Habit tracker

62417 Mobile Application Development with Swift

Sandie Petersen s224323

Stiina Salumets s250088

31/03/2025

Introduction and problem statement

Habits are something that affects all humans to some degree. We all have good and bad habits and most of us have a desire to better ourselves and build new better habits. But how is it done? This has been the subject of many books and talks. But importantly there has to be a trigger for the new habit and it has to be engaging (Clear, 2022)

So *how can we use the functionality of an application to help the user build new habits*.

Analysis

The following section is going to detail the concept of habits and how it is defined in this context as well as what needs to be included in a habit tracker.

What is a habit

A habit is something done consistently at a certain interval or time of day, it is often done subconsciously.
However this is still a wide description therefore in the context of this project a habit is defined as: *something you do at least once a week and maximum once a day.*

What is needed in a habit tracker

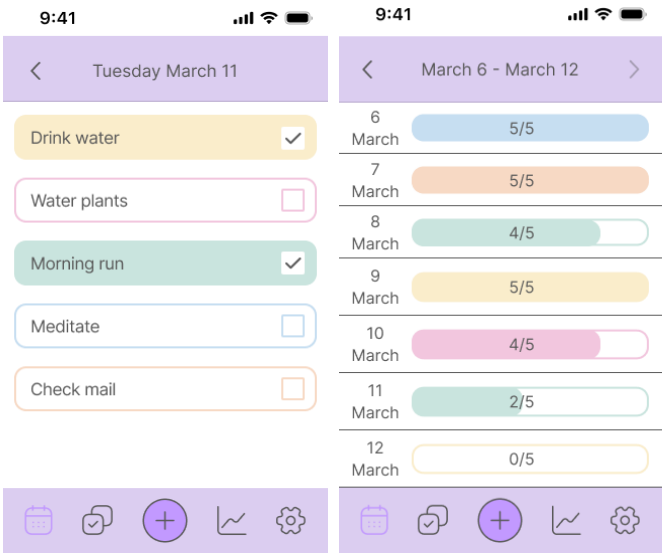
The most important functionality is the ability to create, see, edit and delete habits (also known as CRUD). However the app should also have reminders that will trigger the user to perform the habit. Lastly it would be good if the app is engaging so that the user wants to use it and thereby develop the habits.

Design

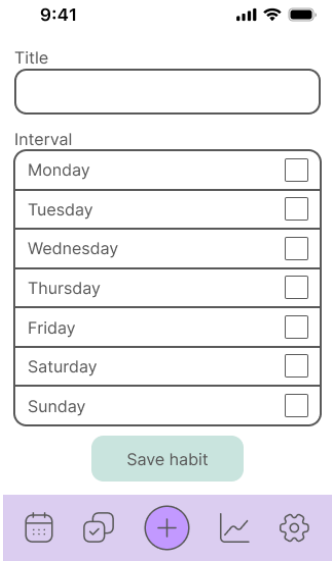
The following sections detail the design of the project, including mockups and diagrams.

Visual design

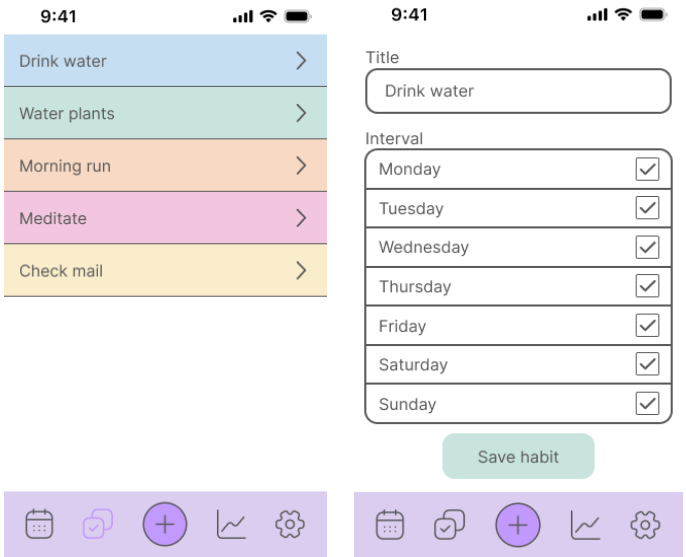
To get a general idea of what the app should look like when completed a mockup was created. So far only the screens currently implemented are included in the mockups.



Firstly the calendar section, here a week is shown where the bottom date will always be the current date. The bar on the right of each day shows the progress of that day. If the user clicks on one of the days they will go to the daily view where they can see the specific habits for that day, only the current date is able to be updated.



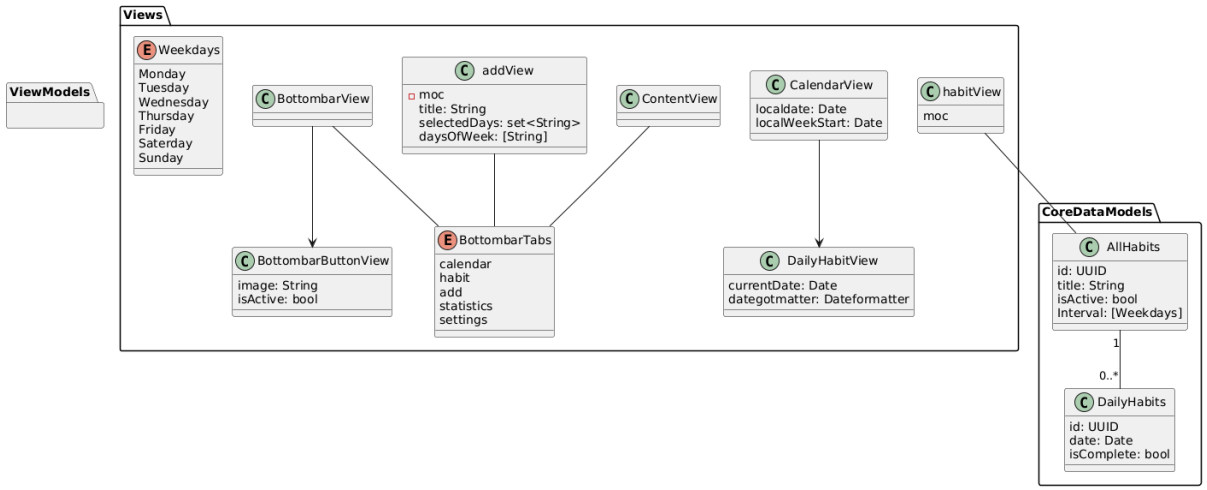
Using the center + button the user is able to create new habits by filling out the form.



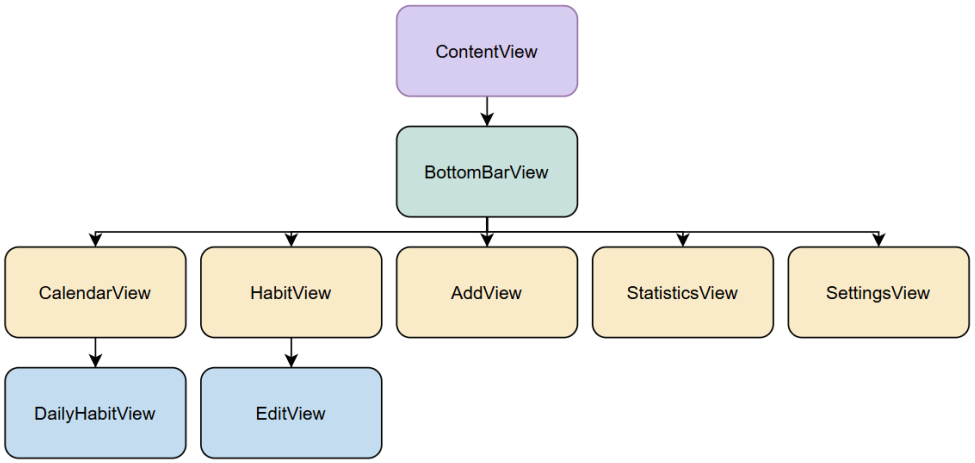
The user can also see a list of all habits by clicking on the second button in the nav bar. From here the user is able to edit the habits, this functionality uses the same form as when creating, however the form will be populated with the habit data.

Code design

To better understand how the code is going to be implemented a class diagram was created. The diagram depicts how the code is currently structured. Looking at this we realise that a restructuring focusing on implementing propper View Models is something we want to focus on.



The second diagram depicts the hierarchy of the interface screens to give a better understanding of how the user will experience the app. As of now we are satisfied with this, but will be adding new screens to it.



Implementation

The following section details the implementation of the app in its current state.

Core data

Core data currently has two entities: AllHabits and DailyHabits. They have a relationship of one(AllHabits) to many(DailyHabits).

DailyHabits have attributes: date, id and isCompleted. This enables us to tie each dailyHabit to a specific date and for the user to mark the habit as complete.
AllHabits has attributes: id, interval, isActive, title. IsActive enables the habit to be “deleted” but the old cases of this habit will still be shown to the user in their history. The interval attribute enables for the user to indicate on which days of the week a dailyHabit object should created

Navigation

The main navigation is based in the bottom nav bar, the bar is inspired by the code in the following article (source), but has been modified to fit the needs of the specific project.
The bottom bar works by switching between views based on what button is selected.

In other areas of the app, navigation stacks are used to create detailed views on lists, meaning dynamic views that are based on which list item was selected.

Views

Currently the views are implemented in a way where they work, however they are quite messy and handling responsibilities outside the scope of what a view should handle. Therefore a major cleanup has to happen where we properly implement viewmodels and helper files and so on.

Results and further development

Though good progress has been made, the app is still very much a work in progress. The basic CRUD functionalities have been implemented as well as core data and navigation but there are many further functionalities to be implemented as well as styling the app to look more like the mockups.

The following list are features to be implemented in the final project (we would like feedback on these)

- Will include
    - Notifications once a day to remind the user to track.
    - API calls to cute cat/dog pictures to be displayed when the user completes a habit.
    - Statistics view
    - Update code structure to have more MVVM structure
- Might include
    - Some settings
        - cat or dog person
        - colors
    - Detailed views on the daily habits with descriptions or sub tasks

Conclusion

In conclusion the current state of the app is great progress but we would have liked to be further ahead. however this has not been possible as we have spent a lot of time dealing with xCode specific issues rather than being able to code. This will hopefully be better in the next phase of the project.

Sources

Clear James. 2022. *Atomic Habits.* Cornerstone Press