D5: Sonic Score Saxophonics

Lin Zhan, Junrui Zhao, Jordan Li

Use Case/Application

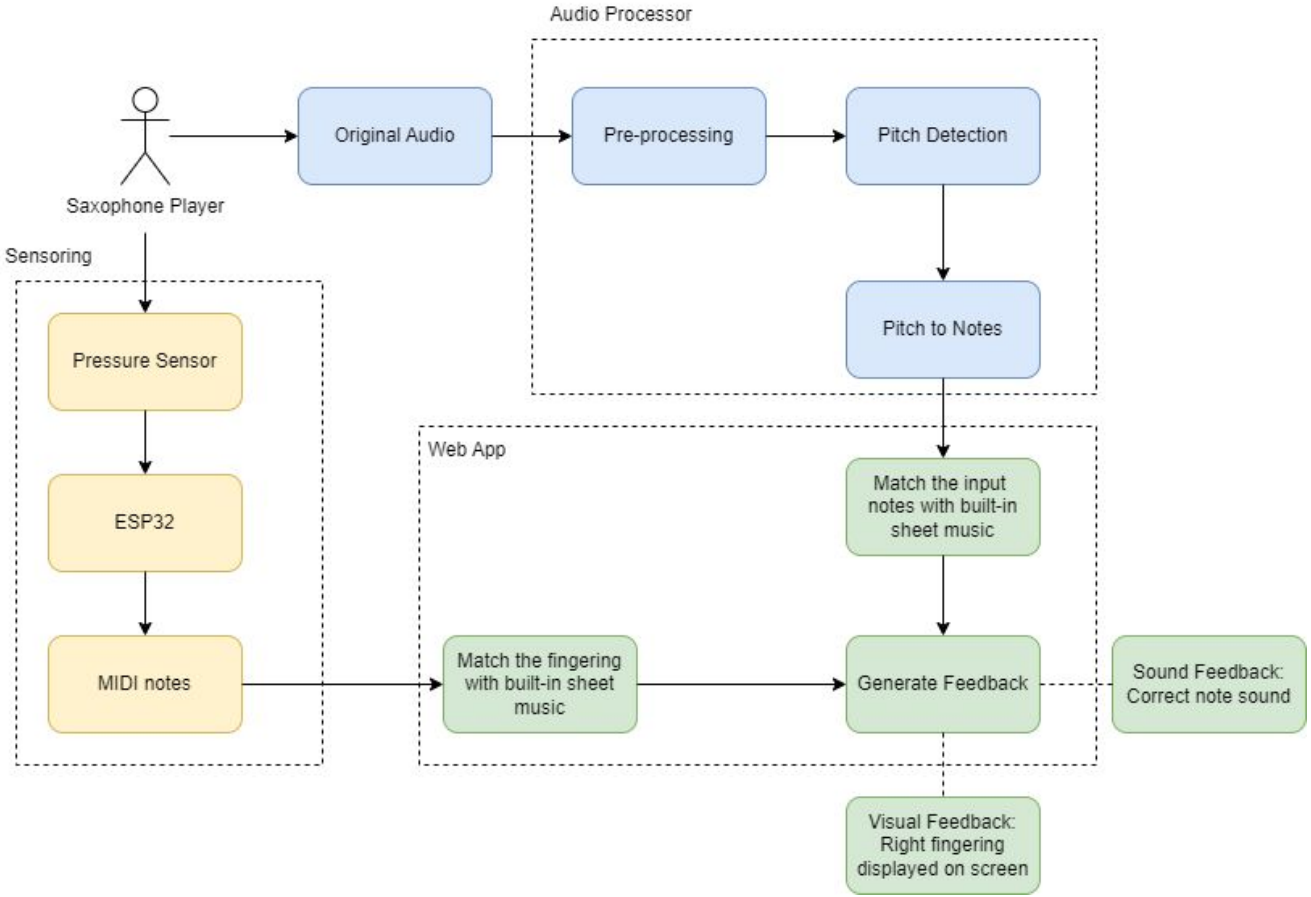
- Problem: Learning saxophone, especially at the beginning, is impractical at home
    - Lessons needed: self-practice can result in undetected errors
        - Note pitch can be different from expected
- Solution: An add-on system of a saxophone to detect fingering and combines fingering and audio data to detect player errors and provide feedback

Quantitative Use Case Requirements

- Accuracy
  - Fingering collection (>= 90%)
  - Audio note detection (>= 90%)
  - Accurate feedback (>= 95%)
    - At most 5% miss when the user’s fingering/audio input is incorrect (false positive)
- Latency
  - Feedback given within 1s (audio and fingering feedback)
  - Overall feedback for a 1-minute playing session given within 3s of finishing session
    - Including error rate, out-of-tune feedback, and suggestions on how to improve

Solution Approach

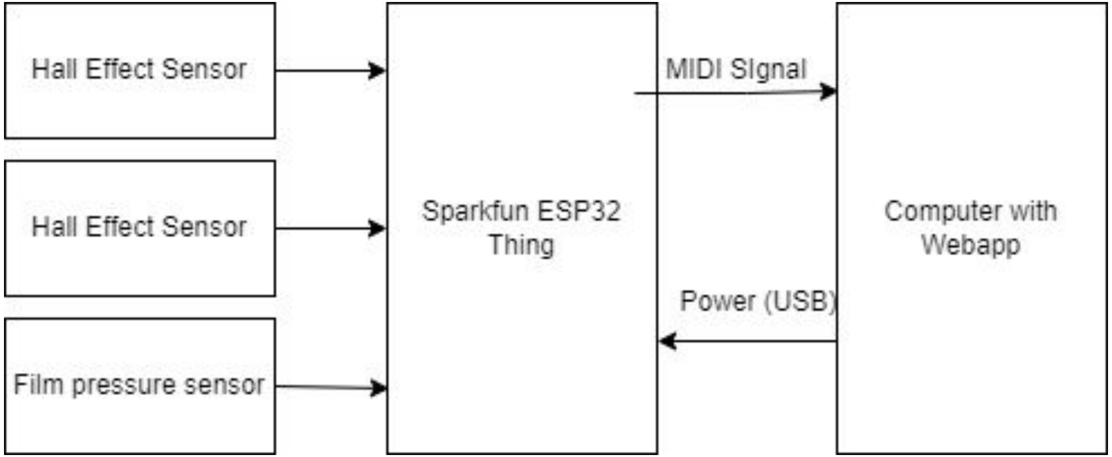
- Fingering Collection
- Audio Processor
- Web App



Audio Processor
Saxophone Player
Original Audio
Pre-processing
Pitch Detection
Pitch to Notes
Sensoring
Pressure Sensor
ESP32
MIDI notes
Web App
Match the input notes with built-in sheet music
Match the fingering with built-in sheet music
Generate Feedback
Sound Feedback: Correct note sound
Visual Feedback: Right fingering displayed on screen

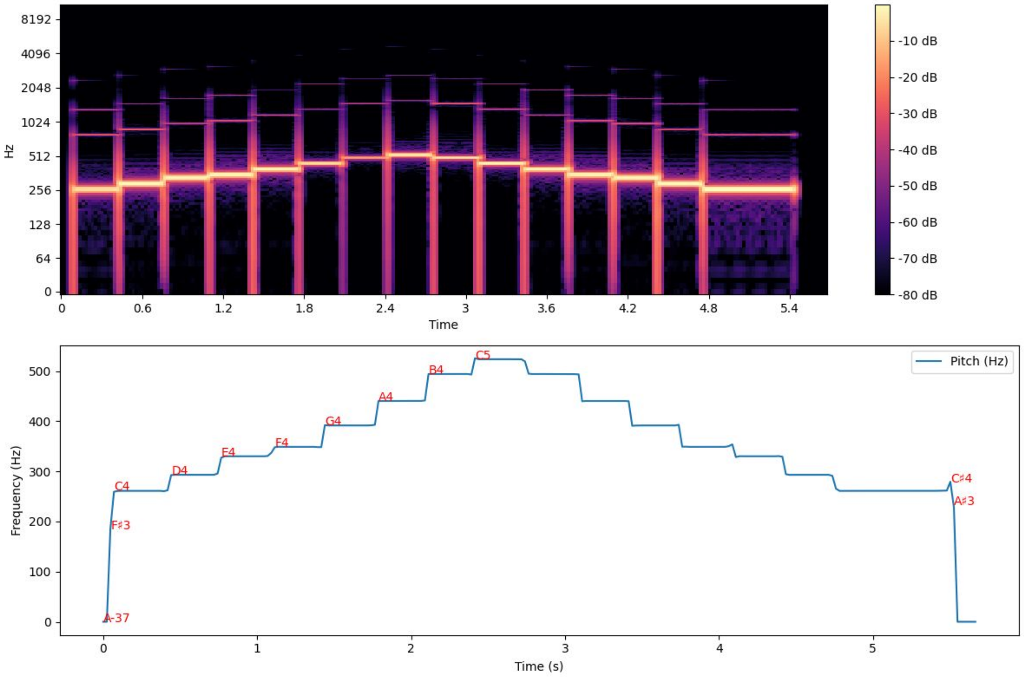
System Specification - Fingering Collection

- Data analysis in ESP32
- MIDI data sent through USB



System Specification - Audio Processor

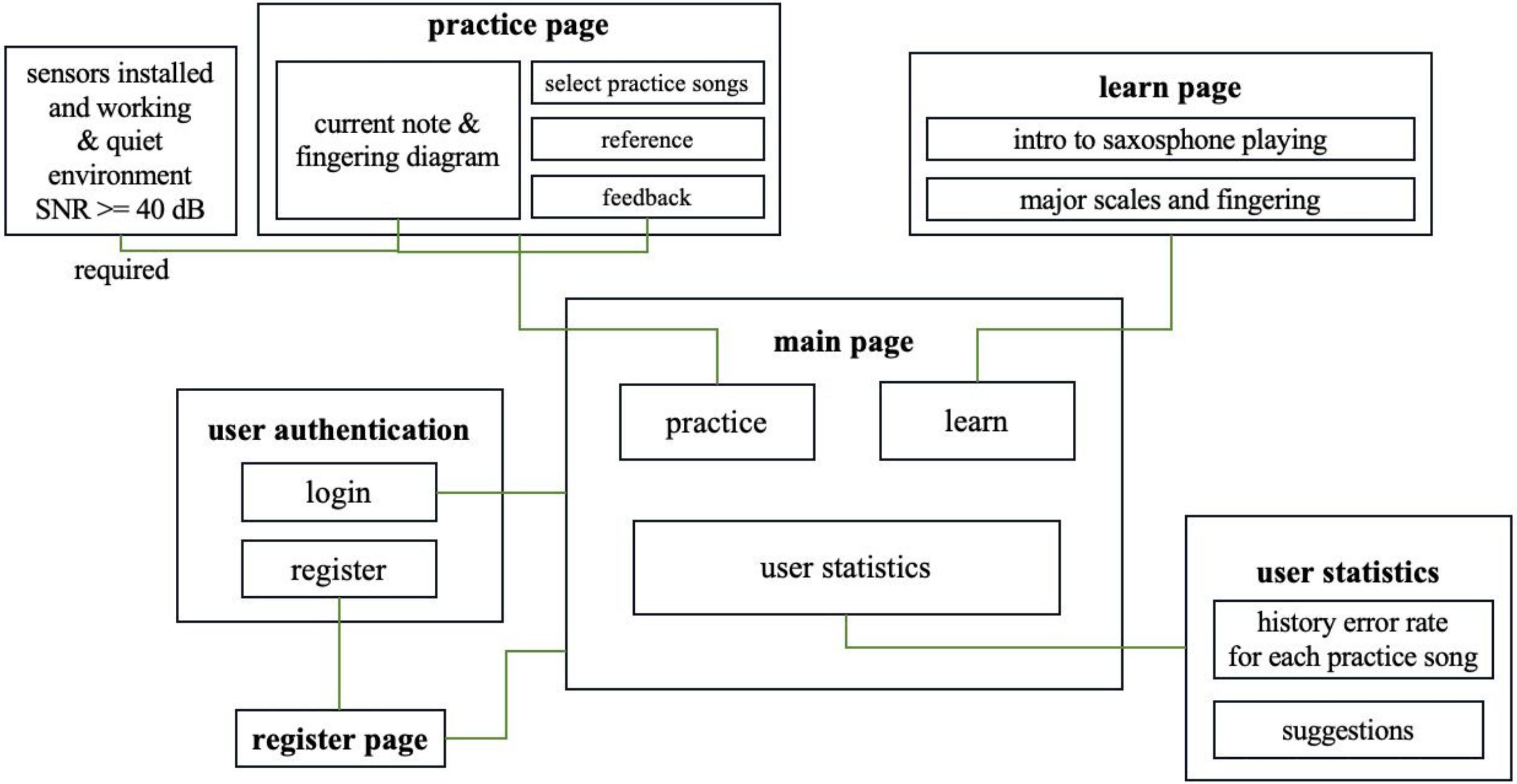
- Pre-processing
  - SNR check
- Pitch Detection
  - Short-time Fourier transform(STFT)
- Pitch to Note
  - Convert frequency to MIDI notes
  - Convert MIDI notes to music notes



Frequency output of an scale music input

Audio source: https://commons.wikimedia.org/wiki/File:12tet_diatonic_scale.ogg

System Specification - Web App


sensors installed and working & quiet environment SNR >= 40 dB
required
practice page
current note & fingering diagram
select practice songs
reference
feedback
learn page
intro to saxosphone playing
major scales and fingering
main page
practice
learn
user statistics
user authentication
login
register
register page
user statistics
history error rate for each practice song
suggestions

Implementation Plan

- Sensor/controller
    - Hall effect, with octave key using film pressure
        - Integration may require additional time
    - ESP32 Thing to collect data and convert to MIDI
- Audio Processor
    - Pre-processing: Band-pass filtering
    - Pitch detection: Short-time Fourier transform(STFT)
        - Back-up plan: Discrete Fourier Transform
    - Python libraries used: Librosa, Scipy

Implementation Plan - Web App

Mockup practice playing page (main function)

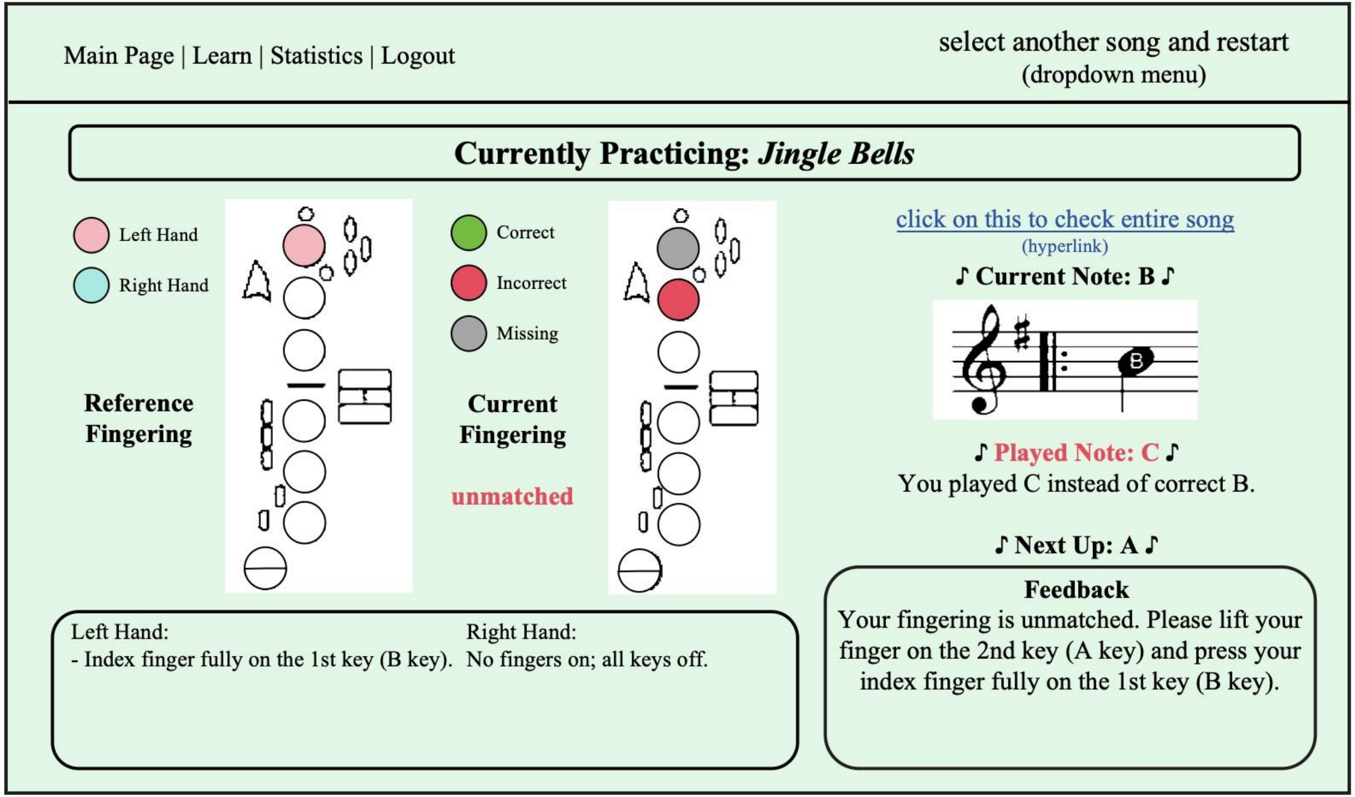
Framework: Django

Database for user info:

SQLite

Info displayed on practice page:

- Current fingering & reference fingering
- Fingering explanation
- Current note & reference note
- feedback


Main Page | Learn | Statistics | Logout
select another song and restart
(dropdown menu)
B

Currently Practicing: *Jingle Bells*

● Left Hand

● Right Hand

Reference Fingering

● Correct

● Incorrect

● Missing

Current Fingering

unmatched

click on this to check entire song
(hyperlink)

♪ Current Note: B ♪

♪ Played Note: C ♪
You played C instead of correct B.

♪ Next Up: A ♪

Left Hand:
- Index finger fully on the 1st key (B key).

Right Hand:
No fingers on; all keys off.

Feedback
Your fingering is unmatched. Please lift your finger on the 2nd key (A key) and press your index finger fully on the 1st key (B key).

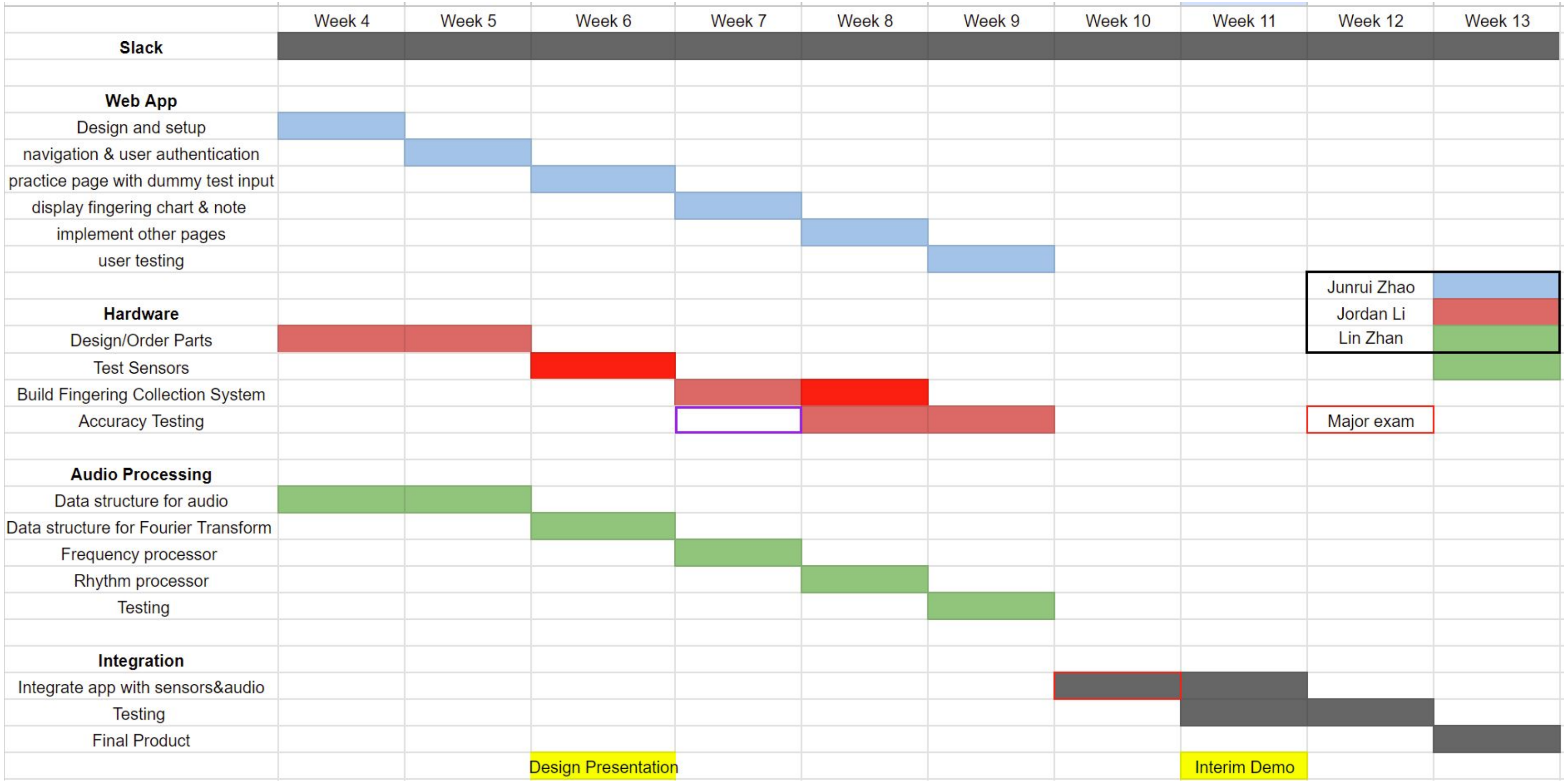
Test, Verification and Validation

Area	Testing Strategy	Testing Input	Metrics
Fingering collection	Test through all combinations of fingerings	Chromatic scale from low B flat to high F (entire range of saxophone)	>=90% of cases match input
Audio note detection	Use tone generator to test our system against TE Tuner	Tone generator, with notes covering entire range of tenor saxophone (Ab2 to E5)	>=90% of cases are within 5 percent of existing tuner app
Feedback error detection	Run previous two tests at same time w/correct and incorrect combos	C major scale / Jingle Bells / Mary had a little lamb with correct and incorrect version (played by Jordan)	>=95% of mismatch cases detected
Latency of feedback	Play one/a series of notes and count the time for feedback generations	C major scale / Jingle Bells / Mary had a little lamb	<=3s for all session, <=1s on average for one note

Risk Factors/Unknowns and Mitigation

- Sensors malfunction
  - Has enough time and $ to buy another model
- Audio Processing inaccuracy
  - Use a different length of sliding window
- High latency
  - Optimize communication between hardware and software
  - Improve algorithm of web app

Project Management



	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
Slack	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛
Web App										
Design and setup	🟦									
navigation & user authentication		🟦								
practice page with dummy test input			🟦							
display fingering chart & note				🟦						
implement other pages					🟦					
user testing						🟦				
									Junrui Zhao	🟦
Hardware									Jordan Li	🟥
Design/Order Parts	🟥	🟥							Lin Zhan	🟩
Test Sensors			🟥							🟩
Build Fingering Collection System				🟥	🟥					
Accuracy Testing				□	🟥	🟥			Major exam	
Audio Processing										
Data structure for audio	🟩	🟩								
Data structure for Fourier Transform			🟩							
Frequency processor				🟩						
Rhythm processor					🟩					
Testing						🟩				
Integration										
Integrate app with sensors&audio							⬛	⬛		
Testing								⬛	⬛	
Final Product										⬛
			Design Presentation					Interim Demo		