**Pocket Chef**

****GCA1 IoT – Dylan Smyth, Joshua Donnelly, Luke Clarke

**Introduction**

In this project we will discuss what our project is? why this project? And how are we going to achieve our goals. We outline what hardware we will be using, and how they will be implemented in the final project. We showcase our early prototype and detail how we adapted that prototype as a result of feedback from our testing.

**What is Pocket Chef?**

Pocket chef is a tool for inexperienced cooks, such as students, to learn the difficult but necessary skill of cooking. Cooking can be dangerous, so we have decided to focus on helping specifically with high-risk food (e.g. Chicken or pork). Pocket Chef connects with a thermometer through a connection with the Raspberry Pi over Flask. The application is simple and intuitive, but informative and helpful.

**Hardware List**

(Luke Clarke)

The hardware that will be required to develop this application:

1. Raspberry PI
2. Li-polymer battery PI HAT
3. Li-polymer battery -> if not included in HAT type
4. Led
5. Male / Male connector
6. Male / Female connector
7. Female / Female connector
8. Female Jack Male Plug Panel Mount
9. Waterproof Thermometer
10. Breadboard

**Total: roughly 60€**

Rough Pi Zero breadboard schematic

**A battery and circuit board

Description automatically generated**

**Data, Data Storage and Data Processing**

(Dylan Smyth)

Data, Data Storage and Data Processing

Data Gathered by Sensors

Our IoT cooking system uses a Bluetooth thermometer connected to a Raspberry Pi. The thermometer measures the internal temperature of meat during cooking and the Pi acts as a bridge between the thermometer, the Internet and the database. The system may also use a camera module for photo-based doneness detection but we will try to use the users phone if possible.

1. Bluetooth Thermometer

* Data collected:
  + Internal meat temperature
  + Battery level of the thermo
  + Timestamp of each reading
* How it works:  
  The thermometer uses a thermistor sensor inside the metal probe. A thermistor changes resistance with temperature. The onboard chip converts this change into a digital temperature reading and sends it over bluetooth to the Raspberry Pi.
* Frequency of readings:
  + Default: 1 reading every 5 seconds
  + High accuracy mode: 1 reading every 1 second
  + Low power mode: 1 reading every 30 seconds – 1 minute for slow cookers
* Sample data format:
* {
* device\_id: Dylan’s Pi,
* thermometer\_id: bluetooth\_thermometer,
* temperature\_c: 63.5,
* battery\_percent: 80
* }

2. Raspberry Pi Camera Module

* Data collected: Photos of the meat’s surface during cooking to visually detect rawness of the meat.
* How it works: The users takes a picture of the meat inside and out and it gets compared to photos in the database to confirm if its cooking right or done.
* Frequency of capture:
  + Whenever the internal temperature rises by 5°C
  + User is pinged to take photo if they don’t know if meat is cooked or not
* Stored data:  
  Each photo is stored with a timestamp and associated cooking session ID.

3. Third-Party API Data

* Data collected:
  + Recommended cooking temperatures
  + Recipe details and images
* Source: Third-party cooking/food safety APIs (BBC Foods)
* Use: These are stored in the local database and used to give the user guidance on safe cooking temperatures for specific meats.

**Data Storage**

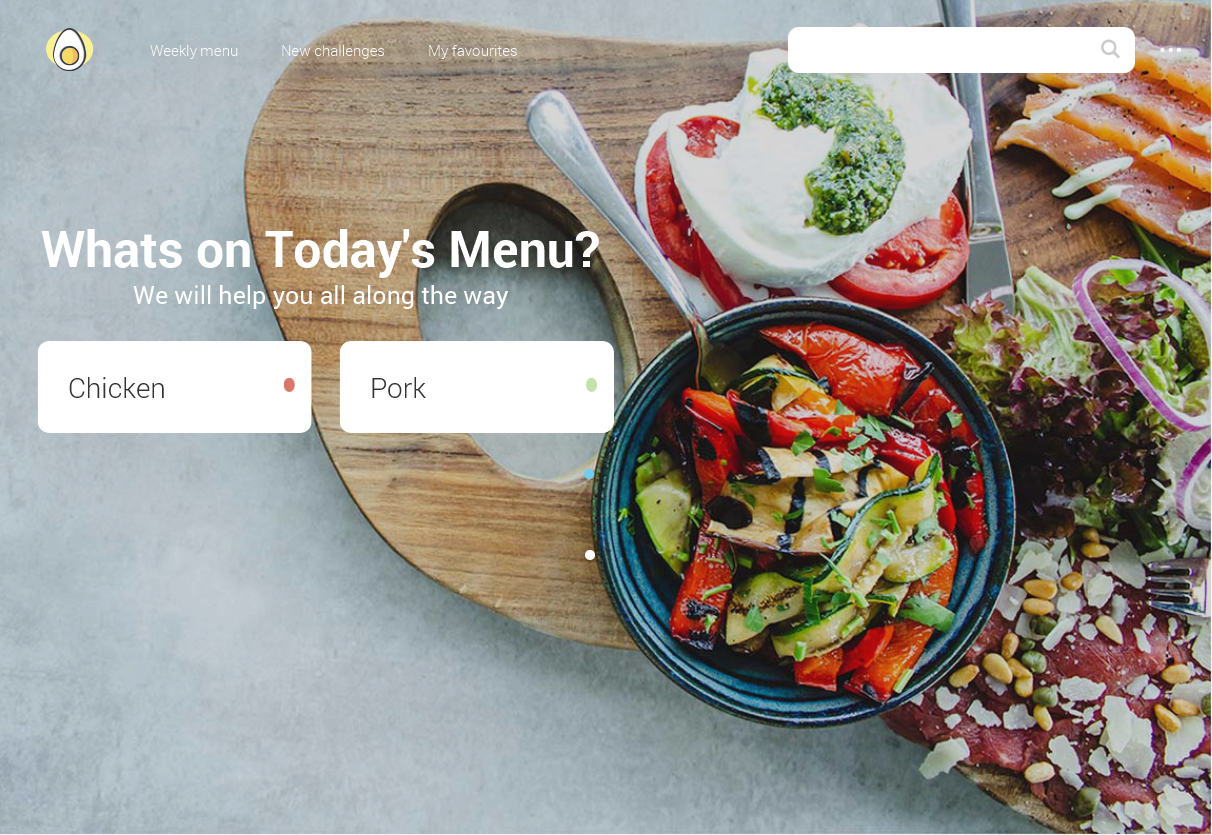
All sensor data and cooking records are stored in a MySQL relational database. MySQL is used because it provides data integrity, supports structured queries. It is also the database software we are most familiar with.

**The UI, User and Testing**

(Joshua Donnelly)

**Ui Mock-up**

**Screen 1** – The landing page. Consists of a simple straight forward, with a choice between the high-risk meats that we have decided to focus on. Its important for all of these screens to follow our usability guideline, Simple and intuitive.

****

A screenshot of a computer

AI-generated content may be incorrect.

**Screen 2** – App responds with this screen once meat is selected. The user is requested to connect the necessary hardware. The design is simple and clear what is requested from the user.