NEA 2025

Sam Hunt

Contents

[Analysis 2](#_Toc200621688)

[Documented Design 3](#_Toc200621689)

[Technical Solution 4](#_Toc200621690)

[Testing 5](#_Toc200621691)

[Evaluation 6](#_Toc200621692)

[References 7](#_Toc200621693)

# Analysis

Introduction

The aim of this project is to build a program that assists musicians in easily recording sheet music – to be specific, tab sheets. To demonstrate the issue that this will solve, an example will be given. James is a guitar player / songwriter who has a riff he produced for a new song and wants to share it with his band in advance of band practice later that day. There wouldn’t be enough time to transcribe a complex riff, so James must instead wait until next week’s practice to play it.

A program could exist that allows him to either upload an mp3 file or optionally record audio containing the riff he wishes to transcribe. The program would automatically analyse the recording and create a tab sheet accordingly, with notes laid out and proper timing.

The general problem this project aims to solve is the time it takes to write down and manually transcribe each note, as well as the musical knowledge required, creating a barrier for new musicians and wasting songwriter’s time unnecessarily. Using the planned program to automatically transcribe these notes will save a lot of time and allow them to create more music in less time.

Note

Tablature format (or ‘tab sheets’) refers to the visualisation of a series of notes, typically displayed in an ASCII format, with numbers laid out along six different lines. ‘--4--‘, for example, would indicate that you must pluck the string with your finger on fret four. The specific string to pluck is shown by the line of dashes it appears on. The line second from the bottom of a tab sheet would refer to the second lowest string relative to the others. By convention, the bottom line on tablature format refers to the thickest, lowest pitched string and the top refers to the thinnest, highest pitched one. Using this layout, the correct notes to pluck can easily be identified.

Client

The general client group for this project includes songwriters who want to easily transcribe a guitar solo, and people who play as a general hobby wanting to share riffs they produce. Overall, it is directed towards guitar players, including guitar teachers (by them receiving / sending riffs that they make in the app to their students) and bands (someone creates a riff and can now easily share this to others in the band).

To research this project, an interview will therefore be done on someone easily accessible who has been playing guitar for multiple years now. In doing so objectives essential to regular players can be identified and feedback can be gathered on the outcome of the project, discovering any issues with the programming.

Objectives

This guitar app has a range of objectives, some required for a minimum viable product, and others less essential to the fundamentals of the program. These will be added if there is enough time, and have been clearly marked by section below:

(no letter) – Fundamental objectives relating to key function of the program

B – Other objectives that can be added with less urgency

List of objectives:

* 1: Input handling
  + 1.1: User can upload files in a set format, e.g. mp3.
  + 1.2: User can use files of multiple formats.
  + 1.3: Provide suitable errors for incorrect formats.
  + B1.4: Allow user to record audio directly into the program.
* 2: File processing
  + 2.1: Process the audio to identify pitches of notes.
  + 2.2: Identify the length of notes accurately.
  + 2.3: Ignore any common sources of noise from processing for accurate results.
  + B2.4: Store above data in a suitable variable type.
* 3: Converting to tab
  + 3.1: Map the detected notes in objectives 2.1 and 2.2 onto a tablature format
* 4: Displaying tab sheets
  + 3.1: Have a method of displaying the tab sheet that works every time.
  + 3.2: Display notes in correct positions from objective 3.1.
  + 3.3: Allow user to copy the tab sheet as text with suitable spacing for readability.
  + B3.4: Allow user to save the tab sheet as an image.
  + B3.5: Implement feature to adapt to different guitar tunings, if manually input, and have the notes move to the correct notes.
  + B3.6: Display the selected tuning in the exported tab format to make it clear for other musicians.
* 4:

Project Research

Existing tools

There are existing programs that allow users to manually transcribe their songs to tab format, or even newer ones that can use artificial intelligence to transcribe for them. However, these have their individual drawbacks.

One example is an online tool called ‘Guitar2Tabs‘. This is a tool which utilizes artificial intelligence to automatically transcribe notes and chords as accurately as possible.

A screenshot of a website

AI-generated content may be incorrect.Their home page states that you can not only record and upload sound files, but can also use a YouTube video link for transcription. Additionally it has a range of output options, not just tablature format.  
Below is another image taken from this website listing powerful features.

A screenshot of a music tab

AI-generated content may be incorrect.Most of these features are out of scope for this project and require a much larger scale to implement. One example is the multi-instrument support. This project will only transcribe guitar input, and multiple tracks playing at once will not work.  
This is an example of a more advanced version of this project’s outcome. Instead of adding lots of additional features, the basics will be focused on.

An example of a fully manual transcription tool is a feature of a mobile application called ‘Ultimate Guitar’.

A screenshot of a computer

AI-generated content may be incorrect.Ultimate Guitar makes users upload tab sheets in pure ASCII format, which is tedious, and formatting manually wastes time. Interview – add alex opinions

A screenshot of a computer

AI-generated content may be incorrect.

Above: Ultimate Guitar homepage

A program that combines the simplicity of Ultimate Guitar and fast transcription offered by Guitar2Tabs is ‘My Guitar Tabs’.

A screenshot of a computer

AI-generated content may be incorrect.

This program allows users to record the position of each note played but does not require them to manually format the tab sheet.

A screenshot of a computer

AI-generated content may be incorrect.This is an example of the controls provided to allow users to place notes in the transcription, with support for note slides, vibrato and more.

Overall, there are tools that go all in on ai to analyse recordings for conversion but end up having subscription fees, whilst the rest rely on fully manual note entry.

This project aims to create a program in between the two, with automatic note detection but using a more lightweight method. It would be nice to allow for editing of tabs in the final product, but this is an optional task.

# Documented Design

….

# Technical Solution

…

# Testing

…

# Evaluation

…

# References

AQA. (2025, 06 19). *NEA Information*. Retrieved from aqa.org: https://www.aqa.org.uk/subjects/computer-science/a-level/computer-science-7517/specification/subject-content/non-exam-assessment-the-computing-practical-project

Hills Road Staff. (2025, 06 12). *HRSFC Computer Science*. Retrieved from Sharepoint: https://hrsfc.sharepoint.com/sites/CompSci-Student

*NAudio Documentation*. (2025, 10 13). Retrieved from GitHub: https://github.com/naudio/NAudio