HCLI - Habit Tracker CLI

Conception Phase

Author: Alejandro Moral Aranda Date: February 9, 2025

1 Introduction

HCLI (Habit Tracker CLI) is a command-line-based habit tracking application that provides users with a lightweight, efficient, and distraction-free way to track their habits. Unlike traditional mobile or web-based applications, HCLI focuses on speed and usability through a command-line interface (CLI), allowing users to add, check, and analyze their habits through simple commands.

The goal of HCLI is to provide a seamless experience for users who prefer a terminal-driven workflow, ensuring that they can manage their habits without the need for graphical interfaces. This document outlines the technical foundation, system architecture, and workflow behind HCLI.

2 Core Functionalities

The HCLI application is designed with the following core functionalities:

- Adding habits: Users can define habits with a name and periodicity (daily/weekly).
- Tracking progress: Users can mark habits as completed.
- Viewing streaks: The system calculates the longest streaks for each habit.
- Analytics and summaries: Users can analyze which habits they struggle with and view overall progress.
- Reminders: Displays pending habits that need to be completed.
- Dashboard: Provides visual analytics (ASCII or graphical).
- Configuration: Users can adjust file storage paths and customize settings.

3 User Interaction and Workflow

The general user flow of HCLI is designed to be intuitive and efficient. The interaction follows a simple command-driven workflow:

1. User initializes the application by setting up their username:

   ```
   python main.py setup-user
   ```

2. User adds a new habit specifying the habit name and periodicity:

```
python main.py add "Workout" daily
```

3. User checks off a habit when completed:

   ```
   python main.py check "Workout"
   ```

4. User views analytics and streaks:

   ```
   python main.py summary
   python main.py streaks
   ```

5. User manages configuration:

   ```
   python main.py config --show
   ```

6. User accesses dashboard visualization:

   ```
   python main.py dashboard
   ```

4 Data Management

All habit data is stored in JSON files to ensure simplicity and portability. The data files include:

- **habits.json** - Stores user habits, their periodicity, and timestamps.
- **user.json** - Stores user preferences, including the username.
- **config.json** - Stores the application configuration settings.

5 System Architecture

The HCLI application consists of several key components:

- CLI Interface: Handles user input and executes commands.
- Data Management: Reads and writes habit data to JSON.
- Analytics Engine: Calculates streaks, summaries, and habit trends.
- Reminder System: Checks for pending habits.
- Dashboard: Provides ASCII or graphical visualizations.

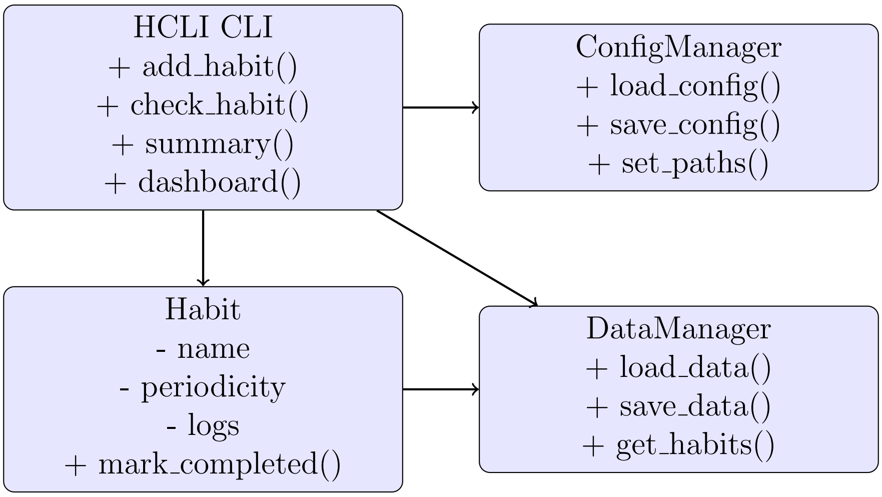

Figure 1: UML Class Diagram for HCLI

6 Tools and Technologies

HCLI is implemented using the following technologies:

- Python - The core programming language.
- Typer - Used for handling CLI commands.
- Rich - Provides colorful terminal output and tables.
- JSON - Stores persistent data for habits and user preferences.
- Pytest - Used for unit testing the application.

7 Implementation Plan

The development of HCLI follows a structured roadmap:

- Phase 1: Core functionality (add, check, list habits).
- Phase 2: Advanced features (analytics, reminders, streaks).
- Phase 3: Optimization, visualization, and testing.

8 Conclusion

HCLI is a lightweight, efficient, and terminal-based habit tracking solution designed for users who prefer CLI-driven workflows. By leveraging simple commands, structured data management, and insightful analytics, HCLI provides a seamless experience for habit formation and tracking. This document lays the foundation for development, ensuring a well-structured implementation and feature roadmap.