Test report for Ripped

# 1 Introduction

This application is for persons that exercises and want to have an overview what they have done recently. The application can register:

* Cardio exercises (in example sports)
* Static exercises(in example sitting in 90 degress,
* Dynamic exercise(in example push-ups, bench-press etc)

# 1.1 Purpose of the application

This application is important since the most of the people nowdays exercise somehow and this application makes it easy to remember for the user on what they did and what they should do.

# 1.2 General characteristics of application

The application contains six different sections, they are:

1. The Workout – Here a user can have a routine that they should do, maybe a workout at the gym or a training pass. It is able to create routines in this section and edit the existing ones.
2. Exercises – List all the available exercises that the user can add into the workout. If something doesn’t exist, the user is able to add it! An exercise can also be edited with a new name or adding a note to it.
3. History – In this section the user can see the past workouts that have been made. See what you did and which date it was made.
4. Statistics – Not implemented yet
5. Profile – Enter name, age, weight and height. Not implemented yet.
6. Settings – There is an about button where it is explained why the application was built and when, also in this section there are three options of settings. They are:  
   * Language - Able to change language to Swedish or English
   * Distance - The user is also able to change distance unit to either kilometers or miles.
   * Weight – Is able to change unit in weight to either pounds or kilograms.

# 2 Test environment

The test code exists in the Android-project but has to be created as a new project since the current one is not a JUnit-testing project. When the GUI was tested a test-program called “Robotium” was used. How to install it is easy! Follow those instructions below to start the the test-project:

1. Start with File->New -> Other and thereafter take Android Test Project
2. Enter a name of the project
3. Select the application as the “Test target”.
4. Then insert the robotium-solo.jar and the javadoc.jar into the new project.
5. Then right-click into the project and select properties. Thereafter press Java Build Path and choose libraries and press the button “Add JARs..” which is placed on the left side. Then take the robotium-solo.jar in the selection and now the file will be shown in the library.
6. Click on the robotium-solo file in the library, press Javadoc location and click in the radio-button called “Javadoc in archive” and choose browse and thereafter pick the javadoc.jar.
7. Click the tab “Order and Export” and click in the robotium checkbox.
8. Now pick the “test code”:s source files that are located in the project and copy them into the JUnit test project.
9. Now the test can be started!

Before testing the application it is highly recommended that the database is cleaned since some of the test cases won’t go through.

# 2.1 Hardware environment

Since the GUI is based on Android 2.2 any device can be used since “timeouts” are implemented so the phone or the emulator can’t fail because it is to slow.

# 2.2 Software environment

In the testing Robotium(version 3.4.1) and JUnit testing was used. For the GUI Robotium was used and for the database JUnit was used.

# 2.2.3 Softwares

Since we want to test the GUI robotium is a perfect alternative

# 2.2.4 Software settings

No extra settings are needed since the configuration is done when Robotium is “installed”.

# System information

# 3.1 System version

The current version is 1.0.

# 4 Known bugs and limitations

Before testing a few bugs where found, some of them weren’t bugs, just misunderstandings. The issues were:

1. Distorded menu on ldpi devicess
2. Name for “add exercise” is not cleared
3. Not uniform design
4. ListWorkoutActivity.java does not update after creating a workout
5. Wrong string on cancel button
6. “Add new workout” does not work correctly
7. “Add exercise” in Exercise activity crashed
8. When submitting a exercise
9. Main menu is distorted on ldpi devices
10. All Swedish strings aren’t implemented
11. Time in database is set to UTC, not the local timezone
12. Problems aborting add exercise
13. Not uniform design for profile

# 5 Test specification

## Test cases for version 0.7

* 1. Launching the app

Description: When you launch the app it will starts and Main Activity will appear

Precondition: An installed app

Test steps:

1. Launch the app
2. Verify that the menu is visible

Related user stories: All

* 1. List Workouts in Workout-Activity

Description: List your Workouts in Workout-Activity

Precondition: The app is started(test case 1.1)

Test steps:

1. Select the Workout button
2. Verify that the Workout-activity viewing the content in WorkoutActivity

Related user stories: 1.1, 1.2

* 1. List all Exercises in Exercise-activity

Description: List all the Exercises in the Exercise-activity,

Precondition: The app is started(test case 1.1)

Test steps:

1. Select the Exercise-Button.
2. Verify that the Exercise-activity viewing the content in Exercise

Related user stories: 2

* 1. List the History in History-activity

Description: List your history in the History-activity

Precondition: The app is started(test case 1.1)

Test steps:

1. Select the History-Button.
2. Verify that the History-activity viewing the content in history

Related user stories:

* 1. See your Statistic in Statistic-Activity

Description: See your Statistic in Statistic-Activity

Precondition: The app is started(test case 1.1)

Test steps:

1. Select the Statistics-Button.
2. Verify that the Statistics-activity viewing the content in statistics

Related user stories:

* 1. Show your Profile in Profile-Activity

Description: Show your Profile

Precondition: The app is started(test case 1.1)

Test steps:

1. Select the Profile-Button.
2. Verify that the Profile-activity shows the profile

Related user stories:

* 1. List the Settings-menu

Description: List the Settings-menu

Precondition: The app is started (test case 1.1)

Test steps:

1. Select the Settings-Button.
2. Verify that the Settings-activity starts

Related user stories:

1. Add a workout  
    Description: Add a workout  
    Precondition: Accessibility to the Workout-Activity(test case 1.2)  
    Test steps:
2. Select the Add Workout button
3. Verify that a dialog pops up
4. Try to enter text into the text field
5. Click the Add workout button
6. Verify that the Edit Workout-Activity starts with the text from the text field as the title
7. Verify that the workout exist in the database

Related user stories: 1.1.1

1. Cancel add-workout dialog  
    Description: Cancel add-workout dialog  
    Precondition: Accesibility to add-workout dialog(test case 2.1)  
    Test steps:
2. Select the cancel-button
3. Verify that dialog is cancelled

Related user stories: 1.1.1

1. Search for Exercises by muscle group in Edit Workout  
    Description: Search for Exercises by muscle group in Edit Workout  
    Precondition: Accessibility to Edit Workout works(test case 2.4)  
    Test steps:
2. Select the spinner that contains the muscle groups
3. Verify that the spinner opens
4. Verify that the spinner is scroll-able
5. Select an item in the spinner
6. Verify that the spinner is closed after an item is selected
7. Verify that the searched exercises appear

Related user stories: 1.1.2

1. Open edit a workout  
    Description: Open edit a workout  
    Precondition: Accessibility to Workout-Activity(Test case 2.1)  
    Test steps:
2. Select a workout in Workout-activity
3. Verify that the Edit Workout-Activity is listing Exercises
4. Verify that the list is scroll-able

OR

1. Longclick a workout in Workout-activity
2. Press “Edit” on the pop-up screen
3. Verify that the Edit Workout-Activity is Listing Exercises

Related user stories: 1.1.4

1. Add exercise to Workout

Description: Add Exercise to Workout  
 Precondition: Accessibility to Edit Workout-Activity(Test case 2.4)  
 Test steps:

1. Select an unchecked Exercise
2. Verify that the Exercise got a checkbox
3. Save the workout
4. Open the workout and see if the exercise is checked

Related user stories: 1.1.4

1. Remove exercise from Workout  
    Description: Remove exercise from Workout  
    Precondition:Accessibility to Edit Workout(test case 2.4)  
    Test steps:
2. Select an already checked Exercise
3. Verify that the Exercise don’t have a checkbox
4. Save the workout
5. Open the workout and see if the exercise is not checked.

Related user stories: 1.1.4

1. Clone the workout  
    Description: Make a duplicate of the workout   
    Precondition: The user has selected Workout from the main menu.  
    Test steps:
2. Long click a workout in the list.
3. Select “Clone” from the pop up menu.
4. Verify that there is two workouts with same name and exercises.

Related user stories: 1.1.5

1. Delete a workout  
    Description: Deletes a workout   
    Precondition: The user has selected Workout from the main menu.  
    Test steps:
2. Long click a workout in the list.
3. Select “Delete” from the pop up menu.
4. Press ok in the dialog
5. Verify that the workout doesn’t exists.

Related user stories: 1.1.3

1. Register cardio exercises:  
    Description: Register result for performed cardio exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Click on a cardio exercise
3. Verify that the activity for register cardio is active

Related user stories: 1.2.2

1. Register cardio exercises: Not valid time  
    Description: Register result for performed cardio exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Set both minutes and seconds to zero (or blank), distance whatever you want (can only be decimal numbers and blank).
3. Press “Add set”.
4. Note that the set is not added to “Current sets” label and an error will appear.

Related user stories: 1.2.2

1. Register cardio exercises: Valid time  
    Description: Register result for performed cardio exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Set either minutes or seconds to zero and the other one to a number greater than 0, distance whatever you want (can only be decimal numbers).
3. Press “Add set”.
4. Note that the set is added to “Current sets”.
5. Verify that it is written to database when clicking “Done”.

Related user stories: 1.2.2

1. Register static exercises:  
    Description: Register result for performed static exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Click on a cardio exercise
3. Verify that the activity for register static is active

Related user stories: 1.2.3

1. Register static exercises: Not valid time  
    Description: Register result for performed static exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Set both minutes and seconds to zero and the other one to a number greater than 0, weight whatever you want (can only be decimal numbers).
3. Press “Add set”.
4. Verify that the set is not added to “Current sets” and an error is displayed.

Related user stories: 1.2.3

1. Register static exercises: Valid time  
    Description: Register result for performed cardio exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Set either minutes or seconds to zero and the other one to a number greater than 0, weight whatever you want (can only be decimal numbers).
3. Press “Add set”.
4. Note that the set is added to “Current sets”.
5. Verify that it is written to database when clicking “Done”.

Related user stories: 1.2.3

1. Register dynamic exercises:  
    Description: Register result for performed dynamic exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Click on a cardio exercise
3. Verify that the activity for register dynamic is active

Related user stories: 1.2.6

1. Register dynamic exercises: Not valid reps  
    Description: Register result for performed dynamic exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Set number of reps to zero (or blank), weight either blank, zero or decimal number)
3. Press “Add set”.
4. Verify that the set is not added to “Current sets” and an error is displayed.

Related user stories: 1.2.6

1. Register dynamic exercises: Valid reps  
    Description: Register result for performed dynamic exercise  
    Precondition: The user has selected to start a workout  
    Test steps:
2. Enter a number greater than zero in Reps and whatever you want in weight (blank, zero or decimal number).
3. Press “Add set”.
4. Note that the set is added to “Current sets”.
5. Verify that it is written to database when clicking “Done”.

Related user stories: 1.2.6

1. Sets from last time  
    Description: See sets from last time  
    Precondition: The user has performed and saved the sets of an exercise  
    Test steps:
2. Choose to register new results for an exercise you have performed before.
3. Verify that the last time sets is listed in the view.

Related user stories: 1.2.9

1. Add a workout with no name(SHOULD NOT BE POSSIBLE)  
    Description: Add a workout with no name  
    Precondition: Accessibility to Workout-Activity(Test case 2.1)  
    Test steps:
2. Choose to register new results for an exercise you have performed before.
3. Verify that the last time sets is listed in the view.

Related user stories: 1.2.9

1. View all exercises  
    Description: View all exercises  
    Precondition: the start menu is shown test case 1.1  
    Test steps:
   * + 1. Push the exercise button
       2. Verify that list exercise view is shown

Related user stories: 2.1

1. Test Exercise home button  
    Description: Test Exercise home button  
    Precondition: Testfallet som testar Exerciseknappen  
    Test steps:
   * + 1. Push Exercise Button
       2. In Exercise Activity push the home button
       3. Verify that the home screen is shown

Related user stories: 2.1

1. Enter name on new exercise  
    Description: Enter name on new exercise  
    Precondition: The exercise window is active and the test case 1.1.1 successful  
    Test steps:
   * + 1. Push ADD EXERCISE! Button
       2. Write Exercise name
       3. Push Add
       4. The activity edit Exercise should show up with the exercise name as title

Related user stories: 2.9, 2.2

1. Enter no name on a new exercise  
    Description: Enter no name on a new exercise  
    Precondition: The exercise window is active and the test case 1.1.1 successful  
    Test steps:
   * + 1. Push “ADD EXERCISE!” - Button
       2. Push Add without adding a name
       3. A text will appear in the textbox, “Please enter a name”

Related user stories: 2.8, 2.2

1. Add data for a new exercise and press cancel(With no data)

Description: Add data for a new exercise and also adds the exercise

Precondition: That the Edit Exercise activity is active and also test case 1.2.1  
 Test steps:

1. Push “ADD EXERCISE!” - Button
2. Write a exercise name
3. Edit Exercise activity will pop-up
4. Choose musclegroups/Cardio and click Save.
5. Verify that List Exercise activity starts.

Related user stories: 2.9

1. Add a new exercise with all the data(Maximum settings)  
    Description: Add a new exercise with all the data  
    Precondition: The activity Edit exercise is working and test case 3.3 is working.  
   Test steps:
   * + 1. Push “ADD EXERCISE!” – Button
       2. Write a exercise name
       3. Edit Exercise activity will pop-up
       4. Choose musclegroups/Cardio, fill in notes and data and click Save.
       5. Verify that List Exercise activity starts.

Related user stories: 2.2, 2.10

1. Edit an existing exercise  
    Description: Edit an existing exercise  
    Precondition: The list exercise view is active testcase 1.2.2, 1.2.3  
    Test steps:
   * + 1. Click on an existing exercise
       2. Edit something in the exercise
       3. Click Save button.
       4. Verify that List Exercise activity starts.

Related user stories: 2.3

1. Delete an existing exercise  
    Description: Delete an existing exercise  
    Precondition: The list exercise view is active and test cases 1.2.2, 1.2.3  
    Test steps:
   * + 1. Longclick an exercise
       2. On the pop-up menu, click “Delete”
       3. Verify that the current activity is List Exercise Activity

Related user stories: 2.3

1. Abort adding exercise  
    Description: Abort adding exercise  
    Precondition: The list exercise view is active test case 1.2.2, 1.2.3  
    Test steps:
   * + 1. Push ADD EXERCISE! Button
       2. Fill in Exercise name
       3. Push Add
       4. Go to the bottom and push cancel
       5. Verify that the current activity is List Exercise

Related user stories: 2.6

1. Cancel when editing an exercise  
    Description: Cancel when editing an exercise  
    Precondition: The list exercise view is active test case 1.2.2, 1.2.3  
    Test steps:
   * + 1. Push an exercise
       2. Edit some info
       3. Go to the bottom and push cancel
       4. Verify that the exercise not have been changed

Related user stories: 2.7

1. Open profile  
    Description: Open profile  
    Precondition: The app is started  
    Test steps:
   * + 1. Select the Profile-button
       2. Verify that the right activity starts

Related user stories: 3.1

1. Add data to profile  
    Description: Add data to profile  
    Precondition: Able to go into the profile menu(4.1)  
    Test steps:
   * + 1. Add data to all fields
       2. Click the button “update profile”
       3. Verify that the main activity starts

Related user stories: 3.1

1. Add data to profile without filling in all fields  
    Description: Add data to profile without filling in all fields  
    Precondition: Able to go into the profile menu(4.1)  
    Test steps:
   * + 1. Add data but not to all of the fields
       2. Click the button “update profile”
       3. Verify that a message pop-up that says “Fill in correct information”.

Related user stories: 3.1

# JUnit test cases.

1. Test add 100 exercise  
    Description: Test add 100 exercises  
    Test steps:
2. Check how many exercises that is in the database
3. Adds 100 exercises
4. Verify that 100 exercises is added
5. Test add one exercise  
    Description: Test add one exercise   
    Test steps:
6. Check how many exercises that is in the database
7. Adds one exercise
8. Verify that one exercise is added
9. Test delete all exercises  
    Description: Test delete all exercises  
    Test steps:
10. Check how many exercises that is in the database
11. Delete all exercises
12. Verify that there are no exercises in the database
13. Test delete one exercise  
     Description: Test delete one exercise  
     Test steps:
14. Check how many exercises that is in the database
15. Delete one exercise
16. Verify that one exercise is deleted
17. Test receive exercise with specific id  
     Description: Test receive exercise with specific id   
     Test steps:
18. Receive an exercise with specific ID
19. Check that ID is right
20. Verify that the name isn’t null
21. Verify that the TypeID isn’t null
22. Test edit one exercise  
     Description: Test edit one exercise   
     Test steps:
23. Receive an exercise with specific ID
24. Change the description of the exercise
25. Edit the exercise
26. Verify that the edits that recently were done works.
27. Test add profile  
     Description: Test add profile   
     Test steps:
28. Checks number of profiles in the database
29. Adds a profile
30. Verify that the profile is added
31. Test to get performed workouts   
     Description: Test get performed workouts   
     Test steps:
    * + 1. Checks how many workouts it is in the database
32. Test add cardio set  
     Description: Test add cardio set   
     Test steps:
33. Checks how many sets that are currently in the database
34. Adds one cardio set
35. Verify that the set is added
36. Test add dynamic set   
     Description: Test add dynamic set  
     Test steps:
37. Checks how many sets that are currently in the database
38. Adds one dynamic set
39. Verify that the set is added
40. Test add static set   
     Description: Test add static set   
     Test steps:
41. Checks how many sets that are currently in the database
42. Adds one static set
43. Verify that the set is added
44. Test delete set   
     Description: Test delete set  
     Test steps:
45. Checks how many sets that are currently in the database
46. Delete one set
47. Verify that the set was deleted

# 6 Automatic test

# 6.1 Code coverage

Since Emma wasn’t implemented in this application we don’t know how much coverage we had. The goal is of course 100% but since we didn’t implemented Emma we don’t know how much we covered.

# 6.2 Nightly builds

None.

# 6.3 Unit test

Things that are difficult to test are example the scrolling since some of the activities doesn’t have enough workouts or exercises in it.

# Test report