TAB2MXL: Text to MusicXML File Converter

System Requirements Specification Document

EECS 2311 - Professor Tzerpos

February 28, 2021

Group 12

Sara Araibi (215700255)

Vishwa Perera (216155947)

Savneet Gill (217386400)

Nishiket Singh (216521197)

Kaneez Fatima (215711534)

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 Product Features

The entity-relationship diagram below demonstrates how the key features of the program are interpreted from the text tablature in a hierarchical manner. If available in the provided text file, all mentioned attributes will be extracted from the tablature and translated to musicXML file format.

These features are essential to build a properly formatted musicXML file. Removal of any of the major components from the text tablature will result in a corrupt musicXML file.

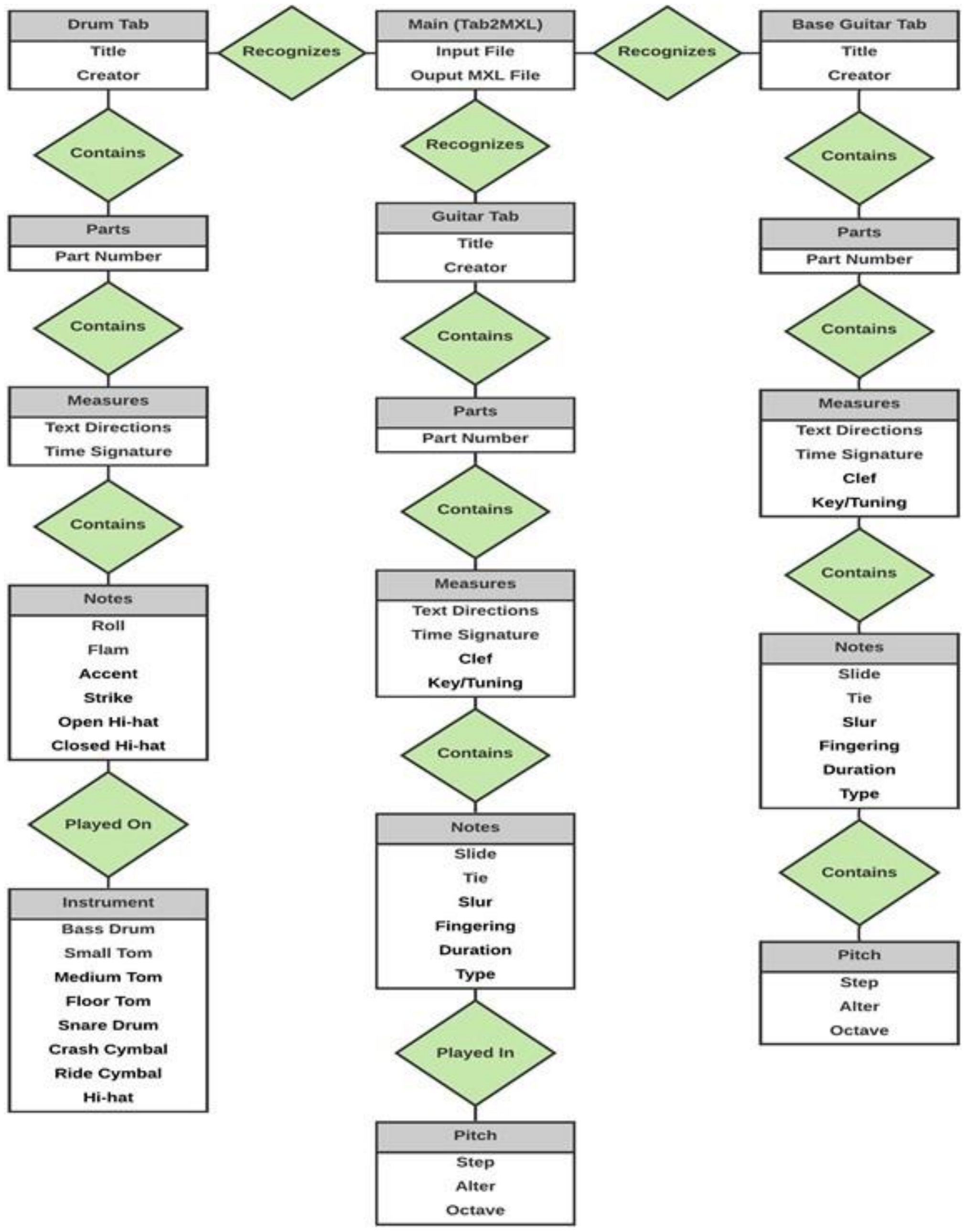

Figure 1: The textboxes are labelled with major components of the tablature (grey). Underneath each component is a list of attributes that may be parsed (if provided) from the input text file (white). The diamonds (green) represent the relationship between the components.

2.3 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. This information may include text direction indicating the tempo above the staff or lyrics and fingering located below the staff. 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.

2.3.1 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
- Lyrics
- Fingering

2.3.2 Features allow the user to insert the following:

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

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 2.6.2.

2.4 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.

2.5 Design and Implementation Constraints

2.5.1 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
- Hi-Hat with Foot: Hf
- 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

2.5.2 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.

2.6 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.

2.6.1 Text Location

- Any sequence of alphabetic characters above the staff will be registered as text direction and will appear in the same way on the output musicXML file.
- 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.

2.6.2 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

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 wil 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 text tab includes the guitar string tuning at the start of each line of measures

- Each measure is separated by a column of vertical bars (|)
- The text tab does not include any notation not mentioned under section 2.5.1.

Successful Scenario:

1. The user uploads a .txt file
    a. The user uploads a file that is not txt 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
    a. The user uploads a blank .txt file
    b. The program produces an error window notifying the user that the file is blank
    c. The user uploads a txt file containing unrecognized notation
    d. The program produces an error window notifying the user that the file contains notation that is not accepted
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 text tab includes the bass string tuning at the start of each line of measures
- Each measure is separated by a column of vertical bars (|)
- The text tab does not include any notation not mentioned under section 2.5.1.

Successful Scenario:

1. The user uploads a .txt file
    a. The user uploads a file that is not txt 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
    a. The user uploads a blank .txt file
    b. The program produces an error window notifying the user that the file is blank
    c. The user uploads a txt file containing unrecognized notation
    d. The program produces an error window notifying the user that the file contains notation that is not accepted
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 text tab includes the drum piece name abbreviation at the start of each line of measures
- Each measure is separated by a column of vertical bars (|)
- The text tab does not include any notation not mentioned under section 2.5.1.

Successful Scenario:

1. The user uploads a .txt file
    a. The user uploads a file that is not txt 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
    a. The user uploads a blank .txt file
    b. The program produces an error window notifying the user that the file is blank
    c. The user uploads a txt file containing unrecognized notation
    d. The program produces an error window notifying the user that the file contains notation that is not accepted
4. The user saves the produced musicXML file and imports it into an external music editing software compatible with musicXML file format

4.0 External Interface Requirements

4.1 Software Interfaces

- The user will need to have Java downloaded on their machine in order to run the program.

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.