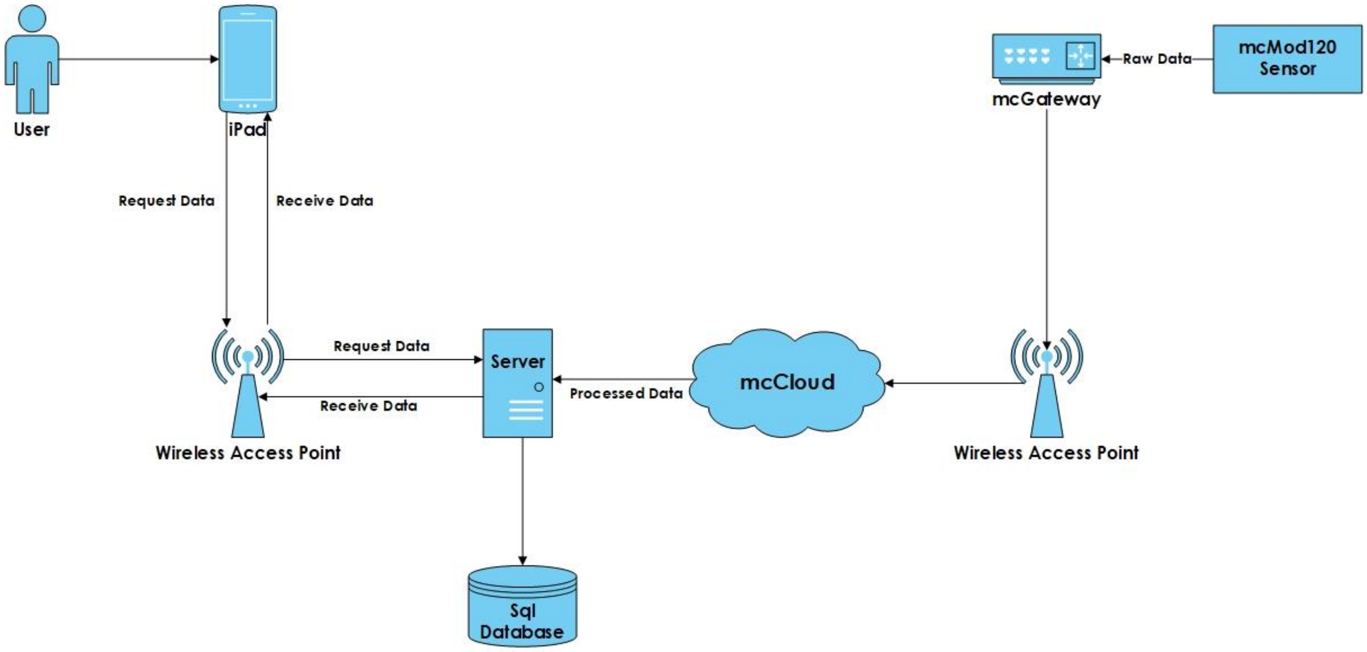
Final Design Proposal

ArraysAtOne: Maple Leaf Foods

Kevin Baumgartner, Jesse Berube, Marc Harquail, Alex Ireland

Software Architecture

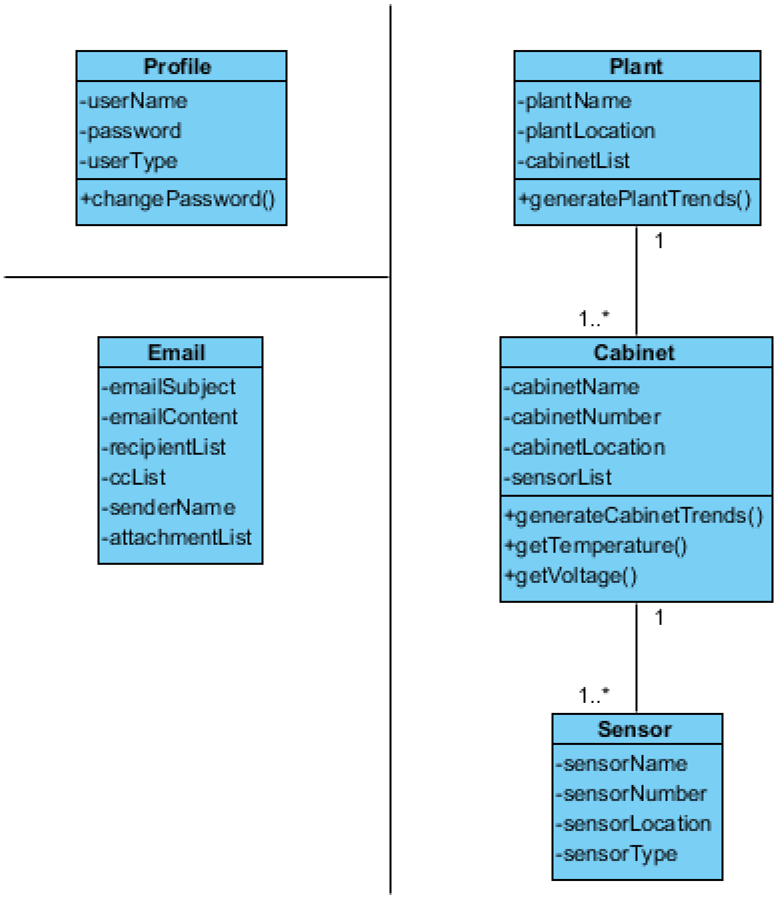


For this project, we have procured hardware and software from IoT Provider mcThings. They have provided us with the hardware we need, as well as the software that will assist us with transmitting the raw data from said hardware, to our database.

The system begins with the temperature sensor (Labeled mcMod120). This sensor captures raw, real-time temperature data. Each sensor automatically connects to the nearest mcGateway, which allows the data to be forwarded to the database. Before being sent to the database, mcGateway sends the data to mcCloud by connecting to the nearest wireless access point. mcCloud serves as our raw data interpreter, as well as a system monitor for all the connected sensors. Within mcCloud, we have written scripts which interpret the raw data, and outputs the formatted data that will be used in the application itself. This data then gets sent to our database using a Webhook. The database will eventually be located on a Maple Leaf server, once the project is up and running.

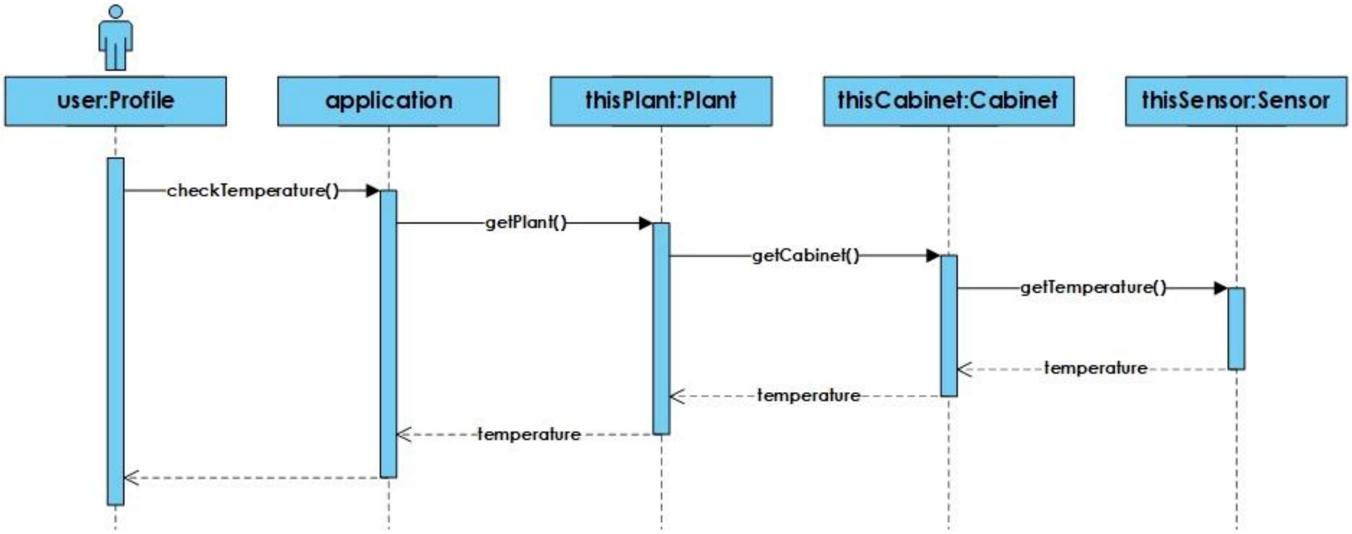
On the other end, once the user accesses the application, a request is made to get the most recent data from the database. Once this data is received, it is displayed to the screen. This request is furthermore continually made on a regular and frequent basis (likely every minute, or several seconds) to provide near real-time information on the status of the cabinets.

Class Diagram

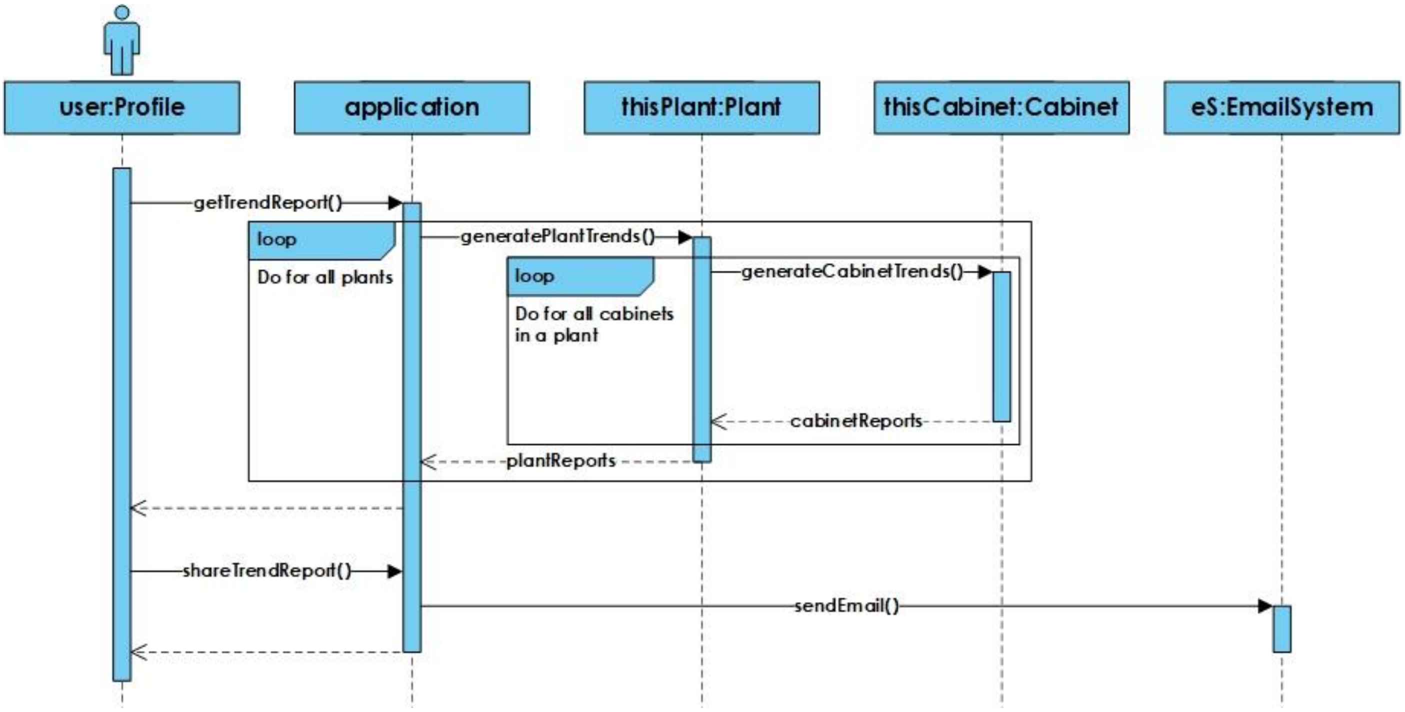


Sequence Diagrams

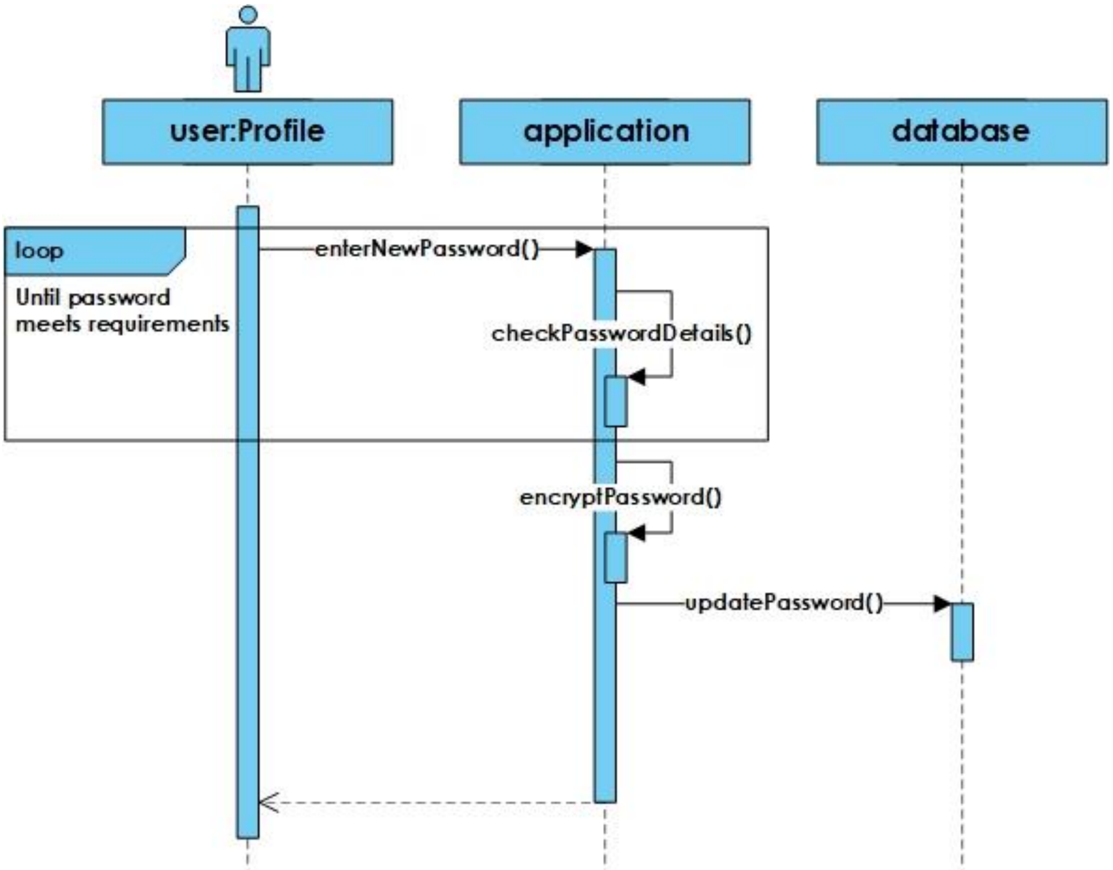
View Temperature Sequence



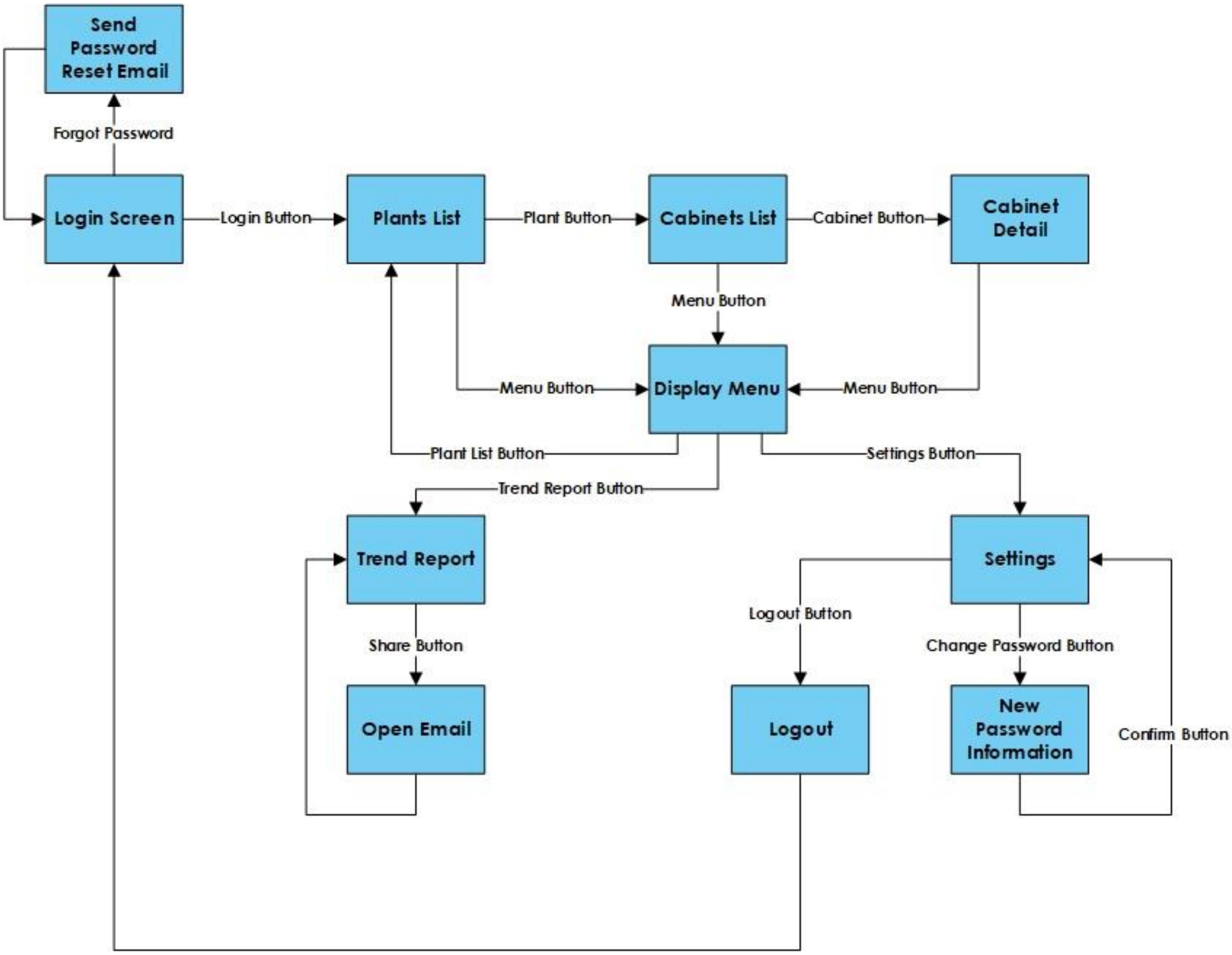
Generate Trend Report Sequence



Change Password Sequence

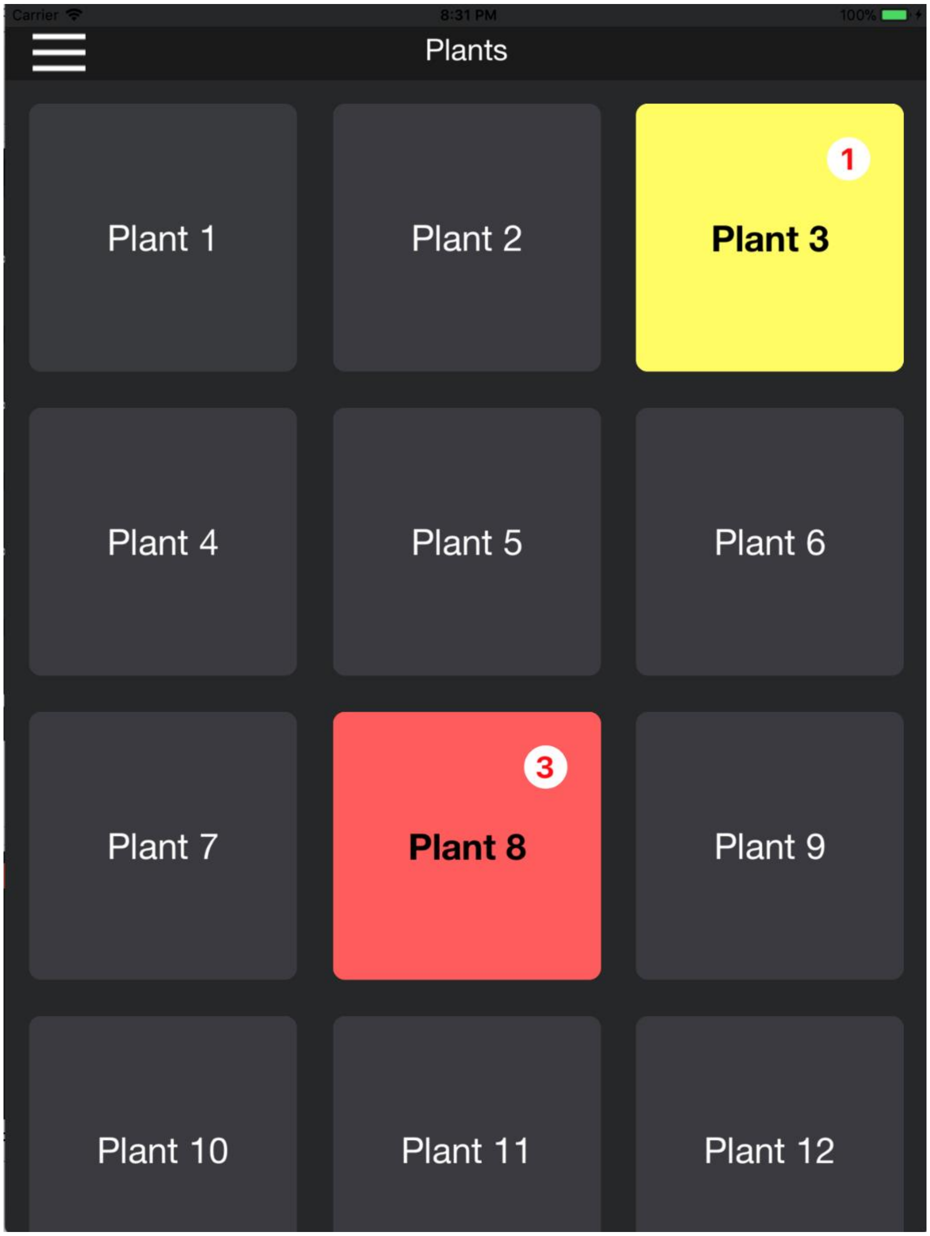


GUI Storyboard



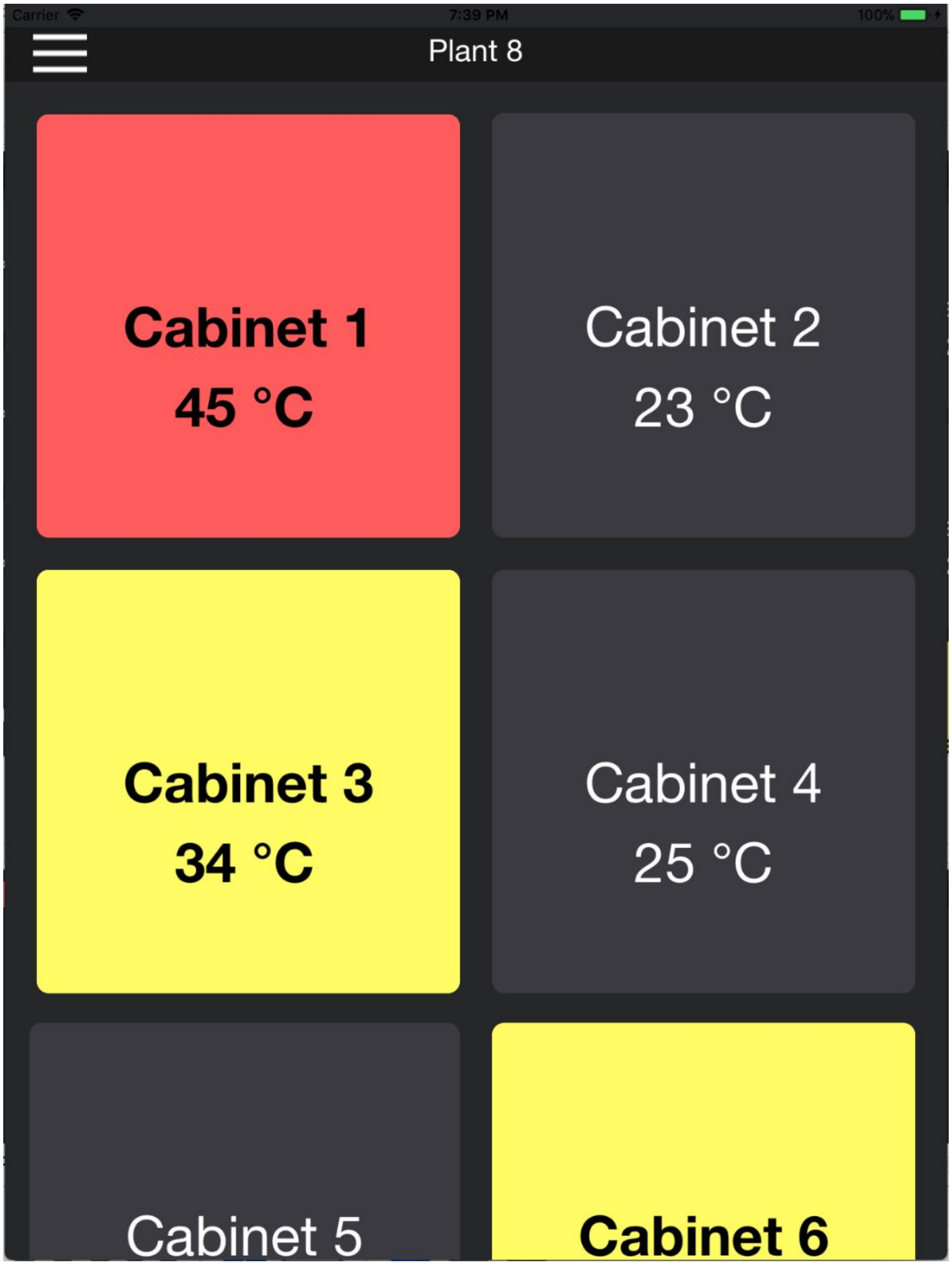
<u>GUI Wireframes</u>

Plants List view



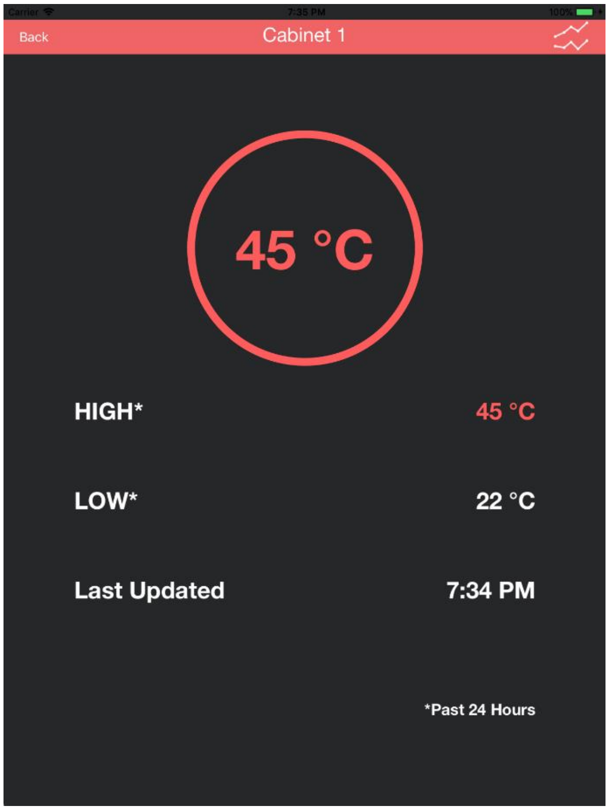
This view lays out all the plants in an easy to navigate and read grid view, with each item in the grid being a button that allows you to move on to see a list of cabinets for that specific plant. When any number of cabinets at a specific plant is experiencing issues, the colour of the button will change (yellow indicates temperatures at abnormal levels, red indicates temperatures at unsafe levels requiring immediate attention). Furthermore, a white badge is added to the top right of the plant's button indicating how many cabinets are experiencing issues.

Cabinets List View



The above is a view of a list of the cabinets located within a given plant. This would follow a similar design to the Plants list, to maintain stylistic consistency. Like the previous view, colour is employed to convey urgency, with each element being a clickable button. A current temperature reading is also added to the button to allow for a quick overview without having to click on each individual cabinet.

Cabinet Detail View



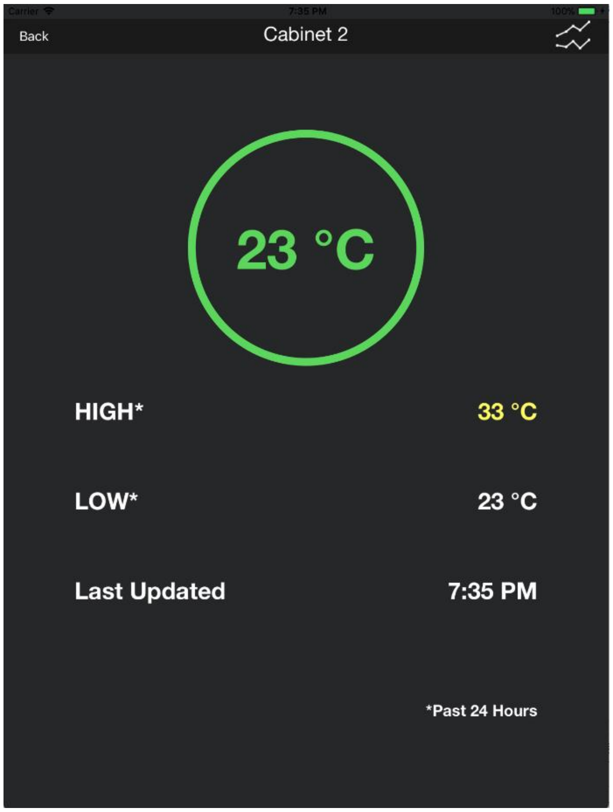
45 °C

HIGH* 45 °C

LOW* 22 °C

Last Updated 7:34 PM

*Past 24 Hours



23 °C

HIGH* 33 °C

LOW* 23 °C

Last Updated 7:35 PM

*Past 24 Hours

The above mock-ups are two different versions of the same view. Once again, the use of colour is employed to quickly convey the status of a cabinet, without having to think too much about the specific temperature. The high and low temperatures of the specific cabinet over the course of the previous 24 hours is also included, as well as the time the temperature was last updated. This can help with indicating potential issues with the cabinets environment or operation (ex: in cases where the low and high temperatures are 15 degrees apart).

Development Methodology

The software development methodology we will be using throughout development is Agile, as we will be working closely with Maple Foods during this project. There are a few tools that we will be using as well to achieve our goals.

To write our IOS application, we will be using XCode, as we will be using Macs to develop our app. During the development of our application, we will be using Bitbucket for version control, as it will be free for our needs. We will also need a database for our application, as we will need to store the measurements that our sensors pick up, therefore, we will be using SQLite, as it will be easy to use and manage, and has everything we need. For testing, we will be using both White Box and Black Box testing philosophies, as the application will be tested by both us and Maple Foods.

For our project, we have partnered up with mcThings, an IoT hardware and software provider. They have provided us with the sensors we will need, as well as the software we need to program their sensors. We will be using mcStudio, an application which allows us to directly program the sensors provided to us by mcThings.

mcStudio allows us to write, compile, modify, and debug code for our sensors. This will allow us to write scripts and load them directly onto hardware made by mcThings.   Alongside mcStudio, we will also be using mcCloud, a web service that will allow us to manage our mcThings devices from anywhere. Using mcCloud, we will be able to upload the scripts we write in mcStudio, as well as register our sensors to the website, which will allow us to compile and run our code on our sensors. mcCloud will even allow us to compile and run our code to the sensor from anywhere, so long as the sensor is connected to mcCloud.