Food Spy
Final Project Report

Abigail Kwan, Yamin Yee, Jeremy Escamilla

California State University, Long Beach

Department of Computer Engineering and Computer Science

December 16, 2019

What is our Project?

FoodTracker is an IoT integrated Fridge in which the user can track the status of food in the fridge, and keep the user visually aware of the fridge's status.

Why are we doing it?

People are busy (especially college students)! In today's world, people often have to juggle around multiple responsibilities at once: work, social life, and the dreaded bills. Food expiration and the state of one's fridge often falls by the wayside, assuming that they'll work without fault. Unfortunately, this isn't always the case: sometimes the power gets cut, other times multiple food items are already expired long before someone can have the time to go through each item and check for spoilage.

That's where we come in. We want to make it easier to quickly check the contents of the fridge, even when on the go!

Who are we?

Abby is the primary App developer who's responsible for the inception, design, development and testing of the app on mobile devices. In addition to the front-end, Abby is responsible for the back-end/server portion of the system and will share responsibilities with Yamin (given that the back-end is accessed via the Raspberry Pi 3) in that respect. She will also share the responsibility of the wireless communication between Raspberry Pi and Sensors with Jeremy, to ensure that the app can correctly update based on the sensors.

Yamin is the primary Fridge and Raspberry Pi developer of the group, and will be responsible for the layout/schematics, construction and testing of the fridge, in addition to the initialization, and setup of the Raspberry Pi. In the hardware implementation of the sensors with the Fridge and Raspberry Pi, she will be working with Jeremy, and in terms of the App programming on the Raspberry Pi and how it accesses the backend of the app, she will cooperate with Abby.

Jeremy is responsible for the Peripheral Devices of the System. From the temperature sensing to the visual acquisition via Camera Module, his job is to: research and find which hardware will be best for the cold (and sometimes cramped) fridge environment, to find the best placement for said devices to properly do their function, to program and test said devices, and work with Yamin to install said devices into the fridge. Currently, his responsibilities overlap with both the Fridge Hardware responsibilities (Yamin) and App programming into the Raspberry Pi (Abby), due to the integrated nature of the system.

How are we organized?

For the organization of our schedules and setting of deadlines, we utilize Wrike and the Gantt Chart functionality. For file-sharing and version control (with regards to documentation) we utilize Google Drive, and for basic person-to-person communication (when face-to-face isn't an option) we utilize the Discord App.

Task List

I. Abby

1. Create Menu (static display)
2. Create Food Profile Display
3. Finish Menu w/ clickable elements
4. Add navigation in app
5. Test connection to server
6. Incorporate options (edit, delete) for profiles
7. Input camera command into Raspberry Pi
8. Test camera command
9. Add settings
10. Test entire app

II. Yamin

1. Start building the Fridge's foundation.

2. Start building the detailed components (e.g. Raspberry Pi).
3. Integrate components with Fridge, and begin Raspberry Pi debug.
4. Test the fridge as a whole with the other components of the system (app and peripheral devices).

III. Jeremy

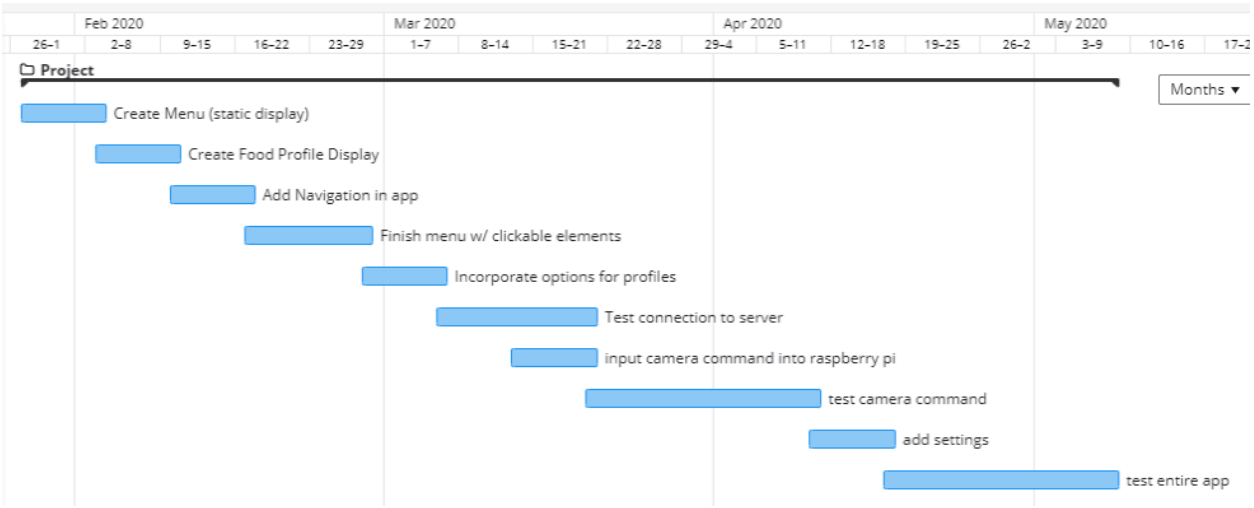
1. Begin driver installation necessary for the ESP32.
2. Upon product arrival, begin testing of the DHT device.
3. Upon verifying DHT functionality, test the bluetooth functionality of the ESP32.
4. If both function correctly, then verify that integration of both the DHT device and Bluetooth functionality are not only possible, but functional.
5. Test the life-expectancy of the temperature sensor, and verify that an alert can be sent once the battery runs low.
6. Test the complete Temperature Sensor System in fridge conditions and verify whether it can function as intended for a long period of time.
7. Work with Yamin to install the Raspberry Pi 2.8 Camera Module
    i. Figure out the placement, programming, etc. of the device with Yamin. In addition, interior light installation may become necessary for a better user experience and proper image acquisition.
    ii. Adjust the lense/focus and/or adjust placement for the sake of a better image for the customer.

iii. Work with Abby regarding food identification or whether it is based on user selection (or region-based, where the camera marks a region of the fridge for a specific item).

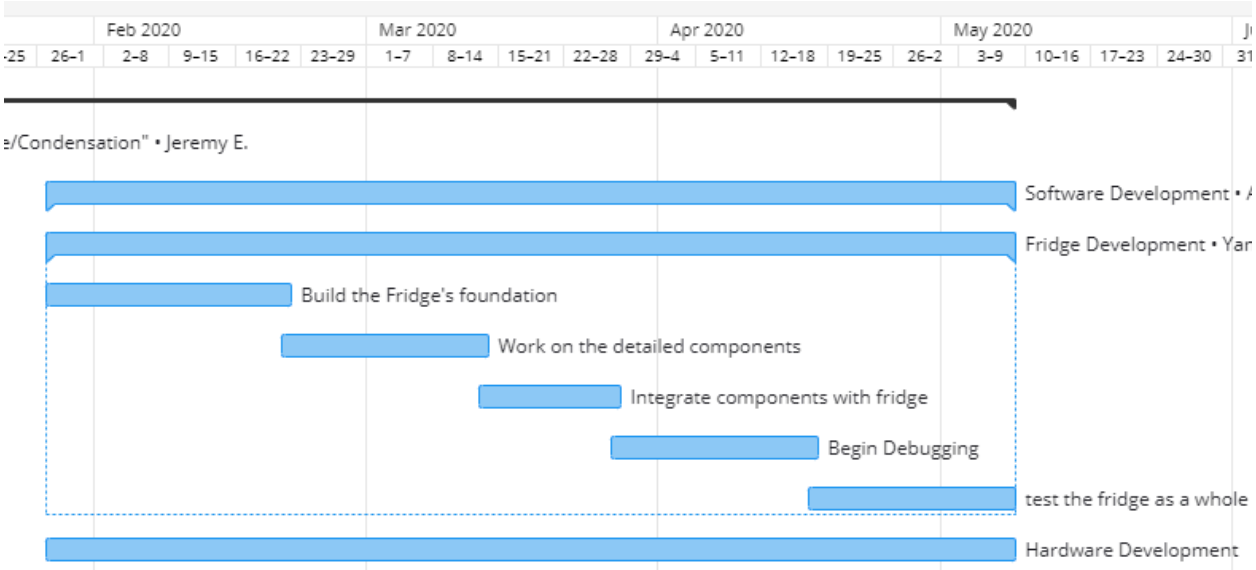
8. Work with Abby regarding the bluetooth functionality of the Temperature Sensing System.
    i. Test the updating of the temperature on the App, and ensure its accuracy.
    ii. Verify the activation of all alerts: (Low Sensor Battery, High Fridge Temperature, Food Expiration)

Timelines

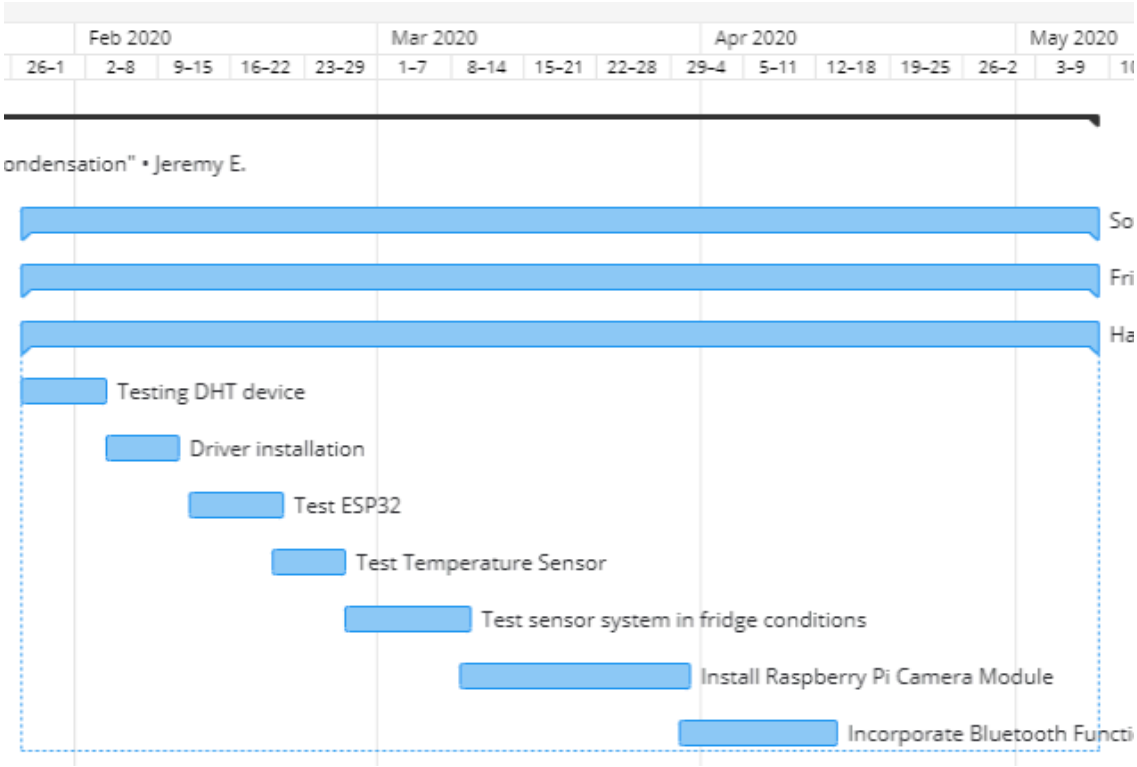
Abby:



Yamin:

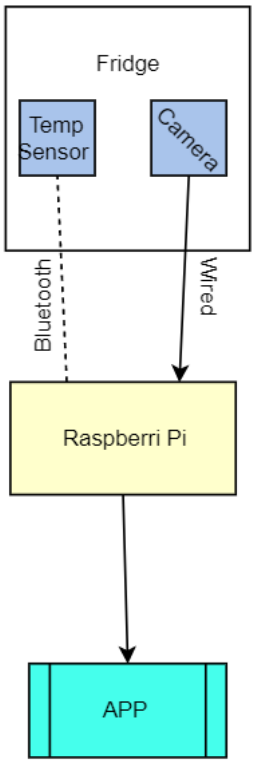


Jeremy:



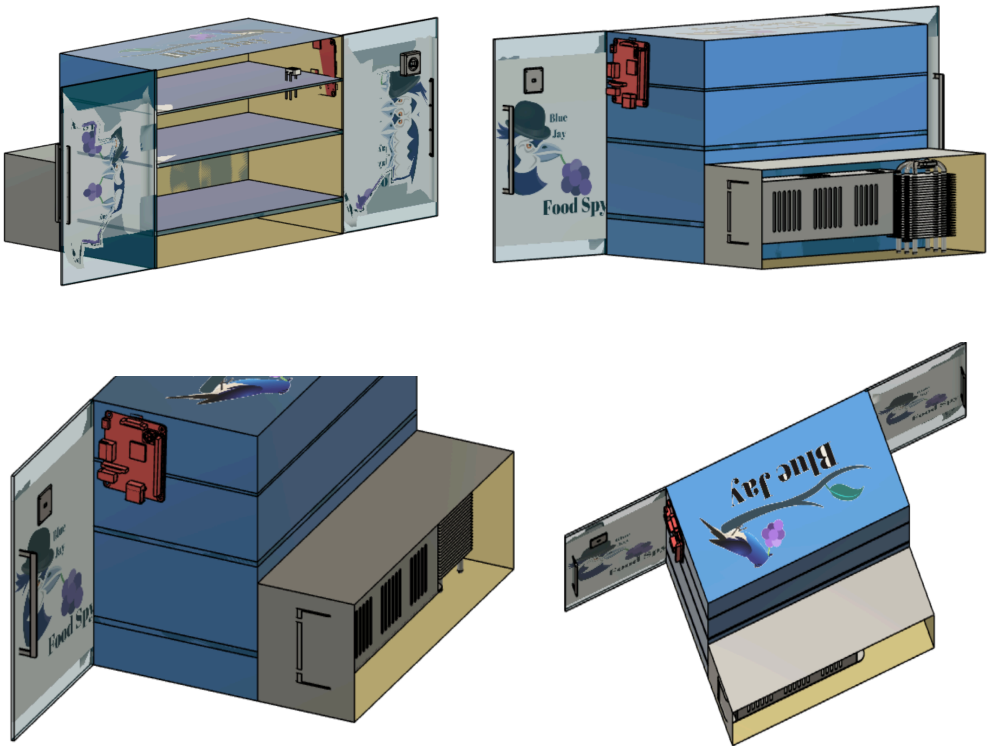
Hardware:

High Level Diagram



Fridge

Conceptual Drawing



Yamin created the design of the refrigerator, since she was the one who initially thought of this idea. Due to her friend's advice, she decided to use Fusion 360 to model the fridge so that the product looks alive rather than use 3D as the program. There will be two handles on the glass door of the fridge. The whole design is seen in the pictures above and also in this Youtube link: **the video of our 3D design Fridge.**

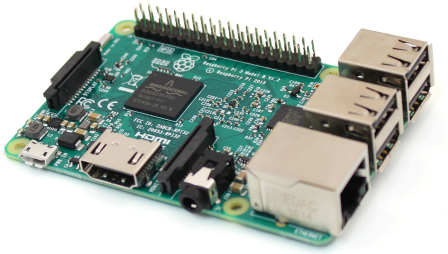
The Raspberry Pi 3 will be placed on the left side of the fridge in order to able to connect to pi camera since we installed pi camera on the inner right door of the fridge.

The transformers and the cooler will be in the back. As you can see we will have our formal logo on the top of the fridge and then the two stickers of the fridge will be placed on the doors.

Expected Features

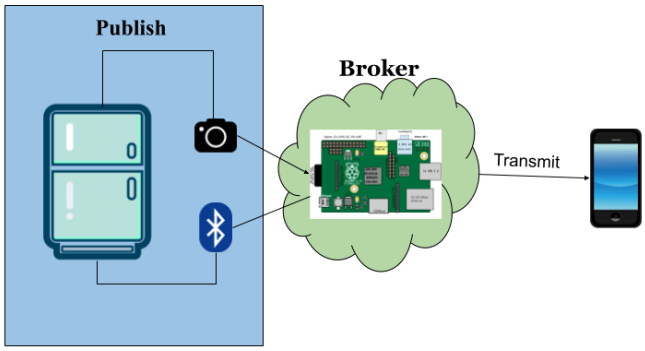
Features Required	Description
Low Temperature	Must be able to reach sub-40 degrees temperature to preserve food (USDA regulation)
Size	800mm (width) 600mm (height) 500 mm (depth)
Power/Voltage	110V - 240V (wired)
Cooling System	Placed in the back of the fridge to cool down the temperature
Design	2 sky-blue glass doors, possibly 3 rows/shelves

Central Computer



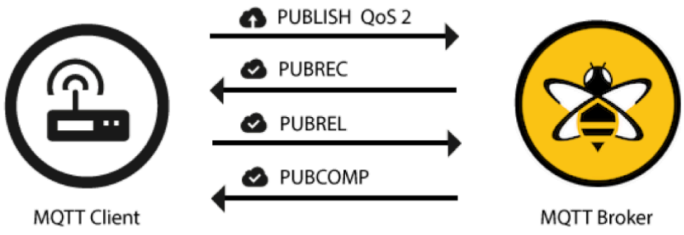
The Raspberry Pi 3 will be performing as the central Receiver/Transmitter. It will function as the broker/middleman/heart of the system, and the programming therein will affect all aspects of the project. The tasks it must perform are:

Receive and interpret the data from the Sensors (camera and temperature, update data specific for the user (such as whether or not a food item is “deleted”, change of temperature, or sending alerts), and also have access to the backend of the server aspect of the app.



With Regards to QoS (Quality of Service) for MQTT front/back ends

According to research and the link Yamin found, there will be 3 different types of Quality of Service which are 0, 1 and 2. QoS is the Quality of Service between the sender and receiver of messages. There are 3 different types of QoS but the best QoS will be QoS 2 which will perform exactly once. If you see the visual diagram below, it is the best quality and it can provide at least two request and response flows between sender and receiver.



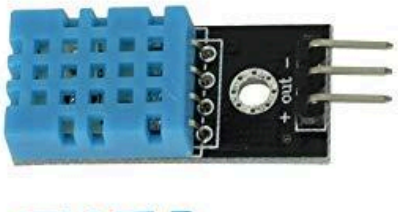
Yamin also documented the Raspberry Pi 3 for our project and she researched upon it. This is the link of her **documentation** of Raspberry Pi.

Peripheral Devices

Camera Sensor - Raspberry Pi Camera Module - V-2.8

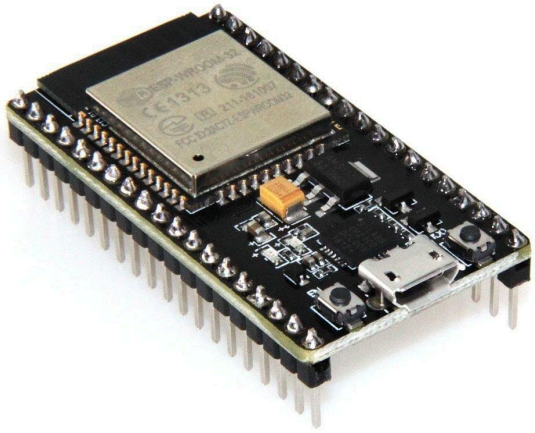
Features	Description
Expected Quantity	1 Camera will be needed
Camera Specs	Utilizes Sony IMX219 Sensor with a fixed focus lens. Programmable with Raspberry Pi's Linux IDE.
Image Quality	With the 8 Megapixel camera, image quality can be up to as much as: 3280 x 2464 resolution. Can capture video at 1080p30, 720p60, etc.
Size	2.54 x 2.28 x 0.762 cm
Power	Connected to the Raspberry via Ribbon Cable, w/ 2A Supply Recommended

Temperature Sensor Features - DHT11



Features	Description
Expected Quantity	2 Sensors will be utilized,
Output	Pre-Calibrated Digital Output
Temperature Range	0 - 50 Degrees Celsius (32 - 122 Degrees Fahrenheit)
Size	28 x 12.8 x 7.2 mm
Power Supply	3.3V - 5.5V

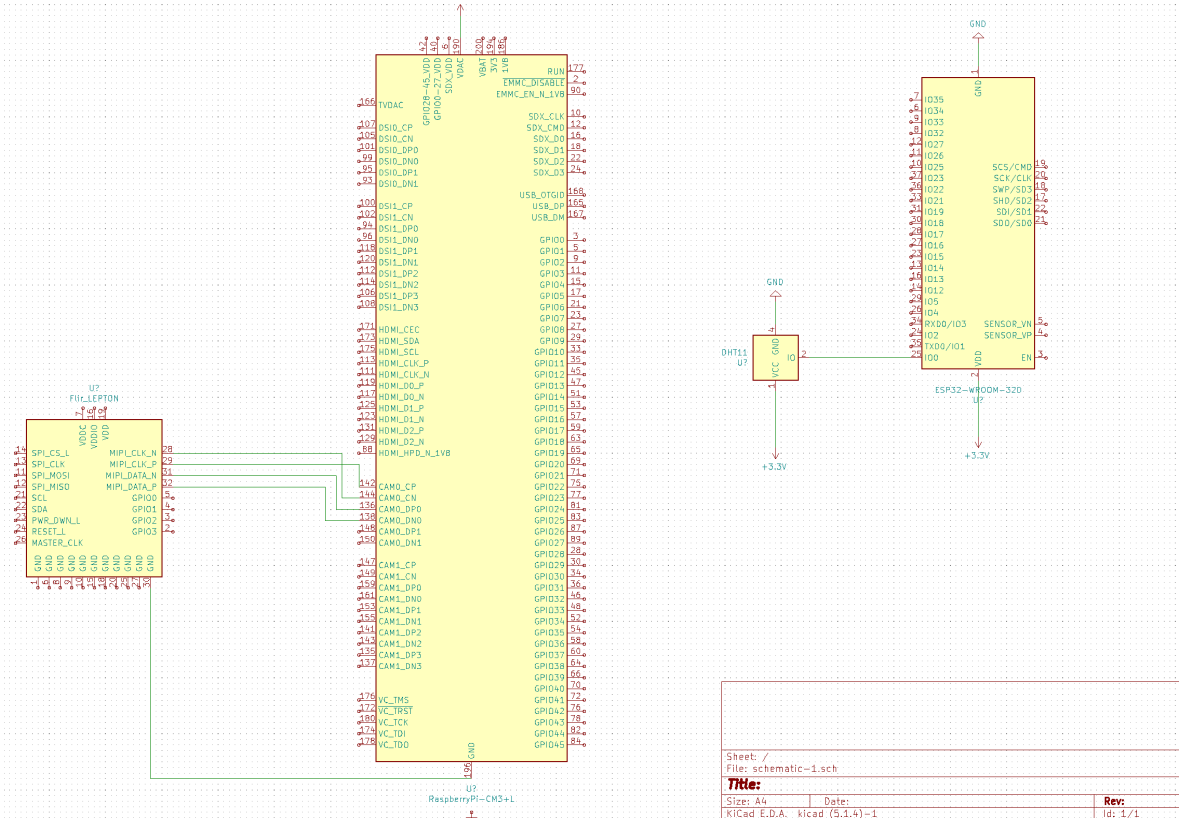
Microcontroller Features - ESP32 WROOM



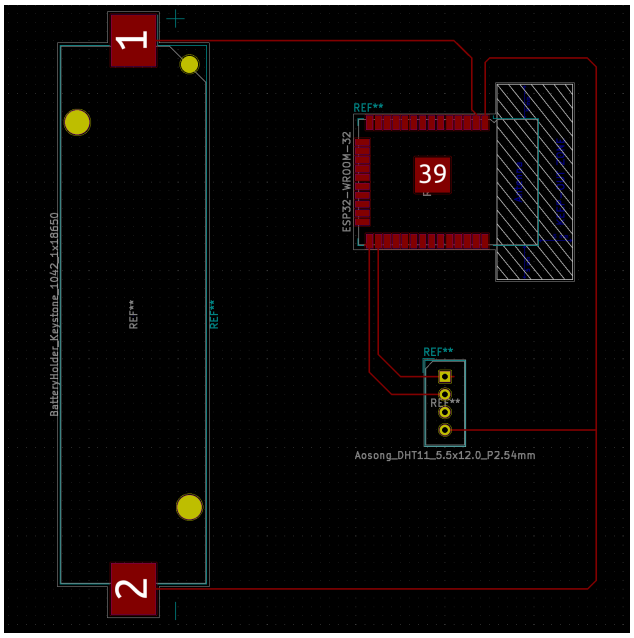
Features	Description
Expected Quantity	2 MCUs (MicroController Units) will be utilized,
Functions	Can operate as a standalone wireless system, or a slave device. Has Hybrid Bluetooth/2.4 GHz Dual Mode Wifi Functionality.

Temperature Range	-40 to 125 Degrees Celsius (-40 to 257 Degrees Farenheit)
Size	145 x 102 x 6 mm
Power Supply	2.3V - 3.6V

ESP32 - DHT11 and Raspberry 3 - Camera Module Schematic



ESP32 - DHT11 PCB File



Material Chart

Item No.	Name	Description	Qty	Unit Price
1	ESP-WROOM-32	Development Board with Hybrid 2.4 GHz WiFi and Dual Core Bluetooth capabilities with Arduino IDE	2	$10.99
2	DHT-11	Pre-Calibrated Temperature/Humidity Sensor (5-Pcs)	2	$10.49
3	ZMI Powerpack	10000mAh Portable Charger	2	$19.99

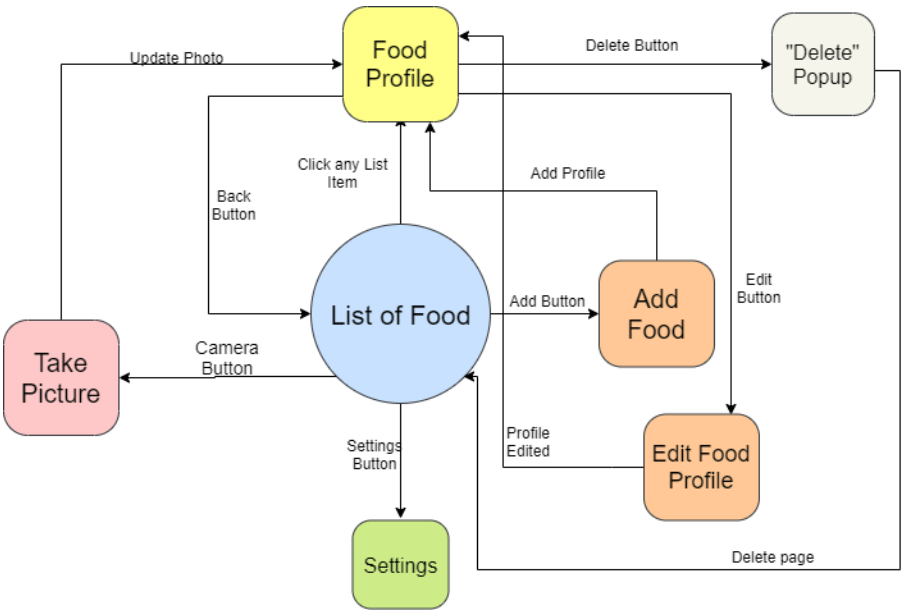
4	**Raspberry Pi Camera Module V2-8**	**1080p 60fps Camera Module with Sony IXMOR Sensor with Ribbon Cable, interfaces with Raspberry Pi 3**	**1**	**$24.98**
5	**Raspberry Pi 3**	**Model B, Motherboard with Wifi and Bluetooth capabilities**	**1**	**42.99**
6	**Refrigerator Radiator Cooling Equipment**	**12V 6A, DIY, Electronic**	**1**	**33.34**
Total	**x**	**x**	**x**	**142.78**

Software:

The app will be made using Flutter tool kit with Visual Studio Code as the editor. There are six main planned features: the menu, “add food” function, settings page, “snapshot” function, temperature reading, and food profile page. The menu is the most important part of the app because it contains all the food profiles that the user has currently entered into the system. Other important features are the food profile and “add food” function. They allow the user to browse specific details about each food and also add new entries into the system. The food profile should list information such as the date the food was added into the system, the expiration, and the picture of the food. The picture will be taken using the “Snapshot” feature of the app. The temperature is always listed on the top of the app regardless of what page the user

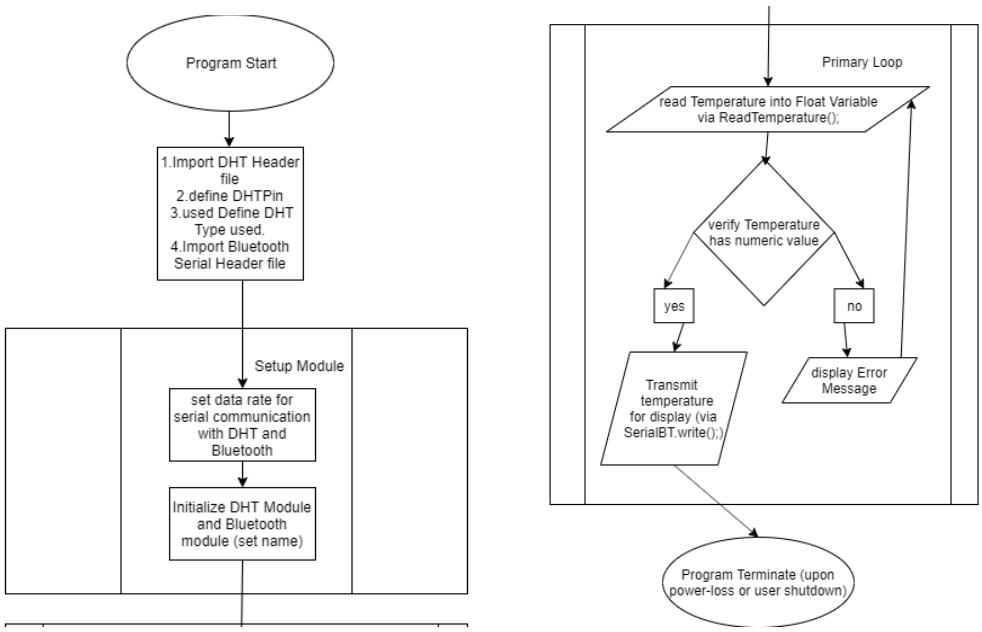
is currently on. The settings page is just for wiping data and for changing the display (light/dark).

Code/UI Flowchart for App



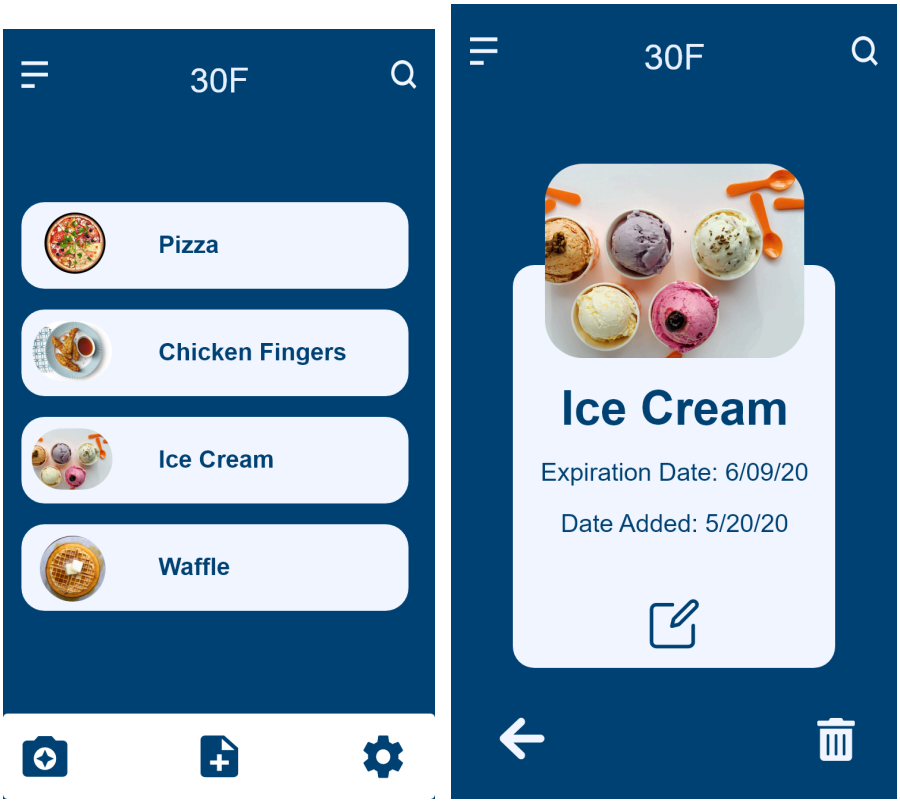
To elaborate on the software plans, Abby made this chart which details the flow of the user interface. The center of all operations will always be the menu. Each arrow dictates an action. For example, one of the arrows says “Add Button” which indicates that the user will click it in order to go to the “Add Food” page. The square shapes indicate actual pages. The coloring on the chart indicates if the pages are related to each other or not. For example, The “Add Food” and “Edit Food Profile” page are colored the same because Abby consider them to have a similar layout, despite being two separate pages. This is because they fulfill the same feature, which is to allow the user to list out information about their food. The only difference in their function is that one is about *adding* new entries, while the other is for *editing* new entries. Overall, this chart is about planning out the overall organization of the app.

Code Flowchart for ESP32 MCU



Due to the abundance of resources available on the Arduino IDE, and the abundance of header files for the ESP32's Bluetooth functionality and DHT functionality, the above flowchart is fairly small and references certain functions (such as "readTemperature();") available from the DHT.h file. One challenge heading forward is the desirable data rate for both the DHT sensor-input and the Bluetooth functionality of the ESP32, or whether it is possible to set unique data transfer rates for both and utilizing delays to allow for varying data rates for the DHT and Bluetooth.

Mock Screenshots of App in action/ UI



For the app, Abby mainly wanted to focus on the core functions, such as the menu and the food profile. She will try to keep it as minimal and focused as possible in order to be able to finish the app on time. For example, the main page of the app only consists of several features: the menu, the snapshot function, the "add" function, and settings. The food profile also only contains the most relevant information such as the temperature and the expiration date.

Stretch Goals

- We can possibly have a live feed for the camera in the future, so the user can see a constant stream of information.

Expo Poster

We had this printed on a mounted board at Staples Center. This is the design we are using for the Senior Design Expo.

Website

Yamin has also put up our project information on a website, which can be located here: https://emu-bellflower-3x6w.squarespace.com/blue-jay-food-spy-project

Appendix

1. ESP32 Datasheet
2. Raspberry Pi Camera Module v2-8 Documentation
3. DHT11 Documentation (Mouser)
4. Fridge Temperature and Food Storage Safety Article by FDA
5. Flutter Tutorial & Documentation
6. HiveMQ - MQTT Protocol Client, Broker
7. Pub.dev - flutter plugins & packages (camera)
8. How to Build the Fridge
9. Pi Camera Tutorial