Habit Tracker: Technical Concept

1. Introduction

This document provides a technical overview of the Habit Tracker application, a command-line tool designed to help users build and maintain positive habits. The application allows users to define, track, and analyze their habits.

2. System Architecture

The application is built with a layered architecture that separates concerns and promotes modularity. The architecture consists of the following layers:

- **Presentation Layer (cli.py)**: This is the user-facing layer, responsible for handling user input and displaying output. It is implemented as a command-line interface (CLI) using the click library.
- **Logic Layer (database.py)**: This layer contains the core functions of the application, such as creating habits, completing tasks, and calculating streaks. It acts as an intermediary between the presentation layer and the data access layer.
- **Data Access Layer (models.py, database.py)**: This layer is responsible for all interactions with the database. It uses SQLAlchemy, an Object-Relational Mapper (ORM), to map Python objects to database tables. The data models are defined in models.py, and the database connection and session management are handled in database.py.

3. Data Model

The data model is the heart of the application, representing the structure of the data stored in the database. It is defined using SQLAlchemy's ORM capabilities, which allows us to work with Python objects instead of raw SQL queries.

The core components of the data model are:

- **Habit**: This is the base class for all habits. It uses a single-table inheritance pattern to store different types of habits in a single database table. The type column serves as a discriminator to distinguish between different habit types.
- **DailyHabit and WeeklyHabit**: These are subclasses of Habit that represent habits with different periodicities. They inherit the common attributes of the Habit class and can have their own specific behaviors.
- **HabitInstance**: This class represents a specific occurrence of a habit. Each time a habit is due, a new HabitInstance is created. This allows us to track the completion status of each habit over time.

The classes' relationships can also be seen in figure 1.

4. Data Storage and Retrieval

The application uses an SQLite database to persist data. SQLite is a lightweight, file-based database that is well-suited for small to medium-sized applications. The database file, habits.db, is stored in the Source directory.

SQLAlchemy is used as the Object-Relational Mapper (ORM) to abstract away the complexities of database interaction. This allows us to work with Python objects and methods instead of writing raw SQL queries. The database.py module contains all the functions for creating, reading, updating, and deleting data.

Key data management functions include:

- save_habit(): Saves a new habit to the database.
- save_instance(): Saves a new habit instance to the database.
- complete_task(): Marks a habit instance as complete and creates the next instance.
- get_all_habits(): Retrieves all habits, with an option to filter by type.
- current_streak_for_habit(): Calculates the current streak for a given habit.
- backfill_instances(): Ensures that there are no gaps in the timeline of habit instances.

5. User Interaction and Flow

The application provides a command-line interface (CLI) for user interaction. The CLI is built using the click library, which makes it easy to create a structured and user-friendly command-line tool.

The main commands available to the user are:

- add-habit: Creates a new habit.
- list-all-habits: Lists all existing habits.
- list-all-active-habits: Lists all active habit instances.
- complete-task: Marks a habit as complete for a given date.
- show-longest-streak: Shows the longest streak for a habit or for all habits.
- delete-habit: Deletes a habit and all its associated data.

UML Diagram

The following diagram provides a comprehensive overview of the application’s modules, classes, and their interactions.

Figure 1: Class diagram. m of the habit tracker