Test Report Version 0.2.0

1 Introduction

1.1 Purpose of application

The purpose of the application is to generate a randomized route for a running session, and to track the runner during the run.

1.2 General characteristics of application

The mobile application generates a route by sennding coordinates to Google Maps API and then tracks the runner using GPS.

2 Test enviroment

The code is available at https://github.com/Salking/MakeMyRun. The test project (MakeMyRunTest) is located within the main project. Import both into Eclipse (as separate projects) and run the test project as Android JUnit Test.

2.1 Hardware enviroment

Unit tests will be run in the emulator on your machine. Manual tests are performed with an Android smart phone. Some kind of connection is required so that GPS positioning will be available.

2.2 Software environment

JUnit 3 using Eclipse. The Android emulator is set to Google API Level 15.

2.2.3 Softwares

See 2.2.

2.2.3 Software settings

---

3 System information

3.1 System version

Version 0.2.0

4 Known bugs and limitations

No known bugs. Although the application is limited to generating a new route, starting it and stopping it. During the run we display distance and time elapsed.
That is all we do, nothing else.

5 Test specification

See Test Cases version 0.2.0

6 Automatic test

6.1 Code coverage

Emma through ant is used. We have an 80% requirement for unit testable classes, and a 90% goal.

6.2 Nightly builds
Nightly builds are not yet being made.

6.3 Unit test
JUnit
Unit testing actual position of the user is impossible. Mocking it can only assert so much for us. Here we will have to rely on manual tests.

7 Test report
Manual acceptance tests. See (Acceptance Test Report 0.2.0.pdf)
Automatic Android Unit Test Report. See (Unit Test Report 0.2.0.html)

RouteGeneratorTest.testGetCurrentRoute() fails due to difficulties with mocking GPS Locations.