Developer manual

This document contains of a start-up guide for the developer for further development of the application.

The guide contains five major parts

- Getting started
- Project structure
- Storage
- Test specification
- Release procedure

1. Getting started

Get access to the application project

1. Open your terminal
2. Change directory to your workspace (cd workspace location)
3. Write git clone git://github.com/sofiase/ReadyForAPet
4. Open Eclipse, Import, General, Existing projects in workspace
5. Select ReadyForAPet

Dependencies

- Java development environment with Android SDK 4.2.2
- An Android device or virtual Android device within the Android SDK target
- Libraries
    - Google APIs (Android 4.2.2)
    - Android Dependencies

Android API level

Minimum:	10
Target:	17

2. Project structure

The project architecture has been built on inspiration from the design model Model View Controller (MVC), which consists three components; model, view and controller. The design model has been modified for this unique project since the Activities in Android both consists the .xml files (views) and Java files (controllers, views, utils and models). The project therefore consists of the following three packages.

- ViewControllers
- Models
- Util

The decision of this design model has mainly been based on the advantage of separating the view from the model and the utils, although in the android projject the view cannot be separated from the controller as explained below. Another aspect while deciding design pattern has been the groups' earlier experiences of MVC which has enabled the group to have a larger focus on the implementation of the functionality and user experience. The definition of the three packages and their relations are explained below.

The ViewController package consists of all the direct interaction with the user in the .java files. In an Android project the view this is limited to all of the Activities. In this project all of the activitites are also controllers, which makes them ViewControllers. The ViewController therefore both represents the View and the Controller part in the application. The ViewController calls on methods in the component Model. The view also consists of the .xml files that are used by the ViewControllers to specify the user interface.

The Model consists of the representation of all the information that is used in the application. It does not communicate, structurally, with any other package but is called by other packages, without knowing where the call is coming from.

The util package consist of utility classes. In this project the util package consists of a LocationHelper class that handle creation of the location manager and location listener and keeps track of geopoints. The project also consists of an imported util class from another project that is used to calculate the distance in meters between these geopoints. Methods in the package util are called by activities.

3. Storage

The storage requirements are 5,2 MB for the application on an android device. Additional storage from the application, for example the pet with its name and mood, is stored in the device's internal storage. The internal storage has been chosen to use because saved items should only be available for the actual application. By default files saved on internal storage are not available for other applications. It is only necessary to save the latest game, which therefore only requires little storage space and motivates the decision.

4. Test specification

The project was tested with two different types of tests; acceptance- and unit tests. The acceptance tests were tested with two different Android devices within the SDK target group.

Get access to acceptance tests:

git clone git//github.com/sofiase/ReadyForAPet/doc

The unit test was tested through Java with an Android device and a virtual device and simulated for different situations.

Get access to the automatic tests:

git clone git//github.com/sofiase/ReadyForAPet/ReadyForAPetTesterTest

5. Release procedure

A specific release procedure was set up for every application release of ReadyForAPet.

1. Test the project

- Acceptance tests
- Automatic tests

2. Clean the project

3. Build an apk-file

1. If it is the first release of the project, create a folder, in other case just continue with step 2.
2. Build an .apk-file of the specific application
    a. Right click on the project and select Export
    b. Select Android, Export Android Application
    c. Select the project
    d. Select new keystore and create a new location inside the folder for releases and pick a password. In our case this is a folder named after the release version.
    e. Note your alias name and password
    f. When completed all process with details, you will get one .apk.-file and a keystore file in the picked location.
    g. Right click on the project again on do steps b-c. Then select existing keystore and type in location and password.
    h. Your alias name will now appear in a dropdown, select it and enter your password.
    i. Select the same location as in step d to store your final .apk file

Release requirements

Every release should include an updated version of the following files.

- .apk-file
- Release notes
- Test report