Habit Tracking Application

Table of contents

[Overview 1](#_Toc99993409)

[Problem statement 1](#_Toc99993410)

[Requirements 1](#_Toc99993411)

[Use cases 1](#_Toc99993412)

[Business Rules 1](#_Toc99993413)

[Design 2](#_Toc99993414)

[UML Diagrams 2](#_Toc99993415)

[Methodology 2](#_Toc99993416)

[Technology choices 2](#_Toc99993417)

[Questions 3](#_Toc99993418)

[Selection/parameterization and use of programs (software,tools,app) 3](#_Toc99993419)

[Assumptions 3](#_Toc99993420)

[References/Bibliography (See guidelines portfolio) 3](#_Toc99993421)

# Overview

Everybody wants to stop bad habits and create good habits in its place. They are turning to technology for assistance to achieving this. We want to create a habit tracking application to assist them to achieve their goals.

## Problem statement

We need to create a backend for our habit tacking application.

## Requirements

### Use cases

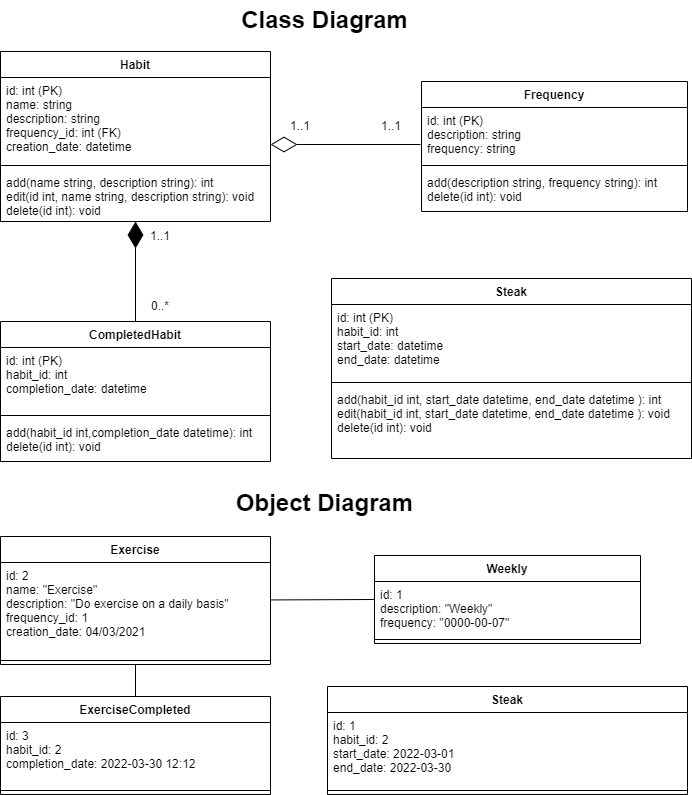
1. As a user I want to create a habit with a description or specification and frequency (how often must it be repeated)
2. As a user I want to flag a task as completed at a specific time
3. As a user I want to analyse the data as follows
   1. List of currently tracked habits
   2. List of habits with the same periodicity
   3. What is the longest run streak for a habit?
   4. List of daily habits
   5. Which habits do I struggle with?
4. As a user I want to store the data between sessions

### Business Rules

1. The user must complete a task once during the period otherwise he breaks the habit
2. The user must complete a task for x consecutive periods to establishes a streak

# Design

## UML Diagrams



## Methodology

What methodology was used to plan or to solve the issue

## Technology choices

* Python version 3.10.3
* Database – ***sqlite3***
* Make use of ***pytest*** and follow a TDD approach
* Visual Studio code for my IDE
* Sqlite – How to store/retrieve
* FastApi
* CLI- fire or click?
* Docker container + Readme

## Questions

1. Think about components and how to do the communication. Component diagram
2. How do we store the data
3. How does the user interact with the application. (Create new habits, complete a task, check progress)
4. What is the general flow of the application (Flow diagram)
5. Explain structure, tools for each component (Component diagram?), component interaction
6. Explain analysis and consideration. Capture problem
7. Follow TDD
8. How to structure the application and tech choices
9. Look at Evaluation criteria p6 and formal guidelines p8
10. (Do we need to store a streak, or just calculate when they require?)

## Selection/parameterization and use of programs (software,tools,app)

integral part of the portfolio for conceptual topics and must be documented accordingly

## Assumptions

1. Frequency – string to specify frequency
2. Pre populate Daily, Weekly, Monthly
3. Pre populate 5 predefined habits (1 weekly, one daily)

## References/Bibliography (See guidelines portfolio)

1. <https://www.youtube.com/watch?v=dcsvl3YqAEk&ab_channel=MichaelHadley>
2. <https://drawio-app.com/uml-class-diagrams-in-draw-io/>
3. <https://towardsdatascience.com/timestamp-vs-timedelta-vs-time-period-afad0a48a7d1> (Time frequency)
4. Naming conventions - <https://peps.python.org/pep-0008> <https://google.github.io/styleguide/pyguide.html#316-naming> <https://realpython.com/python-pep8/#naming-conventions>
5. Learning Python 5th Edition – How to structure a program p 671
6. https://fastapi.tiangolo.com/tutorial/first-steps/