Project NutrientMatch

Authors: Surya Chandramouleeswaran, Grace Liu, Steven Zeng
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*— NutrientMatch is a system that seeks to track inventory of foods and calorie consumption in peoples' daily lives. The aim of this device is to improve the state of the art pantries and refrigerators which currently do not have inventory capabilities and/or calorie-tracking functions. The product will consist of a wooden box with a scale at the center. Additionally, the box will also have a door with a camera for scanning purposes as well as a ring light to illuminate the image for proper image recognition. The system is designed to improve the daily nutritional intake of consumers and provide a stress-free food tracking system that can be easily integrated into larger smart food appliances.**

***Index Terms*—Arduino Integration, Camera Processing, Front-end Design, Image Classification and Recognition, Label Reading, SQL Database**

1 INTRODUCTION

Calorie tracking has become tedious for a wide range of people, from body builders to the average person trying to live a healthier lifestyle in this age where physical wellness is heavily emphasized. The conventional method to track calories is to weigh every ounce of consumed food and to use applications like MyFitnessPal to compute calories by hand. Likewise, a lot of the food we buy is often forgotten and sometimes left in the back of the fridge or depths of the pantry that will eventually go to waste. It is quite difficult and tedious to constantly keep track of how much food one has and also who that food belongs to, especially when living with a group of people. Being a health conscious group who has also experienced the frustrations of family members and friends taking food from a shared fridge or pantry, we seek to create an inventory tracking product which accumulates daily caloric intake for users.

Our product incorporates a scale and camera into a physical box with the output forwarded to a website to display inventory and caloric intake. The expanded goal of our project beyond the scopes of the Capstone curriculum is to incorporate our product into pantries and refrigerators while implementing macro-nutrient tracking. However, for the scope of this course, we aim to create a product that can correctly distinguish between two food categories: fruits and canned foods. Within the fruit category, our product will classify what type of fruit, obtain the weight in grams, and compute caloric values using the weight and online information. And within the canned food category, our product will use text recognition to obtain the name of the product and scan the nutritional label for the calories per serving and total calories. This data will be stored in a SQL database to keep track of inventory. When a user takes out an item to consume, it will be removed from the database and the caloric value will be accumulated into the user's daily calorie tally.

Our product is particularly valuable for two primary reasons. First, it is a minimalist inventory system that can be incorporated into any food storage apparatus. The only alternative currently on the market is Samsung smart fridges which seem to exhibit an extremely high price point. Likewise, people can manually keep a daily inventory, but this has been proven to be a tiring and annoying task. Second, our product allows for users to know roughly how many calories they consume a day from their pantry or fridge. Currently, there is no product that directly accumulates the amount of calories consumed for a given person. At best, most smart appliances have technology to display the caloric intake of various products and accumulate the sum for all users. Our product will require users to log in and track their own intake as well as their own food inventory. This provides a better system to track inventory ownership as well and improve the experience of sharing a smart appliance with others.

Overall, we expect to create an MVP that consists of image recognition and classification as well as database integration between a website, an Arduino, and periphery sensors (camera and scale). Our project incorporates the ECE areas of software systems and circuit design. Our group is very excited to learn more about the design process that goes into making a real-world product and the challenges that arise with it. Moreover, we seek to gain a more practical experience in circuit design, ML integration, and secure web development that will extend beyond our time here at Carnegie Mellon University and into industry and research for future use.

2 USE-CASE REQUIREMENTS

With an increase in physical wellness trends, we envision creating a product that not only helps users keep count of their daily calorie intake in a simple manner but also allows for inventory tracking when a storage appliance is used among multiple people. As previously mentioned, problems arise when food tracking becomes a tedious task, especially for those with relatively busy lifestyles. This was the main factor that influenced the objectives of our product. Recent statistics have shown that 41.9% adults in the United States are obese. In addition to promoting a healthier lifestyle, the shockingly high amount of spoiled food items thrown out per week should also motivate more people to reduce waste during their daily consumption of foods and beverages.

The Use-Case requirements are delineated into a both a physical design experience and the intended effect on users of our product. This section is intended to provide an overview of such processes, and is more focused on the desired experiences from the consumer perspective; subsequent sections are dedicated to exploring design choices in greater detail.

2.1 Structural Overview

Our use-case requirements can be defined both from the design perspective and the user experience we want people to have with NutrientMatch. Structurally, our MVP will be a wooden cubic structure with an edge length of 2 feet that holds all of the components of our design together. In the first case scenario when users want to log canned foods, they can hold up their item to a 2 megapixel, 15 frames per second (fps) video camera for image capturing and label reading. An ambient light source of 55 Watts, covered with a diffuse plastering material and dimming capabilities to enhance the user experience will be placed around the camera and provide requisite illumination for the object for effective recognition. On the other hand, when users want to log fruits to their food inventory, they will follow the same steps but also place their item on the scale where the weight reading will be forwarded to the ESP8226 WiFi module for backend calorie calculations. This reading will be done automatically once the scale senses weight on it. Both of these image recognition cases require users to hold their items around 12-18 inches away from the camera for accurate classification.

2.2 Accessibility and Design Implications

To ensure that our product meets the correct market and can target users who are highly interested in a more elevated food tracking experience, we explored various public safety, safety, and welfare, as well as global, cultural, social, environmental, and economic factors to help us initially identify our use-case requirements.

In regards to public welfare, our main goal is to help users improve their overall health through better calorie tracking habits. As people can become forgetful during busier days while tracking their food inventory, our product also aims to reduce user stress levels by keeping track of every item that enters their fridge. We have ambitious goals of reducing general food waste and improving people's lifestyles through a better food tracking system. With respect to the safety aspect, our product will prioritize the safety of our users by ensuring the base of our structure is built sturdy along with well built components of the camera, light, and scale attached to it through a series of vigorous testing plans. The light will also have enough dimming capabilities so users won't be blinded by such component.

In consideration of social factors, we want to create our product under the condition that it can be accommodating for people of different social groups, whether it is regarding dietary preferences or medical concerns, our algorithm will perform with minimal bias during its classification process. This accessibility is achieved through a detailed image classification training process and a strong user interface that allows for the selection of various preferences such as dietary restrictions and a transparent sourcing of nutritional information. We are building such a product to improve the average lifestyle of a user, but in addition to this, little systematic bias is key to our emphasis on being inclusive and creating something for everyone to comfortably utilize.

Lastly, with consideration to economic factors including production, distribution, or consumption of goods and services, concerns arose when we found out the food waste generated daily in the United States during our extensive research process. This does not only pose sustainability challenges economically, but also contributes to a substantial amount of landfill waste that would require extensive efforts to clean up and maintain. While the environment is being harmed, taxpayers are also paying for the damage. Our solution not only concerns the distribution of foods but also aims to help users reduce their waste during their process of food purchasing and consumption as well. By providing accurate nutritional information, from label reading to food classification, along with better control over food inventory, our solution helps users make better use of food items at hand and decrease the number of their food items going to waste.

We anticipate building a product that will directly address calorie tracking concerns but also provide diverse options for users and make a positive impact on the environment. Careful consideration of design decisions and their impact on the environment and consumer base we intend to target is key to developing a tailored product. Thus, surverys and incorporating active user feedback is an important aspect of our design process as we seek to optimize our product's performance and accessibility.

3 PRINCIPLE OF OPERATION

Fig. 10 on the final pages of this design plan provides a block diagram of our product and its components. In summary, there are two peripheral sensors (the camera and the scale) which will forward outputs to an Arduino to synchronize with the Cloud and a SQL database. The web application will interact with the Cloud and the database to facilitate user interaction. We encourage readers of this project report to consult the back pages for a detailed view of the block diagram description of the project.

Fig. 1 below represents the physical product that we will build to engage with the web application user interface and Cloud server.

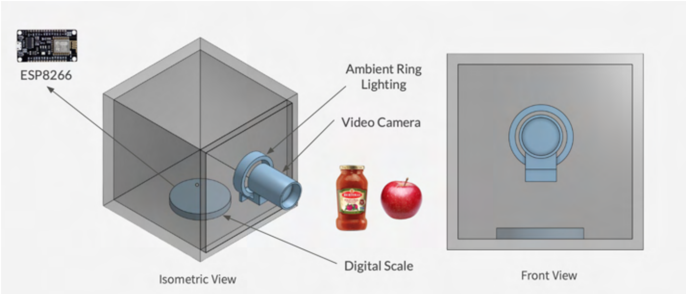

Figure 1: CAD Diagram

The Arduino will connect to the scale and camera to populate the database. After obtaining the scale reading and images of the item (front and back), the Arduino will store these values in the database followed by computations of caloric value.

User authentication and management will be handled through Google OAuth service, and separate registration and login pages. To be able to login, users will need to first register using a registration page that also gives the option of logging in through OAuth as shown by Fig. 2:

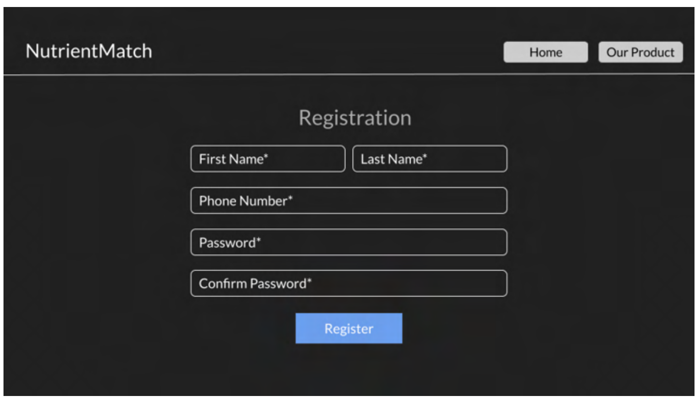

Figure 2: Registration Page

For users who have created profiles and are registered in the database, they will be navigated to our Fig. 3, scripted with React and Javascript as seen below:

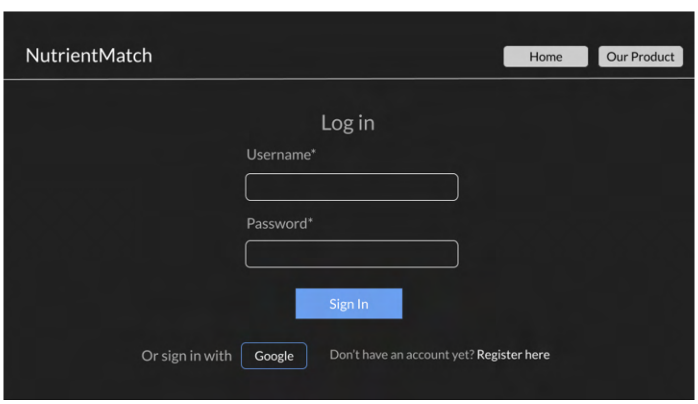

Figure 3: Login Page

After logging in, users will be able to see an inventory list for their fridge as shown by Fig. 4:

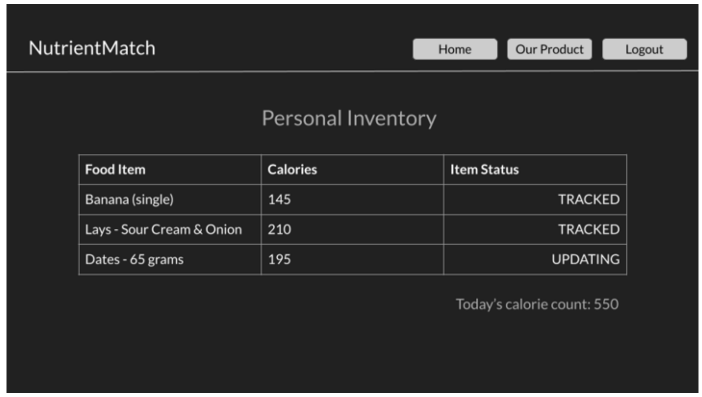

Figure 4: Inventory Page

This page operates purely on database logic implemented through Django's MVC architecture. The database is accessed through the `models` framework, and `views.py` handles the processing and rendering of data to our front end to be displayed to users.

4 DESIGN REQUIREMENTS

Fig. 5 below summarizes and divides our overarching design requirements into three components: physical, computational performance, and user experience.

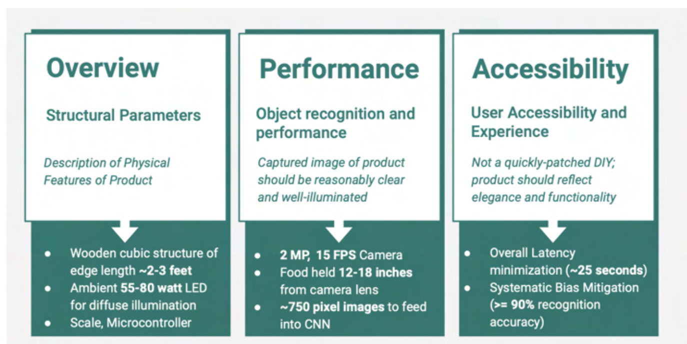

Figure 5: Summarized Design Requirements

Design requirements, from both the hardware and software perspectives, are enumerated in this section and centered around the three implementation themes in this project below.

4.1 Physical Design Requirements

- **Size Constraints:** The scale and camera setup must fit within specified dimensions not exceeding 12 inches x 12 inches x 6 inches to ensure compatibility with food storage systems. This is centered around the dimensions of the Oxo scale that will be placed within the apparatus.

- **Material Selection:** Since walnut wood has a consistent and monochromatic finish, it is our best option for the background to facilitate optimal image recognition.
- **Mounting Mechanism:** The hinge mechanism must support a weight of up to 2 kg and allow for adjustments with an angular precision of 5 degrees. This constitutes the class of food items we seek to classify in the project.

4.2 Functional Performance Considerations

- **Camera Specs:** The camera must capture images at a minimum frame rate of 30 fps with a resolution of at least 5 megapixels to support accurate image processing.
- **Database Interaction:** Database read and write operations must be completed within 25-30 seconds to ensure real-time data acquisition and user interaction. *This constitutes a full-cycle latency goal for each update operation in usage of our product.*
- **Calorie Tracking Accuracy:** The accumulated calorie amount should be within a 10% deviation of the actual consumption amount to meet the project's goal of promoting healthy eating habits.
- **Inventory Tracking:** The system must achieve 100% accuracy in inventory tracking, with a maximum allowable error rate of 1% per item.
- **Automatic Data Forwarding:** Upon detection of nonzero scale readings, the scale and camera inputs should be automatically forwarded to the database within 2 seconds.

4.3 User Experience Requirements

- **Website Responsiveness:** The web application must load within 3 seconds and maintain a response time of under 500 milliseconds for user interactions. This statistic is derived from social psychology experiments determining ideal latency times for users to not notice "lag" in application usage [5].
- **Cloud Deployment Statistics:** The web application must be deployed on a cloud infrastructure with a guaranteed uptime of 99.9% and supports concurrent user access of up to 10 users. This is a rather small number in comparison to the applications that modern databases are developed around but in our opinion strongly suffice for the project.
- **Security:** We have determined that password authentication must use bcrypt hashing with a minimum work factor of 12 and Google OAuth2 tokens must have an expiration time of 1 hour. These will be implemented through Django MVC and settings files.
- **Accessibility:** Users should be able to input corrections within 10 seconds of error detection and a confirmation prompt for data validation.

The statistics mentioned above are not arbitrary; they were developed as a function of market research and consultation with faculty and peers and are centered around a balance of optimal and achievable performance benchmarks for to help meet our design requirements.

5 DESIGN TRADE STUDIES

In this section, we will discuss the various design choices and evaluations made regarding the integration of various hardware and software components.

5.1 Camera and Light Selection

Proper camera readings are critical to proper ML classification. As a result, we did a lot of research on the choice of camera to be cost-effective and compatible with our product. We experimented with the MacBook camera (60 fps and 7MP), and the image was clear, resulting in accurate image recognition. However, the resulting image was high in resolution which would potentially lead to ineffective memory usage. As a result, we began pivoting towards a simple Arduino OV2640 camera (15 fps and 2 MP). This worked just as well as the MacBook camera with lower image quality, leading to greater memory compatibility and data retrieval from our database. Likewise, this camera syncs with the Arduino seamlessly which is another big plus during our considerations. We will still test ML algorithms directly on the MacBook camera due to the convenience of receiving an image and directly running computations locally. However, our ultimate end goal is to integrate the Arduino camera with computation done solely on the Cloud.

Another consideration involves the light to illuminate the item well so the camera can properly scan it. Our original design choice involved LED strip lights with its primary plus being low wattage; however, the image quality was not exceptional and the label reading process was negatively impacted. Additionally, the light quality was too dull. We then began testing with ring lights. The object became more clear with proper illumination. The only drawback to using this light is the need for more power, which is around 50 watts. However, for the scope of this project, we will prioritize performance over power efficiency. But in the long-run, we hope to maximize the energy usage of our product to achieve our overall goal of addressing environmental concerns.

5.2 Scale Selection and Communication

From measuring load-cell to the cloud database, low-latency communication of scale readings is essential to user caloric tracking. Such a process must be tightly-integrated with the classification process from the database perspective so that item-caloric content pairs are effectively established.

While evaluating options to achieve this, the most obvious approach to us would be wiring a microcontroller chip into the USB/COM port of the scale hardware and write scripts to sample and forward readings from such a device into the computer. The **RS-232 Serial Communication Protocol** is a standard communication method that is followed by scales to forward their data to external applications [10]. Most microcontroller ADC (analog to digital) inputs operate on TTL (transistor-transistor logic), suggesting the need for an RS232 to TTL converter as pictured below:

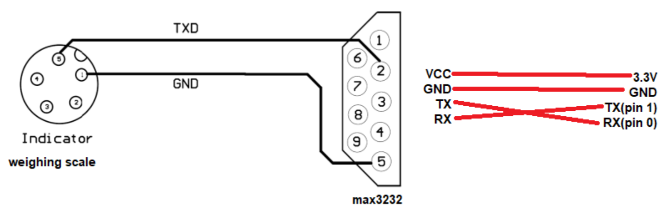

Figure 6: Wiring Schematic for RS232 → TTL Converter

Since we have established some basic expectations for scale communication architecture at this point, the details remaining are in scale selection and subsequent database integration. In the pursuit of a simple MVP, a basic scale with a COM port would suffice. In particular, we pay special attention to the *OXO Good Grips Stainless Steel Food Scale with Bluetooth Connectivity* option. Market research has yielded the following choices, enumerated by our relative preferences:

- **Aideepen ESP32-CAM W BT Board ESP32-CAM-MB:** The Aideepen ESP32-CAM board offers the convenience of both a camera module and Bluetooth functionality. It is built around an IoT (Internet of Things) approach. Learning ESP32 protocol is a requirement for this project and is well-incorporated into our proposed schedule for scale integration and development. This system-on-chip (SoC) approach is ideal for its low power consumption and respectable memory storage capacity (520 kB).

- **ESP-32 Thing Plus (SparkFun):** From our knowledge in previous electronics course experiences, SparkFun is a well-known and reputable manufacturer. This is considered an elementary option for integration: with a rich array of interfaces and expansion headers supporting I2C, SPI, and UART (these are all just jargon for interface protocols), the SparkFun option is most the flexible for incorporating various peripherals and sensors. However, since this option is general, we have found that the Aideepen option is much more tailored for our recognition regarding communication and update features that we seek to implement. We have included this product purely in consideration for its proof-of-concept potential and highly general application.

- **Xiaomi Smart Scale with Bluetooth RPI communication:** We consider this option as a stretch goal. The benefits of this approach are its embedded integration of Bluetooth technology with RPI technology, offering an entirely wireless solution for data transmission. We are intrigued by this option given the elegance and convenience of a wireless approach, and would be eager for the learning opportunity of Bluetooth management (MAC recognition and scripting, specifically). However, the main drawback to this approach is our unfamiliarity and the learning curve associated with operating a WiFi communication protocol. For the purposes of online support, simplicity, and based on prior experience, the former two options offer a better solution for meeting our MVP goal.

5.3 Database Management

Database operation is a critical component of this project as it constitutes the backbone of information recording and operation in the context of caloric tracking. During the evaluation of current database options to satisfy our MVP, we narrowed down these selections based on four factors: compatibility with existing hardware, scalability, performance, and familiarity. Among such valid options include MySQL, PostgreSQL, and MongoDB.

The following section evaluates the relative merits of each option and our rationale for selecting MySQL in the end as our database:

- **Reliability and Stability:** MySQL has a long-standing reputation for its reliability and stability in handling large datasets and concurrent transactions. Its robust architecture ensures consistent performance, making it generally more suitable for our project's needs.

- **Resources and Support:** MySQL is an extensively used relational database management system (RDBMS) with strong sources of support, both online and through CMU faculty and coursework from a Web Applications course (17-437) we have all previously taken. This ensures access to comprehensive documentation, support forums, and readily available resources for troubleshooting and optimization.

- **Scalability:** MySQL offers excellent scalability options, allowing us to accommodate for future growth and increased data volume without having to make significant architectural changes. Features like replication, sharding, and clustering demonstrate how MySQL can efficiently handle a growing user base and data-intensive operations.

- **Compatibility:** MySQL is compatible with various programming languages, frameworks, and platforms, which would allow for seamless integration with our software systems. This compatibility extends to our chosen technologies for web development and Arduino programming, ensuring smooth database integration across the project. Note that mySQL also demonstrates compatibility with ESP8266 and Raspberry Pi microcontrollers as well.

- **Performance:** MySQL is renowned for its performance optimization capabilities, including indexing, query caching, and storage engine options. By leveraging these features, we can achieve optimal performance for data retrieval, storage, and manipulation, which is important to achieve our real-time image recognition and classification tasks.

PostgreSQL and MongoDB were also well-established options for database management in our project, but there were some considerations that limited them from being viable competitors to MySQL in the end:

- **PostgreSQL:** Although PostgreSQL offers advanced features such as support for JSON data types and robust transactional capabilities [1], it is more suitable for complex data structures and applications requiring advanced querying capabilities. For our MVP, which primarily focuses on image recognition, classification, and basic data storage, the additional complexity of PostgreSQL may not be warranted. Furthermore, although PostgreSQL is built into deployment options such as Heroku through Nginx for static file management, our familiarity for deployment through Amazon Elastic Cloud (EC2) better-serves us to pursue integration with mySQL as opposed to PostgreSQL.

- **MongoDB:** MongoDB is a document-oriented NoSQL database known for its flexibility and scalability, particularly for unstructured or semi-structured data [6]. While MongoDB could potentially offer benefits in terms of flexibility and schema-less design, it may also introduce complexity in terms of maintaining data consistency and integrity, especially when dealing with relational data models or transactional requirements. Given the structured nature of our data and the relational aspects of our project, MySQL's strong support for Atomicity, Consistency, Isolation, Durability (ACID) transactions and relational integrity constraints align better with our needs.

5.4 ML Design Choices

There are 3 options we have analyzed for the ML design portion of this project. The first is the soft-margin SVM formulation where its optimization problem is formulated as follows:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \quad (1)$$

$$\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i = 1,\dots,n \quad (2)$$

$$\xi_i \geq 0, \quad \forall i = 1,\dots,n \quad (3)$$

The goal is to utilize this formula to create a decision boundary between canned foods and fruits. The slack variable, ξ_i, allows for data points that are either misclassified or between the minimum margin and decision boundary. The primary reason we plan to use this for the binary classification is because the SVM method will not be feasible for the other ML design steps. The primary benefit of using soft-margin SVM is that fact that it does not rely on all the training data but rather on support vectors which are closer to the decision boundary (on the margin or outliers). Likewise, soft-margin SVM is more resistant to outliers due to the introduction of slack variables, and canned food and fruits are such broad categories, thus resulting in the use of anticipating a moderately large number of outliers in our training set. Additionally, using kernels will allow for non-linear classification, which is why decided to choose SVM over logistic regression and Naive Bayes classification. The goal is to optimize the weight variables and intercept variable to correctly classify an image based on its position with regards to the decision boundary created by the solution to the optimization problem. Likewise, we plan to use knowledge from the course ML for Engineers (18-661) and assistance from professors to effectively utilize the benefits of this approach.

The second ML algorithm we plan to use is GoogLeNet. The primary purpose of this is to differentiate fruit (mainly between bananas, oranges, and apples). This will test our abilities in understanding neural networks and fine-tuning them to classify specific items. Using a library like this has benefits in accuracy and efficiency due to its extensive documentation and testing by professionals in the field. But the primary drawback of this approach involves complications with library compatibility and difficulty in altering pre-existing code. Likewise, GoogLeNet is used primarily for image classification, and the scope of our project is narrowed down to fruit classification. This has the potential to lead to some accuracy concerns that may be harder to repair due to such a large code base. Overall, it will be a good experience testing out a commonly used library, and we expect to get results in terms of fruit classification that will determine if this method is appropriate to integrate into our project.

The last algorithm we plan to use is ChatGPT4 API that is by far the most intuitive and easiest to implement. It requires familiarity with using the API in which we would like to gain experience in. We expect this method to yield the highest accuracy results due to the vast training data and network that ChatGPT4 utilizes. The primary con includes speed. We plan to use this primarily for label reading by inputting an image into the API to output values from labels. However, this API will merely be a backup option in the event that the previous two algorithms are inefficient, inaccurate, or both at the same time. The primary drawback of starting with this approach is a greatly reduced learning experience that we want to gain from the Capstone course.

We plan to experiment with all three of these approaches with the expectation of using all of them throughout various components of our classification pipeline. Each of these choices have their own benefits and drawbacks which we have just previously explored and anticipate as demonstrated in our risk mitigation section.

An honorable mention that we have already explored is an online library to extract valuable information off nutrition tables created by Open Food Facts [9]. The issue with this approach boils down to its low accuracy and buggy design. This library correctly read the correct protein values but was very ineffective in extracting the calorie amount. Likewise, the training data was quite limited and unrepresentative. The primary benefit of trying this design was being able to obtain valuable knowledge using various libraries like OpenCV and Tesseract. Likewise, Steven got practice fine-tuning parameters to improve results slightly. Despite reaching a dead end in this approach, we obtained invaluable information that will go a long way in the design process.

6 SYSTEM IMPLEMENTATION

The architecture for this project is centered around a modular technology approach. We plan to focus first on software development, including web development and classification, and then move onto scale hardware integration with our cloud-based database.

6.1 ML Algorithms

There will be a sequence of steps that each consist of varying ML algorithms to correctly classify and store values for an image in the database.

The first step is IC which categorizes an image into two groups: fruits or canned foods. To do this task, the product will use soft-margin SVMs [3] to properly classify images into these two groups. To find the correct decision boundary, the ML algorithm will take into account various features to best formulate the respective boundary. An example of what this would consist of can be seen below in Fig. 7.

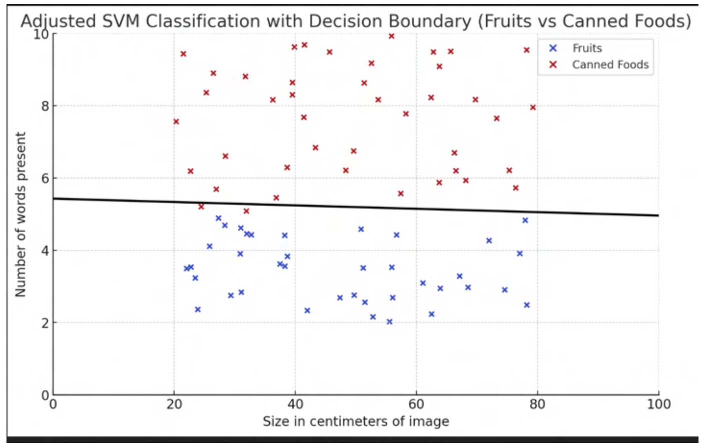

Figure 7: Classification Decision Boundary

After classifying these images into two groups, each group has its own stage of required computations. For the fruit group, the image will then be offloaded to a program that utilizes GoogLeNet, a twenty-two layer CNN, to classify the fruits into either apple, banana, or orange. The twenty second layer will be fine-tuned using OpenCV and color recognition to better differentiate between apples (red), bananas (yellow), and oranges (orange). After completing these computations, the classified result will be forwarded to the database with a corresponding weight and caloric value which will be obtained using ChatGPT's API call.

The second group, which is canned foods, will be passed into a label reading algorithm. The algorithm to conduct this functionality will also be ChatGPT's Image Reading API which will take in an image and output a formatted response that can easily be forwarded to the database. The response will consist of the number of servings, serving size, and caloric information alongside the name of the canned food product.

6.2 Scale Integration

This project necessitates seamless integration of scale readings with the MySQL database to enable efficient inventory tracking and caloric consumption accumulation. Leveraging the Aideepen ESP32-CAM board in conjunction with a simple Bluetooth scale such as the *OXO Good Grips Stainless Steel Food Scale with Bluetooth Connectivity* [2], we offer a robust solution for capturing weight measurements and transmitting them to the database:

- **Hardware Setup**: The Aideepen ESP32-CAM board is connected to the scale via General-Purpose Input/Output (GPIO) pins, allowing for direct communication between the two devices. The scale's output, typically in analog or digital form, is interfaced with the ESP32-CAM's analog or digital input pins, respectively, for real-time data acquisition.
- **Software Implementation**: Upon receiving weight measurements from the scale, the ESP32-CAM board

employs microcontroller firmware programmed in Arduino IDE to process the data and transmit it to the MySQL database. The firmware utilizes appropriate libraries and protocols to establish a secure connection with the database server and execute SQL queries for data insertion.

- **Polling and Data Transmission**: To facilitate efficient inventory tracking, the ESP32-CAM firmware implements a polling mechanism that periodically samples weight readings at predefined intervals. This aligns with faculty discussion which discussed the desire for a real-time busy-polling feature for weight sampling. The sampled data is then packaged into structured JSON or CSV format and transmitted to the MySQL database via HTTP POST requests for seamless integration with the backend system.
- **Error Handling**: Addressing failure cases is essential in any human system which is prone to failure. We look to implement error handling mechanisms within the firmware to mitigate data loss or corruption due to network instability or hardware malfunctions. Error logging and retry mechanisms ensure that failed data transmissions are logged and reattempted to maintain data integrity and reliability. Furthermore, we will partition on-chip storage to cache recent entries that may have been corrupted during database updating, once again in the event of network instability or hardware malfunction.

6.3 Web Application

We envision a strong web application to play a pivotal role in the accessibility of our product. In the design of this, such an application should provide intuitive access to inventory tracking, caloric consumption accumulation, and data visualization functionalities. Built on the Django MVC framework and utilizing MySQL for the database backend, our web application aims to deliver a seamless and responsive user experience:

- **User Authentication and Authorization**: Our web application implements robust user authentication and authorization mechanisms, leveraging Django's built-in authentication system to authenticate users and manage access control. We will require users to register accounts and log in securely to access personalized features and data.
- **Real-time Updates and Notifications**: To enhance user engagement and provide timely feedback, our web application integrates real-time updates and notifications functionality. WebSocket technology, coupled with Django Channels, enables bidirectional communication between the server and clients, facilitating instant notifications for inventory changes, weight measurements, and system updates. This offers an opportunity to build on similar concepts covered in our web applications course (17-437).
- **Front-end Compatibility**: Responsive layout principles ensure optimal viewing and usability across various home devices, including laptops, smartphones, and tablets. We will employ Bootstrap or materialize CSS frameworks to achieve consistent and visually appealing user interfaces across different platforms.

7 TEST & VALIDATION

Our exhaustive testing plan is centered around a modular unit-testing structure for each of the sub-components of the project, followed by comprehensive final product testing, as reflected in our planned schedule on the last page. This section elaborates on the specifics of our testing structure.

7.1 Tests for Scale and DB Design

With regards to validating the scale readings, we will test two metrics: latency and accuracy. Latency will be measured with an Arduino Latency Measurement Kit that we obtained online. The expected time for the database to fully update and retrieve information to be printed on the Arduino serial monitor is set to 20 seconds. Our goal is to reach that time and reduce latency as much as possible. The test will include 100 measurements of various weights, and the names will be hard coded to asynchronously test without needing the ML classification algorithm to work fully yet. Then, the accuracy of the scale reading that is displayed on the serial monitor from the SQL database will be analyzed for correctness with the goal of 100% accuracy.

7.2 Tests for ML Design

Regarding the ML portion of the design, there will be multiple components to execute and test. The first component is the differentiation between fruits and canned food using SVMs. The desired result is 90% which was obtained through careful analysis of similar ML projects [12] and the limitations of SVMs. After training the soft-margin SVM model, Fig. 8 below demonstrates the optimal C value which is used in determining how much slack our model gives to outliers or points that fall between the decision boundary and the support vectors [3]. These points could be misclassified data values or points that are not as confidently classified. As seen by Fig. 8, the optimal C value appears to be 0.005 with around 86% accuracy. Our hope is to fine-tune the algorithm by adjusting the various weights given to the two features as well as experimenting with different features like weight and light exposure to achieve a 90% accuracy or higher.

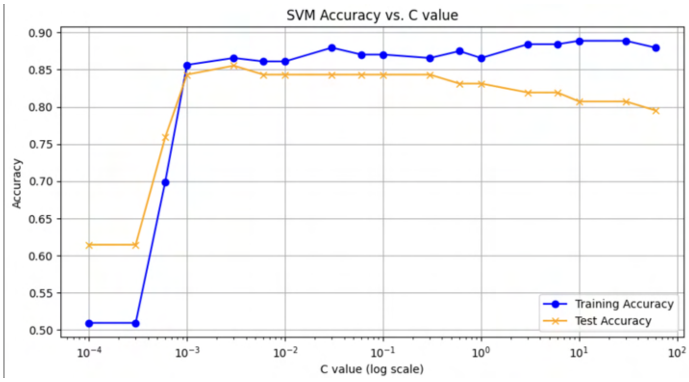

Figure 8: Optimization of C Parameter

The next series of tests involve the GoogLeNet classification accuracy. Fig. 9 shows the desired confusion matrix results for a sample of 400 fruit and canned food images obtained online. As seen by Fig. 9, the goal is to have correct fruit classification with an accuracy of 90%. To achieve this accuracy, we will have to gain a deep understanding of GoogLeNet to better alter the algorithm for it to work primarily for classifying these three fruits.

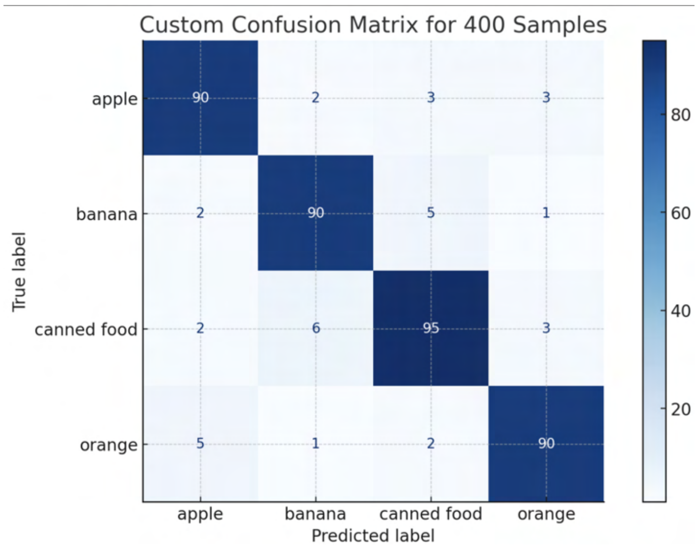

Figure 9: Ideal Confusion Matrix

Then, the second to last series of tests will involve label reading accuracy tests. OpenAI has a reputation of having around 97% accuracy regarding text-reading capabilities. As a result, the desired result is 95% accuracy when it comes down to reading the nutritional label of canned foods [8]. This test will consist of 100 images of various canned food images online. The accuracy is determined by correctly identifying what brand of canned food the image belongs to, the number of servings it consists of, and lastly, the amount of calories per serving. This will be conducted manually to better monitor the results and identify potential issues.

Lastly, the database will be consistently monitored to see if it is properly storing names and values from the various ML computations as described in the previous tests.

7.3 Website Test and Validation

The website will be tested in two separate phases: local and cloud. The local phase will be ran locally on a computer, and the primary concerns with it are usability and functionality. Usability tests will involve surveys in which we ask volunteers to interact with our website and report different components regarding their satisfaction levels. The two primary goals our design hopes to achieve are easy navigation and formatted display of information. The registration, login, inventory, and caloric intake pages will be extensively tested as a volunteer interacts with it for roughly around 5 to 10 minutes. Then, a populated database with dummy data will be displayed on the inventory page as well as the caloric intake page. Likewise, tests regarding security of the website will be implemented. Attacks like SQL injections and privacy attacks will be monitored and analyzed to ensure a secure web application is established.

After passing the local tests, the next step will involve cloud deployment. Latency will be tested to ensure a good user experience. AWS EC2 instance will be the designated platform used for cloud deployment. Since the ML algorithms will be computed on the cloud server, the *Route 53* console will be used to conduct latency analysis. Likewise, accuracy tests regarding database reads and writes will be conducted with expected accuracy of 100%. Finally, an exhaustive test will be conducted at the end which combines the hardware input with the ML algorithm, and all that will lastly be re-directed to the web application. This will be the final product validation stage.

8 PROJECT MANAGEMENT

8.1 Schedule

The detailed work schedule is shown in full in the back closing pages of this report in Fig. 11. The following aims to enumerate the key milestones to track our progress to MVP completion:

- Milestone 1 is the creation of the physical box which has already been completed
- Milestone 2 is forwarding scale input to the Arduino which then forwards it to the database
- Milestone 3 is completing the ML classification process with input from our system's camera
- Milestone 4 is the completion of our website design including cloud deployment
- Milestone 5 is reaching our MVP by combining all the components into a cohesive product

8.2 Team Member Responsibilities

The following work distributions amongst us are as follows:

Grace will be working primarily on the web applications portion. This consists of front-end design to create a user-friendly website. The website will be composed of various pages using the MVC template from Django to display values in our database. She will create a login and registration page that allows users to create an account and monitor their inventory and caloric intake. Furthermore, she will work with Surya to accept user-input in the form of user IDs and buttons on the website to be forwarded to the other hardware components when writing to the database. Lastly, she will help with the creation of the physical product and the integration between ML computation and the back-end database.

Surya will lead efforts in integrating physical components with the computational elements of the design. To be clear, although this is primarily a software project, it will rely on an elegant and effective interplay of sampling, transmission, and updating that must be done in a concurrent and timely manner. Towards this goal, he will work with Steven on microcontroller scripting and with both Steven and Grace on database management, which will be the most challenging aspect of this project. Additionally, he will work with Steven on the camera integration and the "autocapture" feature which will filter and reroute captured images of food items to directories for the classification algorithm to read from.

Steven will primarily focus on the ML computation and classification component of the product. This consists of gathering the data, training the model, and testing it on verification and validation data. Likewise, he will work with using online libraries and APIs to perform the remaining computations. Lastly, he will extensively work on connecting the results to the database in addition to receiving input from the hardware components. As a result, Steven will work closely with Surya in terms of communication protocols between the peripheral sensors, the Arduino, and the computer.

8.3 Bill of Materials and Budget

Please find on the following page, Table 1, which displays the necessary resources needed and their corresponding prices.

8.4 Risk Mitigation Plans

Risk Mitigation is an important consideration

- **Signal Amplification:** In microcontroller-based systems utilizing load cells (such as a scale, which translates mechanical pressure into an electrical signal), there exists the potential need to amplify load cell signals to address inadequate sensitivity in the Analog-to-Digital (ADC) conversion process. Towards this, we will incorporate dedicated load cell amplifiers and transducers such as the Sparkfun HX711, which provides the desired high-resolution ADC conversion and minimizes the chance of measurement inaccuracies.

- **Mismatched baud rates** are a significant (but thankfully easily-resolved) issue in the context of micro-controller-computer communication. In the context of our project, the "polling feature" will require logic on the baud rate which can be controlled in the 'Serial Monitor' of the micro-controller IDE. Through `Serial.begin()` functions, the synchronization can ensure that data is transmitted and received at the same rate, ensuring proper sampling.

- **Buffer overflow** and **data loss** pose challenges in micro-controller-based systems that have to interface with external devices. This can occur when incoming data from the scale exceeds the capacity of the micro-controller `receive` buffer, potentially causing data corruption or subsequent instability. To address this issue, our team plans to implement control mechanisms in the controller scripting, following XON/X-OFF software flow control protocols. We also plan to incorporate interrupt-driven processing for asynchronous data management and effective buffer forwarding.

  In the context of data loss, we plan to log incoming data to non-volatile storage options. Arduino holds the well-established EEPROM as an example of this, or SPIFFS in the ESP8266 WiFi microchip. These are hardware backups which the database can read from in the event that the system crashes.

The database is the most vulnerable component in this entire project. It is by far the most important, yet holds the most potential for bottleneck-related issues and crashing, which will frustrate potential users. Several steps must be taken in the context of efficient database access and retrieval, and protection against concurrency, injection, and CSRF attacks:

- **Transactions for concurrency:** In multi-threaded and concurrent environments such as employed in this project, where the database can be simultaneously updated by inventory and calorie values, preventing race conditions and data consistency are of paramount interest. Thus, we will employ Atomicity, Consistency, Isolation, and Durability (ACID) design properties when implementing transactions. In particular, we look to implement a rolling 2-phase locking approach through Django's `Transactions` package to help maintain data integrity and reliable operation of microcontroller-based systems interacting with our MySQL databases.

Table 1: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
Aideepen ESP32-CAM W BT Board	ESP32-CAM-MB	Aideepen	1	$15.99	$15.99
Xiaomi Smart Scale	Xiaomi Scale	Xiaomi	1	$29.99	$29.99
Load Cell Amplifier	HX711	Sparkfun	1	$5.99	$5.99
Adafruit Jumper Cables	Male-Female	TechSpark Inventory	1 pk	FREE	FREE
Raspberry Pi 4 Model B	Raspberry Pi 4	Raspberry Pi Foundation	1	$35.00	$35.00
OXO Good Grips Stainless Steel Food Scale	OXO Scale	OXO	1	$49.99	$49.99
RS232 to TTL Converter	MAX3232	Texas Instruments	1	$9.99	$9.99
					$144.95

- **Data Sanitization**: SQL injection attacks and Cross-Site Scripting (XSS) can occur due to a failure to properly sanitize and validate input data when constructing SQL queries. To mitigate this risk, we will employ data sanitization on request inputs to separate data from commands, ensuring security and integrity.

- **Database Manipulation**: In relational databases like MySQL, the storage and retrieval of data are crucial for efficient operations. One common method for organizing data is through tree-like structures, such as B-trees or B+ trees. These structures enable logarithmic access and retrieval times; namely, the balanced hierarchy of nodes allows for efficient searching, insertion, and deletion operations by maintaining a balanced hierarchy of nodes. However, there are scenarios where performance can degrade. For example, query calls using the reverse accessor, which involves traversing the tree in the opposite direction from its typical orientation, can have a worst-case time complexity of $O(n)$, where n is the number of elements in the tree. This degradation in performance can impact the overall efficiency of data retrieval operations.

- **Classification Concerns**: A final consideration pertains to to classification accuracy. If either the SVM or GoogLeNet algorithm does not reach the expected accuracy results in the desired amount of time, we have prepared to mitigate this issue by fully integrating the ChatGPT4 API. This has been tested extensively and has showcased successful accuracy results and computational efficiency. The absolute worst case involves consolidating the ML design pipeline into the ChatGPT4 API. Likewise, Steven has experience using the Imagga ML library which also exhibits high performance in classification and label reading. The hope is still to showcase our knowledge and skill by making our own classification design, fine-tuning a pre-existing design, and finally integrating an API into our project; however, we have backup plans to account for the uncertainty.

9 RELATED WORK

There are two Capstone projects that are closely related to our idea. The first being Food Tracker (team B6 Spring 2022) [11]. This group implemented a food tracking system to keep track of inventory for grocery items using very similar features as us such as camera recognition and database display. Likewise, A Smart Kitchen Assistant (team D3 Spring 2021) also completed a project involving inventory tracking [4]. These projects have provided us with good inspiration and advice as we seek to develop our product.

In addition to these projects, Samsung has created smart refrigerators that do inventory tracking. These have costs of a couple thousand dollars due to their extensive AI technology, significantly beyond the planned design costs for our design. Nonetheless, the blueprint of Samsung's modern refrigerator technology informs and inspires much of the features we aim to highlight in our MVP.

Overall, our product has similarities with both the Capstone projects and Samsung smart appliances[7]. However, there is novelty in our calorie-tracking feature which we hope to highlight in our design. Likewise, we aim to leverage versatility in product operation; our vision for this product is integration into smart appliances in the kitchen setting, with interfacing capabilities from a mini-fridge to a pantry.

10 SUMMARY

NutrientMatch is a novel project that involves efficient inventory tracking and accumulation of caloric consumption that can easily be integrated into food storage systems and displayed on laptops and smartphones. We seek to incorporate camera and scale readings with ML classification and recognition algorithms to store data that can easily be accessed via a website.

As we move forward with our design, we expect to address several potential complications. The first one is effective scale integration with the Arduino and website with low latency. To address this, we have consolidated a series of specific design choices to consider when building the physical apparatus. The second risk is potentially lower expected accuracy results from our ML models; this can be mitigated through alternative design choices involving the ChatGPT4 API and the Tesseract OCR to boost accuracy.

Lastly, we anticipate potential concurrency, synchronization, and security complications regarding the website. To address these, we plan to do intensive testing and conduct a greater in-depth research into mitigating these concerns.

Glossary of Acronyms

Please find below frequently-referenced acronyms usd throughout this design report:

- API - Application Programming Interface
- ADC - Analog to Digital Converter
- CNN - Convolutional Neural Network
- FPS - Frames per Second
- GPIO - General Purpose Input/Output
- IC - Image Classification
- ML - Machine Learning
- MP - MegaPixels
- MVP - Minimum Viable Product
- MVC - Model-View-Controller Architecture
- OCR - Optical Character Recognition
- SVM - Support Vector Machines
- TTL - Transistor-Transistor Logic
- XSS - Cross-Site Scripting

References

[1] URL: https://www.postgresql.org/docs/current/datatype-json.html.

[2] URL: https://www.bbcgoodfood.com/review/best-kitchen-scales.

[3] Dishaa Agarwal. URL: https://www.analyticsvidhya.com/blog/2021/04/insight-into-svm-support-vector-machine-along-with-code/#:~:text=Soft%20margin%20SVM%20maximizes%20the,maximizes%20the%20margin%20between%20classes..

[4] A Smart Kitchen Assistant. URL: http://course.ece.cmu.edu/~ece500/projects/s21-teamd3/.

[5] Jonathan Deber. "How Much Faster is Fast Enough? User Perception of Latency & Latency Improvements in Direct and Indirect Touch". In: *Conference on Human Factors in Computing Systems* (2015).

[6] Alexander Gillis. URL: https://www.techtarget.com/searchdatamanagement/definition/MongoDB.

[7] Team HomeServe. URL: https://www.homeserve.com/en-us/blog/buying-guide/samsung-smart-refrigerator/.

[8] Jeremy Hsu. "ChatGPT outperforms humans at labelling some data for other AIs". In: *The Scientist* (2023).

[9] OpenFoodFacts. URL: https://world.openfoodfacts.org.

[10] Tamana Sharma. URL: https://circuitdigest.com/article/rs232-serial-communication-protocol-basics-specifications.

[11] Food Tracker. URL: http://course.ece.cmu.edu/~ece500/projects/s22-teamb6/.

[12] Yue Yuan. "Vegetable and fruit freshness detection based on deep features and principal component analysis". In: *Current Research in Food Science* (2024).

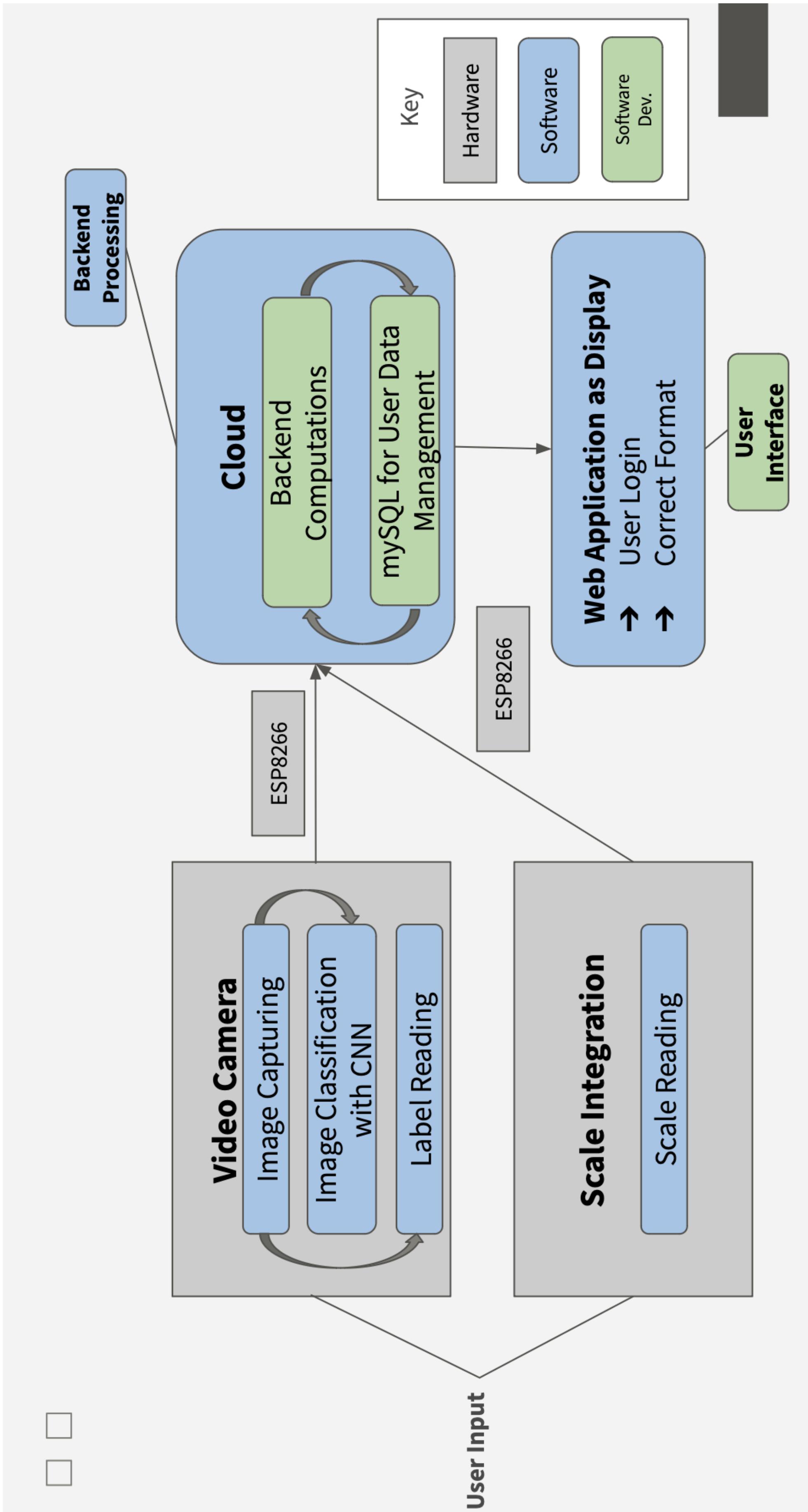

Figure 10: Full-page version of the system block diagram as described earlier.

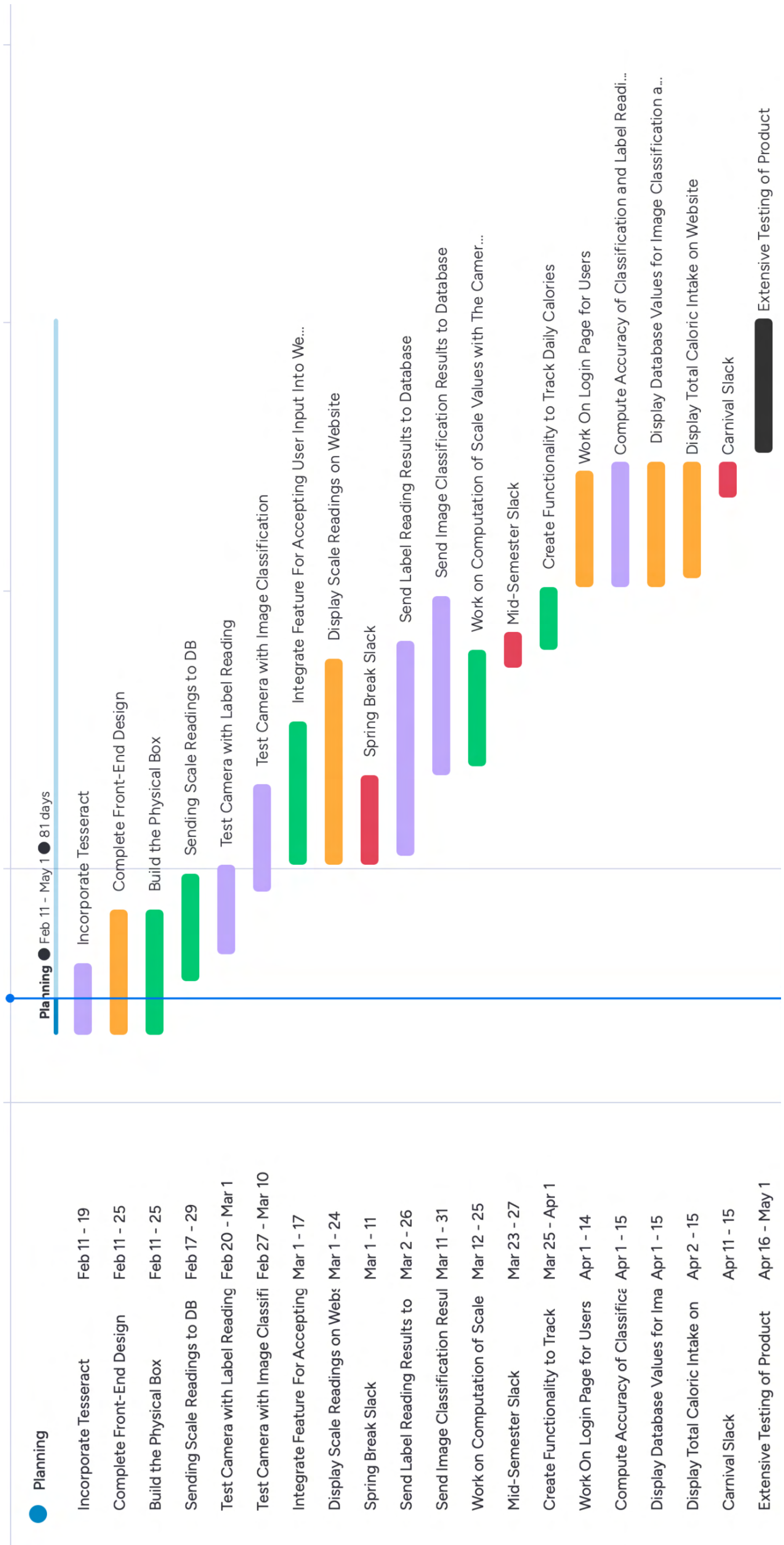

Figure 11: Gantt Chart