**OBJECT ORIENTED AND FUNCTIONAL PROGRAMMING PYTHON**

**Student: Hugo albuquerque cosme da SILVA**

**ID:92126125**

**1 – Conception Phase**

The goal of this project is to build a simple python backend for a habit tracking app using Object oriented and functional programming paradigms. This section will go into details about how the functionalities of the app are intended to be implemented. A top down approach will be used to describe the parts of the project, which means The user interface will be described first and the inner parts of the application will come later.

User Interface

The application is going to the called ***habit\_monitor*** . The user will interact with the health tracking app via a CLI(command line interface) that will be created with the help of the Python package ***fire.*** The CLI will come with five predefined habits:

* Run (weekly)
* Workout (weekly)
* Dancing (weekly)
* Brush the teeth (Daily)
* Drinking water (Daily)

These predefined habits come with example tracking data with a period of 4 weeks that allow the user to test the app’s functionalities. In case the user decides to wipe the default history, it can be done via command line.

The following actions are allowed to be performed:

Define a habit:

*habit\_monitor create\_habit <habit\_name> <habit\_periodicity>*

Delete habit:

*habit\_monitor delete <habit\_name>*

Complete habit:

*habit\_monitor complete\_habit <habit\_name>*

Analyze habit:

*habit\_monitor analyze <operation\_number> <additional\_parameters>*

In the above command, operation\_number will be a digit identifying the operation and additional parameters if necessary. The following operations are available:

* 1: list all current habits tracked.
* 2: list all habits with the same periodicity. Additional parameter – Periodicity: d, w, m (daily, weekly or monthly).
* 3: get longest run streak among all habits being tracked
* 4: get longest run streak for a give habit
* 5: Ask custom question. This operation allows the user to query the application’s database using SQL. Additional parameter: <sql\_query>

Clean history:

*habit\_monitor clean\_history <habit\_name>*

The concept of Habits and the operations that will be performed over it will be encoded using Object Oriented and functional programming paradigms. Habits will represented as a Class and methods can be used to perform the actions over it.

The data about the habits will be persisted using a sqlite3 database. The app’s functionality will be tested using pytest.

The following diagram represents the Conceptual architecture of the application, with the three main classes that are going to be used:

: