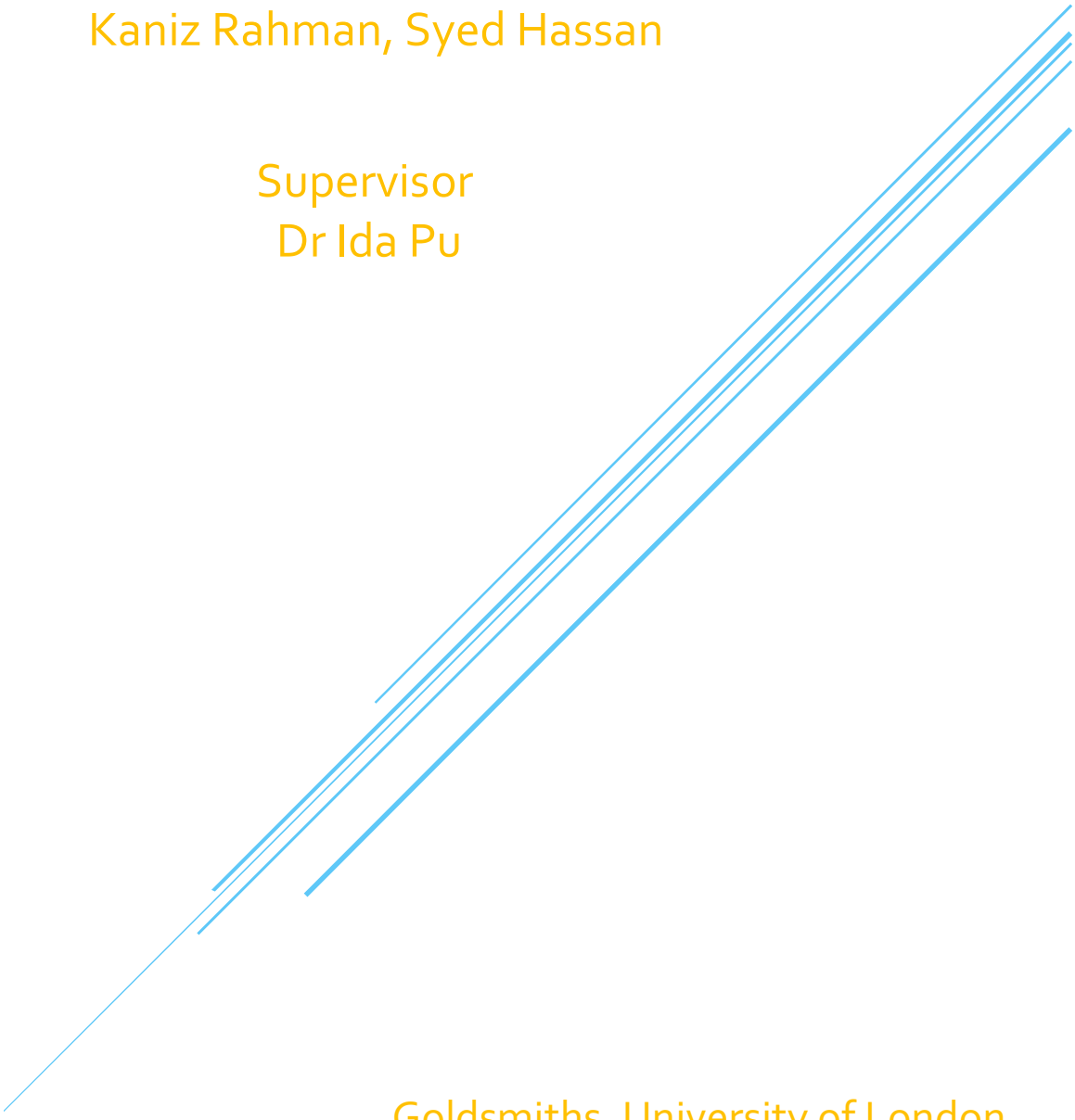
DEVELOPMENT OF A FRIDGE INVENTORY SYSTEM TO REDUCE FOOD WASTE

Done by

Grzegorz Rybak, Keefe Chan, Laila Majeed,
Kaniz Rahman, Syed Hassan

Supervisor
Dr Ida Pu



Goldsmiths, University of London
Computer Science, Software Projects

ABSTRACT

The goal of our project is to develop Scidge, a system that combines both hardware and software to create a fridge inventory to keep track of the quantity of food and when it expires. The inventory then can be viewed and managed through the application.

The main problem that we are trying to solve is food wastage. It is one of the biggest problems that we face in the 21st century as globally, almost one third of edible food that are meant for human consumption is wasted every year. It has been ascertained by a study conducted by the National Food Standard Agency (FSA, 2017) that in the UK, we throw away 7 million tonnes of food and drink from our homes every year, the majority of which could have been eaten.

ACKNOWLEDGEMENTS

The entire team would like to express our gratitude towards our supervisor, Dr. Ida Pu, for her patient and friendly approach to guiding us through the entire software project and offering valuable suggestions which guided this project from its idea conception to a successful completion.

We would also like to thank Konstantin Leonenko for patiently guiding and helping us with the equipment in the Hatchlabs.

Lastly, we would also like to thank the Department of Computing at Goldsmiths, University of London, for providing access to Hatchlabs and for the opportunity to work on this project as a team.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

According to the Food Standard Agency (2018), in the UK alone 7 million tonnes of food and drink are thrown away each year, and with the global population set to rise to over 9.5 billion by the year 2050, the initial pressure facing the food system is maximised. The main reasons behind the sheer volume of food wastage is down to; people buying more than they need (Smithers, R 2013), lack of clarity on the way expiry dates work and storage of certain items, coupled with the contemporary schedules of work/studying and appointments, resulting in people often changing their food preparation plans, or failing to remember to use food items on time (Conserve Energy Future, 2018).

Our motivation was driven by the reasons behind the sheer volume food wastage and playing our part in helping the country cut down on wasting food and thus reducing the carbon footprint (Dibenedetto, B 2013). The initial idea is to create a unique, innovative device designed to enhance the life-style and well-being of contemporary individuals who are interested in maintaining a healthy lifestyle but do not have the time to do so. The busy individual who may find it difficult to keep up with the groceries, meal ideas and the expiry dates of the items they already own and thus fall in line with the millions who throw away food to waste. However, many could argue this has already been dealt with, with the innovation of Smart Fridges - that have been on rise since the year 2000 (Ridden, P 2011) which aims to learn the users pattern such as determine how often you use milk and automatically add items to shopping list, help map out meal plans and help manage your grocery budget (Ramey, K 2017) but these products are arguably very expensive (costing from £2000). Debatably, with the rise of inflation, active households are more likely feeling the pain after the cost of living has rose to a six year higher of 3.1% since November 2017 (Guardian, 2017) and with the national living wage rising by only 4.7% from last year (Munbodh, E 2018), the financial strain on the working class/student's lifestyle is inevitable. Our aim is to provide our target market a cheaper alternative with the same quality concepts and specification obtained in many smart fridges.

1.2 SCOPE AND OBJECTIVES

The main objectives of this project is to create a system that is usable, intuitive and functions well consistently which the target groups and final beneficiaries of the project will benefit from. Our chief concern is that our software be user friendly which means making the software and hardware design effortlessly navigable and simple to use. We want our "Scidge" to be able to not only contribute to the fight against food waste but also reducing CO_2 emissions and food prices in the long run. Ensuring the hardware component is integrated seamlessly with our software to improve the efficiency of users lives by monitoring and managing their food.

The software is responsible for a host of functions. Our software must be able to identify the item of which barcode has been scanned, produce an estimated expiry date/or allows the user to manually input the data and displays it on the web application linked. Alerts of when the item is near to expiry and the option to remove the item from the 'virtual fridge' must be viable. As well as outputting accurate weight when the scale option is selected. For a more detailed functionality see Chapter 5 - Design.

1.3 COVERAGE

The report contains seven chapters, excluding the introduction:

Chapter 2 – Literature Review: The literature review of the application framework and Arduino environment.

Chapter 3 – Project Development Process: The development process throughout the entire course

Chapter 4 – Analysis: The requirements analysis of the application to be developed is given in this chapter.

Chapter 5 – Design: This chapter describes the design of the application, database and the hardware.

Chapter 6 – Implementation: This chapter describes how the application and hardware is implemented.

Chapter 7 – Evaluation and Testing: Evaluation of the project which is based on functional testing, expert evaluation, the tests conducted on the software and hardware, findings and fixes are presented in this chapter.

Chapter 8 – Conclusion: Shows the main achievements of the project and proposed future work.

2 LITERATURE REVIEW

2.1 BACKGROUND

2.2 THE APPLICATION FRAMEWORK

The software architecture consists of several tools and frameworks that form the final web application. All of the technologies mentioned are open-source and considered as well-established in the web development industry. ("State of JavaScript 2017")
The hierarchy of the main technologies are discussed in the following subsections.

2.2.1 NODE.JS

Node.js is an open-source server environment that lets the developer write back-end code in JavaScript. This is because it is using Google's V8 JavaScript engine, which in fact is a JavaScript compiler written in C++, and allows to run JavaScript code in the system environment outside of the browser.
For the purpose of the Scidge software development, Node.js was used because of package manager called NPM (Node Package Manager). It allowed the team to download JavaScript extensions enhancing the web application features (like Quagga.js for barcode image recognition) and manage all the dependencies of the JavaScript files in the project. In fact, it would be significantly difficult and time-consuming (if not impossible) to configure such a complex front-end technology stack without NPM as hundreds of dependencies of the used framework would need to be reviewed and manually downloaded.

2.2.2 WEBPACK.JS

Webpack is a build tool that bundles the JavaScript files into static assets that can be served by a server. To understand the necessity of Webpack in the project it is important to compare the modern and the old web development. A JavaScript file can be referenced by an HTML file using the following code:

```
1  <head>
2    <script src="script.js" type="text/javascript"></script>
3  </head>
```

Figure 2-1 An example of a basic JavaScript file import

However, when there are multiple files with a complex dependency hierarchy, there appears a need for a Method Resolution Order (MRO) algorithm as it is in other programming language dealing with multiple inheritance, like in Python having a "C3 linearization" algorithm (Simionato, 2003). Additionally, the above way of referencing files is asynchronous; the files are tried to by processed by the JavaScript engine in all at once and this causes another problem: the referenced files need to be loaded synchronously to prevent a situation when a file calls a method from another file that has not yet been imported.

That is why a concept of JavaScript "Modules" has been established. It allows to "export" certain functions of a JavaScript file which then can be "imported" by another.

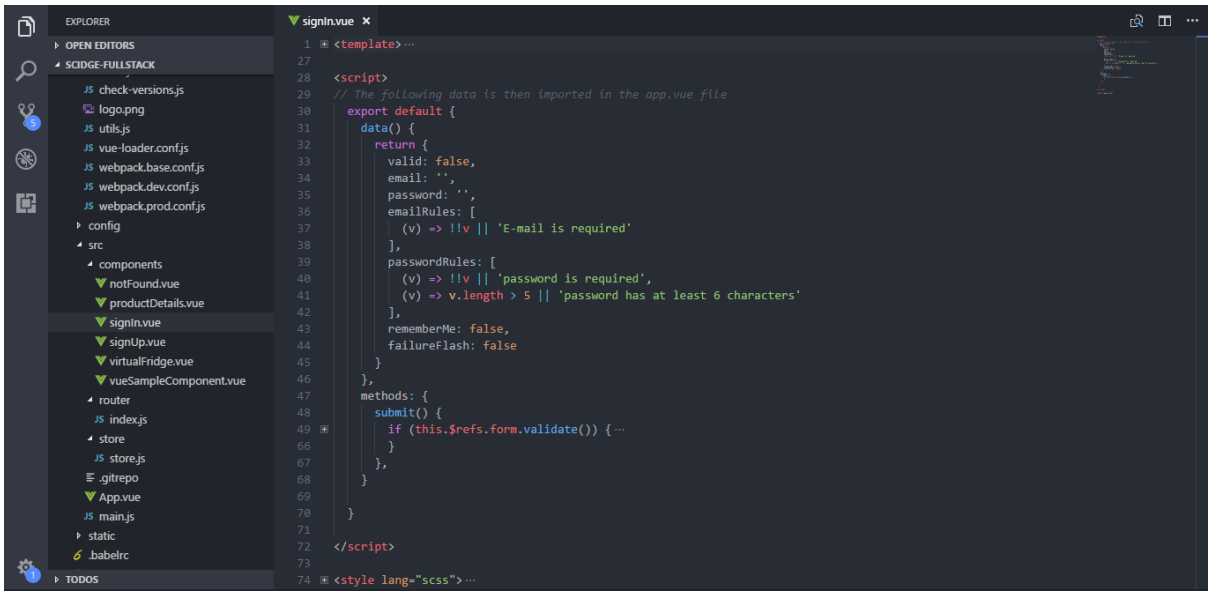


Figure 2-2 A typical .vue component exporting its business logic. This logic is then used in a file in the next figure

Figure 2-3 A root-view '.vue' file which also exports its logic, shown in the next Figure

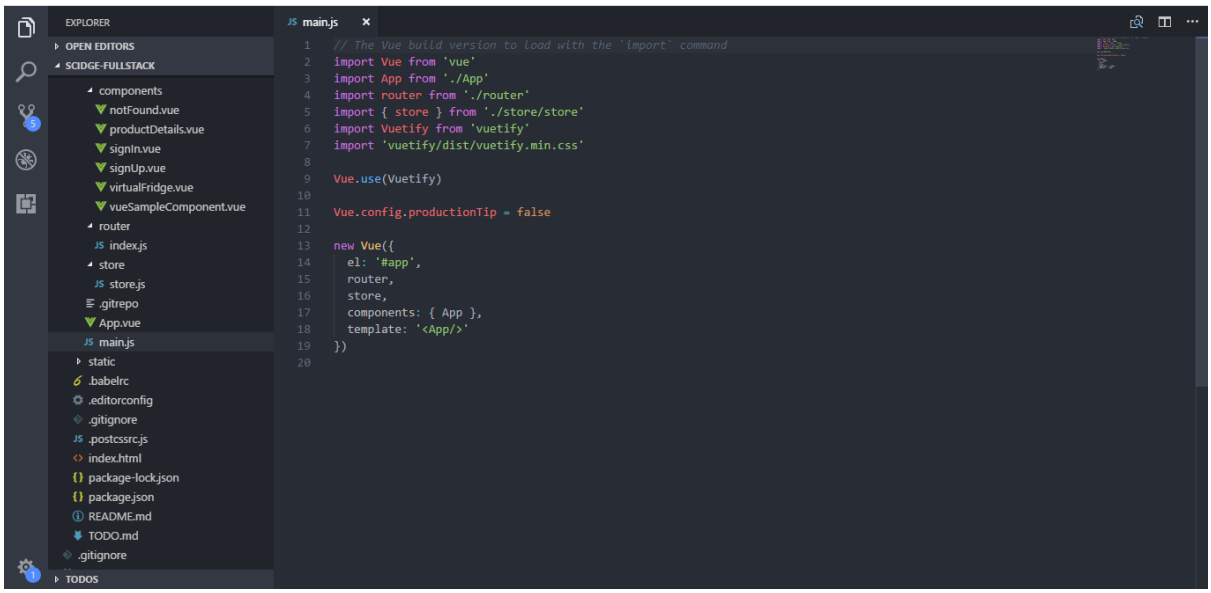


Figure 2-4 The hierarchy of the dependencies from various logic gathered in the main.js file, which initiates the Vue object accessible through every file in the chain

Webpack leverages that concept and scans all the hierarchy of the imported and exported modules in all the JavaScript files in the front-end's workspace in order to "bundle" all the files in a small set of minified JavaScript files (usually one or two files). These files can be now served by the server.

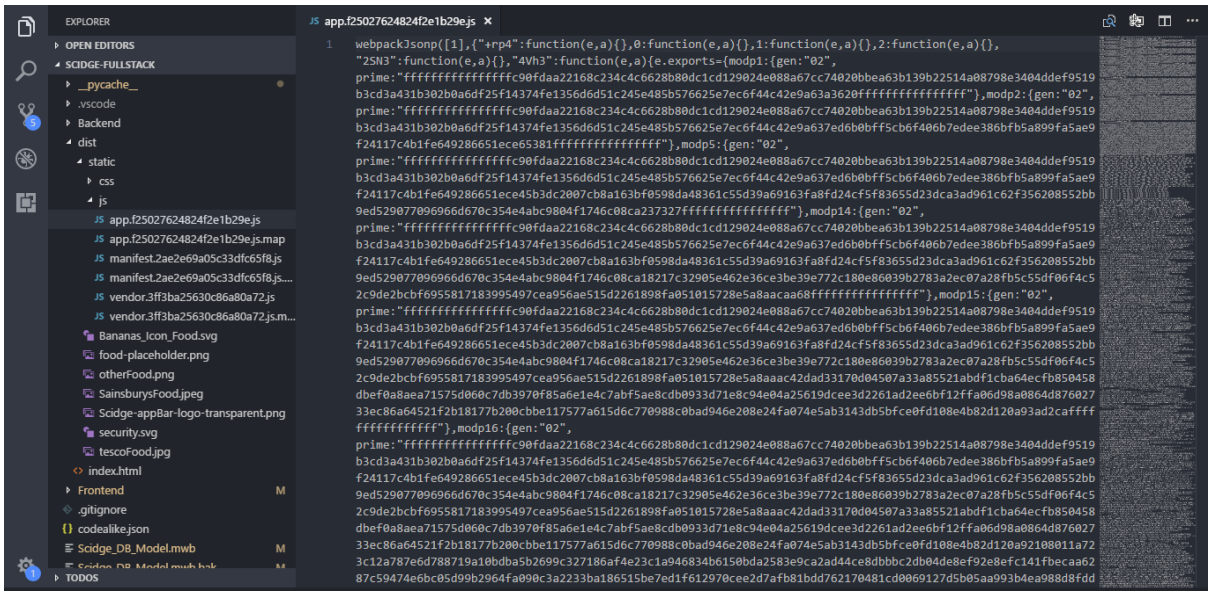


Figure 2-5 All the JavaScript files in the project were bundled into a set of 6 minified static JavaScript assets

2.2.3 BABEL.JS

Babel is a JavaScript "transpiler"; it analyzes the code written in any ECMAScript edition and reproduces it to the ECMAScript 5 edition version, which is the most commonly supported in the browsers. This is because of the fact that JavaScript is the most widely known implementation of the

ECMAScript and that the new versions of the ECMAScript – ES 2015, ES 2016 & ES 2017, which greatly enhance the language capabilities, are in fact not commonly implemented across all the browsers. To avoid cross-browser compatibility problems and enhance the efficiency of the code, the team has included Babel in the build process at the stage of bundling the Webpack assets.

2.2.4 VUE.JS

Vue is a widely popular JavaScript framework providing a vast set of features accelerating the development of a web app.
By using the Vue, the Scidge app is developed not as ".html" pages containing JavaScript & CSS references, but in a ".vue" components that contain all the HTML, JavaScript and CSS logic for the part of the app.
It changes the way the web application is developed, as the developer can create various-sized, reusable or nested components inside every page. In this situation, it is easy to stop considering the web app as a set of pages, but as a list of "views" formed from multiple, differently-sized components (hence the naming of the framework).

Figure 2-6 example of a ".vue" component. This component only consists of an h1 heading with its very own customizable logic and styling. Therefore, it can be easily reused in many places inside a bigger, maybe even page-sized component

2.3 ARDUINO

Arduino is an open source electronics prototyping platform based on flexible, lightweight easy to use hardware and software. It is tool to develop computer systems that can sense and control the physical environment more than the normal computers that we commonly use. It is an open source physical computing platform based on a basic microcontroller board and a development environment for writing programs for the board. The software is written in C or C++. The single chip microcontroller has a microprocessor which differs over the different boards.

Figure 2-7 Several types of Arduino Boards

A range of Arduino Boards exist (as shown in Figure 2-7), which has different specifications which suits different purposes. Although there are many official Arduino versions, there are also unofficial Arduino based platforms as both the manufacturing process and the programs are open sourced. So, anyone can freely build their own Arduino clones without paying royalties.

2.3.1 ARDUINO IDE

Arduino integrated development environment (IDE) is a programming environment that allows users to write different sketches (Arduino programs) and load them into any Arduino microcontroller. After the sketches are compiled, it is translated to the assembler language and uploaded into the microcontroller. The Arduino IDE has a built-in code parser that will check the sketches before sending it to the Arduino board.

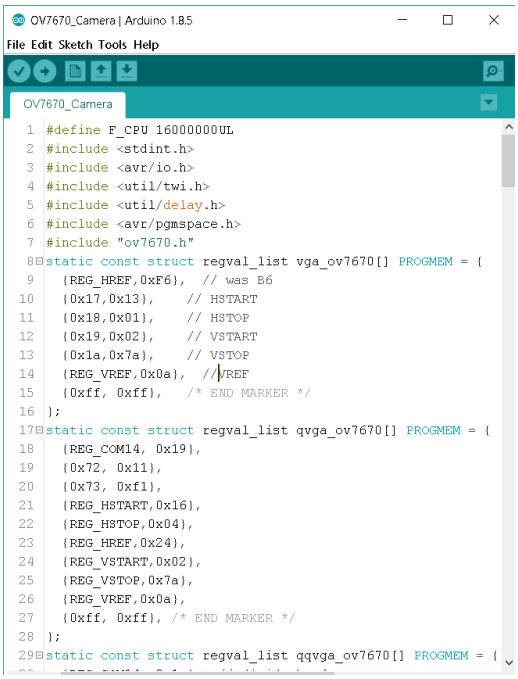


Figure 2-8 Arduino IDE sketch example

Thus, it is an easy to use software which is supported by all the Arduino boards. Arduino also has a large community and a huge assortment of libraries which makes the learning curve gentler. However, the libraries are optimized for easy usage and not efficiency. It is also unsuitable for industrial use and is limited for skilled programmers.

2.3.2 ARDUINO SENSORS

Arduino belongs to embedded systems, which means it communicate with its environment through sensors. These sensors are able to gather data of many different variables such as temperature, motion, pressure to light intensity. By combining several different sensors in a system, we can monitor and extract information from our physical environment. Which we can then provide appropriate responses to change or alert the system based on the relevant inputs.

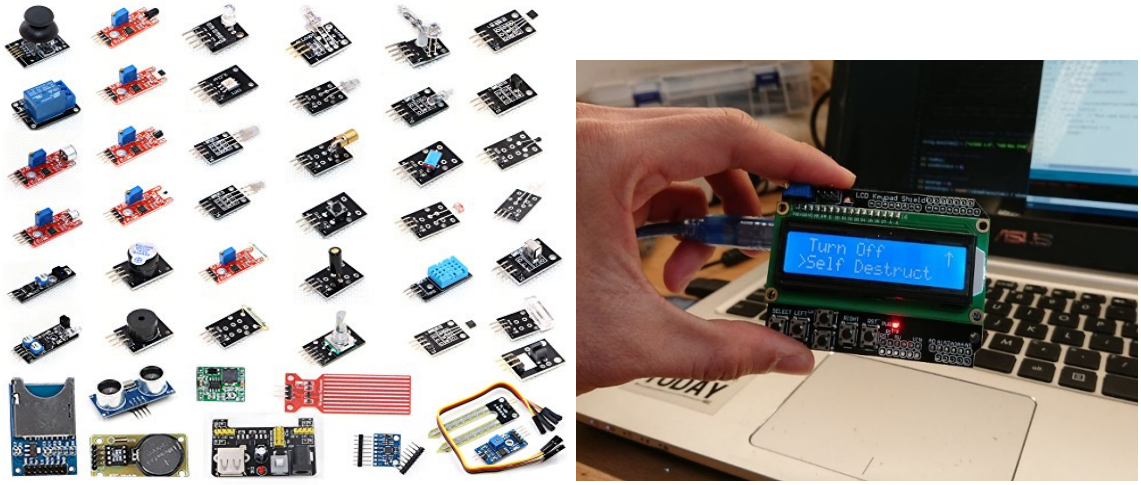


Figure 2-9 Several types of Arduino Sensors

As Arduino becomes more widespread, more and more sensors were designed and manufactured. At the same time, the need of measuring different variables led to better and more accurate control and monitoring systems.

3 PROJECT DEVELOPMENT PROCESS

3.1 HUMAN RESOURCE MANAGEMENT

The team has started the development in term 2 from discovering the most efficient approach of allocating the human resources. After the group meetings in the first 2 weeks of the second term, the created 5 separate areas of the development and assigned each member of the team as a leader of one area. In the second week of the term 2 each of the team members have decided which areas they would like to lead and what is their reasoning for it.

No.	Team member	Which area chosen to lead	Decision argumentation	Area's main objectives (if applicable)
1.	Keefe Chan	Hardware development	Keefe has chosen to create the hardware of the "Scidge" system; He was the person in the team with the biggest knowledge and experience in creating a hardware and through the first term, he was the main person to shape the vision about how should the Scidge scanner fit into the project.	
2.	Laila Majeed	Software back-end development	Back-end area needed to be developed in much bigger correlation to the other areas, mainly the software Front-end one and Laila had the highest record of the regularity in the communication in the team.	- designing and implementing a database - Researching and discovering ways to get information about the scanned product from external APIs - creating an application server for connecting with the database data on the front-end
3.	Grzegorz Rybak	Software front-end development	Grzegorz already had an experience in creating full-stack web applications so the project's development could leverage on his knowledge on both how to create a user interface and how to connect the front-end with the back-end. Additionally, if Laila ever needed any help with the back-end, he could easily help her given the close relation between those two areas.	- creating the application layout - designing the user interface - developing a good UX - connecting front-end with the back-end - creating the software version of scanning the product's barcode and dynamically rendering the data acquired from the database

4.	Syed Hassan	External promotion & website development	Desire to use and expand the knowledge gained from the Web Development module from year 1 and self-teach creativity skills such as use of Photoshop to design to the logo and design of website. .	- Creating a website showcasing the project - Using his research behind the project, Syed designed the logo for the brand.
5.	Kaniz Shaba Rahman	Testing	Shaba described herself as not so technically confident as the rest of the team and was worried she won't be able produce a quality code that could be used in the project. At the same time, she wanted to help the team, get the insight and a knowledge of the technology used in the project and use the fact English is her first language (as opposed to most of the team) in any administrative tasks related to testing (e.g. in user testing)	- Researching and discovering the testing needed - Software test cases - Hardware test cases - User tests survey and analysis of data
6.	Grzegorz Rybak	Team leadership & project management	The team has decided to let Grzegorz continue carrying the tasks of the group leader. Grzegorz has also agreed to stay on the leader's position for the good of the project.	- Resources management - Time tracking - Administering project git repository. - Notifying the rest of the team members about tasks and dates

There are 2 main efficiency factors in favor of the chosen approach of allocating the human resources:

1. Smaller possibility of tasks being "blocked".

Because of dividing the development into 5 areas, the tasks have become more separated and distinct from each other, therefore, there were smaller chances that at any point of time multiple team member work on a range of connected tasks so that one cannot continue the task until the second person finishes. The only chance of that happening was in the software development between the front-end and the back-end areas. This was resolved firstly by allocating the most communicative member in the team to the back-end area and secondly, by making the front-end area leader more involved in the back-end development process, therefore, taking on the work from the back-end tasks that were, or had chance to being blocked.

2. Avoidance of relying on communicational problems of the team.

The team has proved to have problems in a regular communication between the team members in the previous term. While it was desired by all the projects participants to have this problem resolved, the chosen human resources management approach decreased the chances of the project development being damaged should the communication problem have not been fixed.

3.2 METHODOLOGIES

3.2.1 SCRUM TECHNIQUE DEVELOPMENT ATTEMPT

Once an optimal human resources strategy has been established, the team started to implement a project management methodology. The initial choice was an agile, SCRUM technique. The team leader, has taken a role of the SCRUM master and designed a work area in the 'Trello' project management system. He created a "Kanban board" divided into main development 5 areas (the ones mentioned in the previous section) where each of the leaders was asked to insert all their tasks in. Next, he created 4 'Sprint' boards for the duration of the whole term 2. Each "Sprint" was adequate to a 2-weeks-time of the development after which the team was supposed to deliver a "potentially shippable" part of our system (end of the sprint guidelines reference). The strategy was the following: Kanban board served as a pool of all the tasks to be done in the project from each of the 5 development areas. Then, a desired amount of the tasks from the Kanban was chosen by each of the leaders. Those tasks were placed in the "Sprint" board at the beginning of every new "Sprint" and aimed to be done within the period of the "Sprint". At the end of every "Sprint" the team was meant to review the project's progress and evaluate the status or the result of every task.

The SCRUM methodology seemed to have 2 main benefits that appealed to the team:

3.2.1.1 EFFICIENT MICRO-MANAGEMENT OF EACH AREA BY THE AREA LEADERS.

This system had a high chance of leveraging that every team member was a leader in a development area. It is because with SCRUM every member adds new tasks and break them down into their own development areas, which previously needed the group leader. In the best case, that could increase the efficiency of the team factor of 5. It could also help in discovering and forming more informative tasks to do as each of the leaders of the development area had a bigger chance of having the knowledge of the area needs than the group leader, who needs to share his attention across all the areas.

3.2.1.2 DECREASE OF TENSION-CAUSING SITUATIONS ON THE LEADER – MEMBER LEVEL.

SCRUM allowed each team member to have them decide their set of tasks and distribute its execution along the sprints in their own liking. This could significantly reduce the chance of situations causing a tension between the team members when the group leader assigns the tasks and deadlines to the team members and expects them done in a given time.
With this methodology, the deadline's distribution seemed to be fair and uniform for each project participant – the only one deadline was the end of the sprint. Additionally, this had a chance of reducing the stress on the leader to distribute the tasks in a fair manner to all the team members.

In practice, however, the SCRUM technique proved to be very inefficient for various reasons and the group leader has decided to abandon it. Below is the list of the arguments and their explanation:

3.2.2 AREAS LACKING FOR SCRUM

3.2.2.1 DEVELOPMENT AREAS HAVE BEEN POORLY ADMINISTERED

The assumption of having leaders in every area, while beneficial in terms of the autonomy of actions and making them less prone to being "blocked" by other team members, contained a very big drawback: from then on, every team member needed to handle the administration activities previously handled only by the group leader. It was evident the group leader needed to handle those activities well

(after all, that is why a leader was chosen and that was their main duty in that role). However, assigning this task to everyone in the team, quite reasonably, revealed not everyone was equally skilled in it. This was not acceptable as all 5 areas were crucial for a successful project completion and the team could not allow for having particular areas with no tasks recorded despite having a common knowledge there is some work that needs to be done in them.

3.2.2.2 INABILITY TO JUDGE TEAM'S PROGRESS

While some of the areas were empty with no tasks altogether, the level of details of tasks in the areas that had them has drastically differed. This made it significantly hard for the group leader to estimate the group's progress at a given time. What is worse, it also made the team lazier. Because the real number of tasks to be done was not easily noticeable, it was easy for the team to get a false feeling that there is much less tasks to do than there was in reality.

3.2.2.3 DIFFICULTY IN FOLLOWING SCRUM MEETINGS REQUIREMENT

SCRUM technique required regular meetings with the team to work efficiently as it is one of the principles of the agile development. However, the team has found it particularly challenging to organize such meetings. Meeting in person often times proved to be impossible due to having only one moment during the week when the team could meet (supervisor meetings). The team has also experimented with phone and computer conferences but then realized of ineffectiveness as the team consisted of members that had extremely limited contact with the rest of team and, as a result, could not be considered as reliable in meeting organization. This made logistical problems and further decreased the productivity of the team in this methodology.

3.3 REPOSITORY-FOCUSED PROJECT MANAGEMENT.

After discovering the problems of the Scrum technique, the team started establishing a new project management system.
Due to the reduced progress the group leader has decided to build the new methodology that is quick to implement into the project and is based on the experience gained from the first term. It was decided to create a project management git repository "SCIDGE" with an "activity log" to record the contribution of every team member and a "TODO list" that included all 5 development areas of the project.

From then, the repository was regularly managed by the group leader overlooking the TODO and Activities log files, giving the rest of the team notices and updates about new tasks needed to be added and activities to be recorded on a weekly basis. Additionally, he also added a documentation of the particularly important group meetings and project deliverables such as the group's presentation. From March, this repository was being sent to the supervisor at a weekly basis with its current state at a given time.

4 ANALYSIS

4.1 SYSTEM REQUIREMENTS

In this section, we present the Functional and Non-Functional requirements of both our software and hardware. Functional requirements define the properties and functionality that our system must have to perform successfully. Non-Functional requirements define the qualities and criteria that we can use to judge the operation of our system.

4.1.1 SOFTWARE

4.1.1.1 FUNCTIONAL REQUIREMENTS

One of the most crucial requirements of the software was that it needed to deliver uniform User Experience (UX) across all the devices and platforms as only then the web app would leverage the fact of not being developed in multiple platform-specific languages and gain bigger user audience.
The table of the functional requirements is presented below. The team has decided to distinct the requirements that form the "core" functionality of the Scidge software and the "regular" ones determining the right work of features that can be built on top of the core requirements

ID	Description	Core/regular requirement
SFR01	System must scale its User Interface (UI) correctly across all the devices and their screens width & resolution	Core requirement
SFR02	Users must be able to sign up	Core requirement
SFR03	User must be able to sign in	Core requirement
SFR04	User must have their own, non-dependant space in the database (that is, User's data cannot be designed in a way so that it is possible to be altered by other user's actions)	Core requirement
SFR05	System must show all user's products divided into 2 categories: general and "near expiry"	Core requirement
SFR06	User must be able to edit the product's details	Core requirement
SFR07	User must be able to scan a barcode of the product	Regular requirement
SFR08	User must be able to delete the product from their products list	Core requirement
SFR11	User's credentials must be stored in a secure manner	Core requirement
SFR 12	System must be able to show user recipe propositions based on the products they have stored in the database	Regular requirement
SFR 13	User must be able to find a recipe proposition by typing the recipe name or any ingredient	Regular requirement
SFR 14	System must show the correct recipe upon click on a given recipe by the user	Regular requirement

SFR 15	User must be able to add new products to their shopping list	Regular requirement
SFR 16	User must be able to delete any product from their shopping list	Regular requirement
SFR 17	User must be able to store more than one product of the same type in the products list	Regular requirement

Table 4-1 Software Functional Requirements

4.1.1.2 NON-FUNCTIONAL REQUIREMENTS

The crucial non-functional requirement discovered by the team after sharing the conceptual prototype is that the users should find the able as m easy-to-follow medium of displaying the products they scanned via the fridge scanner. The table of the functional requirements is presented below. Requirements are labeled as "crucial" and "non-crucial" depending on how important were highlighted by the participants in the Scidge Survey from the first term [appendix 1]

ID	Description	Crucial/Non-crucial requirement
SNFR01	User should be able to easily find all the desired food information with minimum of actions	Crucial
SNFR02	User should have a feeling of interacting with an application adjusted to a mobile device screen	Crucial
SNFR03	User should be able to easily locate and distinct any food product in their virtual fridge	Crucial
SNFR04	User should find the Scidge application visually appealing	Non-crucial
SNFR05	User should be able to hide sensitive data (for example their password and user information)	Non-crucial
SNFR06	User should be able to scan a product via the app in a relatively fast and convenient manner	Non-crucial
SNFR07	User should find the shopping list easy to use and useful	Non-crucial
SNFR08	User should find the recipe suggestions relatively useful	Crucial
SNFR09	System should show the recipe search results in a way that is clear for a user to process the information	Crucial

Table 4-2 Software Non-Functional Requirements

4.1.2 HARDWARE

4.1.2.1 FUNCTIONAL REQUIREMENTS

ID	Requirement	Crucial/Non-crucial requirement
HFR01	Arduino board must be able to retrieve information from the camera module, load cell and load amplifier	Crucial
HFR02	Scidge must be able to retrieve information from both sensors and send it to the Arduino IDE through USB	Crucial
HFR03	Scidge must be able to connect to WiFi through the ESP8266 module	Non-crucial
HFR04	Scidge must be able to send information from the relevant modules to the database through WiFi	Non-crucial

Table 4-3: Hardware Functional Requirements

4.1.2.2 NON-FUNCTIONAL REQUIREMENTS

ID	Requirement
HNFR01	The design of Scidge must be easy enough for anyone to use
HNFR02	Scidge should be portable
HNFR03	Scidge should be able to accommodate the weight of commonly found food without compromising on accuracy
HNFR04	The camera module should be able to capture image of sufficient quality for the application to read the barcode and recognise the food item
HNFR05	Scidge should have a fast response time
HNFR06	Scidge must be designed to be able to accept new operations and features
HNFR07	The cost of the entire system should be below £40

Table 4-4: Hardware Non-Functional Requirements

4.2 STAKEHOLDERS

Whilst exploring our likely stakeholders we found that the households which produces the most food waste are mostly between the age of 16-45, young professionals and households with children (Tucker & Farrelly, 2015, p.8) as there are time constraints for both shopping and eating healthy. Therefore, we decided to target this demographic of consumers, taking into consideration of their needs and wants by the results of a survey we composed (appendices ref 1) to tend to their preferences.

4.3 USE CASE DIAGRAMS

In software development, a use case is a list of steps which defines interactions between actors and a system to achieve a goal. In this project, our use case diagram shows the sequence of actions that the Scidge system is supposed to do. This diagram explains to the stakeholders what functionalities Scidge have, their roles and who will be interacting with our system.

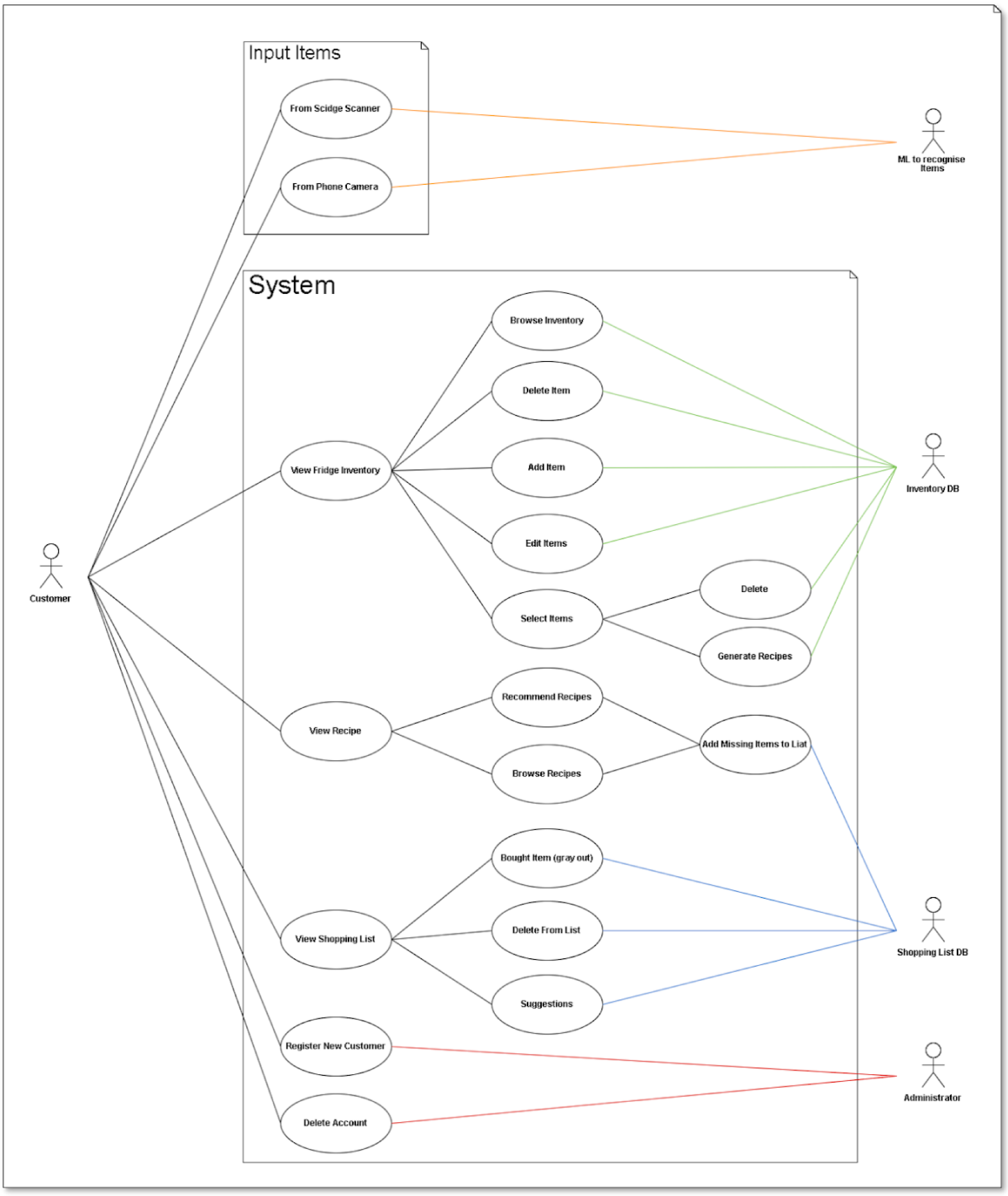


Figure 4-1 Use Case Diagram

4.3.1 USE CASE SCENARIOS

In this section, two use case scenarios are depicted. The first use case scenario in Table 4-5 describes the way a user logs in to the application to view the fridge inventory to delete an item. The second scenario in Table 4-6 describes the way a user can add an item through the Scidge hardware.

Use Case Name	Delete item
Context	Deleting an item that has been added previously
Primary Actor	Application user
Precondition	User has an existing account and has added at least one item into the Scidge inventory
Success Post Condition	Item was deleted correctly
Trigger	User has finished the food and wants to delete the item from the application inventory
Main Success Scenario	1. User opens application and is prompted to log in or create an account 2. User types in user name and password 3. System displays main inventory page 4. User selects the item and chooses the option to delete the item 5. System prompts to confirm delete action and deletes item from the list when user clicks on confirm.

Table 4-5 Delete item use case scenario

Use Case Name	Add item through Scidge
Context	Adding in a new item through the Scidge scanner
Primary Actor	Application User
Precondition	User has an existing Scidge account, synced with Scidge hardware through WiFi and user has turned on Scidge scanner
Success Post Condition	Scidge successfully captured information from both modules and send it to the application
Trigger	User bought a packet of sliced cheese and wants to add it into the Scidge inventory
Main Success Scenario	1. User places item on the Scidge scanner 2. User presses the button on the device 3. The weight and image of the item is captured and sent to the database through the ESP8266 module 4. The application then displays the new item in the Scidge Inventory

Table 4-6 Add item through Scidge use case scenario

5 DESIGN

5.1 DESIGN MODELS

This section describes the structure of the system and how the system will be implemented. Several different models have been developed to help with the Scidge application design. They all represents the structure of the system, the flow of different type of data and the overall activities of the application.

5.1.1 ACTIVITY DIAGRAM

Activity diagram shows the combination of activities performed by the system. It shows the overall control flow of Scidge system. It also represents high level understanding of the system's functionalities.

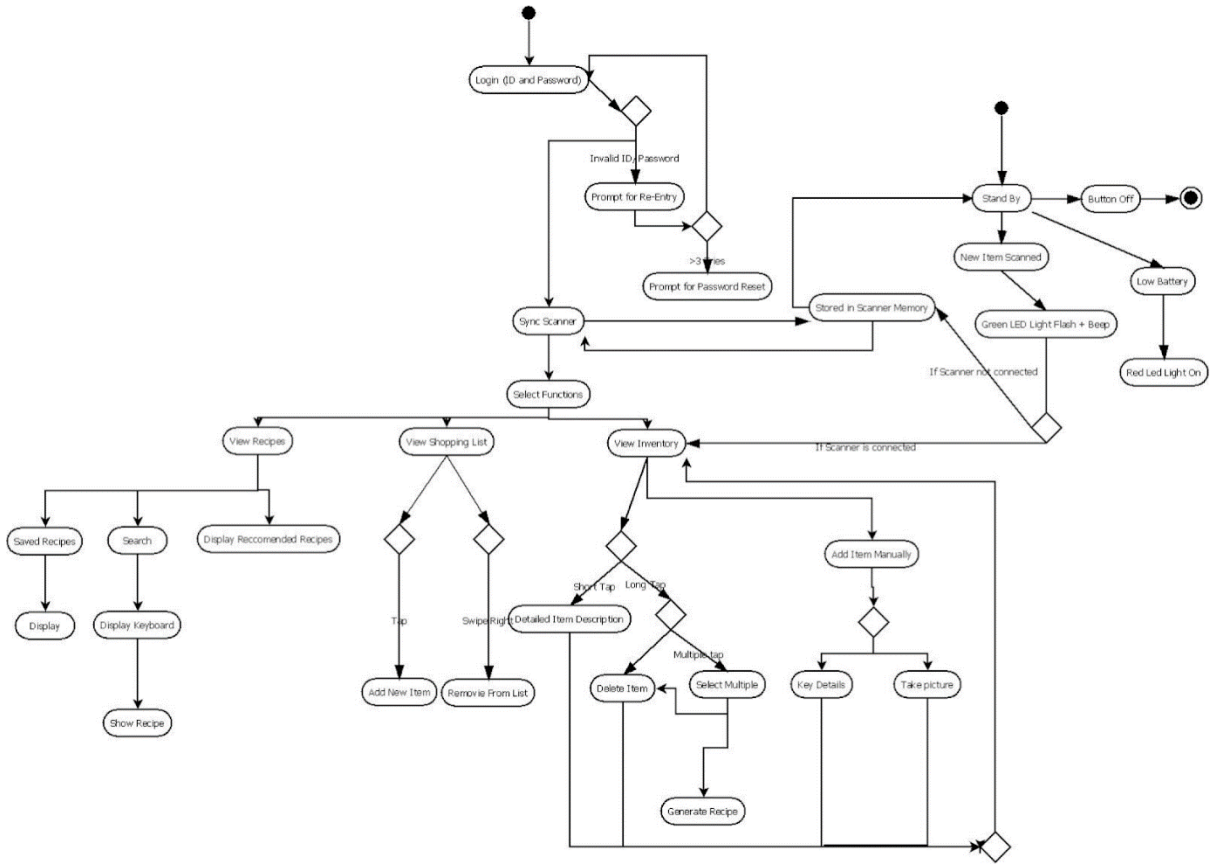


Figure 5-1 Activity Diagram

5.1.2 CLASS DIAGRAM

The following simplified diagram presents the main classes of the system. This shows the conceptual model of Scidge and was very useful during coding for creating different classes.

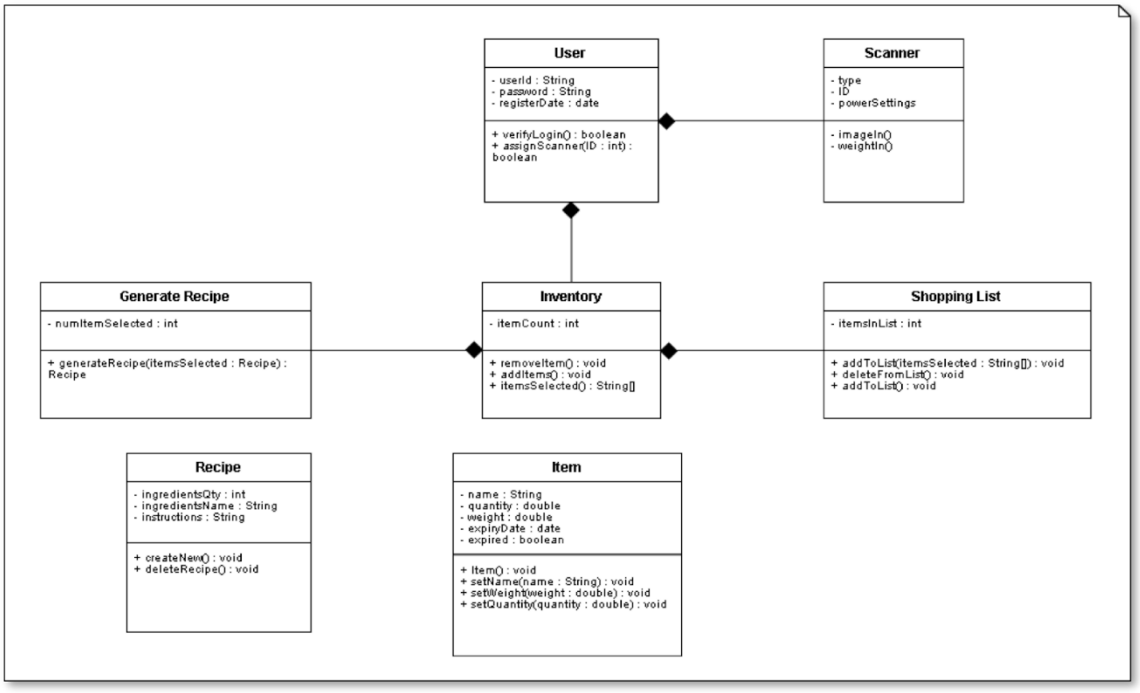


Figure 5-2 Class Diagram

5.1.3 SEQUENCE DIAGRAM

The sequence diagram is a communication design of the system that shows how the objects operates with one another and in what order the messages are exchanged between them. This diagram was taken into account in the development of front-end application as it explains well which objects the user interacts more and should be accessible and viewed easily on the software application.

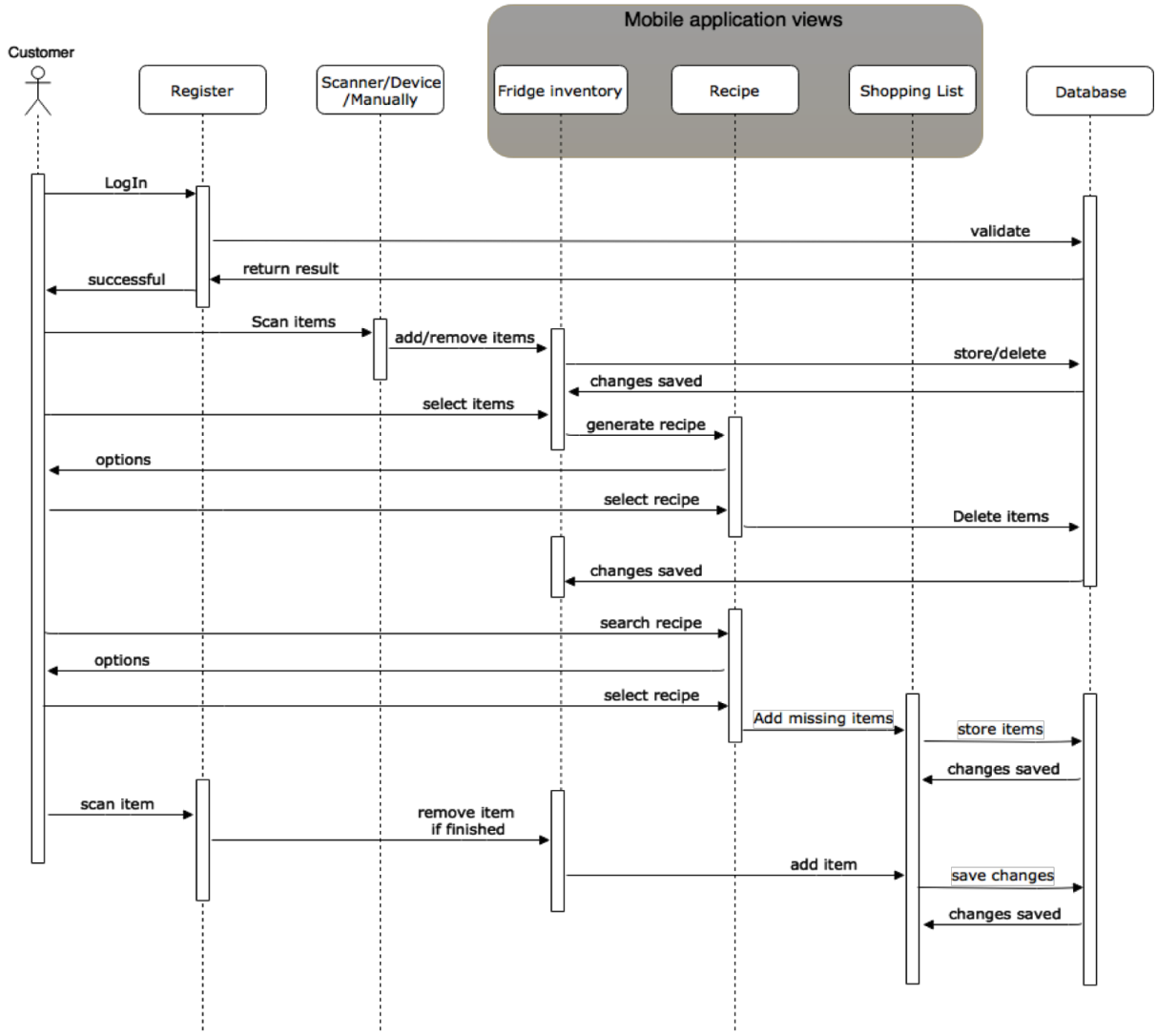


Figure 5-3 Sequence Diagram

5.2 DATABASE DESIGN

The ER diagram represents the structure of tables and the connection between the data stored within those tables. This ER diagram was developed and used to design the database for Scidge, it visually illustrates information on Scidge's users and products and how they are related in the database.

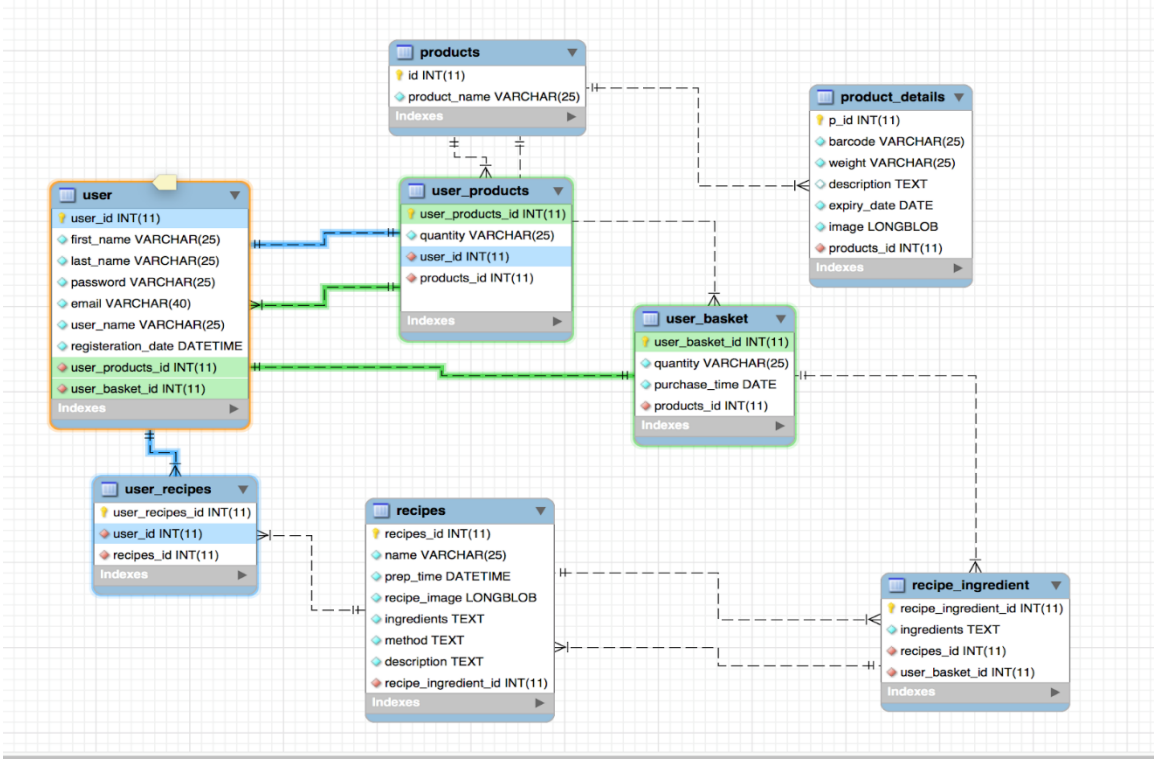


Figure 5-4 Database Design

5.3 UI DESIGN

Last term, our team has created the user interface design that has been shown in the project proposal:

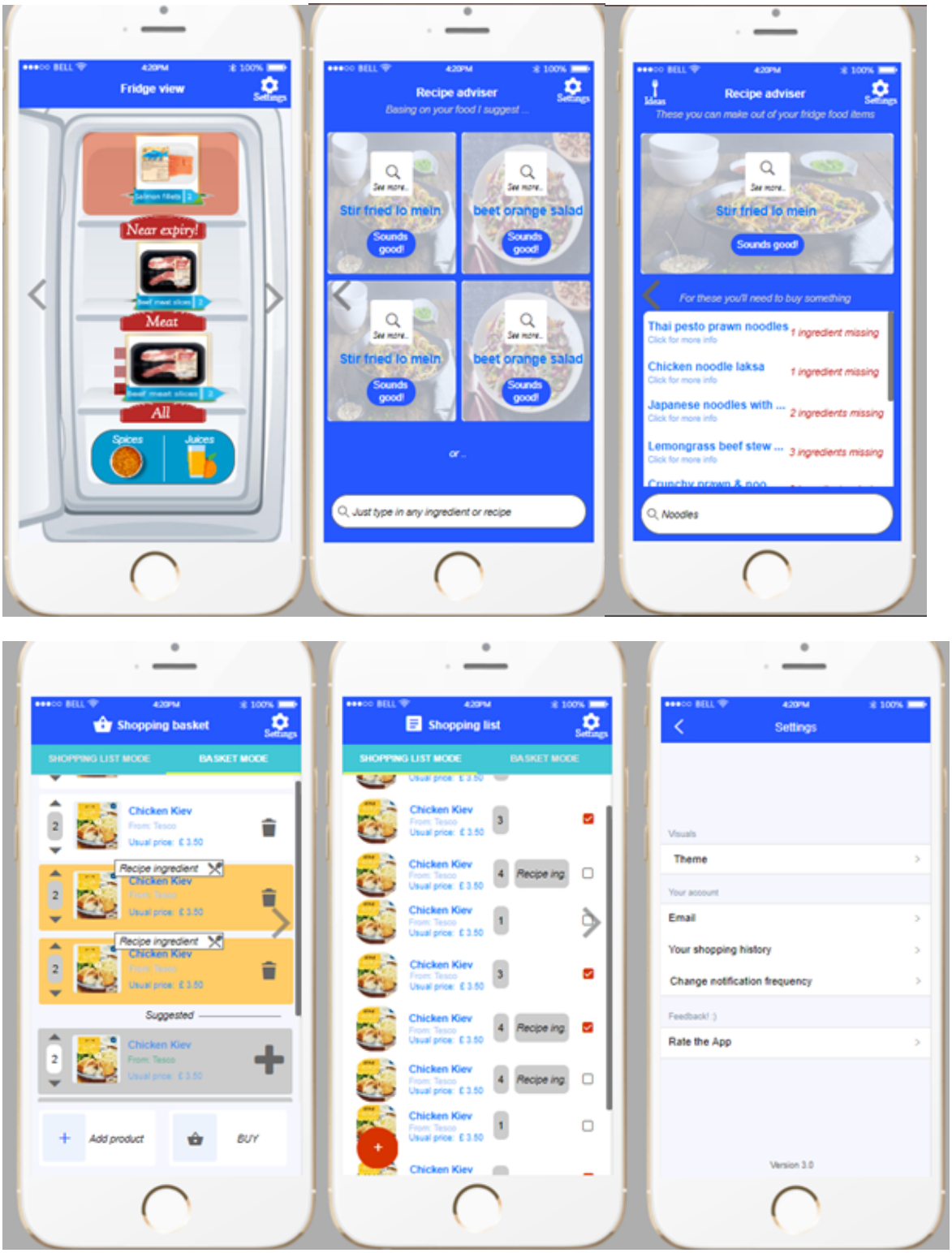


Figure 5-5 Term 1 UI Design Concept

This term, the team reviewed the layout of the main component – the virtual fridge view – and decided to change the following elements:

No.	Old UI element to change	Justification:	New UI element
1.	Fridge drawing shape around all the products	The drawing of the fridge together with part of the fridge doors takes too much space, which is so crucial to displaying the product information in a clear manner to the user. Additionally, it deepens the connotation that the SCIDGE system is only dedicated to food products that are inside of the fridge (products stored outside of the fridge, e.g. from cupboards, are also perfectly valid to scan into the system)	Products will be displayed in a grid extended to the border of the device screen.
2.	"Juices" and "Spices" bottom drawers	The team has realized that with such distinction of the products, there should also be drawers like "yoghurts" or "crisps" etc. So many drawers would bloat the screen and decrease the User Experience (UX) as user would constantly need to click through the drawers. We decided to abandon the drawers altogether and just group these products on a corresponding shelf.	Juices and spices to be show as regular products on the corresponding shelf.
3.	Left and right arrows pointing to next pages	The arrows – used as links to the recipes and basket pages – could hurt the UX with another unnecessary click required from the user in the interaction with the app. Moreover, they can become frustrating if accidentally clicked being so close to the product tiles (considering removal of the fridge drawing in point 1.). Instead, the user can "swipe" the screen to travel between the pages.	N/A "Swipe" interaction with the screen.

Table 5-1 Virtual fridge view UI review

Additionally, we wanted to implement the extra feature of scanning in the food products via the software in addition to the one available via the hardware. This was a common request by the participants of the Scidge Survey (taken in the term 1 and presented in the project proposal).

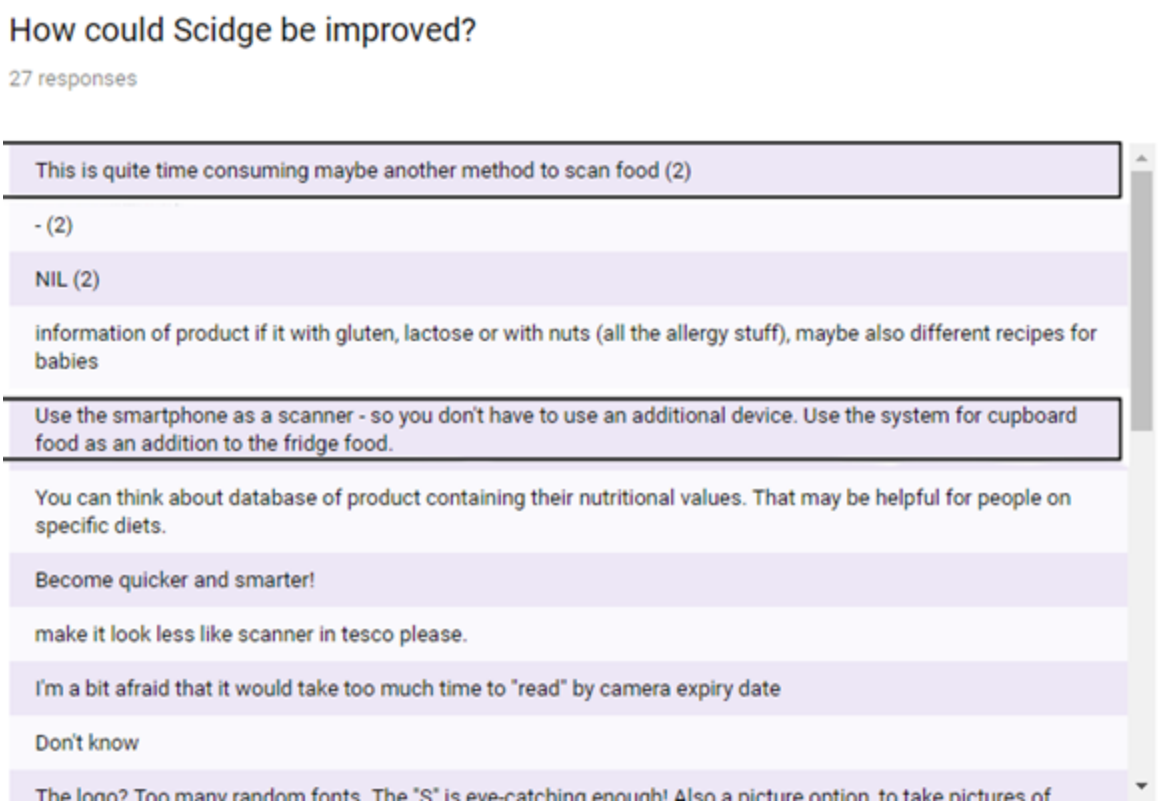


Figure 5-6 Survey answers suggesting software scanner part 1

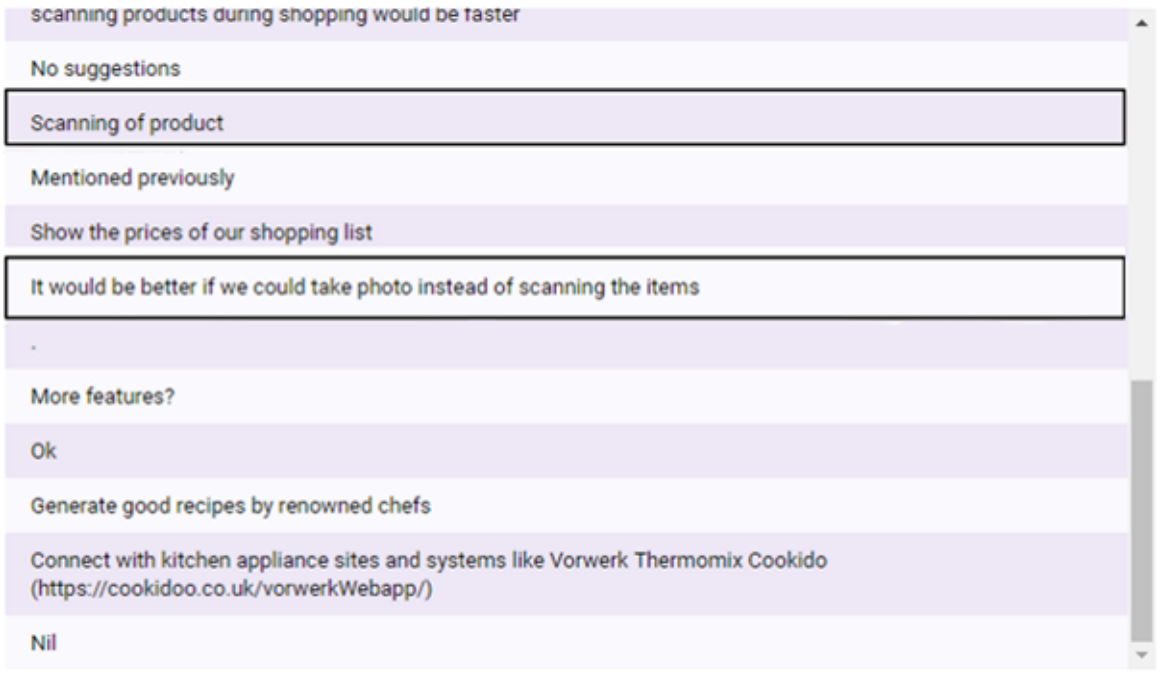


Figure 5-7 Survey answers suggesting software scanner part 2

As a result of the above, a new, enhanced UI of the Virtual Fridge View component has been created:

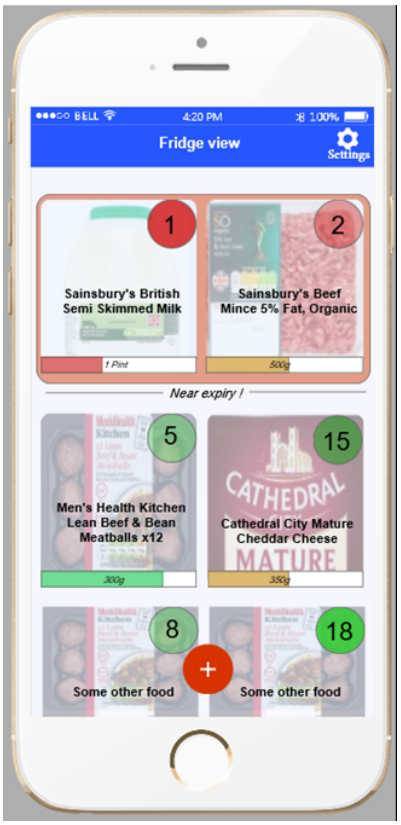


Figure 5-8 New UI in the Virtual Fridge View component enhanced according to the UI review results and the Software Scanner addition implied by in the Scidge Survey

This UI layout from now on served as a reference in creating the UI of the real application. It is also important to mention that it was not aimed to be depicted one-to-one in the actual implementation, but used as guidelines and a reminder of the justification of this particular layout.

6 IMPLEMENTATION

6.1 FRONTEND IMPLEMENTATION

Given the knowledge that the front-end area: the layout, the user interface and the user experience is the means through which the users interact with and judge their contact with our system, an extra effort was put to develop the front-end using the modern, industry-accepted approach. As a result, the team developed software's front-end as a mobile-first web application using one of the modern JavaScript frameworks designed as an "SPA" – a Single Page Application and a PWA – Progressive Web App.

This allowed the software to contain the following capabilities compared to a development without such technology stack:

1. Modern visual appearance due to incorporating the "Vuetify" UI components library into the software and a usage of the SASS CSS-preprocessor.
2. Dynamic data render and enhanced interface actions user can interact with.
3. Enhanced user experience and efficiency of the application.

While points number 1 & 2 will be covered in the following sections, let us describe point number 3.

By incorporating a Vue-router extension to the Vue.js framework our application relies on together with using the Single-Page-Application design, the data in the application is persistent across the pages and the page traversal across the web application does not require the user's browser to load any extra data. This reduces both the time for the app to become interactive for the user as well as user's network data, as the rest of the components on the page doesn't need to be refreshed.

```
// Firstly, import the actual framework to this file
import Vue from 'vue'
// Then, import the vue-router extension
import Router from 'vue-router'
// Next, all the components (pages) are imported to the file as a reference
import virtualFridge from '@/components/virtualFridge'
import signIn from '@/components/signIn'
import signUp from '@/components/signUp'
import notFound from '@/components/notFound'

// We tell Vue framework to use the Router object (which is the imported vue-router extension) as our
// "middleware"
Vue.use(Router)

// Finally, we initiate the Router:
export default new Router({
  // ... specifying all the URL routes
  routes: [
    {
      // Firstly the URL path
      path: '/',
      // ... and then a name of the component we want to show at that route (as shown in the next route)
      // ... or (as shown here) we may want to redirect to an other, defined route by using the "redirect" name.
      redirect: '/signin'
    },
```

Figure 6-1 Vue

```
    {
      path: '/signin',
      name: 'signIn',
      component: signIn
    },
    {
      path: '/signup',
      name: 'signUp',
      component: signUp
    },
    {
      path: '/virtual-fridge',
      name: 'virtualFridge',
      component: virtualFridge
    },
    {
      path: '*',
      name:'notFound',
      component: notFound
    }
  ],
  mode:'history'
})
```

Figure 6-2 Vue 2

6.1.1 LOGIN COMPONENT

The login component was the first to be created on the front-end. It contains the most basic functionality with the form validation and a business logic to obtain the user account.

Figure 6-3 Presenting the login page

Figure 6-4 Displays a situation when the user provides wrong credentials

```
signUserIn({ commit }, credentials) {
  commit('changeLoadingState', true); // a method is called to show the loading spinner while the data is
  beint fetched
  // We do an HTTP GET request (using a JavaScript extension called "axios" ) to our server
  // to get a salt of the user with the email provided by the user in the login form.
  return axios.get(`http://localhost:5000/db/users/getInfo/${credentials.email}`)
  .then(res1 =>{
    if (!res1.data) { // If there is no such email in the DB , we return false and display a following
    pop-up under the login fomr
      return false;
    }
    // Else, we hash the password candidate with the obtained salt
    let salt = res1.data.toString();
    let hash = bcrypt.hashSync(`${credentials.password}`, salt );
    credentials.password = hash; // .. end swap the password to the hashed password
    // Then we sent an HTTP POST request with user's credentials to be checked on the server.
   return axios.post(`http://localhost:5000/db/users/log_user_in/${credentials.email}`, credentials)
    .then(res2 => {
      // IF the server responds with any the user's account object, it means the passwords matched and we
      set them as our user account. We also stop showing the spinner at that time and return truth.
      if (res2.data) {
        commit('setUserAccount', res2.data);
        commit('changeLoadingState', false);
        return true;
      }
      // Else, we still stop showing the spinner and return false triggering the form error message again
      else {
        commit('changeLoadingState', false);
        return false;
      }
    })
  })
},
```

Figure 6-5 The business logic managing the user's login

6.1.2 SIGN UP COMPONENT

The registration page contains advanced handling and verification of the data in the input forms.

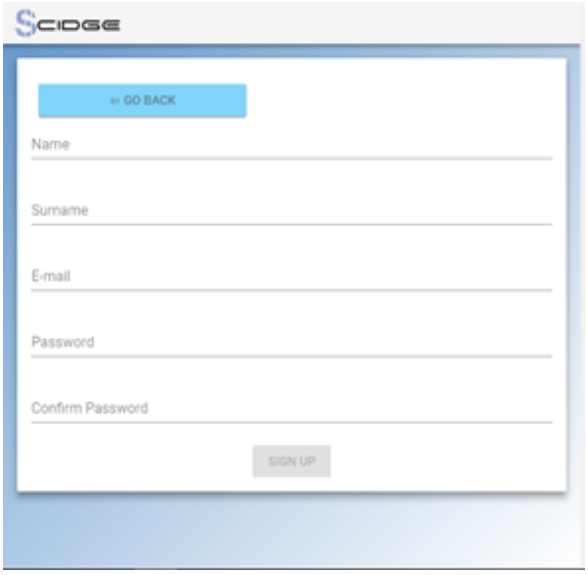


Figure 6-6 shows the initial state of the signup page

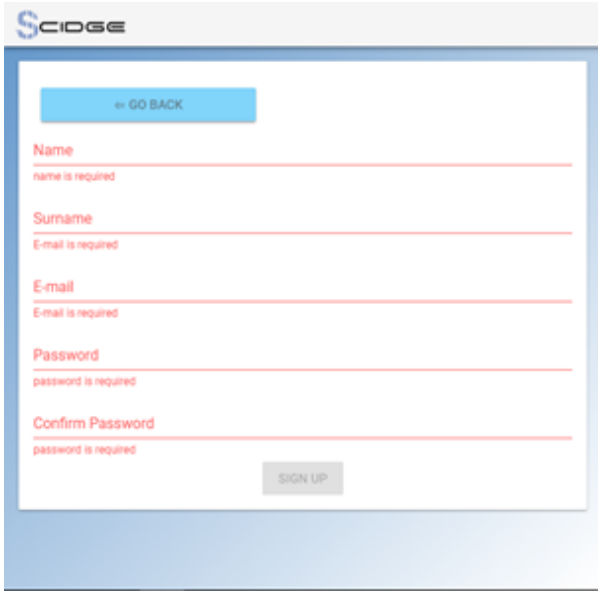


Figure 6-7 displays a form verification of non-empty input fields. The "sign up" button is disabled until the form is valid

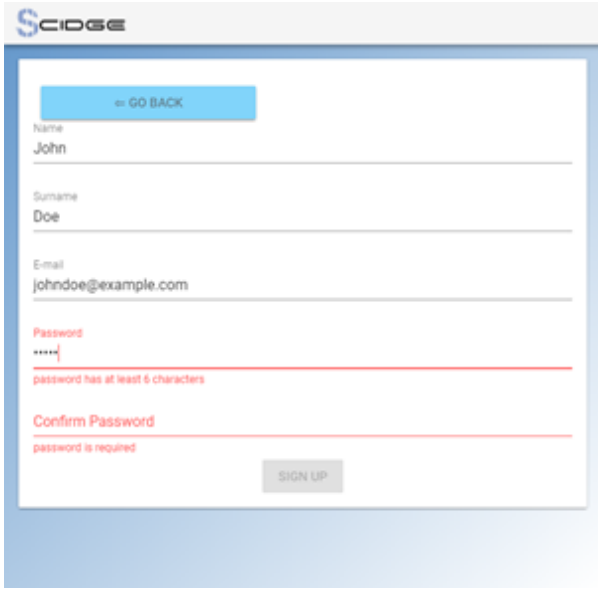


Figure 6-8 shows the password fields validation

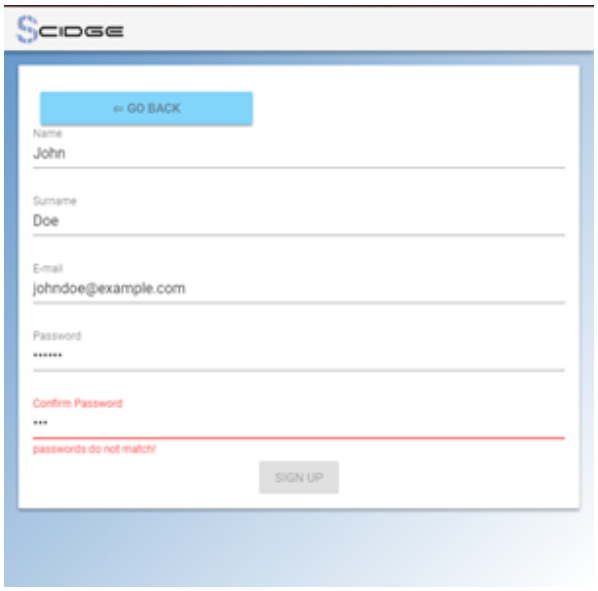


Figure 6-9 presents what happens if the user provides a password not equal to the original one in the "confirm password" input field. The "sign up" button is still disabled

Figure 6-10 presents the situation when user provided the same passwords in both password input fields. As all the input fields are correctly filled, the "Sign up" button stops being disabled.

Figure 6-11 shows the prompt message that is displayed after the user clicks the "Sign up" button. The app redirects the user to the login page shortly afterwards.

```
signUserUp({commit}, credentials) {
  commit('changeLoadingState', true);
  let salt = bcrypt.genSaltSync(10);
  console.log("the salt:");
  console.log(salt);
  let hash = bcrypt.hashSync(`${credentials.password}`, salt);
  credentials.password = hash;
  credentials = {
    ...credentials,
    salt: salt
  };
  console.log("vuex:");
  console.log(credentials);
  return axios.post('http://localhost:5000/db/users/add_user', credentials)
  .then(res => {
    if (res.data) {
      console.log(res);
      commit('setUserAccount', res.data);
      commit('changeLoadingState', false);
      return true;
    }
    else {
      commit('changeLoadingState', false);
      return false;
    }
  })
  .catch(err =>{
    commit('changeLoadingState', false);
    console.log("error while posting to:  http://localhost:5000/db/users/add_user !");
    console.log(err);
  })
},
```

Figure 6-12 showcases the business logic behind the registration procedure. Commentary snippets removed for brevity

6.1.3 VIRTUAL FRIDGE COMPONENT

The virtual fridge view was the core feature in our software. The enhanced UI design used as a reference in developing the component's layout. The visual aspect of was achieved due to implementing the "Vuetify" UI components library into the project.

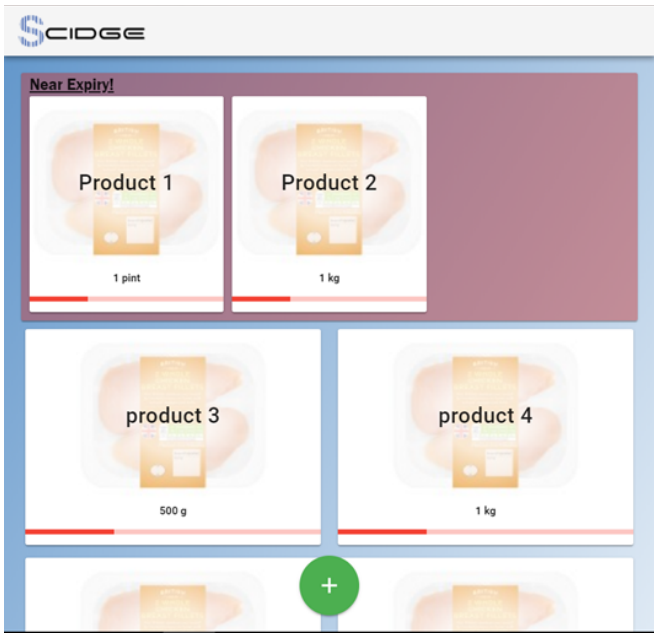


Figure 6-13 presents the view of the virtual fridge components. Food products in figure serve only as an example. Real food products have their name displayed and an image fetched from Open Food Facts API via the URL: "http://world.openfoodfacts.org/api/vo/product/BARCODE-OF THE-PRODUCT.json" and referenced through "product. image_front_url". Additionally, by using the Vue-router in the web application, the data is saved compared to the standard approach as clicking on the plus button or on the product to see the product details doesn't transfer the user to any new page but renders new data on the same one.

6.2 BACKEND IMPLEMENTATION

The database was built in two parts. The design of the database and the implementation part. Using the use case diagrams as a reference we developed several designs and after discussion with the team we came up with the database design which explained the functionality of our back-end very well. The database has tables for different objects that stores all the data received through hardware and application and from open source APIs.

For Implementation we started with creating packages such as Flask, datetime, json, requests and urllib.parse and moved further with the classes but every time we added the new contents we also did the test for any small changes to be sure that the system is working alright. Then we worked on the connection method here is dbConnect() which allow to connect with the database and fetch the result of query into the database tables. And adding the GET and POST so we can work around with the data. Creating classes in python for database tables to manage the data manually from here by adding, editing or deleting elements.

Database was implemented with MySQL on virtual server (hosted on the Computing department intranet) with PHPMYAdmin as the database administrative. MySQL is open source database management system and free of charge. Its secured and expandable for data storage. Also, Python works well with MySQL database. Back-end coding was done by python as it's a popular language and works great with Flask framework packages. Flask framework was chosen because it's simple and flexible and is also used in front-end development. For our database, currently we have db connection which connects to the database and a JSON converter to convert open sources code from JSON. pymysql package was imported to connect MySQL and Python. For API, currently we have GET router

that can retrieve data with getX and POST router that can reform data with postX. When a user tried to sign up to web application or hardware they GET and POST request are used to receive and store the data of the user and if the user is already signed up it examines the database for the the user data and allow the user to use the application.
The database interacts with both the hardware and the software front-end and is responsible for accepting queries and providing them with the data or storing data received by them. We faced many difficulties to connect the database with front-end as the front-end is written on JavaScript after few adjustment and using Vue framework in front-end the server was connected and we did a test where the barcode received by application was accessible through the database and through the server the data was retrieved for that product. The database stores all the data about the users and products received from hardware and application when the user signup it stores the data so next time the user can login and when the product is scanned through the hardware or application the barcode is received and with GET and POST requests data of the product is fetched from the open source APIs.

At one stage the packages in python were hard and problematic to download as we had both version 2 and 3 of python and some of the packages in one version were not accessible through another one and we couldn't use both at the same time so the team decided to create a virtual environment for python which was very time consuming and took us long time to figure out how it works and was still not very helpful as it was hard to connect to the database server so we decided to use python anywhere which an online python development environment and after all the hard work with that environment we realized that it provides a very small storage for storing data and it didn't worked out as well. During this process of testing different environments for python we learned about a lot of environments and the way it works.

Testing was done for back-end when the product was searched through manually adding barcode with getProductByBarcode and postProductByBarcode request the data for that product was returned through server from open source API [figure 6-16] and the same test was done for the user login as well.

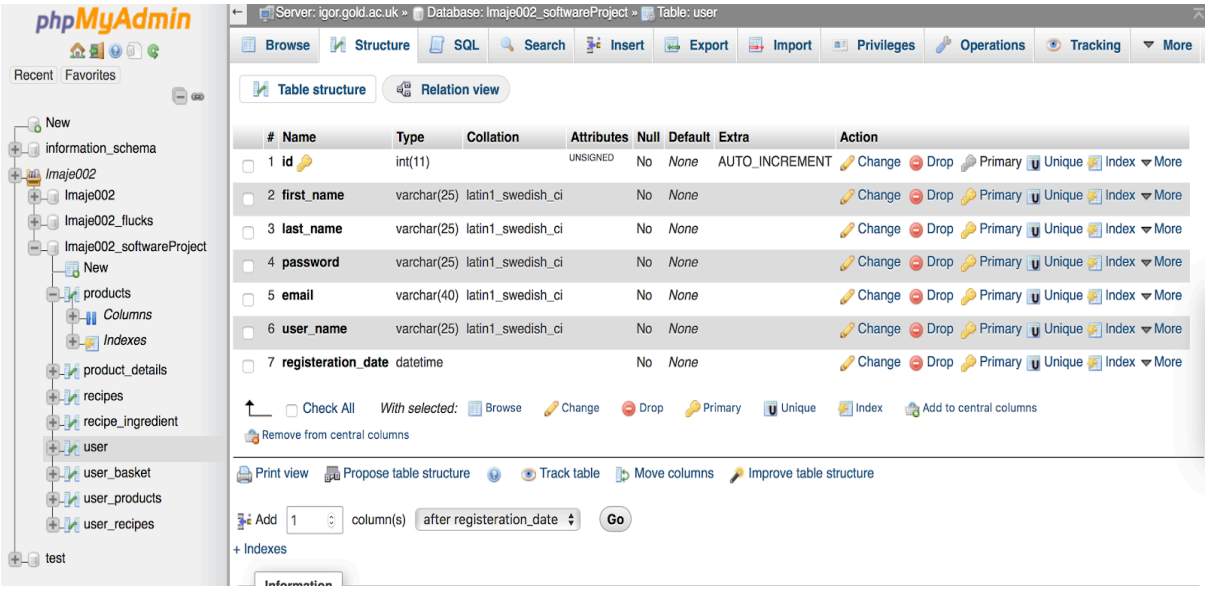


Figure 6-13 shows the user table in the database with the name and type of data used (user id, name, last name, password and email)

```
import pymysql
class Database:
                        host="localhost",                    #  localhost
                        user="root",                        # username
                        passwd="********",                  # password
                        db="lmaje002_softwareProject"
    def __init__(self):
        self.connection = pymysql.connect(self.host, self.user, self.passwd, self.db)
        self.cursor = self.connection.cursor()
    def insert(self, query):
        try:
            self.cursor.execute(query)
            self.connection.commit()
        except:
            self.connection.rollback()
    def query(self, query):
        cursor = self.connection.cursor( pymysql.cursors.DictCursor )
        cursor.execute(query)
        return cursor.fetchall()
    def __del__(self):
        self.connection.close()
if __name__ == "__main__":
    db = Database()
    # Data Insert into the table
    query = """
        INSERT INTO user
        (`first_name`, `last_name`)
        VALUES
        ('Mike', 'Bob'),
        ('Michael', 'Roff'),
        ('Imran', 'Khan')
        """
    # db.query(query)
    db.insert(query)
```

Figure 6-14 shows the insertion of user's data to the database tables and check if the user of the same name already exists

```
from flask import Flask, render_template, request, redirect, url_for import pymysql
import json
import requests
import urllib.parse import datetime
db = pymysql.connect(host="localhost", user="root",passwd="********", db="lmaje002_softwareProject")
# cursor object helps to execute all the queries needed
cursor = db.cursor()
cursor.execute("SELECT * FROM product_details")

#opening json data in python
barcode = '737628064502'
main_api = 'http://world.openfoodfacts.org/api/v0/product/[barcode].json'
url = main_api + urllib.parse.urlencode({'code': barcode})
json_data = requests.get(url).json()
#print(json_data)
db.commit()
db.close()
#data_entry()
app = Flask(__name__)
@app.route('/api/v1/product_details/<p_id>')
def getProductByBarcode(p_id):
    data = {
    'barcode': getBarcode(),
    'weight': getWeight()
    }
    return json.dump(data)

@app.route('/api/v1/user/<id>')
def getUserByID(id):
    data = {
    'username': getUserName(),
    'password': getPassword(),
    }
    return json.dump(data)
```

Figure 6-15 shows the Flask router that fetches data from open source APIs into database by using the product barcode number

6.3 HARDWARE ARCHITECTURE AND IMPLEMENTATION

The main core of the device consists of four parts. The Arduino UNO R3 board, the OV7670 camera module, the HX711 load cell amplifier and the ESP8266 WiFi module. The system is designed and built in such a way that the weight and the image of the food item can be captured with just a push of a button once the food is placed on the device.

6.3.1 SENSORS

6.3.1.1 CAMERA MODULE OV7670

The OV7670 Camera Module is a low voltage CMOS image sensor that provides the full functionality of a single-chip VGA camera and image processor in a small footprint package. The OV7670 provides full-frame, sub-sampled or windowed 8-bit images in a wide range of formats, controlled through the Serial Camera Control Bus (SCCB) interface. It has image array capable of operating at up to 30 frames per second (fps) in VGA with complete user control over image quality, formatting and output data transfer.

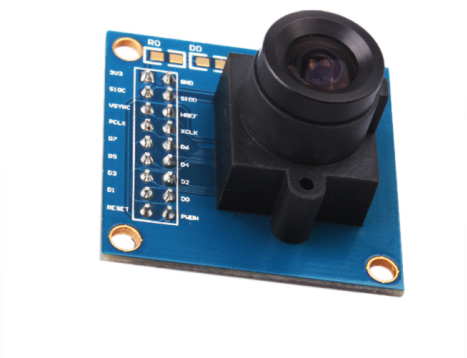


Figure 6-16 OV7670

The main features and why we chose the OV7670 module are:

- High sensitivity suitable for low light operation
- ISP has a compensation function to eliminate noise and dead pixels
- Support for image scaling
- Saturation automatically adjust (UV adjustment)
- Automatically adjust edge enhancement
- Automatically adjust the noise reduction
- Automatically affect the control functions include: automatic exposure control, automatic gain control, automatic white balance, automatic elimination of light stripes, automatic black level calibration image quality control including colour saturation, hue, gamma, sharpness

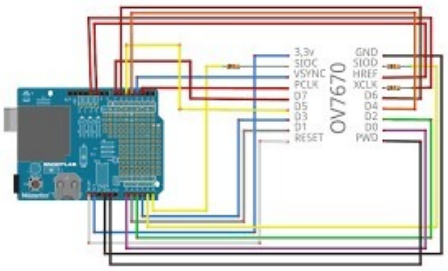


Figure 6-17 Connections for OV7670

For more details, please refer to the data sheet in reference.

6.3.1.2 1KG LOAD CELL

Figure 6-18 Load Cell

This straight bar load cell can translate up to 1KG of pressure (force) into an electrical signal. Load cells are designed to measure a specific force, and ignore other forces being applied. The electrical signal output by the load cell is very small and requires specialized amplification.

6.3.1.3 LOAD CELL AMPLIFIER HX711

The Load Cell Amplifier is a small breakout board for the HX711 that allowed us to easily read load cells to measure weight. By connecting the amplifier to the Arduino UNO R3, we can determine the changes in the resistance of the load cell, and after some calibration we were able to get very accurate weight measurements.

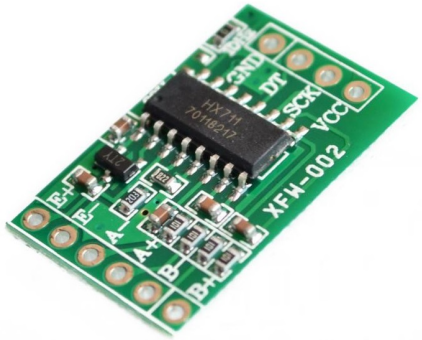


Figure 6-19 HX711

After our research, we determined that most of the food that were wasted were the perishables such as vegetables and dairy products which were mostly under 1kg, hence we decided to use the 1kg load cell coupled with the HX711 Load Cell Amplifier to accurately determine the weight of the food item placed on the Scidge scanner.

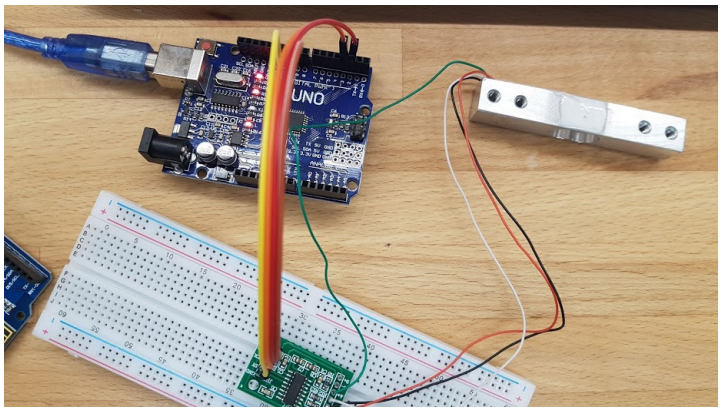


Figure 6-20 Calibrating the Load Cell and Load Cell Amplifier

For more details, please refer to the data sheet in reference.

6.3.2 ARDUINO UNO R3 MICROCONTROLLER

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as Pulse Width Modulation (PWM) outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an in-circuit serial programming (ICSP) header and a reset button. It can also be powered with an AC-to-DC adapter or battery to get started.

Figure 6-21 Arduino UNO R3 Microcontroller

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Table 6-1 Technical Specifications for Arduino UNO R3

6.3.3 COMMUNICATION MODULE

The Arduino Uno does not have a built in WiFi modules, thus it would require an additional module as described below.

6.3.3.1 WIFI MODULE ESP8266

The WiFi Module ESP8266 is a self-contained system on chip (SoC) with integrated TCP/IP protocol stack that can provide the Arduino UNO R3 Microcontroller(6.3.2) access to any WiFi network. The module is capable of either hosting or offloading a Wi-Fi network application.

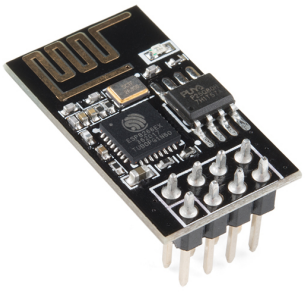


Figure 6-22 ESP8266

The module is pre-programmed with an AT command set firmware, which allowed us to connect it to our Arduino board easily. This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime.

The main features summed up for the Wi-Fi module are:

- Integrated TCP/IP protocol stack
- Integrated flash memory
- Standby power consumption
- Supports P2P and Soft-AP

For more details, see datasheet in reference.

6.3.4 CONSTRUCTED PROTOTYPE

After several design sketches, the main structure of the Scidge prototype was created from only two pieces of 6mm clear acrylic sheets. After coming up with several prototypes and receiving feedback, we modified several aspects of Scidge, such as removing hinges which would allow the scanner to be folded. We also took into consideration that most of the food would be from the fridge, thus Scidge would need to be waterproof to prevent the sensors from getting destroyed by condensation.

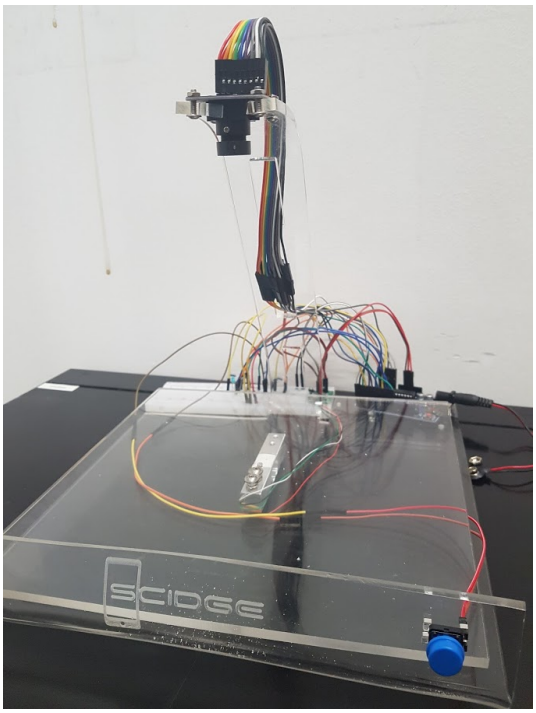


Figure 6-23 Scidge prototype

We then recreated the design in Adobe Illustrator with precise measurements of the sensors position and used the laser cutter from the Goldsmiths Hatch labs to create the main structure. In order to produce a clean finish, we heated the acrylic and sculpted it to its current form.
Scidge was created to hold the OV7670 Camera Module in place while elevated above the scale and the holes drilled allowed all the sensors and boards to be held firmly in place with screws.

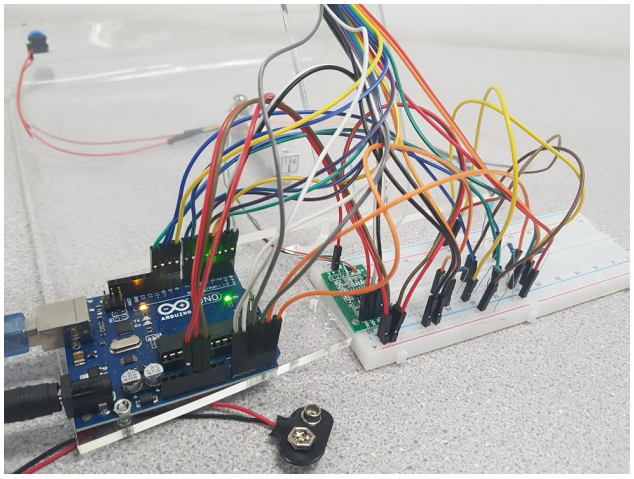


Figure 6-24 Wiring for the prototype

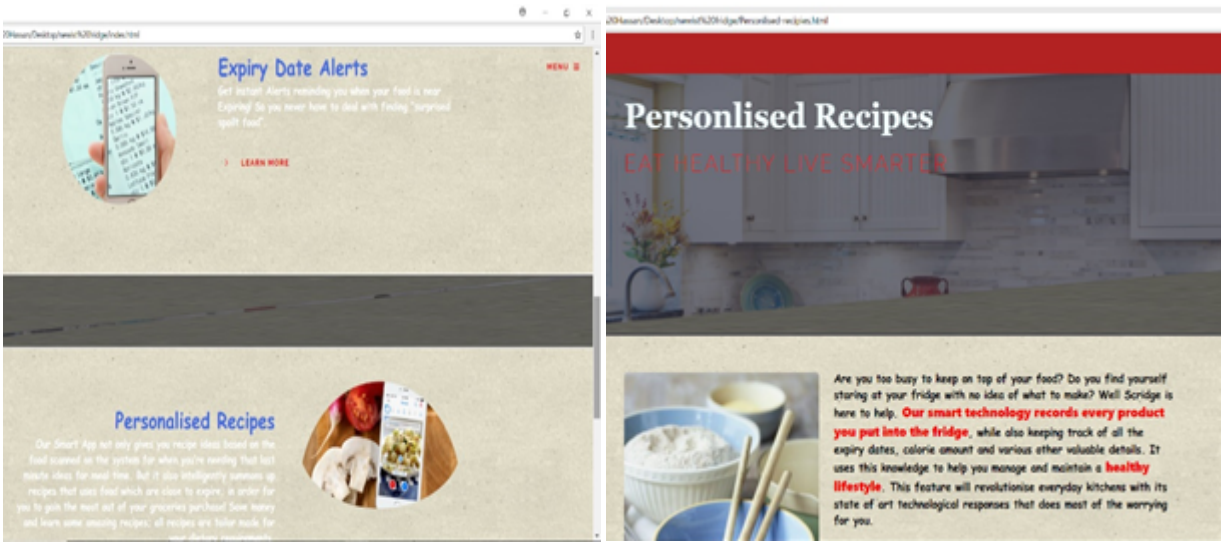
6.4 WEBSITE DEVELOPMENT

The main role of a promoting a company is to increase brand awareness, provide the appropriate information to the potential customers and to build sales and profit (Mishra, A 2015). Choosing the right platform to showcase the promotion of the brand for the desired target customers is imperative in achieving promotional success and connect with the potential customers (Bradley, J 2018). For this project, the decision was to create a website to promote and advertise our new product; as a website would not only be cost effective in producing, but also accessible to the public as the use of internet is more in the rise than ever before.

Creating a website designed to promote the brand is an innovative and thought-provoking task as the website did not only have to be attractive and appealing enough to draw the attention of potential customers, but also carry interesting and enlightening information that would both benefit and intrigue the users into buying the product. The website is required to be user friendly and engaging to the users as well advertise the product itself.

The decision was to create a dynamic website, with a complementary colour scheme and a range of pages that will be filled with necessary information. Colour can often affect the perception of consumers, as it often plays a key role in marketing and advertising. (Lant, K 2017). Colours are considered to be "85%" the reason behind why consumers decide to choose a certain item, and "90%" of the impulsive decisions are based solely on colour (Lant, K 2017) and researchers have discovered that 42% of consumers often will form a biased opinion of an item based on the website design, compared to any other factors. Taking this on board, the decision to choose the colour scheme on the final design was based on numerous research and it was acknowledged the most sophisticated colours, especially in relation to technology brands, were neutral colours as these implied you have "nothing to hide" and your products will stand out from the crowd with the sense of cleanliness and transparency (Miah, K 2016).

The overall design and presentation of a website is heavily visual and consumers are known to make aesthetic judgements quite quickly, in most cases almost 50 seconds into browsing (Lindgaard, et al 2006) (Chen, J 2009). Research by Khuang (2001) demonstrated that emotion can increase the value of physical product, and emotions play an important role in capturing the public attention and retaining certain information into their memories (Chen, J 2009). In this website, the decision to have a 'kitchen-healthy eating' theme throughout, was mainly down to its relation to the specified target market. The item that is being sold is a home electronic item, that is to be placed in the fridge. Carefully selected the ideal images that would reflect the information on each page was imperative in capturing the user audience and drawing on those strings of emotions. Photoshop played a key role in this, to ensure certain edits such as contrast tweaks and lighting, enabled to pictures to really stand out.
In order to successfully promote the brand, it was substantial to provide the consumers with the required information that informed the readers of the qualities of the product and out motive behind it, while giving them a reason to buy and own the item themselves, without boring the readers with unnecessary vast volume of words. They key is to be engaging and persuasive. Ensure the reader can relate to your content, draw and emotion connection and agree with your motives and reasons behind the concept, which was well demonstrated through the webpages produced.



A successful website will deliver the concept behind the idea whilst also persuading the users to buy the item. Outside the website, various platforms may be used to promote the website, banners, posters and TV commercials being only a few. For this product, an alternative idea was used whereby the item was pitched to the London Mayor Entrepreneur Award. Winning the London Mayor Entrepreneur Award will allow the "brand" to stand out from the crowd and not only send a positive message to existing customers but also attract links with new business and promotion by generating media coverage in the press and offering guidance on how to communicate the success on to existing and potential clients (best business award, 2017).

6.5 UNFORESEEN PROBLEMS & SOLUTIONS

6.5.1 SOFTWARE

During the development of the software, 2 major difficulties were encountered:

1. Establishment of a correct connection between the front-end build architecture and back-end server.
2. Creating custom solutions to the native Flask methods unavailable due to handling some server-typical tasks to the client-side.

The first problem arose due to a non-trivial architecture of the front-end and the multiple tools it needs to be processed with in order to be served as a file via the server.
As explained in the section 2.2.1., the front-end architecture builds with Vue.js is not a standard set of ".html" files with references to JavaScript and CSS. In fact, before being bundled by the Webpack a static JavaScript file and injected into a single "index.html" file, it is a list of Vue components with framework-specific features and a complex correlation that is not understandable by any other framework-external agent.

Because of this an extra time was needed to find the right solution. Firstly, the team explored the "build process" (i.e. a process of transforming the framework-specific business logic to its framework-agnostic equivalent in the programming language it was written) of both back-end and front-end architectures. Then, an effort was made to discover how to configure the Flask server so that it serves the static JavaScript files that were bundled and "built" by the Webpack from the Vue components while the

routes handling is not managed by the server, but passed down to the Vue-router, which resides inside those static JavaScript files.

The second problem was also related to the back-end – front-end complex relation. Because in this technology stack most of the business logic is handled on the client-side and that the back-end and front-end operate in different programming languages (Flask in Python, Vue in JavaScript), the team was not able to use some of the mature and well-established extensions of Flask and needed develop own custom approaches. One of the particular examples of this is the login system. To make an authentication system secure, the system should handle this matter predominantly on the back-end as it is easier to exploit sensitive data on the client-side (as it is downloaded to the user's browser, thus accessible in the development panel). However, it was not possible to handle the authentication process fully on the back-end and use flask-login – a production-ready user session management extension in this scenario. This is because, as mentioned above, Scidge's Flask server perceives the application as one index.html file referencing a big JavaScript file of code and has no ability to distinct the pages inside as the Vue-router does. Therefore, the team could not use the standard way of preventing unauthorized users from accessing restricted pages in Flask.

As a solution, the software front-end leader implements a custom method inside the Vue-router that checks if the user's data exists in the browser's Local Storage (indicating that the user is logged) and then, according to the result, lets them enter the desired page or sends them back to the login page.

Equally complicated situation applied to storing the passwords in the db. First of all, it was necessary to alter the standard approach of sending the user's data to the server and hash their password there before storing it in the database. Because of the same issue of not seeing pages by the Flask server, it was unworkable to use the standard Flask way of processing the user login forms with "Flask-WTForms" which has a built-in way of securing the data flow to the server. Therefore, we the application was exposed to the Man-In-The-Middle attacks while sending unsecured data to the server.

To resolve the issue, team has decided to hash the passwords already at the client-side with a random salt generated basing on the user device's time. Next, both the hashed password and the salt were store in the database for the particular user. Then, when the user wanted to log in, the client-side fetched the salt for this particular user (if only they provided the right email) hashed the "password-candidate" (i.e. the password provided by the user to be checked if same as the one in the DB), sent the hashed password to the server where it was then checked if the passwords matched and the server sent a True or False.

6.5.2 HARDWARE

One of the problems that I faced when developing the hardware was that the learning curve was very steep, and it was difficult to troubleshoot as there was no one to assist or guide me when I was stuck. This resulted in many hours spent searching for answers online only to find out that I made a minor mistake which could have been avoided.

Another problem was due to an over ambitious project aims where we wanted to connect the Scidge scanner with the database through WiFi as the development of Scidge scanner was more complex than we thought.

7 EVALUATION & QA TESTING

7.1 FORMATIVE EVALUATION AND INTERACTION WITH USERS

During development a survey (ref 2 appendices) was composed to make sure our system met the objectives and aims we initially wanted to tackle. Questions were designed to determine validity, reliability, and statistical significance of our device in relation to the user needs. 5 test users that matched our stakeholders criteria were selected and as one final model of 'Scidge' was produced, collecting results was a gradual process over a 5 week period, with each test user trialling 'Scidge' for one week each to gain a definitive perspective. In doing this, it allowed us to discover any faults and facilitate any changes early if needed.

Outcomes of the testing were extremely positive (ref 3, appendices). 5/5 of test users said they have positive opinions of the quality and innovation of the product(ref 3, question 1-3, appendices) An average of 40% said they would "probably need" 'Scidge', 20% expressed they would "definitely need", whilst the remaining 40% selected neutral. Question 6 asked "If the product were available today, how likely would you be to buy the product?" 60% said they were "somewhat likely" and the 40% responded "very likely". The diagram (in ref 3, question 7, appendices) is a graph showing how likely our test users would recommend.

Proceeding onto questions 8 (ref 3, question 8, appendices) and 9 (ref 3, question 9, appendices) on the survey it allowed any opinions to be expressed of 'Scidge'. Analysing the responses to those questions we understood that users felt positive using the device as it gave them a sense of achievement knowing that they are contributing in helping reduce a global issue and encourages people to become more eco-friendly. Internal benefits were also felt, as it saved money due to inciting cost-effective methods in cutting down over-purchase of food. The main feature that needed to be improved is the size of our external hardware scanner. Our target audience included young professional and students, which they usually don't have the luxury of space, especially living in a high-paced city. Enabling push-notifications were also conveyed on what could improve, as it increased the users inefficiency due to having to check back onto the website, in which sometimes the food has already expired. This user testing taught us that 'Scidge' is an device that isn't a fundamental need for all the specific stakeholders we initially thought would benefit most, however, in long term perspectives it heightens convenience for those who have the disposable income and time.

In the original project proposal the features we were aiming to include were image recognition of the item rather than a barcode, push notifications, automatic shopping list (where it notifies the user if their 'essential' items are running low - with the option of automatically updating the basket of their shop of choice), a recipe advisor which gives recipes based off the users items and a health tracker so if the user wants to gain/lose weight or has health implications such as diabetes, it would notify them if the item they scanned in would beneficial towards their goal. After conducting research, we found that personalisation is what the existing market lacked and so that's why we created those features; to make "Scidge" tailored to each unique individual. During development we learned that image recognition involved learning new technologies such as machine learning, as well as having a considerably large data set and neural network to accommodate all the real world influences (lighting, the item being perfectly centred to be recognised, size etc.). Time was also a resource that we lacked and therefore executing that feature would be challenging based off these limitations.

7.2 FUNCTIONAL REQUIREMENTS EVALUATION

In order to test if the application and hardware meets the defined functional requirements in section 4.1. We tested all aspects of the application during and after the implementation. The following tables presents the test cases written to evaluate the correct functionality of the application.

7.2.1 SOFTWARE

Tested Requirement	**SFR01**
Test Content	Checks that the application allows the user to sign up
Input	User fills up form using their name, email and password
Pass Criteria	Application displays a confirmation message and redirects to the main screen

7.2.2 HARDWARE

Tested Requirement	**HFR01**
Test Content	Transfer of information from OV7670 module, load cell and load cell amplifier to the computer
Input	Capture and send a picture and weight through USB
Pass Criteria	Able to send and receive image and weight information to the computer

7.3 NON-FUNCTIONAL REQUIREMENTS EVALUATION

In this section, we checked whether the non-functional requirements of the application which were defined in Section 4.1 were successfully fulfilled.

7.3.1 SOFTWARE

SNFR01: The application was designed so that once the user is logged in, they will be directed to the inventory overview page where they will be able to see all the items in their fridge with the expiry date and the amount of food left.

SNFR02: We were able to accommodate different screen sizes of different phones.

SNFR03: We developed the interface of our application to be as clean as possible after several feedback loops and modifying our interface (described in Section 5.3).

SNFR04: We believe that through the modifications we made from the feedback we received has made the application visually appealing.

SNFR05: We were able to come up with a solution to solve the problem hiding personal data which was described in Section 6.5.1.

SNFR06: We were able to implement the barcode scanning function which made the scanning of the product into the app relatively convenient.

7.3.2 HARDWARE

HNFR01: Scidge scanner was designed to be user-friendly and easy to use, it only requires the user to place the food on Scidge and push one button for both the weight and image of the food item to be captured and sent to the computer.

HNFR02: Scidge was designed to function with a 9V power supply or a 9V battery so that it is portable.

HNFR03: We determined that most of the food item in the fridge are less than 1kg, thus we decided to use a 1kg load cell. However, the 5kg load cell is also available and can be easily changed.

HNFR04: We chose the OV7670 camera module (described in 6.3.1.1) which satisfies our image quality requirements.

HNFR05: The response time for the data transfer to the database through WiFi has yet to be determined as we were unable to connect to the database.

HNFR06: Scidge was designed so that the microcontroller chip and sensors are easily changeable with just minor adjustments in the laser cutting of the acrylic.

8 PROJECT CONCLUSION

8.1 SUMMATIVE EVALUATION

The main aim of the project was to create a device with a system that is practical, inexpensive and intuitive for the target market in order to cut down on food wastage and thus reduce the carbon footprint. The main specification the device would entail were; to identify the item of which the barcode has been scanned, give an expiry date automatically or allow the users to register this manually, produce recipes using the inventory of the fridge and allow users to view the inventory of their "virtual fridge" by removing products as they finish. This project was eight months project, which proved complex and time consuming at most.

Because of the chosen approach, our front-end was heavily complex and featured not only the common front-end functionalities, but also many typical back-end processes: route-handling, managing user permissions in navigation (also known as route-guarding), outside-APIs calls, input forms validation, and even some part of the authentication system.
While it took us some extra effort and time to firstly learn our front-end tech stack, connect it properly with the back-end and come up with workarounds in some scenarios, all in all, it gave us extraordinary capabilities during the development of the software. The ones particularly useful, making the designing process significantly faster were:

- vast advanced options of manipulating the HTML DOM (Document Object Model): each element on the page could be easily manipulated with the Vue.js templating engine using the " {{ }} " syntax or conditionally rendered using the "v-if / v-else/ v-for" appended to any html tag.
- Vuex central state management: thanks to the "Vuex" Vue.js extension, we had a one central "store" JavaScript file which can be considered as a single logistical centre from which we could easily pass the dynamic data to different page components (for example a user data object we could keep easy reference to across or our sub-pages of the application) or dispatch any asynchronous AJAX calls "action" (for example to our server or the OpenFoodFacts API) with fetched data automatically rendered or updated in the desired places of our application.
- Further data rendering options: once a JSON containing the desired data was fetched using the Vuex "action" it then could be conveniently displayed in a variety of ways, where our favourite was rendering all the object properties values with the looping "v-for" directive appended to the html element we wanted the data listed and then having it's CSS class dynamically bound based on the condition in "v-if" directive. This amount of easily accomplished ways of representing the data is almost invaluable in the application focused on displaying user's food information.

To conclude, we believe choosing to develop the software as a mobile-first, "PWA", "SPA", modern-JavaScript-framework-driven web application allowed us not only to deliver an almost native-like application an understandable quality (given the time and the human resources) but also a software that is available across multiple operating systems without the need of the time-consuming development in many programming languages.

Furthermore, in terms of the external promotion, due to the limited time given for the website development because of the upcoming deadline for presentation of the project and the young mayor enterprise award, the time that would have been spent self-teaching and
researching the skills required to produce a dynamic virtually appealing website was restricted. Skills on how to Photoshop and produce a logo that would be unique, eye-catching, relatable to the target market and produce images for the website and the overall structure of the website was one of the many strengths. However due to the delay of the overall prototype of the device, no real images of the device could be used on the website, which was one of the downfall of the overall website. If more time

was allocated for the website and promotional aspect of the project, additional effort would be spent on producing promotional value on different platforms such as a video that could have also been embedded into the website.

8.2 FUTURE WORK

When constructing an idea for the project, the ultimate goal was to produce an innovative, unique product that would not only test and enhance our skills but also fit a purpose to benefit our potential target market. The overall project idea was a complex and difficult one, proving to be a project that needed more time. The specification required of this device was vast and time consuming and with the stress of exams and other academic priorities, certain aspects of the project were substantially delayed or not achievable.

The main specifications of the project were;

To create a device with a software that allowed the users to be able to add items into their fridge by scanning each item.

Allow users to keep an eye on the inventory of their fridge by using their smartphone (removing and adding items as they wish)

Give users recipes using the items in their fridge

Update users of food that may be running out or approaching their expiry date and allow users to re-purchase items from their phone.

The last two specification became the two most "unrealistic ideas" due to their complexity and were overall dropped when creating an final item. If more time were given for the project we would potentially plan the following;

To add in a machine learning algorithm to improve the system so that it does not solely depend on reading the barcode to identify the food items but is able to determine what item it is from a picture.

After developing the hardware, we were not able to connect the Scidge hardware with the database through WiFi, so we hope that given enough time this could be achieved.

For the Scidge hardware, after building the prototype, we are planning to design and print our own printed circuit board (PCB) so that the system can be

If more time was allocated for the website and promotional aspect of the project, additional effort would be spent on producing promotional value of the product on different platforms such as a video that could have also been embedded into the website; a video that would advertise the qualities of the device.

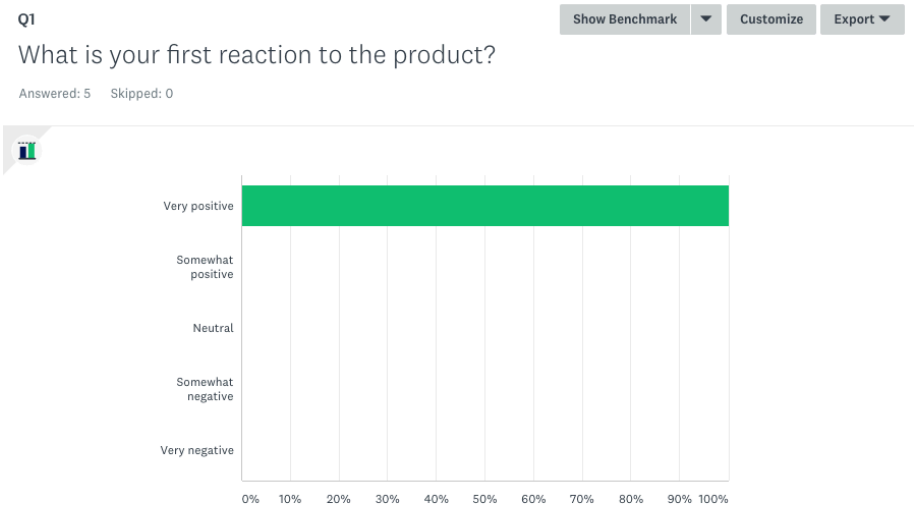
9 APPENDICES

Ref 1 – “Research Scidge Survey” results spreadsheet URL:
https://docs.google.com/spreadsheets/d/1N91yy-r-ZnaOgHKJtdXM8_Vst9w16ssOcNU4KYE2pns/edit?usp=sharing

Ref 2 - “Completed Scidge Survey Questions”
URL: https://www.surveymonkey.co.uk/r/88L2T6R
QR Code:

Ref 3 - “Completed Scidge Survey Responses”
URL: https://www.surveymonkey.com/results/SM-YFSM6RZVL/

Question 1:



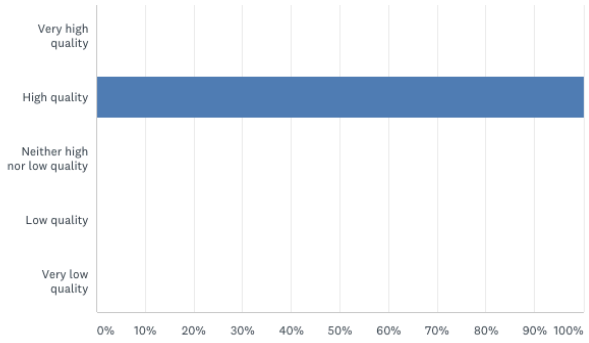
ANSWER CHOICES	RESPONSES	
Very positive	100.00%	5
Somewhat positive	0.00%	0
Neutral	0.00%	0
Somewhat negative	0.00%	0
Very negative	0.00%	0
TOTAL		5

Q2

Show Benchmark | Customize | Export

How would you rate the quality of the product?

Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES	
Very high quality	0.00%	0
High quality	100.00%	5
Neither high nor low quality	0.00%	0
Low quality	0.00%	0
Very low quality	0.00%	0
TOTAL		5

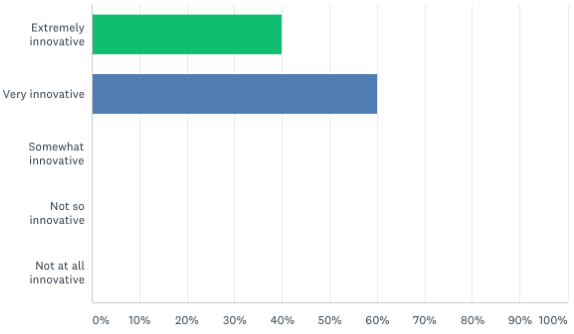
Question 2:

Q3

Show Benchmark | Customize | Export

How innovative is the product?

Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES	
Extremely innovative	40.00%	2
Very innovative	60.00%	3
Somewhat innovative	0.00%	0
Not so innovative	0.00%	0
Not at all innovative	0.00%	0
TOTAL	5	

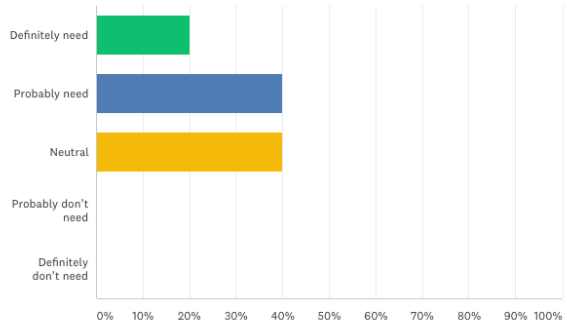
Question 3:

Q4

Show Benchmark | Customize | Export

When you think about the product, do you think of it as something you need or don't need?

Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES	
Definitely need	20.00%	1
Probably need	40.00%	2
Neutral	40.00%	2
Probably don't need	0.00%	0
Definitely don't need	0.00%	0
TOTAL		5

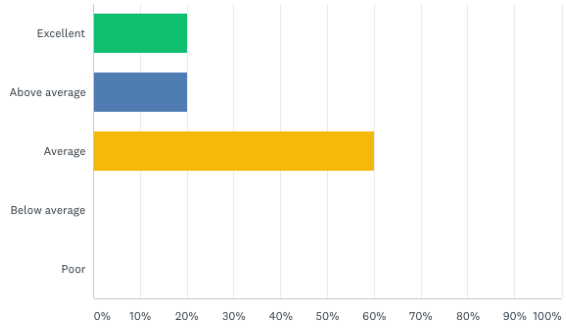
Question 4:

Q5

Show Benchmark | Customize | Export

How would you rate the value for money of the product?

Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES	
Excellent	20.00%	1
Above average	20.00%	1
Average	60.00%	3
Below average	0.00%	0
Poor	0.00%	0
TOTAL		5

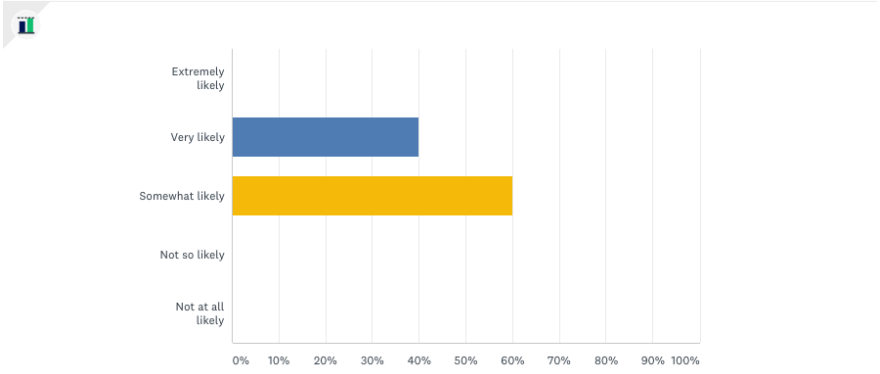
Question 5:

Q6

Show Benchmark | Customize | Export

If the product were available today, how likely would you be to buy the product?

Answered: 5 Skipped: 0



ANSWER CHOICES	RESPONSES	
Extremely likely	0.00%	0
Very likely	40.00%	2
Somewhat likely	60.00%	3
Not so likely	0.00%	0
Not at all likely	0.00%	0
TOTAL	5	

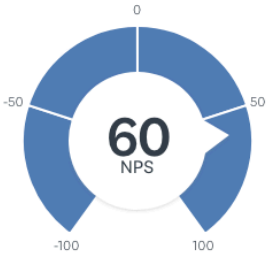
Question 6:

Q7

Show Benchmark | Customize | Export

How likely is it that you would recommend our new product to a friend or colleague?

Answered: 5 Skipped: 0



DETRACTORS (0-6)	PASSIVES (7-8)	PROMOTERS (9-10)	NET PROMOTER® SCORE
0 0	40% 2	60% 3	60

Question 7:

Q8

Export

In your own words, what are the things that you like most about this new product?

Answered: 5 Skipped: 0

RESPONSES (5) TEXT ANALYSIS TAGS

Add Tags Filter by Tag Search responses

Showing **5** responses

saved me money on buying things that i would think i need more of as i can see whats in my fridge on my phone.

View respondent's answers

I like that fact that it's a new and fresh (pun intended lol) way to prioritize my food consumption ensuring that I utilize my grocery shop to its fullest potential, which it turn saves me money and food wastage.

View respondent's answers

i liked how the design is sleek and futuristic with the exposed wires and what this product stands for with reducing food waste. seeing when all my food is going to expire on my phone makes shopping a lot easier and cost effective.

View respondent's answers

My favourite thing about the product is I will be able to save money by not throwing food away; by helping me decide which one I should eat first

View respondent's answers

being able to look at all my food expiry dates in one place

View respondent's answers

Question 8:

Q9

Export

In your own words, what are the things that you would most like to improve in this new product?

Answered: 5 Skipped: 0

RESPONSES (5) TEXT ANALYSIS TAGS

Add Tags Filter by Tag Search responses

Showing **5** responses

receiving notifications on when food is about to go off

View respondent's answers

I think visually it looks great,I love the transparency of the product showing all its mechanical glory which is edgy and attractive to the eye. However if the size could be descaled I think it would be more appealing, but I have taken into consideration, that this is just a prototype. Good job guys ????

View respondent's answers

although i liked the design, the size is a little on the larger side so i had to rearrange my other appliances around to make it fit. wish it had notifications that automatically pop up on my phone instead of always having to check on the website.

View respondent's answers

perhaps a compact model for a lunch box, you can put inside the fridge and it can tell you which one to eat first

View respondent's answers

the size

View respondent's answers

Question 9:

10 REFERENCES

Food Standard Agency (2017) Food Waste.
Retrieved 9 April 2018, from https://www.food.gov.uk/news-updates/campaigns/food-waste

Ridden, P. (2011).LG launches first Smart-Grid appliance: the Smart Fridge. Newatlas.com. Retrieved 10 April 2018, from https://newatlas.com/lg-smart-fridge/18502/

Elliott, L. (2017). Inflation rises to 3.1%, adding to UK cost of living squeeze. the Guardian. Retrieved 12 April 2018, from https://www.theguardian.com/business/2017/dec/12/uk-inflation-rises-uk-cost-of-living-squeeze

Munbodh, E. (2018). How much the National UK living wage rises for millions of workers in April. mirror. Retrieved 11 April 2018, from https://www.mirror.co.uk/money/new-minimum-wage-rates-april-12277215

Ramey, K. (2017). How Does a Smart Refrigerator Work — And What's on the Horizon? - Use of Technology. Use of Technology. Retrieved 10 April 2018, from https://www.useoftechnology.com/how-a-smart-refrigerator-work-whats-on-the-horizon/

Conserve Energy Future. (2016). Causes, Effects and Solutions of Food Waste - Conserve Energy Future. Retrieved 10 April 2018, from https://www.conserve-energy-future.com/causes-effects-solutions-food-waste.php

Consultancy.uk. (2017) UK smartphone penetration continues to rise to 85% of adult population. Retrieved 10 April 2018, from https://www.consultancy.uk/news/14113/uk-smartphone-penetration-continues-to-rise-to-85-of-adult-population

The Best Business Awards. (2018) Winner's Benefits | The Best Business Awards. Retrieved 09 April 2018, from https://www.bestbusinessawards.co.uk/winners-benefits/

DiBenedetto, B. (2018). Food Waste Has a Big Impact on Climate, Water, Land and Biodiversity. Triple Pundit: People, Planet, Profit. Retrieved 10 April 2018, from https://www.triplepundit.com/2013/09/food-waste-big-time-hit-climate-water-land-biodiversity/

Bradley, J. (2018). The Importance of Promotional & Marketing Strategies. Smallbusiness.chron.com. Retrieved 10 April 2018, from http://smallbusiness.chron.com/importance-promotional-marketing-strategies-57205.html

Tucker & Farrelly, 2015, p.8 Retrieved 11th April 2018, from https://www.anzca.net/documents/2015-conf-papers/849-anzca15-pearson-mirosa-andrews-kerr/file.html

Simionato, M (2003). The Python 2.3 Method Resolution Order. Retrieved 15th of April 2018, from https://www.python.org/download/releases/2.3/mro/

"State of JavaScript 2017" report. Retrieved 15th of April 2018, from https://stateofjs.com/2017/build-tools/results/

Djirdeh, H (2018). "Let's Build a Custom Vue Router", Retrieved 18th of April 2018, from https://css-tricks.com/build-a-custom-vue-router/

(n.d.). Retrieved from https://store.arduino.cc/arduino-uno-rev3
Load Cell Amplifier HX711 Philippines Makerlab Electronics. (n.d.). Retrieved from https://www.makerlab-electronics.com/product/load-cell-amplifier-hx711/
OV7670 Camera Module. (n.d.). Retrieved from https://wiki.eprolabs.com/index.php?title=OV7670_Camera_Module
Rogers, R. M. (n.d.). WiFi Module - ESP8266. Retrieved from https://www.sparkfun.com/products/13678
Zen Cart™ Team. (n.d.). OV7670 Camera Module. Retrieved from http://www.elecfreaks.com/store/ov7670-camera-module-p-705.html