Automated Inventory Tracking System

Gaspar Dantas, Sonu Thummar, Justin Rehg, and Lody Morillo

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The Automated Inventory Tracking System (AITS) has the intention to cover holes in the day to day process of managing our personal items. More directly intended to prevent waste from our grocery shopping by allowing users to access a view with all of their belongings, including product name, details, and expiration dates. This project detects the barcode of grocery items and add it to user's inventory and perform various CRUD operations. The software uses an API service that returns the details about the product scanned. The data received in the application is stored in the database where each user has their own inventory.**

***Index Terms* — Barcode, grocery, inventory, scanner tracking.**

I. INTRODUCTION

According to the EPA (Environmental Protection Agency) in 2018, there were a total of 292.4 million tons of waste in the USA alone. Our focus was to create a product that can benefit society and these numbers really struck us. We sought a way to diminish these values and produce a more efficient way of handling our inventories, leading us to create the Automated Inventory Tracking System. The main goal for this project is to create a user-friendly interface so any user can manage their items in a self-explanatory manner, with a simplistic UI and very clear expectations on how to deal with the given features. To maximize the number of users, we aimed to make this application platform independent, being capable of utilizing android or IOS devices. This system can be used by any person who intends to manage their inventory in an advance and efficient way.

This project has 2 major components: software and hardware. The hardware is used to scan in an item that users intends to add to their inventory. Upon scanning the item, the software application will receive the barcode string via Bluetooth and users will be able to see the details of the product on their screen. There are multiple operations that a user can perform in their list of products such as search, add or delete quantity and edit the details of the product. The application keeps track of the expiration dates of the items so users can plan ahead. This system intends to help users to shop only the items that are required and hence, preventing food waste

II. SOFTWARE APPLICATION

For the software of this project, we utilized Flutter, an open-source SDK (software development kit) that allows a single codebase to apply for both operating systems. making it a very convenient approach to handle most mobile devices.

A. Workflow

The workflow for attaining the software is by downloading the application in the Google Play Store, next, the user will be viewing the login screen. We performed social authentication mechanisms to Gmail, which simply brings the user to an authentication page, where they will provide their credentials and Flutter will ensure the user has a valid email and return it. From that point on, we store the email onto our users table in Firebase. We opted for Gmail upon researching the most utilized social authentication services and discovered over 40% of all users complete login via Gmail. To make the application as user friendly as possible, we have implemented a silent login, which essentially stores your email authentication for the next time you access the application. Successful login will redirect user to the main inventory where various CRUD operations can be performed. Users can search for items in the main inventory if the list is longer. Application will be synced with the database and can make real time changes once user start adding or scanning in the items.

B. User Interface

Each user will have their own products, which contains a set of inventory items. Upon a successful verification from the login screen, we bring the user to the main inventory view. This view is where the bulk of our application takes place. Users will see a list of all of their products (containing names, measure and descriptions), which upon click can see their underlying inventory items (containing expiration dates and amounts). Before deep-diving into the software development of AITS, we had considered only having a products table, but after further analysis we realized the need to have a lower level table of inventory items which contain a list of items with specific dates of expiration and amounts. We can clearly observe the need for this when thinking about having multiple Coca-Cola bottles, these can be categorized as the same product but we can have an inventory of those

products, containing multiple items of different expirations. We really underlined the concern for wastes and attempted to minimize it at all costs, and having a separation of concerns for each product, making them different inventories really took care of meeting those expectations.

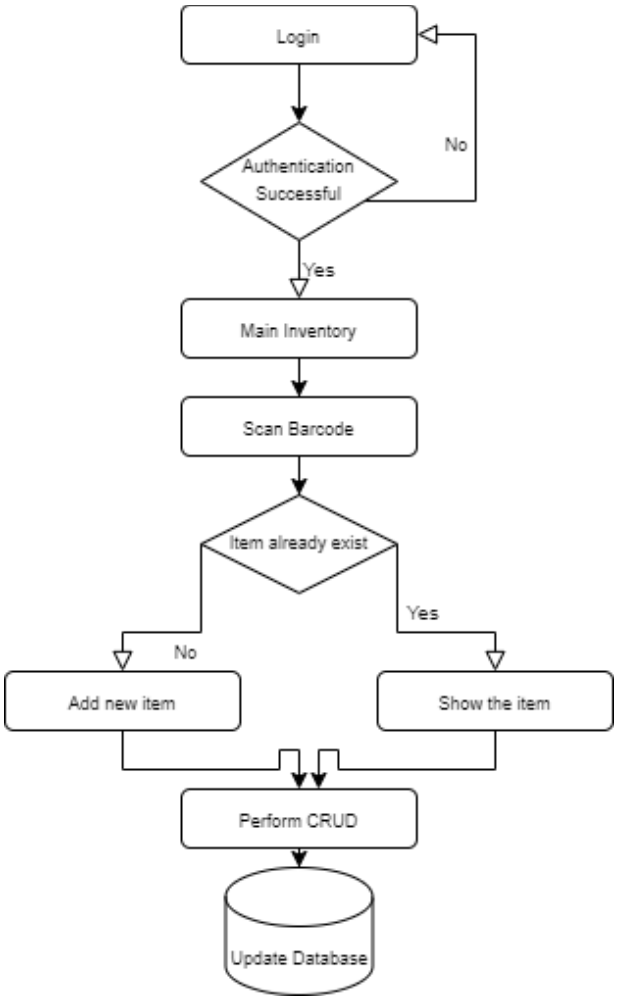

Fig. 1. Software Framework Layout

Upon product selections (based on click), users can view the different items underlying each product and visualize their expiration dates and amounts. Users may also adjust the amounts upon clicking the "+" or "-" buttons on the list and when the item has a count of 0, this removes the element from the product inventory, also if the product has a total of 0 items within the list, the product gets deleted from the UI and Firebase. To make adjustments to either the measures or expiration dates, upon a click of the item, users can access the product details and modify as necessary. At the top of the main inventory view, users can also search for specific products based on UPC code, inventory name, etc. and filter the content seen to the appropriate search criteria entered, this feature was mostly intended for a user with a diversified range of inventory items and products, to prevent searching manually, we decided to filter down upon the search criteria in case changes need to be made to a specific element from the list.

Lastly, our application has a button on the corner right, which allows the user to create a new product, adding an initial inventory item as well upon the insertion of product details, such as name, measure, expiration dates, UPC code, etc. For this section, some fields are required, such as name and measure, UPC codes are not required but upon insertion, users can utilize a lookup button, which will ping the Nutritionix API, retrieving the product details automatically, except expiration dates. Upon the save button, both the inventory and initial product get saved into Firebase and the application brings the user back to the main view, with the added product/item, where they can customize the amounts, depending on how many they have in their household.

Our intention is to sync the HW with the SW via Bluetooth modules. Upon the user opening the creation view, Bluetooth listeners would activate and await a scan. Upon a scan entry, the string would be received and inserted into the UPC code text field and automatically call the Nutritionix API to extract further details of the product. This capability will also be allowed on the application itself. There is a scanner button on the top of the screen, which activates the camera from the phone and awaits a barcode for the scan detection and validation. Upon the retrieval of the scan, we plan on approaching the same feature as the HW/SW integration, by inserting the string into the UPC code text field and automatically extracting details from Nutritionix.

By implementing this feature, we focused on the overall user-friendliness of the application and not requiring hardware components to utilize the software. We wanted to implement a self-sufficient app that can really help people manage their inventory, help users get organized and be more optimal with their products. Not only will there be benefits by reducing waste, there will also be relief factors on both finance and time, as users will mitigate the risks of throwing away a perfectly good item that they simply forgot in the fridge or cabinet due to other day-to-day tasks. By providing users a view of everything they have to manage, releases the efforts of trying to memorize all items, when to consume them by, what is missing for grocery lists, etc. Not only can this be used by a single individual, if a family opts to generate a family account or email, they can manage all items at once within their own devices. The application handles

multiple users and has real-time capabilities, with a very quick and responsive interface. Though, the most gratifying accomplishment overall is how diverse this application can be, handling users from various mobile devices, either android or IOS, which brings all these benefits to a wide range of potential users.

We do not intend to market this product, AITS was meant to be an application designed solely to help other people and impact society in a positive manner, with no vision to gain a financial return. We want to release the application and keep developing potential features for AITS, such as sending user alerts when it is close to expiration and other useful capabilities. At the time, we plan on releasing the application only on the Google Play Store for now, as for IOS applications, we would need to pay additional fees to release it in the App Store.

C. Framework

The framework allows us to add, replace, and remove features without having to make significant changes to our code. We integrated a model-based framework where we have a data model, business model, and view model. A model represents a set of data such as color, value, name etc. which represents an entity for the system. The data model represents the entity from a provider such as a table from a database. The business model represents an entity of system that is populated through different data models. It is an independent and isolated identity in a way that the data models can also be used to construct same and exact business model. The view model is the model that represents the data that a specific view is going to use. It is constructed though business logic by utilizing different business models. The bloc events are the events that will cause business logic functions to trigger and apply all the changes needed. The Dao or data access objects will pull in the data from the database that is readable by the data models. The presenter will act as a middleman between the bloc and the view. It will present the events to either of the entity. When the presenter will receive and event from the view, it will pipe in the data to the bloc and when the presenter receives an event from the bloc, it will pipe out the data to the bloc.

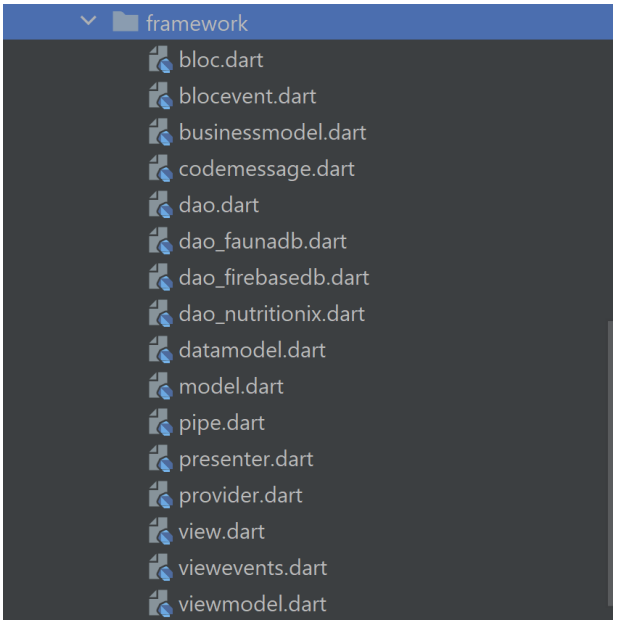

Fig. 2. Software Framework Layout

D. API

This project uses Nutritionix API. This database has more than 22000 UPC codes registered. Creating an account provides us with an API key and Secret Key. Once configured in the application, it returns the JSON format with all the information that is available from the passed UPC code.

E. Testing

The testing was done by keeping the requirements in the mind. We were able to pass most of the tests without any errors. Some of the main things to test was to determine if the CRUD operations are working correctly without any errors. The API was able to pull in data from the source to the application. The application was able to successfully login a user and pull up the inventory specific to that user. The application was able to search for the product based on the names using a search bar. The application was able to scan in the items and retrieve the barcode. There were many other small tests done to make sure that the workflow of the application is working as desired. We were not able to pass the notifications test and the reason for that is because we were not able to figure out the intervals that needs to be set for the triggers to execute notifications. Here is a table which explains the testing process in a detailed manner.

Status	Requirements
Pass	The software should be able to store items from the inventory in a database
Pass	The software should have an API service to retrieve product details.
Pass	The software should be able to insert records directly from the scan and allow users to add inventory items.
Pass	The software should be able to delete the data
Pass	The software should allow the user to add/remove expiration dates through datetime pickers
Pass	The application should be reusable across different platforms
Fail	Push Alert Notification

Fig. 3. Test Status Table

III. HARDWARE APPLICATION

A. Fridge Inventory Client Tracker

Based The fridge inventory tracker needed a way to input the items being inserted in the user's fridge. In this age in time, almost all items purchased in a grocery store contain a UPC code printed on the grocery store item. Implementing a barcode scanner would be ideal to take advantage of what most items have in common, to track the grocery store items. Several options were considered when thinking of how the user interface will be implemented. In the end, the choice selected was implementing an Android application. Since the system would be operation from Android, it was also decided that a microcontroller would be needed to handle the information retrieved from the scanner to then be sent to the Bluetooth module. It was required that the system had up to 16MIPS throughput at 16MHz and combines 32 KB ISP Flash memory with read-while-write capabilities. Furthermore, the fridge inventory tracker hardware needed to be compact, so that the product would not take much space in a kitchen or on the go. The board was designed to be no more than four inches squared.

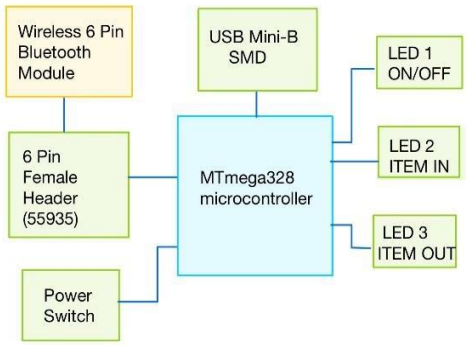

Fig. 4. Dimension of Q45U Ultrasonic Range Finder

The microcontroller also has pins which connect to the three LEDs, USB mini-B host, push button power switch, and Bluetooth module pins connector. The PCB board uses a 5V power connector to power the microcontroller and other components. This configuration can be seen in figure 1.

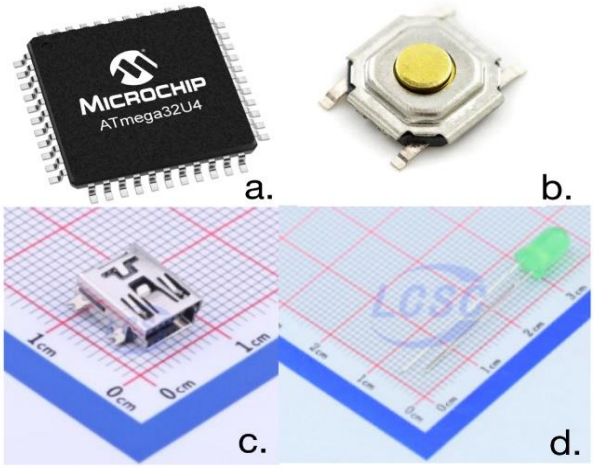

Fig. 5. (a.) ATmega32 Microcontroller, (b.) Mini Pushbutton Switch – SMD, (c.) USB Connectors SMD, and (d.) Through Hole 5mm Light Emitting Diodes

To redeem the information for each grocery store item, the fridge inventory system needed to be able to interface a scanner with the microcontroller.
Several scanners were investigated, most of which were connected via USB port connected to a long cord. The best option, the Tera Barcode Scanner, was selected by virtue of various reasons. It is an enduring, featherweight, wireless automatic single line 1D bar code scanner that advances productivity as it raises scanning flexibility. It can be used as a stationary desktop scanner with its included scanner holding stand.

Fig. 6. Tera Barcode Scanner. (Reprinted with permission from Tera Instruments, Inc)

As in Fig. 6. Tera Barcode Scanner shows (Reprinted with permission from Tera Instruments, Inc) a covenant, fixed mount presentation barcode scanner. Its most valuable aspect is that as soon as the barcode scanner is it is plugged into a PC, and the barcode is given to the scanner, the numbers are immediately taken into the computer. The process is almost like entering inputs into a keyboard, no unique drivers are required to be able to use the barcode scanner device. It is compatible with PC and Laptop where you can plug the 2.4GHz wireless dongle. When connecting via 2.4GHz wireless, the transmission distance can be up to 328ft under a barrier-free environment and about 30-100ft under an indoor obstacle environment, depending on the complexity of the environment. Tera introduced the unique Battery Level display function to Tera 5100. No other product on the market has such a feature. There are four levels of the indicator light, each time a 500mAh battery is used up, and one level indicator light will go out. Most of the 1D laser wireless barcode scanners on the market are equipped with battery capacities of 800 to 1200mAh. However, Tera 5100 is one of the few products that are equipped with an extra-large 2000mAh battery. As a result, Tera 5100 doubles the service time and standby time with the huge battery capacity, allowing over 40 hours continuous scanning time and over 60 days standby time.

A connection is required to connect to the UPC database and to be able to keep all the databases synchronized. With this system in mind, wireless network connection allows for better system integration. The next main concern was selecting a medium for the data scanned from the scanner to update in the Android App. A Wi-Fi module and a Bluetooth module were each considered. The objective was to select a module which was compatible with Android without the need of custom drivers.

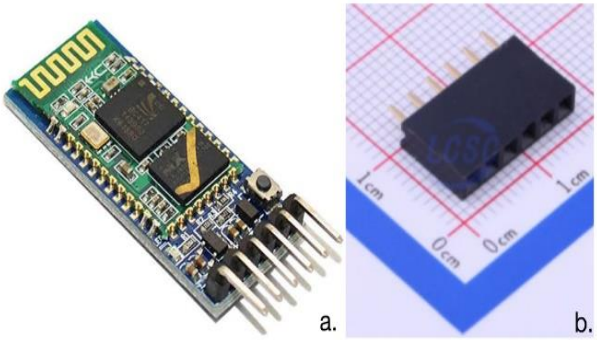

Fig. 7. (a.) HiLetgo HC-05 Wireless Bluetooth RF Transceiver Master Slave Integrated Bluetooth Module 6 Pin, (b.) Female Header 6

The fridge inventory program will be able to be seen on the client's smart phone via an App. This adds convenience to the user, as they do not have to take up anymore real estate in their personal home space, since almost everyone already owns a smart phone. Using the users already owned smart phone was selected as the best option. This selection adds an element of speed and ease to the user experience when using the system. UPC code.

B. Printed Circuit Board Layout

The printed circuit board can be described by the different components that make it up. There are three important interfaces that are essential to have functioning hardware. The Bluetooth component, the microcontroller, the USB port are the important components needed for the implementation of the system. The Bluetooth will send the barcode information the cellular phone. The microcontroller will take in and send out signal so it can interpret the data we need to get it where it needs to go. The USB port is the only connection the system has to the scanner to retrieve the barcode information from the products surface. On the board the placement of the Bluetooth can be found under the J1 and J2 label. The microcontroller can be found under the U1 label. The USB port can be found under the J5 label. This can be seen in Figure 8.

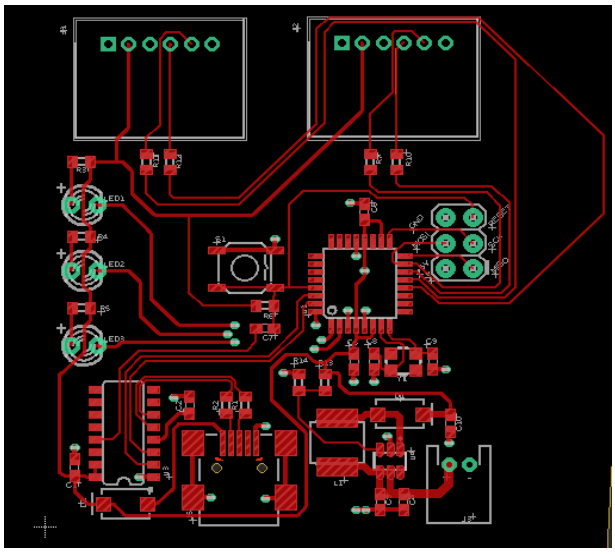

Fig. 8. Board file of the printed circuit board

While these three components are the bulk of our hardware, there are other component parts that are equally important. For the hardware system to turn on and off there must be a push button switch, which can be located under the label S1. The system also needs to power to function. The board will bring in power through the component labeled J3, via two females to male wires. A final detail that was added to the hardware was LED lights so that the user can receive indication that the board is turned on, an item has been scanned in and that an item has been scanned out.

C. Barcodes

Because our project is for tracking inventory, it is necessary to understand barcodes and how they work. For our project we decided to keep it simple and work with one-dimensional barcodes.

Fig. 9. UPC-A Barcode

Most commonly for products, it is referred to as a UPC or universal product code and is generally a set string length but can vary in size based on the product. To access the data the barcodes refer to requires access to one of a few specific databases.

Hardware

The purpose of the hardware is to take in our barcode data, interpret it properly, and send it wirelessly to our application. The Diagram below shows a general flow of how the data will move through our hardware.

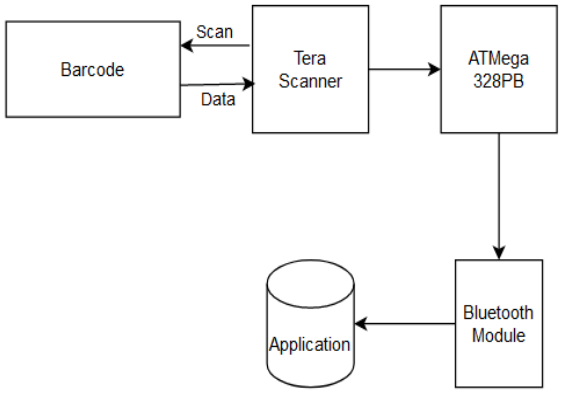

Fig. 10. Hardware Diagram

The scanner will take the data from the barcode, the data will be converted to a UART signal, sent to our microcontroller, and then sent through our Bluetooth module.

D. UART Set up

UART stands for Universal Asynchronous Receiver Transmitter and is the method our hardware uses to translate and understand the data we are receiving and sending. Because our HC-05 module operates at a 9600 baud rate automatically we want our MCU to operate at this same baud rate. To set this up, we need to use an equation to determine what to set our registers to.

$$UBRRn = \frac{f_{OSC}}{16BAUD} - 1$$

Fig. 11. Baud Rate Formula

Here UBRRn is the number we're looking to set our registers to. F_{OSC} is our crystal which is 16MHz, and we need Baud to be 9600. This yields a value of 103. When converted to hexadecimal we have a value of 0x67. Since the UBRRn is broken into high and low this means our high will be set to 0x00, and our low will be set to 0x67 to achieve a 9600 baud rate to match our Bluetooth module. The baud rate can also be looked at as bits per second, each byte send through UART also requires a

start and a stop bit, so each byte sent takes 10 bits, and therefore the speed should be approximately 960 bytes per second. Each string from the scanner should be 12 characters and should take 120 bits.

E. ATMEGA328PB Microcontroller

The microcontroller we're using for our project is the ATMega328PB. It is an 8-bit microcontroller that has 32 registers, a 1KB EEPROM, and a 2KB RAM, and 32KB Memory.

Fig. 12. ATMega328PB

For us it will be operating at 16MHz frequency. The key function of this MCU is to read in the UART signal. Some MCUs only have 1 UART, but this MCU allows for multiple. To connect properly to the Bluetooth component, we will need to make sure to set the baud rate properly to match the Bluetooth. This means we need to set the baud rate to 9600 and properly set this in the code. We will also be utilizing the SPI communication to flash code onto our ATMega. This requires a device of some kind, along with some software to properly flash the code. For this, we will use the Atmel 51 AVR MCU programmer.

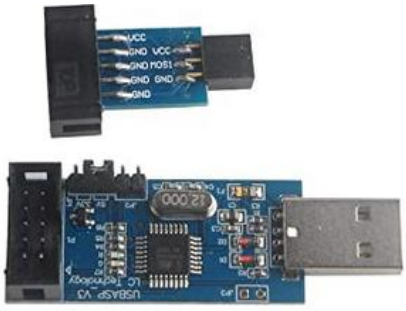

Fig. 13. AVR MCU Programmer

It is a USB based tool that connects to the SPI pins and writes the data to the MCU. Tera Barcode Scanner the Tera Barcode Scanner is the device we used to read in the barcode data. It is specifically designed to read one-dimensional barcodes. This happens using a laser, a sensor, and the internal processor of the scanner. The laser will reflect from the white spaces between the barcode and will allow the sensor to determine the spaces and the resulting data.

It then takes this signal and sends it through USB. For our project we will be using a USB to mini-USB adapter, and a USB to UART component to translate this data to the appropriate form. This scanner also has a wired or wireless option, which we felt made it more flexible and easier to use.

F. HC-05 Bluetooth Module

To make sure our hardware connects to our application we are using a HC-05 Bluetooth Module. This will enable us to send the barcode string data as a UART string through a Bluetooth connection.

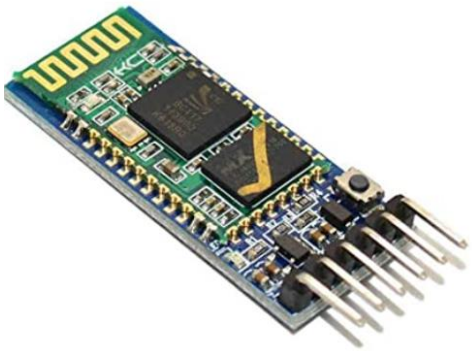

Fig. 14. HC-05 Bluetooth Module

The HC-05 operates on a 9600 baud rate for connection, and a 38900 baud rate to change commands. Fortunately, we don't need to alter the Bluetooth module any, and can use the default settings. The TX and RX pins refer to UART. TX refers to sending data whereas RX refers to receiving data. When connecting to the pins we need to make sure we connect the RX on Bluetooth to a TX pin on the MCU.

CONCLUSION

In conclusion, our software is designed to be very user friendly where we have integrated social authentication, database CRUD operations, API, and a way to manage the inventory better. We have used Flutter as our SDK which can be used to develop platform independent applications. Our hardware is a lightweight design so that users can move freely without any problem. We have talked about the scanner, the microcontroller, the PCB design, and the Bluetooth module. We created this project because we want to help the planet and community by reducing the wastage of food and groceries items that people use daily.

ACKNOWLEDGEMENT

We would like to thank all the consultants and class mentors for guiding us with their valuable information. The professors for the class were very helpful in giving feedback and helping us in times of need. We would also like to thank the writers and providers of all the resources we used to complete this project. Finally, we would like to thank all the peers for providing us with the feedback and also guiding us to understand some of the administrative parts that we came across during the time of senior design course.

REFERENCES

[1] ASP Microcomputers. (2020, March 30). *Barcode Scanners: What Are The Main Interfaces Of Barcode Scanners?* ASP Microcomputers. https://www.asp.com.au/what-main-interfaces-barcode-scanners/.

[2] *EAN/UPC - Barcodes.* GS1. (n.d.). https://www.gs1.org/standards/barcodes/ean-upc.

[3] Fathabadi, A. S., Butler, M. J., Yang, S., Maeda-Nunez, L. A., Bantock, J., Al-Hashimi, B. M., & Merrett, G. V. (2017, December 20). *A model-based framework for software portability and verification in embedded power management systems*. Journal of Systems Architecture. https://www.sciencedirect.com/science/article/abs/pii/S1383762117305234.

[4] *Flutter documentation*. Flutter. (n.d.). https://flutter.dev/docs.

[5] *HC-05 - Bluetooth Module*. Components101. (2021, July 16). https://components101.com/wireless/hc-05-bluetooth-module.

[6] M-Short. (n.d.). Installing an Arduino Bootloader. https://learn.sparkfun.com/tutorials/installing-an-arduino-bootloader/all.

[7] Nutritionix. (n.d.). https://www.nutritionix.com/business/api?gclid=Cj0KCQjwl_SHBhCQARIsAFIFRVWQTeCqz5PTzOuW9i-xyIHw_1-IDbcppJnM2M9RCI4HcJdRHv92u-gaAh_sEALw_wcB.

[8] W. H. Cantrell, and W. A. Davis, "Amplitude modulator utilizing a high-Q class-E DC-DC converter," *2003 IEEE MTT-S Int. Microwave Symp. Dig.*, vol. 3, pp. 1721-1724, June 2003.

[9] *Tera 2D Wireless Barcode Scanner USB Wired 2D QR Bar Code Reader with Stand model: HW0005-Z.* Teradigital. (n.d.). https://tera-digital.com/collections/all-products/products/tera-2d-wireless-barcode-scanner-2-in-1-2-4ghz-wireless-usb-2-0-wired-2d-qr-bar-code-scanner.

BIOGRAPHY

Gaspar Dantas, a senior Computer Engineering student at University of Central Florida. He is passionate about learning new things everyday. He aspires to be a software developer after graduating from UCF

Sonu Thummar, a senior Computer Engineering student at University of Central Florida. He currently works as a Technical Support Analyst and Software Developer Intern at AceApplications LLC. He wants to become a full stack programmer after graduating from UCF.

Lody Morillo, is currently a senior at the University of Central Florida and will receive her Bachelors of Science in Electrical Engineering in August of 2021. She plans on going into the industry once she graduates.

Justin Rehg, a senior at the University of Central Florida. He plans to graduate with his Bachelor of Science in Computer Engineering in August of 2021. He is currently in pursuit of an internship, and plans to attend University of Central Florida's Master Program FIEA(Florida Interactive Entertainment Academy).