University of Kent

An IoT (Internet of Things) Food Weighing Scale for Counting and Tracking Calories

Specification and Preliminary Design

Group 8

Joshua Kolimbarides (JJK29)
Matthew Skingley (MS916)
George Cobbold (GC396)
Patrick Cullinan (PPC2)

Supervisor: Benito Sanz-Izquierdo

13/11/19

Table of Contents

1. Introduction

“Latest figures show that more than half of the UK population – 66% of men and 57% of women – are overweight or obese” [1]. “Overweight and obesity are defined as abnormal or excessive fat accumulation that presents a risk to health.” [2]. In order to lose body fat, in combination with other healthy habits such as reducing dietary sugar and exercising regularly, calories (a shorthand term for kilocalorie) should be limited. An individual should eat no more than their Recommended Daily Allowance (RDA) of calories per day, 2500kcal for men and 2000kcal for women. [3]

SensePlate is a smart IoT food scale with an integrated User Interface (UI) allowing users to track the calories and micronutrients in their diet, so that they may have a greater understanding of their diet and habits. The user interacts with the scale by placing an item on the weighing plate and choosing a foodstuff from the item menu. On the press of an “ok” button, the database will be updated with relevant data relating to their input. This database is shown interactively in a browser, where the user can track their caloric intake over time, providing them with greater awareness of how much they are consuming in reference to the RDA.

2. Functional Specification

A user places an amount of food onto the weighing plate. The user will then use the menu described below to identify that food . This menu consists of folders (henceforth known as categories) and food items. If this were not the case the user would have to scroll through a large, disorganised library of food items, which would drastically increase the time to use the scale and make it less user friendly. This data (food, weight) is sent to a connected server and pushed to a database which may hold a large number of these entries. From here, columns in this database are filled in that give useful context about that entry. These are as follows: Timestamp, when the data was received; Calories, KCal of energy stored in that amount of food; Salt, salt (mg) contained in that amount of food.

A web page (HTML file stored on the server) displays the data held in the database in a format meaningful to the user. The data will be filtered by day, so the user can see their entries to the database on any day they wish. A graph will show the calories consumed each day, so that the user may track their eating habits over time.

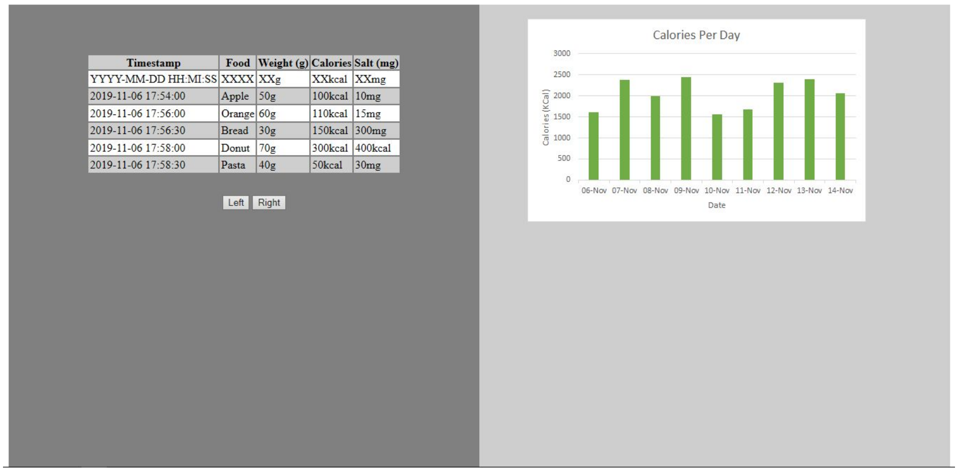

Figure 1. Mockup Web Page Design

In regards to the hardware that will be manufactured for the weighing scale, one of the pivotal components will be the load cell, in which the load cell which convert mechanical force into a measurable electrical output. For a kitchen scale with a load range between 100g to 10kg, a 4-terminal load would be suited best.

Alongside this, the UI will comprise of an LCD toggle menu. The anticipated production of the LCD scrolling menu shall show the various categories of food products and within that there will be sub menus showing the different variations of the food categories. This LCD scroll menu will be aimed so that it is user friendly for individuals to use.

On completion of the project, the following deliverables will be produced:

- A zeroing weighing scale. (Hardware)
- An LCD toggle menu (Hardware)
- A server capable of processing data sent from the arduino microcontroller. (Software)
- An end output webpage for user display, making use of the timestamp to show daily intake and comparing it to prior days. (Software)
- A reference database containing the food categories and nutritional values that the user will request. (Software)

Design requirements

Precision expectations

The reference database used for storing foods needs to have accurate nutritional values. To ensure this, the information is being sourced from the United States Department Of Agriculture (USDA)[5]. This organisation is responsible for researching a vast and expansive array of various food groups and has to rigorously examine a vast amount of different samples to ascertain accurate values. The food records in we will use in our database will be a uniform one gram (1g). This is so that a nutritional value can be quickly yielded and passed over to the webpage for display to the user. This could be a potential risk with regard to compromising our data values, so careful consideration will have to be taken, in order to ensure the converted data has the correct ratio to the source values.

Power requirements

The Arduino microcontroller receives power through the Raspberry Pi via USB. The maximum average load current for the raspberry pi is 1000ma at a voltage of 5.1V, therefore the power supply should be capable of outputting 5.1 Watts. Necessary precautions will be taken to ensure that this value is not exceeded.

The following table includes the current values and subsequent total power the circuit will have to be capable of handling:

Component	Current Draw (mA)	Power (mW)
LCD Display 1	1.1	5.61
LCD Display 2	1.1	5.61
LED	20	102
Arduino UNO	40	204
Load Cell	2	10.2
Raspberry Pi Zero W (H)	250	1275
Total	314.2	1603

Table 1. Power requirements

To calculate total wattage, the total current draw is multiplied by the source wattage to obtain a value of 1.603 watts.

Security Authentication

To ensure that the server is optimally secured, it will be necessary to undergo server hardening checks. This will consist of:

- Using up-to-date software (Long Term Stable release)
- Mandatory SSH Key use
- Restricted root access
- Maintaining good etiquette & practises when adding new data
- Limiting access to users and those responsible for upkeep
- System monitoring
- Regular backups

Departures from briefing and supervisor comments

One of the notable departures from the briefing that the group took was to send the data from the Arduino microcontroller to the Raspberry Pi via a serial connection as opposed to bluetooth. Bluetooth connections are subject to latency delays and are still subject to interference dependent on neighbouring bluetooth devices. The reliability of serial connections, alongside the capabilities of the devices the group has chosen in this project, made transferring the data via serial the far more compelling option.

Another departure the group took from the briefing was programming the user interface for the food results with Javascript and PHP as opposed to the prescribed Java. This allows for greater functionality between the databases the group will be using as they are in SQL.

Thirdly, although the briefing mentions both a phone app and a web app to display values to the user, the group opted for a natively run web page. This is advantageous because the functionality of the web development languages being used makes it easier to output the content to multiple devices and browsers. A web app can be displayed on mobile browsers, and is just as legible when responsive styling is introduced.

3. Technical Specification

Microcontroller

Project members have experience using the Arduino microcontroller environment so it makes a good choice for gathering user inputs. Another microcontroller could be used, but the Arduino is chosen because of these features in relation to its price: high core clock speed, C like code (easy to write and read), speed of compilation and deployment, small form factor and enormous amount of online resources and project documentation

Server

We have chosen to use a local server to store the database of entries. This is chosen as opposed to a cloud server like Google's Firebase so that we have more freedom to work on a Local Area Network (LAN) connection not connected to the internet. Therefore drops in internet upload and download speed will not affect us. The device may be used in any country, including those where Google is blocked. A Raspberry Pi is chosen because of its specifications and price in relation to its competitors. These are:

High single core clock speed, adequate working memory size, large storage capacity with inexpensive MicroSD card, WiFi capability, USB and HDMI interfacing, Linux ready with support for many distros (Operating systems built on Linux) and low power consumption.

The Linux "Distro" we will be using is called Ubuntu Core 18, an IoT focussed release of Ubuntu 18.04 LTS (Long Term Support). Ubuntu Core is chosen because of its numerous safety features and widespread industry adoption in the IoT market [ref eclipse IDE survey]. These safety features include immutable and sandboxed applications (snaps) and mandatory SSH key use. Ubuntu core is also a very small install, being only 650mb on a fresh install on a virtual machine.

Note: Internet access may be used so that the device may generate an accurate timestamp for each entry.

Relational Database Software

An SQL (Structured Query Language) database is used to store entries of food items, specifically MySQL, on the aforementioned server. This is mainly because it is lightweight and free. In addition, MySQL is optimised for use with PHP, so web development with database interaction is both simple and heavily documented. SQL is relational, meaning that SQL tables "talk" to each other and can be linked to provide useful functionality. In our project, we will need a main entry database to store items, weights and timestamps (A format of time in MySQL) and also a lookup database that uses an algorithm to return the calories and salt in these entries.

Display

The input for the item selection is displayed on a two part LCD (Liquid Crystal Display) menu where the user can cycle through folders (categories of food) and food items. This input system will be comprised of two separate vertically aligned LCD panels each with two navigation buttons for selection.

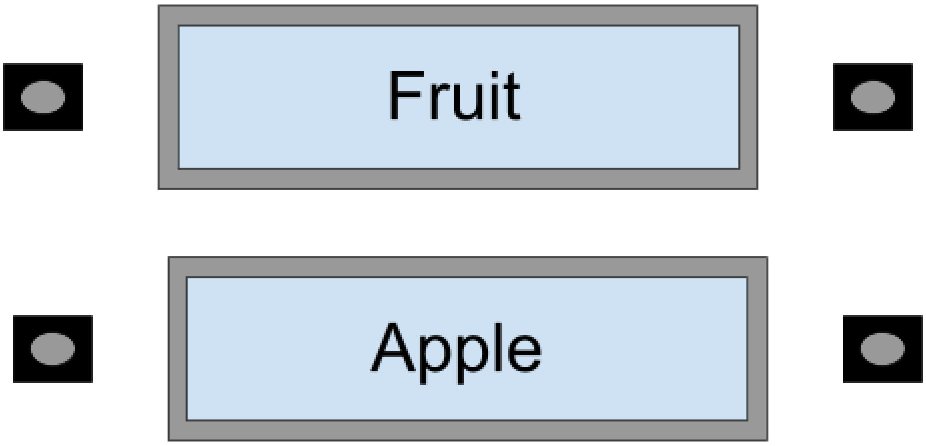

Figure 2. LCD panel layout

We will be using a pair of 1602A-1 boards for the above implementation, as shown in figure 2. It has an associated Arduino library called LiquidCrystal, which prints strings (lists of characters) to the display with minimal code footprint. The display allows 16 characters per line, of which there are two. For this reason, every category and food must be identifiable by a string of 32 characters or less.

Figure 3. 1602A-1 LCD Panel

4. Data Flow

The block diagram Figure 4, shows the elements compromising SensePlate. The user has two methods with which to interface with the device. The first interaction method is through the scale itself. As previously described, the user will place food items on the weighing plate and use an interactive menu system to select that foodstuff. The second method of interaction is an interactive website that gives the user insight into their habits and the data they have collected via the scale. This GUI takes the form of a table, showing the contents of the internal database, and a graph window that help users visualise the data that has been gathered. The database will be SQL based, hence objects on the page can call scripts written in PHP to manipulate and display the data. This enables the following features.

- User may view the human-readable data stored in the database. This means that the column containing the primary key (A feature of SQL databases necessary for their function) will be hidden.
- User may delete database rows.
- User may filter data by day.
- User may scroll through these days.
- The day's data will be graphed alongside the table.

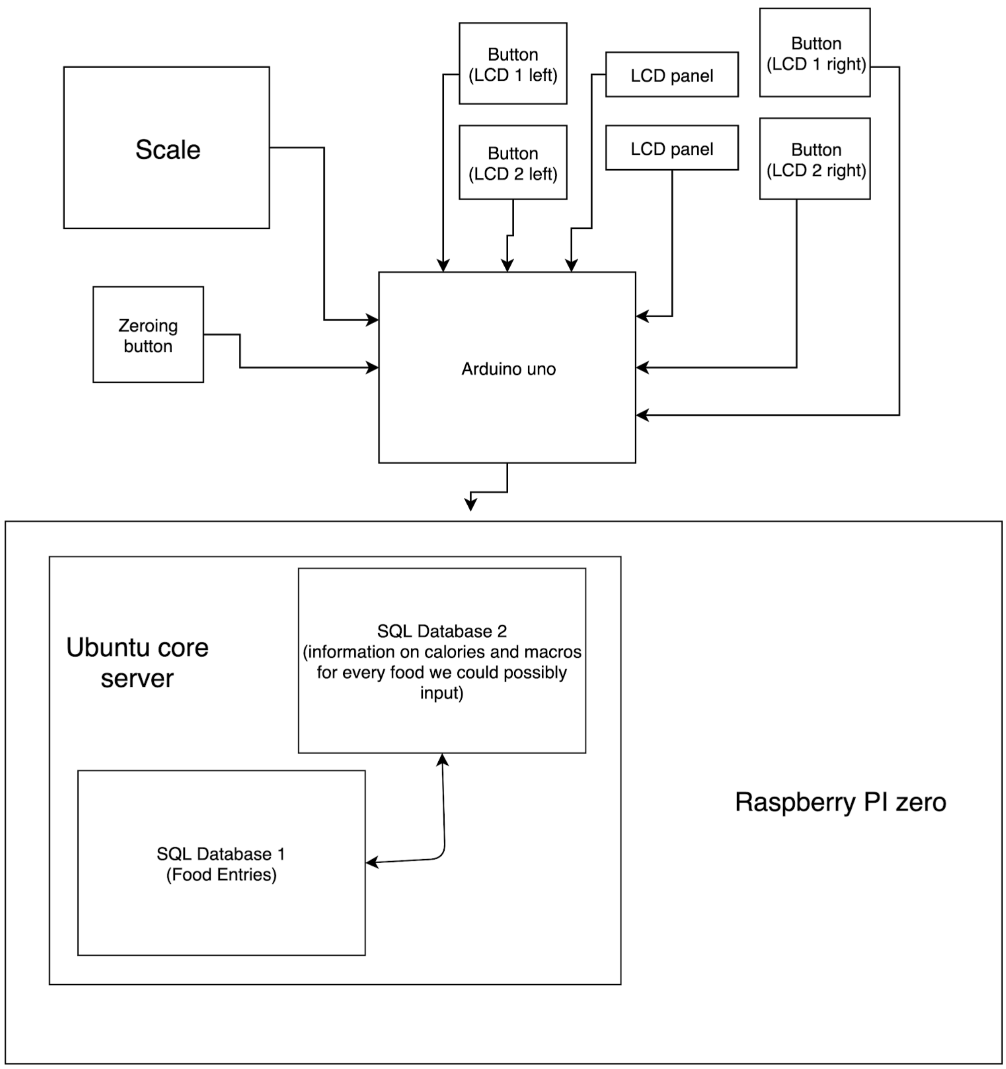

Figure 4. Block diagram

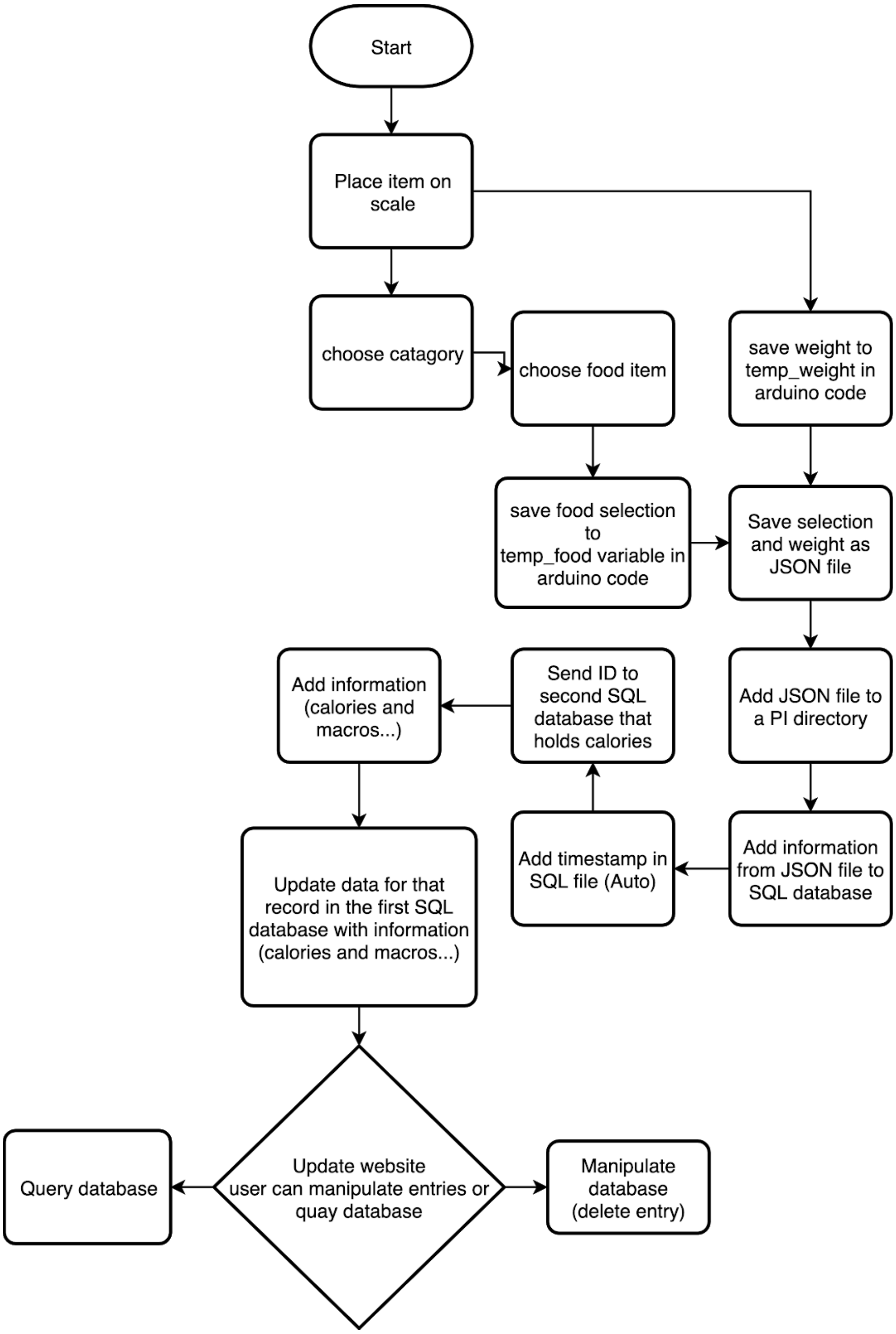

Figure 5. Software flowchart

5. Project Plan

a. Breakdown of tasks

Phase 1: Design and research

- Choose project from list of briefs (Group)
  Make a form on Google Forms so that project members may vote on which project to undertake based on their confidence with the necessary tasks. The top five projects are filtered to one by a group discussion, SensePlate being the overall winner. SensePlate was chosen because: George is comfortable with the hardware aspect and can demonstrate the skills he has learnt. Matthew has an interest in IoT and is excited to be an integral part of an IoT device deployment. Josh is not only good at Arduino programming, but also has experience in serial data transmission in industry. Patrick has a penchant for programming languages and is adaptable, being able to work with many solutions based on specification and the most appropriate implementation for the given hardware.
- Task allocation (Group)
  George is allocated to creating the weighing plate.
  Josh is allocated to creating the UI (User Interface) for the scale and is in charge of data transmission to the server.
  Matthew is allocated to storing the data from the weighing scale in a database and creating a system to display that database to the user.
  Patrick is allocated to data retrieval from a secondary database, to add calories and other data to the main database. He will also be working on displaying and manipulating the database.
- Specification (Group)
  Report that provides information on research, design, control flow, time allocation, task allocation and budget.

  - Front Page (Group)
  - Table of contents (Josh)
  - Introduction (Matthew)
  - Technical Specification(Josh, Matthew, Patrick)
  - Functional Specification Part 1 Front End (Matthew, Josh, Patrick)
  - Functional Specification Part 2 Back End (George)
  - Data Flow (Josh, Matthew)
  - Block Diagram (Josh)
  - Software Flowchart (Josh)
  - Project Plan (Matthew)
  - Hardware Implementation (Josh, George, Matthew)

- Software Implementation (Matthew)
- Testing and Evaluation (Matthew)
- Final report (Josh)
- Gantt Chart (Josh)
- Costs (Josh, George)
- Group management (Josh, Patrick)
- Referencing (Josh)

Functional Specification Part 1, Hardware:

This section illustrates the hardware development milestones for the project. The individual(s) responsible for these tasks is listed.

(George)

- Real time weight sent to Arduino and stored in a variable.
- Weight can be zeroed with a button.
- Weight is accurate to test weights.

(Josh)

- LCD displays print test strings.
- LCD display 1 prints categories, LCD display 2 prints food items.
- Both menus can be scrolled through using left and right buttons.
- Scrolling the top menu resets the position of the bottom menu.
- Bottom menu shows only foods in the category displayed by the top menu.
- Bottom menu food item can be stored in a new variable when button "ok" is pressed.
- Production of a PCB for the front panel.

(Josh, George)

- Weight and Food Item can be converted to a JSON file.
- JSON file can be saved to a directory on the Raspberry Pi.

Functional Specification Part 2, Software:

This section illustrates the software development milestones for the project. The individual(s) responsible for these tasks is listed.

(Josh)

- Python script to generate a JSON file, simulating an input with suitable random values.

(Matthew)

- Prototype a Linux server environment on a virtual machine.
- Create an SQL database on the server.
- Port the server to on a microcomputer.
- Create a one way data link from the Arduino to the server.
- Handle JSON files added to the directory from the Arduino, add them to the database via a script.

(Patrick)

- Add data to the database relevant to the input data (Calories, Salt etc.).
- Building the infrastructure necessary to deal with queries.

(Matthew, Patrick)

- Develop a website that can display this data onto a webpage.
- Establish a two-way link between this webpage and the database, so that the user may edit values.
- Graph and filter this data.

Phase 3: Testing and evaluation

Testing involves using the SensePlate from input to output. SensePlate will be tested using various foodstuffs of various types over a period. If the project is working the following will be true: (The individual(s) that will carry out the test is listed next to each test.)

1. (George) The user can weigh items on the scale. Weight will be 0g when there is nothing on the scale. Weight is accurate with test weights.
2. (George, Josh) The user can scroll through folders of food types and food types in those folders.
3. (Josh) When the "ok" button is pressed, the JSON file described above is sent to the server with the weight of the food and a string of the food's name.
4. (Matthew, Josh) JSON files are added to the database reliably.
5. (Matthew, Josh) The timestamp from a transmission is recorded as a column.
6. (Matthew) The database is secure and cannot be reset accidentally.
7. (Patrick) The database can be viewed on a webpage.
8. (Patrick) Database values can be deleted via a script on the website.
9. (Matthew) The database can be filtered by day.
10. (Matthew, Patrick) The data on, and only on, the screen at any given time is graphed reliably.

Phase 4: Final report and documentation

- Group Presentation (Group)

  The project will be present. Josh will be in charge of this presentation but each member will explain their contribution to the project.
- Demonstration of project (Group)

  The product will be demonstrated by placing different items on the scale and observing the updated website.
- Documentation and final report (Group)

  A final report will be written by the group in a similar fashion as the specification .
- Individual report (Josh, Matthew, George, Patrick)

  This report will mainly consist of what each individual worked on.

6. b. Program of work: Gantt Chart

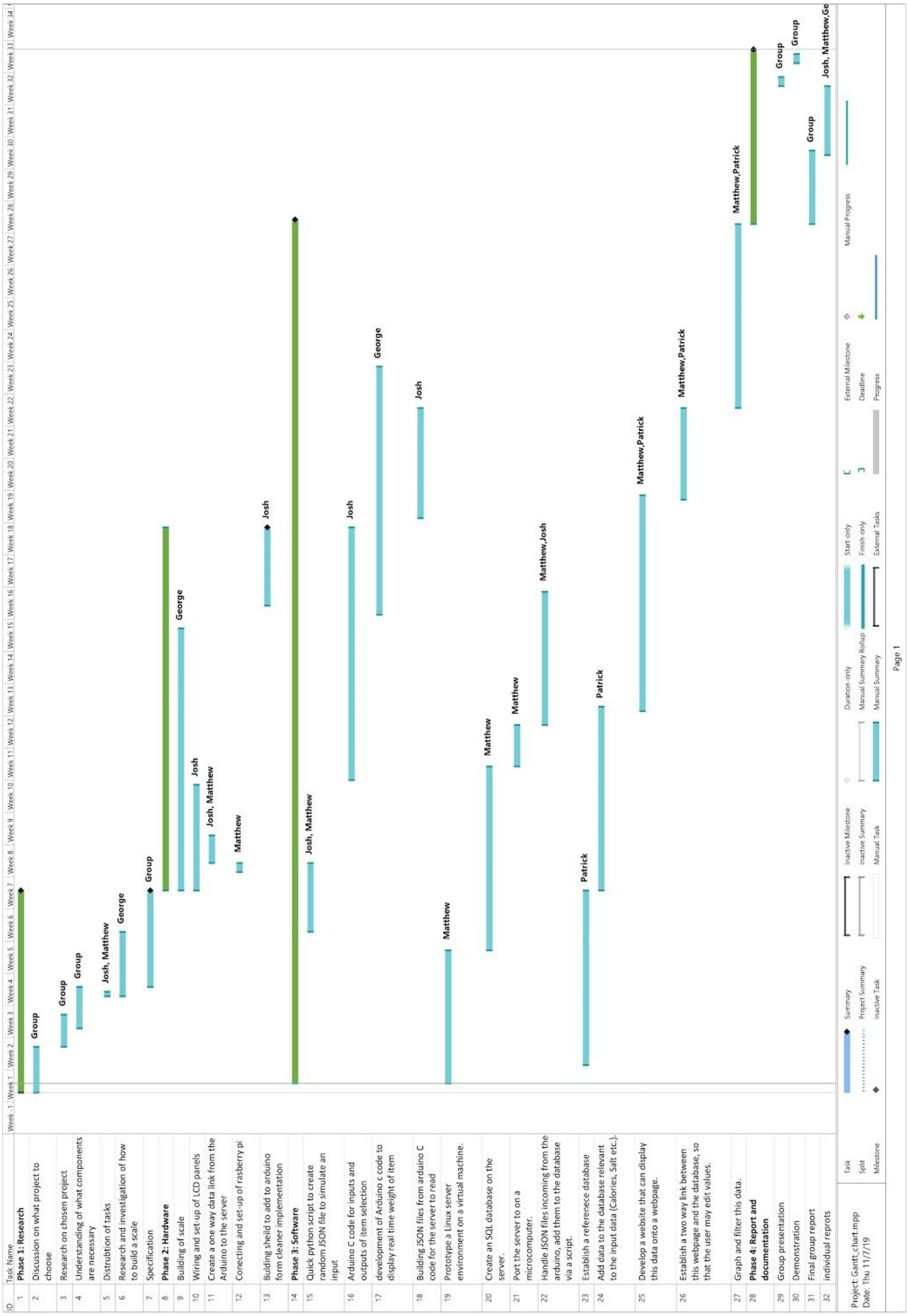

Figure 6. Gantt chart

7. Costs

	Part	Purchase from	Amount	Cost (each inc. VAT)	Total
Input module (LCD)	1602 LCD IIC/I2C/TWI Module	https://hobbycomponents.com/our-brand-exclusives/920-hobby-components-1602-smart-lcd	2	£5.98	£11.96
	Buttons	https://hobbycomponents.com/switches/379-12mm-tactile-push-button-switch-pack-of-10	1 pack of 10	£1.69	£1.69
	Jumper wires	https://www.rapidonline.com/rapid-jw-003-breadboard-jumper-wires-bundle-of-75-34-0673	1 pack with 75 wires	£4.14	£4.14
Raspberry Pi (Server)	Raspberry Pi Zero WH	https://www.rapidonline.com/raspberry-pi-zero-wh-raspberry-pi-zero-wireless-with-pre-soldered-header-75-1030	1	£12.98	£12.98
Load Cell (Scale)	Load Cell	https://www.amazon.co.uk/Weight-Sensor-Weighing-Arduino-Raspberry-Black/dp/B07G2BDJQ5/ref=sr_1_1_sspa?crid=1QVEU3C545US7&keywords=load+cell&qid=1573153169&sprefix=load+%2Caps%2C865&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyMEVDM1o1VU9UUTlTJmVuY3J5cHRlZElkPUEwOTg1MzAzMzhFTzkwMTcxN0ozVSZlbmNyeXB0ZWRBZElkPUEwMjIwNjg4MjgwUVczWEkzQzZZNiZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=	1	£8.50	£8.50
				Total	£39.25

Table 2. List of components and costs

8. Group Management

Project Manager: Josh, Matthew
Group Members: George, Josh, Matthew, Patrick

Date of Meeting	Topics discussed
16/10/19	Discussion of project briefs
17/10/19	Group leader and chosen project
20/10/19	Research of scale and input devices
25/10/19	Research of database and server
05/11/19	Specification and Preliminary Design Report
11/11/19	Report feedback

Table 3. Meeting log

<u>Measures taken to ensure the success of the project:</u>

- Google form created for everyone to vote on a project to reduce the risk of arguments such that the most voted project is chosen.
- Whatsapp group created for communication between group members. Additionally, it provided members a platform to voice their opinions regarding decisions the group had to make.
- Regular meetings with Benito Sanz-Izquierdo, to ensure targets are being met (Table 3).
- Regular meetings between ourselves to ensure everyone is happy with their current task.
- Cloud based documents used, to encourage collaboration between group members in real time - notably for the Specification.

For the sake of assigning tasks and assuring that the group would undergo the project harmoniously, it was necessary to appoint a group spokesman. For this, we appointed Josh. Josh is in charge of organising meetings and prepares the presentation material used. Meetings are done on a regular basis. As mentioned previously, there is the group meeting conducted with our supervisor which is scheduled every wednesday at 2pm. Additionally there is a meeting between group members on Tuesdays at 1pm, which we outline what progress has been made to date.

In order to optimise productivity, it was necessary to breakdown the tasks involved in the project and assigning group members to it in order to cater to their strengths. This allows for group members to be specialised in the tasks they perform and increases the groups efficiency with regard to managing workload.

9. References

[1] NHS Choices., (2019). Eat Well [online]. National Health Service. [Viewed 5th Nov. 2019] Available from: https://www.nhs.uk/live-well/eat-well/cut-down-on-your-calories

[2] Who | Obesity., (2014). Health Topics - Obesity. [online]. World Health Organisation. [Viewed 5th Nov. 2019]
Available from: https://www.nhs.uk/live-well/eat-well/cut-down-on-your-calories

[3] NHS Choices., (2019). Cut down on your calories - Eat well. [online]. National Health Service. [Viewed 7th Nov. 2019]
Available from: https://www.nhs.uk/live-well/eat-well/cut-down-on-your-calories

[4] United States Department Of Agriculture (USDA) Standard Reference 28 Food Table (2019). [Viewed 8th Nov. 2019] Available from: https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/methods-and-application-of-food-composition-laboratory/mafcl-site-pages/sr17-sr28/

[5] Iot.eclipse.org. (2019). [online].[Viewed 8th Nov. 2019]. Available from: https://iot.eclipse.org/resources/iot-developer-survey/iot-developer-survey-2019.pdf

Figure 1 | Mock-up design of website. Created by Matthew Skingley

Figure 2 | Layout of LCD Panels and buttons. Created by Josh Kolimbarides

Figure 3 | LCD Display Sales Market Report., (2019) [Viewed 7th Nov. 2019]
Available from: https://cryptonewsplus.com/tag/lcd-display-sales-market

Figure 4 | Block diagram. Created by Josh Kolimbarides

Figure 5 | Process flow diagram. Created by Josh Kolimbarides

Figure 6 | Gantt chart. Created by Josh Kolimbarides

Table 1 | Power requirements. Created by Patrick Cullinan

Table 2 | List of components and costs. Created by Josh Kolimbarides and George Cobbold

Table 3 | Meeting log. Created by Josh Kolimbarides