PORTFOLIO
Object Oriented and Functional Programming with Python (DLBDSOOFPP01)

Phase 1: Conception phase

Student Name: Akawa Johannes Sheepo
Matriculation Number: 92131324

1. Introduction

In a present world that is goal-oriented and productivity-driven, building and maintaining positive habits is crucial. This project aims to develop a basic habit that enables users to define, monitor and analyse their habits more effectively. This application will focus on developing a backend of a habit tracking application using the Python 3.7+ with the following functionality; habit creation, task completion, streak tracking, and analytics. The final product will be a command-line interface(CLI) application, that is able to store data persistently in the SQLite and be able to provide analysed data to identify user habit patterns.

2. Scope

The Habit tracking application will be developed within this scope. The following features below make up the scope of this application. The application should allow users to create, update, and delete habits. Users should be able to mark habits as complete within their defined period such as daily or weekly and track habit completion timestamps. Moreover, analyzing user behavior is an essential functionality. Hence, this application should be able to perform the following analyses. Firstly viewing all current habits and filtering habits by periodicity. Secondly, viewing the longest streak across all habits and viewing the longest streak for a specific habit. All data used in this application should be stored and retrieved from SQLite database. There will be no graphical user interface, therefore a command-line user interface is found sufficient for this application. Finally, to ensure an error-free application and correctness of the logic all modules shall be tested.

3. Specification

The specifications include functionality, features and requirements of the Habit Tracking Application. Below is a list of the components:

1. **Habit Class**: Comprises the individual habits. Each habit has a name, periodicity, creation date, completion history, and current streak.
2. **HabitTracker Class**: This is a controller class where the user can manage multiple habits. It has functions that allow users to add and remove habits. It will provide functions to retrieve all habits, filter by periodicity, and calculate streaks.

3. **Database Storage**: The application data will be stored in a SQLite database, which will be read and written to for persistence.

4. **Analytics Module**: this is where the functional programming techniques utilised. The purpose of functional programming is to analyse habits, providing functions to list all the habits stored, filter habits using the period and determine the longest streaks.

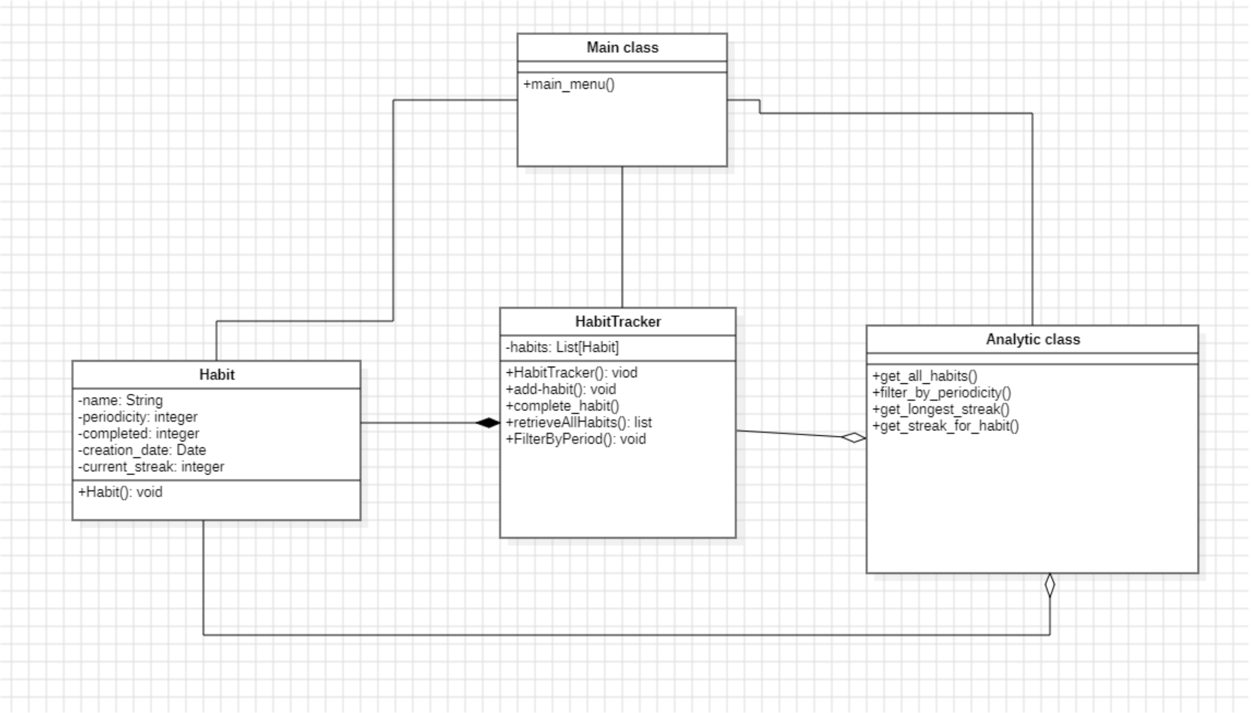
5. **Command-Line Interface (CLI)**: A user-friendly CLI will allow users to interact with the application by selecting the actions they want to perform. The interface should have options to create habits, complete tasks, view habits, and access analytics.

4. Deliverables

Up on the completion of this project, the following deliverables will be provided:

1. A fully functional command-line interface habit tracking application implemented in Python 3+.
2. Documentation detailing installation instructions, usage guidelines, and code comments.
3. A set of predefined habits with example tracking data for testing purposes.
4. A conceptual diagram illustrating the structure and relationships of the application components.
5. A unit test suite to validate the functionality of the application.

5. Conceptual diagram



Main class
+main_menu()
Habit
-name: String
-periodicity: integer
-completed: integer
-creation_date: Date
-current_streak: integer
+Habit(): void
HabitTracker
-habits: List[Habit]
+HabitTracker(): viod
+add-habit(): void
+complete_habit()
+retrieveAllHabits(): list
+FilterByPeriod(): void
Analytic class
+get_all_habits()
+filter_by_periodicity()
+get_longest_streak()
+get_streak_for_habit()