GTA-Group

WaterMe
Use-Case Specification:
Change Settings

Version 1.3

Revision History

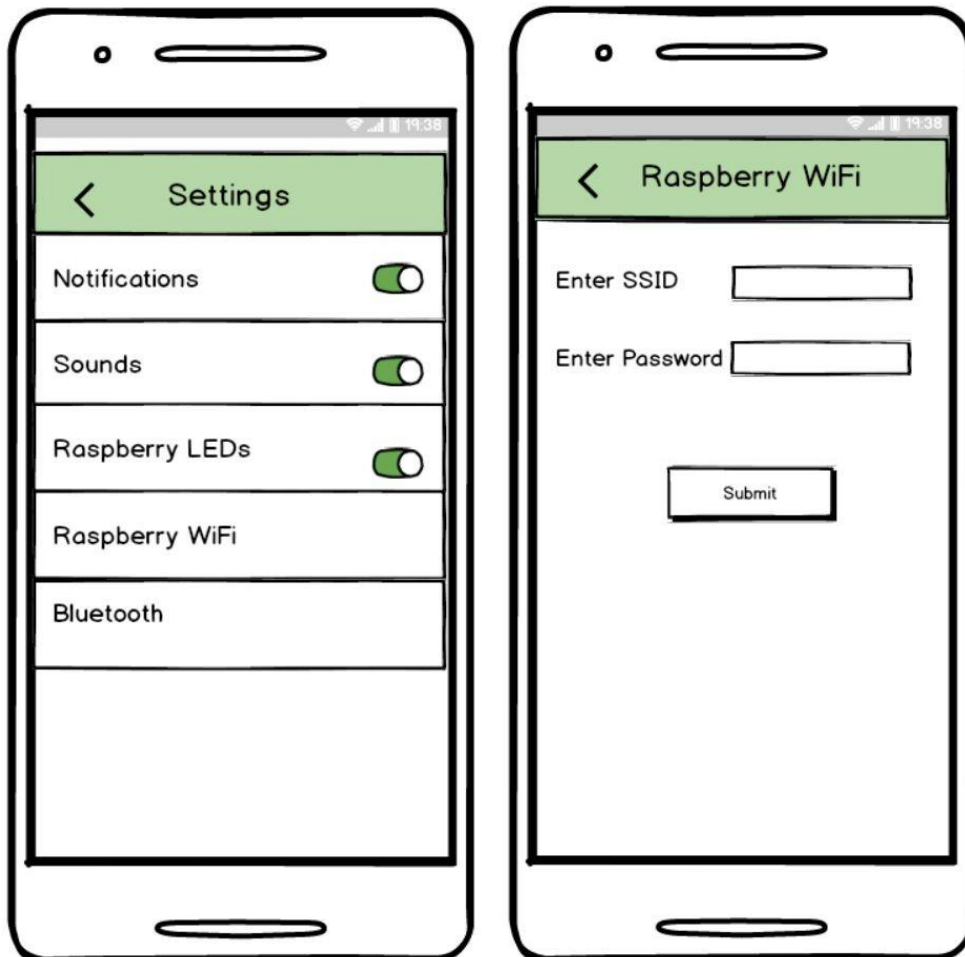| Date	Version	Description	Author
27/Okt/2016	1.0		Chris Todt Paul Giesa
01/Nov/2016	1.1		Olga Akymenko Chris Todt Paul Giesa
07/Dez/2016	1.1.1	Update UML + Mockups	Chris Todt
07/Dez/2016	1.2	Add Espresso test	Chris Todt
01/Mai/2017	1.3	Add FP	Olga Akymenko

Table of Contents

Use-Case Specification: Change Settings

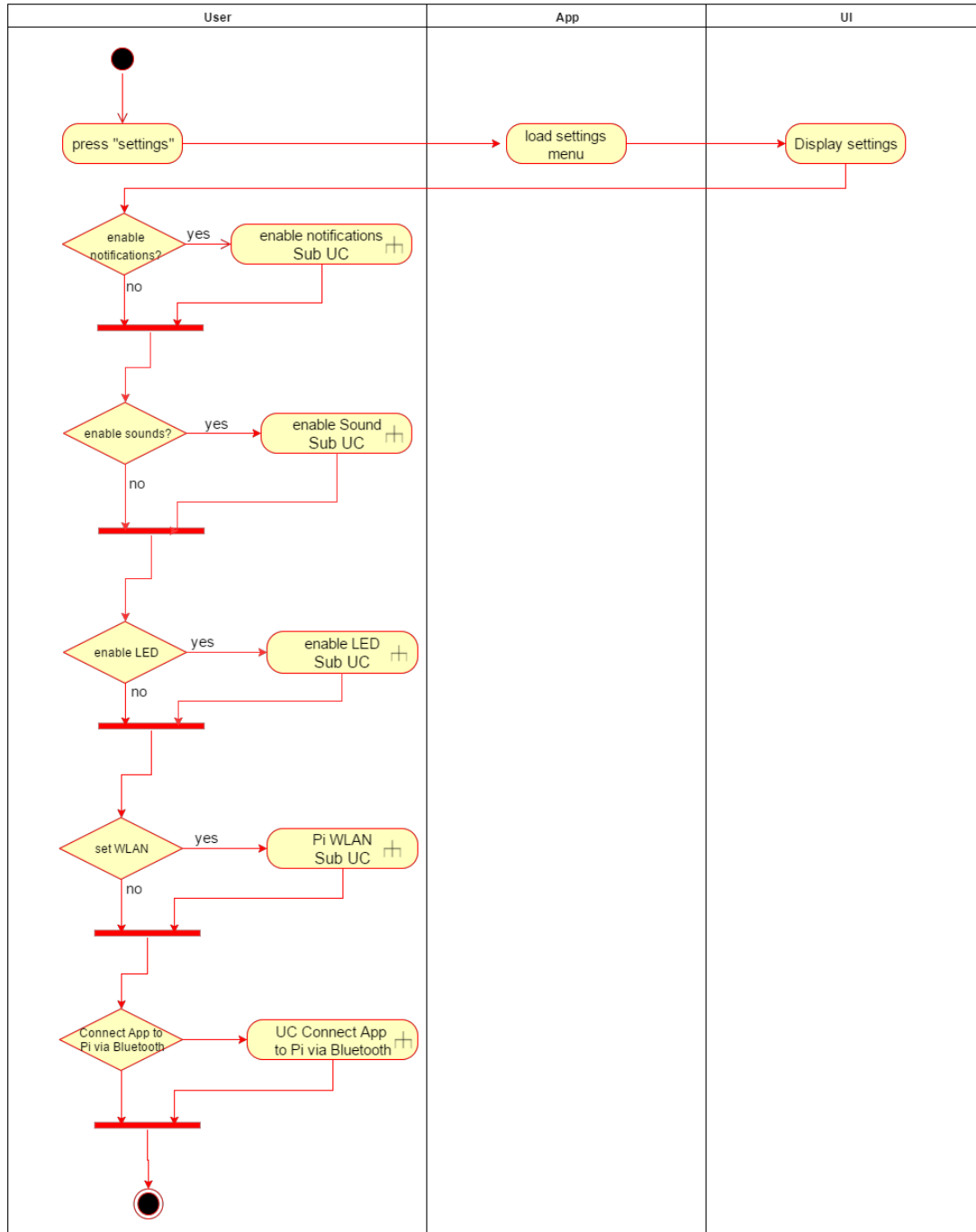## 1.   Change Settings

1.1   Brief Description

In "Settings" the user can activate/deactivate notifications, sounds, LEDs and change WLAN settings for the Pi.
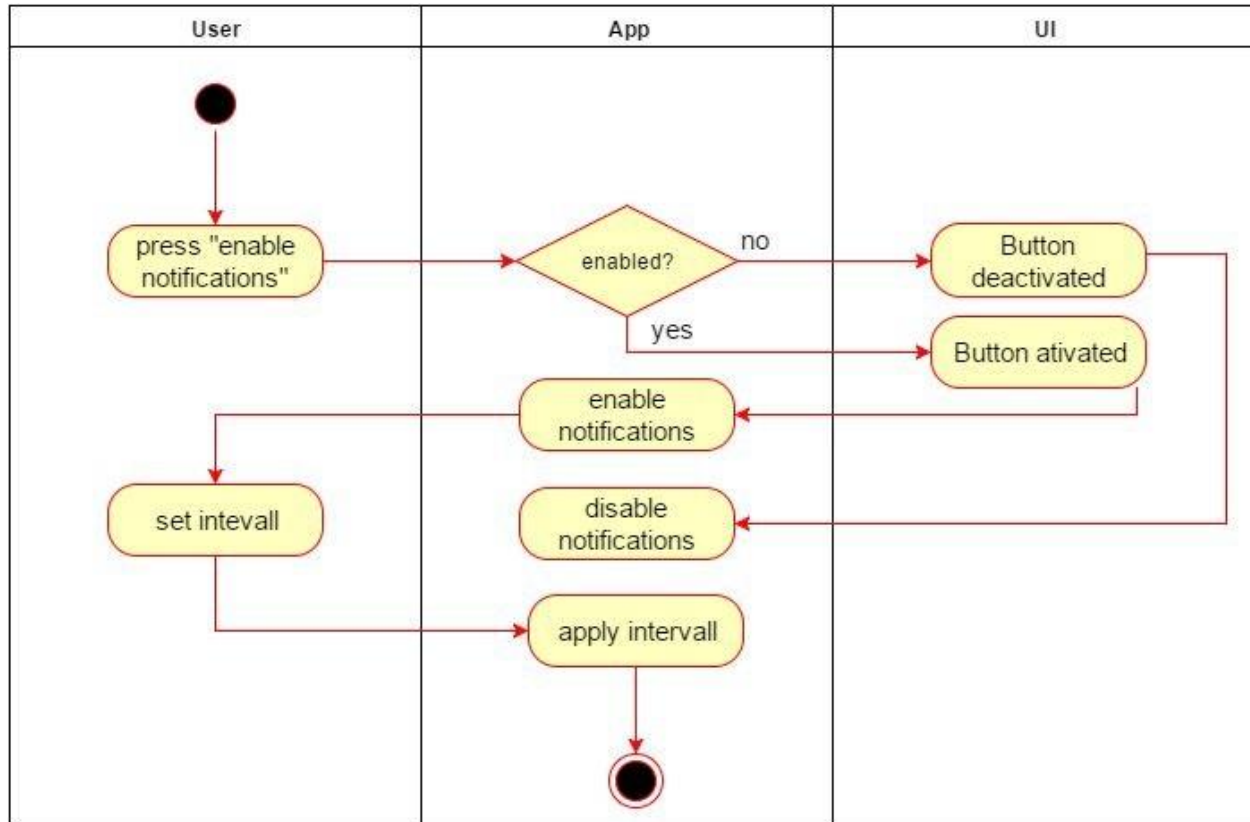
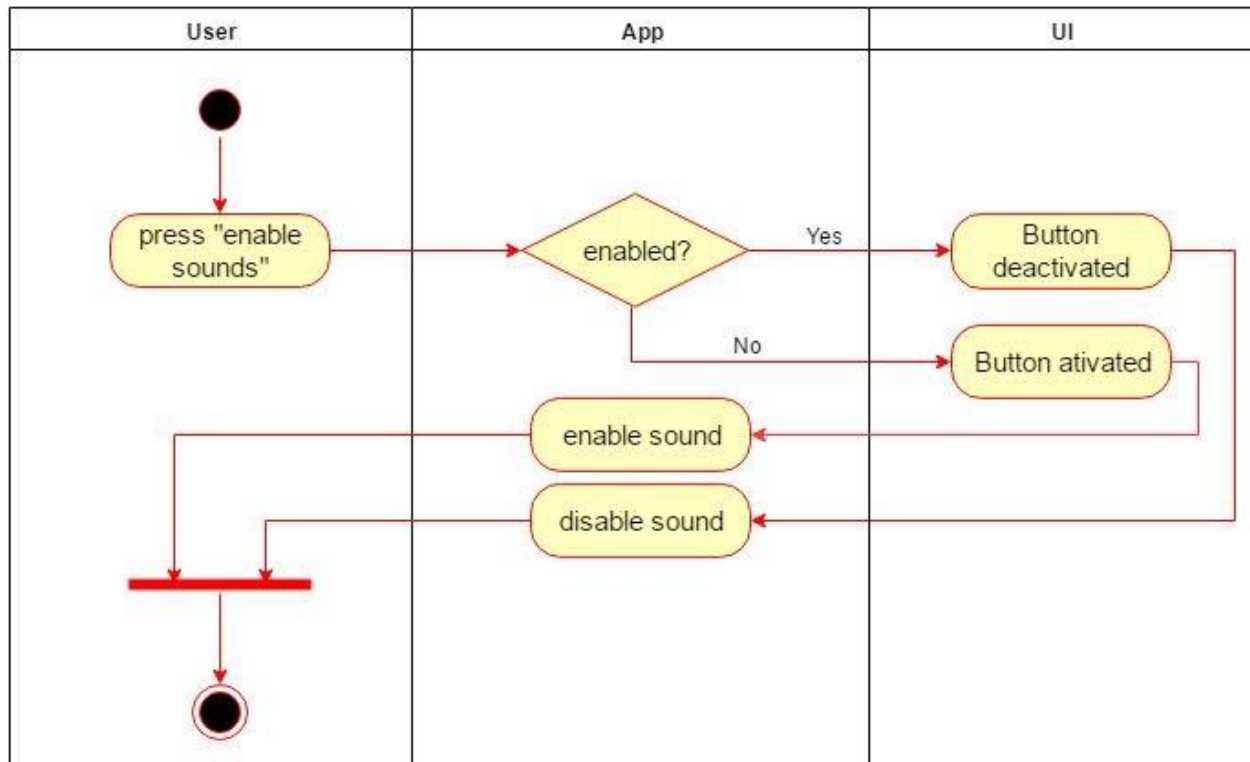### 1.2   Mockup

2. Flow of Events
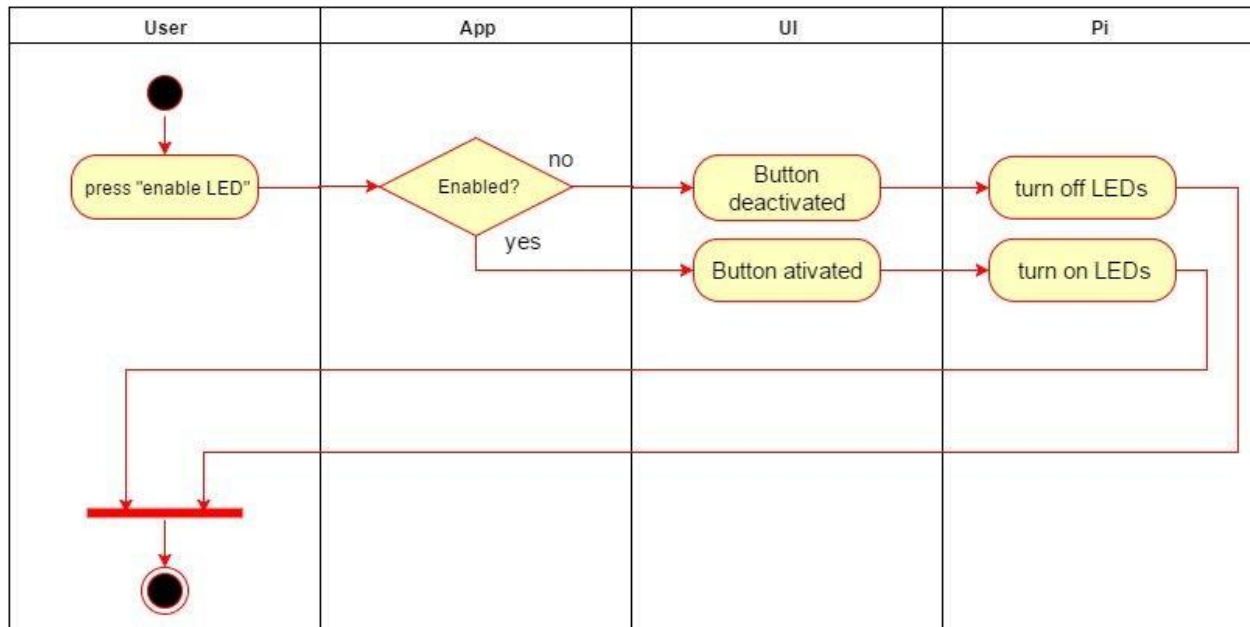
2.1 Basic Flow

2.2   Alternative Flows

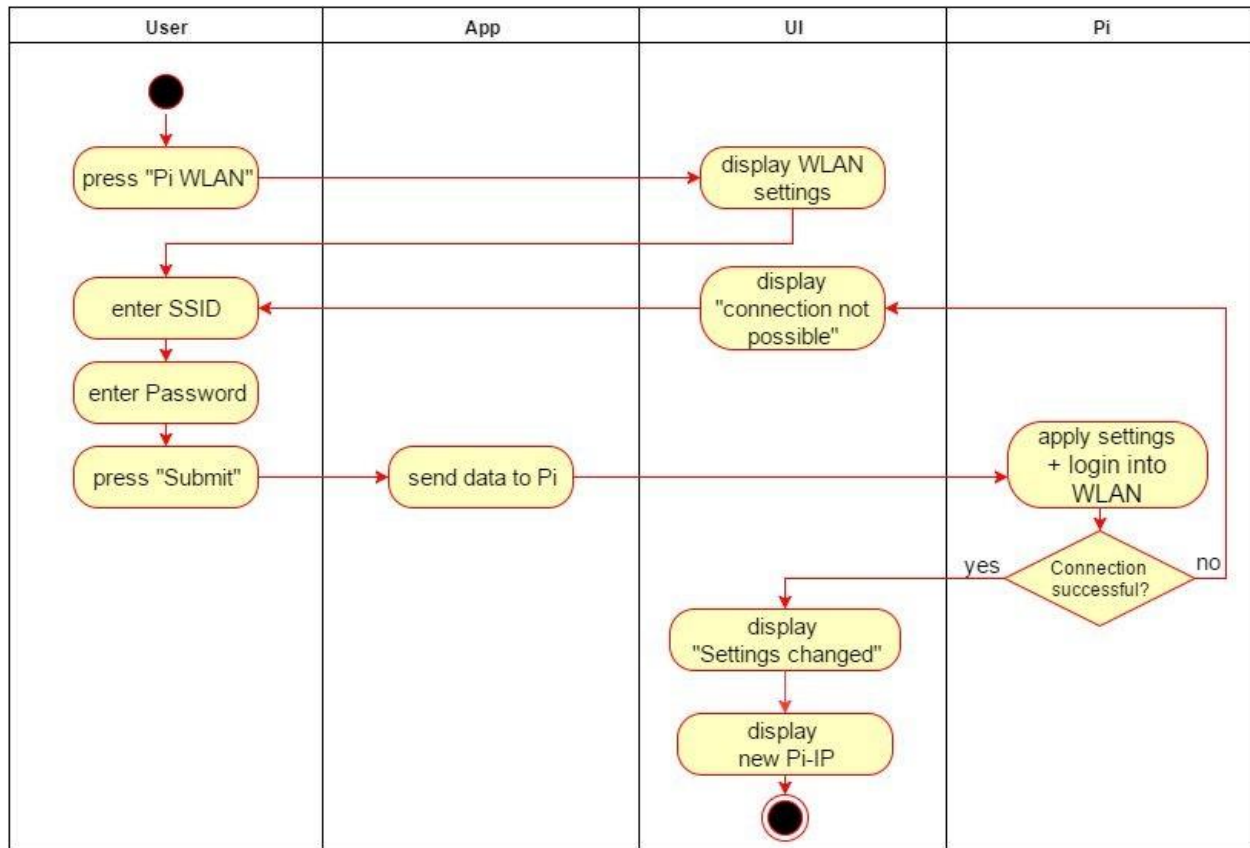2.2.1   *Enable notifications*

2.2.2   Enable sounds

2.2.3 *Enable LEDs*

2.2.4 Set WLAN

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.action.ViewActions.closeSoftKeyboard;
import static android.support.test.espresso.action.ViewActions.replaceText;
import static android.support.test.espresso.matcher.ViewMatchers.isDisplayed;
import static android.support.test.espresso.matcher.ViewMatchers.withContentDescription;
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;
// ------------------------Bluetooth muss aktiviert sein!------------------------
@LargeTest
@RunWith(AndroidJUnit4.class)
public class UCChangeSettings {

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

    @Test
    public void uCChangeSettings() {
        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 appCompatButton3 = onView(
                allOf(withId(R.id.BTNconnect_bt), withText("connect"), isDisplayed()));
        appCompatButton3.perform(click());
        SystemClock.sleep(9000);

        ViewInteraction appCompatImageButton = onView(
                allOf(withContentDescription("Nach oben"),
```

```java
                withParent(allOf(withId(R.id.action_bar),
                        withParent(withId(R.id.action_bar_container)))),
                isDisplayed()));
    appCompatImageButton.perform(click());

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

    ViewInteraction switch_2 = onView(
            allOf(withId(R.id.SWnotifications), isDisplayed()));
    switch_2.perform(click());

    ViewInteraction switch_3 = onView(
            allOf(withId(R.id.SWsounds), isDisplayed()));
    switch_3.perform(click());

    ViewInteraction switch_4 = onView(
            allOf(withId(R.id.SWsounds), isDisplayed()));
    switch_4.perform(click());

    ViewInteraction switch_5 = onView(
            allOf(withId(R.id.SWled), isDisplayed()));
    switch_5.perform(click());

    ViewInteraction switch_6 = onView(
            allOf(withId(R.id.SWled), isDisplayed()));
    switch_6.perform(click());

    ViewInteraction appCompatButton4 = onView(
            allOf(withId(R.id.BTNraspberrywifi), withText("Raspberry WiFi"),
isDisplayed()));
    appCompatButton4.perform(click());

    ViewInteraction appCompatEditText = onView(
            allOf(withId(R.id.TXTwifi_id), isDisplayed()));
    appCompatEditText.perform(click());

    ViewInteraction appCompatEditText2 = onView(
            allOf(withId(R.id.TXTwifi_id), isDisplayed()));
    appCompatEditText2.perform(replaceText("test"), closeSoftKeyboard());

    ViewInteraction appCompatEditText3 = onView(
            allOf(withId(R.id.TXTwifi_password), withContentDescription("Password"),
isDisplayed()));
    appCompatEditText3.perform(replaceText("passwort"), closeSoftKeyboard());

    ViewInteraction appCompatButton5 = onView(
            allOf(withId(R.id.BTNsubmit), withText("Submit"), isDisplayed()));
    appCompatButton5.perform(click());

    ViewInteraction appCompatImageButton2 = onView(
            allOf(withContentDescription("Nach oben"),
```

```
        appCompatImageButton2.perform(click());

        ViewInteraction appCompatImageButton3 = onView(
                allOf(withContentDescription("Nach oben"),
                        withParent(allOf(withId(R.id.action_bar),
                                withParent(withId(R.id.action_bar_container)))),
                        isDisplayed()));
        appCompatImageButton3.perform(click());

    }

}
```

4.   Special Requirements

(n/a)

5.   Preconditions

5.1   Main screen

The user has to be in the main screen to be able to access the settings.

6.   Postconditions

6.1   Changed buttons

The buttons will be transparent if the setting is disabled / colored if the setting is enabled.

7.   FP- Calculation

Calculation of DET/RET/FTR: DET/RET/FTR Table

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	Simple	Average	Complex
Number of User Input	0	●	○	○
Number of User Outputs	4	●	○	○
Number of User Inquiries	5	●	○	○
Number of Files	0	●	○	○
Number of External Interfaces	0	●	○	○

(Columns Simple/Average/Complex under WEIGHTING FACTOR)

8.   Extension Points

To be determined