Software Requirements Specification

Version 1.0

February 26, 2021

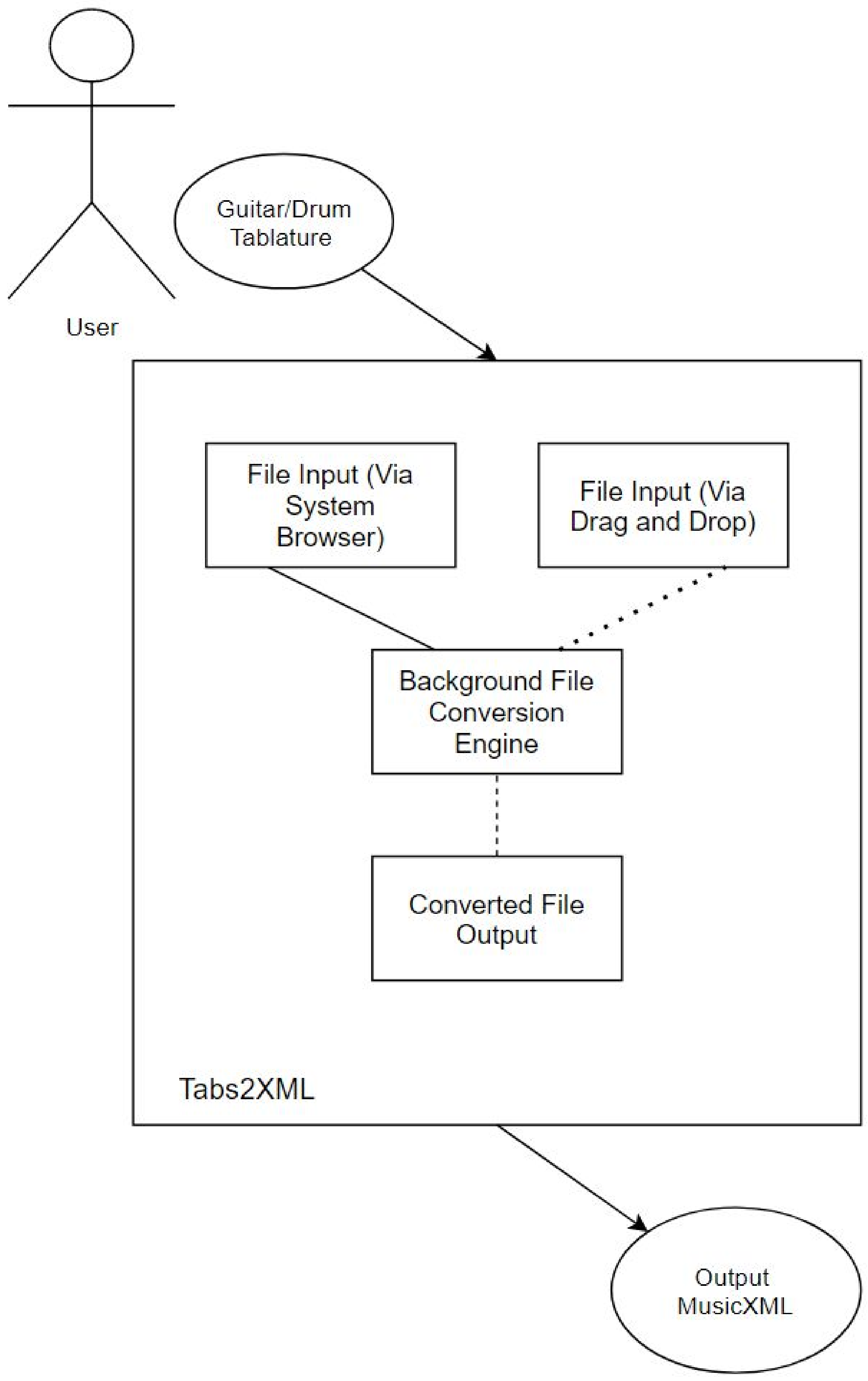
Tabs2XML

Alp Sirek
Andrew Ngov
Arjit Johar
Daniel Santorelli
Muhammad Azizi

Submitted in Partial Fulfillment of The Midterm of EECS 2311 Software Development Project

Table of Contents (Click to go to section)

1.0 Introduction



User
Guitar/Drum Tablature
File Input (Via System Browser)
File Input (Via Drag and Drop)
Background File Conversion Engine
Converted File Output
Tabs2XML
Output MusicXML

Figure 1 - Diagram aid for sections 1.1, 1.3, and 1.4. Depicts the core flow the User would go through and a little behind the scenes of what the Tabs2XML is doing in the background.

1.1 Purpose

The purpose of this document is to present a detailed description of the Tabs2XML system. This document will explain the purpose and features of the system, the interface of the system, what the system will do, the constraints under which it must operate, and how the system will behave when interacted with.

Tabs2XML is being developed for the purpose of converting guitar tablature files and drums tablature files into MusicXML files. Due to the relatively new format, there aren't many music pieces written in MusicXML. While tablature for guitars and drums are easy to understand, they offer a low degree of readability and modification. The MusicXML format builds on these shortcomings to allow readers to better understand the music piece and easily play it. Tabs2XML is being developed for those who want to play songs in the format of MusicXML, but find that they can only find the tablature of those pieces (common occurrence as there aren't as many pieces of song in the MusicXML format). Tabs2XML is also being developed for those that would like to play their songs in different keys. Tabs2XML will have implemented features to allow the user to change the key of their songs (given that the song being played is locally stored and that it is a file of type tablature or MusicXML).

1.2 Intended Audience

Tabs2XML is being built for guitarists and drummers. These two groups of people will get the most out of our software as they would have access to another format to play songs from – one that can more easily read and modify. With our application they can convert their existing tablature collection or new tablature they get to MusicXML. Since MusicXML allows the reader to modify the key of the song they are playing, Tabs2XML is also intended for such people.

While guitarists and drummers have the most to gain, others like music teachers, students learning how to play the guitar or drums, and anyone wanting to convert tablature (for guitars and drums) to MusicXML will also benefit from the use of Music2XML.

1.3 Intended Use

The main use of MusicXML is its capability to convert tablature of guitars to MusicXML. To add more value for the users of the program, there are built in features to help users find tablature for songs that are not already in the MusicXML format, resources for those hoping to learn more about the Music XML format, and Tutorials to help users better understand the capabilities of the MusicXML format. In the event that the User incurs any troubles that are not dealt with in our resource section, they are encouraged to research the issue independently.

1.4 Scope

The software system will be a file conversion system which will turn locally stored TABLATURE into MUSICXML files. The system is designed to help musicians who prefer to view their music in music sheet form but only have access to the tablature for a given song.

The system will meet the customers needs while remaining easy to understand and use. To elaborate, this system is designed to allow a user of the system to convert locally stored TABLATURE for guitar and drums to MUSICXML. The conversion is also locally stored. The system will also be implemented with a resource section with helpful links.

Definitions and Acronyms

Term	Definition
MusicXML / XML	The modern file format for editable music that Tabs2XML converts text to.
Tablature Format / Tab	The rudimentary, text-based music format that Tabs2XML converts from.
UI	User interface. Aspects of the program that the user can see and directly interact with.
Conversion Engine	The algorithm that Tabs2XML runs to convert tablature to MusicXML

References

Krüger, Nico. "How to Write a Software Requirements Specification (SRS Document)." *Perforce*, Perforce Software, 23 October 2018, https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document. Accessed 26 2 2021.

2.0 Overall Description

2.1 User Needs

Tabs2XML is a public program which will have many consumers with a wide range of users-. Because of this, it is important for it to be very flexible in how it can accomplish its task while remaining user friendly. Further needs of the user is discussed in detail in the Intended Audience section (1.2).

2.2 Assumptions and Dependencies

- The tablature the user is trying to converted is of a type that is supported
- The user has access to the internet (Tabs2XML will work without the internet but with limited functionality)

2.3 User Stories

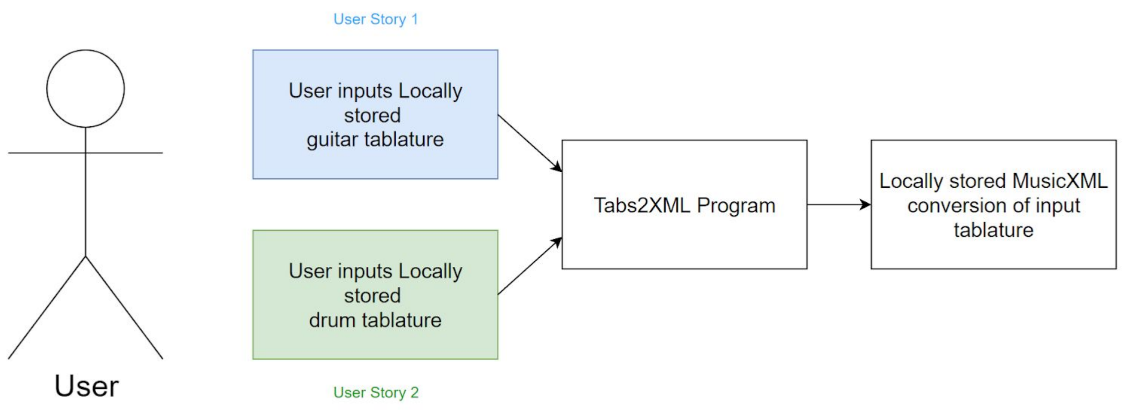

User Story 1
User inputs Locally stored guitar tablature
User inputs Locally stored drum tablature
User Story 2
User
Tabs2XML Program
Locally stored MusicXML conversion of input tablature

Figure 2 - Diagram aid for User Story 1 and User Story 2. Depicts the steps the User would go through for a successful interaction with Tabs2XML.

Title: User Story 1

Primary Actor: User (can be anyone described in the Intended Audience section (1.2))
Success Scenario:

1. User imports guitar tablature into Tabs2XML.
2. Tabs2XML identifies which instrument the tablature is for (either guitar or drums).
3. Tabs2XML sends the input through the conversion engine.
4. User then saves the output MusicXML file anywhere on their local device.

Precondition: User has locally stored tablature for drums and User has enough local storage for the converted file (~10kb).
Extensions: In the event that the User incurs any difficulties while using Tabs2XML, there are several support systems available to offer them help. This includes detailed error messages and a troubleshooting section in the user manual for Tabs2XML. In the event that they require further assistance, they can react out to our support staff at hiangel@my.yorku.ca

Title: User Story 2

Primary Actor: User (can be anyone described in the Intended Audience section (1.2))
Success Scenario:

1. User imports drum tablature into Tabs2XML.
2. Tabs2XML identifies which instrument the tablature is for (either guitar or drums).
3. Tabs2XML sends the input through the conversion engine.
4. User then saves the output MusicXML file anywhere on their local device.

Precondition: User has locally stored tablature for drums and User has enough local storage for the converted file (~10kb).
Extensions: In the event that the User incurs any difficulties while using Tabs2XML, there are several support systems available to offer them help. This includes detailed error messages and a troubleshooting section in the user manual for Tabs2XML. In the event that they require further assistance, they can react out to our support staff at hiangel@my.yorku.ca.

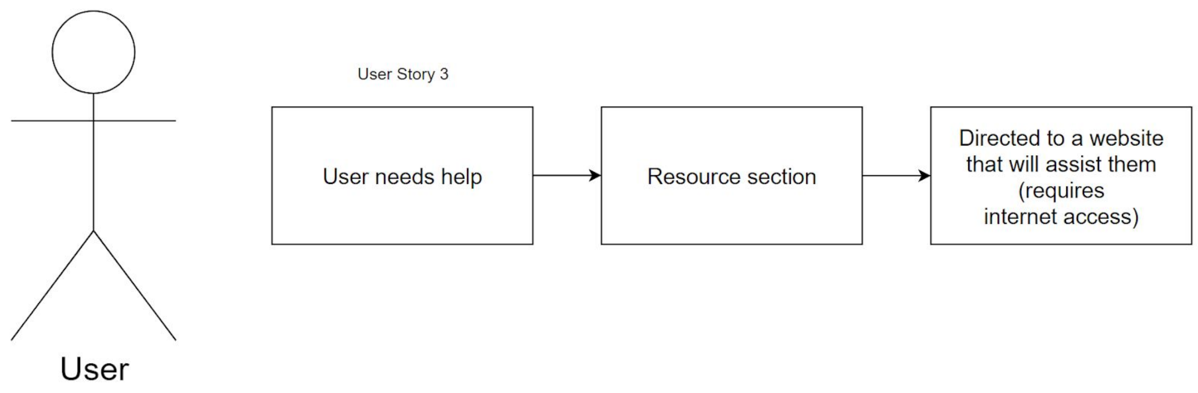

User Story 3
User needs help
Resource section
Directed to a website that will assist them (requires internet access)
User

Figure 3 - Diagram aid for User Story 3. Depicts the steps the User would go through if they needed help from the resource feature of Tabs2XML (this feature is described extensively in the Intended Use section (1.3))

Title: User Story 3

Primary Actor: User (can be anyone described in the Intended Audience section (1.2))
Success Scenario:
1) User requires assistance (the cases and extent of assistance Tabs2XML offers is detailed in the Intended Use section (1.3))
2) User navigates to the Resource section and finds an appropriate link to asset them
3) User is then directed to a resourceful site which which will help/guid/assets/teach the User
Precondition: User has an internet connection and is able to navigate web pages.
Extensions: In the event that the User incurs any troubles that are not dealt with in our resource section, they are encouraged to research the issue independently.

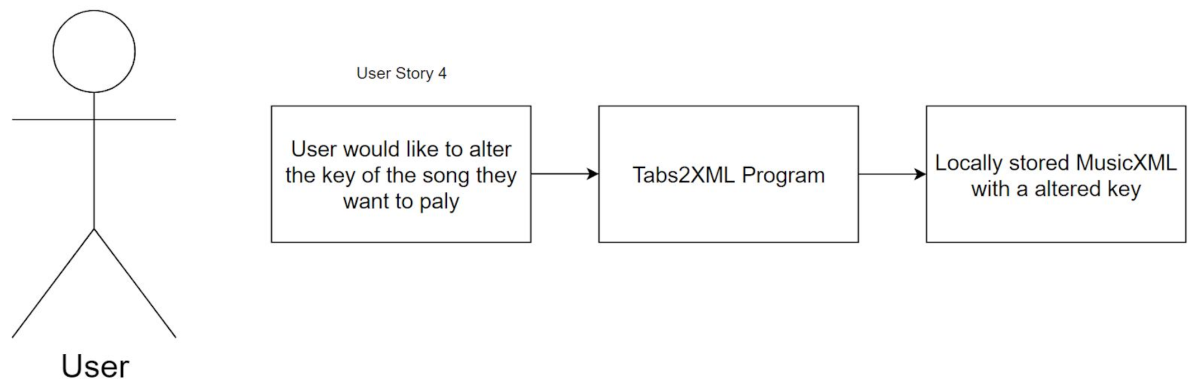

User Story 4
User would like to alter the key of the song they want to paly
Tabs2XML Program
Locally stored MusicXML with a altered key
User

Figure 4 - Diagram aid for User Story 4. Depicts the steps the User would go through if they wanted to change the key in which their song is played in.

Title: User Story 4

Primary Actor: User (can be anyone described in the Intended Audience section (1.2))
Success Scenario:
1) User wants to play a song, but would like the key that it is played in to be changed.
2) User inputs their file and chooses which key they want to have the converted file in.
3) User then saves the converted file to their local system
Precondition: In the event that the User incurs any difficulties while using Tabs2XML, there are several support systems available to offer them help. This includes detailed error messages and a troubleshooting section in the user manual for Tabs2XML. In the event that they require further assistance, they can react out to our support staff at hiangel@my.yorku.ca.

3.0 System Features and Requirements

3.1 Functional Requirements

- System allows users to import a tablature (either drum or guitar) to be converted.
- System is able to convert tablatures (only for guitar and drums) into MusicXML files.
- MusicXML file converted can be used in other applications that can view MusicXML files as music sheets.
- System is able to identify if the input is a guitar or drum tablature.
    - Users can also choose whether it will be a guitar or drum tablature.
- System can identify notes, chords, frets, strings, etc when trying to convert into a MusicXML file.
- System will give an error if tablature is not properly formatted or it is not a drum or guitar tablature.
- System is able to run Python programs. Please refer to the following link for a list of OS system requirements in order to execute programs: https://docs.python.org/3.9/faq/general.html

3.2 External Interface Requirements

- Users can change time signature/measure, give a name to the converted tablature, etc.
- In the event that the user would like to access the resources in the resources section, it is required that they have access to the internet.

3.3 System Features

- Browse button to select the directory where the tablature(s) are in.
- List of files in the directory which can be selected.
- Display on the right side which shows a preview of the file that is inputted.
    - Display changes the preview of MusicXML file when the file is converted.
- An input area to enter the name of the piece.
- List of time signatures to select from(1/4, 2/4, 3/4, 4/4)
- Another browse button to choose the directory that the converted MusicXML file will be saved in.
- Convert button which converts the file to MusicXML and saves it in the directory that was chosen.

3.4 Nonfunctional Requirements

- The file that is to be inputted to convert to MusicXML should be in txt format.
- The file that is to be inputted should be either a guitar or drums tablature.
    - Many different tablatures should be able to be used.
- The file should be formatted correctly and contain no errors.

3.5 Assumptions and Dependencies

- Text files that are input by the user will only be of drums and guitar tablatures.
- That the text imported is a tablature of correct formatting and design.
- Everything such as beat, key, etc. will be set to system default unless specified by the user.
- Due to the variety of forms that tablature comes in, Tabs2XML will have a list of acceptable forms that are guaranteed to work. Other forms may be used, but may result in an incorrect output conversion.

4.0 Timeline

February 3: Completion of Requirements Document

February 17: Completion of GUI interface and parser.

February 28: Completion of Music XML conversion for at least one instrument along with the first rendition of the user manual

March 16: Completion of MusicXML conversion for all instruments with little to no errors.

March 28: Debugging system complete

March 30: Draft of Design Document must be submitted

April 13: Final submission