GTA-Group

Water Me
Use-Case Specification: Manage plants

Version 1.2

Revision History

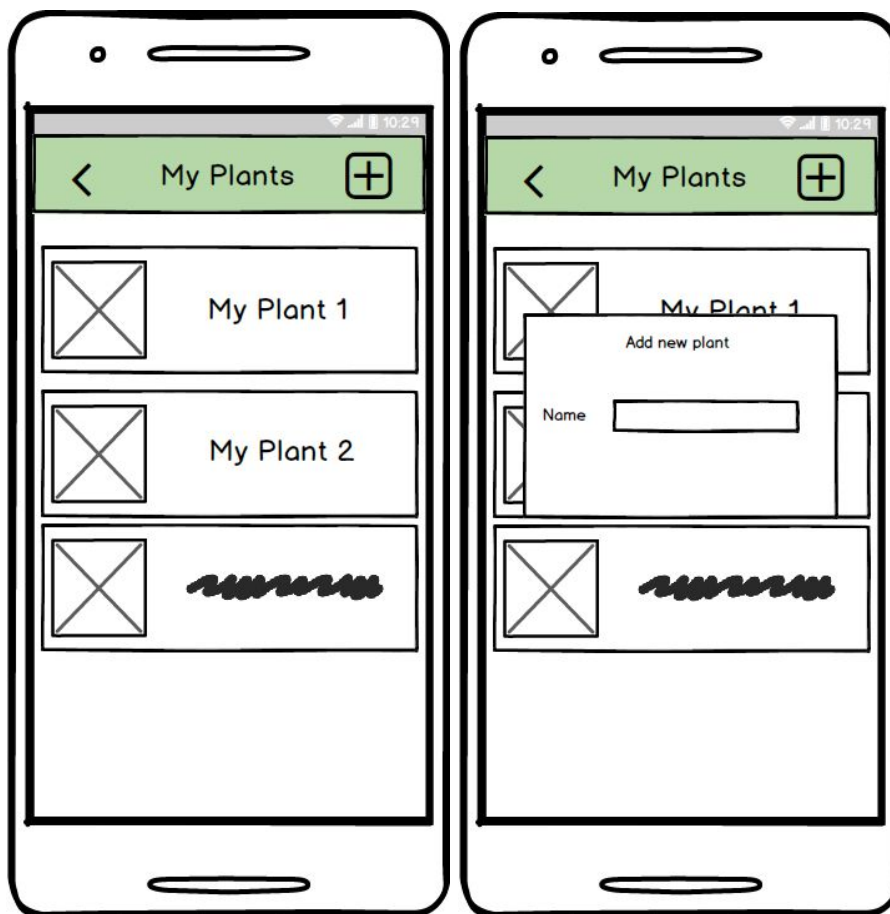| Date	Version	Description	Author
04/Dez/2016	1.0		Olga Akymenko
08/Dez/2016	1.1	Added Espresso test and mockup	Chris Todt
19/Dez/2016	1.2	Changed Diagramm	Olga Akymenko

Table of Contents

Use-Case Specification: Manage plants

1.   Manage plants

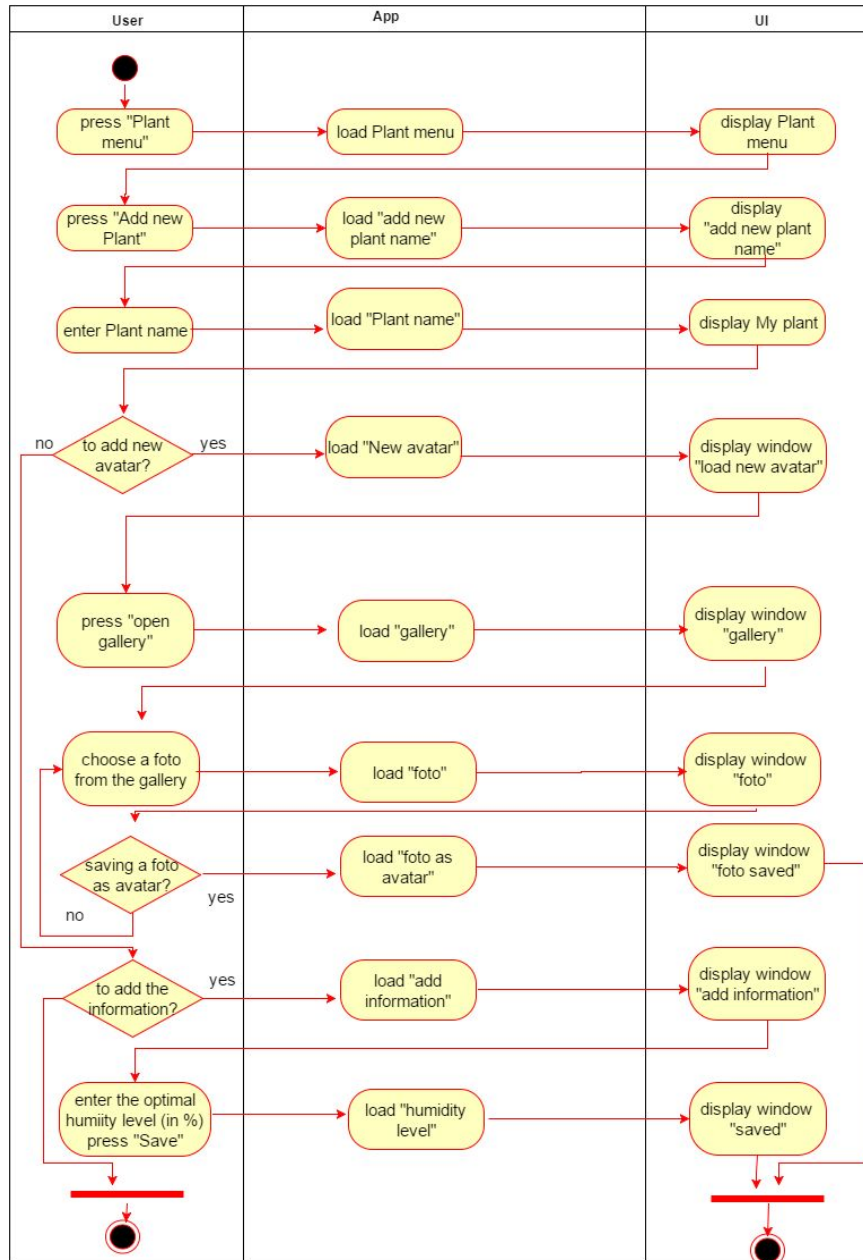### 1.1   Brief Description

User has the ability to add a new plant in the list. Also, change the image and add information about the plant (the optimum humidity level).

1.2   Mockup

2. Flow of Events

2.1 Basic Flow

User	App	UI
press "Plant menu"	load Plant menu	display Plant menu
press "Add new Plant"	load "add new plant name"	display "add new plant name"
enter Plant name	load "Plant name"	display My plant
to add new avatar? (no / yes)	load "New avatar"	display window "load new avatar"
press "open gallery"	load "gallery"	display window "gallery"
choose a foto from the gallery	load "foto"	display window "foto"
saving a foto as avatar? (yes / no)	load "foto as avatar"	display window "foto saved"
to add the information? (yes)	load "add information"	display window "add information"
enter the optimal humiity level (in %) press "Save"	load "humidity level"	display window "saved"

2.2   Alternative Flows

(n/a)

3.   Espresso test

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


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 android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;

import org.hamcrest.Description;
import org.hamcrest.Matcher;
import org.hamcrest.TypeSafeMatcher;
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.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;

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

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

   @Test
   public void uCManagePlants() {
       ViewInteraction appCompatButton = onView(
               allOf(withId(R.id.BTNmyplants), withText("My plants"),
                       withParent(allOf(withId(R.id.mainmenu),
                               withParent(withId(R.id.activity_main_menu)))),
                       isDisplayed()));
       appCompatButton.perform(click());
```

```java
        ViewInteraction tableLayout = onView(
                allOf(childAtPosition(
                        allOf(withId(R.id.LSTplants),
                                withParent(withId(R.id.activity_plant_menu))),
                        0),
                        isDisplayed()));
        tableLayout.perform(click());

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

    }

    private static Matcher<View> childAtPosition(
            final Matcher<View> parentMatcher, final int position) {

        return new TypeSafeMatcher<View>() {
            @Override
            public void describeTo(Description description) {
                description.appendText("Child at position " + position + " in parent ");
                parentMatcher.describeTo(description);
            }

            @Override
            public boolean matchesSafely(View view) {
                ViewParent parent = view.getParent();
                return parent instanceof ViewGroup && parentMatcher.matches(parent)
                        && view.equals(((ViewGroup) parent).getChildAt(position));
            }
        };
    }
}
```

4. Special requirements

(n/a)

5. Preconditions

5.1 Gallery

User has installed the application and is in the "Plant menu". The phone has at least one photo in the gallery.

6. Postconditions

6.1 Avatar/Information

The plant will be have avatar.

6.2 Humidity data

The information about the level of humidity data will help to ensure that the user would received a message about the need for watering in time.

7. Extension Points

(n/a)