Object Oriented and Functional Programming with Python – Development Phase

DLBDSOOFPP01
Task: Create a Habit Tracking App

Eric Anton Hauck
B.SC. Data Science
Matriculation Number 102305503

May 2025

INTERNATIONALE
HOCHSCHULE

Introduction

What?

- This habit tracker is a Python-based backend application designed to help users build positive habits and break bad ones. It allows users to create, track, and analyze habits over time.

Why?

- Developing good habits or breaking bad ones can be challenging. A habit tracker provides motivation by visualizing progress and ensuring accountability.
- The app focuses on essential functionality, offering a lightweight solution with a command line interface.
- It is designed for users who want to stay organized and monitor their routines with minimal setup.

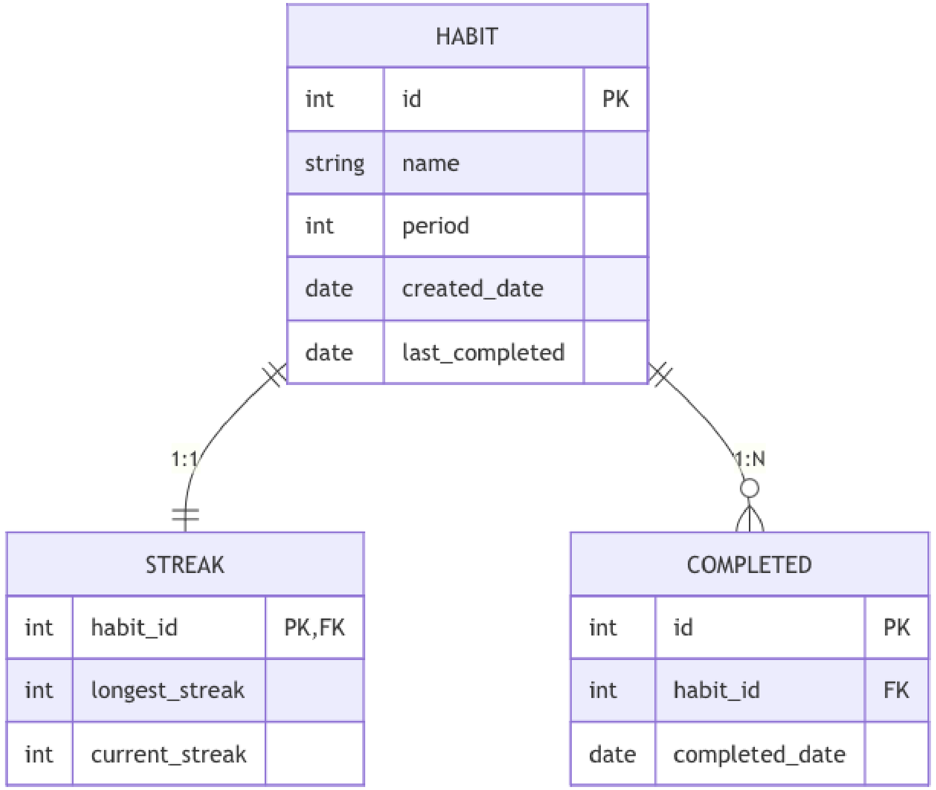
Database

The database schema consists of three main tables:

- **Habit:** Stores habit definitions, including the name periodicity, and creation date.
- **Completed:** Logs the completion dates for each habit, enabling streak calculations.
- **Streak:** Tracks the user's progress by storing the longest streak and current streak for each habit.

Why this structure?

- This normalized schema ensures data integrity and minimizes redundancy.
- It enables efficient queries for analytics.
- SQLite was chosen for its simplicity and lightweight nature, making it ideal for this project.



HABIT		
int	id	PK
string	name	
int	period	
date	created_date	
date	last_completed	

STREAK		
int	habit_id	PK,FK
int	longest_streak	
int	current_streak	

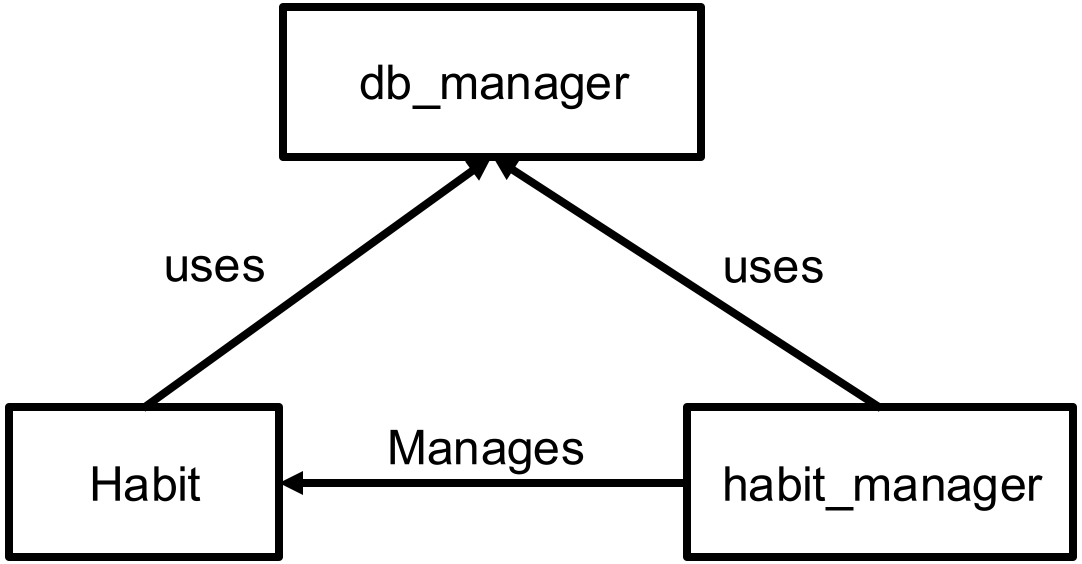
COMPLETED		
int	id	PK
int	habit_id	FK
date	completed_date	

Class diagram

The class diagram showcases the relationships between the following key components Habit Class (represents individual habits), Habit Manager Class (Manages all habits) and Database Manager Class (Responsible for database connection)

Reasoning:

- The object-oriented design ensures modularity and reusability.
- Encapsulation of habit-specific logic within the Habit class simplifies the Habit Manager's responsibilities, adhering to the single-responsibility principle.
- Creating a db_manager class allows for easy testability and scalability.



Modules and functions

Habit

- Can be completed
- Can be broken
- Checks if it is open
- Gets the due date
- Gets current streak
- Gets longest streak
- Updates longest streak

Habit_Manager

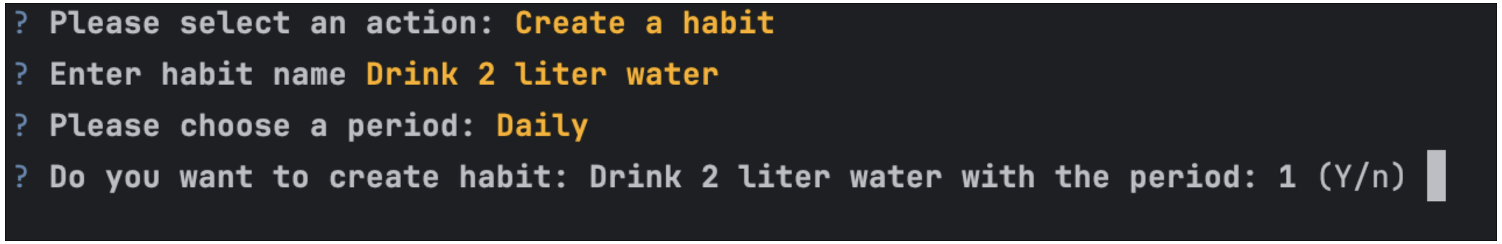
- Initialises the habit list
- Creates habits
- Deletes habits
- Lists open habits
- Lists habits periodically
- Shows habits with a specified period
- Lists longest streaks
- Lists current streaks

DB_Manager

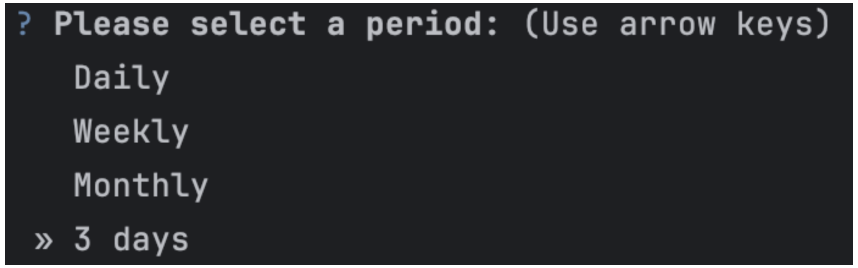
- Initialises the db
  - In normal mode
  - Or in test mode with 4 weeks of sample data
- Creates tables when needed
- Closes the connection

Command line interface

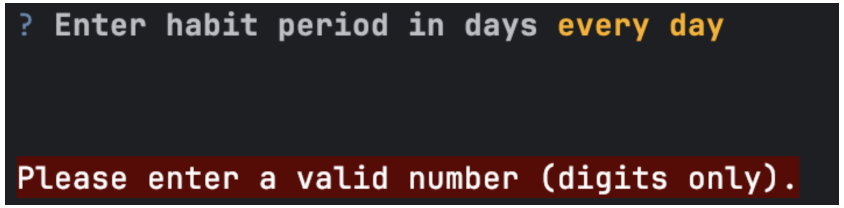
Users navigate through the app using a simple and intuitive CLI powered by questionary. Selections can be made using the arrow and enter keys. On-Screen instructions promt the user to input text or a period when needed. All functions changing the database need to be confirmed again by the user.



```
? Please select an action: Create a habit
? Enter habit name Drink 2 liter water
? Please choose a period: Daily
? Do you want to create habit: Drink 2 liter water with the period: 1 (Y/n)
```



```
? Please select a period: (Use arrow keys)
   Daily
   Weekly
   Monthly
 » 3 days
```



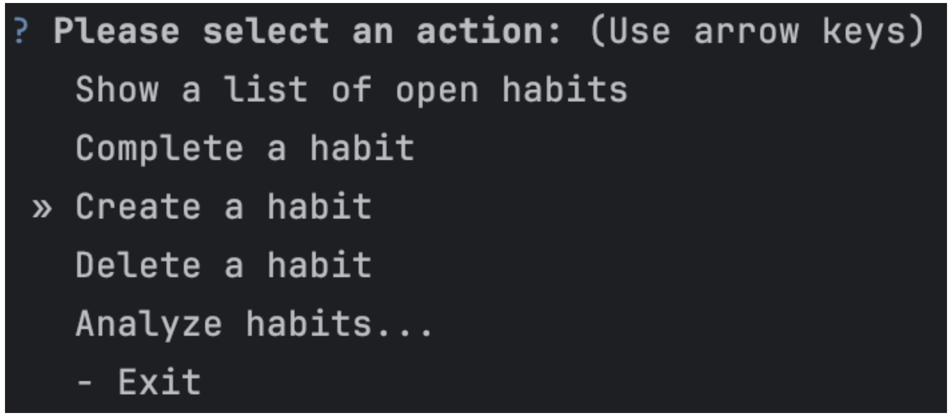
```
? Enter habit period in days every day


Please enter a valid number (digits only).
```

User input is validated and restricted if necessary, in the CLI to avoid mistakes when writing and reading from the database.

Main Menu

The main menu is the entry point for the user when starting the program. Here habits can be managed, and open habits can be shown with their respective due date.

From the main menu the user can enter the submenu “Analyze habits…” and exit the program.

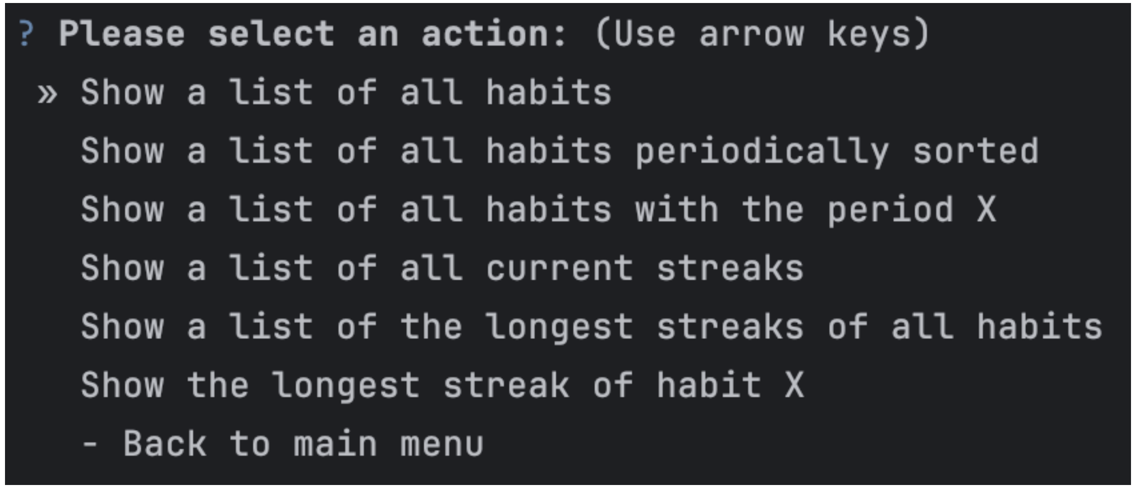


```
? Please select an action: (Use arrow keys)
   Show a list of open habits
   Complete a habit
 » Create a habit
   Delete a habit
   Analyze habits...
   - Exit
```

Submenu Analyze habits…

In the submenu there are all actions regarding the analysis of habits.

After choosing a menu point and the corresponding return from the function, the user is still in the analyze submenu and needs to go back to the main menu via the menu point.



```
? Please select an action: (Use arrow keys)
 » Show a list of all habits
   Show a list of all habits periodically sorted
   Show a list of all habits with the period X
   Show a list of all current streaks
   Show a list of the longest streaks of all habits
   Show the longest streak of habit X
   - Back to main menu
```

Tools

Python: The primary programming language used for its versatility and rich ecosystem.

SQLite: A lightweight relational database for habit data storage, ensuring persistence across sessions.

Questionary: A library for creating interactive command-line interfaces, enhancing user experience.

Git: Version control system used for tracking changes and collaborating effectively.

Pytest: Testing framework to ensure the reliability and correctness of the app's functionality.

Installation

Prerequisites

- Python 3.7 or higher
- pip (Python package manager)

Steps

Clone the repository:

```
git clone https://github.com/anton-hauck/IU_habit_tracker.git
cd IU_habit_tracker
```

Install dependencies:

```
pip install -r requirements.txt
```