

# Habit Tracker Concept Paper

In this concept paper, I want to explain my idea for a habit tracker. I'll start by discussing the technical concept and then move on to a brief overview of the User Interface.

## I. Technical Concept:

### Class Structure

- **Habit**  
-> will be the main class of the program: Handling all relevant logic concerning checking off habits and breaking habits. This class will also be able to control whether habit is still "in time."
- **database\_modifier**  
-> will update, insert, remove relevant data from database
- **habit\_analyser**  
-> will be the class responsible for analysing the data i.e. finding the highest streak in database or calculating the average streak.

Please note: For readability and organization, I will split the program into two separate files. One will handle the program's logic, acting as the "backend", while the other will solely focus on the user interface.

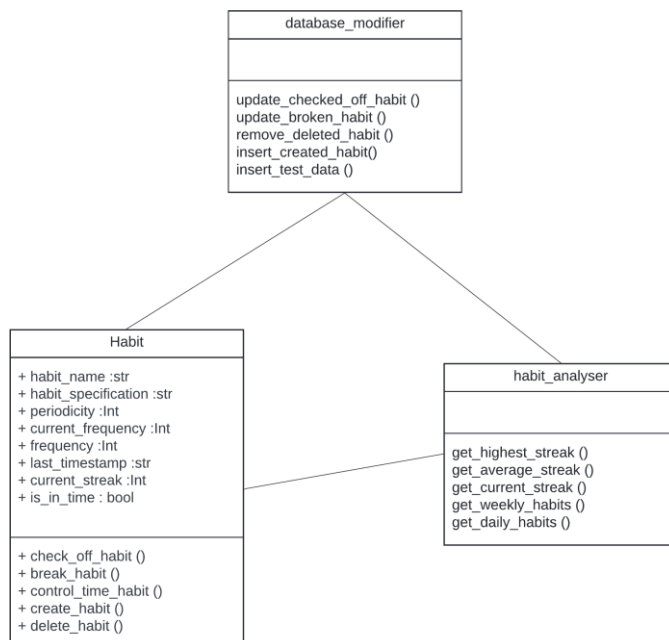


Figure 1: Class diagram

## Modules

- **SQLite3** as permanent Database
- **Datetime** to retrieve and handle dates
- **Unittest** to test the main functions of the program

The Interface will use a simple **input** function in combination with a **while** loop

## Database Structure:

The database will hold all the details about habits, split into two tables. One table, called "habits," will keep track of active habits, while the other, "habit\_history," will store records of past streaks when habits were discontinued i.e. broken. This setup helps the "habit\_analyse" class to find the longest streak and figure out the average streak length.

In the "habits" table, there will be fields like name, description, periodicity and the current streak. It will also save the current frequency and the total frequency, which can be useful for habits like brushing teeth, which might need to be done more than once a day or week. Using timestamps allows users to start habits whenever they want, including even in the future.

Table: habits

	habit_name	habit_specification	periodicity	current_frequency	frequency	timestamp	current_Streak
Format in SQLITE	TEXT   PRIMARY KEY NOT NULL	TEXT   Description of task	TEXT  Daily Weekly	INTEGER   Limit at frequency, default at 0	INTEGER   1=once 2=twice 3,4,5,6 ....	TEXT in YYYY-MM-DD	INTEGER how many times was the habit checked off
Example:	Brush teeth	Brush your teeth twice a day	Daily	0	2	2024-02-02	7

Table: habit\_history

	ID	timestamp	streak_count	habit_name
Format in SQLITE Explanation	INT AUTOINCREMENT   automatically assigned by SQLite	TEXT in YYYY-MM-DD   timestamp when habit was broken	INTEGER   Current streak	TEXT   FOREIGN KEY
Example	1	2024-04-14	5	Brush teeth

## II. USER INTERFACE

After launching the program, users will be greeted with a friendly welcome question to kick things off. They will then see a list of their current habits right away. Since checking off habits will probably be the most common action, I have made it the focal point of the **start page**. Users can quickly mark their habits as completed right from the get-go. Additionally, they have the option to press "0" to access the menu for more functionality.

The **menu** lets the user choose between 1. Insights, 2. Schedule, 3. Add new habit and 4. Delete habit

In the **Insights** section the program will list all the current streaks, the High score streaks and the average streak length.

In the **Schedule** section the user sees a list of the habits sorted by periodicity (daily and weekly) for easy tracking and planning.

In the **Add new habit** section the user can create a new habit. They will be prompted to input the habit's name, periodicity, frequency, and the option to start immediately or at a specific date.

For the **Delete Habit** feature, the program will present a list of all current habits. Users can simply select the habit they wish to delete by pressing the corresponding number for removal.

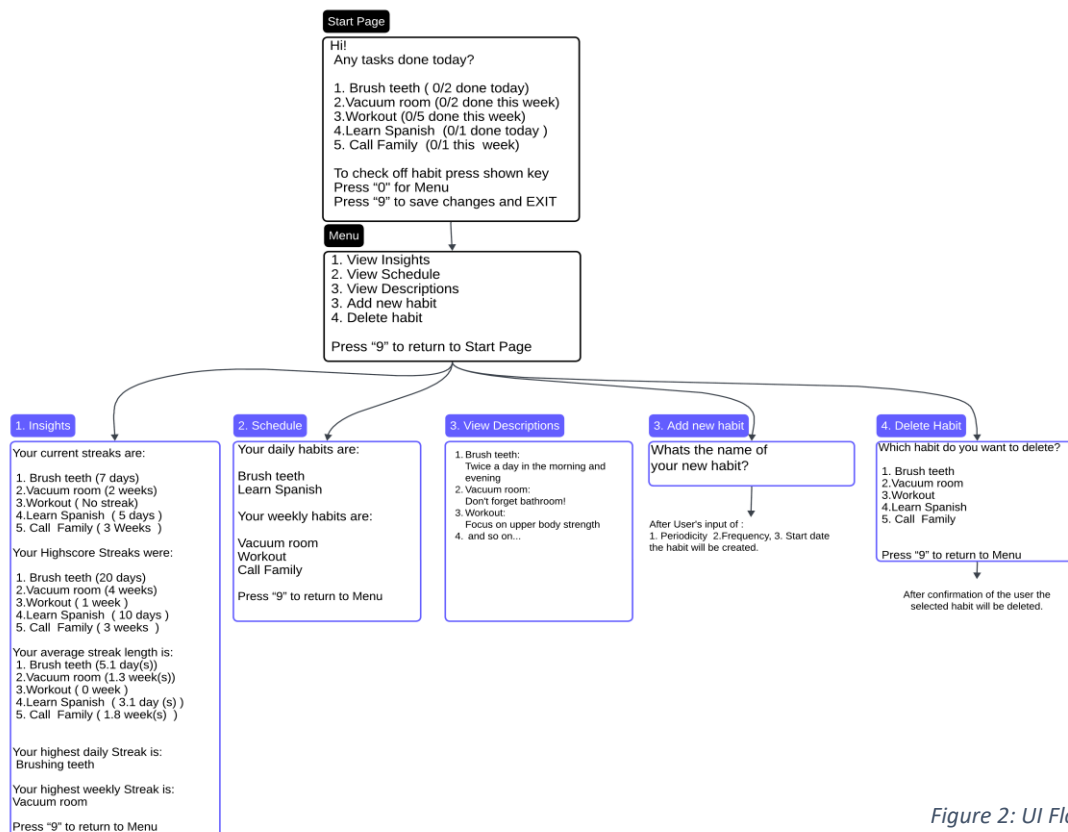


Figure 2: UI Flowchart