Audio to Guitar Tablature Conversion

Project Team

Shahzaib Ansar	20P-0473
Sakhawat Ali	20P-0478
Rafay Ather	20P-0457

Session 2020-2024

Supervised by

Ms. Sara Rehmat

Department of Computer Science

National University of Computer and Emerging Sciences
Peshawar, Pakistan

June, 2025

Student's Declaration

We declare that this project titled "*Audio to Guitar Tablature Conversion*", submitted as requirement for the award of degree of Bachelors in Software Engineering, does not contain any material previously submitted for a degree in any university; and that to the best of our knowledge, it does not contain any materials previously published or written by another person except where due reference is made in the text.

We understand that the management of Department of Computer Science, National University of Computer and Emerging Sciences, has a zero tolerance policy towards plagiarism. Therefore, We, as authors of the above-mentioned thesis, solemnly declare that no portion of our thesis has been plagiarized and any material used in the thesis from other sources is properly referenced.

We further understand that if we are found guilty of any form of plagiarism in the thesis work even after graduation, the University reserves the right to revoke our BS degree.

Shahzaib Ansar Signature: ________________

Sakhawat Ali Signature: ________________

Rafay Ather Signature: ________________

____

Verified by Plagiarism Cell Officer
Dated:

Certificate of Approval

The Department of Computer Science, National University of Computer and Emerging Sciences, accepts this thesis titled *Audio to Guitar Tablature Conversion*, submitted by Shahzaib Ansar (20P-0473), Sakhawat Ali (20P-0478), and Rafay Ather (20P-0457), in its current form, and it is satisfying the dissertation requirements for the award of Bachelors Degree in Software Engineering.

Supervisor

Ms. Sara Rehmat Signature: ________________

____

Mr. Haroon Zafar

FYP Coordinator
National University of Computer and Emerging Sciences, Peshawar

____

Mr. Fazl-e-Basit

HoD of Department of Computer Science
National University of Computer and Emerging Sciences

Acknowledgements

We would like to express our deepest gratitude to our supervisor, Ms. Sara Rehmat, for her invaluable guidance, support, and encouragement throughout the development of our Final Year Project, "Audio to Guitar Tablature Conversion" Her expertise and insights were instrumental in shaping the direction and successful completion of this project. We extend our sincere thanks to the faculty and staff of the Department of Computer Science at FAST University for providing us with the necessary resources and a conducive environment for research and development.Lastly, we would like to acknowledge the efforts and cooperation of our team members, whose dedication and hard work made this project possible.It has been an enriching and rewarding experience to work together towards a common goal.

Shahzaib Ansar

Sakhawat Ali

Rafay Ather

Abstract

This report describes the completion of a system aimed at supporting novice guitarists by enabling them to convert audio files containing guitar sounds into guitar tablatures. This project aims to assist those people who are either unable to read music or can't learn music by ear. It is capable of identifying guitar notes in a recording from trained audio processing and machine learning techniques and putting them into tablature or notes. This makes the task of practicing and repeating songs much easier for the users.

The project encompasses research on pitch estimation, source separation of audio recording and sound to score transcription with a view to providing a comprehensive and simple-to-use product. The system is intended to work with different audio files and implements some of the features such as picking out guitar sounds, identifying the pitch and transcription into printable formats. The product is implemented within a multi-tier development strategy and all-in-one testing solutions, ensuring soundness, efficiency and performance. This project straightforwardly eliminates the barrier between ears and fingers so that all self-motivated guitar players would be able to improve their musical skills with ease.

Contents

1 Introduction 1

1.1 Introduction 1

1.2 Problem Statement 2

1.3 Motivation 2

1.4 Objective 3

1.5 Project Scope 3

2 Review of Literature 4

2.1 Related Research 4

2.1.1 Source Separation of Piano Concertos Using Musically Motivated Augmentation Techniques (2024) 4

2.1.2 Continuous-Valued Pitch Contour Streaming for Polyphonic Guitar Tablature Transcription (2022) 5

2.1.3 Leveraging synthesized data for guitar tablature transcription (2021) 6

2.1.4 User guided one-shot deep model adaptation for music source separation (2021) 6

2.1.5 Towards Complete Polyphonic Music Transcription: Integrating Multi-pitch Detection And Rhythm Quantization (2018) 7

3 Project Vision 9

3.1 Problem Statement 9

3.2 Business Opportunity 9

3.3 Objectives 9

3.4 Project Scope 10

3.5 Constraints 10

3.6 Stakeholders Description 10
3.6.1 Stakeholders Summary 10
3.6.2 Key High-Level Goals and Problems of Stakeholders 11

4 Software Requirements Specifications 12
4.1 List of Features 12
4.2 Functional Requirements 13
4.3 Quality Attributes 13
4.4 Non-Functional Requirements 13
4.5 Test Plan 14
4.5.1 Introduction 14
4.5.2 Test Levels 14
4.5.3 Testing Techniques 14
4.5.4 Test Cases 15
4.6 Software Development Plan 16
4.6.1 Agile Development Methodology 16
4.6.2 Development Timeline 17
4.6.3 Resource Allocation 17
4.6.4 Risk Management 17
4.6.5 Project Deliverables 18
4.7 UI Screens 19
4.8 Diagrams 20

5 Iteration Plan 24

6 Iteration 1 27
6.1 Activity Diagram 27

7 Iteration 2 28
7.1 Architecture Diagram 28

8 Iteration 3 29
8.1 Domain Model/ Class Diagram 29

8.2 Component Diagram 29
8.3 Layer Diagram 29
8.4 Structure Chart 29
8.5 Flow Diagram 30
8.6 Data Flow Diagram (DFD) 30
8.7 Data Dictionary 30
8.8 Activity Diagram 30
8.9 Network Automata/ Graphs or State Machine 30
8.10 Call Graph or Sequence Diagram 30
8.11 Interaction Overview Diagram 30
8.12 Schema Design/ ER Diagram 30
8.13 Data Structure Design 30
8.14 Algorithm Design 30
8.15 Development Phase 31
8.15.1 Unit Test 32
8.15.2 Suites or Test Cases 32
8.16 Maintainable Phase 32
8.16.1 CI/ CD 32
8.16.2 Deployment Diagram 32
8.16.3 System-Level Test Suites, Test Cases 32
8.16.4 SVN or GitHub (Optional) 32
8.16.5 Configuration/ Setup and Tool Manual (Optional) 32

9 Iteration 4 **33**
9.1 Domain Model/ Class Diagram 33
9.2 Component Diagram 33
9.3 Layer Diagram 33
9.4 Structure Chart 33
9.5 Flow Diagram 34
9.6 Data Flow Diagram (DFD) 34
9.7 Data Dictionary 34

9.8 Activity Diagram 34
9.9 Network Automata/ Graphs or State Machine 34
9.10 Call Graph or Sequence Diagram 34
9.11 Interaction Overview Diagram 34
9.12 Schema Design/ ER Diagram 34
9.13 Data Structure Design 34
9.14 Algorithm Design 34
9.15 Development Phase 35
9.15.1 Unit Test 36
9.15.2 Suites or Test Cases 36
9.16 Maintainable Phase 36
9.16.1 CI/ CD 36
9.16.2 Deployment Diagram 36
9.16.3 System-Level Test Suites, Test Cases 36
9.16.4 SVN or GitHub (Optional) 36
9.16.5 Configuration/ Setup and Tool Manual (Optional) 36

10 Implementation Details 37
10.1 System Workflow 37
10.2 Preprocessing and Source Separation 37
10.3 Transcription System 38
10.4 Notation Generation 38
10.5 Machine Learning Enhancements 38
10.6 User Interaction and System Integration 39

11 User Manual 40
11.1 11.1 Introduction 40
11.2 Getting Started 40

12 Conclusions and Future Work 42
12.1 Conclusions 42
12.2 Future Work 43

List of Figures

4.1 Upload a mp3 file 19
4.2 Show the progess of MP3 file 19
4.3 Flow Chart 20
4.4 Audio Processing Activity Diagram 21
4.5 Use Case 22
4.6 Sequence 23
4.7 Architecture 23

6.1 Audio Processing Activity Diagram 27

7.1 System Architecture Diagram 28

List of Tables

Chapter 1

Introduction

1.1 Introduction

This project aims to create an accessible tool that helps beginner and non-skilled guitarists play music from audio clips without requiring advanced music-reading skills. Many individuals who enjoy guitar music struggle to learn songs by ear, especially if they cannot read musical notation or tablature. This problem limits their ability to engage with and learn their favorite guitar pieces effectively.

Previous research in audio processing and machine learning, such as instrument detection and pitch extraction, has laid the foundation for identifying and isolating individual instrument sounds within complex audio clips [1].

Building on these studies, our approach combines audio processing and machine learning to develop a system that can detect guitar sounds from an audio clip and automatically convert them into readable musical notes or tablature. The system's success will be measured by its accuracy in detecting guitar sounds and generating notes that are easily understandable for beginner guitarists. This research hopes to bridge the gap between listening and learning, offering a tool that brings guitar practice closer to a broader audience.

- To develop a tool that aids beginner guitarists in playing music by detecting and converting guitar sounds from audio clips into notes.

- Non-skilled guitarists often struggle to play by ear, facing challenges in replicating their favorite music without music-reading skills.

- Research in audio processing and machine learning has advanced, enabling the identification and analysis of various instrument sounds, yet tools specifically aiding guitarists remain limited.

- By combining audio feature extraction and a pre-trained model for guitar sound detection, this project will convert detected sounds into playable notes or tablature.

- The system will be considered successful if it can accurately detect guitar sounds in audio clips and produce clear notes for users to understand and play.

1.2 Problem Statement

Many aspiring guitarists struggle to replicate music by ear, particularly when they lack formal music-reading skills. For beginners, interpreting and playing guitar from audio clips is challenging without a structured way to convert sounds into notes or tablature. This project addresses the need for an accessible tool to help non-skilled guitarists practice and play guitar music more easily.

1.3 Motivation

The motivation behind this project is to make guitar learning more accessible. Traditional methods often require advanced musical knowledge, creating a barrier for beginners. By developing a tool that converts guitar sounds into readable notes, we aim to simplify the learning process, allowing enthusiasts to practice and play without needing extensive musical background.

1.4 Objective

The objective of this project is to create a system that identifies guitar sounds in audio clips and converts them into musical notes or tablature. This system will empower beginners by providing an easy way to interpret and play guitar music, bridging the gap between listening to and performing guitar songs without advanced skills.

1.5 Project Scope

This project will develop a system for analyzing audio clips, detecting guitar sounds, and converting them into musical notes or tablature.

The scope includes research on audio processing techniques and the use of machine learning to accurately identify guitar-specific frequencies.Additionally, the project will involve developing a user-friendly interface for uploading audio clips and displaying the generated notes.

The focus will be on providing non-skilled guitarists an accessible tool for learning and playing guitar music, with evaluation criteria centered around the accuracy and ease of use of the generated notes.

Chapter 2

Review of Literature

2.1 Related Research

This section reviews key research papers that form the foundation of this project. Each study has contributed significantly to advancements in audio processing, pitch detection, and music transcription.

2.1.1 Source Separation of Piano Concertos Using Musically Motivated Augmentation Techniques (2024)

In 2022, researchers Yigitcan Özer and Meinard Müller proposed a deep learning-based approach to tackle the complex problem of separating piano and orchestral tracks from piano concertos. The study introduced musically motivated data augmentation techniques that utilized the harmonic, rhythmic, and structural elements of piano concertos to enhance model training. These augmentation methods were integrated into hybrid deep learning models, which combined spectrogram- and waveform-based approaches for music source separation (MSS).

The researchers evaluated their models using the Piano Concerto Dataset (PCD), a dataset featuring recordings from various acoustic environments and performers. Results demonstrated that hybrid models trained with musically inspired augmentations outperformed traditional models, achieving superior separation quality. Performance was measured using metrics like Signal-to-Distortion Ratio (SDR) and 2f-scores, alongside perceptual evaluations using the MUSHRA framework.

Despite its strengths, the study's limitations include a focus on classical piano concertos, potentially limiting the generalizability of the results to other music genres. Nonetheless, this research represents a significant advancement in music source separation, offering valuable techniques for applications like transcription, karaoke systems, and music production.

2.1.2 Continuous-Valued Pitch Contour Streaming for Polyphonic Guitar Tablature Transcription (2022)

In 2023, researchers addressed a critical gap in Automatic Music Transcription (AMT) and Multi-Pitch Estimation (MPE) by introducing a novel formulation for Guitar Tablature Transcription (GTT). While AMT has gained significant attention in recent years, the accurate estimation of continuous-valued pitch contours remains a challenge, particularly for applications like GTT, which require detailed representation of pitch modulation techniques. Traditional AMT approaches typically quantize pitch information to the Western music scale, limiting their ability to capture nuanced pitch variations.

The study proposed a GTT formulation that estimates continuous-valued pitch contours while grouping them by their string and fret of origin. This approach addresses the limitations of conventional AMT models, which either neglect pitch modulation or increase model complexity to reduce quantization. Experiments demonstrated that the proposed method significantly enhances the resolution of MPE and produces tablature estimation results competitive with existing baseline models.

This research represents a significant advancement in both AMT and GTT, offering a framework to better handle pitch modulation and continuous pitch contours. It holds potential for applications in music transcription and education, particularly for instruments like the guitar.

2.1.3 Leveraging synthesized data for guitar tablature transcription (2021)

In 2023, researchers proposed a novel approach to improve Guitar Tablature Transcription (GTT), a critical task in music education, composition, and entertainment. GTT involves converting audio into guitar tablature, which not only captures the musical content but also includes the instrument's specific implementation and ornamentation. Existing datasets for GTT are limited in size and diversity, leading to overfitting and poor generalization to out-of-domain data.

To address these challenges, the study introduced SynthTab, a dataset synthesized using commercial acoustic and electric guitar plugins. SynthTab is derived from DadaGP, a large, richly annotated collection of symbolic tablature. The dataset encompasses multiple guitars, styles, and techniques, producing audio that adheres faithfully to the original fingerings and tablature specifications.

Experiments demonstrated that pre-training a baseline GTT model on SynthTab improved transcription performance when fine-tuned and tested on individual datasets. Moreover, cross-dataset experiments highlighted that pre-training significantly reduced overfitting, allowing models to generalize better across diverse datasets. This research represents a major advancement in GTT, providing a scalable methodology for data synthesis and improved transcription performance.

2.1.4 User guided one-shot deep model adaptation for music source separation (2021)

In 2021, researchers tackled the challenge of isolating individual instruments from mixed musical pieces, a task known as music source separation. While state-of-the-art models achieve reasonable results, their ability to generalize to unseen test data remains limited. This study introduced a novel approach that leverages user-provided temporal segmentation, indicating when each instrument is active, to fine-tune pre-trained deep models for improved source separation.

The proposed method adapts the generic model specifically to the target song instance, a paradigm referred to as user-driven one-shot deep model adaptation. By focusing the model's adaptation on the observed signal, the researchers demonstrated significant performance improvements, particularly for underrepresented instruments in the training data.

This approach highlights the potential for integrating user input to enhance the capabilities of deep learning models in source separation, addressing the limitations of generalization and improving the accuracy of instrument isolation in complex musical mixtures. The results indicate promising directions for future research in adaptive music source separation.

2.1.5 Towards Complete Polyphonic Music Transcription: Integrating Multi-pitch Detection And Rhythm Quantization (2018)

In 2018, researchers proposed an integrated system for polyphonic music transcription that aims to convert complex audio recordings into human-readable musical scores. The study addressed key challenges in multi-pitch detection and rhythm quantization by refining existing methodologies and introducing innovative techniques. Specifically, multi-pitch detection was improved with better note tracking, handling repeated notes, and optimizing onset times. Rhythm quantization utilized a noisy metrical hidden Markov model (HMM) to eliminate extra notes and enhance timing accuracy.

The system was evaluated using classical piano datasets, demonstrating significant advancements over traditional methods and commercial software like MuseScore 2 and Finale 2014. The results highlighted its superior performance in reducing pitch errors and improving rhythm alignment, particularly for polyphonic recordings. The outputs were generated in versatile formats, including MIDI, MusicXML, and PDF, making the system practical for a wide range of applications.

Despite its advancements, the study's limitations include challenges with musically complex configurations and handling pitch errors in dense polyphony. Nevertheless, this research represents a significant step forward in polyphonic music transcription, offering valuable insights for applications in music education, automated transcription, and digital music production.

Chapter 3

Project Vision

3.1 Problem Statement

Musicians often struggle to transcribe guitar parts from audio recordings, making it difficult to learn and play their favorite songs. Existing methods for converting audio to notation are often complex and time-consuming

3.2 Business Opportunity

The growing popularity of online music learning platforms highlights a demand for tools that simplify the music learning process. This project provides an opportunity to create a scalable solution targeting beginner and intermediate guitarists who need an easy-to-use transcription tool. By combining advanced audio processing with a user-friendly interface, the system fills a significant gap in the market.

3.3 Objectives

Audio to Guitar Note Tablature Converter is to create an efficient tool that automatically extracts guitar sounds from audio clips and converts them into accurate tablature, facilitating easier learning for musicians

3.4 Project Scope

The project focuses on:

- Supporting audio input in MP3 and WAV formats.
- Handling monophonic and polyphonic guitar tracks.
- Generating guitar tablature in a readable and printable PDF format.
- Providing support for files up to 100 MB in size.

The system does not aim to support real-time transcription or transcription for instruments other than the guitar.

3.5 Constraints

- The system processes audio files up to 100 MB with a maximum duration of 10 minutes.
- Supported audio formats are limited to MP3 and WAV.
- Real-time transcription is not supported; processing may take up to 5 minutes.
- The system requires a reliable internet connection for advanced processing modules.

3.6 Stakeholders Description

3.6.1 Stakeholders Summary

The key stakeholders in this project include:

- **Guitarists**: Primary users who require an efficient tool for transcription.

- **Music Educators**: Secondary users who can utilize the system for teaching purposes.
- **Developers**: Individuals interested in expanding the tool's capabilities for other instruments.
- **Amateur Musicians**: Hobbyists looking for a simple way to play and learn guitar music.

3.6.2 Key High-Level Goals and Problems of Stakeholders

- **Guitarists**:
  - **Goals**: Simplify the process of learning and playing songs by generating accurate tablature.
  - **Problems**: Lack of reliable tools to isolate and transcribe guitar sounds from mixed audio.
- **Music Educators**:
  - **Goals**: Use the tool to teach students how to play guitar using generated tablature.
  - **Problems**: Manual transcription takes time and reduces teaching efficiency.
- **Amateur Musicians**:
  - **Goals**: Easily replicate songs from audio without needing advanced skills.
  - **Problems**: Difficulty interpreting musical notation and identifying guitar sounds.

Chapter 4

Software Requirements Specifications

4.1 List of Features

The system provides the following key features:

- **Audio Upload**: Users can upload audio files in MP3 and WAV formats.
- **Guitar Sound Isolation**: The system separates guitar sounds from other instruments.
- **Pitch Detection**: It detects the pitch and timing of the guitar notes with high accuracy.
- **Tablature Generation**: Detected notes are converted into guitar tablature.
- **PDF Export**: The generated tablature is saved as a PDF for easy sharing and printing.
- **Real-Time Updates**: The system provides progress updates during audio processing.
- **Error Handling**: Notifications for unsupported file formats or when no guitar is detected.

4.2 Functional Requirements

- Users must be able to upload audio files in MP3 or WAV format.
- The system must isolate guitar audio from mixed soundtracks.
- It should accurately detect pitch and timing of guitar notes.
- Generate readable tablature in a printable PDF format.
- Notify users of processing progress and errors, if any.

4.3 Quality Attributes

- **Performance**: Process 2-minute audio files within 5 minutes.
- **Accuracy**: Achieve at least 95% accuracy in pitch detection.
- **Usability**: Provide an intuitive interface for beginners.
- **Scalability**: Handle audio files up to 10 MB.
- **Security**: Securely process and delete uploaded files post-processing.

4.4 Non-Functional Requirements

- The system must be accessible via web browsers and support mobile devices.
- Ensure compatibility with MP3 and WAV audio file formats.
- Provide feedback to users within 10 seconds of an action (e.g., upload confirmation).
- Maintain reliable performance with minimal downtime.

4.5 Test Plan

4.5.1 Introduction

- **Objective**: To validate the functionality, accuracy, and reliability of the *Audio to Guitar Tablature Conversion* system.

- **Scope**: The testing will cover audio file uploads, guitar sound isolation, pitch detection accuracy, tablature generation, PDF export functionality, user interface usability, security, and overall system performance.

4.5.2 Test Levels

- **Unit Testing**: Test individual components, such as audio upload validation, sound isolation algorithms, and pitch detection, to ensure they function correctly in isolation.

- **Integration Testing**: Verify the interaction between modules, including sound isolation, pitch detection, and tablature generation, ensuring seamless communication and data flow.

- **System Testing**: Evaluate the entire system's functionality, including file upload, processing accuracy, and PDF generation, along with usability and security features.

- **Acceptance Testing**: Conduct user acceptance testing with real users to validate the system's usability, accuracy of tablature, and overall user satisfaction.

4.5.3 Testing Techniques

- **Manual Testing**: Testers will interact with the application as end-users, evaluating user interfaces, usability, and overall system behavior.

- **Automated Testing**: Write automated test scripts to perform repetitive and regression tests, ensuring that new changes do not break existing functionalities.
- **Exploratory Testing**: Testers will dynamically explore the application, identifying potential issues and usability problems not covered by formal test cases.
- **Performance Testing**: Use load testing tools to simulate various user loads, ensuring the system performs well under different levels of user activity.
- **Security Testing**: Conduct penetration testing to identify vulnerabilities in the system, ensuring that user data and the application are secure against potential attacks.

4.5.4 Test Cases

- **Test Case 1: Audio File Upload Validation**
  - **Inputs**: MP3 and WAV files of varying sizes (e.g., 5 MB, 7 MB, and 10 MB).
  - **Expected Output**: The system accepts valid files and displays errors for unsupported formats or excessively large files.
- **Test Case 2: Guitar Sound Isolation Accuracy**
  - **Inputs**: Mixed audio files containing guitar and other instruments.
  - **Expected Output**: The system successfully isolates guitar sounds without significant distortion or noise.
- **Test Case 3: Pitch Detection Accuracy**
  - **Inputs**: Guitar audio files with known pitch patterns.
  - **Expected Output**: The system detects and displays pitches with at least 95% accuracy.
- **Test Case 4: Tablature Generation and Export**
  - **Inputs**: Isolated guitar audio with various complexity levels (monophonic and polyphonic tracks).

  - **Expected Output**: The system generates accurate tablature and exports it as a correctly formatted PDF.

- **Test Case 5: Usability Testing**
  - **Inputs**: Actual users interacting with the application.
  - **Expected Output**: Users can easily upload files, view processing progress, and download the tablature without difficulty.

- **Test Case 6: Performance Testing**
  - **Inputs**: Audio files of varying sizes and simultaneous user requests.
  - **Expected Output**: The system processes files within 5 minutes and remains responsive under heavy load.

- **Test Case 7: Security Testing**
  - **Inputs**: Simulated unauthorized access attempts and security attacks (e.g., SQL injection, file tampering).
  - **Expected Output**: The system resists unauthorized access, secures user data, and maintains system integrity.

- **Test Case 8: Error Handling**
  - **Inputs**: Unsupported audio formats and audio files without guitar sounds.
  - **Expected Output**: The system displays appropriate error messages and does not crash.

4.6 Software Development Plan

4.6.1 Agile Development Methodology

- **Sprints**: The project will be divided into 2-week sprints, enabling continuous development, feedback, and adjustments.

- **Daily Standups**: Regular team meetings will be held to share updates on progress, discuss challenges, and plan future tasks.
- **Iterative Development**: The system will be developed incrementally, with functional releases at the end of each sprint.
- **User Involvement**: Stakeholders and end-users will provide feedback throughout the process, ensuring the system meets their needs and expectations.

4.6.2 Development Timeline

- **Sprint 1-2 (2 months)**: Initial project setup, UI/UX design, and development of audio upload and validation features.
- **Sprint 3-6 (4 months)**: Implementation of core modules, including guitar sound isolation, pitch detection, and tablature generation.
- **Sprint 7-8 (2 months)**: Final system integration, thorough testing, bug fixing, and deployment preparations.

4.6.3 Resource Allocation

- **Development Team**: Backend developers, UI/UX designers, audio processing engineers, and QA testers.
- **Tools and Technologies**: Python (for audio processing), TensorFlow (for pitch detection), and a web-based framework for the user interface.
- **Infrastructure**: Cloud-based servers for storage and processing, with secure data handling.

4.6.4 Risk Management

- **Identify Risks**: Potential risks include delays in module integration, scope changes, and performance issues with large audio files.

- **Mitigation Strategies**:
  - Conduct regular sprint reviews and testing to identify and address issues early.
  - Maintain flexible timelines and contingency plans for unexpected challenges.
  - Ensure clear communication among team members and stakeholders to manage scope changes effectively.

4.6.5 Project Deliverables

- **Functional Application**: A web-based system capable of processing audio files, generating guitar tablature, and exporting results as PDFs.
- **Technical Documentation**: Comprehensive documentation covering system architecture, APIs, data flow, and processing algorithms.
- **User Manuals**: Guides for end-users explaining how to upload files, view progress, and download tablature results.
- **Testing Reports**: Detailed results of unit, integration, and system testing.

4.7 UI Screens

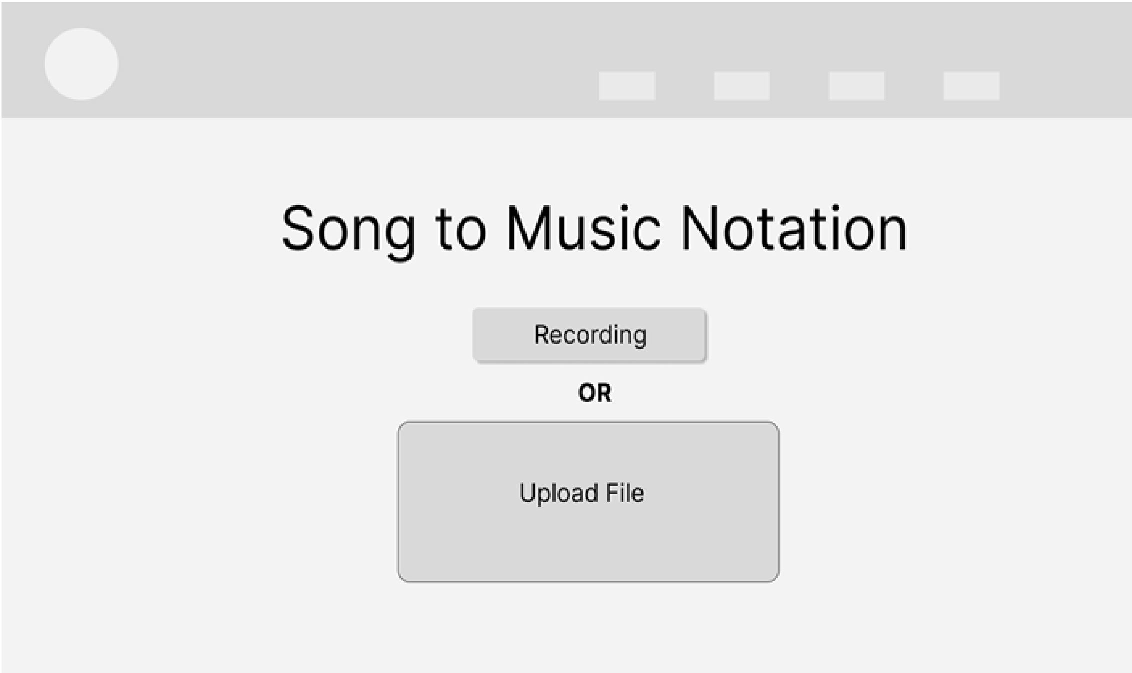


Figure 4.1: Upload a mp3 file

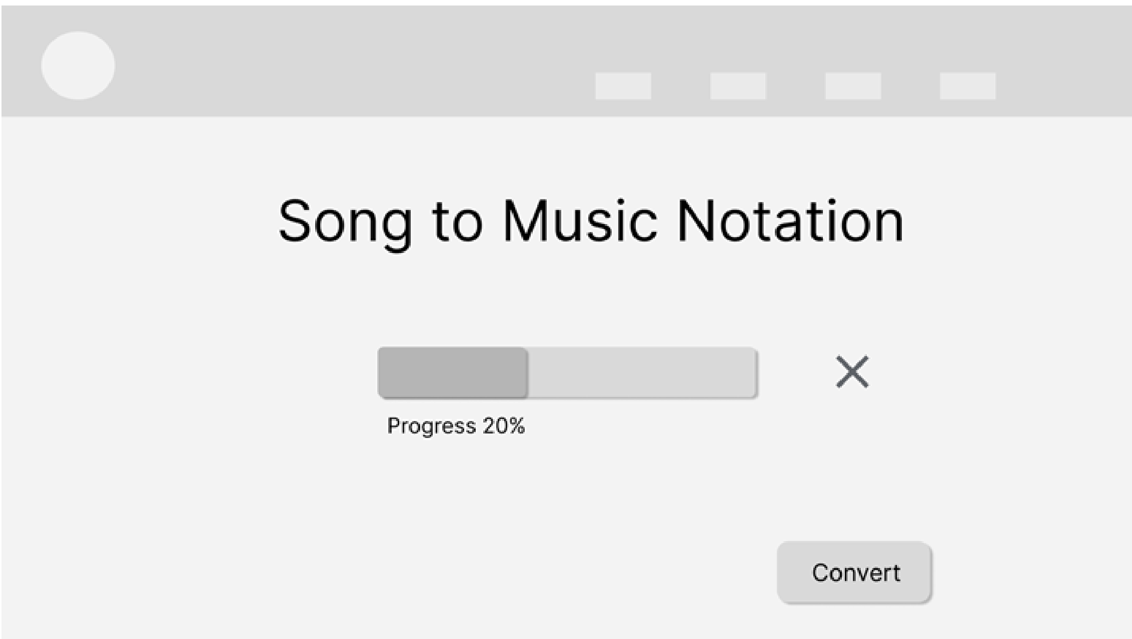


Figure 4.2: Show the progess of MP3 file

4.8 Diagrams

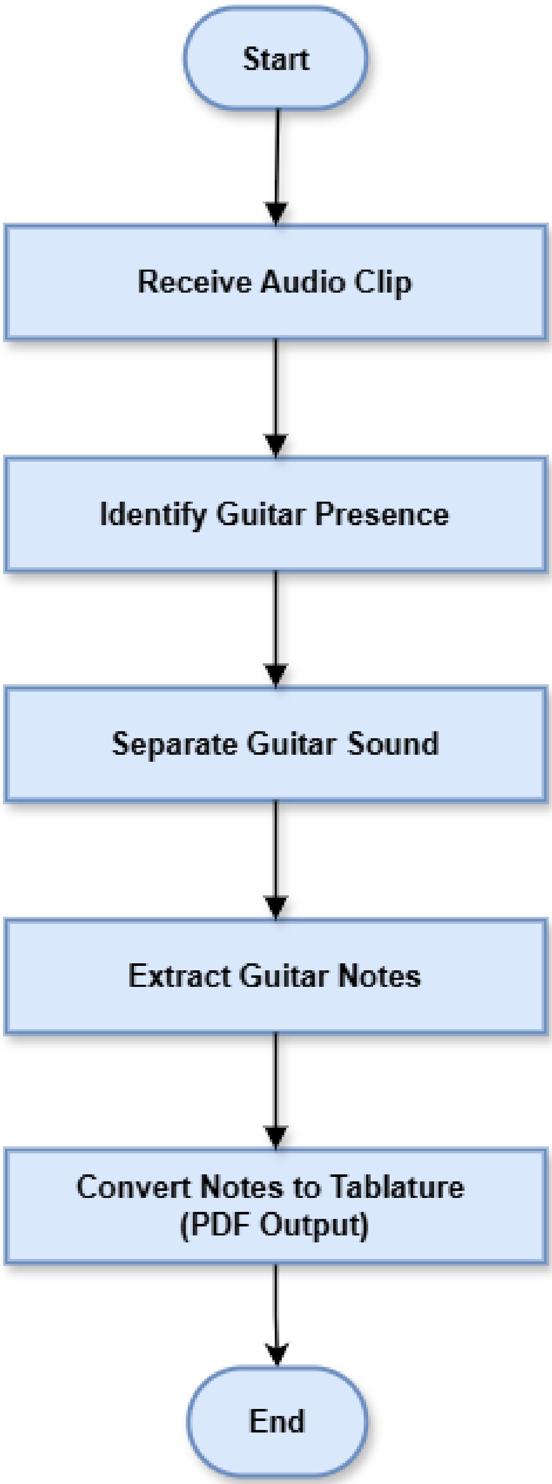


Figure 4.3: Flow Chart

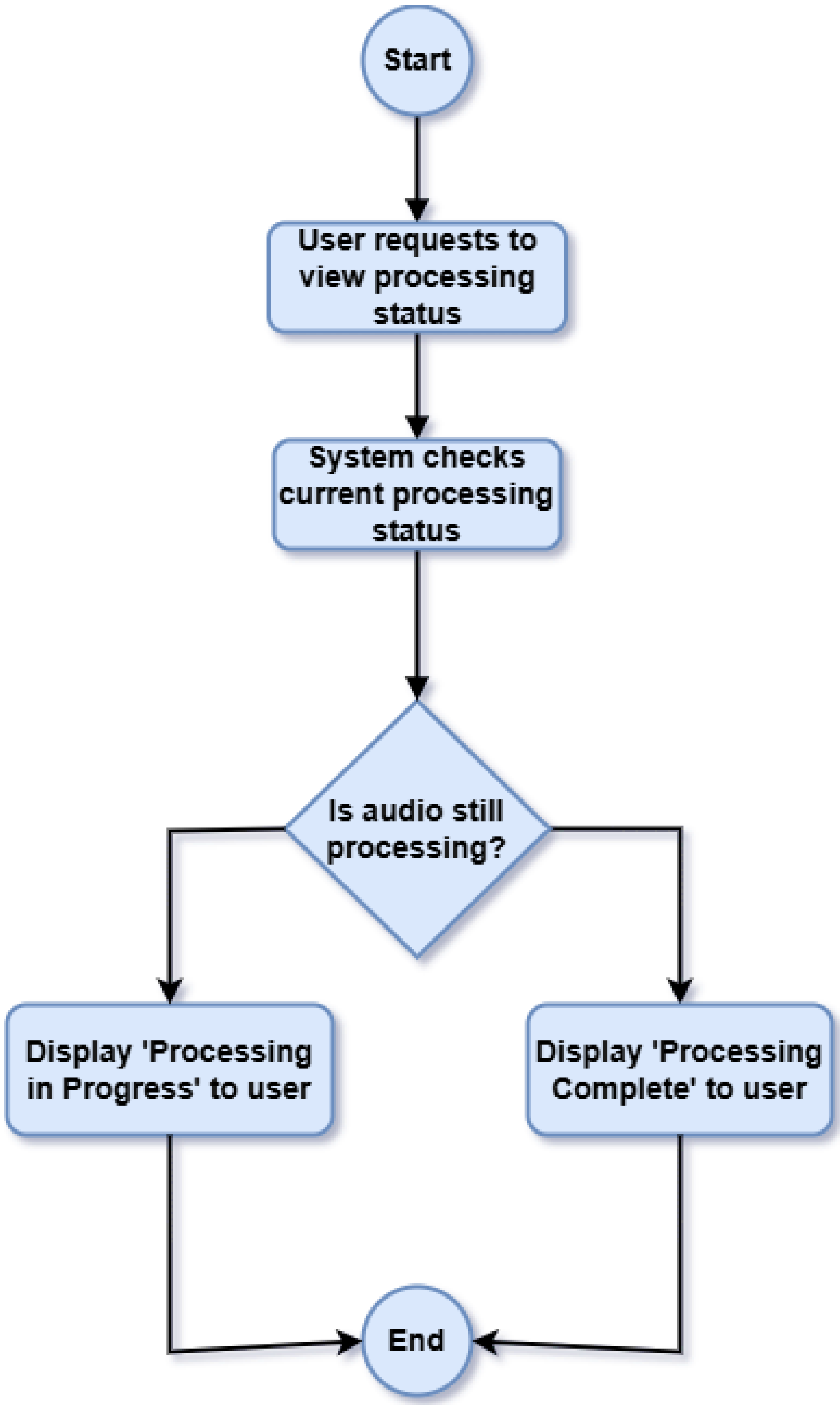


Figure 4.4: Audio Processing Activity Diagram

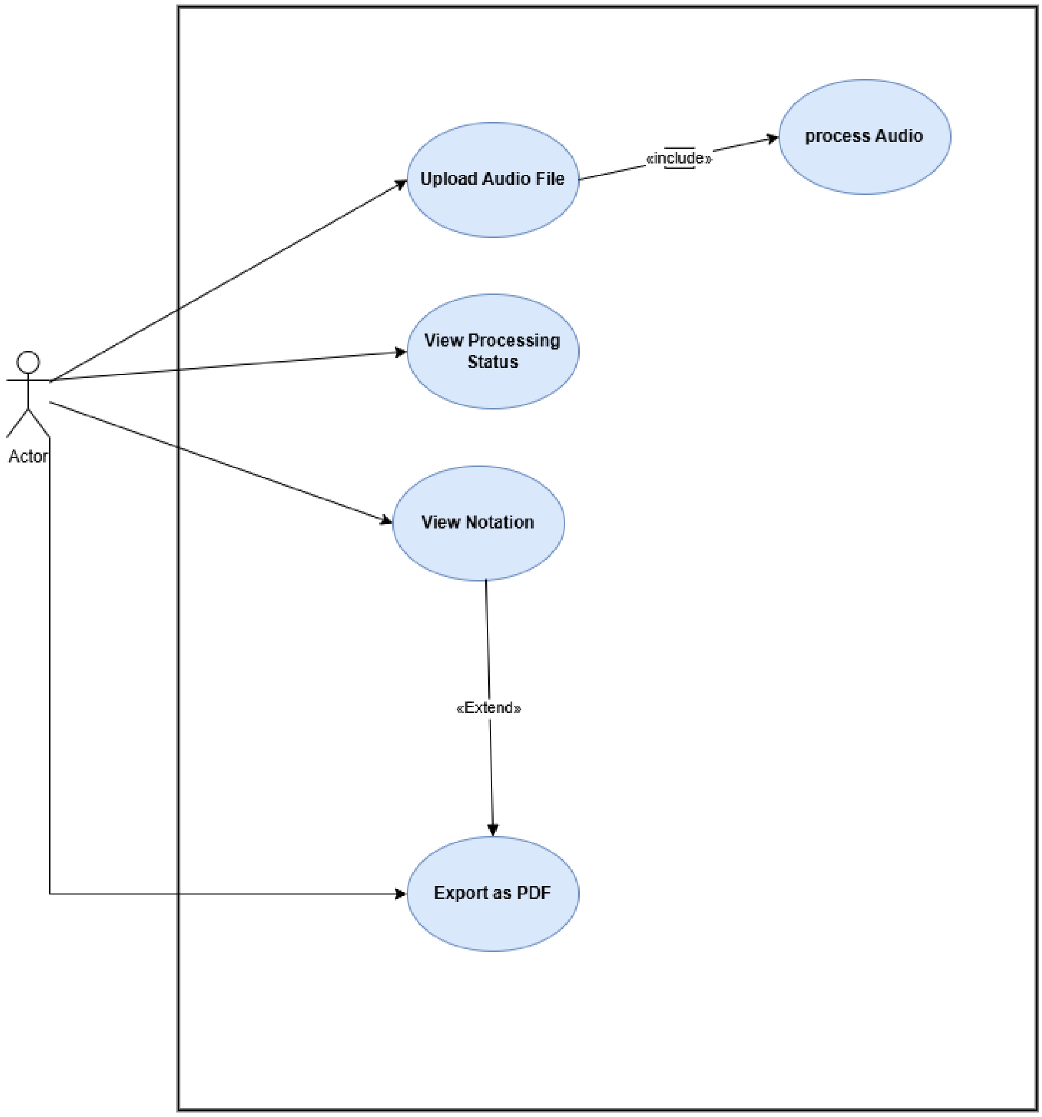


Figure 4.5: Use Case

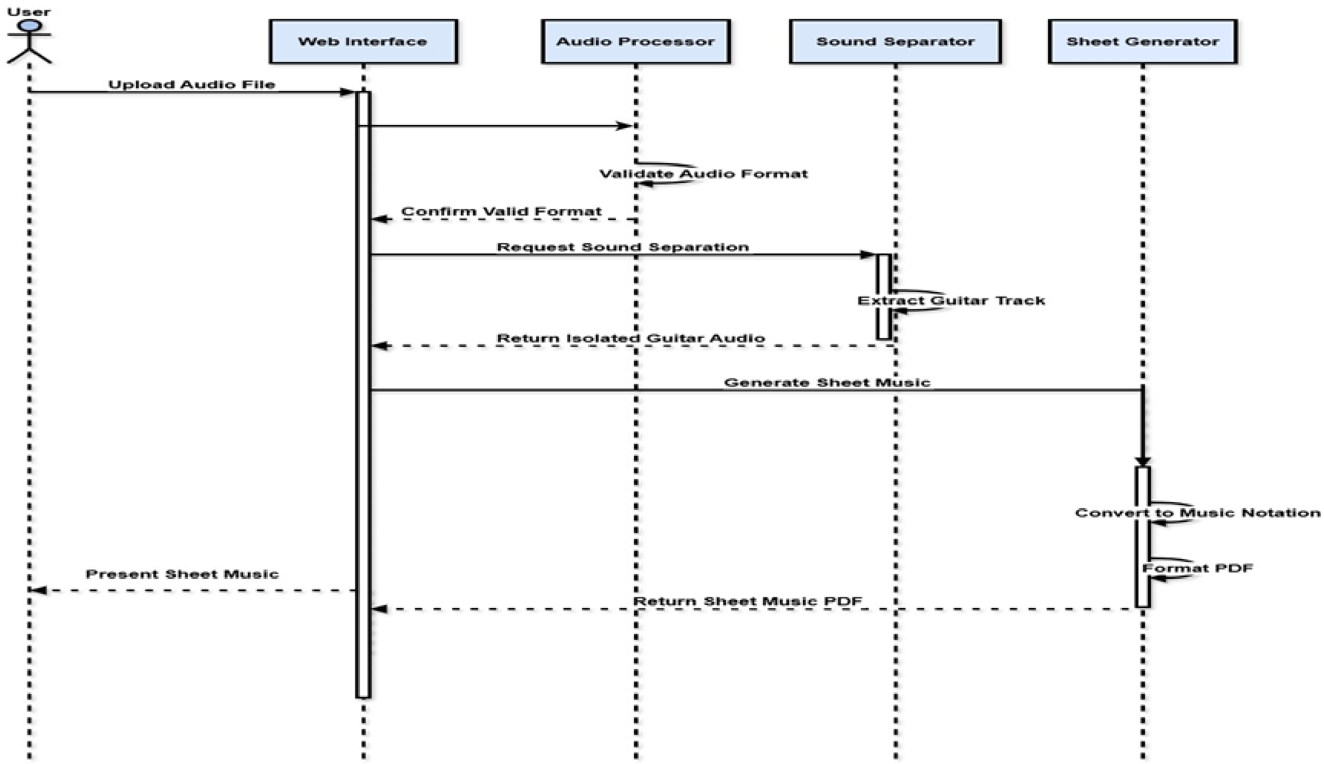


Figure 4.6: Sequence

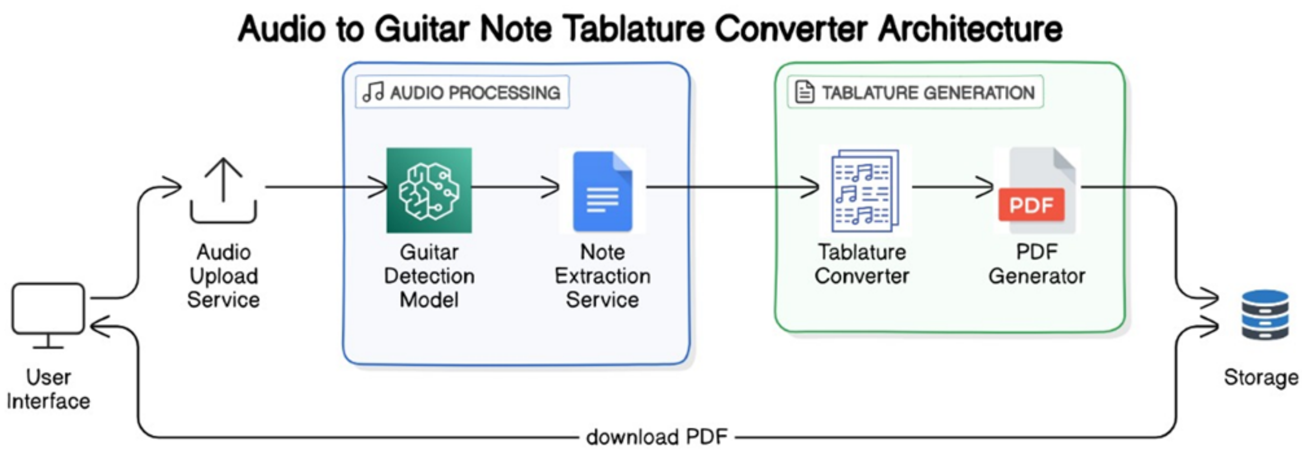


Figure 4.7: Architecture

Chapter 5

Iteration Plan

- **Midterm FYP 1**

  For Midterm FYP 1, our primary objective centered around defining the core functionalities of our *Guitar Transcription System*. During this phase, we developed three essential diagrams—Activity Diagram, Use Case Diagram, and Sequence Diagram—to establish a clear understanding of the system's workflow. Our emphasis was on the conceptualization and refinement of these aspects, without delving into the actual implementation.Additionally, we focused on designing UI/UX wireframes to outline the visual structure and user interaction flow of our application. This step ensured a well-structured foundation for the user experience before transitioning into development.Parallel to these activities, we prepared a comprehensive report that aligns with the specified project requirements. This document encapsulates the project scope, methodology, and strategic decisions taken during the initial phases of development. Our goal was to establish a solid conceptual groundwork that would facilitate smooth implementation in subsequent phases.

- **Final FYP 1**

  In Final FYP 1, we moved from planning to actually building our Guitar Transcription System. We started by creating an architectural diagram, which showed how different parts of the system would work together. This acted as a guide for developing our project.After finalizing the design, we began the implementation process.

One of our biggest achievements was setting up the preprocessing and source separation pipeline. We used FFmpeg to prepare the audio and Demucs to separate the guitar track from the rest of the music. This step was essential for making sure the transcription was as accurate as possible.Next, we worked on the transcription model, using Basic Pitch to convert the isolated guitar sound into MIDI notes. This helped us accurately detect notes for tablature and sheet music generation. Finally, we used Verovio to turn the MIDI data into MusicXML and PDF formats, so users could easily access the results in standard notation.At the same time, we started exploring machine learning models, analyzing datasets, and improving the accuracy of our transcription. To present our progress effectively, we also designed a poster that summarized the key aspects of our project in a clear and simple way.This phase was a big step forward in developing our system, bringing together advanced audio processing and transcription technologies to make our project more effective and reliable.

- **Midterm FYP 2**

  In the Midterm of FYP 2, our focus will shift towards the comprehensive testing of our Guitar Transcription System. This phase involves a meticulous evaluation to ensure that the functionality aligns with our intended design and operational workflow. We will rigorously assess the accuracy of the audio transcription pipeline, ensuring that the extracted guitar notes and generated sheet music meet our predefined standards.Beyond basic validation, our objective is to enhance the precision of the system by refining the machine learning model and improving note detection accuracy. This iterative process of testing and optimization is crucial in achieving high transcription fidelity and ensuring a reliable user experience.By continuously refining our system, we aim to establish a robust foundation for the final implementation, guaranteeing that the system performs optimally in real-world scenarios.

- **Final FYP 2**

  In the Final FYP 2 stage, our main focus will be on deploying our Guitar Transcription System. This means making sure the system is fully working and ready to process guitar recordings into tablature and sheet music.We will also perform detailed testing to check how well the system works in different conditions. This includes making sure that the audio transcription, MIDI conversion, and sheet music generation are accurate. If we find any issues, we will fix them and improve the system before finalizing it.Our goal in Final FYP 2 is to complete the project successfully by delivering a well-functioning system. This phase ensures that our system is reliable and ready to be used, combining advanced audio processing and machine learning to achieve high-quality transcription results.

Chapter 6

Iteration 1

6.1 Activity Diagram

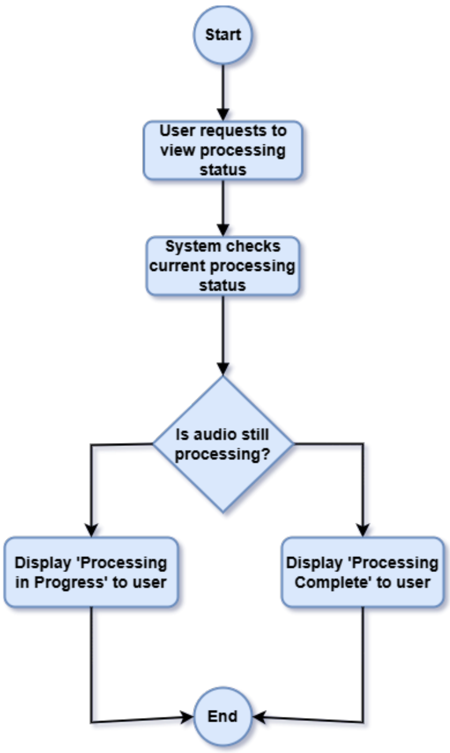


Figure 6.1: Audio Processing Activity Diagram

Chapter 7

Iteration 2

7.1 Architecture Diagram

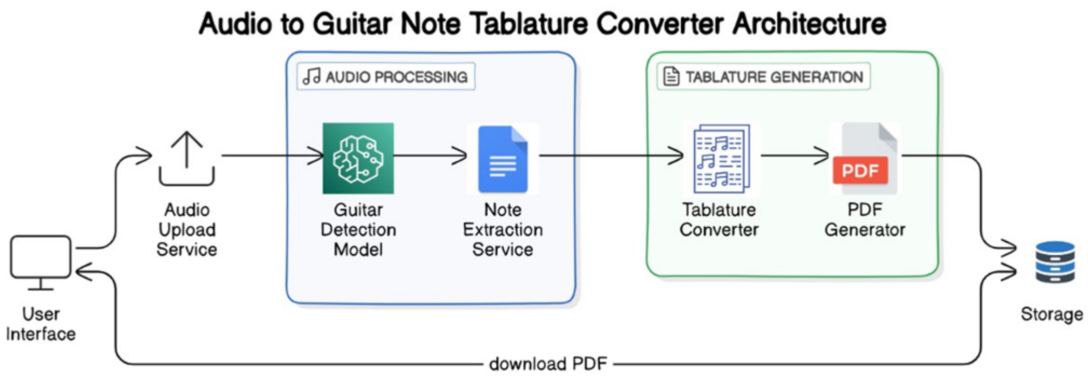


Figure 7.1: System Architecture Diagram

Chapter 8

Iteration 3

The first iteration is expected to be completed by the midterm of the FYP-2. This chapter will have some of the artifacts based on system design. The requirements analysis section is same for all the systems while the design may vary. There may have two types of designs the structural design or . First section is for the structural design.

structural design

8.1 Domain Model/ Class Diagram

8.2 Component Diagram

8.3 Layer Diagram

8.4 Structure Chart

Behavior Design

8.5 Flow Diagram

8.6 Data Flow Diagram (DFD)

8.7 Data Dictionary

8.8 Activity Diagram

8.9 Network Automata/ Graphs or State Machine

8.10 Call Graph or Sequence Diagram

8.11 Interaction Overview Diagram

For all above designs

8.12 Schema Design/ ER Diagram

8.13 Data Structure Design

Any information

8.14 Algorithm Design

Any information

8.15 Development Phase

Comments, Naming Conventions, Static Analysis of Code, etc.,

8.15.1 Unit Test

8.15.2 Suites or Test Cases

8.16 Maintainable Phase

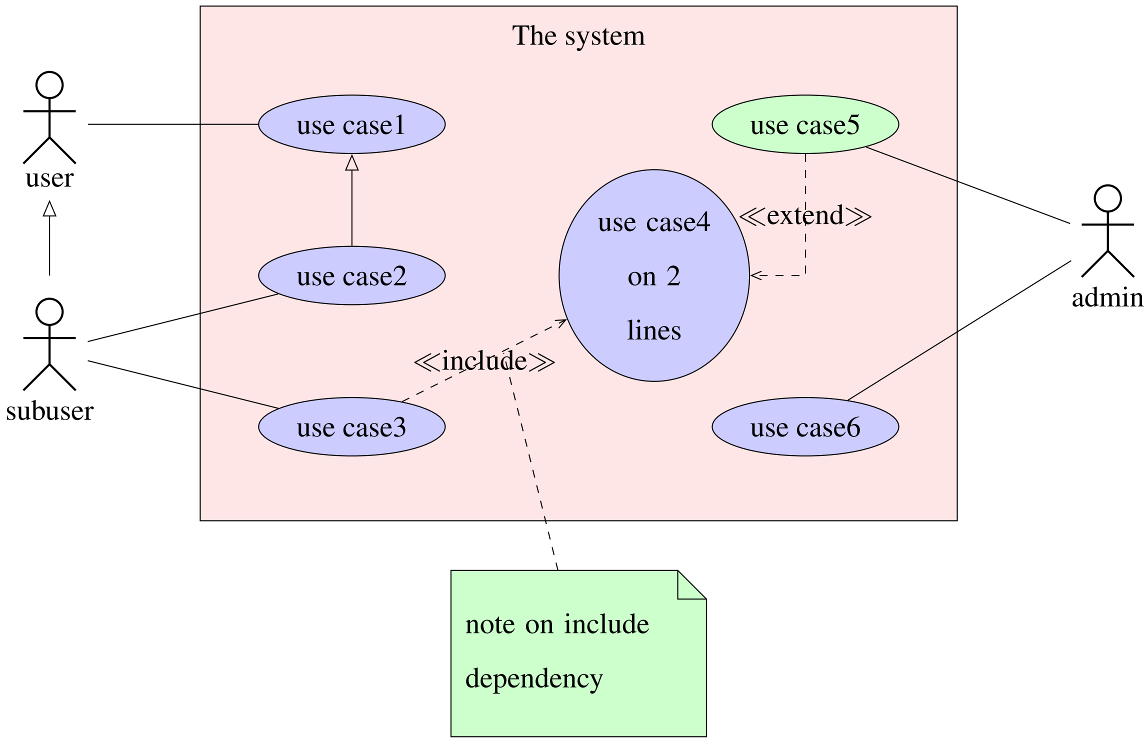
8.16.1 CI/ CD

8.16.2 Deployment Diagram

8.16.3 System-Level Test Suites, Test Cases

8.16.4 SVN or GitHub (Optional)

8.16.5 Configuration/ Setup and Tool Manual (Optional)



Chapter 9

Iteration 4

The first iteration is expected to be completed by the final of the FYP-2. This chapter will have some of the artifacts based on system design. The requirements analysis section is same for all the systems while the design may vary. There may have two types of designs the structural design or . First section is for the structural design.

structural design

9.1 Domain Model/ Class Diagram

9.2 Component Diagram

9.3 Layer Diagram

9.4 Structure Chart

Behavior Design

9.5 Flow Diagram

9.6 Data Flow Diagram (DFD)

9.7 Data Dictionary

9.8 Activity Diagram

9.9 Network Automata/ Graphs or State Machine

9.10 Call Graph or Sequence Diagram

9.11 Interaction Overview Diagram

For all above designs

9.12 Schema Design/ ER Diagram

9.13 Data Structure Design

Any information

9.14 Algorithm Design

Any information

9.15 Development Phase

Comments, Naming Conventions, Static Analysis of Code, etc.,

9.15.1 Unit Test

9.15.2 Suites or Test Cases

9.16 Maintainable Phase

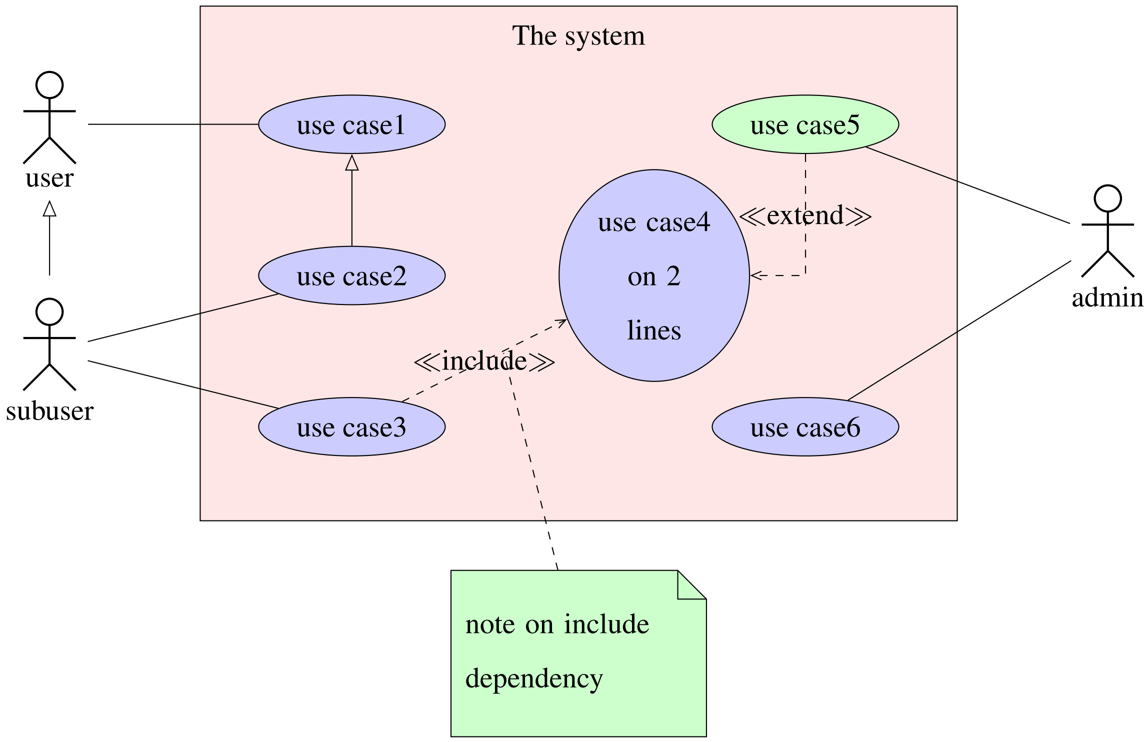
9.16.1 CI/ CD

9.16.2 Deployment Diagram

9.16.3 System-Level Test Suites, Test Cases

9.16.4 SVN or GitHub (Optional)

9.16.5 Configuration/ Setup and Tool Manual (Optional)



Chapter 10

Implementation Details

10.1 System Workflow

This project utilizes state-of-the-art machine learning and signal processing techniques to transcribe guitar recordings into sheet music. The system follows a structured pipeline to ensure accurate and efficient transcription.

10.2 Preprocessing and Source Separation

To prepare audio for transcription, the system applies a preprocessing pipeline:

- **Audio Conversion:** The uploaded audio files are converted to a standardized format (**WAV**) using **FFmpeg** for consistency in processing.

- **Source Separation:** The **Demucs** deep learning model is employed to isolate the guitar track from multi-instrument recordings, ensuring cleaner transcription results.

10.3 Transcription System

The isolated guitar track undergoes the transcription process using a neural network-based model:

- **MIDI Generation:** The **Basic Pitch** transcription model is applied to detect onsets and note events, converting the audio into MIDI format.
- **Error Handling:** Post-processing techniques such as noise filtering and note smoothing are implemented to enhance transcription accuracy.

10.4 Notation Generation

The MIDI data is further processed to generate readable sheet music:

- **MusicXML Conversion:** The MIDI file is converted into **MusicXML**, a format suitable for musical notation.
- **PDF Export:** The **Verovio** rendering engine is used to generate high-quality PDF sheet music, allowing users to download and print their transcriptions.

10.5 Machine Learning Enhancements

To further refine transcription accuracy, machine learning techniques are explored:

- **Dataset Evaluation:** Various guitar-specific datasets are analyzed to improve model training.
- **Hyperparameter Tuning:** The transcription model is fine-tuned using optimization techniques to enhance note detection and timing precision.

10.6 User Interaction and System Integration

The system is designed to provide a seamless user experience through a web-based interface:

- **File Upload:** Users can upload guitar recordings directly to the website.
- **Processing and Download:** After automatic processing, the system generates a downloadable PDF containing the transcribed sheet music.
- **Real-time Feedback:** Future iterations may include interactive features for users to visualize transcription progress.

This structured approach ensures that the **Guitar Transcription System** provides an efficient and accurate solution for musicians seeking high-quality sheet music from their recordings.

Chapter 11

User Manual

11.1 11.1 Introduction

The **Guitar Transcription System** is a web-based application designed to help musicians convert their guitar recordings into sheet music effortlessly. The website allows users to upload an audio file, process it automatically, and download the generated sheet music in PDF format.

This user manual will guide you through the steps to use the website efficiently. By following these instructions, you can easily transcribe your guitar recordings and obtain high-quality sheet music for practice, learning, or sharing. The system ensures accuracy and ease of use, making guitar transcription accessible to all users.

11.2 Getting Started

1. **Open the Website**: Access the Guitar Transcription System through your web browser. There is no need to download or install any software.

2. **Upload Your Audio File**: Click on the **"Upload Audio"** button and select your guitar recording. The system supports common audio formats like WAV and MP3.

3. **Processing the Audio**: Once uploaded, the system will automatically process the

file to extract guitar notes and convert them into sheet music. This may take a few moments.

4. **Download the Sheet Music**: After processing is complete, click on the **"Download PDF"** button to save the generated sheet music.

5. **Use Your Transcription**: Open the downloaded PDF to view your guitar tablature and sheet music, ready for practice, learning, or sharing.

Chapter 12

Conclusions and Future Work

12.1 Conclusions

The **Guitar Transcription System** represents a significant advancement in the automation of music transcription. By leveraging state-of-the-art audio processing and machine learning techniques, this system provides musicians with a seamless way to convert guitar recordings into sheet music.

The key achievements of this project include:

- **Efficient audio preprocessing and separation**, utilizing **FFmpeg** for format conversion and **Demucs** for isolating guitar tracks from mixed audio.
- **Accurate note detection and transcription**, integrating **Basic Pitch** to convert isolated guitar audio into MIDI with high precision.
- **Automated notation generation**, using **Verovio** to convert MIDI files into **MusicXML** and export them as **PDF sheet music** for easy access and use.
- **User-friendly web interface**, allowing users to upload their recordings and download transcriptions without requiring any software installation or account creation.
- **Seamless end-to-end workflow**, eliminating manual transcription efforts and providing musicians with quick, high-quality results.

The development of this system has the potential to revolutionize the way guitarists transcribe music, making sheet music generation more accessible and efficient.

12.2 Future Work

While the current version of the **Guitar Transcription System** offers a fully functional pipeline for guitar audio transcription, there are several areas for future improvements and enhancements:

- **Enhancing transcription accuracy** by integrating advanced deep learning models for better note detection and timing precision.
- **Supporting multiple instruments**, expanding beyond guitar to allow transcription of other stringed instruments like bass and ukulele.
- **Improving tablature generation**, refining note-to-string mapping for more accurate fretboard representation.
- **Optimizing processing speed**, reducing the time required to analyze and generate sheet music from uploaded recordings.
- **Adding real-time transcription capabilities**, enabling users to see the notes being transcribed while playing live.
- **Expanding file format support**, allowing users to export transcriptions in **MusicXML, and MIDI** formats.
- **Conducting user studies** to gather feedback and improve usability, ensuring a more refined experience for musicians at all skill levels.

By pursuing these future advancements, the **Guitar Transcription System** can continue evolving into a more powerful and indispensable tool for musicians, ultimately streamlining the transcription process and fostering musical creativity.

Bibliography

[1] A Kolyshkin and S Nazarovs. Stability of slowly diverging flows in shallow water. *Mathematical Modeling and Analysis*, 2007.