

Habit Tracking App

Technical Concept Document

Executive Summary

The habit tracking app is supposed to allow users the expected range of functionalities, in an intuitive and easy to handle interface. The envisioned functionalities include the ability to create an unlimited amount of trackable habits, record an unlimited amount of habit events, analyze the user's historical data and delete or reset habit events as needed.

Additionally, the user's weekly and daily streaks shall be tracked and new records communicated. All in all, the app is supposed to be controlled via a clean and error resistant command line interface. Despite being entirely within the command line, where possible analytical insight shall be communicated graphically.

Technical structure envisioned

In order to keep a clean code base an object oriented approach has been chosen for the project. As such it will be necessary to split the code base into several categories, each of which shall be able to be referred to in any part of the program. The categories envisioned are as follows:

- MAIN
 - here the order of actions will be decided and the rest of the code controlled
- DATABASE
 - here all the upload, download and general interaction with the database will be handled
- HABITS class
 - in this class object the abstract habit functionalities will be built, with the aim to handle everything that needs to go into a realistic habit here
- ANALYSIS Module
 - here the analytical cross-reference will be taken care of
 - streak tracking and communication will be set up here
- TEST
 - In this section a reliable way to unit test will be established

- fake data will also be provided in order to allow for the full testing of all functionalities
- README
 - A short summary of the way the app works, as well as everything needed to successfully install the app will be communicated in this section
- REQUIREMENTS
 - In this .txt file all dependencies will be listed
 - The file will be directly referenced in order to provide a smooth install experience
- GRAPHICAL OUTPUT
 - Here all graphics will be produced

Libraries used & their purpose

All involved libraries will be commonly used open-source tools with clearly established track-records for reliability and efficiency. In this way the likelihood of security breaches, or significant loss of performance shall be minimized.

Wherever possible shortcuts can be found by using already existing libraries to cover desired functionalities they will be used in order to keep development time to a minimum. As long as the desired library fulfills the mentioned hallmarks of quality.

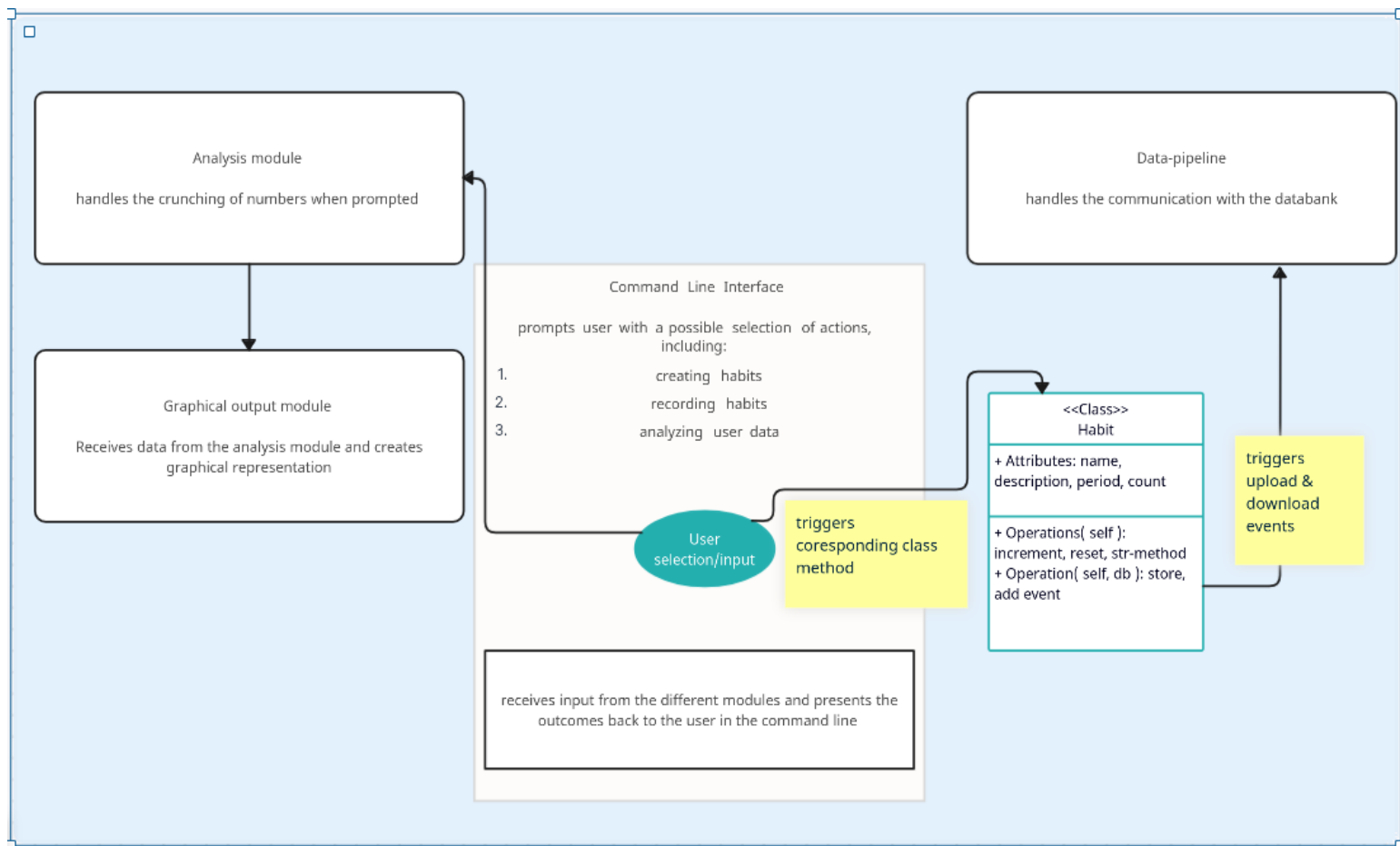
During the time of writing the following list of libraries will be involved:

- DATETIME
 - The role of this library will be to provide accurate and efficient dates for the recording of habits
 - The timedelta function of the datetime module will be used to track and evaluate different streaks
- QUESTIONARY
 - This library has been chosen for the simple and clean command line interface which can be quickly established using its choice, ask and text functionalities
- SQLITE3
 - To build a database without over-complicating the process this library has been chosen for all our data-pipeline needs
- MATPLOTLIB

- In order to keep user retention high and provide a product that is fun to use, this library will be used to build out the graphs providing a high-level overview over the different ongoing and past streaks

Structure

The interplay of the different elements can be represented as follows:



Conclusion

By limiting what needs to be freshly programmed due to the incorporation of several powerful libraries it should be possible for the technical team to create an application at low cost. By including a clean and intuitive interface, as well as graphical representation of the most relevant data the application should nonetheless be fun to use and contribute value to each user.