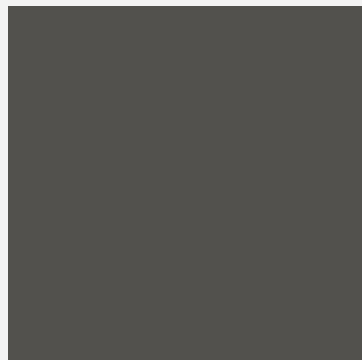
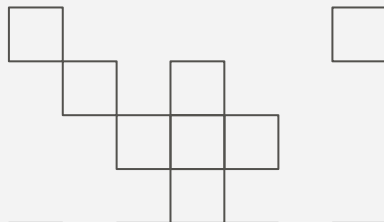


TEAM E2

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
- Lack of easy inventory tracking in pantries/fridges



SOLUTION

NutrientMatch is a product that helps users simplify their food tracking and inventory management process



FINAL CHANGES

- Used Raspberry Pi and two USB cameras to record image and scale readings
- Shifted towards usage of ResNet-18 model with background subtraction
- Implemented 7-segment OCR in lieu of wired serial communication on the scale



Quantitative Design Requirements

Overview

Structural Parameters

Description of Physical Features of Product

- Wooden cubic structure of edge length **~2-3 feet**
- Ambient **55-80 watt** LED for diffuse illumination
- Scale, Microcontroller

Performance

Object recognition and performance

Captured image of product should be reasonably clear and well-illuminated

- **1080p, 15 FPS** Camera
- Food held **12-18 inches** from camera lens
- **~750 pixel images** to feed into CNN

Accessibility

User Accessibility and Experience

Not a quickly-patched DIY; product should reflect elegance and functionality

- Overall Latency minimization (**~25 seconds**)
- Systematic Bias Mitigation (**>= 90%** recognition accuracy)

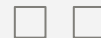
Evolution of Solution Approach

Themes	Object Recognition(CNN)	Back-end Management	Hardware Considerations
STEP 1	Image Classification: -> Packaged Items -> Standalone Foods	mysql manages users within family, corresponding data	Diffuse Lighting Structure
STEP 2	Label Reading: -> Directed towards packaged foods	Low-latency (25-seconds) updating process	Scale Communication: Raspberry Pi (RPI)
STEP 3	Robust Error Handling	“Polling strategy” on weight measurement	Automated scripts for microchip control

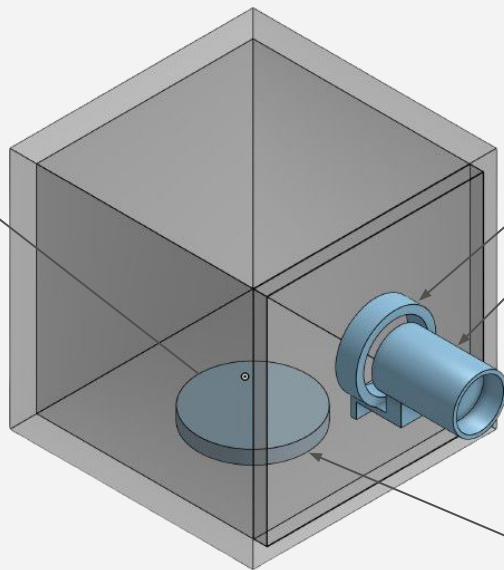
Societal/Ethical Considerations

- Creating a social platform to foster an encouraging and supportive online community (reach goal)
- Be cautious in the design process to avoid the development of eating disorders and body dysmorphia
- Food wastage concerns

Solution Approach: Overview



RPI 4 Model B



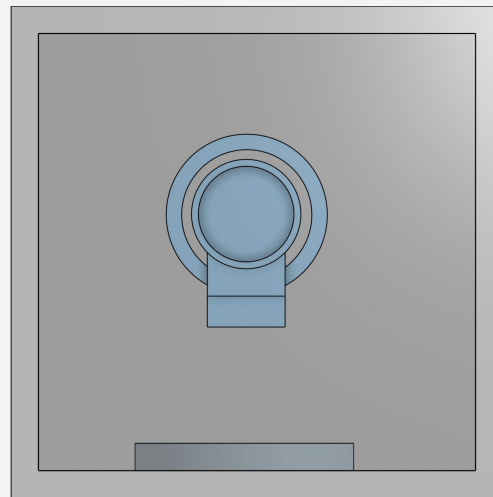
Isometric View

Ambient Ring Lighting

Video Camera



Digital Scale



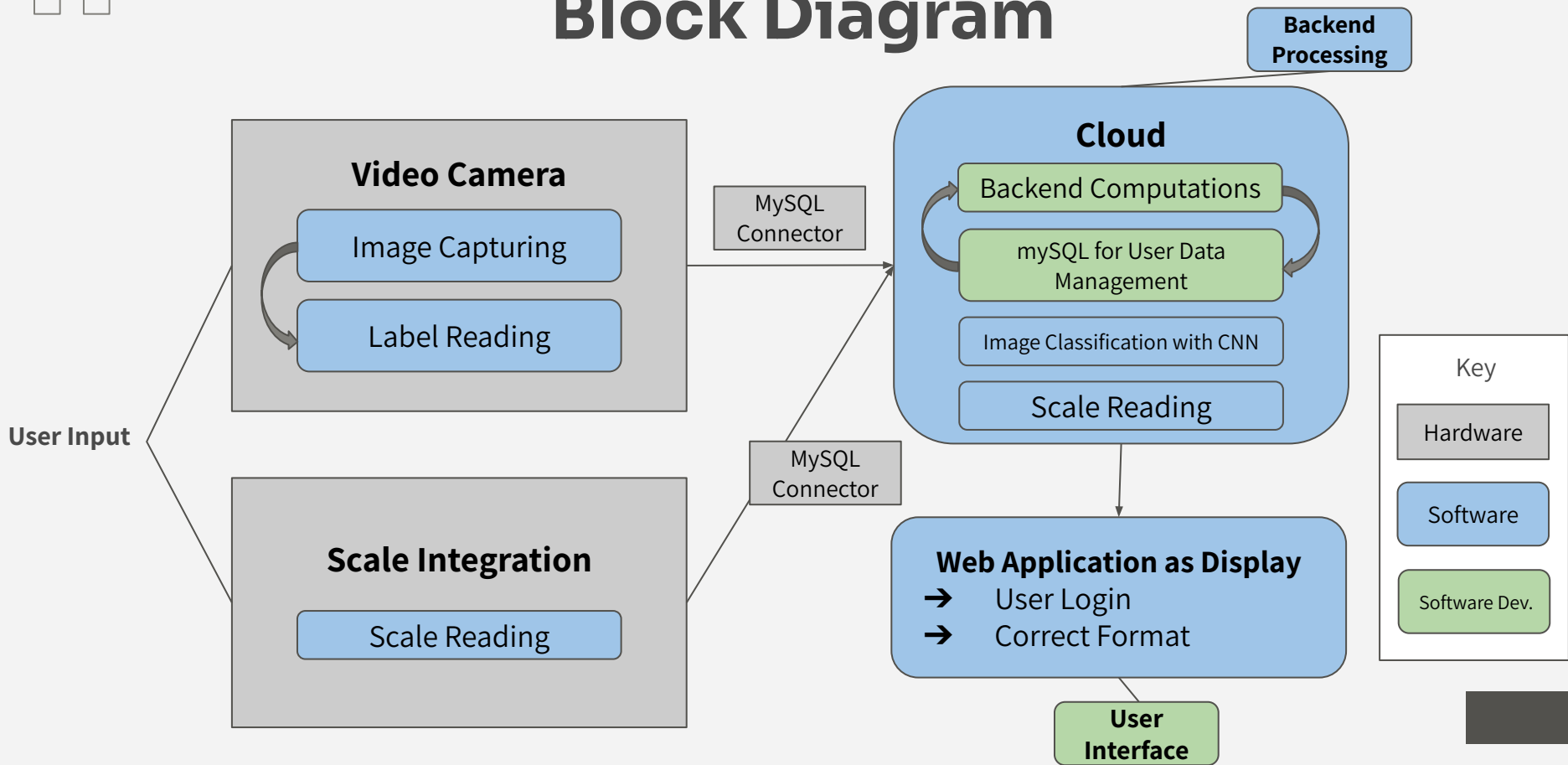
Front View

Users can either choose to

Scan nutritional label on packaged foods

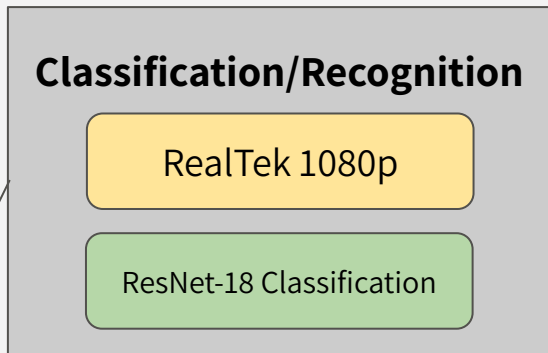
Image classify & weigh fresh produce on the scale

Block Diagram

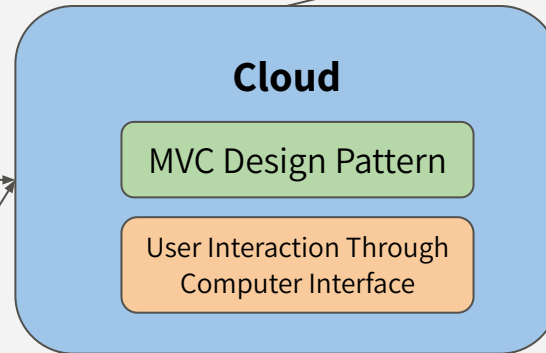


Implementation Plan

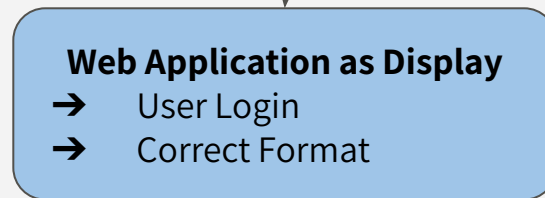
Backend Processing



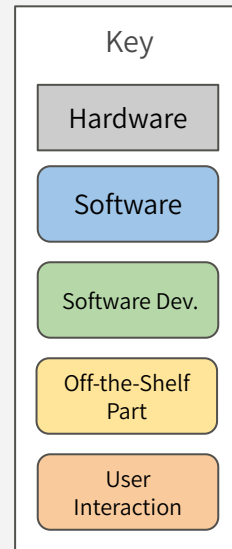
RPI



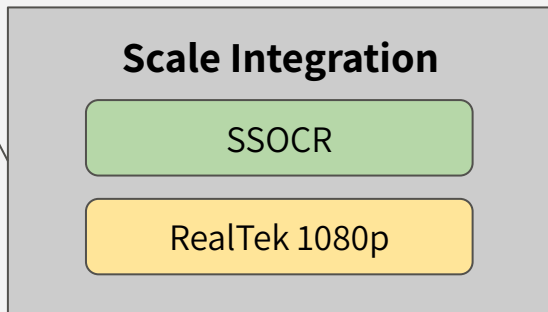
RPI



User Interface

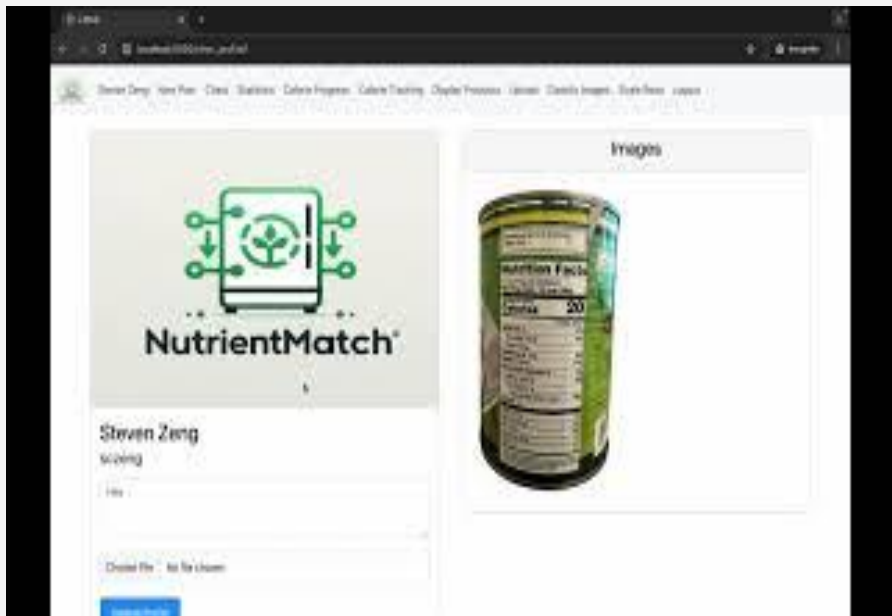


User Input



Complete Solution Plan

Web App Demo



RPI/Camera Demo



Testing, Verification, Validation

01

Image Recognition System

- Canned Foods
- Fruits
- Invalid Groups

02

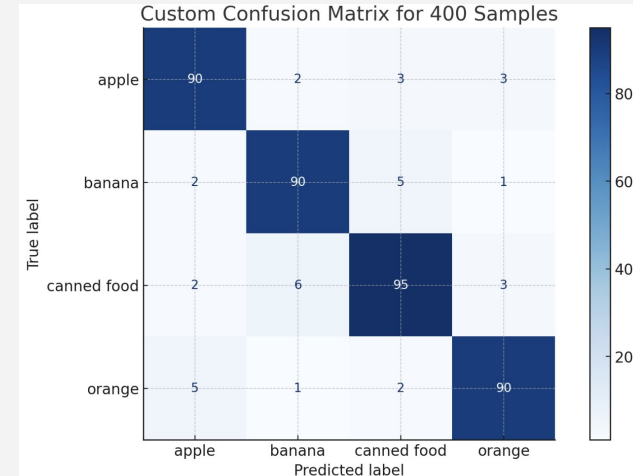
Overall Performance Considerations

- Weight Detection
- Caloric Accuracy
- Overall Process Latency

03

Risk Mitigation: Invalid Measurement

- Ability to edit incorrect entries/errors
- Ability to interrupt/restart measuring process in flexible manner

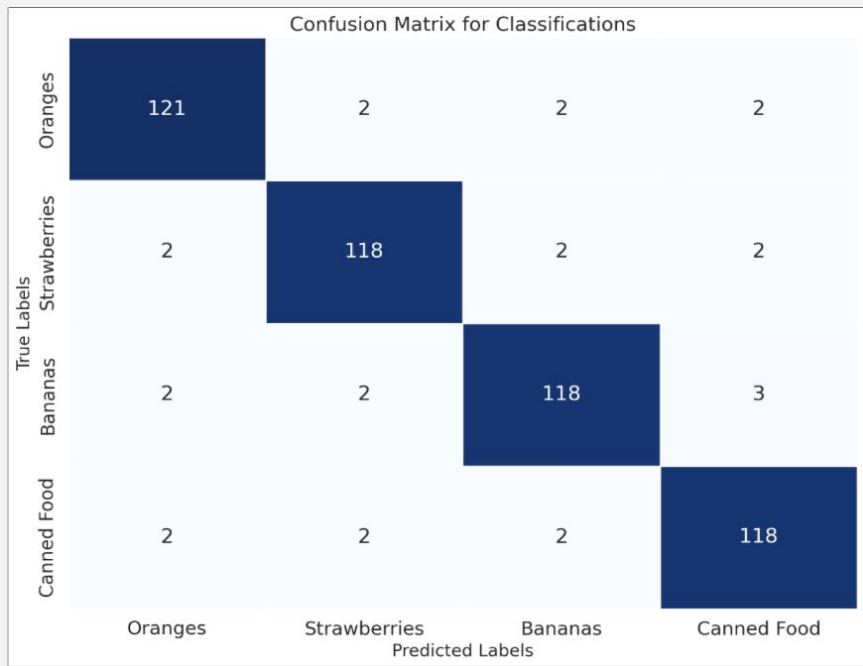


- Benchmark: 95% category classification
- After classification, caloric values should be within 10% range from actual amount

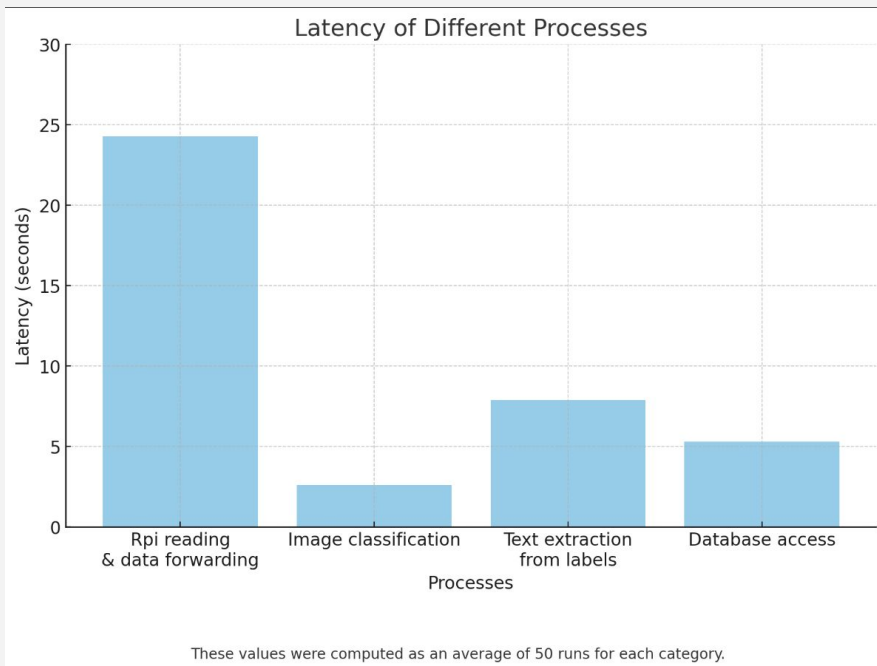
Goal: Modular structure around imaging, scale integration, and backend management

Testing Results

Classification Results



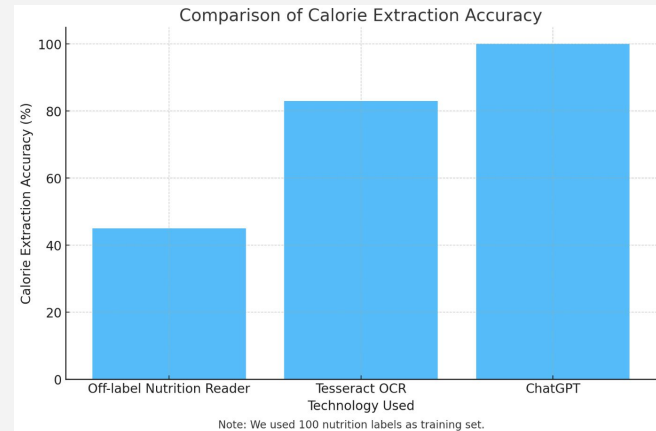
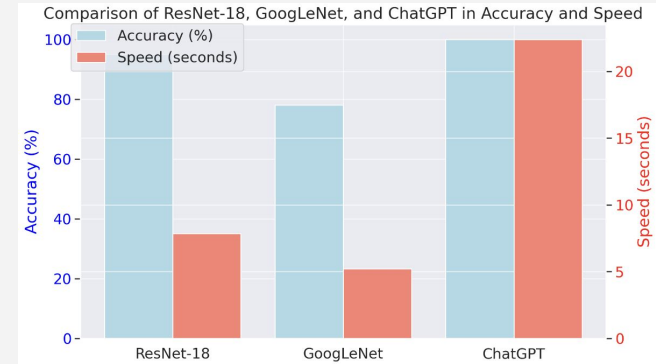
System Latency Results



Design Tradeoffs

ML Models	GoogLeNet	Tesseract OCR*	Nutrition Reader	ResNet-18*
Benefits	-> Pre-trained IC Library -> Quick to Run	-> Label-reading -> Pre-trained Library with High Accuracy	-> Scanning Capabilities -> Label-reading	-> Pre-trained IC Library -> Highest Accuracy
Problems	-> Relatively Low Accuracy -> Not a Representative Dataset	-> Poorly Formatted Text Output	-> Low Calorie Extraction Accuracy (~50%)	-> A lot of Computation Power -> High Latency

System Design Choices	Where to perform ML computations?	What information to store in database?	Hardware design choices





Project Management

Key	Category	Task Name	Total Weeks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
				02/12 - 02/18	02/19 - 02/25	02/26 - 03/03	03/04 - 03/10	03/11 - 03/17	03/18 - 03/24	03/25 - 03/31	04/01 - 04/07	04/08 - 04/14	04/15 - 04/21	04/22 - 04/28	04/29 - 05/04
Everyone	Project Planning	Define Project Scope	1	█											
Grace		Design Presentation Preparation	1	█											
Steven		Design Report	1.5	█	█	█									
Surya		Final Presentation Preparation	1												
		Final Report/Submissions	3										█	█	█
	Slack Time	Spring Break	1				█								
		Mid-Semester	0.5							█					
		Carnival	0.5									█			
	Frontend Implementation	Mockup Designs	1.5	█	█	█									
		Basic Frontend Design	2	█	█										
		Web Request Rendering	2			█	█								
		User Login Functionality	1				█								
		Weight Tracking Functionality	1				█								
		Database Retrieval System	4				█	█					█	█	
		User Chat Functionality	3.5				█	█	█						
		Additional Functionality Inclusion	8					█	█	█	█	█	█	█	█
		Frontend Improvements	2.5					█	█	█					
	ML Algorithm Training	Tesseract Incorporation	3	█	█					█	█	█			
		Design Path Considerations	2	█	█										
		Image Classification Training	4			█	█	█	█						
		OCR Training	4		█	█	█	█		█					
		Food Product Differentiation	3				█	█	█						
		Accuracy Training	1					█	█						
		API Adjustments	2							█	█				
		Physical Camera Integration	2.5							█	█	█			
		Image Classification Adjustments	3									█	█	█	
		ResNet18 Incorporation	3.5						█	█	█	█	█	█	
		Accuracy Improvements	3									█	█	█	█
	Scale Implementation	Design Path Considerations	3	█	█	█									
		Hardware Scoping & Ordering	3	█	█	█									
		Scale Testing	2			█	█								
		Integration with Remote Database	4				█	█	█	█					
		Scale Modifications	2						█	█					
		Scale Sensitivity Testing	1								█				
		7-Segment Display OCR	2.5								█	█	█	█	█
	Camera Implementation	Camera Selection & Testing	4	█	█	█									
		ArduCam Testing	4	█	█	█									
		Raspberry Pi Configuration/Setup	3		█	█	█								
		Camera Performance Improvements	2							█	█				
		Qualitative Verification	3									█	█	█	█
	System Integration	Label Reading Forwarding	1.5								█	█	█		█
		Food Classification Forwarding	2								█	█			
		Caloric Scale Calculations	3								█	█	█		
		Global Page Usage	2						█	█					
		Latency Testing	3.5									█	█	█	█
		User Testing	1									█	█	█	█
		Final Checks	0.5												█

