EECS2311: SOFTWARE DEVELOPMENT PROJECT

TAB2XML

System Requirement Specifications

April 11, 2022

PREPARED FOR

Vassilios Tzerpos
Lassonde School of Engineering

PREPARED BY

Hiba Jaleel - 215735020
Kuimou Yi - 216704819
Kamsi Idimogu - 216880288
Maaz Siddiqui - 216402927

Table of Contents

1. Introduction

1.1 Purpose

The purpose of the Software Requirement Specifications document is to describe the functions our software, TAB2MXL, the MusicXML previewing system will be able to perform and how exactly our software will be able to achieve the desired results.

1.2 Intended Audience

This document is intended for:

Customer/Client: The customer or client will be expressing what features they would like the software to have, and through this document ensure that the software will cover the aspects they would like.

Project Manager: The project manager will be a point of contact for any conveyance between the developers and the customer/client and communicate any new demands or changes in the customer/client's requirements. The project manager will also be setting deadlines for the developers to follow and assist in maximising developer efficiency.

Developers: The developers will be working on implementing the requirements of this software as laid out by the customer/client through their development of the software.

Musicians (also referred to as user/users): Musicians are defined as people who enjoy listening to music and partake in playing text-based tablature music. They will be the end-users of this software.

1.3 Intended Use

The software development timeline of the TAB2XML is from January 10, 2022 to April 11, 2022. The main costs involved in this project are time, energy, and electricity. The software's ability to perform the required tasks will be the measure of success. Our team has determined the core features/functionality, important features, and nonfunctional requirements to be the deliverables of this project. Risk will be evaluated by grading it's exposure, probability of occurrence, and loss size (in days). Based on the context and severity of the risk, effective mitigation measures will be implemented.

2. Overall Description

2.1 Product Perspective

TAB2XML can be accessed through GitHub, under the following repository: https://github.com/CCSCovenant/TAB2XML/releases

2.2 Product Overview

TAB2XML will allow users to convert their text-based tablatures into music sheets, which users will then be able to play. Additionally, users will have the ability to customise the look and feel of the music sheet, along with access to changing the tempo of the music and playback features.

2.3 User Class Characteristics

2.3.1 Use Case #1

Title*:* Preview Music Sheet
Primary Actor: Musician
Preconditions*:* The musician has started the application and input the text-based tablature music.
Success Scenario*:* The system displays sheet music. The musician previews the music sheet.
Alternative Flow: The system displays sheet music. The musician adjusts the spacing of the sheet music to his/her liking. The musician previews the music sheet.

2.3.2 Use Case #2

Title: Saving Music Sheet
Primary Actor: Musician
Preconditions: The musician has started the application, input the text-based tablature and previewed the music sheet.
Success Scenario: The musician saves the music sheet.
Exception: The music sheet does not save.

2.3.3 Use Case #3

Title: Play Music
Primary Actor: Musician
Preconditions: The musician has started the application, input the text-based tablature and previewed the music sheet.
Success Scenario: The system identifies the type of instrument that the inputted text-based tablature represents. The system plays the music.

Alternative Scenario: The musician inputs the desired tempo. The system identifies the type of instrument that the inputted text-based tablature represents. The system plays the music.
Exception: The system does not identify the type of instrument that the text-based tablature represents.

2.3.4 Use Case #4

Title: Musician Selects Where Music Plays From
Primary Actor: Musician
Preconditions: The musician has started the application, input the text-based tablature and previewed the music sheet. The system identifies the type of instrument that the inputted text-based tablature represents. The system plays the music.
Success Scenario: The musician selects the note that they wish to hear the music begin from. The music continues from that note.
Alternative Flow: The musician inputs which measure they want the music to begin from. The music continues from that measure.
Exception: The system does not identify the type of instrument that the text-based tablature represents.

2.3.5 Use Case #5

Title: Play Music on Repeat
Primary Actor: Musician
Preconditions: The musician has started the application, input the text-based tablature and previewed the music sheet. The system identifies the type of instrument that the inputted text-based tablature represents.
Success Scenario: The system plays the music. The musician enables repeats. The system repeats the music.
Alternative Flow: The musician enables repeats. The system plays the music on repeat.
Exception: The system does not identify the type of instrument that the text-based tablature represents.

2.4 User Stories

Although the scope of TAB2XML is vast, here are some of the user stories where it would be well implemented.

2.4.1 User Story #1

As Max, a music student, I want to hear the music sheet I am practising so I can ensure that I am playing the music sheet correctly.

2.4.2 User Story #2

As Bob, a music composer, I enjoy hearing how my music sounds when played with different instruments.

2.4.3 User Story #3

As Ben, a music writer, I like to be able to write my music easily and hear how it sounds.

2.4.4 User Story #4

As Bob, a music composer, I like to be able to write my music easily and save my music sheet for later use.

2.4.5 User Story #5

As Sam, a music teacher, I like to be able to print music sheets out for my students.

2.4.6 User Story #6

As Albert, a 64 year old music student, I want to be able to adjust my music sheet according to my weakening vision so that I can comfortably see the music sheet when I play.

2.4.7 User Story #7

As Bob, a music composer, I write large pieces of music so I like to be able to see what note or measure is currently playing.

2.4.8 User Story #8

As Bob, a music composer, I write large pieces of music so I like to be able to jump from note to note depending on which part of the song I am working on.

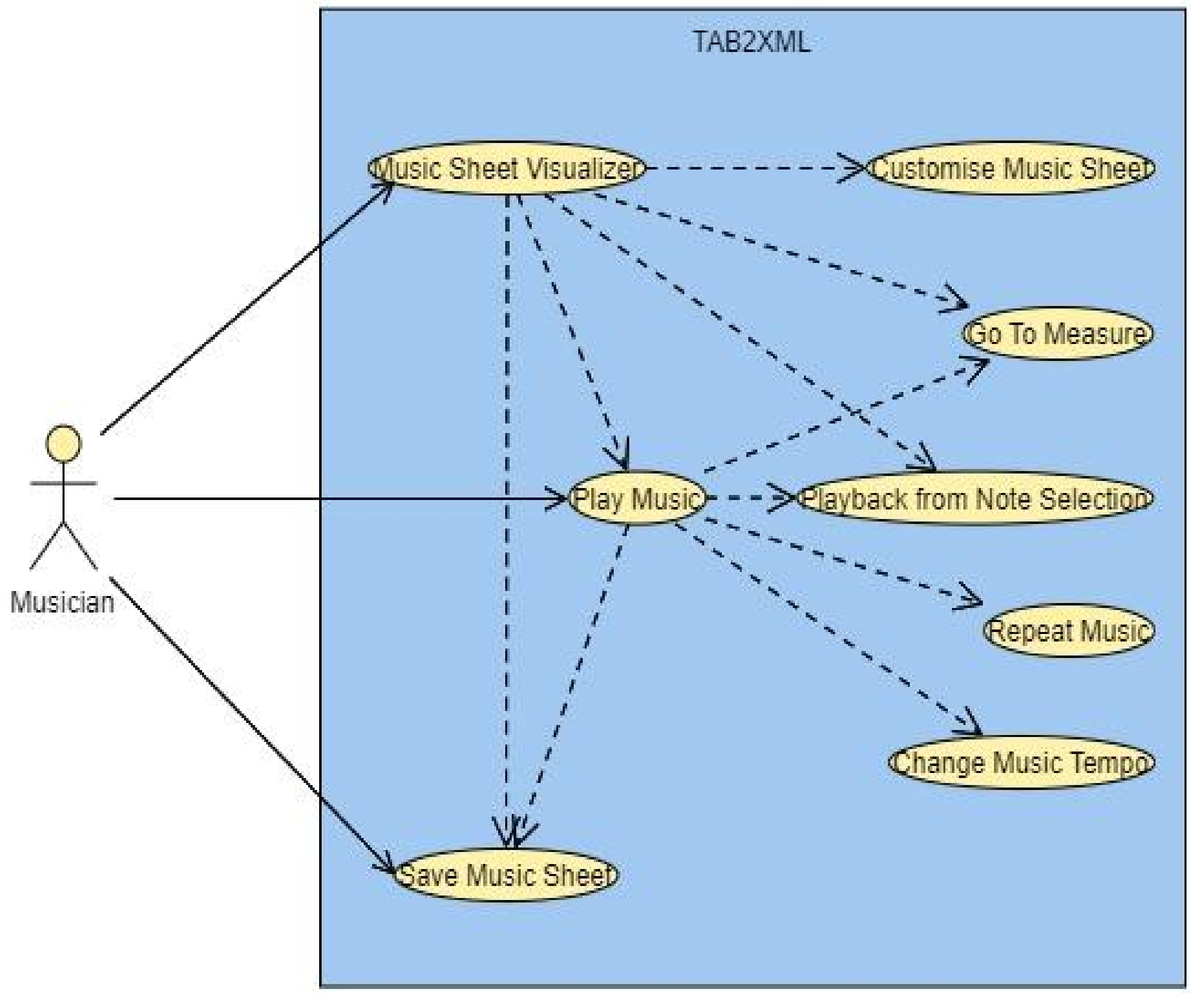
2.4.9 User Story #9

As Ben, a music writer, I like to be able to hear my music on repeat to hear what sounds the best.

2.4.10 User Story #10

As Sam, a music teacher, I like to be able to experiment with the music sheets with different tempos to see what my students will enjoy the best.

2.5 Use Case Diagram



3. System Features and Requirements

3.1 Functional Requirements

3.1.1 Tablature to MusicXML conversion

3.1.1.1 Description

This feature allows the users to convert their text-based tablature into a MusicXML file.

The MusicXML converter will also detect and support the two most popular instruments which use text-based tablatures; the guitar and drum.

3.1.1.2 Stimulus/Response Sequences

- The user inputs their text-based tablature into the designated text input area in the system.
- The system identifies the type of instrument that the inputted text-based tablature represents.
- The system outputs it into a MusicXML file

3.1.2 MusicXML Visualisation - Music Sheet

3.1.2.1 Description

This feature allows the user to visualise the MusicXML file. The visualisation feature previews the text-based tablature in the form of a music sheet and allows users to save the file.

This feature also allows the user to be able to view the customised visual output of the music sheet, such as the spacing between notes, size of the notes, spacing between the line of the staffs, alignment of left or right side of the notes, etc.

Additionally, when the music is being played, the previewer will highlight the music note that is being played and the measure that the player is on.

3.1.2.2 Stimulus/Response Sequences

- The user will select desired MusicXML-based tablature.
- The user will preview the music sheet.
- After customising the sheet, the user can view the custom music sheet.
- The user can export it as a PDF file in order to print or share.
- When playing music, the user can view the note/measure that is currently being played.

3.1.3 MusicXML Visualisation Interaction

3.1.3.1 Description

This feature allows the users to interact with the visualiser.

Through this feature the user will be able to indicate how they would like the visual output of the music sheet to be customised, by indicating their preferred spacing between notes, size of the notes, spacing between the line of the staffs, alignment of left or right side of the notes, etc.

This feature will allow users to select a measure or note for the player to play from, along with allowing the user to go to whichever measure they wish in the sheet music previewer.

3.1.3.2 Stimulus/Response Sequences

- The user will select desired MusicXML-based tablature.
- The user will preview the music sheet.
- The user can customise the music sheet according to his/her liking.
- The user can select a measure or note to play the music from.
- The user can select a measure that they wish to go to.

3.1.4 MusicXML Playing

3.1.4.1 Description

This feature allows the user to play the MusicXML file, along with the ability to repeat their MusicXML file.

When a note or measure is selected in the MusicXML Visualisation Interaction, this feature will play the music from that note or measure.

3.1.3.2 Stimulus/Response Sequences

- The user will select desired MusicXML-based tablature.
- The user will preview the music sheet.
- The user can play the music sheet.
- The user can play the music sheet on repeat.
- The user can play the music sheet from a selected note or measure.

3.2 Nonfunctional Requirements

3.2.1 User Interface

The user interface will be designed to ensure simplicity and ease of use. The user interface will prioritise efficiency in both speed and use, along with accuracy of results.

3.2.2 Software Abilities

The software should:

- Be visually appealing
- Be efficient, fast and responsive in processing requests (under 5 seconds)
- Provide accurate results, i.e, the music sheet prints accurately and the music is played correctly.
- Be easy to use and understand through its user-friendliness
- Provide documentation that assists and supports the usability and ease of use
- Be robust, and handle large text-based tablatures
- Be memory-efficient and take up a small amount of drive space