

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
SUMMER 2023**



**MELODY MASTERS
SOUNDSYNC**

**ANGEL AGUIRRE
BENJAMIN FARMER
PATRICK FERGUSON
EDGAR HERNANDEZ
ADRIAN RAMOS**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	07.17.2023	GH	document creation
0.2	07.26.2023	AA, BF, PF, EH, AR	complete draft V1

CONTENTS

1	Introduction	5
2	System Overview	6
2.1	Database	6
2.2	UI	6
2.3	Sheet Scanner	6
2.4	Sheet Music tracker	7
2.5	Note Recognition	7
3	Subsystem Definitions & Data Flow	8
4	UI Subsystems	9
4.1	Login/Signup	9
4.2	Homepage	10
4.3	Settings	11
4.4	Personal Sheet Library	12
4.5	New Sheet Image Import	13
4.6	Playing Mode	14
4.7	Post-Play Statistics	16
5	Database Subsystems	18
5.1	Account Information/Maintenance	18
5.2	Account Storage	19
6	Sheet Music Tracker Subsystems	21
6.1	Pre-Play Configuration	21
6.2	Page Turner/Scroller/Tracker	22
6.3	Sheet-To-Audio Comparer	23
7	Sheet Music Scanner Subsystems	25
7.1	Image Import	25
7.2	Optical Music Recognition	26
7.3	File Translator/Compiler	26
7.4	Data Formatter	27
8	Note Recognition Subsystems	29
8.1	Audio Recorder	29
8.2	Data Processing	29
8.3	Note Profiler	30
9	Manual Page Turning Pedal	32
9.1	Pedal	32

LIST OF FIGURES

1	A simple architectural layer diagram	6
2	Data flow diagram	8
3	Example subsystem description diagram	9
4	Example subsystem description diagram	10
5	Example subsystem description diagram	12
6	Example subsystem description diagram	13
7	Example subsystem description diagram	14
8	Example subsystem description diagram	15
9	Example subsystem description diagram	17
10	Account Info/Maintenance Subsystem	18
11	Account storage subsystem	19
12	Pre-Play Configuration diagram	21
13	Example subsystem description diagram	22
14	Example subsystem description diagram	24
15	Image import diagram	25
16	OMR diagram	26
17	File Translator/Compiler diagram	27
18	Data Formatter diagram	28
19	Audio recorder diagram	29
20	Data Processing diagram	30
21	Note profiler diagram	31
22	Pedal diagram	32

LIST OF TABLES

2	Subsystem interfaces	10
3	Subsystem interfaces	11
4	Subsystem interfaces	12
5	Subsystem interfaces	13
6	Subsystem interfaces	14
7	Subsystem interfaces	16
8	Subsystem interfaces	17
9	Subsystem interfaces	19
10	Subsystem interfaces	20
11	Subsystem interfaces	22
12	Subsystem interfaces	23
13	Subsystem interfaces	24
14	Subsystem interfaces	25
15	Subsystem interfaces	26
16	Subsystem interfaces	27
17	Subsystem interfaces	28
18	Subsystem interfaces	29
19	Subsystem interfaces	30
20	Subsystem interfaces	31
21	Subsystem interfaces	32

1 INTRODUCTION

SoundSync is a mobile application with the main goal of eliminating the need to manually turn a sheet music page when playing an instrument. SoundSync will allow its users to scan a physical piece of sheet music that will then be copied into a digital format that the user can then save to their account. Using this digital sheet music, SoundSync will also utilize audio recording to detect which notes are being played by the musician in order to keep track of where the user is on the page. Once the user is close to playing the notes displayed on the screen, the application will automatically display the next page to be played for them. If this automatic note detection fails, the user will have the option to use our bluetooth pedal to turn the page manually, but still hands-free. In order to perform these tasks, SoundSync will utilize cloud services, live audio processing techniques, and optical character recognition software.

2 SYSTEM OVERVIEW

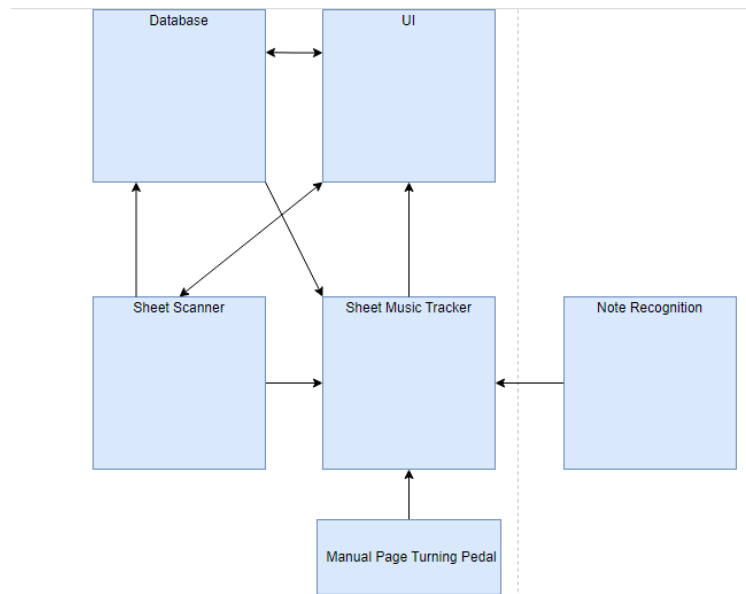


Figure 1: A simple architectural layer diagram

2.1 DATABASE

The database layer is going to be our Firebase database which will be responsible for storing all account information. When a user uses our app they will need to login or create an account. The login confirmation and account creation will be done through Firebase. It will store the user login and account details. It will also store all sheet scans the user has done. This will allow the user to load previously scanned music sheets from Firebase on to the app.

2.2 UI

The UI will be the overall interface of the app. It will include several screens starting with a login screen, Create Account screen, Forgot Password screen, and a Home Page screen. The user will start in the login screen and go through the process of logging in or creating an account. If they forgot their password they will be prompted to another screen to enter their email associated with their account. Once the user has logged in they will be directed to the home screen. From here they will have tabs at the bottom of the screen to display different information. This will include the scan tab, view tab, and Account information tab. The scan tab will allow the user to select an image or PDF of a music sheet from their mobile device to be displayed. Or the user can import a previously scanned music sheet from Firebase. Once a music sheet has been scanned the user will be redirected to the view tab which will display their music sheet. This is where the user will play their music and have the music sheet follow along. The Account Information tab will allow the user to change information on their account.

2.3 SHEET SCANNER

The sheet scanner will be the part of the app that does all the work in scanning PDFs or images and displaying them properly in the view tab. It will accept imported documents from Firebase or from the user's mobile device. It will then push these to an API we have that uses Audiveris to scan the music sheet and pick out all the musical notes. It will then format these musical notes into a display in the view tab so the user can see their music sheet and begin to play.

2.4 SHEET MUSIC TRACKER

The sheet music tracker is the part of the system that will take in the contextual inputs from the sheet scanner and the note recognition module in order to identify the current position on the digital music sheet that the user is playing on. Taking in these inputs, the sheet music tracker will generate the current context for the UI's 'playing mode' screen, showing the user exactly where the system thinks they are when they are playing. This tracker will then be used to direct when to turn the page. This page-turning will also listen for the foot pedal, just in case the musician wants to manually turn the page. The system will also be highly configurable, allowing the user to fill in any potentially missing info from their scanned sheet, such as tempo.

2.5 NOTE RECOGNITION

The note recognition module will take in the user's live playing as audio input, using an audio recorder. With this live audio input, this layer will constantly perform Fourier transformations and note-to-frequency comparison algorithms in order to identify what note or group of notes is being played at any given time. This system will then send this contextual information to the sheet music tracker.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

This is a diagram of the overall structure of our app with it included subsystems for each layer.

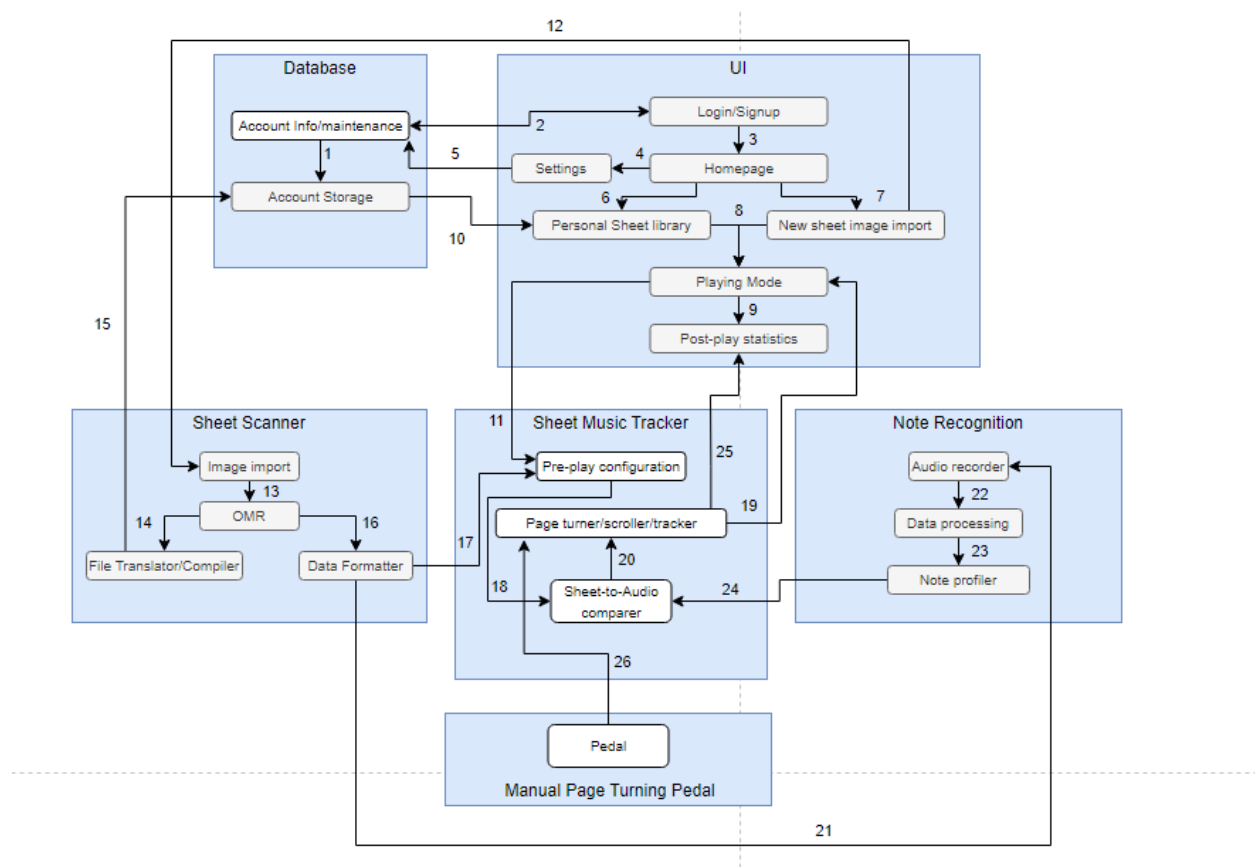


Figure 2: Data flow diagram

4 UI SUBSYSTEMS

4.1 LOGIN/SIGNUP

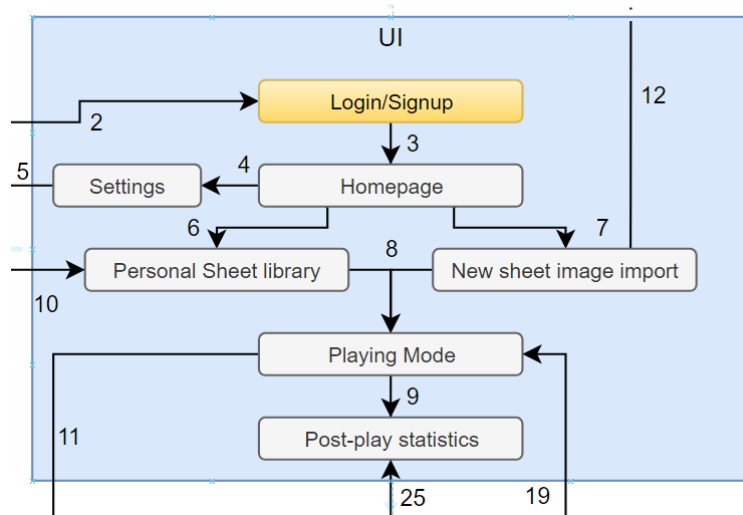


Figure 3: Example subsystem description diagram

4.1.1 ASSUMPTIONS

- The user will first be presented with a login page if the system does not remember them.
- The sign-up page will be a pressable option on the login page
- When a user signs up, the system will store their credentials in a database and designate a place for them to store all user info.

4.1.2 RESPONSIBILITIES

The Login/Signup subsystem will be responsible for allowing any user to create a new account or use one they have previously created. This subsystem is vital for allowing the user a way to verify who they are so that the system knows what saved data to send to them.

4.1.3 SUBSYSTEM INTERFACES

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#2	The subsystem will authenticate the user's entered credentials or create new ones, depending on what has been stored in the account info subsystem.	Account Info	Login/Signup Page Info
#3	Upon successful account creation or authentication, the subsystem will redirect the user to the homepage with their relevant data, if any.	N/A	UI redirect to Homepage

4.2 HOMEPAGE

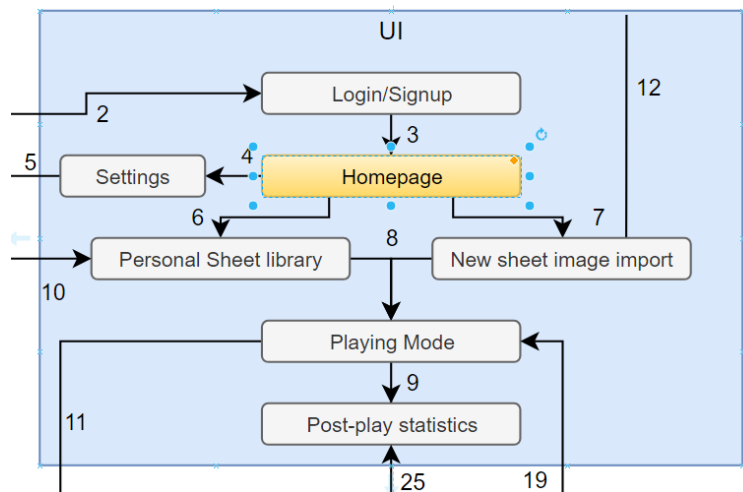


Figure 4: Example subsystem description diagram

4.2.1 ASSUMPTIONS

- The user will first be presented with a section containing any previously saved sheets to their account, as well as a section allowing them to scan a new sheet.
- The subsystem will also contain a button that allows the user to change their account preferences and information.

4.2.2 RESPONSIBILITIES

The Homepage subsystem will be responsible for showing the user what pages they have scanned previously in an easily browseable format. This homepage will act as a sort of dashboard and is the main UI component the user will see when they are not actively playing an instrument. This homepage will

be where the user can visually see what data has been stored/generated from their use, such as sheets, performance data, and account info.

4.2.3 SUBSYSTEM INTERFACES

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#3	The subsystem will be entered through the login page once the user has signed in with an existing account.	A login verification from the login screen	N/A
#4	The subsystem will allow the user to navigate to the settings page, where they can use an interface to change account preferences and information.	N/A	UI redirect to Settings
#6	The subsystem will allow the user to navigate to their personal sheet library, where all previously scanned pieces of sheet music are stored on their account.	N/A	UI redirect to Personal Sheet library
#7	The subsystem will allow the user to navigate to the New sheet image import page, where the note sheet scanning functionality will be available for the user to use in order to make a digital copy.	N/A	UI redirect to New Sheet image import

4.3 SETTINGS

4.3.1 ASSUMPTIONS

- The user will be able to use various settings, such as changing a password, logging out of their account, or giving preset info about their preferences towards the performance mode.

4.3.2 RESPONSIBILITIES

The settings subsystem will be responsible for allowing the user to configure their account information and settings, giving them more control over their experience with the application.

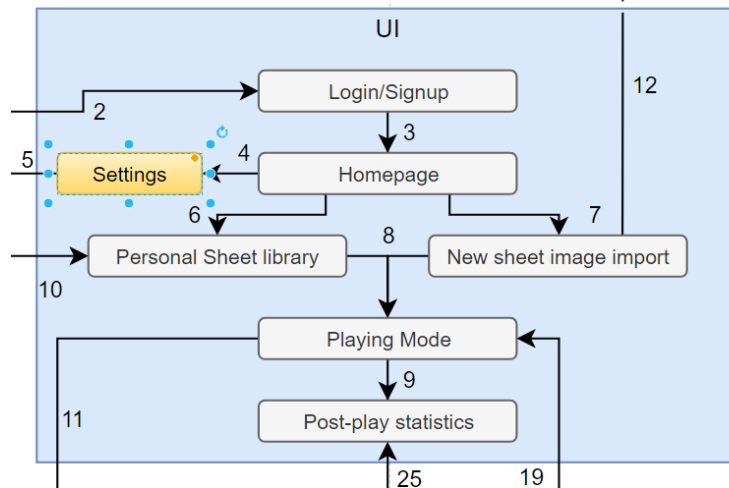


Figure 5: Example subsystem description diagram

4.3.3 SUBSYSTEM INTERFACES

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#4	The settings page will be accessible from a button on the Homepage	user button input from the home-page	N/A
#5	The settings page will be able to modify account information in the database, depending on what the user selects or changes from settings.	N/A	account info modifications

4.4 PERSONAL SHEET LIBRARY

4.4.1 ASSUMPTIONS

- The user will be able to browse their previously scanned sheets, giving them an easy interface to quickly select what song they want to play from their library.

4.4.2 RESPONSIBILITIES

The Personal Sheet library subsystem will be responsible for organizing and presenting the previously scanned sheets to the user in a way that is both aesthetically pleasing and practical. When a musical piece is selected from the library, the subsystem should redirect the user to Playing Mode.

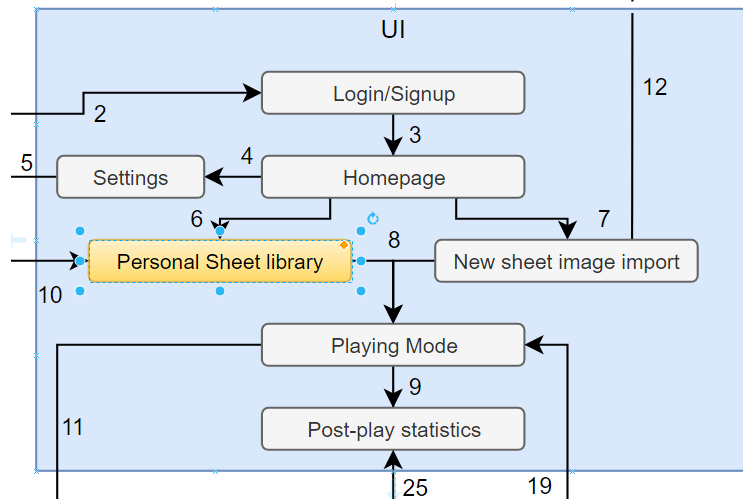


Figure 6: Example subsystem description diagram

4.4.3 SUBSYSTEM INTERFACES

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#6	The subsystem will be accessible from the Homepage by a button press.	The homepage personal sheet library button	N/A
#8	The subsystem will be able to redirect the user to the playing mode when the user selects one of their saved musical pieces.	N/A	UI redirect to Playing Mode selected sheet music data
#10	The subsystem will derive what musical pieces are available from what the user has saved in their account's database location.	Musical Sheet data	N/A

4.5 NEW SHEET IMAGE IMPORT

4.5.1 ASSUMPTIONS

- The user will be able to interact with the Sheet Scanner layer via an interface with instructions of use. These instructions will allow the user to import an image on their device's image library, allowing them to provide the input for the sheet scanner.

4.5.2 RESPONSIBILITIES

The New sheet image import subsystem is responsible for providing the user a way to take their real sheet music and digitally upload it to our application. This subsystem will take an image import, confirm

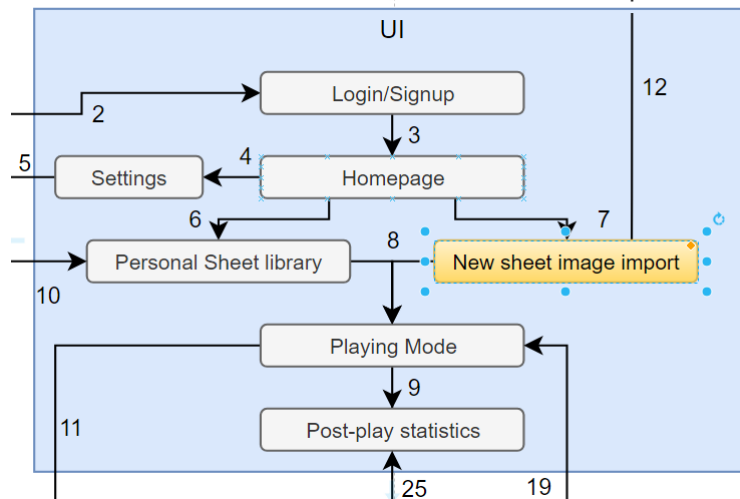


Figure 7: Example subsystem description diagram

it to the user, and then prompt the user if they would like to save the rendered result from the scanner.

4.5.3 SUBSYSTEM INTERFACES

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#7	The subsystem will be accessible from the homepage by a button press.	Homepage redirect to the subsystem via button press	N/A
#8	The subsystem will allow the user to enter the Playing Mode subsystem once they confirmed the generated digital copy.	N/A	UI redirection to Playing Mode Selected sheet music data
#12	The subsystem will allow interaction with the Sheet Scanner layer, giving it the image input it needs, provided by the user.	N/A	User-provided image

4.6 PLAYING MODE

4.6.1 ASSUMPTIONS

- The user will be able to interact with the Sheet Music tracker while playing their instrument.
- The user will be able to see where they are on the sheet music copy while playing

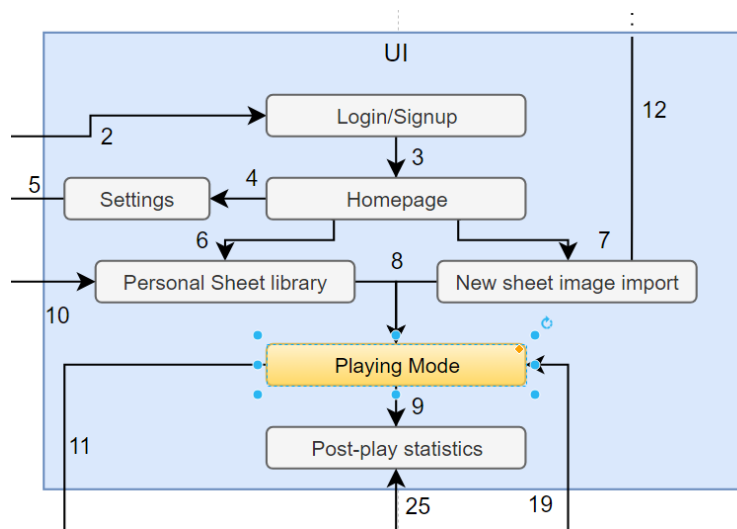


Figure 8: Example subsystem description diagram

- The digital copy of sheet music will display more sheet music when the user is close to completing the sheet they are on.

4.6.2 RESPONSIBILITIES

This subsystem will be responsible for rendering the required information to the user, such as their digital sheet music copy and an indication of where the system thinks they are on the page. The subsystem will also provide the next sheet in the book if the user is close to finishing the current one and there is more than one.

4.6.3 SUBSYSTEM INTERFACES

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#8	The subsystem will be accessible from either the Personal Sheet library or the New Sheet image import. Upon entering the Play Mode page, relevant musical sheet data will be rendered to the user.	A confirmation button from either sheet image import or personal sheet library Selected sheet music data	N/A
#9	The subsystem will redirect to a post-play statistics page after performance	N/A	UI redirect to Post-play statistics
#11	The subsystem will give the user a small list of items to configure for the playing mode if the system detects missing data	N/A	configuration data to Sheet Music Tracker
#19	The subsystem will be directed by the page-turner/scroller/tracker to re-render relevant visual information to the user, such as new music sheet pages and indicators showing where the system thinks they are.	system tracking data	N/A

4.7 POST-PLAY STATISTICS

4.7.1 ASSUMPTIONS

- The user will be able to see how they did on the piece they just played

4.7.2 RESPONSIBILITIES

The Post-Play statistics subsystem is responsible for showing the user how our system thinks they performed in terms of accuracy and time precision.

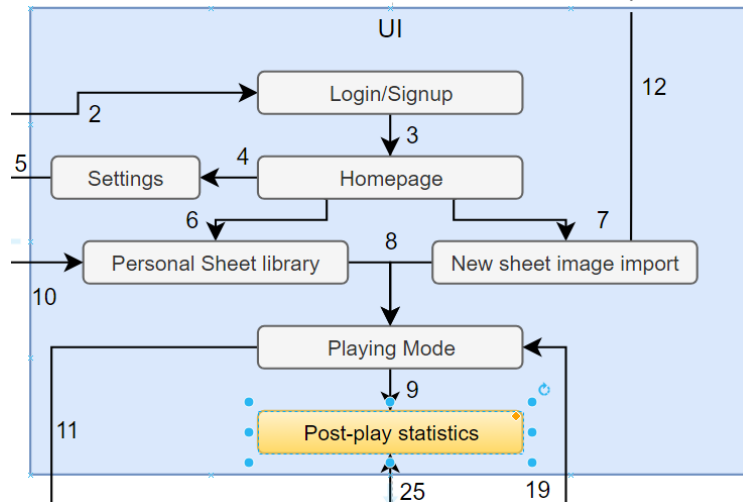


Figure 9: Example subsystem description diagram

4.7.3 SUBSYSTEM INTERFACES

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#9	The subsystem will be accessible after the user is finished playing their piece in Playing Mode.	UI redirect from Playing Mode	N/A
#25	The subsystem will derive its rendered information from the tracker in the Sheet Music Tracker layer.	accuracy data	N/A

5 DATABASE SUBSYSTEMS

Description of the database subsystems.

5.1 ACCOUNT INFORMATION/MAINTENANCE

The Account Info/Maintenance subsystem will communicate with Login/Sign Up, Account storage, and Settings to manage access to the music database and update user account info.

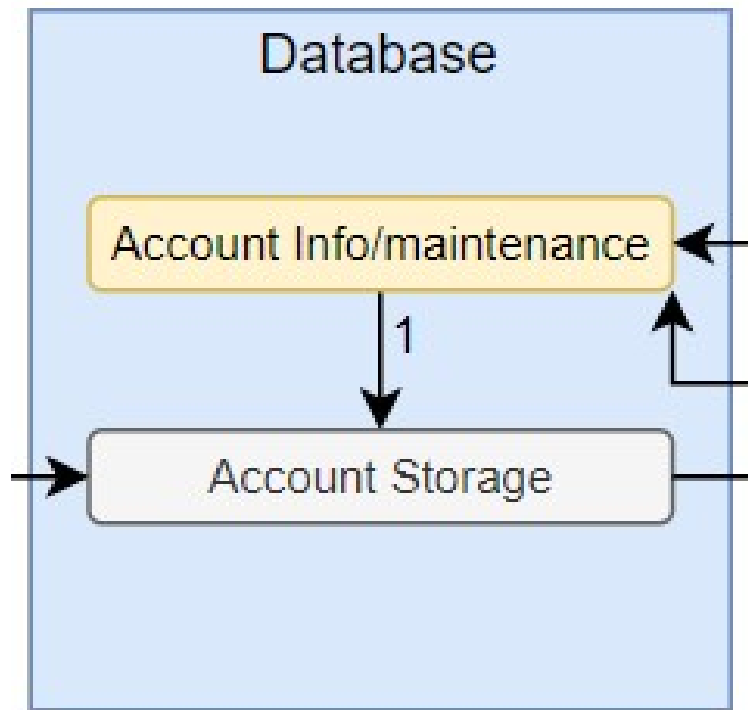


Figure 10: Account Info/Maintenance Subsystem

5.1.1 ASSUMPTIONS

- An account will consist of at least a username and password.
- The user will be able to change their account information.

5.1.2 RESPONSIBILITIES

The account info/maintenance subsystem will be responsible for storing account info for logins and updating account info when requested by the user.

5.1.3 SUBSYSTEM INTERFACES

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	The subsystem will make account storage accessible after a successful login	N/A	Account Info
#2	The subsystem will receive the login info from the UI to compare to stored account info. If successful, it will relay that back to the Login/Sign Up subsystem to move to the homepage	Login credentials	Login/Sign Up status
#5	If requested in settings, the subsystem will update the stored account info	New account info	N/A

5.2 ACCOUNT STORAGE

The account storage subsystem will communicate with the File Translator/Compiler, Personal Sheet Library, and Account Info to handle storing a signed-in user's previously translated music data and show it in their library.

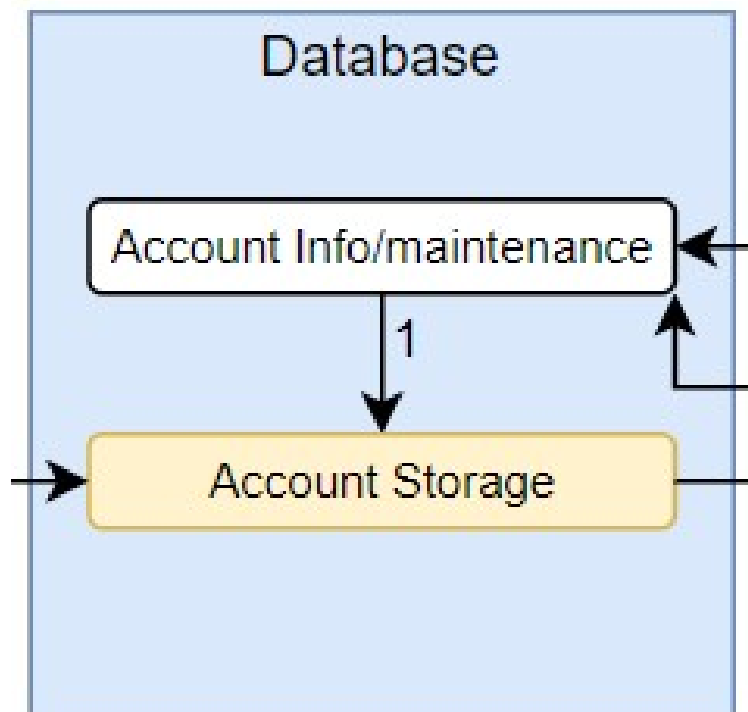


Figure 11: Account storage subsystem

5.2.1 ASSUMPTIONS

- Stored music data will already be in a readable format.

5.2.2 RESPONSIBILITIES

The Account Storage subsystem will be responsible for storing ready-to-use music data from the sheet music that the user scans in. It will be accessible by the personal sheet library for later use.

5.2.3 SUBSYSTEM INTERFACES

Table 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	The subsystem will receive info from the account info subsystem to know which data should be accessible.	Account Info	N/A
#10	Account storage will provide music data to the sheet library subsystem	N/A	Personal Sheet Library
#15	The subsystem will receive music data to store	File Translator	N/A

6 SHEET MUSIC TRACKER SUBSYSTEMS

Description of each subsystem for the Sheet Music Tracker.

6.1 PRE-PLAY CONFIGURATION

The Pre-Play Configuration is supposed to account for any errors in building the sheet music page. In case the beats per minute cant be found or an incorrect musical note is in the music sheet, the user can go in and modify it themselves.

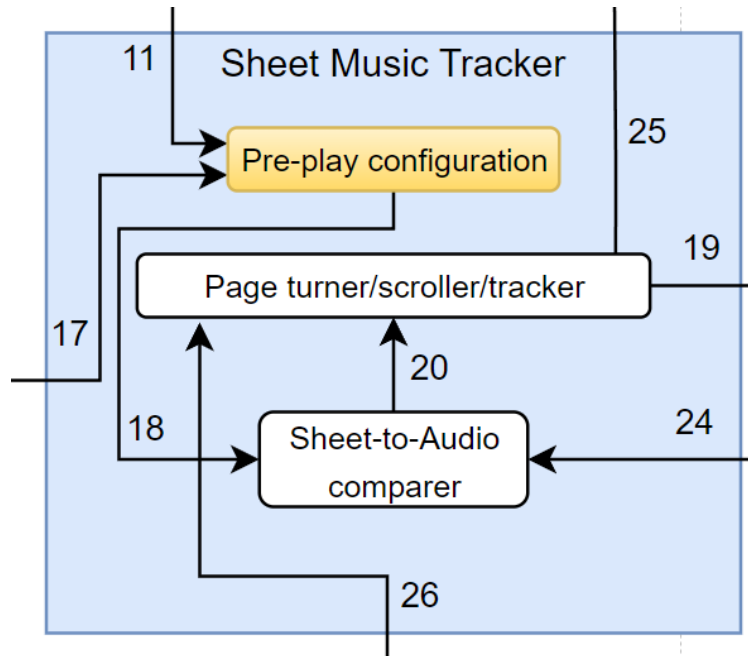


Figure 12: Pre-Play Configuration diagram

6.1.1 ASSUMPTIONS

- The Pre-Play Configuration will receive it information from the data formatter.
- The user will pick out any errors and supply the correct solution.

6.1.2 RESPONSIBILITIES

The Pre-Play Configuration is responsible for auto-filling for any errors in the music page. It then passes this information to the sheet-to-audio subsystem.

6.1.3 SUBSYSTEM INTERFACES

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#11	The user will use the UI to manually adjust the music sheet for any errors	Playing mode/UI user input	N/A
#18	The corrected music sheet is passed to the sheet-to-audio subsystem.	N/A	Corrected music sheet
#17	The scanned music sheet is passed to the Pre-play configuration from the data formatter subsystem	Music Sheet	N/A

6.2 PAGE TURNER/SCROLLER/TRACKER

This subsystem is in charge of tracking where the user is in the current song and determining when it needs to turn the page of the music sheet.

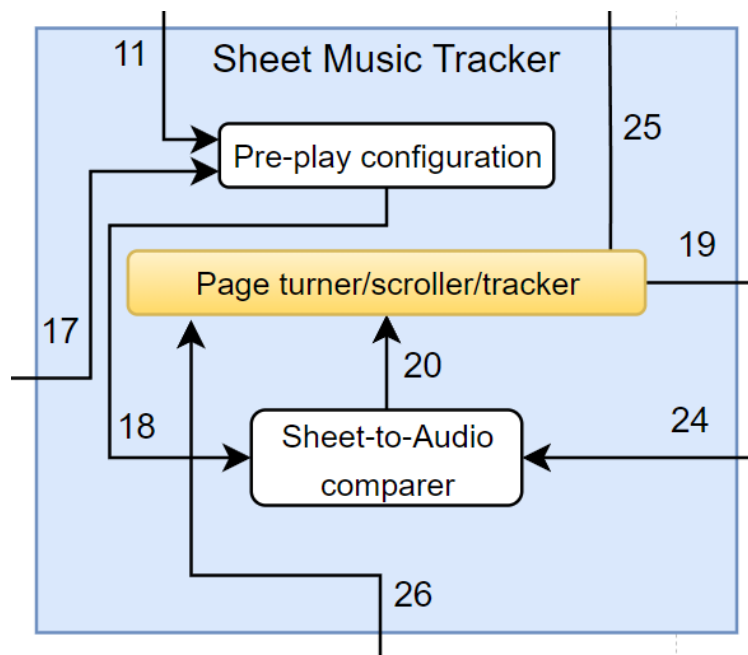


Figure 13: Example subsystem description diagram

6.2.1 ASSUMPTIONS

- A correct music sheet will be provided with no errors from the sheet-to-audio comparer subsystem.
- The Sheet-To-Audio Comparer subsystem will provide the information needed to know where the user is in a song.
- This subsystem will signal at the correct time when to turn the page.

6.2.2 RESPONSIBILITIES

The role of this subsystem is to track where the user is on the music sheet by processing information from the Sheet-to-Audio compaper subsystem. It will pass this Information to the UI play mode subsystem so it can be displayed. This subsystem will also keep track of how well the user played the song so the post play statistics subsystem can get the information it needs.

6.2.3 SUBSYSTEM INTERFACES

Table 12: Subsystem interfaces

ID	Description	Inputs	Outputs
#19	Information about where the user is in the current song is passed to the play mode subsystem/UI. This will allow the user to know when a page is about to be turned.	N/A	Tracking Information
#20	Sheet-to-Audio comparer passes information on where the user is, in the current song based on the audio being played.	sheet to audio information	Updated tracking
#25	Information about how well the user played the song is passed to post played statistics in the UI.	N/A	Accuracy rating
#26	Signals from the pedal tell the page to turn manually.	Pedal Signal	Turn Signal

6.3 SHEET-TO-AUDIO COMPARER

This subsystem uses audio to determine where the user is on the music sheet.

6.3.1 ASSUMPTIONS

- All notes will be correctly picked up as audio without interference.
- The music sheet will have been scanned correctly and displayed without error.
- A connection between the audio produced by the user and notes on the page will be made.

6.3.2 RESPONSIBILITIES

This subsystem is in charge of making the connection between the user playing audio and musical notes on a music sheet. This information will then be passed to the page turner/scroller/tracker subsystem so it can perform its purpose.

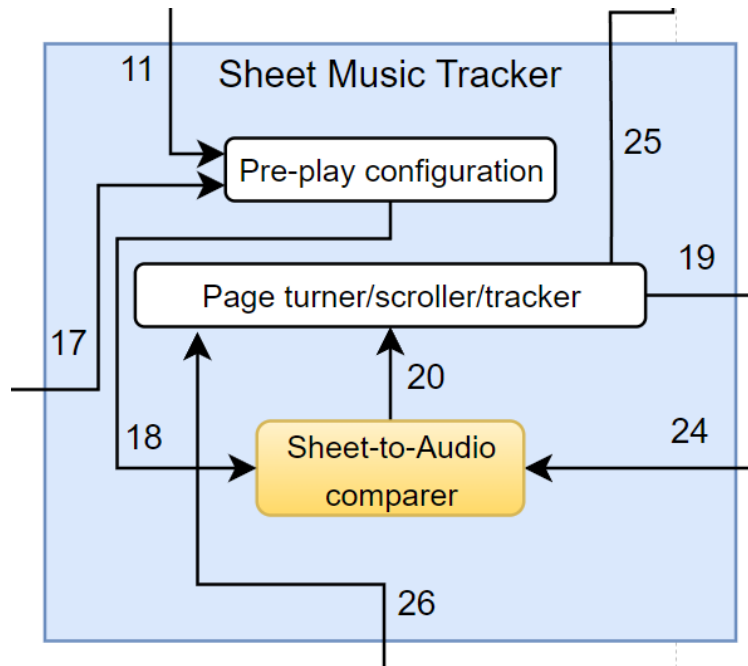


Figure 14: Example subsystem description diagram

6.3.3 SUBSYSTEM INTERFACES

Table 13: Subsystem interfaces

ID	Description	Inputs	Outputs
#18	A formatted music sheet is passed in so a connection can be made to audio.	Formatted Music Sheet	N/A
#20	Information connecting audio played to the music sheet on screen is passed to the page turner/scroller/tracker subsystem.	N/A	Sheet to Audio information
#24	Notes are picked out from audio and passed to this module.	Musical notes audio	N/A

7 SHEET MUSIC SCANNER SUBSYSTEMS

7.1 IMAGE IMPORT

This subsystem is in charge of letting the user choose a new sheet music picture to import to the system and will be sent to the OMR for further processing.

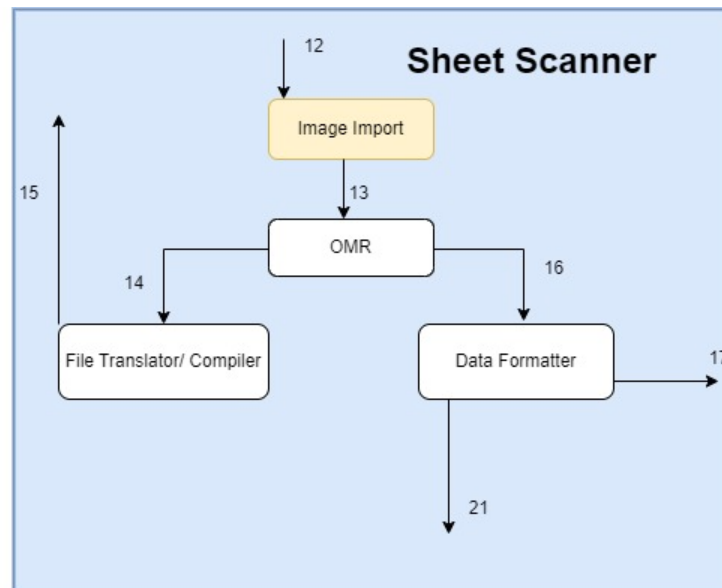


Figure 15: Image import diagram

7.1.1 ASSUMPTIONS

- The image import function will be activated from the UI/New Sheet Image Import selection by the user.
- The image import will support basic image formats.

7.1.2 RESPONSIBILITIES

This subsystem is a check to ensure the user selects a valid picture format. It then allows the OMR subsystem to begin decoding.

7.1.3 SUBSYSTEM INTERFACES

Table 14: Subsystem interfaces

ID	Description	Inputs	Outputs
#12	System is activated by user to import new music	New Sheet Image Import/UI	N/A
#13	Verifies file type is valid and music notes are present	N/A	Picture with music notes

7.2 OPTICAL MUSIC RECOGNITION

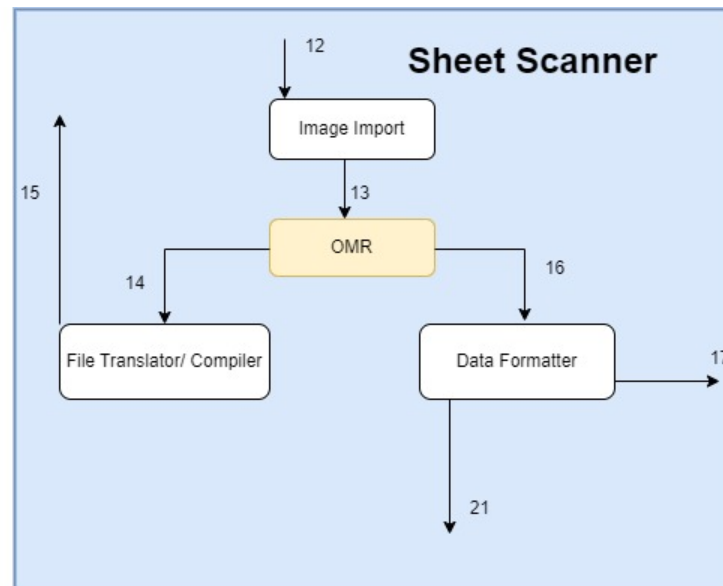


Figure 16: OMR diagram

7.2.1 ASSUMPTIONS

- The OMR will be able to identify all types of notes on the music sheet.
- The note recognition should be able to identify notes that are not readable by the image.

7.2.2 RESPONSIBILITIES

The OMR will identify as many notes as possible and pass the results to other subsystems for further decoding.

7.2.3 SUBSYSTEM INTERFACES

Table 15: Subsystem interfaces

ID	Description	Inputs	Outputs
#14	The image is sent to the compiler to be converted to a music sheet that has the identified notes	N/A	OMR
#16	The identified notes are sent to the data formatter for further decoding	N/A	OMR

7.3 FILE TRANSLATOR/COMPILER

7.3.1 ASSUMPTIONS

- The file translator will compile and translate all supported file types.
- The compiler will compile the data sheet in a fast and efficient time.

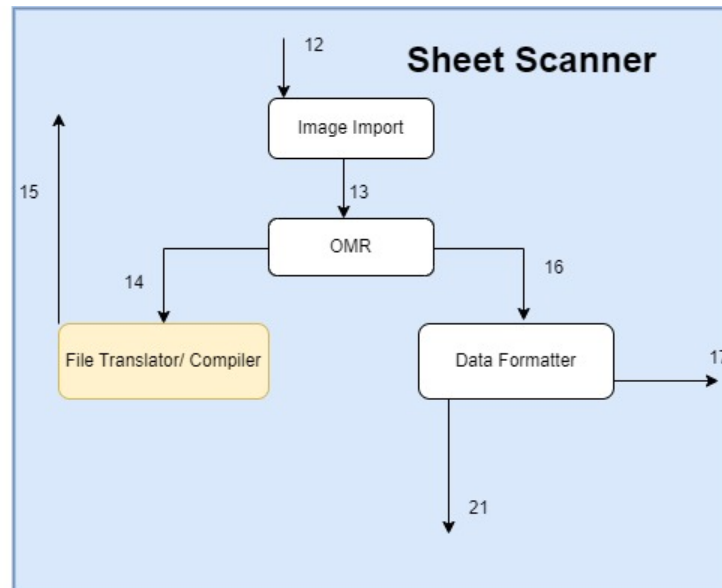


Figure 17: File Translator/Compiler diagram

7.3.2 RESPONSIBILITIES

This subsystem is responsible for compiling and translating the sheet music that has been sent from the OMR to a readable form that will be saved to the user account.

7.3.3 SUBSYSTEM INTERFACES

Table 16: Subsystem interfaces

ID	Description	Inputs	Outputs
#14	Music data of identified notes to be passed for compiling	Data sheet to be compiled	N/A
#15	Compiled and translated sheet music to be saved to the user account	N/A	Translated sheet music

7.4 DATA FORMATTER

7.4.1 ASSUMPTIONS

- Data formatter output will have the output in a format where other systems will easily be able to decode the information.
- The formatter will efficiently parse the notes for easier note, chord, pause...etc identification in other systems.

7.4.2 RESPONSIBILITIES

The Formatter will be responsible for formatting the data for other systems to use the information and recognize the various musical elements the user's music sheet contains. The formatter will use this information given by the OMR and optimize for performance.

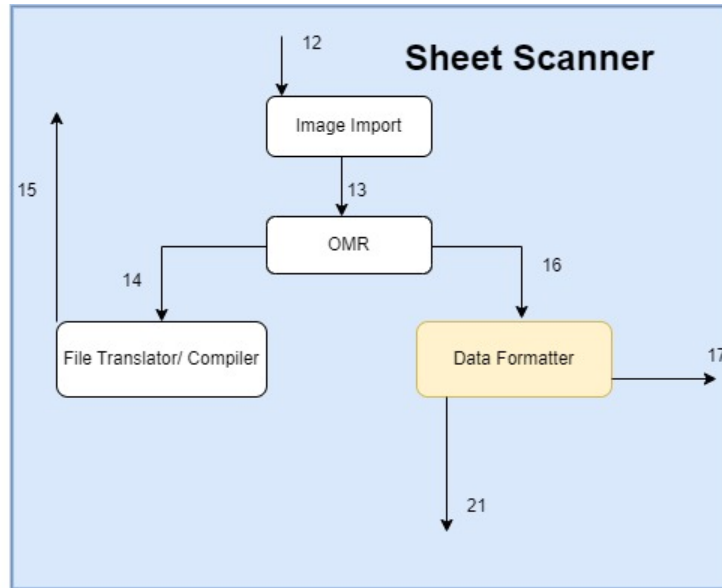


Figure 18: Data Formatter diagram

7.4.3 SUBSYSTEM INTERFACES

Table 17: Subsystem interfaces

ID	Description	Inputs	Outputs
#16	The OMR will send the identified notes to formatter for further optimization to other systems	OMR notes identified	N/A
#17	The formatter will send the pre-checked data to the pre-play configuration to give a final verification from the user	N/A	Pre-Play Configuration
#21	The formatter will send the data to the audio recorder and cross check with the notes being played by the user for a match	N/A	Audio Recorder

8 NOTE RECOGNITION SUBSYSTEMS

8.1 AUDIO RECORDER

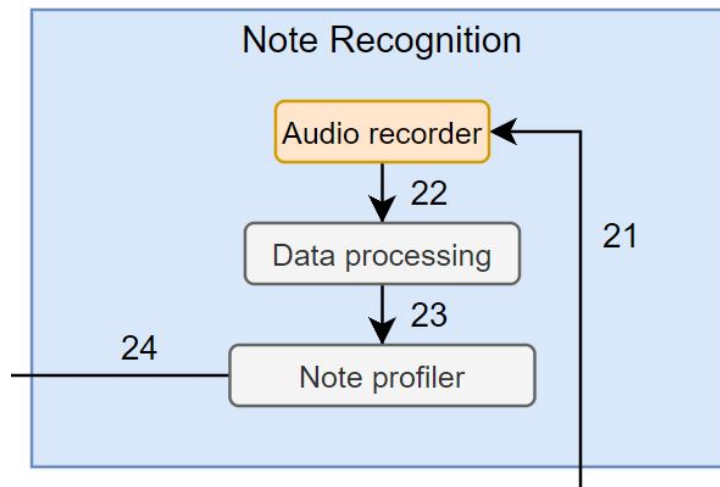


Figure 19: Audio recorder diagram

8.1.1 ASSUMPTIONS

- Audio array will be spilt properly.
- The data will be in a correct format to process information to notes in later steps.

8.1.2 RESPONSIBILITIES

This subsystem will take in the audio of the instrument being played, and break data into elements in an array for processing. This will help with note profiling by having notes within an element.

8.1.3 SUBSYSTEM INTERFACES

Table 18: Subsystem interfaces

ID	Description	Inputs	Outputs
#21	The sheet music note data will be brought in to help split audio into elements	sheet music note data	N/A
#22	Audio array sent to get processed for note detection	N/A	Audio array

8.2 DATA PROCESSING

8.2.1 ASSUMPTIONS

- Audio array will be spilt properly.
- The data will be in a correct format to process information to notes.

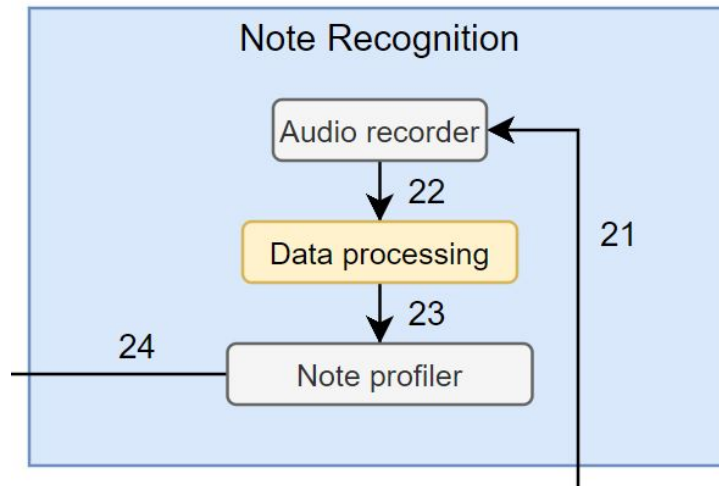


Figure 20: Data Processing diagram

8.2.2 RESPONSIBILITIES

This subsystem will take the audio array from audio recorder subsection and apply some algorithms to simplify the data for note profiler subsection.

8.2.3 SUBSYSTEM INTERFACES

Table 19: Subsystem interfaces

ID	Description	Inputs	Outputs
#22	Audio array sent to get data processed for note detection	Audio array	N/A
#23	Processed Audio array ready for profiling	N/A	Audio array

8.3 NOTE PROFILER

8.3.1 ASSUMPTIONS

- Audio array will be spilt properly.
- The data will be in a correct format to process information to notes.
- Data from audio array will compare to to known notes/chords data within a certain tolerance.

8.3.2 RESPONSIBILITIES

Takes processed audio array and will compare that data to data known for notes/chords to determine what is being played. This will then be stored in an array that will be sent to sheet-to-audio comparer.

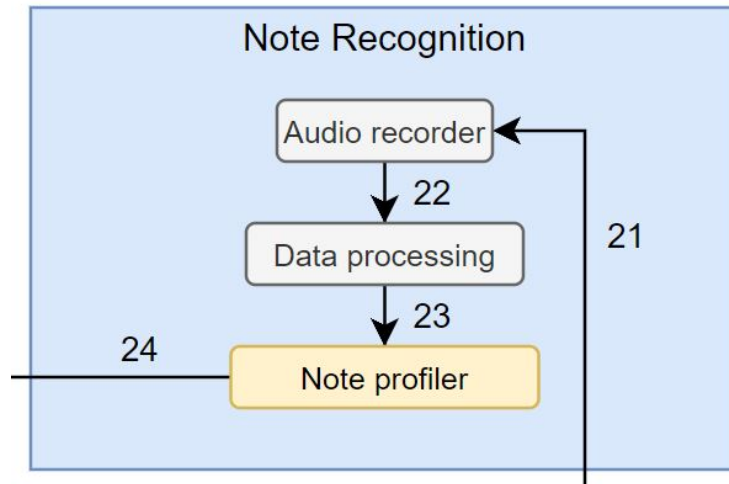


Figure 21: Note profiler diagram

8.3.3 SUBSYSTEM INTERFACES

Table 20: Subsystem interfaces

ID	Description	Inputs	Outputs
#23	Processed Audio array ready for note profiling	Audio Array	N/A
#24	Note Array sent to be compared with sheet music note	N/A	Note Array

9 MANUAL PAGE TURNING PEDAL

9.1 PEDAL

The pedal will be a hardware piece that will send a signal to change the music page if the user chooses to even if the end of the music sheet is not reached.

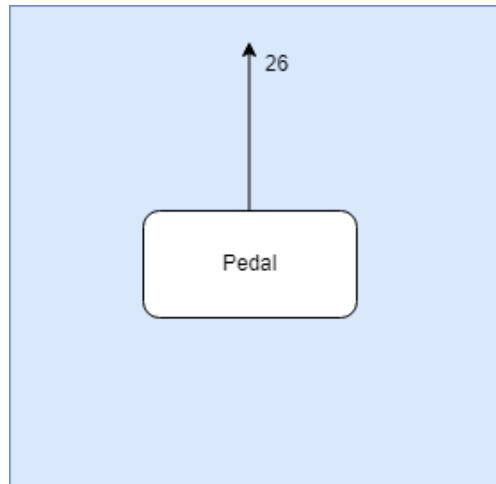


Figure 22: Pedal diagram

9.1.1 ASSUMPTIONS

The pedal will be able to manually turn the page early or as a fail safe if software fails to change the page.

9.1.2 RESPONSIBILITIES

The pedal can be used at any moment by the user to change the music page if the user chooses to do so early. The pedal can also be used in case of a software hang or glitch where the page is not turned if the end of the page is reached and all notes have been played.

9.1.3 SUBSYSTEM INTERFACES

Table 21: Subsystem interfaces

ID	Description	Inputs	Outputs
#26	A signal will be sent to the tracker/ page turner system to turn the page	N/A	Signal to Page turner/ Tracker

REFERENCES