

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 from day to day as a student, businessman or if you simply are a forgetful person it can be hard too keep or create new habits. This app will keep track of habits the user wants to create and keep. It also uses notifications that are set by the user to help them remember their habits. The app also helps motivate the User with the help of streaks and achievements.

1.1 Definitions, acronyms, and abbreviations

- **Habit:** A habit is an action that the user wants repeat and keep track of. For example a habit could be washing their face every morning or water the plants once every week.
- **User:** A user is a person that has an account in the app and uses the application.
- **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.
- **Achievement:** When users have a number of habits or have performed their habits a number of times they unlock achievements.
- **Streak:** When a user has done a Habit five days in row, the user will receive a streak. The streak is present until the day the user forgets to do the habit.

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 is connected to the user and saved

Non-functional:

2. 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:

- check off habits
- make them go to done list

Non-functional:

3. **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 the previous day goes back to undone
- You'll keep your streak when the daily reset goes through

Non-functional:

4. **Story name:** Set themes

Description: As a user I wanna be able to set a color theme to make it more personal

Confirmation:

Functional:

- set different themes

Non-functional:

5. **Story name:** Undo checks

Description: As a clumsy person i wanna be able to undo my done 'check off' if I press the wrong habit so i don't forget to do it

Confirmation:

Functional:

- check off habits
- undo check

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
- receive some kind of reward when achieved a set number of days

Non-functional:

7. **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
- get reminders

Non-functional:

8. **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

Non-functional:

9. **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:

10. **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:

11. **Story name:** Rewards

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

Confirmation:

Functional:

- save user progress
- receive achievements when goals are reached

Non-functional:

12. **Story name:** Statistics

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

Confirmation:

Functional:

- save information about done habits over time
- put saved information together and show to user

Non-functional:

13. **Story name:** Access account

Description: As a user I want to be able to create an account so 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:

14. **Story name:**

Description: As a user I want my user account and habits to be saved when I exit the application so when I start the application again I can continue working on my good habits

Confirmation:

Functional:

- create account for user
- save data when exiting app

Non-functional:

15. **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 gotten little attention due to the focus of this project being to create an object oriented program and the design of the code itself. But in this section the GUI design choices will be presented and explained.

In general navigation you use touch to navigate around in the app, and with this being an android application we have not added back buttons thus they already exists in the phone it self. This due to it might create confusion if there is two back buttons available.

The first part of the application that the user will meet is the log in screen. This is a very basic layout with two fields with input prompts where the user is to put it's Username and Password. Afterwards you have a login button to take you to the application if your inputs are valid.

If the user do not have an account there is a button 'NEW USER' that will take the user to a page where he/she can create a new one.

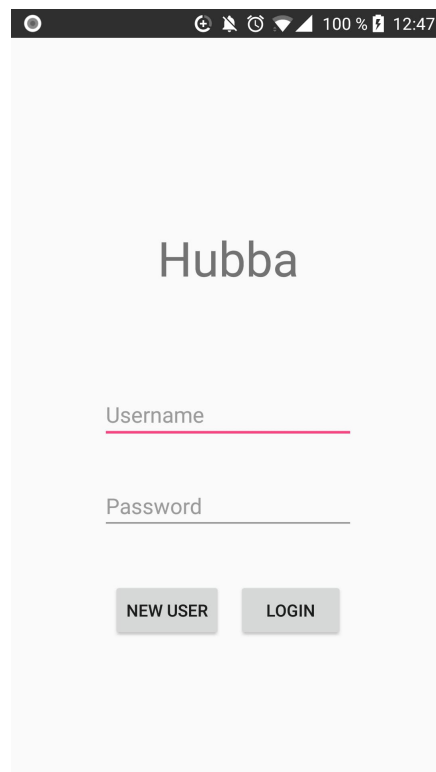


Figure 1: Screen shot of log in screen

When you want to create a new User there are some fields necessary to fill in so the app gets enough information to be able to operate. The Username and Password is needed to be able to login and a Email for when you want to be found by potential friends in the future as a unique ID.

In the bottom you can see a big button to create your new user. (See fig 2)

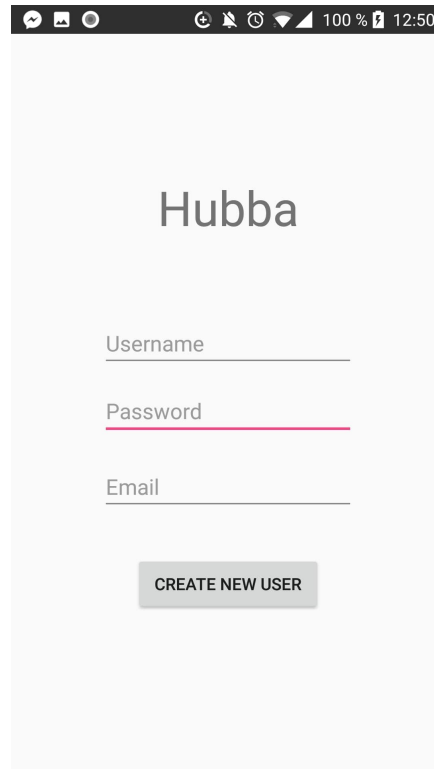


Figure 2: Screen shot of Create new user

On the front/main page (see figure 3) The habits will be shown under the categories Morning, Midday, Evening, Night and Done depending on what state and category the habit have, as you can see in fig 3. This is to more quickly and easily get an overview of what you should do during the day with clear separations between the different categories.

You also have three navigational buttons. Up to the left is the menu button which takes you too the menu, up to the right is the button to get to calendar and down to the right the button that takes you to the create habit screen. Due to this being an habit application for phones and most users will be right handed the button to add habits was put down to the right for easy access.

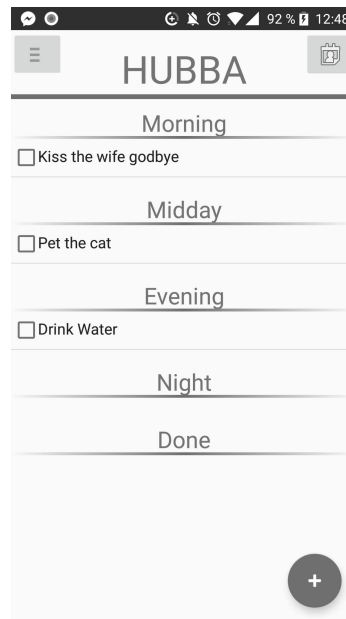


Figure 3: The front/main page with a few habits added

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 fig4.

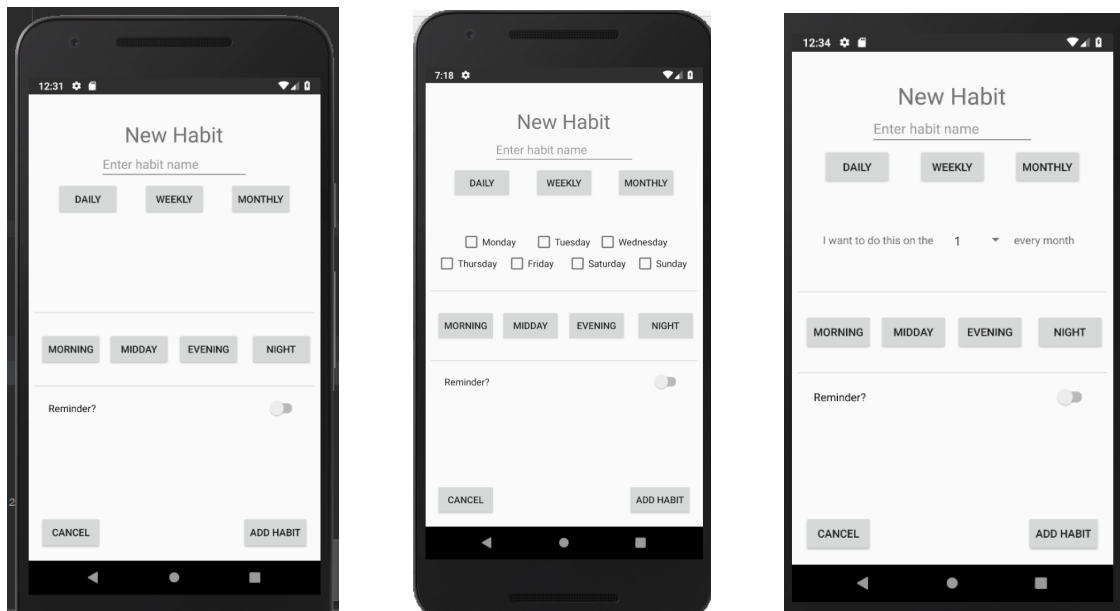


Figure 4: Different layout depending on if daily, weekly or monthly have been chosen in that order

- Daily: If the habit is to be done daily there is no special options to be chosen.
- Weakly: 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.

3 Domain Model

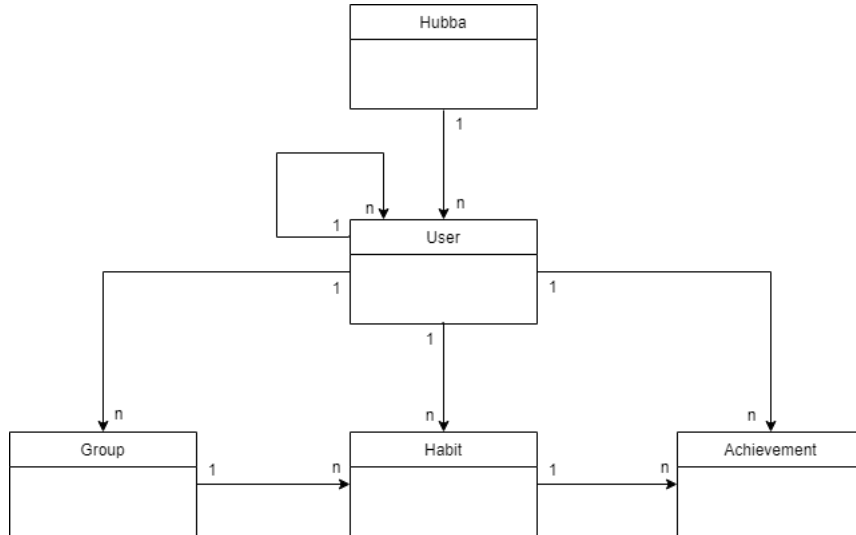


Figure 5: Hubbas domain model

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 is responsible for habits, 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 categories of states. Time of day, how frequently, if it's finished or not, if it's a habit for a group.

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

3.6 Achievement

To help keep the user motivated to continue building good habits, achievements are added. This class has the information of an achievement, such as name, an image, and if it has been achieved or not.

References