

Requirements and Analysis Document for Habba

Grupp 15

**Johannes Gustavsson, Nils-Martin Robeling, Li Rönning,
Alexander Selmanovic, Jian Shin, Camilla Söderlund**

Objektorienterat Programmeringsprojekt TDA367
Chalmers tekniska högskola
Sverige
October 2018

1 Introduction

With a schedule that can vary greatly from day to day either as a student, a businessman or as a forgetful person it can be hard too keep or create new habits.

This android application will keep track of habits the user wants to create and keep. By creating a habit to do either daily, weekly or monthly and having it to show up in the app on those days, you can easily keep track and remember to do them. It uses notifications that are set by the user, to help them remember when to do their tasks even if you don't have the app running. The app also helps motivate the user with the help of streaks and achievements.

1.1 Definitions, acronyms, and abbreviations

- **User:** A user is a person that has an account for the application and therefore can log in and use the application.
- **Habit:** A habit is an action that the user wants to repeat regularly and keep track of. For example a habit could be washing your face every morning or watering the plants once every week.
- **Group:** A user can be a part of a group of users. In this group they have shared habits. For example if the users want to work out two times a week together they can use groups and keep track of each other and stay motivated.
- **Streak:** When a user has done a habit five days in a row, the user will receive a streak next to the habit on the front page. The streak represents the number of times in a row the user has completed a habit. The streak is visible on the front page until the day the user forgets to do the habit.
- **Achievement:** Users can unlock achievements when they use the application, there are two different types of achievements:
 - **Streak achievement:** when a user has performed a habit a number of times in a row they unlock streak achievements. For example doing a habit ten days in a row.
 - **Number of habits achievement:** When the user has created a number of habits they unlock this kind of achievement. For example having a total of five habits.

2 Requirements

2.1 User Stories

1. **Story name:** Creating habits

Description: As a user I want to be able to to create a new habit so that I can keep improving my self.

Confirmation:

- Functional:
- Be able to create a new habit.
 - Have the habit added to the right list.
 - The habit should be connected to the correct user and saved.

Non-functional:

2. **Story name:** Delete habit

Description: As a user I want to be able to remove habits so that I only keep track of relevant habits.

Confirmation:

Functional: - Being able to delete a habit.

Non-functional:

3. **Story name:** Check off habits

Description: As a user I want to be able to check of the habits I have done so I can see what I have left to do.

Confirmation:

Functional: - Boolean isDone in a Habit can be set true or false.

- When checked of it should be moved to done list.

Non-functional:

4. **Story name:** Undo checks

Description: As a clumsy person I wanna be able to undo my 'check off' so that if I pressed the wrong habit it returns to its previous state.

Confirmation:

Functional: - Internal logic for when it's done to be reset to not done.

- Should be shown in their previous category again.

Non-functional:

5. **Story name:** Reset habits

Description: As a user I want the list of daily tasks to be reset each day so I can check them off again the next day

Confirmation:

Functional: - The tasks that was marked as done should be set back to undone daily.

- Streak should be kept when the daily reset goes through.

Non-functional:

6. **Story name:** Get streaks

Description: As a user I wanna be able to get streaks when I've done a specific task multiple days in a row to make it more fun to do my habits.

Confirmation:

Functional: - Keep track of number of consecutive days habit has been done.

- To be shown when streak reaches a set number.

Non-functional:

7. **Story name:** Change themes

Description: As a user I wanna be able to change the color theme on the application to make the user experience more personal.

Confirmation:

Functional: - Being able to change theme.

- The theme should be changed on all pages it's supposed to.

Non-functional:

8. **Story name:** Reminders

Description: As a user I want to be able to set a reminder so that I get reminded about a task and don't forget about it.

Confirmation:

Functional: - Be able to set reminders.

- Reminder should be shown when app is turned off.

- Reminder should be shown at the right time.

Non-functional:

9. **Story name:** Notifications

Description: As a user I want to be able to adjust the way the app communicates (Notifications) with me to fit my personality.

Confirmation:

Functional: - Enable or disable notifications.

- Receive notifications if enabled.

- Different types of notifications depending on set mood.

Non-functional:

Comment: Not Implemented.

10. **Story name:** Share habits

Description: As a user I want to be able to add friends to the app so that we can share our habits.

Confirmation:

Functional: - be able to find friends

- add friends

- share habits with friends

Non-functional:

Comments: Not implemented.

11. **Story name:** Have friends

Description: As a user I want to be able to add friends so I can quickly find them without remembering their username.

Confirmation:

Functional: - Being able to add friend.

- Your friends should be shown under friends menu.

Non-functional:

12. **Story name:** Remove friends

Description: As a user I want to be able to remove friends so I have relevant friends in my list.

Confirmation:

Functional: - Being able to remove friend.

Non-functional:

13. **Story name:** Compete with friends

Description: As a user I want to be able to add friends so that we can compete in who completes their habits.

Confirmation:

Functional: - Add friends.

- Compare achievements with friends.

Non-functional:

Comments: Not implemented.

14. **Story name:** Achievements

Description: As a user I want confirmation that i am progressing through achievements so that I feel encouraged to continue.

Confirmation:

Functional: - Having Achievements to fulfill.

- Receive achievements when goals are reached.

- Being able to see achievements your achievements

- Getting a notification when an achievement is fulfilled.

Non-functional:

Comments: Notifications not implemented for achievements yet.

15. **Story name:** Statistics

Description: As a user I want to keep track of my progress so that I feel motivated.

Confirmation:

Functional: - Access information about done habits over time.
- Put saved information together and show to user

Non-functional:

Comments: Not implemented.

16. **Story name:** Access account

Description: As a user I want to be able to create an account so that if I get a new phone I can have access to my account on the new one.

Confirmation:

Functional: - Create a user account.
- Externally save accounts.

Non-functional: When logged out, should not be able to press back to get into the account again.

Comments: Only local accounts are available.

17. **Story name:** Save data

Description: As a user I want my user account and habits to be saved when I exit the application so that what I've done won't disappear.

Confirmation:

Functional: - All important data should be saved.
- Data should be saved when exiting the application.
- Data should be saved when logged out.

Non-functional: When new types of data is added it should be easy to have it saved.

Comments: Current system is a temporary solution so that functionality that is dependent on the save and load functions will work. A better more sustainable one need to be implemented later.

18. **Story name:** Load data

Description: As a user I want my user account and habits to be loaded so that when I start the application I can continue working on my good habits.

Confirmation:

Functional: - Correct data should be loaded.
- Data should be loaded when logged in.

Non-functional:

19. **Story name:** Have an overview

Description: As a user I want to be able to overlook upcoming and past habits so I can easily take them into account when planning.

Confirmation:

Functional: - Implement a calendar to have an overview.
- Keep information of habit frequencies to show in calendar.

Non-functional: -

2.2 User Interface

The user interface has received less attention due to the focus of the project being to create an object oriented program and the design of the code. In this section the GUI design choices will be presented and explained.

The first part of the application that the user meets is the login screen (see fig 1 (left)). It has a basic layout with two fields with input prompts where the user can enter their username and password. After entering valid credentials and pressing the login button the user is taken to the applications main page. If the credentials are invalid, the user will remain on the log in page.

If the user doesn't have an account there is a button called "NEW USER" that will take the user to a page where they can create one.

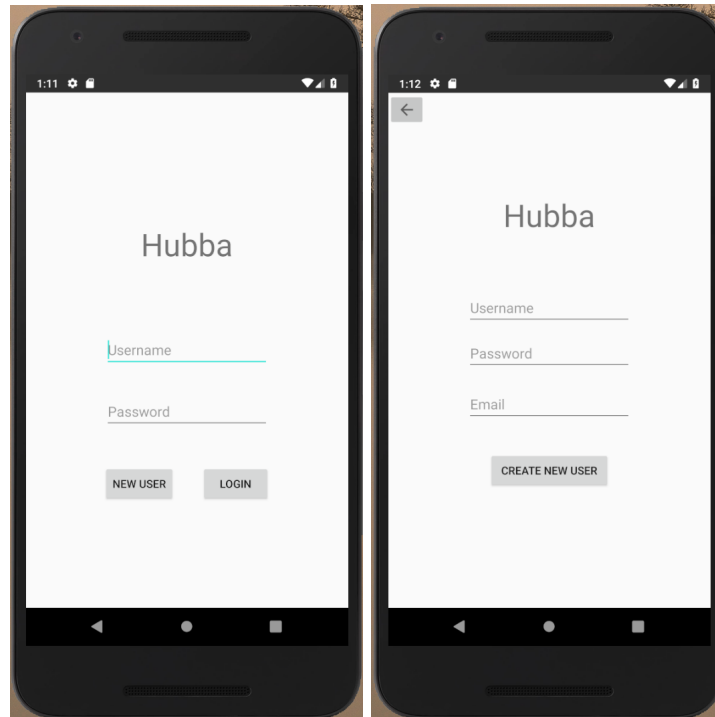


Figure 1: Screenshot of login screen(left) and Screenshot of create new user(right)

To create a new user there are some fields that are necessary to fill in so that the app will have enough information to be able to operate. The username and password are needed to be able to login. The email is required for future functionality, such as recovering a username or resetting a password. At the bottom of the interface a big button that says "CREATE NEW USER" can be seen (See fig 1 (right)). When pressed, an account is created for the user and they can use it to log into the app.

On the main page the habits are shown under the categories "Morning", "Midday", "Evening" or "Night" depending on what state and category the habit has been given, as you can see in fig 2. It can also be placed under "Done" if the user has checked it off. These categories are there so that the user easily can get an overview of what habits should be done during the day with clear separations between the different categories.

On the main page there are three navigational buttons. In the top left corner there is a menu button which takes you too the menu. In the top right corner there is a button to get to a

calendar view of when the habits are to be done and down in the right corner there is a button that lets you create habits on a separate screen. Due to this being an habit application for phones and that most people are right handed the button to add habits was put down to the right for easy access.

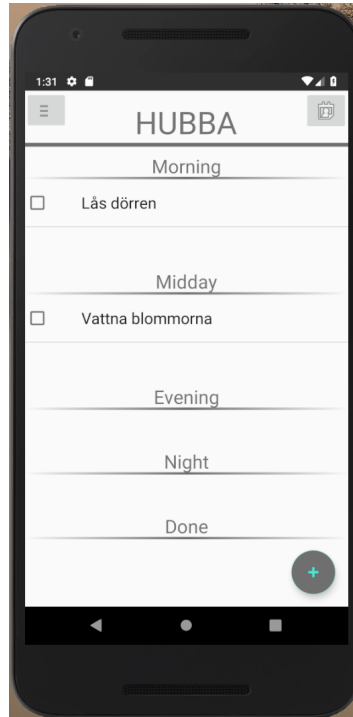


Figure 2: The front/main page with a few habits added

When the button for adding a new habit is pressed a new activity is shown, create habit, that has with only have a few buttons. These buttons impact the GUI and depending on the users choice different options will be shown as you can see in figure 3.

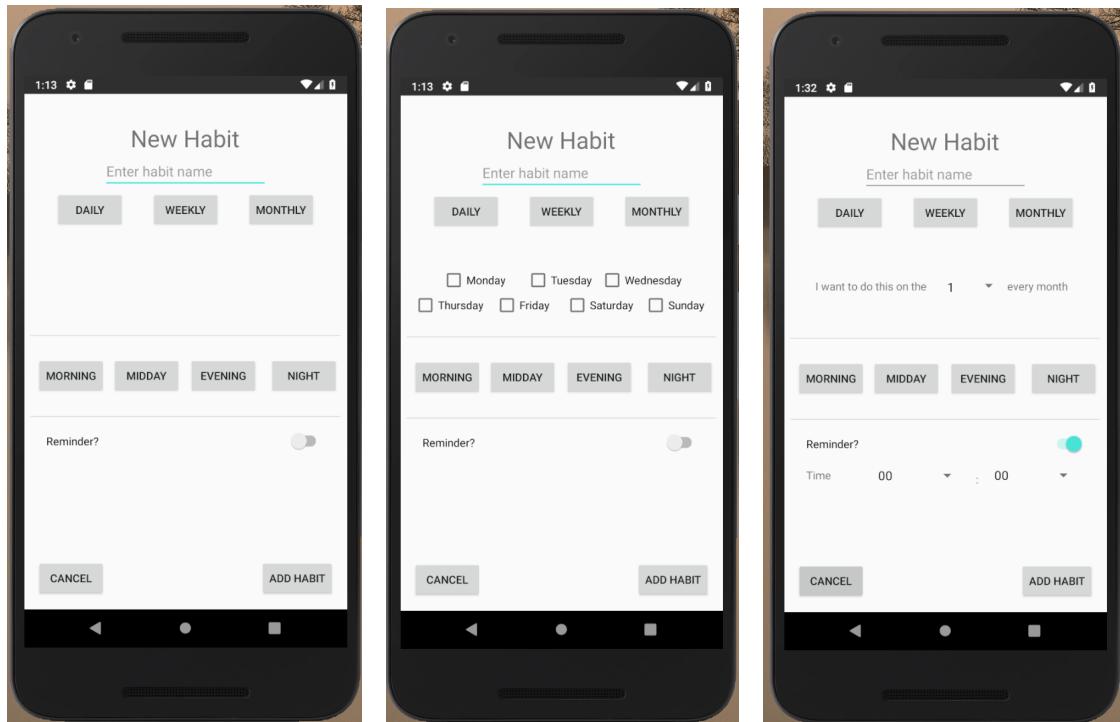


Figure 3: Different layout depending on if daily, weekly or monthly have been chosen in that order

- Daily: If the habit is to be done daily there is no special options to be chosen.
- Weekly: When weekly is chosen the user is able to chose on what days of the week it is supposed to be done.
- Monthly: When monthly was chosen the user are able to chose what date to do it on.

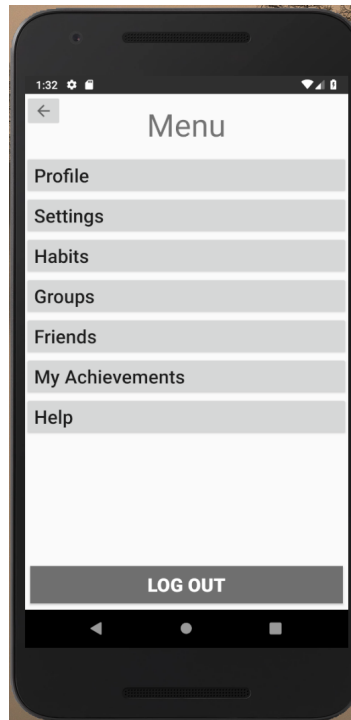


Figure 4: Layout for menu page

When you are open the menu from the front page the menu will be shown as fig 4. From this page you navigate around to different pages. To make the user more sure about the app and log in function, the app gives the user an option to log out with a prominent done button.

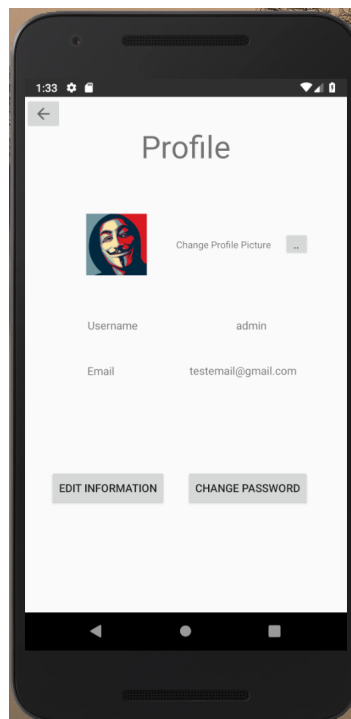


Figure 5: The user's profile page.

On the profile page the user can overview their current profile information, including a profile picture (see fig 5). It is also possible to change the information.

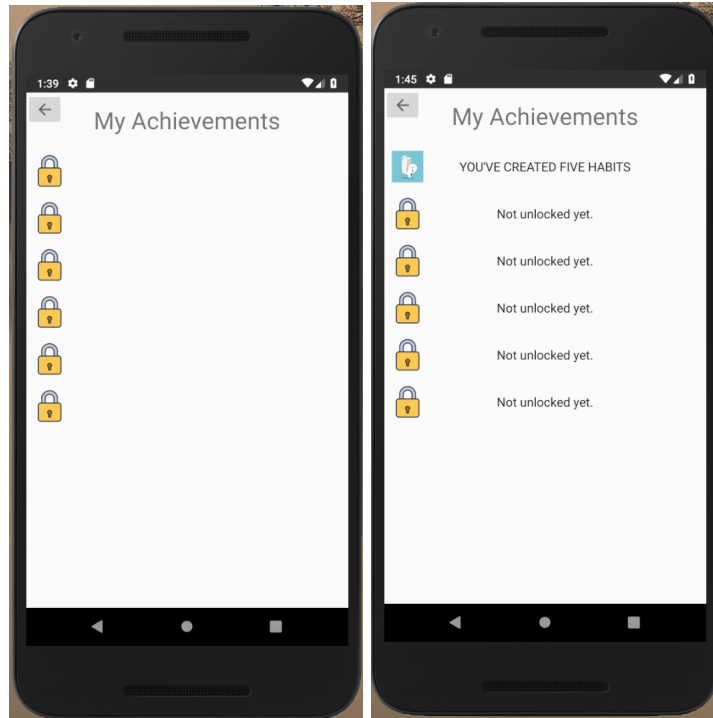


Figure 6: When one achievement has been unlocked

When a user reaches the criteria for a certain (preset) achievement, said achievement is unlocked on the achievements page (see fig 6).

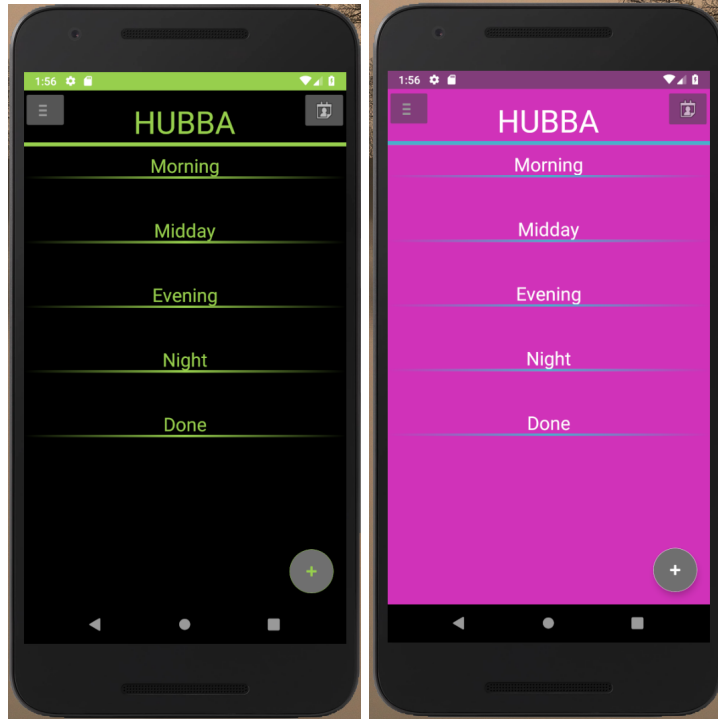


Figure 7: The front page shown in Elite theme on the left, Pink fluffy theme on the right.

A user can choose between two built in themes as shown in fig 7. When a theme is set, all the pages in the app changes appearance accordingly.

3 Domain Model

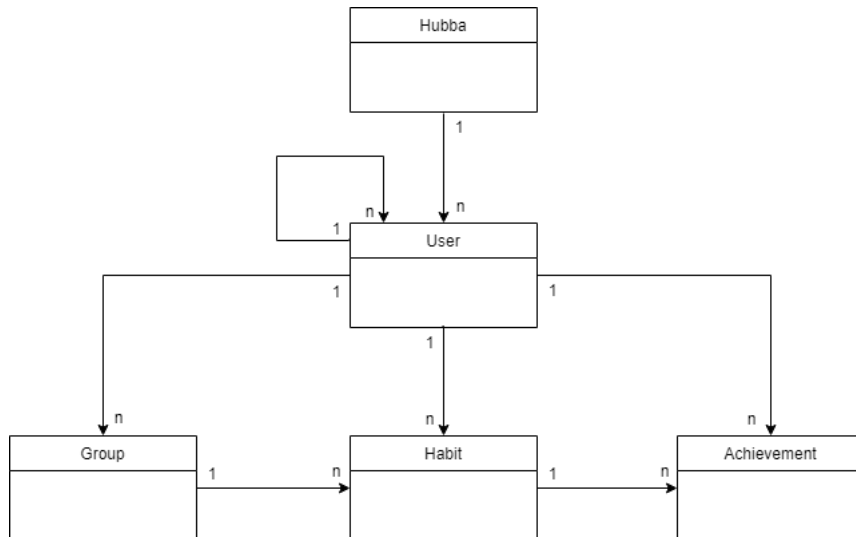


Figure 8: Hubbas domain model

3.1 Class Responsibilities

The different classes that are shown in the domain model have different responsibilities. They all belong to the model module, to which there are more classes, but these are the most relevant ones.

3.2 Hubba

Hubba works as a facade for the model module, which mainly consists of the classes shown in the domain model. All communication between the model and the other modules, view and ViewModel, go through Hubba. Another one of the main responsibilities for this class is to keep a list of all the users, and to know which user is currently using the app.

3.3 User

The user class represents the user and keeps information about the users name, email, password, habits, groups and achievements. It also keeps track of whether the user wants notifications from the app or not, and sound settings. So when the saved user logs in the settings will be the same as they were last time.

3.4 Group

If the user wants, it is possible to create group within the app, and share a habit with the other users in the group. This class is responsible for keeping the name of the group, a list of the users in the group, and a habit that the group shares. It also checks if the habit is completely done, meaning that all users have completed the habit.

3.5 Habit

This class represents a habit, which is what the application is based on. It holds all the detailed information about a habit:

- Title: what the habit is called.
- Streak: how many consecutive times the habit has been performed.
- State: a habit has several different states. Such as time of day, how frequently they should be done, and if it's a single user habit or for a group.

The class also makes sure that the different states can be updated.

3.6 Achievement

To keep the user motivated and to make them continue to build good habits, achievements are added. This class has the information of an achievement, such as name, a target number to reach, if it has been achieved or not and method to evaluate if it's achieved.

References