

Conception Phase

Habit tracker app

Objective:

The main objective of this project is to develop a basic backend for a Habit tracking application that is permits to keep tracking of certain habits. To achieve this goal, I will first perform a conceptual study of the application. latter will allow me to reach easily the realization of the application by organizing the ideas and by structuring the process of coding.

Specification of functional requirements:

The following functional requirements outline the desired capabilities and actions that should be supported by the habit tracker application:

1.Create habit

2.Manage habit

- Delete habit
- Delete category
- Modify periodicity
- Go back to the menu

3.Check off habit

4.Show my analytics

- All your current habits with their details.
- All habits with a same time-period
- The longest streak you have out of all your current habits.
- The longest streak you've had for a specific habit

5.Exit

Choice of technology:

Several technologies play a crucial role in the development of a Habit Tracker application. Python 3:11 serves as the programming language of choice, providing a powerful and versatile platform for developing the application's back-end logic and functionality. PyCharm, an integrated development environment (IDE), enhances the development process with features such as automatic code completion, debugging tools, and project management capabilities. It enables developers to write, test, and maintain their code efficiently.

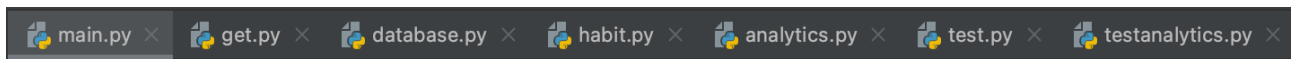
Data storage:

To handle the app's data storage requirements, SQLite3 is used as the database system. SQLite3 is a lightweight, serverless, and self-contained database engine that seamlessly integrates with Python. It provides a relational database management system within the app itself, eliminating the need for external database servers. SQLite3 offers simplicity, efficiency, and cross-platform compatibility, making it well-suited for managing the habit tracking data efficiently.

Organization structure:

The code is coded based on object-oriented programming (OOP). This means that classes are created with methods. The "Habit" class contains all methods and attributes related to the management of habits, and an "Analytics" class that handles habit analytics.

Fig.1 Screenshot PyCharm classes



Source: Own representation

Command Line interface:

In order for the user to be able to add, delete, manage and check off the habit, it is important to create a way to do this. Therefore, I decided to create a CLI where the user selects the task he wants to do, and after selecting the desired task, the program starts asking questions, and then the user enters the required information's.

Checking off habit:

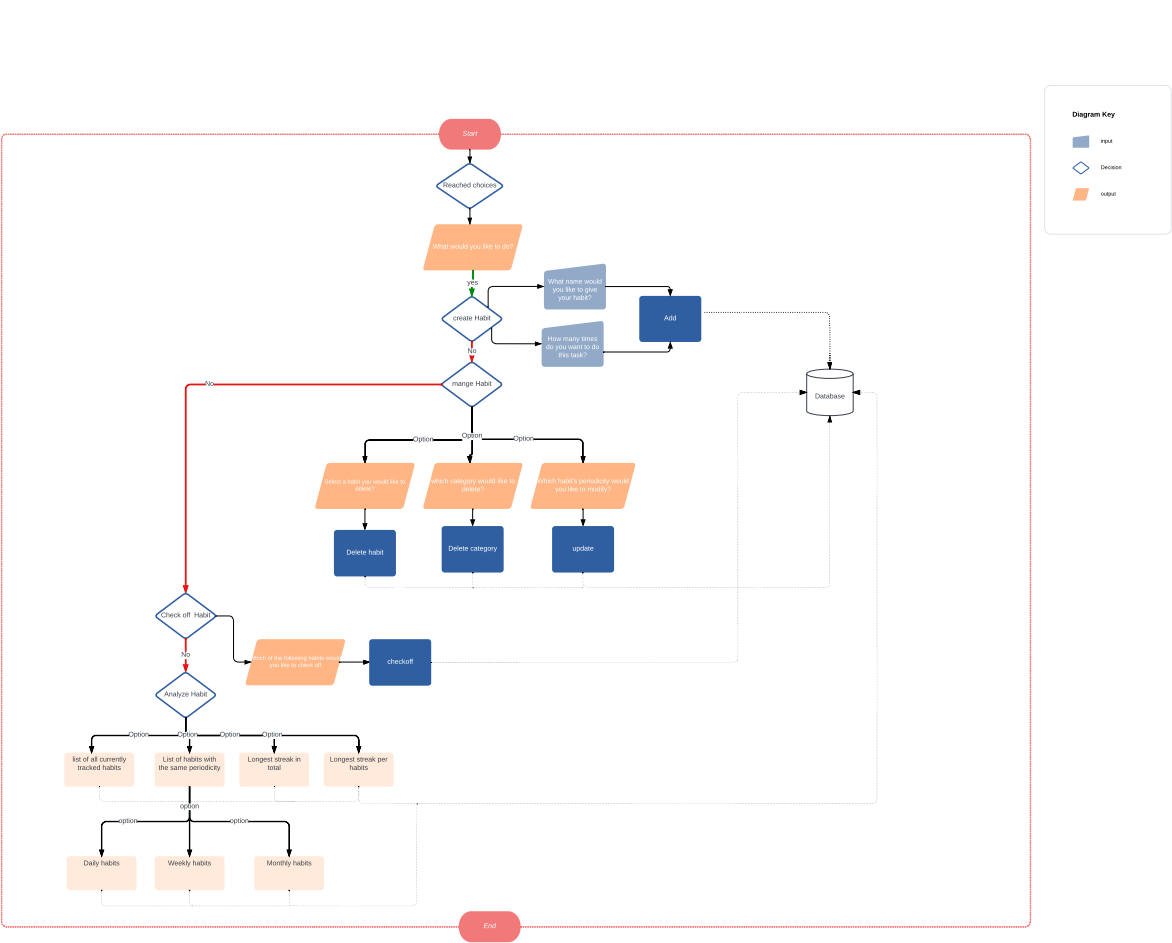
By increasing the streak by 1 when a habit is checked off and resetting the streak to 0 when it is not completed, we enforce the requirement that each habit be checked off at least once in the specified time period. If the habit is not completed, the streak is reset, indicating that the user has "broken" the habit for that time period.

Testing:

To ensure the validity and reliability of the components of the Habit Tracker and the analytics modules, incorporating unit tests is a critical aspect of developing a Habit Tracker app. That's why the Unittest framework will be employed to carry out the testing of the app.

Flowchart:

Fig.1 Habit tracker flowchart



Source: Own representation