

# TAB2MUSIC DESIGN DOCUMENT

EECS 2311 SOFTWARE DEVELOPMENT PROJECT

---

**Group 11**  
March 2022

## Table of Contents

<b>1. Introduction</b>	<b>2</b>
<b>2. System Design Overview</b>	<b>2</b>
2.1 Major Packages of TAB2Music	2
2.2 Major Classes of TAB2Music	2
<b>3. Frontend Design</b>	<b>3</b>
3.1 Visualisation	3
3.1.1 Guitar	3
3.1.2 Drum	3
3.2 Playing notes	4
3.2.1 JfugueForGuitar	4
3.2.2 JfugueForDrum	5
<b>4. Backend Design</b>	<b>5</b>
4.1 XML Parser	5
4.2 Guitar Parser	6
4.3 Drum Parser	6
<b>5. Sequence Diagrams</b>	<b>7</b>
<b>6. Maintenance Scenarios</b>	<b>9</b>

## 1. Introduction

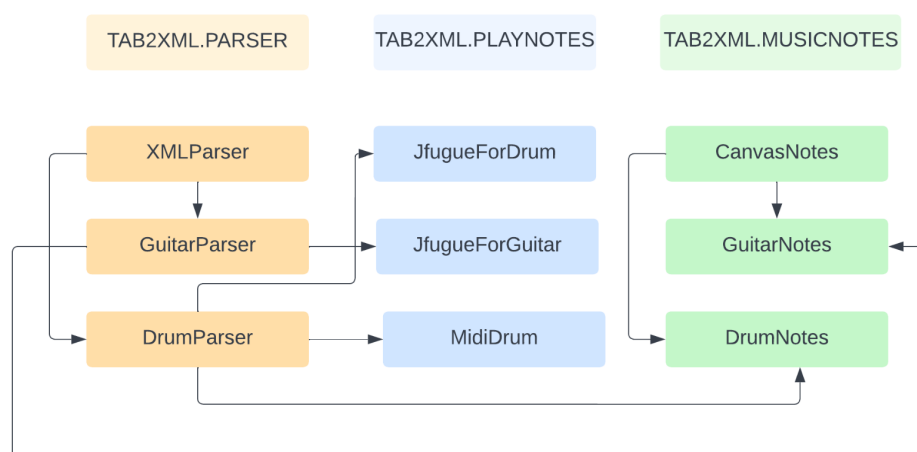
TAB2XML is a system that can convert text tablature to MusicXML. This document details the design of the system, in order to aid future developers of the system. It describes both the high-level design of the system as a whole and the design of many individual components.

## 2. System Design Overview

### 2.1 Major Packages of TAB2Music

TAB2XML is split into multiple packages. The major ones are:

- GUI - Frontend and GUI related code.
- Parser - contains all the code related to parsing the xml file.
- playNotes - contains all the code for playing the parsed notes.
- MusicNotes - Code that visualises the notes.



**Figure 1:** Package Diagram showing the relationship between major classes and packages in the project.

### 2.2 Major Classes of TAB2Music

Here is a brief description of each class and interface in the above diagram. Once again, full descriptions can be found in the classes' respective sections.

- GuitarParser - Parses the Xml for instrument type guitar and stores all the important data into ArrayLists.

- DrumParser - Parses the Xml for instrument type drums and stores all the important data into ArrayLists.
- PlayNotes - The class receives the notes from the respective parser class and stores them in a string in order to play them.
- CanvasNotes - The class receives data from the respective parser class and visualises them.

### **3. Frontend Design**

#### **3.1 Visualisation**

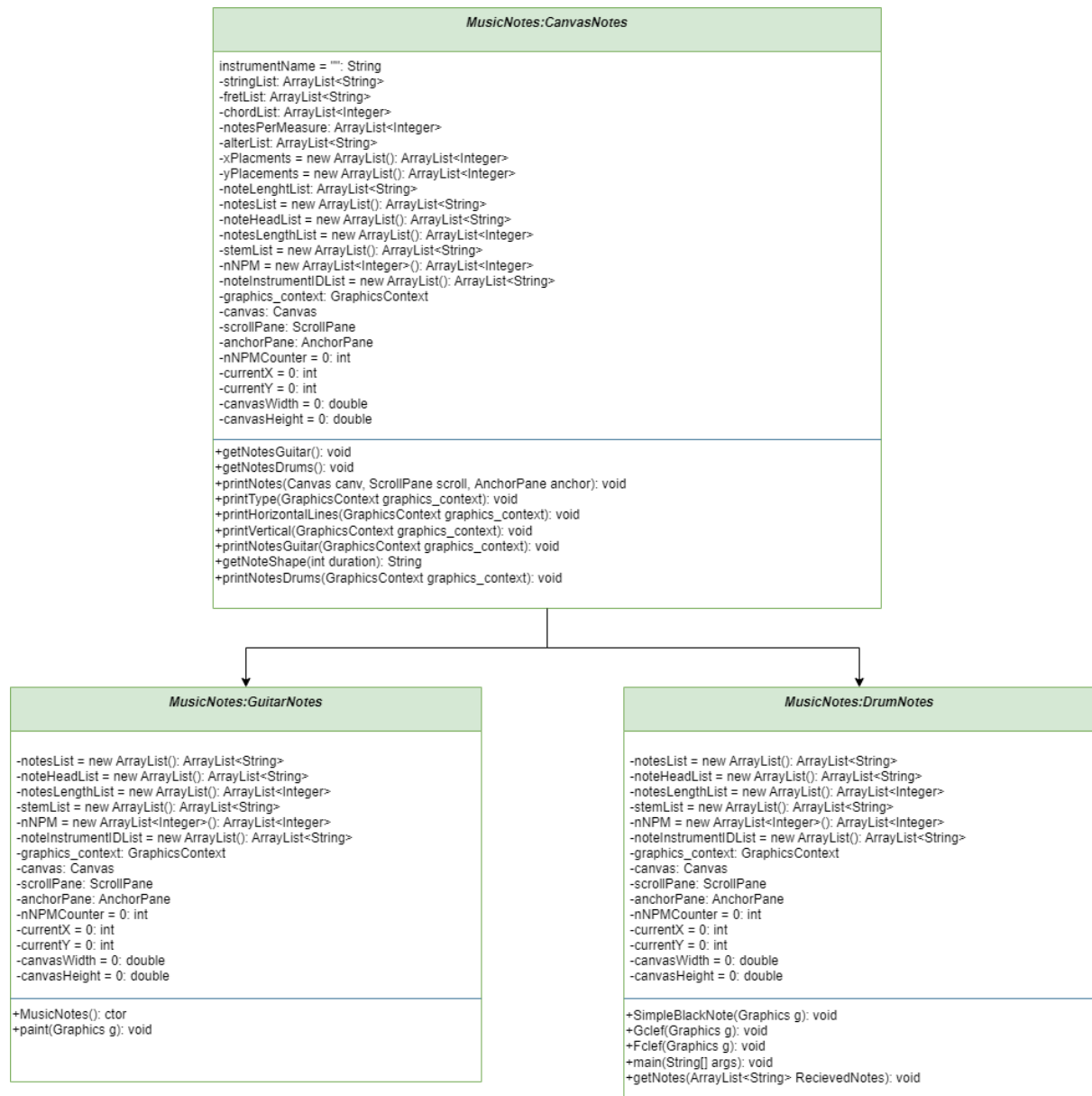
Input which is a tablature will be given by the user. Then the input will be parsed. As a result, an array list containing strings named noteLenghtList will be produced which stores the length of each note. There are also three other important array lists named nNPM, fretList, chordList and stringList which store number of notes per measure, frets that each string is played on for each note, the existence of chord for each note and the number of string for each note respectively. There is CanvasNote.java class which contains the methods that help to printNotes.

##### **3.1.1 Guitar**

There is a printHorizontalLines method which receives the GraphicsContext to draw lines on and based on the instrument name, it draws the horizontal lines. There is also another method named printType. It receives GraphicsContext objects as well. If the name of the instrument is equal to guitar, it will print TAB in vertical form for the first set of vertical lines. The method which prints the guitar notes is printNotesGuitar. The same GraphicsContext object will be passed to that function to draw the notes on the canvas. This method checks stringList and based on the values of that, prints down the value of fret. After printing each fret, the x and y coordinate will be updated. Also, after getting to the number of notes per measure, a vertical line will be drawn.

##### **3.1.2 Drum**

There is a printHorizontalLines method which receives the GraphicsContext to draw lines and based on the instrument name, it draws the horizontal lines. A method is named There is printNotesDrums that prints the notes for drums. This method draws a vertical line when the number of notes per measure has been achieved. Moreover, it goes through all the elements in noteList and stores the octave part. Based on the value of the octave and the outcome of getNoteShape, it draws the note at the specific place.



**Figure 2:** Class diagram visualisation of classes under the CanvasNotes Class and MusicNotes package.

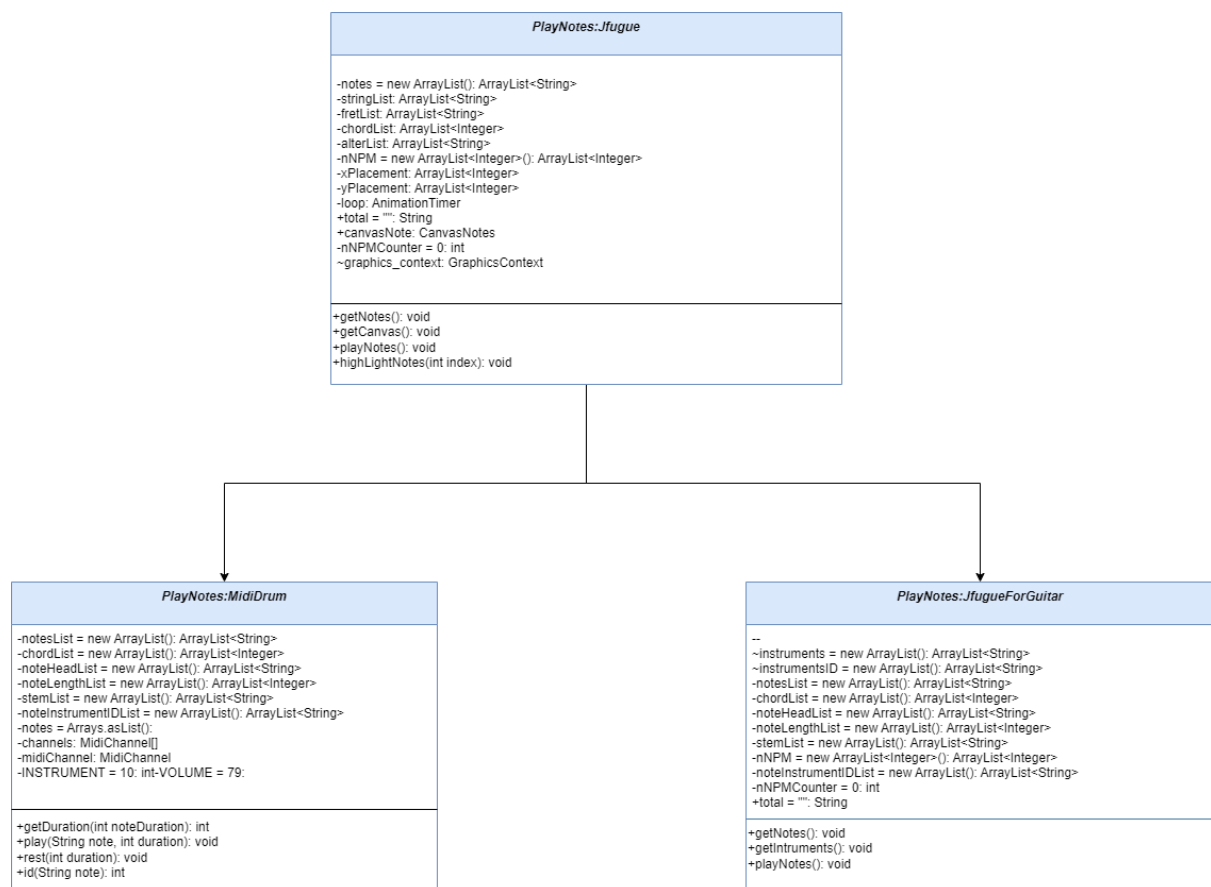
## 3.2 Playing notes

### 3.2.1 JfugueForGuitar

This class receives ArrayLists of notes and their duration from the guitar parser class. A player object is made and the notes along with their duration are stored in a string called total. The class makes use of jfugue java library to play the notes stored in the total string.

### 3.2.2 JfugueForDrum

This class makes use of arrayLists of drum notes, their lengths, chords, midi-instruments and their IDS. The getInstrument method pairs each midi-instrument with its respective ID so that the notes are played on the correct instrument. The class stores the notes received from the drum parser in a string called total and uses jfugue to play them. It also supports the functionality for playing/pausing music.



**Figure 3:** Class diagram visualisation of classes under the Jfugue Class and PlayNotes package.

## 4. Backend Design

### 4.1 XML Parser

XML is generated from the user input. This class receives the XML as a string and gets the attributes from the XML that are required to identify the instrument, number of measures and

number of notes in each measure so that a call to the appropriate method is made to parse further information. The attributes are stored in an ArrayList and sent to the class (as parameters of the method) that is responsible for further parsing.

## 4.2 Guitar Parser

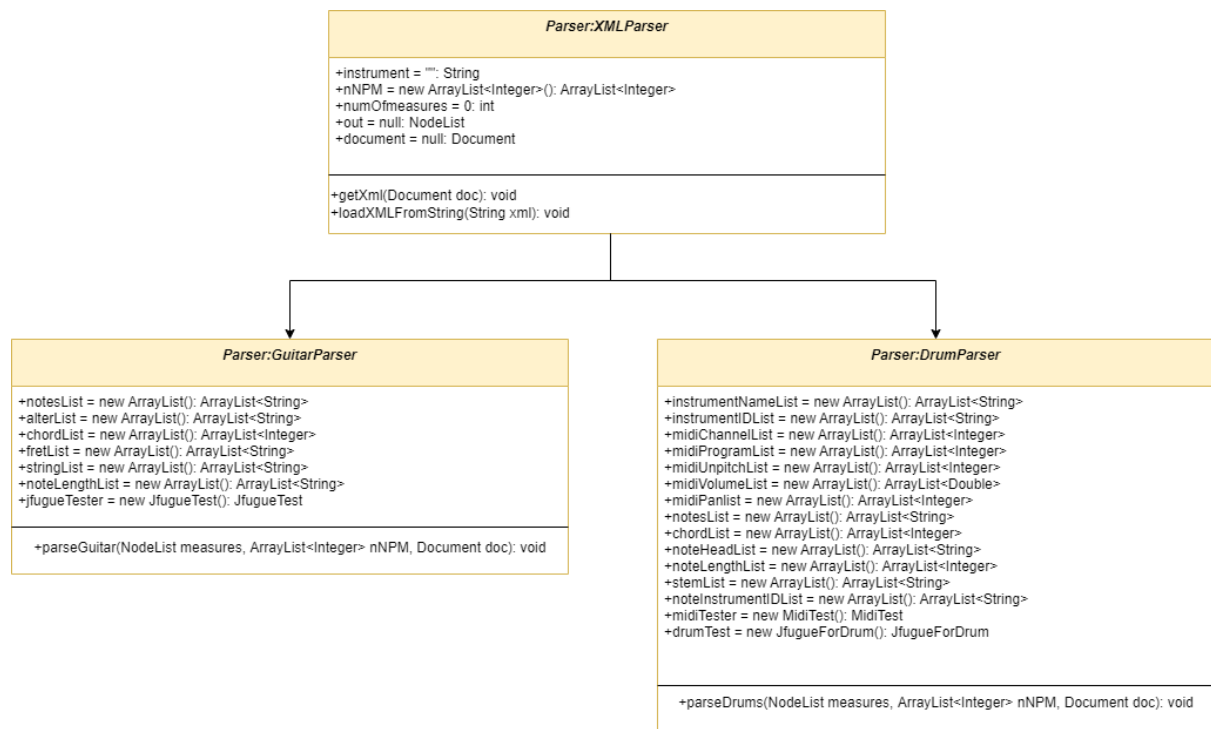
This class is responsible for parsing the guitar XML. Data from the XML is retrieved by tag names and stored in their respective ArrayLists. Everything that is parsed from the XML is also printed out to the console for user's reference. Calls to methods for visualising and playing are made with the relevant ArrayLists as parameters.

For visualisation the String and fret number for each note are sent over so that the notes can be visualised correctly whereas, for playing the notes along with their durations are sent over.

## 4.3 Drum Parser

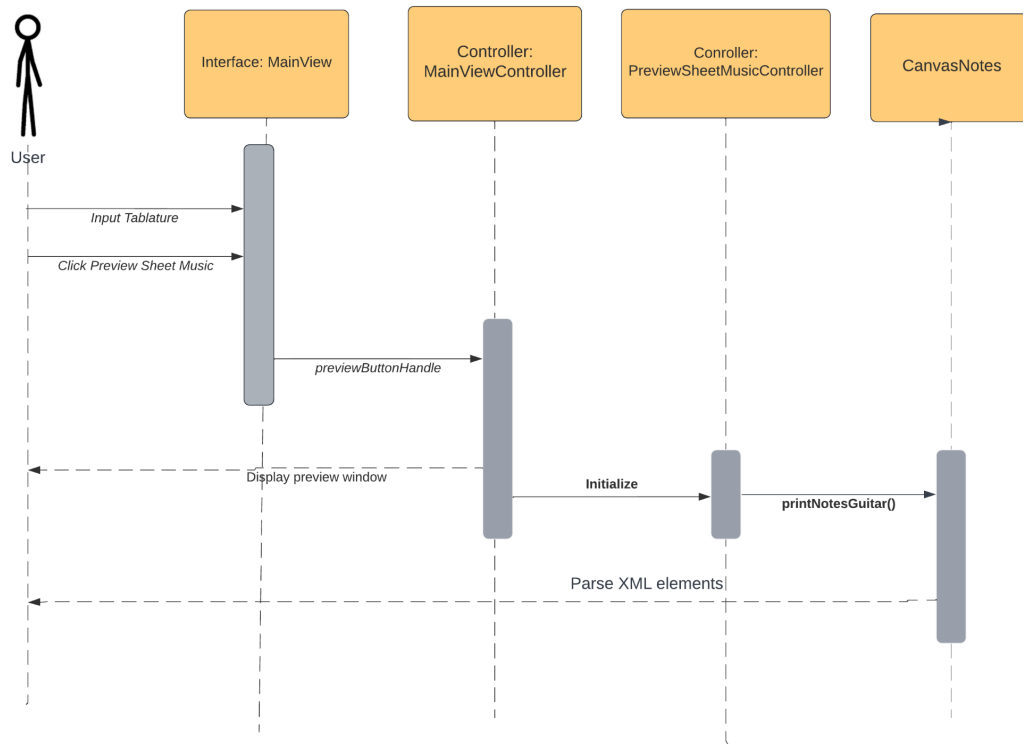
This class is responsible for parsing the drums XML. Data from the XML is retrieved by tag names and stored in their respective ArrayLists. Everything that is parsed from the XML is also printed out to the console for user's reference. Calls to methods for visualising and playing are made with the relevant ArrayLists as parameters.

For playing the drums, the notes, their duration as well as the midi-instrument, with its name and ID, that each note is played on is sent over to the JfugueForDrum class. For visualising the notes, the notes as well as the stemList is sent.



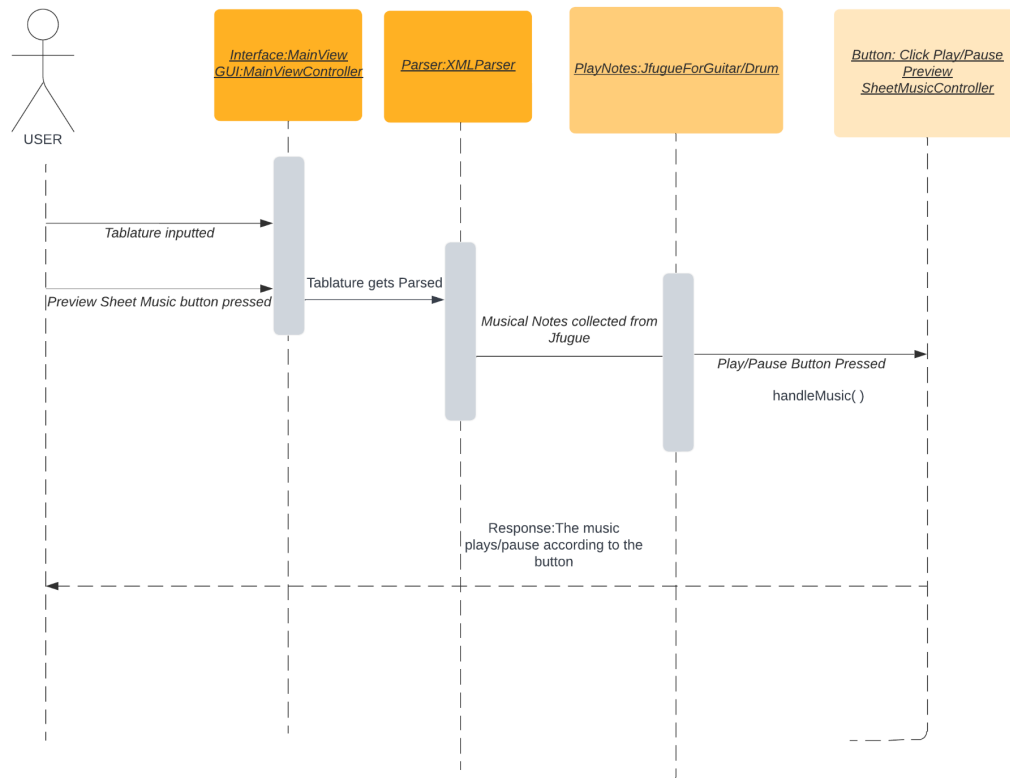
**Figure 4:** Class diagram visualisation of classes under the XMLParser Class and Parser package.

## 5. Sequence Diagrams

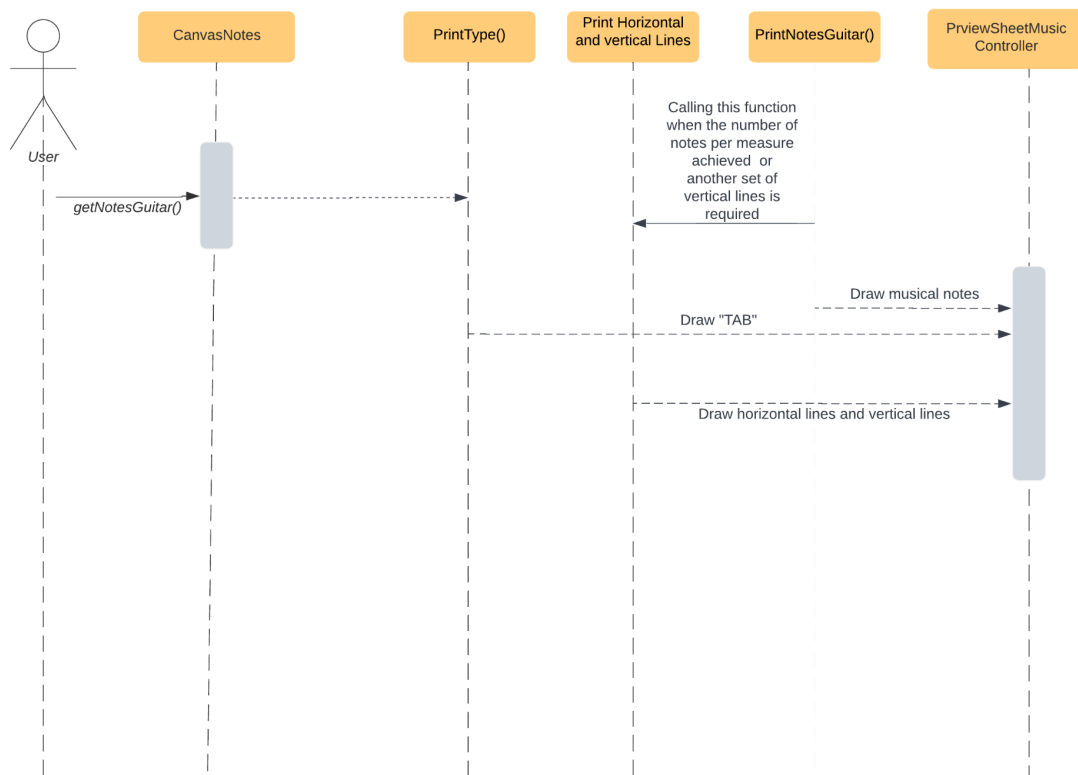


**Figure 5:** Sequence diagram showing the process of displaying guitar tablature.





**Figure 6:** Sequence diagram demonstrating the process of playing and pausing music



**Figure 7:** Sequence diagram demonstrating the process of printing musical notes for the tablature inputted

## **6. Maintenance Scenarios**

### **Adding New Features**

In order to add new features, we need to have maintenance checks.

For instance, for visualisation it needs to be parsed first, then it needs to be collected by the respective XML Parser for the instrument.. After being collected it needs to be pushed to Canvas Notes in order to show the visual representation.