

I. Overall Architecture

Starting from the user view, an HTML form will allow the user to input the data. After the form is completed, it is submitted to the database as PHP data. The database query is then run through our python back end that takes the queried data and turns it into a MIDI file. This file is then sent back to the HTML page in an embedded media player.

II. Front End Description

The front end for the RandoMIDI application is very simple, and involves two major components: the midi generator form, and an embedded midi player.

1. **MIDI Generator Form**

The main page of the application is a form that allows the user to select all of the options he or she would like the application to use when generating a MIDI file. Here is the current working list of available options:

- * Style
 - * Classical
 - * Blues
 - * Rock
- * Tempo
- * Mode
 - * Major
 - * Minor
 - * Phrygian
- * Instruments
 - * Acoustic Guitar
 - * Electric Guitar
 - * Bass Guitar
 - * Piano
 - * Violin
 - * Cello
 - * Flute
 - * French Horn
 - * Percussion
- * Motif
 - * (See below)
- * Key
 - * A-flat
 - * C

- * F-sharp
- * etc.
- * Time Signature
 - * 4/4
 - * 6/8
 - * 3/4
 - * 2/2
- * Main Chord Progression
 - * (See below)

If any of the options are left blank, then the system will make a choice at random. Leaving every field blank will generate a completely random MIDI file. If the user wishes to create a unique motif, then they will be taken to a new page where they can edit a small (three measures at most) musical phrase. That phrase will then be saved to the database and be available for future use by all users. Additionally, the motif dropdown will contain English descriptions about what the motif sounds like, for example, "bright," "cheery," "dreary," "weird," "ominous," etc. The user can also specify a primary chord progression to use. For example, the most common progression for popular music is I, IV, V, I. The program will primarily use the progression the user entered, but will intersperse other chords as well.

2. **Embedded MIDI Player**

After the user submits all their choices from the form, the application will generate the MIDI file on the server and send it back to the browser. To give the user the satisfaction of immediately hearing their creation, we will embed a simple media player in the browser. The user can play the song as many times as they like. We will probably have to use an existing JavaScript library for this.

III. PHP Back End

1. **Form Processing**

When the HTML form is filled out to the user's specification, PHP will retrieve the data requested in the form via a PHP script. It will run a query (examples in III.3.) on the database. After the query is ran, the output will be formatted and sent to our Python backend (part IV).

2. **Database**

The database is where all of our data for Motifs, Instruments, Progressions, etc. are stored. It is queried by the PHP script and returns the data that is queried. In the future, we may also use it to store the user created music. This is unforeseen.

3. **Queries**

Select a random motif in the key of C major:

```
`SELECT *`  
`FROM Motifs`  
`WHERE Key='C'`  
`AND Mode='major'`  
`ORDER BY RAND()`  
`LIMIT 0,1;`
```

Get all chord progressions and instruments for a blues song:

```
`SELECT Instruments.Name AS Name, Chords`  
`FROM Instruments, Progressions`  
`WHERE Instruments.Style='blues'`  
`AND Progressions.Name LIKE 'blues%'`
```

Get a motif that harmonizes a C major chord:

```
`SELECT Motif`  
`FROM Motifs, Scales`  
`WHERE Motifs.Key='C'`  
`AND Motifs.Mode='major'`  
`AND Scales.Tonic='C'`  
`AND Scales.Mode='major'`  
`AND (`  
  `Motif LIKE '[(\C%'`  
  `OR Motif LIKE '[(\E%'`  
  `OR Motif LIKE '[(\G%'`
```

`)`

IV. Python Back End

After the PHP code gets the user input, retrieves the appropriate info from the database, and formats the data as input for MIDI generation, a Python script is called, which will take the data and programatically create and return a MIDI file. We will use a Python music package called "mingus" to do all MIDI processing. The arguments to calling the script will be contained in a Python-style dictionary. Here is an example of calling the script from PHP:

```
`$command = escapeshellcmd("python3 ./python/createMidi.py $arguments");`  
`$output = shell_exec($command);`
```

where `\$arguments` is a variable that contains a dictionary like the following:

```
`{'tempo': 120, 'mode': 'major',`  
`'instruments': ['acoustic guitar', 'piano'],`  
`'key': 'E', 'time signature': '4/4',`  
`'progression': ['I', 'IV', 'V', 'I'],`  
`'motif': [('C4', 'e'),('D4', 'e'),('A4b', 'q')]}`
```

This string will be built up in PHP from the form and database information. The script will then use this data to generate a MIDI file using the methods contained in the mingus library. The return value will be a .mid file, which will then be played by the embedded player.