

# **TAB2XML**

## **Design Document**

---

### **Group 6**

**Elmira Onagh  
Irsa Nasir  
Long Lin  
Harjap Randhawa  
Daniel Di Giovanni**

**Winter 2022**

# Table of Content

<b>1. Visualization of MusicXML</b>	<b>2</b>
<b>1.1. Instrument: <i>Guitar</i></b>	<b>3</b>
<b>1.1.1. Sequence Diagram</b>	<b>3</b>
<b>1.1.2. UML Class Diagram</b>	<b>4</b>
<b>1.2. Instrument: <i>Drum</i></b>	<b>5</b>
<b>1.2.1. Sequence Diagram</b>	<b>5</b>
<b>1.2.2. UML Class Diagram</b>	<b>6</b>
<b>1.3. Instrument: <i>Bass</i></b>	<b>7</b>
<b>1.3.1. UML Class Diagram</b>	<b>7</b>
<b>1.3.2. Sequence diagram</b>	<b>8</b>
<b>2. Playing the tablature</b>	<b>9</b>
<b>2.1. Activity Diagram</b>	<b>9</b>
<b>2.2. UML class diagram: <i>Guitar</i></b>	<b>10</b>
<b>2.3. UML class diagram: <i>Drum</i></b>	<b>10</b>
<b>2.4. UML class diagram: <i>Bass</i></b>	<b>11</b>
<b>4. Printing the Music sheet</b>	<b>12</b>
<b>4.1. Sequence Diagram</b>	<b>12</b>
<b>4.2. Activity Diagram</b>	<b>13</b>
<b>5. Go to measure</b>	<b>14</b>
<b>5.1. Activity Diagram</b>	<b>14</b>

# 1. Visualization of MusicXML

Based on the instrument specified in the input tablature, three different objects are created: *Guitar*, *Drum*, and *Bass*. In the following section, we look at different diagrams related to the creation and relationships of these classes.

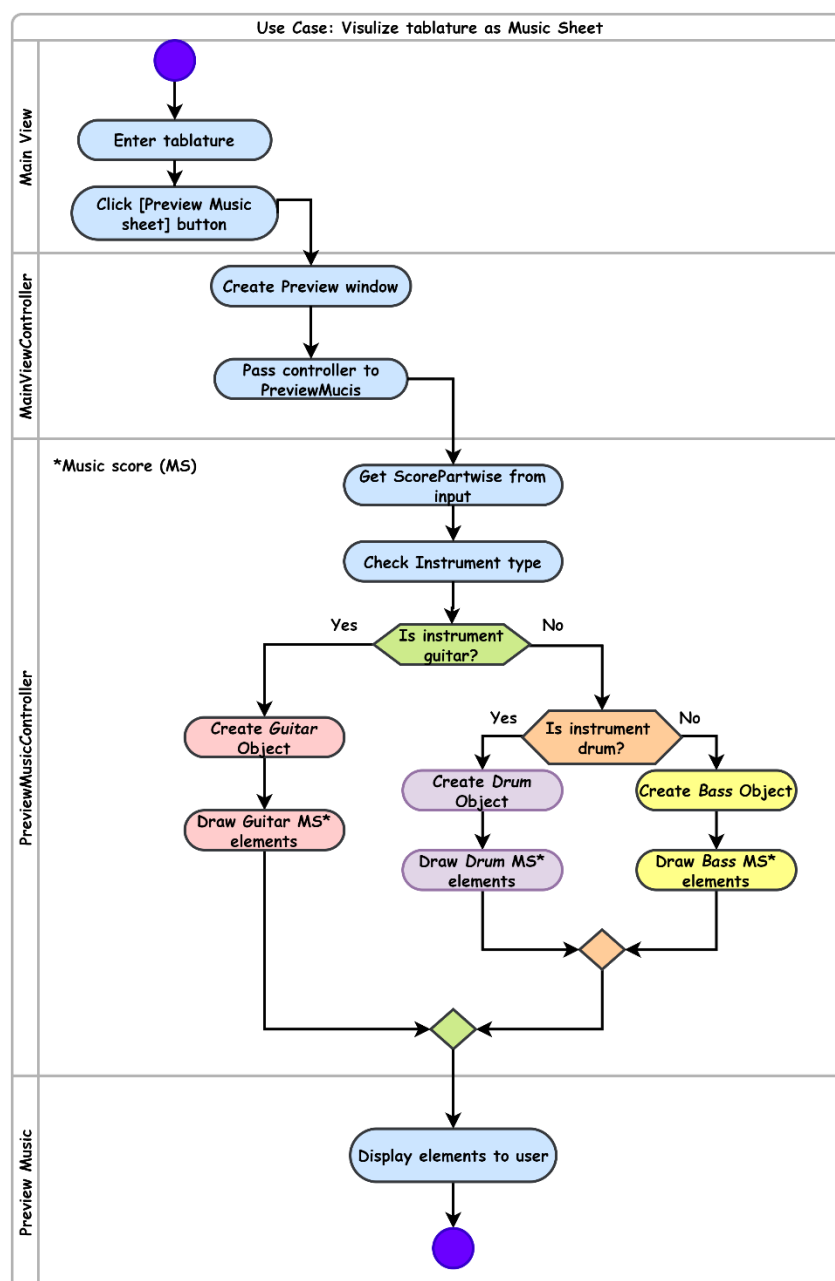


Figure 1. Overall activity diagram of the visualizing a tablature as a music sheet.

## 1.1. Instrument: *Guitar*

If the input tablature is a guitar tablature, then a *Guitar* class is instantiated. The creation and displaying of the elements of the tablature in form of a music sheet are done through the *drawGuitar* method.

In section 1.1.1 the sequence diagrams depicting the sequence of events taken to visualize guitar notes can be found. The Diagrams are broken down into parts to ease the understanding and visualization.

In section 1.1.2 UML Class diagram is included to show the interactions between the *Guitar* class and other classes that result in displaying the musical elements on the screen.

### 1.1.1. Sequence Diagram

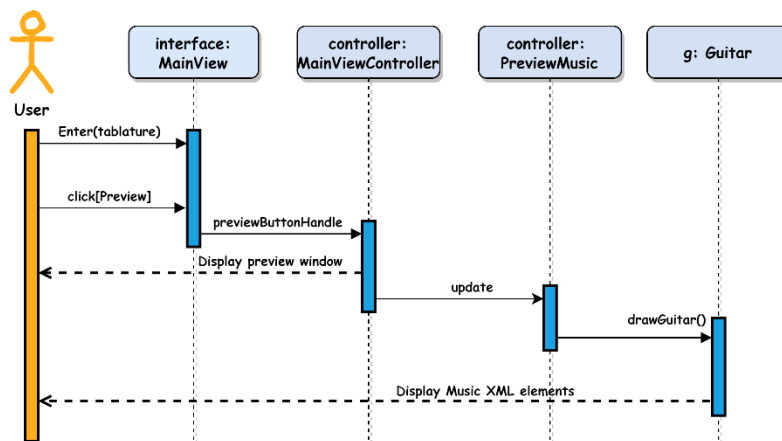


Figure 2. Overall Sequence diagram of displaying guitar tablature.

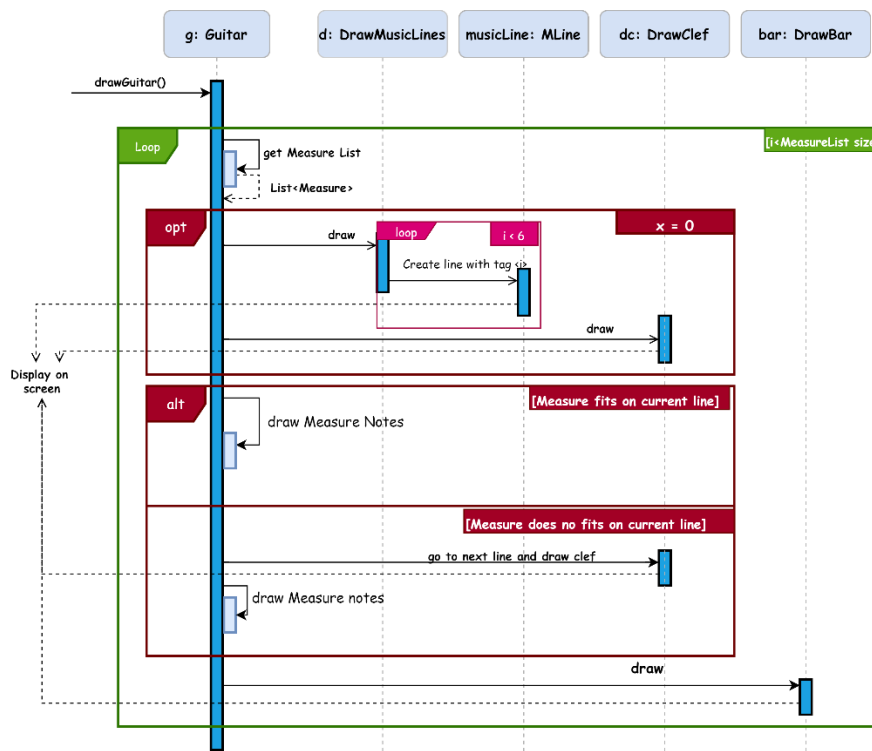
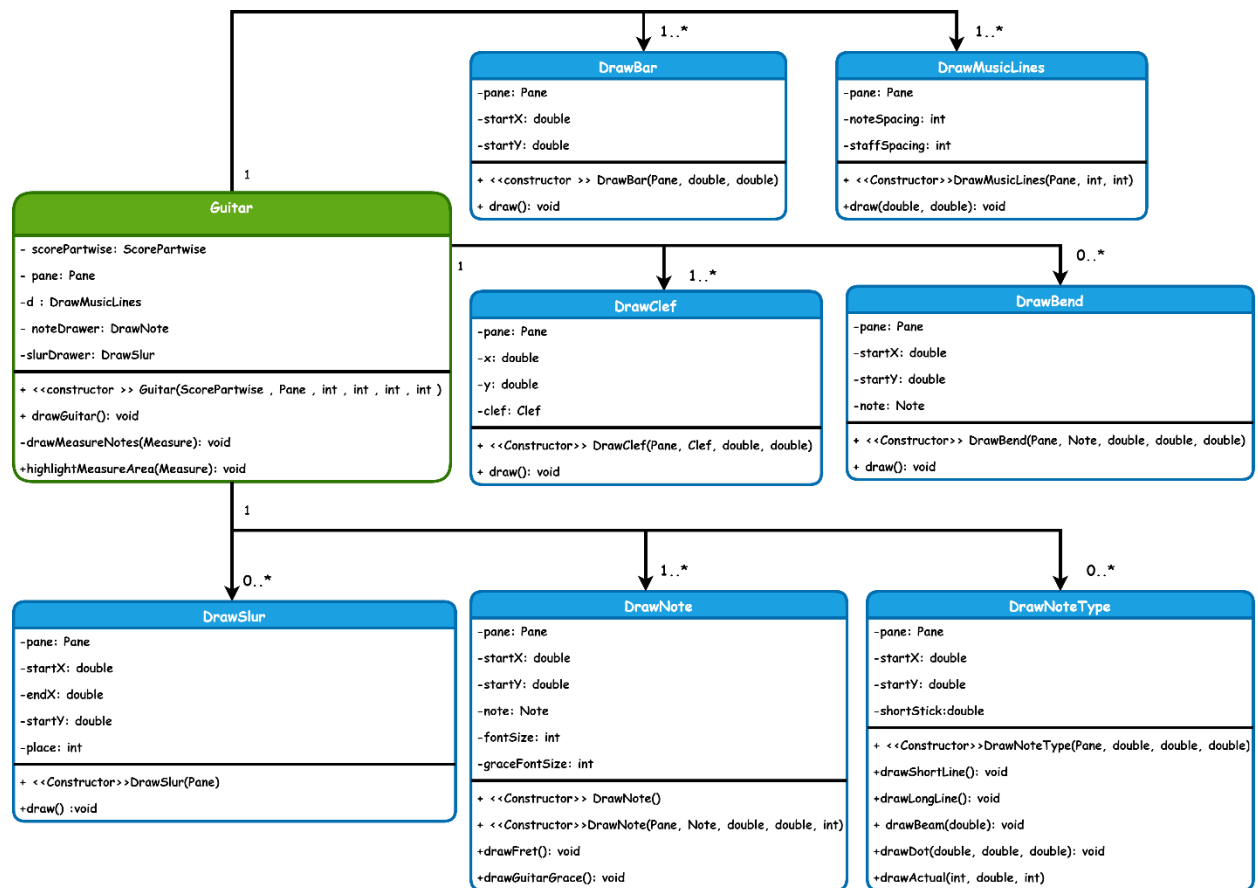


Figure 3. Sequence diagram describing the events taken in *drawGuitar* method of *Guitar* class.

### 1.1.2. UML Class Diagram

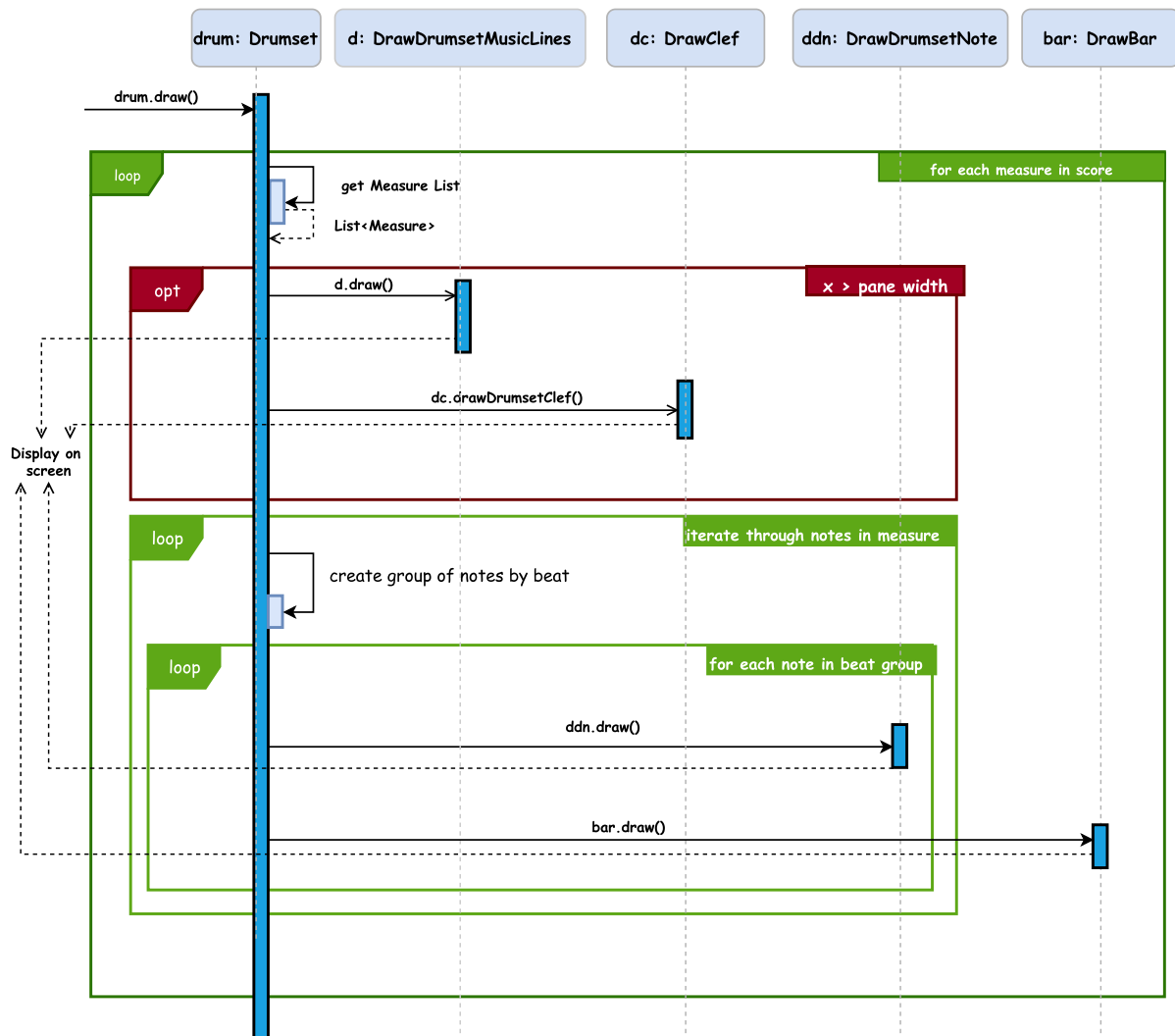


**Figure 4.** Class diagram of the *Guitar* class and its interactions. The green coloured *Guitar* class belongs to the *instrument* package while the blue classes belong to the *GUI.draw* package. The public, private, protected attributes and operations are denoted by “+,” “-,” and “#” respectively.

## 1.2. Instrument: *Drum*

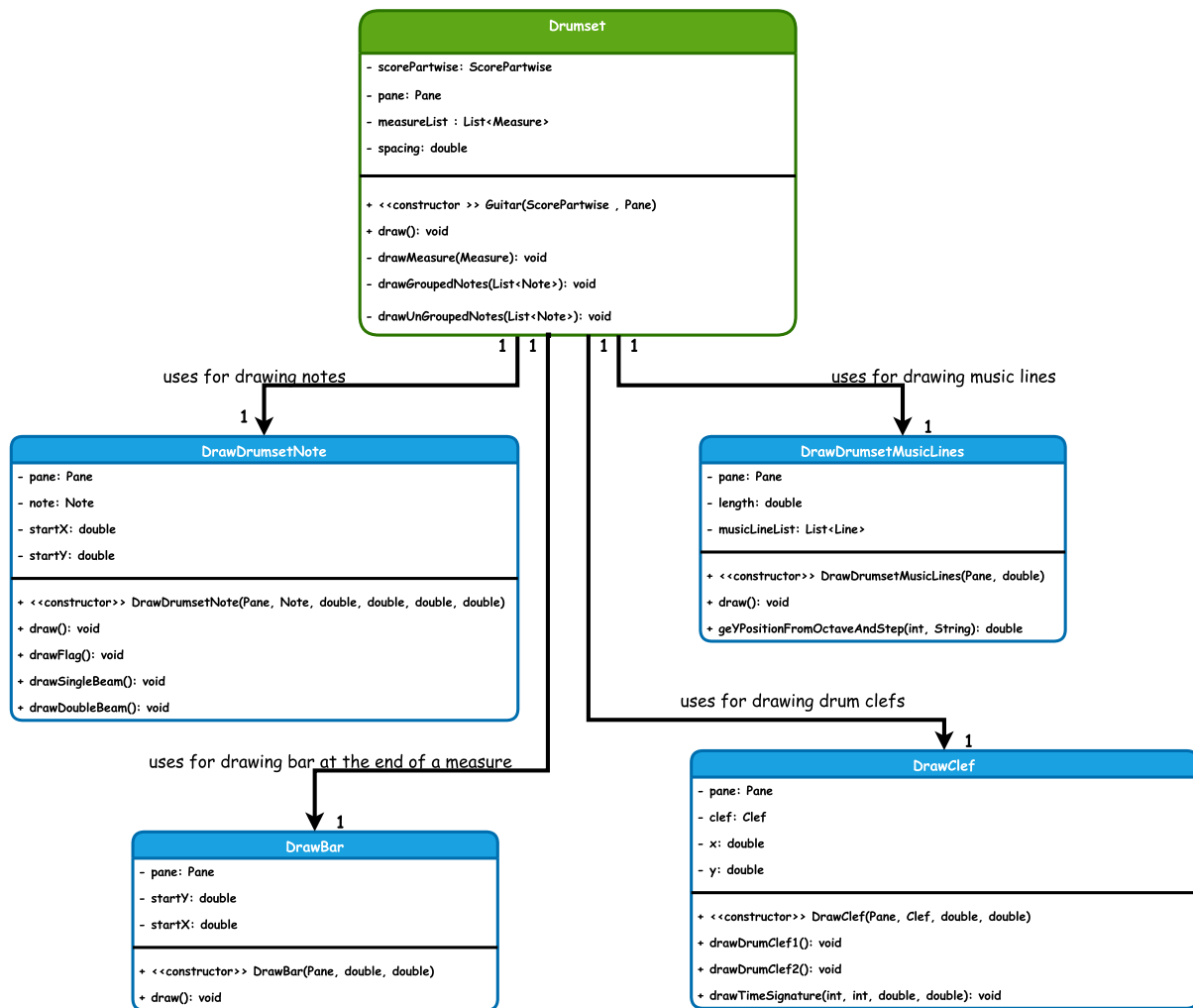
The following two diagrams are sequence and class diagrams for the *Drumset* class, showing its workflow and the other classes it uses/interacts with.

### 1.2.1. Sequence Diagram



**Figure 5.** Sequence diagram of the *Drumset* class and its interactions with the classes *DrawDrumsetMusicLines*, *DrawClef*, *DrawDrumsetNote*, and *DrawBar*.

### 1.2.2. UML Class Diagram



**Figure 6.** Class Diagram of the *Drumset* class and the classes it uses.

## 1.3. Instrument: *Bass*

### 1.3.1. UML Class Diagram

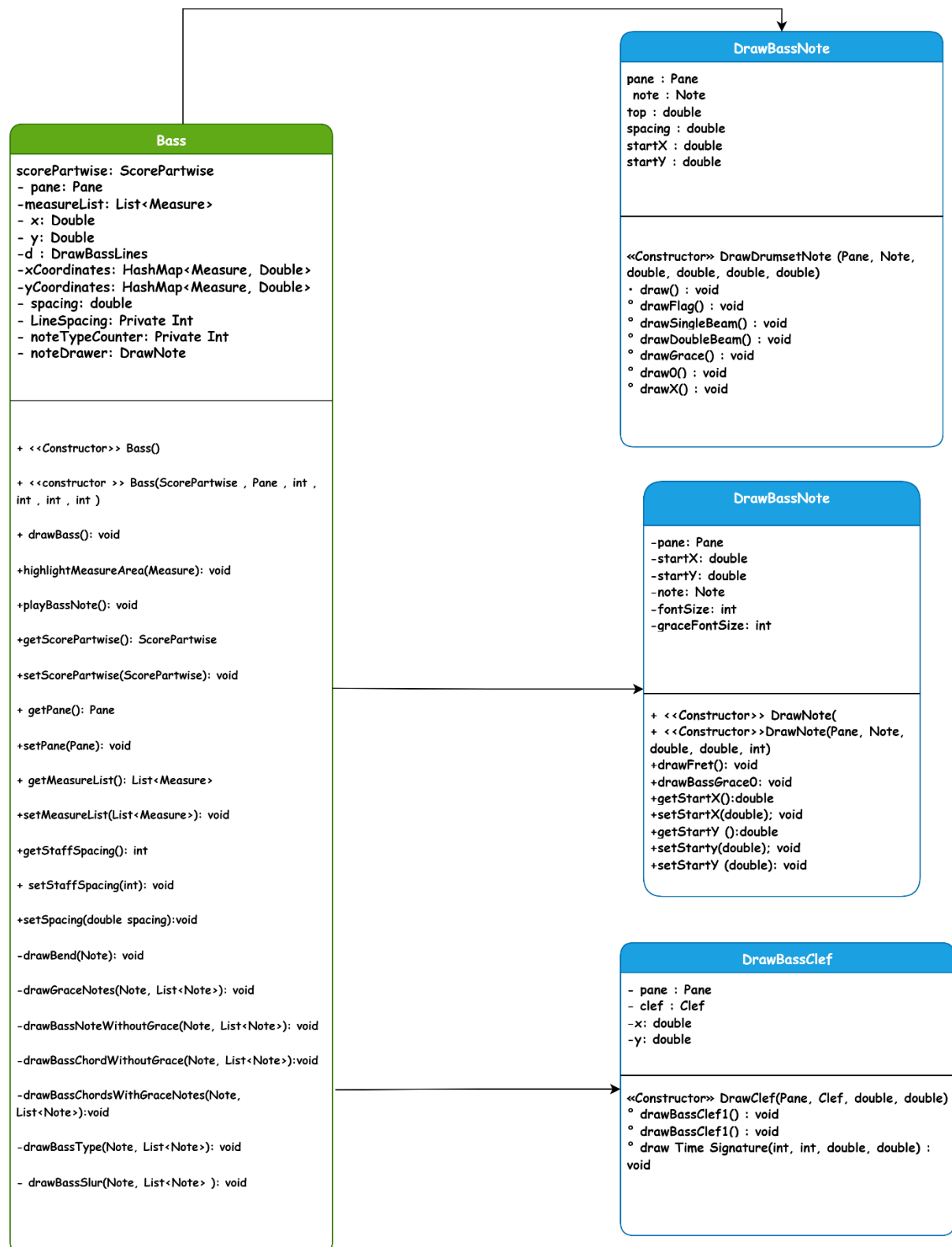
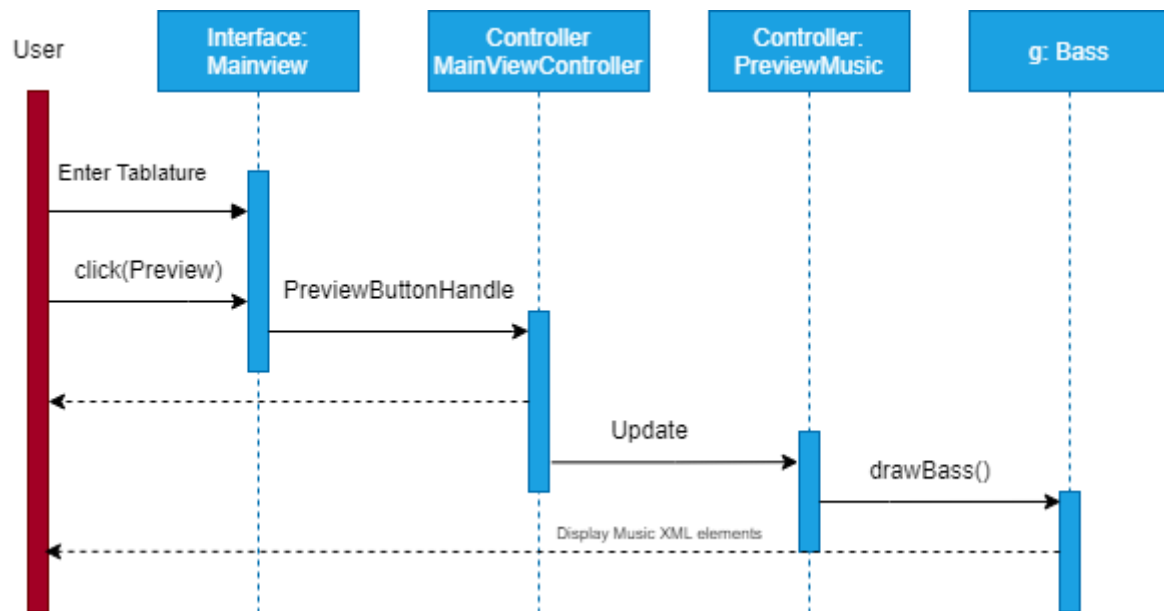


Figure 7. Class diagram indicating the attributes and methods of *Bass* class and classes that it interacts with



### 1.3.2. Sequence diagram



## 2. Playing the tablature

In this section, we will cover the design diagrams related to the Play functionality of the system. In section 2.1 an Activity diagram describing the overall events taken to play the notes is included. In the following sections, UML class diagrams depicting the specific methods of *MusicPlayer* class used for each instrument can be found.

### 2.1. Activity Diagram

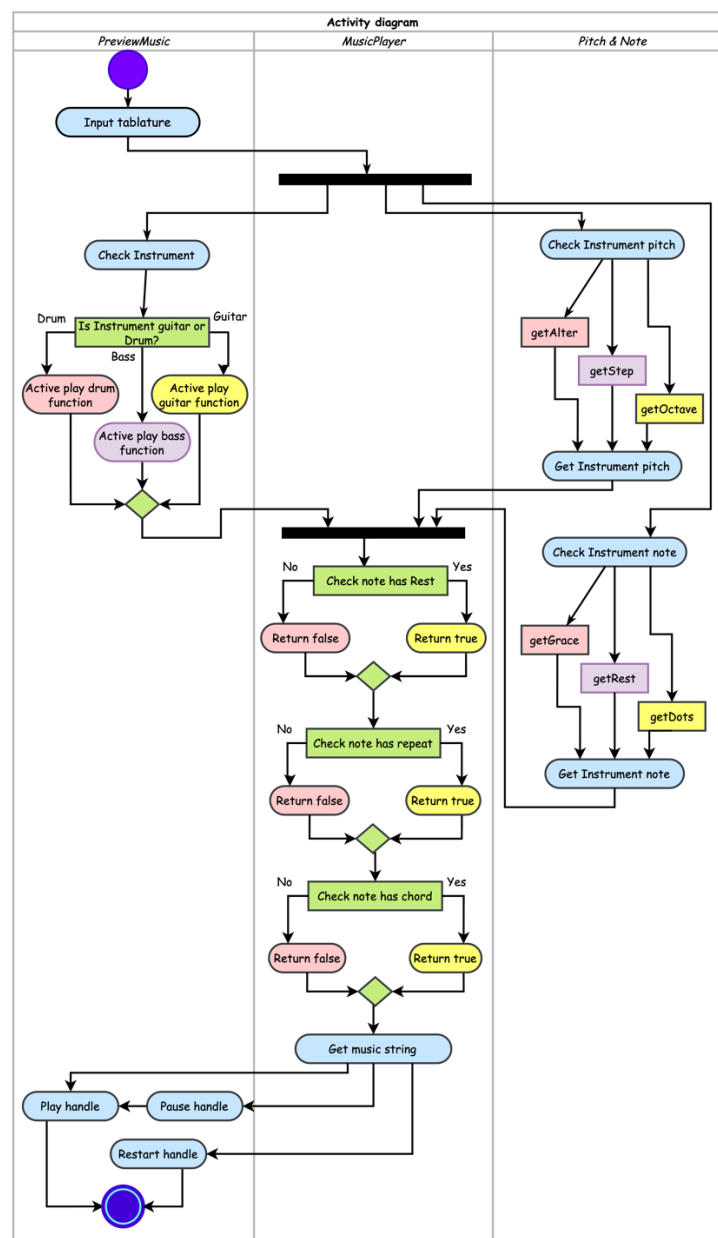


Figure 8. Activity diagram of play music note function for guitar and drum

## 2.2. UML class diagram: *Guitar*

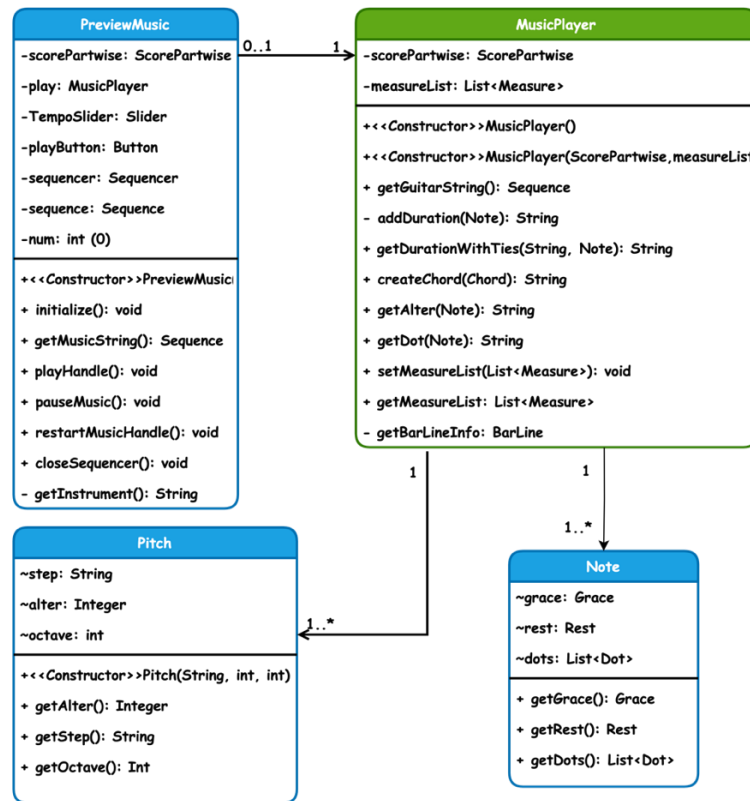


Figure 9. The public, private, protected attributes and operations are denoted by “+,” “-,” and “~” respectively.

## 2.3. UML class diagram: *Drum*

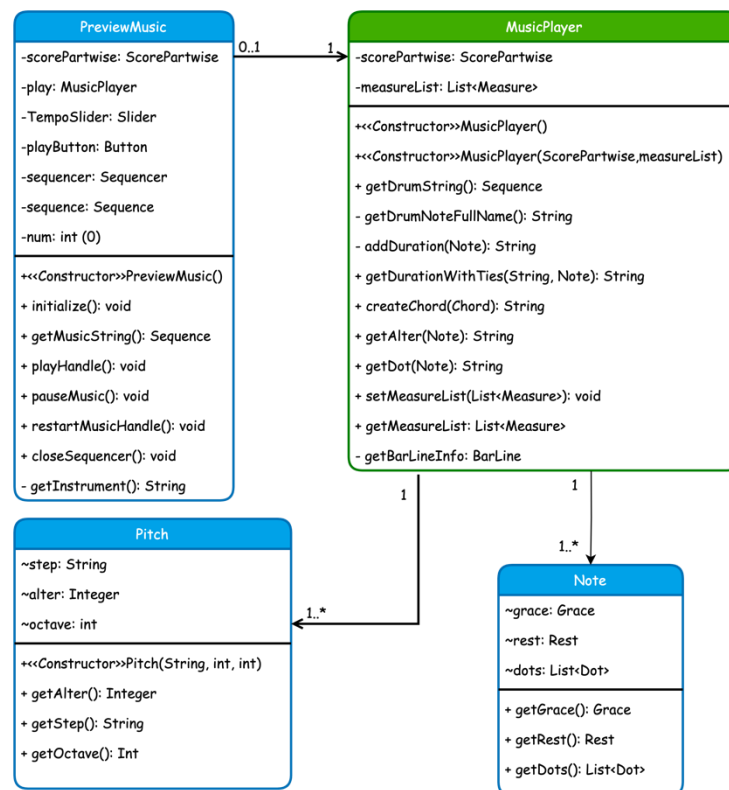
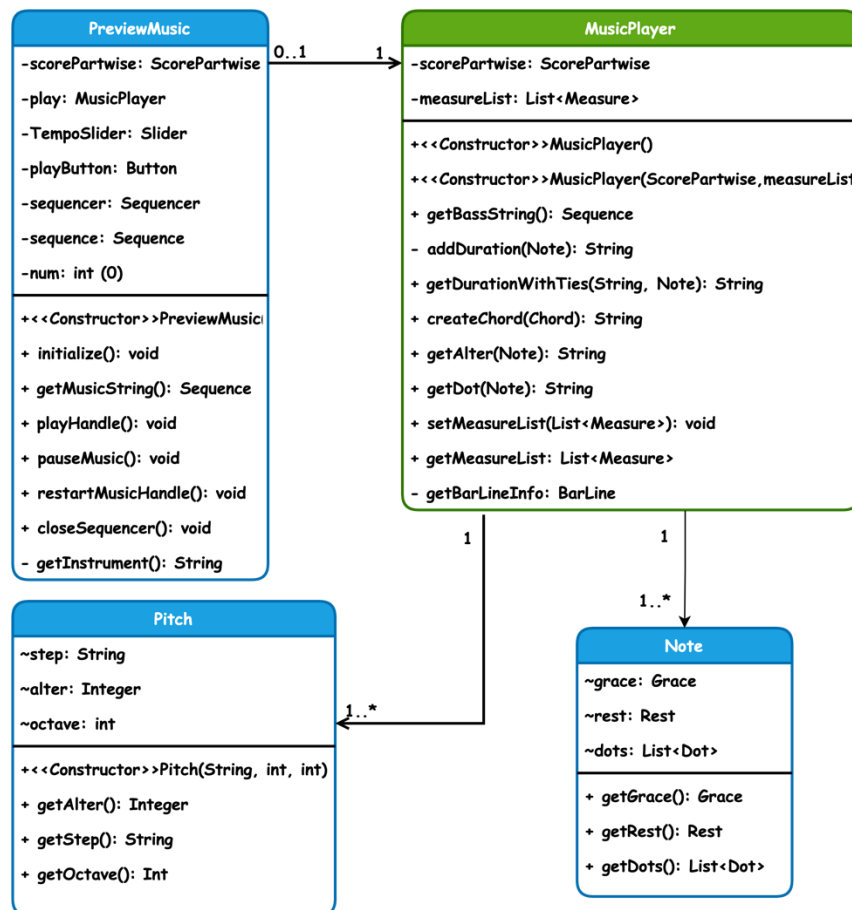


Figure 10. The public, private, protected attributes and operations are denoted by “+,” “-,” and “~” respectively.

## 2.4 UML class diagram: *Bass*

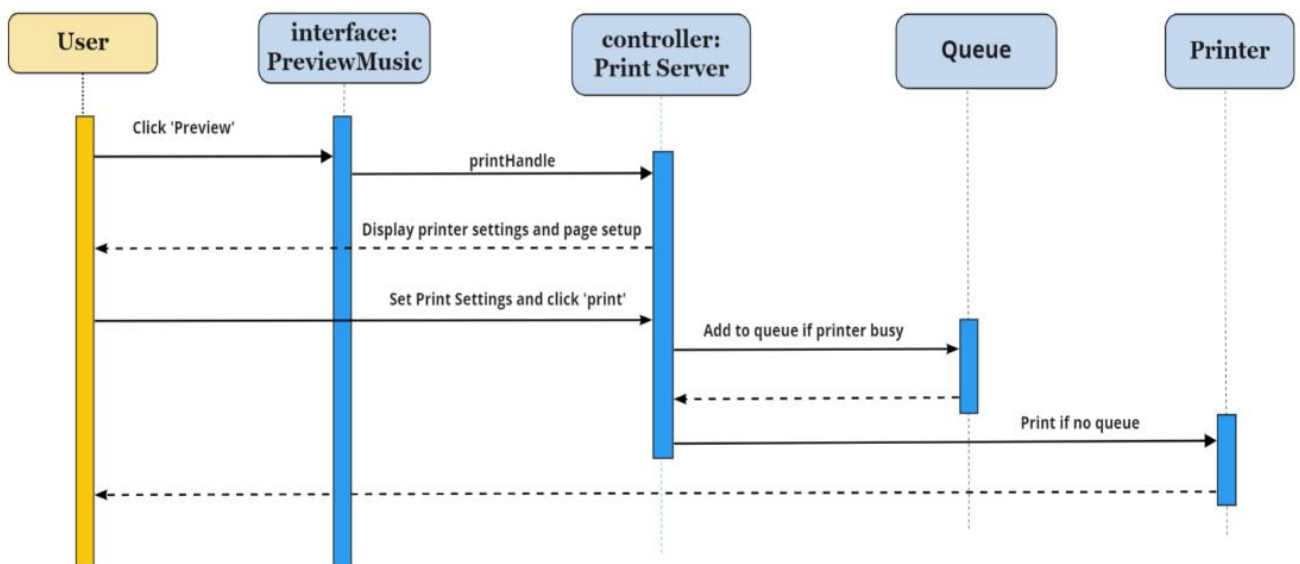


**Figure 11.** The public, private, protected attributes and operations are denoted by “+,” “-,” and “~” respectively.

## 4. Printing the Music sheet

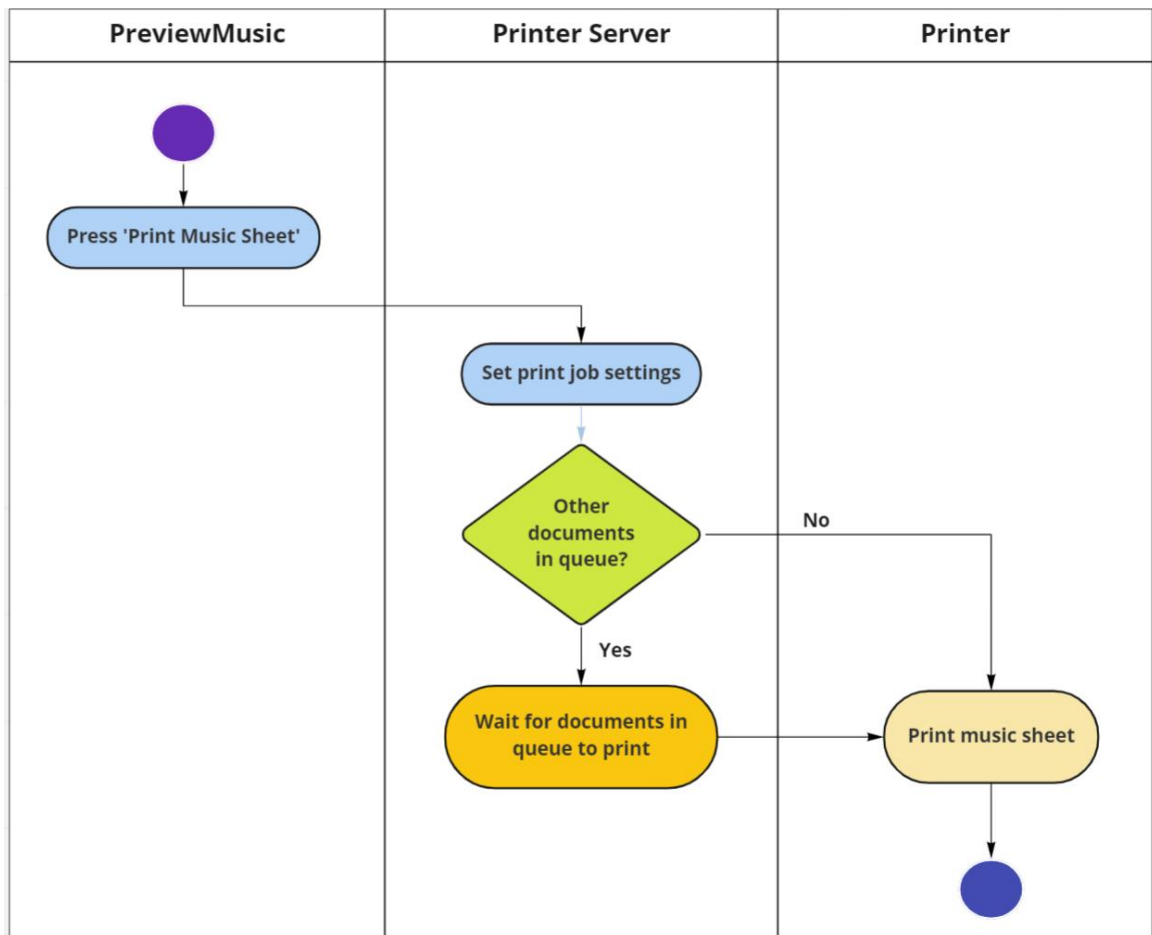
In this section, we will discuss the sequence and activity diagrams describing the events taken to print/save the music sheet from the moment that the user clicks on the print button on the preview window.

### 4.1. Sequence Diagram



**Figure 12.** Sequence diagram for the print function.

## 4.2. Activity Diagram

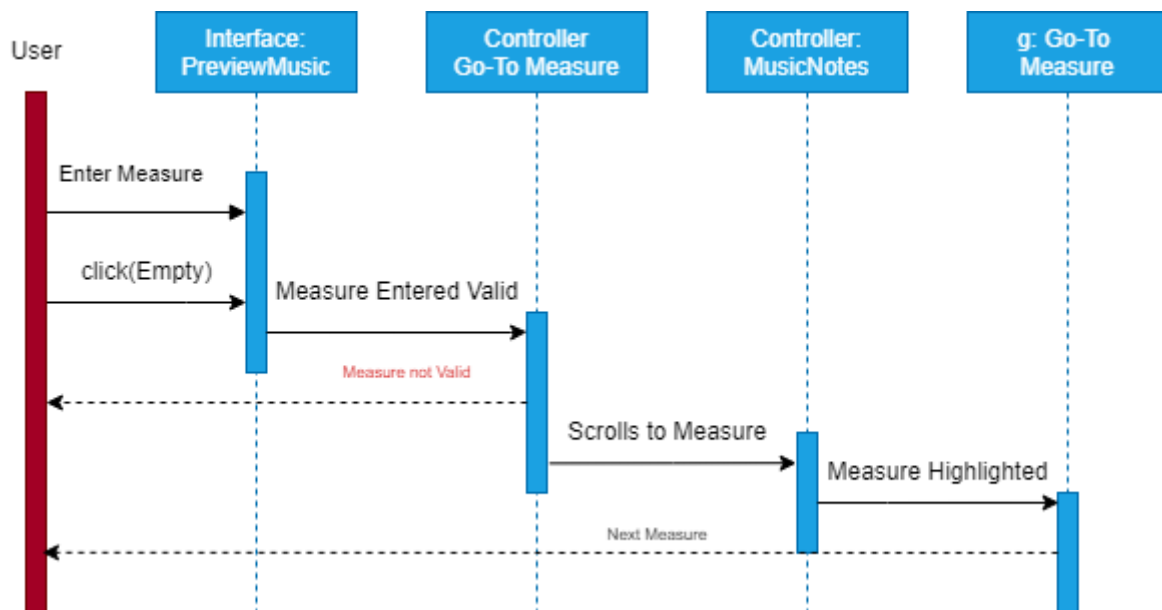


**Figure 13.** Activity diagram for the print function.

## 5. Go to measure

This section includes an activity diagram describing the event taken from when a user enters a specific measure number to highlight the measure.

### 5.1. Activity Diagram



**Figure 14.** Activity Diagram showing the events that result in highlighting a measure.