

GTA-Group

WaterMe

**Use-Case Specification: Connect App to Raspberry Pi
via Bluetooth**

Version 1.3

Revision History

Date	Version	Description	Author
29/Nov/2016	1.0	First version	Chris Todt
01/Dez/2016	1.1	Added Basic Flow	Chris Todt
07/Dez/2016	1.2	Added Mockups and updated UML	Chris Todt
08/Dez/2016	1.3	Added Espresso test	Chris Todt

Table of Contents

Connect App to Raspberry Pi via Bluetooth	4
Brief Description	4
Mockup	4
Flow of Events	5
Basic Flow	5
Espresso test	6
Special Requirements	7
Bluetooth	7
Preconditions	7
Settings menu	7
Paired	7
Postconditions	7
Connected	7
Not connected	7
Extension Points	7

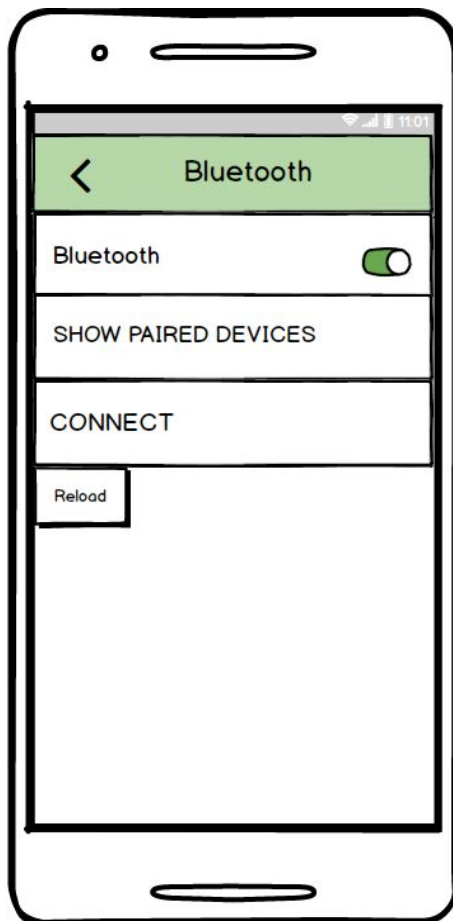
Use-Case Specification: Connect App to Raspberry Pi via Bluetooth

1. Connect App to Raspberry Pi via Bluetooth

1.1 Brief Description

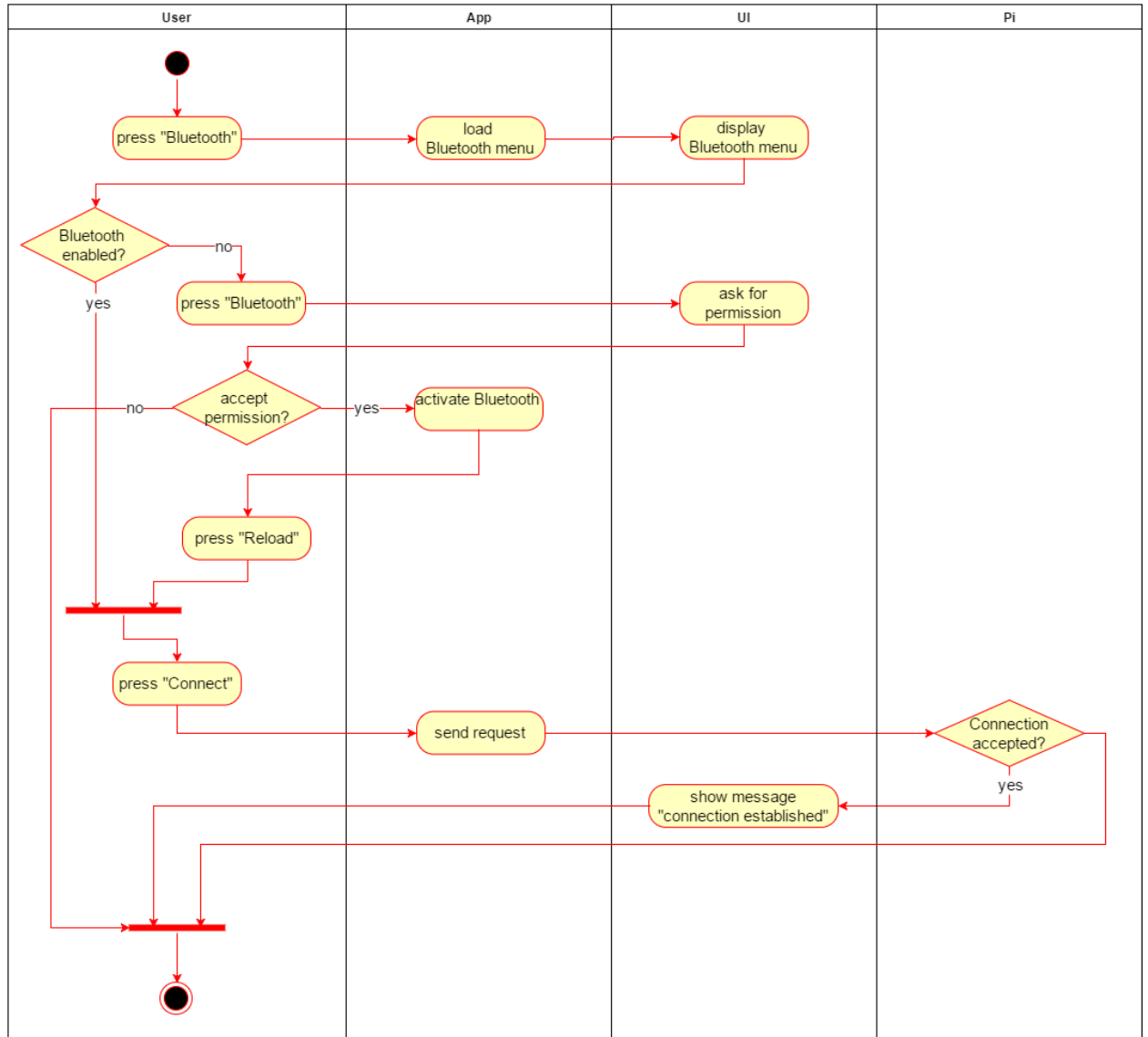
In “Bluetooth” the user can connect the application with the Raspberry Pi via Bluetooth.

1.2 Mockup



2. Flow of Events

2.1 Basic Flow



3. Espresso test

```
package com.project.gta.demo;

import android.os.SystemClock;
import android.support.test.espresso.ViewInteraction;
import android.support.test.rule.ActivityTestRule;
import android.support.test.runner.AndroidJUnit4;
import android.test.suitebuilder.annotation.LargeTest;

import org.junit.Rule;
import org.junit.Test;
import org.junit.runner.RunWith;

import static android.support.test.espresso.Espresso.onView;
import static android.support.test.espresso.action.ViewActions.click;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withId;
import static android.support.test.espresso.matcher.ViewMatchers.withParent;
import static android.support.test.espresso.matcher.ViewMatchers.withText;
import static org.hamcrest.Matchers.allOf;

@LargeTest
@RunWith(AndroidJUnit4.class)
public class UConnectApptoRaspberryPiviaBluetooth {

    @Rule
    public ActivityTestRule<MainMenu> mActivityTestRule = new
    ActivityTestRule<>(MainMenu.class);

    @Test
    public void uConnectApptoRaspberryPiviaBluetooth() {
        ViewInteraction appCompatButton = onView(
            allOf(withId(R.id.BTNsettings), withText("Settings"),
                withParent(allOf(withId(R.id.mainmenu),
                    withParent(withId(R.id.activity_main_menu)))),
                isDisplayed()));
        appCompatButton.perform(click());

        ViewInteraction appCompatButton2 = onView(
            allOf(withId(R.id.BTNbluetooth), withText("Bluetooth"), isDisplayed()));
        appCompatButton2.perform(click());

        ViewInteraction switch_ = onView(
            allOf(withId(R.id.SWbluetooth), isDisplayed()));
        switch_.perform(click());

        ViewInteraction appCompatButton3 = onView(
            allOf(withId(R.id.BTNreload), withText("Reload"),
                withParent(withId(R.id.LLbluetooth)),
                isDisplayed()));
```

```
appCompatButton3.perform(click());

ViewInteraction appCompatButton4 = onView(
    allOf(withId(R.id.BTNconnect_bt), withText("connect"), isDisplayed()));
appCompatButton4.perform(click());
SystemClock.sleep(9000);
}
}
```

4. Special Requirements

4.1 Bluetooth

The smartphone has to support Bluetooth.

5. Preconditions

5.1 Settings menu

The user has to be in the settings menu.

5.2 Paired

The smartphone and the Pi must be paired.

6. Postconditions

6.1 Connected

The smartphone is connected to the Pi and the user can see the message “connection established”.

6.2 Not connected

The smartphone could not connect to the Pi. No message is displayed.

7. Extension Points

(n/a)