Universitat Politècnica de Catalunya

Intelligent System Project

Final Document

Master in Artificial Intelligence

Group 1

Authors:

Gerard Caravaca Ibáñez
Benjamí Parellada Calderer
Armando Rodriguez Ramos

Supervisor:

Miquel Sànchez Marrè

Fall Term 2023/2024

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Contents

1 Executive Summary

This report provides a succinct overview of the Wave2Page project, aimed at equipping stakeholders with essential insights for informed decision-making.

Statement of the Problem

Wave2Page addresses the significant challenge of converting audio recordings into digital sheet music. The market currently lacks automated systems capable of efficiently and accurately transcribing complex musical compositions into readable and usable sheet music, a gap Wave2Page aims to fill.

Basic Background Information

Wave2Page integrates advanced techniques from Signal Processing, Deep Learning, and Music Theory. The demand for such technology is driven by the needs of musicians, composers, and educators for reliable transcription tools, particularly for complex polyphonic music.

Analysis

The Solution: Wave2Page presents an innovative solution to convert audio recordings into MIDI format and subsequently into digital sheet music. This encompasses high-fidelity instrumental separation and sophisticated audio-to-MIDI transcription, addressing a key need in the music industry for accurate and comprehensive music transcription, particularly for complex, multi-instrument compositions.

The Design of the System: Developed with a modular architecture, the system ensures scalability and adaptability. It supports various input formats and guarantees output of high quality, aligning with professional music transcription standards, thus meeting the diverse needs of the music community.

Financial: The project is financially viable, with potential revenue streams from direct sales, subscriptions, and licensing. The initial project cost of €17,779.4 is offset by the projected revenue, which shows substantial growth over three years, indicating a successful recovery of the initial investment and long-term profitability.

Sustainability: Our commitment to sustainability is integral to the project. We've implemented practices to minimize our environmental impact, such as using energy-efficient cloud services for model training and operation. Socially, we're dedicated to making music transcription accessible to all, promoting inclusivity in the music community. Economically, our cost-effective and scalable solution ensures long-term viability, contributing positively to the industry and society.

Main Conclusions

Results: The developed system demonstrates high accuracy in transcribing various musical genres. It efficiently handles the challenges of over-fidelity and artifacts commonly found in existing automated systems.

Recommendations: For further enhancement, it is recommended to expand the system's capabilities to include more diverse musical instruments and styles. Continuous user feedback and iterative improvements are suggested to maintain the system's relevance and effectiveness.

Business Analysis: A comprehensive economic cost analysis and a detailed revenue estimation underscore the project's financial sustainability and market viability. With a strategic pricing plan and a growing user base, the project is set to not only recover its initial costs but also generate significant profits, making it a lucrative venture for potential investors and stakeholders.

2 Problem to be Solved

Wave2Page is an intelligent system set to transform the landscape of musical transcription by offering a comprehensive solution for converting multi-instrument audio recordings into digital sheet music. This cutting-edge system is meticulously designed to accommodate a carefully selected array of instruments that are pivotal in a wide variety of popular music genres, including vocals, guitar, bass, piano, and percussion. This focus mirrors a practical approach, aligning with the current capabilities and advancements in the realms of audio processing and music transcription technologies.

At present, the music industry lacks a holistic solution capable of processing complex, multi-instrument audio tracks into a coherent and readable sheet music format. The Wave2Page project addresses several critical gaps in this sector:

- **Innovative Multi-Instrument Conversion**: Unlike existing solutions that handle only single instruments, Wave2Page is engineered to transcribe entire songs into sheet music. This innovation marks a significant leap from the current industry standards.
- **Access to Intermediate Outputs**: We provide users with access to various stages of the transcription process. This includes the ability to obtain isolated tracks of each instrument, offering flexibility and a deeper insight into the music. As well as the individual post-processed MIDI file for each of the instruments separated.
- **Enhanced Audio Quality of Separated Stems**: We place a strong emphasis on refining the audio quality of separated stems, ensuring that the output material is not just accurate but also polished and of high fidelity.
- **Revolutionizing Drum-to-Sheet Transcription**: Addressing a significant void, our system includes a reliable drum-to-sheet pipeline, a feature notably absent in current technologies, to convert percussion elements accurately into written music.
- **Incorporation of Vocals**: Our system stands out by automatically integrating vocals into the sheet music, an advancement not found in existing automated solutions.
- **Advanced Post-Processing for Readability**: Unlike most systems that rely solely on direct MIDI-to-sheet conversion, Wave2Page introduces a crucial post-processing step. This allows us to refine and simplify the output, enhancing readability and capturing the nuances of the music more effectively for human readers.

The Wave2Page project, with its array of innovative features and focus on multi-instrument processing, is poised to redefine the standards of musical transcription. It aims to fill the existing gaps in the market, providing musicians, composers, and music enthusiasts with a tool that not only translates music from audio to sheet form but does so with an unprecedented level of detail, quality, and user-friendliness.

3 Background and State-of-the-Art

Automatic Music Transcription (AMT) stands as a pivotal domain in music technology, aiming to convert audio recordings into symbolic musical notation. This field has seen diverse methodologies and transcription types, each catering to different aspects of music and presenting unique complexities [1, 2, 3]. In AMT, transcription types range from piano solo transcription, where the challenge lies in accurately detecting a wide range of notes from a solo piano performance, to multi-instrument polyphonic transcription, which deals with the complexities of overlapping harmonics

and timbres from different instruments. Other notable transcription types include drum transcription, focusing on identifying percussive events; vocal transcription, which extracts vocal lines from polyphonic mixes; chord recognition, analyzing harmony and identifying chord progressions; and beat/downbeat tracking, pinpointing rhythmic elements in music.

The depth of transcription in AMT varies, with levels such as frame-level focusing on the estimation of note pitches in short time frames [1, 2, 4, 5, 6, 7], note-level connecting pitch estimates over time [1, 2, 8], and stream-level grouping notes into streams corresponding to different instruments [9, 2]. These levels reveal the intricate layers within AMT, from basic frequency estimations to detailed musical notation development.

Prominent studies in the AMT field have contributed significantly to its evolution. For example, Hung et al. used musical scores for supervised learning in source separation, focusing on different instrument sounds [10]. Jansson et al. applied U-Net CNN, initially used in medical imaging, to separate vocal and backing tracks [11]. Lin et al. introduced a unified model for source separation [12], transcription, and synthesis, while Manilow et al. developed an architecture for separating and transcribing musical mixtures [13].

Despite these advancements, the AMT market reveals certain limitations. Tools like ScoreCloud [14], MuseScore 4 [15], AnthemScore [16], and Noteflight [17], each with unique features, still face common challenges such as octave errors, misaligned notes, and issues in handling dense chords and unseen timbres. Current systems often fall short in accuracy and reliability, suggesting a need for further improvements and innovations in this field.

3.1 Context and Alternatives

Our problem involves multi-instrument transcription, a process crucial for understanding complex musical pieces containing multiple instruments. Deep learning-based methods have been pivotal in this field. For instance, architectures like those in [18, 19] integrate instrument recognition, transcription, and source separation. The pioneering MI-MPE method [20] focuses on piano rolls for different instruments. Additionally, [21] use multi-object semantic segmentation for pitch detection and instrument recognition. Other approaches first separate sources and then transcribe each instrument [13].

Unlike previous works focusing on finishing the transcription in MIDI or similar formats, we aim to parse it into symbolic notation, which none of the mentioned methods ends up doing. To aid in this process, we will experiment with MIDI-sheet [22] music alignment and audio-to-score alignment [23] which are used in piano music information retrieval. The former involves converting MIDI data into a simplified score format for alignment, while the latter uses neural networks to identify performance-score discrepancies.

For Wave2Page, our approach aligns with this latter methods of separation and those found in [13, 18, 19]. Our system will undergo stages of separation, transcription, and parsing to ensure accurate and musically faithful accuracy. Nevertheless, research in the field of accurately aligning MIDI with corresponding sheet music is currently nonexistent.

The audio separation module is responsible for isolating different audio tracks from a single audio file. This isolation is key to accurate transcription, as it separates distinct musical elements from a polyphonic composition. The primary model employed for this task is Meta's Demucs [24, 25] which stands out for its efficiency in isolating individual instruments and vocals from mixed audio tracks. Its architecture, which evolved from the U-Net convolutional model, incorporates a hybrid spectrogram/waveform separation approach combined with a Transformer architecture. This design allows it to process signals both in temporal and spectral domains, making it highly

effective in handling complex musical accompaniments. However, alternatives to Demucs exist, such as Spectral Clustering [26], Wave-U-Net [27], and the previously described [18, 19]. Spectral Clustering, based on Spectral Graph Theory, offers a traditional approach to audio source separation, particularly useful for sources with pronounced spectral differences. The Wave-U-Net model, tailored for 1D waveform processing, presents another alternative, especially for fine-tuning specific aspects like piano sound extraction.

The transcription module is responsible for converting the different audio tracks into a combined MIDI file. Spotify's Basic Pitch [28], a convolutional neural network, currently leads in mono-instrument transcription due to its efficient and precise audio-to-MIDI conversion capabilities. Renowned for its ability to accurately detect pitch bends, it also demonstrates versatility across a range of instruments, including vocals. Nevertheless, alternatives like CREPE [29], a model specializing in pitch tracking, and the Differentiable Digital Signal Processing (DDSP) library [30], which integrates signal processing with deep learning, offer additional avenues for refinement and customization in the transcription process.

4 Technical Project Description

4.1 Solution Design

In this part, we will explain the technical solution proposed, delving into each of the subtasks.

4.1.1 Task Analysis

The process of transforming audio into sheet music in the Wave2Page system involves several critical tasks. Each task plays a pivotal role in ensuring the system functions efficiently and accurately. The tasks, while specific in nature, are designed to be adaptable to a range of audio inputs and music genres. Moreover, for each of these tasks, the subtask of researching the state-of-the-art, and training/testing models to achieve the best performance is implied.

1. **Audio Format Standardization**: Given the variety of audio formats, the system initially converts all incoming audio files to a uniform format. This standardization is crucial for consistent processing across different file types.

2. **BPM and Time Signature Detection**: The first step in processing any audio file is to determine the Beats Per Minute (BPM) and time signature. This task sets the foundational tempo and rhythm structure for the subsequent transcription process.

3. **Audio Track Separation**:
   (a) **Track Splitting**: The core of the system is the track separation or *Splitter* system, which deconstructs a mixed audio track into separate stems for each instrument.
   (b) **Source Separation Enhancement**: To improve the quality of separated tracks, a dedicated task involves enhancing source separation.
   (c) **Silent Track Detection**: Post-separation, the system needs to identify and omit silent tracks.

4. **MIDI Conversion**:
   (a) **Instrumental Tracks**: Non-percussive, non-silent audio tracks are processed through an audio-to-MIDI conversion system that maintains their musical content.
   (b) **Percussion Processing**: Given that percussion does not conform to standard pitch-based transcription methods, a specialized model is required to accurately transcribe these elements.

(c) **Vocal Transcription**: If vocals are present, they are processed using a speech-to-text system to capture lyrical content accurately.

5. **Sheet Music Generation**:
   (a) **MIDI Post-Processing**: The generated MIDI files undergo post-processing, ensuring they align with conventional music notation standards and are visually appealing.
   (b) **Parsing to sheet**: The final task involves converting the processed MIDI files into sheet music. This task captures all the individual components into a musical score.

Each task is designed to handle specific aspects of the audio-to-sheet music conversion process, ensuring that the final output is a faithful and polished representation of the original audio.

4.1.2 Methods Implementing each Task

Preprocessing

In the Wave2Page system, the initial task is receiving the audio that is likely to be converted, specifically in WAV format. The WAV format is chosen due to its widespread use and lossless nature, ensuring that the audio quality is preserved during processing. This standardization is vital for consistent processing across different file types and is the first step in preparing the audio for subsequent stages of the transcription process.

The second task, is detecting the BPM of an audio piece, a crucial step for setting the foundational tempo for accurate music transcription and downstream tasks. Using the Librosa library [31], the process analyzes the onset strength envelope of the audio signal, a measure reflecting the intensity and frequency of rhythmic elements in the audio. By evaluating these onsets, it can efficiently estimate the track's BPM through the temporal dynamics of the signal.

For time signature detection, the process is more complex and less straightforward. Time signature detection in music remains an open research question [32]. While we have experimented with various models and approaches for time signature detection, we recognize the challenges and limitations in this area. We use a self-trained Support Vector Machine (SVM) model for classifying song excerpts by meter [33]. This SVM model conducts both local and broader acoustic analyses, examining properties like the spectrum around the beat and pitch over longer intervals. It utilizes similarity features to distinguish acoustic variations at and between beats, which is crucial for accurate time signature classification. Nevertheless, on the [34] corpus, we only achieved 87.2% accuracy, which is less than just assigning all the samples as 4/4. Considering that our system primarily targets popular music, which mainly features a 4/4 time signature [34], we have set this as the default in our system. This simplification streamlines the transcription process without affecting the accuracy for most of our target music genres.

Audio Track Separation

Once the audio is in the correct format, the next step is its separation into individual stems. We chose the Hybrid Transformer Demucs 6s [35] model for this purpose, recognizing its proficiency in segmenting key elements like drums, bass, vocals, guitar, and piano, despite some limitations in piano processing. Demucs is the current state-of-the-art in the field and, with our resources, attempting to surpass its separation capabilities is not feasible.

Upon reviewing the outputs from Demucs, it's evident that some artifacts are present in the separated stems, which is a common challenge in stem extraction processes. To mitigate this we utilized a denoising approach based on Real Time Speech Enhancement in the Waveform Domain [36]. This model, specifically optimized for vocal tracks, employs a real-time, CPU-friendly encoder-

decoder architecture with skip-connections, optimized using multiple loss functions in both time and frequency domains. This approach effectively removes various types of background noises and room reverb, making it highly suitable for enhancing vocal clarity.

In an effort to further refine the audio quality of individual instruments, we experimented with fine-tuning separate models on the MUSDB18 dataset [37], a benchmark dataset for music separation. The process involved extracting stems using Demucs, followed by fine-tuning models for each instrument with an L1 loss function. While this approach yielded an improvement in Signal to Distortion Ratio (SDR) for drums, bass, guitar, and piano, it inadvertently introduced a stuttering effect in longer notes and led to the emergence of ghost notes in the MIDI transcription, ultimately diminishing the overall system's quality. Consequently, we decided to limit the use of this enhancement to vocals only, where it demonstrated a clear benefit.

Demucs consistently outputs a fixed number of tracks, regardless of the presence or absence of these instruments in the source. To address this, we developed a silent track detector to identify and omit tracks that are mainly silent, which is crucial for avoiding the transcription of non-existent notes, enhancing the accuracy of the sheet music, and reducing our computational cost. This detector works by converting the audio track into amplitude and then to decibels, followed by calculating the duration of potential silences using a threshold. By computing the ratio of silence to the total duration of the song, we can effectively identify and exclude tracks which are predominantly silent. This approach was validated, accuracy of 96.7%, through the process of obtaining stems for fine-tuning the previous denoiser model, where we established our threshold by averaging the maximum values from tracks manually identified as silent.

MIDI Conversion

Following stem separation, we convert each into MIDI using the Basic Pitch convolutional neural network [28], known for its efficient and accurate audio-to-MIDI AMT. Basic Pitch's ability to handle polyphonic instruments makes it ideal for our needs. It transforms the separated inputs into individual MIDI files, which we then combine to recreate the complete musical composition in MIDI format, capturing the essence of the original piece.

However, handling percussion elements presents a unique challenge due to their distinct representation in the MIDI protocol and their amalgamation into a single audio stem by Demucs. Unlike other instruments, percussion sounds in MIDI are allocated a specific channel, and their transcription requires a different approach than the Basic Pitch model, which is not effective for these audio sources. Additionally, no freely accessible model is available for percussion transcription, thus, requiring a tailored approach.

For successful transcription of percussion, three key components are essential: the onset of the note, the specific percussion sound (or note), and the note's duration [38]. Given that drums typically lack sustain, we assume each note duration to be an eight of the time signature, which adequately preserves the granularity of the music without loss of detail. The onset detection is achieved through Librosa, which operates by identifying peaks in the onset strength envelope.

The core challenge, however, lies in classifying the type of percussion sound. To achieve this, we extract Mel-Frequency Cepstral Coefficients (MFCCs) [39] from each detected onset, utilizing them to train a small Convolutional Neural Network (CNN). This approach is chosen after testing other features like spectral centroid, spectral rolloff, and zero-crossing rate, which did not significantly enhance accuracy while impacting computational efficiency.

To train this CNN, we compiled a custom dataset by scraping the web for freely available drum sounds, ensuring they were license-compliant. This dataset, comprising over 3,000 sounds typical

of a rock drum-kit, was then categorized based on their names. The trained model achieved 87.9% accuracy in correctly classifying the drum sounds during testing. Additionally, in cases of low confidence scores in the predictions, we opted to use multiple instruments in the MIDI transcription, reflecting the common practice in percussion where multiple sounds are often played simultaneously. This nuanced approach to percussion transcription significantly enhances the authenticity and richness of the transcribed MIDI files in our system.

If vocals were detected in the audio files, we additionally transcribed it using a speech-to-text model. The *OpenAI Whisper* model [40], known for its accuracy and efficiency, is then employed to transcribe the audio data into text, specifically utilizing the Large English model to ensure accuracy and prevent misinterpretation from other languages. These are then parsed into each of the different notes to symbolize that each note represents a word.

Sheet Music Generation

MIDI files, despite their precise capture of performance nuances, often don't align seamlessly with the symbolic representation in sheet music. They include all performance details, which, while accurate, can result in a cluttered representation, unlike the cleaner output of a human transcribed sheet. Musicians who are accustomed to reading sheet music typically prefer transcriptions that prioritize readability over an exhaustive detailing of every performance aspect. Furthermore, the automatic audio-to-MIDI process can inadvertently capture unintended overtones from the original performance, resulting in more notes than performed.

To address this, our system incorporates a thorough rule-based post-processing stage before converting MIDI into sheet music. This process involves instrument-specific adjustments such as quantizing MIDI notes and eliminating unnecessary overtones. For example, bass and vocal tracks typically don't feature harmonies, so we remove chords and keep the bass note. The post-processing also involves filtering out low-velocity notes, setting limits on note durations, correcting overlapping notes, removing pitch outliers, and rearranging chords into more conventional patterns. These rules were developed based on our analysis of the Lakh dataset [41].

We also explored enhancing our system with a language model, specifically using RWKV [42], by first converting MIDI into a string format and then processing it through the model. However, this approach proved challenging in maintaining control over the MIDI content, often leading to the removal of essential notes and unwanted alterations. Consequently, we decided not to incorporate it into our system. Interestingly, this method showed potential for music generation, although the output tended to be somewhat repetitive and uninteresting to listen to.

The final step in our process involves using MuseScore 4 to transform the refined MIDI into a sheet music PDF. Measuring the accuracy of our transcriptions quantitatively was challenging, so we conducted a focus group with three musicians to evaluate the system's performance. While they noted some discrepancies and inaccuracies compared to the actual performance, they found our processed version significantly more readable and accurate than other competing software, indicating a successful balance between accuracy and usability in our approach.

Addressing the performance metric for our final audio-to-score system is a complex and somewhat ill-defined problem, primarily due to the limited research in this specific area. This complexity arises from two key factors: firstly, a musician's performance of the same musical piece can vary with each rendition, and secondly, the same piece of music can be transcribed in multiple ways. The current alignment metrics we explored are inadequate for tackling this issue. Therefore, a novel, research-worthy alignment approach is necessary. One potential solution we propose involves collecting a set of clean MIDI performances alongside the original composer scores for the

same pieces, converted into MIDI format[1]. With this approach, we aim to understand what a 'clean' score-derived MIDI looks like. This understanding could then inform the development of a system that learns to align and evaluate MIDI transcriptions by comparing them with these high-quality, composer-intended MIDI representations. Such a comparison could offer insights into the fidelity of our audio-to-score transcriptions, allowing for a more accurate and contextually relevant assessment of the system's performance.

4.2 Functional Architecture of the System

Figure 1 depicts the final schema of our intelligent system. It is divided into 4 regions that progressively convert from an audio or YouTube link to a complete music sheet. Each of the parts is deeply described in its corresponding previous sections. However, this figure allows us to fully comprehend the whole process and functional integration.

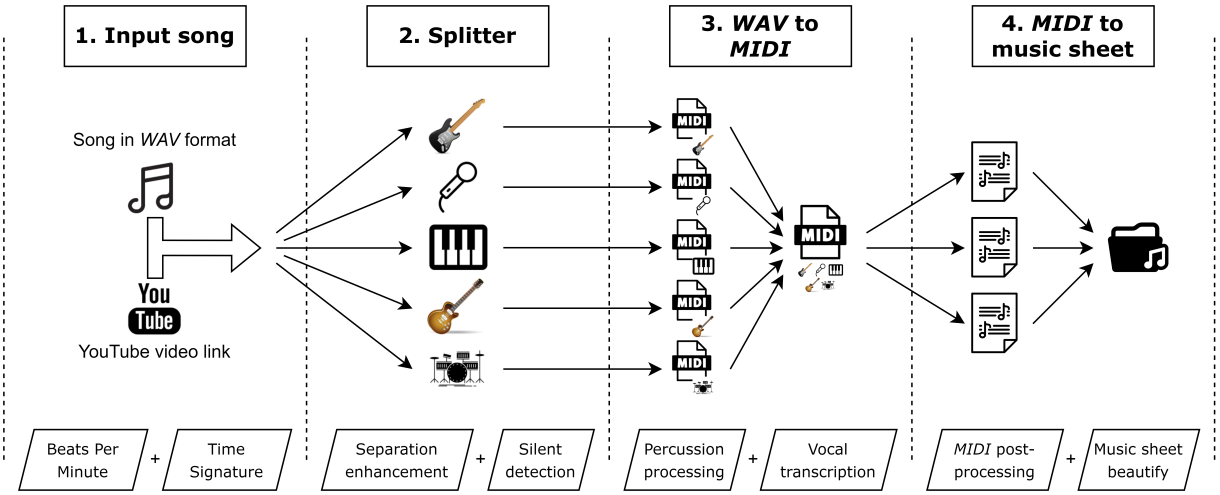


Figure 1: Final schema of the whole project. The boxes at the bottom represent the corrections and work done in each of the phases.

The tilted boxes at the bottom of each section try to represent the extra processes and steps made during the completion of the section. For the first part, after the song is introduced, we compute the beats per minute and the time signature. In the second part, we improve the Demuc's state-of-the-art source separation with original ideas and perform the silence detection to discard the instruments that didn't make an appearance in the song.

In the third step, we created innovative percussion processing in order to enhance the quality of its conversion and also used a state-of-the art automatic speech-recognition system to obtain the lyrics of the song. In the last step, we perform post-processing on the midi in order to obtain a much better representation. Lastly, we beautify the generation of the music sheet to achieve the required high-quality standards. The system's modular design allows for flexibility and adaptability, accommodating future enhancements and adjustments to improve accuracy and efficiency.

5 Economic Cost Analysis

This section provides a comprehensive analysis of the final economic costs associated with the project, building upon the expected costs outlined in the *Definition of the Project* document. The costs are aggregated by the various stages and components of the project to offer a complete financial overview. For this purpose, a comparative analysis will be made between the expected

[1]This differs from Lakh dataset, which lacks sheet music-derived MIDI and only has performance-based MIDI

costs and the final costs, as well as an estimate of the future benefits of the project.

5.1 Human Resources & Development Cost

The staff profiles involved in this project consist of a project manager and three developers, each with their assigned roles: a Team Leader (Benjamí Parellada), responsible for overseeing the progress and coordination of development tasks; a Solutions Architect (Armando Rodriguez), who focuses on designing robust and scalable solutions and addressing technical challenges; and a Software Developer (Gerard Caravaca), tasked with implementing and testing the designed solutions in accordance with the project requirements. The project manager was played by the professor Miquel Sànchez. Table 1 shows the detailed hourly wage breakdown for each role.

Role	**Gross salary**	**Social security**	**Remuneration**
Director (D)	21.65 €\h	6.50 €\h	28.15 €\h
Developer (P)	12.69 €\h	3.81 €\h	16.50 €\h

Table 1: Summary of staff costs. In order to calculate the estimated salaries for each role, estimates provided by the *Indeed* [43] website have been used. We assume that the different roles do not impact the salary of the developers, as they are all three Junior Software Developers.

After determining the individual cost of each staff profile and tallying the total hours worked (Table 5), we can derive the final economic costs. Table 2 displays the overall breakdown of the respective costs grouping certain Work Packages together. Our initial cost analysis took a conservative approach, intentionally overestimating personnel expenses. Nevertheless, the overestimation was still below the actual resource costs, where we were 17% above our initial estimates.

Role	**Estimated hours**	**Dedicated hours**	**Estimated cost**	**Final cost**
Director (D)	50	50	1,407.5	1,407.5
Meetings	36	36	594.0	594.0
System Analysis	140	185	2,310.0	3052.5
System Implementation	150	189	2,475.0	3,118.5
System Testing	14	16	231.0	264.0
System Documentation	110	119	1,815.0	1,693.5
Total	450 h + 50 h	595 h	€8,832.5	€10,400.0

Table 2: Comparison between estimated staff costs and final costs. The director is assigned with different tasks than the developers, which equitably (∼180 h each) do the rest of the work.

5.2 General expenses

In this section, we calculate the generic costs that are not directly tied to specific project tasks. These include depreciation, workspace, electricity, and internet consumption.

- **Workspace:** The project required a workspace and a meeting room. We used a room at home and the FIB classroom. The average cost of a room in Barcelona is approximately 400 €/month, and a meeting room like the one at the FIB is about 16 €/hour. Over 5 months, with 36 hours of meetings and co-working, the total workspace cost amounts to $400 \times 5 \times 3 + 16 \times 36 =$ €6,576.
- **Hardware:** The hardware includes two mid-range laptops (€650 each) and a high-range computer (€1,000). Assuming a five-year lifespan, the amortization comes to $\frac{5}{60} \times 650 \times 2 + \frac{5}{60} \times 1000 =$ €191.7.
- **Software:** All software used in this project is open source, resulting in an estimated amortization cost of €0.

- **Internet Bill:** Based on personal bills, the internet cost is around 43 €/month. Thus, the total for 5 months is 43 €/month × 5 months = €215.
- **Electricity Consumption:** With electricity prices between 0.23 and 0.31 €/kWh in Spain, we use an average rate of 0.27 €/kWh. A laptop consumes about 49.5 kWh over 180 hours, so the total electricity cost for the three laptops is 0.27 × 49.5 × 3 = €40.1.

These calculations provide a comprehensive overview of the general costs associated with the project, amounting to a total of approximately €7,022.8.

5.3 Server Costs

In our project, unexpected expenses emerged due to training deep learning models and hosting cloud services on Microsoft Azure (as detailed in Section 6.3). We utilized the Azure NC6 server with 56 GB RAM and a K80 GPU, suitable for our needs [44]. Under a pay-as-you-go pricing model, where the server cost is €0.915 per hour.

We used the server for two distinct purposes. First, to fine-tune deep learning models for 10 hours. The total cost for this training was 0.915 €/hour × 10 hours = €9.2. Second, we used the server for web hosting and testing over two weeks (24-hour daily operation), the cost was 0.915 €/hour × 24 hours/day × 14 days = €307.4. Furthermore, a Domain name for one year costs €10 at GoDaddy [45]. Hence, the total expenditure for setting up the website, including model training and initial web hosting, was €326.6.

For cloud processing, the cost per song, assuming each requires 1 minute of GPU time, is calculated as 0.915 €/hour/60 = €0.01525 per song. This estimate helps in understanding the operational costs for processing individual songs on the platform.

5.4 Final cost evaluation

Our project underwent comprehensive financial assessments, leading to the final cost evaluation detailed in Table 3. This table compares the initially estimated costs with the actual final costs. Initially, we allocated a contingency fund of 20% of the estimated cost for unforeseen expenses. Notably, a portion of this contingency was utilized, reflecting the dynamic nature of project management. The total estimated cost of the project was €19,003.3, while the actual final cost was €17,749.4. This resulted in a saving of €1,223.9 from the initial estimates, demonstrating efficient cost management despite the challenges encountered.

Concept	**Estimated cost**	**Final cost**
Staff cost	8,832.5	10,400.0
General cost	7,022.8	7,022.8
Server cost	-	326.6
Contingency cost	3,148.0	-
Total	€19,003.3	€17,779.4

Table 3: Final project cost evaluation.

5.5 Benefits estimation

Once the development costs have been studied, the next step is to estimate the benefits of the project. For this, we have designed a **pricing plan**. It is structured to cater to a broad spectrum of users, from casual enthusiasts to professionals.

Our website offers unique music services such as audio to sheet conversion, audio source separation, a YouTube Chrome extension, MIDI to sheet conversion, and whisper transcription, among others. These services cater to a diverse range of users including musicians, producers, educators, and

music enthusiasts. After a brief study of the market [46, 14, 16, 17], we believe our price point is just and accurate for our services provided. Thus, the proposed pricing structure contains the following pricing plans:

- **Free Tier:**
  - Purpose: Attract new users and let them experience the basic functionalities.
  - Features: Limited usage (e.g., 5 conversions per month), basic features of each service, ad-supported.
  - Ideal For: Casual users or those who want to test the services.
- **Standard Subscription:**
  - Purpose: Offer full access to more dedicated users.
  - Pricing: A monthly or annual fee (e.g., €4.99/month or €49/year).
  - Features: Unlimited access to all basic services, no ads, priority customer support.
  - Ideal For: Regular users, amateur musicians, and educators.
- **Premium Subscription:**
  - Purpose: Cater to professional and heavy users with advanced needs.
  - Pricing: Higher tier pricing (e.g., €14.99/month or €149/year).
  - Features: All Standard features plus advanced options (like high-resolution outputs), exclusive tools (advanced source separation, etc.), and premium support.
  - Ideal For: Professionals, music producers, and studios.
- **Pay-Per-Use:**
  - Purpose: Flexible option for users who need services occasionally.
  - Pricing: Based on usage (e.g., €0.5 per conversion).
  - Features: Access to specific services on a per-use basis.
  - Ideal For: Users with one-time or infrequent needs.
- **Educational/Non-Profit Discount:**
  - Purpose: Support educational institutions and non-profit organizations.
  - Pricing: Discounted rates (e.g., 30% off on subscriptions).
  - Eligibility: Verification required for educational institutions and non-profit organizations.

In order to estimate the yearly benefits of our proposed pricing plan, we made several key assumptions that reflect typical user behavior and market trends. We started with an initial user base of 300, anticipating a consistent growth rate of 5% per month. The distribution of users across different pricing tiers was assumed as follows: 70% in the Free Tier, 25% in the Standard Subscription, 5% in the Premium Subscription, and 5% in the Pay-Per-Use category. Additionally, we estimated that 10% of Standard and Premium subscribers would be eligible for an Educational/Non-Profit Discount. Furthermore, we subtracted the average cost per song to the previous revenue to obtain the profits after running the simulation. Table 4 shows the profit estimation by respecting all these constraints.

Year	Users	Standard (€)	Premium (€)	Pay-Per-Use (€)	Educational (€)	Total (€)
1	524.70	4,712.87	9,438.34	39.35	2,337.55	16,528.11
2	917.71	8,242.84	16,507.71	68.83	4,088.38	28,907.77
3	1,605.08	14,416.78	28,872.09	120.38	7,150.61	50,559.86

Table 4: Yearly Profit Estimation, achieved through simulation with the previous constraints.

Analyzing the revenue estimations over three years against the initial project cost of 17,779.4 €, it is evident that the project is financially successful and sustainable. Starting from the first year, the revenue exceeds the project cost, indicating a quick recovery of the initial investment.

This positive trend continues with substantial year-on-year growth, both in revenue and user base, reflecting strong market acceptance and effective pricing strategy. By the third year, the revenue more than triples the initial cost, underscoring the project's profitability and market viability.

6 Sustainability Analysis

This section provides a detailed exploration of current environmental thinking from a variety of perspectives, including social, economical and environmental edges. This analysis recognizes that the success of our project extends beyond mere technological achievement, emphasizing our commitment to responsible practices that benefit the environment, society, and the economy.

6.1 Social Edges

Social sustainability is a cornerstone of our project, as we aim to create positive impacts within the music community and society at large. One key aspect is inclusivity, where our AI-generated music sheets have the potential to make music notation more accessible to everyone.

By offering free access to foundational music sheet generation tools, and a discount premium plan for educators, we endeavor to level the playing field for musicians from varied socio-economic backgrounds. This approach aims to engender a sense of fairness within the music domain, cultivating a musical environment that is more socially sustainable and inclusive.

6.2 Economical Edges

On the economic sustainability of the project, we recognize the importance of conducting a comprehensive cost-benefit analysis to ensure the long-term viability of our AI-driven music sheet generation system. Through market research and analysis, we have identified a growing demand for AI-generated music sheets among music educators, composers, and students. Additionally, we have assessed potential competitors and devised strategies to differentiate our offering, ensuring our long-term sustainability and relevance in a competitive market.

On the other hand, as our project expands, we anticipate the creation of job opportunities in various domains. This includes software developers, AI specialists, musicologists, and customer support staff. By contributing to job growth, our project not only enriches the employment landscape but also supports the local economy.

On the cost optimization side, we have meticulously analyzed the incurred cost, and previsedpossible costs due to risks and unforeseen events. This gives us a certain credibility and security in the economic growth of the project. You can review this economic analysis in Section 5.

6.3 Environmental Edges

Environmental sustainability is a crucial facet of our project's overarching strategy. In our commitment to reduce the ecological footprint, we diligently evaluate the energy consumption associated with training and operating the Intelligent System.

To minimize environmental impact, we explored the adoption of energy-efficient hardware and cloud computing solutions, harnessing technology that maximizes computational efficiency while minimizing power usage. For this, we use *Microsoft* services to train and analyze our Deep Learning models. A joint Microsoft-WSP study found that cloud computing may reduce energy consumption by 93% and carbon dioxide emissions by 98% compared to traditional on-premises IT infrastructure [47]. In consequence, this approach not only lowers energy consumption but also aligns with our commitment to environmental responsibility.

In terms of carbon emissions, we calculate the project's carbon footprint. For instance, during the

model training phase, we measure the electricity consumption and compute the associated carbon emissions. By doing so, we can demonstrate our commitment to carbon neutrality and contribute to global efforts in combating climate change. For doing this calculation, we take into account three main branches:

- **Fine-Tuning on Microsoft Cloud Services**: Specific data on CO2 emissions per hour on the Microsoft's cloud NC6 server is not readily available. Nevertheless, on average, a web page produces about 0.8 grams of CO2 equivalent per pageview [48]. We can safely assume that our webpage will be above this average due to the expensive computations.

- **Use of three laptops**: Assuming each laptop is used for an average of 2 hours per day, the total usage over 3 months (approx. 90 days) is 180 hours per laptop, so 540 hours in total. If we use the median figure of 61.5 kg CO2eq for 4 years of laptop use [49], the emissions for 3 months can be estimated by scaling this number down. Emissions per laptop for 3 months $= \frac{61.5 \text{ kg } CO_2\text{eq}}{4 \text{ years}} \times \frac{3}{12}$ years $= 3.84$ kg CO_2. Thus, the average emissions for one laptop in three months is $= \frac{3.84 \text{ kg } CO_2\text{eq}}{30\text{days/month} \times 24\text{hours/day}} \times 180$ hours $= 0.96$ kg CO_2. Which implies a total consumption of 2.88 kg CO_2 for the three laptops.

7 Project Management

Project management is the central piece that must ensure that the work is always being done in the right direction. To ensure this, an adequate working methodology has been used.

7.1 Project Management Methodology

We have opted for the *Agile* methodology as our project management strategy. Agile, with methodologies like Scrum and Kanban, prioritizes flexibility and iterative development–a fitting choice for our project's evolving requirements. This approach enables us to swiftly adapt to changes, enhances team collaboration, and ensures efficient issue resolution.

Specifically, we have chosen to implement the **Scrum** [50] methodology for our project. Scrum is designed to optimize team coordination and productivity. It emphasizes frequent and incremental deliveries of work, prioritizing tasks based on their value to the project's stakeholders. This approach is particularly effective for complex projects involving multiple participants and diverse requirements, as is the case with Wave2Page.

The Scrum methodology goes through the following different phases, as described in [50].

- **Sprint Planning:** A phase where the team identifies and prioritizes tasks for the upcoming Sprint, based on the project's overall goals and current progress. This meeting involves discussing detailed requirements and planning the work to be completed. It is essential for setting the direction for the Sprint, allowing the team to adapt and evolve their approach based on learnings and feedback from previous Sprints.
- **Sprint Execution:** This is a time-boxed period, typically no longer than one month and often lasting two weeks, where the team focuses on completing the tasks defined during the Sprint Planning. This phase is dedicated to developing and implementing features, fixing bugs, or carrying out other tasks necessary to achieve the Sprint goals.
- **Daily Stand-up:** A daily meeting, where the progress of the project is reviewed. In this brief meeting, team members discuss what they worked on the previous day, what they plan to work on today, and any blockers they're facing.

7.2 Project Management Tools

To support our Agile project management approach, we employed a comprehensive suite of digital tools carefully selected to align with our project's unique requirements and objectives. These tools encompass a wide range of functionalities, ensuring that every aspect of project management and collaboration is optimized for efficiency and effectiveness.

Jira [51] serves as the central hub for our project management activities. This versatile tool enables us to track tasks, issues, and user stories, facilitating sprint planning, backlog management, and progress monitoring. Its flexibility and scalability make it an ideal choice to adapt to our evolving project needs. In conjunction with Jira, **Confluence** [52] plays a pivotal role in knowledge sharing and documentation. This collaborative platform empowered our team to create, edit, and organize project documentation, user stories, and meeting notes. The seamless integration with Jira ensures that our documentation remains up-to-date and veracious.

Version control and source code management are paramount in our software development efforts. **Github** [53] enabled us to maintain a well-structured codebase, supporting collaborative coding and code reviews. In addition to this, we also used **Microsoft Azure Services** [54] to share the training and evaluation code of the Deep Learning models implemented during the project. This approach ensures code quality and enables us to respond swiftly to changing requirements.

For meeting management and team coordination, we utilize **Discord** [55], chosen for its efficient voice and text chat features, as well as its capacity for creating dedicated servers that enhance collaborative communication. Additionally, **Google Workspace** [56] is integral to our workflow, providing cloud-based document, email, and calendar management services, thereby boosting our productivity and facilitating easy document sharing, meeting scheduling, and team synchronization. For document preparation, including this report, we rely on **Overleaf** [57].

7.3 Final IS Project Scheduling

In the following section, as we conclude our project, we will provide a detailed examination of its foundational aspects by breaking down the tasks involved. This retrospective analysis is crucial, as it formed the basis for all phases of the project's development. We will review the various stages and elaborate on each task, offering insights into our strategic approach and decision-making processes. This section will also highlight key milestones, effectively mapping out the project's trajectory and achievements. Additionally, we will discuss the methods and alternatives we considered, thereby shedding light on our tactical choices. This thorough evaluation is essential for understanding the structure and efficacy of our project execution.

7.3.1 Project Work Packages

We structured our project planning by dividing tasks into distinct work packages for clarity and organization. The phases we identified and completed are:

WP1 - Project Management: This phase involved defining, organizing, and overseeing the project's progression.

WP2 - Preliminary Study: Focused on developing a solid theoretical foundation in key concepts in digital music production, providing a solid theoretical base for the project.

WP3 - Viability Analysis: We assessed the project's economic and social feasibility, including a thorough risk analysis.

WP4 - Documentation: This area encompassed creating and maintaining all project documentation, such as technical documents, user manuals, and progress reports.

WP5 - Application Design: Centered on designing the application's user interface and experi-

ence, ensuring intuitiveness and aesthetic alignment with project goals.

WP6 - Implementation: The practical phase of executing the project plan, developing a basic prototype and incrementally enhancing its features.

WP7 - Testing and Evaluation: Involved rigorous testing of the project to verify its performance and functionality against the set objectives and specifications.

7.3.2 Task definition

The following section outlines the tasks that constituted our project, organized into distinct blocks corresponding to each phase. These tasks were defined in line with our initial goals, and task dependencies were established as shown in the Task Dependency Diagram (Figure 2).

Project Management - WP1

M1 - Problem Definition: We clarified the specific challenge or issue that the project aimed to address.

M2 - Study Requirements: We analyzed and documented the essential features and constraints of the project.

M3 - Define Goals: We set clear, measurable objectives for the project's outcome.

M4 - Develop a Plan: We created a comprehensive strategy, including timelines, resources, and milestones.

M5 - Conduct Weekly Meetings: We held regular sessions to discuss progress, obstacles, and plan subsequent steps.

In the Project Management phase, we started with Problem Definition (M1) to set the project's scope and focus. Following this, we tackled Study Requirements (M2) to define the project's needs. Defining Goals (M3) often occurred concurrently with M2 or immediately afterward, establishing clear project objectives. Development of a Plan (M4) hinged on understanding the problem, requirements, and goals established in M1, M2, and M3. Lastly, Conducting Weekly Meetings (M5) was an ongoing task that started with M1 and persisted throughout the project.

Preliminary Study - WP2

S1 - Research Music Source Separation: We researched techniques and methods to distinguish individual sounds within a music piece.

S2 - Research MIDI Transcription: We investigated approaches to convert audio signals into MIDI format.

S3 - Research Sheet Music Generation: We explored tools and algorithms for transforming audio or MIDI into notated music.

S4 - Analyze Prominent Studies: We reviewed leading research related to the project's focus.

During the Preliminary Study phase, tasks such as Music Source Separation (S1), MIDI Transcription (S2), Sheet Music Generation (S3), and analyzing Prominent Studies (S4) were typically conducted as separate activities. These tasks began following the completion of M1 to ensure they aligned with the project's objectives.

Viability Analysis - WP3

V1 - Identify and Analyze Risks: Detected potential challenges and evaluated their impact.

V2 - Conduct Sustainability Analysis: Assessed the long-term feasibility and ecological responsibility of the project.

V3 - Perform Economic Analysis: Examined the financial implications, including costs, benefits, and ROI.

In the Viability Analysis phase, Identifying and Analyzing Risks (V1) was initiated after completing

the Study Requirements to understand potential challenges. Conducting Sustainability Analysis (V2) and Performing Economic Analysis (V3) could start concurrently with V1.

Documentation - WP4

D1 - Develop "Definition of the Project Document": Gathered initial findings, plans, and research from the intial phase.
D2 - Develop "Midterm IS Project Document": Delivered a detailed report on the project's mid-stage achievements and challenges.
D3 - Prepare Midterm Presentation: Developed slides for the midterm showcase.
D4 - Develop "Final IS Project Document": Detailed the entire project, from inception to completion.
D5 - Create a User Guide: Produced a comprehensive manual for end-users detailing functionality and usage.
D6 - Prepare Final Presentation: Developed slides and materials for the final showcase.

The phase involved developing the "Definition of the Project Document" (D1) after M1, M2, and M3 were completed. The "Midterm IS Project Document" (D2) and the preparation of the Midterm Presentation (D3) began after significant progress had been made in the Implementation phase (I1, I2, I3). Towards the project's end, the focus shifted to developing the "Final IS Project Document" (D4), creating a User Guide (D5), and preparing the Final Presentation (D6), after most implementation (I4 to I7) and testing tasks (T1, T2, T3) were completed.

Solution Design - WP5

DS1 - Design the Architecture: We structured the system's high-level components and their interactions.
DS2 - Define Interaction Design: We outlined how users would engage with the system.
DS3 - Develop User Interface Design: We crafted visual and functional elements for user interaction.

In the Solution Design phase, we began Designing the Architecture (DS1) post M2 to ensure the system architecture aligned with the project requirements. This was followed by Defining Interaction Design (DS2) and Developing User Interface Design (DS3).

Implementation - WP6

I1 - Implement Initial *Splitter*: We developed the first version of the audio separation tool.
I2 - Implement Initial MIDI Transcription: We built the preliminary system for converting audio to MIDI.
I3 - Implement Initial Sheet Music Generation: We produced the first iteration of the music notation generator.
I4 - Postprocess *Splitter*: We refined and enhanced the audio separation tool based on feedback and testing.
I5 - Improve MIDI Transcription: We continuously enhanced the accuracy and efficiency of the MIDI transcription process.
I6 - Customize Sheet Music Generation: We optimized the conversion process from audio or MIDI to sheet music, adding features to refine the output.
I7 - Implement User Interface: We constructed interactive front-end components for user navigation and utilization.

The Implementation phase started with the initial implementation tasks (I1, I2, I3) after the Architecture Design (DS1). This was followed by subsequent improvements and post-processing

tasks (I4, I5, I6). The Implementation of the User Interface (I7) commenced after completing the User Interface Design (DS3).

Testing and Evaluation - WP7

T1 - Test Individual Components: We evaluated each module separately to ensure functionality.
T2 - Test the System: We examined the integrated system for overall performance and reliability.
T3 - Conduct User Testing: We gathered feedback from real or potential users to identify usability issues and improvement areas.

Lastly, in the Testing and Evaluation phase, we began Testing Individual Components (T1) after the initial implementation tasks (I1, I2, I3). System Testing (T2) was conducted after completing all implementation tasks (I1 to I7). User Testing (T3) was most effective after T2, ensuring the system was fully integrated and functional.

7.3.3 Milestones

The project is structured around a series of four significant milestones, each with associated deliverables and targeted deadlines. The final Gantt Chart can be visualized in Figure 3.

MS1 - Definition of the Project: The first milestone focused on the initial phase, involving the clear articulation of the project's scope and objectives. The associated deliverable for this phase was the "Definition of the Project Document" (MS1-D1), which was expected to be completed and submitted by *September 28th, 2023*. This deliverable was successfully completed and submitted by the designated deadline.

MS2 - Midterm Reporting: As the project progressed to its mid-point, a comprehensive report capturing the project definition, achievements, and challenges encountered was required. This was encapsulated in the "Midterm IS Project Document" (MS2-D2) and "Midterm Presentation" (MS2-D3). The deadline for this deliverable was *November 16th, 2023*, and it was successfully completed and submitted by this deadline.

MS3 - Final Software and Documentation: This decisive milestone marked the culmination of the project's primary objectives, where the final product was ready for delivery. Three key deliverables were associated with this phase: the "Final IS Project Document" (MS3-D4), "Create User Guide" (MS3-D5), and the "User Interface" (MS3-I7), which culminated the end of the software development. All these deliverables shared the same deadline, *January 14th, 2024*, and were completed successfully.

MS4 - Public Project Exposition and Defense: The project's journey is set to culminate with a public presentation, where the team will present and defend the project's achievements, challenges, and learnings. The preparation for this defense includes the creation of the "Final Presentation" (MS4-D6). This document will serve as the main tool for the public defense of the project, and it is due on *January 15th, 2024*.

7.4 Final Task Assignment

In our complex multifaceted project, as detailed by our Task Dependency Graph (Figure 2), clear task distribution was essential for efficiency and accountability. Our team, comprising three developers and a director, followed this structure: the director provided overarching support and feedback, while the developers focused on technical aspects. Despite individual task assignments, our team emphasized collaboration and mutual support. Table 5 succinctly illustrates our workflow, serving as both a guide and a historical record of our approach.

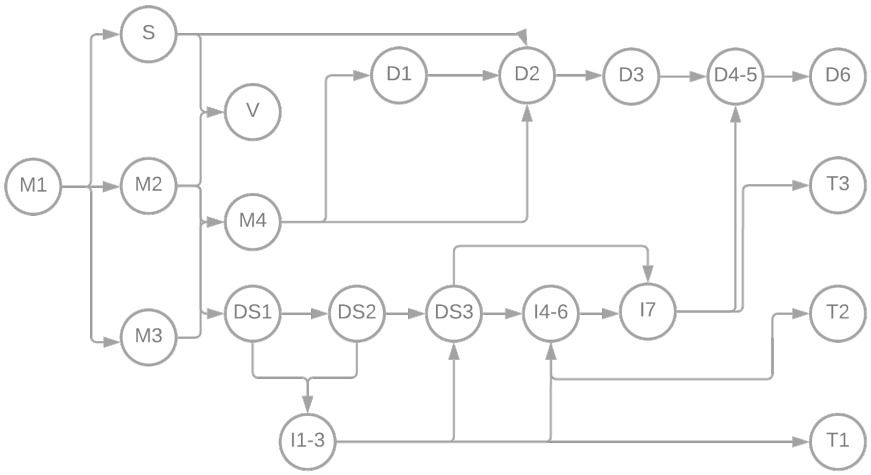


Figure 2: Graph of Task Dependencies.

ID	Task	Assigned Member	Duration (h)	Dependencies
M (WP1)	**Project Management**		**60**	
M1	Problem Definition	P1	4	
M2	Study Requirements	P2	7	M1
M3	Define Goals	P1, P2, P3	3	M1
M4	Develop a Plan	P3	10	M2, M3
M5	Conduct Weekly Meetings	P1, P2, P3	36	
S (WP2)	**Preliminary Study**		**129**	
S1	Research Music Source Separation	P3	35	M1
S2	Research MIDI Transcription	P1	35	M1
S3	Research Sheet Music Generation	P2	35	M1
S4	Analyze Prominent Studies	P1, P2, P3	24	M1
V (WP3)	**Viability Analysis**		**22**	
V1	Identify and Analyze Risks	P1, P2	14	M2, S
V2	Conduct Sustainability Analysis	P3	3	M2, S
V3	Perform Economic Analysis	P3	5	M2, S
D (WP4)	**Documentation**		**119**	
D1	Develop "Definition of the Project Document"	P1, P2, P3, D	14	M1, M2, M3
D2	Develop "Midterm IS Project Document"	P1, P2, P3, D	37	I1, I2, I3, M4, S
D3	Prepare Midterm Presentation	P1, P2, P3, D	4	D2
D4	Develop "Final IS Project Document"	P1, P2, P3, D	51	I7, T1-T3
D5	Create a User Guide	P1, P2, P3, D	6	I7, T1-T3
D6	Prepare Final Presentation	P1, P2, P3, D	7	D4
DS (WP5)	**Solution Design**		**10**	
DS1	Design the Architecture	P2	5	M2, M3
DS2	Define Interaction Design	P1	2	DS1
DS3	Develop User Interface Design	P3	3	DS2, I1, I2, I3
I (WP6)	**Implementation**		**189**	
I1	Implement Initial *Splitter*	P3	12	DS1, DS2
I2	Implement Initial MIDI Transcription	P1	12	DS1, DS2
I3	Implement Initial Sheet Music Generation	P2	12	DS1, DS2
I4	Postprocess *Splitter*	P3	34	I1, DS3, S
I5	Improve MIDI Transcription	P1	45	I2, DS3, S
I6	Customize Sheet Music Generation	P2	34	I3, DS3, S
I7	Implement User Interface	P3, P2	40	I4-I6, DS3
T (WP7)	**Testing and Evaluation**		**16**	
T1	Test Individual Components	P1, P2, P3	3	I1, I2, I3
T2	Test the System	P1	7	I1-I6
T3	Conduct User Testing	P2, P3	6	I7
Total			**545**	

Table 5: Summary of the task assignment, with their dependencies, and recorded workload. Take into consideration the dependencies are sequential, so while the dependencies on M4 are M2 and M3, we need to recursively complete all dependencies of each before we can begin M4. The final workload hours sum is computed from the top level tasks, which in turn are the sum of their corresponding low level tasks. For some tasks, the Director is needed to give feedback which we will act upon. (Benjamí - P1, Armand - P2, Gerard - P3, Miquel - D).

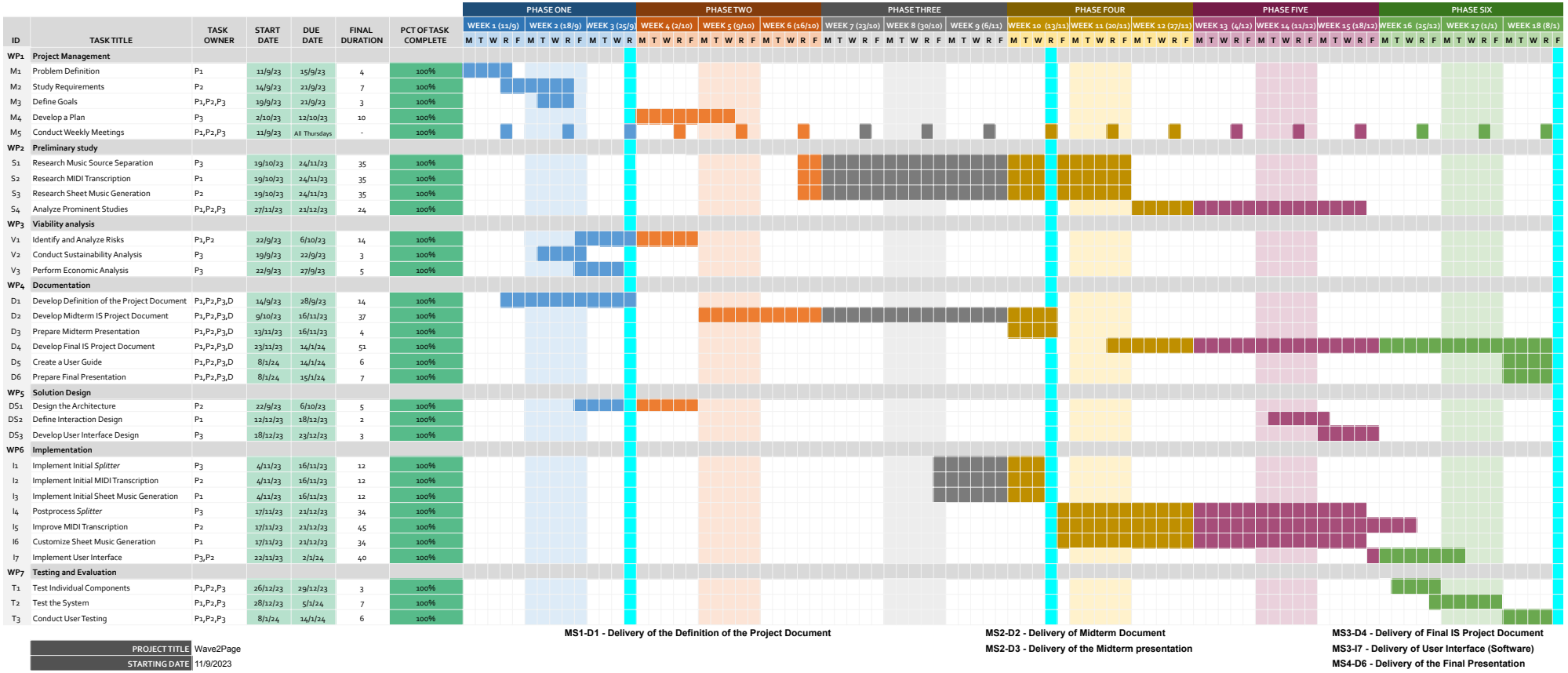


Figure 3: Gantt Diagram of the tasks. The tasks of our project are systematically visualized in a Gantt Diagram. Key milestones, as detailed in Section 7.3.3, are highlighted in Arctic Blue for ease of reference. The first significant milestone is the completion of the "Definition of the Project," marking a crucial initial stage of our work. The second milestone aligns with the completion of the "Midterm Document" and "Midterm Presentation," signifying a phase where the problem specifications have been more thoroughly refined and developed. The final milestone, marked by the last blue line, represents the deadline for submitting a copy of the "Final IS Project Document," "User Guide," and "Final Presentation" in preparation for the project defense. This deadline also includes the submission of the developed User Interface. The completion of the User Interface is contingent upon the full functionality and readiness of all backend components.

The Gantt Diagram is intuitively designed to be self-explanatory. Each square on the diagram approximates four hours of work, aligning with the task assignments detailed in Table 5. This arrangement reflects the executed workload, where the added flexibility to accommodate unforeseen delays in the planning stage was paramount to achieving an accurate and reliable timeline.

			P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3	
WorkPackage	Tasks	Subtasks	W1			TW1	W2			TW2	W3			TW3	W4			TW4	W5			TW5	W6			TW6
WP1																										
	M																									
		M1	4			4				0				0				0				0				0
		M2		5		5		2		2				0				0				0				0
		M3				0	1	1	1	3				0				0				0				0
		M4				0				0				0			7	7	3			3				0
		M5	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2
	Total M		5	6	1	11	2	4	2	7	1	1	1	2	1	1	8	9	4	1	1	5	1	1	1	2
Total WP1			5	6	1	11	2	4	2	7	1	1	1	2	1	1	8	9	4	1	1	5	1	1	1	2
WP2																										
	S																									
		S1				0				0				0				0				0			6	6
		S2				0				0				0				0				0	10			10
		S3				0				0				0				0				0		2		2
		S4				0				0				0				0				0				0
	Total S		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	2	6	18
Total WP2			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	2	6	18
WP3																										
	V																									
		V1				0	5			5	1	7		8		1		1				0				0
		V2				0			3	3				0				0				0				0
		V3				0			1	1			4	4				0				0				0
	Total V		0	0	0	0	5	0	4	9	1	7	4	12	0	1	0	1	0	0	0	0	0	0	0	0
Total WP3			0	0	0	0	5	0	4	9	1	7	4	12	0	1	0	1	0	0	0	0	0	0	0	0
WP4																										
	D																									
		D1	1	1	1	3	2	3		5	3		3	6				0				0				0
		D2				0				0				0	1	1	1	3	2		2	4	3	3		6
		D3				0				0				0				0				0				0
		D4				0				0				0				0				0				0
		D5				0				0				0				0				0				0
		D6				0				0				0				0				0				0
	Total D		1	1	1	3	2	3	0	5	3	0	3	6	1	1	1	3	2	0	2	4	3	3	0	6
Total WP4			1	1	1	3	2	3	0	5	3	0	3	6	1	1	1	3	2	0	2	4	3	3	0	6
WP5																										
	DS																									
		DS1				0		1		1		2		2		2		2				0				0
		DS2				0				0				0				0				0				0
		DS3				0				0				0				0				0				0
	Total DS		0	0	0	0	0	1	0	1	0	2	0	2	0	2	0	2	0	0	0	0	0	0	0	0
Total WP5			0	0	0	0	0	1	0	1	0	2	0	2	0	2	0	2	0	0	0	0	0	0	0	0
WP6																										
	i																									
		I1				0				0				0				0				0				0
		I1				0				0				0				0				0				0
		I3				0				0				0				0				0				0
		I4				0				0				0				0				0				0
		I5				0				0				0				0				0				0
		I6				0				0				0				0				0				0
		I7				0				0				0				0				0				0
	Total I		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total WP6			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WP7																										
	T																									
		T1				0				0				0				0				0				0
		T2				0				0				0				0				0				0
		T3				0				0				0				0				0				0
	Total T		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total WP7			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL			6	7	2	14	9	8	5	22	5	10	8	22	2	5	9	15	6	1	3	9	14	6	7	26

			P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3	
WorkPackage	Tasks	Subtasks	W7			TW7	W8			TW8	W9			TW9	W10			TW10	W11			TW11	W12			TW12
WP1																										
	M																									
		M1				0				0				0				0				0				0
		M2				0				0				0				0				0				0
		M3				0				0				0				0				0				0
		M4				0				0				0				0				0				0
		M5	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2
	Total M		1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2
Total WP1			1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2
WP2																										
	S																									
		S1			8	8			6	6			7	7			5	5			3	3				0
		S2	8			8	4			4	4			4	6			6	3			3				0
		S3		12		12		6		6		7		7		5		5		3		3				0
		S4				0				0				0				0				0	4	4	4	12
	Total S		8	12	8	28	4	6	6	16	4	7	7	18	6	5	5	16	3	3	3	9	4	4	4	12
Total WP2			8	12	8	28	4	6	6	16	4	7	7	18	6	5	5	16	3	3	3	9	4	4	4	12
WP3																										
	V																									
		V1				0				0				0				0				0				0
		V2				0				0				0				0				0				0
		V3				0				0				0				0				0				0
	Total V		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total WP3			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WP4																										
	D																									
		D1				0				0				0				0				0				0
		D2	3	3	3	9	3	3	3	9	2	2	2	6				0				0				0
		D3				0				0	2	1	1	4				0				0				0
		D4				0				0				0	2	2	2	6	2	3	2	7	3	1	2	6
		D5				0				0				0				0				0				0
		D6				0				0				0				0				0				0
	Total D		3	3	3	9	3	3	3	9	4	3	3	10	2	2	2	6	2	3	2	7	3	1	2	6
Total WP4			3	3	3	9	3	3	3	9	4	3	3	10	2	2	2	6	2	3	2	7	3	1	2	6
WP5																										
	DS																									
		DS1				0				0				0				0				0				0
		DS2				0				0				0				0				0				0
		DS3				0				0				0				0				0				0
	Total DS		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total WP5			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WP6																										
	i																									
		I1				0			4	4			4	4			4	4				0				0
		I1				0	3			3	5			5	4			4				0				0
		I3				0		4		4		3		3		5		5				0				0
		I4				0				0				0				0			9	9			8	8
		I5				0				0				0				0	12			12	13			13
		I6				0				0				0				0		7		7		7		7
		I7				0				0				0				0				0				0
	Total I		0	0	0	0	3	4	4	11	5	3	4	12	4	5	4	13	12	7	9	28	13	7	8	28
Total WP6			0	0	0	0	3	4	4	11	5	3	4	12	4	5	4	13	12	7	9	28	13	7	8	28
WP7																										
	T																									
		T1				0				0				0				0				0				0
		T2				0				0				0				0				0				0
		T3				0				0				0				0				0				0
	Total T		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total WP7			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TOTAL			12	16	12	39	11	14	14	38	13	14	15	42	13	13	12	37	18	14	15	46	21	13	15	48

			P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3		P1	P2	P3		Total
WorkPackage	Tasks	Subtasks	W13			TW13	W14			TW14	W15			TW15	W16			TW16	W17			TW17	W18			TW18	
WP1																											
	M																										
		M1				0				0				0				0				0				0	4
		M2				0				0				0				0				0				0	7
		M3				0				0				0				0				0				0	3
		M4				0				0				0				0				0				0	10
		M5	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	36
	Total M		1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	60
Total WP1			1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2	60
WP2																											
	S																										
		S1				0				0				0				0				0				0	35
		S2				0				0				0				0				0				0	35
		S3				0				0				0				0				0				0	35
		S4	2	2	2	6	2			2		2	2	4				0				0				0	24
	Total S		2	2	2	6	2	0	0	2	0	2	2	4	0	0	0	0	0	0	0	0	0	0	0	0	129
Total WP2			2	2	2	6	2	0	0	2	0	2	2	4	0	0	0	0	0	0	0	0	0	0	0	0	129
WP3																											
	V																										
		V1				0				0				0				0				0				0	14
		V2				0				0				0				0				0				0	3
		V3				0				0				0				0				0				0	5
	Total V		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
Total WP3			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22
WP4																											
	D																										
		D1				0				0				0				0				0				0	14
		D2				0				0				0				0				0				0	37
		D3				0				0				0				0				0				0	4
		D4	3	2	3	8				0				0	3	3	3	9	2	2	2	6	3	3	3	9	51
		D5				0				0				0	2	2		4			2	2				0	6
		D6				0				0				0				0				0	3	2	2	7	7
	Total D		3	2	3	8	0	0	0	0	0	0	0	0	5	5	3	13	2	2	4	8	6	5	5	16	119
Total WP4			3	2	3	8	0	0	0	0	0	0	0	0	5	5	3	13	2	2	4	8	6	5	5	16	119
WP5																											
	DS																										
		DS1				0				0				0				0				0				0	5
		DS2				0	1			1	1			1				0				0				0	2
		DS3				0				0			3	3				0				0				0	3
	Total DS		0	0	0	0	1	0	0	1	1	0	3	4	0	0	0	0	0	0	0	0	0	0	0	0	10
Total WP5			0	0	0	0	1	0	0	1	1	0	3	4	0	0	0	0	0	0	0	0	0	0	0	0	10
WP6																											
	i																										
		I1				0				0				0				0				0				0	12
		I1				0				0				0				0				0				0	12
		I3				0				0				0				0				0				0	12
		I4			7	7			5	5			5	5				0				0				0	34
		I5	8			8	5			5	7			7				0				0				0	45
		I6		8		8		7		7		5		5				0				0				0	34
		I7				0				0				0		15	12	27		5	8	13				0	40
	Total I		8	8	7	23	5	7	5	17	7	5	5	17	0	15	12	27	0	5	8	13	0	0	0	0	189
Total WP6			8	8	7	23	5	7	5	17	7	5	5	17	0	15	12	27	0	5	8	13	0	0	0	0	189
WP7																											
	T																										
		T1				0				0				0	1	1	1	3				0				0	3
		T2				0				0				0	2			2	5			5				0	7
		T3				0				0				0				0				0		3	3	6	6
	Total T		0	0	0	0	0	0	0	0	0	0	0	0	3	1	1	5	5	0	0	5	0	3	3	6	16
Total WP7			0	0	0	0	0	0	0	0	0	0	0	0	3	1	1	5	5	0	0	5	0	3	3	6	16
TOTAL			14	13	13	39	9	8	6	22	9	8	11	27	9	22	17	47	8	7	13	28	7	9	9	24	545

Figure 4: Time Sheet: This image displays a comprehensive breakdown of the weekly hours dedicated to the project by each of the three developers, categorized by task. It cumulatively totals 545 hours, with an equitable distribution of approximately 181 hours per developer. (Benjamí - P1, Armand - P2, Gerard - P3, Miquel - D)

8 User Manual

This section provides a concise yet comprehensive guide on using our system and its services. We start with an overview of the system's primary functions and then explore the additional features that enhance the overall user experience and complement the product's quality.

8.1 Description of the Intelligent System's Purpose

The purpose of the intelligent system is to allow the user to convert their favorite songs into readable and comprehensible music sheet for each one of the instruments involved in the performance. The focus is on popular music, thus, we limit the instruments to *vocals*, *piano*, *guitar*, *bass*, and *drums*. We offer an intuitive interface for various musical operations, from the primary functionality of sheet music conversion to auxiliary features like MIDI conversion and integration with YouTube.

8.2 Access to the system

Our system is hosted on a web page developed using Flask [58], a Python web framework known for its simplicity and minimal code requirement. For usage instructions, refer to *README.md*. Once the necessary packages are installed, launch the web page by running "`python3 app/app.py`" and access it through the IP displayed in the terminal. We encourage users to access the app through Google Chrome [59], which is where we developed and tested all the functionalities. Moreover, it is the only explorer that allows the incorporation of our web extension, which allows using our system directly from any YouTube video. We will talk more about this extension in Section 8.3.3.

8.3 Functionalities of the system

All of our functionalities have been integrated into an easy-to-use webpage. The website features four sections with different purposes: *Home* main interface; *About* provides user-level details of the system; *Contact* allows users to contact us, and *Services*, with the functionalities described next.

8.3.1 Conversion to music sheet

The centerpiece of our system is the conversion of user-uploaded WAV audio files into high-quality, coherent music sheets. We prioritize excellence in this process. As the conversion takes place, a dynamic loading animation keeps the user informed that their song is still being processed.

Figure 5 illustrates the selection of instruments available for the conversion process. The time required for conversion varies based on the number of instruments selected. Notably, the interface includes a premium option, delineating the differences in output between free and premium users. Free users receive watermarked music sheets without lyrics transcription, while premium users enjoy additional benefits (Figure 8). Upon completion of the conversion, a download button enables users to easily access and use their custom-generated music sheets.

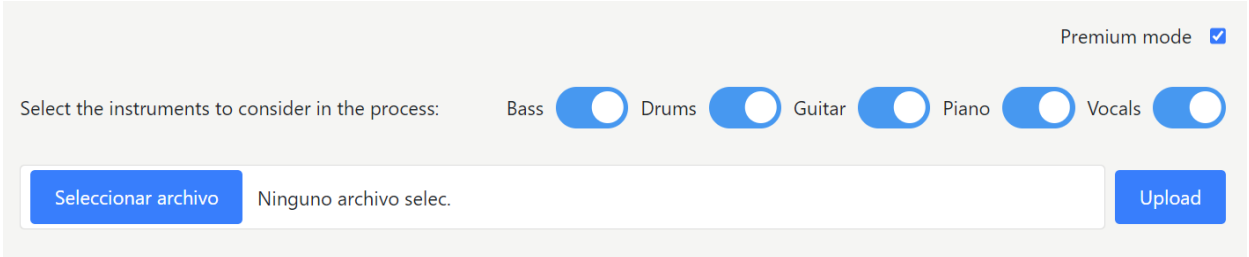


Figure 5: Main functionality from the Home Page: From audio to music sheet conversion.

8.3.2 Stem Separation

Have you ever wished to isolate specific parts of a song, like hearing it without vocals, focusing solely on the piano to learn its chords, or experimenting with unique combinations? Our stem separation feature makes this possible. Now, you can enjoy your favorite songs in a new way by selectively muting or adjusting the volume of different components. This is enabled by our system's ability to separate a song into distinct instruments, thanks to the *Splitter* module. Users can manipulate the track volumes and download them separately for personalized musical exploration (Figure 6).

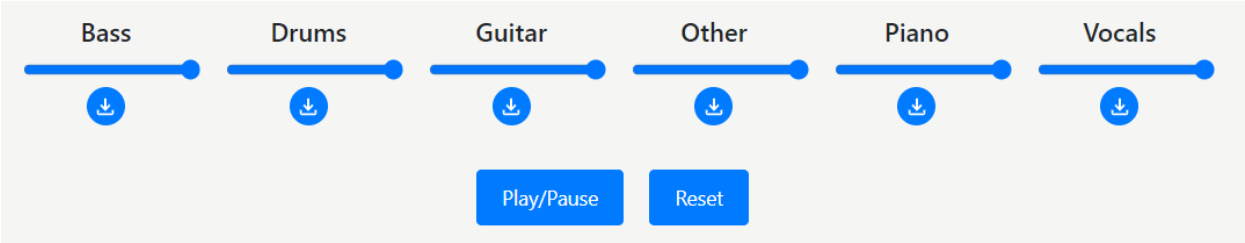


Figure 6: Stem Separation interface in the webpage.

8.3.3 YouTube Integration and Chrome Extension

Our system includes a feature that enables users to input a YouTube URL directly on our webpage. By entering an audio name, which the system will use and save, users can initiate the conversion process by clicking *Convert*. A loading widget will appear, and shortly after, the music sheet will be displayed alongside a download button for obtaining the generated sheets.

Additionally, we developed a Chrome extension (see Figure 9) which enhances the integration of our service with YouTube. When a user is browsing YouTube and decides to convert a video, they can simply enter a filename and click *GO*. This action seamlessly redirects them to our webpage's YouTube conversion service, where the process of transforming the video into music sheet begins. To ensure efficiency and avoid potential memory issues, we recommend not using excessively long videos for this conversion.

8.3.4 Audio to MIDI conversion

The final feature of our system is the conversion of audio files to MIDI format, and allowing users to download the MIDI file. This process is straightforward, as the music sheet generation initially involves converting the song to MIDI. Therefore, we offer users easy access to this intermediate MIDI file, simplifying the conversion process and adding significant value.

8.4 Examples of Use

We have already presented examples of the music sheet conversion process for both free and premium user plans. Additionally, in the files sent, we have added some more examples of the output of our system. Figure 8 illustrates the crispness and harmony of the notes in the generated sheets. The premium plan additionally offers song transcription, as showcased in Figure 7.

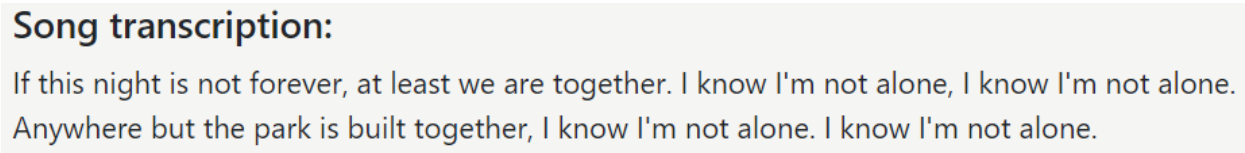
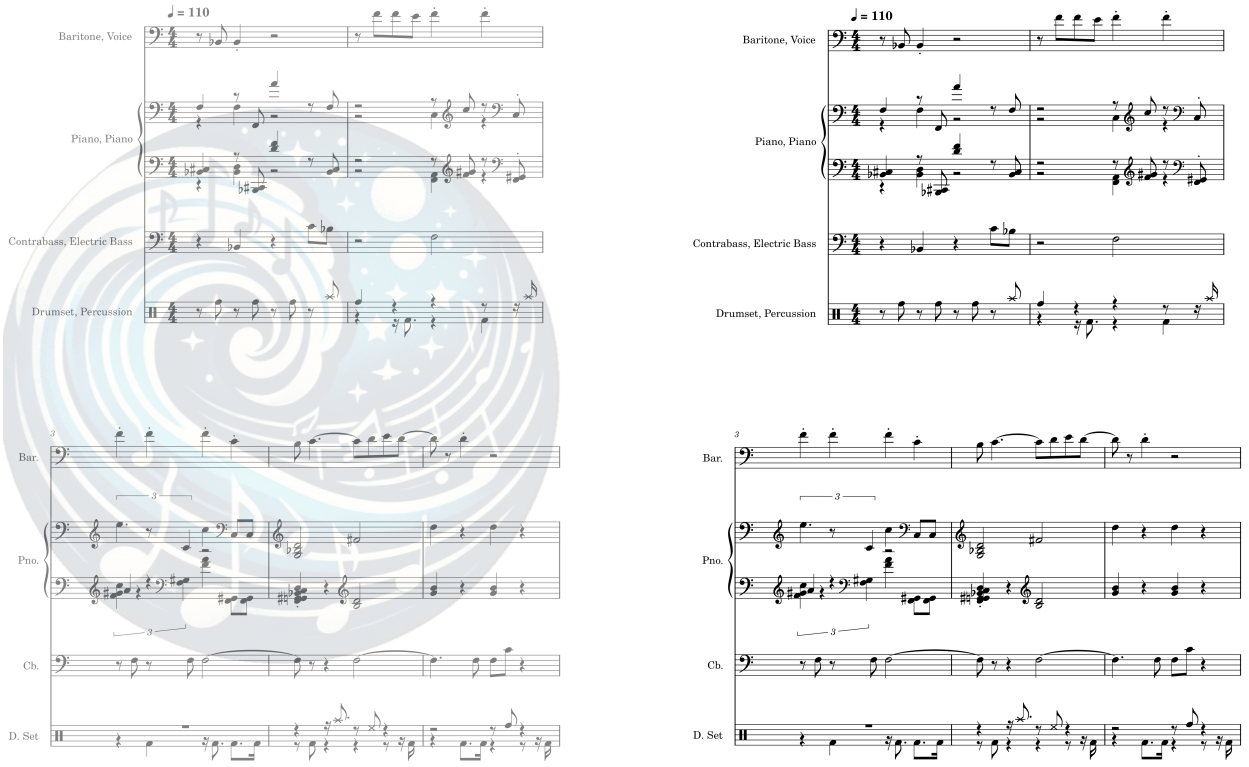


Figure 7: Example of lyric transcription with the premium plan.



(a) Generated music sheet for Free user (b) Generated music sheet for Premium user

Figure 8: Music sheet generated examples with the Free and Premium plans respectively.

8.5 Interactions of the system

In this section, we briefly explain some of the key interactions with respect to the input and output data provided to the webpage, and the list of error messages that one can encounter.

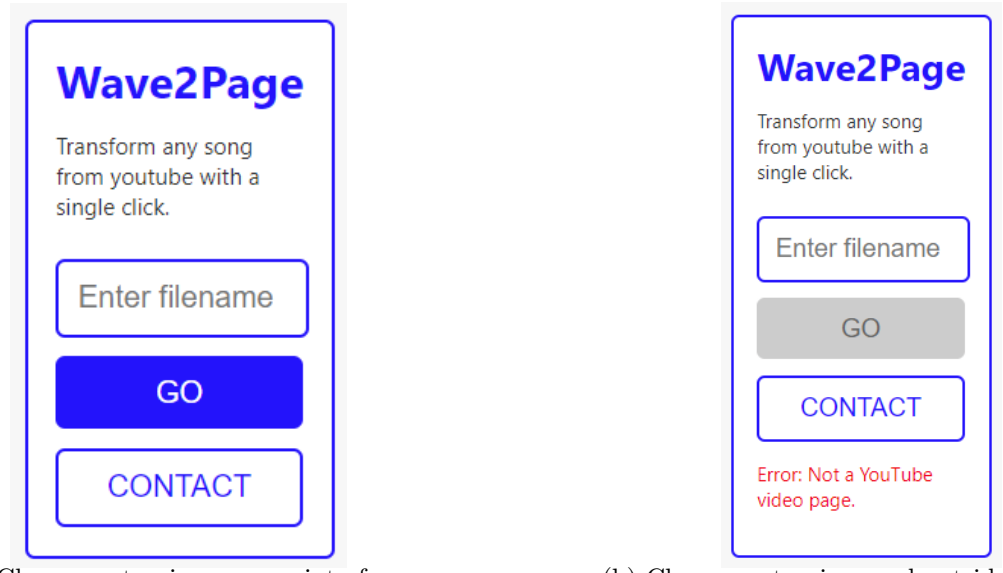
8.5.1 Input/Output

Our system's functionalities fall into two categories: those requiring a song input and those that do not. For song inputs, we exclusively use the WAV format, ideal for our needs due to its uncompressed nature. Non-song functionalities include the YouTube service, which requires a YouTube link, and audio divisibility, accessible after saving a song through other functionalities. To facilitate integration with the website, outputs are generated as PNG music sheets, one for each page of the score. This format was chosen for its portability and ease of use. To facilitate downloads, we package these files into a single ZIP file.

8.5.2 List of Error Messages

Our webpage is designed to minimize user-facing errors by restricting the upload of specific data types. In rare cases of conversion failure, an error message is displayed. Additionally, we provide warnings or alerts to enhance user experience. For instance, if certain instruments are not detected in a song or filtered out due to silence, users are notified that these instruments won't be included in the conversion.

For our Chrome extension, operation is limited to YouTube videos. If a user attempts to use it elsewhere, the error shown in Figure 9b appears. This approach eliminates most issues; if a filename isn't provided, the system uses the sanitized original song title.



(a) Chrome extension pop-up interface (b) Chrome extension used outside YouTube

Figure 9: Chrome extension in two different scenarios

9 Discussion

9.1 Conclusions

Wave2Page, our innovative intelligent system, marks a significant advancement in musical transcription technology. Designed to convert multi-instrument audio recordings into digital sheet music, it addresses critical gaps in the music industry with its array of unique features and functionalities. The system's ability to handle a diverse range of instruments, including vocals, guitar, bass, piano, and percussion, positions it as a versatile tool in various popular musical genres.

In the development of Wave2Page, we have rigorously tackled several tasks, from BPM and time-signature detection to sophisticated MIDI conversion and sheet music generation. Our approach to audio track separation using state-of-the-art models, enhanced by our novel silent track detection method, showcases our commitment to accuracy and efficiency, enhancing the final output. The subsequent MIDI conversion process, especially our unique method of handling percussion elements and vocal transcription, demonstrates our system's ability to deal with complex audio elements.

However, the quest to generate human-like, nuanced transcriptions remains a challenging endeavor. Our current methodology, though efficient, may not fully capture the depth and subtleties of musical expression. Musical transcription is an inherently subjective art, often leading to varied interpretations of the same piece by different musicians or transcribers [60]. This variability, underscored in studies on musical interpretation, highlights the complex and artistic nature of musical transcription and necessitates further exploration and refinement in our approach.

In line with this, it's important to acknowledge the limitations inherent in our system. Currently, our system supports only a limited range of instruments and does not accommodate multiple instances of the same instrument, which can be a significant limitation in complex musical compositions. Additionally, there is a noticeable discrepancy at times between the input audio and the final output, as indicated by feedback from our specialized focus group. Furthermore, the post-processing and cleanup of MIDI files to produce more polished sheet music is an area that can be significantly improved. Enhancements could be achieved either by expanding the rules in our rule-based system or by developing more sophisticated algorithms, possibly utilizing Reinforcement Learning, to learn how to refine MIDI files for better presentation. Specializing these rules for each instrument could also enhance the alignment performance.

Additionally, the absence of a reliable alignment metric in our context is a notable gap. Although we have identified two potential metrics during our research [22, 23], they don't seem suitable for our specific needs, as they presuppose a static version of the correct sheet music. This presents an opportunity for further research and development in creating or adapting metrics that can more effectively measure the alignment in dynamic and variable musical transcriptions.

On the economic and sustainability front, Wave2Page demonstrates a robust and feasible model. Our comprehensive economic analysis and thoughtful pricing strategy, coupled with a keen understanding of market demands and competitor offerings, have positioned us well to enter the industry. Additionally, our system offers more functionalities at a competitive price point compared to similar products in the market, ensuring strong market acceptance and future growth.

Furthermore, our sustainability analysis reveals a deep commitment to environmental and social responsibility. Our use of energy-efficient cloud services and sustainable practices aligns with our goal to minimize our ecological footprint. The project's social impact, especially in terms of inclusivity and accessibility in music transcription, further underscores our dedication to contributing positively to the music community and society at large.

9.2 Future Work

The Wave2Page project presents substantial opportunities for future enhancements, which can be broadly categorized into two domains: technical advancements and user experience improvements.

Technical Advancements:

- **Enhancing MIDI Conversion Process**: Conducting research on advanced methods for musical alignment and training more potent deep learning models could substantially improve the MIDI conversion process. This approach involves not just optimizing current algorithms but potentially incorporating innovative, more effective methodologies.
- **Expanding Instrument Range**: Adding a wider array of instruments to the system would not only enhance the richness of the output but also broaden the appeal and applicability of the product to a larger user base.
- **Lyrics Alignment**: One of the more challenging yet impactful improvements could be the precise alignment of lyrics with the music. This would greatly enhance the user experience by providing more coherent and comprehensive music sheets. Although we have developed this feature, its current level of accuracy does not meet our standards for deployment.

User Interaction and Visual Quality Enhancements:

- **Webpage Development**: As non-expert frontend developers, we acknowledge the potential for significant improvements in the website's design and functionality. Enhancing the webpage's responsiveness, interactivity, and aesthetic appeal is crucial for user engagement.
- **Informative Loading Features**: Implementing more informative loading indicators, such as progress bars with estimated completion times, would significantly improve the user experience by providing clarity on the status of their requests.
- **Application Deployment**: Prior to deployment, adding diverse payment options and enhancing user control features is essential for a streamlined and user-friendly experience.

These future tasks, ranging from technical improvements to user interface enhancements, are essential for evolving Wave2Page into a more robust, efficient, and user-friendly system. Each step will contribute to the overall goal of establishing Wave2Page as a leading solution in musical transcription technology.

References

[1] Emmanouil Benetos et al. "Automatic music transcription: challenges and future directions". In: *Journal of Intelligent Information Systems* 41 (2013), pp. 407–434.

[2] Emmanouil Benetos et al. "Automatic music transcription: An overview". In: *IEEE Signal Processing Magazine* 36.1 (2018), pp. 20–30.

[3] Yu-Te Wu et al. "Omnizart: A General Toolbox for Automatic Music Transcription". In: *Journal of Open Source Software* 6 (Dec. 2021), p. 3391. DOI: 10.21105/joss.03391.

[4] Li Su and Yi-Hsuan Yang. "Combining spectral and temporal representations for multipitch estimation of polyphonic music". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.10 (2015), pp. 1600–1612.

[5] Rainer Kelz et al. "On the potential of simple framewise approaches to piano transcription". In: *arXiv preprint arXiv:1612.05153* (2016).

[6] Zhiyao Duan, Bryan Pardo, and Changshui Zhang. "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.8 (2010), pp. 2121–2133.

[7] Paul H Peeling, A Taylan Cemgil, and Simon J Godsill. "Generative spectrogram factorization models for polyphonic piano transcription". In: *IEEE transactions on audio, speech, and language processing* 18.3 (2009), pp. 519–527.

[8] Juhan Nam et al. "A Classification-Based Polyphonic Piano Transcription Approach Using Learned Feature Representations." In: *Ismir*. Citeseer. 2011, pp. 175–180.

[9] Zhiyao Duan, Jinyu Han, and Bryan Pardo. "Multi-pitch streaming of harmonic sound mixtures". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.1 (2013), pp. 138–150.

[10] Y.-N. Hung, G. Wichern, and J. Le Roux. "Transcription is all you need: Learning to separate musical mixtures with score as supervision". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 46–50.

[11] A. Jansson et al. "Singing voice separation with deep u-net convolutional networks". In: *ISMIR* (2017).

[12] L. Lin et al. "A unified model for zero-shot music source separation, transcription, and synthesis". In: *arXiv preprint arXiv:2108.03456* (2021).

[13] E. Manilow, P. Seetharaman, and B. Pardo. "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 771–775.

[14] *ScoreCloud - Play your music – ScoreCloud notates!* 2023. URL: https://scorecloud.com/.

[15] *MuseScore - The world's most popular notation app*. URL: https://musescore.org/es.

[16] *AnthemScore 4 - Music AI for Your PC*. URL: https://www.lunaverus.com/.

[17] *Noteflight*. URL: https://www.noteflight.com/.

[18] Kin Wai Cheuk et al. "Jointist: Joint learning for multi-instrument transcription and its applications". In: *arXiv preprint arXiv:2206.10805* (2022).

[19] Kin Wai Cheuk et al. “Jointist: Simultaneous Improvement of Multi-instrument Transcription and Music Source Separation via Joint Training”. In: *arXiv preprint arXiv:2302.00286* (2023).

[20] Keitaro Tanaka et al. “Multi-Instrument Music Transcription Based on Deep Spherical Clustering of Spectrograms and Pitchgrams.” In: *ISMIR*. 2020, pp. 327–334.

[21] Yu-Te Wu, Berlin Chen, and Li Su. “Polyphonic music transcription with semantic segmentation”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 166–170.

[22] Thitaree Tanprasert et al. “Midi-sheet music alignment using bootleg score synthesis”. In: *arXiv preprint arXiv:2004.10345* (2020).

[23] Ruchit Agrawal, Daniel Wolff, and Simon Dixon. “Structure-aware audio-to-score alignment using progressively dilated convolutional neural networks”. In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 571–575.

[24] Simon Rouard, Francisco Massa, and Alexandre Défossez. “Hybrid Transformers for Music Source Separation”. In: *ICASSP 23*. 2023.

[25] Alexandre Défossez. “Hybrid Spectrogram and Waveform Source Separation”. In: *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*. 2021.

[26] Andrew Ng, Michael Jordan, and Yair Weiss. “On spectral clustering: Analysis and an algorithm”. In: *Advances in neural information processing systems* 14 (2001).

[27] Daniel Stoller, Sebastian Ewert, and Simon Dixon. *Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation*. 2018. arXiv: 1806.03185 [cs.SD].

[28] Rachel M. Bittner et al. “A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 781–785. DOI: 10.1109/ICASSP43922.2022.9746549.

[29] Jong Wook Kim et al. *CREPE: A Convolutional Representation for Pitch Estimation*. 2018. arXiv: 1802.06182 [eess.AS].

[30] Jesse Engel et al. “DDSP: Differentiable digital signal processing”. In: *arXiv preprint arXiv:2001.04643* (2020).

[31] Brian McFee et al. *librosa/librosa: 0.10.1*. Version 0.10.1. Aug. 2023.

[32] Jeremiah Abimbola, Daniel Kostrzewa, and Pawel Kasprowski. “Time Signature Detection: A Survey”. In: *Sensors* 21.19 (2021), p. 6494.

[33] Matthias Varewyck, Jean-Pierre Martens, and Marc Leman. “Musical meter classification with beat synchronous acoustic features, DFT-based metrical features and support vector machines”. In: *Journal of New Music Research* 42.3 (2013), pp. 267–282.

[34] Trevor De Clercq and David Temperley. “A corpus analysis of rock harmony”. In: *Popular Music* 30.1 (2011), pp. 47–70.

[35] Simon Rouard, Francisco Massa, and Alexandre Défossez. “Hybrid Transformers for Music Source Separation”. In: *ICASSP 23*. 2023.

[36] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. “Real time speech enhancement in the waveform domain”. In: *arXiv preprint arXiv:2006.12847* (2020).

[37] Zafar Rafii et al. “MUSDB18-a corpus for music separation”. In: (2017).

[38] Richard Vogl, Gerhard Widmer, and Peter Knees. "Towards multi-instrument drum transcription". In: *arXiv preprint arXiv:1806.06676* (2018).

[39] Beth Logan et al. "Mel frequency cepstral coefficients for music modeling." In: *Ismir*. Vol. 270. 1. Plymouth, MA. 2000, p. 11.

[40] Alec Radford et al. *Robust Speech Recognition via Large-Scale Weak Supervision*. 2022. DOI: 10.48550/ARXIV.2212.04356. URL: https://arxiv.org/abs/2212.04356.

[41] Colin Raffel. *The lakh midi dataset v0. 1*. 2016.

[42] Bo PENG. *RWKV-LM*. Version 1.0.0. Aug. 2021. DOI: 10.5281/zenodo.5196577. URL: https://github.com/BlinkDL/RWKV-LM.

[43] Indeed. *Indeed salary estimator*. https://es.indeed.com/career/. 2023.

[44] *Azure Machine Learning Pricing*. Accessed: 2023-12-10. 2023. URL: https://azure.microsoft.com/en-us/pricing/details/machine-learning/.

[45] GoDaddy. *GoDaddy Official Website*. https://www.godaddy.com/. Accessed: 2023-12-20. 2023.

[46] *Automatic Drum Transcription - Drumscrib*. https://drumscrib.com/. Accessed: 2024-01-02. 2024.

[47] *Study: Carbon, energy efficiency benefits of the Microsoft cloud*. Accessed: 2023-11-13. 2018. URL: https://www.microsoft.com/en-us/download/details.aspx?id=56950.

[48] Wholegrain Digital. *Website Carbon Calculator v3 — What's your site's carbon footprint?* 05. URL: https://www.websitecarbon.com/.

[49] *What Is The Carbon Footprint Of A Laptop?* Accessed: 2023-12-03. 2023. URL: https://circularcomputing.com/news/carbon-footprint-laptop/.

[50] Atlassian. *Jira blog about Scrum methodology*. https://www.atlassian.com/es/agile/scrum. 2023.

[51] Atlassian. *Jira product website*. https://www.atlassian.com/es/software/jira. 2023.

[52] Atlassian. *Confluence product website*. https://www.atlassian.com/es/software/confluence. 2023.

[53] Github. *Github official website*. https://github.com/. 2023.

[54] Miscrosoft. *Miscrosoft Azure introductory tutorial*. https://azure.microsoft.com. 2023.

[55] Discord Inc. *Discord official website*. https://www.discord.com/. 2023.

[56] Google. *Google Workspace blog*. https://workspace.google.com/. 2023.

[57] Overleaf. *Overleaf Documentation*. https://www.overleaf.com/learn. 2023.

[58] Armin Ronacher and contributors. *Flask Documentation*. Accessed: 2023-11-6. Pallets Projects. https://flask.palletsprojects.com/.

[59] Google Chrome. *Google Chrome official website*. https://www.google.com/chrome/. 2023.

[60] Hendrik Vincent Koops et al. "Annotator subjectivity in harmony annotations of popular music". In: *Journal of New Music Research* 48.3 (2019), pp. 232–252.