Developer Manual

Getting started:

1. Visit our repository at https://github.com/HampusLilja/AndroidProject and download the files.

2. Download and install Java jdk from http://www.oracle.com/technetwork/java/javase/downloads/index.html

3. Open Eclipse(can be downloaded from http://www.eclipse.org/downloads/) and choose file -> import -> Existing project into workspace ->browse, select the folder in which you saved the project files and press ‘ok’. Make sure that the checkboxes that will show up on the screen are filled in and then press ‘Finish’. The project should now be ready to work with inside Eclipse.

4. Make sure that the project is using Google API, this is necessary for the application to work, since it uses methods from com.google.android for both GCM(explained later) and the map.

5. To get the map to work correctly on the emulator or if you load the application into your cell phone you will have to get your own api key for google maps. On https://developers.google.com/maps/documentation/android/mapkey you will find out how you can get your key. When you have your own key you have to go to res/layout/activity_map.xml and paste your code into android:apiKey=”*******” (instead of the *-symbols).

6. You will have to set up your own server if you want to send and receive messages.

Setting up the server:

Visit https://github.com/HampusLilja/AndroidProject/tree/master/server-package and download all the files. Notes on how to install the server can be found in the README.txt file. In the server-package there is also included a test web interface

Code Structure:

For information about classes, methods and their relationship please view the UML diagram provided in our repository(https://github.com/HampusLilja/AndroidProject/blob/master/doc/AndroidProjectUMLetc.pdf)

Application:

The messages and the user settings are saved in the application(except if you change nickname, this will be stored in the server).

GCM:
GCM is short for Google Cloud Messaging and is used to send data between the application and the server.

Server:
The server links the nickname chosen by the user with the registration ID sent from the users android device and save them. The users' location(latitude and longitude), which(if any) room the user is currently in and a list of all the users in all the rooms.

Documentation:
All the information about how the project was created and how the work was completed is uploaded on https://github.com/HampusLilja/AndroidProject/tree/master/doc.

Testing:
To run the tests you have to right click on the TestShutApp project and select Run As -> Android JUnit Test. Unfortunately some of the tests are depending on the server to be up and running. We have written some manuel tests that can be found on https://github.com/HampusLilja/AndroidProject/blob/master/doc/test.pdf.