CTHmaps: Developer Manual

A guide and overlook of the CTHmaps android application.

Getting started:

1. Clone the git repository from https://github.com/henke101/CTHmaps.
   to clone our git repository do following:
   Run following code in command line:
   $ git clone https://github.com/henke101/CTHmaps
   This command will clone the git repository into the current directory in terminal

Dependencies:

1. Android SDK (API level 16 with Google APIs Add-On)
2. a virtual Android Device
3. Java 6 SE development environment.

Scrum:

We are developing the project CTHmaps according to the agile software development method scrum. This means that we have a product backlog which contain all features, requirements etc. we want to implement for our project. Every week we have a sprint meeting where we decide what all members of the team should do. For more information about Scrum see https://scrum.org/.

How to use git:

- **Branches:**
  Developing of a new feature should be done in a local branch. If other persons need the code from your branch but you are not ready to push your feature to the remote master branch it is best to push it to a remote branch different from the master. Before features developed in a branch is pushed to master it should be merged into the local master branch, which should be up to date by doing a pull if necessary. When adding code to master it should not contain bugs and the feature you are working on should be finished.

- **Commit message**
  When committing the message should be as short and descriptional as possible. If possible it should be summarized in 72 characters.

Ant

After you have cloned the project from Github repository we recommend you to install ant. Then you can build and test the project from the command line. If you don´t have Ant installed you can visit Ant's homepage https://ant.apache.org/ for more information how to set it up.

Building and installing

A build.xml file is included in the root directory which may be used to build the project and also run the test from the command line assuming you have Ant installed at your computer.

To build the project run following command line:

```
$ ant clean debug
```

To build and install CTHmaps.apk on an Android device run following command line:

```
$ ant clean debug install
```

For more information what you can do with Ant run following comand line:

```
$ ant -p
```

Release procedure

When creating a new release of CTHmaps application it is important to do the following:

1. Create a release note with known bugs and limitations of the application.
2. Create a git tag which contain the release version number.
   To create a git tag do following:
   From directory where your repository is found run following:
   $ git tag -a version number (for example v.1.0) -m "comment".

Editor

You can choose whatever editor that suits you but we recommend you to use Eclipse.
For more information how to install Eclipse visit: http://www.eclipse.org/downloads/ and download the latest version.

Tests

Automatic tests are included in a separate project which are connected to CTHmaps. There are two ways of running the automated test. Both cases requires the emulator (AVD).

1. To run included tests using ant, go to the the command line and from CThmaps test directory run following command line:

```
$ ant test
```

To make sure running the newest application run following command line before running $ ant test:

```
$ ant clean debug install
```

To generate a test report from emma you can run following command line:

```
$ ant clean emma debug install test
```

After running this command you will find a coverage.html file in directory bin.

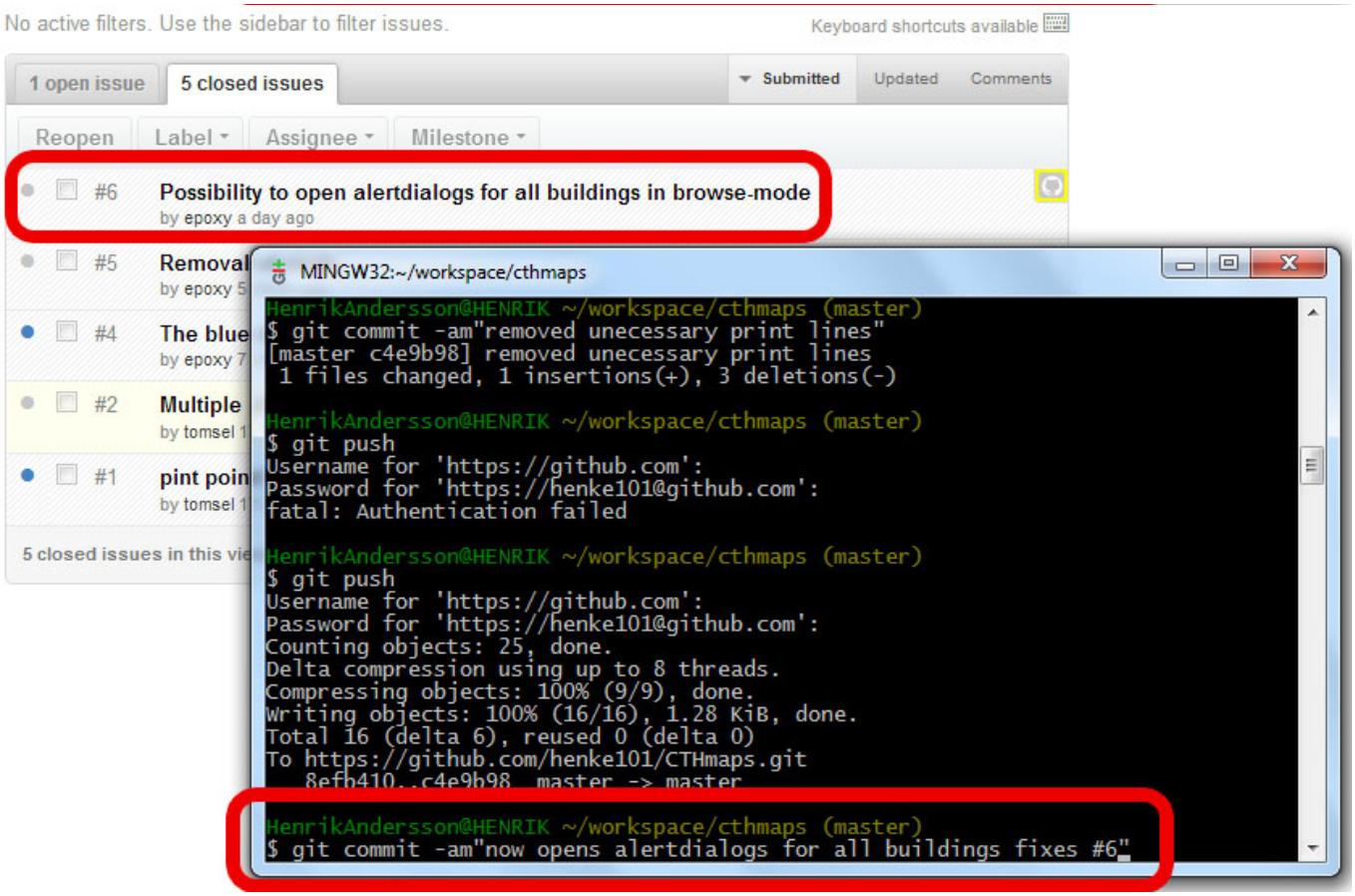
2: The other option is to run the testsusing Eclipse
Right click on the test project and choose Run As -> Android JUnit Test

Bug system

When finding bugs in the code we are using the githubs bugsystem. Of course the best way is to solve the bug directly but if problem occurs use githubs bugsystem.

How to use Githubs bugsystem:
First, go to https://www.github.com and choose our repository, “CTHmaps”. Press at the button “Issues” and create a “new issue” mark the bug label, write a title and a comment, and press submit new issue.
See picture below:

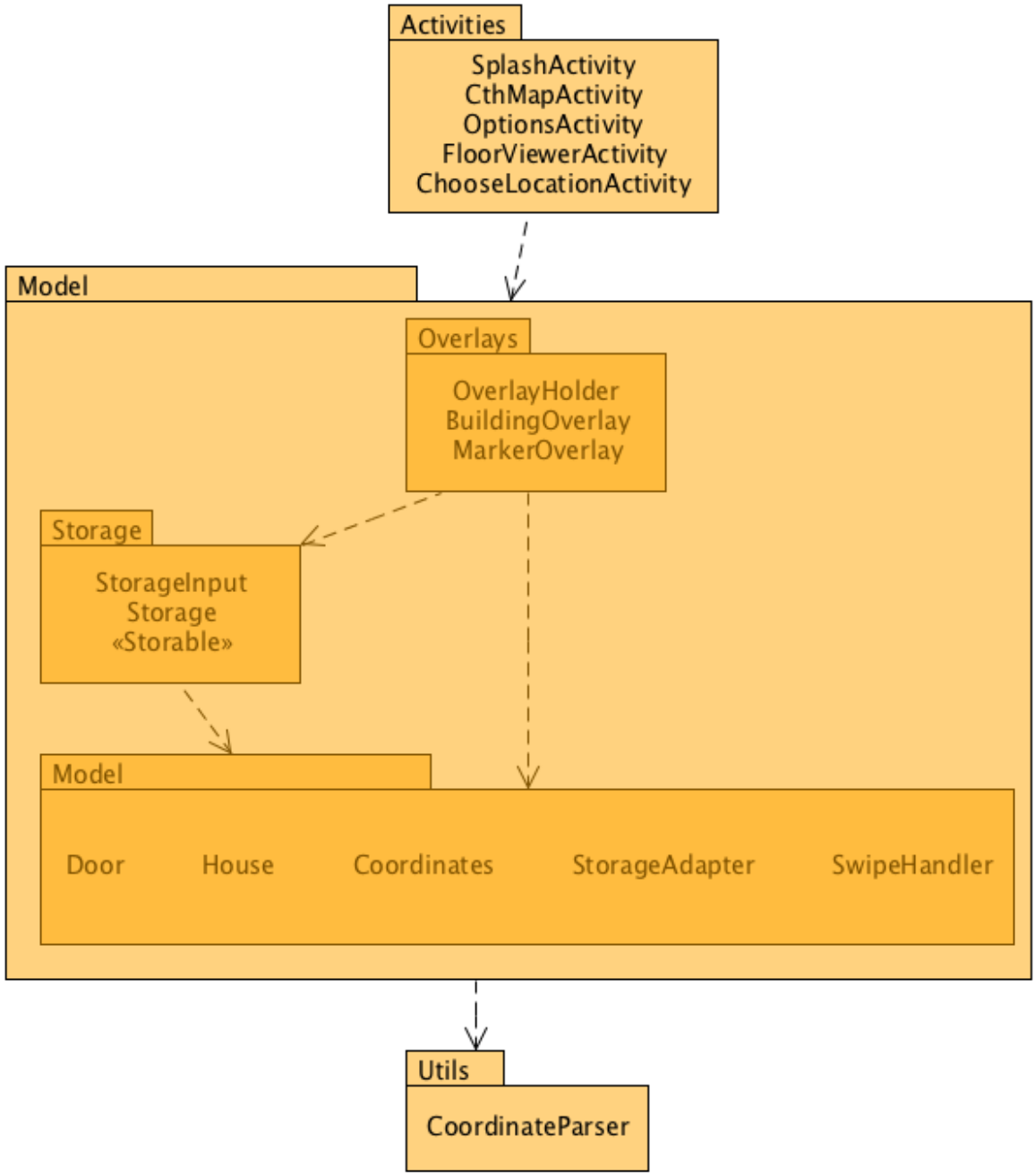
When you have solved a bug, run following command line:
$ git commit -am “message (finish you message with following)-> fixes #bugId”
You can find the bug Id in the tab issues at GitHubs homepage.



Architecture

The package structure is organized according to different areas such as activities in package activity and the database in package database and so on. Help methods are found in package utils. We have two main packages: the activity and the model package. In model we have different subpackages such as database and overlays. An overall architecture of our package

structure can be found in the picture below. For more information about the project see next page and Architecure specification



Package storage

The database is backed with an SQLiteDatabase that gets its input from a .txt file that can be found in the directory assets.

Package overlays

Everything from classroom icons to destination flags that can be seen on the map exist in overlays. These overlays are all placed out and handled in the OverlayHolder class.