

Team D0: Karaoke

Anna Gerchanovsky, Anita Ma, Kelly Woicik

Use Case

KaraoKey is a karaoke-inspired vocal coach web app targeted towards helping people improve their singing in real time. We aim to make vocal lessons more fun and accessible to everyone.

Implementation

- Real-time pitch detection
- Melodies and scales for users to sing
- Visual feedback to the user
- Scoring system based on performance

ECE Areas:

- Software
- Signal Processing

Competitive Analysis

- S21: Pitch Perfect
 - Less gamified
 - No real time feedback
- Rock Band
 - Requires specialized equipment
 - Good pitch detection + visualization <3
- Real Karaoke machine
 - Less information
 - Either no feedback or no real time feedback
 - Bulky and inaccessible for at-home consumers



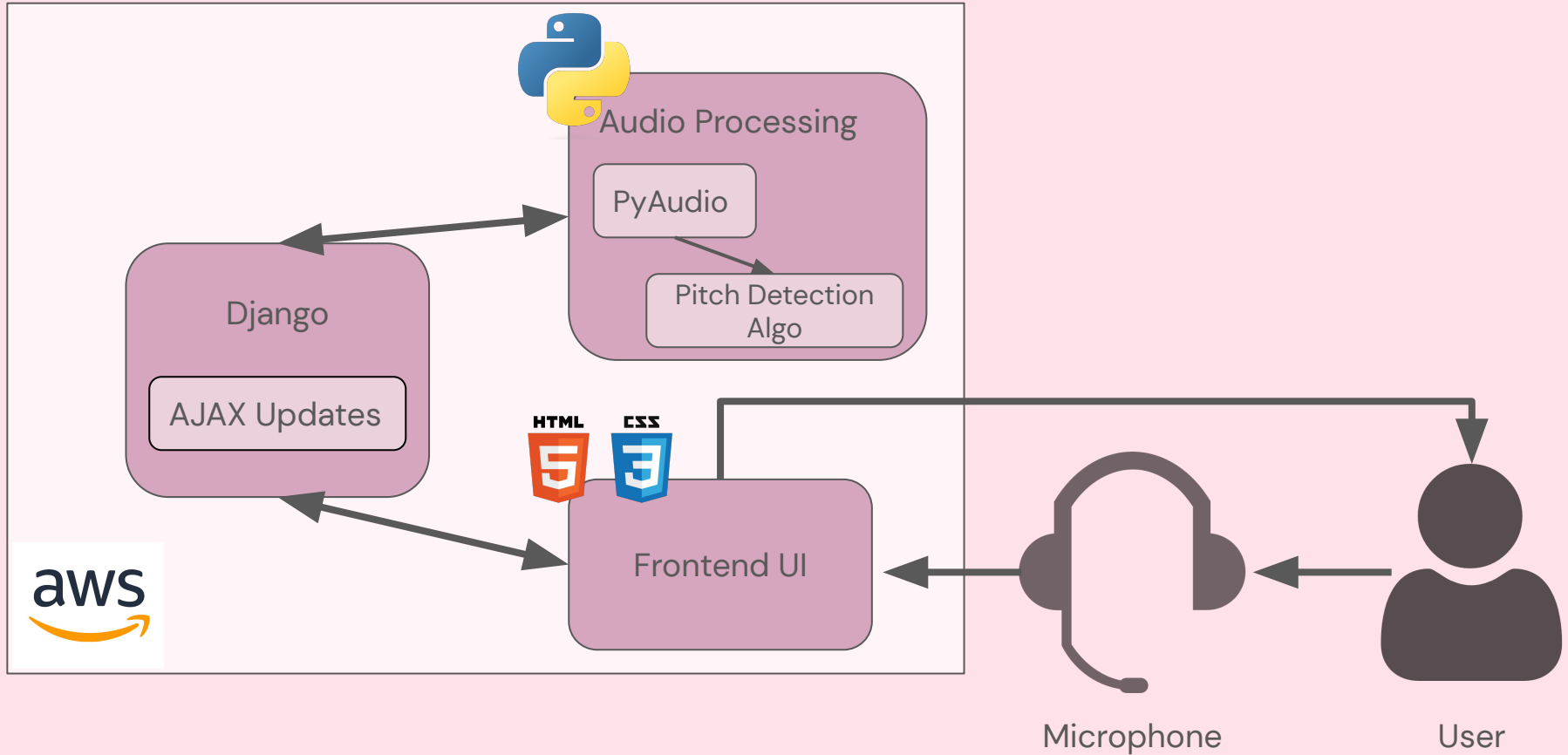
Use-Case Requirements

Requirement	Quantitative Limit (current estimate)
Latency in real time feedback	Under 0.25s
Latency in inputted song processing	\leq the length of the song
Latency in analysis	$< 5s$
Pitch accuracy in user singing (real time)	$\geq 90\%$
Pitch accuracy in inputted songs (asynchronous)	$\geq 95\%$
Latency in AJAX updates	Every .1 seconds

Technical Challenges

Requirement	Quantitative Limit (current estimate)	Technical Challenge
Latency in real time feedback	Under 0.25s	Pitch Detection/ Feedback Algorithm
Latency in inputted song processing	\leq the length of the song	Pitch Detection
Latency in analysis	$< 5s$	Graphing Algorithm
Pitch accuracy in user singing (real time)	$\geq 90\%$	Pitch Detection
Pitch accuracy in inputted songs (asynchronous)	$\geq 95\%$	Pitch Detection
Latency in AJAX updates	Every .1 seconds	Web App Latency

Solution Approach



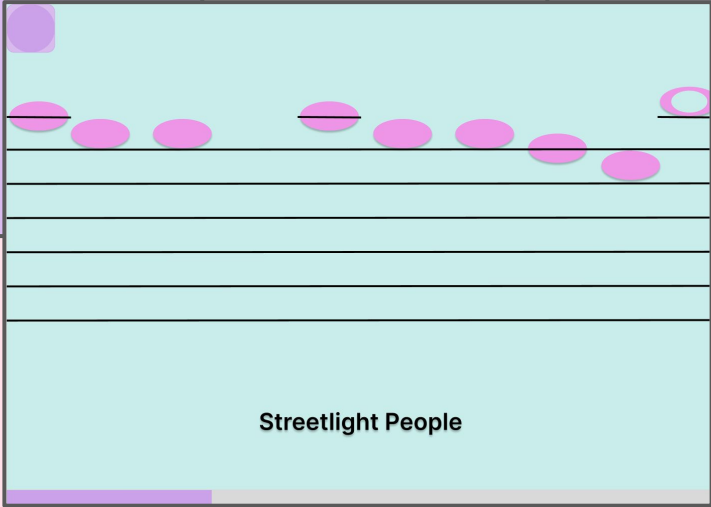
User Interface Mockup



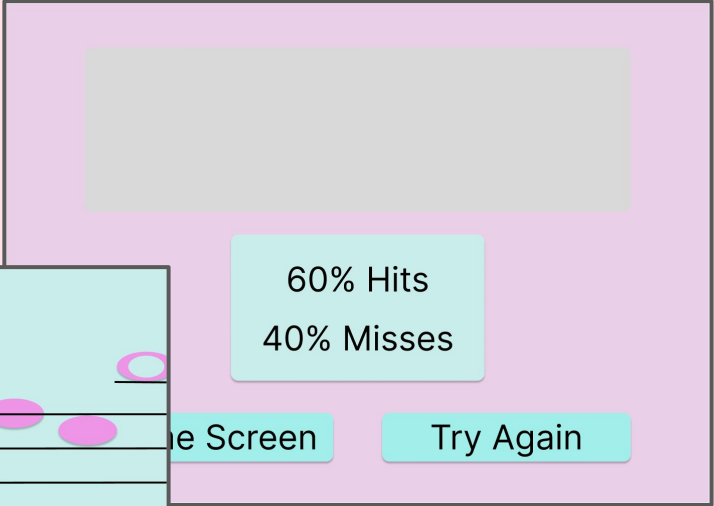
KaraoKey

Login

Sign Up



Streetlight People



60% Hits
40% Misses

Game Screen

Try Again

Audio Analysis and Pitch Detection

- Many pitch detection algorithms (PDA) exist.
 - YIN, MPM, Frequency Domain Approaches
- Goal: Choose a PDA that is efficient enough to meet our latency and accuracy requirements.
- If homegrown algorithms are not efficient enough, we will fall back onto module usage.

Testing, Verification, Metrics

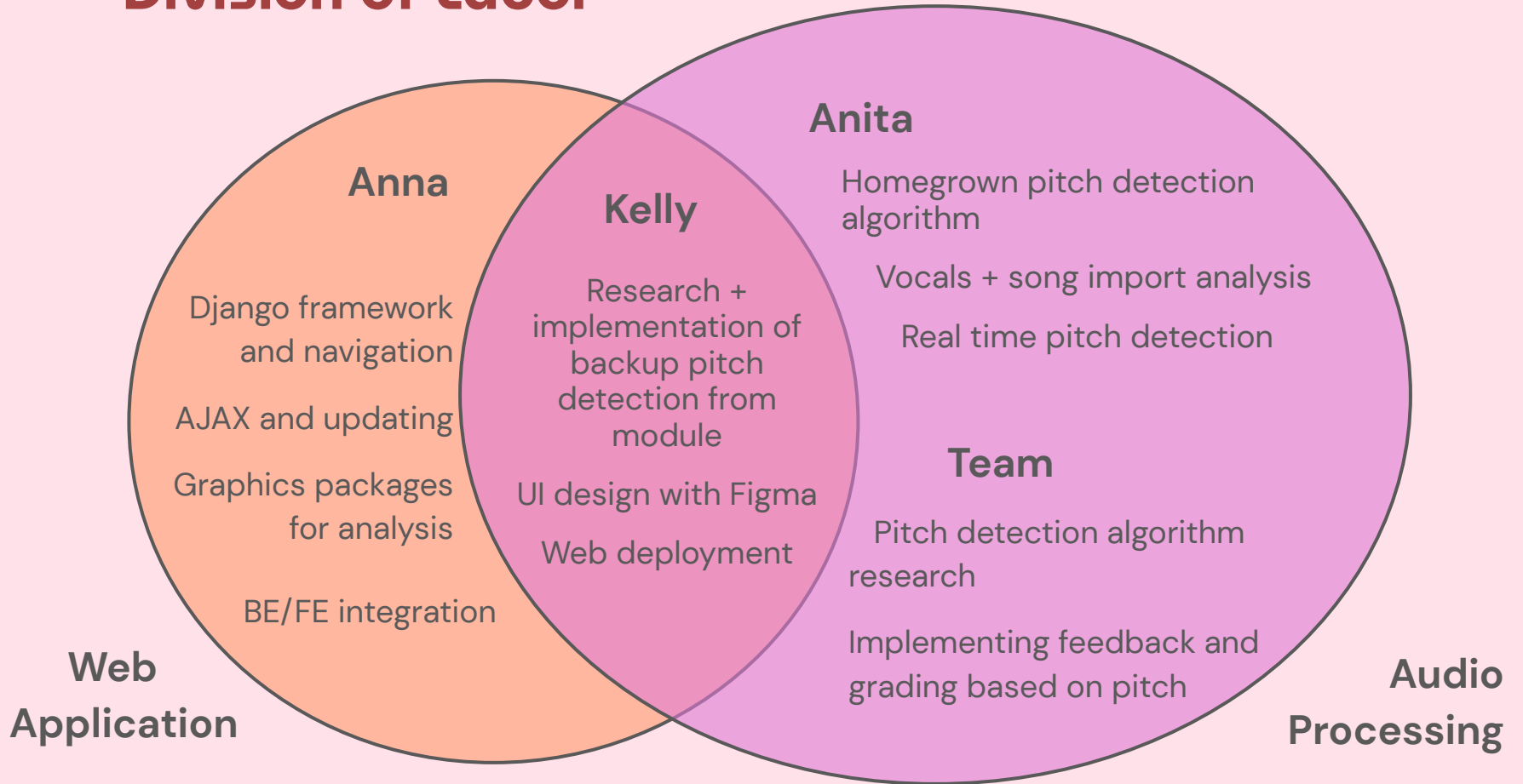
Quantitative Testing

- Testing against vocals / melodies with known pitch
 - Self recorded, vocals + instrumental?
- Pitch detection accuracy
 - Imported song analysis
 - Vocals analysis
 - Real time
 - Post analysis
- Pitch detection latency
 - Automatic time measurements

Qualitative Testing

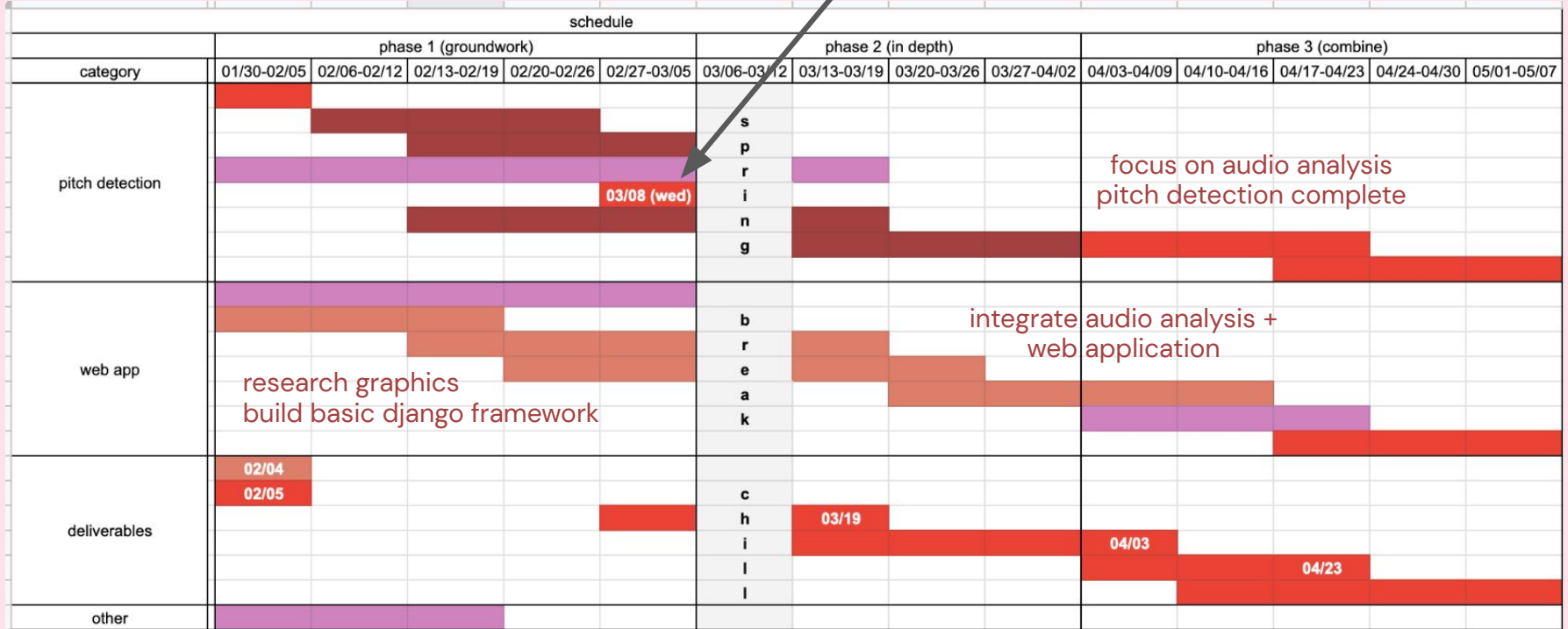
- User Testing
 - Good pitch / bad pitch
 - Random user sampling
 - Improvement
 - Over one song (real time)
 - Over multiple songs

Division of Labor



Schedule

pitch detection finalized
(use own algorithm, or fallback on module)



Moving Forward

- Increase accessibility of vocal lessons
- Foster a casual experience
- Provide real time feedback