*Florida International University*

*School of Computing and Information Sciences*

Feature Document

User Story ID 229

**Team Member(s):** Jordan Laing, Galo Romero

**Project:** Biosensing 2.0

**Product Owner(s)**: Shekhar Bhansali, Yogeswaran Umasankar

**Mentor(s)**: Vishal Chopade, Apurva Sonawane

**Instructor**: Masoud Sadjadi

**User Story:** Merge modules into finished app

### **Description:**

* As a developer, I want to merge the BLE connectivity module (made by Galo) with my server connectivity/data display module into a single app

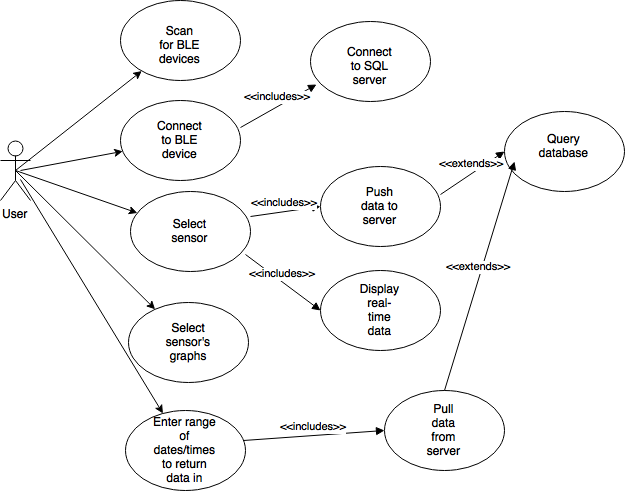
### **Acceptance Criteria:**

1. App can connect to BLE device and collect data
2. App can store data in SQL server
3. App can display data in graphs

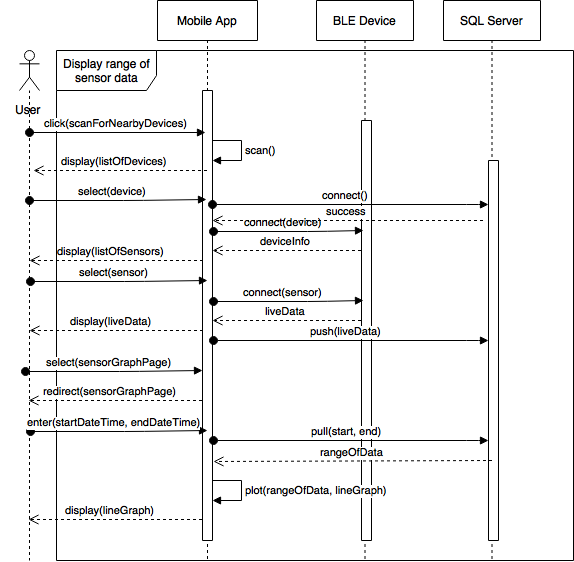
**Use Case:**

* **I**D: 229-01 - Connect to a sensor on the device and display a graph with a given range of data.
* **Actor:** User
* **Entry Condition:** This use case starts when the user scans for nearby BLE devies..
* **Flow of Events:**
  + The user clicks the button to scan for nearby BLE devices.
  + The system scans for for nearby BLE devices and lists those found.
  + The user clicks on the device they want to connect to.
  + The system connects to the device as well as the SQL Server and lists the sensors it recognizes.
  + The user clicks the sensor they want to connect to.
  + The system connects to the sensor, begins displaying real-time data, and begins pushing data collected from that sensor to the server every 30 seconds.
  + The user selects the graph for a sensor and clicks the Go button.
  + The system redirects the user to that sensor’s page.
  + The user enters a start date and time and an end date and time and clicks the Go button.
  + The system returns pulls the data in the given range for that sensor and displays it in a line graph.
* **Exit Condition:** This use case ends when the system displays the graph to the user.

**Use Case Diagram**



**Sequence Diagram**



**Testing**

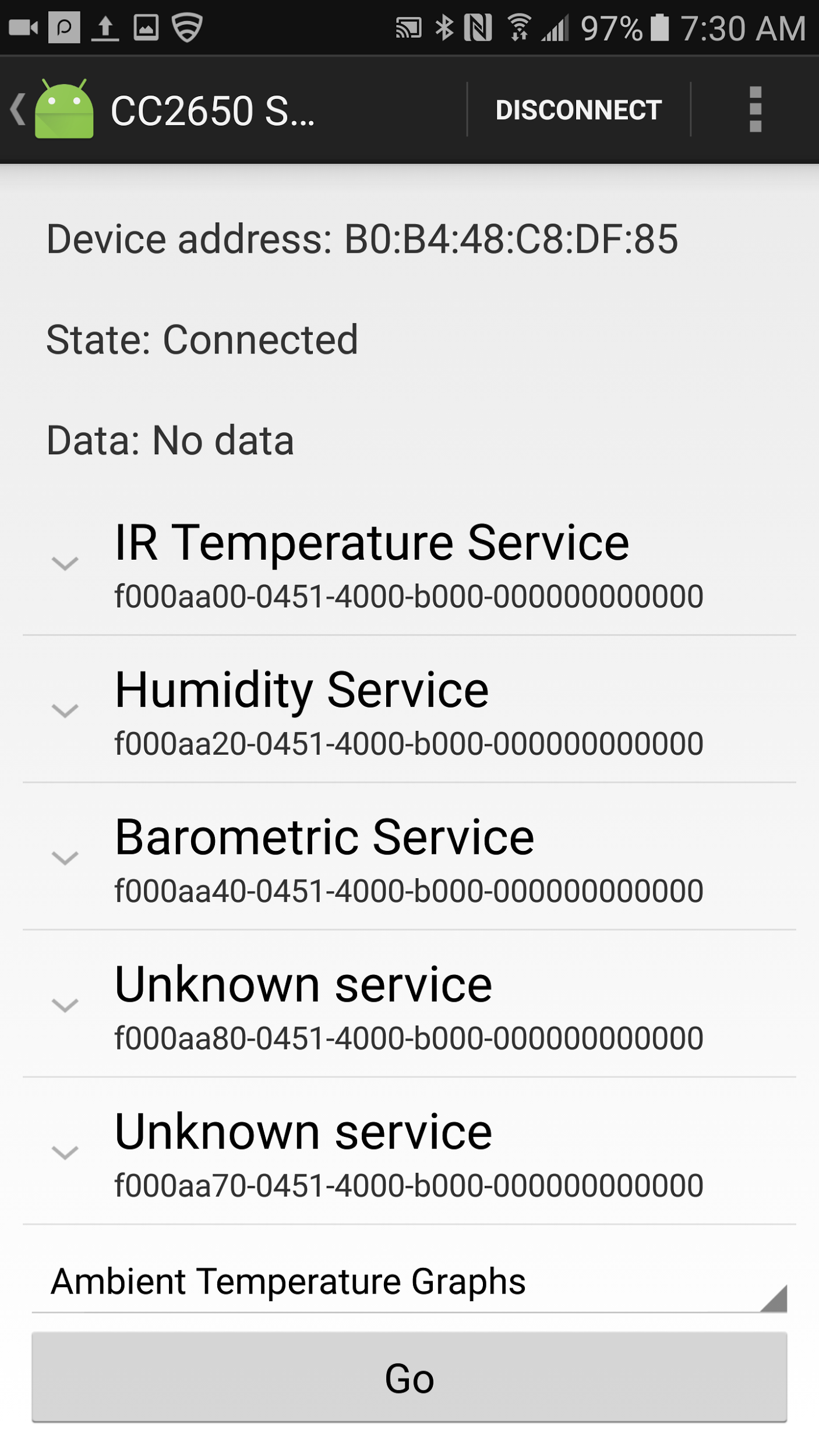
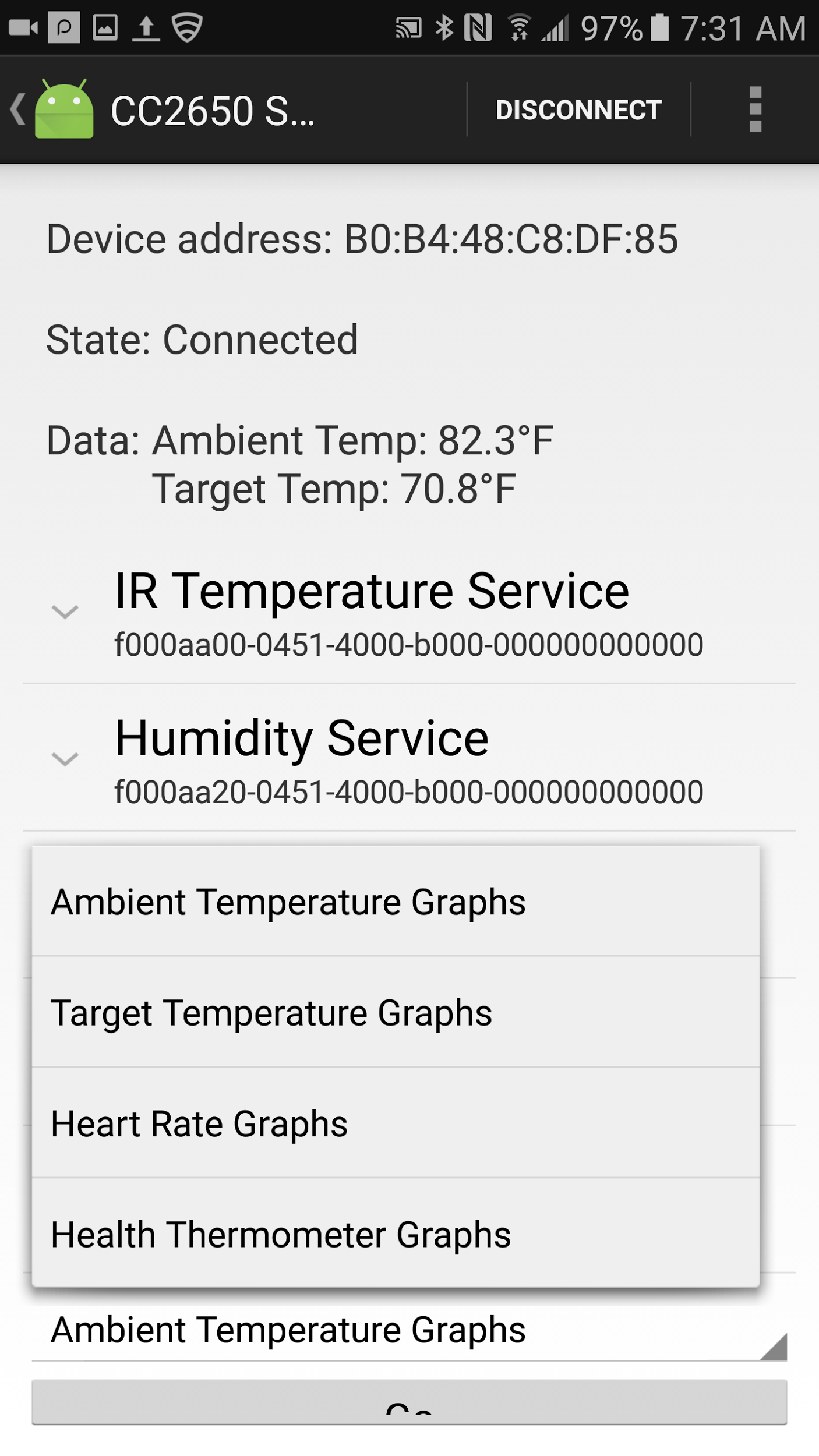
* Test Case ID: 229-001
* Purpose: To test if the mobile app can connect to a BLE biosensing device’s sensors, store the data it collects in a server, and display graphs with data pulled from the server
* Preconditions:
  + Android application package installed on mobile device
  + SQL Server running on same local network as mobile device
  + A BLE biosensing device (such as the T.I. SensorTag) is nearby and turned on
* Expected Result:
  + The app connects to the device’s temperature sensor and pushes the data to the server every 30 seconds
  + The app displays a user-given range of temperature data in a line graph
* Actual Result:
  + The app successfully connects to the T.I. SensorTag’s temperature sensor and pushes temperature data to the server every 30 seconds.
  + The app successfully displays a line graph with that sensor’s data within the range of date and times the user specifies

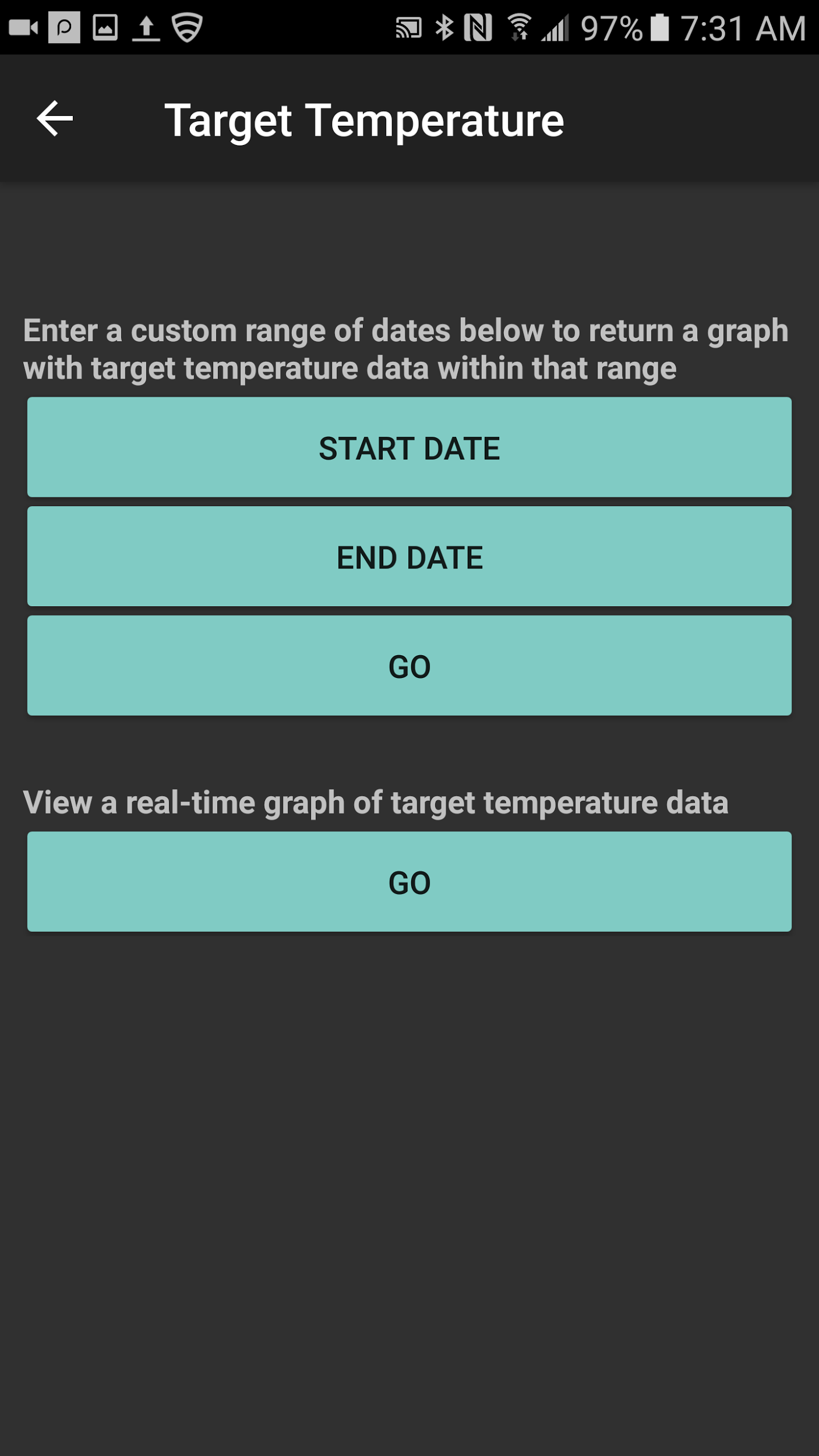
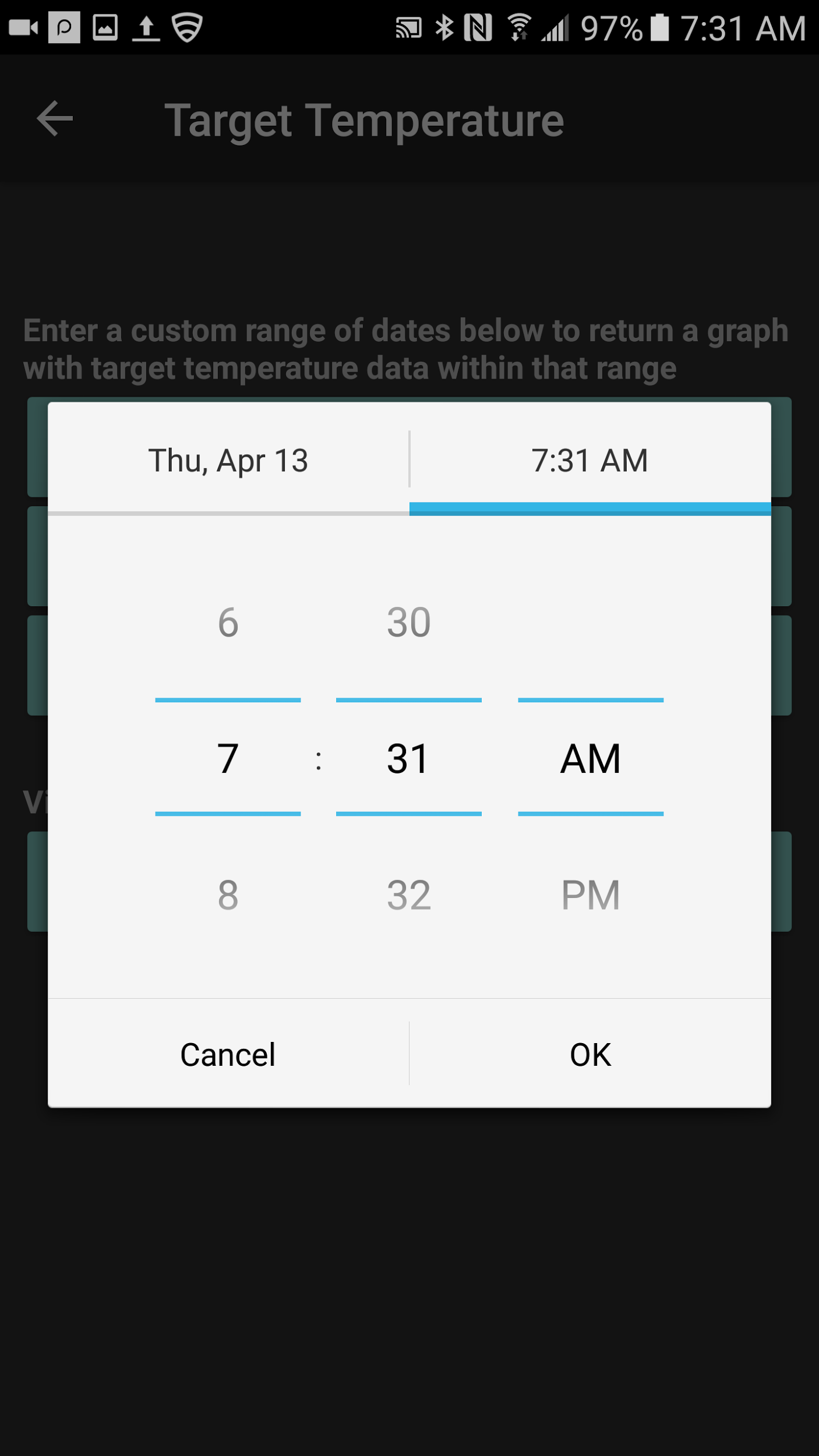
**User Guide**

Once the user opens the app, they can click the button in the top right corner to scan for nearby BLE devices. The app will display to the user a list of all the devices it finds. The user can then select the device it wants to connect to from this list, as shown below.

.

The user can select the device they want to connect to by clicking on its name. The app will then display all the services (including sensors) provided by the app. It is likely that it will not recognize some of the services, as shown below on the left.

Click on the service that corresponds to the sensor you want to access, such as the Temperature service. The app will connect to the sensor and begin displaying live data from it at the top of the page, as shown above on the right. Also shown in that screenshot is the spinner that allows the user to select which sensor’s graphs they want to view. Pressing the Go button will take them to the corresponding page, shown below on the left. 

On this page, there are two types of line graphs the user can view. The first represents the data in the range the user specifies using the date and time picker shown above on the right. The second is a real-time graph that plots the most recent 20 values for that sensor in the database and then begins updating the graph with new data every 30 seconds. Pressing either of the Go buttons will display the graph they chose to them, as shown below.

