Concept of the Habit Tracker

Introduction

A habit is an action which repeats overtime. The habit tracking program is a simple application that aims to help users build positive habits and track their progress over time in an efficient way, using object-oriented programming principles in Python. This concept outlines the technical foundations of the program.

The program should contain the following general functions:

- "Add" Function: This function is designed to establish new habits, providing the option to set goals, a time frame, streak, best streak, and status, for each habit.
- "Check" Function: This function allows users to mark the completion of a task associated with a habit and update the status and streaks of that habit.
- "Remove" Function: This function enables the removal of existing habits from the system.
- "Analyze" Function: This function facilitates the tracking and analysis of habits over time.

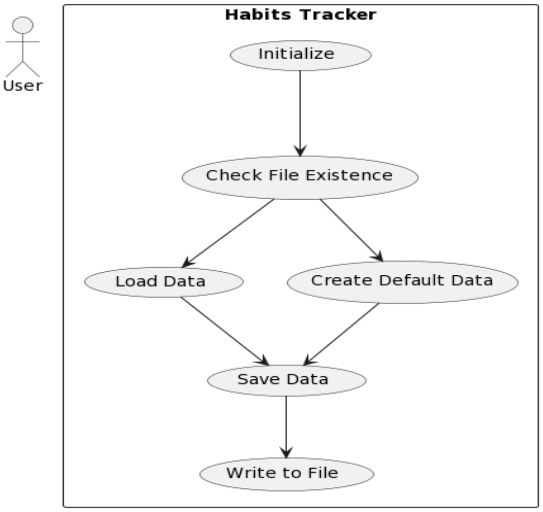
Concept

The program will function based on classes and objects. It will begin by initializing a "Tracker" object, which will record the current date, and load the existing information and habits from the data file. Habits will be represented as a class to ensure an organized approach for habit management. There will be three files, one JSON file for storing data, and two python files consisting of a Main file and a file for the functions and classifications.

Data Storage

For simplicity and efficiency, the program will utilize a JSON file for streamlined data storage and to avoiding reliance on external frameworks or libraries. Implemented in Python, the application leverages standard libraries for JSON handling.

The program manages data through read and write operations to a JSON file. During initialization, existing data is loaded from the file, if not it will create a default data file, maintaining continuity.



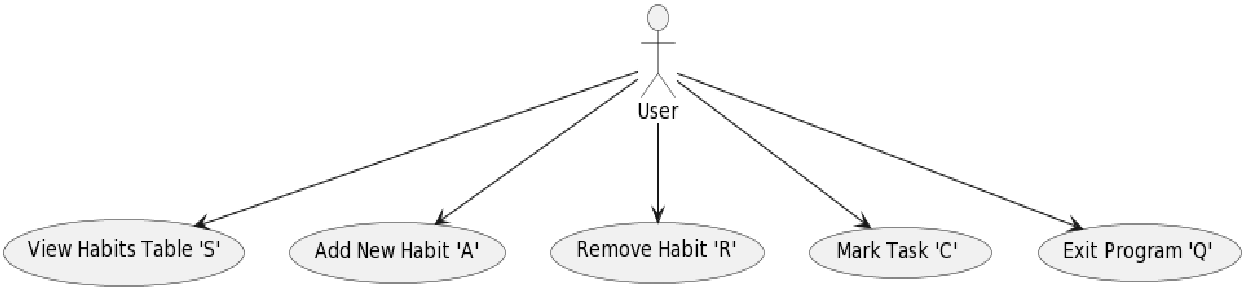
User Interface

Color-coded console output will enhance the visual presentation of messages. This approach not only improves the overall aesthetic but also contributes to effective communication. This visual enhancement contributes to a more intuitive and user-friendly experience.

The program should generate formatted tables, enhancing the display of habit information in the console, improving readability. This utilization of formatted tables not only enhances the visual presentation but also provides a comprehensive overview of habit-related information for analysis.

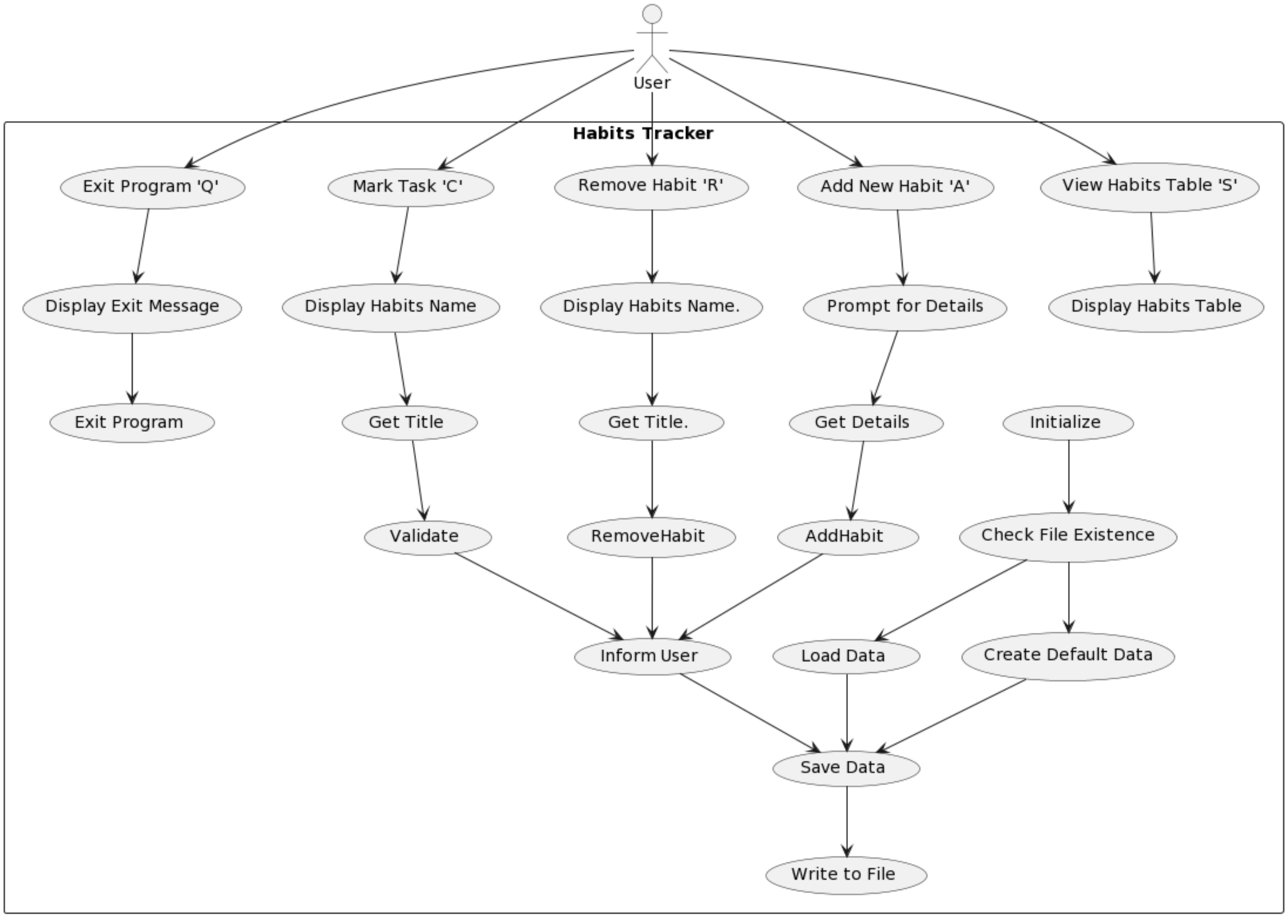
The program features a user-friendly command-line interface (CLI) with distinct commands for seamless interaction: The CLI displays a table of habits, and prompts users for input by incorporating intuitive single-letter commands. This design aims to simplify navigation and enhance the overall user experience.

- "Add" Function: Users can add a new habit by entering the letter "A" or "a".
- "Remove" Function: To remove a habit, users can use the letter "R" or "r".
- "Check" Function: Users can mark a task as complete by entering "C" or "c".
- "Analyze" Function: To view all habits and the information, users can use "S" or "s".
- "Exit" Function: Users can quit the program by entering the "Q" or "q".



Input validation will be required to guarantee accurate data type entry and to avoid user input errors. In the event of an invalid command, the program shouldn’t crash but instead notify the user and correct them. This approach enhances the reliability and user-friendliness of the program by mitigating errors and guiding users towards correct inputs.

Full Diagram:


User
Habits Tracker
Exit Program 'Q'
Mark Task 'C'
Remove Habit 'R'
Add New Habit 'A'
View Habits Table 'S'
Display Exit Message
Display Habits Name
Display Habits Name.
Prompt for Details
Display Habits Table
Exit Program
Get Title
Get Title.
Get Details
Initialize
Validate
RemoveHabit
AddHabit
Check File Existence
Inform User
Load Data
Create Default Data
Save Data
Write to File