Fresh Food App

Prepared by Group 27:
Christian Gutierrez, Jordan Nguyen, Natalie Reyes, Wasay Ahmed

CS 440
University of Illinois Chicago
Spring 2022

Table of Contents

List of Figures

List of Tables

I Project Description

1 Project Overview

The Fresh Food application will let the user know if food has gone bad by analyzing the chemical components of the odor released by food. Some chemical odors include methane (greenhouse gas given by rotting vegetation) or Ammonia (rotting fish and meat). The chemical makeup of the odor is read by external odor sensors and the data is sent to the app which will calculate edibility and estimate days til inedible depending on the type of food. This will remove the guesswork of deciding whether food has gone bad.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Food supply and vendor businesses like grocery stores display to customers to pick. Most of the food sold have sell-by and expiration dates that indicate the grocery store's limit of putting the food on display before getting thrown away. This is one of the main sources of food waste because the dates on these labels are not accurate. Restaurants buy food in large quantities to keep their business running and some of that food gets thrown out to prevent risks of customers getting sick. Food transporters receive the food from their sources and distribute it to the suppliers and vendors. The conditions of the environment the food is stored in, such as temperature and humidity, affects the rates of how quickly food goes bad. This lowers further the accuracy of the original expiration dates. Therefore shipping companies need specialized trucks to control their storage environment to keep the food fresh.

2b Goals of the Project

The goal of the Fresh Food app is to save money for ordinary consumers, restaurants, food suppliers, and food transporters by accurately determining if food is safe to eat by using gas/odor sensors. Over 108 billion pounds of food is wasted every year in the US which amounts to about $218 billion dollars. The app can also save people from food borne illnesses and reduce the number of related hospitalizations and people taking time off work to recover. The app would also have an impact on the environment by reducing the greenhouse gasses released by rotting food which contributes to about 6% of overall greenhouse emissions.

2c Measurement

Sensors for the detection of orders and gasses emitted from the food will be the source of all measurements of this project. Measurements such as methane and ammonia, depending on the type of food being monitored will be used to determine if the food is still edible and used to estimate the days until the food is inedible. The sensors will send their measurements by Bluetooth or Wi-Fi for analysis. Other sensors such as temperature and humidity sensors can also be factored in for the estimations.

3 The Scope of the Work

The work done by this app is to make accurate predictions of how long the food will be edible by analyzing the gas and odors emitted by the food or notify that the food is inedible. To do this work, the user manually sets up the sensors and makes sure that the measurements

are being received successfully to the Fresh Food app. Then the user will see the estimates for the ediblity time left for the food.

3a The Current Situation

Currently there are no such widely used food freshness apps or devices. Most foods sold have either sell-by date or expiration date labels that are not entirely accurate. Food suppliers and vendors don't want to risk selling expired food and throw out everything past those dates. This is one of the main issues that contributes to the massive amounts of food being wasted.

3b The Context of the Work

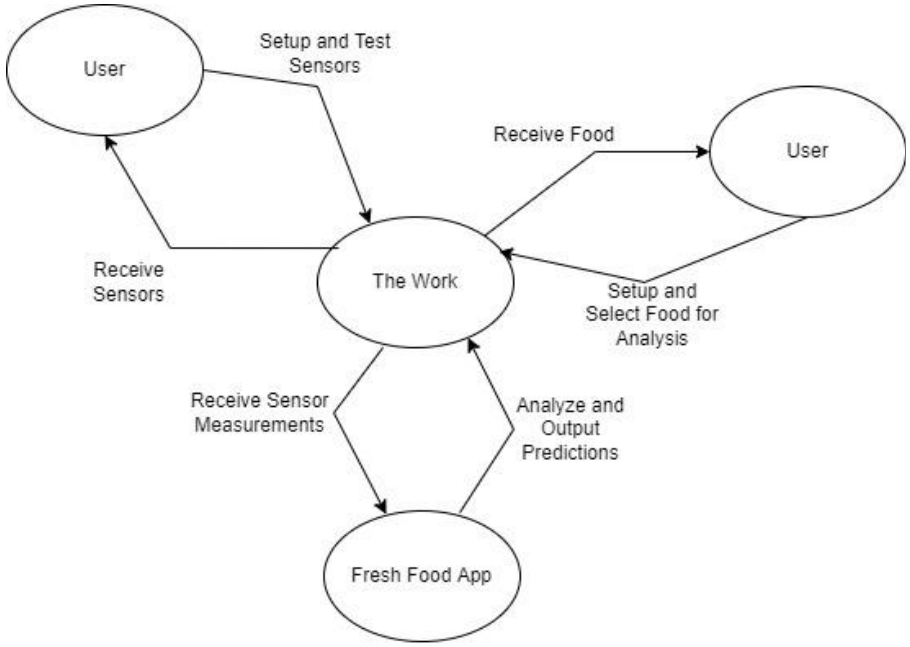

Figure 1. *Work diagram of Fresh Food App.*

3c Work Partitioning

Table 1. *Work Partitioning table of Fresh Food App.*

Event Name	Input/Output
User Sensor Setup	Manually place sensors correctly for food gas/odor detection (Input) Test connection and data transfer with sensor and app (Input)

User Food Setup	Place food near sensors (input) Select food type for accurate calculations (input)
Sensors Measurements and Data Analysis	Measure gas/odor from the food (input) Analyze measurements and display estimated time left for food freshness (output)

3d Competing Products

Currently there is only one commercial product called FOODsniffer which is a handheld device that determines the freshness of meats. This device is limited to only meats. Therefore, the Fresh Food app has the advantage of measuring other types of foods and can be used for continuous monitoring anywhere the sensors are properly set up.

4 The Scope of the Product

- **Consumers** - This could be anyone who eats produce, meat, or dairy products. The Fresh Foods App would allow the consumer to check their produce, meat, or dairy products for any harmful chemicals and get an indication of whether it is okay to consume that food item or not. The consumer will also be able to view safety and expiration information regarding the product. Lastly, the consumer will be able to rate or review the place they got the product from.
- **Buyers** - This could be customers who shop for produce, meat, or dairy products. This could also be individuals or businesses who are buying produce, meat, or dairy products for their business. The Fresh Foods App will allow the buyer to check their produce, meat, or dairy products for any harmful chemicals and get an indication of whether it is okay to buy that food item or not. The buyer will also be able to view safety and expiration information regarding the product. Lastly, the buyer will be able to rate or review the food products for the place or company that the product is at.
- **Suppliers/Vendors** - This could be farms, companies, restaurants, business owners, markets, stores, employees, or anyone who supplies or sells fresh produce, meat, or dairy products. Suppliers and vendors can use the Fresh Foods App to check their produce, meat, or dairy products for any harmful chemicals and get an indication of whether it is okay to supply or sell that food item or not. The supplier and vendor will also be able to view safety and expiration information regarding the product. Lastly, the supplier or vendor can receive any ratings or reviews they have received from their shoppers, buyers, or consumers.
- **Shippers/Transporters** - This could be drivers, delivery employees, delivery companies, or anyone who transports goods from the supplier to their destination. This can also be the supplier if the supplier chooses to deliver their own products. Shippers/transporters can check to ensure the products did not go rotten during their time of being transported

to their destination. The shipper/transport can communicate or inform the supplier and vendor of how the supply is doing.

- **System** - The system for the Fresh Foods App can process results from testing the food. The system can also use the internet (Google) to help present information regarding the results, expiration, and safety to the user.

4a Scenario Diagram(s)

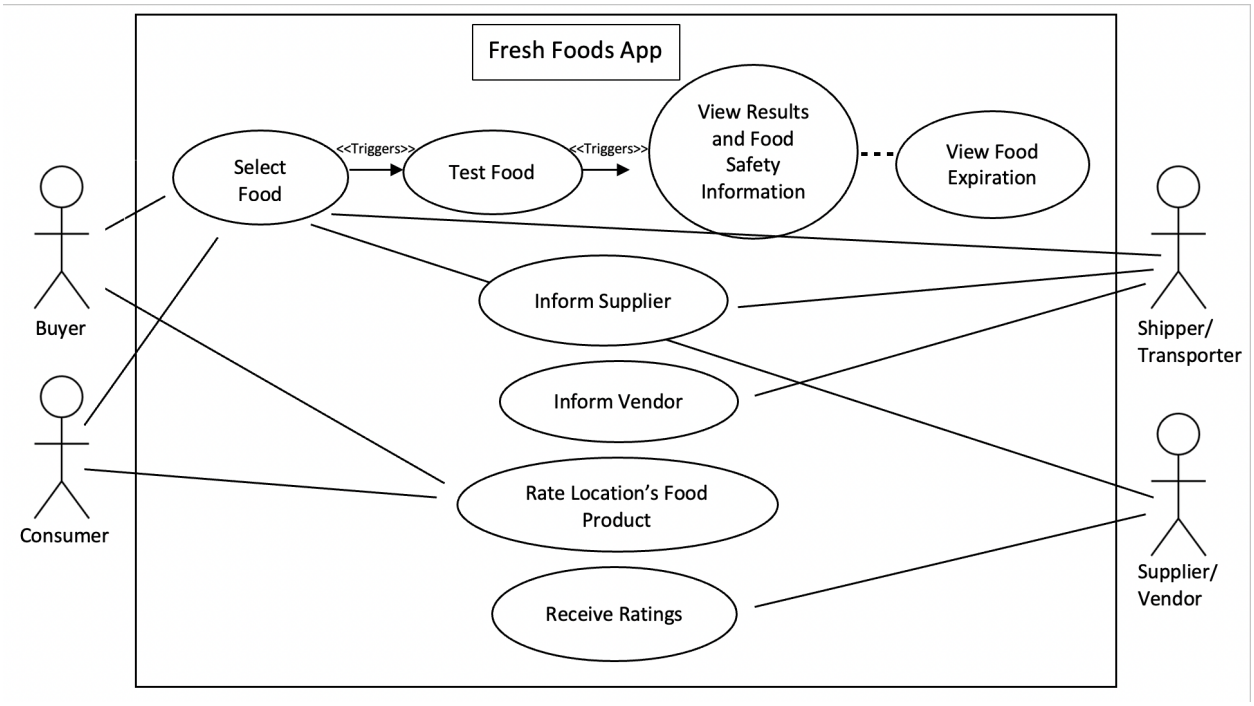

Figure 2. *Scenario Diagram of Fresh Food App.*

4b Product Scenario List

Table 2. *Product Scenario List of Fresh Food App.*

Scenario Name	External Actors
Select Food	A database of produce, meat, or dairy items can be used to identify the specific food item selected
Test Food	Sensor (separate hardware through bluetooth connection) will be used to collect data from food
View Results and Food Safety Information	Consumer, buyer, shipper/transporter, supplier/vendor can view results, and safety information regarding the food gathered from Google
View Food Expiration	Consumer, buyer, shipper/transporter,

	supplier/vendor can view the expiration date of the food determined by the amount of harmful chemicals detected by the sensor and expiration data on the specific food item from Google
Inform Supplier	Shipper/Transporter can separately communicate that the food product is rotten
Inform Vendor	Shipper/Transporter can separately communicate that the food product is rotten and can't be sold
Rate Location's Food Product	Consumer and buyer can separately rate or review the quality of the foods products from where they are seeing them or buying them from
Receive Ratings	Suppliers and vendors can receive ratings or reviews from people who have considered buying their food products or have bought them

4c Individual Product Scenarios

All users of the Fresh Foods App can test produce, meat, or dairy products to check whether they are okay to eat, view results, food safety information, and food expiration information. Buyers of produce, meat, or dairy products can use the Fresh Foods App to ensure that they are buying good quality and safe products. If they find a product is not safe to eat according to the sensor, then they can leave a negative rating or review to that vendor or supplier. If the buyer finds the products are safe and with good quality, they can leave a positive rating or review to that vendor or supplier. Consumers who are about to eat a food product can test the product before eating. This can be useful especially at restaurants to help prevent food poisoning. The consumer can leave a rating or review regarding the quality of the product from the location it's sold. Suppliers of food products can use the Fresh Foods App to ensure they are supplying good quality food products. Suppliers can also receive ratings from buyers or consumers to get further feedback on their products. Vendors can use the Fresh Foods App to ensure they are not selling harmful foods to anyone. They can receive ratings from consumers or buyers for further feedback on their products they sold. Shippers/Transporters can test the food products before and after shipping. They can inform the supplier and vendor if there is anything wrong with the food product.

5 Stakeholders

5a The Client

The clients of the Fresh Foods App will be the everyday consumer that shops for fresh produce, meat, and dairy products. The application will be used by the user to determine how

fresh the products they have purchased are in the comfort of their own home. Other clients of the application could include vendors or suppliers of the food that is to be sold such as grocery stores , as well as the producers of the food. to ensure quality products. Instead of having to rely on taste , smell, and how the products appear visually we can reduce the risk by using an attachment near or directly inside product containers to detect and monitor the presence of harmful gasses that can come from food that has expired.

5b The Customer

Users of this application will have the responsibility of picking out which product they would like to purchase as inputting the correct settings for said chosen product. The user will be able to choose from a variety of food types and categories from dairy, products, meat, fruits, and vegetables. The customer can learn and read what the detection results mean and whether or not the food is edible or what stage the food product is in.

5c Hands-On Users of the Product

The clients and customers will be the hands-on-users for the app. They will use the application to test individual food items and after the testing is complete a log will be made to indicate when the product will need to be discarded using a calendar. The pages will be defined by the UI designer so that customers can navigate through and complete the action of testing.

5d Maintenance Users and Service Technicians

Our application will be maintained by the developers who will be in charge of testing the performance of the app and ensuring it is accurately updated from time to time. Similar to the maintenance the developers are responsible for, engineers will perform tasks to the database to make sure it runs smoothly. Both will need to continue checking and updating the product for the whole duration of the product's life.

5e Other Stakeholders

Other stakeholders will include legal experts to make sure everything runs smoothly if anything occurs. Another stakeholder would be a marketing team who can assist with making the product attractive to potential customers and users. A financial advisor will also most likely be used to figure out how to manage the money made from the application and budget accordingly.

5f User Participation

User participation will occur once the app has been established and the functional aspects have been implemented.

An example would be once the database of various foods have been inputted and data can be accurately read from the gasses emitted then user testing can be done. Once these steps have been completed and the interface and database is ready to go then the user can give feedback on how the app is designed.

5g Priorities Assigned to Users

The key users will be the at home users. The people who purchase fresh produce on a regular basis. It's very important to establish a good relationship with our main users that way we can eventually transition to more mainstream use.
Other users can include the vendors who sell the items such as a grocery store. Once the application has had some marketing and the main users are satisfied then there will be a smoother transition from at home use to everyday use for stores to use.
Users that are not as important as the product producers. More than likely it will be difficult for a fast paced environment with producing so many products that they already have their own maintenance checks for bigger amounts of inventory.

6 Mandated Constraints

6a Solution Constraints

1.) The app will need to keep an updated database to be able to accept different kinds of products within the app. Each kind of food will have its own section and will be listed in alphabetical order.
The best way to sort out each kind of food is to alphabetize them in each of their respective categories to make them easily accessible.
After choosing the food item you can then test for gasses such as:

- Methane (greenhouse gas given by rotting vegetation)
- Hydrogen sulfide (emitted from dead plant and animal matter)
- Noxious solanine gas (produced by rotting potatoes, dangerous if inhaled in large amounts)
- Ammonia (rotting fish and meat)

2.) The Fresh Foods App will be available on android and iOS
Making the app downloadable by a wider variety of people who have different devices.
In order to be downloadable it will first need to be evaluated so it is available in the app store for both apple and android products.

6b Implementation Environment of the Current System

The app will be on the Android and iOS systems. The sensors can be connected to Arduinos or other microcontrollers that communicate with the app via Bluetooth and WiFi.

6c Partner or Collaborative Applications

- Database: access different foods for testing
- Cloud: access safety tips for food storage

6d Off-the-Shelf Software

- Sharing API
    - Connect and share data to google calendar or with others when getting gas detection results.
- Smartphone
    - Needed by the client in order to have access to the application.

- Login API
    - Will allow clients to easily login using gmail accounts.

6e Anticipated Workplace Environment

The product is developed to be easily accessible due to the versatility when it comes to users of the product. This application can be used at the clients home and even accessible when grocery shopping at the grocery store.

6f Schedule Constraints

In order to distribute the application we must have consumers who are able and ready to utilize it and hopefully implement it into their routine. Whether the routine be shopping for food at a grocery store, the grocery store employees maintaining and checking their own products, or the distributor of the food.

6g Budget Constraints

The product itself will be an application that will not require a large amount of funds besides changing and updating the database, maintaining the UI, and other expenses related to the development or infrastructure of the application.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Buyer: An individual, shopper, or business that is planning on purchasing or has purchased produce, meat, or dairy product(s).
Consumer: An individual who is about to eat or has eaten produce, meat, or dairy product(s).
Shipper/Transporter: An individual, or employee of a business who ships/transports produce, meat, or dairy products from the supplier to the intended destination.
Vendor: Farms, companies, restaurants, business owners, markets, stores, employees, or anyone who sells fresh produce, meat, or dairy products.
Supplier: Farms, companies, restaurants, business owners, markets, stores, employees, or anyone who supplies fresh produce, meat, or dairy products to a buyer.

7b UML and Other Notation Used in This Document

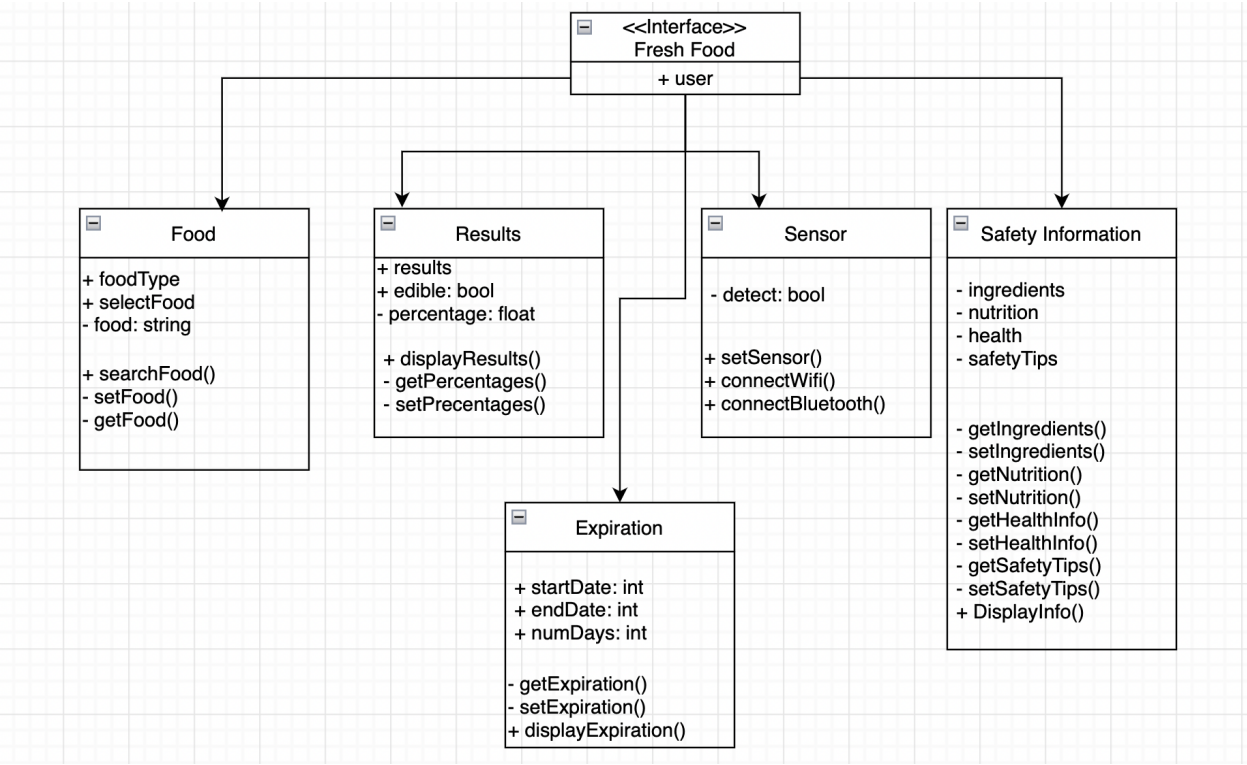

Figure 3. *UML diagram of Fresh Food App.*

7c Data Dictionary for Any Included Models

Select Food:

- Food type
- Specific food name
- Search
- Select

Test Food:

- Sensor detection
- Sensor connection to bluetooth
- Sensor connection to wifi
- Set up sensor

View Results:

- Percentages on gasses and chemicals emitted from food
- Mention whether safe to eat or not

View Safety Information:

- Nutrition information
- Ingredients
- Health information regarding food
- Safety tips on that particular food item

Food Expiration:

- Expiration date
- Number of days until safe to eat

Buyer:

- Identifier: company, business owner, customer, individual

Consumer:

- Identifier: customer, individual

Shipper/Transporter:

- Identifier: employee, company

Vendor:

- Identifier: seller, company, employee, owner, business

Supplier:

- Identifier: company, employee, individual, owner, business

8 Relevant Facts and Assumptions

8a Facts

Every year, an estimated 1 in 6 Americans (or 48 million people) get sick, 128,000 are hospitalized, and 3,000 die from foodborne diseases.
Each year, 108 billion pounds of food is wasted in the United States. That equates to 130 billion meals and more than $408 billion in food thrown away each year. EPA estimates $218 billion dollars wasted.
Food wastage is responsible for around 6% of total global greenhouse gas emissions.
The share of Americans that own a smartphone is now 85%(2021)
https://www.cdc.gov/foodsafety/food-poisoning.html
https://www.feedingamerica.org/our-work/our-approach/reduce-food-waste
https://ourworldindata.org/food-waste-emissions
https://www.pewresearch.org/internet/fact-sheet/mobile/
https://www.epa.gov/international-cooperation/international-efforts-wasted-food-recovery#:~:text=In%20the%20United%20States%2C%2040,emissions%20in%20the%20United%20States.

8b Assumptions

We assume that the users can obtain the odor sensors and set them up correctly to get accurate samples.
We assume that the amount of specific odors can correlate to edibility.
We assume that all food that rots will emit a traceable odor. If not, we can list specific foods for the user that cannot be checked by the app.
We assume odor sensors are affordable for any user.

II Requirements

1 Product Use Cases

1a Use Case Diagrams

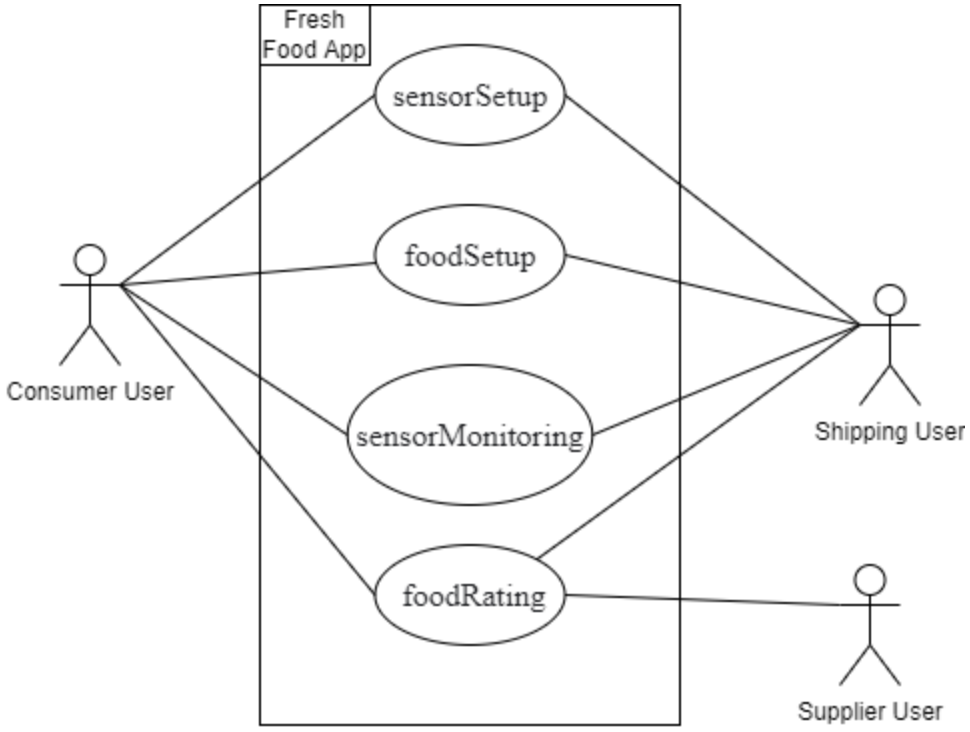

Figure 4. *Use Case diagram of Fresh Food App.*

1b Product Use Case List

Not applicable

1c Individual Product Use Cases

Use case ID: sensorSetup Name: Sensor Setup
pre-conditions: User is registered and has received the sensors with the instructions.
post-conditions: User is allowed to the next screen for food monitoring.
Initiated by: Any type of user.
Triggering Event: Pressing Sensor Setup button.
Additional Actors: None

Sequence of Events:

1. The app requires User to press the sensor setup button to begin setup.
2. User manually installs the sensors to the location where the food will be stored.
3. The app requires the User to test the connection of the sensors to the app with Bluetooth or Wi-Fi.
4. The app requires the User to calibrate the sensors for accurate readings.
5. User finishes the setup by pressing the Finish Setup button.

Alternatives: The app should allow the User to recalibrate and relocate the sensors.
Exceptions: The app prevents the User from advancing when failing to test the connection of the sensors.

Use case ID: foodSetup Name: Food Setup
pre-conditions: User has finished setting up the sensors.
post-conditions: User can view the data received and the app's calculations.
Initiated by: User
Triggering Event: Pressing the Food Setup button.
Additional Actors: None

Sequence of Events:

1. User starts food setup by pressing the Food Setup button.
2. User selects the type of food to be monitored.
3. User places the food at a specified distance from the sensors.
4. User finishes the setup by pressing the Finish Setup button.

Alternatives: If the food is not listed on the app, then the user can pick a similar type of food.
Exceptions: User can't advance without selecting a food type for the app to monitor.

Use case ID: sensorMonitoring Name: Sensor Monitoring
pre-conditions: User finished the setups.
post-conditions: User is required to set up a new food for monitor. User can access the data for that specific food.
Initiated by: None, activated by itself after Food Setup.
Triggering Event: View Food Monitoring button to see data.
Additional Actors: None

Sequence of Events:

1. After the User finishes the Food Setup, the user can press View Food Monitoring to see the data and current predictions for their food.
2. After the food expires, the app notifies the user to throw it out.
3. If the User consumes or throws out the food, the User must press the Stop Monitoring button to stop the app from receiving measurements.
4. The app will now notify the User to do the Food Setup again to view monitoring again.

Alternatives: If the app loses connection with the sensors, it might require the User to do test connections again.
Exceptions: Only Consumer and Shipping users should do setups for food monitoring.

Use case ID: foodRating Name: Food Rating pre-conditions: Sensor Monitoring is finished. post-conditions: Supplier and User can see reviews. Initiated by: User. Triggering Event: Pressing the Submit Review button. Additional Actors: Supplier, User, and Shipping users.
Sequence of Events: 1. After the sensor is done monitoring a food the user can begin a review. 2. The user writes a review, the review automatically matches the supplier. 3. User finishes the review by pressing the Submit Review button.
Alternatives: Can't send multiple reviews of the same Food. Exceptions: If the database fails to receive, the user can write again.

2 Functional Requirements

1 - Consumer

Description: The consumers shopping for the food must learn how to navigate the app and choose the food they test, adjust settings/sensors to measure the gasses emitted, access the estimated expiration date and safety tips/health information, and rate locations of food products.

Rationale: Consumers must know all the information they have access to so they can utilize all the features and information they need.

Fit Criterion: Different food categories such as dairy, meat, fruits, and vegetables available within the database along with a variety of foods within each individual category.

Acceptance Tests: Choose Food Test

2- Transporter

Description: Check if products are safe during their time of being transported to their destination. Will inform the supplier of how the supply is doing.

Rationale: If food is transported it's important to make sure they don't arrive in poor condition as a last check to its destination.

Fit Criterion: test during transportation to ensure quality and shelf life.

Acceptance Tests: Inform supplier Test

3 - Supplier

Description: Check their own supply for any harmful chemicals and get an indication of whether it is okay to supply or sell that food item or not. They will also be able to view safety and expiration information. Lastly they will receive any ratings or reviews they have received from their shoppers, buyers, or consumers.
Rationale: Be able to monitor feedback in regards to their products.
Fit Criterion: Receive ratings and feedback regarding product quality.
Acceptance Tests: Receive Ratings Test

3 Data Requirements

4 - Food Data

Description: The app must provide a setup screen that requires the user to identify the type of food that will be monitored.
Rationale: Different foods expire at different rates. Therefore, it's important for the app to know what type of food is being monitored so it can adjust the boundary values for non-expired food for the sensors and time estimates.
Fit Criterion: The food selected by the user can be matched with the food data from the database to retrieve the data for the sensors.
Acceptance Tests: Food Data Test

5 - Sensor Monitoring Data

Description: The sensors must measure the environment conditions and the odors emitted by the food while monitoring and send the data to the app to make accurate predictions of the food's life span. The data is also stored in the database for future reference.
Rationale: To determine if the food is safe to eat, the odor and gasses emitted by the food must be analyzed. The data can also be used to determine the remaining life span of the food.
Fit Criterion: The sensors should measure methane and ammonia emitted from the food. Temperature and humidity must be measured from the environment. All these values are used to make life span predictions.
Acceptance Tests: Sensor Data Test

6 - Consumer Record

Description: The app must provide a registration page for the consumer to submit their information to a database that can be accessed from the app.
Rationale: For tracking the food throughout its life span.
Fit Criterion: New consumers need to register their information such as their name and contact information.
Acceptance Tests: Consumer Data Test

7 - Supplier/Vendor Record

Description: The app must provide a registration page for the supplier/vendor to submit their information to a database that can be accessed from the app.
Rationale: For tracking the food throughout its life span and allow customers to rate them.
Fit Criterion: New suppliers/vendors need to register their information such as their name, location, contact information, and their products. The necessary data should be retrieved to share with the consumer and/or shipping user.
Acceptance Tests: Supplier/Vendor Data Test

8 - Shipper/Transporter Record

Description: The app must provide a registration page for the shipping user to submit their information to a database. Food environment measurements and time to reach destination data should be provided.
Rationale: For tracking the food throughout its life span. Environment conditions during shipping need to be tracked because it affects the foods lifespan.
Fit Criterion: New shipping users must register their information, and shipping details such as destinations and shipping time. Environment conditions such as temperature and humidity during shipping can be useful for future reference and food life span predictions.
Acceptance Tests: Shipping Data Test

4 Performance Requirements

4a Speed and Latency Requirements

9 - Food Monitoring

Description: The sensors must reliably transmit their measurements to the app for estimating the remaining life span of the food. The data and calculations should be submitted to the database.
Rationale: Reliable transfer of sensor measurements to the app is essential for the app to output the remaining life span of the food and if the food is still edible.
Fit Criterion: The sensors can transmit the measurements over Wi-Fi and/or Bluetooth to the app. The sensors can transmit their measurements at an adjustable rate set by the user. For example, 1 transmission per hour. All the data and calculations should be submitted to the database, but the database submission speed shouldn't matter for the user's experience.
Acceptance Tests: Food Monitoring Test

10 - Database Registration Handling

Description: Database should reliably and quickly store registration data for any type of user.
Rationale: Registration is necessary to access the functionality of the app. When submitting the registration, the database must be quick and reliably store the data before moving on.
Fit Criterion: The user should be allowed to pick the type of user they want to be and sent to the proper registration page to be filled out and then submitted. The data submission should not take more than 1 second.

Acceptance Tests: Registration Test

11 - Backend Database

Description: Database must quickly and reliably provide data that is queried from the app.

Rationale: Slow data retrieval will affect the user's experience making the app feel slow. Missing data can affect the calculations and give inaccurate results. Therefore, a reliable database is required.

Fit Criterion: A cloud database from AWS or GCP can provide these services. The database can be relational or document-oriented databases depending on what is compatible with the app software.

Acceptance Tests: Query Test

4b Precision or Accuracy Requirements

12 - Sensor Accuracy

Description: The sensors must accurately measure the variables of the environment and food gas emissions.

Rationale: Having the most accurate sensor measurements is necessary to make accurate calculations to determine the remaining life span of the food that will be consumed by the user.

Fit Criterion: Calibrations and testing can be done for accurate measurements.

Acceptance Tests: Sensor Accuracy Test

4c Capacity Requirements

13 - Database Capacity

Description: The database must have enough capacity to store all the data for all the users.

Rationale: Keeping records of the foods being monitored can be useful for all types of users to improve the environments where they store their food.

Fit Criterion: The database must provide an unique ID for every food being monitored that is used to track all the sensor measurements over time. When the food changes user or location with different sensors, the food keeps the unique ID, so the data is stored in the same database location. The database must also store the registration data separately. Capacity must be sufficient to store everything.

Acceptance Tests: Database Capacity Test

5 Dependability Requirements

5a Reliability Requirements

14 - 98% Application and cloud database uptime

Description: The application and cloud database system must be available 98% of the time to the consumer, shipper, supplier, and vendor. This product can be widely used with various businesses which may become reliant on it. They should be able to view health/factual information regarding the food items after testing.

Rationale: This product can connect multiple businesses together which can rely on it to get accurate factual information on the food item and decide whether it is okay to sell or buy. Consumers also play a big role in helping run a business, without consumers, there

would be no business. For these reasons, this product's cloud database should be working effectively 98% of the time.
Fit Criterion: The system cloud database can go down 2%, or at most 8 times a year.
Acceptance Tests: Cloud Database Uptime Test

15 - 98% Sensors and testing food uptime

Description: The hardware sensors and food testing functionality must work effectively and accurately 98 percent of the time for the consumer, shipper, supplier, and vendor. This product can be widely used with various businesses which may become reliant on it. They should be able to use it to effectively test their food.
Rationale: This product can connect multiple businesses together which can rely on it to get accurate test results and expiration information on the food item in a timely manner. This can let them know whether it is okay to sell or buy. Consumers also play a big role in helping run a business, without consumers, there would be no business. For these reasons, this product's testing functionality should be working effectively 98% of the time.
Fit Criterion: The testing functionality can go down 2%, or at most 8 times a year.
Acceptance Tests: Testing Food Uptime Test

16 - Backups

Description: In events of anticipated or spontaneous failures, the application must notify the consumer, shipper, supplier, and vendor. A backup database cloud can be used temporarily until the system gets back up.
Rationale: Many people and businesses can rely on this product, so it must provide a backup database cloud to be temporarily used until the system is fixed.
Fit Criterion: Data is backed up every 1 hour, and can be available to use after a system outage.
Acceptance Tests: Backups Test

5b Availability Requirements

17 - 24/7 Availability

Description: The application and cloud database must be available 24/7 to the consumer, shipper, supplier, and vendor.
Rationale: Businesses and people can be reliant on this product to ensure food quality. If the product isn't available 24/7, this can halt business operations and transactions.
Fit Criterion: Features of the applications need to be updated frequently, after every hour in the back up system. New accounts must also be properly registered and made available to the user. Existing accounts must continue to work.
Acceptance Tests: Availability Test, Cloud Database Uptime Test, Testing Food Uptime Test, Backups Test

5c Robustness or Fault-Tolerance Requirements

18 - Temporary system

Description: During failures, the consumer, shipper, supplier, and vendor must be able to continue using the application and sensors through a temporary backup system.

Rationale: This will ensure that business operations and transactions are not halted, and can continue until the main system comes back online.
Fit Criterion: This temporary system must be tested separately to ensure that it will work.
Acceptance Tests: Temporary System Test

5d Safety-Critical Requirements

19 - Food safety test

Description: Testing of sensors must be done every day to ensure that results are accurate and not skewed for tested food.
Rationale: If a food is harmful to consume and is tested as safe to eat, then that would be a huge health/safety concern. If food results are continuing to skew, then the system will be temporarily disabled and the users will be notified.
Fit Criterion: The sensors will be extensively tested by the engineers, and the system will resume once cleared. In replace of the food testing functionality, the users can still search factual information regarding the food item, so business operations and transactions aren't completely halted. All testing will need to be done under technology and food laws.
Acceptance Tests: Food Safety Test

6 Maintainability and Supportability Requirements

6a Maintenance Requirements

20 - Cloud database maintainability

Description: The cloud database must be reviewed and updated frequently with accurate information from google regarding the food items. The cloud database also must be updated with user login information.
Rationale: The system must ensure that up to date and accurate information regarding the food item is provided to the users from google and the cloud database. The cloud database should also be able to have user login information updated if the user registers a new account or changes the name or password of an existing account.
Fit Criterion: Search engines like Google, and a cloud database must be used. The revision of information should be done by the system daily, and updated at least once a month.
Acceptance Tests: Cloud Database Maintainability Test

6b Supportability Requirements

21- Technical support

Description: There must be support provided for any user of the app, the consumer, shipper, supplier, and vendor.
Rationale: Support will help the product users with any questions or problems they may have on the product or application.
Fit Criterion: Support must be provided 24/7 and can be contacted through call or email. A queue will be used to manage the callers on a first come first serve basis. Calls will be limited to at most 10 minutes per customer so everything is being done in a timely manner.

Acceptance Tests: Technical Support Test

6c Adaptability Requirements

22- Platforms

Description: The hardware and application must be used with mobile operating systems. In the future, the application can be used on a PC and web browser.
Rationale: The hardware sensors can be accessed through the mobile application. Here, the user can calibrate and set/connect the sensors for the hardware. We want to start off small with mobile systems, then expand to other platforms such as having a PC application and a web browser for this app in the future.
Fit Criterion: Modern mobile operating systems should be used. The application can be used through android or ios.
Acceptance Tests: Platforms Test

6d Scalability or Extensibility Requirements

23 - Cloud database scalability

Description: The cloud database must scale with time as the food item information, number of different food items, and users of the app may increase.
Rationale: If the database has reached its maximum capacity, the system can get slow.
Fit Criterion: Cloud database will need to be monitored daily to ensure that it is not racing its maximum capacity.When the cloud database gets close to its capacity, the size of the database will scale double in size.
Acceptance Tests: Cloud Database Scalability Test

24 - Application platform scalability

Description: The application must be extended to other platforms in the future such as PC or web browser.
Rationale: The hardware sensors and all functionalities of the application can be used through PC and web browser just like how it's used through mobile operating systems.
Fit Criterion: Support should be in place for modern operating systems and old operating systems too as updates are made.
Acceptance Tests: Application Platform Scalability Test

6e Longevity Requirements

25 - Variable longevity

Description: The longevity of the product and application must rely and depend on the users, such as the consumer, shipper, supplier, and vendor.
Rationale: The product is projected to increase multiple food business involvements. As food businesses stay active, people will also stay active with the product.
Fit Criterion: the hardware sensors and application should work accurately, effectively, and in a timely manner to satisfy the users needs.
Acceptance Tests: N/A

7 Security Requirements

7a Access Requirements

26 - Transporter Access

Description: Have access to the information of where the supplier is located and the desired destination is located.
Rationale: To inform the supplier of how the supply is doing and deliver to the correct place.
Fit Criterion: Transporter only has access to the two locations for deliveries.
Acceptance Tests: Transporter Access Test

27 - Supplier Access

Description: Receive access to all ratings, reviews, complaints they have received from their shoppers, buyers, or consumers.
Rationale: Suppliers will be able to benefit by improving food based on feedback received.
Fit Criterion: Only able to see complaints filed to them but ratings and reviews are public.
Acceptance Tests: Supplier Access Test

28 - System Access

Description: Process results from tested food using accurate resources
Rationale: There needs to be a system to be able to read what the information from the tests tell the user.
Fit Criterion: Only place to see the results and get documentation on what they are via internet/google.
Acceptance Tests: Internet Access Test

7b Integrity Requirements

29 - Email Login

Description: All users can access the app and their information using their personal email address.
Rationale: Easy way to sign in for users and it makes it difficult for others to sign in when using unique email addresses and passwords to them.
Fit Criterion: If email and password don't match then don't give access.
Acceptance Tests: Email Credentials Test

7c Privacy Requirements

30 - Privacy Laws

Description: Any privacy laws that relate to this application will be applied to the data that is being stored.
Rationale: Keeping user data confidential so no one else has access to it.
Fit Criterion: Only users have access to their personal information.
Acceptance Tests: N/A

7d Audit Requirements

31 - Name

Description: N/A
Rationale: N/A
Fit Criterion: N/A
Acceptance Tests: N/A

7e Immunity Requirements

32 - Authentification

Description: Authenticating requests, verifying permissions of users, using a two way factor authentication to make users are who they say they are
Rationale:.Prevents harmful data to enter the app, prevents cyber attacks, and is used to verify identity.
Fit Criterion: 2 way factor authentication should be used
Acceptance Tests: Authentication Test

8 Usability and Humanity Requirements

8a Ease of Use Requirements

33 - Usability

Description: Create an application for all ages and experience levels.
Rationale: We will have a variety of users who vary in age, simplifying is better to understand when some users may not be as experienced as others.
Fit Criterion: Make sure users of all ages can use the application to meet their needs.
Acceptance Tests: N/A

8b Personalization and Internationalization Requirements

34 - International inclusion

Description: Have the option to translate important information/directions into other languages not just in english.
Rationale: Expanding the language will include many other people in different parts of the world.
Fit Criterion: simple checklist for language change.
Acceptance Tests: Language Checklist Test

8c Learning Requirements

35 - System information

Description: In the app there will be a section of instructions under “help” in case the user needs to review how to use the app.
Rationale: We will need a way to address how to use the app so it reduces confusion for users.
Fit Criterion: Common issues and questions will be answered and monitored.
Acceptance Tests: N/A

8d Understandability and Politeness Requirements

36 - Simplicity layout

Description: Simple easy to read instructions, try to reduce wordy sentences, keep sentences simple and easy to read.
Rationale: Making the software readable for everyone.
Fit Criterion: Short and informative wording.
Acceptance Tests: N/A

8e Accessibility Requirements

37 - ADA Guidelines

Description: The font, colors, and layout of the software will be created with the the accessibility clause in the Americans with Disabilities Act (ADA) as requiring all the mobile apps to be accessible to the disabled without any discrimination.
Rationale: Everyone can have the same accessibility to all features of the application.
Fit Criterion: The ADA guidelines will be met for users with disabilities.
Acceptance Tests: N/A

8f User Documentation Requirements

38 - How To Guide Document

Description: Instructions on how to set up the sensor and the different functionalities will be available.
Rationale: Having access to the guide will make sure the users have the correct setup to utilize the app efficiently.
Fit Criterion: N/A
Acceptance Tests: N/A

8g Training Requirements

39 - Teaching system

Description: The people transporting the food should learn how to use the app fast and effectively because of how fast paced their job is.
Rationale: The delivery of products is crucial due to the different conditions food may be in so being quick will make sure products can be delivered quickly.
Fit Criterion: N/A
Acceptance Tests: N/A

9 Look and Feel Requirements

9a Appearance Requirements

40 - Color palette

Description: To avoid overwhelming users and those transporting the food items we would like to have a relatively cool color scheme and exclude loud neon or bright colors.
Rationale: Helps readability for multiple users young and old.
Fit Criterion: Eye strain can be caused by bright screens or colors.
Acceptance Tests: N/A

41 - Font

Description: The size of the font will be mid sized and average no smaller than size 12.
Rationale: Users want to be able to take quick glances when performing tasks instead of spending time struggling to read off of their devices.

Fit Criterion: Optometrists don't recommend small fonts as it relates back to eye strain and can worsen eyesight.
Acceptance Tests: N/A

42 - Visuals

Description: Including visual representations for buttons and foods along with words can help navigation and learnability of the app.
Rationale: Help with navigation and food identification.
Fit Criterion: A variety of users will use the app and some people learn visually and others feel comfortable with just labels.
Acceptance Tests: N/A

9b Style Requirements

43 - Style Guidelines

Description: We want a modern yet efficient layout so all audiences young and old feel comfortable using it.
Rationale: The layout will be easy to remember when navigating which makes tasks quicker and style will be simple which will draw in more users.
Fit Criterion: N/A
Acceptance Tests: N/A

10 Operational and Environmental Requirements

10aExpected Physical Environment

44 - Physical environment

Description: The system must be able to make a wireless connection to the sensors of the hardware through the mobile application. The separate hardware can be placed, wrapped around, or hovered over the food item being tested.
Rationale: Cross contamination or wind interference can be factors that can impact the sensor's ability to get an accurate reading of the food item.
Fit Criterion: The food item being tested will need to be in an enclosure with the sensors to prevent any cross contamination or wind interference, and has to be an open non packaged item. The food item also has to be tested in a controlled peaceful environment.
Acceptance Tests: Physical Environment Test

10b Requirements for Interfacing with Adjacent Systems

45 - Adjacent operating systems

Description: Application must work with mobile ios or android operating systems.
Rationale: Mobile operating systems the user uses should work with the application.
Fit Criterion: Mobile operating systems for ios or android should be used, and should work with current systems.
Acceptance Tests: Adjacent Operating Systems Test

46 - Interfacing hardware and application

Description: Application and hardware sensors must work together adjacently.

Rationale: A big part of the functionality relies on the application being able to connect to the hardware sensors through wifi, and working properly.
Fit Criterion: Wifi connection should be established as strong between both the application and hardware, so the sensors can work properly.
Acceptance Tests: Interfacing Hardware and Application Test

10c Productization Requirements

47 - Installation

Description: The application must be purchased from the app store or google play store. The hardware sensors can be purchased from a retailer or online.
Rationale: Once purchased, the user must connect the hardware sensors with the application following the instructions on the manual that comes with the hardware. Instructions are also provided on the app.
Fit Criterion: N/A
Acceptance Tests: N/A

10d Release Requirements

48 - Updates

Description: The application must be updated from time to time to account for any bugs or changes.
Rationale: Updates and maintenance will fix any bugs, and will work towards improving speeds, and the storage for the cloud database.
Fit Criterion: Maintenance and updates will be done at least once a month.
Acceptance Tests: N/A

11 Cultural and Political Requirements

11a Cultural Requirements

49 - US Culture

Description: App will be set to English by default, but can have chosen language options later.
Rationale: Product will only be available in the United States as there are alot of types of foods around the world unaccounted for.
Fit Criterion: N/A
Acceptance Tests: N/A

50 - Professional Use

Description: App will have option to provide more in-depth information about spoiled food such as percent of types of gasses detected
Rationale: Shoppers have no need for in depth information like gas compounds and only need the final result. Professionals like stores and delivery need in depth information as they are managing more food and need more information to make choices.
Fit Criterion: N/A
Acceptance Tests: N/A

11b Political Requirements

51 - Anonymous Data

Description: All data received by the app will be sent to our database, however, made anonymous. Data used to improve the system.
Rationale: Some clients have reputations to uphold quality food, and anonymizing data will give clients freedom to use sensors freely while also providing valuable data to the database.
Fit Criterion: Database checked for accurate anonymous data.
Acceptance Tests: Database Anonymous Test

12 Legal Requirements

12aCompliance Requirements

52 - Liabilities

Description: We have all users sign terms and conditions that show our app is not liable for any damages that are caused by spoiled food not identified by the app.
Rationale: With possibly many foods being scanned everyday, there are bound to be some mistakes like improper setup of sensors. These conditions will protect our company from lawsuits.
Fit Criterion: N/A
Acceptance Tests: N/A

12b Standards Requirements

53 - Compatibility

Description: Ability to be compatible with as many unique odors sensors as possible. Read data from most odor sensors,
Rationale: Having an app that is compatible with most sensors allows the user to have flexibility when choosing an odor sensor.
Fit Criterion: N/A
Acceptance Tests: N/A

13 Requirements Acceptance Tests

13aRequirements – Test Correspondence Summary

Test	Requirements																																									
	Consumer	Transporter	Supplier	Food Data	Sensor Monitoring Data	Consumer Record	Supplier/Vendor Record	Shipper/Trasporter Record	Food Monitoring	Database Registration Handling	Backend Database	Sensor Accuracy	Database Capacity	98% Application and cloud database uptime	98% Sensors and testing food uptime	Backups	24/7 Availability	Temporary System	Food safety test	Cloud database maintainability	Technical support	Adaptability Requirements	Cloud database scalability	Application platform scalability	Physical Environment	Adjacent operating systems	Interfacing hardware and application	Transporter Access	Supplier Access	System Access	Email login	Authentification	International inclusion	Anonymous Data
Choose Food Test	X																																								
Inform supplier Test		X				X																																		
Receive Ratings Test			X	X																																				
Food Data Test				X																																				
Sensor Monitoring Test					X																																			
Consumer Data Test						X																																		
Supplier Data Test							X																																	
Shipping Data Test								X																																
Food Monitoring Test									X																															
Registration Test										X																														
Query Test											X																													
Sensor Accuracy Test												X																												
Database Capacity Test													X																											
Cloud Database Uptime Test														X																										
Testing Food Uptime Test															X																									
Backups Test																X																								
Availability Test														X	X	X	X																								
Temporary Systems Test																		X																						
Food Saftey Test																			X																					
Cloud Database Maintainability Test																				X																				
Technical Support Test																					X																			
Platforms Test																						X																		
Cloud Database Scalability Test																							X																	
Application Platform Scalability Test																								X																
Physical Environment Test																									X															
Adjacent Operating Systems Test																										X														
Interfacing Hardware and Applications Test																											X													
Transporter Access Test																												X												
Supplier Access Test																													X											
Internet Access Test																														X										
Email Credentials Test																															X									
Authentification Test																																X								
Language Checklist Test																																	X							
Database Anonymous Test																																		X						

Table 1 - Requirements - Acceptance Tests Correspondence

13b Acceptance Test Descriptions

Choose Food Test

Description: Ensure the user can pick an individual food item to test in their category (dairy, meat, fruits, and vegetables available within the database).

Inform Supplier Test

Description: Ensure the transporter can inform suppliers of the food's status.

Receive Ratings Test

Description: Suppliers are able to receive feedback in regards to their products.

Food Data Test

Description: Test the return of food type data from the database to adjust boundary values for non-expired food for the sensors and life span estimates.

Sensor Monitoring Test

Description: Test the sensors if they are measuring gas emissions from the food and environment data. Test that the sensor sends the data to the app.

Consumer Data Test

Description: Test the registration for consumer users in the database.

Supplier Data Test

Description: Test the registration for supplier users in the database.

Shipping Data Test

Description: Test the registration for shipping users in the database.

Food Monitoring Test

Description: Test that the sensor is measuring the emissions of the food, making life span predictions for the user, and determining if the food is safe to eat.

Registration Test

Description: Test the registration screen in the app.

Query Test

Description: Test sending queries to the database.

Sensor Accuracy Test

Description: Test the accuracy of the sensor measurements. Test calibrating the sensor first.

Database Capacity Test

Description: Ensure that the database has enough capacity for all the data collected in this app.

Cloud Database Uptime Test

Description: Tests the availability time of the cloud database.

Testing Food Uptime Test

Description: Tests the availability time of the hardware sensors and food testing system.

Backups Test

Description: Tests the backup cloud database.

Temporary Systems Test

Description: Tests the temporary backup system.

Food Safety Test

Description: Tests accuracy of food results from hardware sensors and application.

Cloud Database Maintainability Test

Description: Tests and ensures that the cloud database is being updated with accurate information on the food items, and also log in information.

Technical Support Test

Description: Ensure that lines are working properly for calls. Ensure that emails are also being properly received.

Platforms Test

Description: Ensure that the hardware and application is properly being used with modern mobile operating systems.

Cloud Database Scalability Test

Description: Ensure that the cloud database is scaling double in size as more log in or food item factual information is added.

Application Platform Scalability Test

Description: Ensure that other platforms such as PC or web browser work properly with connecting to the hardware sensors.

Physical Environment Test

Description: Ensures that the connection with the application and hardware sensors are properly made, and that the open unpackaged food item is properly tested in a controlled environment inside the hardware. The food item can be tested by being placed in, wrapped, or hovering over the hardware.

Adjacent Operating Systems Test

Description: Mobile operating systems for android and ios should work fine with the sensors and system overall.

Interfacing Hardware and Applications Test

Description: Ensure that a strong connection is established between the application and hardware so the sensors can work properly.

Transporter Access Test

Description: Ensure Transporter only has access to the supplier and delivery locations for deliveries.

Supplier Access Test

Description: Only able to see complaints filed to them but ratings and reviews are public.

Internet Access Test

Description: Ensure the system has internet access when processing results and information.

Email Credentials Test

Description: Ensure if email and password don't match then don't give access.

Backup Data Test

Description: Prevent data from becoming corrupt or in case there are user errors we have another way to access the database.

Authentication Test

Description: Prevents harmful data to enter the app, prevents cyber attacks, and is used to verify identity.

Language Checklist Test

Description: Ensure use of a simple checklist for language change.

Database Anonymous Test

Description: Ensure data recorded on database is anonymous and data can't be traced back to users.

III Design

1 Design Goals

The design goal for the application will be to provide an easy to use simple UI. Many different kinds of users will have access to the app including consumers, employees, and or companies. It is crucial that it is easily accessible to all. It is also important to have an efficient database, the database is where the information of the foods and their data will be stored for an accurate reading when foods are being tested. The main idea is to provide expiration dates to each food tested so it will eliminate possible food waste and make monitoring foods easier for vendors/ suppliers.

2 Current System Design

There currently is not a pre-existing system at the moment.

3 Proposed System Design

For the proposed system design we have included a type of Measurement System which will be what we use the Arduino for. The Arduino will then need compatible gas sensors so that functionality of reading the harmful gasses emitted from outdated food is possible. Along with the sensors is a Breadboard to connect the components of the sensors together for the Fresh Food App to operate correctly and efficiently. A key to testing whether or not the sensors are connected and working as expected is to include the Bluetooth arduino module and the WiFi arduino module. These two modules will give the user feedback when connected and will be responsible for giving the data from the sensor to the user. A cloud database is necessary to keep track of what food has been tested, what the results indicate, and all user information.

To create an Android application these tools will be very useful in doing so. Creating the app in Android Studio will make it relatively easy to see progress using the emulator , Java programming language is what is currently compatible with android studio , XML is also what android studio uses for its activity files.

iOS app: Xcode, Swift, SwiftUI

3a Initial System Analysis and Class Identification

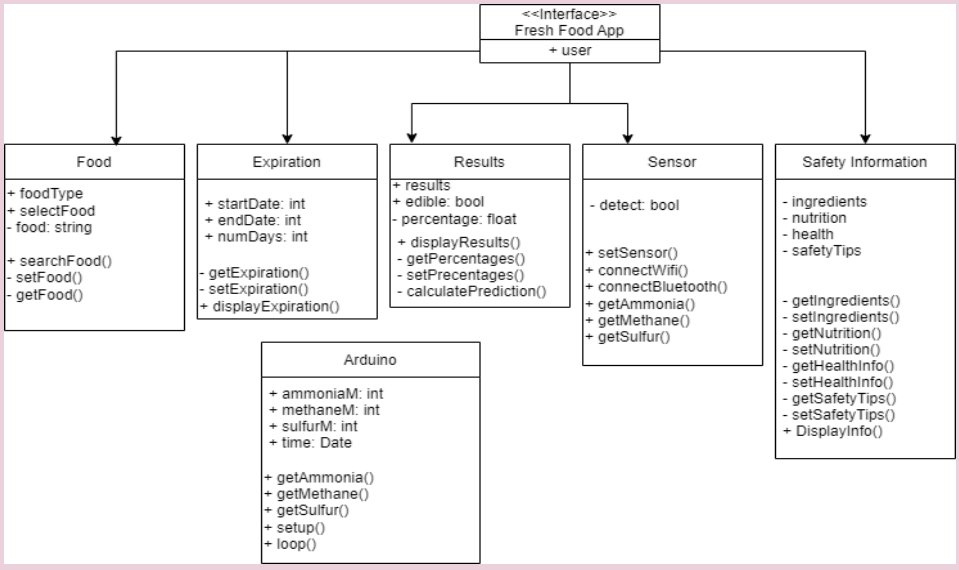

Figure 5. *Class diagram of Fresh Food App.*

Some of the most important classes that will need to be created and utilized are the Food, Expiration, Results, Sensor, Safety, and lastly we will need to include the Arduino class to help with testing for the gasses. Each of these classes serves a specific purpose. Food class gives the user a chance to input the food item they're testing, the Expiration class gives the dates starting and ending with days included to keep track of when to dispose of food, and the Results class gives a percentage and whether it is edible or not. The Sensor class will read the gasses emitted, Safety class will include ingredients, nutrition, health, and safety tips, lastly the arduino is what will be used to measure the gasses we are looking for.

3b Dynamic Modeling of Use-Cases

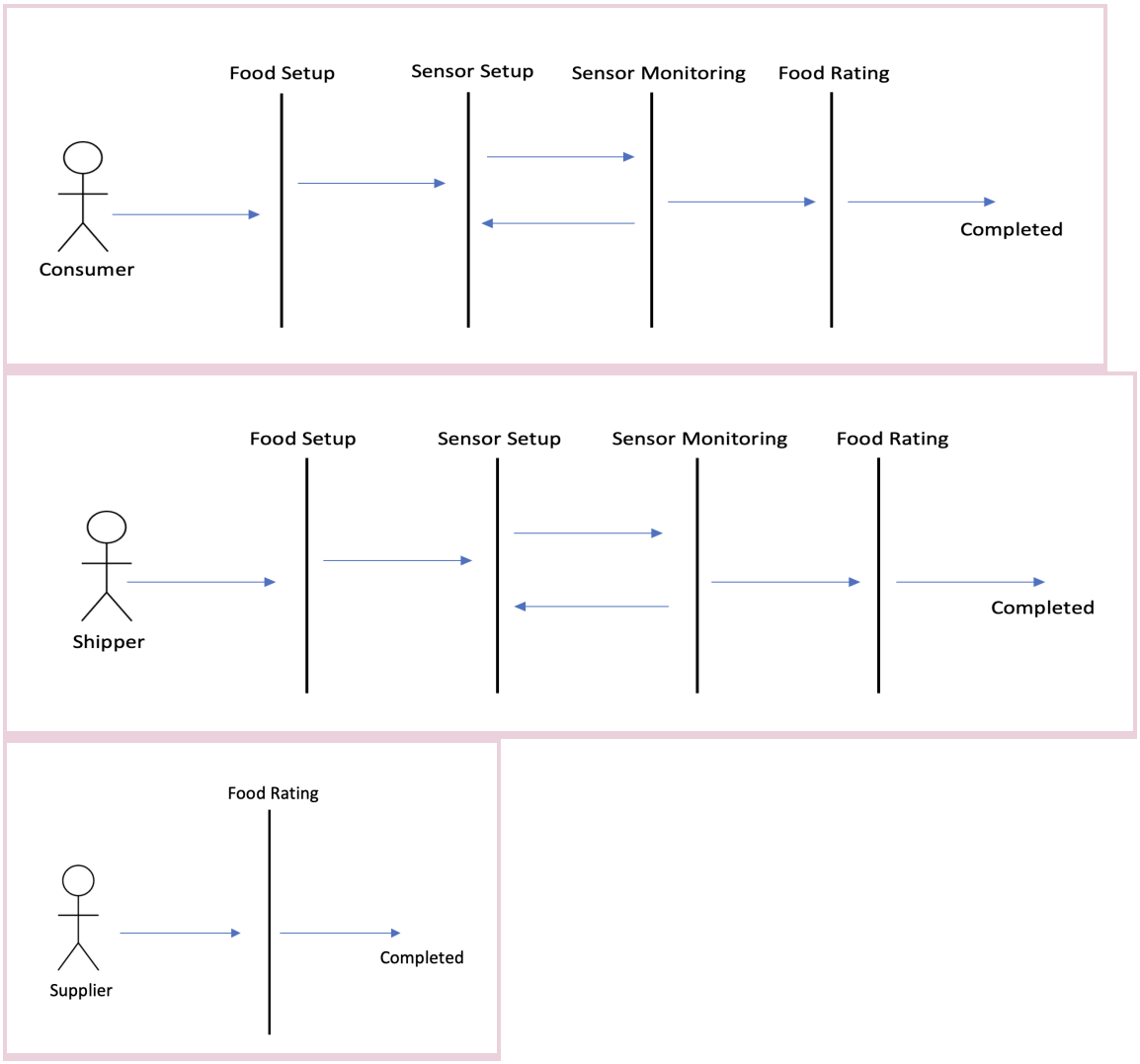

Figure 6. *Sequence diagram*

In the sequence diagram above it shows what a typical user of that type is expected to see and do while accessing the Fresh Food Application. The consumer and shipper share the same capabilities while the Supplier has access to the ratings it receives from the Consumer and Shipping user.

3c Proposed System Architecture

The Fresh Foods applications proposed system architecture is a client -server architecture. The application will begin by having a welcome screen with all choices readily available to click on. Client will choose food to input and the server will test what amount of gas is

emitted and what that reading indicates. The information the server collects of food data will then be given back to the client.

3d Initial Subsystem Decomposition

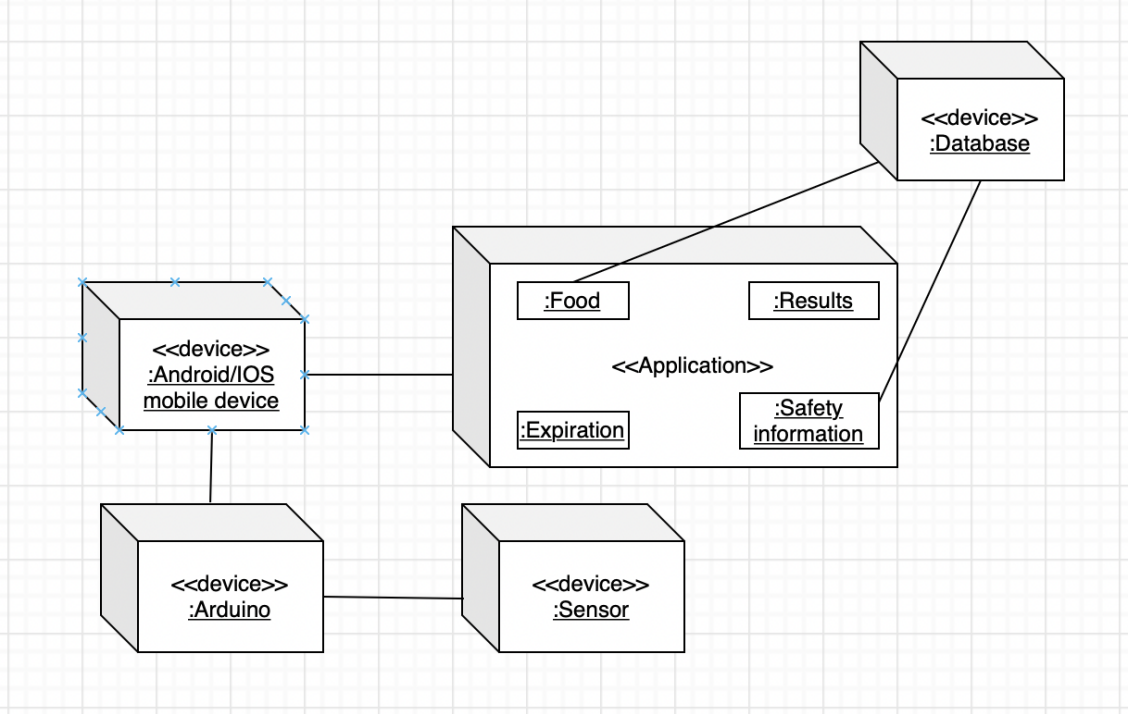

Figure 7. *Subsystems diagram*

Food
This class is responsible for being able to search and select a food to test. The database can be used to retrieve the particular food item.

Expiration
This class will be responsible to calculate the expiration date for the food being tested from the real time data that is sent from the sensors.

Results
This class will be responsible for calculating the percentages of different gasses/odors that are detected from the sensors. Those percentages will be displayed to the user.

Safety information
This class will be responsible for accessing the database for a particular food item, and displaying factual information, such as health information, nutritional information, and safety tips for that food item.

Arduino
This class will be responsible for setting up the arduino system and the sensors with it.

Sensor
This class will be responsible for setting each sensor accordingly to a specific gas/odor. For example, there should be a sensor for detecting ammonia, a sensor for sulfur, and another for methane.

4 Additional Design Considerations

4a Hardware / Software Mapping

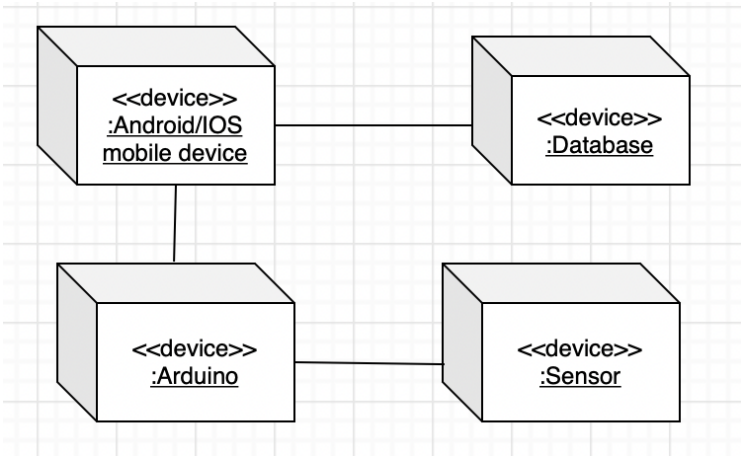

Figure 8. *Hardware/Software Mapping*

4b Persistent Data Management

The system will need a backup database in the case the main database goes down or gives technical issues. A server can be used for the backup database. When the main database has issues or needs maintenance, the server can be switched to the backup database. The backup database server class and the main database server class are the same as they fulfill the same purpose. The backup database can be kept offline to prevent technical issues, and will need to be updated frequently from the data that's in the main database.

4c Access Control and Security

The possibility of users creating an account can have security concerns just in case if the system was under attack. Though the information the user has is not personal, and should not be relevant to an attack. A greater concern would be if the database that holds factual information about various food items got corrupted. This could result in important information about the food being corrupted or false. As a result this can look really bad for the product and the company which could retract customers away. For this reason, the main database will need to be monitored frequently, and if anything got corrupted, the backup database could be used temporarily.

4d Global Software Control

The actual hardware will take intensive research and testing, for demonstration purposes an arduino system can be used. For the final product, any updates that are made to the software, should be compatible with the hardware, and these should be reflected globally. The Database should also be able to work well globally without any concerns. Latency issues can be a possible concern with the database. Improving and updating the efficiency of using the database can help with this concern.

4e Boundary Conditions

During maintenance or updates, the user of the app can be informed of maintenance. Depending on the type of maintenance, the user may or may not have access to the database or the testing functionality of the app. Technical support for the app and hardware can be available to help the user with any technical issues.

4f User Interface

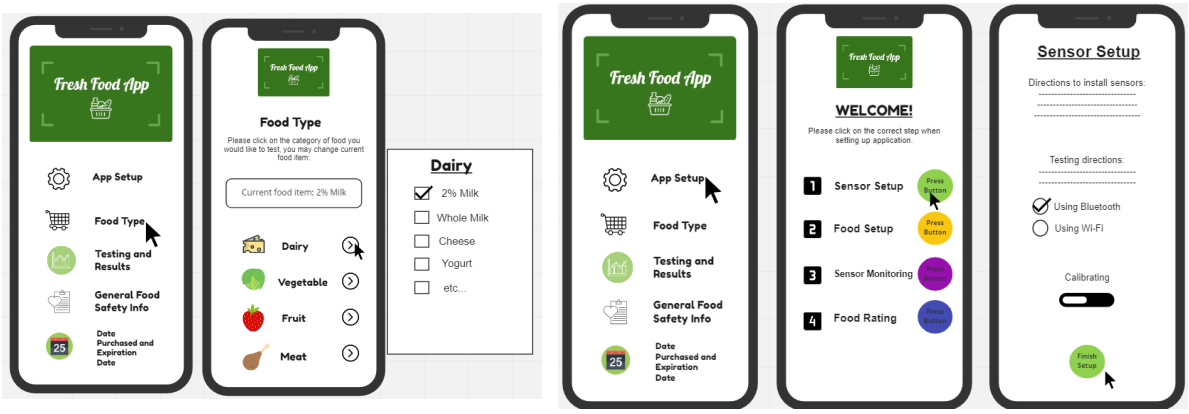

Figure 9. *User Interface of the Fresh Foods App*

The UI design above is showing what choosing the Food type option of the application will do and what is expected and how to choose the specific food item the user would like to test. Another UI design included is the Sensor setup settings which initiates how the user will go about setting up the app so the sensors work correctly and efficiently.

4g Application of Design Patterns

N/A

5 Final System Design

The Fresh Food App will be a three-part system, with a sensor(Arduino), a mobile app, and a database. The sensor will pick up gas odors and measure them with classes that control the sensor and a class to handle the transmission of data between the sensor and mobile app. The mobile app will show all relevant information to the user in the form of classes of food and

its inner classes such as result and expiration. The mobile app must also have a class that controls the sensor as well as and class that handles database calls. Lastly, the database must hold previous samples with data including food names, gas percentages, and edibility.

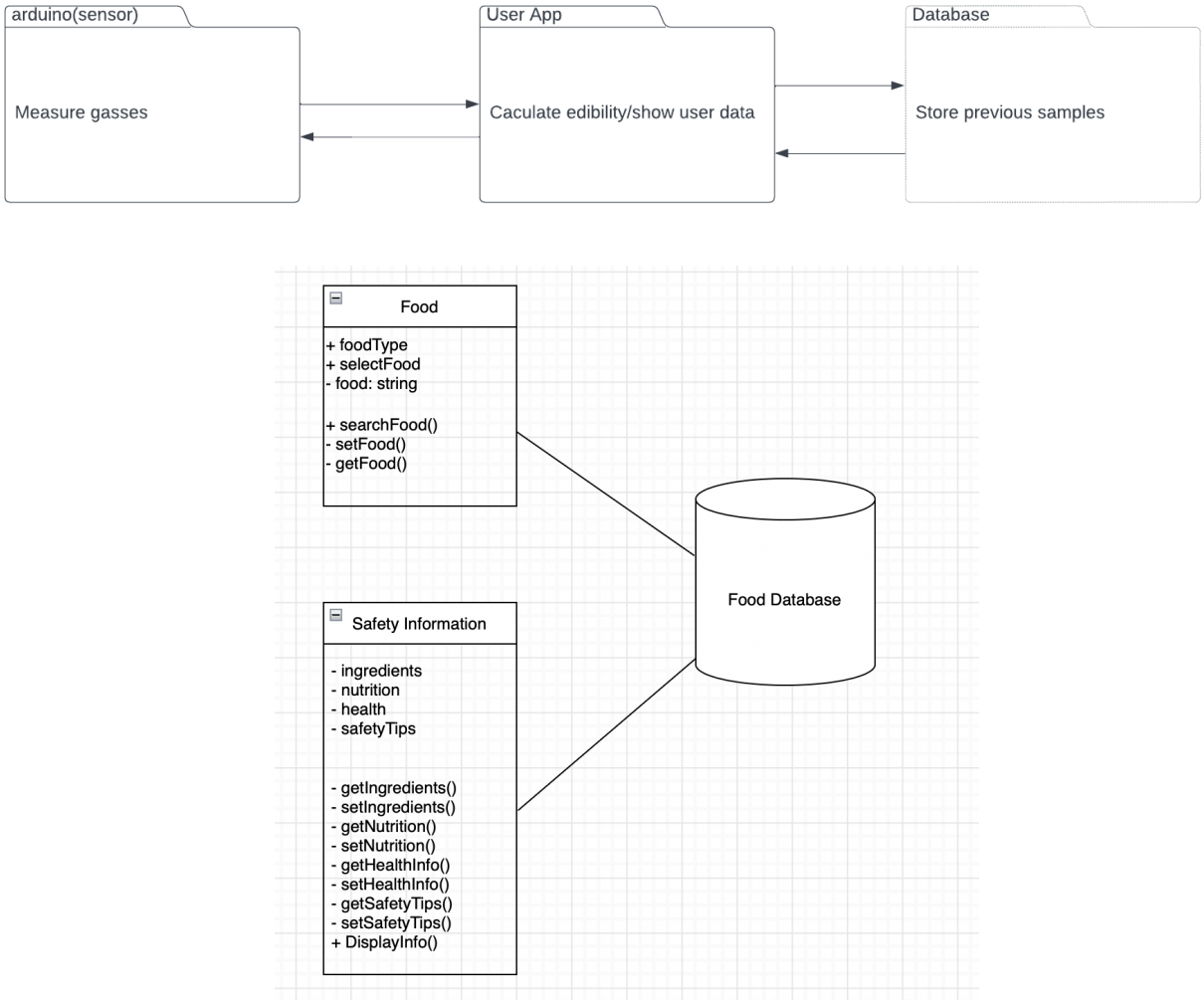

Figure 10. *Final System Design*

6 Object Design

6a Sensor (Arduino) Subsystem

The class for sensors will need methods that can handle all of the gas sensors such as when they are activated, how long sensors are active to get accurate samples, and checking if sensors are working properly. The sensor class will also need to handle sending and receiving requests from the main application including sending of gas data.

6b App Subsystem

The food class will be what kind of food the user is testing. It will hold the name of the food and what type of food(dairy, grain). The food class will also have multiple inner classes including results, expiration, and safety information. The result will have the results from the sensor, including percentage gasses, and edibility boolean. If the edibility is true, the expiration object will be used with the estimated date/time until food will be inedible. Lastly, food objects will come with safety information, including ingredients and nutrition which can be added by the user or estimated by similarly named foods from the database.

The app will also need a class to handle sensor commands to the odor sensor or Arduino. Methods to activate the sensors as well as when to start collecting data. There also needs to be some methods to handle the calculation of gas data when received from the Arduino. There should be a method that controls the sensors' deactivation and activation because the sensor remaining on would use a lot of electricity.

A Util class will be needed to hold methods that contact the database including sending and receiving data such as edibility estimations, gas makeup, and types of food users are scanning. These methods should be implemented/used in a multithreaded environment to prevent the freezing of the UI of the user. The Util class should offer some protection on incorrect, ill measured data such as preventing foods without a legit name from entering the database.

IV Project Issues

1 Open Issues

For the next group of developers that decide to take on this project they are most likely to encounter the following problems:

1. Arduino Bluetooth and WiFi
   Working with the Bluetooth and WiFi modules for Arduino is difficult. From previous experience, sending multiple data streams over the serial ports of an Arduino will be difficult, especially for inexperienced developers. Also, the developers will need to learn how to use the bluetooth functionality of an iOS or Android app which might not be compatible. Therefore, the developers should skip the direct Bluetooth and WiFi data transfer from the Arduino system to the app system. The Arduino will measure data, calculate predictions, and send data to the cloud database. The app should only pull data from the cloud database and display the measurements and predictions. This should make developing the project much easier.

2. Container
   Some of the gasses emitted by spoiling food have lower density than air which will cause the gas to rise making measurements inaccurate in an open environment. For example, Methane and Ammonia gasses are lighter than air. Therefore, future developers must consider and test using an enclosed container to put the food and sensors in. The buildup and sudden releases of these gasses will affect the prediction calculations. The developers

need to account for this in the prediction calculations and allow some ventilation for more accurate measurements.

3. Types of Gas Emissions
   Not all the gasses emitted by spoiling food is going to be pure Methane, Ammonia, and Sulfur. Many variant compounds of these chemicals are emitted and should be factored in the prediction calculations if they are determined to be important enough.

4. Database Library for Arduino
   Arduino IDE is C++ with many Arduino libraries. Depending on what cloud database is selected, the developers need to make sure that it will be compatible with Arduino C++. If this becomes a problem, then the developers should consider using Python to program Arduino and then send the data and calculations to the database. Most cloud database libraries should be available for Python.

2 Off-the-Shelf Solutions

Microcontrollers with gas sensors wired together with Bluetooth and WiFi modules should all be wired together to make the gas measurement system. All of these components should be easily available to make a prototype. The final product sold should be a container that already has a single Printed Circuit Board with the sensors and microcontroller in a secured position. Any compatible cloud database service can be used for the app and data collection.

2a Ready-Made Products

An Arduino microcontroller should be used to handle the sensor measurements. Arduino kits with a breadboard and enough wiring can be easily found to be used for this project. Figure X shows an example of an Arduino kit for $40. A single arduino should have enough pins to control everything. But if that is not the case, more Arduinos should be used to measure and send data.

Figure 11. *Arduino Kit example from Amazon.com.*

Arduino is compatible with the MQ series of gas sensors. These sensors can be easily ordered online. Figure X shows a collection of MQ sensors that can be ordered for $10.58. Figure X shows the MQ-137 Ammonia Gas sensor available to order for $37.50. Other sensor types exist and are available to order online. Developers need to test which of these sensors are suitable enough for the project. Figure X shows an example of an Arduino Bluetooth module. Figure X shows an example of an Arduino WiFi module. Other Arduino Bluetooth and WiFi modules exist in online stores and should be considered.

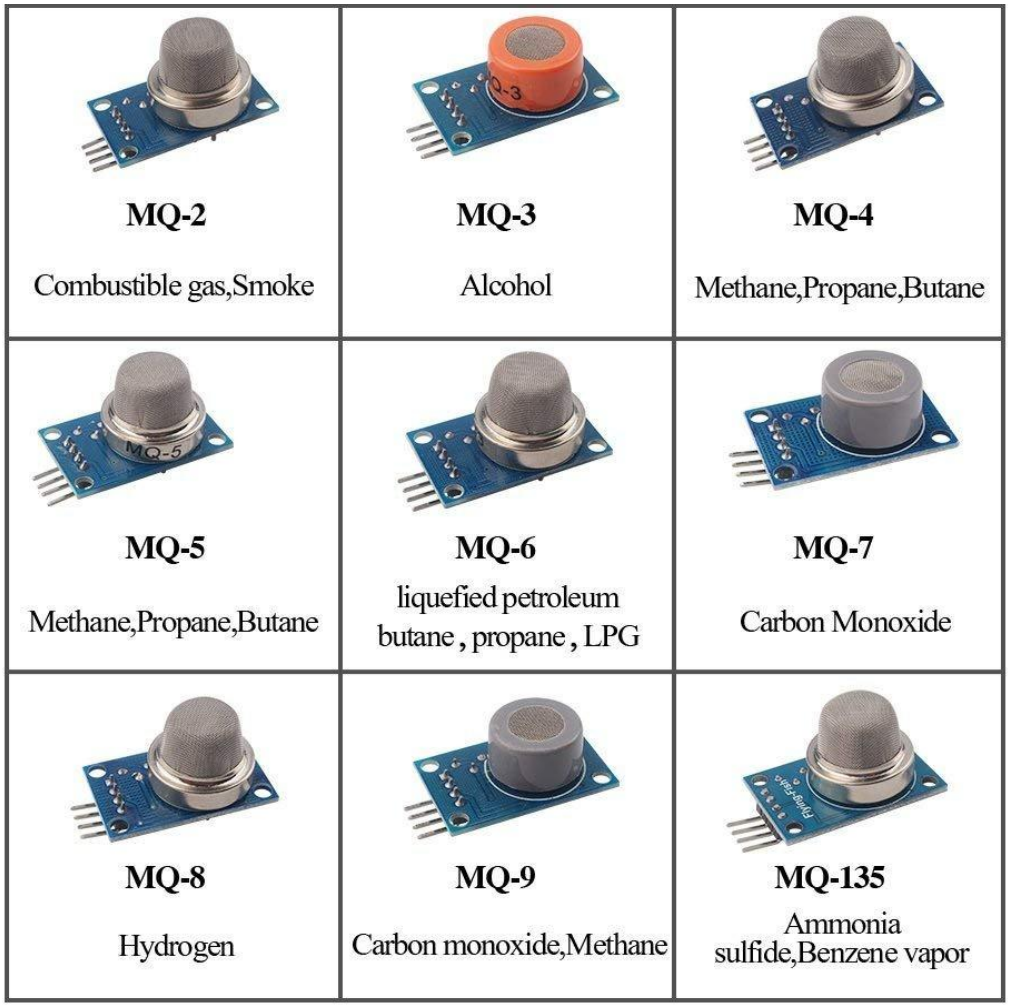

Figure 12. *MQ Gas Sensors from Alibaba.com.*

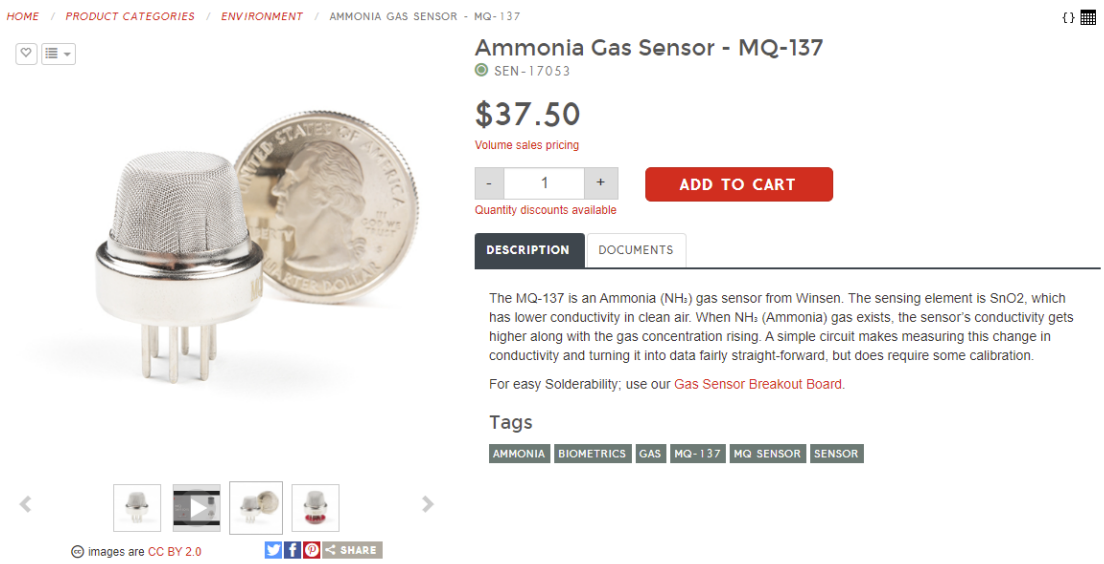

Figure 13. *MQ-137 Ammonia Gas Sensor from Sparkfun.com.*

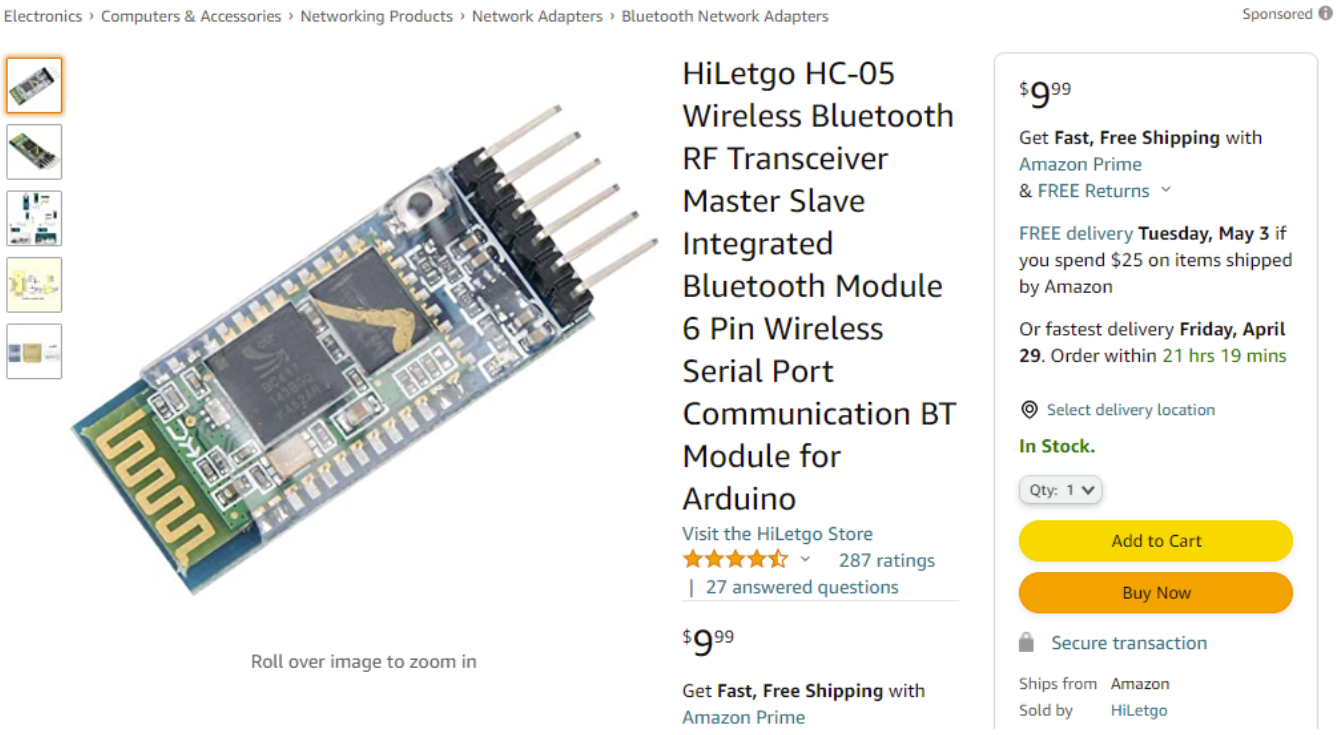

Figure 14. *Arduino Bluetooth module example from Amazon.com.*

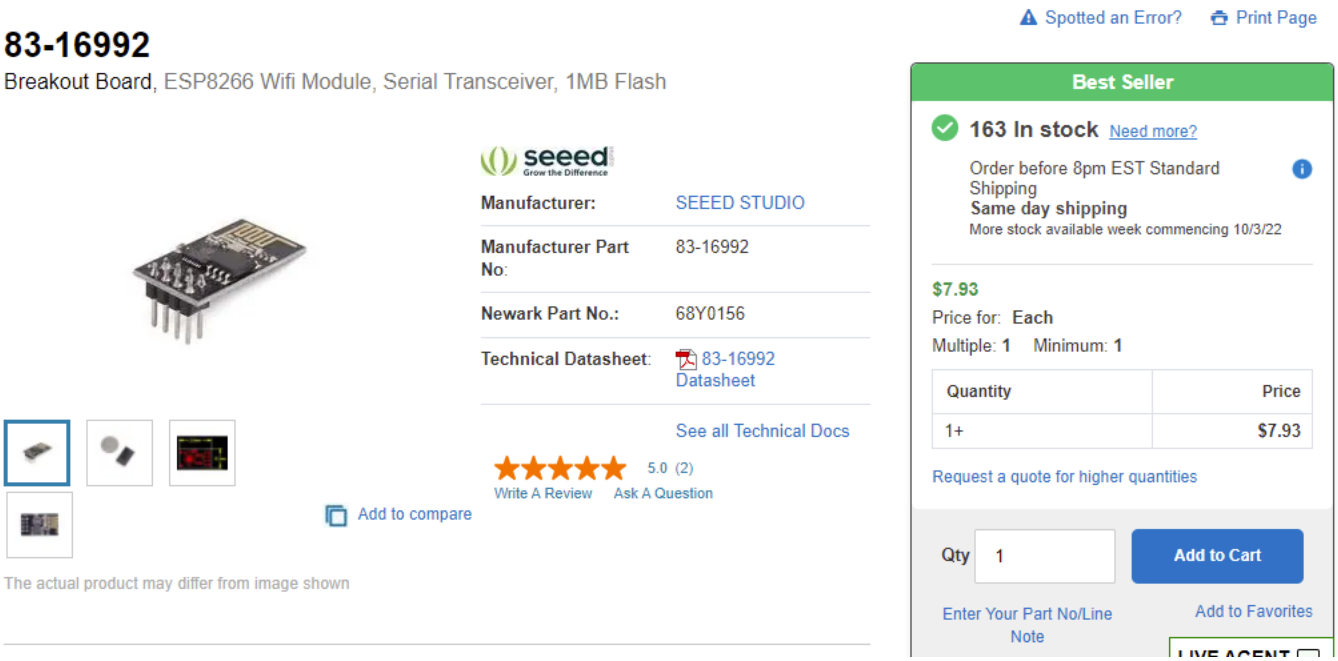

Figure 15. *Arduino WiFi module example from Newark.com.*

Multiple cloud databases exist that can be used for this project. The most popular being Amazon Web Services (AWS), Google Cloud Product (GCP), Microsoft Azure, and Firebase. For a demo, these services should be free.

2b Reusable Components

The Arduino measurement system should be reusable until the sensors begin to fail or begin to accuracy. Some of the environments the sensors will be in will shorten the measurement systems life span. The measurement system should be recycled or shipped back to salvage and reuse parts that still work.

2c Products That Can Be Copied

NA

3 New Problems

3a Effects on the Current Environment

If the systems are successful less food will be wasted. Lowering the amount of food wasted will save the environment from pollution caused by spoiling food and transportation. The measurement system can endup in a landfill if the owner decides to throw it out instead of recycling or shipping it back.

3b Effects on the Installed Systems

NA

3c Potential User Problems

Users that fail to properly install the measuring system can have a frustrating experience when the measurements are not being received or are not accurate. Users that are not proficient in using apps could find the system difficult to use. Therefore, the app and installation should be easy enough that the vast majority of users are able to use it without a problem.

3d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The measurement system can have problems in a wet environment either caused by condensation or water leaks because of the electronics that are used. Droplets could also block the sensors affecting the measurements. Vibrations during transportation could also affect the reliability of the measurement system.

3e Follow-Up Problems

Faulty sensors and manufacturing quality control mistakes.

4 Migration to the New Product

NA

4a Requirements for Migration to the New Product

NA

4b Data That Has to Be Modified or Translated for the New System

NA

5 Risks

The biggest risk of this project is the case of inaccurate sensor readings that can mislead users into thinking that the food is safe to eat. Sensor calibration tests and other safety features in the app and measurement systems must be available for the user. Sudden sensor failure will also affect the prediction calculations. When these types of failures happen the user must be alerted and told not to trust the measurements.

6 Costs

Using the same components shown in Section 4.2:
Arduino Kit: $40
MQ Gas Sensor Collection: $10.58
Bluetooth Module: $10
WiFi Module: $7.93
Cloud Database: $0
Total: $68.51
A demo product of the Fresh Food App could be made for about $68.51. The cost most likely will go up if the sensors are not good enough or different pricer sensors are used.

7 Waiting Room

Develop a system that takes the environment values of where the food is contained and adjust that environment to match the ideal conditions for that food to extend its life as much as possible. This can be done by designing a container that adjusts ventilation, temperature, and humidity that works with the Fresh Food App measurement system.

8 Ideas for Solutions

For the app creation, I would recommend android studio as it is decently user friendly with an emulator. There are also a lot of guides that talk about how to hook up android studio to databases.
For databases, I would recommend Google Firebase because it is free for 50,000 pulls/updates as well as it offers real time updates without the user having to actively send a request. However, threading will be needed to handle the calls to the database.
For arduino, CS 362 in 2021 previously required an arduino kit, so if that is an option, I think it would be a useful asset.

9 Project Retrospective

The application that we have created and used for our development project contains many elements that we tried to break down as much as possible to think about every aspect that is needed. A segment of the project we may have not thoroughly thought about and planned efficiently is how the sensors will work. We decided to use an arduino as we have some prior experience with them but there may be easier alternatives out there that we are not familiar with but we applied what information we thought was helpful and easily accessible at this time. Another aspect that could be evaluated more was the container or how the sensor will be reading the food item.

There are so many details that come with developing an application that it is very easy to overlook what is important. Developing this report emphasizes how important research is to get an application that operates exactly how you intended. As we complete our development project we are happy with the time, effort, and amount of research we each did. Our group enjoyed the topic and application we have chosen to develop and believe it is something that will be developed in the future as tech evolves. The project idea we have come up with was ambitious and innovative, we are satisfied with our choice and execution of our development project.

V Glossary

Buyer: An individual, shopper, or business that is planning on purchasing or has purchased produce, meat, or dairy product(s).

Consumer: An individual who is about to eat or has eaten produce, meat, or dairy product(s).

Shipper/Transporter: An individual, or employee of a business who ships/transports produce, meat, or dairy products from the supplier to the intended destination.

Vendor: Farms, companies, restaurants, business owners, markets, stores, employees, or anyone who sells fresh produce, meat, or dairy products.

Supplier: Farms, companies, restaurants, business owners, markets, stores, employees, or anyone who supplies fresh produce, meat, or dairy products to a buyer.

VI References / Bibliography

[1] Robertson and Robertson, Mastering the Requirements Process.

[2] A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013.

[3] J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012.

[4] M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004.

Shop Links:
https://www.amazon.com/ELEGOO-Project-Tutorial-Controller-Projects/dp/B01D8KOZF4/ref=sr_1_1_sspa?gclid=CjwKCAjwsJ6TBhAIEiwAfl4TWEAPNw3oOQVsG7lc2XMcI1mszMjfAPoQgvf3jeomUY5aovbjRnMgmxoCE4wQAvD_BwE&hvadid=410002106846&hvdev=c&hvlocphy=9021541&hvnetw=g&hvqmt=e&hvrand=10818002164915531210&hvtargid=kwd-79704347864&hydadcr=24664_11410922&keywords=arduino+kit&qid=1651037200&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFUOVJWUEdGRzU5UzcmZW5jcnlwdGVkSWQ9QTAwOTYxMDgxNjVJV0VKSlk2MzhVJmVuY3J5cHRlZEFkSWQ9QTEwMDEzNzYzVFNPV0pHQ01HTlNCJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==

https://www.alibaba.com/product-detail/9PCS-Lot-Gas-Detection-Sensor-Module_62340496855.html

https://www.sparkfun.com/products/17053

https://www.amazon.com/HiLetgo-Wireless-Bluetooth-Transceiver-Arduino/dp/B071YJG8DR

https://www.newark.com/seeed-studio/83-16992/esp8266-wifi-module-breakout-board/dp/68Y0156?COM=ref_hackster

VII Index