rAppla
Master Test Plan

Version 1.0

Revision History

Date	Version	Description	Author
24.04.2014	1.0	Android App für rapla Stundeplan	rAppla Team

Table of Contents

Master Test Plan

Introduction

Purpose

The purpose of the Master Test Plan is to gather all of the information necessary to plan and control the test effort for a given iteration. It describes the approach to testing the software, and is the top-level plan generated and used by managers to direct the test effort.

This *Test Plan* for the rAppla App supports the following objectives:

- Identifies the required resources
- Outlines the testing approach that will be used
- Identifies the items that should be targeted by the tests

Scope

- User-Interface Test
- State-Based Test (synchronisation)

Intended Audience

- Project Members
- People interested in Android-Testing

Evaluation Mission and Test Motivation

Background

- Ensure a flawlessly working User-Interface
- Ensure a flawlessly working Update Process

Cucumber

Alle erfolgreichen Cucumber Tests sind in den Use-Case Spezifikationen aufgelistet

Evaluation Mission

- Verify specifications
- Finding as many bugs as possible
- Advise about testing

Test Motivators

- Existing Use Cases
- Performance
- Workflow

Target Test Items

The listing below identifies those test items□software, hardware, and supporting product elements that have been identified as targets for testing. This list represents what items will be tested.

- Client operations
- Rappla synchronization

Test Approach

Testing Techniques and Types

Function Testing

Technique Objective:	• Ensure successful Rapla update • Ensure successful Parsing • Ensure correct Initialisation
Technique:	Based on Android-Unit-Tests
Oracles:	Result of the Android-Unit-Test and the corresponding test log.
Required Tools:	Android-Unit-Tetsing integrated in Eclipse IDE
Success Criteria:	All test return the correct and expected result
Special Considerations:	The Android-Unit-Test does not create a visible version of the graphical user interface

User Interface Testing

Technique Objective:	• Ensure correct displaying of events and graphical objects
Technique:	Based on Android-Unit-Testing
Oracles:	Result of the Android-Unit-Test and the corresponding test log.
Required Tools:	Android-Unit-Tetsing integrated in Eclipse IDE
Success Criteria:	All test return the correct and expected result
Special Considerations:	The Android-Unit-Test does not create a visible version of the graphical user interface

Entry and Exit Criteria

Test Plan

Test Plan Entry Criteria

An android emulator or device is connected to the testing computer

Test Plan Exit Criteria

The test is terminated, when the tests are finished or the device is disconnected

Deliverables

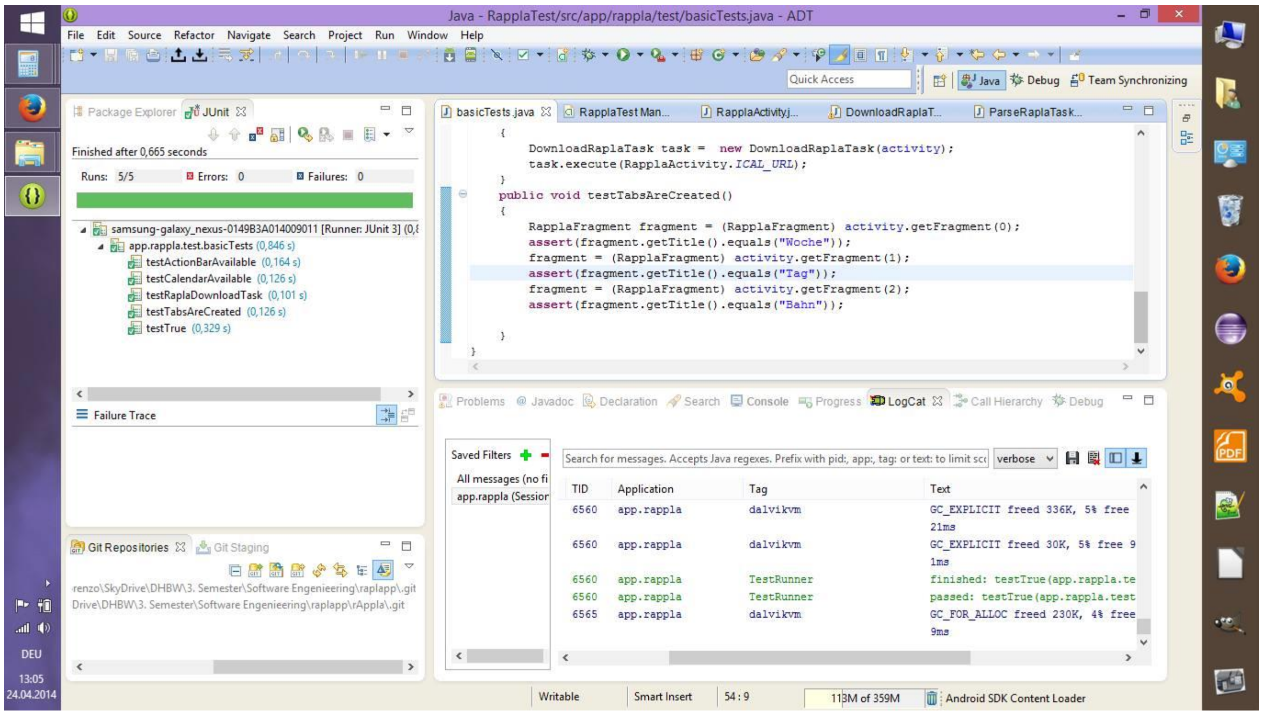
Test Evaluation Summaries

Results are output in testlogs

Reporting on Test Coverage

Results are output in testlogs

Proof of successful test and integration of Unit-Testing in Eclipse



```
Finished after 0,665 seconds
Runs: 5/5    Errors: 0    Failures: 0

samsung-galaxy_nexus-0149B3A014009011 [Runner: JUnit 3] (0,8
  app.rappla.test.basicTests (0,846 s)
    testActionBarAvailable (0,164 s)
    testCalendarAvailable (0,126 s)
    testRaplaDownloadTask (0,101 s)
    testTabsAreCreated (0,126 s)
    testTrue (0,329 s)
```

```
renzo\SkyDrive\DHBW\3. Semester\Software Engenieering\raplapp\.git
Drive\DHBW\3. Semester\Software Engenieering\raplapp\rAppla\.git
```

```
    {
        DownloadRaplaTask task =  new DownloadRaplaTask(activity);
        task.execute(RapplaActivity.ICAL_URL);
    }
    public void testTabsAreCreated()
    {
        RapplaFragment fragment = (RapplaFragment) activity.getFragment(0);
        assert(fragment.getTitle().equals("Woche"));
        fragment = (RapplaFragment) activity.getFragment(1);
        assert(fragment.getTitle().equals("Tag"));
        fragment = (RapplaFragment) activity.getFragment(2);
        assert(fragment.getTitle().equals("Bahn"));

    }
}
```

```
TID   Application   Tag          Text
6560  app.rappla    dalvikvm     GC_EXPLICIT freed 336K, 5% free
                                 21ms
6560  app.rappla    dalvikvm     GC_EXPLICIT freed 30K, 5% free 9
                                 1ms
6560  app.rappla    TestRunner   finished: testTrue(app.rappla.te
6560  app.rappla    TestRunner   passed: testTrue(app.rappla.test
6565  app.rappla    dalvikvm     GC_FOR_ALLOC freed 230K, 4% free
                                 9ms
```