Final Project

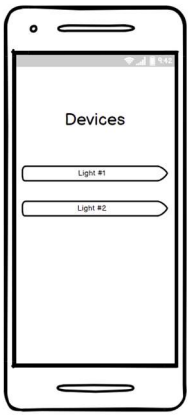
In the Final Project we will bring our learning together to build an IoT App. We will need to use the following modules in Expo to complete this project; native-base, react-navigation and react-native-paho-mqtt.

Before we can communicate with our IoT devices we need to establish a connection to our IoT Bridge.

A good starting place would be to specify the URL to the Bridge and send this data to the next screen.

A more advanced option would be to store the URL in local storage and load a default Bridge. The user could then edit the Bridge URL if required.

Use React Navigation to send the URL data to the next screen.

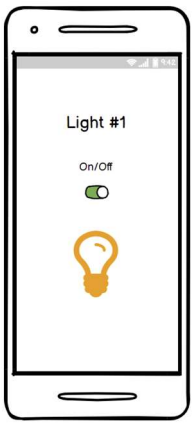


Our Devices screen should list all the devices currently attached to the Bridge.

We can use react-native-paho-mqtt to send a command to the Bridge to receive a list of connected devices.

Flatlist will enable us to loop over the array of devices returned by the Bridge. We will have to filter the array to remove items we do not want.

The device button should send all the necessary data for each device to the next screen.



The final screen should enable us to turn our IoT device on and off. We could use a graphic to display the status of the device.

A more advanced solution for this screen would be to create components from the data from the previous screen. For example; on/off, brightness and colour.

react-native-paho-mqtt

The Zigbee2MQTT documentation shows us how to send commands to the Zigbee2MQTT bridge.

https://www.zigbee2mqtt.io/guide/usage/mqtt_topics_and_messages.html

Here is a snippet of how to send an on/off command to a device using react-native-paho-mqtt. It would be better to get the bridge device list first and make the friendly name, 0x001788010b2efdb0, dynamic.

```
// connect the client
client.connect()
  .then(() => {
    // Once a connection has been made, make a subscription and send a message.
    console.log('onConnect');
    return client.subscribe('zigbee2mqtt/bridge/devices');
  })
  .then(() => {
    const message = new Message("on");
    message.destinationName = 'zigbee2mqtt/0x001788010b2efdb0/set';
    client.send(message);
  })
  .catch((responseObject) => {
    console.log(responseObject)
    if (responseObject.errorCode !== 0) {
      console.log('onConnectionLost:' + responseObject.errorMessage);
    }
  })
;
```