SDD

System design document for LEEP, a timetracker application, Version: 1

Date: 2017-05-27

Author: Linnéa Bark, Eli Knoph, Evelina Stridsberg and Paulina Palmberg

This version overrides all previous versions.

1. **Overview of program**

LEEP is an android application whose purpose is to organize students’ time. It gives an overview on how much time they actually spend on different things. The application allows students to log time when they are doing specific things, to start an activity and then to stop it.

How to log:

1. Choose the category which describes the activity that will be executed.
2. Click on the start-button. To start a “timer”.
3. The “timer” will start counting the time that the activity is spent doing.
4. Press the stop-button to stop the activity.

The application will take the information about the stopped activity and put this in a list where all the activities are stored. Other than that the application can log time when an activity is being executed, the application has a statistics view where a day and a month can be overviewed. There is more functionality in the LEEP application but the two already mentioned is the main (functionality).

**1.1 Definitions, acronyms and abbreviation**

*LEEP* is an abbreviation of the creator's first name acronyms.

An *activity* in the sense of the LEEP application is a collection of a date, start time, total time and a category.

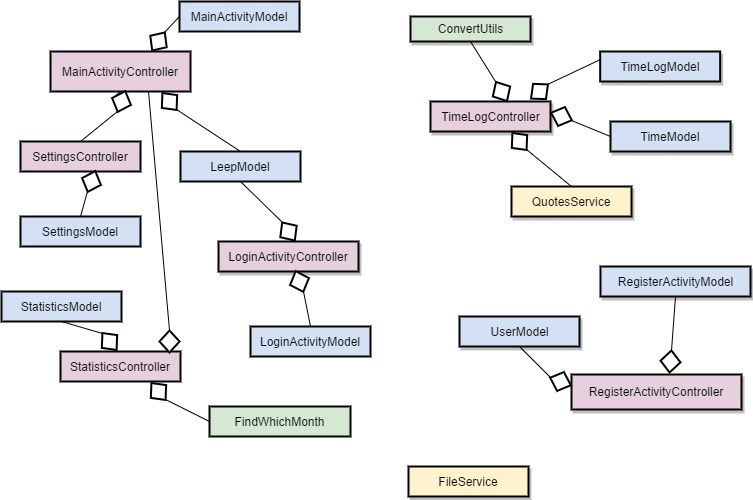
(To) *log* is to save an activity for it to appear in the statistics.

*Timetracker* is the most common word used for these kind of activities. Generally it means that you can track and view what kind of activities you’ve done over the day.

**2 System Design**

**2.1 Overview**

The projects follows the structure of the Model-View-Controller pattern. The Models are in no way dependant of the controllers, and the view classes are totally independent as they are xml-files in Android Studio.

**Class diagram of the LEEP project**

As the user makes an account an User object creates with a username, password and email. When the user logs in, the LoginActivity reacts and compares the username and password in Leep, to (if the log in was successfull) start the activity MainActivity which holds the three fragments; Settings, Time Log and Statistics. The main part of the system is Time Log in which the user can start and stop/save an activity. The user starts an activity and the timer in the Time class reacts and updates the TextView in TimeLog. As the user stops the activity the activity is being saved in FileService where it’s saved into a text file in the internal storage of the android device.

**2.3 General facts about the development tools**

The LEEP android application has been made with the latest version of **Android Studio 2.3.2.** and **Android Studio 2.3.1.** The project has been programmed in the programming language Java.

To create the LEEP project we used computers for the coding and to test the code we used an **emulator** of an Android phone, also that on our computers, but that is optional.

The object Leep contains an User object which handles all the saving of category names, usernames, email and password. The name of the categories can be changed by the user and saved.

The project is mainly built out of Android Studio’s **“activities”,** (which will be explained further down) and **fragments.**

In the application some of the views are using classes called *adapters*. These adapters help different forms of lists to be shown in a specific way. The webpage developer.android.com describes it as “***Adapter:*** A subclass of **RecyclerView.Adapter** responsible for providing views that represent items in a data set” [1]. In these adapters there are more functionality that help to provide views.

SharedPreferences are a built in saving tool in Android Studio. When using sharedPreferences you are saving data in the internal storage space of your android device.

**3. System decomposition**

The LEEP application are created with the purpose to help unorganized students to be more organized, or at least help them to get more knowledge of what they are spending their time on. The application are being viewed as three main views. The three views are described as settingsview, timelogview and statisticsview. Except for these three main views the application contains a view of how to register a user-account and a view of how to log in at a already existing user-account.

The first time the application is started the registerview comes up for the user to create an account. After that the loginview appears so that the user can login. At the loginview the user can choose to click at the radiobutton for “Keep me logged in”, to give the user the choice to not have to enter the password everytime they start the application.

When the user is logged in the timelogview will appear. On timelogview the user can see a motivational quote and either start an activity or start a timer, also called a count down. Before the user start an activity the user have to decide which category that will be logged. You can also start the timer, but when it is down at zero (0) there will be no sound.

The other views can be found from the toolbar located at the top of the screen. When the user clicks on the text “SETTINGS” the settingsview will . An expandablelistview appears with the headers: “CATEGORIES”, “QUOTES” and “HELP”. The headers are called a *group* in Android Studio, and contains *children* (The rows that appears when you click on a group). The children under the group will let the user change name of the category names and the quotes.

At last when clicked on “STATISTICS” , a view with different buttons that will determine which activities that will be displayed at the bottom of the screen. There is a piechart on the middle of the screen that are supposed to show how much of the time that are spent on one category.

**4. MVC in the program**

MVC is in some ways already implemented in Android Studio and its way of structuring classes. The view-classes are the xml-files in which every part of the graphics are implemented. Android Studio is built on activities, (and not the same as used in our application, although they are called the same). Every activity handles input from the user. Hence, every Activity class is a controller class. The controllers have their own model classes in which they as a model class behaves, calls for to make calculations.

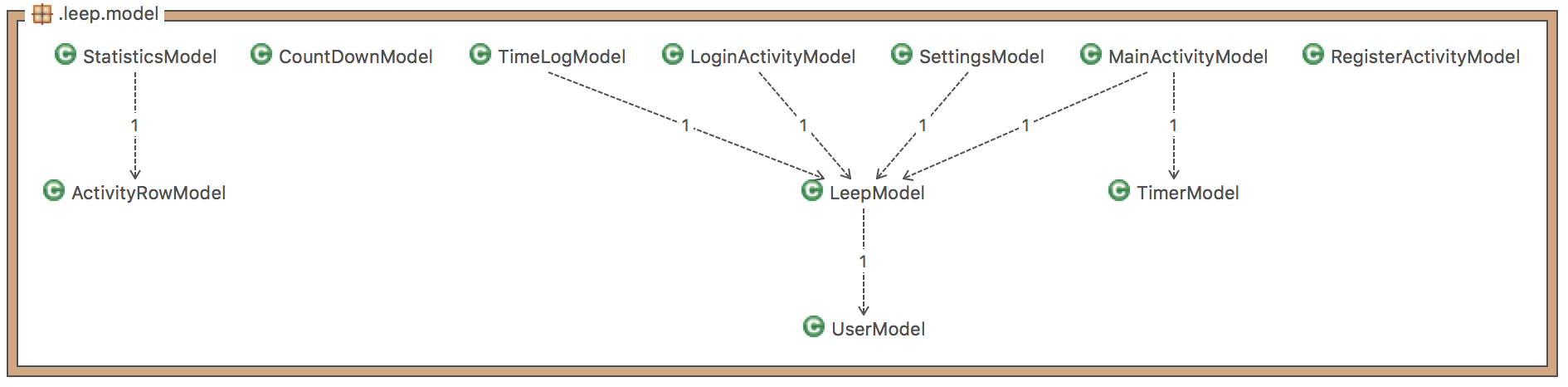
The main parts of the program is

**5. Domain model**

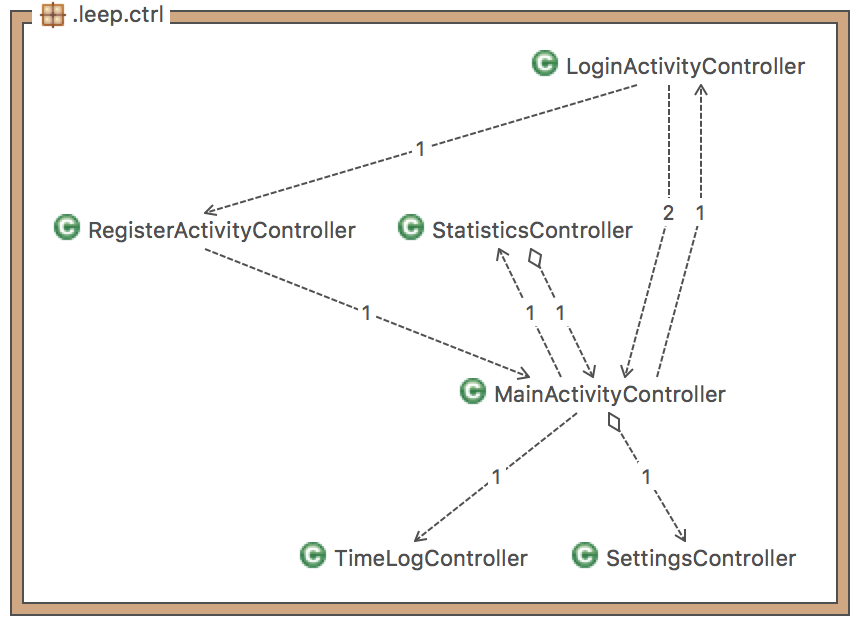
The name of the program is LEEP. When the applications starts the user has to create a user, which can be created an infinity of times. Every user has got three categories which can be a part of an infinity of activities. And every category can log time, (press start and stop) an infinity of times. These are the restrictions and limits in this program. The category names are all specific for just your user log in, because they are editable for each user.



**5.1 Dependencies of Models**



**5.2 Dependencies of Controllers**



**6. The tests**

The tests can be found at github.com at the following link: github.com/linneabark/komigen/app/src/test/java/edu.chl.leep

**7. Persistent data management**

The data stored in the project is account details and activities. The account details such as username, password and email is stored in **Android’s Sharedpreferences**, which saves in the internal storage of the android device. Activities are also saved with SharedPreferences but with a list instead of single element, also in the device’s internal storage. The data is held and retrieved in SharedPreferences, the methods can be found in the FileService class and LEEP class.

**8. Access control and security**

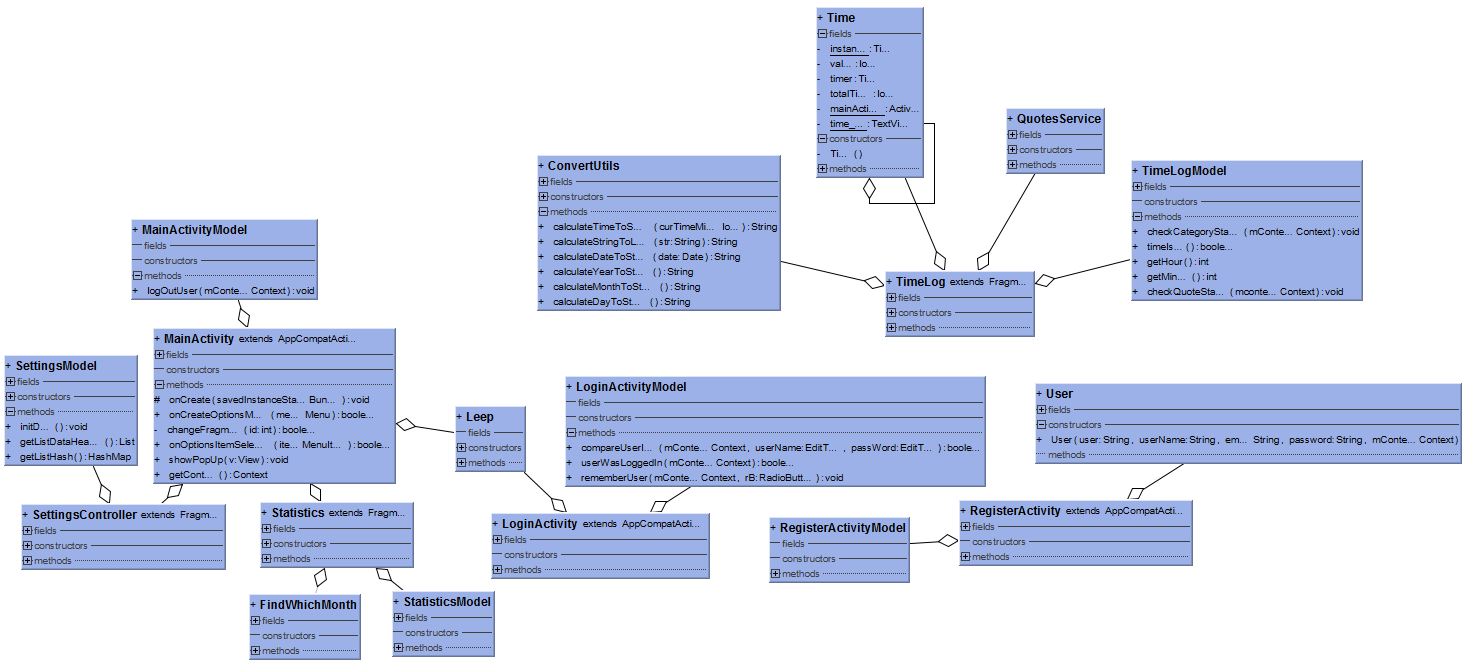
The LEEP application do not have an admin account. The main purpose of the architecture of this application is to be able to have several users and to save information separately. The application is built to use on different android mobile devices. As a user you have to create an account to start using the application. At present there is no way to communicate with another user on another android device, or even other users you have created at your own device. They are all stand alone. But it’s open for modification.

**9. References**

[1] <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html> (Acessed 28th May 2017)

**Repository for project:**

https://github.com/linneabark/komigen

(UML-diagram)