**First Draft of Design**

**ABC Grammar to be implemented**:

abcFile ::= abcHeader abcBody

abcHeader ::= fieldNum title comment\* optionalFields\* key  
fieldNum ::== “X:” [0-9]+ end  
title ::= “T:” text end  
key ::= “K:” baseNote (‘#’ | ‘b’)? (’m’)? end  
optionalFields ::= composer | noteLength | meter | tempo | voice | comment  
composer ::= “C:” text end  
noteLength ::= “L:” fraction end  
meter ::= “M:” (“C” | “C|” | fraction) end  
tempo ::= “Q:” [0-9]+ end  
voice :: “V:” text end

abcBody ::= line+  
line ::= (element\* endOfLine) | voice | comment  
element ::= noteRep | tuplet | barline | repeat | space  
noteRep ::= note | chord  
chord ::= “[“ note+ “]”  
note ::= noteType (noteMultiplier)?  
noteType ::= pitch | rest  
noteMultiplier ::= ([0-9]\*) (“/” [0-9]\*)?  
rest ::= “z” (noteMultiplier)?  
pitch ::= (accidental)? baseNote (octave)?  
accidental ::= “^” | “^^” | “\_” | “\_\_” | “=”  
baseNote ::= “C” | “D” | “E” | “F” | “G” | “A” | “B” | “c” | “d” | “e” | “f” | “g” | “a” | “b”  
octave ::= “’”+ | “,”+  
tuplet ::= “(“ [2-4] (noteRep{2} | noteRep{3} | noteRep{4})  
repeat ::= repeatSimple | repeatAlternate

repeatSimple ::= ( repeatStart noteRep+ (barline noteRep+)\* repeatEnd)

repeatAlternate ::= ( repeatStart noteRep+ (barline noteRep+)\* “[1” noteRep+ (barline noteRep+)\* “[2” noteRep+ repeatEnd)

barline ::= “|”

repeatStart ::= “|:”

repeatEnd ::= “:|”

majorEnd ::= “||” | “|]”

text ::= [.]  
newLine ::= “\n”  
space ::= “ “ | “\t”   
comment ::= “%” text endOfLine  
end ::= endOfLine | comment

endOfLine ::= newline | endOfFile

**Datatypes, Classes and Interfaces:**

**Accidental:**

**/\*** '^','=','\_','^^','\_\_' \*/

Enum Type {

Sharp, Neutral, Flat, DoubleSharp, DoubleFlat}

**Voice:**

***Attributes:***

Identifier, ArrayList<musicSequences> musicSequences

**class** **MusicSequence** // we will enforce each implementer to have a play function that will handle how each

// type is handled

**Chord extends MusicSequences:**

***Attributes:***

ArrayList<Note> notes

**Tuplet extends MusicSequences:**

***Attributes:***

ArrayList<Note> notes

**Repeat extends MusicSequences:**

***Attributes:***  
 ArrayList<Note> notes, type

**Note extends MusicSequences:**

***Attributes:***

baseNote, octave, accidental, length

**Pitch extends Note**:

***Attributes:***

baseNote, octave, accidental, length

**Rest extends Note:**

***Attributes:***

baseNote=z, octave = null, accidental = null, length

**Bar** **implements MusicSequences:**

***Attributes:***

List<MusicSequence>

**Player:**

***Attributes:***

musicSequence body, Header header

**Header:  
 *Attributes:***

String composer, title

Key key

Fraction meter, tempo, defaultNoteLength

**Body:  
 *Attributes:***

List<Voice> voiceList

**Key:  
 *Attributes:***

baseNote, accidental, minor