AUTOMATIC GUITAR TRANSCRIPTION FOR EDUCATIONAL PURPOSES

Kittitat Bamrung

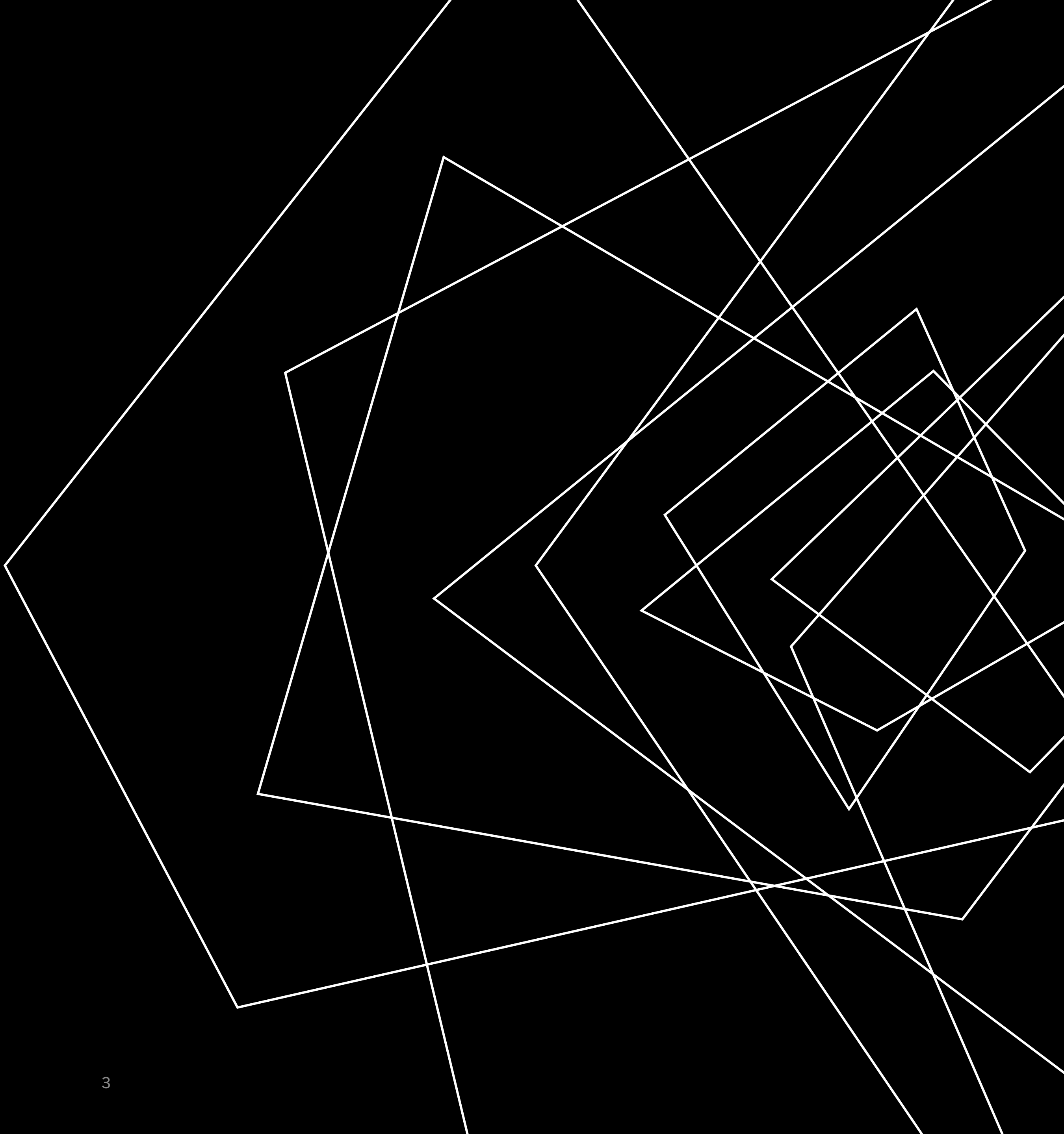
INTRODUCTION

The objective of the project was to develop an application that could transcribe a song into guitar tablature, in a hope that it could help students find a faster and easier way to learn guitar.

RESEARCH

This involves research and study on topics such as:

- Music Source Separation (MSS) – separate guitar track
- Automatic Guitar Transcription (AGT) – uses the result of MSS to output a guitar tablature (tab).
- Display results in Front-end with ability to edit the tab.



TECHNOLOGIES

BACK END

First, a model needed to be developed and trained to transcribe the audio into a guitar tab. In order to do this, I would need libraries such as:

- TabCNN – guitar transcription model.
- TensorFlow – Machine learning library.
- Librosa – Audio processing library to retrieve music information.
- MusicXML – To turn the output of the predictions into a musical sheet format.

TECHNOLOGIES

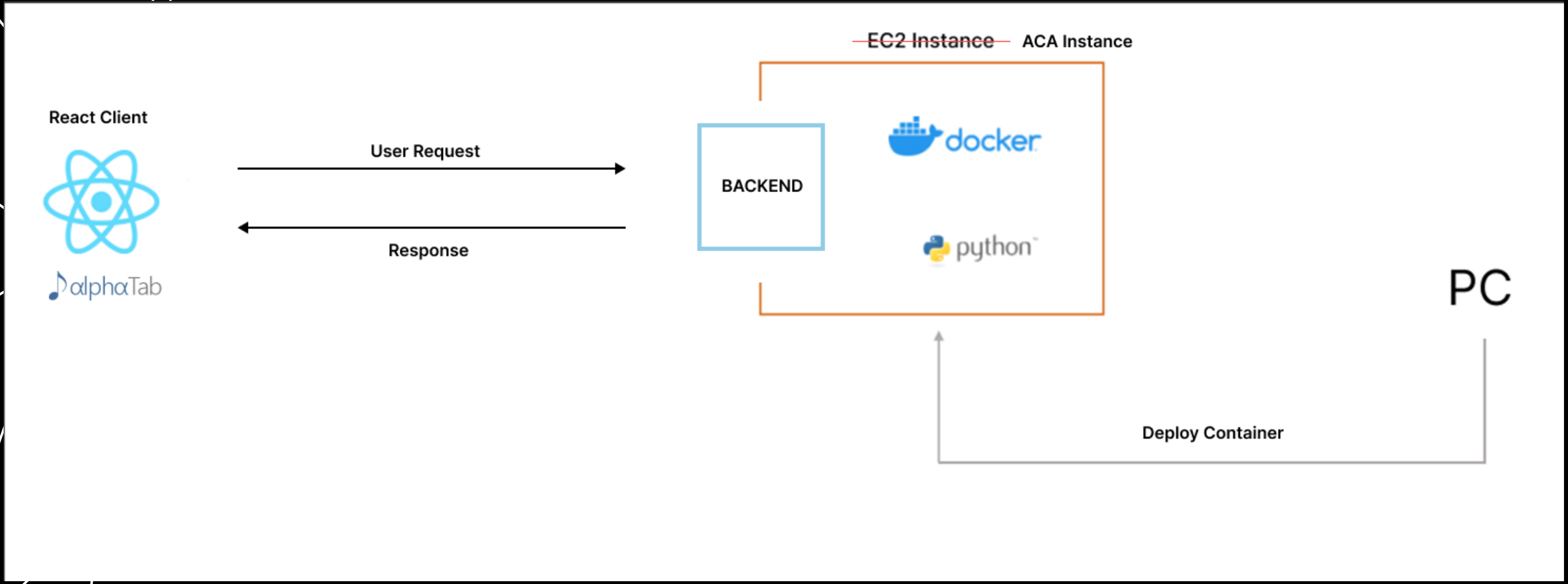
FRONT END

I needed some sort of visualisation library to display the data from the server. I’ve landed on the following libraries:

- AlphaTab – a visualiser for music sheet including MusicXML.
- ReactJS – a framework to build my application.

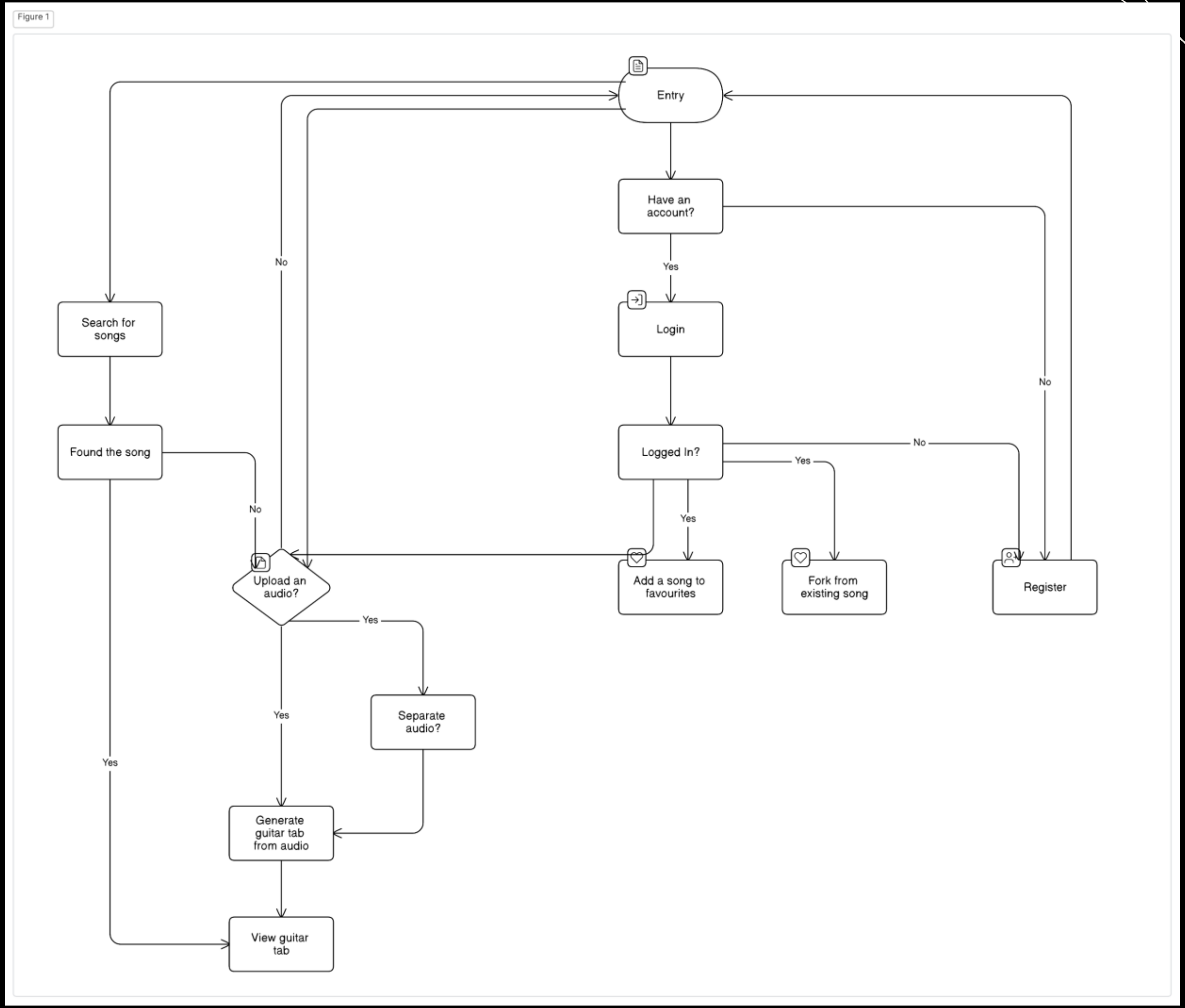
TECHNOLOGIES

ARCHITECTURE



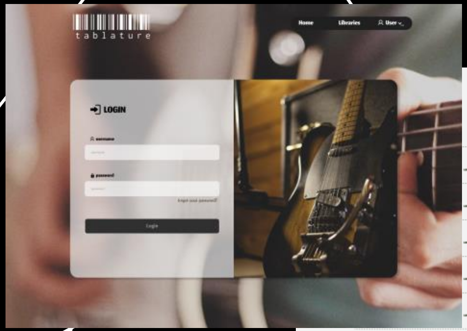
FUNCTIONAL REQUIREMENTS

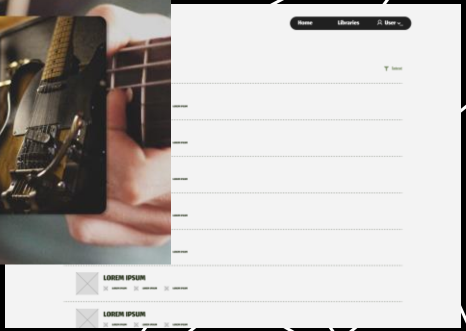
- Functional requirements were outline to what the application needed to be able to achieve. A flowchart was developed to aid in visualising the final look of my application:

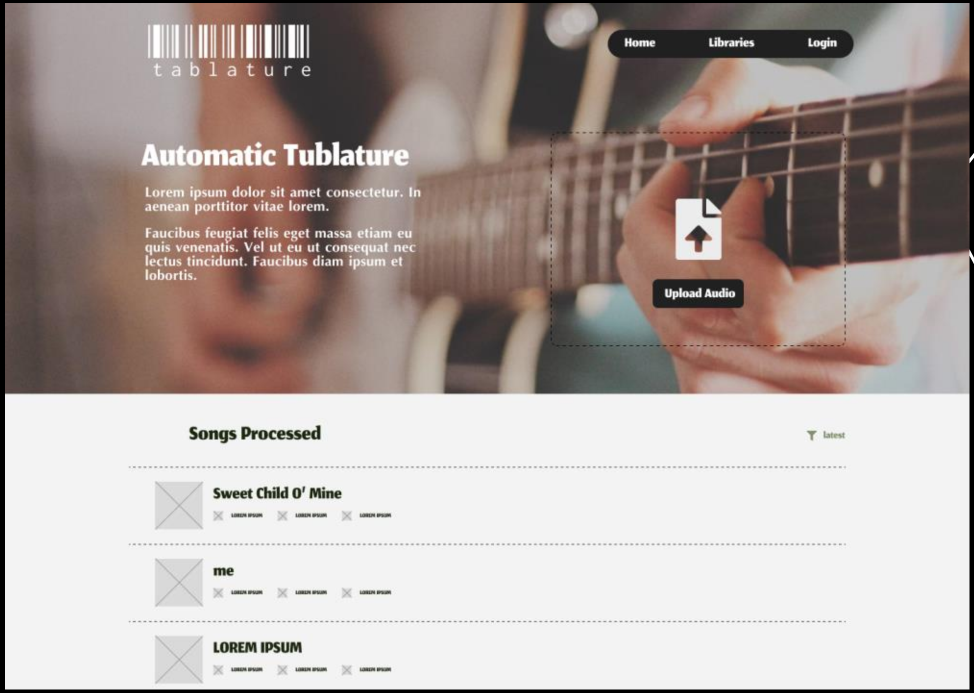


DESIGN

The prototype of the application was designed to act as a checklist for the functional requirements.







Automatic Tublature

Lorem ipsum dolor sit amet consectetur. In aenean porttitor vitae lorem.

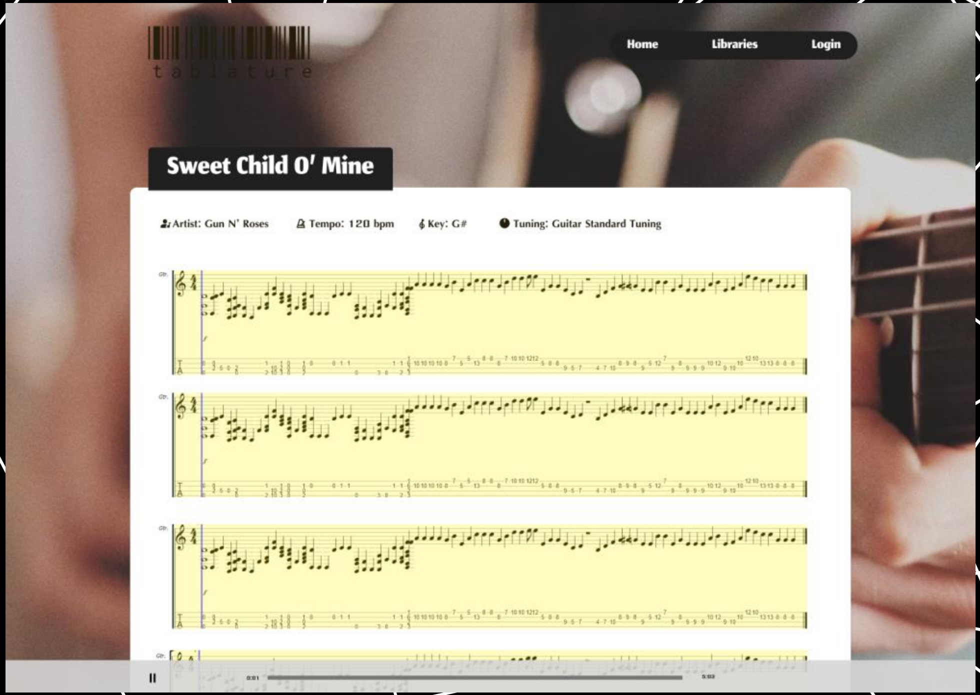
Faucibus feugiat felis eget massa etiam eu quis venenatis. Vel ut eu ut consequat nec lectus tincidunt. Faucibus diam ipsum et lobortis.

Songs Processed

Sweet Child O' Mine

me

LOREM IPSUM



Sweet Child O' Mine

Artist: Gun N' Roses　Tempo: 120 bpm　Key: G#　Tuning: Guitar Standard Tuning

TESTING

AUDIO

Along the course of the project, I’ve performed some audio tests. Here’s what I’ve found:

1. The audio that came from an acoustic guitar achieved better results.
2. If two guitar tracks were present, the model would output the prediction of one guitar tab.
3. The model works well with chords.
4. Even though, there was some misplacement in the prediction of the notes, the actual pitch of these sounds were mostly correct.
5. There can be a lot of bleeding from one note to the next. It became very apparent when an arpeggio was played.

REFLECTION

Here are what I did not managed to achieved:

- The ability to fully add, edit, and/or delete notes on the tab editor.
- Train another model to dynamically recognise guitar techniques i.e. palm-muting, harmonics, strumming etc.
- A better accuracy model that could have provide a more accurate placement of the note/chord.
- A better onset detection to accurately distinguish between a bleed of a note and an actual note.

REFLECTION

On the bright side, here are what I did managed to achieved:

- Less bleeding between notes/chords and a better onset detection through the use of noise reduction methodology.
- The option to separate guitar track from its mixture.
- The chord detection model to help seeing the chord progression.
- The ability to fork and edit song(s).
- The ability to favourite song(s) for later practices.
- In terms of the development, I have gained a lot of knowledge in setting up Python to serve as a server.