Conception Phase: Habit Tracking App

1. Objective

The aim of this project is to build a simple habit tracking app using Python. The app will let users create and delete daily or weekly habits, mark them as done, and track how many times in a row they completed each habit. The program will also give users feedback on how consistent they have been withing completing habits.

The app will run in the command line, where users are prompted to provide inputs. It will use object-oriented programming to organize how habits are created and tracked, and functional programming to analyze the user's progress. The project will focus on the core functionality of a habit tracker and will not include any graphical interface.

2. Core Concept

The app is built around three classes:

Habit Class: This defines what a habit is. Each habit has a name, a frequency (daily or weekly), and a record of when it was completed. It can also calculate streaks (how many times in a row the habit was completed without missing).

HabitTracker Class: This handles all the habits. It can create, delete, store, and manage multiple habits at once. It also helps users mark habits as complete and saves the data to a file using JSON.

Analytics Module: This is a separate file that uses functional programming. It helps answer questions like:

- What habits am I tracking?
- What's my longest streak?
- What daily or weekly habits do I have?
- Which habit did I miss most often?

All user data will be saved in a .json file so changes made by the user are saved when the app is closed. When the app starts, it loads the habits from this file so the user can continue where they left off.

3. Tools and Technologies

- Python 3.7+
- JSON for data persistence

- Command-line interface for user interaction
- unittest module for testing
- GitHub for version control
- UML diagram created using Lucid Chart for documentation

4. User Experience and Flow

The app works as follows:

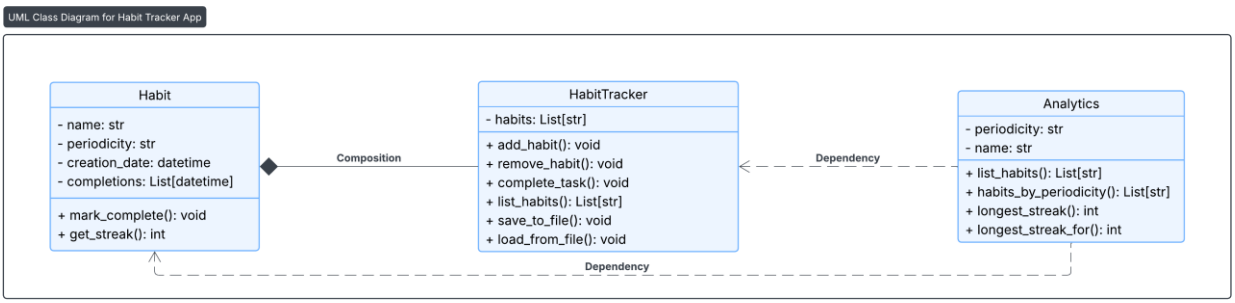
1. The user opens the program.
2. The app loads any saved habits from a file.
3. The user can:
    - Create a new habit
    - Delete an existing habit
    - Mark a habit as completed for the day/week
    - View a current list of habits
    - See habit statistics
4. The app tracks when each habit was completed and calculates if the user has kept or broken their streak.
5. All progress is saved in a file so the user can return and continue to make changes.

5. Diagram: System Components

The system is made up of three main parts:

- Habit (class)
- HabitTracker (class)
- analytics.py (module)

These parts interact with each other. For example, when the user marks a habit as done, the HabitTracker updates the habit and saves it. The analytics module looks at the data from all habits and shows results.



This diagram illustrates the relationship between the core components of the app, demonstrating how habits are defined, tracked, and analyzed.

6. Why I Chose This Design

- Classes make it easier to keep each habit organized with its own data and actions.
- Functional programming is efficient for handling user input and analysing data in a clear, reusable way.
- JSON is a simple way to save and load data, so users do not lose their progress.
- Command line interaction is simple and allows the focus to be put on creating a user-friendly code in Python instead of building a full app interface.

Additionally, this structure is clean, easy to understand, and allows the flexibility of adding more features later if needed.