Habit Tracking App - Concept Document

---

1. Introduction

This document outlines the system design and architecture of a Habit Tracking Application written in Python. The aim is to build a lightweight command-line tool that allows users to track recurring habits, record progress, and analyze habit consistency. The app follows object-oriented principles and applies functional programming for analytical components.

---

2. Key Components

2.1. Class Design

Class: Habit

- name: str
- periodicity: str ("daily" or "weekly")
- created_at: datetime
- completions: list[datetime]
- Methods:
    - complete_task(date: datetime)
    - is_broken(period_start: datetime, period_end: datetime)
    - get_streak()

Class: HabitManager

- habits: list[Habit]
- Methods:
    - create_habit(name, periodicity)
    - delete_habit(name)
    - get_habit(name)
    - list_habits()
    - load_data() / save_data() (JSON or SQLite)

Module: Analytics (Functional)

- get_all_habits(manager: HabitManager) -> list
- get_habits_by_periodicity(manager, periodicity: str) -> list
- get_longest_streak(manager: HabitManager) -> Tuple[str, int]

- get_longest_streak_by_habit(habit: Habit) -> int

---

3. Data Storage

The app uses file-based persistence with two options:

- **JSON file** for easier setup and portability
- **SQLite database** (optional for more robust solution)

---

4. User Interaction Flow

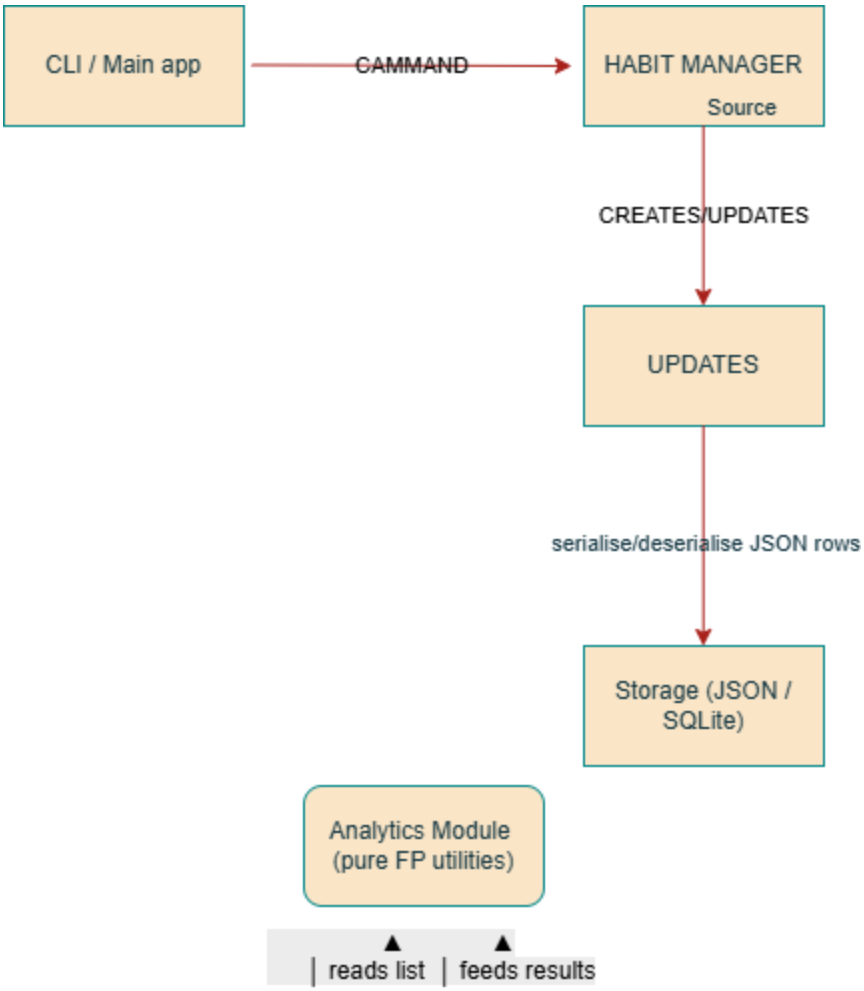
Users interact with the app via a CLI. The interface allows users to:

- Create new habits
- Mark habits as complete
- View habits and their streaks
- Analyze habits using analytics commands

Tools: Built-in input() or third-party click

---

5. Diagram



---

6. Summary

The application isolates concerns: Habit objects hold state, HabitManager orchestrates CRUD and persistence, and a pure functional analytics.py offers reusable insights. Data is saved in habits.json by default, making the app runnable without extra setup, while an SQLite adapter can be added later. The modular layout, CLI entry-point, and a pytest suite keep the codebase both extensible and verifiable.