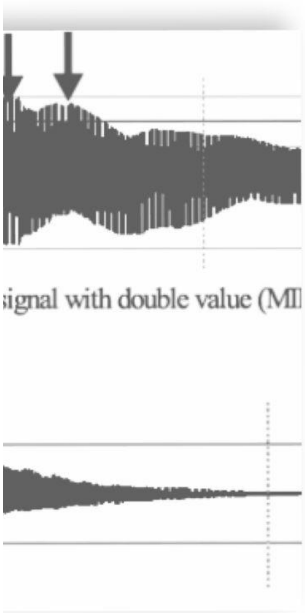
Analysis Report

Group 20



pianissimo

Team Members:
Melisa Kamacı
Tamer Alkım Tokuç
Nazem Yıldırım
Begüm Hatipoglu

Analysis Report

1. Introduction

This paper aims to provide a detailed analysis of Pianissimo by covering all major project-related concerns. In order to provide stakeholders with a thorough and improved understanding of what Pianissimo has to offer, the content is intended to develop a complete, systematic, and verifiable analysis model through formulating the required functionality.

2. Current System

Since the current system is not available, the planned system is explained in the proposed system part.

3. Proposed System

Pianissimo aims to make it easier for anyone who is interested in this instrument to play the piano by providing notes for the desired piano compositions. Its main feature is to extract base notes using various technologies and provide them in PDF or MIDI format after loading plain piano music in an audio format. The system has only the user category.

User

In Pianissimo, the user is the person who joins the system as a member of the system's website using the application. Becoming a member requires obtaining a user/member ID for each online user and a password to log into the account taken. As a member, there are many options that a user can perform in this web application.

The first option is for the user to see the existing data in the system. After logging into the user account, they can view the previously uploaded and converted note documents in the system.

In addition, the user can take printouts from the system by uploading documents to the system. The output in the system can be in MIDI or PDF format. Perhaps the most important feature of the system for users, whose ten minutes are so valuable, is that the system is fast because it is held in a maximum of 600 seconds when the system is overloaded, and in an average of 90 seconds when Pianissimo notes are played. This means users will get the output in the fastest possible time.

The user can upload the input, that is, the translation, by dragging the desired file or by selecting a file from the upload file section. In addition, the user can view the conversion in the previous conversion, new conversion and conversion in progress sections from the conversion page.

When the user loses his/her password for the system or wants to renew it optionally, they will be able to renew the password. The security of passwords is provided by the system. Users save all their personal information in the profile management section of their accounts. The customer can make the necessary changes on his/her previously saved personal information.

3.1 Overview

Pianissimo is designed as a computer application to address the difficulty of obtaining the notes of piano songs by utilizing a range of technologies. The major functionality which makes up the main reason why Pianissimo is needed by pianists focuses on the conversion of any solo piano music to their sheet music. After processing the uploaded piano composition as an audio file, the notes are extracted and made available in different formats including PDF or MIDI. To provide easy access to previously transcribed recordings, the membership system is planned to be integrated.

3.2 Functional Requirements

- The program shall identify notes (including accidentals) with %95 accuracy.
- The program shall calculate subdivided notes with 99% precision.
- The application shall be 90% accurate when determining BPM and key.
- Depending on the user's preferences, the software will either output a MIDI file or a PDF file.
- In addition to lossless files like FLAC and WAV, the application shall also accept lossy compressed files like MP3 and AAC. Other file types should not be considered.
- The program shall log users with correct credentials in and disallow access to ones with wrong credentials.
- The program shall offer a registration page where new users can sign up.
- Whenever access is required, the application shall make previously transcribed recordings accessible for display and removal.
- The program shall enable users to update their profile in respect of their preferences.
- The program shall grant users a new password whenever the password recovery system is initiated.

3.3 Nonfunctional Requirements

- The program shall have a clear UI.
- The program shall have %90 uptime per month.
- The program shall comply with local musical note naming convention.
- The program shall not use more than 2 GB of RAM.
- The program shall run cross-platform as well as on widely used browsers.

3.3.1 Performance

The performance of the application heavily depends on the audio quality and the size of the audio file. User's internet connection stability and speed must also be considered. Processing and transforming the audio should not take more than 2-3 minutes provided the mentioned criteria meets certain standards.

3.3.2 Usability

The graphical user interface (GUI) is to be designed minimalistic in order to keep the user experience non-complex. The instructions and limitations on how to use the application efficiently is to be displayed on the interface.

3.3.3 Reliability

The downtime of the application user interface should be limited to a maximum of 5 hours a week. In case of a crash in the application, it must be repaired in at most an hour. The system shall display appropriate messages to the user while the system is down.

3.3.4 Supportability

The application is to be cost-effective to maintain and the maintenance of the system shall be done regularly. The application must be suitable for the addition of new features and the updating of the current features.

3.3.5 Scalability

The scalability of the system will be taken into consideration according to the changing needs of the users and the industry. While scaling up the application in terms of the conditions mentioned in hardware and software constraints, the performance of the system shall be maintained if not improved.

3.3.6 Extensibility

The application is to be designed considering SOLID principles in software engineering, namely open-closed principle. Therefore the system will be open to extensions such as the addition of new features according to the needs and suggestions of the users without changing the majority of the foundation of the code.

3.4 Pseudo requirements

- To avoid adding to the complexity and maintain the usability of the application, the GUI design must be made basic and adhere to material design patterns.
- The GUI must follow the industry standards to appear identical across all operating systems.
- Python 3's minimum hardware requirements must be met for the program to run.
- To reduce reliance on specific external packages, management of extra packages will be handled via shipping frameworks like Docker.
- If the program's web interface was used, an internet connection must be set up.
- Data transfer for Web apps will be shielded from any outside access.
- Future plans call for a web application to be constructed using the React framework.
- The website will verify that the audio file has been submitted by looking at the file extension.
- HTTPS will be used by the website as a connection protocol.
- TCP/IP must be used for system communication.
- Mobile applications will use Kotlin/Java as a future development plan.
- The efficiency of the system is maintained at a maximum of 600 seconds when the system is heavily loaded and an average of 90 seconds when Pianissimo outputs the notes.
- Minor faults such as screen problems including erroneous data display will be limited to no more than 5 per KLOC.

- Major Bugs will be limited to a maximum of 0.25 errors/KLOC, including system failures, security breaches, and the inability to generate output.
- Critical errors will be limited to no more than 1 error/KLOC, such as users with excessive or inadequate access.
- 240 hours will be the Mean Time Between Failures (MTBF) for the system.
- The system's Mean Time to Repair (MTTR) will be two hours.
- The system will have a maximum of 50 projects stored and a maximum of 100 users may be logged in at once.
- The OS will manage printer connections.
- Users are only allowed to upload audio files that are no larger than 250 MB in size.
- The users' audio files must have a bitrate of 224 Kbps or above for the application to process them.
- Under their needs, the system must be able to interface with other components.

3.5 System models

This section describes the process-oriented representation of Pianissimo by using system models to outline the flow of information between the models of the system as well as to depict process interactions and the operations performed by these processes.

3.5.1 Scenarios

- The user can sign up to Pianissimo in order to create a profile.
- The user can log in to view their profile and the sheet music they have transformed before, if they have.
- A user can log out directly after logging into the pianissimo web page.
- The user can load sheet music before logging out.
- The user can download the output of the sheet music file or output without downloading it.
- The user can obtain an existing music file in the user's database.
- The user can remove a chosen file of sheet music from their profile which they have transformed before.

3.5.2 Use case model

The use case diagrams of the subsystems of Pianissimo are depicted in Figures 1, 2 and 3. The diagrams are followed by further descriptions.

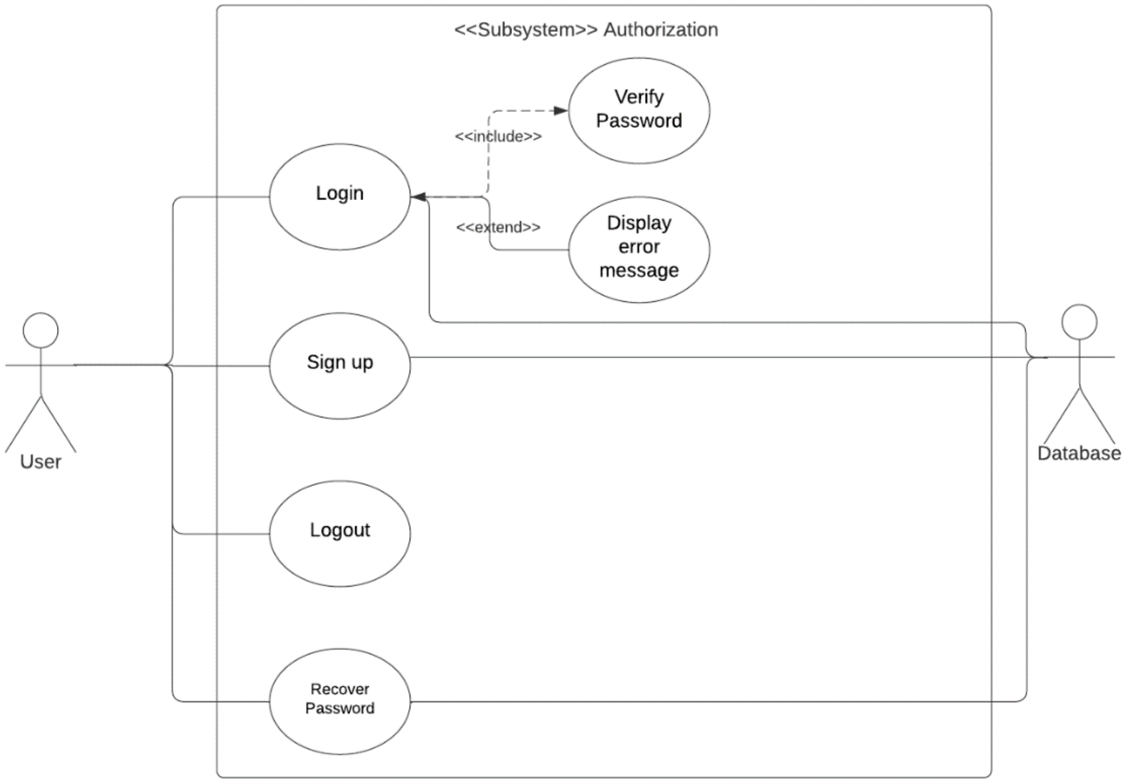


Figure 1: Authorization subsystem use case diagram

Use Case	Login
Actor	User Database
Description	Logging into the system allows users to access their accounts and use system features. Since emails and passwords are stored in the database, each time a user logs in to the system, verification of the password is done by encrypted data retrieved from the database. If verification fails, an error message is displayed.

Use Case	Sign up
Actor	User Database
Description	Signing up to the system creates new users through registration. Submitted data such as email and password are saved to the database for future use.

Use Case	Logout
Actor	User
Description	Logging out from the system allows users to exit the system to avoid unauthorized operations.

Use Case	Recover Password
Actor	User Database
Description	Password recovery may be needed by users who end up forgetting their credentials to recover their accounts. The credentials are replaced in the database with a standard procedure.

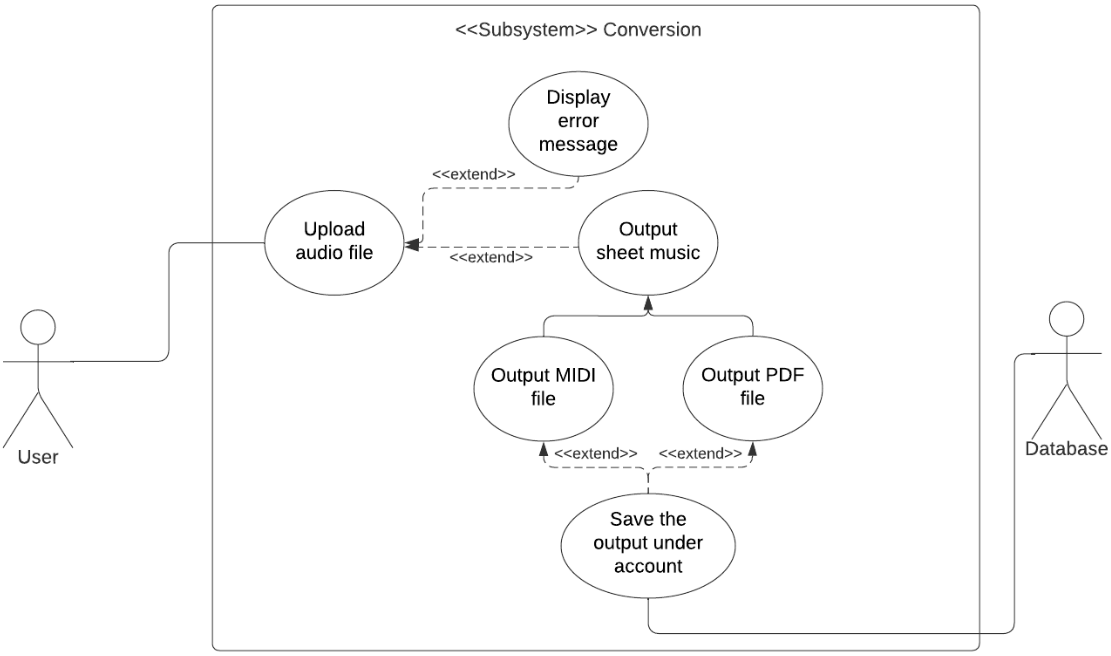


Figure 2: Conversion subsystem use case diagram

Use Case	Upload Audio File
Actor	User Database
Description	The user initiates the upload of an audio file in either FLAC, WAV, MP3, or AAC formats containing solo piano music to be processed by the system. If the format is invalid or does not meet the file expectations, an error message is displayed. If the file has no problems, the system outputs the sheet music in either MIDI or PDF file. Depending on the format choice, the output is saved under the account in the database.

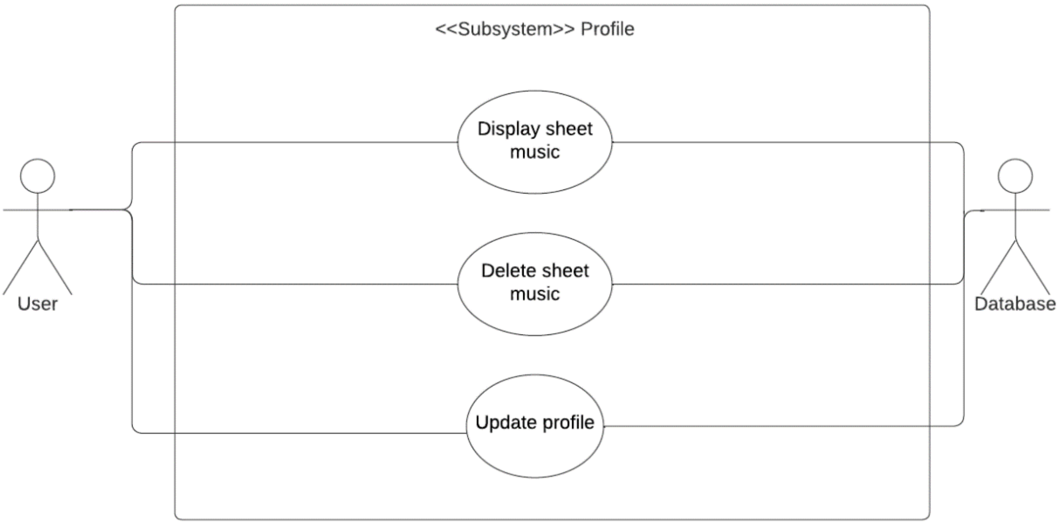


Figure 3: Profile subsystem use case diagram

Use Case	Display sheet music
Actor	User Database
Description	The user will be able to display the sheet music of previously transformed recordings on their profile with the connection to the database and view the selected ones.

Use Case	Delete sheet music
Actor	User Database
Description	The user will be able to remove the selected sheet music from their profile by initiating a removal operation on the database.

Use Case	Update Profile
Actor	User Database
Description	The user will be able to update their profiles according to their personal preferences. The settings such as password change and theme change will be updated right after submitting changes.

3.5.3 Object and Class Model

The object and class model of Pianissimo that shows planned classes along with their internal properties and their interactions is illustrated in figure 4.

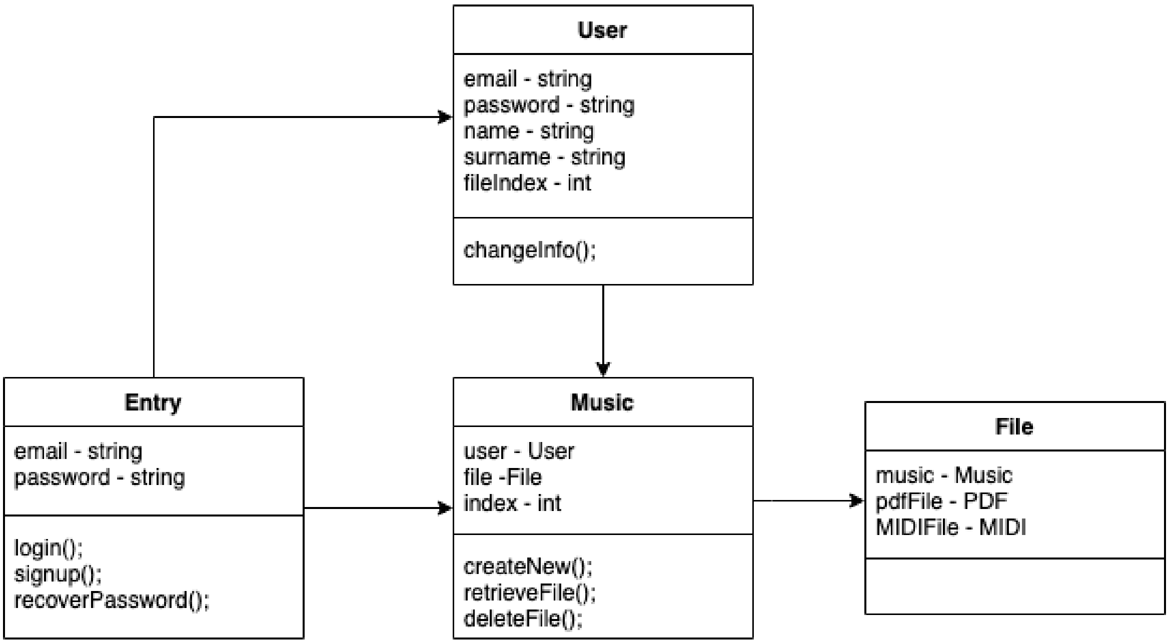


Figure 4: Object and Class Model

3.5.4 Dynamic models

The dynamic models of Pianissimo including state diagrams, sequence diagrams, and activity diagrams are modeled in this section to express the behavior of the system over time. The data download workflow shown in figure 5 gives an idea of the general working principle of the system by establishing interconnections of system components.

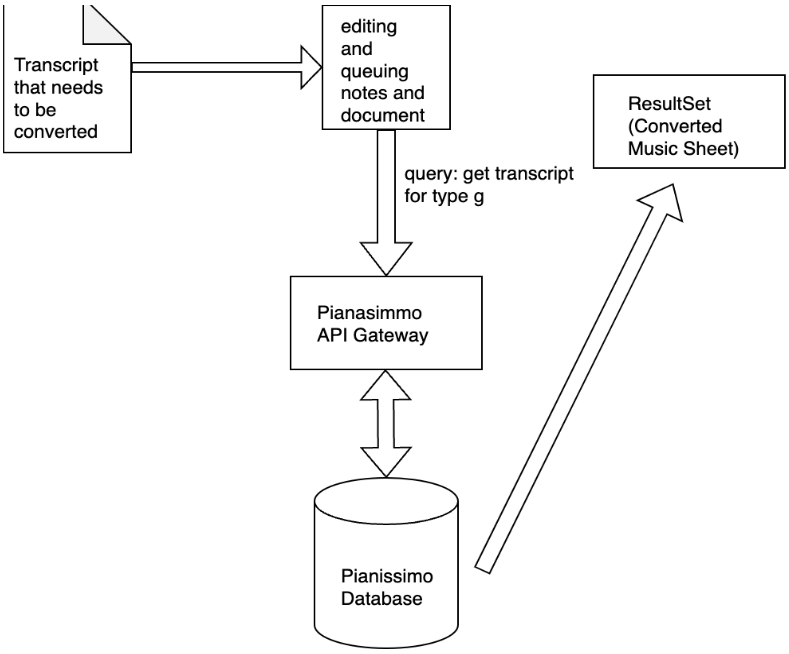

Transcript that needs to be converted
editing and queuing notes and document
query: get transcript for type g
ResultSet (Converted Music Sheet)
Pianasimmo API Gateway
Pianissimo Database

Figure 5: Data Download Workflow

3.5.4.1 State Diagrams

The state diagrams which illustrate the states an object can attain as well as the transitions between those states are used to point out the functionality of the main menu (figure 6) and the login page (figure 7).

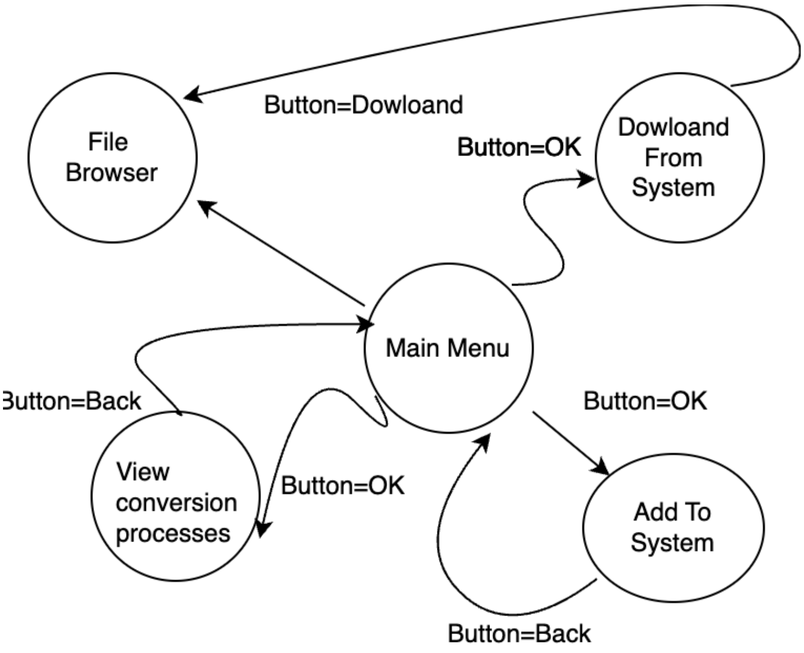


Figure 6: State diagram for Main Menu

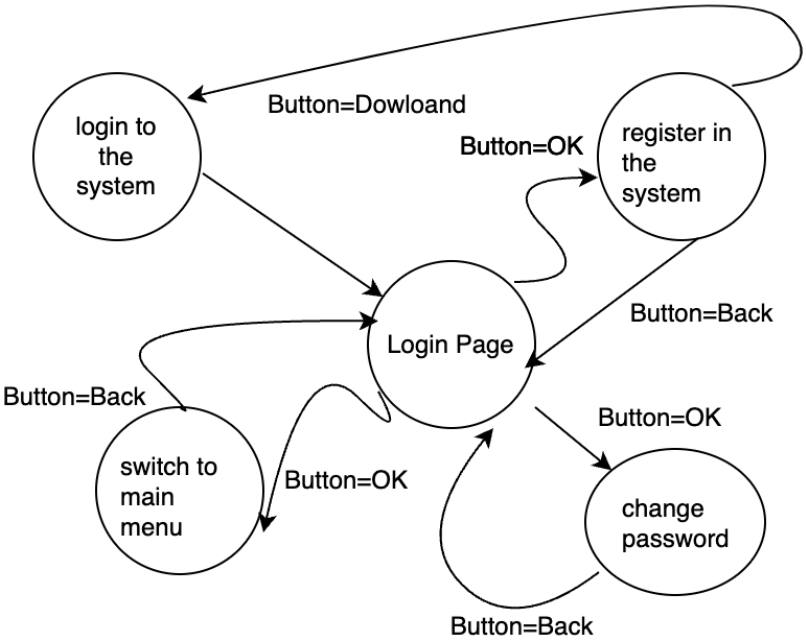


Figure 7: State diagram for Login Page

3.5.4.2 Sequence Diagrams

This section models the sequence diagrams to create interaction diagrams that detail how operations are carried out in a time sequence. The diagram titles are indicated in subsection names.

1. **User Sign Up**

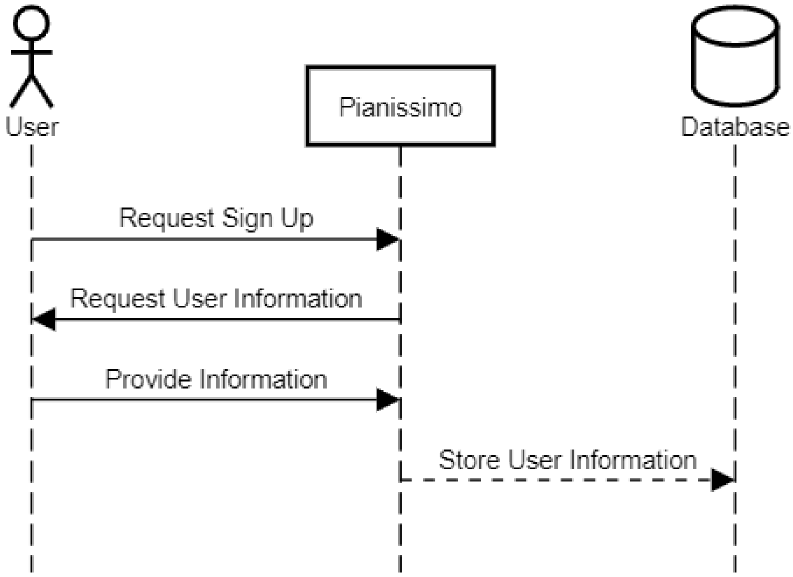


Figure 8: User Sign Up Sequence Diagram

2. User Login-Password Recovery

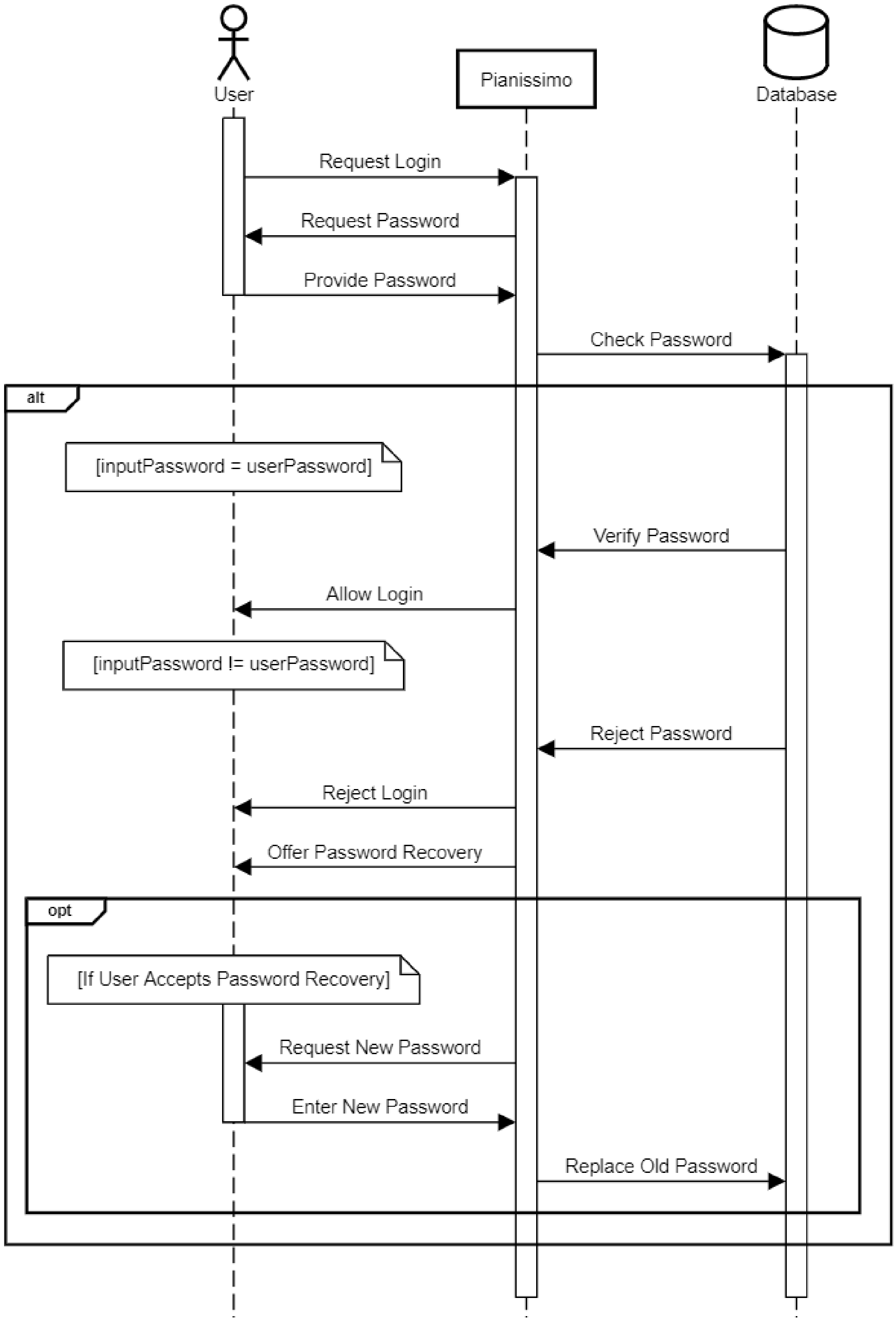


Figure 9: User Login-Password Recovery Sequence Diagram

3. Uploading Audio Files

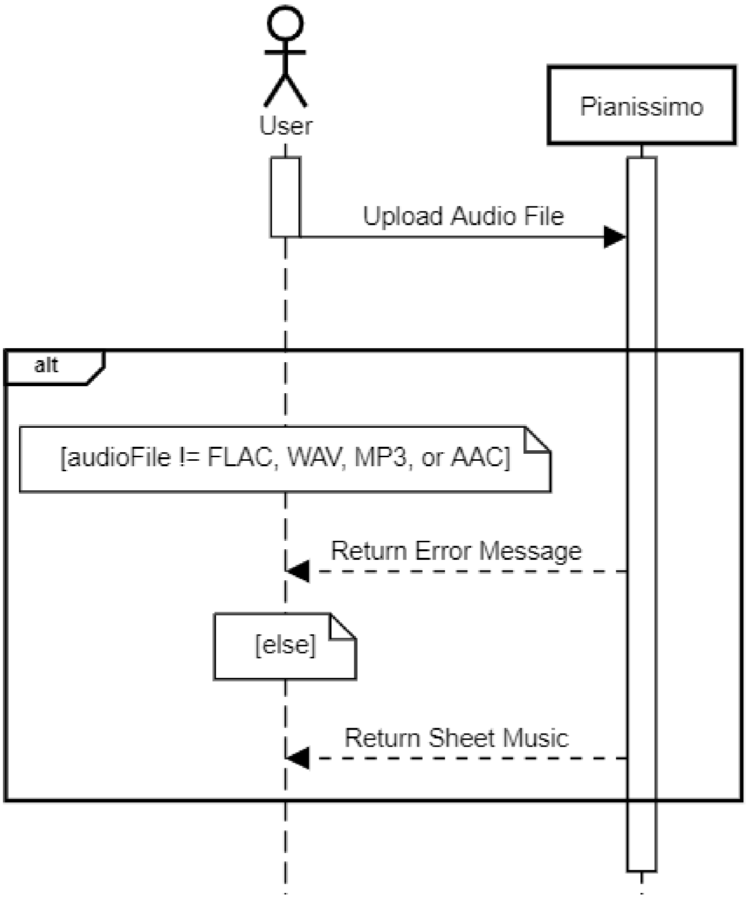


Figure 10: Uploading Audio Files Sequence Diagram

4. Searching Sheet Music in Profile

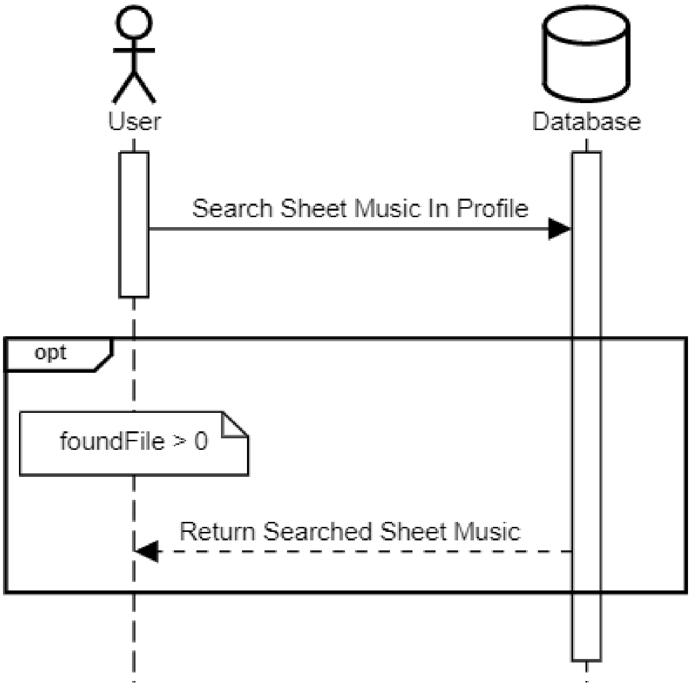


Figure 11: Searching Sheet Music in Profile Sequence Diagram

5. Deleting Sheet Music

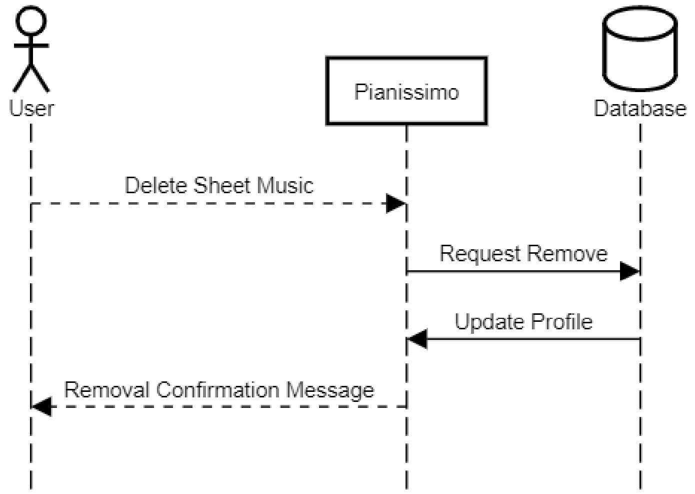


Figure 12: Deleting Sheet Music Sequence Diagram

3.5.4.3 Activity Diagrams

This section models the activity diagrams to visually present a series of actions and the flow of control in the system. The diagram titles are indicated in subsection names.

1. User Sign Up/Login

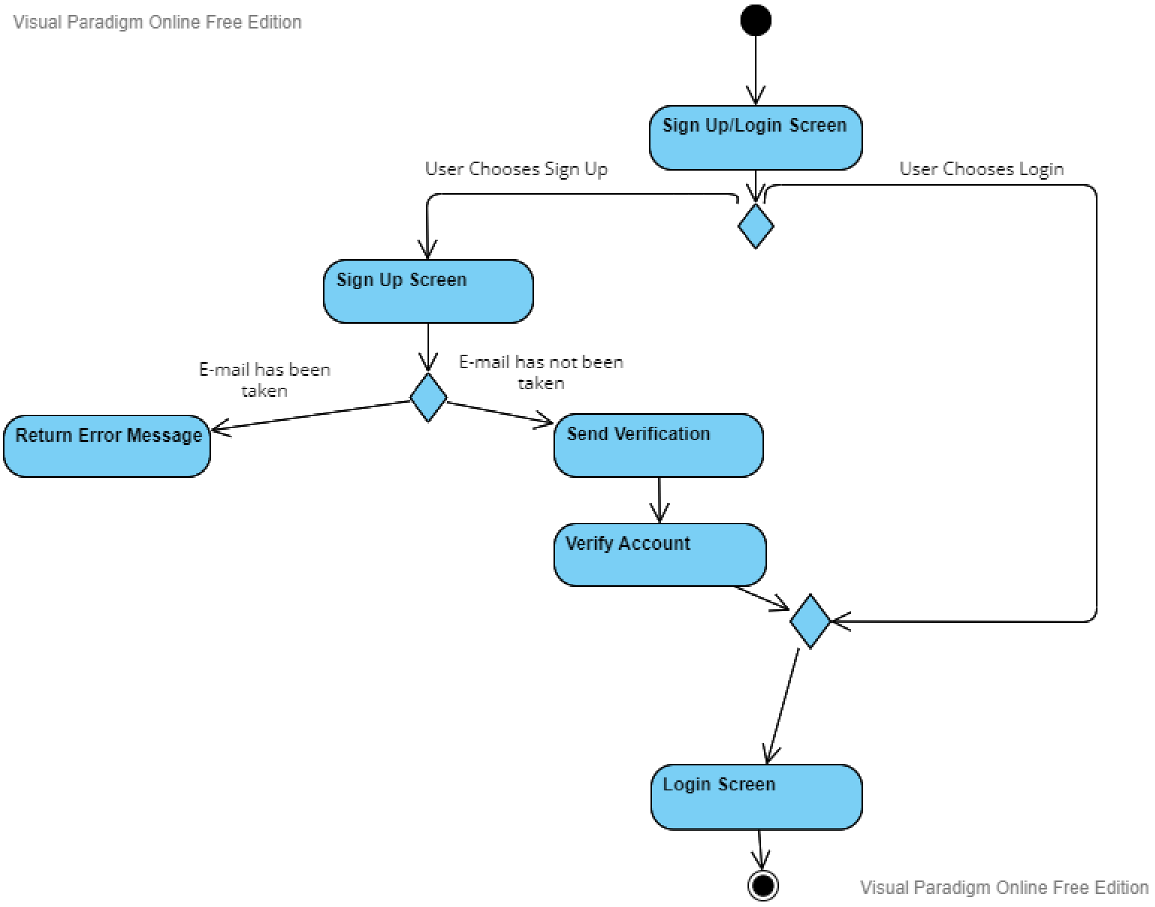


Figure 13: Activity Diagram of User Sign Up/Login

2. User Login-Password Recovery

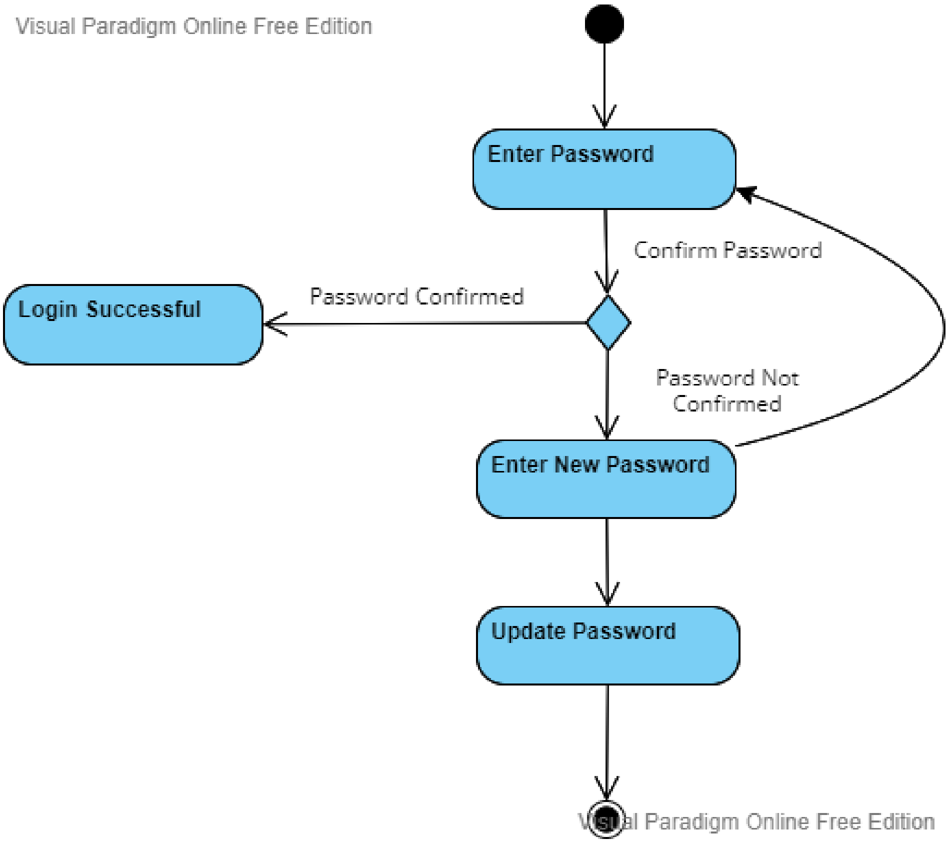


Figure 14: Activity Diagram of User Login-Password Recovery

3. Uploading Audio Files

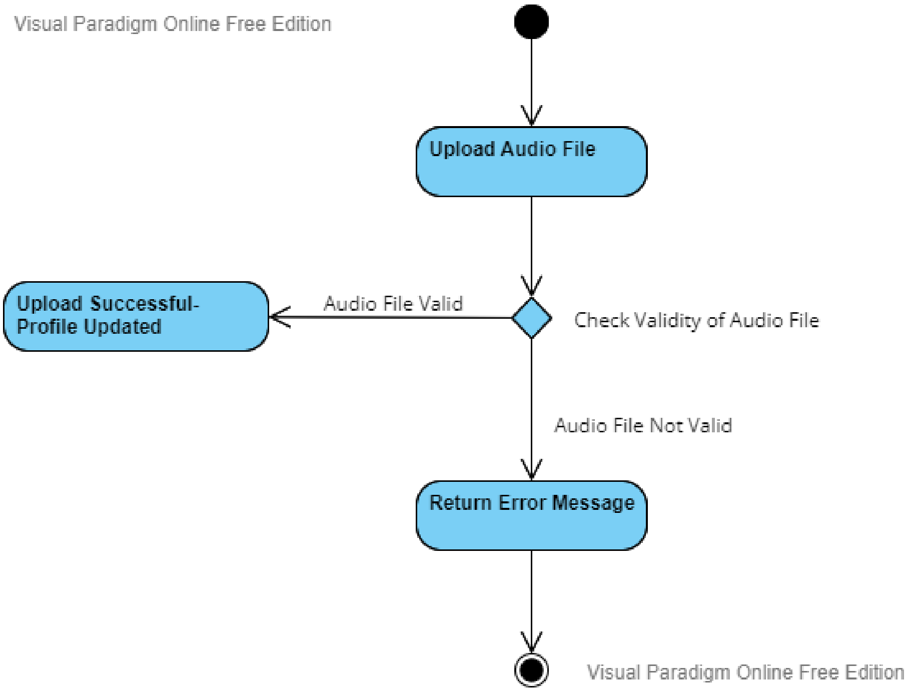

Visual Paradigm Online Free Edition
Upload Audio File
Upload Successful-Profile Updated
Audio File Valid
Check Validity of Audio File
Audio File Not Valid
Return Error Message
Visual Paradigm Online Free Edition

Figure 15: Activity Diagram of Uploading Audio Files

3.5.5 User Interface - Navigational Paths and Screen mock-ups

This section illustrates the UI designs of several pages to demonstrate the visual aspects of the system and models a navigational path diagram to specify the type of actions users can take.

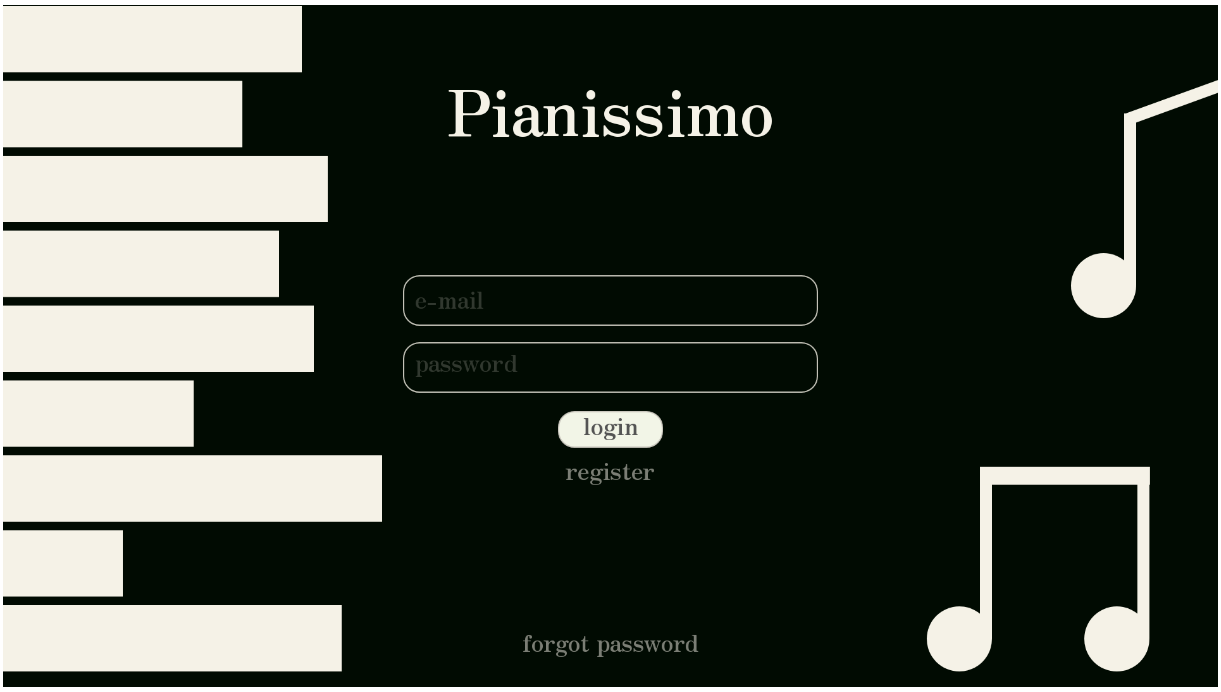


Figure 16: Login page

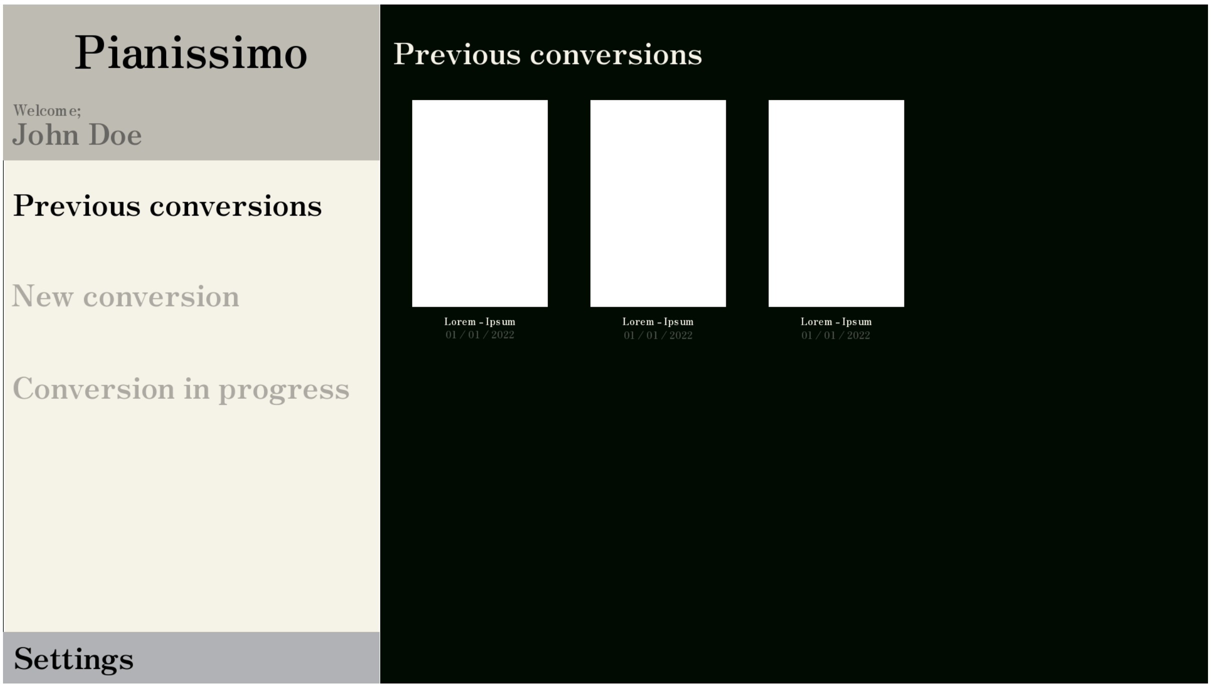


Figure 17: Main interface with previous conversions tab active

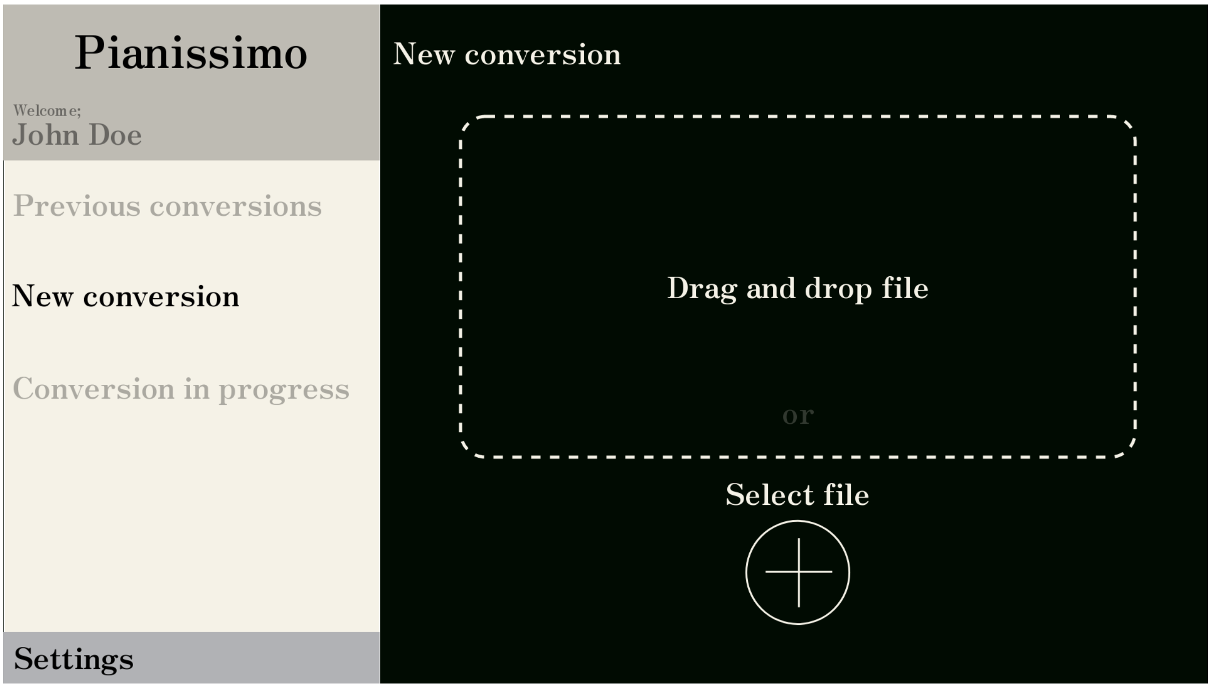


Figure 18: Main interface with new conversion tab active

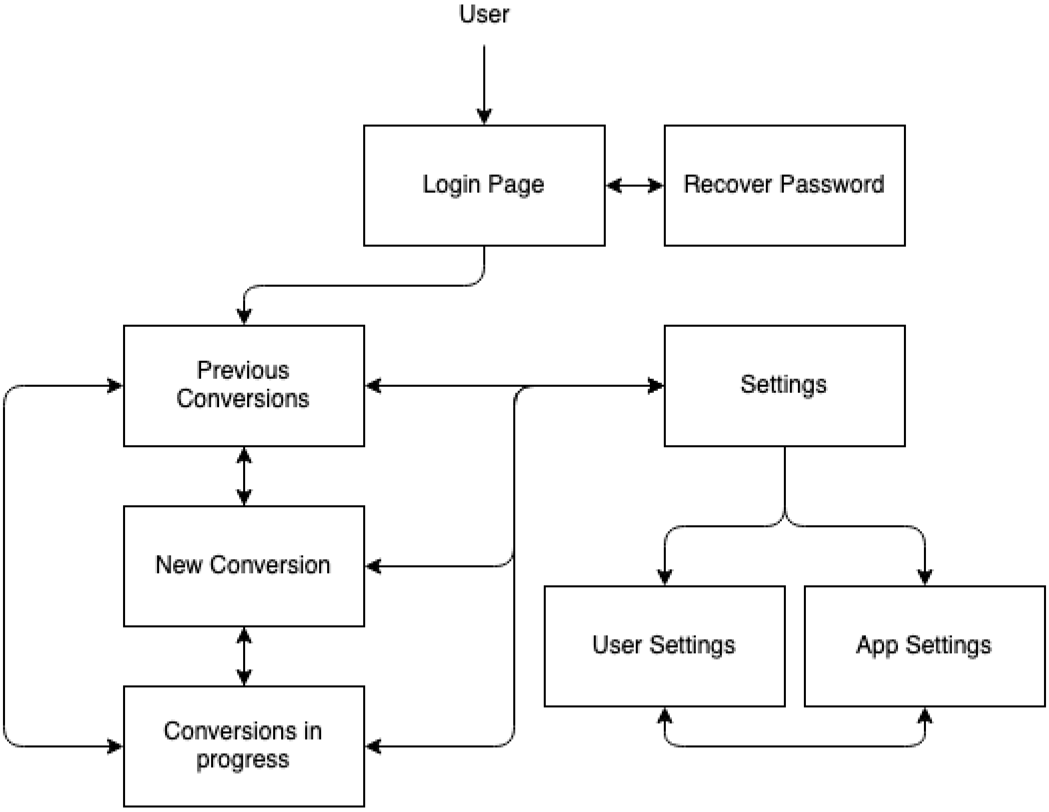


Figure 19: Navigational path for user

4. Glossary

Pianissimo **(noun):***a passage marked to be performed very softly.*

MIDI (Used in Section 3): Musical Instrument Digital Interface

PDF (Used in Section 3): Portable Document Format

BPM (Used in Section 3.2): Beats per Minute

FLAC (Used in Section 3.2): Free Lossless Audio Codec

WAV (Used in Section 3.2): Waveform Audio File Format

MP3 (Used in Section 3.2): MPEG-1 Audio Layer 3

AAC (Used in Section 3.2): Advanced Audio Coding

UI (Used in Section 3.3): User Interface

GB (Used in Section 3.3): Gigabyte

RAM (Used in Section 3.3): Random Access Memory

SOLID (Used in Section 3.3.6): Single responsibility principle, Open-closed principle, Liskov substitution principle, Interface segregation principle, and Dependency inversion principle

HTTPS (Used in Section 3.4): Hypertext Transfer Protocol Secure

TCP/IP (Used in Section 3.4): Transmission Control Protocol/Internet Protocol

KLOC (Used in Section 3.4): Thousands (Kilos) of Lines of Code

OS (Used in Section 3.4): Operating System

MB (Used in Section 3.4): Megabyte

Kbps (Used in Section 3.4): Kilobits per Second

5. References

[1]: Ohio State University Department Of Computer Science and Engineering, Systems Requirements Specification (SRS), 2006
https://web.cse.ohio-state.edu/~bair.41/616/ReqDoc_Outline.htm