Test plan

Project: Social Buddy (7/8 2024)

Requirements that are being tested are listed in the "Features_Requirements_Resit" excel sheet. *Note that the requirements are listed based off the cell row. e.g [A2 corresponds to the row with A2]*

This report details the planned testing strategies for the various requirements and features of the Social Buddy Project. Each heading section details the testing plan of the developed features (Flic button, Distance Calculation).

Test Environment:

- Android Tablet
- Tile Pro tag
- Tablet with the test version of the app

Test Setup:
1. Open the test application on the Android tablet.
2. Ensure that the Flic button is always connected to the tablet.
3. Ensure that the Tile tag is active.

Flic Button

This section covers testing related to the flic button feature

[REQ A2] Case A1: The buddy bot receives input from the flic button

The objective of this test is to verify that the buddy bot receives input from the flic button. The Flic button keeps track of how many times it has been pressed in its lifetime. The amount the button is clicked during this test will be compared with the amount it has been pressed in its lifetime to control that the button presses are received correctly

Execution

1. Connect the tablet to the laptop
2. Connect the flic button to the buddy app
3. Note the number of times the button has been clicked throughout its lifetime
4. Click on the flic button
5. Observe the terminal for button presses
6. Check if the number of times the button has been clicked throughout its lifetime has changed

Authors: Ahmet Oral, Binh Nguyen Do, Khizer Butt, Terrence Zhong
Supervisors: Sandra Hekkelman en Alexander Slaa

Acceptance criteria

The terminal displays a “button pressed” message. The number of clicks has changed to the correct amount.

[REQ A4] Case A2: Display the button press on the buddy Bot

The objective of this test is to verify that the button press correctly triggers the medicine taken function and plays an animation on the buddy app.

Execution

1. Ensure that the buddy app is opened on the tablet
2. Click on the flic button
3. Observe the app for any popups prompted by the flic button

Acceptance Criteria

The buddy app displays the “medicine taken” notification when the flic button is pressed. And updates the candidate's next medicine date.

Distance tracker (Tile Tag)

[REQ A3] Case B1: Determine the distance between patient and the buddy bot

This test was executed on a prototype feature test app. The aim of this is solely to test the distance tracking feature between the tag and the buddy app.

Execution:

1. **Open the testing app**
2. **Click on the tile tag on the list of connectable devices**
3. **Connect to the tile tag and connect to it**
4. **Observe the distance values calculated by the application.**

Acceptance Criteria:
The distance calculation and the actual distance has a discrepancy of less than 5m’s

[REQ A5] Case B2: Detect when the user strays further than 10m’s from the buddy bot

Context: due to low reliability of the current implementation of the distance determination feature. The testing scope has decreased from 50m to 10m.

Before execution: Have the scanResultlist print each incoming result into the terminal. And calculate the RSSI value for 10m's, set the boundary point at that RSSI value.

Execution:

1. **Open the buddy bot app on the bluetooth24 branch.**
2. **Open the Distance sensor screen from the main app drawer.**
3. **Observe the UI element with deivceID, rssi, est distance and status.**
4. **Note down the observations of UI and terminal output**

Acceptance criteria:

The scan Result updates frequently and prints on the terminal , and the UI element turns red when the distance exceeds the set boundary.

[REQ A8] Case B3: The buddy bot can send a message to the firebase instance when a user is out of bounds.

Before execution: Open the firebase, instance and there should be an entry in the bot entry fields for all tile tracker messages.

Execution:

1. **The buddy bot should be in the near vicinity of the user (less than the warning distance from the buddy bot (x<10m)**
2. **The user moves beyond the range of 10m**
3. **Observe the firebase entries in the meantime**

Acceptance criteria:
The buddy bot sends a message to the firebase **ONLY** when the user has strayed beyond the set limit. And it sends a message every time the user has passed the limit.

[REQ A7] Case B4: The buddy bot can reconnect to the determined tag when their connection is broken due to several reasons

This test aims to determine the reliability of the distance determination of these 2 devices. During the development, the connection between the tag and the buddy bot can be deemed unstable, as the connection can be broken by:

- User terminated connection
- Unknown reasons

The first reason may be caused by the tag, as it does not want a constant connection from any unknown devices. This connection is established via direct Bluetooth Low Energy (BLE), so the tag may not know who is connecting to it. The main way to connect to the tag is via an application made by the manufacturer. However, there is no API available in flutter for us to make use of it.

The second reason can also be caused by the way the 2 devices are connected. The BLE signal can be interfered with other signals, such as Wi-Fi, as they operate on the same frequency (2,4 GHZ).

A constant connection is therefore not possible. However, can reconnections upon disconnections work? How reliable are the reconnections? This test is here to find out.

Execution:

1. **Make sure the tablet and the tag are in connection range. Preferably the smallest possible distance first to determine if further testing is necessary.**
2. **Open the BLE test app and choose the tag on the device list.**
3. **Turn on the reconnection mode and connect to the tag.**
4. **Observe the reconnection when a disconnection occurs.**
5. **Notes down the time it takes to reconnect, or whether the connection can be re-established.**
6. **Based on the first findings, determine if further testing is necessary. If applicable, repeat all the steps with different distances.**

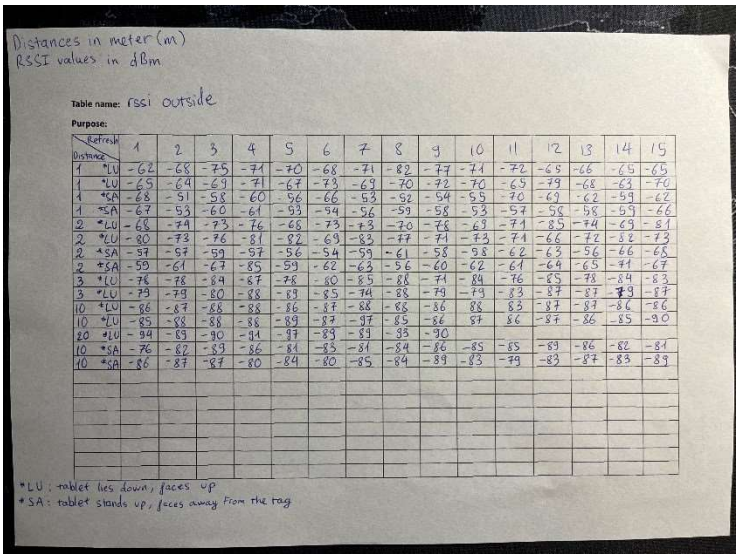
Acceptance criteria:

The reconnections can be established when a disconnection happens within the connection range. The time it takes to reconnect should not exceed 5 seconds.

Appendix Test Report.

Case B1:

indoor result



Distances in meter (m)
RSSI values in dBm

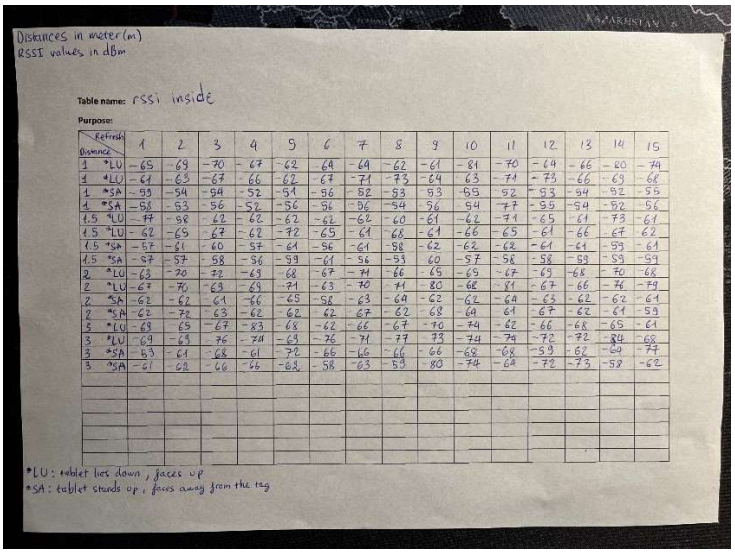
Table name: rssi outside

Purpose:

Refresh / Distance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 *LU	-62	-68	-75	-71	-70	-68	-71	-82	-77	-71	-72	-65	-66	-65	-65
1 *LU	-65	-64	-69	-71	-67	-73	-69	-70	-72	-70	-65	-79	-68	-63	-70
1 *SA	-63	-51	-58	-60	-56	-66	-53	-52	-54	-55	-70	-69	-62	-59	-62
1 *SA	-67	-53	-60	-61	-53	-54	-56	-59	-58	-53	-57	-58	-58	-59	-66
2 *LU	-66	-74	-73	-76	-65	-73	-73	-70	-78	-63	-71	-85	-74	-69	-81
2 *LU	-80	-73	-76	-81	-82	-69	-83	-77	-71	-73	-71	-66	-72	-82	-73
2 *SA	-57	-57	-59	-57	-56	-54	-59	-61	-58	-58	-62	-63	-56	-66	-68
2 *SA	-59	-61	-67	-85	-59	-62	-63	-56	-60	-62	-61	-69	-65	-71	-67
3 *LU	-76	-78	-84	-87	-78	-80	-85	-88	-71	-84	-76	-85	-78	-84	-83
3 *LU	-79	-79	-80	-88	-89	-85	-74	-88	-79	-79	-83	-87	-87	-79	-87
10 *LU	-86	-87	-88	-88	-86	-87	-88	-85	-86	88	83	-87	-87	-86	-86
10 *LU	-85	-88	-88	-88	-89	-87	-97	-85	-86	87	86	-87	-86	-85	-90
20 *LU	-94	-89	-90	-91	-97	-89	-89	-93	-90						
10 *SA	-76	-82	-89	-86	-81	-83	-81	-84	-86	-85	-85	-89	-86	-82	-81
10 *SA	-86	-87	-87	-80	-84	-80	-85	-84	-89	-83	-79	-83	-87	-83	-89

*LU : tablet lies down, faces up
*SA : tablet stands up, faces away from the tag

outdoor result



Distances in meter(m)
RSSI values in dBm

Table name: rssi inside

Purpose:

Refresh / Distance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 *LU	-65	-69	-70	-67	-62	-69	-69	-62	-61	-81	-70	-64	-66	-80	-74
1 *LU	-61	-63	-67	-66	-62	-67	-71	-73	-64	63	-71	-73	-66	-63	-68
1 *SA	-53	-54	-54	-52	-51	-56	-52	-53	53	-55	-52	-53	-54	-52	-55
1 *SA	-58	-53	-56	-52	-56	-56	-56	-54	56	54	-77	-59	-54	-52	-56
1.5 *LU	-77	-58	-62	-62	-62	-62	-62	-60	-61	-62	-71	-65	-61	-73	-61
1.5 *LU	-62	-65	-67	-62	-72	-65	-64	-68	-61	-66	-65	-61	-66	-67	-62
1.5 *SA	-57	-51	-60	-57	-61	-56	-61	-58	-62	-62	-62	-61	-61	-55	-61
1.5 *SA	-57	-57	-58	-56	-53	-61	-56	-53	-60	-57	-58	-58	-58	-55	-57
2 *LU	-63	-70	-72	-69	-68	-67	-71	-66	-65	-69	-67	-69	-68	-70	-68
2 *LU	-67	-71	-69	-69	-71	-63	-70	-71	-80	-68	-81	-67	-66	-76	-79
2 *SA	-62	-62	-61	-66	-65	-58	-63	-69	-62	-62	-64	-65	-62	-62	-61
2 *SA	-62	-72	-63	-62	-62	-62	-67	-62	-68	-69	-61	-67	-62	-61	-59
3 *LU	-68	-65	-67	-83	-68	-62	-66	-67	-70	-74	-62	-66	-68	-65	-61
3 *LU	-69	-69	-76	-71	-69	-76	-71	-77	-73	-71	-79	-72	-72	-84	-68
3 *SA	-53	-61	-68	-61	-72	-66	-66	-66	-66	-68	-68	-59	-62	-69	-77
3 *SA	-61	-68	-66	-66	-68	-58	-63	-59	-80	-74	-69	-78	-73	-58	-62

*LU: tablet lies down, faces up
*SA: tablet stands up, faces away from the tag

Conclusion:

There are some discrepancies between measured and actual distance but it shouldn't impact the purpose of the implementation.

Link to test app: https://github.com/NguyenBDo/Project78-Integratie_beacons_met_Social-Buddy/tree/main/Source_Code/ble2/lib/src

Case A2:

Results

Test iteration	1	2	3	4	5	6	7
distance in m 3	Pass	Pass	Pass	Pass	Pass	Pass	Pass
10	Pass	Pass	Pass	Pass	Pass	Pass	Pass
15	Pass	Pass	Fail	Fail	Pass	Pass	Pass

Conclusion:

Based on the test iterations, it can be concluded that for the most common distances, the Flic button always responds. At longer distances, the response may sometimes fail, but this is not significant since the patient is usually not far from the Buddy Bot.