SIMEON AKPANUDO

24th December 2022

OBJECT ORIENTED AND FUNCTIONAL PROGRAMMING WITH PYTHON.

DESIGN AND USER - FLOW OF THE APP.

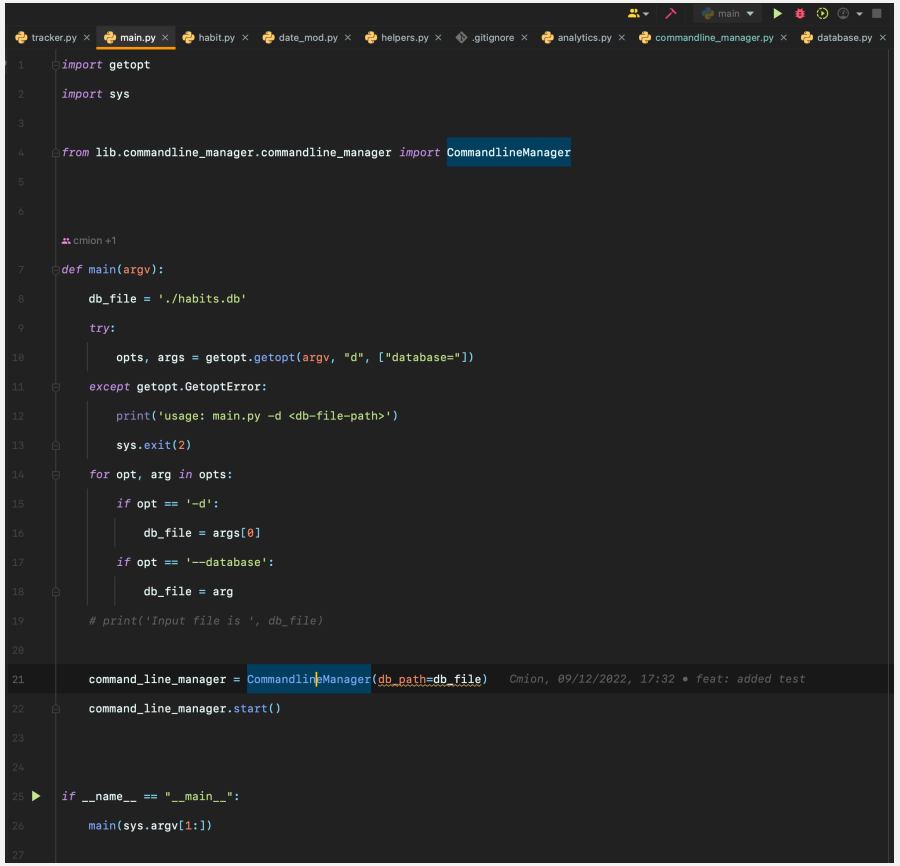
1. App runs a loop and only quits when the user quits the program.
2. Asks user what they'll like to do, eg. (Add habit, Manage habits, show stats e.t.c).
3. Performs the user's preferred action, until user goes back or closes the application.

CODE ENTRY POINT

Main.py and commandline_manager.py

- Code uses python's conventional entry point. (main function).
- Main function accepts a command-line argument that is used to set a custom **.db** file where the habits would be store. If none is provided, default 'habits.sb' would be used.
- Main calls a command-line manager class.
- Command-line manager class has start method, that uses a while-loop that only exits based on the user's input or a forced ctrl-c.
- Command-line manager also interacts with other classes, such as the HabitTracker and Analytics classes, it only accepts the user's command-line input, processes them and calls the required method(s) of this classes.
- **Imports**: colorama (for styling), simple_term_menu (for terminal menu), Analytics class, HabitTracker class.

Code snapshot

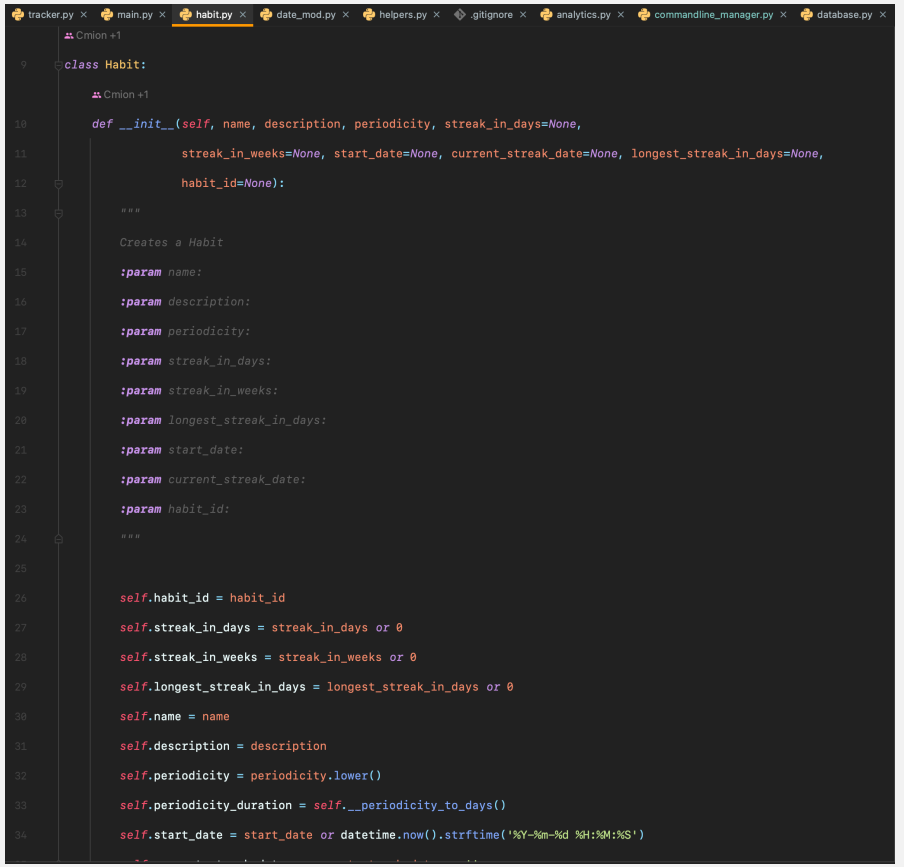


HABIT MODEL

habits.py

- Contains a habit class/model.
- Habit class is a physical model for a habit, it creates a class using the instance parameters.
- Handles data evaluation (e.g: Streaks calculation and modification).
- Also contains static methods that transforms a database response to a list of habits and vice versa.
- All through the app, habits are interacted with using this class. This class serves a convenience, instead of interacting with the raw data retrieved from the sqlite database.

Code snapshot

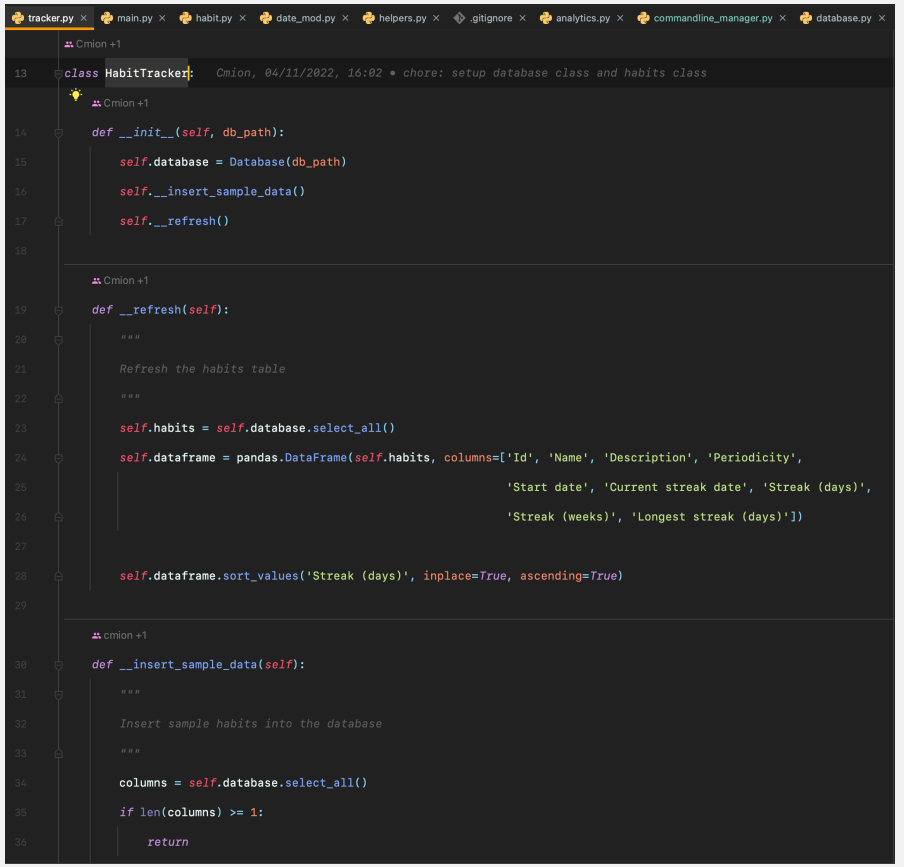


DATABASE, TRACKER AND ANALYTICS CLASSES

database.py, tracker.py and analytics.py

- **Database:** This class manages the creation of the apps database table.
  - Implements insertion, update and deletion methods.
  - Exposes a database cursor.
  - **Imports**: sqlite3.
- **Tracker:** This class implements methods that performs the core functionality of the app.
  - Implements methods such as add_habit, delete_habit e.t.c.
  - Creates a human readable table using pandas and PrettyTable
  - Loads sample data upon app initialization.
  - Prints user's habits to terminal.
  - **Imports**: colorama, pandas, prettytable, Database class and Habit class.
- **Analytics:** This class performs simple analytics on the user's habits data.
  - Retrieves the longest and current longest streak.
  - Retrieves habits with same periodicity.

Code snapshot



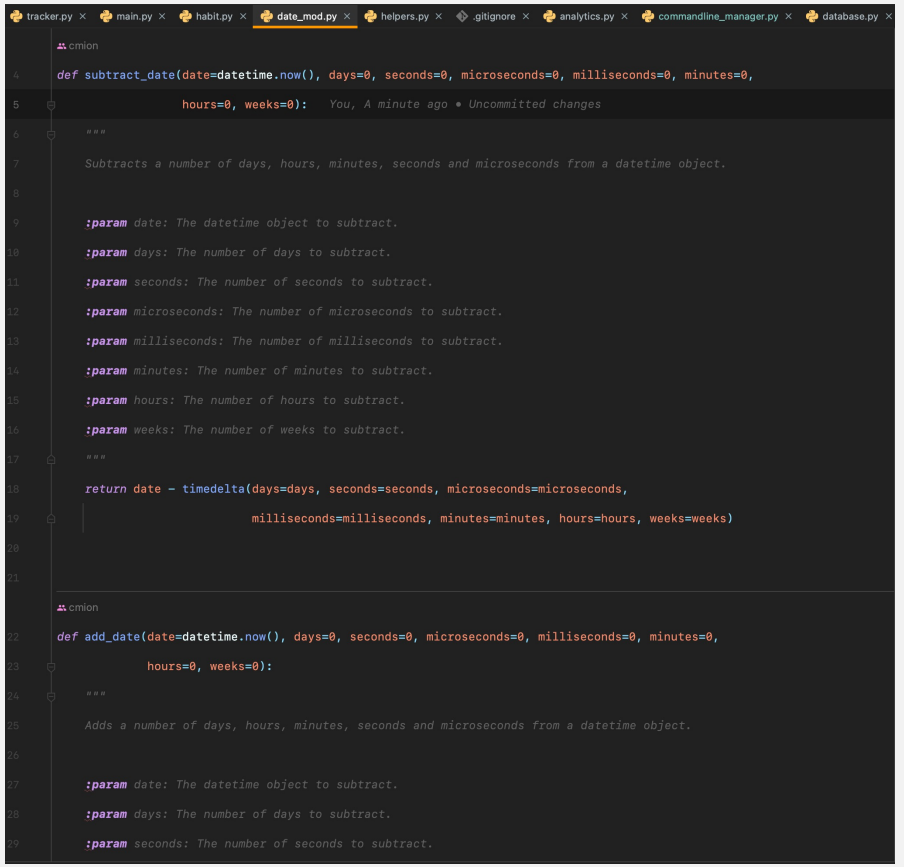
HELPER FUNCTIONS

date_mod.py and helpers.py

- Data_mod.py: This file contains functions that handles data subtraction and addition.

- Helper.py: Contains a simple habit sample function.

Code snapshot

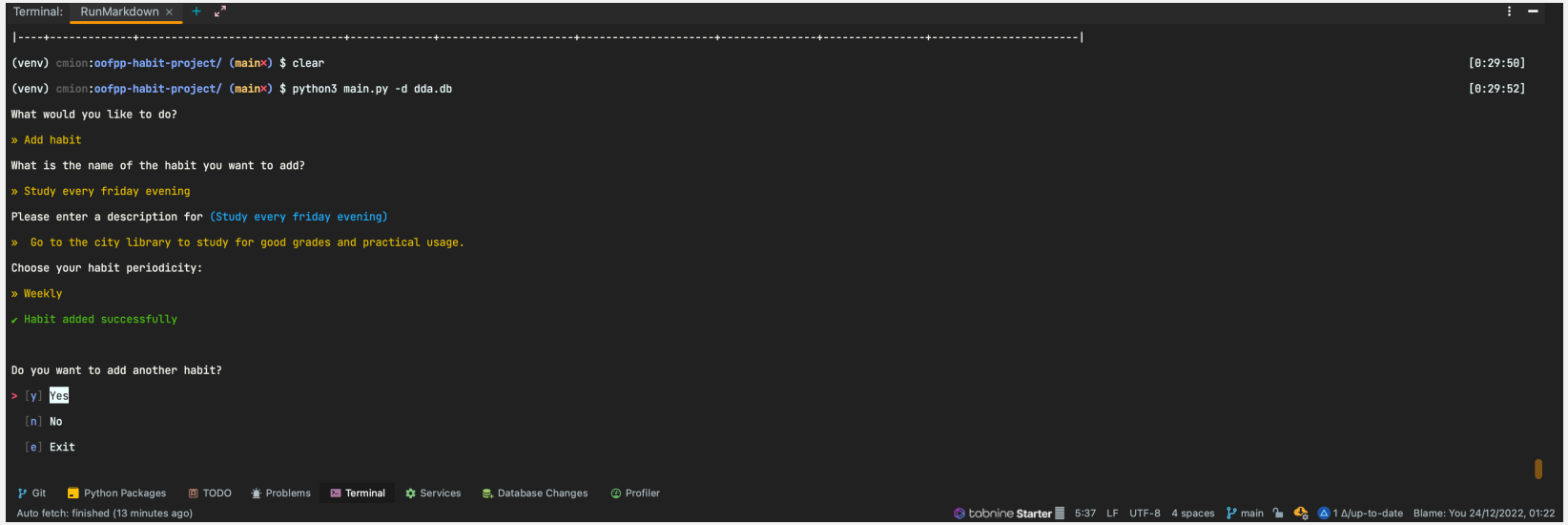


ADDING HABITS

- This process follows the interactive design used in the app.
  1. What is the name of the habit you want to add?.
  2. Please enter the description for (name-of-habit).
  3. Choose your habit periodicity (Here the user can either press keys such as: **w** for weekly, **d** for daily and **c** for cancel. They can also use the **up and down arrows** to navigate through the options and press **Enter** to choose a preferred periodicity.

  Finally, the user is greeted with a message, if the action was successful.

  Note: If a user enters whitespace characters, the app would prompt them again to enter either (the name or description).

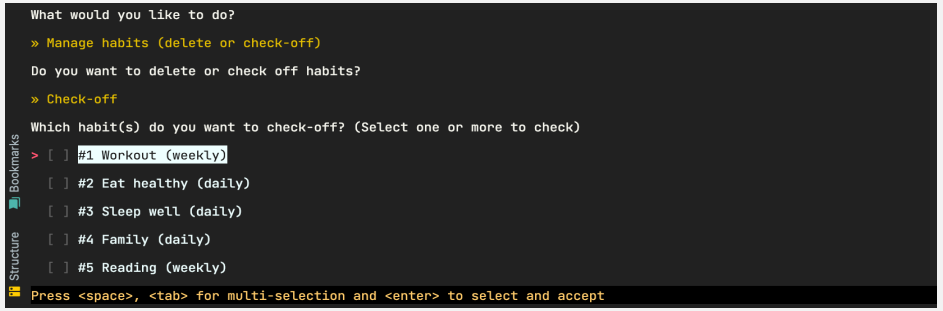


MANAGING HABITS

Checking off

User is presented with a multi-selection menu of habits to pick which habit(s) they'll like to check-off.

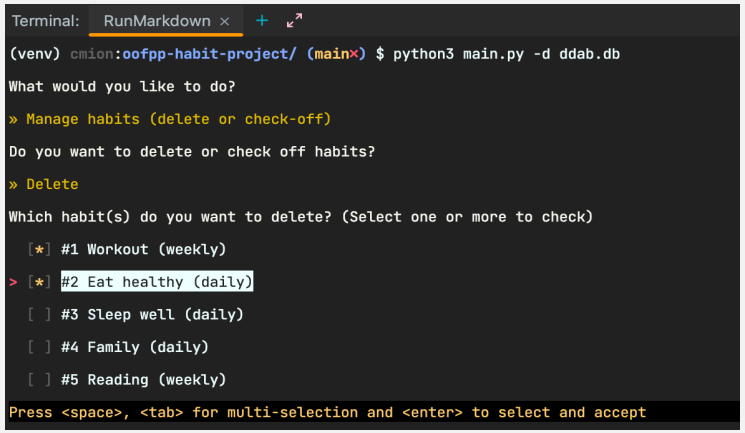
- User cannot check-off an habit twice within it's periodicity.
- When performing habit check-off, the streaks are recalculated, and if a user failed to mark an habit as complete within it's periodicity, the current streak is set to 0, and the longest streak is set, if the most current streak is greater than the last recorded.



Deleting

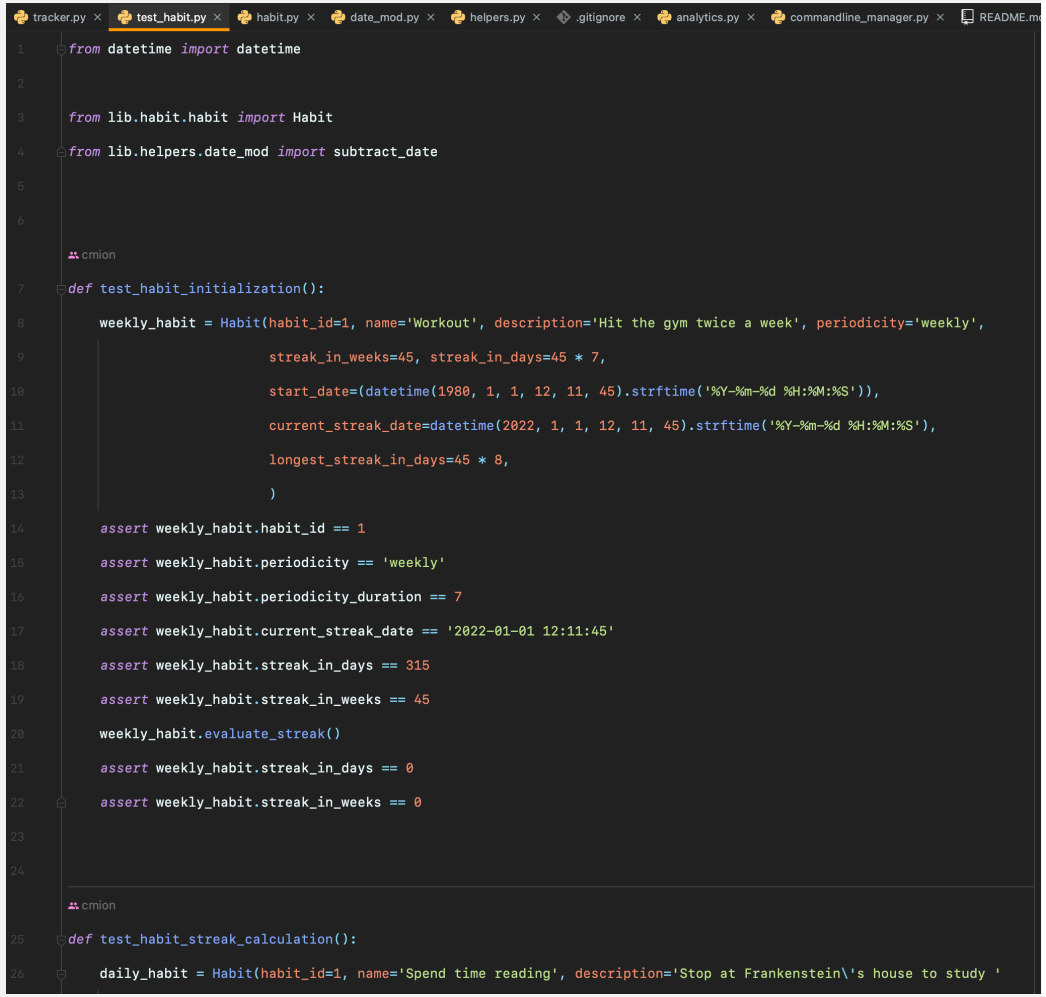
User is presented with a multi-selection menu of habits to pick which habit(s) they'll like to delete.

After user picks, they're prompted to confirm whether they want to delete the selected habits or go back.



TESTING (WITH PYTEST)

- The type of testing implemented is a unit test.
- Only two modules were tested, the habits model/class, and the helper functions.
- **Reason for not testing other modules**: Other untested modules requires the usage and connection to the SQL database. Testing these modules would require an integration test or an e2e (End to End) testing. Another alternative would be a mock of the database objects. Mocking is currently not implemented in this project.



CHANGES BETWEEN PHASE 1 & 2

1. Adding of console styling with (colorama).
2. Usage of terminal menu with (simple_term_menu).
3. Introduction of a command line manager to help interface the user with the business logic.