TAB 2 MUSICXML™

System Requirements Specification

V3 - April 13

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 file format converting system. It will take tablatures of Guitar, Bass, and Percussion music, and convert them into a musicXML file format. 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 the 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

We recommend you 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

Only Classical Guitar, Bass and Percussion music will be supported. 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 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, bass and drums in ".txt" format to MusicXML code, which will output as a downloadable file in musicXML format. The final output at minimum, will contain work details (Such as default blank titles and musicians) and attributes (Such as time signature and sufficient bars) when a blank tablature set of strings is uploaded such as the following.

```
|-----------------|-----------------|
|-----------------|-----------------|
|-----------------|-----------------|
|-----------------|-----------------|
|-----------------|-----------------|
|-----------------|-----------------|
```

b. Product Features

1. Uploading the input text file (Tablature):
    a. User will be able to browse their personal file manager
    b. User can paste tablature into text box
2. Modify small errors present in XML file with GUI.

3. Go back to make manual adjustments after the final output.
4. The user can choose classical guitar, bass, or percussion tablature.
5. Able to download the final XML code.
6. Able to back up and/or cancel at any time.
7. The user has the option to go to the “optional” composer name, work title page, change the time signature and key.

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. Robustness

1. Users can enter a “.txt” file by browsing and choosing a “.txt” file or pasting a “.txt” file.
2. 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.
3. Users can go back edit tablature even after converting.

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 time signature will be defaulted to 4/4 time unless specified otherwise.
5. The key will be defaulted to A minor or C major unless specified otherwise.
6. The Composer and piece's Title will be defaulted to blank 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. Classic guitar will always be in treble clef.
2. If the instrument is unidentifiable the program will default to treble clef.

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 with an error message indicating the type of mistake they made. Once the user has made appropriate changes to match passable formatting for 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

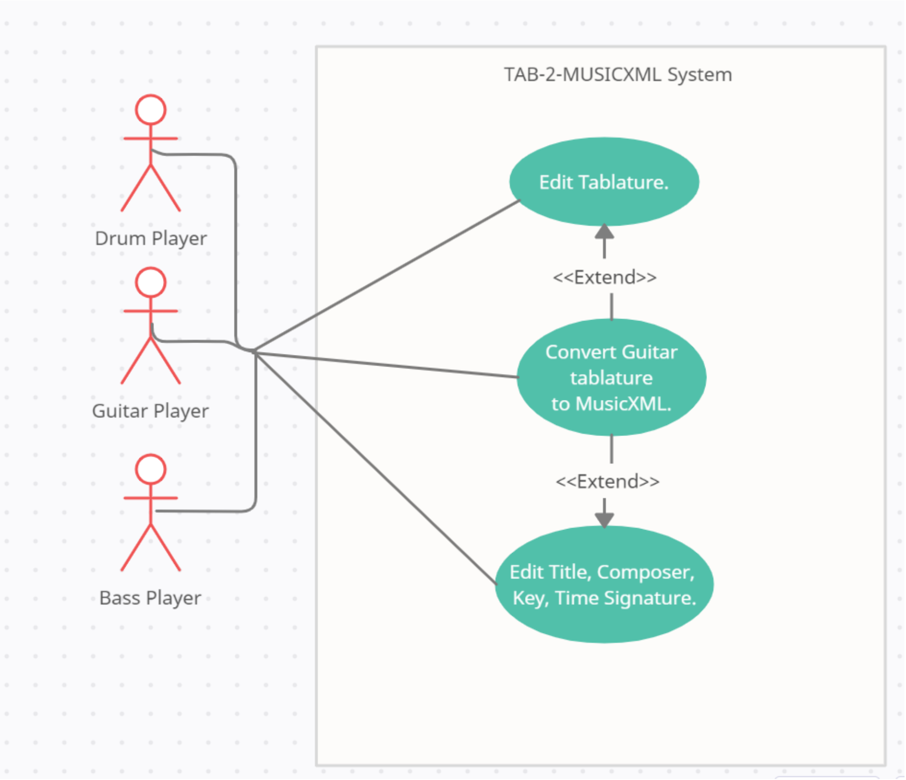
- As a self taught music student, I’m more comfortable reading musical notes than text tablature. I want to learn how to play a song that I’ve only found tablature versions of, I’d like to be able to convert it to musical notes so I can further my music knowledge and become better.
- As a music teacher, 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.
- As an amateur music maker, I want to be able to easily edit existing music tabs so that I can create some new music by editing existing pieces.

c. User Scenarios

- 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 input 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.
- An amateur music maker finds a text tablature online and they want to try and edit the music to make a remix. The program lets them edit the text tablature and convert it to musicXML so that they can share their new creation.
- A guitarist finds an online website that has a collection of text tablatures, but they feel more comfortable with reading music notes. They can copy and paste the tabs on those sites and paste it into the program so that it can convert it. This allows the guitarist to read the music in their most comfortable form.

d. Use Cases



- Use Case: A user wishes to convert a text tablature into musicXML format
  - Steps:
    - User imports the text tab into the program
      - User can choose to paste the text tab
      - User can choose browse the system for their text tab file
    - User checks the text tab in the program
      - User can also edit the text tab
      - User can also add metadata modifications
    - User converts the tab
    - User receives the converted tab
    - User can save the tab to their computer
      - User can also choose to edit the tab if unsatisfied

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 upload their tablature via browsing their file manager or pasting directly into the text box. There will be a button next to the ‘Confirm’ button, called ‘Modifications’ which will opeitonally allow the user to make small changes to their final output. On the Modifications' window the user should be allowed to add the Composer, the music piece's Title, change the Time signature and Key. To reiterate, the modifications page is optional to the user andis upto their preference, otherwise it will be defaulted to settings explained above. Once the user is satisfied with the modifications, they can hit the 'Confirm' button and will return back to the Upload page. If the file isn't in the correct format the software will give an error message. The user can make edits within the text box to the tablature they uploaded.

Once the correct file is inputted, the software will convert the file into a MusicXML file, the GUI will show what that file contains and give the user options to:

- Restart by pressing the home button.
- Save the file.
- 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. Safety Requirements

The file entered needs to be complete and readable.

b. 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.

c. 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.
- O'Connor, M. (2019, December 09). How to read drum tabs - quick start guide. Retrieved April 13, 2021, from https://www.electronicdrumadvisor.com/how-to-read-drum-tabs-beginners/#:~:text=Each%20bar%20is%20separated%20with,or%20cymbal%20is%20being%20played.&amp;text=On%20the%20%E2%80%9Ctimeline%2C%E2%80%9D%20%E2%80%9C,an%20accent%20or%20stronger%20hit