TAB-2-MusicXML™ (T.2.M)

System Requirements Specification

V2 - February 24

By EECS2311 Team 4

- Patchanon Suepai
- Suha Siddiqui
- Nobaiha Zaman Rayta
- Zhilong Lin
- Martin Brejniak

Table Of Contents

1. Introduction

a. Purpose

The purpose of this project is to create a format converting program. It will take tablatures of Guitar and Drum music, and convert them into a musicXML file. Tablatures, while useful, can sometimes not be the best way to learn how to play a music piece. This software will allow users to use both the tablature and MusicXML version of a music piece (Considering they found the tablature first) and will support users from different musical backgrounds and expertise.

b. Document Conventions

GUI = Graphical User Interface

UI = User Interface

c. Intended Audience and Reading Suggestions

Please read through the table of contents and thoroughly understand the intended purpose of this software so the system requirements described in this document are understood with clarity.

d. Project Scope

The purpose of this project is to create a format converting program. It will take tablatures of Classical Guitar, Bass and Drum music, and convert them into a musicXML file. The software is officially set to release on April 13, 2021, so testing users can get their hands on it sooner, and begin converting their favourite tablatures into MusicXML files. The software will only be limited to converting three types of instrumental tablature (guitar, bass and non-pitched percussion) but is currently only implemented for guitar.

The developers of the software will meet with the client every week to talk about progress, and take any suggestions they might have.

e. Contact Details

Should the user have any further questions or concerns, they can reach the developers on the email provided in this document. (EECS2311Team4@gmail.com).

2. Overall Description

a. Product Perspective

The TAB-2-MusicXML™ interface system is a java based application that can be downloaded onto a personal computer. The software will allow users to convert tablature made for classical guitar in “.txt” format to musicXML code, which will output as a downloadable file in musicXML format that can be used as sheet music or tablature sheet music. The final output at minimum, will contain work details (Such as titles and musicians) and attributes (Such as time signature and sufficient bars).

b. Product Features

i. User capabilities

a. Uploading the input text file (Tablature):

    i. User will be able to browse their personal file manager

    ii. User can paste tablature into text box

b. Modify small errors present in XML file with GUI.

c. Manual time signature adjustments.

d. The user can input classical guitar tablature.

e. Manually enter title and composer.

f. Able to download the final XML code.

g. Able to back up and/or cancel at any time.

h. The user can return to home once they have processed a single tablature.

i. The user has the option to go back to the "optional" composer name and work title page.

c. User Classes and Characteristics

i. User function

1. The user can use GUI to personalize the final output to preferences.
2. The user can choose whether to upload the file to the converter, or to simply paste the text into the input box.

ii. Admin function

1. Developers will be able to access code, and customers will not have access to alter functionality of the final marketed product.

iii. Robustness

1. .txt and not .txt in file upload.
    a. Users can enter a .txt file by browsing and choosing a .txt file or pasting a .txt file.
    b. Users can enter a non .txt file and will be notified with an error message that will allow them to change their input or restart the software.
    c. Users can enter an invalid input into a text file and will be notified with an error message that will allow them to change their input or restart the software.

iv. Usability

1. The user will be given a user-manual in order to gain a grasp on the functionality of the program.
2. The user will also be able to perform clicking options and editing when it comes to changing the tablature.

d. Operating Environment

i. *Operating system:* Windows/Mac

ii. *Platform:* Java

iii. *Language:* English

iv. Downloadable desktop application

e. Design and Implementation

The program will have some constraints which will could potentially affect user accessibility:

1. There is no control over the stem direction of each note.
2. Imported note heads and orientation will not be implementable, the notes that are embedded into the program are the only note heads and orientation available to the user.

f. Assumptions and Dependencies

i. Assumptions

1. The program will assume the user is entering tablature which follows the format fit for the program described in the user manual.
2. The installation environment will be on a windows OS or macOS.
3. The user will enter music that does not require a change in note head shape.
4. The user will enter music that does not require a change in time signature and enter music that is in 4/4 time.
5. The key will be defaulted to A minor or C major unless specified otherwise.

ii. Missing information

1. The user can rely on the software to fill in the following component if they chose to not upload:
   - *a. Missing time signature:* Will set default to 4/4 time.
   - b. *Missing composer/musician/title:* Will default to a blank space.
   - *c. Blank tablature file or entry:* Will return 32 measures of whole notes, in 4/4 time and treble clef.

iii. Instrument assumptions

1. The user can only expect viable output when entering tablature for classical guitar.
2. Classic guitar will always be in treble clef.
3. If the instrument is unidentifiable the program will default to treble clef.

iv. Syntax

1. The user can enter music that includes special syntax that could represent components like ties, slurs, hammer-ons and pull offs in tablature but will not display such a result in the musicXML code.

3. System Features

a. Functional Requirements

Functional requirements of the system are that it should be able to take in “.txt” music tablatures in either pure text form via copy and pasting, or via a text file upload from the user. Afterwards, it will attempt to parse and convert the text tablature into a musicXML file by looking for coherent syntax inside the text, then reading and converting that into its respective MusicXML form. If there are any errors or conflicts in parsing, it will prompt the user to manually enter in any missing values. The program will then alter the XML file to include the users changes. Afterwards, the program will save the output to an XML and give the user a musicXML file, which they can then save to their computer, and open it in a compatible program.

b. User Stories

- As a self taught music student, I want to learn how to play a song that I’ve only found tablature versions of, so I can further my music knowledge and become better.
- As a music teacher, I want to be able to convert my tablature music into MusicXML so my student can get a better understanding of the piece.
- As a music composer, I want to be able to convert my tablature music to MusicXML so the music I create can reach a broader audience.

c. Use Cases

- A music teacher found a text tablature online but they prefer to use a more traditional music sheet layout rather than text tablature, as they find that their students learn better that way. So, they would want to convert it. They downloaded a tablature in ".txt" format and uploaded it to the program. They then choose where on their computer to put the newly converted file.

- A music student finds a tablature they want to convert to a MusicXML file. They copy and paste it into the input window and click convert. However, they accidentally inputted a few characters into the input window unknowingly. The program lets them know that the tablature they inputted didn't work, and to try again. They then get a fresh copy of the tablature, and this time they don't make the same mistake. They then choose where on their computer to put the newly converted file.

4. Graphical User Interface Requirements

The software has a GUI that will greet the user and prompt the user to press the start button. Once the button is pressed it will allow the user to insert the title of the tab and its composer. The insertion of the title and the composer of the song should be optional.

Upon hitting enter it will give the user to open a tablature file with the browser button and edit it or simply just paste it in the text editor box. If the file isn't in the correct format the software will give an error message.

Once the correct file is inputted, the software will convert the file into a MusicXML file and the GUI will show the XMLfile and give the user options to either restart by pressing the home button, save the file or exit the software.

In summary, the user will be able to navigate through the program with the usage of buttons and scrolling with the instructions provided on each screen.

5. Nonfunctional Requirements

a. Performance Requirements

The .txt file input must be in tablature format. Any other .txt that is inputted will give an error message.

b. Safety Requirements

The file entered needs to be complete and readable.

c. Security Requirements

The only two files that should be entered in the program:

1. A file that is copied and pasted from an official website or any website that can be trusted.
2. The user can input their own file designed in the required format.

The user can not read the functional codes. The musicXML file can not be used in any illegal fields.

d. Software Quality Attributes

The software will be easy to use, and the time needed to convert for the musicXML file will be short by design. The software is free for anyone to download and the code is open sourced and available on Github.

6. References

- “Hello World: A One-Bar Song with a Whole Note on Middle C in 4/4 Time.” MusicXML, 15 Dec. 2017, www.musicxml.com/tutorial/hello-world/.
- Bandakkanavar, Ravi, et al. “Software Requirements Specification Document with Example.” Krazytech, 17 Oct. 2019, krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database#:~:text=A%20Software%20Requirements%20Specification%20(SRS,a%20project%2C%20software%20or%20application.&amp;text=This%20includes%20the%20purpose%2C%20scope,hardware%20requirements%20of%20the%20project.