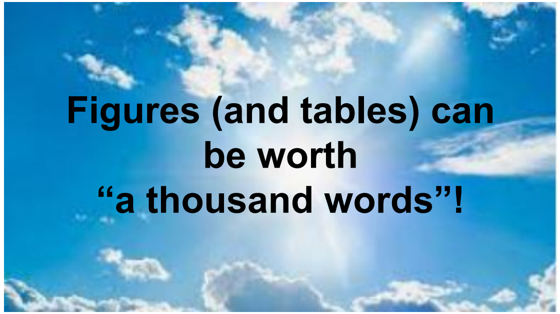
Team D3 – Meal By Words

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 Requirements

- 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

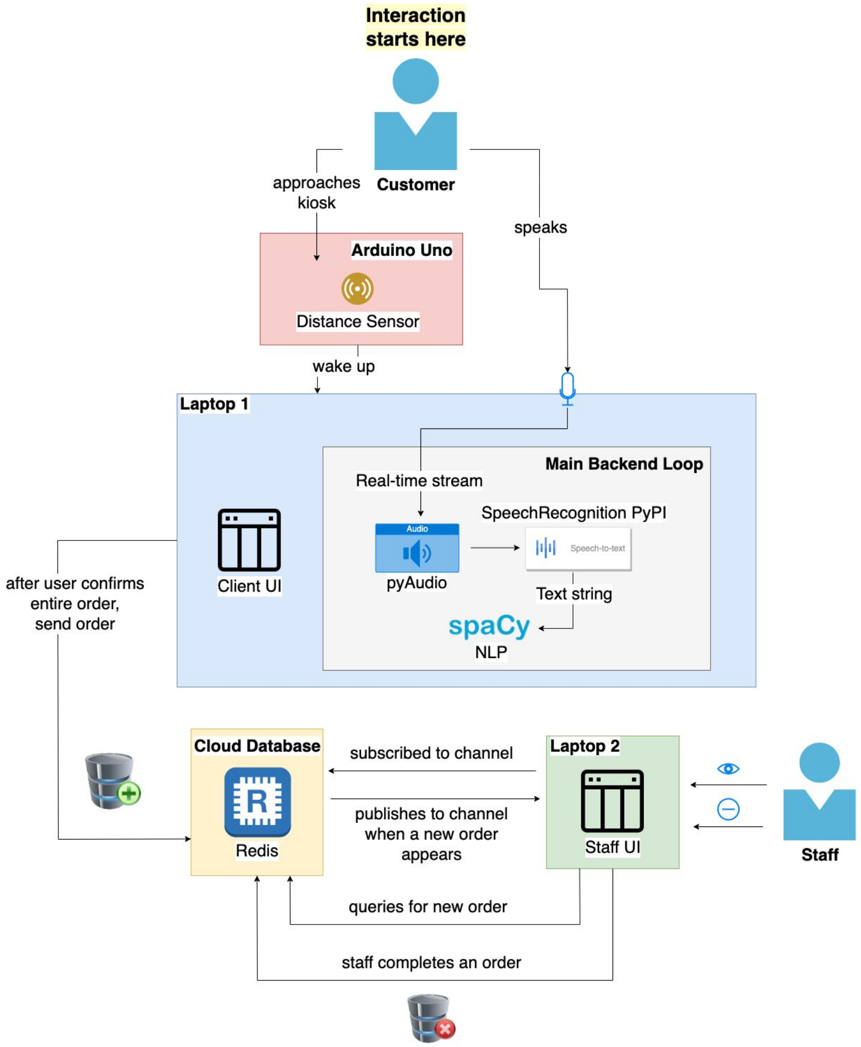
Solution Approach - Overview

1. **Ultrasonic distance 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 staff **UI**)

- **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 ⬆️

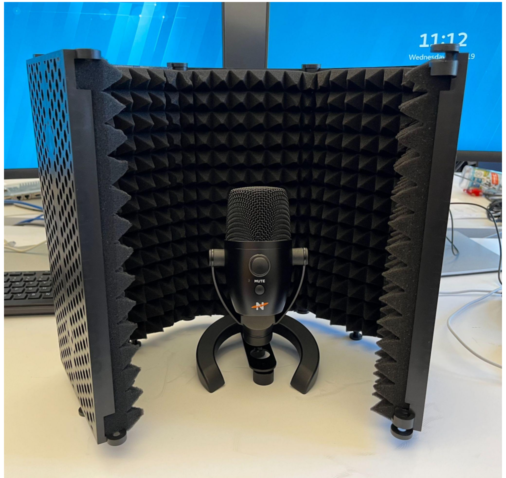
Solution Diagram

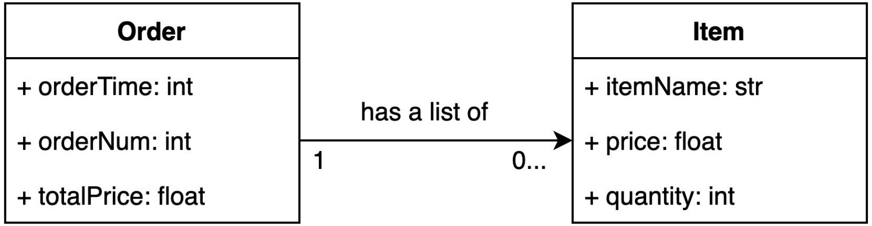
- Raspberry Pi → Arduino Uno
- Microphone connects to Raspberry Pi → laptop that runs customer-side UI
- Amazon DynamoDB → Redis
- Staff-side UI busy waiting loop → subscribe to DB channel for uploads



Complete Solution

- **Speech Recognition**: Python SpeechRecognition library (voice input -> text)
- **Speech Parsing**: spaCy (text -> menu items & quantities)
- **Data storage:** Redis cloud database
  - Document model (JSON-like)
  - Publisher-subscriber relationship between customer-side and staff-side

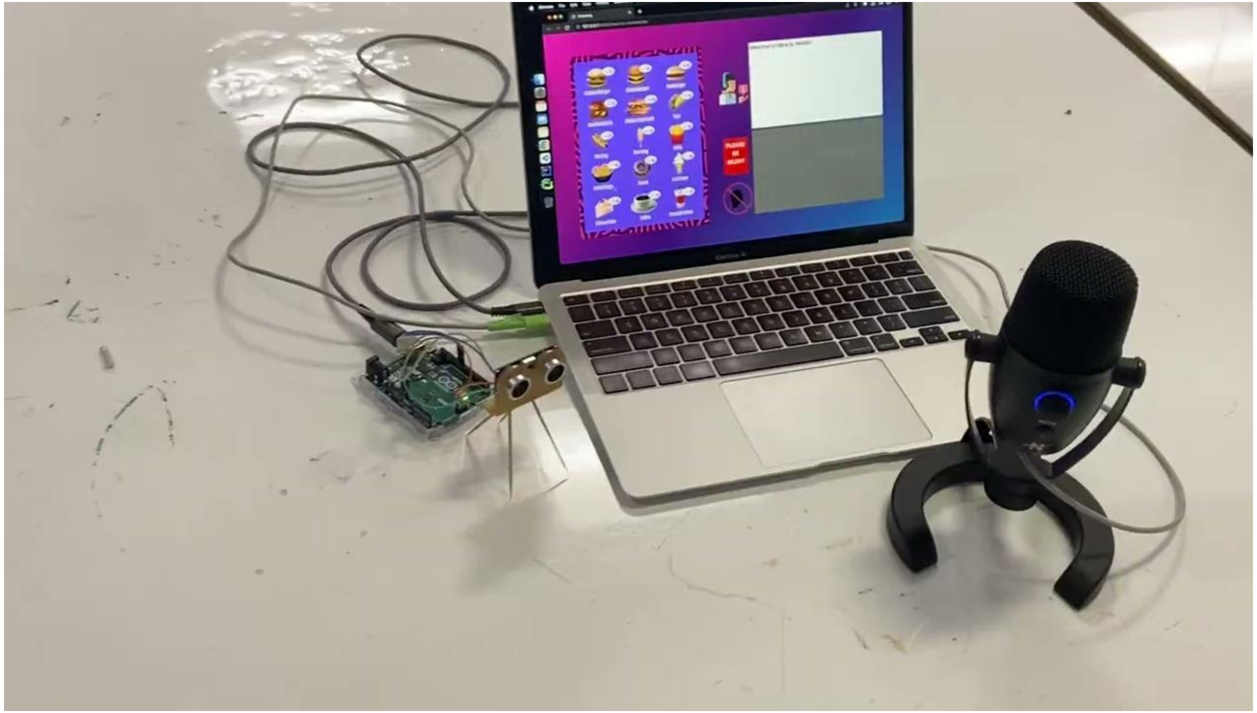




Complete Solution - Termination

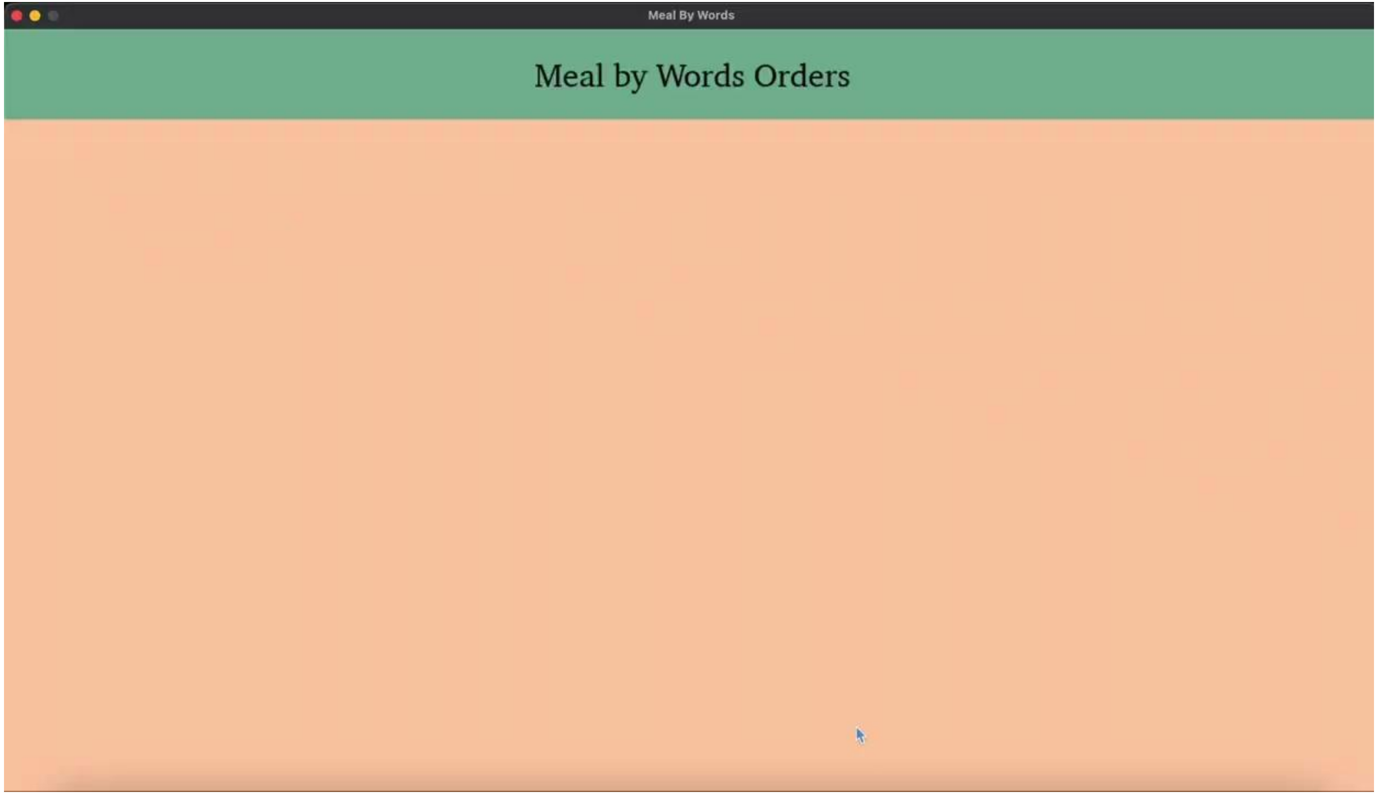
- **Unfinished orders**
  - Distance sensor detects that customer has walked away
  - System automatically terminates the interaction and deletes the order
- **Checkout**
  - Look for termination cues
    - "check out", "finished", "done"
  - Ask customer to confirm order before uploading information to cloud
    - "yes", "correct", "confirmed"

Complete Solution - Customer-side UI



Link: https://drive.google.com/file/d/1rhk311Kv55q_o_GuzdvgGkcVGhh3ORdK/view?usp=drivesdk

Complete Solution - Staff-side UI



Link: https://drive.google.com/file/d/16vVGAu4JFwMCLz74GkGXaG3jXaCvqeDD/view?usp=drivesdk

Engineering Trade-Offs

Say all items in one go		Say items one by one	
Feels natural	Hard to determine when to stop	Increases speech detection accuracy	Lengthens service time
Upload entire order at checkout		**Upload item to database when detected**	
1 DB access per order	May lose current ongoing order	All order data saved as soon as recognized	Repeated DB accesses → latency
Busy waiting		**React to trigger events**	
Intuitive Easy to implement	Wasteful Hard to determine sleep time interval	Saves CPU power System can wake up at any time	Harder to implement Harder to terminate background events

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][*]
- **Kiosk activation time**
  - Customer-side system wakes up within 2s
- **Cloud response time**
  - Staff UI, customer UI ⇄ cloud/database (1s)

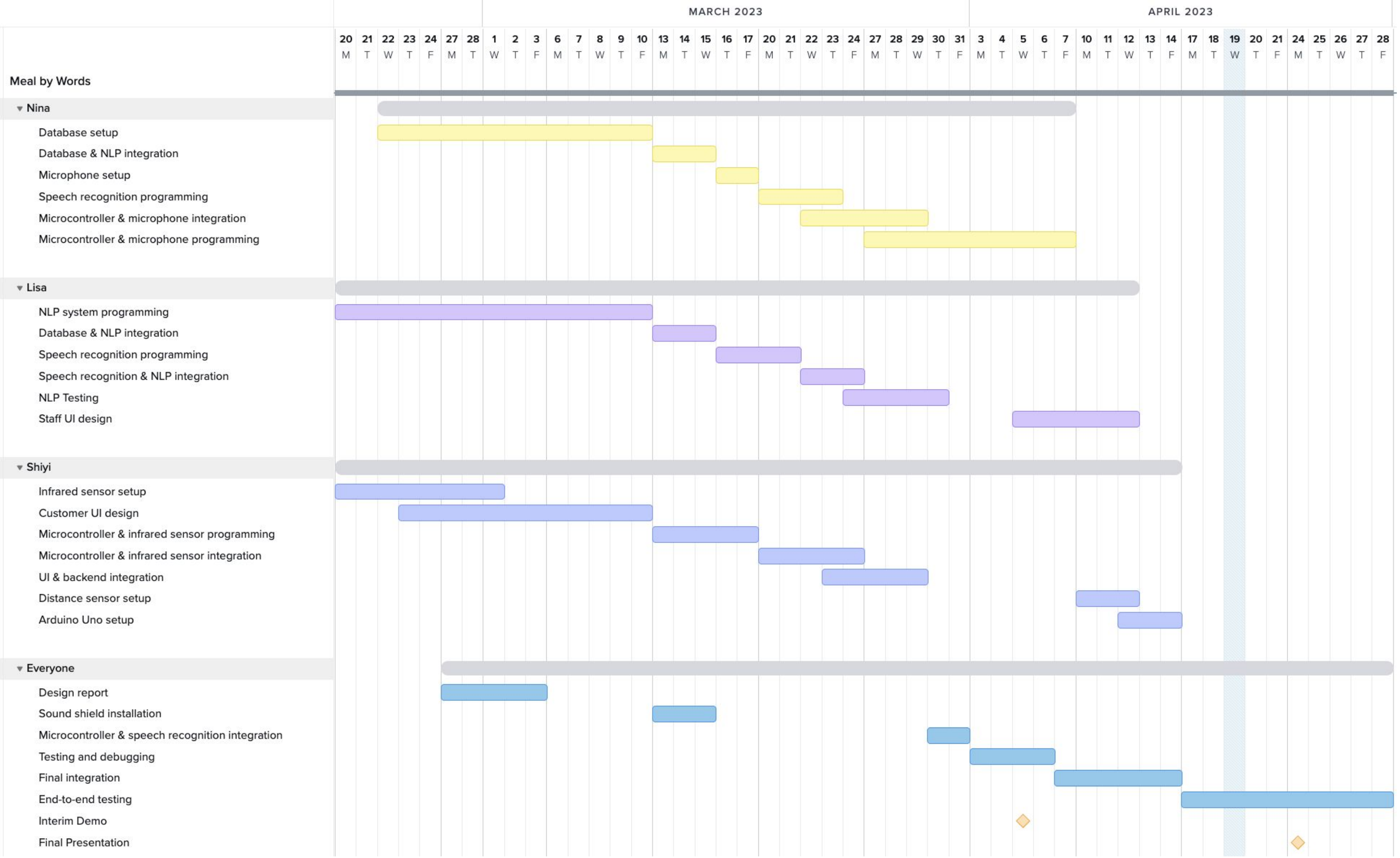
Implementation Performance

Criteria	Expected	Actual	Issues
Staff-side UI latency	1s	Average: 1.638s Median: 1.021s	Latency is highly dependent on internet speed.
Order upload accuracy	100%	Average: 100%	N/A
Text to command accuracy	95%	Average: 100% for basic sentences	Potential edge cases
Audio to text accuracy	85%	Average: 87.9%	Failure on “verb + two” pattern (e.g. “I want 2 hamburgers” → “I want to hamburgers”)
Kiosk activation latency	2s	Average: less than 2 sec, typically 1 sec	Rare when kiosk mistakenly goes to sleep mode due to false distance reading, but it has happened before.

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


MARCH 2023
APRIL 2023
20 21 22 23 24 27 28 1 2 3 6 7 8 9 10 13 14 15 16 17 20 21 22 23 24 27 28 29 30 31 3 4 5 6 7 10 11 12 13 14 17 18 19 20 21 24 25 26 27 28
M T W T F M T W T F M T W T F M T W T F M T W T F M T W T F M T W T F M T W T F M T W T F M T W T F
Meal by Words
Nina
Database setup
Database & NLP integration
Microphone setup
Speech recognition programming
Microcontroller & microphone integration
Microcontroller & microphone programming
Lisa
NLP system programming
Database & NLP integration
Speech recognition programming
Speech recognition & NLP integration
NLP Testing
Staff UI design
Shiyi
Infrared sensor setup
Customer UI design
Microcontroller & infrared sensor programming
Microcontroller & infrared sensor integration
UI & backend integration
Distance sensor setup
Arduino Uno setup
Everyone
Design report
Sound shield installation
Microcontroller & speech recognition integration
Testing and debugging
Final integration
End-to-end testing
Interim Demo
Final Presentation