

Test Report Version 0.3.1

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 sending coordinates to Google Maps API and then tracks the runner using GPS.

2 Test environment

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 environment

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 0.3.1

4 Known bugs and limitations

When generating routes we have not yet implemented a fix to avoid tolls in order to avoid generating route over certain paths (i.e. a ferry crossing). Progress bar not showing correctly, only flashes at the end of the Async task. Current location dot sometimes disappears randomly.

5 Test specification

See *Test Cases version 0.3.1*

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

Acceptance test results

Test id	Test Result	Comment
1	Not yet implemented	
2	Not yet implemented	
3	Pass	The progress bar is shown very briefly
4	Not yet implemented	
5	Pass	
6	Pass	
7	Pass	
8	Pass	
9	Pass	

Automatic unit tests

Picture below is a screenshot from a document generated by Emma, providing information about code coverage from unit tests. We have 50 JUnit test cases.

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %	line, %
all classes	84% (27/32)	65% (115/176)	72% (2685/3729)	69% (522.4/755)

OVERALL STATS SUMMARY

total packages: 6
 total executable files: 25
 total classes: 32
 total methods: 176
 total executable lines: 755

COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %	line, %
com.pifive.makemyrun	64% (9/14)	44% (36/82)	41% (591/1427)	42% (143.6/338)
com.pifive.makemyrun.model	100% (5/5)	79% (19/24)	89% (865/976)	90% (145.6/161)
com.pifive.makemyrun.database	100% (2/2)	84% (16/19)	92% (469/508)	89% (75.6/85)
com.pifive.makemyrun.external	100% (1/1)	50% (1/2)	92% (111/120)	95% (21.8/23)
com.pifive.makemyrun.drawing	100% (9/9)	86% (37/43)	93% (606/655)	91% (126.8/139)
com.pifive.makemyrun.geo	100% (1/1)	100% (6/6)	100% (43/43)	100% (9/9)

[all classes]

EMMA 2.0.5312 (C) Vladimir Roubtsov