

# Application for Transcribing Grand Piano Recordings to Sheet Music in PDF

Michał Jagoda

May 25, 2024

## Contents

<b>1</b>	<b>Topic Analysis</b>	<b>1</b>
1.1	Overview . . . . .	1
1.2	Objectives . . . . .	1
1.3	Key Components . . . . .	2
<b>2</b>	<b>Functional Requirements</b>	<b>2</b>
2.1	Core Functionalities . . . . .	2
<b>3</b>	<b>Non-Functional Requirements</b>	<b>3</b>
3.1	Performance Requirements . . . . .	3
3.2	Usability Requirements . . . . .	3
3.3	Compatibility Requirements . . . . .	3
3.4	Maintainability Requirements . . . . .	3
3.5	Scalability Requirements . . . . .	3
<b>4</b>	<b>Conclusion</b>	<b>3</b>

## 1 Topic Analysis

### 1.1 Overview

This project involves creating an 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.

### 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 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.
- **AI Model:** Generating an initial transcription output from the uploaded MP3 file.
- **Postprocessing Algorithm:** Creating a MIDI file from the AI model's output.
- **PDF Generation:** Using the MuseScore API to generate a PDF from the generated MIDI file.
- **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.

## 2 Functional Requirements

### 2.1 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.

## 3 Non-Functional Requirements

### 3.1 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).

### 3.2 Usability Requirements

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

### 3.3 Compatibility Requirements

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

### 3.4 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.

### 3.5 Scalability Requirements

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

## 4 Conclusion

By addressing both the functional and non-functional requirements outlined above, this application can provide a robust, user-friendly solution for transcribing grand piano recordings into sheet music. This will not only meet the immediate needs of users but also ensure the application's longevity and adaptability to future demands.