SYSTEM REQUIREMENTS

Project: TAB2XML
Course: Software Development Project (EECS 2311)

THE TEAM (GROUP 12):

Matteo Pulcini - [217536756]
Krishna Raju - [218199497]
Patrick Qi - [218095091]
Riffi Manoj - [218061986]

1.0 Introduction

1.1 Purpose

The main objective of the system requirement document is to describe all the required features of the software in great detail, the needs of the customer, and the use cases/user stories of TAB2XML. The software is intended to aid musicians and students in converting music from one format to another. For example, the user is able to input text-based tablature for guitar/drums which can then be converted into sheet music and music XML. They are also presented with the option to play the converted tablature via a virtual drum/guitar instrument.

1.2 Intended Audience

The intended audience for the software can ultimately be anybody with an interest in learning how to play or read sheet music or to convert from one music format to another with simplicity. The user could be a curious teen interested in music all the way to an elderly who practices music as a hobby.

1.3 Intended Use

The intended use of the software is for the user to provide tablature for a certain instrument they are interested in learning. The tablature is then converted into different types of formats such as MusicXML, a MIDI audio file, or Sheet Music which can be saved to a personal computer. The user is also presented with the option to play the tablature they had just converted via a guitar/drum virtual instrument. They also have the freedom to choose the tempo of the music player.

2.0 Product Outline

2.1 Product Overview

The general overview of the product is for the user to pass through the tablature for the instrument of their choice. The imported tablature is then converted into sheet music, where they will have various different options to choose from such as playing the music with a virtual instrument of their choice, pausing/playing the music, and saving the sheet music to their personal machine, etc.

3.0 Functional Requirements

3.1. View input music in different formats

The system must allow the user to input the drum/guitar tablature and output the option to …

1. Listen to the input music
2. See the music in a tablature format
3. MusicXML format
4. Download the music as a PDF or MIDI file or music XML

3.2. Play music

The system must play the input music with the option to …

1. Pause the Music
2. Change the tempo of the music
3. Skip through certain parts of the music

3.3. Go to desired measures

The system must allow the user to highlight and jump to certain measures with respect to the whole system except for the audio component.

3.4. Support Guitar & Drum Tablature Notations

The system must display tablature from the input with the following notations …

1. Chords
2. Slurs
3. Slides
4. Grace Notes
5. Gliss Slides
6. Repeats

3.5. Allow the user to print the music

The system must be able to print the tablature to their local printer.

3.6. Allow the user to save the music

The system must allow the user to save music as a PDF file.

3.7. Allow the user to adjust output aesthetics

The system must allow users to change the output of the aesthetics. Some features that should be adjusted are:

1. Size of Notes
2. Font of Notes
3. Spacing Between Notes/Measure

3.8. Allow easy maneuvering

The GUI of the application is set in a spacious way where a first-time user is not overwhelmed by the various buttons. The GUI is very simple and aesthetic which means, it is very hard for the user to be confused about what they are doing.

4.0 Non-Functional Requirements

The system should be:

4.0. Available & Reliable

The system should be available to be used at all times and it should address a user's request in a matter of seconds consistently.

4.2. Usable

The system should be usable at all times and handle multiple requests one after another.

4.3. Compatible with macOS & Windows

The system should perform the same regardless of macOS or a Windows operating system.

4.4. Works without Internet Access

4.5. Response Time

The system should take several seconds to launch and to convert the input into tablature.

5.0 Use Cases

Use Case 1

Title: Preview Sheet Music

Primary Actor: Customer

Success Scenario:

1. Users input text-based tablature into the application.
2. The application identifies the instrument the tablature is based on.
3. The application converts the tablature into Sheet Music.

Precondition: Music tablature must be inputted correctly / error-free.

Post-condition: Music is displayed in tablature format

Exceptions:

2a: Unidentifiable Instrument

i) Relaunch Application

ii) Create Github Issues

2b: Undesired Output

i) Adjust aesthetics in the settings menu in trying to get a more desired output

Use Case 1

Title: Download MIDI file

Primary Actor: Customer

Success Scenario:

1. Users input text-based tablature into the application.
2. The application identifies the instrument the tablature is based on.
3. The application converts the tablature into Sheet Music.
4. The user clicks “Preview Sheet Music” to preview the Sheet Music.
5. In the new GUI that appears, there is an option that allows the user to download the sheet music onto their machine.

Precondition: Customer inputs correct / error-free tablature in main GUI.

Post-condition: Midi file gets downloaded in a specified location

Exceptions:

2a: Unidentifiable file

i) Relaunch Application

ii) Create Github Issues

2b: Undesired Output

i) Adjust the input tablature to your liking

Use Case 1

Title: Playing Music

Primary Actor: Customer

Success Scenario:

1. Users input text-based tablature into the application.
2. The application identifies the instrument the tablature is based on.
3. The user clicks on the button labeled “Play MusicXML”.
4. The user clicks “Preview Sheet Music” to preview the Sheet Music.
5. A new GUI appears with an interactive button that allows the user to play the audio.

Pre-condition: Tablature must be inputted correctly and error-free.

Post-condition: Music audio is then played flawlessly

Exceptions: 2a: Music is played incorrectly

i) Some features are still in development, desired functionality may not be supported yet.

Use Case 2

Title: Move through Audio output

Primary Actor: Customer

Success Scenario:

1. Users input text-based tablature into the application.
2. The application identifies the instrument the tablature is based on.
3. The user clicks on the button labeled “Play MusicXML”.
4. A new GUI appears with an interactive slider that allows the user to control the playback of the Audio.

Precondition: Tablature must be inputted correctly and error-free.

Post-condition: Music is scrubbed to the specified position.

Exceptions: **2a: Music is played incorrectly**

i) Some features are still in development, desired functionality may not be supported yet.

Use Case 3

Title: Print sheet music

Primary Actor: Customer

Success Scenario:

1. User inputs text-based tablature into the application. The application then presents a new menu with the sheet music along with a print menu to print the sheet music by sending it to a printer after completing the interaction with the printer menu the sheet music should be printed.

Precondition: User must have view output tablature, a printer connection, printer ready to print.

Post-condition: The user receives a printed copy of the sheet music

Exceptions:

2a: Default printer not available. The user’s computer needs to have a default printer configured in order to print the sheet music

2b: The default printer has an error while printing. The user will need to Troubleshoot the issue and try again

Use Case 4

Title: Change Aesthetics

Primary Actor: Customer

Success Scenario:

1. User inputs text-based tablature into the application. The application then presents a new menu with the sheet music along with a settings menu to customize the appearance after clicking the preview sheet music button. After modifying the settings and reopening the preview sheet music window, the customizations are reflected in the sheet music display.

Precondition: User must have input valid text-based tablatures, user must have picked one of the default fonts available. The user must then click save then reopen the preview sheet music GUI

Post-condition: The sheet music will have a customized appearance according to the user.

Exceptions:

2a: Invalid font selection. Only the fonts which appear in the dropdown menu are implemented

2b: null font size. The user must make a selection of a font in the otherwise if the user clicks save without inputting a font size a null pointer exception will be thrown when reopening the window

Use Case 5

Title: Go-to-Measure

Primary Actor: Customer

Success Scenario:

1. User inputs text-based tablature into the application. The application then presents a new menu with the sheet music along with a text field that allows the user to input the measure, after inputting the measure number and clicking go the window will move to the desired measure

Precondition: User must have view output tablature, the measure number must exist and be valid

Post-condition: The window moves to and highlights the desired measure

Exceptions:

2a: User inputs a negative integer or decimal value, The user needs to input nonnegative integers in the text field

2b: The user inputs a value that exceeds the number of measures in the piece, the measure number needs to be less than or equal to the number of measures in the music piece

Use Case 6

Title: View music XML

Primary Actor: Customer

Success Scenario:

1. User inputs text-based tablature into the application. Upon clicking the “Show MusicXML” button the user is shown the music xml of the tablature music

Precondition: User must have valid input tablatures

Post-condition: The user will receive the music xml for the inputted tablatures

Exceptions:

2a: Invalid tablature. The tablatures input needs to be valid

5.0 User Stories

1. As a customer, I want to download the sheet music so I can view it outside the application.
2. As a customer, I want to hear the music so I can see if the sheet music is correct.
3. As a customer, I want to be able to control the speed at which the notes are being played.
4. As a customer, I want to be able to print the sheet music.
5. As the Instructor, I want to see an intuitive application so I can navigate with little help from the user manual.
6. As the Instructor, I was to be able to view the source code to create a JUnit test to verify the validity of the software.