Jun 9, 2019
Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
British Columbia, V5A 1S6

RE: ENSC405W/440 Requirement Specification for FooMinder

Dear Dr. Rawicz,

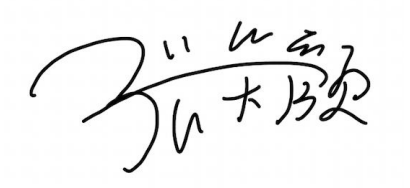
This requirement specification document for FooMinder was prepared by Freshist for ENSC405W/440. Our intention for this capstone project to apply image recognition and sensor data analysis to provide a reminder and organizer for groceries. FooMinder is built to be a tool to use in households for daily usage.

FooMinder will use a camera to capture food images and send the data to the microprocessor to perform food recognition. It will provide storing time estimations and set up a reminder for the users to keep track of their groceries. The detector is used to detect the current freshness by extracting the concentration of Nitrate thus determine the safeness to eat.

The requirements specification will review from the general requirements to more detailed requirements such as our software and hardware components. This document will discuss safety and sustainability requirements as well as the engineering standards our project is following. The requirements are covered in terms of proof of concept, the prototype and the final product which provide a timeline for our project. Lastly, this document provided an acceptance test plan including software and hardware plans.

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

Sincerely,



Xiaoyan Zhang
CEO
Freshist

FRESHIST

Requirement Specification:

FooMinder

Partners:

Harry Gong
Yixin Hu
Wei Liang
Xiaoyan Zhang

Contact:

Yixin Hu
huyixinh@sfu.ca

Submitted to:

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

Issued Date:

June 9th, 2019

Abstract

This document specifies the functional requirements of the food reminder system, FooMinder. Firstly, it will provide the requirements that describe the general functionality. Secondly, it will present the requirements for each system component. This document also gives the safety and sustainability requirements as well as Engineering standards our project must satisfy. The purpose of this document is to give readers a thorough understanding of the device and its system, including its background, scope and basic operations while excluding detailed design which will be presented in a later date.

The FooMinder is a system that can be separated into four components: a camera to capture images of the food, a detector to detect the concentration of Nitrate, a microprocessor to perform image processing, data analysis thus output the approximated storing time and current freshness of the food, moreover a screen to visualize the database and provide reminder feedback to the user. The device consists of the camera and the detector. The user interface will consist of a screen, a power button and a LED indicating whether the camera and detector are on or off. This will make this device user-friendly.

This requirements for FooMinder cover the following core components:

- General Requirements: including system and functional requirements.
- Firmware/Software Requirements: including general, image classification, graphical user interface software and performance requirements.
- Hardware Requirements: including general, sensor, processor, camera and display, wired and wireless, power supply and performance requirements.
- Acceptance Test Plan: including software and hardware plans.

This document will conclude by discussing the planned deliverable for the proof of concept which is going to be shown in August 2019.

Table of Figures

Table of Tables

Glossary

Confusion Matrix - A confusion matrix is a table that allows the visualization of the performance of an algorithm.

SBC - A single-board computer is a small computing device built on a single circuit board with a microprocessor, memory, I/Os, and other necessary hardware for a functional computer.

Table of Content

1. Introduction

FooMinder is designed to be a user and cost-friendly device that is able to provide food identification, detect food freshness, and automatically set the alarm to remind people of the expiry date for certain food. This compact and portable device uses a small camera to capture the pictures of the fruit or vegetable, and detect food freshness by a detector. These images would be sent to a microprocessor (SBC) that performs image processing using an image recognition API on food images. The sensor which is built in the processor can be inserted into the fruit or vegetable to performance the freshness sensing. Finally, based on the image processing and the sensor data analysis, the system would send feedback to the user regarding the food freshness and the duration that it can be stored.

We aim to make the FooMinder a tool for, from a small point of view, the person who often forgets certain food he/she bought and find out after the expiry date or big families that need a way to organize available food. It could also be a tool for large food manufacturers that need to organize and monitor the freshness of certain foods. By using FooMinder, we believe it would be an effective way to get the best nutrition from the food and help reducing food waste and saving money for the users.

1.1 Background

When people go shopping and bring a lot of groceries back home. After putting them into the fridge, they tend to forget what they bought and when they realize, some groceries are already bad. Processed foods are easy to determine the expiry date by reading information from the boxes. However, fresh fruits and vegetables are ambiguous to tell the date. FooMinder aims to provide assistance for determining those expiry dates. To achieve this, our system will use a camera to capture the images of food and send the data to the microprocessor to perform image processing and output an estimated expiry date. The detector is used to detect the current freshness by extracting the concentration of nitrate thus determine providing extra data in terms of safeness of the food.

Furthermore, according to the Food and Agriculture Organization (FAO), there is roughly one-third of all food produced globally is either lost or wasted. Food is wasted in various ways. For example, foods that are close to, at or beyond the “best-before” date are often discarded from household kitchens by consumers. [2] Less food waste would have positive influences on climate change and sustainability. Reducing food loss and waste is crucial to creating a Zero Hunger world and reaching the world’s Sustainable Development Goals. [9]

1.2 Intended Audience

This document serves as FooMinder's functional requirements for Freshist members, instructors, Craig Scratchley, Dr. Andrew Rawicz, and teaching assistants supporting us in this course. Future revisions will draw from the framework detailed in this document.

1.3 Requirement Classification

The requirement in this document will follow the following convention:

Req {Domain}.{Subsection}.{Requirement Number} {Stage of Development}

Example: Req 1.1.1.P

Domain of Requirement	Encode
General	3
Firmware/Software	4
Hardware	5
Safety/Sustainability	6
Engineering Standards	7

Table 1.1: Domain of Requirement Encoding

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

Table 1.2: Stage of Development Encoding

2. System Overview

FooMinder is a portable device using image processing to detect the food freshness and alert the user of the expiry date when a certain food is going bad. FooMinder is designed for households that need reminders on the food expiry date so that reduce the waste and save money. It is our goal to provide a small, simple-use device to help families with organizing their groceries.

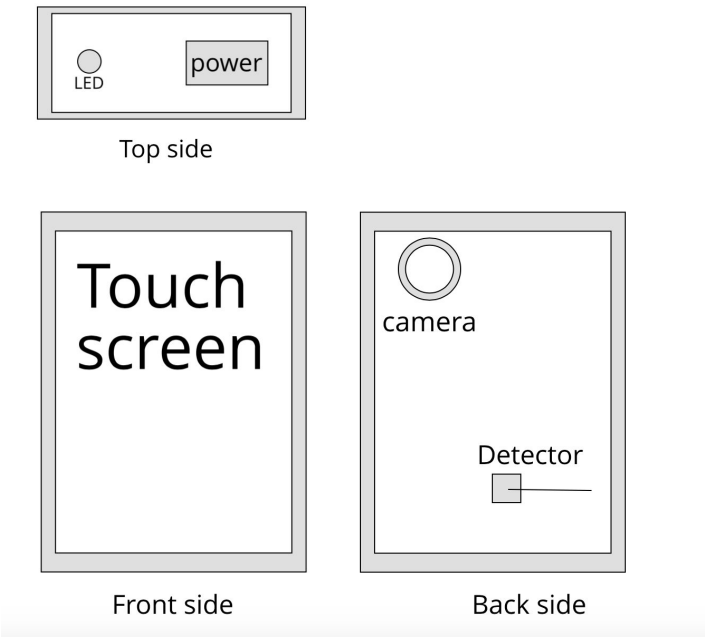


Figure 2.1: Concept Design

The figure above shows the concept design of FooMinder. We are going to use a touch screen to display information and control user input. The power button is on the top side. Camera and detector are on the back of the controller. There is a tri-colour LED to indicate the camera or detector is ON or OFF.

The actions needed by users are simple. Users can control the camera and the detector using the touch screen. By holding the detector with some distance from the food. The camera will capture the image and send the image data to the microprocessor (SBC). The processor will perform image processing and recognize the item. With common store durations, the system will calculate an approximate storing time. Users are able to detect the current freshness by inserting the detector head into the food. The sensor on the detector head will extract information of nitrate concentration and return the freshness value in terms of safeness of eating. These data will be visualized on the screen.

3. General Requirements

This section describes the general requirements for the system as a whole. The requirements for each component of the system will be detailed in the following sections.

3.1 System requirements

The following table includes the general requirements for the system.

Requirement ID	Description
Req 3.1.1 P	The system consists of a microprocessor (SBC), a camera, a detector, a battery, a tri-colour LED and a touch screen.
Req 3.1.2 C	The device must recognize the food that the user wants to detect.
Req 3.1.3 P	The device must determine the food freshness that the user wants to detect.
Req 3.1.4 P	The screen must provide visual alert according to the estimated expiry date.
Req 3.1.5 P	The device must have a power on/off button.
Req 3.1.6 P	The price of our device must be under $200.
Req 3.1.7 P	The device must have a battery life of at least 24 hours.
Req 3.1.8 P	The device must have a camera and detector control interface.

Table 3.1: System Requirements

3.2 Functional Requirements

The following table includes the general requirements for the functionality of the system.

Requirement ID	Description
Req 3.2.1 P	Correctly determine and distinguish the food that the user wants to examine.
Req 3.2.2 P	Complete the image processing and output the estimated storing duration within a delay of 0.5 seconds.

Table 3.2: Functional Requirements

4. Firmware/Software Requirements

4.1 Software Overview

The core of the system consists of an image recognizer that can detect different fruits and vegetables as well as their freshness based on their physical appearance. Our language of choice for the project will be Python 3.5 as it is a highly abstract language having user-friendly data structures and the ability to accept flexible objects. To realize the image classifier we will be using Google's Tensorflow library which supports Python 3.5.

The classifier shall output the fruit/vegetable class along with a freshness value that is used as input to an estimator function. The freshness value is based on the appearance of the fruit/vegetable. Using this value, the estimator function is then able to determine an expiry date for the item as well as setting a reminder for the user. A database is used to store all of the fruits and vegetables in stock with their respective amount and expiry date attributes. Then, this information will be displayed on a GUI. This relational database will be queried and maintained using SQLite. The figure below depicts the software flow diagram for the system.

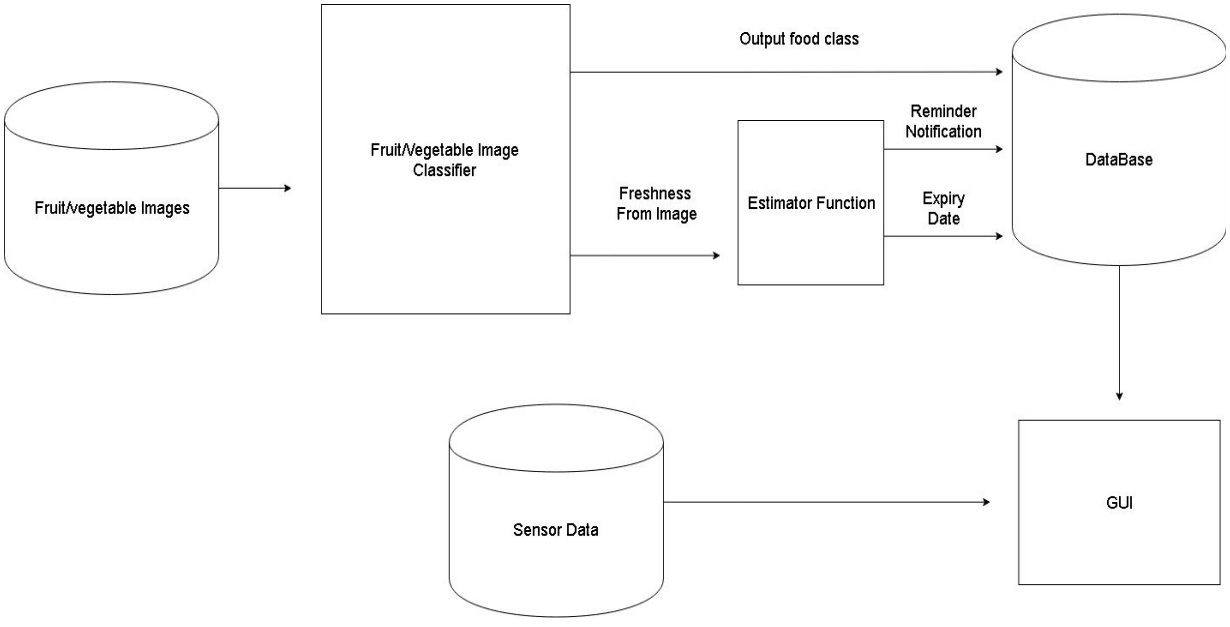


Figure 4.1: Software Diagram of the Overall System

4.2 General Requirements

The following table outlines the general requirements for the firmware/software system.

Requirement ID	Description
Req 4.2.1 P	All software will be written in Python 3.5 and C using the ISO/IEC 9899:2011 standard.
Req 4.2.2 P	Storage of data will be using a relational database with SQLite.
Req 4.2.3 P	Fruit images used for training shall be acquired through pictures taken from the camera and google images.
Req 4.2.4 P	The estimator function must be able to determine a future expiry date for the fruit or vegetable registered.
Req 4.2.5 P	The system shall use a GUI to display reminders before the expiry date.
Req 4.2.6 P	The software must be able to insert a new item into the database after fruit/vegetable classification.
Req 4.2.6 F	The SBC must be able to read in real-time sensor data through serial GPIO with no data loss.
Req 4.2.7 P	The SBC shall not be actively detecting sensory inputs when no data is being transmitted.
Req 4.2.8 F	The SBC must handle interrupts correctly and ensure the previous context is brought back to its proper state after being serviced.

Table 4.2: Software General Requirements

4.3 Image Classification Requirements

Image classification involves assigning a label to an entire image. Following are the software requirements that are essential for us to develop an image classifier.

Requirement ID	Description
Req 4.3.1 C	The Python Tensorflow r1.12 library shall be used for image classification.
Req 4.3.2 C	The fruit/vegetable classifier will utilize cv2 for image processing.

Req 4.3.3 P	Training images used for classification must be constrained with dimensions 100x100 pixels at 35.92 PPI.
Req 4.3.4 P	Training images must have no background. It shall only show the fruit/vegetable.
Req 4.3.5 P	Each fruit and vegetable class shall contain 50-100 training images for 360° rotations about each x, y, and z axis.
Req 4.3.6 F	The training images shall contain fruit and vegetable of 3 different varieties. All varieties shall have the same amount of images.
Req 4.3.7 F	The classifier must be able to recognize and identify more than 30 different types of fruit and vegetable images captured by the camera.

Table 4.3: Image Classification Requirements

4.4 Graphical User Interface Software Requirements

Following table shows software requirements for the graphical user interface on the LCD display.

Requirement ID	Description
Req 4.4.1 C	The python Tkinter library will be used to build the graphical user interface.
Req 4.4.2 C	The GUI shall be interacted by touch screen.
Req 4.4.3 P	The GUI must contain navigation arrows for scrolling through the main menu.
Req 4.4.4 P	The GUI shall have two buttons for selecting menu options, and cancelling/back functionality.
Req 4.4.5 P	The main menu shall contain a display to show all of the reminders for items that are close to their expiry date.
Req 4.4.6 P	The main menu shall also have an option that can be selected to display all of the currently registered food from the database.
Req 4.4.7 P	The GUI shall allow the user to delete items from the database.

Req 4.4.8 P	The main menu shall have an option to start “freshness detection” using the detector head.
Req 4.4.9 P	Sensor data should be read every second for 10 seconds 5 seconds after selecting the “freshness detection” option for fruit/vegetable.
Req 4.4.10 F	The sensor data shall be displayed on the GUI in real-time.

Table 4.4: Graphical User Interface Software Requirements

4.5 Performance Requirements

The below table lists the performance requirements for the overall software system.

Requirement ID	Description
Req 4.5.1 F	The GUI software must restart in the event of a crash.
Req 4.5.2 P	The GUI software must have a response time of less than 1 second when selecting and cancelling main menu options.
Req 4.5.3 P	Pre-processing of training images shall not exceed a time of 1 second.
Req 4.5.4 P	Fruit/vegetable images must be correctly classified within 0.5 seconds
Req 4.5.5 P	The image classifier must be able to identify fruits and vegetables with greater than 95% True Positive Rate (TPR) for each and every class [6].
Req 4.5.6 F	The image classifier must be able to identify fruits and vegetables with greater than 98% ACC (Accuracy) for each and every class.

Table 4.5: Software Performance Requirements

5. Hardware Requirements

5.1 General Requirements

The following table includes the general requirements for the system's hardware.

Requirement ID	Description
Req 5.1.1 C	The device must be eligible to use as a power supply.
Req 5.1.2 C	All electrical components shall be grounded.
Req 5.1.3 C	The device shall operate at a maximum voltage of 110V.
Req 5.1.4 P	The device shall be easily handled when on or off.
Req 5.1.5 P	The needle can be hidden inside the device and pulled out when needed.

Table 5.1: Hardware General Requirements

5.2 Sensor Requirements

The following table includes the sensor requirements for the system's hardware.

Requirement ID	Description
Req 5.2.1 C	The sensor can detect with stabbing into fruits or vegetables.
Req 5.2.2 P	The sensor can recognize the internal concentration of Nitrate and give us related parameter to determine the freshness level.
Req 5.2.3 P	The sensor gives final feedback to tell whether the food is still edible.
Req 5.2.4 P	Communication between the sensor and SBC will be based on wireless.

Table 5.2: Sensor Requirements

5.3 Processor Requirements

The following table includes the processor requirements for the system's hardware.

Requirement ID	Description

Req 5.3.1 P	The microprocessor, Raspberry Pi, must integrate a variety of systems to be a big system perfectly.
Req 5.3.2 P	The SBC is able to be programmed to sense and control objects.
Req 5.3.3 P	The SBC, Raspberry Pi, should be able to interact with a large array of outputs such as LEDs, motors and displays by responding to sensors and inputs.
Req 5.3.4 P	The center processors of this system should interact with cameras, sensors and display.
Req 5.3.5 P	The processor should have USB, Ethernet and HDMI port outputs.

Table 5.3: Processor Requirements

5.4 Camera and Display Requirements

The following table includes the requirements for camera and display.

Requirement ID	Description
Req 5.4.1 P	The camera should be able to scan and tell the food type with an approximate shelf time.
Req 5.4.2 P	The camera should support wireless data transmission.
Req 5.4.3 P	The camera size should be as big as a mobile phone camera and mounted on the back of the device.
Req 5.4.4 P	LED indicators must indicate the status of the device and give feedbacks.
Req 5.4.5 P	The touch screen of the device must show the status of scanning.

Table 5.4: Camera and Display Requirements

5.5 Wired and Wireless

In our system, there are two ways for signals transmission, both wired and wireless. The following table includes the wired and wireless requirements for system hardware.

Requirement ID	Description
Req 5.5.1 P	Camera image data should be transmitted by wireless technology.

Req 5.5.2 P	Sensor results should be transmitted by wireless technology.
Req 5.5.3 P	Communication between processor and display should be connected by a USB cable.

Table 5.5: Wired and Wireless Requirements

5.6 Power Supply

The following table includes the power supply requirements for system hardware.

Requirement ID	Description
Req 5.6.1 P	The power supply must be 5V.
Req 5.6.2 P	The power supply shall be a portable power solution that can be mounted inside the reminder.
Req 5.6.3 P	The power supply shall be easily recharged or replaced.
Req 5.6.4 P	The power supply shall guarantee long-lasting power and reliable performance.
Req 5.6.5 F	The size of the power supply shall not exceed 7cm*5cm*5cm.

Table 5.6: Power Supply Requirements

5.7 Performance

The below table lists the performance requirements for the hardware system.

Requirement ID	Description
Req 5.7.1 F	The whole system should be integrated and working properly (fluently).
Req 5.7.2 P	By scanning analysis, the display should tell the latest information about kinds, freshness and quantities of detected objects.
Req 5.7.3 F	The whole scanning and display process should occur within 5 seconds.
Req 5.7.4 F	Updated information must get any time from the display when scanning.

Table 5.7: Hardware Performance Requirements

6. Safety and Sustainability

6.1 Safety Requirement

The below table lists the safety requirements for the overall system.

Requirement ID	Description
Req 6.1.1 F	The device shall have no sharp edges.
Req 6.1.2 F	The device shall use at 0 °C - 40 °C.
Req 6.1.3 F	Sensor and microprocessor need to be fixed firmly and not be exposed to the surface.
Req 6.1.4 F	All devices and cables have to be installed correctly according to the Canadian Electrical Code.
Req 6.1.5 F	The screen shall not be broken easily.

Table 6.1: Safety Requirements

6.2 Sustainability

According to the cradle-to-cradle design principle, we use the principles of ecology, economy and equity [10]. Our project is composed of several main parts, which are screen, sensor and control system. In order to make our project sustainability, the “cradle to cradle” is satisfied in each part.

Fewer materials and less energy is a cradle-to-cradle, industrial ecology, products are not used for a finite length of time[8]. With regards to the minimum waste, we will do the research for individual parts to ensure each part we used are recyclable and eco-friendly. Moreover, the idea we use the wireless application is sustainability, because it decreases materials like wires might be waste.

Our device is low cost and competitive in the market. We will share electrical parts, screen and some components, such as Raspberry Pi with other projects to minimum cost. Nowadays, products that can detect freshness are not more systematic and versatile than our product. And some people have a need for this, but there is no suitable product.

Social equity, restorative environmental impact, and profitability are the three primary pillars in this project [7]. During decision-making progress, we will keep fairness for those three pillars in the project.

7. Engineering Standards

Since this device is used by humans, we will follow the standards to ensure health and safety.

7.1 Electrical

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

Requirement ID	Description
IEC 61558-1:2017	Safety of transformers, reactors, power supply units and combinations thereof - Part 1: General requirements and tests[5]
IEC 60086-1:2015	Primary batteries - Part 1: General[4]

Table 7.1: Electrical Standards

7.2 Environmental

The following are the environmental standards used during the development:

Requirement ID	Description
IEC 60050-904:2014	International Electrotechnical Vocabulary (IEV) - Part 904: Environmental standardization for electrical and electronic products and systems[3]
CAN/CSA-ISO/ TR 14062-03 (R2013)	Environmental Management - Integrating Environmental Aspects into Product Design and Development (Adopted ISO/TR 14062:2002, first edition, 2002-11- 01)[1]

Table 7.2: Environmental Standards

8. Conclusion

This document clearly outlines the general system requirements as well as the specific requirements for software and hardware components. This document also provided safety and sustainable requirements and Engineer standards we are going to follow during the design and prototyping of this project. In addition to the above requirements, we also provide an acceptance test plan for future record. This document will provide a solid guide to assure the satisfaction of functional requirements for each step.

FooMinder is designed to be a food reminder system consists of a camera, a detector and a microprocessor. It uses image processing to detect the food freshness and set a reminder to alert the user of the expiry date. The detector can be used to detect the current freshness of the food. All data acquired during the processing will be visualized on a touch screen which is also a controller for the camera and the detector. Our intention is to provide a small, user-friendly and cost-friendly device.

As a group of engineers that provide solutions to real discovered problems. We aim to give the effort for reducing food waste and meet the world's sustainable goals.

9. Appendix

9.1 Proof of Concept Deliverables

For the proof of concept deliverables, the following deliverables will be presented.

- Food (fruit and vegetable) recognition performed with the microcontroller and laptop
- Touch screen circuit demo working
- Proof of concept sensor detecting

9.2 Acceptance Test Plan

Our acceptance test plan is broken down into two sections: software and hardware. The table shown below depicts a list of categories of test cases to be conducted.

<table>
<tr><td colspan="2">Software Tests</td></tr>
<tr><td>1 - Fruit/vegetable Classifier</td><td>Comments:</td></tr>
<tr><td>100 fruit images with 95% TPR:

Pass ☐ Fail ☐</td><td rowspan="3"></td></tr>
<tr><td>100 fruit images with 98% ACC:

Pass ☐ Fail ☐</td></tr>
<tr><td>Able to correctly classify images within 0.5 seconds:

Pass ☐ Fail ☐</td></tr>
<tr><td>2 - GUI operations</td><td>Comments:</td></tr>
<tr><td>Able to navigate the main menu with arrow keys:

Pass ☐ Fail ☐</td><td></td></tr>
</table>

Able to select/cancel main menu options with “YES” and “NO” keys: Pass ☐ Fail ☐	
Able to display updated fruit/vegetable database information when selecting “Display Items” option: Pass ☐ Fail ☐	
Able to display freshness values when selecting “Start Detection” option: Pass ☐ Fail ☐	
3 - DataBase Operations	Comments:
Insertion of new items into the database is successful: Pass ☐ Fail ☐	
Deletion of item from the database is successful: Pass ☐ Fail ☐	
4 - Sensor Data	Comments:
Sensor data can be read by SBC: Pass ☐ Fail ☐	
SBC start reading in sensor data 5 seconds after the start of detector head insertion: Pass ☐ Fail ☐	
Sensor data can be displayed on the GUI in real-time. Pass ☐ Fail ☐	
5 - Expiry Date	Comments:

The expiry date generated by the image classifier is in the future. Pass ☐ Fail ☐	

Table 9.1: Software Test Plan

Hardware Tests	
1 - Device Operations	Comments:
The small screen can display letters, digits and picture. Pass☐ Fail☐	
LED should light on when the camera or detector is on. Pass☐ Fail☐	
To prevent the device from overheating during use, cooling should work regularly. Pass☐ Fail☐	
Proper and stable functioning by the power supply. Pass☐ Fail☐	
2 - Sensor Operations	Comments:
The sensor can detect the concentration of Nitrate, and show the results on display. Pass☐ Fail☐	
The allowed processing time is 1 second. Pass☐ Fail☐	
The needle of the sensor can be pulled out from the device and insert back easily. Pass☐ Fail☐	

3 - Camera Operations	Comments:
The camera works properly when the power switch is on, and it can be switched between ON and OFF easily. Pass☐ Fail☐	
The camera scans and does image analysis, it shows us the information on the display. Pass☐ Fail☐	
4 - Interface Operations	Comments:
The buttons on the device can be pressed and work regularly (i.e. select numbers and letters on the display). Pass☐ Fail☐	
The small screen displays the corresponding contents without messy code. Pass☐ Fail☐	
5 - Performance Operations	
The whole system can be integrated and functioning properly in a short period of time (a few seconds), then displays us the accurate and updated information of the detected objects. Pass☐ Fail☐	
When the food will expire after one day, there is a notice message shown on screen. Pass☐ Fail☐	

Table 9.2: Hardware Test Plan

Reference

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

[2] Grosso, M., & Falasconi, L. (2018). “Addressing food wastage in the framework of the UN Sustainable Development Goals”. Waste Management & Research, 36(2), 97-98.

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

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

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

[6] Narkhede, Sarang (2018). “Understanding Confusion Matrix”. Retrieved from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[7] “Raspberry Pi 2, Model B”(2015). Retrieved from http://www.adafruit.com/pdfs/raspberrypi2modelb.pdf

[8] Robinson, W. D., ed (1986). “The Solid Waste Handbook”. New York: Wiley. James W. Patterson

[9] Usubiaga, A., Butnar, I., & Schepelmann, P. (2018). “Wasting Food, Wasting Resources: Potential Environmental Savings Through Food Waste Reductions”. Journal of Industrial Ecology, 22(3), 574-584.

[10] W. McDonough and M. Braungart (2002). “Cradle to cradle”. New York: North Point Press