ECE 24

DEVELOPMENT OF A SMART REFRIGERATOR

Yelnur Abilakim
Oluwasijibomi Akinkugbe
Wyatt Brisbane
Keidi Xhagolli

FACULTY ADVISOR
Prof. Jaudelice de Oliveira

ECE 493

May 27, 2022

Smart Devices have been making huge strides in the world today. From Amazon's Alexa devices to Apple's Home Dot and even Google's Lifx Mini Smart Bulb, smart devices are making the lives of individuals and businesses easier and more convenient. Smart Refrigerators are the next generation of smart devices.

There are smart refrigerators on the market, however their smart features usually provide help with cooking or maintenance. Almost none of them target the issue of food waste directly. Implementing features that can notify the owner of expired or close to expiring food can help fill a niche that no fridge manufacturer has yet filled.

We developed a smart refrigerator device that can help combat food waste by allowing the user to always have a correct idea of the contents and its respective state in their refrigerator and make choices that will help reduce the amount of wasted food. Our solution includes an (a) application, (b) touchscreen, and (c) barcode reader. The touchscreen is mounted on the refrigerator to allow the user to manually input and view refrigerator contents without opening it. The barcode reader is used to automatically add food items to the smart refrigerator database to accurately predict the expiration date. The application is used to send notifications about the approaching end date of food items and additionally displays the refrigerator's contents to the user.

Table of Contents

List of Figures

List of Tables

1 Quad Chart

1.1 Quad Chart

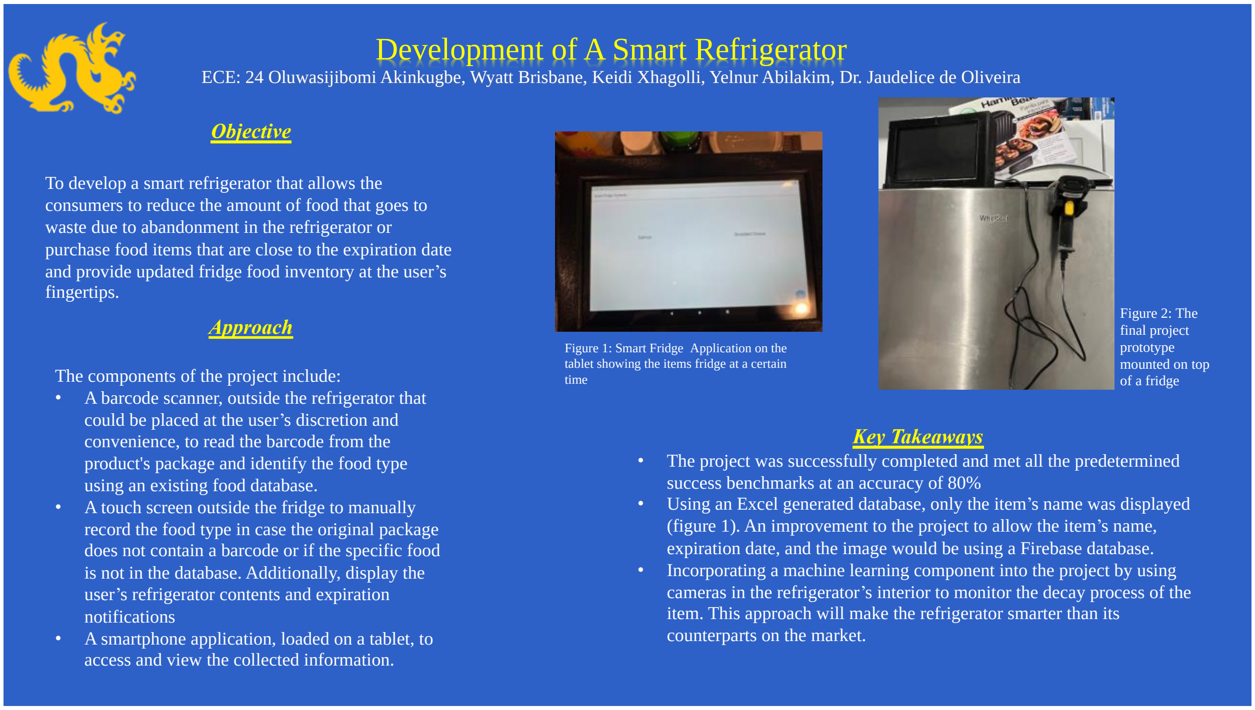


Figure 1: Quad chart detailing the summary of the project.

2 Introduction

2.1 Motivation

Food wastage is a problem that everyone has run into, especially college students, and it results in loss of both time and money. We are interested in working on this problem so we can help college students reduce the amount of food they waste and help them save money and time when budgeting for the desired period (monthly, weekly, bimonthly, etc.). Looking at the bigger picture, the food waste issue has major impacts on various aspects of life, with the most prominent ones on the Earth's environment. Environmental impacts of food waste include, but are not limited to, soil damage, increased trash production, less available land for wildlife, and increasing ocean dead zones, which lead to massive death of aquatic organisms[4]. With our work, we will get a step closer to putting a stop to these destructive consequences of food waste.

2.2 Problem Statement

To develop a smart refrigerator that allows the consumers to reduce the amount of food that goes to waste due to abandonment in the refrigerator and purchase of food items that are close to the expiration date and provide updated fridge food inventory at user's fingertips.

2.3 Stakeholder's Needs

Studies have shown that America wastes about 40% of the food it consumes[1], and billions of pounds of food wasted every day are due to overbuying, poor planning, and simply forgetting when a food item was bought in the store and placed in the fridge. We aimed to reduce this amount of food waste by developing a smart refrigerator and application to help people keep track of what and when they buy groceries.

2.4 State of the Art

The first emergence of a smart refrigerator was introduced by Lucky- Goldstar (LG) in 2000. These smart refrigerators are connected to the internet and can be synchronized with other smart home device applications like Alexa to look up recipes based on their refrigerator contents and create grocery lists, allow users to remotely control the refrigerator's temperature and send alerts to users' smartphones when its door has been left open for a long period of time.

Despite these amazing capabilities, most of the smart refrigerators still do not have the popularity that most smart home appliances like Amazon's Ring Doorbell, or Apple's Home Dot due to its limited functionality. To increase the popularity regarding these smart home refrigerators, most technology companies are incorporating new advanced technologies like cameras and artificial intelligence. Most of Samsung's smart refrigerators have a touchscreen that performs like a tablet with the same capabilities of an Amazon Alexa with the ability to play music, control temperatures and alert consumers of recipes based on current refrigerator contents. On the other hand, Lucky-Goldstar's current smart refrigerators only inform consumers when the filter of the refrigerator needs to be changed.

Our proposed smart refrigerator will have more capabilities and will be better suited to solve the food waste problem that other smart refrigerators have not been able to solve. The final protype of the project will be able to identify the food item once scanned with the barcode scanner, check the predicted expiration date of the item according to the database and send notifications when the food item is approaching its predicted expiration date. It will also be able to give the user an exact description of the current contents of his/her refrigerator to prevent overstocking on items that are already present in the refrigerator.

2.5 Approach

We aim to develop a smart fridge with the following components:

- A barcode scanner outside the fridge to read the barcode from the product's package and identify the food type using an existing food database such as MyFitnessPal, for example.
- A touch screen outside the fridge to manually record the food type in case the original package does not contain a barcode or if the specific food is not in the database.

- A smartphone application to access and view the collected information.

The main constraint is the size of the fridge and limited available space inside it. We first wanted to integrate a set of scales and possibly cameras into the design. To make the project more affordable and accessible, it was decided that they would be cut from the project. This allows for our product to work with any fridge without any major modifications to the fridge itself.

3 Design Specifications and Results

3.1 Specifications, Constraints and Standards

There are only a few self-imposed specific constraints and standards that we followed for this project's hardware. Since there are no comparable products on the market currently, there are no common specifications we had to follow due to market pressures. Our primary specifications for this design are portability and compatibility with existing fridges. This is considered essential to the project because we want it to be accessible to low-income families and students who might not have enough funds to purchase a whole new refrigerator just for this feature. We need not worry about the Raspberry Pi transmit frequencies or output power as it is certified by the FCC to meet those standards [12].

On the software side, there are a few standards we must follow such as having a clean, intuitive application that can be easily used by anyone, and our system must be able to recognize at least the most common food items and reliably guess their expiration date. Also, our software must be compliant with common communication standards such as the ones used for Wi-Fi[11] and internet data transfers to transfer information from the Pi directly to the user's smartphone. We will not collect the user's personal information so there is no need to comply with privacy of data standards. We are currently working on satisfying these standards.

There are little ethical concerns regarding this project, the worst thing that could happen is someone taking a bite out of a food item that has slightly gone past its expiration date. As far as environmental impacts are concerned, the hardware will eventually fail and turn into e-waste, but we believe that all the food items that will be saved from becoming garbage will severely outweigh the small amount of e-waste that will be produced at the end of its lifecycle. The only significant concern with the product is its cyber security – one may "hack" a user's smart device (phone or tablet) and gain access to their fridge's contents. It can be solved, however, by encrypting the connection between the app and the tablet.

3.2 Concepts

To follow these standards, the concept we decided to focus on was a small portable box to fit on the side or top of the fridge. This allows for the device to be utilized on any number of refrigerators, not just the one we use for our project. The device itself will be mounted onto the fridge using strong neodymium magnets. The device should be able to scan food in a way that it can accurately tell what it is to estimate its expiration - this could be done with a camera or a barcode scanner. As for software, the device should compare input items to those in a large database of common items, to minimize the amount of manual input needed.

3.3 Detailed Design

3.3.1 Overall Design

3.3.1.1 Description

We started with the question of how we can reduce overall food waste efficiently. Since most businesses have some framework in place to minimize food waste, even though they are not perfect, we thought we would make more of an impact if we helped consumers directly. Since we all had run into the issue of forgetting food in the fridge, it seemed like the best opportunity for improvement. From there we thought of diverse ways to notify the consumer of food that is close to its expiration date.

The original idea involved a camera mounted on the door of the fridge pointed towards the fridge's cavity. We would use machine learning to identify the food in the fridge and using a database, it would get an estimate of the expiration date. That was a good start, but after some brainstorming, we realized that, not only would the camera have trouble with items behind other items, but it would also need a very precise learning algorithm. So, we decided on using a barcode scanner on top of the fridge, which the user can use to simply scan their food before they put it in the fridge. As far as the notification part, we were debating between developing an application that receives the data from the sensor and lists it on your phone. Or we would have the sensor connected to an Arduino which would simply send a text message to your phone whenever something was about to go bad. We finally decided on the former but using Raspberry Pi rather than Arduino because of its superior power at comparable price and size.

Delving further into the concept we realized that adding an interactive interface would be necessary, because not everyone has a smartphone to begin with, and even if they do, they might want to enter a food item manually into the system without the hassle of having to go through their phone every time. It also offered the advantage of showing the foods that are closest to their expiration date directly on the refrigerator, so the consumer is more aware of their near expired food, and they choose to consume that first. Also, integrating scales into the Raspberry Pi would help the user automatically keep track of certain foods that you cannot reliably count, such as grapes or cheese. Integrating them directly into the fridge's shelves would be ideal, but that would require the building of a refrigerator from the ground up. So instead, to save on the cost and complexity of the final product, it was decided that the scales would be removed from the project entirely.

The exact design includes three major physical components, a Raspberry Pi microcontroller, a barcode scanner module, and a touch sensitive screen. The Raspberry Pi will act as the brains of the system, taking in and processing the information input into it from both the barcode scanner module, the touch sensitive screen, and the digital scales. The Raspberry Pi and screen are inside a custom-designed housing to keep them together. Attached to this is the barcode scanner, which has a magnetic hook to hold it onto the fridge when not in use. There will also be several software components to the system, which include a data analyzer, an interactive interface, and a phone application. The data analyzer will receive the barcode data and cross-reference it with a database that contains data, such as the name of the food as well as rough estimates of expiration dates, for most common foods. The interactive interface will handle both the output and input of the touchscreen. It will show what is currently inside the fridge and at what quantities, it will also allow the user to reduce the amount of each item in the fridge and manually input an item if it does not have a barcode or it so happens not to be in the database. The phone application will allow the user to do exactly what the tablet does, but in a more convenient way since the user can use the phone on the go.

3.3.1.2 Results

The system works as intended – the barcode scanner receives the product name from the barcode, the program finds the corresponding expiration date in the database, and the result is shown to the user on the tablet screen. The system identified the food items correctly 80% of the time, which meets the benchmark the group has set out as measures of success. The 3D-printed physical parts were sturdy and durable enough to safely hold the tablet screen and the barcode scanner. The power cords and jumper wires were neatly organized to make the product more visually appealing. The codes, drawings, and physical prototypes are included in Appendices.

4 Discussion

4.1 Hardware

Once all the hardware for the project was acquired, a housing to contain all of it for the final product needed to be designed. Each piece was measured and had all dimensions noted to allow an accurate 3D model of the tablet screen and Raspberry Pi to be made in Fusion360. With this model, a housing could be designed to hold the screen, while still allowing room for any cables that may attach underneath. The goal was to be able to utilize a team member's own 3D printer, a Creality Ender 3v2, to produce the model, which gave the group more freedom to prototype and try new designs when earlier ones failed. These early failures were due to faults in the design where parts were made too thin for the printer to create. This was solved through slight design modifications to the model in Fusion360. The overall design of the casing included a lower base, which supports the screen and contains all the internal components, and a cover, which magnetically snaps to the base to hold the screen down. An issue that was soon discovered was that the modeled parts would be too large to fit in the print space of the 3D printer. The solution for this was to split each part in half and design small pins to lock each part together. This had the benefit of decreasing print cost in the event of a print failure as well as fitting each piece comfortably within the print space. A holder for the barcode scanner was also designed, allowing for the user to easily mount the handheld barcode scanner to the side of the refrigerator magnetically. After all parts were printed and assembled using the custom pins, a test fit with the touch screen and other hardware was done with no issues.

4.2 Raspberry Pi Controller

The Raspberry Pi controller needed to be set up with a proper environment to implement the database and the application. At first, a Windows 10 environment was thought to be the most appropriate by the team due to its familiarity and built-in features. Once Windows 10 was running on the Pi, it proved less than optimal because the loading and response times were unacceptable. From there, a team discussion led to the possibility of loading a Linux based system such as Ubuntu onto the Pi due to its commonality and programming ease. Further research into the matter raised the possibility of using Android 10, another Linux based system, as our operating system. Android 10 was chosen because an Android compatible application would be developed anyway, so having the Pi run Android as well makes the application development a little easier overall. Once Android 10 was loaded, the system proved to be quite responsive, and it seemed to be running well on the Pi. At this point, the system's compatibility with a touchscreen and a barcode scanner was tested. Luckily for us, the touchscreen and the barcode scanner were compatible with the system right out of the box. Further testing was conducted to make sure all the hardware was fully compatible, and all the built-in software was stable.

4.3 Database

To estimate how long each product the user places into the fridge should last, the system must have access to the information about each food item's expiration date. As such, a database containing shelf lives of various products was acquired [10]. It originally came in a PDF format but was transformed into an Excel table because the latter is easier to parse. The table currently features approximate shelf-life ranges for over 180 most common food items found in an average household. The mean of the lowest and highest values of the range will be used to provide the user an expected period the food item can be safely stored in the refrigerator before consumption.

4.4 Application Development

Before the start of the winter term, the flutter was the primary application software that the team planned to use to develop the app for the project. Due to prior knowledge and experience with XCode to develop iOS applications, the software team decided to begin with the iOS application development and then later begin the Android development using Android Studio to mimic the iOS application. Problems arose during Android application development as the two cross-platform application development software had different functionalities and syntax, thus making it harder to create the same application on two different operating systems. Despite this delay, the dart script used for the program was completed and tested on both Android and iOS operating systems by week 7 of the Spring Quarter. During the project integration, however, there were two main issues that arose.

The first issue was the tablet failed to connect to the MacBook Pro which contained the DART script for the application. The team tried various ways of connections including USB and Wireless through Wi-Fi connection. Both connection methods failed because Visual Studio and Android Studio both required an Android 11+ device to establish this connection. This issue was later resolved by downloading the project as an executable Android Package Kit (.apk) file, loading this file into a portable USB device and then connecting this device to the Raspberry Pi microcontroller. Doing this allowed the final prototype to be cleaner as it required less wires to open the application.

After the application was successfully launched on the tablet, it was discovered that the barcode scanner was not recognized as the input for the in-app barcode scanner. This was because the module used in the script only recognizes inputs from a camera source to read the item's barcode. This issue was fixed modifying the Dart script so that that barcode scanner reads from a keyboard input instead of the initial barcode scanning module. After the modifications were finalized, the application development was complete as the application was able to read from the barcode scanner and display the name on the screen.

4.5 Future Work and Improvements

Given more time the following improvements would be made to the project:

- Using a Firebase generated database to allow the item's name, expiration date, and the image to be displayed on the application as using the Excel generated database, only the item's name was displayed.
- Incorporating a machine learning component into the project by using cameras in the refrigerator's interior to monitor the decay process of the item. This approach will make the refrigerator smarter than its counterparts on the market.

- Including odor or humidity sensors which could monitor the overall health of the items in the refrigerator, that does have some drawbacks such as non-specific information and the reliability of the sensor is questionable.
- Placing scales inside the refrigerator that would be put inside the fridge to keep track of the amount of certain foods. This will allow the user to label the scale to the food it would measure, and it would inform the user of how much of the food is left. For example, the user could label the scale as grapes, and it would keep track of how many grapes are left on that scale automatically.

5 Impact Statement

5.1 Environmental Impact

The main environmental impacts of food waste are larger ocean dead zones, increased pest populations, increased trash production, soil damage, and less available land for wildlife [4]. We believe that the overwhelming decrease of food waste will thoroughly decrease worldwide waste. While the manufacturing and the disposal of the equipment will cause some waste to be produced, the overall impact of this project will be a net benefit to the environment and the world at large.

5.2 Economic Impact

According to USDA's Economic Research Service, around $161 billion worth of food was wasted in 2010 in the United States [3]. This money could have been used for other purposes. Estimates show that if money were invested into cost-effective solutions to reduce food waste by 20% by 2030 would generate a net financial benefit of $73 billion [9]. We believe our project could be satisfy as one of these solutions because it would be marketed at an affordable price to the user.

5.3 Social Impact

According to USDA's Economic Research Service, around 133 billion pounds of food were wasted in 2010 in the United States [3]. This issue was made worse due to the COVID-19 Pandemic, which increased farm food loss and food business closures[9]. This food could have been distributed to people in need.

6 Materials and Methods

6.1 Software

- ❖ Python used for a program that can correctly identify the item scanned through the data and display the item on the touchscreen
- ❖ Fusion 360 for hardware case
- ❖ Flutter for App development on both android and iOS devices

- ❖ Microsoft Excel database for identifying product information key to predicting food expiration date

6.2 Hardware

- ❖ Raspberry Pi microcontroller for controlling the barcode scanner and weight scales
- ❖ Arduino kit for incorporation with the scales
- ❖ Touch screen for manually recording food type and amount
- ❖ Barcode scanner module compatible with the Raspberry Pi
- ❖ Digital scales for measuring amounts of certain foods
- ❖ Fridge for prototyping and testing
- ❖ Miscellaneous wires and connectors

7 Budget

The table below shows the project's updated budget.

	ECE 24: Development of A Smart Fridge Project Proposal				
	Equipment	**Description**	**Quantity**	**Base Price ($)**	**Amount ($)**
1	Barcode Scanner	The barcode scanner will be used to scan the food item and cross reference the item with the database to correctly identify the item and its expiration date.	1	$13.99	$13.99
2	Raspberry Pi Microcontroller and Touch Screen	The Raspberry Pi Microcontroller will be used to interface with the software program and barcode scanner.	1	$135.99	$135.99
3	3D Printer Filament	PLA filament used to build the housing along with any prototypes.	1	$44.99	$44.99
				Total	**$194.97**

Table 1: A detailed table showing the equipment and its corresponding description that made up the budget for the project

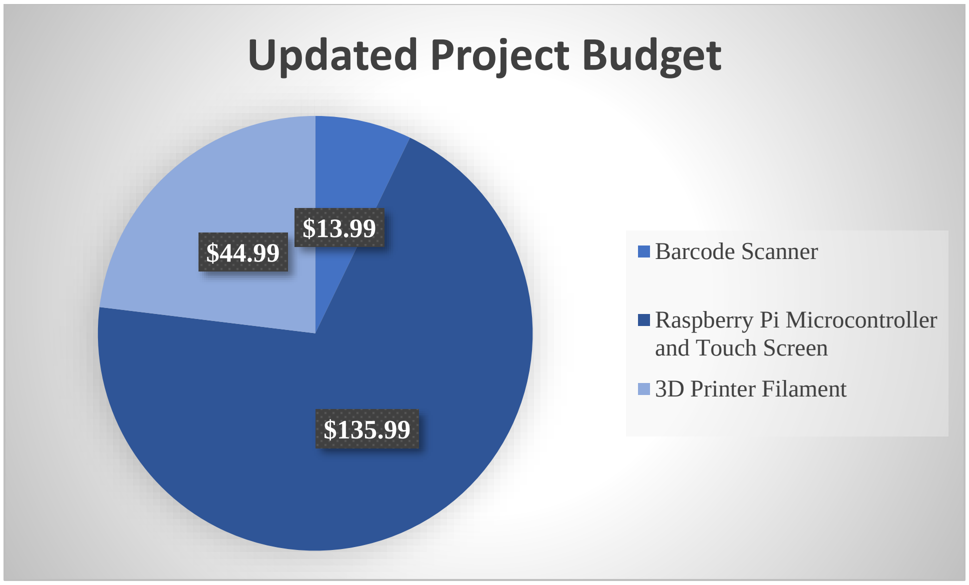
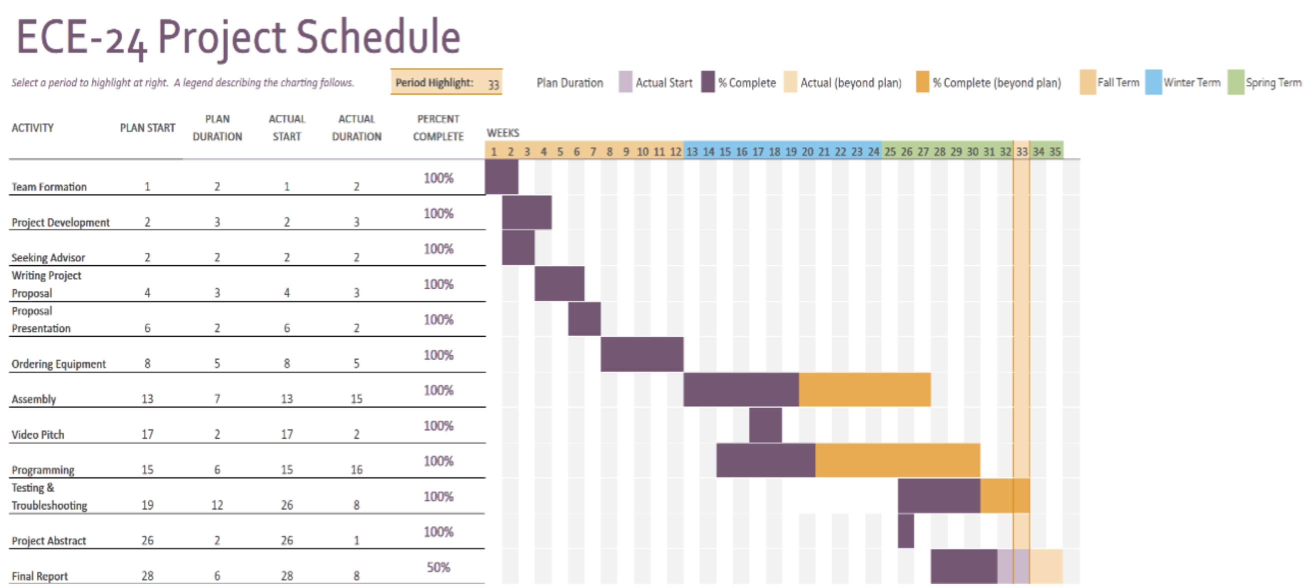


Figure 2: A detailed pie chart showing the percentage of equipment on the total budget.

8 Project Management

8.1 Schedule

The planned schedule for this project was based off various due dates along with group estimates on how long each task should take. This plan is visualized in a Gantt chart, shown below in Figure 3.



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Team Formation	1	2	1	2	100%
Project Development	2	3	2	3	100%
Seeking Advisor	2	2	2	2	100%
Writing Project Proposal	4	3	4	3	100%
Proposal Presentation	6	2	6	2	100%
Ordering Equipment	8	5	8	5	100%
Assembly	13	7	13	15	100%
Video Pitch	17	2	17	2	100%
Programming	15	6	15	16	100%
Testing & Troubleshooting	19	12	26	8	100%
Project Abstract	26	2	26	1	100%
Final Report	28	6	28	8	50%

Figure 3:Gantt chart showing the planned schedule for the project, including information for how long each project. The selected period shows the current point in the timeline.

8.2 Team Member Responsibility

Our team comprised of four students at the College of Engineering specifically the Electrical and Computer Engineering department. Each member's responsibility remained consistent throughout the quarters of senior design. A detailed description is shown below:

Team Member's Name	***Responsibility***	***Description***
Yelnur Abilakim	Data Analysis	Created a Python script for data analysis which matched the barcode scanned item to the database through the barcode text displayed, and found the database used to display an item's name rather than barcode text on the application.
Oluwasijibomi Akinkugbe	Application Development	Developed the application where notifications were sent to the user at appropriate times and barcode scanned items were displayed
Wyatt Brisbane	Hardware and Circuit design	Designed and printed the housing for microcontroller and touchscreen.
Keidi Xhagolli	Microcontroller programming	Programmed the Raspberry Pi microcontroller, touchscreen to read items from the barcode scanner and display the items on the mounted touchscreen.

Table 2: A detailed showing each member's responsibility for the project

9 References

[1] Food Waste FAQs. *U.S. Department of Agriculture* [Online]. https://www.usda.gov/foodwaste/faqs (Accessed October 21, 2021).

[2] Buzby, Jean C., Hodan F. Wells, and Jeffrey Hyman. The Estimated Amount, Value, and Calories of Postharvest Food Losses at the Retail and Consumer Levels in the United States, EIB-121. *U.S. Department of Agriculture*, Economic Research Service, February 2014.

[3] Food Loss and Waste. *U.S. Food and Drugs Administration* [Online]. https://www.fda.gov/food/consumers/food-loss-and-waste (Accessed October 28, 2021).

[4] Ovidiu Sandru. The Environmental Impact of Food Waste. *The Green Optimistic* [Online], February 11, 2021. https://www.greenoptimistic.com/the-environmental-impact-of-food-waste-20210211/#:~:text=The%20Environmental%20Impact%20of%20Food%20Waste%201%20Larger,Damage.%20...%205%20Less%20Land%20for%20Wildlife.%20 (Accessed October 28, 2021).

[5] Kore, Utkarsh and Akre, Parth and Mashayak, Usama, Smart Refrigerator (May 18, 2020). 2nd International Conference on Communication & Information Processing (ICCIP) 2020, Available at SSRN: https://ssrn.com/abstract=3647969 or http://dx.doi.org/10.2139/ssrn.3647969

[6] Mukesh P. Mahajan, Rohit R. Nikam, Vivek P. Patil, and Rahul D. Dond "Smart Refrigerator Using IOT "Vol. 2, No. 3, PP – 86- 91, March – 2017

[7] Nikhil Kakade, and Prof. (Dr.) S. D. Lokhande, "IoT based Intelligent home using Smart Devices", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, No. 6, June 2016.

[8] Ahmed, M., & Rajesh, R. (2019). Implementation of Smart Refrigerator based on Internet of Things. International Journal of Innovative Technology and Exploring Engineering, 9(2), 3419-3422. doi: 10.35940/ijitee.b6343.129219

[9] Mukhebi, V., Costantino, J. and Downey, P., 2021. 2020 Annual report. [online] Refed.org. Available at: <https://refed.org/downloads/refed-2020-annualreport-final-optimized.pdf>.

[10] The Food Keeper: A Consumer Guide to Food Quality & Safe Handling. Food Marketing Institute with Cornell University Institute of Food Science Cornell Cooperative Extension [Online], December 2012. foodkeeperinside4stripes (ncsu.edu) (Accessed January 24, 2022)

[11] "IEEE 802.11-2020," IEEE Standards Association, 26-Feb-2021. [Online]. Available: https://standards.ieee.org/ieee/802.11/7028/. (Accessed: 27-May-2022).

[12] "FCC ID 2ABCB-RPI32," FCC ID. [Online]. Available: https://fccid.io/2ABCB-RPI32. (Accessed: 27-May-2022).

10 Appendices

10.1 Appendix A – tablet housing and its Fusion360 design


Figure 4. 3D-Printed Tablet Housing (left) and Fusion360 tablet housing design (right)

10.2 Appendix B – barcode scanner holder and its Fusion360 design

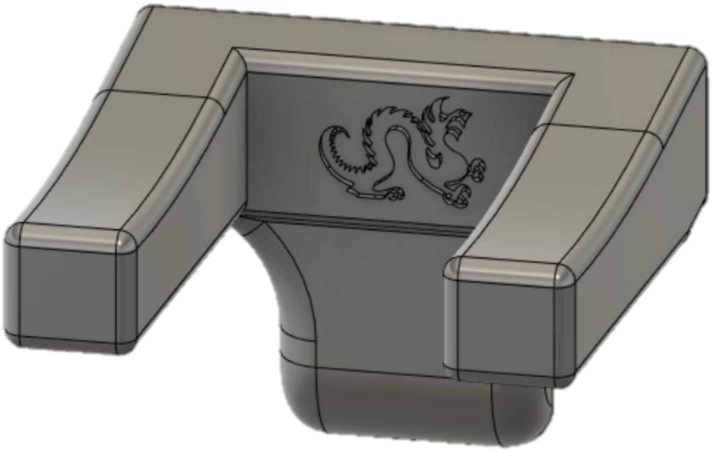



Figure 5: 3D-Printed Barcode Scanner Holder (left) and Fusion360 barcode scanner holder design (right)

10.3 Appendix C – Excel database excerpt (to see the full database, refer to Appendix F)

	A	B	C	D	E	F	G
1	product	opened_min	opened_max	unopened_min	unopened_max	barcode	pic
27	butter	30	90			688267074714	
28	buttermilk	7	14				
29	cabbage	7	14				
30	canned goods (high acid)	5	7				
31	canned goods (low acid)	3	4				
32	carrots	21	21				
33	cauliflower	3	5				
34	caviar	2	2	7	28		
35	celery	7	14				
36	cheddar	21	28	180	180		
37	cheesecake	7	7				
38	cherries	1	2				
39	chicken	1	2			282004009948	
40	chile pepper	4	5				
41	chitterlings	1	2				
42	chocolate syrup	180	180				
43	chutney	30	60				
44	cilantro	7	7				
45	clams	1	2				
46	coconuts	14	21				
47	cod	1	2				

Figure 6: Cropped section of the Microsoft Excel database used for Data Analysis

10.4 Appendix D – Python code for data analysis

```
In [1]:

import xlrd
import difflib
fpath=([redacted])
workbook=xlrd.open_workbook(fpath)
keeper=workbook.sheet_by_index(0)
products=[]
dates_min=[]
dates_max=[]
for i in range(1,keeper.nrows):
    products.append(keeper.cell_value(i,0))
    dates_min.append(keeper.cell_value(i,1))
    dates_max.append(keeper.cell_value(i,2))

In [2]:

def matches(word):
    return difflib.get_close_matches(word,products,n=3,cutoff=.5)
user_word="chicken"
matches_list=matches(user_word)
#print(matches_list)

['chicken', 'ground chicken', 'cheesecake']

In [3]:

the_match=matches_list[0]
#print(the_match)

chicken

In [4]:

idx=products.index(the_match)
#print(idx)

28

In [6]:

exp_min,exp_max=dates_min[idx],dates_max[idx]
exp_avg=(exp_min+exp_max)/2
#print(exp_avg)

1.5
```

Figure 7: Cropped section of Python script used for identifying the item's name and expiration date from the barcode scanned text

10.5 Appendix E – Code for Flutter Development

```dart
import 'package:flutter/material.dart';
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:firebase_core/firebase_core.dart';

//import 'firebase_options.dart';

Run | Debug | Profile
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(

      //options: DefaultFirebaseOptions.currentPlatform,
      );

  runApp(const MyApp());
}

class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  var itemController = TextEditingController();
  var _scanResult = '';
  var itemList = [];
  @override
  void initState() {
    super.initState();
    final fbm = FirebaseMessaging.instance;
    fbm.requestPermission();
  }

  Widget build(BuildContext context) {
    const title = 'Smart Fridge Contents';

    return MaterialApp(
      title: title,
      theme: ThemeData(
        appBarTheme: const AppBarTheme(
          backgroundColor: Colors.white,
          foregroundColor: Colors.black,
        ), // AppBarTheme
      ), // ThemeData
```

```dart
        appBarTheme: const AppBarTheme(
          backgroundColor: Colors.white,
          foregroundColor: Colors.black,
        ), // AppBarTheme
      ), // ThemeData
      home: Scaffold(
        appBar: AppBar(
          title: const Text(title),
        ), // AppBar
        body: Center(
          child: GridView.count(
            // Create a grid with 2 columns. If you change the scrollDirection to
            // horizontal, this produces 2 rows.
            crossAxisCount: 2,
            // Generate 100 widgets that display their index in the List.
            children: List.generate(itemList.length, (index) {
              return Center(
                child: Text(
                  itemList[index],
                  style: Theme.of(context).textTheme.headline5,
                ), // Text
              ); // Center
            }), // List.generate
          ), // GridView.count
        ), // Center
        floatingActionButton: Builder(
            builder: (context) => FloatingActionButton(
                  onPressed: () {
                    showModalBottomSheet(
                      isScrollControlled: true,
                      context: context,
                      builder: (builder) {
                        return StatefulBuilder(
                            builder: (BuildContext context, setState) {
                          return SizedBox(
                              height: MediaQuery.of(context).size.height / 2,
                              child: Scaffold(
                                appBar: _buildBarcodeScannerAppBar(),
                                body: _buildBarcodeScannerBody(),
                              )); // Scaffold // SizedBox
                        }); // StatefulBuilder
                      },
                    );
                  },
                  tooltip: 'Scan Barcode',
                  backgroundColor: Colors.blue,
```

```dart

  _showBarcodeScanner() {
    return showModalBottomSheet(
      isScrollControlled: true,
      context: context,
      builder: (builder) {
        return StatefulBuilder(builder: (BuildContext context, setState) {
          return SizedBox(
              height: MediaQuery.of(context).size.height / 2,
              child: Scaffold(
                appBar: _buildBarcodeScannerAppBar(),
                body: _buildBarcodeScannerBody(),
              )); // Scaffold // SizedBox
        }); // StatefulBuilder
      },
    );
  }

  AppBar _buildBarcodeScannerAppBar() {
    return AppBar(
      bottom: PreferredSize(
        child: Container(color: Colors.blueAccent, height: 4.0),
        preferredSize: const Size.fromHeight(4.0),
      ), // PreferredSize
      title: const Text('Scan Your Barcode'),
      elevation: 0.0,
      backgroundColor: const Color(0xFF333333),
      leading: GestureDetector(
        onTap: () => Navigator.of(context).pop(),
        child: const Center(
            child: Icon(
          Icons.cancel,
          color: Colors.white,
        )), // Icon // Center
      ), // GestureDetector
      actions: [
        Container(
          alignment: Alignment.center,
          padding: const EdgeInsets.only(right: 16.0),
        ), // Container
      ],
```

```dart
  Widget _buildBarcodeScannerBody() {
    return SizedBox(
      height: 400,
      child: AlertDialog(
        content: Form(
          child: Column(
            children: [
              TextFormField(
                controller: itemController,
                decoration: InputDecoration(
                  labelText: 'Scan your item',
                  icon: Icon(Icons.scanner),
                ), // InputDecoration
              ), // TextFormField
            ],
          ), // Column
        ), // Form
        actions: [
          TextButton(
            onPressed: () {
              // Send them to your email maybe?
              var item = itemController.text;
              setState(() {
                _scanResult = item;
                itemList.add(_scanResult);
                //Navigator.of(context).pop();
              });
            },
            child: Text('Add'),
          ), // TextButton
        ],
      ), // AlertDialog
```

Figure 8: Screenshots from Visual Studio Code showing the Flutter (dart) script used to develop the application

10.6 Appendix F – Links

10.6.1 Video Presentation

https://youtu.be/nnHXSs-a6Nc

10.6.2 Video Pitch
https://youtu.be/pbAn8yog85s

10.6.3 LinkedIn Post
https://www.linkedin.com/posts/wyattbrisbane_for-my-teams-senior-final-project-we-created-activity-6936065415676661760-2I5k?utm_source=linkedin_share&utm_medium=member_desktop_web

10.6.4 Full Excel database
https://docs.google.com/spreadsheets/d/1V3DHmKiN2Xal4Y8ZU5NOEYglDAuMhmmQ/edit?usp=sharing&ouid=105983955909892693873&rtpof=true&sd=true