MARCH 6, 2022

OFFICIAL USER MANUAL

EECS 2311 SOFTWARE DEVELOPMENT PROJECT

GROUP 3:
Ahmed
Devesh
Ion

Table of Contents

Table of Contents 2

1. **About TAB2XML** 3
   1.1 Product Name and Intended Use
   1.2 Features of Product

2. **System Requirements** 3

3. **Installing TAB2XML using Gradle** 4
   3.1 IntelliJ
   3.2 Eclipse

4. **How to use TAB2XML** 10
   4.1 Main Window
   4.2 File Tab and Input
   4.3 Save Tablature and Save MusicXML
   4.4 Music XML
   4.5 Preview Sheet
   4.6 Music Player

5. **Input Requirements** 15
   5.1 Measure instructions (Repeats and time signature)
   5.2 Guitar

1. About TAB2XML

1.1 Product Name and Intended Use

TAB2XML is a tool designed to convert musical tablature in a text format into MusicXML, a popular open-source file format used for exchanging digital sheet music. TAB2XML is a java-based application that is currently actively supported, with more features being added every week.

1.2 Features of Product

This product is able to convert the tablature of following instruments: Guitar, Drum, Bass.

Note: Since this product is still in development, instruments that the product can convert are limited, as of now. Reference How to Use.

2. System Requirements

Minimum System Requirements:

CPU:Core i3 2.4 Ghz
Ram: 4 GB RAM
GPU:Intel HD 4000
OS: Windows 8/10 64-bit
Storage: 500 mb free Storage
Java: JDK 17 or better

Recommended System Requirements:

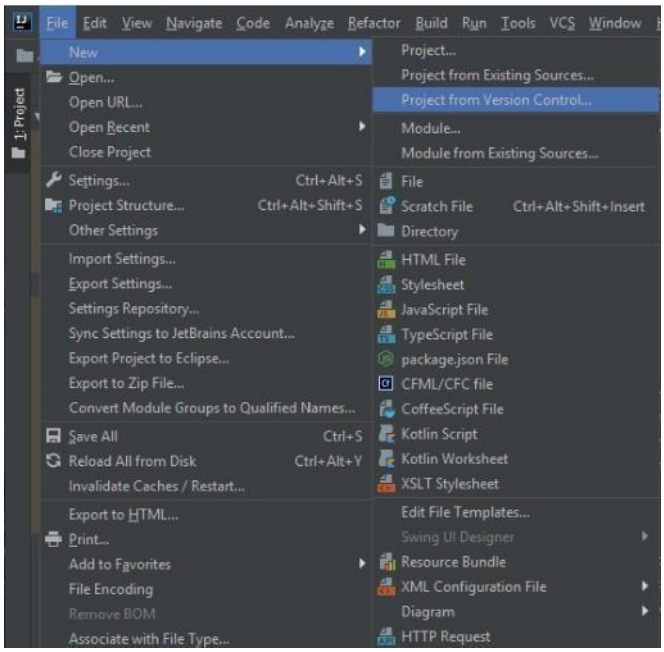
CPU:Core i5 2.8 Ghz
Ram: 8 GB RAM
GPU: Nvidia GTX 660 or AMD Radeon HD 7870 equivalent DX11 GPU, 2 GByte VRAM
OS: Windows 8/10 64-bit
Storage: 1 GB free Storage
Java: JDK 17 or better

3. Installing TAB2XML using Gradle

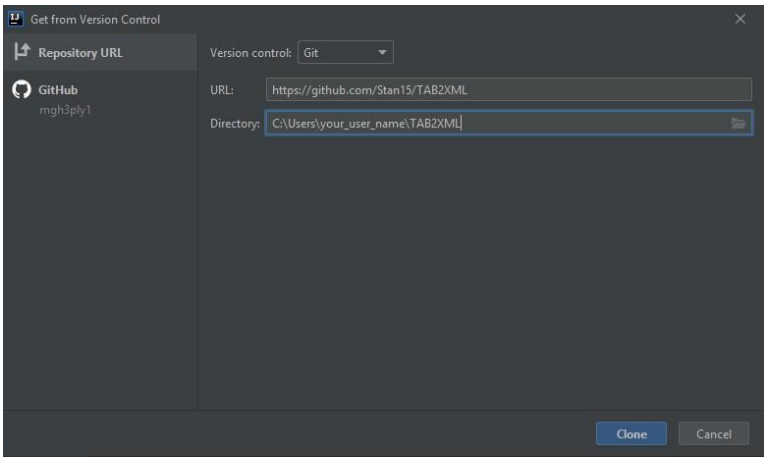
TAB2XML is built as a Gradle project and thus should work on any IDE of your choice. However, we will only go over how to run the program using the IntelliJ and Eclipse IDE.

3.1 IntelliJ

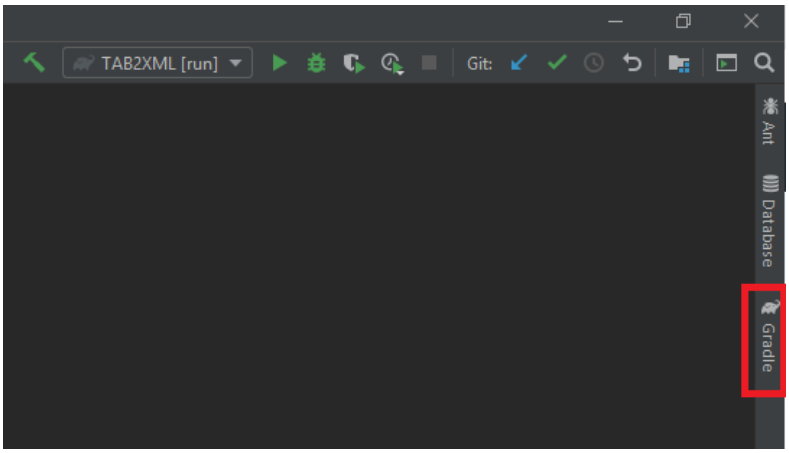
1. *We first need to clone the project from the online GitHub repository onto our local device. To do that, first open IntelliJ. At the top right, select File > New > Project from Version Control*



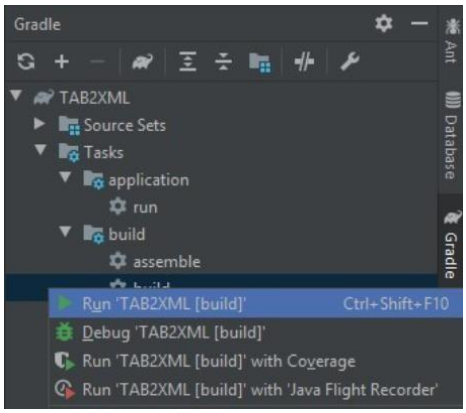
2. *This will direct you to a window which prompts you to enter a repository URL. Enter the link* https://github.com/eldibyorku/TAB2XML.git *as the URL, and select a directory where the project should be saved.*



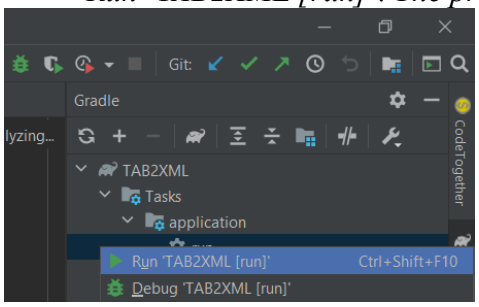
3. *Once the project opens, click the Gradle toolbar on the right side of your screen.*



4. *When the toolbar opens, click Tasks > build, and then right click "build" and the following dialog should pop up. Click "Run TAB2XML [build]"*

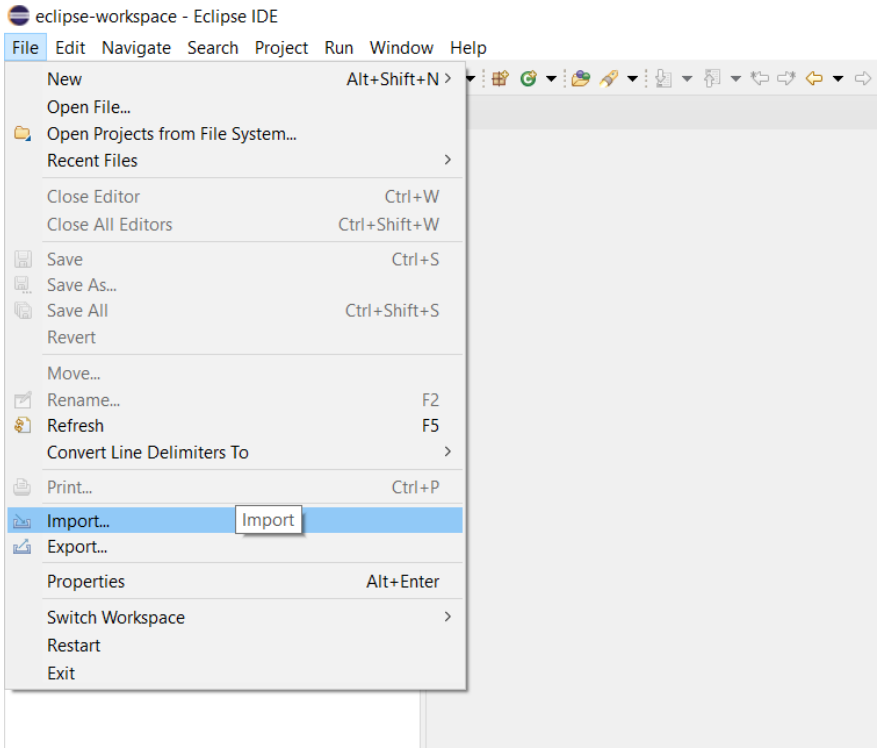


5. *Finally select Tasks > application, and then right click the run task and select Run 'TAB2XML [run]'. The program will now launch.*

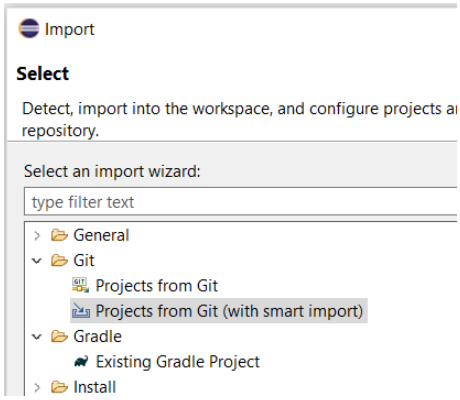


3.2 Eclipse

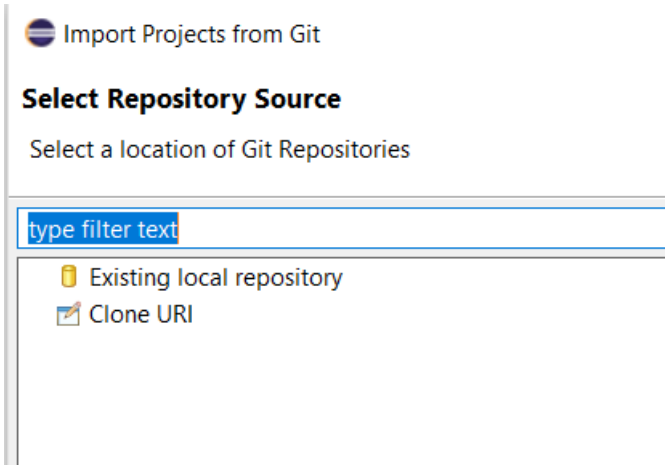
1. *Like with IntelliJ, we want to clone the project from the GitHub repository. To do this, we select File > Import.*



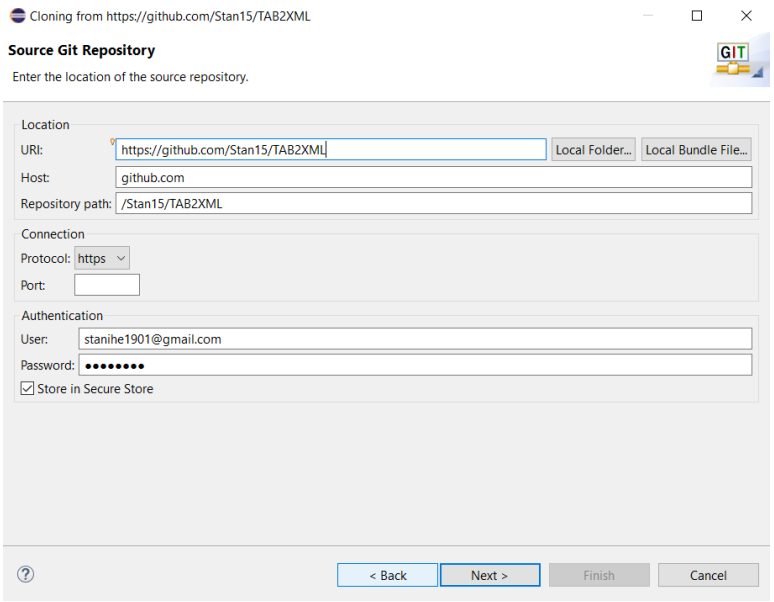
2. *This will take you to a new window like the one below. Under the Git folder, click Projects from Git (with smart import)*



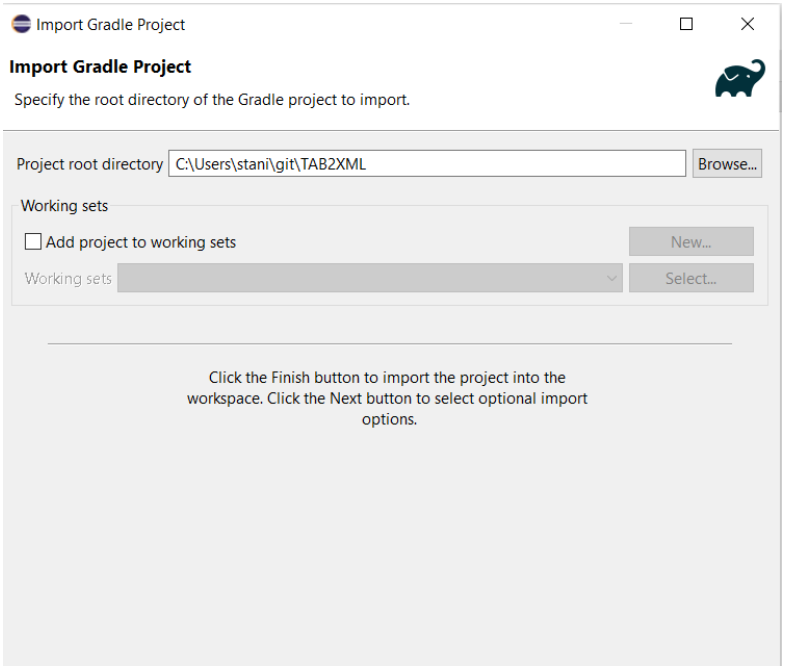
3. *Next, click Clone URI*



4. *This will take you to a window where you are prompted to input a URI. Paste in the link* https://github.com/eldibyorku/TAB2XML.git *and click proceed with the steps (clicking next) until you see the finish button. Click on that.*

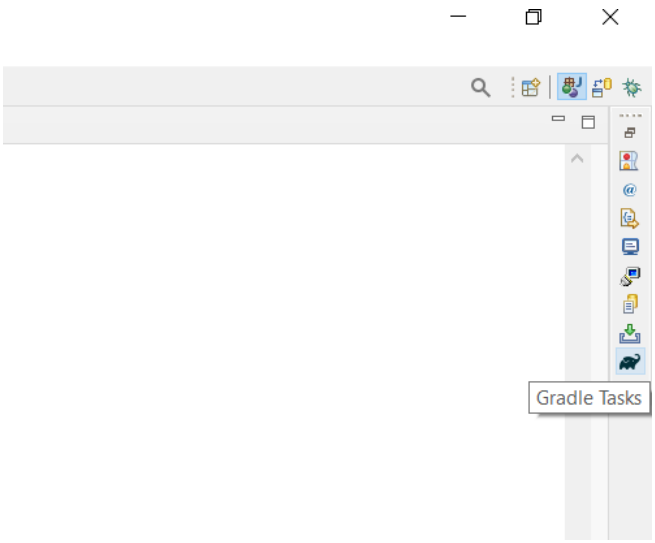


5. *Now we have imported the project onto our local device, we want to import it again as a Gradle project. Repeat step 1. When you get to the view shown in step 2, instead of*
   clicking projects from Git, click Existing Gradle project. Proceed till you come to this page.

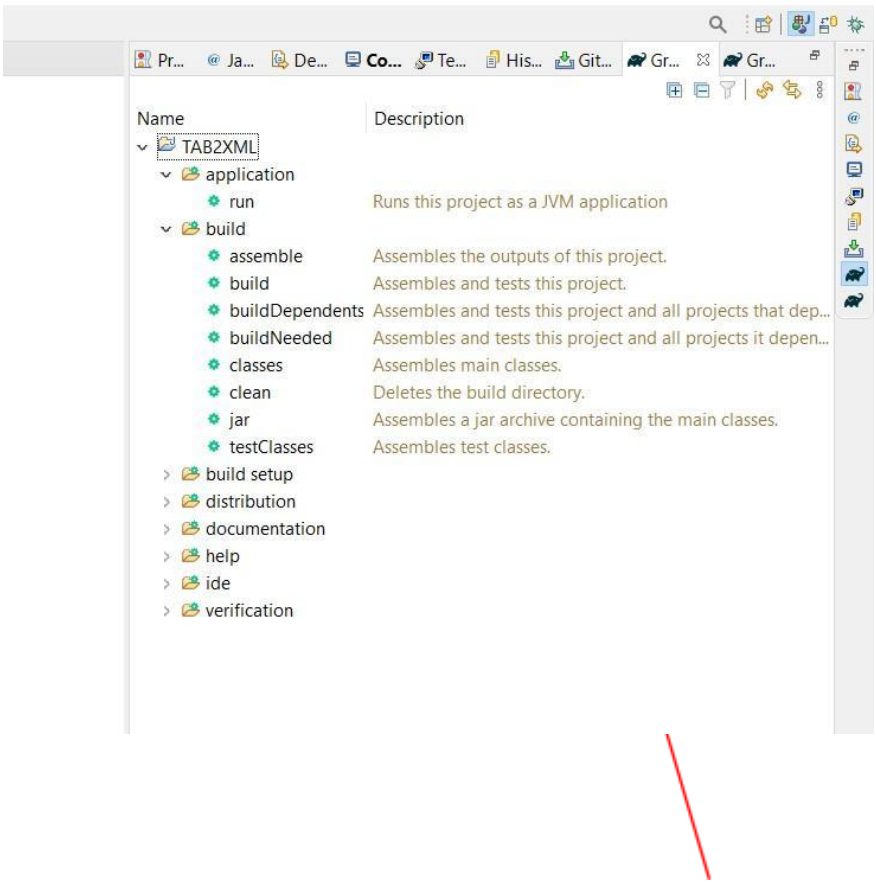


6. *Specify the path of the Git project you cloned in the previous steps and click finish.*

7. *Click on the "Gradle tasks" icon on the right of your screen*



8. *Double click on the Gradle "build" task at TAB2XML > build > build*



9. *Finally, click on the run task on TAB2XML > application > run*

4.How to use TAB2XML

4.1 Main window

When you run the program, you will be able to see a text field at the center of screen (**Figure1**). This is where you paste in your tablature txt file.

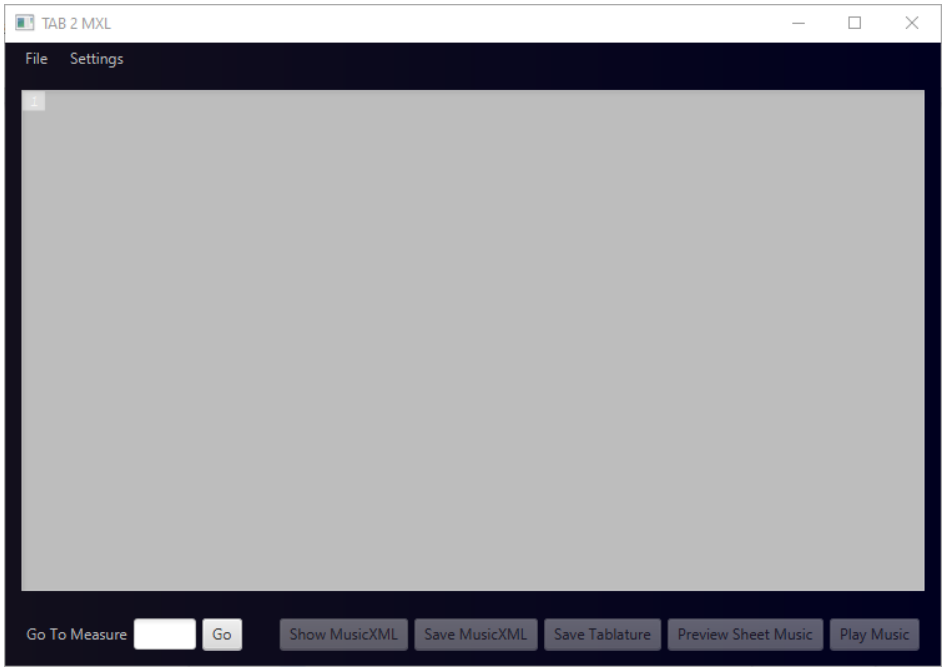


Figure 1

4.2 File tab and input

To put your input, click file-open and choose your file, or copy and paste your text file in the text field.`

1. Once you put your text input, the system identifies errors in your input if any exists, and it notates them using a color-coded highlighting system.

   There are 4 levels of highlighting:

   i. Red highlight: This is used to identify errors which may critically affect the output of the conversion.
   ii. Yellow highlight: Errors with this highlight are less critical, but we do not guarantee an accurate output with these errors.
   iii. Grey highlight: This highlight is used to identify content which may have little to no effect on the output.

Detailed below are a few examples of different error highlighting scenarios:

Grey highlight: “This text can’t be understood.” (**Figure 2**).
This is used to identify text which were not identified to be a score object (i.e measure, no te, repeat instruction, e.t.c.)

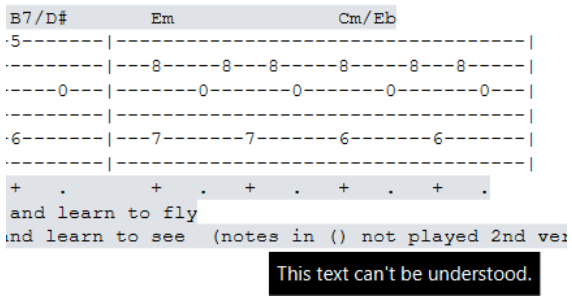


Figure 2

- Yellow highlight: “A guitar measure should have 6 lines.” (**Figure 3**).
  As a warning, if you get such an error on a measure which seems to be accurate, make sure no text is written on the side of the measure as this makes the system identify it as two different measures. Reference the *Input Requirements* section of this manual for more information regarding this.

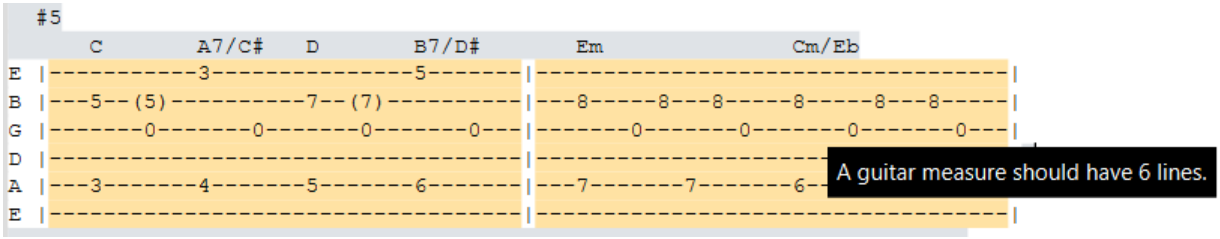


Figure 3

- Red highlight: “This annotation is either unsupported or invalid” This is used to identify elements which are either not supported or not identified as valid measure annotations.

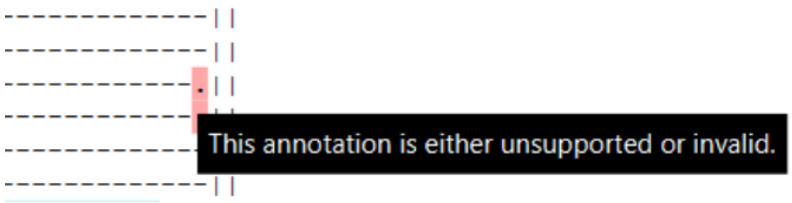


Figure 4

Note: There are more error scenarios which may occur, but they are all categorized into the three groups identified above.

2. For a more accurate output, be sure to resolve any errors highlighted red or yellow. The

grey errors can usually be ignored without consequence.

How to resolve errors:

Be sure the number of lines of the instrument is right.

Remove all unrecognizable notations in tablature and replace with dash '-'.

Remove all text placed around measures, except for measure instructions (time signature and repeats)

Note: If you removed all yellow and red highlights, the score is ready to be converted. However, if there are no yellow or red highlights in the score, you can skip this step.

4.3 Save MusicXML and Save Tablature

Click the "Save MusicXML" or "Save Tablature" button. This opens a new window for some options (**Figure5**).

Figure 5

Title – You can set the title of the song.
Artist – You can set the artist name.
File Name – You can name your file.

- Note: All of the functions have not been implemented. It will be updated gradually.

If you set all of them, click the save button. Navigate to the location where you want to save your converted file and save it. You can also change the file name (**Figure 6**).

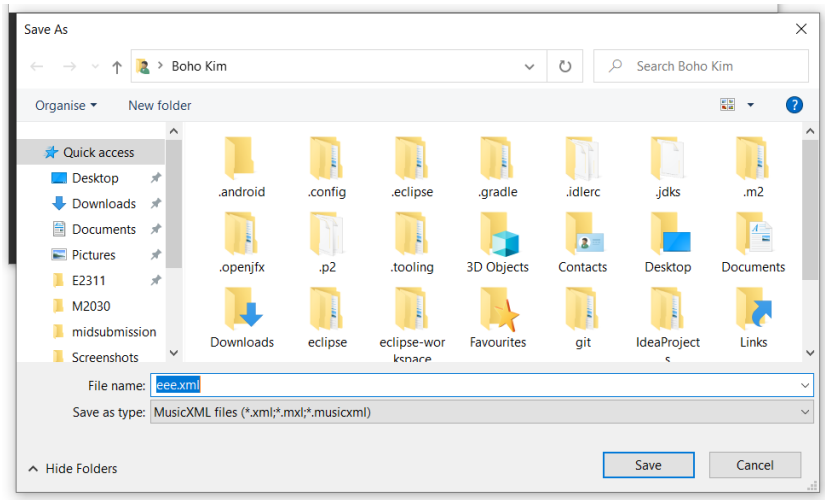


Figure 6

4.4 MusicXML

Click the "Show MusicXML" button. This opens a new window (**Figure 7**).

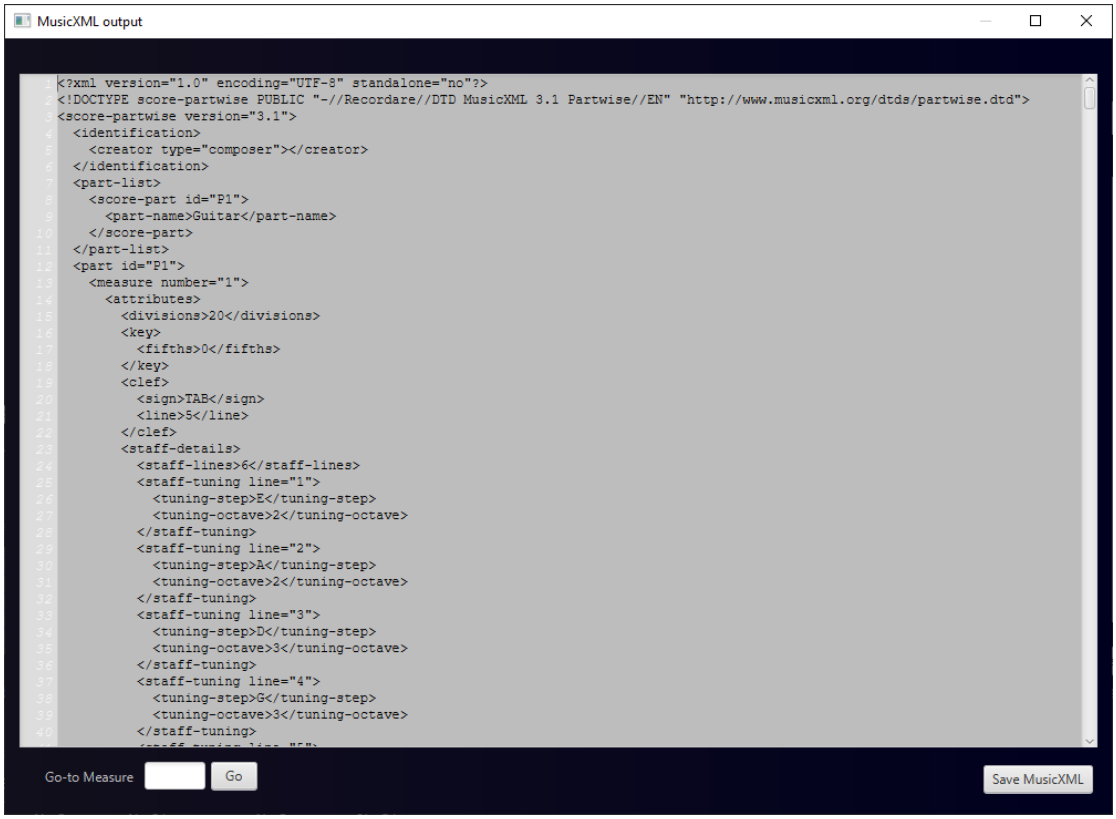


Figure 7

You may navigate the data shown on this window by using the scroll button or by manually inserting a measure and pressing the "Go" button.

If you'd like to, you may also save the data displayed on this window using the "Save MusicXML" button, which is also present on the main console window. Return to section 4.3 of this manual for more info on this function.

4.5 Preview Sheet

Return to the main window, and click the preview button, which will open a window similar to **Figure 8**.

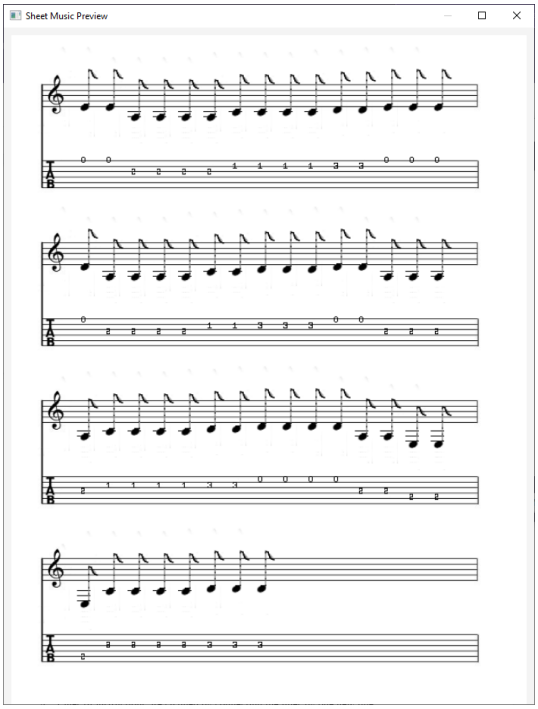


Figure 8

You will be greeted with a window showing the pdf of the tablature translated into a music sheet document.

4.6 Music Player

Return to the main window, and click the “Play music”, which will open a new window as in **Figure 9**.

IMAGE HERE

Figure 9

Information about music player here

---

5. Input Requirements

5.1 Measure instructions (Repeats and time signature)

This program allows for the application of repeats and time signatures to individual measures. Here, we will go over the input restrictions governing these features.

For instructions in general, the following requirements are outlined:

1. For a line to be interpreted as having instructions, it must only be composed of valid instructions separated by spaces and nothing else.
2. The ‘tab’ button should not be used in your instruction lines as this might result in the system not applying the instructions to the correct measure.
3. For your instructions to be recognized, the line directly below the instruction line must be a measure line or another instruction line (instruction chaining is allowed).
4. Lines of instructions are chained by connecting the lines by one new line.
5. The order of priority for applying instructions is left to right, up to down.

Repeats:
For repeats, the following input requirements must be followed for repeats to be correctly applied

1. Repeats must start and end with a vertical bar, and can have any combination of spaces or dashes "-" in-between, as seen below.

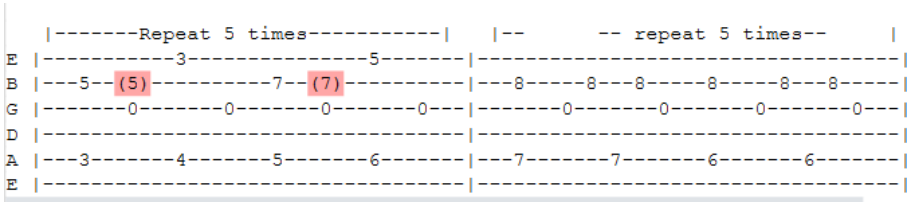


Figure 7

2. Repeats can be notated in any one of the three below stated ways.

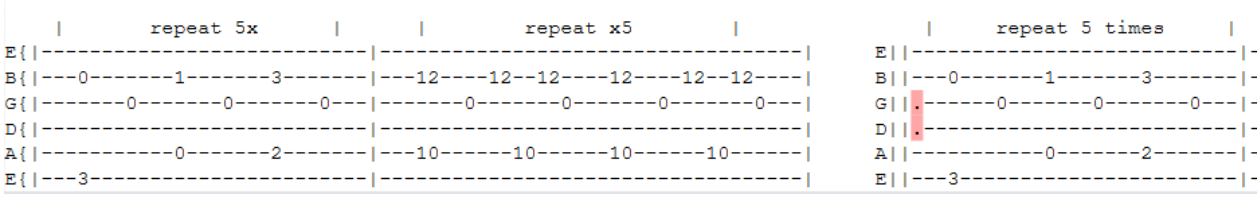


Figure 8

Time Signatures:

1. The list of possible time signatures has been artificially restricted to the following generally accepted time signatures: 2/4, 2/2, 3/8, 3/4, 4/8, 4/4, 4/2, 6/8, 6/4, 9/8, 9/4, 12/8, and 12/4.
2. If an invalid time signature is provided, the following error is received:

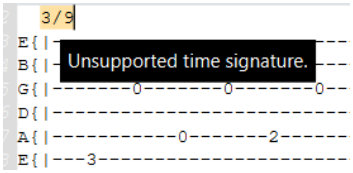


Figure 9

3. The default time signature is 4/4 if no time signature instruction is provided.
4. Time signatures with a beat or beat count consisting of three or more values are not recognized as instructions and will make the line of instructions invalid.

5.2 Guitar

Some sample tablature text files that meet the below requirements can be found in the project folder in the directory TAB2XML/src/test/resources/test_tab_files. The tablature file input into the program must meet the following requirements:

1. A measure must start with a vertical line after the string name.
2. you may not have text by the side of a measure which itself is not a measure.

Note: System does not guarantee an accurate output if measure collections do not contain blank line dividing, and if they do not have clear line names specified. (i.e string names/drum names).

Note: Information of other instruments will be updated gradually.