SWEN325 Assignment 3: IoT

Report by Thomas Herdson 300391820

# Introduction

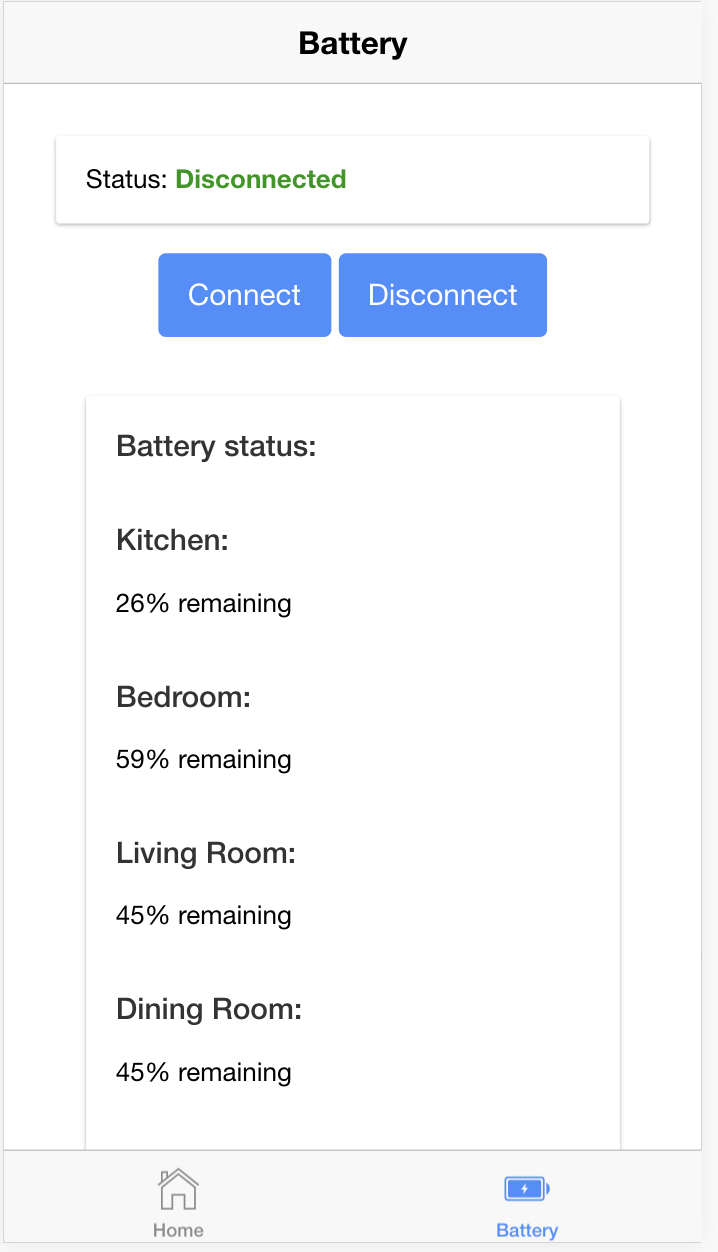
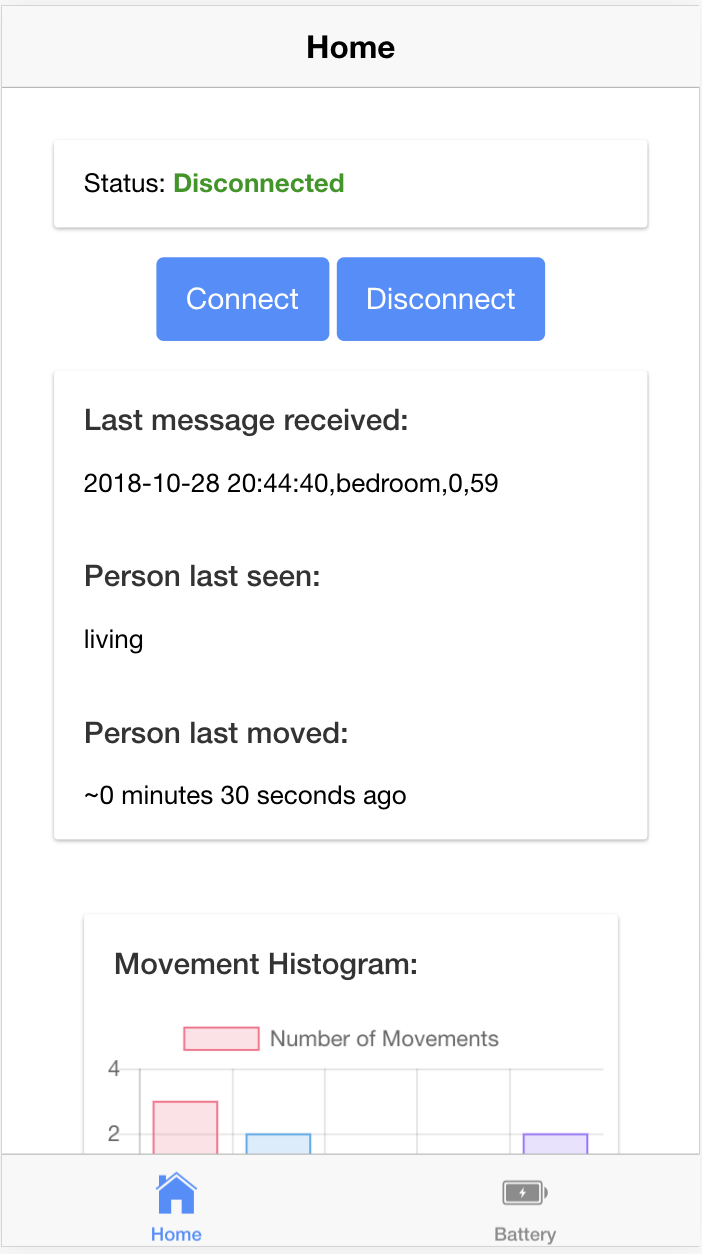
In this assignment, I created an Ionic mobile application that allows the user to track an elderly person in their home. The user can see where in their accommodation the elderly person was last detected, as well as how long ago they were last detected. As well as this, the application triggers a notification if the user was not seen in the last five minutes.

The development of this application was facilitated through MQTT, and the Paho MQTT library.

# UX Decisions

## Decision One: Cards

The core user interface of my application is achieved using Ionic cards.



The screenshots show how each section of information is grouped together into a card, this makes it clear to the user what information belongs together and makes each page look more coherent. For example, there is a clear seperation of the graphs from the information about where the person was last seen etc.

Cards are becoming an increasingly common design trend in mobile application development Card layouts put information — images, text, buttons, links, etc. — into a series of rectangular containers. These blocks can be layered or moved and tend to adjust to the size of the screen, stacking and falling into columns if you turn your phone on its side. The cards used on the analytics page allow users to neatly and cleanly see lots of information at once, and clearly shows the limits of each separate component. (source: https://thenextweb.com/dd/2015/11/16/why-cards-are-dominating-mobile-design/)

The cards are also good for extensibility, meaning that in the future functionality can be added such that cards can be swiped away and interacted with. Alternatives to this design that I considered were Ionic Segments, which would be useful for displaying the graphs. For example the user could select the tab at the top of the segment of which graph type they want to see.

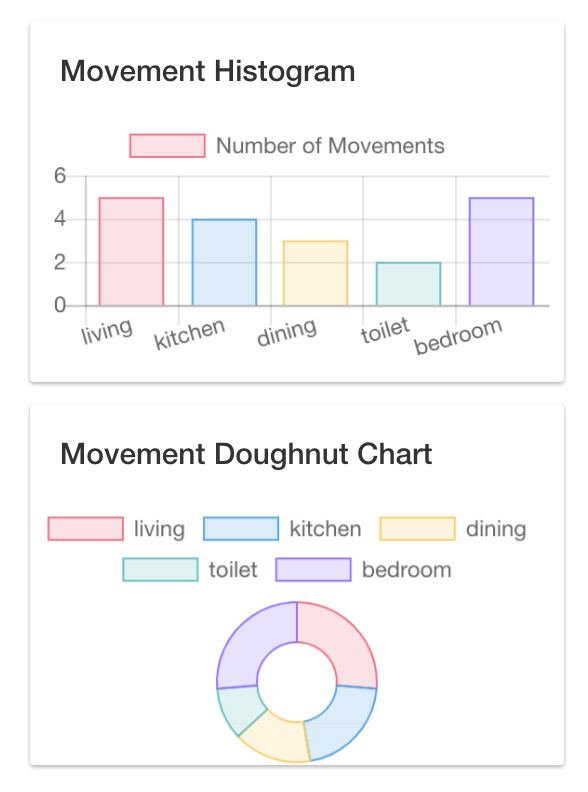


The problem I found with this type of display is that less information can be seen at a glance, and the user has to specifically go looking for information to find it. I also believe that with current mobile development trends, cards seem to be the obvious choice.

Another alternative to the cards would be to simply display all the information in one large container, without cards. The main reason I decided to use cards over this, is that cards provide a neater and cleaner appearance, and clearly show the limits of each component within each page. As well as this, cards are very easy to use when writing code.

## Decision Two: Visualisation of movements

In my assignment, for the displaying of a visualisation of the total number of movements in different areas of the house, I decided to use graphs. For this, I used the chart.js library, which is able to quickly draw beautiful charts. These charts are updated automatically each time there is new data.



The decision I made was to use a bar chart and doughnut chart, as opposed to another form of visualisation such as a heatmap. The first reason for this choice is that I believe a heatmap is less maintainable. For the heatmap, if the user were to change accommodation the heatmap would have to be changed, and potentially also the code. The charts inmy application can be easily adapted as required, for example if new rooms were to be added. Furthermore, a heatmap may be less easy to understand, and as well as this the layout of the accommodation may be irrelevant.

Another choice I had to make is which charts to use. I decided to use a bar chart, because it shows an easy to understand visualisation in which movement in different areas of the home can be compared by the height of the bar. More importantly, the exact number of times motion was detected can also be understood by using the numbers on the y-axis.

Additionally, I decided to use a doughnut chart, as they are excellent at showing the relational proportions between data (source: https://www.chartjs.org/docs/latest/charts/doughnut.html). This is especially useful in the case of understanding where motion is being detected within the accommodation.

# Possible Issues with IoT Application

## False Sensor Trigger

A key flaw in the current setup is that the sensors can be triggered by something other than the human in question. For example, pets can cause the motion detector to go off, as well as things like curtains blowing in the wind. This will cause the application to incorrectly report when the person was last detected, as well as the charts displaying incorrect information. This could be especially bad as the application is designed for tracking elderly people, so if something went wrong with their health and the application kept detecting movement, their life could be put in danger.

The solving of the issue lies mostly in the actual hardware implementation within the home, as opposed to anything that can be done in the software. For example, the sensors should have small levels of tolerance, so they do not trigger when a small object moves, such as a cat. This is similar to burglar alarms, that do not trigger with pets in the home. Furthermore, the sensor can be placed higher in the room.

## Poor connection

It must be an expectation that at times the network will go down, meaning that updates cannot be sent to the MQTT broker. This is a problem as the user of the application will stop getting updates, meaning that if the elderly person has issues, the user of the application will not be notified.

This can potentially be solved by the brokers using a mobile connection, as opposed to a Wi-Fi connection, as this could potentially be more reliable. Or for them to use both forms of connection. Additionally, they could be wired with Ethernet as opposed to Wi-Fi,

Furthermore, software changes should be made such that the application makes an alert if the broker has not sent a message in the past 30 seconds.

## Unnecessary Alerts

In the current implementation, even if the senior is sleeping, or leaves their apartment, an alert will be sent to the mobile application. This will be annoying as well as wasteful sending false alarms.

It would therefore be necessary for the elderly person or a carer to be able to switch off the sensors when leaving the house. Additionally, the sensors can be disabled from within the application.

## Sensors Failing

A number of factors could cause the sensors to fail. For example, if the sensors are battery powered they could run out of battery, or they could have a hardware defect, or be damaged. This situation needs to be taken care of, and I suggest that the broker sends a ‘not responding’ message for a sensor, which triggers an alert in the application that maintenance needs to be performed on a sensor.

As well as this, the sensors should be powered by mains as opposed to a battery, and potentially have a backup power supply.

# UX Design for IoT and Non-IoT Applications