TAB2MXL: **Text to MusicXML File Converter**

System Requirements Specification Document

Table of Contents

1.0 Introduction

1.1 Purpose

The purpose of the program is to allow users to input a text file containing guitar, bass, or drums tablature for a song and produce an equivalent MusicXML file that can then be used in various music editing software programs such as MuseScore, Finale, Reaper, etc.

1.2 Intended Audience

The program is intended for beginners to professional musicians and educators alike. It may serve as a useful learning or teaching tool for beginners and educators, or a convenient way to edit text tablature for a musician or artist in their program of preference.

1.3 Project Scope

The program application extends to converting guitar, bass and drums tablature to MusicXML file format. The program will be able to accommodate up to 12-string guitar tablature and up to 6-string bass guitar tablature.

2.0 Overall Description

2.1 Product Perspective

Tab2MXL accepts plain text input, as well as .txt files, and produces MusicXML file. The output file can then be imported into any existing music software that is compatible with MusicXML files, such as Finale, MuseScore, Reaper and many more. This allows a user to easily edit an existing piece of text tablature.

Instrument details:

- The program will be able to recognize the type of instrument the tablature is meant for based on the unique notation detected in the input text file.
- The program can differentiate between percussion, guitar and bass guitar.

Work details:

- The program will prompt the user to insert the name of the piece, as well as the musician or composer. If these fields are left blank, the piece will be given the name of the input text file.

Attribute details:

- The program will prompt the user to enter the time signature and key of the piece. If these fields are left blank, the piece will be given a time signature of 4/4 in C major key.

2.2 User Classes and Characteristics

If the text tablature file contains information above and below the staff include information pertinent to how the piece if played or sounds it will be included in the output MusicXML file. The duration of notes will be reassessed and recalculated for each measure, such that the program will be able to adjust to any changes in the length of each measure.
The user may choose the upload a .txt file into the program or paste the plain text into a text window. Users will be given access to features that allow them to modify certain components of the piece before the program produces the completed output file.

2.2.1 Features Allow the User to Modify the Following:

- Text (in text box before loaded into the program as the input file)
- Text Directions (above and below the staff)
- Tuning

2.2.2 Features Allow the User to Insert the Following:

- Work Title
- Creator (Composer)
- Time Signature
- Key
- Any necessary text directions (arm VII, andantino, etc)

If essential information is missing from the input text and not specified by the user, the program will automatically assign default values to the missing features. Specifications of default values will be outlined in section 2.5.2.

2.3 Operating Environment

- Application will use high-level language written in Java language
- The client will need to have Java installed on their machine to run the program

2.4 Design and Implementation Constraints

2.4.1 User Restrictions

If the program does not recognize a character(s) in the input text file, it will return an error message and produce a text file outlining the problematic characters and the measure number they're located in. The user is expected to manually edit unrecognized characters in the text file before re-uploading.

If uploading an input file, the program will only accept .txt files.

- Tuning for guitar and bass must be present at the beginning of each line of measures
- Drum abbreviations must be present at the beginning of each line of measures. The abbreviations must be in the form mentioned below.

The program will not accommodate customized notation within the lines of the staff beyond the following for guitar and bass tablature:

- Hammer on: h
- Pull of: p
- Slide: s, /, \
- Vibrato: ~
- Harmonics (natural, artificial, tapped): *, [ ], ( )
- Muted strum: x
- Bend: b
- Reverse: r

Drum tablature will only produce the following standard 8 drum pieces (Hi-hat will occupy 2 staff lines):

- Bass Drum: b or Bd
- Snare Drum: SN
- Hi Tom: T1
- Low Tom: T2
- Floor Tom: FT
- Hi-Hat: HH
- Crash Cymbal: CC
- Ride Cymbal: Rd

The program will not accommodate customized notation within the lines of the staff beyond the following for drum tablature:

- Accent (Drum):O
- Flam(Drum):f
- Drag(Drum):d
- Roll(Drum):b
- Strike(Cymbal):x
- Hit hard or loose hi-hat (Cymbal): X
- Open hi-hat (Cymbal): o

2.4.2 Implementation Constraints

Due to inconsistencies in text tablature, the program will sometimes expect information that is not readily available or easily interpreted from the input text file. Such exceptions will require additional input from the user.

If the missing information is not properly modified by the user when prompted, the resulting musicXML file may produce inaccurate tablature or sheet music.

2.5 Assumptions and Dependencies

Information that can be extracted from text tablature and converted into MusicXML files depends on the elements and attributes available in the MusicXML features index:

https://www.musicxml.com/for-developers/alphabetical-index/

In the event there is an aspect of tablature music that does not readily translate to any of the features in the above linked index, it cannot be represented in a MusicXML file.

2.5.1 Text Location

- If the program detects a blank link below the staff, it will start the next measure and the following text will be interpreted as the text above the following measure or the first line of the following measure if there is no additional text.

2.5.2 Default Values

- If the time signature field is left blank during the time of upload, the time signature will default to 4/4.
- If the key field is left blank during the time of upload, the default key will be C major / A minor (numerical value 0 in the circle of fifths)
- If a blank .txt file is uploaded, the program will notify the user with an error message
- Drum tablature will use a staff including the 8 standard drum pieces regardless of whether each piece is used

2.5.3 User Needs and Expectation

- The user will need to input a valid text that contains a tablature that has instruments of a drum set, guitar or bass.
- If the user uploads an invalid text file, the user will be notified that the upload is invalid due the text file not containing a tablature of specific instrument stated above or the text file contains random characters.
- The user expectation is if the file is valid when uploaded then the user will get the expected MusicXML file and can then use MuseScore.

3.0 Use Cases

3.1 Common Interactions

- If the text tab format is valid, the user will be prompted with a file browser window to select where the system will output the MusicMXL file will be saved, as well as the name of the file
- If the text tab format is invalid, the user will be prompted with an error message which will ask the user to provide a different file

3.2 Use Case: Guitar Text Tab to MusicXML Path Requirement

When prompted for file upload, the user selects a text file (.txt) and it's recognized by the system as a guitar text tab.

Preconditions:

- The user will need to input a valid file containing the instrument of a guitar to convert to the expected MusicXML file.
- The user must make sure the text tab does not include any notation not mentioned under section 2.4.1.

Successful Scenario:

1. The user uploads a .txt file
    a. The user uploads a file that is not text format.
    b. The program produces an error window asking the user to upload a .txt file.
2. The program gives the user the option to add a title, musician, time signature and key to the output file.
3. The program converts the .txt file to the equivalent MusicXML file
4. The user saves the produced MusicXML file and imports it into an external music editing software compatible with MusicXML file format.

3.3 Use Case: Bass Text Tab to MusicXML Path Requirement

When prompted for file upload, the user selects a text file (.txt) and it's recognized by the system as a bass text tab.

Preconditions:

- The user will need to input a valid file containing the instrument of a bass to convert to the expected MusicXML file.
- The user must make sure the text tab does not include any notation not mentioned under section 2.4.1.

Successful Scenario:

1. The user uploads a .txt file
    a. The user uploads a file that is not text format.
    b. The program produces an error window asking the user to upload a .txt file
2. The program gives the user the option to add a title, musician, time signature and key to the output file.
3. The program converts the .txt file to the equivalent MusicXML file
4. The user saves the produced MusicXML file and imports it into an external music editing software compatible with MusicXML file format

3.4 Use Case: Drum Text Tab to MusicXML Path Requirement

When prompted for file upload, the user selects a text file (.txt) and it's recognized by the system as a drum text tab.

Preconditions:

- The user will need to input a valid file containing the instrument of a drum set to convert to the expected MusicXML file.
- The user must make sure the text tab does not include any notation not mentioned under section 2.4.1.

Successful Scenario:

1. The user uploads a .txt file
    a. The user uploads a file that is not text format
    b. The program produces an error window asking the user to upload a .txt file
2. The program gives the user the option to add a title, musician, time signature and key to the output file.
3. The program converts the .txt file to the equivalent MusicXML file
4. The user saves the produced MusicXML file and imports it into an external music editing software compatible with MusicXML file format.

4.0 Software Requirements

4.1 Software Interfaces

- The user will need to have Java downloaded on their machine in order to run the program.
- The user will need to install eclipse 2020 IDE in order for the software system to run as well.

5.0 Non-Functional Requirements

5.1 Performance Requirements

- There are no bugs in the system that may prevent current or future users from accessing the application

5.2 Accuracy

- The program will produce a MusicXML file that accurately depicts the music within the text tablature uploaded.

5.3 Accessibility

- The program is available for download and operates on your machine locally. There is no dependency on internet connection of server traffic.

6.0 User Stories

- As a musician, I want to convert a musical tablature to a western musical notation for better understanding
- As a music instructor, the translation of a text file containing tablatures to a MusicXML file can be helpful to teach students music notation.
- As a beginner to learning music notation, I want to convert a tablature to learn the different music notations in different tablatures.
- As a musician, I want to translate a tablature to a MusicXML file and play it in MuseScore so that I can learn how to play that specific song.

7.0 User Interface

The user interface will consist of the of a GUI where it will show the uploaded text file of a tablature and the converted version of the MusicXML file. The interface will not accept any invalid text files and if the user happens to upload an invalid text file, the software system will prompt an error message. The interface will also consist of different features where the user will have the options to choose from in order to produce the designated MusicXML file. Such features include the composer's name of the tablature uploaded, the XML file name, the different keys that the file can play and the time signatures as well. The user will as well have the options to save the edit version of the text file and the MusicXML file