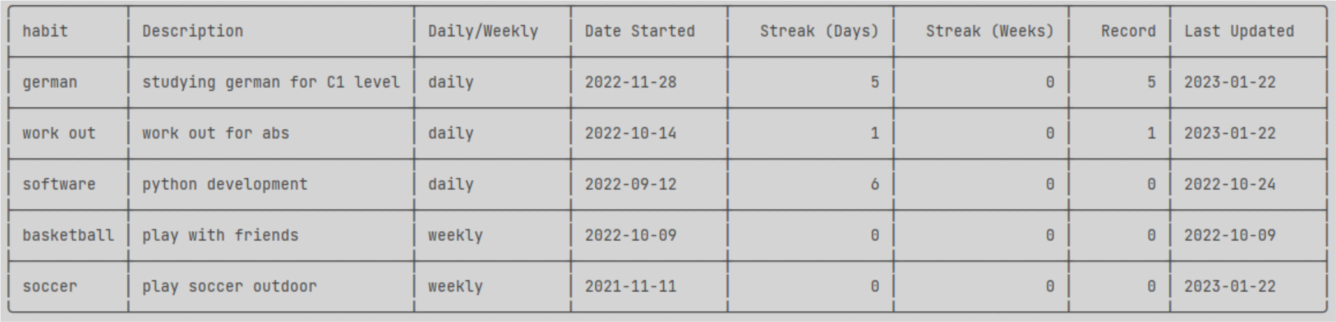
Conception Phase

In this project, I developed a habit tracker app with a user-friendly interface on which user can add, delete, track and see current progress of their habits.

1. Main Structure

I chose to display a DataFrame view in order for the user to see everything at a glance on start-up and various levels of the application. Below is how it looks:



habit	Description	Daily/Weekly	Date Started	Streak (Days)	Streak (Weeks)	Record	Last Updated
german	studying german for C1 level	daily	2022-11-28	5	0	5	2023-01-22
work out	work out for abs	daily	2022-10-14	1	0	1	2023-01-22
software	python development	daily	2022-09-12	6	0	0	2022-10-24
basketball	play with friends	weekly	2022-10-09	0	0	0	2022-10-09
soccer	play soccer outdoor	weekly	2021-11-11	0	0	0	2023-01-22

Figure 1 . Dataframe

In the DataFrame, I will use fields like habit name itself, its description, its type (weekly, daily), the date when it was first added, streak count, its highest number of consecutive check offs, and its last update. The table will always be updated whenever a function is triggered and be printed on screen again.

2. Selection of Data Saving Process

I selected SQL over Json in order for the project to be more professional and incorporate SQL commands learned from previous courses within the project.

3. Modules

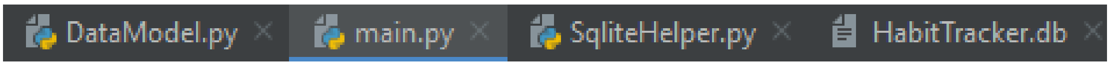


Figure 2. Files

I used two extra files which are imported into main.py called DataModel and SqliteHelper files. Most of the functions are defined in the SQL class from SqLiteHelper module , and called from main.py file. Some others are also defined under main.py with the name of App class.

HabitTracker.db is the database in which the habits and its attributes are stored.

4. Coding Structure

Code has been written in compliance with OOP structure of Python. Structural programming has also been used when necessary, but the main structure is OOP. I created classes and functions inside other files and called from main.py which is also used to start the program.

One main class of the app is in SQL Class. This includes analyzing functions (as shown in FlowChart), adding streak days and other SQL related functions like adding habits and deleting them from table, committing them to SQL server etc.

A relatively smaller portion of the app is in App class in which managing habits, doing a check off for a habit, and listing main menu functions are defined under. This is the first class that app calls.

The last class is Data. This class is about how to show the data, its order and format whose query is run from SQL table after a function is triggered.

When the app starts, it will start off by listing 5 predefined habits as DataFrame using tabulate plugin. Under it, mainmenu function will be called and user will see the options of:

1- Check Off Habit
2- Manage Habits
3- Analyze Habits
4- Exit

- Check off adds a streak to selected habit.
- Manage habits prompts for 2 other options as add and delete habit.
- Analyze habits brings other statistical data options from the Data table as summarized data.

5. Modules and Tools

I used tabulate, datetime, sqllite3 and pandas.

- Sqllite3 is for storing the data in a database table and doing all the management via this table. I used SQL commands with cursor object.
- Datetime is to keep track of weekly and daily habits and their streaks .
- Tabulate is to display the DataFrame in an organized and elegant way.
- Pandas is used to display habit list in the form of a table queried from SQL server.

6. FlowChart

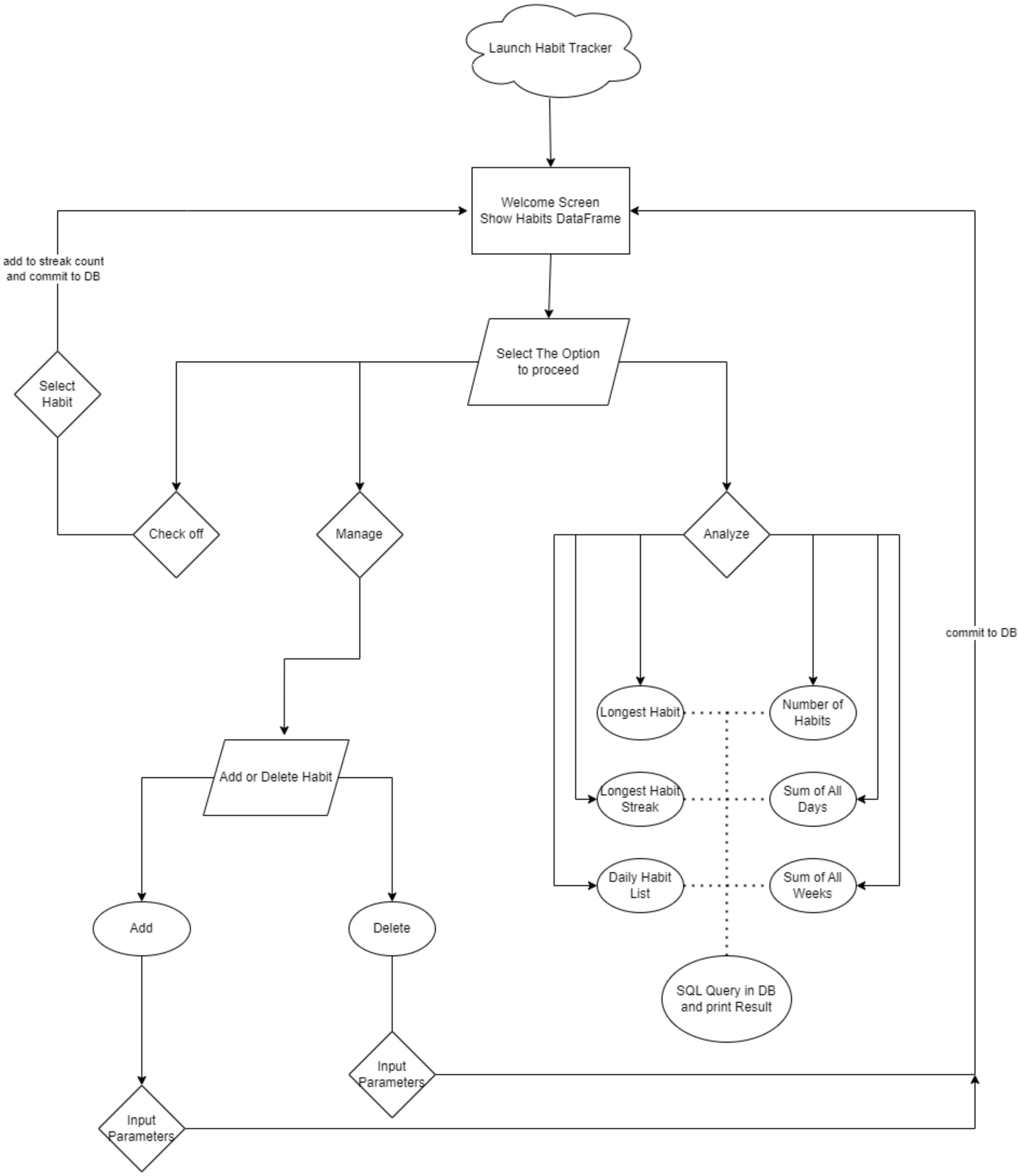


Figure 3. FlowChart of Habit Tracker Application