July 7, 2019

Dr. Craig Scratchley
School of Engineering Science
Simon Fraser University
Burnaby, BC, V5A 1S6

Re: ENSC405W/440 Design Specification for FooMinder

Dear Dr. Scratchley,

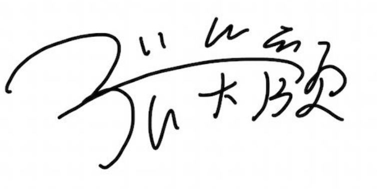
This design specification document for FooMinder was prepared by Freshist formed with Gloria, Harry, Victor and Laura for ENSC405W/440. Our intention for this capstone project to apply image recognition and sensor data analysis to provide a reminder and organizer for groceries. FooMinder is designed to be a tool to use in families for daily routine.

FooMinder consists of two components, a mobile application and a fridge status detector. The application will perform image processing which will convert image to text and extract keywords. The users will scan their receipt and the app will recognize the grocery and create the list. We will pull the expiration date data from the web and set up the reminder.
For the hardware part, we will build a fridge status reminder which will place inside the fridge and remind the user when there is something rotten. The data from the sensors will be sent to our app through Bluetooth. All these data will be stored and visualized in the app.

The design specification will review from the general system design to more detailed specification such as Mobile Application Design, Fridge Environment Detector Design and Hardware design. The Appendix includes a User Interface and Appearance Design which will discuss from Graphical Presentation to Analytical and Empirical Usability Testing. Lastly, this document provided a Supporting Test Procedure including software and hardware plans.

Our team would like to thank you in advance for taking the time to read our design specification. If you have any questions, feel free to contact us at huyixinh@sfu.ca.

Sincerely,



Xiaoyan Zhang
CEO
Freshist

Design Specification:
Group 6
FooMinder

Partners:
Ruihong Gong
Yixin Hu
Wei Liang
Xiaoyan Zhang

Contact:
Yixin Hu
huyixinh@sfu.ca

Submitted to:
Craig Scratchley, ENSC 405W
Andrew Rawicz, ENSC 440
School of Engineering Science
Simon Fraser University

Issued Date:
July 7th, 2019

Abstract

This document specifies the functional design specification of the food reminder system, FooMinder. Firstly, it will provide an overview that describes the general functionality and deliverables in terms of proof of concept, the prototype and the final product which provide a timeline for our project. Secondly, it presents the design specification and process for each system component. This document also gives the User Interface and Appearance Design as well as the Supporting Test Procedure in the Appendix. The purpose of this document is to give readers a thorough understanding of the device and its system with detailed design information.

The FooMinder is a system that can be separated into two main components: a mobile application and a fridge status detector. The users can scan the grocery receipt and the application will perform image processing to extract keywords and generate the grocery list which will be stored in the database. The fridge detector will sense chemical gas inside the fridge and report the status to remind the user when there is something rotten. The application will receive the data from the sensors through Bluetooth. All these data will be stored and visualized in the app.

These specifications for FooMinder cover the following core components:

- System Overview: including Proof of Concept Deliverables, Prototype Deliverables and Final Product Deliverables.
- Mobile Application Design: including General Design, Software Flow, Image Processing Module, Bluetooth Interface and Database and Display.
- Fridge Environment Detector Design: including General and Sensor Specification
- Hardware Design: including General Design, Arduino Microcontroller, Power Consumption, Pin Setup, Communication Method, General Workflow, Design Alternative and Future Consideration.
- User Interface and Appearance Design Appendix: including Graphical Presentation, User Analysis, Technical Analysis, Engineering Standards, Analytical and Empirical Usability Testing.
- Supporting Test Procedure: including software and hardware parts.

Table of Contents

List of Tables

List of Figures

Glossary

API - Application Programming Interface
DAO - Data access object
Database Entities - Objects in a system that we want to model and store information about
GND - System ground
LCD - Liquid Crystal Display
MAC Address - hardware identification number that uniquely identifies each device on a network
MCU - Microcontroller and etc
OCR model - Optical Character Recognition model
PPM - Parts per million dimensionless concentration
PWM - Pulse width modulation
RX - Receive
TX - Transmit
UART - Universal asynchronous receiver-transmitter
VCC - Power supply voltage, typically the operating voltage is +5V

1 Introduction

The name “FooMinder” is a combination of “Food” and “Reminder”. Just like the literal meaning of it. It is a reminder of food. FooMinder is designed to a system that contains a mobile application and fridge detector. The fridge detector is intended to be put inside the fridge and detect the environment of the fridge. The processed sensor data will then send through Bluetooth the mobile application. Users can record their receipt from the grocery stores and the application will set up a reminder to alert users for items are near the expiration date. The fridge environment status will be visualized based on the sensor data and remind users to clean the fridge when necessary.

The purpose of this project is to make a tool that can help users keep track of their groceries as well as organizing them. We believe it would be an effective way to get the best nutrition from every single food and thus help reduce food waste and save money for the users.

1.1 Scope

This design specification document outlines the technical design requirements for our project FooMinder mobile application and fridge detector. This specification is presented with detailed information and requirements that distinguish between different development stages, such as proof-of-concept, prototype and final product. This document will have main discussions on mobile application design and the detector design. Each major section will contain subsections that describe the specific design processes on how the functionality is going to be implemented.

This document also includes the User Interface and Appearance Design in the Appendix section. It includes user analysis and technical analysis as well as Analytical and Empirical Usability Testing. In the Appendix, we also include the Supporting Test Procedure to verify and validate each component of the design.

1.2 Intended Audience

The intended audience of this document is for Freshist members, instructors, Craig Scratchley, Dr. Andrew Rawicz, and teaching assistants supporting us in this course. This document will serve as a reference and guideline for the following development process. Future revisions will draw from the framework detailed in this document.

1.3 Design Specification Classification

The specification in this document will consider the following convention:

Des {Section}.{Subsection}.{Requirement Number} {Stage of Development}

Example: Des 3.1.1.P

Stage of Development	Encode
Proof of Concept	C
Prototype	P
Final Product	F

Table 1.2: Stage of Development Encoding

2 System Overview

FooMinder is a mobile application with an external detector. The detector is intended to be placed inside the fridge to detect the environment change based on the chemical concentrations. The application is going to receive concentration level data from the external hardware through Bluetooth and save it into the database. Users will be able to scan the grocery receipt by using the application. The app will perform image processing on the picture of the receipts and extract out keywords for the first step of setting up the reminder. Then we plan to pull the common expiration date data from the website and finish setting up this reminder. FooMinder is overall a reminder designed to remind the user of the food expiration date and the status of the fridge. The diagram of the system working flow is displayed below.

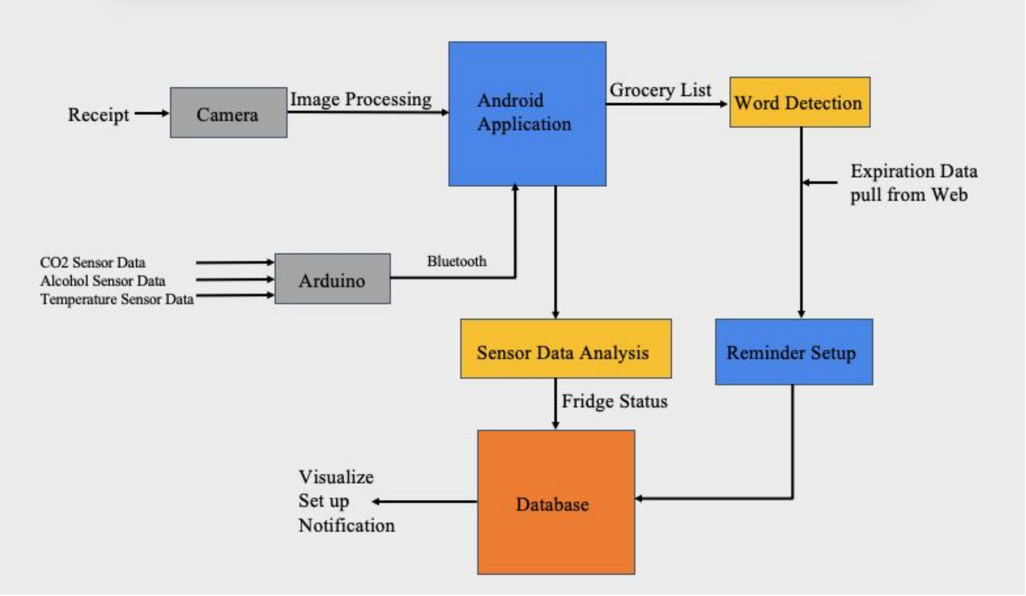


Figure 2.1: General Working Flow

The general design specification is shown in the table below.

Code	Description
Des 2.1.1 C	The detector should be able to detect CO_2 and alcohol gas concentration in the enclosed environment (i.e. fridge).
Des 2.1.2 P	The application should be able to determine the range of different levels of gas concentration.
Des 2.1.3 P	The scanning process should not have a delay of more than 1 second.
Des 2.1.4 F	All system should be thoroughly tested and verified before commercially produced under the guideline of Acceptance Test Procedure in the requirement specification and the Supporting Test Procedure included in the Appendix.

Table 2.1: General Design Specification

2.1 Proof of Concept Deliverables

Based on the various progress achieved, the hardware team is able to present simple data readings. The hardware consists of a CO_2 sensor, an alcohol sensor and a temperature and moisture sensor, an LCD screen. It gets power by connecting to the computer using cables. The LCD screen is able to show raw data and send back to the computer using UART. The sensors are placed in parallel on one side of the circuit board. Alternatively, we may place the sensors on different sides and in different orientations to avoid disturbance and reduce the overall size of the detector. The microcontroller we are using is Arduino Uno. It is used to connect the external circuit together and send data to the computer.
The software team is able to present a simple image-to-text conversion together with a simple tentative user interface. The application is developed under Android Studio using JAVA language. The application can be installed and run on an actual android phone and emulator with API 28. For the Proof-of-Concept Deliverables, the image processing and the user interface will be further enhanced. We will set up the database for storing the grocery data. The deliverable of this stage would be available before August 15, 2019.

2.2 Prototype Deliverables

For the Prototype Deliverables, the LCD screen will be replaced by a single tri-colour LED to reduce the power. This LED will work the same purpose as the LCD which is to indicate the feedback from the application side. For example, when the power is turned on, the LED will turn green. When the power is about to go out, it will turn red to remind the user to charge or replace the battery. when the Bluetooth connection is established, the LED will blink with blue colour. Since this is a single portable device, we will replace power from the cable by

the battery. In this phase, UART will be replaced with Bluetooth which will eventually be connected with the application and transmit data.
The application should be able to connect with Bluetooth devices and receive data package from them. The image-to-text converter should be trained to learn how to distinguish different layout of receipts from different stores and convert keywords from abbreviations. The deliverable of this stage would be available around December 2019.

2.3 Final Product Deliverables

The final product should be fully tested and verified before put it into massive production. User research and business marketing also need to should be done. The Arduino microcontroller and external circuit should be replaced into a single microcontroller chip to reduce the size and the weight of the product, unnecessary components on the board but never used and power required for the system which would significantly extend the battery life. Moreover, the final version of the application should be adapted in most mobile phones and available in major application stores.

3 Mobile Application Design

3.1 System Overview

Our goal is to provide the user with an easy-to-use application so that they can keep track of what they have in the fridge and remind them about the items that are soon to be expired. Instead of capturing images of fruits and vegetables one by one, it is more convenient to scan a receipt and select from the list of groceries to be added for the reminder. This will require the usage of an image-to-text converter and a database to store all of the items which the user has selected. The sensor data, on the other hand, are sent to the application through Bluetooth in which they will be processed into the chemical concentration level. The processed sensor data will be comprehensible to the users in the sense that he can easily decide whether to consume or discard the item.

3.2 Application Design Specifications

The design specifications below details our requirements for building the android application.

Code	**Description**
Des 3.2.1 C	The application will be developed with android Pie API level 28.
Des 3.2.2 P	Text recognition will utilize the Firebase ML kit version 21.0.0.
Des 3.2.3 P	The application will utilize the Room database API for data storage.

Des 3.2.4 P	The application Bluetooth interface shall send/receive data with no loss.
Des 3.2.5 P	The text image to text converter shall recognize receipt with only a few misspelled words.
Des 3.2.6 P	The spell checker API used to correct abbreviations shall accurately recognize a word.
Des 3.2.7 P	The application shall store/update/load all data without any errors and data losses.
Des 3.2.8 P	The user can navigator the application UI with no delay or lag.
Des 3.2.9 P	The application should be easy to use and guide the user to the next steps.

Table 3.2 Application Design Specification

3.3 General Design

Our android application will utilize the FireBase machine learning API to convert captured photo into text[2]. When the user scans the receipt, the captured image is converted into a bitmap format. A FireBaseVisionImage object can be instantiated to load the bitmap image to be converted into text. After that, an image-to-text recognizer will be used to extract the fruit and vegetable names into a list. The user can choose items from this list to set up a reminder.

The raw sensor data from the Arduino Uno microcontroller will be sent across to our application and processed into concentration levels to indicate to the user which of the current items are going bad. For our prototype, the user will be able to make a one time request to get the sensor data through the Bluetooth interface.

To set up a reminder for the selected items, we will be pulling estimated expiry dates from the website EatByDate[8] using a web scraping tool called Jsoup. The item, its corresponding expiry date, date of purchase, and the quantity will be stored in a database which can be visualized in the android application on the request of the user. This is achieved using the Android Room database library. The figure below shows the architectural design of our android application.

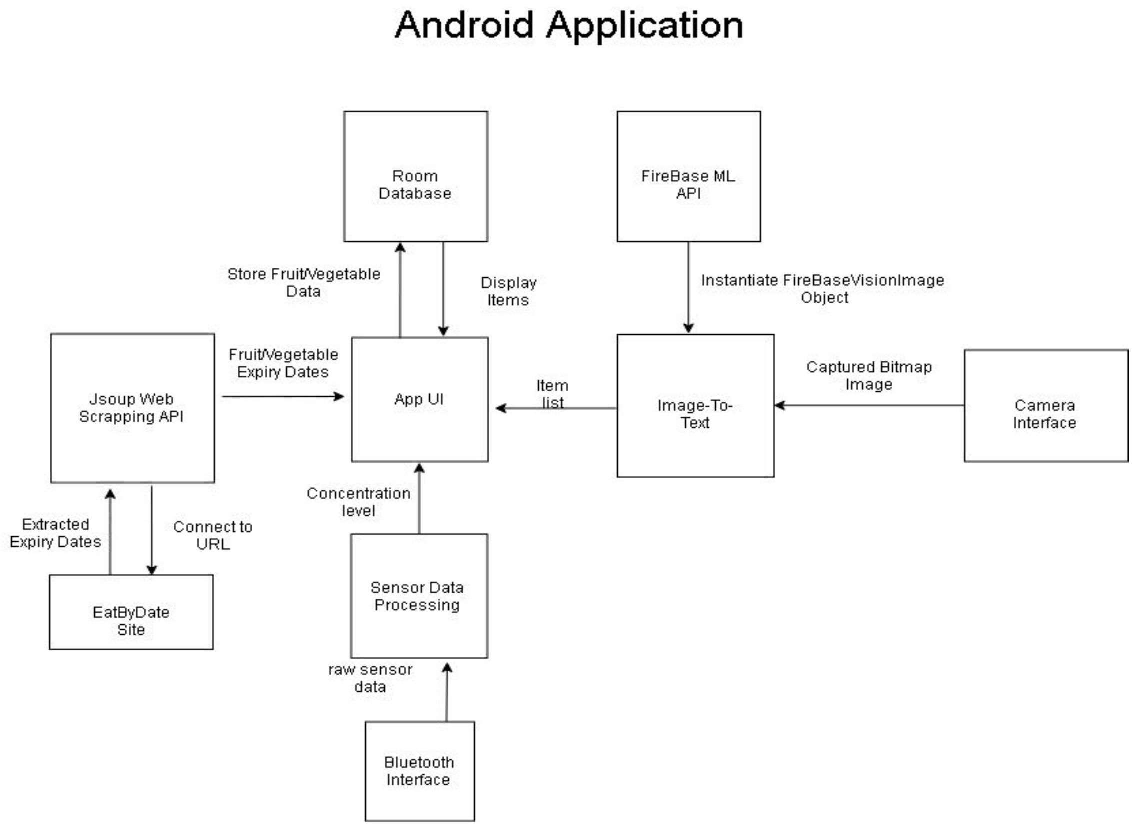


Figure 3.2: Architecture Design of the App

3.4 Software Flow

Due to the variations in receipts, for instance, the font size, the clarity of the ink, and the length of the receipts, it may be difficult to scan a whole piece of paper and expect every key component being extracted even when using a well trained OCR model. Therefore, we included in our application an option for the user to manually enter the items that were not recognized or partially recognized so they can still include them for reminders. Once a list has been composed by scanning or manually entering, the user can choose to rescan his items or re-construct a new item set. Next, the user can either manually enter an expiry duration that best fits his expectations for particular products, or he can pull estimated data from EatByDates in which it will autofill in the expiry duration for the produce. Lastly, the items and their attributes will be stored in a database which can be retrieved and displayed on a UI page. The following figure shows the algorithm flow of the process.

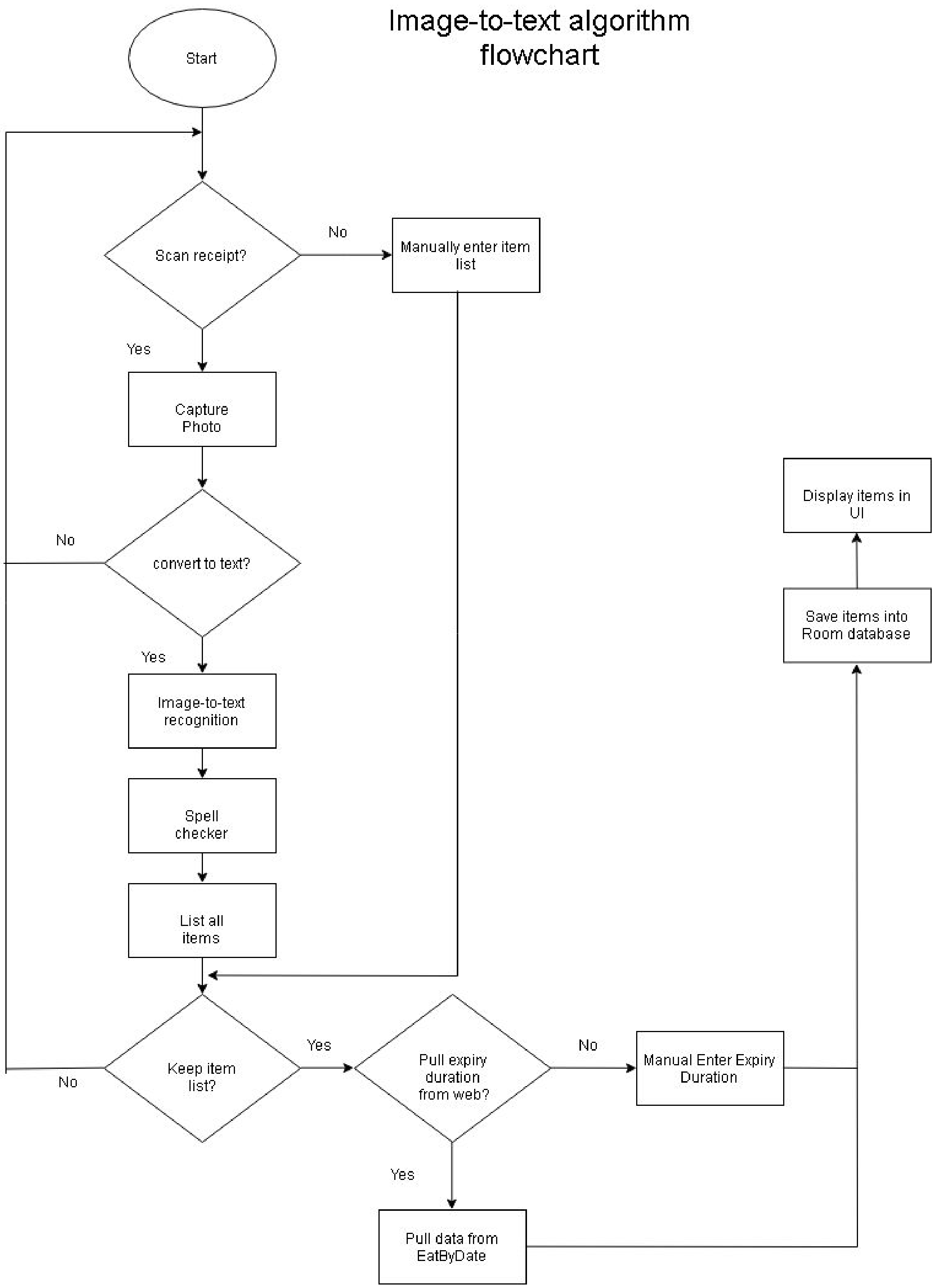


Figure 3.3: Algorithm Flow for Image to Text

The user could request raw data from Arduino through a Bluetooth interface in the app. After that, the raw sensor data will be processed into concentration levels which will be displayed on the application's UI. The user can then select to consume the item by which the application will automatically delete it from the database. The below flow chart visualizes this sequence of operations.

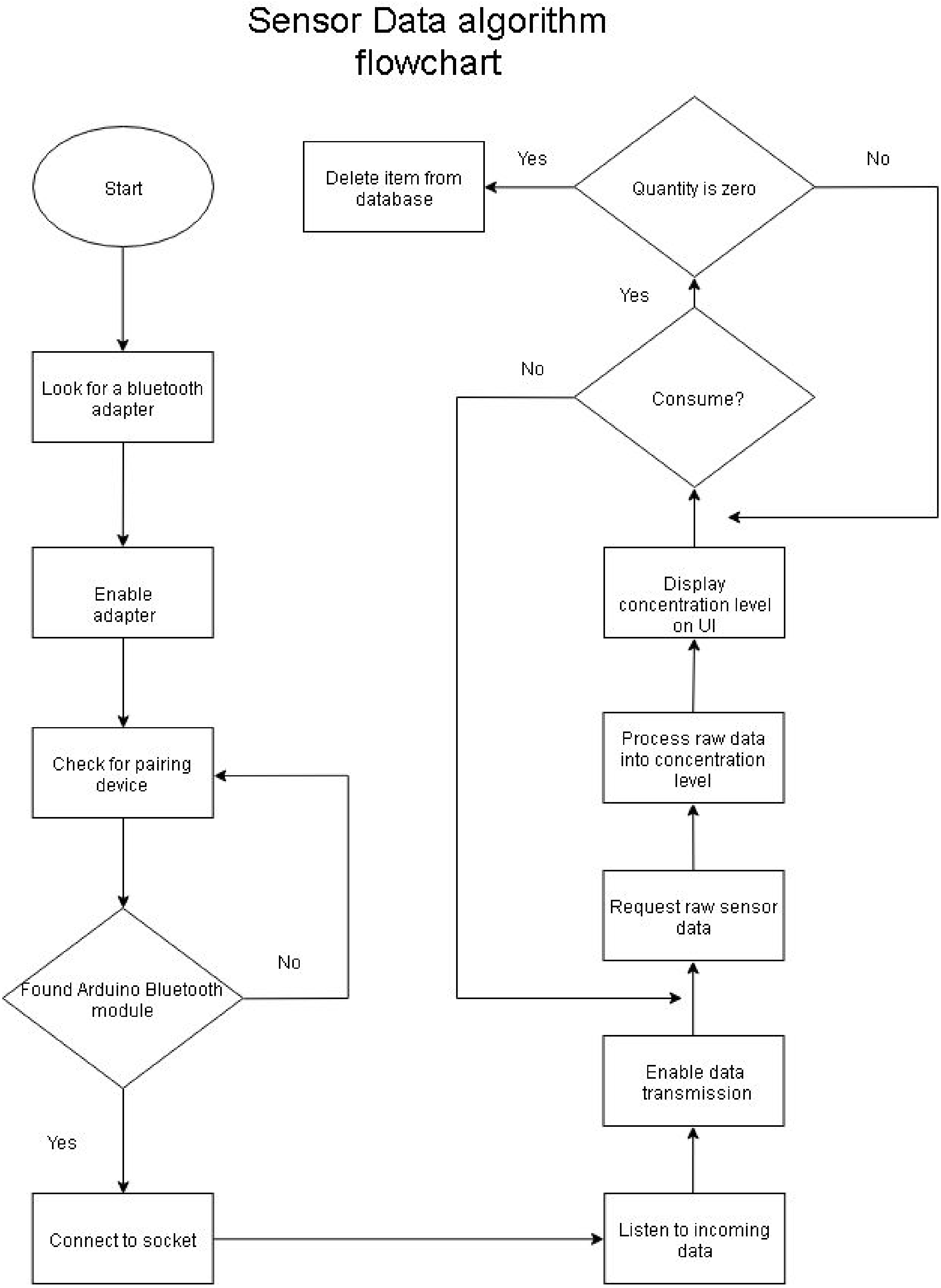


Figure 3.4: Sensor Data Algorithm Flow

3.5 Image Processing Module

The firebase MK kit integrates several Google's ML technologies such as the Google Cloud Vision API, TensorFlow Lite, and the Android Neural Networks API together in a single SDK. The ML kit comes with a set of ready-to-use API for common use cases such as text recognition, face detection, barcode scanning, and image labelling. For our application, we will use the free pricing on-device text recognition feature. The ML kit provides an easy way to recognize text from a variety of image types. Our application will provide two options for the user to either scan a receipt and recognize the text immediately after (Bitmap), or pick a previously taken receipt in his photo album and recognize the text from there (Media Image). Once the user has decided on using either of the two options, a FireBaseVisionImage object can be created from the image type and passed into a FireBase ML model. We can then use the on-device text recognizer within the ML model to convert the text in the image into a text block and segment the text block into a hierarchical structure consisting of lines, sentences down to a single word or an element. The below two figures are the diagrams for the text recognizer and the hierarchical structure for breaking down the text from an image.

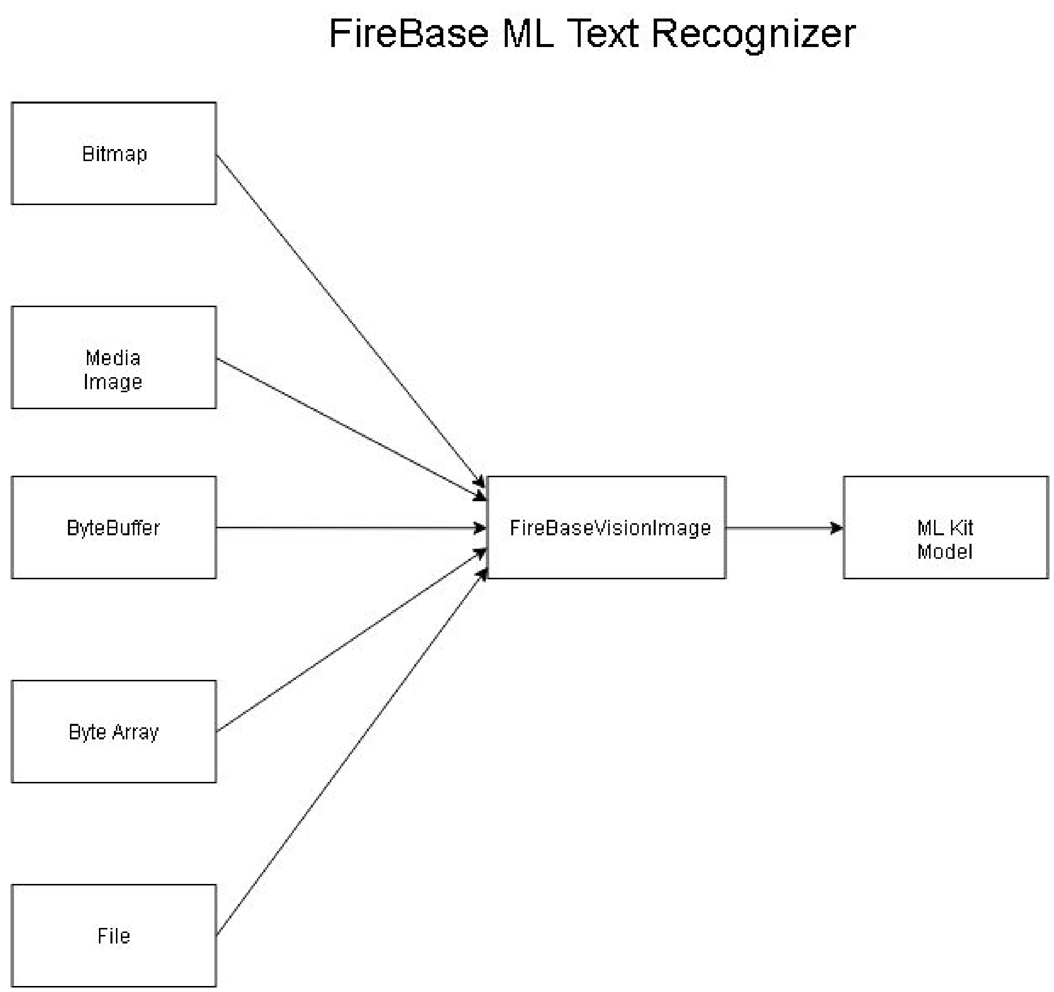


Figure 3.5: FireBase Machine Learning Text Recognizer

```
Image
   |
   |___ TextBlock
   |        |
   |        |___ Line
   |               |
   |               |___ Element
   |
   |___ TextBlock
            |
            |___ Line
            |      |
            |      |___ Element
            |      |
            |      |___ Element
            |
            |___ Line
                   |
                   |___ Element
```

Figure 3.6: Segmenting the Image Into Hierarchical Components

We noticed that some of the grocery receipts may contain abbreviations to conserve ink and paper. For instance, “APPLE” may be shown up as “APPL” in the receipt. Therefore, we thought about utilizing the android spell checker API[21]. The below flow chart illustrates how to correct abbreviated words.

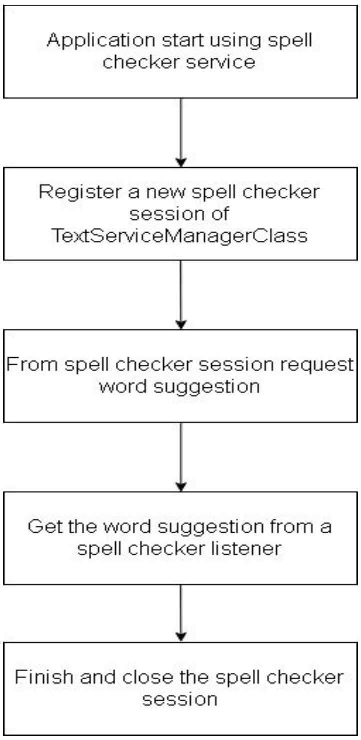


Figure 3.7: Spell Checking Flowchart

3.6 Bluetooth Interface

For setting up our Bluetooth interface, we will first pair our android application with the HC-05 Bluetooth module connected to the Arduino microcontroller. The figure below details the setup of our application's Bluetooth interface. It is important to note that if the MAC address for the HC-05 is not found, it is better to try pairing the phone with the HC-05 module again.

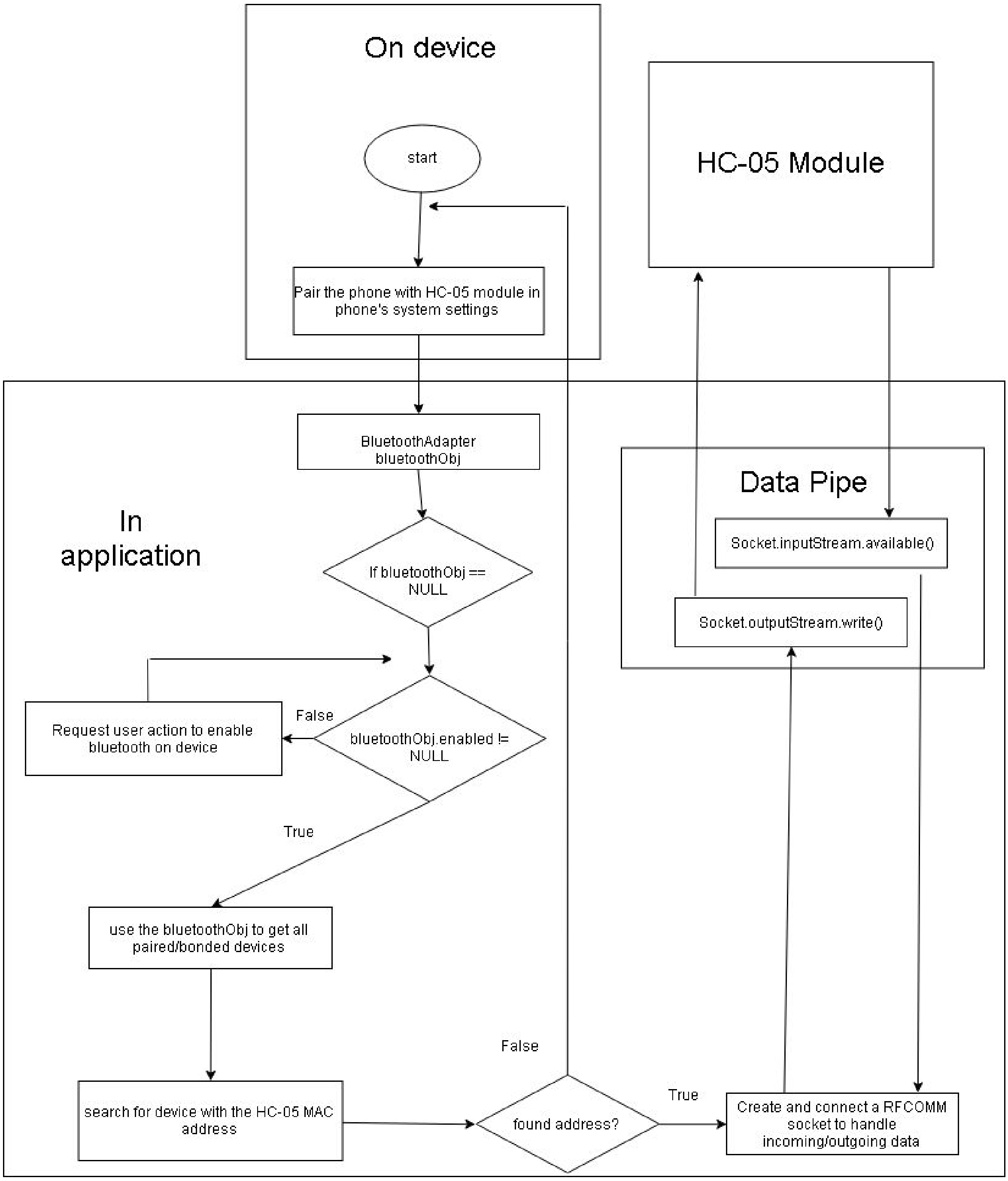


Figure 3.8: Application Bluetooth Setup and Data Transmission

3.7 DataBase and Display

To save our in-app data and all the user information, we will utilize the Android Room database API. The application will use the Room database to get data access objects. Data access object or DAO are main classes where you can define your database interactions. They can include a variety of query methods such as insert, delete, and select. The application will use each DAO to get entities from the database and save any changes to those entities back to the database. Finally, the app uses an entity to get and set values that correspond to the table columns within the database[19]. Upon the startup of the application, it will first load all of the items that the user possesses and display them in the "grocery list" page if the user chooses to view his list. When confirming the list of items to add for the reminder, the list will be stored into the database. If the user closes the application while editing his fruit and vegetable list, all of the temporary data will be stored and loaded upon restart of the application. Below is an architectural representation of the interactions between Room database and our application.

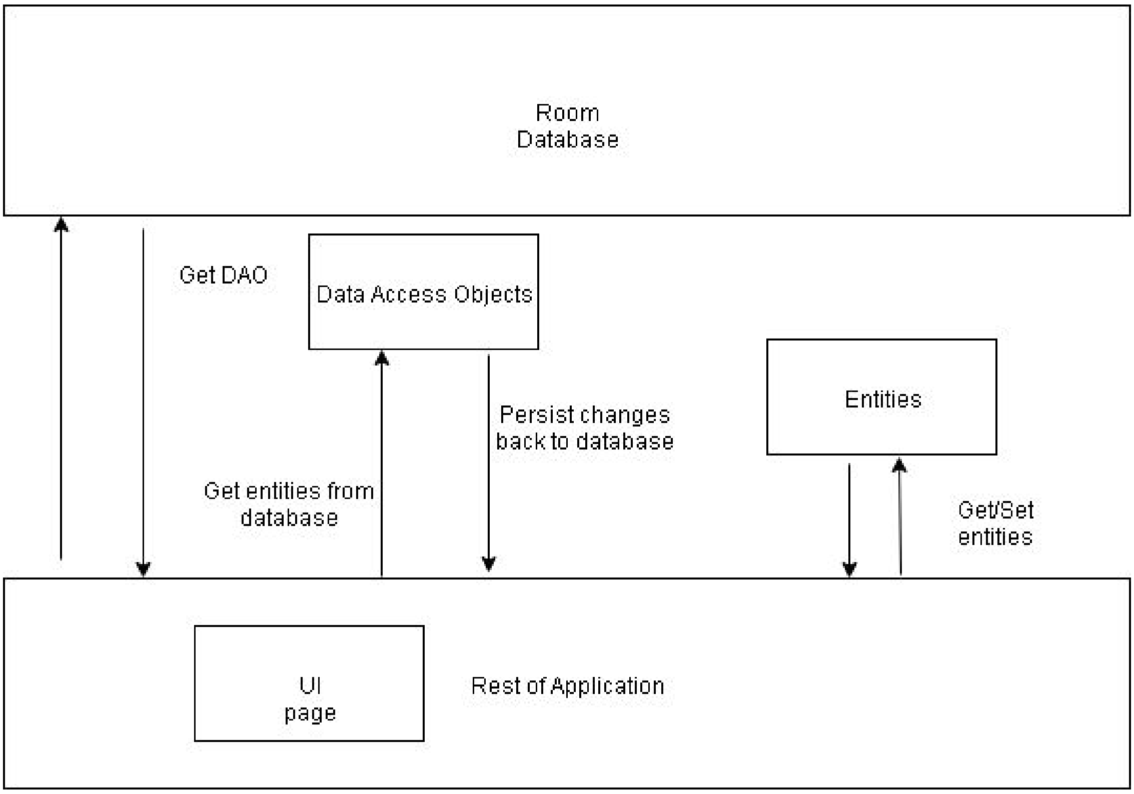


Figure 3.9: Room Database Architecture

4 Fridge Environment Detector Design

To build a detector that detects the environment of the refrigerator and can be installed in a refrigerator or any other enclosed food storage device to detect food spoilage.

4.1 System Overview

The detection result is notified to the user through Bluetooth data transmission, and users can check the internal status of the fridge on the mobile application. Therefore, it not only tells the environmental conditions inside the fridge but also gives us a reference data about food spoilage. By analyzing expired date and measured gas concentration the specific food, we can easily know which food is about to expire or already be spoilt. Then, users can consume or discard the item according to its status. Due to the decay of food, the concentration of certain gases and chemicals such as carbon dioxide, ethanol, are increased. So it can be detected by an array sensor that is the core of the design. The Bluetooth module located on the electronic node and the central UI node will implement the set wireless communication and scalability.

4.2 General Design

In order to keep the system as modular as possible and UI user-friendly, we have designed a separate unit. The power supply consists of four AA batteries and provides a voltage of 6V, which is supplied to each unit via a DC-DC converter. The sensor array unit consists of a CO_2 sensor, an alcohol sensor and a temperature and humidity sensor, all of which work together to provide a value for the detection of chemicals at a specific temperature and humidity settings to provide accurate readings. The input from the unit is fed to the microcontroller, which interprets the reading and passes the signal to the Bluetooth module for transmission to the receiver. The appropriate measured values are displayed on the screen. Then through the Bluetooth connection, the information is transmitted to the mobile phone through our designed app, the data is interpreted. Thus, users can judge the food spoilage conditions based on the measured data and its approximate expired date to inform the user for potential food contamination. Since different food may emit various gases during decay, combining data of the gas concentrations and food expired dates, we can accurately determine and estimate which specific food may spoil or be about to expire. The sensor can be placed anywhere in the refrigerator or in a storage unit close to the food. The internal subsystem can communicate with the display unit placed outside the refrigerator through a Bluetooth module that is paired with each other. We decided to continue this design because it is scalable and economical to manufacture.

4.2.1 Physical Design

The system consists of the sensor array, Bluetooth transmitter, microcontroller and LCD. From the front, this device has an LCD display that can display data and a control switch that can turn on/off the power, as Figure 4.2.1 shown. The inside of this device has sensors array, Bluetooth device and microcontroller, as Figure 4.2.2 shown.

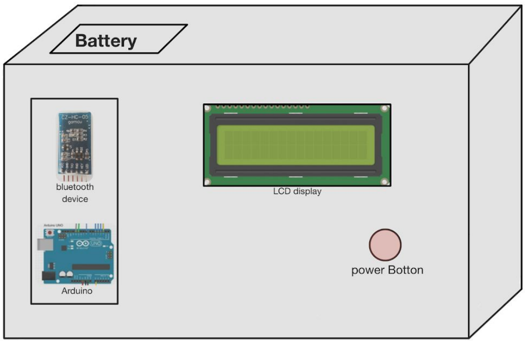


Figure 4.1: Detector Front View

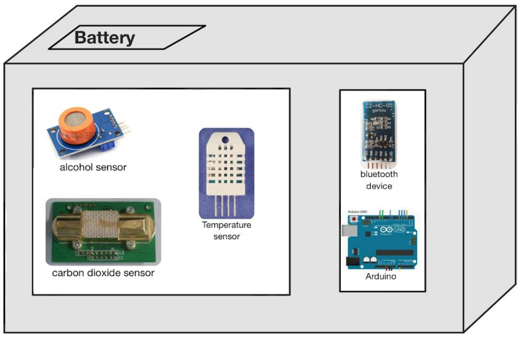


Figure 4.2: Detector Inside View

4.3 Sensor Specification

FooMinder various sensors for measuring the inner of the fridge. There will be 3 sensors that are utilized in conjunction to obtain the data: the carbon di-oxygen sensor, alcohol sensor, and the temperature sensor.

Code	Description
Des 4.3.1 C	All sensors will communicate on a protocol compatible with the microcontroller.
Des 4.3.2 P	Alcohol sensor should be able to detect in the range 20-100 ppm.
Des 4.3.3 P	CO2 sensor should be able to detect in the range 0-5%VOL.
Des 4.3.4 P	The sensors should be able to operate between 0°C - 10°C.
Des 4.3.5 P	The sensors should be able to detect and function in humid conditions with an average relative humidity of 70-80% at 4°C.

Table 4.3: Sensor Design Specification

4.3.1 CO_2 Sensor: MH-Z14

The MH-Z14 carbon dioxide sensor will be used to measure the percentage of carbon dioxide content.

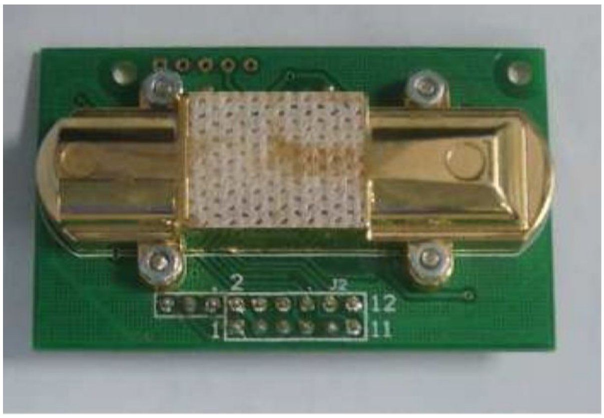


Figure 4.3: MH-Z14 Sensor[16]

The table below shows the general specification of MH-Z14 sensor.

Working voltage	4V ~ 6V
Working current	Max current <100mA,Average <50mA
Detection range	0~10000ppm (optional)
Resolution ratio	5ppm (0~2000ppm)
	10ppm (2000~5000ppm)
	20ppm (5000~10000ppm)
Accuracy	±50ppm±5%
Repeatability	±30ppm
Responsible time	< 30s
Working temperature	0~50℃
Working humidity	0%~90%RH (No condensation)
Storage temperature	-20~60℃

Table 4.3.1: MH-Z14 Specification[16]

The following formula shows how the concentration (C_{ppm}) can be calculated, where TH is the time for high level during an output cycle, and TL is Time for low level during an output cycle.

$$C_{ppm} = 2000 \times (TH - 2ms) / (TH + TL - 4ms)$$

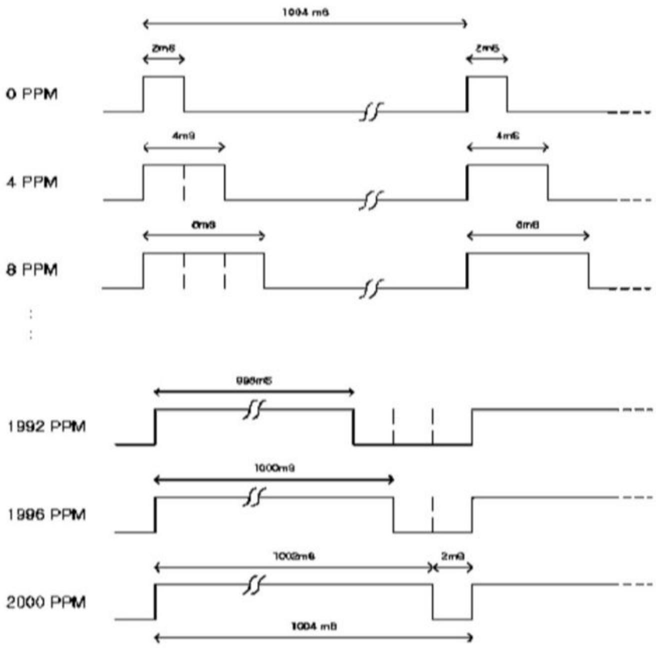


Figure 4.4: PWM Output of CO_2 Sensor

The following table presents the pinouts for MH-Z14.

Pin	**Description**
Pad1, Pad15, Pad17	Vin（input voltage 4.5V～5.5V）
Pad2, Pad3，Pad12, Pad16	GND
Pad4	Vout2 (0.4～2V)
Pad5	Vout1 (0～2.5V)
Pad6	PWM
Pad8	HD
Pad7、Pad9	NC
Pad11, Pad4, Pad18	UART（RXD） 0～3.3V input digital
Pad10, Pad13，Pad12, Pad16	UART（TXD） 0～3.3V output digital

Table 4.3.2: MH-Z14 Pinouts

4.3.2 Alcohol Sensor: MQ-03

The MQ-03 alcohol sensor will be used to measure the percentage of alcohol content.

Figure 4.5: MQ-03 Sensor[17]

The following table shows the general specification for MQ-03 sensor.

Power supply	5.0±0.2V (AC or DC)
Load resistance	200KΩ
Heater resistance	33Ω±5%
Using Tem	-10°C-50°C
Related humidity	less than 95%Rh
Pins	1 Gas sensing layer 2 Electrode 3 Electrode line 4 Heater Coil 5 Tubular ceramic 6 Anti-explosion network 7 Clamp ring 8 Resin base 9 Tube Pin

Table 4.3.3: MQ-03 Specification

The figure below shows the dimensions for MQ-03 sensor.

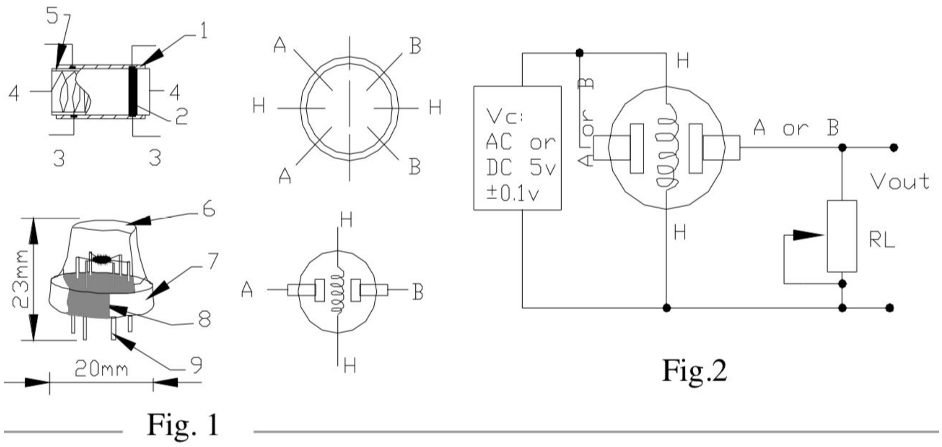


Figure 4.6: MQ-03 Sensor Dimensions[22]

The following table presents the pinouts for MQ-03.

Pin	**Description**
VCC	This pin powers the module, typically the operating voltage is +5V.
GND	Used to connect the module to the system ground.
Digital Out (DO)	You can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer.
Analog Out (AO)	This pin outputs 0-5V analog voltage based on the intensity of the gas.

Table 4.3.4: MQ-03 Pinouts

4.3.3 Temperature Sensor: DHT-22

The DHT-22 Temperature sensor will be used to measure the temperature content.

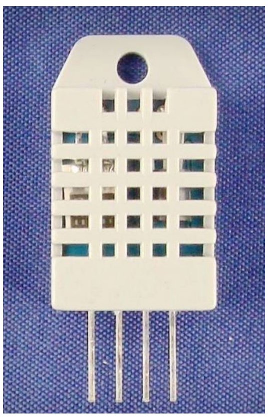


Figure 4.7: DHT-22 Sensor[3]

The table below shows the general specification for DHT-22 sensor.

Power supply	3.3-6V DC
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Pins	(1) VDD (Power Supply) (2) Data - Signal (3) Null (4) GND

Table 4.3.5: DHT-22 Specification[3]

The figure below shows the dimensions for DHT-22 sensor.

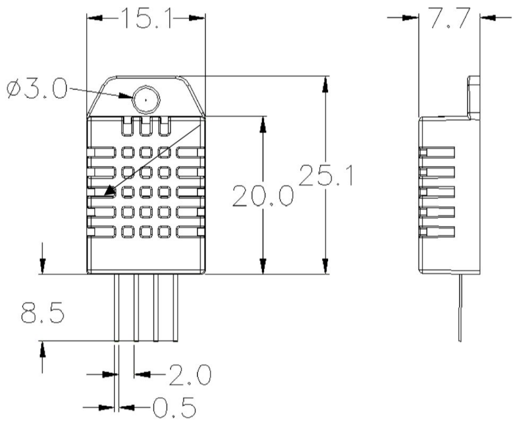


Figure 4.8: DHT-22 Sensor Dimensions[3]

The following table presents the pinouts for DHT-22.

Pin	Description
Vcc	Power supply 3.5V to 5.5V
Data	Outputs both Temperature and Humidity through serial data
NC	No Connection and hence not used
Ground	Connected to the ground of the circuit

Table 4.3.6: DHT-22 Pinouts

4.3.4 HC-05 Bluetooth Module

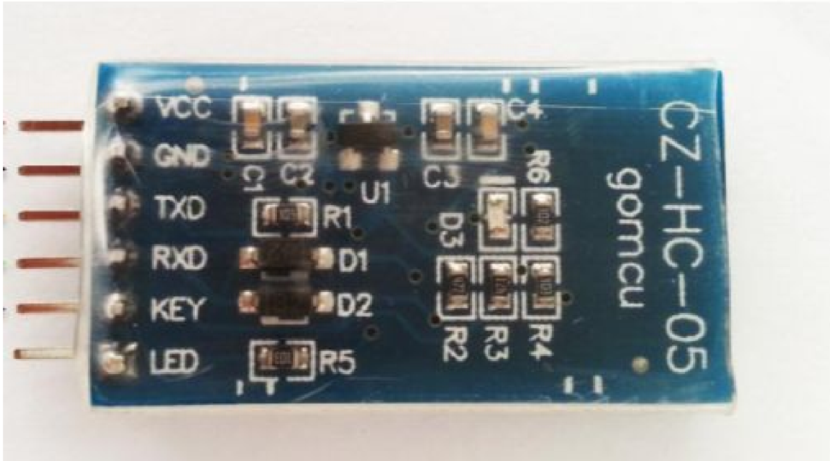


Figure 4.9: HC-05 Module

The table below shows the general specification and pinouts for the HC-05 module.

Pinouts	Descriptions
VCC	Connect to +5V
GND	Connect to Ground
TXD	Connect with the MCU’s RXD PIN.
RXD	Connect with the MCU’s TXD PIN.
KEY	If it is input low level or connects to the air, the module is at paired or communication mode. If it’s input high level, the module will enter to AT mode.

Table 4.3.7: HC-05 Pinouts

5 Hardware Design

5.1 System Overview

In this system, which is mainly consisting of three detectors and an Arduino UNO microcontroller. These detectors are MH-Z14 CO2 module, MQ-2-135-3-7-9 alcohol and smoke module, and DHT22 temperature and humidity module respectively. We can control the system status “ON” and “OFF” by making a simple switch. Once the system starts working, all the measured data can be displayed on the LCD screen. According to the data we measure, by Bluetooth data transmission, the data can be sent out to the mobile terminal for further data analysis.

5.2 General Design

In general, the whole system can be integrated properly and operating regularly by correct physical connection. Once the system is on, these three sensors can detect concentrations of CO2, alcohol, and humidity simultaneously which provide us with three standards to judge the inside status of the fridge. These data can be used not only to judge the internal environment of fridges but also to judge the deterioration or freshness of food. Before the food is put into the fridge, an approximate shelf life would be given. Then we will select the specific kinds of food which may deteriorate according to the different concentrations of gas based on that different foods emit various gases during the deterioration process. Then we can further accurately infer which kind of food is going to be expired or discarded according to the predicted shelf life. Thus, users can know the spoilage conditions of food in real time and eat it before expiration, or clean up some spoiled food based on the fridge internal condition.

5.3 Arduino Microcontroller

Code	**Description**
Des 5.3.1 C	The MQ-03 alcohol sensor should use the analog pins.
Des 5.3.2 C	The MHZ14 CO2 sensor should use the analog pins.
Des 5.3.3 C	The DHT-22 temperature sensor should use the analog pins.
Des 5.3.4 P	HC05 Bluetooth modules use the UART interface provided by the MCU to communicate with it. These are RX and TX pins present on the Arduino.

Table 5.3.1: Microcontroller Specification

The table below shows the technical specifications for the hardware.

Operating Voltage	5V
Recommended Input Voltage	6-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)

DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1KB
Frequency (Clock Speed)	16 MHz

Table 5.3.2: Technical Specification

5.4 Power Consumption

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 6 to 12 volts. When working regularly (6v-12v): 40mA*6v = 0.24w, 40mA*12v = 0.48w, so the range of power: 0.24W~0.48W. We intend to use a 6V battery, so the power consumption is 0.24W.

5.5 Pin Setup for Arduino Uno

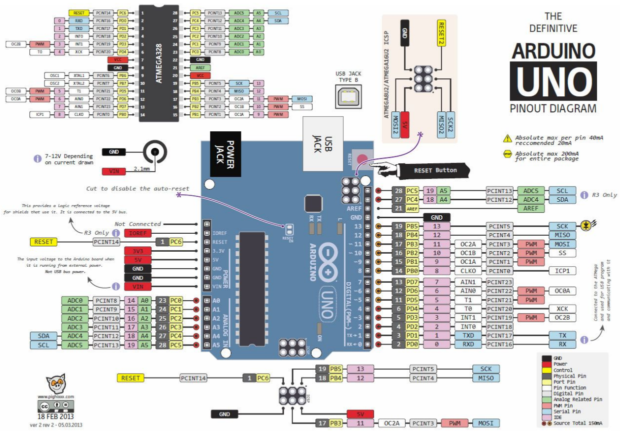


Figure 5.1: Arduino Uno[23]

The following table shows the pin set up of Arduino board.

Pin Type	**Pin Name**	**Description**
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power

		microcontroller and other components on the board. 3.3V: 3.3V supply generated by the onboard voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as an input or output pins.
Serial	0(RX), 1(TX)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.

Table 5.5.1: Arduino Uno Pinouts

5.6 Circuit design and pinouts connection

We integrate all three sensors with the microcontroller unit. And by connecting an LCD screen, we are able to see the feedback information and related results. Basically, we apply an input voltage from the processor to sensors and ground them. The specific circuit connection is shown in the following figure.

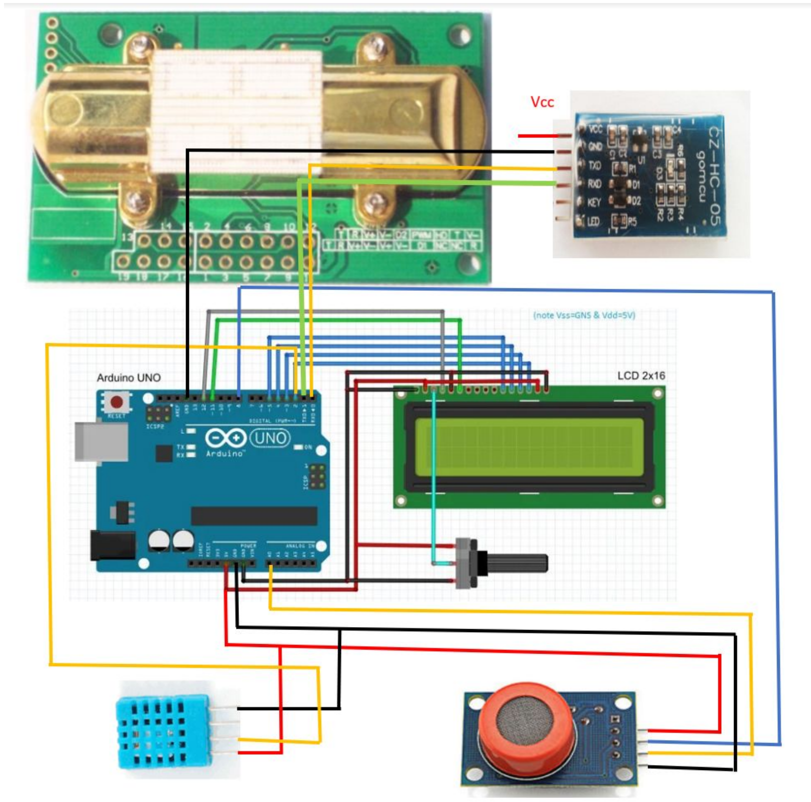


Figure 5.2: Circuit Design

5.7 Communication Method

5.7.1 UART

Code	**Description**
Des 5.7.1 C	The LCD should be connected and maintaining an operating voltage with 5v.
Des 5.7.2 P	Arduino Uno has UART to communicate with the Bluetooth module.
Des 5.7.3 C	By connecting the I2C LCD display to the processor, the communication and data from the processor can be sent and displayed on the LCD screen.
Des 5.7.4 C	By connecting sensors to the processor, so that sensors can be powered (a voltage of 5v) and operating simultaneously. Three pieces of information can be displayed on the screen.

Table 5.7.1: UART Specification

5.7.2 Bluetooth

Code	Description
Des 5.7.5 P	The signal from the processor can be transmitted through the Bluetooth module to our mobile application.
Des 5.7.6 P	The Bluetooth module should be able to work with a serial communication interface after the processor transmits.
Des 5.7.7 P	The module should be set to command mode, connect the Key pin to any pin on the module on the Arduino Uno and set it to high to allow the module to be programmed.

Table 5.7.2: Bluetooth Specification

5.8 General Workflow Overview

For the workflow of the whole system, we integrate sensors array which consists of CO2, alcohol, temperature and humidity sensors with a microcontroller -- Arduino Uno. Once the microcontroller is powered on, then those sensors would be powered by a microcontroller and start to work. Moreover, all the measured data from sensors, like various gas concentration values. In other words, all the data need to be processed and passed through the microcontroller. Next step, we use an HC-05 Bluetooth module to communicate with Arduino Uno. We are supposed to transfer data to the mobile application terminal by Bluetooth. Therefore, we can easily see the data gained from microcontroller on mobile devices. The figure below shows the general working flow.

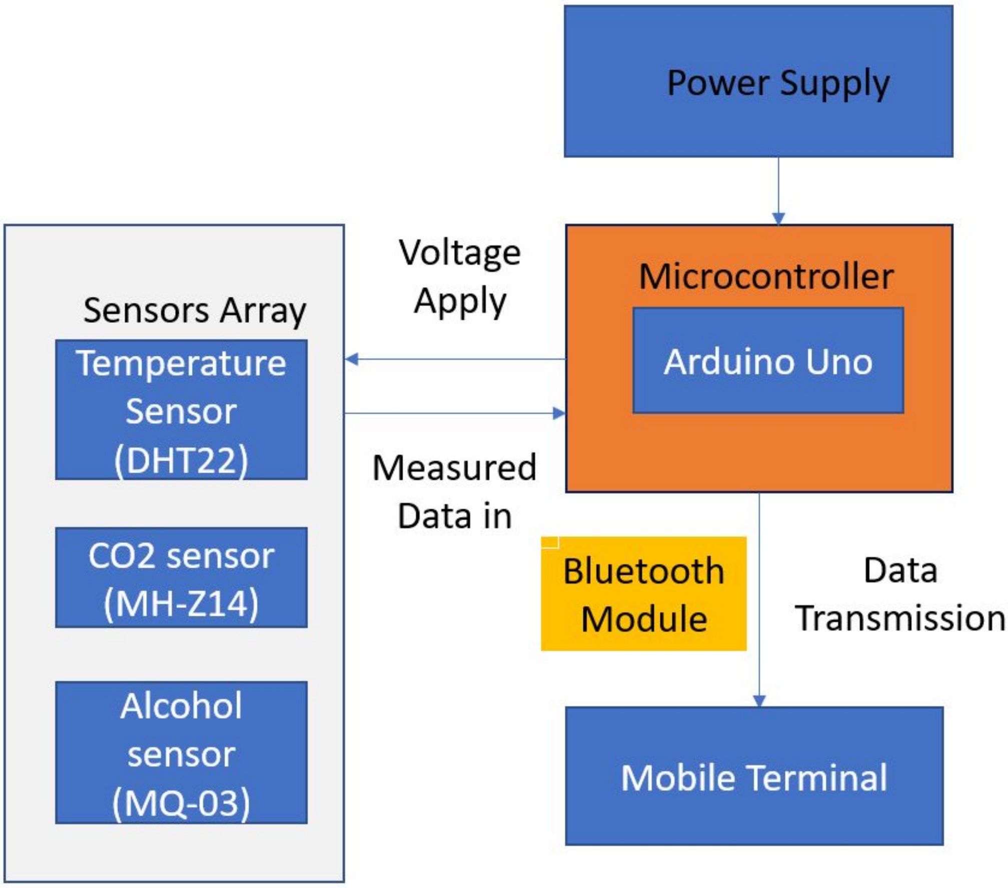


Figure 5.3: Workflow Diagram

5.9 Design Alternatives

In order to ensure that the functions of our products remain unchanged, we have prepared an alternative. In our design products, Arduino and sensor array with detecting CO2, alcohol and temperature are used. In the next progress study, we may encounter a variety of vegetables and fruits which emit a variety of different gases in the process of deterioration. However, the sensors we are using currently may be limited to some vegetables and fruits. Therefore, we may replace sensors to extend the scope of use as far as possible. Moreover, Since we are working with gases which have complex fluid dynamics, the ability of the sensors is to accurately and consistently detect the concentration of gases, which would determine the correctness of the outcome. So, If some of the sensors are not working and measuring accurately, we will try to abandon them for the sake of the accuracy of the results and the reduction of the error range.

5.10 Product Advantages and Future Consideration

Intelligent appliances with multimedia capability have been increasingly apparent in our daily lives. As the fast advances of science and technology, and we are able to widely use the Internet and convenient applications. Our design can be strictly regarded as a part of a smart home system. A smart home is one of the most significant areas of applying intelligent appliances, and fridges are used in people's daily life every day. In other words, this is an indispensable part of people's life. The advantages of our designs are reminding users with the latest information about food conditions of shelf life and quantities, as well as the environmental conditions inside the fridge. Therefore, it can not only greatly reduce food waste because users may forget to eat or not remember what is left in the fridge, but also can improve the internal environment of the fridge. We know that nowadays smart fridges are widely used, but most people can not afford it due to its high price. Basically, smart fridges do not have the function of food expiration warning. First of all, our products are affordable in terms of price. Then, the function of predicting and reminding the shelf life of food will bring different experience and convenience to users. In future plans and considerations, we will continue to improve and optimize our products. In order to meet my purpose of portable and accurate measurement, we will optimize the appearance and reduce the size of products, as well as the location of various detectors. At the same time, without affecting the function of the product, we intend to lower our manufacturing costs as far as possible. It is important to be accepted and satisfied by most users and bringing great convenience to life.

6 Engineering Standards

We mainly divide into electrical standards and environmental standards.

6.1 Electrical

The following table lists the engineering standards will be used during the development.

Standard ID	**Description**
IEC 61558-1:2017	Safety of transformers, reactors, power supply units and combinations thereof - Part 1: General requirements and tests[11]
IEC 60086-1:2015	Primary batteries - Part 1: General[10]

Table 6.1: Electrical Standards

6.2 Environmental

The following are the environmental standards used during the development.

Standard ID	**Description**
IEC 60050-904:2014	International Electrotechnical Vocabulary (IEV) - Part 904: Environmental standardization for electrical and electronic products

	and systems[9]
CAN/CSA-ISO/ TR 14062-03 (R2013)	Environmental Management - Integrating Environmental Aspects into Product Design and Development (Adopted ISO/TR 14062:2002, first edition, 2002-11- 01)[1]

Table 6.2: Environmental Standards

7 Conclusion

This document clearly outlines the general system design specifications and discussed different phase deliverables in perspectives of Proof-of-Concept, Prototype and Final Product. The detailed design requirements and process are presented for software and hardware components. This document also provided a User Interface and Appearance in the Appendix section which analyzed the current user interface design from the aspects of Graphical Presentation, User Analysis and Technical Analysis. In addition to the above specifications, we also provide a Supporting Test Procedure for future record. This document will provide a solid guide to assure the satisfaction of functional requirements for each step.

FooMinder is designed to be a food reminder system consists of a mobile application and a fridge detector. The application uses an image processing algorithm to create an image-to-text converter. Then the application will use this converter after the user takes a picture of the receipt and extract the grocery list. The application will combine the grocery list with the expiration date data from the web to set a reminder to alert the user of the expiry date. The detector can be placed inside the fridge and detect the environment by sensing different chemical gases concentration. All data acquired during the processing will be visualized in the application.

The Proof-of-Concept Deliverables will be presented before August 15, 2019. As a group of engineers who are passionate to solve one problem we discovered in real life, our intention is to provide a user-friendly and cost-friendly tool to help users get organized with their grocery and help reduce food loss and save money. Thus give the effort to meet the world's sustainable goals.

8 Reference

[1] CSA Group (2013). “CAN/CSA-ISO/TR 14062-03 (R2013) - Environmental Management - Integrating Environmental Aspects into Product Design and Development (Adopted ISO/TR 14062:2002, first edition, 2002-11-01)”. CSA, Mississauga.

[2] Dhawan, Hitanshu. (2018, Oct, 15). “Firebase ML Kit 101: Text Recognition - AndroIDIOTS”. Retrieved from https://medium.com/androidiots/firebase-ml-kit-101-text-recognition-3c88f468bff6

[3] “Digital-output relative humidity & temperature sensor/module ”. (n.d). Aosong Electronics [PDF file]. Retrieved from https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf

[4] “Ergonomics of human-system interaction -- Part 11: Usability: Definitions and concepts”. (2018, April 4). Retrieved from https://www.iso.org/standard/63500.html

[5] “Ergonomics of human-system interaction -- Part 161: Guidance on visual user-interface elements”. (2016, February 17). Retrieved from https://www.iso.org/standard/60476.html

[6] “Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems”. (2019, July 4). Retrieved from https://www.iso.org/standard/77520.html

[7] “General Requirements for Battery-Powered Appliances”. (2015, September 19). Retrieved from https://standardscatalog.ul.com/standards/en/standard_2595

[8] “How Long Does Food Last? Guide to Shelf Life & Expiration”. (2011). Retrieved from http://www.eatbydate.com/

[9] “IEC 60050-904:2014 International Electrotechnical Vocabulary (IEV) - Part 904: Environmental standardization for electrical and electronic products and systems”. International Electrotechnical Commission (2014). Retrieved from https://webstore.iec.ch/publication/22029

[10] “IEC 60086-1:2015 Primary batteries - Part 1: General”. International Electrotechnical Commission (2015). Retrieved from https://webstore.iec.ch/publication/23002

[11] “IEC 61558-1:2017 Safety of transformers, reactors, power supply units and combinations thereof - Part 1: General requirements and tests”. International Electrotechnical Commission (2017). Retrieved from https://webstore.iec.ch/publication/26261

[12] IEEE. (2004). “IEEE Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments”. Retrieved from https://standards.ieee.org/standard/1621-2004.html#Additional

[13] IEEE. (2011). “IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)”. Retrieved from https://standards.ieee.org/standard/802_15_4-2011.html

[14] IEEE. (2016). “IEEE Standard for Camera Phone Image Quality”. Retrieved from https://standards.ieee.org/standard/1858-2016.html

[15] “Information technology -- Computer graphics and image processing -- Conformance testing of implementations of graphics standards”. (2016, September 19). Retrieved from https://www.iso.org/standard/18736.html

[16] “MH-Z14 CO2 Module”. (n.d). Futurlec [PDF file]. Retrieved from https://www.futurlec.com/Datasheet/Sensor/MH-Z14.pdf

[17] “Mq3 Gas Sensor Module (alcohol)”. (n.d). Retrieved from https://www.indiamart.com/proddetail/mq3-gas-sensor-module-alcohol-12392978548.html

[18] Norman, D. (2013). The design of everyday things / Don Norman. (Revised and expanded ed.).

[19] “Save data in a local database using Room: Android Developers”. (n.d). Retrieved from https://developer.android.com/training/data-storage/room

[20] “Sensory analysis -- Methodology -- General guidance for measuring odour, flavour and taste detection thresholds by a three-alternative forced-choice (3-AFC) procedure”. (2018, April 24). Retrieved from https://www.iso.org/standard/68901.html

[21] “Spell checker framework: Android Developers”. (n.d). Retrieved from https://developer.android.com/guide/topics/text/spell-checker-framework

[22] “Technical Data MQ-3 Gas Sensor”. (n.d). Hanwei electronics [PDF file]. Retrieved from https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf

[23] “The Full Arduino Uno Pinout Guide [including diagram]”. (2018, November 18). Retrieved from https://www.circuito.io/blog/arduino-uno-pinout/

Appendix

A User Interface and Appearance Design

A.1 Introduction

During the design of FooMinder, we aim to create a product that gives a simple and neat user interface that is intuitive to every user. It is our highest priority to make sure the product is having full functionalities as well as support without adding extra rules to users' lives, especially for elderly people and families. It is important that the product is straightforward to use and easy to get into the habit of using it for the users.

Purpose

The purpose of this appendix is to explain the user interface design of FooMinder. It will provide explanations for the design choices by taking into consideration of "Seven Elements of UI Interaction" which outlined in the ENSC405W lectures and Don Norman's text "The Design of Everyday Things".

Scope

This appendix demonstrates in the following six perspectives during the UI design progress.

1. Graphical Presentation

This section provides an overview of the graphical presentation of our product including the Android Application interface and the hardware appearance.

2. User Analysis

This section discusses the essential information and restrictions for the users when using this product.

3. Technical Analysis

This section explains how we take into account the "Seven Elements of UI Interaction" which are Discoverability, Feedback, Conceptual Models, Affordances, Signifiers, Mapping and Constraints during design development.

4. Engineering Standards

This section outlines the engineering standards related to the user interface of our product.

5. Analytical Usability Testing

This section provides details of the analytical usability testing procedures. It is examined from an analytical perspective to discover any possible ambiguities existing in the product.

6. Empirical Usability Testing

This section provides details of the empirical usability testing methods performed with actual users which will be needed for later iterations of FooMinder.

A.2 Graphical Presentation

The two main components for FooMinder are the fridge detector and the mobile application. The diagrams of the structure and appearance of the fridge detector are illustrated in section 5 in the Design Specification document. Therefore, please refer to the above section to find the description.

In this section, we will go through some graphical representations of how our mobile application may look like.

The following graph shows the main page of our mobile application.

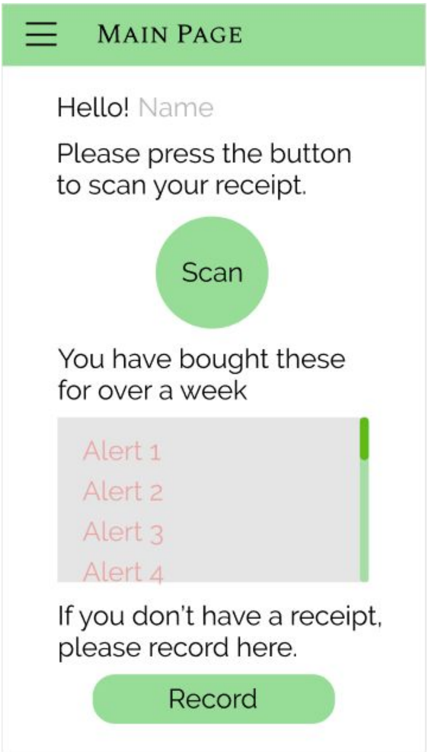


Figure A2.1: Sample App Main Page

As image processing will be one of the main functionalities of our product. The scan button is placed in the middle of the screen. The grocery list is also shown below the scan button which is considered to be a form of notification for the users.

The following series of graphs shows the working procedure that the user will see when using the mobile application.

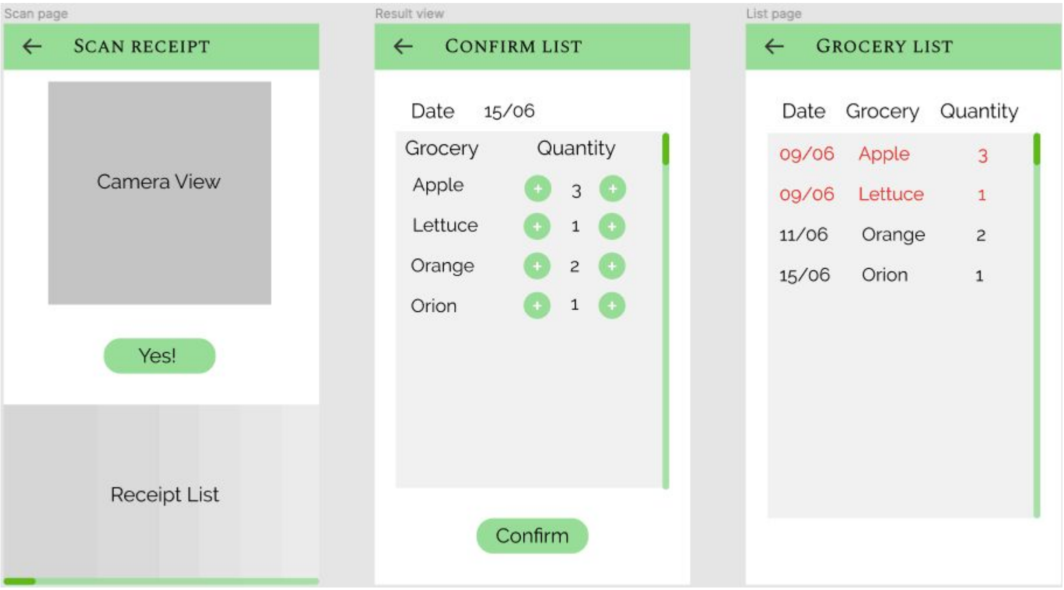


Figure A2.2: Sample Scan Procedure

When the user presses the scan button, they would expect an output of the processed result. Therefore, an output scene will be shown after taking pictures of the receipt. Users can also edit the quantities of the result list in case there are some wrong items. After that, the application will show the final list of this scan.

When the users consume one item from the list, they will be able to change the quantity of it by taping the row of that item. When the quantity goes to zero, this item in the list will be deleted automatically. The figure below shows this feature.

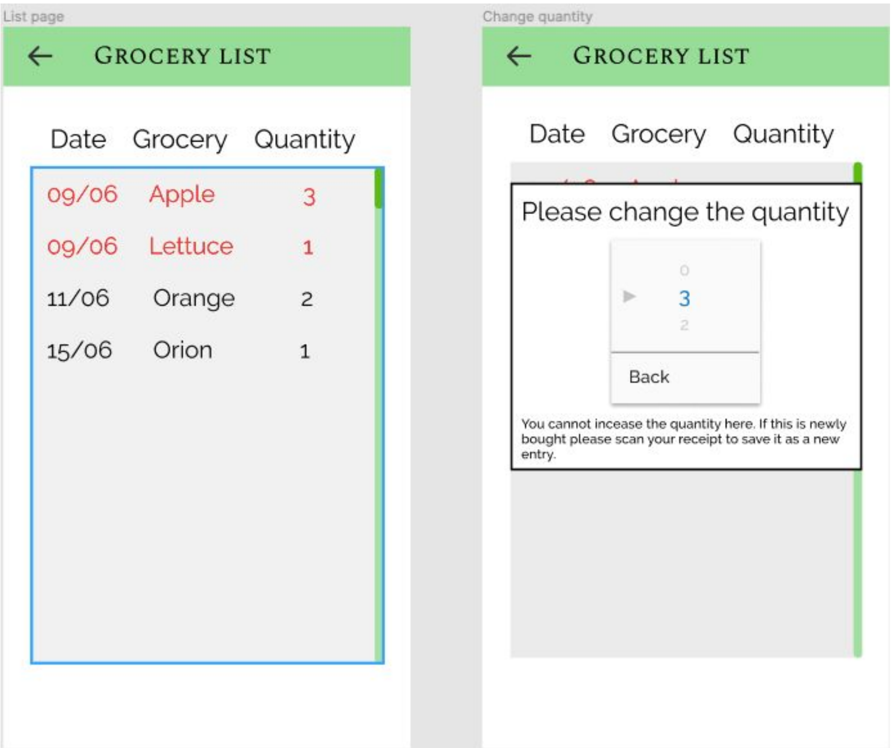


Figure A2.3: Sample Change Quantity Window

A.3 User Analysis

The targeted users of this product are individuals who often forget certain food they bought and find out after the expiry date and families that need a way to organize available food. FooMinder is designed for households that need reminders on the food expiry date so that will reduce waste and save money. There are products that are able to detect the freshness of fruit and vegetables in the market as well as some smart refrigerators. However, smart fridges are usually very pricey and for those old fridges that have been used for decades will not be compatible with these new design products. Therefore, we aim to provide a cost-effective solution for those needing support but don't want to spend a lot of money on it.

The detector device itself should have a reasonable size that won't occupy too much space in the fridge. It also needs to be able to secure inside the fridge so that it can stay in its place and continuously get stable data. There will be also responsible for the users to keep it from the children.

Users will need to download the companion app for the detector so that they can visualize the results from the sensors. The app will basically perform two functionalities. The first one is by taking pictures it will perform image processing and the other one is receiving data from Arduino through Bluetooth and visualize both results in the app. We will be sure to make it simple to understand and easy to use. We expect users would get used to it in just a few minutes by going through the app tour we designed at the beginning of use. The detector should have no delay during connection and give back a status report after everything is set up. The app will also have instructions for setting up the detector on the first tour. The app will have buttons and menus to navigate to different pages and a neat and compact main page to show the top features.

A.4 Technical Analysis

In this section, we will discuss based on the "Seven Elements of UI Interaction" from Don Norman's book "The Design of Everyday Things" which are Discoverability, Feedback, Conceptual Models, Affordances, Signifiers, Mapping, and Constraints.

Discoverability is the principle of making it clear to the users what actions are possible and the current state of the device[18]. We aim to let the users quickly get used to the operations of the app. In order to achieve this, our app will contain the first-time tour for first-time users and this tour is able to review when needed. The app is not going to have too many buttons and repetitive items to minimize the confusion. During the using procedures, the app will have notification pop-up windows to guide the users. For example, the following figure shows an example of this pop-up window. Moreover, we will avoid including unnecessary technical terms without defining them first.

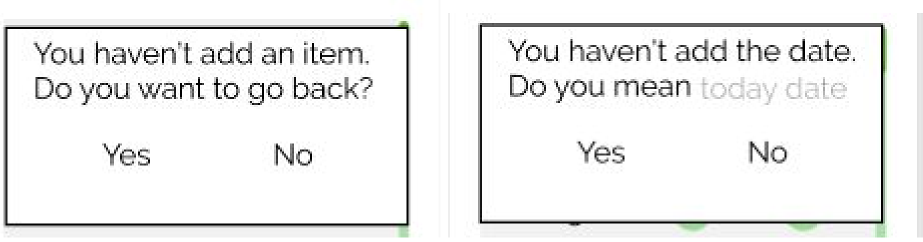


Figure A4.1: Sample Pop-up Windows

Feedback is the principle of making it clear to the users what actions have been taken and what have been accomplished.[18] In order to achieve this, we will have an LCD screen on the detector which shows different status. For example, when successfully establish a connection with the app, LCD will show the message. The app will have a confirmation page after taking a picture of the receipt which lets the user check and change the processed result and then confirm it.

Conceptual Models is the principle of letting the users having understandings and having feelings of control of the product and is able to accommodate similar things[18]. The most general conceptual model could be a user tracking their habit of monthly spending using an app on their phone. There is possibly a market that people who are trying to take notes on their grocery list will be interested in our product.

Affordances are the principle of making it clear for the users about the feature of an object that allows people to know how to use it.[18] In order to achieve this, the app will have well-labelled buttons on the main page and avoid unnecessary menus and interrupts. The layout will keep neat but also have all the essential information. We will have one button for scanning the receipts and one separate menu for connecting detector in the fridge.

Signifiers are the principle of making sure the feedback is well informed and comprehensible[18]. In order to achieve this, the LCD on the detector would be the signifier. The users can be aware of the Bluetooth connection by reading messages on the screen. Therefore, the messages passing on the LCD will be feedback. This information will be communicated back to the user.

Mapping is the principle of making it clear for the users about the relationships between controls and results in the whole product[18]. In order to achieve this, the interaction between the app and the detector will be generally simple. There will be a single page for the detector. The buttons on the page will be well-labelled, for example, Connect the Detector and Request Status. On the other hand, the Scan button is for scanning the receipt and Record Button is for manual input followed with guidance lines.

Constraints are the principle of making it limiting the possibilities of interaction which will simplify the interface and guide the users to the appropriate next action[18]. To use the detector, the users need to keep the device powered by charging the detector or replace the battery when it is low power. The users also need to keep the sensor head clean so that the data will not be disturbed which would be better to keep it from children. To use the mobile app, the user will need to have a smartphone with a camera with enough pixels and Bluetooth to connect with the detector. Users should view the data from the detector through the app.

A.5 Engineering Standards

This section provides particular engineering standards that apply to our proposed user interfaces for the system.

1. **ISO 9241-161:2016**
   Ergonomics of human-system interaction
   Outlines the requirements and recommendations on when and how to use visual user interface elements on software[5].
2. **ISO 9241-11:2018**
   Ergonomics of human-system interaction
   Outlines the framework of the concept of usability and how to apply it to situations where people use interactive systems[4].
3. **ISO9241-210:2019**
   Ergonomics of human-system interaction
   Outlines the requirements and recommendations for human-centred design principles and activities throughout the life cycle of computer-based interactive systems[6].
4. **IEEE 802.15.4-2011**
   IEEE Standard for Local and metropolitan area networks
   Outlines the protocol and compatible interconnection for data communication devices using low-data-rate, low power transmissions in a wireless personal area network[13].
5. **UI-2595**
   General Requirements for Battery-Powered Appliances
   Outlines the general requirements for battery operated devices[7].
6. **ISO 13301:2018**
   Sensory analysis - Methodology - General guidance for measuring odour, flavour and taste detection thresholds by a three-alternative forced-choice (3-AFC) procedure
   Outlines the guidelines for obtaining the data and processing the data on the detection of stimuli that evoke responses to odour[20].
7. **IEEE 1621-2004**
   IEEE Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments
   Outlines the user interface elements for power status control that commonly interact with people in their work and home[12].

8. **IEEE 1858-2016**
   IEEE Standard for Camera Phone Image Quality
   Outlines the quantities of performance of camera-equipped mobile devices[14].
9. **ISO/IEC 10641:1993**
   Information technology - Computer graphics and image processing - Conformance testing of implementations of graphics standards
   Outlines the general framework of testing requirements and procedures for computer graphics and image processing[15].

A.6 Analytical Usability Testing

This section provides the analytical usability testing that we will perform for our product FooMinder. The goal of this test is to help us judge disregarded problems existing in the design of the user interface. The test steps will be presented in two parts: the fridge detector and the mobile application. The detail steps are shown below.

Detector Buttons

1. The power button on the detector device should give a physical feeling of being pressed down.
2. Immediately after the power button is pressed, messages will show on the LCD screen which indicates the current status of the device.
3. After the device is powered on, the Bluetooth module should be turned on at the same time and ready for connecting with the app.
4. After the device is powered on, the sensors should be turned on and ready to transmit data.

Detector Screen

1. The screen should not interrupt the performance of sensor detection.
2. The screen should be in sleep mode when no request for fridge status.
3. The screen should be able to display messages upon associated operations.
4. Word display on the screen should be clear with reasonable brightness.

Mobile Application

1. Every page of the app should have a top bar with a Menu icon and the Title for that specific page.
2. The app should be able to memorize the profile and history of the returned user.
3. The main page of the app should have a button for manual input from the users in case there is no receipt available.
4. The app should contain the first-time tour for users installing for the first time. The tour can be skipped or reviewed.
5. The app should be able to restore the Bluetooth connection if it has been completed once.

A.7 Empirical Usability Testing

In this section, we will include the empirical usability testing procedures that we will perform for our product FooMinder. The goal of this test to get feedback from end users and adjust our product accordingly. The tests will be directed by us and completed by the end users. They are supposed to have no primary knowledge of our product and they will be randomly chosen. We will take down their feedback on the functional design and the user interface design usability. The detailed questionnaire is listed below.

End User Activity 1

Power on the detector

Questions

1. Did you have a physical feeling of the button is pressed down?
2. Did the LCD screen show power on message when you press the button?
3. Were the words on the screen comfortable to read?

End User Activity 2

Turn on the app

Questions

1. Was the first-time tour helpful?
2. How do you feel about the current user interface? It is friendly and intuitive?
3. Did you feel any redundant menu or icon?
4. Is the Scan button on the main page a large size?

End User Activity 3

Connect the detector with the app

Questions

1. Did you have any difficulties locating the connect button?
2. Did you receive connection status feedback of the Bluetooth from both the detector and the app?

End User Activity 4

Scan the receipt

Questions

1. Did you feel any delay when taking the picture of the receipt?
2. If there is a delay, did the waiting indicator show up?
3. Did the Confirm List feedback page come out smoothly?

End User Activity 5

Review the grocery list

Questions

1. Was the layout of the grocery list clear and friendly?
2. Were the items listed in the order from the items bought several days ago to the most recently bought?
3. Were the visual alert set properly, for example, colour and order?

End User Activity 6
User input list
Questions
1. Was the layout clear and friendly?
2. Were you able to add items easily?

A.8 Conclusion

In this appendix, we analyzed our user interface design process including graphical presentation, user analysis, technical analysis, and engineering standards. We also included analytical usability testing and empirical usability testing as a tool in the future which will help us adjust our project to have the simple-use and user-friendly user interface. In the section of technical analysis, we discussed our design based on the "Seven Elements of UI Interaction" from Don Norman's book "The Design of Everyday Things". We were able to discover key features and fundamental components that will support us to improve the user interface. Our goal is to carefully examine each interaction and create a user interface that will be accepted by the users. With our current user interface and accomplished functionalities, we are ready to give the demo. We will exclude all technical information such as sensors and image processing. We plan to demo FooMinder to fellow students and normal families at SFU or in the neighbourhood since they are our primary target group. We are going to record the feedback from each participant and make adjustments to the user interface. For our proof-of-concept and appearance prototype, we will do our best to deliver a user-friendly and simple-use display while having basic function working smoothly.

B Supporting Test Procedure

Hardware Tests plan	
1 - Device Operations	Comments:
The power supply should maintain voltage and current stability at all times Pass☐ Fail☐	
The device should be able to operate between 0°C - 10°C Pass☐ Fail☐	

Proper and stable functioning by the power supply. Pass☐ Fail☐	
2 - Sensor Operations	Comments:
The alcohol sensor can function as an alcometer so that we get an estimate alcohol level. Pass☐ Fail☐	
The CO_2 sensor should function properly, the concentration needs to be displayed on both the LCD screen and mobile application. Pass☐ Fail☐	
The temperature and humidity sensor has to work regularly, the values of measured temperature and related parameter need to be displayed on both the LCD screen and mobile application. Pass☐ Fail☐	
All three sensors should be integrated well and function properly Pass☐ Fail☐	
3 - Display Operations	Comments:
The LCD screen should properly connect to Arduino Uno Pass☐ Fail☐	
LCD should display the text that we want to show Pass☐ Fail☐	

The data of the sensor can be displayed on the LCD screen Pass☐ Fail☐	
4 - Interface Operations	Comments:
The buttons on the device can be pressed and work regularly Pass☐ Fail☐	
The LCD displays the corresponding contents without messy code. Pass☐ Fail☐	
5 - Performance Operations	
For transferring data to the mobile application by Bluetooth, we can directly write or transmit data to the serial port. Pass☐ Fail☐	
All of these data and values should be measured accurately, and measured data should be conformed to the present environmental status (no odd data). Pass☐ Fail☐	
After the contents are sent and check if the transferred data is successfully transferred to the paired mobile application. Pass☐ Fail☐	

Table B.1: Hardware Test Plan

Software Tests	
1 - Image-to-text recognizer	Comments:

Scan 50 different clearly presented receipts. misrecognized words should be no more than 5: Pass ☐ Fail ☐	
Take the image of a partial word. The spell checker is able to correct it: Pass ☐ Fail ☐	
2 - GUI operations	Comments:
Navigating the UI must have no delay or jittering action. The application should not crash: Pass ☐ Fail ☐	
Able to display updated fruit/vegetable database information when selecting "View My List" option: Pass ☐ Fail ☐	
Able to display concentration values on user request Pass ☐ Fail ☐	
3 - DataBase Operations	Comments:
Insertion of new items into the database is successful: Pass ☐ Fail ☐	
Deletion of item from the database is successful: Pass ☐ Fail ☐	
4 - Sensor Data	Comments:
Sensor data can be read by the application without loss: Pass ☐ Fail ☐	

The sensor data checksum byte must equate the calculated checksum byte: Pass ☐ Fail ☐	
Sensor data sent to the application should be fast and within 1 second upon request: Pass ☐ Fail ☐	
5 - Expiry Duration	Comments:
Able to pull up-to-date produce info from EatByDate: Pass ☐ Fail ☐	

Table B.2: Software Test Plan