

# HCLI - Habit Tracker CLI

## Development Phase

Author: Alejandro Moral Aranda

Date: February 9, 2025

---

## 1 Implementation Overview

After defining the conceptual foundation in the previous phase, the HCLI (Habit Tracker CLI) has been developed following a structured approach. The main objective was to ensure a functional, efficient, and user-friendly habit tracking experience within a command-line interface.

Development focused on core functionalities, ensuring stability, data persistence, and analytics features while following the defined roadmap.

## 2 Development Process

The implementation followed a structured plan:

- Building the CLI Framework: The application was built using the Typer library for intuitive command handling.
- Data Storage System: Habit data is stored in JSON files, allowing persistence and easy retrieval.
- Core Habit Management: Users can add habits, check them off, and delete them.
- Analytics and Insights: Features like streak tracking, habit summaries, and pending habit reminders were implemented.
- Testing and Debugging: Pytest was used to ensure the reliability of habit tracking functionalities.
- User Interaction Enhancements: Rich was integrated to improve the command-line interface with color-coded tables and summaries.

## 3 Challenges and Solutions

During development, various challenges were encountered and resolved:

- Handling User Input Robustly: - Issue: CLI commands needed to handle missing or incorrect inputs. - Solution: Implemented validation and error handling for missing arguments and incorrect formats.
- Ensuring Data Integrity: - Issue: Avoiding corruption of JSON files when multiple commands modify them. - Solution: Implemented safe read/write operations with exception handling.
- Managing Habit Streaks Accurately: - Issue: Detecting broken and continued streaks based on habit periodicity. - Solution: Developed an algorithm that dynamically calculates streaks for daily and weekly habits.
- Enhancing User Experience: - Issue: Providing a more engaging CLI without a GUI. - Solution: Used Rich for color-coded outputs, tables, and ASCII dashboards.

## 4 Updated System Architecture

The implementation resulted in a well-structured system, refined from the conceptual phase. The key components are:

- Command Handler: Interprets and executes user commands.
- Data Manager: Stores and retrieves habit-related information.
- Analytics Engine: Calculates streaks, pending habits, and habit performance.
- Visualization Module: Displays data in an ASCII-based or graphical dashboard.

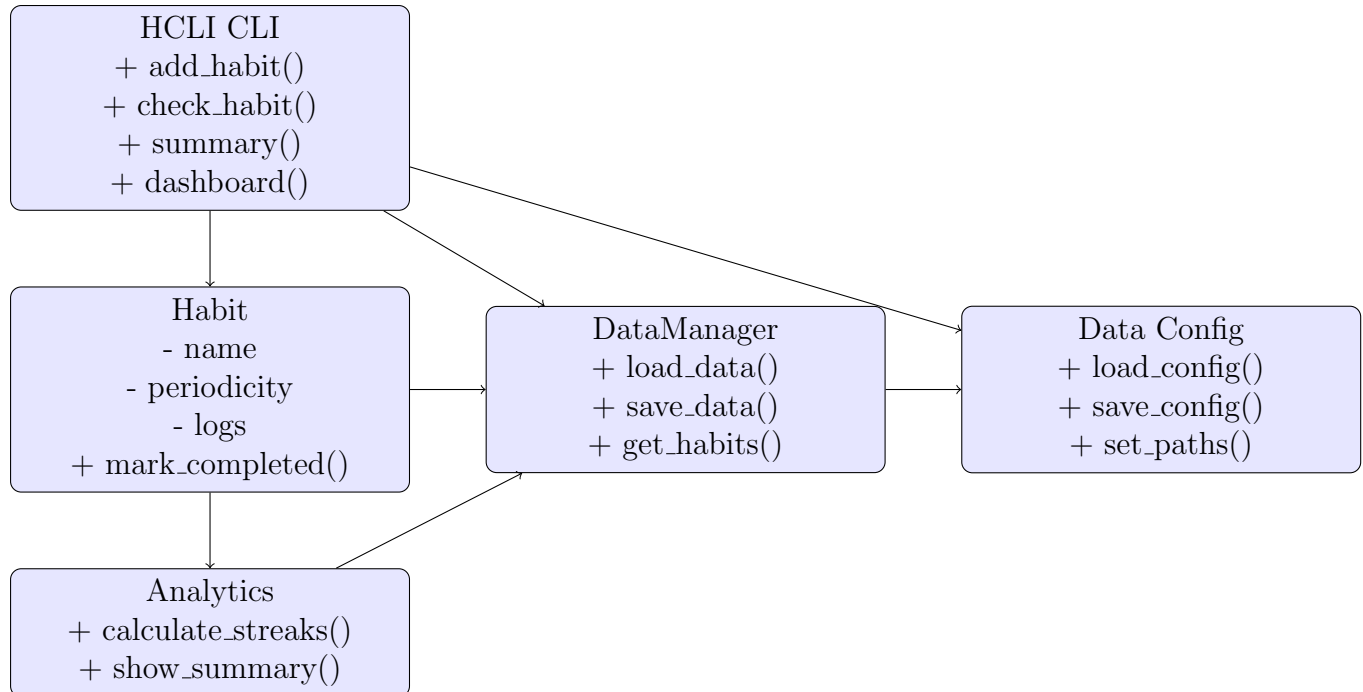


Figure 1: Updated UML Class Diagram for HCLI with Data Config

## 5 Intermediate Progress

Key milestones completed so far:

- Core functionalities fully implemented.
- Successful handling of habit storage and retrieval.
- Streak tracking and habit analytics working correctly.
- Command-line interface refined for better user interaction.
- Initial test cases running successfully with Pytest.

## 6 Next Steps

To finalize the implementation, the following enhancements will be made:

- Improve test coverage by expanding Pytest cases.
- Refine the dashboard visualization to provide better insights.
- Optimize data storage operations for better performance.
- Gather user feedback and implement improvements.

## 7 Conclusion

The development phase of HCLI has successfully transformed the initial concept into a functional CLI-based habit tracker. The system is stable, intuitive, and efficient, and will

be refined further before the final submission.