Conception phase

It is intended to develop an application for habit tracking with Python version 3.7 or later, well-documented installation and run instructions for use as command line interface, and permanent data storage for follow up during different sessions.

For the development of the application, five basic classes should be employed: “Habits”, “Time”, “User”, “Storage”, and “Analytics”. “Habits” class manage properties and functions related to habits such as name of the habit, state of the habit, etc.; “Time” class manage properties and functions related to date and time as tracking period, start of the tracking, finalization of the tracking, etc.; ”User” class will be employed to deal with identification information of the user like username, password, etc.; “Storage” class will contain properties and functions related to long term storage of the information to be used in different sessions; and “Analytics” module will allow the analysis of the habits of the users such as: currently tracked habits, all habits with the same periodicity, longest run streak of all defined habits, and return the longest run streak of a given habit. Figure 1 shows an UML graph of the basic classes.

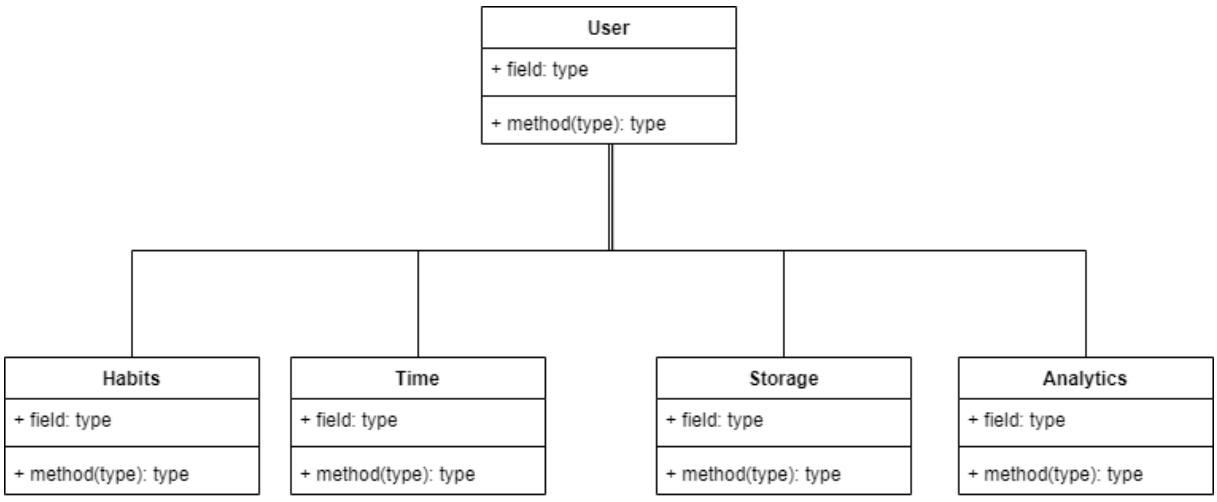

Figure 1. UML graph of the basic four classes required for the development of the habit tracking application. Properties and methods to be assigned.

Long-term information storage will be accomplished by using PostgreSQL as database management system. The DB is expected to have five entities with record coming from the beforehand described classes. For this purpose, the python module “psycopg2” will be required.

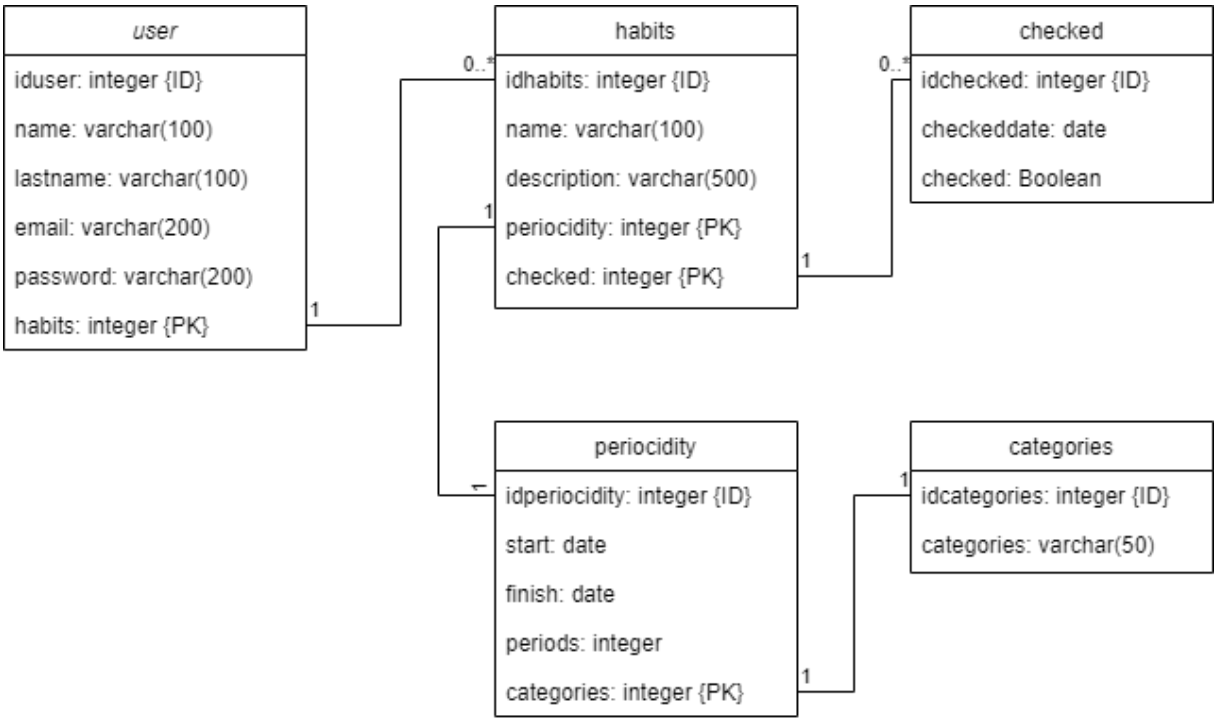

Figure 2. Entity relational model for the project.

Basic statistical analysis could be pursued by using standard python modules such as “pandas”, “NumPy”, and/or “scikit-learn” depending on the complexity or simply through the database management system in the “Analytics” module.

In order to create an API adequate for the use of the tracking habit application it will be used “fire” to convert any python component into a command line interface in just a call.

A view of the general working mechanisms of the application can be seen under in Figure 3

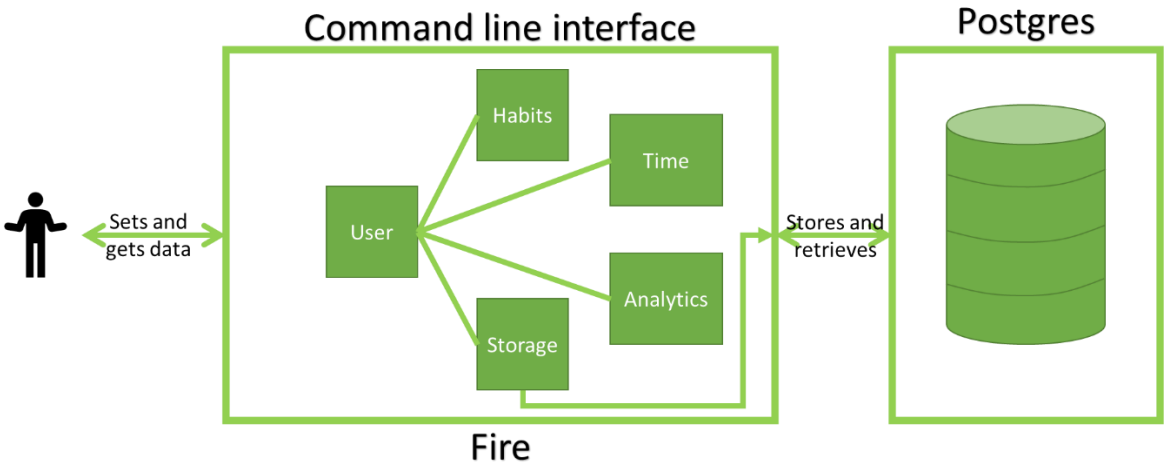

Figure 3. Overview of the tracking habits application.