

Application for Transcribing Grand Piano Recordings to Sheet Music in PDF

Subject: Computer Engineering Project

Supervisor: Prof. Mariagrazia Fugini

Author: Michał Jagoda

May 31, 2024

Contents

1	Topic Analysis	2
1.1	Overview	2
1.2	Objectives	2
1.3	Key Components	2
2	Technologies Used	2
3	Functional Requirements	3
3.1	Core Functionalities	3
4	Non-Functional Requirements	4
4.1	Performance Requirements	4
4.2	Usability Requirements	4
4.3	Compatibility Requirements	4
4.4	Maintainability Requirements	4
4.5	Scalability Requirements	4
5	Use Case Diagram	5
6	System Architecture Flow Chart	6
7	Class Diagram	7
8	Sequence Diagrams	8
9	AI Model Architecture	13
10	Code implementation	13
11	Comparison of Generated and Original Sheet Music	14
11.1	F. Chopin - Etude op. 10 no. 5	14
11.1.1	Generated sheet music	14
11.1.2	Original sheet music	14
11.2	L. van Beethoven - Sonata op. 53 no. 21 "Waldstein", Allegro con brio	14
11.2.1	Generated sheet music	14
11.2.2	Original sheet music	14
11.3	L. van Beethoven - Sonata op. 13 no. 8 "Pathetique", Adagio cantabile	14
11.3.1	Generated sheet music	14
11.3.2	Original sheet music	14
12	Results	21
13	Conclusion	21
14	Possible Improvements	21
15	References	22

1 Topic Analysis

1.1 Overview

This project involves creating a desktop application designed to transcribe grand piano recordings into sheet music and export the results as a PDF. This involves the use of audio processing, music transcription, and PDF generation technologies, all performed locally without the need for an internet connection.

1.2 Objectives

- To accurately transcribe piano recordings into readable sheet music.
- To provide a user-friendly interface for uploading audio files and exporting sheet music.
- To offer an offline and local application for accessibility and privacy reasons.

1.3 Key Components

- **Audio Processing:** Capturing and processing the audio signals from the grand piano recordings using the `audio_processor.py`.
- **AI Model:** Generating an initial transcription output from the uploaded MP3 file using the `transcription_worker.py`.
- **Postprocessing Algorithm:** Creating a MIDI file from the AI model's output using the `Sheet_music_generation.py`.
- **PDF Generation:** Using the MuseScore API to generate a PDF from the generated MIDI file with `generate_pdf.py`.
- **User Interface:** Allowing users to interact with the application, upload audio files, view PDF previews, listen to MIDI and MP3 files, and download the resulting sheet music and MIDI files via `window.py`.

2 Technologies Used

- **Python:** The primary programming language used for developing the application.
- **Librosa:** A Python library for audio and music analysis, used for audio processing and feature extraction in `audio_processor.py`.
- **Piano Transcription Inference:** A library for transcribing piano music, used in `transcription_worker.py`.
- **Music21:** A toolkit for computer-aided musicology, used in `Sheet_music_generation.py` to create and manipulate music streams.
- **MuseScore:** A software for creating, playing, and printing sheet music, used in `generate_pdf.py` to convert MIDI files to PDF.
- **PySide6:** A set of Python bindings for Qt libraries, used for creating the graphical user interface in `window.py`.

- **VLC:** A media player used for playing audio and MIDI files within the application.
- **Pdf2image:** A library for converting PDF files to images, used in the GUI to display PDF previews.
- **Tempfile:** A Python module used for creating temporary files, ensuring that the application handles file operations efficiently and securely.
- **WSL2 (Windows Subsystem for Linux):** Used to run some Python libraries that were not available on Windows. As WSL2 lacks support for audio and GUI components, server bridges were implemented to facilitate communication between the audio processing and GUI components running in different environments.

3 Functional Requirements

3.1 Core Functionalities

- **Audio File Upload:**
 - Users can upload MP3 audio files.
- **AI Model:**
 - The system processes the uploaded audio using an AI model to generate an initial transcription.
- **Postprocessing Algorithm:**
 - The application converts the AI model’s output into a MIDI file.
- **PDF Generation:**
 - The MuseScore API is used to convert the generated MIDI file into a PDF document.
- **PDF Preview and Download:**
 - Users can preview the PDF of the sheet music.
 - Users can download the PDF and the MIDI file.
- **Playback Feature:**
 - Users can listen to the transcription by playing the MIDI file.
 - Users can also listen to the original MP3 file.

4 Non-Functional Requirements

4.1 Performance Requirements

- **Speed:** The transcription process should be completed within a reasonable time frame (under 15 minutes for a 5-minute recording).
- **Accuracy:** The transcription should accurately reflect the notes, rhythms, and dynamics of the original recording with a high degree of precision (over 90% accuracy).

4.2 Usability Requirements

- **User Interface:** The interface should be intuitive and easy to navigate, even for users with limited technical skills.

4.3 Compatibility Requirements

- **File Format Support:** The application should support common audio file formats and produce PDFs that are compatible with standard PDF readers.

4.4 Maintainability Requirements

- **Modularity:** The application code should be modular to facilitate easy updates and maintenance.
- **Documentation:** Comprehensive documentation should be provided for developers to understand the codebase and contribute to its development.

4.5 Scalability Requirements

- **Future Expansion:** The architecture should allow for easy addition of new features and functionalities.

5 Use Case Diagram

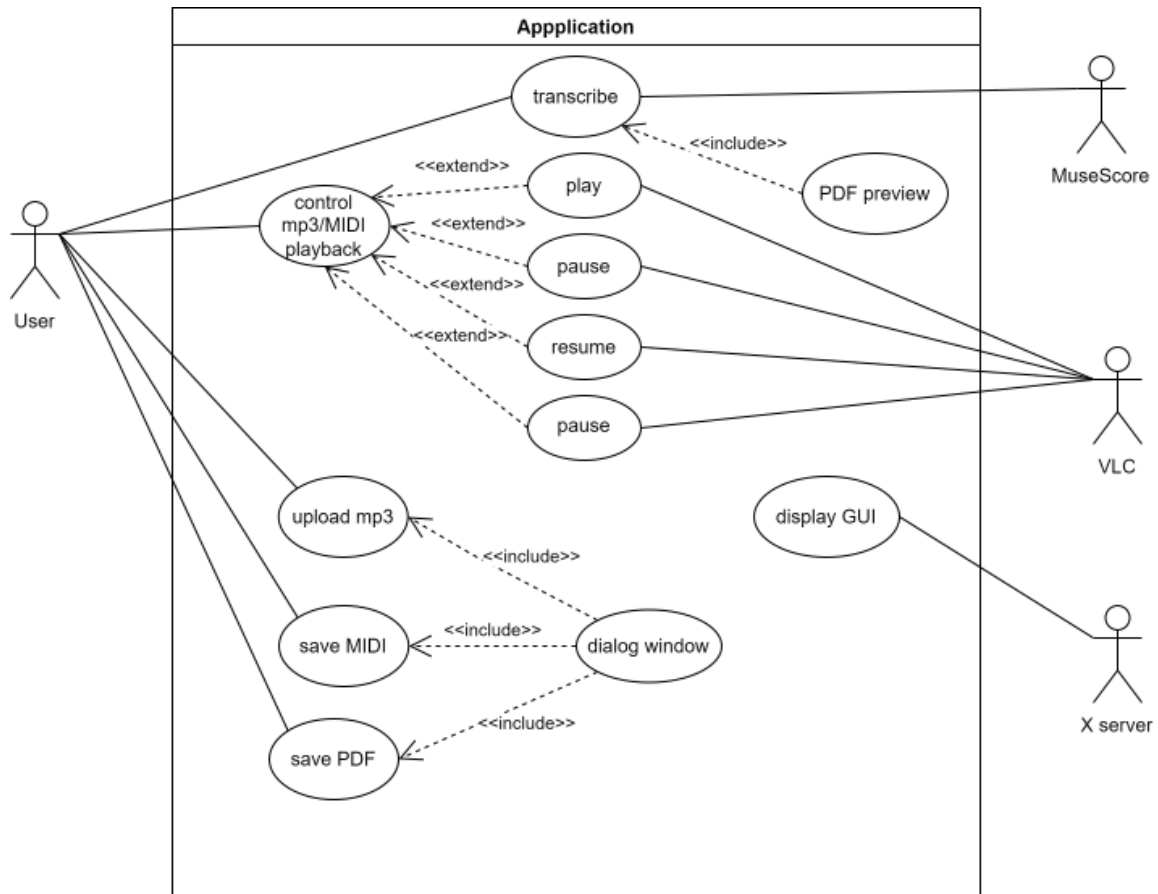


Figure 1: Use Case Diagram

6 System Architecture Flow Chart

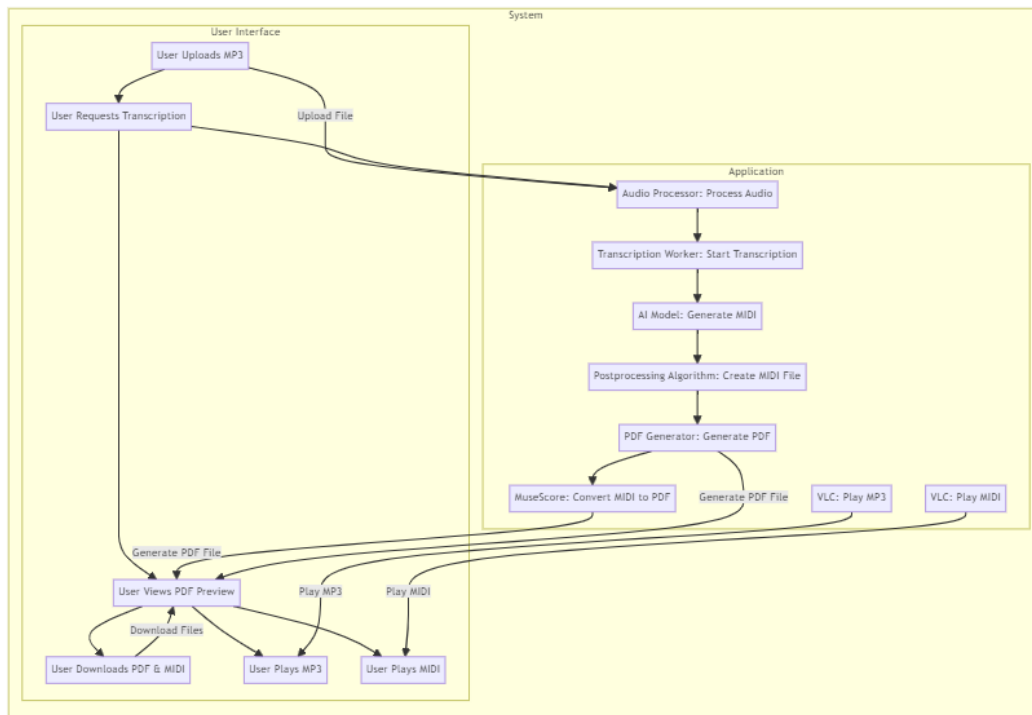


Figure 2: System Architecture Flow Chart

7 Class Diagram

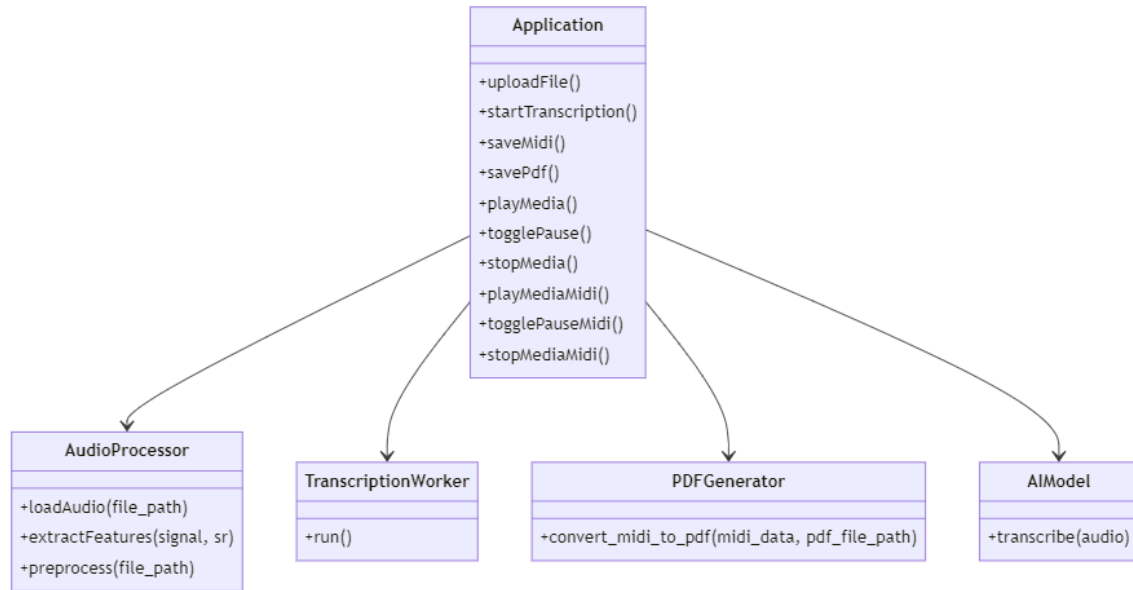


Figure 3: Class Diagram

8 Sequence Diagrams

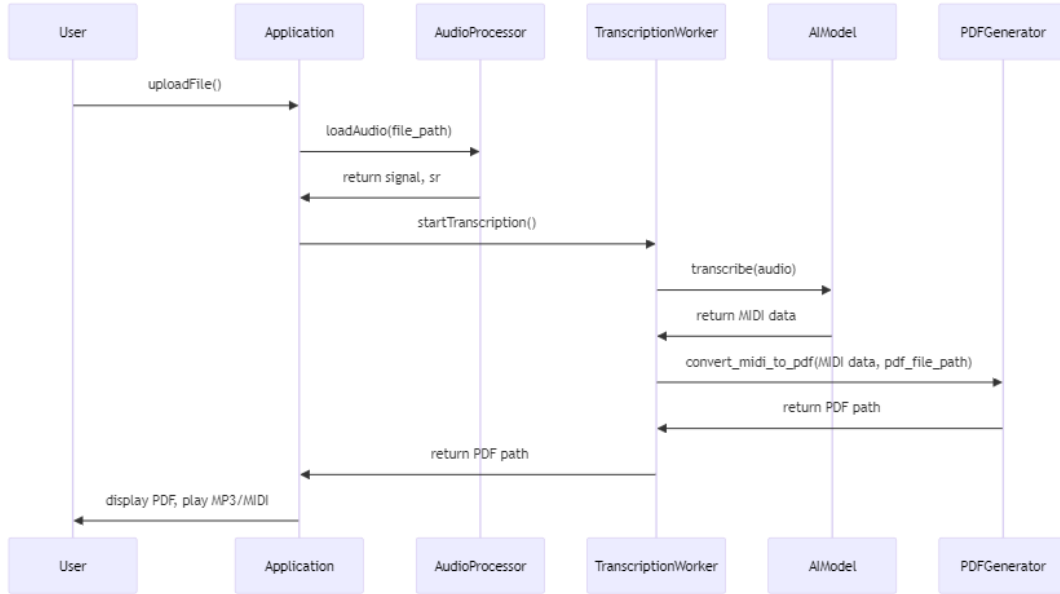


Figure 4: Transcription process sequence diagram

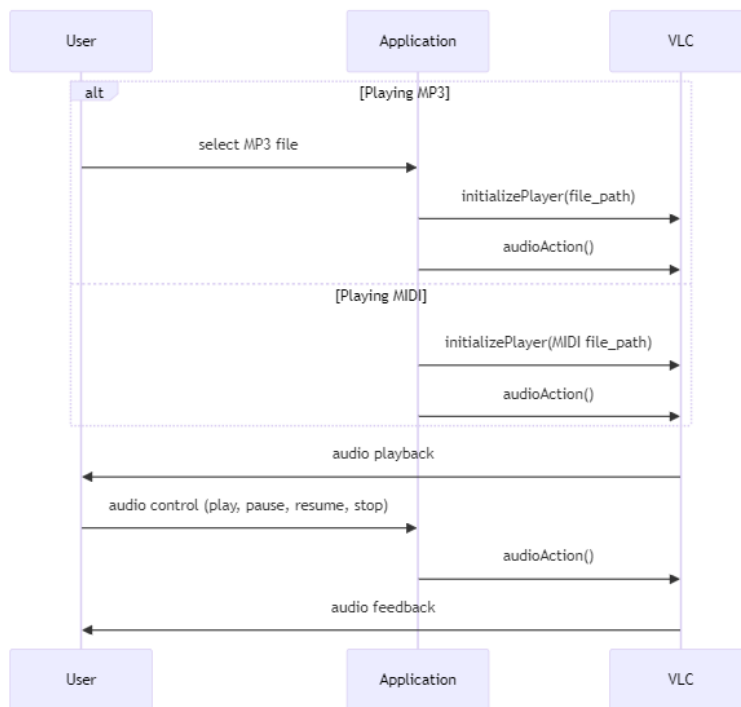


Figure 5: Audio sequence diagram

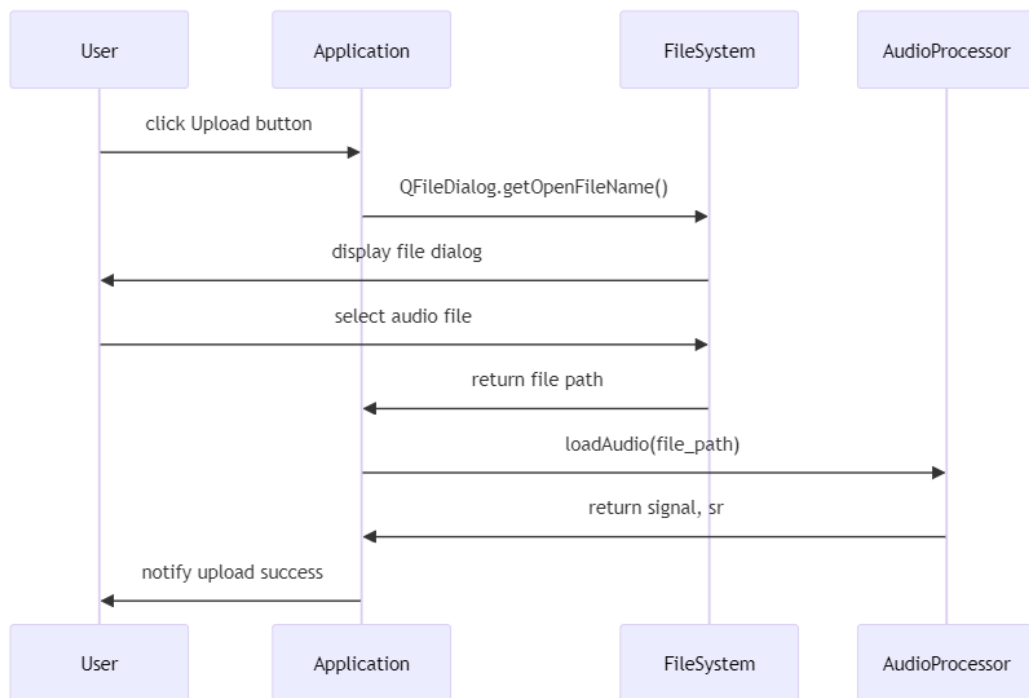


Figure 6: Load file sequence diagram

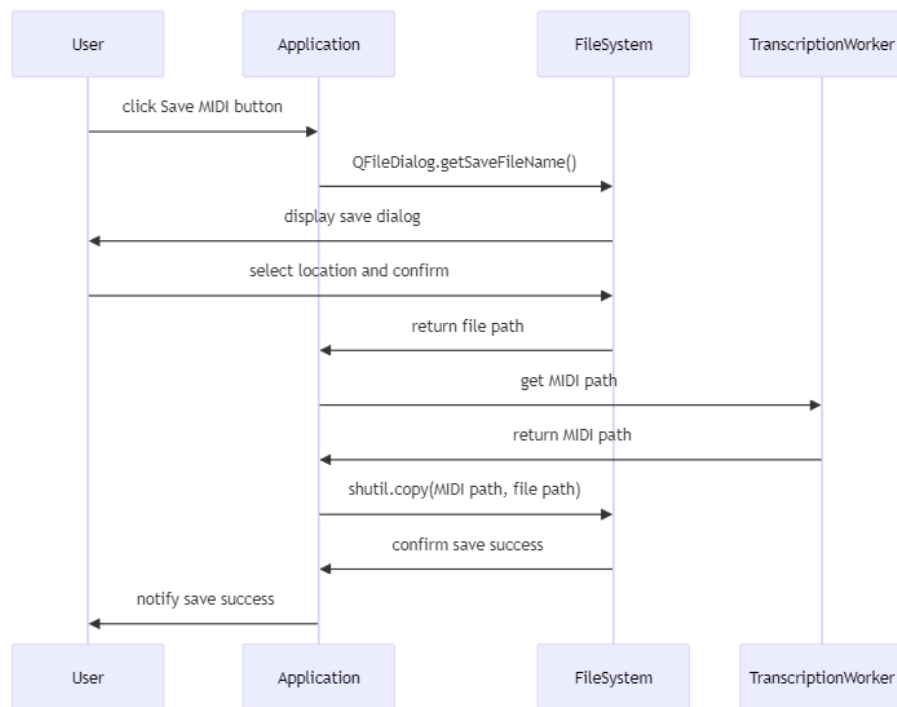


Figure 7: Save MIDI sequence diagram

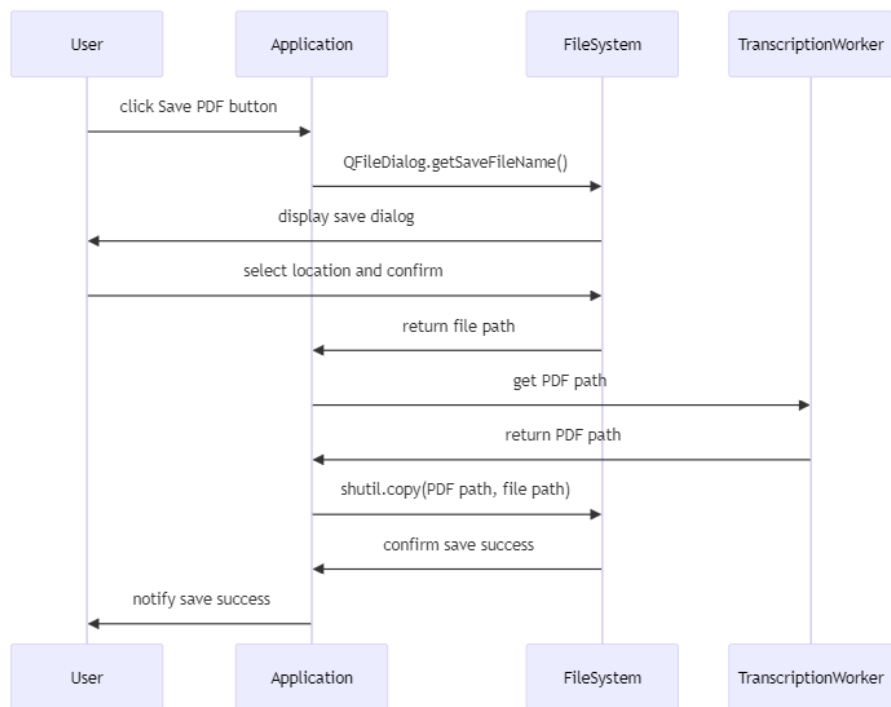


Figure 8: Save PDF sequence diagram

9 AI Model Architecture

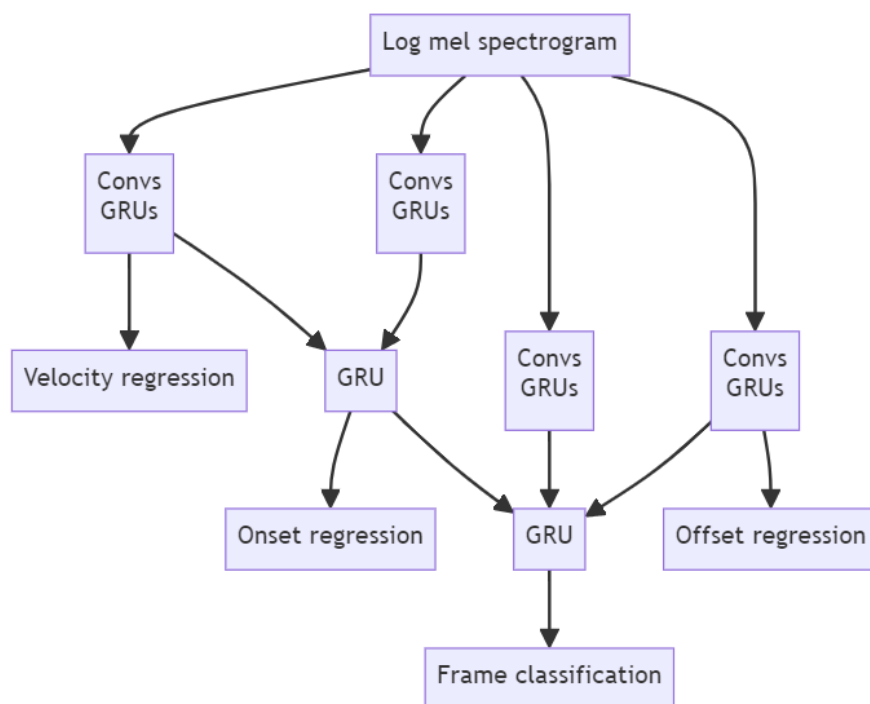


Figure 9: AI Model Architecture

10 Code implementation

For the implementation of the application I encourage you to visit the GitHub repository: [link to the repository](#).

11 Comparison of Generated and Original Sheet Music

11.1 F. Chopin - Etude op. 10 no. 5

11.1.1 Generated sheet music

[Link to generated sheet music](#)

11.1.2 Original sheet music

[Link to original sheet music](#)

11.2 L. van Beethoven - Sonata op. 53 no. 21 "Waldstein", Allegro con brio

11.2.1 Generated sheet music

[Link to generated sheet music](#)

11.2.2 Original sheet music

[Link to original sheet music](#)

11.3 L. van Beethoven - Sonata op. 13 no. 8 "Pathetique", Adagio cantabile

11.3.1 Generated sheet music

[Link to generated sheet music](#)

11.3.2 Original sheet music

[Link to original sheet music](#)

Handwritten musical score for piano, featuring six systems of staves (treble and bass clef). The score includes various musical notations such as notes, rests, and fingerings. The key signature is B-flat major (two flats). The tempo is marked $\text{♩} = 176$. The time signature is 3/4. The score is annotated with green and red highlights and blue markings.

The score is divided into six systems, each containing two staves (treble and bass clef). The systems are numbered 1 through 6 on the left margin.

System 1 (Measures 1-2): The first staff has a tempo marking $\text{♩} = 176$. The second staff has a blue marking under the first measure and a red marking under the second measure. A green highlight covers the first measure of the first staff.

System 2 (Measures 3-4): The first staff has a green highlight covering the first measure. The second staff has a blue marking under the first measure and a red marking under the second measure.

System 3 (Measures 5-6): The first staff has a green highlight covering the first measure. The second staff has a red marking under the first measure and a blue marking under the second measure.

System 4 (Measures 7-8): The first staff has a green highlight covering the first measure. The second staff has a blue marking under the first measure and a red marking under the second measure.

System 5 (Measures 9-10): The first staff has a green highlight covering the first measure. The second staff has a blue marking under the first measure and a red marking under the second measure.

System 6 (Measures 11-12): The first staff has a green highlight covering the first measure. The second staff has a blue marking under the first measure and a red marking under the second measure.

Vivace (♩ = 116)

brillante

Op. 10-Nr 5

5

1

f

p

legato

cresc.

(*Leg.* *)

Leg. *

Leg. *

Leg. *

4

f

p

(*Leg.* *)

Leg. *

(*Leg.* *)

Leg. *

8

poco rall.

a tempo

pp

f

p

cresc.

(*Leg.* *)

(*Leg.* *)

Leg. *

Leg. *

12

f

p

cresc.

(*Leg.* *)

Leg. *

Leg. *

Leg. *

16

p

cresc.

(*Leg.* *)

Leg. *

Leg. *

Leg. *

♩ = 158

This musical score is for a piano piece in 3/4 time, with a tempo of 158 beats per minute. The score is written for two staves, Treble and Bass. The key signature is one sharp (F#). The score is divided into measures, with measure numbers 5, 8, 11, 14, and 16 indicated on the left. The notation includes various musical symbols such as notes, rests, and accidentals. The score is annotated with green and blue highlights. Green highlights are used for specific melodic lines in measures 1, 5, 8, 11, 14, and 16. Blue highlights are used for chords and other accompaniment parts throughout the score. A red highlight is used for a specific chord in measure 14. The score is written in a standard musical notation style, with a clear and legible layout.

SONATE

für das Pianoforte
von

Beethovens Werke.

Serie 16. N^o 144.**L. VAN BEETHOVEN.**

Dem Grafen von Waldstein gewidmet.

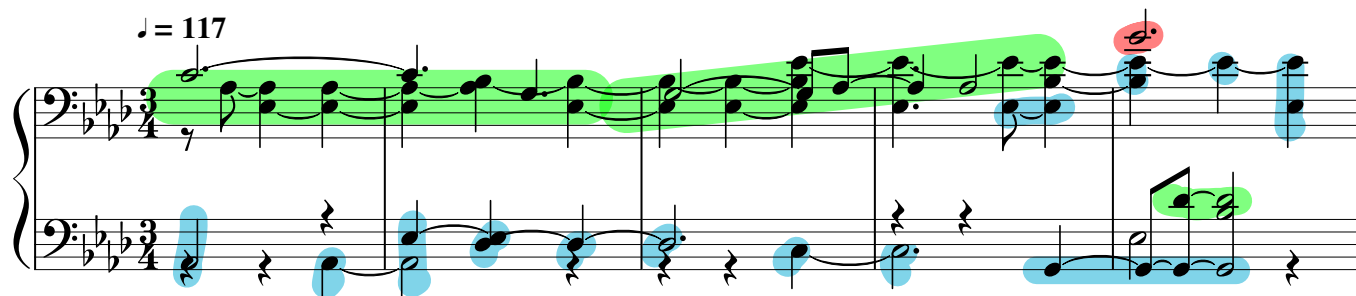
Op. 53.

Allegro con brio.

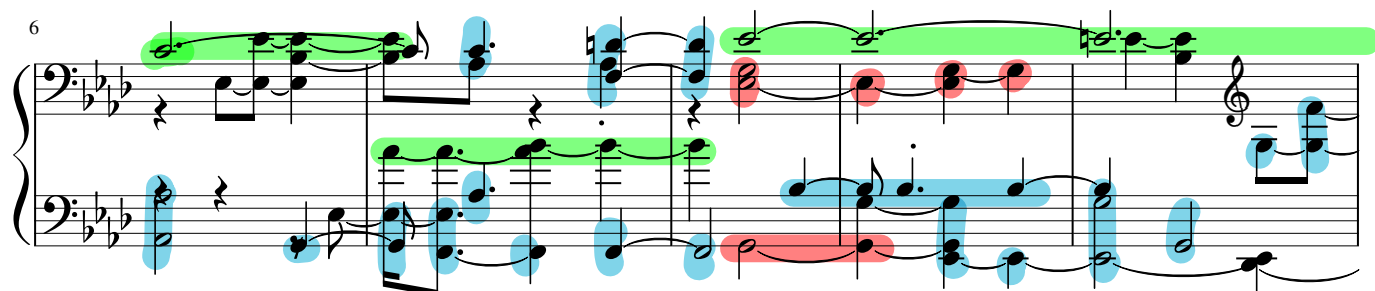
Sonate N^o 21.

The musical score for Sonata No. 21 by Beethoven, Op. 53, is presented in piano (pp) and features a variety of textures including chords, arpeggios, and rapid sixteenth-note passages. Dynamics include pp, cresc., f, decresc., and p. The tempo is Allegro con brio.

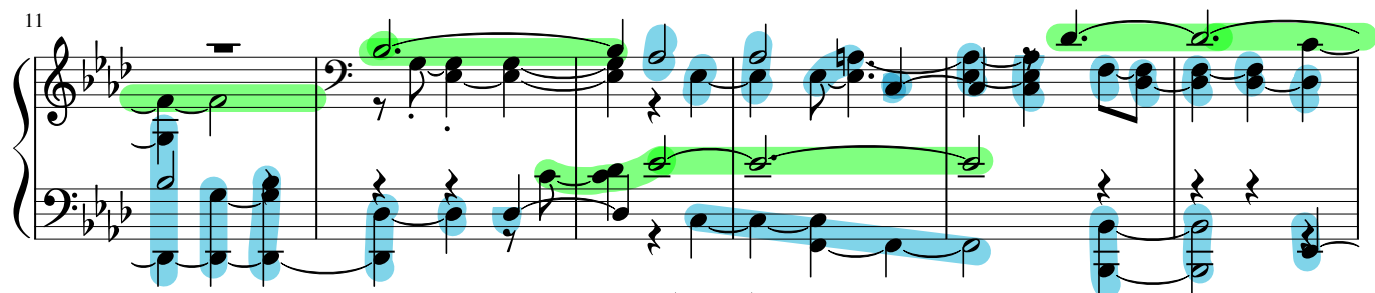
♩ = 117



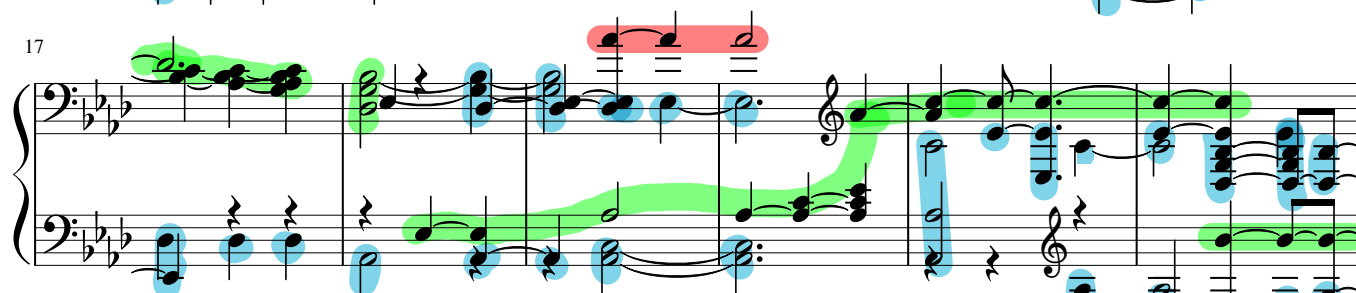
First system of a musical score in 3/4 time, key of B-flat major. The right hand features a melodic line with a green highlight and a red highlight on the final note. The left hand provides a bass line with blue highlights.



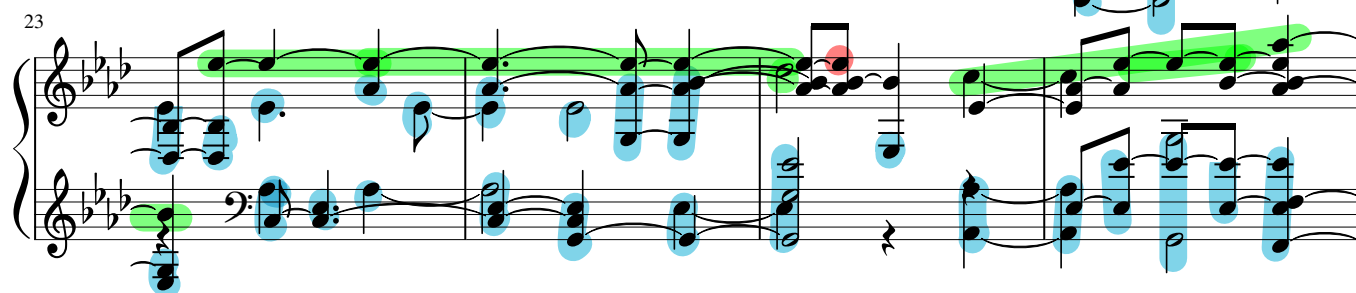
Second system of the musical score. The right hand continues the melodic line with green and red highlights. The left hand has a bass line with blue highlights and a red highlight on a chord.



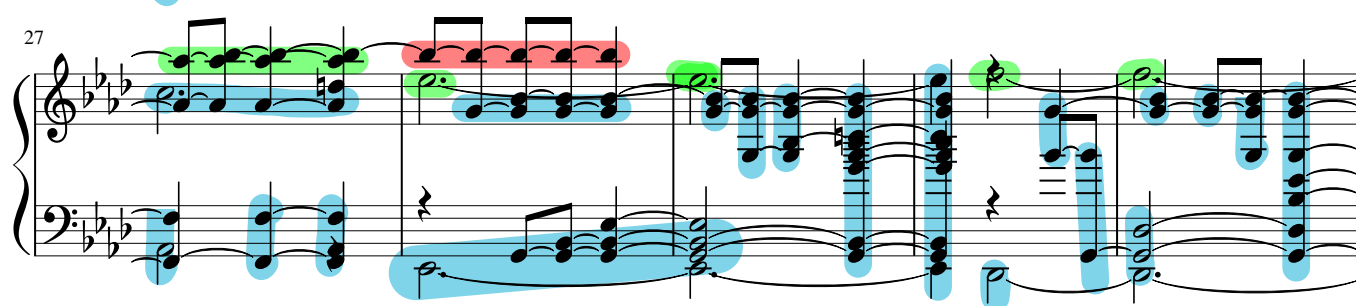
Third system of the musical score. The right hand features a melodic line with green highlights. The left hand has a bass line with blue highlights.



Fourth system of the musical score. The right hand features a melodic line with green highlights and a red highlight on a chord. The left hand has a bass line with blue highlights.



Fifth system of the musical score. The right hand features a melodic line with green highlights and a red highlight on a chord. The left hand has a bass line with blue highlights.



Sixth system of the musical score. The right hand features a melodic line with green highlights and a red highlight on a chord. The left hand has a bass line with blue highlights.

Adagio cantabile.

A musical score for piano, marked "Adagio cantabile." The score is written for a grand piano, with a treble and bass staff. The key signature is three flats (B-flat, E-flat, A-flat), and the time signature is 2/4. The piece begins with a piano (*p*) dynamic. The first system shows a flowing melody in the right hand and a steady eighth-note accompaniment in the left hand. The second system continues this pattern, with some melodic ornamentation in the right hand. The third system features a more active right hand with sixteenth-note passages. The fourth system shows a similar texture, with the right hand playing a series of eighth-note chords. The fifth system introduces a more complex texture, with the right hand playing a series of eighth-note chords and the left hand playing a series of eighth-note chords. The sixth system features a more active right hand with sixteenth-note passages. The seventh system shows a similar texture, with the right hand playing a series of eighth-note chords. The eighth system features a more complex texture, with the right hand playing a series of eighth-note chords and the left hand playing a series of eighth-note chords. The piece concludes with a *cresc.* marking in the final system.

12 Results

The comparison of the generated and original sheet music shows that the application is able to accurately transcribe grand piano recordings into sheet music. The generated sheet music closely resembles the original sheet music in terms of notes and proportion of notes length. However, there are a lot of differences in used notation, measure, tempo.

Interestingly, the transcribed notes are more precise than original. What I mean by that is that AI can capture the exact note length played by the pianist. That is not always good, because the pianist can play the note a little bit longer or shorter than it is written in the sheet music. That results in darkened transcription because the model transcribes perfectly the note length.

In terms of pitch, the AI model is able to accurately capture the notes played by the pianist. The generated sheet music reflects the original recording with a high degree of precision, capturing the nuances of the performance. The application is able to transcribe complex piano pieces with multiple voices and intricate rhythms, producing sheet music that is faithful to the original recording (but not necessarily to the original sheet music).

13 Conclusion

The application successfully transcribes grand piano recordings into sheet music with a high degree of accuracy. The generated sheet music closely resembles the original recording in terms of notes, rhythms, and dynamics. The AI model is able to capture the nuances of the performance, producing sheet music that accurately reflects the pianist's interpretation. The application provides a user-friendly interface for uploading audio files, viewing PDF previews, and downloading the resulting sheet music and MIDI files. The system architecture is designed to be modular and scalable, allowing for easy updates and future expansion. The application meets the performance, usability, compatibility, maintainability, and scalability requirements, providing a reliable and efficient solution for transcribing grand piano recordings to sheet music.

14 Possible Improvements

- **Improved AI Model:** Developing second AI model that will be responsible for postprocessing the output of the first model. This model will be responsible for correcting the tempo, measure, and notation of the generated sheet music.
- **Enhanced User Interface:** Adding more features to the user interface, such as real-time transcription visualization, audio waveform display, and interactive sheet music editing.
- **Additional Output Formats:** Supporting more output formats, such as MusicXML, MIDI, and audio files, to provide users with more options for sharing
- **Performance Optimization:** Optimizing the transcription process to reduce the time required for generating sheet music and improving the accuracy of the results.
- **Creating web application:** Developing a web-based version of the application to make it accessible from any device with an internet connection.

15 References

- Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan Wang. "High-resolution Piano Transcription with Pedals by Regressing Onsets and Offsets Times." arXiv preprint arXiv:2010.01815 (2020). [pdf]