Conception Phase

Simeon Akpanudo

This document focuses on the implementation of a non-GUI Habit Tracker App that allows users to track and store their habit tracking progress, using the command line interface.

Core Modules & Tools

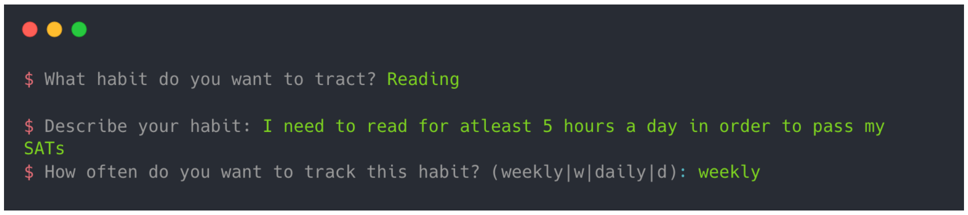
- Programming Language: Python.
- Data Store: SQL (SQLite package in python).
- Unit Test: Pytest.
- Pandas.

Process

This section describes the data storage method, data visualization, how users would interact with the application (UI/UX), code organization, how habits would be tracked and user interaction flow using flowcharts.

User Interaction and Experience Design

One of the pitfalls of non-GUI applications is the overall user experience, as users only use their keyboard while interacting with the application, this requirement counters a day-to-day user experience with other apps since most modern applications are built with GUIs.
This Application would use the command line interface but with easy-to-follow steps.



```
$ What habit do you want to tract? Reading

$ Describe your habit: I need to read for atleast 5 hours a day in order to pass my
SATs
$ How often do you want to track this habit? (weekly|w|daily|d): weekly
```

Fig. 1: *Command Line Interface of App.*

How Habits are tracked

A streak starts when a user marks a habit (daily or weekly) as completed. A weekly habit can only be marked as complete once in a 7-day period, while the daily habit can be marked as complete in a 24-hour period. If a user breaks a habit by not marking it as complete within the habit's periodicity, the current streak is returned to 0, and the historical record would be saved as the longest streak if it exceeds the previous recorded longest streak.

When a user marks a habit as complete for a specific period, a snapshot of the current datetime is saved in the data store to help verify the user's action, for example, if the snapshot is not in the interval of the habit's period (If a user tries to mark a daily habit twice in 24 hours for a daily habit, or twice for a weekly habit) or if the user has broken their current streak.

Saving the data

Data persistence is required to achieve the full functionality of the app. This persisted data would enable tracking and data analysis, for instance, getting the longest streaks or currently active habits. For this implementation, we would use SQL (Structured Query Language) relational database to manage the storing and retrieving of user habits. I have experience using SQL and it better simulates a real-world use case.

Code Organization

This app would be written in the Python programming language, using the Object-Oriented Programming (OOP) and Functional Programming (FP) paradigms. Files in the project are "Habit.py", "Tracker.py", "Analytics.py", "main.py" and "habits.sqlite".

Habit.py – contains the class for managing the habit entity or model. This model contains the attributes that makes up a model. i.e Name, Period, etc.

Tracker.py – contains a class that manages adding, deleting and ticking off habits.

Analytics.py – contains a class that performs simple analysis, such as getting the current streak of habit.

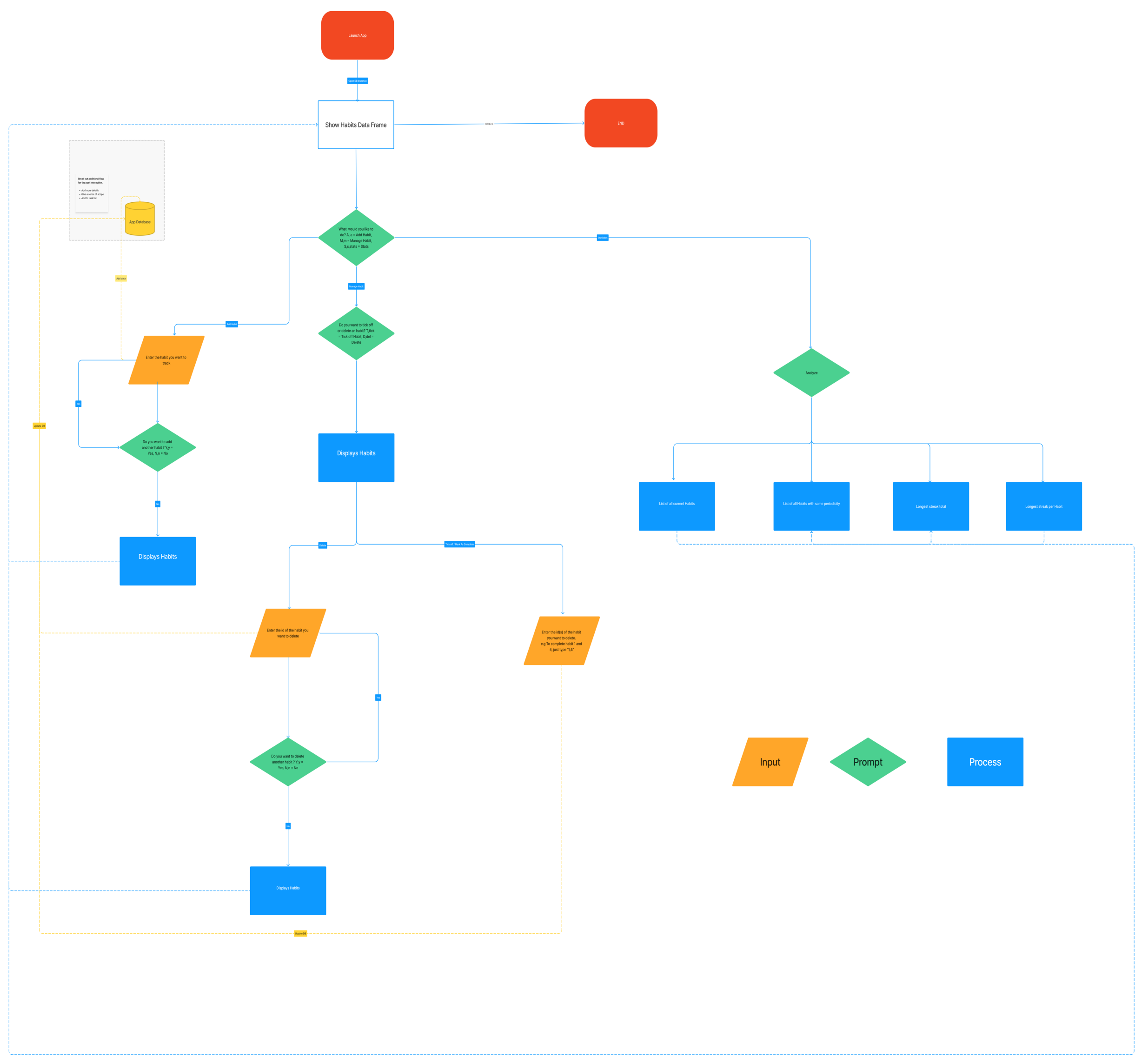
habits.sqlite – is the database file for habits storage.

main.py – contains the app entry-point.

Unit Testing

All classes and their methods within the application would be tested using the **pytest** library.

User Interaction Flow



Launch App
Open DB Instance
Show Habits Data Frame
CTRL C
END
Break out additional flow for the post interaction.
Add more details
Give a sense of scope
Add to task list
App Database
Add data
What would you like to do? A ,a = Add Habit, M,m = Manage Habit, S,s,stats = Stats
Statistics
Manage Habit
Add Habit
Enter the habit you want to track
Yes
Do you want to add another habit ? Y,y = Yes, N,n = No
No
Displays Habits
Update DB
Do you want to tick off or delete an habit? T,tick = Tick off Habit, D,del = Delete
Displays Habits
Delete
Tick off / Mark As Complete
Enter the id of the habit you want to delete
Yes
Do you want to delete another habit ? Y,y = Yes, N,n = No
No
Displays Habits
Enter the id(s) of the habit you want to delete. e.g To complete habit 1 and 4, just type "1,4"
Update DB
Analyze
List of all current Habits
List of all Habits with same periodicity
Longest streak total
Longest streak per Habit
Input
Prompt
Process

Fig. 2: *Flow Chart of Habit Tracker App.*

Possible updates.

- I could be adding a few libraries not listed in the “Core Modules & Tools” section to help with the process of developing this app.