

Team D0: Karaoke

Anna Qerchanovskiy, Anita Ma, Kelly Woicik

KaraoKey is a casual karaoke-inspired web app targeted towards helping beginners improve their singing in real time. We aim to make karaoke less intimidating.

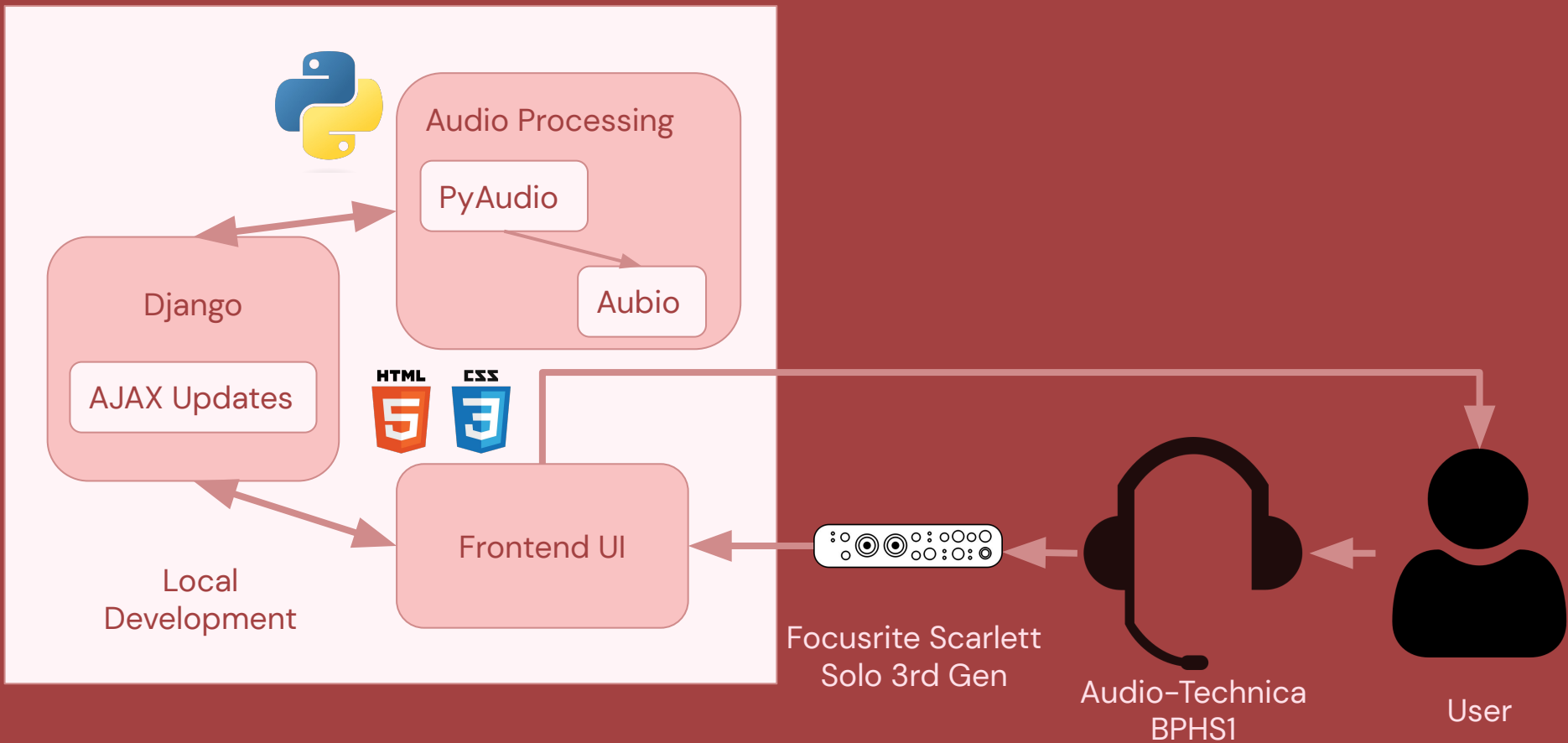
Quantitative Use-Case Requirements

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)	$\geq 95\%$
Headset Microphone	Audio-Technica BPHS1
Audio Interface	Focusrite Scarlett Solo 3rd Gen

Solution Approach

- Feedback
 - Gamified karaoke training
 - Terminology
 - Real time visual feedback
 - Quantitative feedback
- Accessibility
 - Website
 - Well known melodies
- Cultural Considerations
 - Song choices
- Melodies:
 - Twinkle Twinkle Little Star
 - Itsy Bitsy Spider
 - Happy Birthday
- Songs:
 - I'm Yours- Jason Mraz
 - Wonderwall- Oasis
 - Livin' on a Prayer- Bon Jovi
 - I Will Survive- Gloria Gaynor
 - Don't Stop Believin' -Journey

System Specification / Block Diagram



Implementation Plan and Progress

KaraoKey

login

register

quick enter

```
'''
20 Hz is accepted as the lower limit of human hearing, which leads to a
Fast Fourier Transform size of 2205.

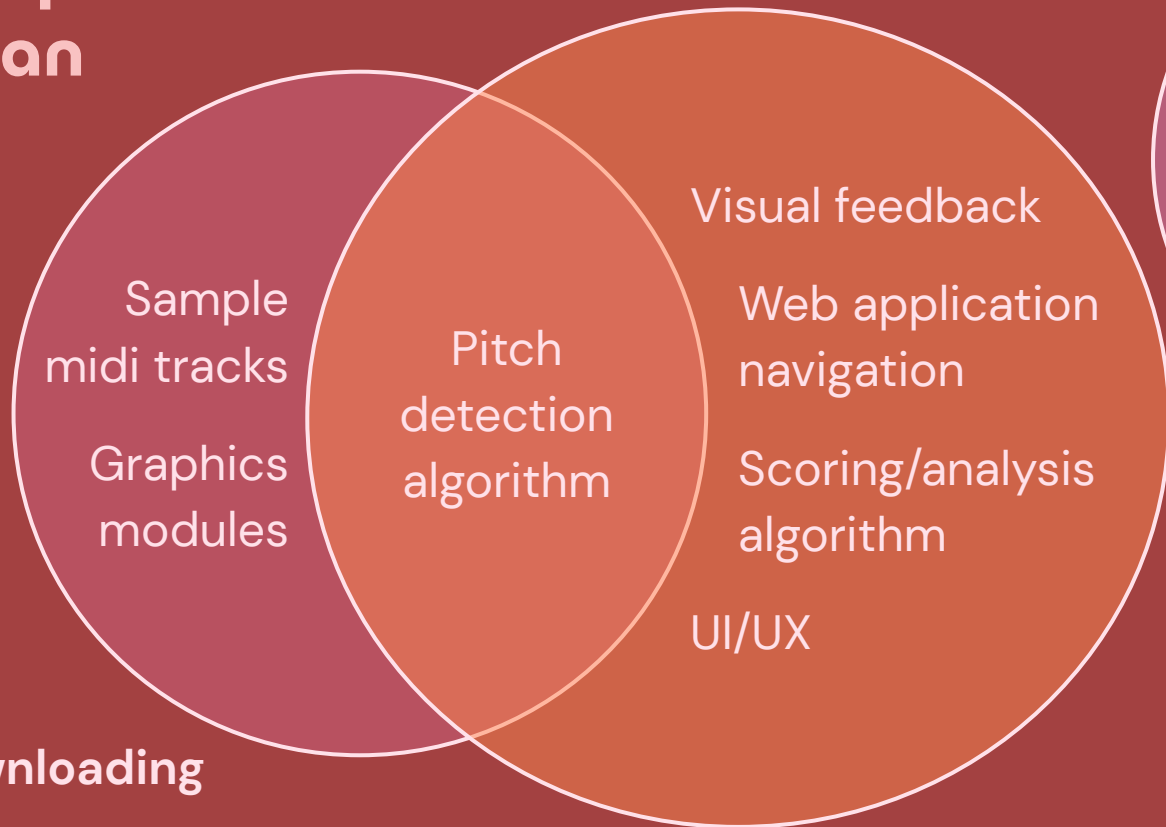
However, this number must be a multiple of 2, so we use 2048 to keep computing
power. Additionally, we don't expect many users to reach this lower limit.
'''

fftSize = 2048
aubioPitch = aubio.pitch("yin", fftSize, frames, sampleRate)
aubioPitch.set_unit("Hz") # make sure output is in Hz
aubioPitch.set_tolerance(0.9) # had to set this value
```

```
257.9384765625
256.5949707031 print("*** starting recording")
258.8707580566 while True:
257.0007324218     try:
250.6801147460         audiobuffer = stream.read(frames)
258.7307128906         signal = np.frombuffer(audiobuffer, dtype=aubio.float_type)
259.1898193359
260.4465026855         signalPitch = aubioPitch(signal)[0]
259.1948852539         confidence = aubioPitch.get_confidence()
259.3930053710
260.25 / 0.991         if (signalPitch != 0.0):
260.3002319335             print(f"{signalPitch} / {confidence}")
259.9943542480     except KeyboardInterrupt:
259.6846313476         print("*** Ctrl+C pressed, exiting")
261.7925415039         break
259.4090881347 print("*** done recording")
259.7821960449     stream.stop_stream()
259.0549011230     stream.close()
259.2745666503906 / 0.9893744587898254
```

Implementation Plan

Developing

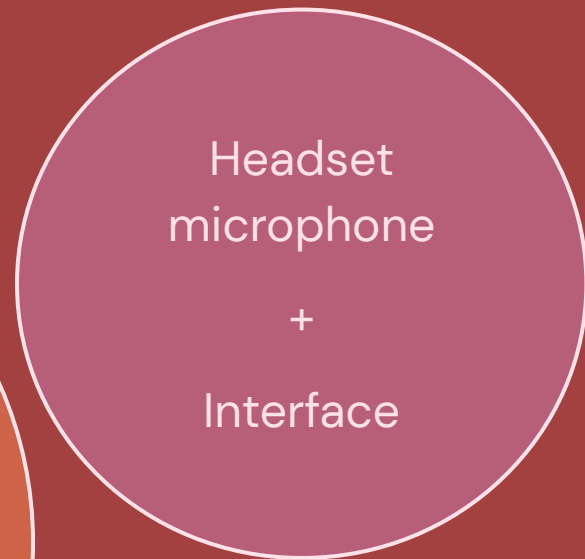


Downloading

Sample midi tracks
Graphics modules

Pitch detection algorithm

Visual feedback
Web application navigation
Scoring/analysis algorithm
UI/UX



Headset microphone
+
Interface

Purchasing

Test, Verification and Validation

How do you test? Comparing recorded wavelength of known pitch to actual wavelength on 2 samples (1 with mistakes, 1 without)

- C major scale
- Twinkle Twinkle Little Star
- I'm Yours

Automated Latency Testing

How do you measure success/passing a test?

- 90% of recorded wavelengths within 5% of actual wavelength
- Notes register within 0.25 seconds of being sung

Murphy's Law

What happens if a test fails?

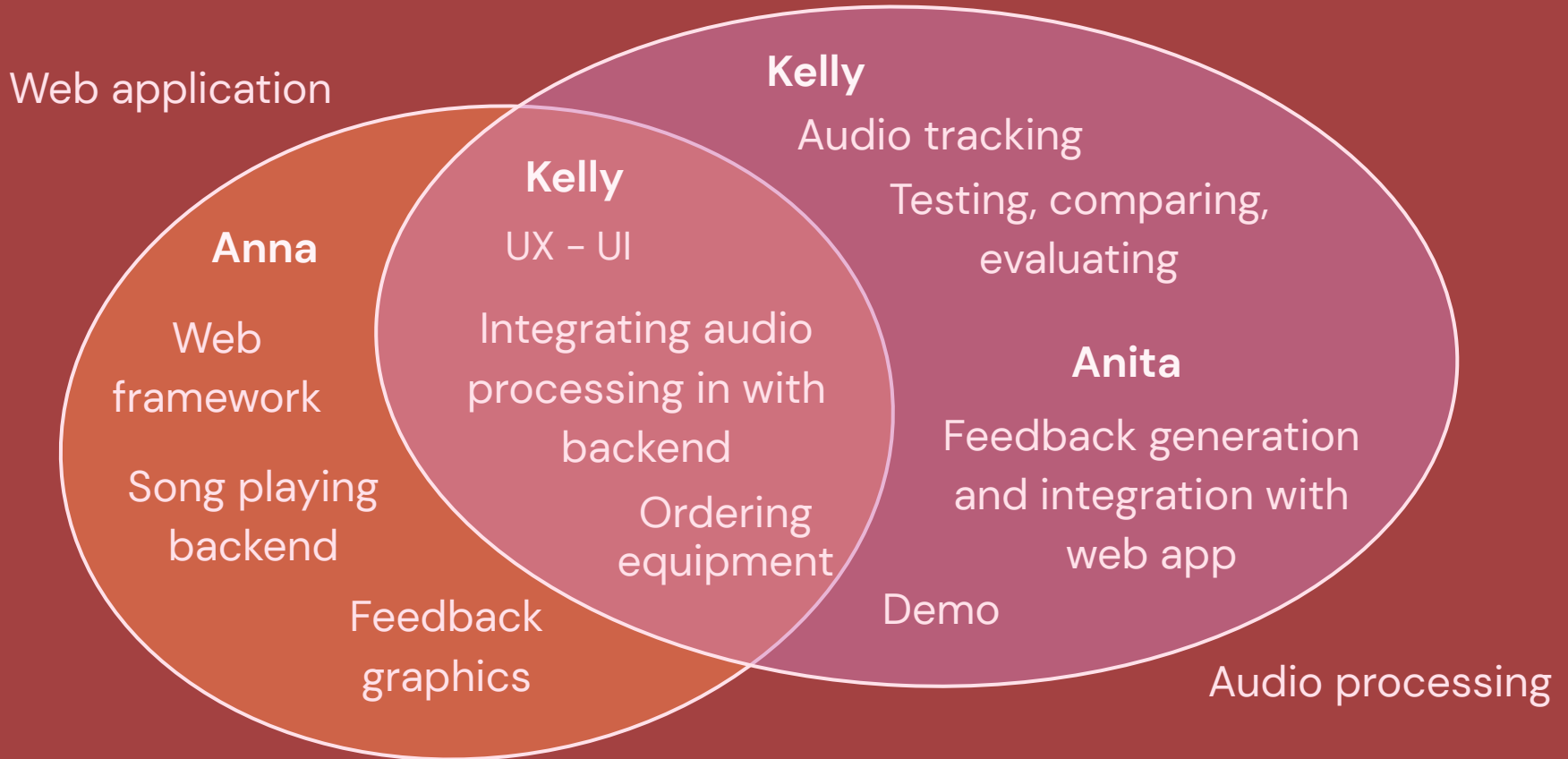
Pitch Tracking Accuracy:

- Test out a different pitch algorithm via Aubio

Pitch Tracking Latency:

- Consider options in C

Project Management - Division of labor



Going Forward

We have adjusted our focus to using existing modules rather than homegrown, and increased emphasis on the gamified karaoke element of the vocal coach.

Our focus is now on creating an enjoyable and helpful experience for users that want to gain confidence in their karaoke skills as they practice.

