Sound Sync

Senior Design

Ramos Adrian,Hernandez Edgar, Farmer Benjamin, Ferguson Patrick, Aguirre Angel

UTA Department of Computer Science and Engineering

Background

Performing music is a craft that takes many hours to complete effectively. One challenge of playing a new song is reading and playing the sheet music repeatedly to understand what spots in the songs are more challenging to play than others. Also, memorizing the start and end of pages allows a smoother transition to turning the page while still playing. This makes the time to learn new music longer than if one could just focus on the notes that need to be played.

SoundSync will be an app that will follow along with the music the user is playing and progress through the sheet music in the app to allow the user to keep playing the song they are on. This will allow users to play a song at their own pace, learning how to play and focusing solely on reading and playing music. Most apps now can write sheet music based on what is being played or play sheet music for the user to hear. These are good for improving a different task at hand. Our objective is unique in allowing users to play the music they want and not worry about having a booklet of music sheets with them or the annoyance of turning the sheet music to keep playing during their performance. Our sponsor explained his passion for music and learning new songs. This app would allow our sponsor and many other music lovers to enjoy learning new music without the hassle of carrying sheet music or turning pages while playing.

Key Requirements

Page Turning/Sliding on Page End – This feature will automatically turn the page of the music sheet being displayed so the user can keep playing without taking their hands off their instrument. It will also do this before the user reaches the end of the page so the page transition doesn't cause a gap in time were the user is waiting to see the next note they need to play.

Sheet Music Scanning – The app will be able to scan music sheets uploaded from the user device. The app will then display these music sheets so the user can play a song and follow along with the music sheet.

Live Note Detection – The app will be able to recognize different notes being played by the user, so it knows when to turn the music sheet page being displayed.

Software Installation – SoundSync will be a web based application. The user will go to the website and begin playing their instrument.

Sheet Music Scanning/Generation Speed – This functionality will allow users to scan music sheets that will be generated as a playable online version for the user to follow along without the need for the physical copy and stored for future sessions.

Note highlighting that follows the user's playing – This feature will highlight the current note the user is on so they know where they are in a song.

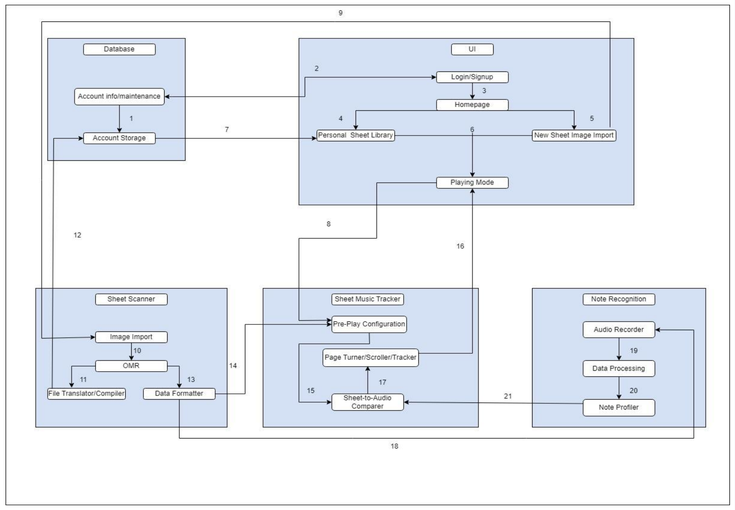
Account Functionality – The application will allow the user to sign up/sign into an account that will then act as a personal database/inventory of the user's previously scanned sheet music as well as their past performance statistics.

User Tutorial Provided on Application Launch – When launched for the first time, the app will give the user a tutorial to show them how to navigate the it and use its features. It will be re-accessible afterwards if needed.

Proper Encryption of Account Information – User-sensitive information such as passwords will be appropriately encrypted.

Architectural Design

We had some very important specifications that our software app needed to meet. The most important being the ability to track where a user is in a song and automatically scroll to the next page when the user is ready to move on. This would entail showing where the user is in the song by highlighting notes so they can be ready for when the page scrolls. But to do this we would need to be able to process audio in real-time and compare that to the notes and chords on the displayed music sheet. Therefore we are constrained by whatever device is recording audio and background interference. Initially, we were gonna rely on the user device to record audio and process it that way. But this may not be reliable as mobile devices don't always make the best audio recorder. So we chose instead to go with sending signals directly to the device from the musical instrument to skip the audio recording and interference entirely. The overall system interacts as so, the user selects music pages from their device and has them scanned through our API. The API puts the music sheets into a desirable format and sends them to the tracker. From there the user can begin the play mode and start playing music. Signals from their instrument will be sent to our C++ server to be processed into notes and chords. This will then match up with the scanned music sheets displayed in the play mode so the highlighting can be shown and the user can see where they are in the song. Once the user reaches the end of the page the program will signal itself to scroll to the next page. This will continue until the user reaches the end of all pages. We initially also wanted to include a pedal for manual page turning but due to time constraint issues we had to opt for not including it.



Implementation Details

Optical Music Recognition (OMR)

SoundSync provides a practical solution for converting sheet music into a digital format. Users upload images of their sheet music, which are then processed on a server with Audiveris, an Optical Music Recognition (OMR) application. Audiveris extracts information about note types and their locations, generating a MusicXML file with the relevant data.

This MusicXML file is then used to recreate the sheet music digitally using LilyPond, a music transcription language and application. The system efficiently parses the MusicXML data to write accurate LilyPond files, which in turn are processed to produce the digital version of the original score.

Once the digital recreation is complete, it's uploaded to a Firebase database along with the parsed data in the form of JSON files. This allows users to retrieve and interact with the digital sheet music, offering a straightforward way to access and utilize musical scores in a digital format.

Audio Recognition Module

SoundSync includes a server-side module hosted on a Raspberry Pi, developed in C++. This module is designed to continuously record audio from the user's microphone in 200ms segments. The recorded audio is analyzed using Fast Fourier Transformation (FFT), a method effective for breaking down audio into its constituent frequencies. This technique enables the module to detect both monophonic notes and more complex polyphonic sounds like chords.

After processing, the audio data is converted into a comma-delimited string format. This format is chosen for compatibility with the evaluation function in our application, which interprets the detected notes.

This module is used in conjunction with the Optical Music Recognition (OMR) output. By combining the real-time audio data with the OMR's note and location information, the system can accurately identify which notes the user is playing and their timing.

Conclusions and Future Work

SoundSync incorporates the overall goals envisioned by Shawn Gieser and our team, capturing the essence of the initial concept. Throughout the development process, we confronted challenges in relation to real-time note recognition, requiring a pivot to ensure a more efficient and accurate note detection. Additionally, the integration of chords into the music sheet tracker posed a comprehensive restructuring of the mapping system for the music notes on the display. Regrettably, these challenges resulted in the inability to fully realize some of the advanced features initially conceptualized for our project. Our commitment to delivering a high-quality app persisted, and we express our gratitude to Shawn Gieser and Jason Losh for their support and invaluable insights. This project has endowed our team with valuable lessons, and we grateful for these experiences for future endeavors.

References

1. Bluetooth. (n.d.). *Specifications*. Bluetooth® Technology Website. https://www.bluetooth.com/specifications/specs/
2. Google. (n.d.). *Server-side encryption | firestore | google cloud*. Google. https://cloud.google.com/firestore/docs/server-side-encryption
3. Ingber, J. (n.d.). *A review of two sheet music apps: SM Music Reader and Sheet Music Scanner*. The American Foundation for the Blind. https://www.afb.org/aw/23/5/17937
4. Joomla. (2020). *Home*. MusicReader. https://www.musicreader.net/en/
5. Library, T. S. L. (n.d.). *Guides: Recording laws: Audio recording*. Audio Recording – Recording Laws – Guides at Texas State Law Library. https://guides.sll.texas.gov/recording-laws/audio-recording
6. Lin, J. (2017, June 12). *Pageflip Firefly Bluetooth/USB page turner review*. The Gadgeteer. https://the-gadgeteer.com/2017/06/12/pageflip-firefly-bluetoothusb-page-turner-review/
7. OSHA. (n.d.). *Department of Labor Logo United Statesdepartment of Labor*. 1926.416 – General requirements. | Occupational Safety and Health Administration. https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926.416
8. Playscore. (2023, May 25). *PlayScore 2 sheet music scanner*. PlayScore. https://www.playscore.co/
9. Texas Government. (2023, May 2). *Privacy and security policies*. Privacy and Security Policies – Texas Parks & Wildlife Department. https://tpwd.texas.gov/site/policies/privacy-and-security-policies
10. Wilson, D. (n.d.). *Understanding scanning requirements*. docufi.com. https://www.docufi.com/how-to-guides/understanding-scanning-requirements

Charts

Feel free to add your data to these charts and add them to your poster.