Use Case: Real-time ASR handling of code-switching…

ID	Name	Description
UL0	Real-time text output	As user actively speaking at a normal rate (100 words per minute), shall return translations in a comparable amount of time compared with typing
UL1	Cross-device support	Our app shall be usable on different laptop devices, laptop OS
UL2	Reasonable output	Shall not display gibberish; shall recognize proper vocabs in English and Mandarin
UL3	Noise tolerance	Shall remain some accuracy when there is noise in the audio
UL4	Reasonable speech length	Shall have the capacity to transcribe a useful quantity of speech in one shot
UL5	Reasonable vocab recognition	Shall recognize vocabularies used in daily life conversations in English and Chinese
UL6	Feasible phone-portability	Shall have the necessary model characteristics to make running on an iPhone feasible

Design Requirements

ID	Name	Description	Value	Test	UL Req.
DR0	End-to-end Latency	Shall return a translation of the first spoken word in no more than 2 seconds	2 secs	ST0	UL0
DR1	Throughput	Shall recognize spoken words into returned text at a rate of .6 secs/word	0.6 secs/word	ST1	UL0
DR2	Translation Error Rate	Shall transcribe speech at a translation error rate of no more than 25%	25%	ST2	UL2
DR3	Noise Tolerance	Shall remain below 30% translation error rate when signal to noise ratio in the audio is higher than 20dB	20dB	ST3	UL3

Design Requirements Cont.

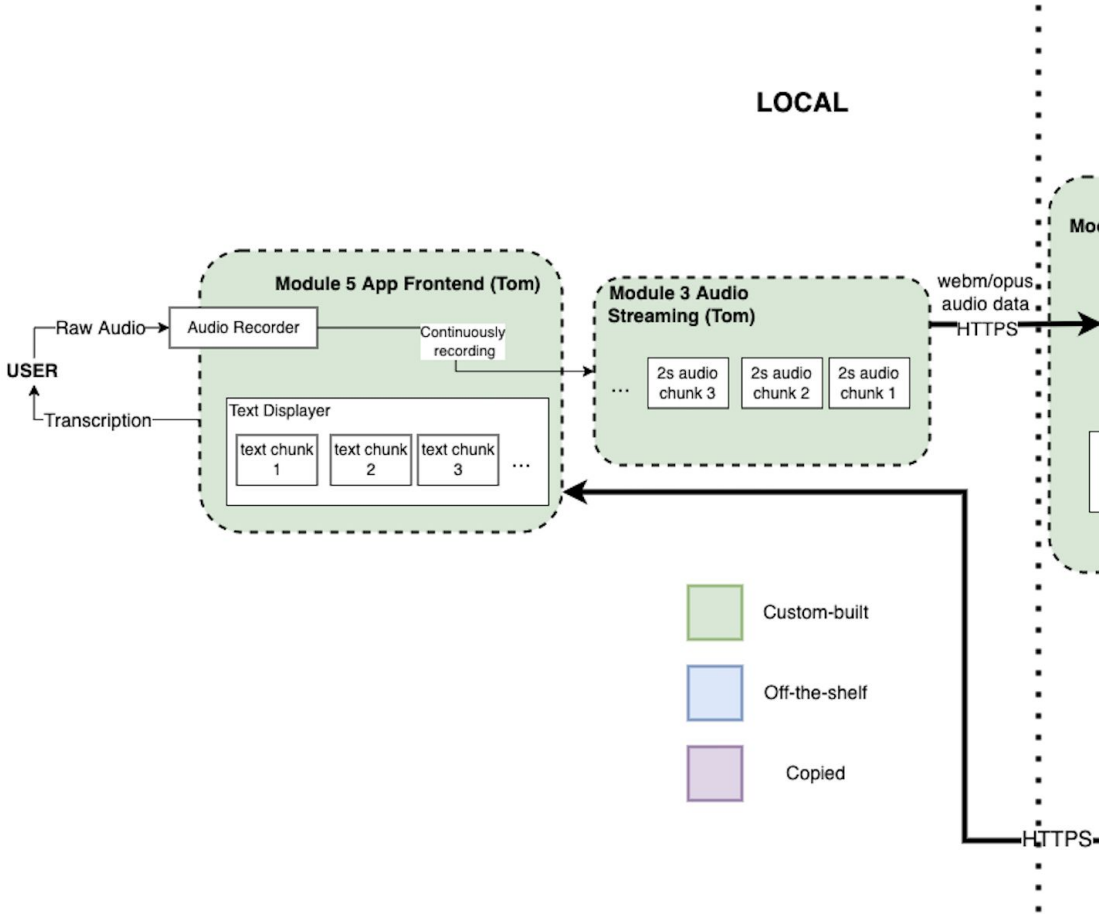
DR4	Character-level Recognition	Shall match up to 10000 characters or sub-words units in vocabulary	10000 words	ST4
DR5	Supporting Environment	Shall be usable on the newest Chrome browser on a laptop	After 1/12/2022	ST5
DR6	Max Data per Input	Shall be able to transcribed a maximum of 1 minute of audio in a single input instance from a user	1 min.	ST1
DR7	Max Model Size	Trained model size (w/o transformer) shall not exceed 20 MB	20 MB	N/A

Solution Approach

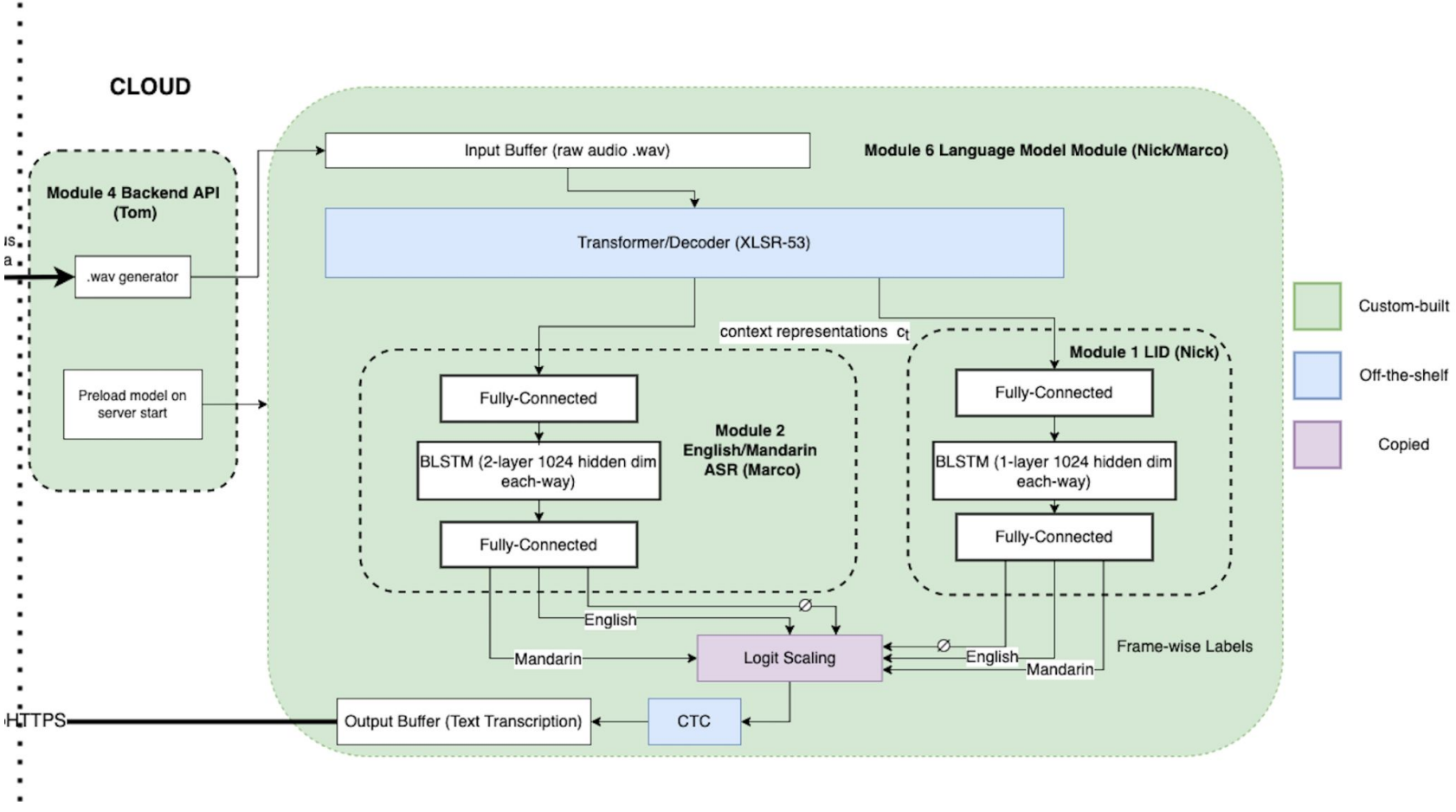
6 Modules, Independently Testable

1. **Language Identification (LID):** Classifies frames into Eng., Mand., Blank
2. **Automatic Speech Recognition (ASR):** Outputs prob. dist. over vocabulary
3. **Audio Streaming:** Chunkifies raw audio, dispatches them to remote API
4. **Backend API:** Receives audio chunks, generates .wav's, requests chunk transcriptions for model, pre-loads model on startup
5. **App Frontend:** Simple, real-time I/O interface for user to record messages and receive transcriptions
6. **Language Model:** Encapsulates and combines submodels, contains all end-to-end training and validation

Solution Approach: Frontend



Solution Approach: Backend



Web Development

- Web Frontend
  - Audio Recorder
  - Real-time Text Transcription Displayer: **continuously** propagates text transcriptions in **mixed Mandarin and English in the order of user's speech**
- Audio Stream Transfer
  - sends new recorded audio chunks **every 2 seconds** to server
  - server on receiving the audio chunks generates .wav files as inputs for backend model
  - **.wav file generation** should < **50ms** to save time for 2s end-to-end latency
  - each request-response pair is marked with a matching **unique sequence ID** for **maintaining order of displayed text transcriptions**
- Backend API
  - **preloads model** on **server start** to save model loading time per request
  - Model runtime for 2-second audio should **< 1.5s**

Language Model

- BLSTMs able to reset for each sequence, provide contextual information
- CTC provides soft-alignment for frame-level classification
- Modularized Initial Training
  - Pre-trained SSL audio-to-phoneme model, needs fine-tuning
  - SpecAugment for training data augmentation
  - Separate loss functions for each module
- Joint End-to-End Training
  - Combined loss functions for LID, ASR, and Phoneme models
  - Allows for overall tuning of model to use-case

Based on: https://arxiv.org/pdf/2110.03504.pdf

Testing & Verification

ID	Name	Procedure
ST0	Latency Test	Continuously recording for 1 minute with our app and log the timestamps when every (2-second audio chunk) request is sent and when corresponding transcription is displayed
ST1	Throughput Test	Pass 1 minute of audio vecs into the model. 1 minute should consist of random frames of English + Mandarin. Begin timer at first audio retrieval, terminate once last label is returned. Divide total number of words/seconds. Repeat 10 times with the same data then avg. Run on AWS g4dn instance. Repeat for Eng., Mand., Mixed.
ST2	Error Rate Test	Use 3 withheld test sets: 1 augmented, 1 English, 1 Mandarin. Calculate total TER for each. None should violate DR2. Run on AWS g4dn instance.
ST3	Noise Test	Using matlab to compute audio samples' SNR to group them by SNR; feed each group to our app to measure average translation error rate for 20, 30, 40, 50dB groups

Unit Testing/Integration

Unit Reqs.

ID	Name	Description	Value	Test	Derived From
M1R1	Throughput	Shall recognize spoken words into returned text at a rate of .6 secs/word	.6 secs/word	M1T1	DR1
M1R2	Classification Error Rate	Shall classify frame speech at error rate of no more than 10%	10%	M1T2	DR2

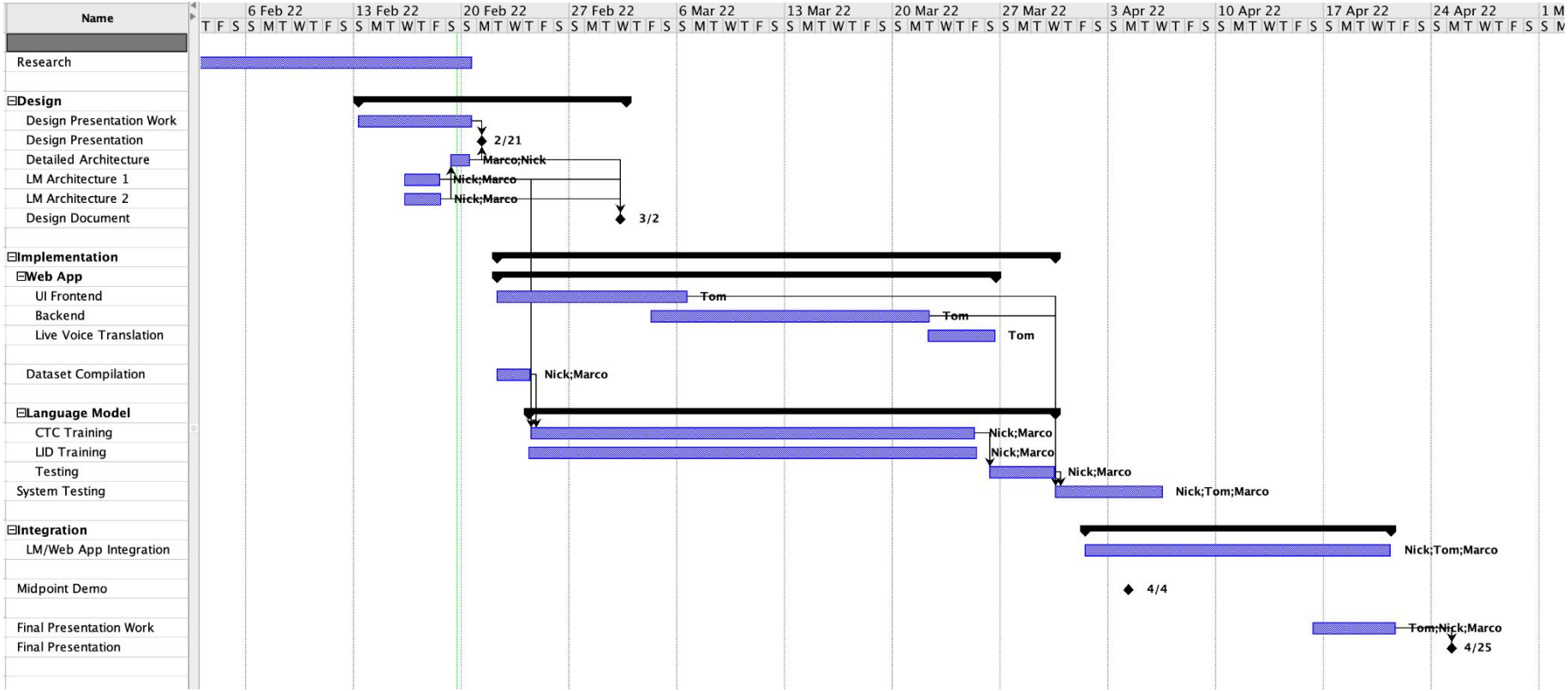
Unit Tests

M1T2	Classification Error Rate Test	Capture the total frame-level classification error rate of the trained LID model when classifying over a withheld augmented (artificially mixed) test dataset.	Use 3 withheld test sets: 1 augmented, 1 English, 1 Mandarin. Calculate total CER for each. Non-should violate M1R2. Run on AWS g4dn instance. Note: Same-sounding but incorrect textual transcriptions should not be calculated as incorrect.

Risks and Mitigation

- Data Issues
  - Have access to SEAME English-Mandarin CS dataset
  - Contingency plans made for mixing English/Mandarin sets
  - SpecAugment for better data value
- Throughput/Latency Issues
  - Shrink/Compress Transformer/Model sizes
  - Shrink audio chunksizes
  - Parallelize inference (multiple instances)
- Accuracy Issues
  - Shrink Vocabulary size
  - Deepen model (not ideal)

Schedule


Name
6 Feb 22
13 Feb 22
20 Feb 22
27 Feb 22
6 Mar 22
13 Mar 22
20 Mar 22
27 Mar 22
3 Apr 22
10 Apr 22
17 Apr 22
24 Apr 22
Research
Design
Design Presentation Work
Design Presentation
Detailed Architecture
LM Architecture 1
LM Architecture 2
Design Document
Implementation
Web App
UI Frontend
Backend
Live Voice Translation
Dataset Compilation
Language Model
CTC Training
LID Training
Testing
System Testing
Integration
LM/Web App Integration
Midpoint Demo
Final Presentation Work
Final Presentation
2/21
Marco;Nick
Nick;Marco
Nick;Marco
3/2
Tom
Tom
Tom
Nick;Marco
Nick;Marco
Nick;Marco
Nick;Marco
Nick;Tom;Marco
Nick;Tom;Marco
4/4
Tom;Nick;Marco
4/25