TEAM E2

NutrientMatch

Surya Chandramouleeswaran, Grace Liu, Steven Zeng

USE CASE

PROBLEM

- An increase in physical wellness trends leads to more people tracking their daily macronutrient intake
- Food tracking becomes a tedious task that often results in lack of consistency
- Inaccurate nutritional data on food tracking apps

SOLUTION

NutrientMatch is a product that can be integrated into smart food storage appliances (i.e. fridge, pantries, etc.) to help users simplify their food tracking process

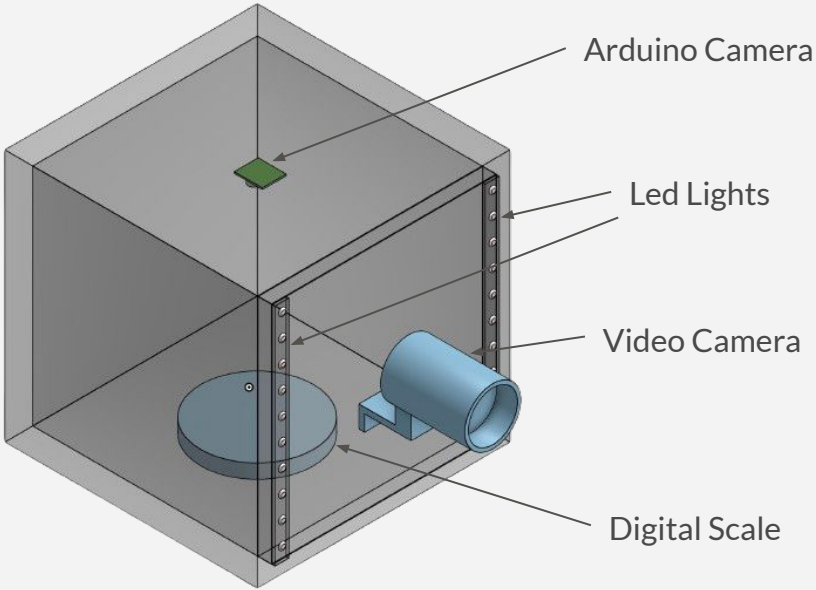
ECE AREAS

- **Circuits:** Integrating communication protocols between hardware and software components
- **Software Systems:** User interface and storing scanned information into cloud database

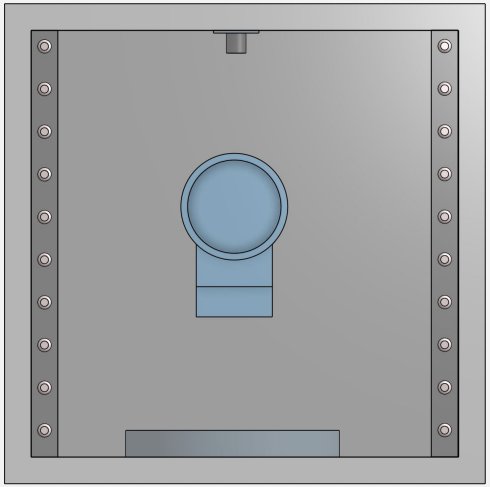
USE-CASE REQUIREMENTS

- Nutrition is of paramount importance to those who want to lead healthy and sustainable lifestyles, and statistics have motivated the notion of our product
    - **41.9% adults** in the U.S. are obese (via CDC)
    - Americans on average throw out **4 spoiled food items** per week (via OnePoll)
- Requirements to achieve success and efficiency of product
    - Wooden cube of around **2-3 feet on each side**
    - Video camera with frame rate of **30 frames per second (fps)** to register item
    - Arduino camera with **2 megapixels** to forward scale reading directly to database
    - Convolutional neural network (CNN) of **input size ~500-100 pixels** to match image size in pixels

SOLUTION APPROACH: Overview



Isometric View



Front View

Libraries: Tesseract Optical Character Recognition (TCR) and food classification library

SOLUTION APPROACH

	Image Recognition	Storing to DB	Website Display
STEP 1	Connect video and Arduino cameras to Arduino board or computer for OCR algorithms	Populate database with specific users (weight information, nutritional goals, etc.)	Create user logins and display necessary information
STEP 2	Testing for classification accuracy	Check food item calculation accuracy	Backend is able to accumulate user's nutritional data in 24-hour periods
STEP 3	Ensure scale reading can be stored somewhere	Computations are stored under respective users	Display correct information under respective user
STEP 4		Items can be removed from DB when prompted by user and reset after 24-hour period	Feature allowing user to remove items from inventory after consumption

TECHNICAL CHALLENGES

COMMUNICATION

- Between different stages of our project by first identifying smaller building blocks
- Cameras send data synchronously to DB

COMPUTATIONAL PROCESSING

- Arduino is only in charge of scale reading, but does it need to be offloaded to computer for higher efficiency?
- Implement real-time synching with cloud

ROBUST ERROR HANDLING

- Two cases to consider
  - Algorithm is unable to classify item
  - Algorithm incorrectly classifies item
- Hardware should easily be able to switch between image recognition models

INTEGRATION OF PARTS

- Sending data synchronously to DB from both scannings
- Processing backend and uploading to user interface in a timely manner

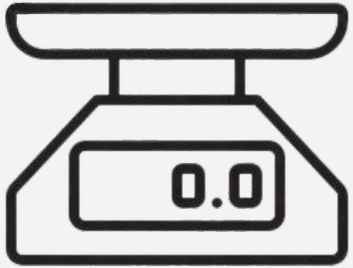
Testing, Verification, and Metrics

Testing Requirement	Verification	Corresponding Metrics
Camera performance and Optical Character Recognition (OCR) algorithm accuracy testing	Ability of camera to forward clear image to image classification system (consideration of downsampling)	To meet a certain computational processing speed/efficiency
Image classification testing	Main focus on classification accuracy	Dataset consists of ~70% training data and ~30% testing data and confusion matrices
Database input and retrieval testing	Ensure data is safely written to database and real time performance tests under certain workload	Web application development using HTML, views.py, and JavaScript; AJAX to meet response time

TASKS

First Stage: Carrying out inventory pipeline

Second Stage: Scale integration to DB



DIVISION OF LABOR

SURYA

- Hardware integration and communication to database
- Either implement wirelessly or connecting components to computer

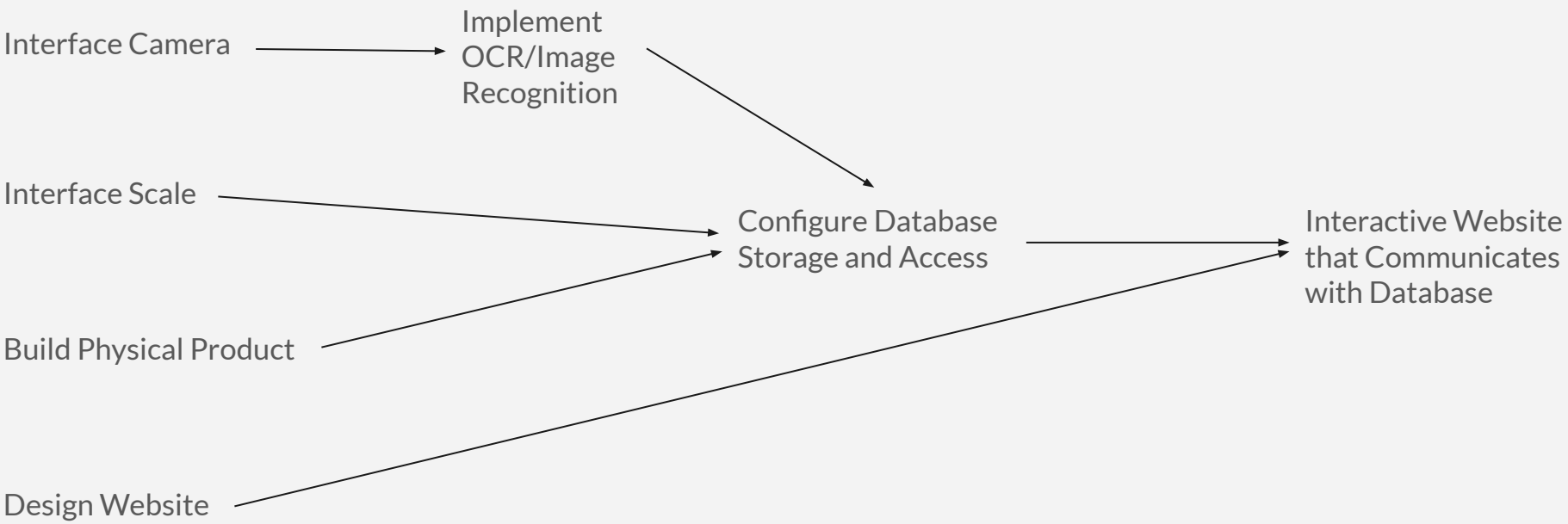
STEVEN

- Testing OCR algorithms for classification accuracy
- Measure scale readings to be forwarded to backend computational software

GRACE

- Populate database with detected information
- Alignment of backend computations with frontend display for effective tracking

OVERALL FLOWCHART



SCHEDULE

Tasks	Week 1 (01/29-02/4)	Week 2 (02/5-02/11)	Week 3 (02/12-02/18)	Week 4 (02/19-02/25)	Week 5 (02/26-03/3)	Week 6 (03/4-03/10)	Week 7 (03/11-03/17)	Week 8 (03/18-03/24)	Week 9 (03/25-03/31)	Week 10 (04/1-04/7)	Week 11 (04/8-04/14)	Week 12 (04/15-04/21)	Week 13 (04/22-04/28)	Week 14 (04/29-05/5)
Create Inventory/Budget Allocation														
Purchase Items														
Build Interface b/n Camera and Computer														
Build Interface b/n Scale and Computer														
Design Website and Front-end Display														
Build the Box/Physical Product														
Incorporate OCR and Image Recognition														
Configure the Database														
Test Camera and OCR Algorithm														
Train/Test Image Recognition Algorithm														
Test Database Access and Web Interface														
Final Presentation Design														
Slack														

CONCLUSION

- **Goal:** develop cloud-based nutrition technology in combination with a growing trend of smart appliances in the kitchen
- Project components
  - Image recognition system to help manage nutrient intake (5-6 weeks)
  - Weighing apparatus to track food volume
    - Integrated asynchronously from image classification system