System Design 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

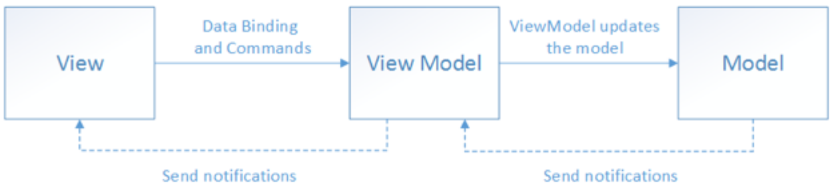
This document describes the design choices made when constructing the android application Hubba.

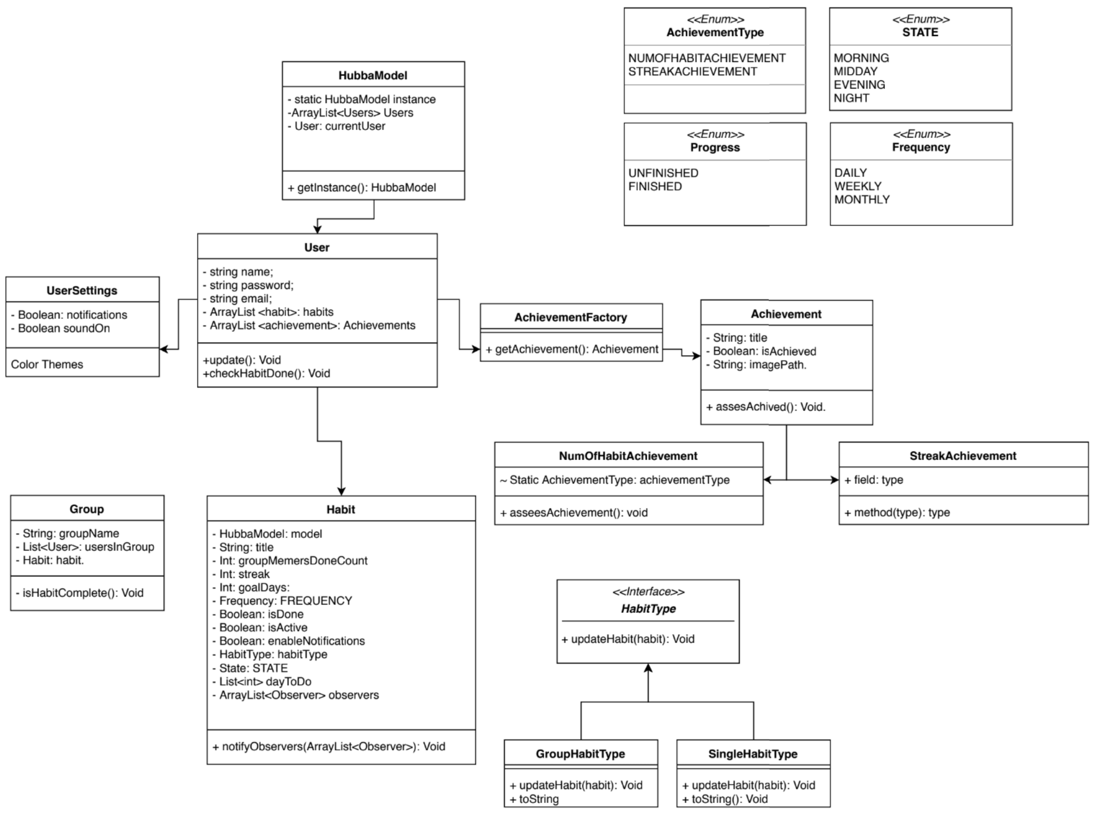
1.1 Definitions, acronyms, and abbreviations

- **Habit**: A habit is an action that the user wants repeat and keep track of. For example washing their face every morning or water the plants once every week.
- **Group**: A user can be a part of a group of users. In this group they can have common habits if they for example want to work out two times a week and keep each other motivated.
- **Achievement**: When users have a number of habits or have performed their habits a number of times they unlock achievements.

2 System architecture

The system is constructed according to the Model-View-ViewModel (MVVM) pattern, an overview of the pattern can be seen in the figure below.





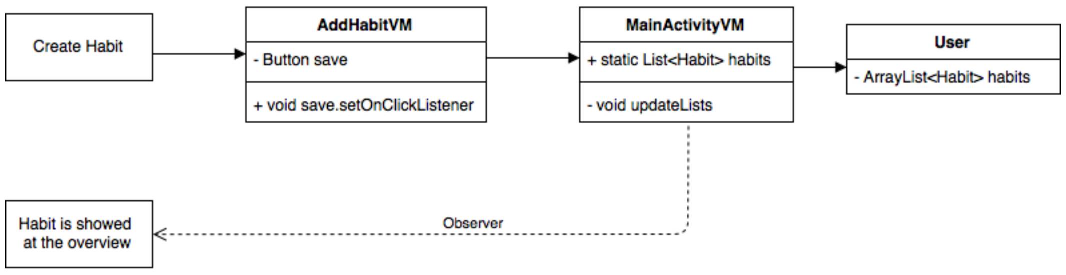
Figur 1: Design model for Hubba

2.1 Subsystem decomposition

2.2 Model

The model package is based on the applications domain model. It contains classes for users, habits, groups achievements and the application. It is also independent of other packages and of android specific implementations.

2.2.1 UML sequence diagram



Create Habit
AddHabitVM
- Button save
+ void save.setOnClickListener
MainActivityVM
+ static List<Habit> habits
- void updateLists
User
- ArrayList<Habit> habits
Habit is showed at the overview
Observer

Figur 2: Diagram over the flow of create a Habit

2.3 ViewModel

The ViewModel package contains all files that keep state for the View and also updates the model.

It also receives notifications from the model to uodate the view when something has changed. When it receives these notifications it sends a notification to the view so that the view updates itself accordingly.

2.4 View

The view package contains all the xml files. The xml files contains names for the parts the user can interact with and builds upp the graphical user interface. This part of the application tells the ViewModel what has happened and then the viewModel handels that information.

3 Persistent data management

4 Access control and security

Referenser