Habit Tracking Application

Table of contents

[Overview 1](#_Toc99691688)

[Problem statement / Purpose (What are we trying to solve) 1](#_Toc99691689)

[Use cases (Requirements): 1](#_Toc99691690)

[Selection/parameterization and use of programs (software,tools,app) 1](#_Toc99691691)

[Design 2](#_Toc99691692)

[UML 2](#_Toc99691693)

[Methodology 2](#_Toc99691694)

[Technology choices 3](#_Toc99691695)

[Assumptions 3](#_Toc99691696)

[References/Bibliography (See guidelines portfolio) 3](#_Toc99691697)

# 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 with this.

## Problem statement

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

## Use cases (Requirements):

1. User create a habit that he wants to track with habit details (Habit, frequency)
2. User complete a task (track activity) – He completed the required action for the habit in the specified time period
3. User must complete a task once during the period otherwise he breaks the habit
4. User complete a task for x consecutive periods he establishes a streak (Do we need to store a streak, or just calculate when they require?)
5. User wants to analyse the data
   1. Longest habit streak
   2. Daily habits
   3. Which habits do I struggle with

## Questions

1. How do we store the data
2. How does the user interact with the application. (Create new habits, complete a task, check progress)
3. What is the general flow of the application (Flow diagram)
4. Explain structure, tools for each component (Component diagram?), component interaction
5. Explain analysis and consideration. Capture problem
6. Follow TDD
7. How to structure the application

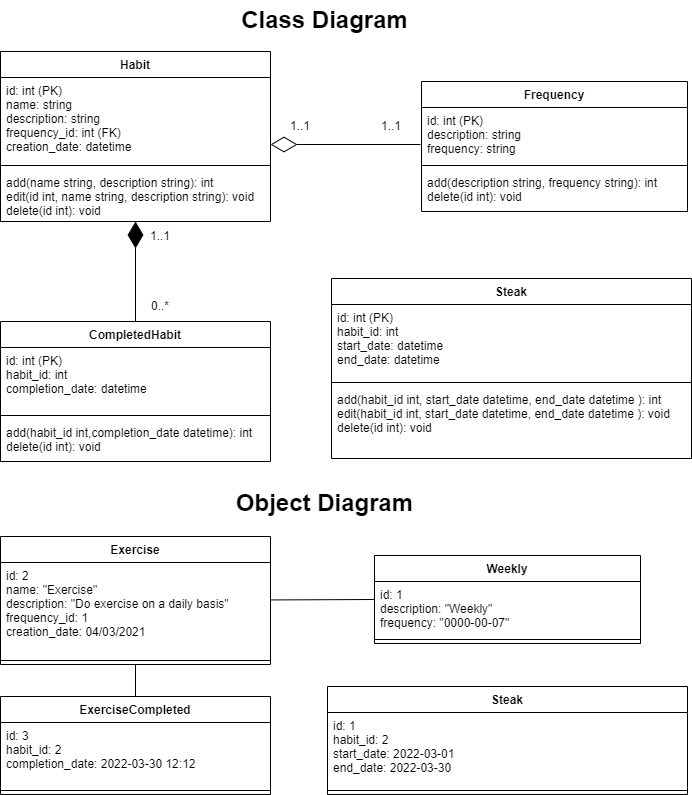
## Selection/parameterization and use of programs (software,tools,app)

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

# Design

## UML

## TODO: Component diagram



## Methodology

What methodology was used to plan or to solve the issue

## Technology choices

* Pytest
* Visual Studio code
* Sqlite – How to store/retrieve
* FastApi
* CLI?

## Assumptions

1. Frequency – string to specify frequency

## 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>