Requirements Document

TablatureConverter

EECS 2311 – Group 6

Tuan Dau

James Le

Temiloluwa Odunfa

Mohammad Amin Mohammadi

02/03/2021

Table of Contents

1. Introduction

1.1 Purpose of Document

The purpose of this document is to outline the description of the Tablature Converter software, the project's functionality, the technologies being used, and use cases and functional requirements that the Tablature Converter project must support.

1.2 Project Summary

The project will consist of a software system that converts a text input, either a .txt file or copy and pasted text input, containing guitar, bass, or drum tablature for a song, into a MusicXML file.

1.3 Background

Although using text tablature is ideally a great way to learn a new song, it cannot be adjusted to a specific key and is sometimes hard to read. There are many alternative formats that do not have these issues , the majority of songs are found in text tablature form. The music research community developed a format called MusicXML where it can accurately denote tablature. Currently there is no way to convert text tablature files to MusicXML.

Tablature notation is a way of writing down an instrument's parts in an accessible manner. The tablature in the context of this project will refer to guitar, bass, and drum tablature. Although tablature is widely used for its ease of use, most tabs exist in a text format, and there does not exist an easy way to turn guitar tab text into a more featured digital file format.

Examples of tablature are listed as follows: Guitar tab, bass tab, drum tab.
In guitar and bass tablature, there exists a number of different techniques that can be notated (definitions here). A comprehensive list of currently supported techniques and their symbols are listed below:

Hammer-ons: h

Pull-offs: p

Ghost notes: x or X

Slides: 1/2, or 1\2 where 1 indicates the starting note and 2 indicates the destination note of the slide. 1/2(note the '/') is conventionally for sliding up the fretboard, and 1\2 (note the '\') is conventionally for sliding down the fretboard, although they may be written interchangeably.

Bends: 1b2, where 1 is the starting note, and 2 is the note that is bent to.

1.4 System Purpose

The software product will consist of a desktop application that, upon the upload of a tablature in *.txt form, will detect which instrument is being described by tablature, and convert the tab to a MusicXML file format.

2. System Features and Requirements

2.1 System Features

Users will be able to upload their text file input.

Users will be able to copy and paste their input.

The application will output a MusicXML file format of the tablature input.

2.2 User Interfaces

Front-end software: JavaFX

Back-end software: Java

2.3 Hardware Interfaces

Windows and Mac operating systems

2.4 Functional Requirements

[DEMO-SRS-001] The application will allow users to upload a .txt file of guitar, bass, or drum tablature to convert into a MusicXML file format.

[DEMO-SRS-002] The application will automatically detect which instrument the upload tablature belongs to.
[DEMO-SRS-003] The application will show users the converted MusicXML file format after conversion as a file preview.
[DEMO-SRS-004] The application will allow users to change the output format of the MusicXML file after conversion in the 'preferences' menu.
[DEMO-SRS-005] The application will allow users to copy tablature from a website and paste it into the system as an alternative to uploading a .txt file.
[DEMO-SRS-006] The application will produce a fully correct MusicXML file, with the exception of note timings, granted that the tablature input is valid. A valid tablature format is defined below:

- Each line of the tablature starts with a letter indicating the note that the string or tuned to, or the designated drum kit piece, immediately followed by the start of a measure
- Measures are separated by vertical lines indicated by either a lowercase L (l) or with the vertical bar glyph (|). Note that multiple measures are not required for a tab to be converted properly.
- Each line, if not consisting of a note or technique (supported techniques are listed in section 1.3) must be the start of a measure (l or |), or a hyphen (-).

2.5 Non-Functional Requirements

Security

Reliability

Performance
- This program will operate quickly, with a conversion time taking less than 3 seconds
- The efficiency of this program depends on the length of th given tablature. The longer the tablature the longer the conversion process.

Maintainability

Scalability
- Tablature Converter supports guitar and bass. Drum's will be added in the near future

Usability

- The uploaded text file should be formatted correctly
- The uploaded text file needs to be complete and readable
-

3. Use Cases

3.1 – File Conversion

Use Case Name:	File Conversion
Primary Actor:	Customer
Summary:	The user's uploaded text file (.txt) should be converted into a MusicXML file.
Basic Flow:	1. The use case starts on program start up. 2. The system requests a text file that contains either guitar, bass, or drum tablature. 3. The system verifies the text file and detects the instrument. 4. The system converts the text file into a MusicXML file. 5. An output preview of the conversion is displayed.
Alternative Flows:	Step 3: If a file other than a .txt is given, the program throws an error to the user and goes back to step 2. Step 3: If neither a guitar, bass, or drum tablature is detected, the program throws an error and goes back to step 2.

3.2 – User Tab Customization

Use Case Name:	User Tab Customization
Primary Actor:	Customer
Summary:	Allows the user to customize the text tablature
Basic Flow:	1. The use case starts when a text file is uploaded to the

	system 2. The file contents are displayed in the textarea 3. Users can customize tablature within the textarea
Alternative Flows:	Step 2. If invalid input is given an error will be sent and the program goes back to step 1. Step 3. If invalid input is given an error will be sent and the program goes back to step 1.

3.3 – Save Converted XML File

Use Case Name:	Save Converted XML file
Primary Actor:	Customer
Summary:	Allows the user to save the converted XML file to local file explorer
Basic Flow:	1. The use case starts when the user uploads a text tablature file 2. The use case starts when a text tab file is being converted. 3. The user can now save to file explorer
Alternative Flows:	Step 2. If the conversion does not compile then the program goes back to step 1.

3.4 – Open XML Conversion in Browser

Use Case Name:	Open XML Conversion in browser
Primary Actor:	Customer
Summary:	Allows the user to customize the text tablature
Basic Flow:	1. The use case starts when the user uploads a text

	tablature file 2. The use case starts when a text tab file is being converted. 3. The user can now open XML contents in browser
Alternative Flows:	Step 2. If invalid input is given an error will be sent and the program goes back to step 1.