Team D3 – Meal By Words

Nina Duan, Lisa Xiong, Shiyi Zhang

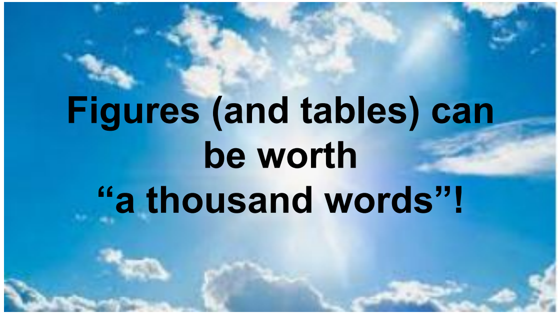
Add your 12 slides after this slide… [remember, 12 min talk + 3 min Q/A]

For more information about formatting or importing slides see:
https://gsuite.google.com/learning-center/products/slides/get-started/

Make sure to cover
(refer to the Design Review Guidance):

- Use Case / Application
- Use-Case Requirements, especially quantitative
- Solution Approach (include Design Requirements here)
- System Specification / Block Diagram
- Implementation Plan (include Design Trade Study(ies) here; i.e why choose that implementation)
- Test, Verification and Validation Plans (including quantitative metrics with target values)
- Project Management

Consider that this slide already works as a introduction slide so use your first slide wisely



Use Case

- Speech-operated ordering kiosk
- Areas: hardware, signal processing, software
- Allow customer to verbally:
  - Order items from menu
  - Remove/change existing item(s)
  - Proceed to checkout
- Allow staff to:
  - See existing orders
  - Delete completed orders

This amazing exclusive offer at the fantastic value-for-money
14 $
TOUCH TO START

Design Requirements

- Avg. service time should be <= 200 seconds

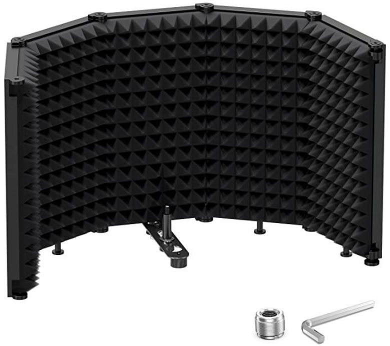
Human Detection	Speech Recognition	User Interfaces
Detect 100% of approaching customer(s)	Reach 85% speech-to-text accuracy at noise level < 70dBA Able to recognize an item by both name (e.g., “burger”) and menu ID (e.g., “A1”) Reach 100% item accuracy at checkout	Customer sees ● Ordered items ● Total cost ● Informative instructions Staff sees a list of orders, sorted by time

Solution Approach - Overview

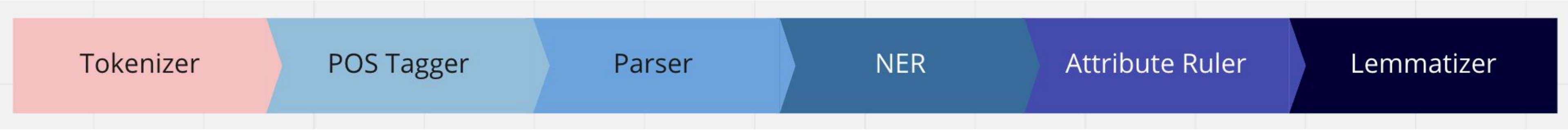
1. **Infrared sensor** detects approaching customer and wakes up system
2. Customer **UI** displays menu and order instructions
3. **Backend system** parses customer speech and fill in order
4. After checkout, order uploaded to **cloud** (accessible to kitchen staff)

- **Public health:** physical contact ⬇️ → risk of transmitting diseases ⬇️
- **Welfare:** workload & stress level of restaurant workers ⬇️
- **Economic:** labor costs ⬇️ throughput of customers ⬆️ → revenue ⬆️
- **Social:** customer satisfaction ⬆️ → time to enjoy their meals ⬆️

Solution Approach - Input Processing



- **Hardware**
  - USB directional microphone: Neat Bumblebee II
  - Professional sound shield: Moukey Microphone Isolation Shield
- **Speech Recognition**: Python SpeechRecognition library (voice input -> text)
- **Speech Parsing**: spaCy (text -> menu items & quantities)



Tokenizer → POS Tagger → Parser → NER → Attribute Ruler → Lemmatizer

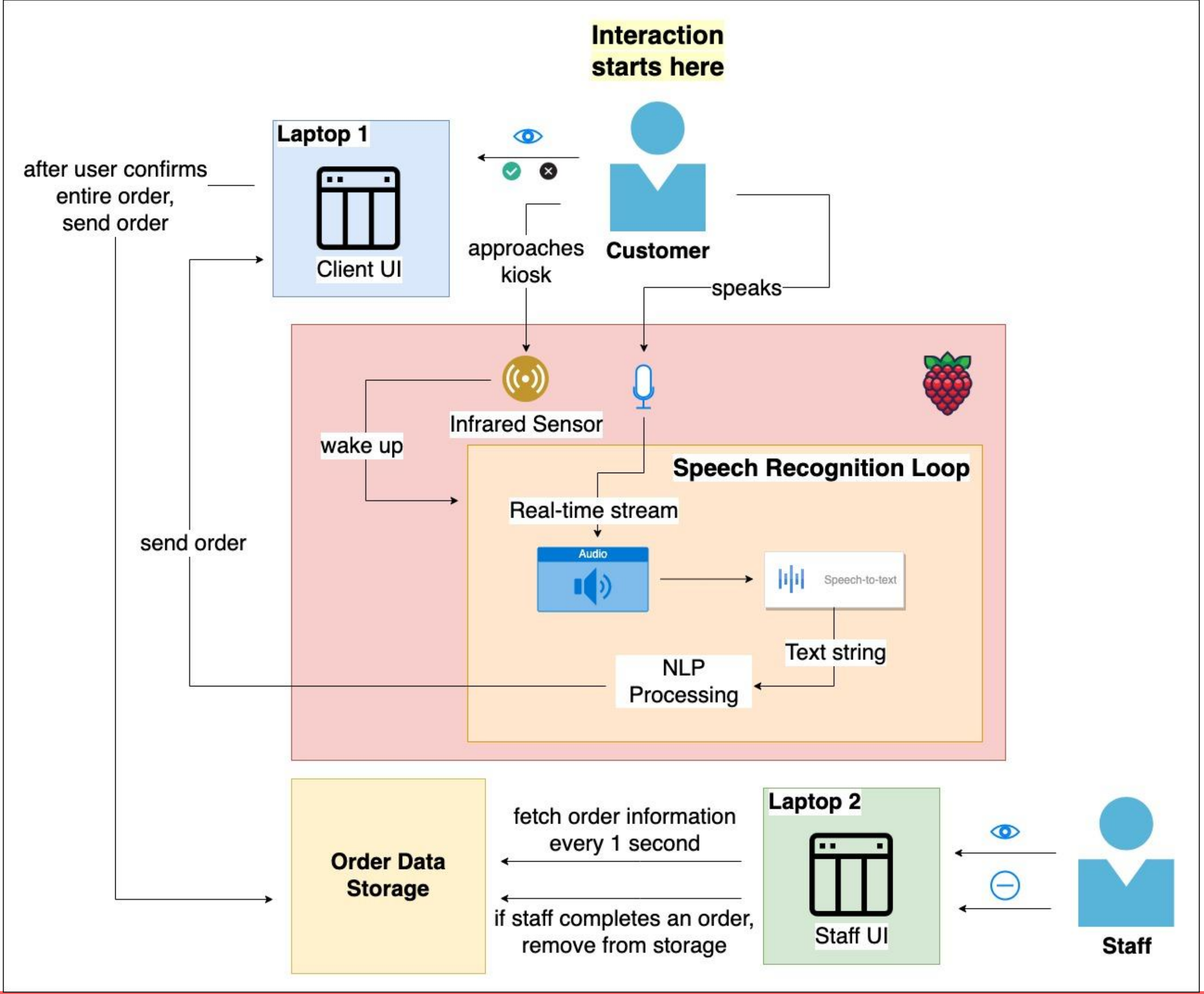
- Identify edge cases and modify the pipeline accordingly

Solution Approach - Termination

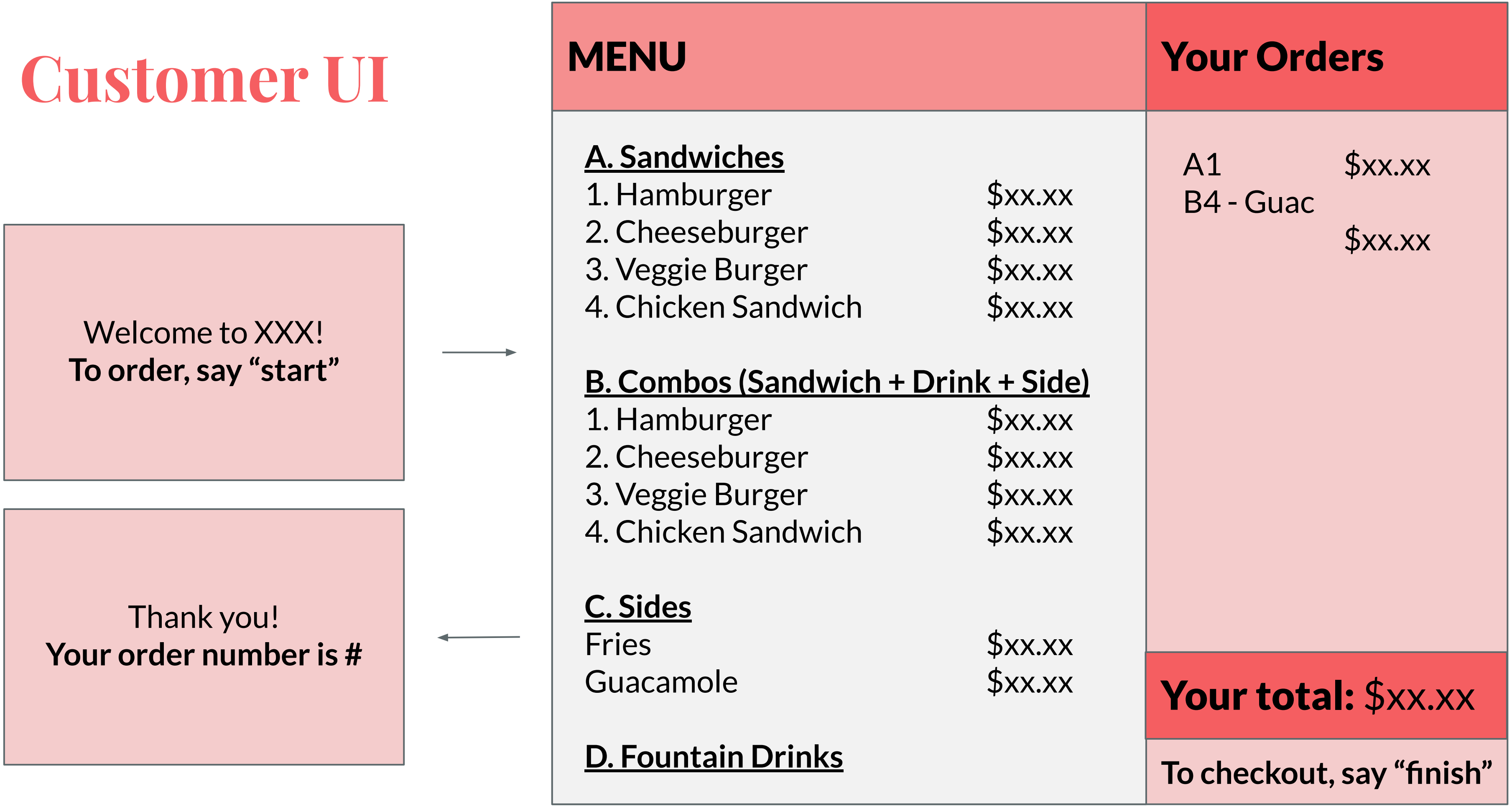
- **Timeout**
    - 1-minute mark: display warning – “System will timeout in <countdown>”
    - 2-minute mark: remove current (incomplete) order
    - Any recognized speech can interrupt countdown and resume process
- **Checkout**
    - Look for commonly used termination cues
        - E.g. “That’s it”, “I’m done”, “All set”, etc.
    - Ask customer to confirm order before uploading information to cloud

Solution Approach - Storage

- **noSQL key-value cloud database**
  - AWS DynamoDB
  - No need for relational database: storing simple, independent objects only
  - Key-value database supports faster inserts and deletes
  - Data persistent even if Raspberry Pi or UIs fail
- **Risk mitigation**
  - Send raw order information to staff end (represented by a laptop) and store as local data


Interaction starts here
Laptop 1
Client UI
Customer
after user confirms entire order, send order
approaches kiosk
speaks
Infrared Sensor
wake up
Speech Recognition Loop
Real-time stream
Audio
Speech-to-text
Text string
NLP Processing
send order
Order Data Storage
fetch order information every 1 second
if staff completes an order, remove from storage
Laptop 2
Staff UI
Staff

Customer UI



Welcome to XXX!
To order, say “start”

Thank you!
Your order number is #

MENU

A. Sandwiches

1. Hamburger	$xx.xx
2. Cheeseburger	$xx.xx
3. Veggie Burger	$xx.xx
4. Chicken Sandwich	$xx.xx

B. Combos (Sandwich + Drink + Side)

1. Hamburger	$xx.xx
2. Cheeseburger	$xx.xx
3. Veggie Burger	$xx.xx
4. Chicken Sandwich	$xx.xx

C. Sides

Fries	$xx.xx
Guacamole	$xx.xx

D. Fountain Drinks

Your Orders

A1	$xx.xx
B4 - Guac	
	$xx.xx

Your total: $xx.xx

To checkout, say “finish”

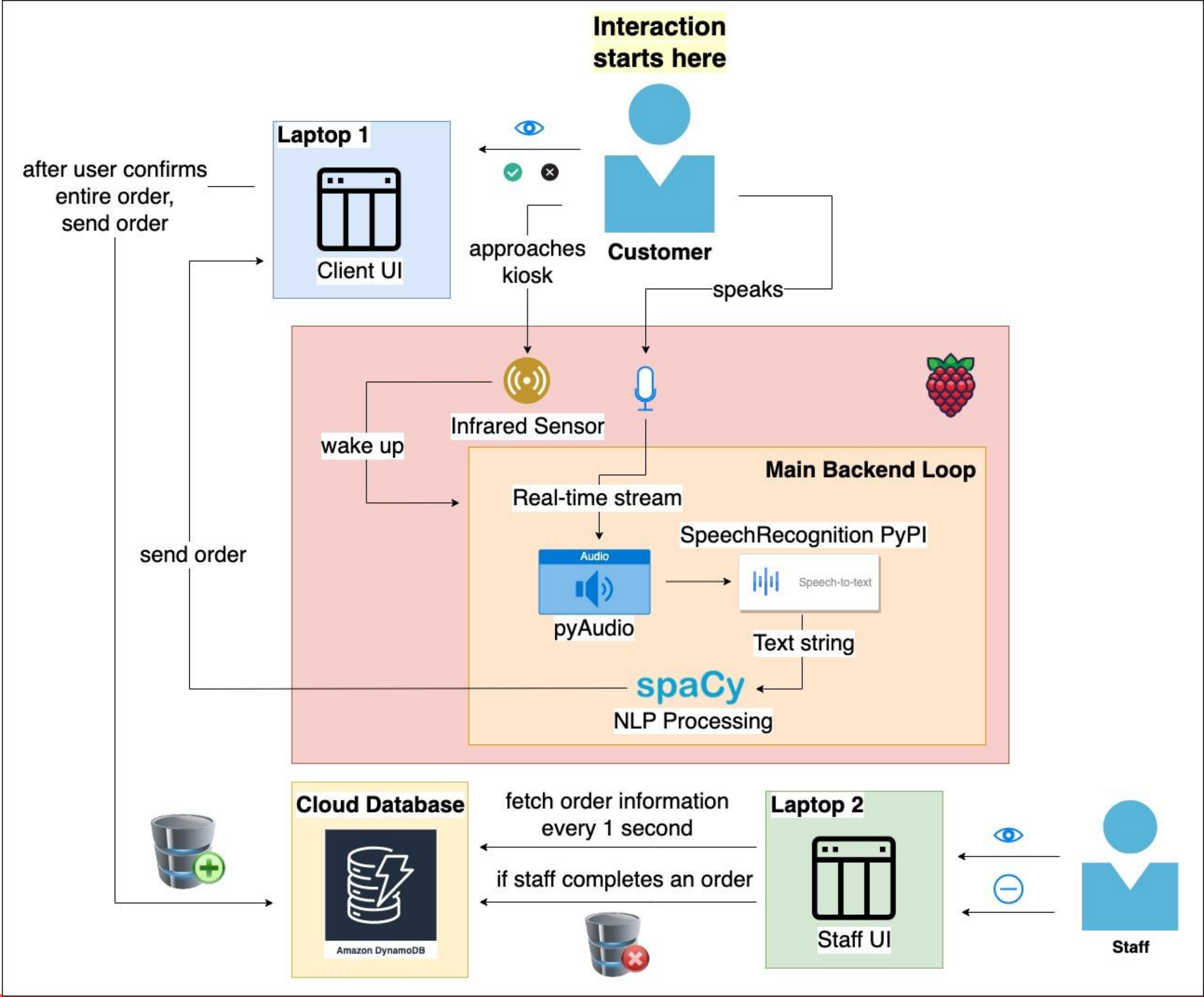
Implementation Plan

Libraries: pyAudio, SpeechRecognition PyPI, spaCy

Hardware components: Raspberry Pi, microphone, infrared sensor

Cloud Database

Custom UIs



Unit Testing

**** 10 volunteers/test, conversational volume (~60dB), distance 0.3-1m****

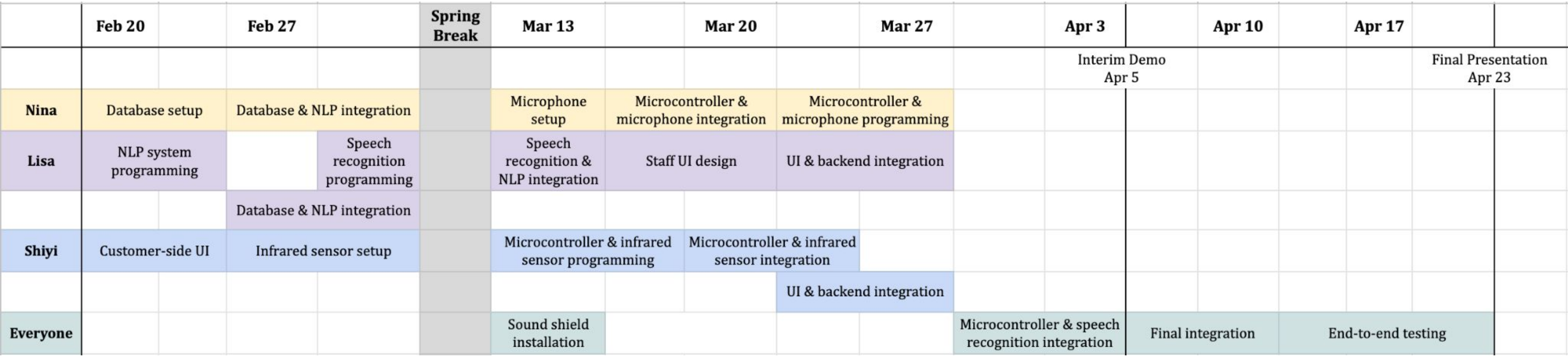
- **Audio to text**
  - # of words in audio correctly transcribed to texts / # of words spoken
- **Text to command**
  - “Two tacos, one milkshake please” → [add] [taco] [2], [add] [milkshake] [1]
  - “Delete tacos” → [del] [taco][*]
- **Noise tolerance**
  - Quiet (<40dBA), normal (50~60dBA), noisy (>70dBA) environments
- **Response time**
  - Staff UI, customer UI ⇄ cloud/database (1s)
  - Presence of a customer → kiosk activated (2s)

End-to-End Testing

**** 10 volunteers/test, conversational volume (~60dB), distance 0.3-1m****

- **Time** to complete a 1-person order (<= 200 seconds)
- **Accuracy** = # of correct item entries staff sees / total # of entries the customer says
- Expect **100% order accuracy** at the checkout in various environments
    - Risk Mitigations
        - Before checkout, system double checks order with customer
        - Customer can verbally remove/change order anytime

Project Management



	Feb 20		Feb 27		Spring Break	Mar 13		Mar 20		Mar 27		Apr 3		Apr 10		Apr 17		
												Interim Demo Apr 5					Final Presentation Apr 23	
Nina	Database setup		Database & NLP integration			Microphone setup	Microcontroller & microphone integration		Microcontroller & microphone programming									
Lisa	NLP system programming			Speech recognition programming		Speech recognition & NLP integration	Staff UI design		UI & backend integration									
			Database & NLP integration															
Shiyi	Customer-side UI		Infrared sensor setup			Microcontroller & infrared sensor programming		Microcontroller & infrared sensor integration										
									UI & backend integration									
Everyone						Sound shield installation					Microcontroller & speech recognition integration		Final integration		End-to-end testing			