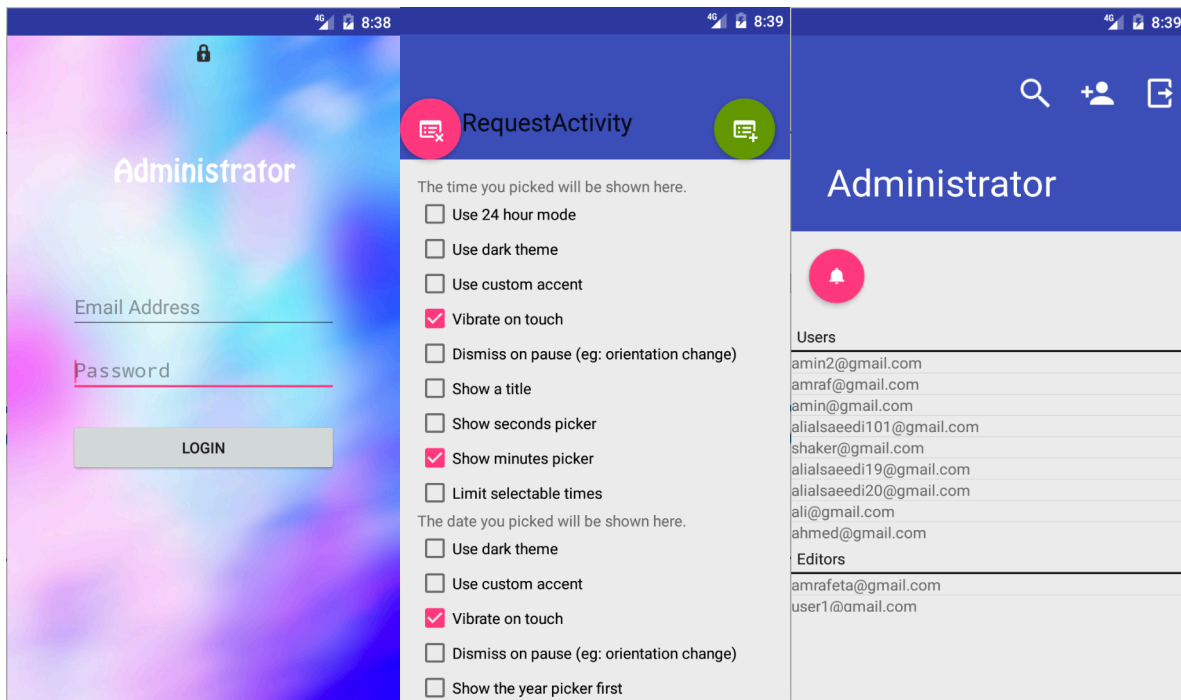


Admin

Prepared by: Mohammadamin Amjadibigvand

Neptun: F5XM1U

December 1, 2016



EXECUTIVE SUMMARY

Overview

The administrator application should be able to give access to the users of it, to log in to the admin account and after that, it should have an option to create the editor account with the specific admin account. The administrator should check if there is an update request from the editor. If there is an update request from the editor the administrator decides whether he publishes or rejects it.

Solution

We can only create the admin account by adding it directly and manually from online web support(firebase).

But it can create editor account for editor application and it can also change the node of the new settings which have been requested by the editor to another node for publishing it , so that the user can read (get) it .

Design

If the user opens the application for the first time , or the user previously has logged out then the first page the user will see the login page .Otherwise the application will just show a progress dialog instead of the log in page ,then in any of the above cases we will be directed in the Main page.

In the main page we have view options such as checking notification for updating the settings, adding the editor ,login out and searching the user .

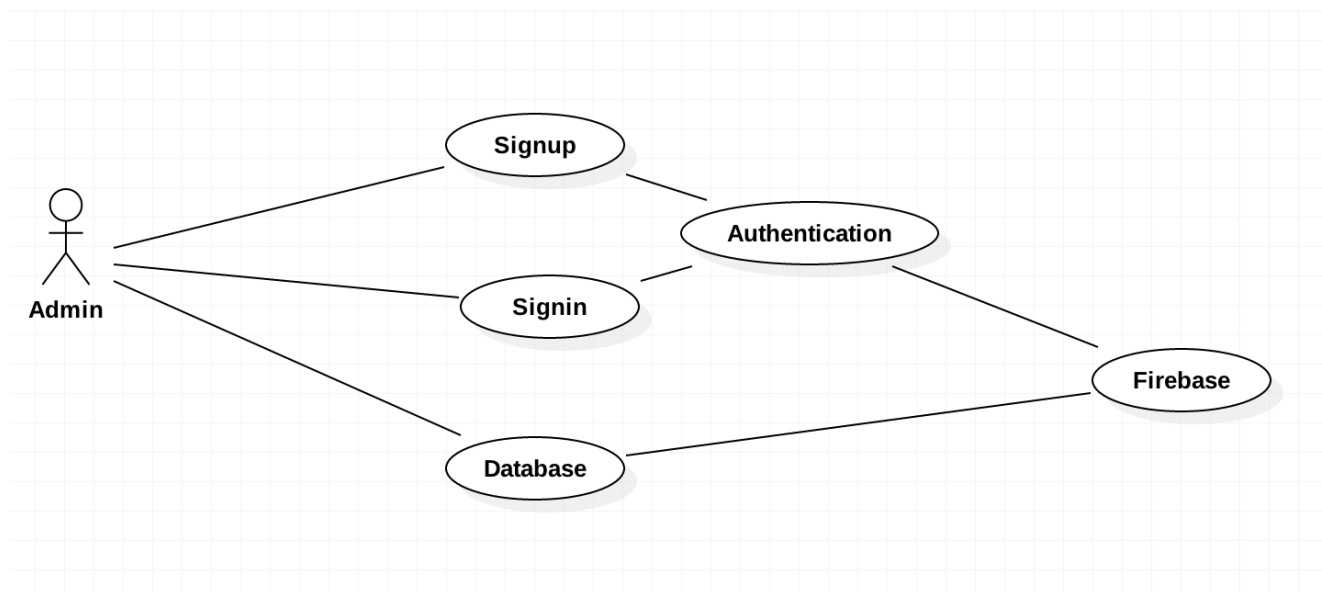
USE CASE DESCRIPTION

Use-case name	Signin
Short description	Will check the input information of the firebase and let the user sign in
Actors	Admin
Scenario	Check email and password and gives the user unique id

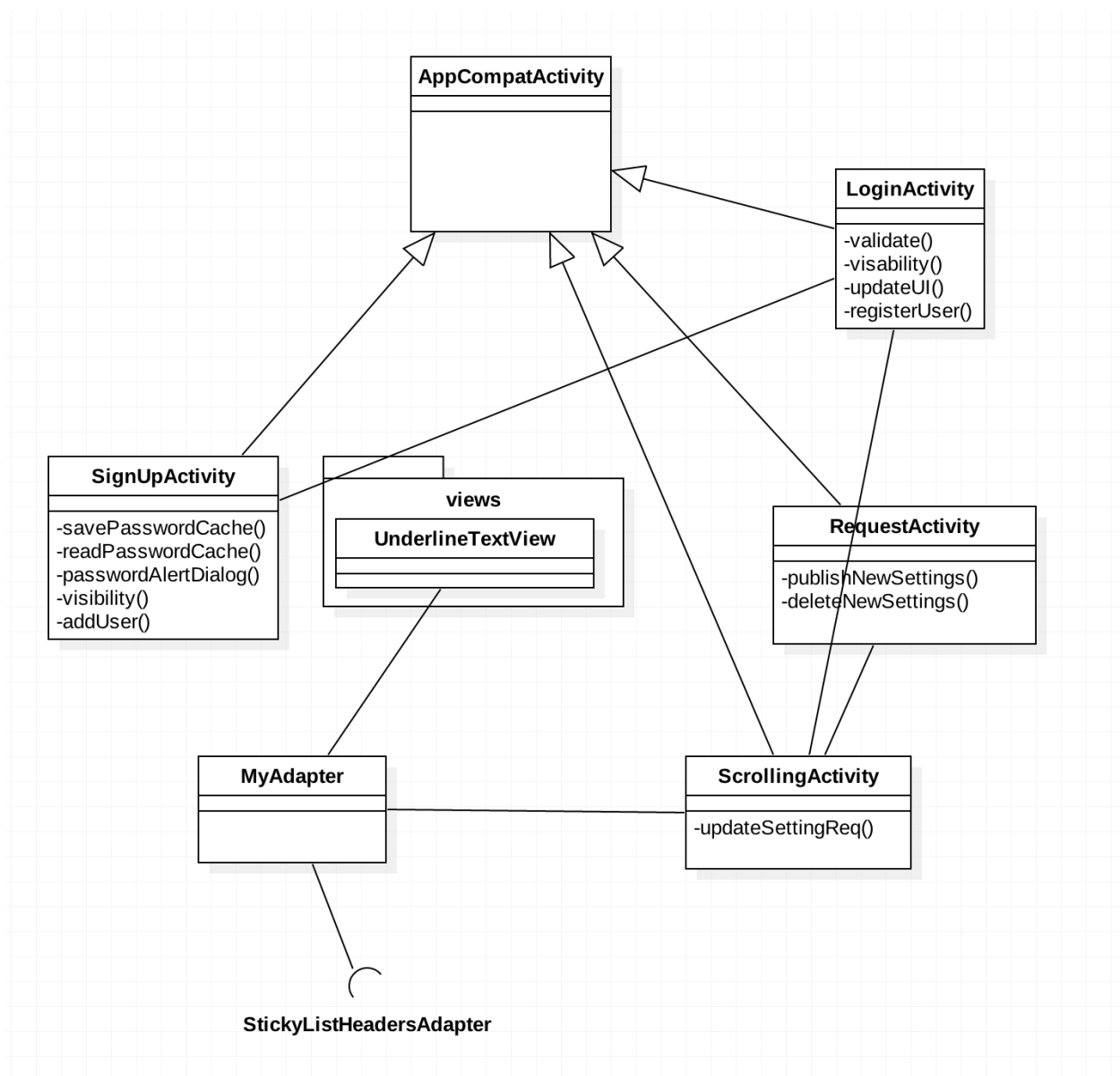
Use-case name	Signup
Short description	Admin will signup the editor by the given information
Actors	Admin
Scenario	At the time of signing up the editor ,the admin will give the editor a tag

Use-case name	Database
Short description	It is used to manage the settings and getting the list of users and editors
Actors	Admin
Scenario	The user will get the setting under the child of the editor , either it will delete it or move it to the general setting , and check the database for the users and editors to print the list of them

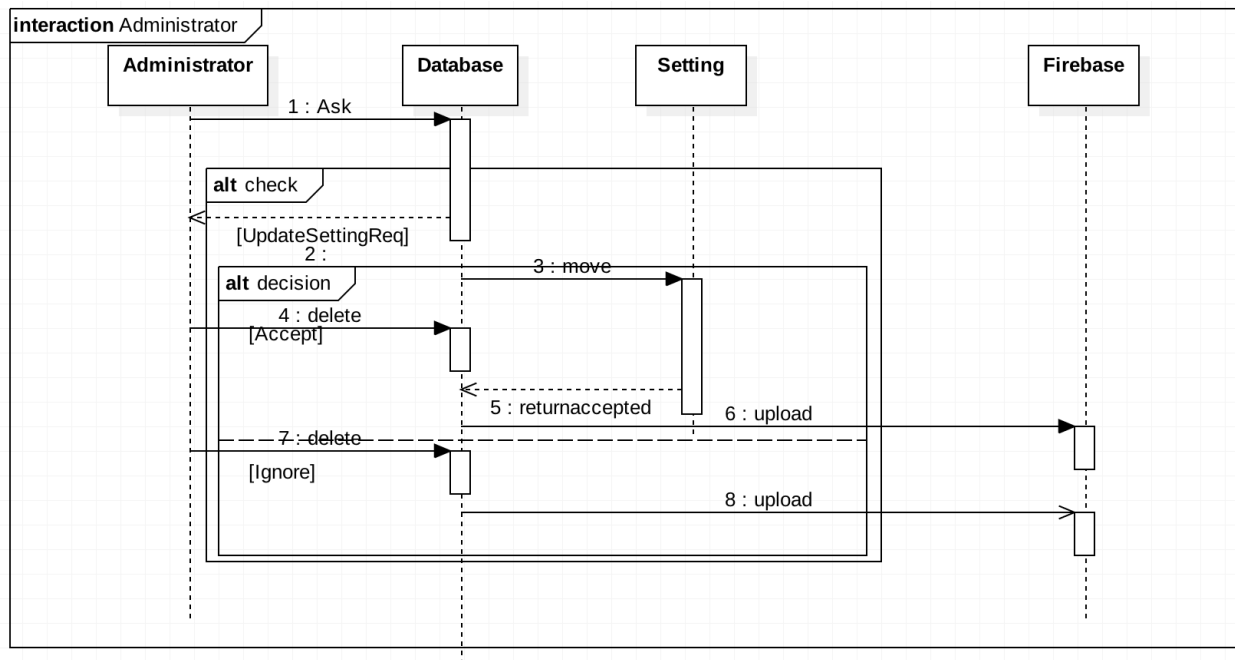
USE CASE DIAGRAM



CLASS DIAGRAM

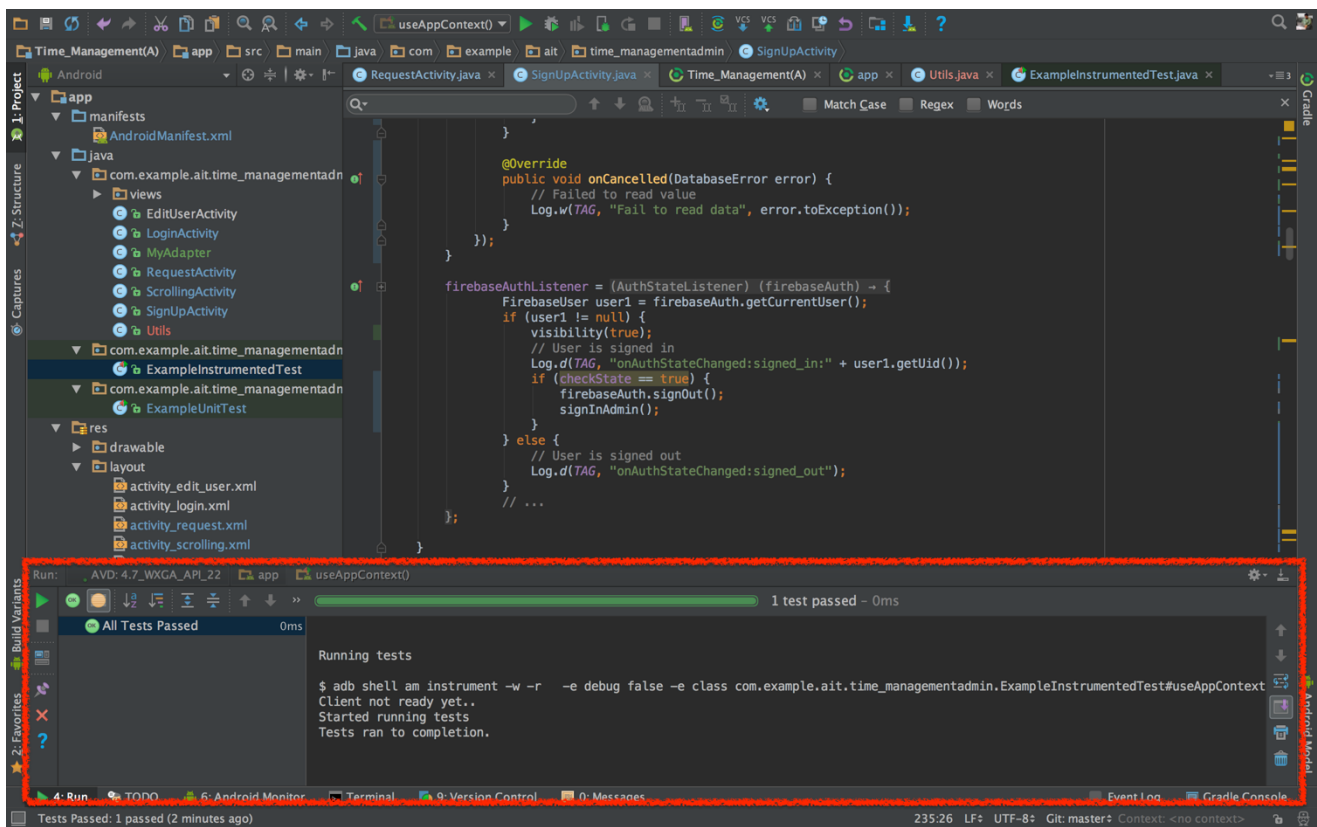


SEQUENCE DIAGRAM



DESCRIPTION OF THE UNIT TESTS

I created a class called Utils which class I am testing through the JUnit(4). What I test is the functions of this class. One of its functions is the validate function which has to test if the email and password are correct or we get any errors on them. Using the newest version of Android (which is 2.2) we can test the functions through the Espresso testing framework with automated UI tests. Local tests units are easier to test because we don't need mocking ,on the other side for Instrumented unit tests we need to mock any database(or prevent any webpage) where I can check easier my functions.



```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:24.1.1'
    compile 'com.google.firebase:firebase-core:9.8.0'
    compile 'com.google.firebase:firebase-database:9.8.0'
    compile 'com.google.firebase:firebase-storage:9.8.0'
    compile 'com.google.firebase:firebase-auth:9.8.0'
    compile 'com.google.firebase:firebase-crash:9.8.0'
    compile 'com.android.support.constraint:constraint-layout:1.0.0-beta3'
    compile 'com.android.support:design:25.0.0'
    testCompile 'junit:junit:4.12'
    compile 'se.emilsjolander:stickylistheaders:2.6.0'
}
```