

325 Assignment 3 - Report
300393699
Gabriel Tennent

2 UX Decisions:

Left -> Right (Right being final product)

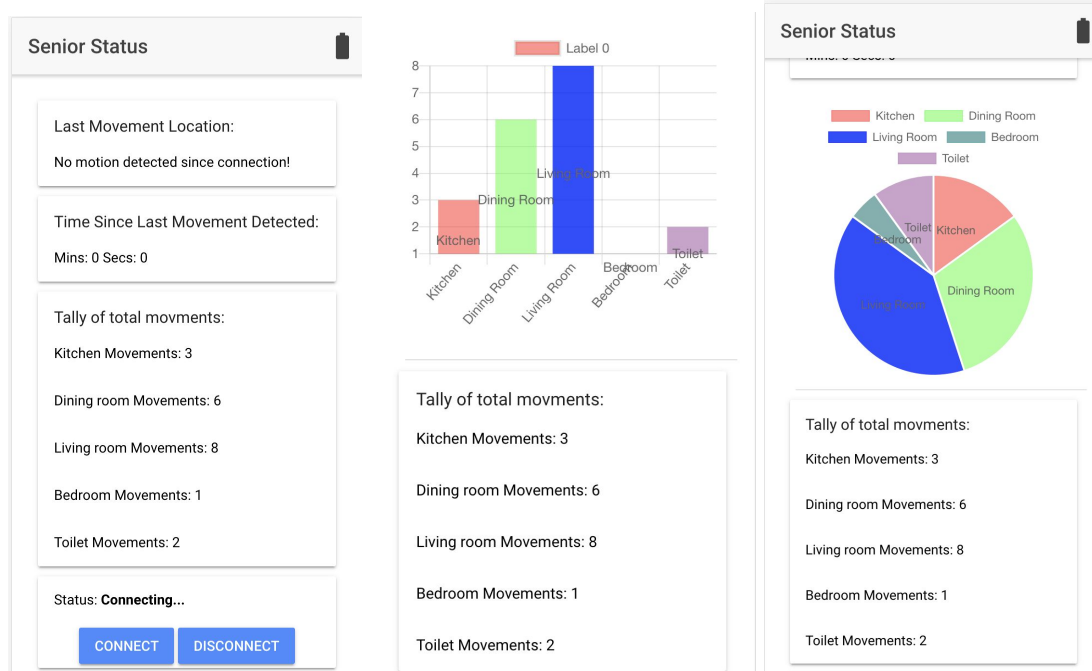
1st UX Design Decision:

The first UX decision I am going to discuss regarding this application is about how the data regarding the rooms visited was displayed. This UX decision defined the main page of the application as the primary purpose of the MQTT broker was to keep track of the person.

Initially I had the data displayed in a list format. The positive of this approach was that it was really clear how many times the person had visited each room without any further input from the user. The negative of this approach was the amount of white space the list displayed. Coupled with how much white space is on the battery screen made the application feel bland. It also didn't draw the eyes of the most important piece of information as it was just a white list.

The second option was a bar chart. I kept the list system as well so people could scroll down if they were confused by the chart to see exactly how much each room had been visited. The bar chart added color to the screen and attracted the eyes which was great. The problem with the bar chart was when the application was loaded a lot of the rooms hadn't been visited or wouldn't be visited for a while so the chart was still white a lot.

This leads me on to the final option which was the pie chart. This chart was only displayed when a room was visited and lead to all the white space being taken up as soon as one room was visited as the whole chart would then be taken up by that chart. I also kept the list for more specific information still.



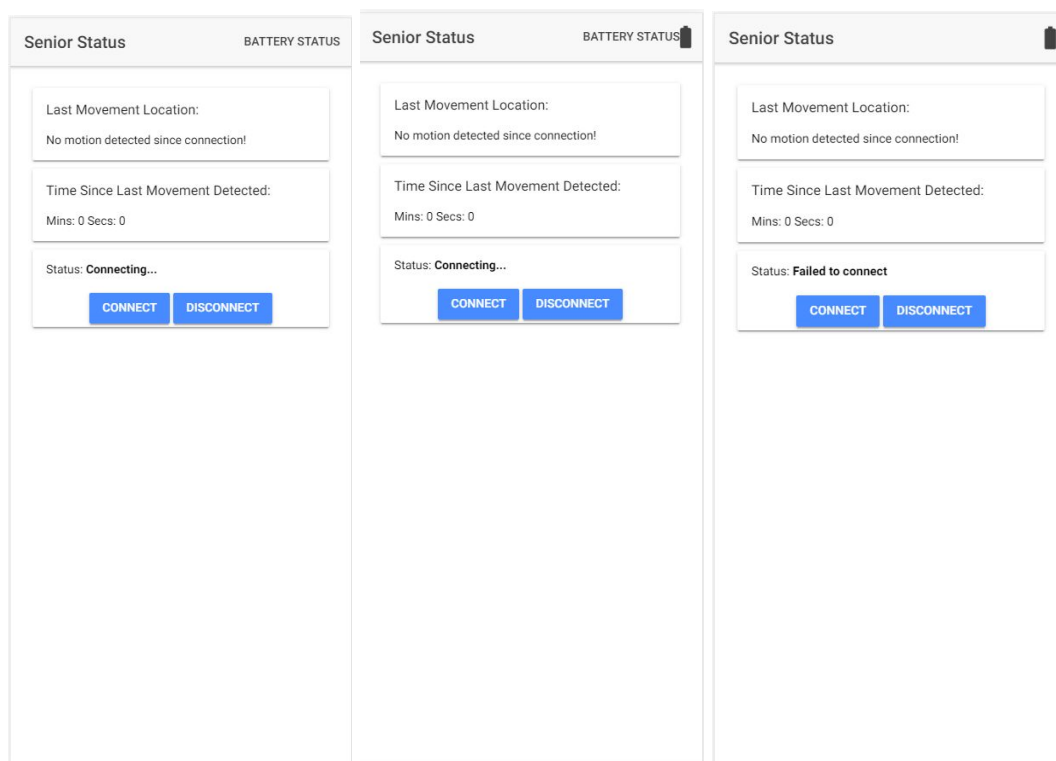
2nd UX Design Decision:

The second UX decision I made was regarding the buttons through the application, more specifically the Battery page button. This button navigates the user to the second page of the application which displays the percentages of all of the batteries in the MQTT server.

Initially I had the button displayed with the writing Battery status over it. The positives of this were that the button was easy to see and clearly indicated where the user would navigate to upon pushing. The issues were that the button looked out of place from the rest of the title bar and looked more like a title as opposed to a button.

The second approach was to add a battery icon into the button. The positives of this were that the user could now understand where the button was going to take them without having to read the text. The negatives being the text still looked out of place.

The final approach I went with was using the Ionic icons format and replacing the entire button with a battery icon instead of also including the text. This didn't feel out of place at all and still indicated to the user where they would navigate without using words.



Completion page discussion:

Though I never got to fully implement my own page as I ran out of time, I will briefly discuss what the purpose of the page was. An issue I found with the application was that it only tracked one room, as the serverStatus page only connected to one page. This means if you were to be taking care of fifty elderly you may need to install a unique application for each of them with all applications providing identical features. My page was working towards providing an additional feature that allowed the user to connect to unique subjects prior to entering the serverStatus page. This would mean the user could select which MQTT connection they would like to receive messages from and in turn have a list of multiple rooms to select from on a single application.

Possible issues for behaviour of the application:

A lot of issues can have detrimental effects on my application. Most of them cannot be prevented but can be handled in an elegant manner to prevent them from having a severe impact.

One of the sensors part of the network could lose connection to the broker. This would prevent any information from the sensor reaching the broker in turn leading to no information being polled. In my case, if one or more of the sensors were to never connect their battery would read as 0% by default. If they were to lose connection during the application their battery percentage would remain what it is before. I think good design could change the battery percentage to -1% if the broker stops receiving information from the sensor. This would let the user know if a sensor failed to send a message to the broker right away. This implementation could be furthered by staying at 0% if the last reading received was 1-2% battery remaining as this would indicate to the user that the sensor ran out of battery as opposed to no longer responding as it should as sensors out of battery would read 0% and sensors that stopped responding with strong battery or upon start up would read -1%.

Another issue for the current design of my application is it doesn't know if the senior has stopped moving but a pet/other movement has been signalling the sensors. My application doesn't have a work around for this issue and it is a severe issue as the senior may be in trouble and the notification may not be triggered as there is still recorded movement from the living space. On the opposite end of the spectrum the senior may just be sleeping and the application is notifying the user that the senior has stopped moving every five minutes and needs to be checked on when in fact the senior is just asleep.

When seniors are asleep or out of the house for example and the application is indicating that they haven't moved for five minutes it is very difficult to know whether they are in trouble or doing one of the above. A work around for this could be that at certain times of the day when the elderly is known to be sleeping or out playing bridge the sensors do not need to notify that there has been no movement to the user and instead during the time when the elderly is known to be

at home and active the notification is issued if there is still no movement. This would prevent the user from being bombarded with notifications that are false calls and lead to the “boy who cried wolf” state of mind assuming that every notification of inactivity is not something to worry about.

A final issue is if the mobile application fails to connect to the MQTT server. This would prevent any information being sent from the broker reaching the phone. This could lead the user to think the sensors are broken when in fact it is their connection to the broker. My current approach to letting the users know the issues in fact with their mobile connection to the broker is by having a connection status displayed above the disconnect and connect button. This implementation could be furthered by having a notification pop up if the connection fails ensuring the user is aware of the problem without having to navigate to the connection status.