Software Design Overview

Version 1.3
12/11/2013

Bassoon Fingering Finder (BFF)

Team Two Buffoons and a Bassoon:
Ryan Sacksteder
Pierce Trey

Prepared for CS481
University of Idaho - Dr. Paul Oman
Fall 2013

Table of Contents

List of Figures

1.0 Introduction

1.1 Purpose

This Software Design Overview provides design details for the Bassoon Fingering Finder (BFF) project. The expected audience of this document includes our customer (Dr. Susan Hess), advisor (Dr. Oman), future project developers and any users wishing to view the project's design description.

1.2 Scope

This document contains an overview of the design of the Bassoon Fingering Finder web application. This document will describe the internal architecture of the Bassoon Fingering Finder application, as well as a couple external entities; an application database and web host (Heroku).

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Heroku - a cloud platform as a service (PaaS) supporting several programming languages
1.3.2 Model View Controller - a software architecture pattern which separates the representation of information from the user's interaction with it
1.3.3 Ruby on Rails - often simply called Rails, is an open source web application framework which runs on the Ruby programming language
1.3.4 PostgreSQL - often simply called Postgres, is an open source object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance
1.3.5 Bassoon - a woodwind instrument in the double reed family that typically plays music written in the bass and tenor clefs, and occasionally the treble
1.3.6 Fingering - the choice of which fingers and hand positions to use when playing certain musical instruments (in this case, a bassoon)

1.4 References

Heroku: https://www.heroku.com
Ruby on Rails: http://rubyonrails.org
Wikipedia: http://en.wikipedia.org
SDD Template: www.jsu.edu/mcis/docs/SDDTemplate.doc (Jacksonville State University)

1.5 Overview

The remaining sections of this document are as follows:

- Section 2 Deployment Diagram: provides a visual representation of the system and gives additional information which describes the possible implementation of the system.
- Section 3 Architectural Design: gives a detailed look at the individual components within the system and how they all work together.
- Section 4 Database Design: looks at the individual tables within the database and briefly overviews their contents.
- Section 5 User Interface Design: describes both the expected output and user interaction with the software.

2.0 Deployment Diagram

BFF is hosted via Heroku web services on a cloud platform. Users can access BFF by visiting bassoonfingeringfinder.herokuapp.com with a web browser. User interaction with the application then results in queries to a PostgreSQL database associated with the application.

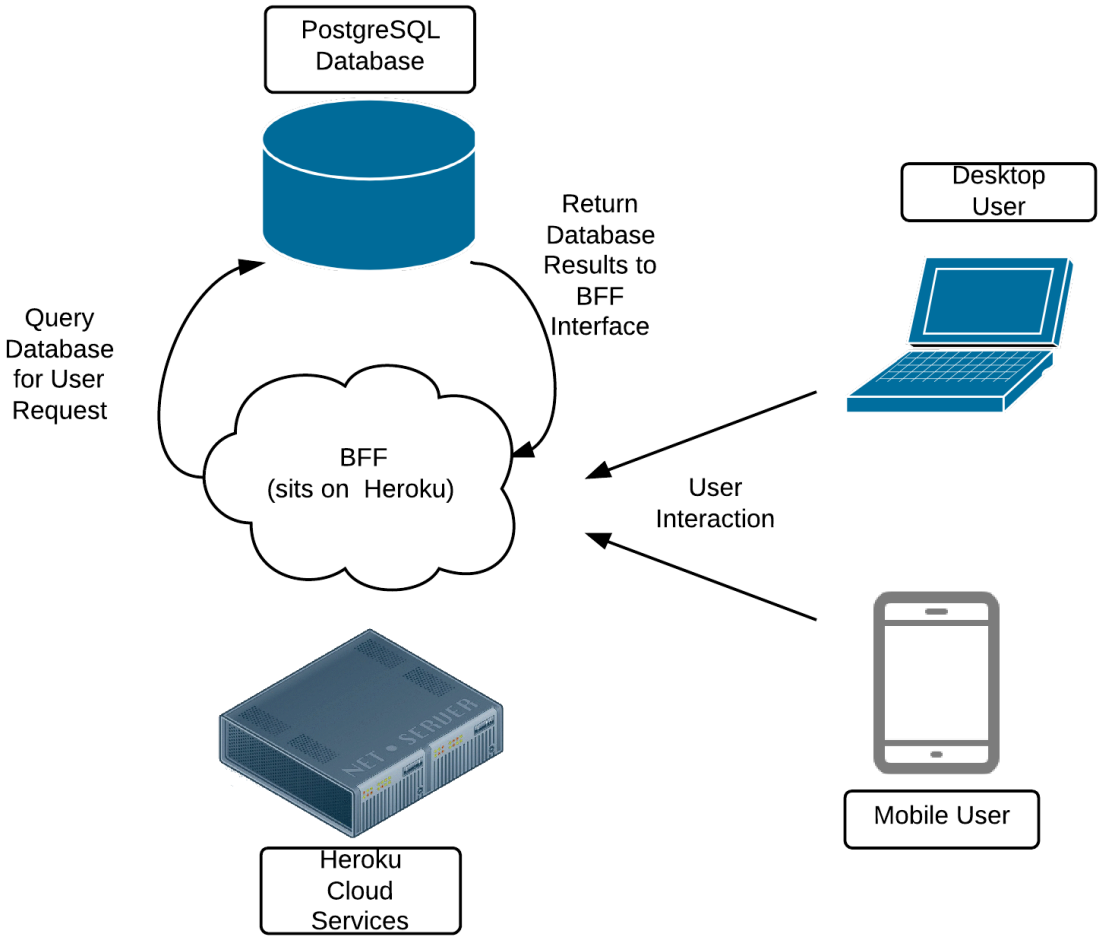
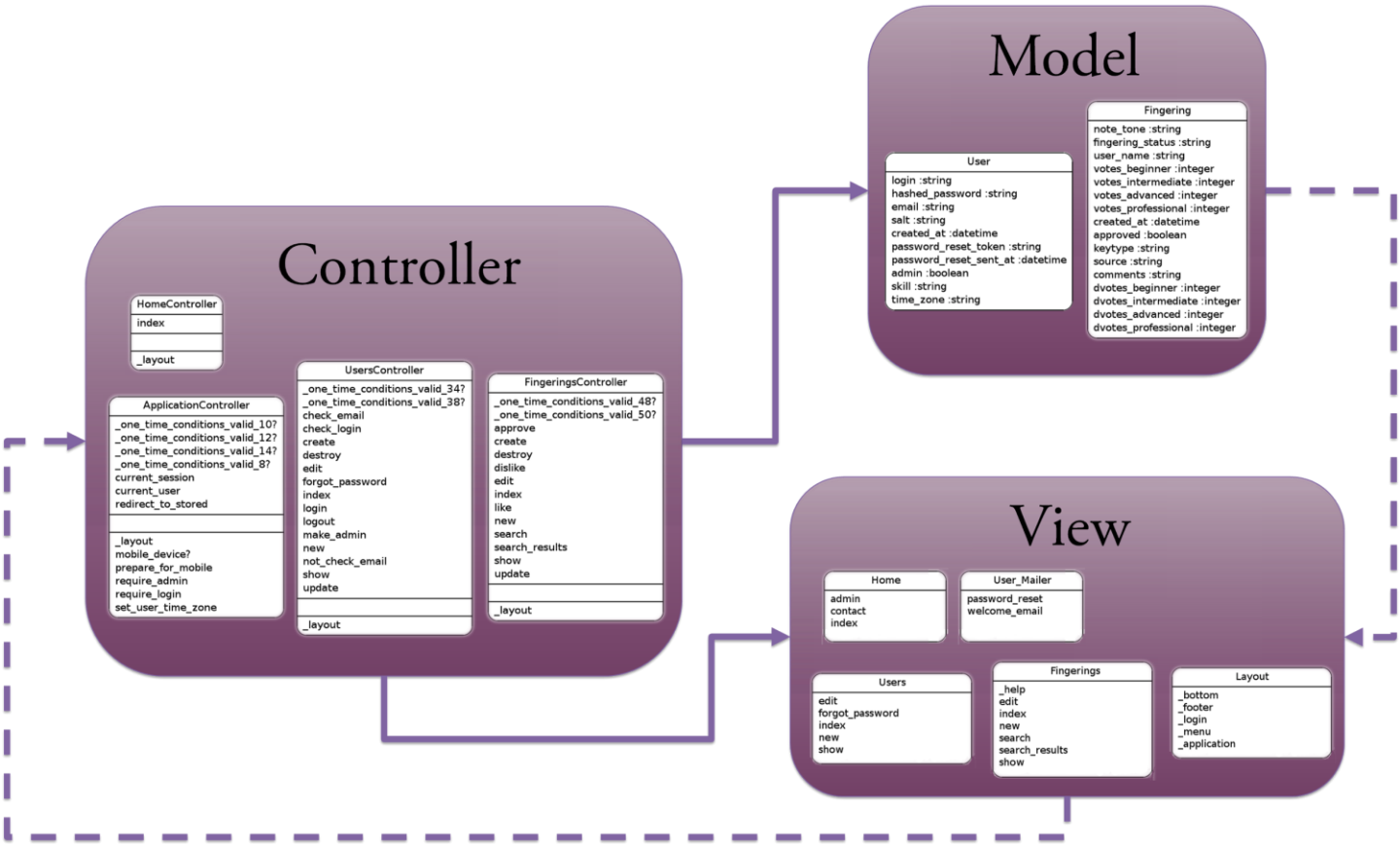


Figure 1: Deployment Diagram

3.0 Architectural Design

3.1 Overview

As a Ruby on Rails application, BFF is built off the Model-View-Controller architecture (seen in Figure 2 below). In this type of architecture, it is easiest to think of models as data or objects in the application, views as different windows and user interfaces shown on the screen, and controllers as the connection between the two. A controller allows the view to communicate with a model. The final component of



BFF's architecture is a PostgreSQL database.

Figure 2: BFF Model-View-Controller Architecture Diagram

3.2 Models

BFF has two models. The first is the User model, and the second is the Fingerings model. Each model contains data pertinent to that object within the application.

3.2.1 User Model

The User model contains the data for an instance of a user. This data includes username, a hashed password, user email, a date/time of when the user was created, the user's skill level and time zone, as well as a flag which specifies if the user is an administrator. The information for a particular user is

loaded into the User model, upon login, from BFF's PostgreSQL database. Changes to a user model are maintained locally in that model until a save event is processed to store those changes to the database, making them permanent.

3.2.2 Fingering Model

The Fingering model contains data specific to a fingering. A Fingering is a much more extensive collection of data than a User, as many more fields are required to construct a Fingering instance. Fingering model data includes: note tone (the note, accidental, and octave associated with the fingering), fingering status (a string representation of the fingering chart for this instance), the type of fingering (standard or alternate), an order for which the fingering should be displayed within a collection of similar fingerings, the source of the fingering (such as a reference book), the name of the user who created the fingering, a date/time at which the fingering was created, a flag for whether or not the fingering has been approved by a site administrator, a series of rating values based on user opinions of the fingering, and finally a comments field for any additional fingering details. Like with the User model, information for a particular fingering is loaded from the application database into the Fingering model during the course of a fingering search, or while adding a fingering. Permanent changes to a fingering take place when a save or remove fingering event is processed.

3.3 Views

BFF views can be broken into four main categories. User views, are views which allow the user to perform user-specific operations. Fingering views display fingering related information to the user. Home views are associated with neither users nor fingerings specifically, but are general views about the application which are served to the user. The last component of views are layouts, which provide a template from which many views are constructed.

3.3.1 User Views

These views present information related to a user account. This includes a view allowing for the creation of a new user (registration), the editing of certain information about an existing user (changing email, password, time zone and skill level) and the listing of information about a user. There is also a view allowing a user to send a password reset (randomly generated) to the email associated with their account. Usernames cannot be changed once set.

Admins are able to edit the information for any existing user, delete any user, as well as view a listing of all users in the database. Standard users are able to see information related to their account only as well as edit their information or delete their account.

3.3.2 Fingering Views

These views present information related to a fingering in the database. This includes a view for the creation of a new fingering, the editing of an existing fingering, the listing of all available fingerings in the database, the searching for fingerings in the database and the displaying of search results from a query.

Admins are able to edit any fingering in the database, view all (approved and unapproved) fingerings in the database, change the order in which fingerings are displayed, change the keytype of a fingering and add individual fingerings from the halves of a trill fingering when adding a new fingering or editing an

unapproved fingering. Standard users can only edit fingerings that they submitted as well as only view fingerings that have been approved by an admin.

3.3.3 Home Views

These views present information about the application. This includes a home view that allows users to see information about the application (current state of development/notices), a contact view that allows users to find information on how to contact the developers of BFF as well as submit any bugs that they discover in the application.

Admins have an additional view that allows them to view currently available tools for interacting with the application. This includes a listing of all users in the application that an admin can use for editing information about users, changing the admin status of users as well as deleting users. These tools may grow pending future development.

3.3.4 Layouts

These views contain the html/ruby for displaying the header and footer of the BFF web application. This includes the login username/password fields, forgot password link, logo, general application info (footer), navigation (header/footer) and everything related to the head section of an html page.

3.4 Controllers

There are four primary controllers in BFF. The first is the application controller, which is a standard controller to track application state. The second controller is the home controller. It process events from user interaction on the home views. The remaining controllers do the majority of work throughout BFF, taking user input from views and operating on the User and Fingering models.

3.4.1 Application Controller

The Application controller manages a number of application specific details. It also handles general application events and ensures that appropriate views are served to users. As far as application specific details, this controller keeps track of session details such as cookies, as well as the current user (a reference to a User model). Using these details, the Application controller manages dates/times based off the user time zone, and serves views based off the device being used during the active session (mobile vs desktop views). In addition this controller manages which views or variations of a view the user will be presented based off their administrator status and whether they have logged in to a user account.

3.4.2 Home Controller

The Home controller enables the application to process events from the Home views. This controller mainly handles travel between the Home views, and also linking to external tools such as the GitHub bug tracker that is associated with development of BFF.

3.4.3 Users Controller

The Users controller interacts between the User model and all User views. When a user invokes an event within a User view, the User controller is notified of the event and carries out a series of routines to read, edit, or save information to the corresponding User model. The BFF Users controller manages the following actions: creating a new user, verifying user credentials such as email address and login information, editing and displaying existing user account settings, deleting a user profile, resetting user

passwords in the event of a forgotten password, logging a user in and out, changing a user's administrator status, and permanently updating user information in the database.

3.4.4 Fingerings Controller

The Fingerings controller binds the Fingerings model to Fingering views. When a user performs an action on a fingering in BFF, the Fingering view emits a signal which is caught and processed by the Fingerings controller. The Fingerings controller then communicates with the Fingering model, compiles results of the interaction with the Model, and passes information back to the view for display. The Fingerings controller in BFF manages the following actions: creating a new fingering, editing and displaying details for an existing fingering, deleting a fingering, updating fingering ratings, searching for a fingering, compiling fingering search results and applying a variety of search filters, as well as updating a fingering in the database.

3.5 PostgreSQL Database

When a controller needs access to information not currently in a model, it performs a database query, crunches the data, and presents it to the user through the view interface. BFF uses a PostgreSQL database since it is the standard for Heroku-based web applications. The database design is discussed further in the section 4 of this design overview.

4.0 Database Design

4.1 Fingering Table

The Fingering database table includes the following columns, which contain all information necessary for displaying the fingering, information about the fingering and for sorting fingerings in a list of queried results.

Column Name	Data Type	Description
id	integer	unique id for this Fingering
note_tone	string	string representation of the musical note(s) associated with this fingering
fingering_status	string	a string representation of the fingering
user_name	string	the application login name of the user who submitted the fingering
votes_beginner/votes_intermediate/votes_advanced/votes_professional	integer	(4 separate columns)the number of positive votes from beginner/intermediate/advanced/professional categorized users that this fingering has received
dvotes_votes_beginner/dvotes_intermediate/dvotes_advanced/dvotes_professional	integer	(4 separate columns) the number of negative votes from beginner/intermediate/advanced/professional categorized users that this fingering has received
created_at	date/time	the timestamp of when this fingering was created
approved	boolean	boolean value indicating whether or not fingering has been approved by a site admin
keytype	string	string representing whether the fingering is of type standard or alternate
source	string	string (entered by user) citing the source where a fingering came from (reference book, etc.)
comments	string	string (entered by user) detailing additional information on how to play the associated fingering
score	float	calculated ranking of the fingering based off of the up/downvotes of an individual fingering

show_first	boolean	column used to temporarily mark a fingering for displaying others in a list (when viewing a specific fingering from the all fingerings table)
octave	integer	integer containing the octave of the note associated with the fingering - used for sorting
accidental	integer	integer containing a representation of the accidental (1 for flat, 2 for natural, 3 for sharp) of the note associated with the fingering - used for sorting
note_name	string	string containing the name of the note (a, b, … , g) of the note associated with the fingering - used for sorting
admin_order	integer	integer containing the position of the fingering in the list of other fingerings with the same associated note - used for sorting

Figure 3: Fingering Database Structure

4.2 User Table

The User database table includes the following columns that contain all necessary information about the user for associating them with their entered fingerings, emailing them notices, determining whether or not they can access admin features, logging into the application via a password that they set and resetting their password via their associated email.

Column Name	**Data Type**	**Description**
id	integer	unique id for this User
login	string	the user's display name (used in the application and with associated fingerings to refer to this user)
hashed_password	string	a hashed rendering of this user's login password
email	string	the user's email associated with this account
salt	string	a random string used for hashing the user's password
created_at	date/time	the timestamp at which this user created their account
password_reset_token	string	string indicating that the user's password was reset
password_reset_sent_at	date/time	the timestamp at which this user's password was reset

admin	boolean	a boolean value indicating whether or not this user has admin status
skill	string	a string indicating whether this user is classified as beginner, intermediate, advanced or professional
time_zone	string	a string indicating which time zone this user is operating in

Figure 4: User Database Structure

5.0 User Interface Design

5.1 User Input

The standard user will be able to:

- View information about the application
- Search for existing fingerings
- Enter new fingerings and submit them for approval
- Edit existing fingerings that they entered
- View all approved fingerings in the database
- View information about the application/how to contact the application developers

The admin user will be able to:

- Do everything that the standard user can do, and…
- View all fingerings in the database (regardless of approval status)
- Edit any fingering
- Approve/disapprove fingerings
- Grant/revoke user admin status

5.2 Home Page

This is a simple informational page that can be used to update users as to the status of the application. There is no interactive functionality present on this page. This page is visible to anyone visiting the site, whether a registered user or not.

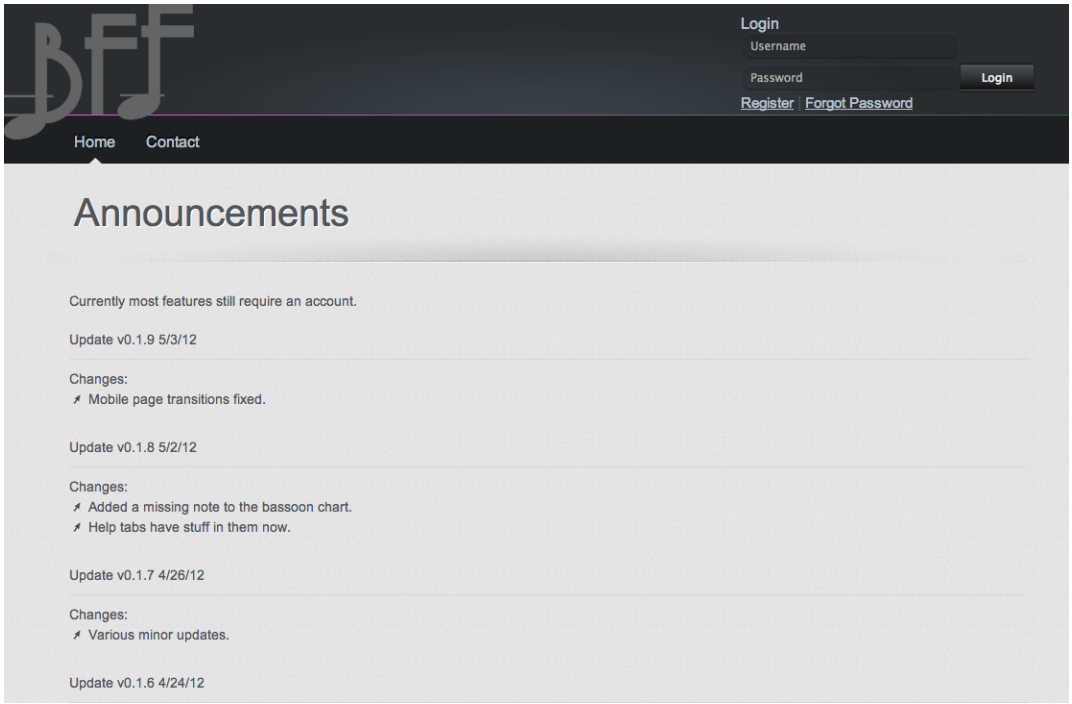


Figure 5: Home Page

5.3 Search Fingering Page

This page contains a diagram of a musical staff that can be used for selecting a specific note (using bass, tenor or treble clef) by clicking on the various elements of the diagram. Multiple clicks on a note head change the quality (accidental) of the note. This page also contains an option for filtering search results by standard/alternate fingerings.

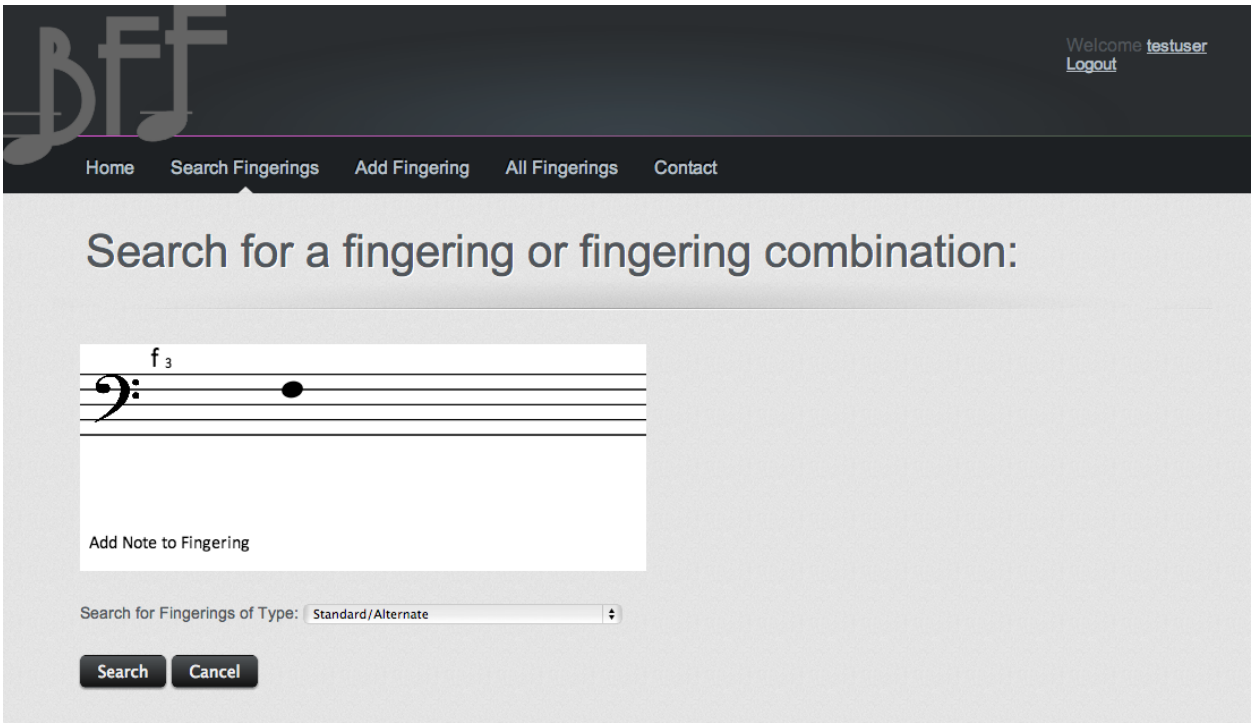


Figure 6: Search Fingerings Page

5.4 Add Fingering Page

This page contains an editable fingering diagram that can be used (by clicking on the various elements inside the diagram) to create a new fingering and select which note it is associated with. This page also contains fields for a user entering a fingering to add a source from which they got this fingering, as well as comments about the fingering before submitting it for approval. The user has the option of adding a second note to this fingering (making it into a trill fingering). This page is only visible to registered, logged-in users.

If the user viewing/using this page is an admin, they will also have options for setting the fingering as standard/alternate (other users' fingerings default to alternate) as well as for setting the display order of the fingering. If the fingering they are entering is a trill fingering, they will also be able to select an option to save either or both of the halves of the trill fingering as individual fingerings.

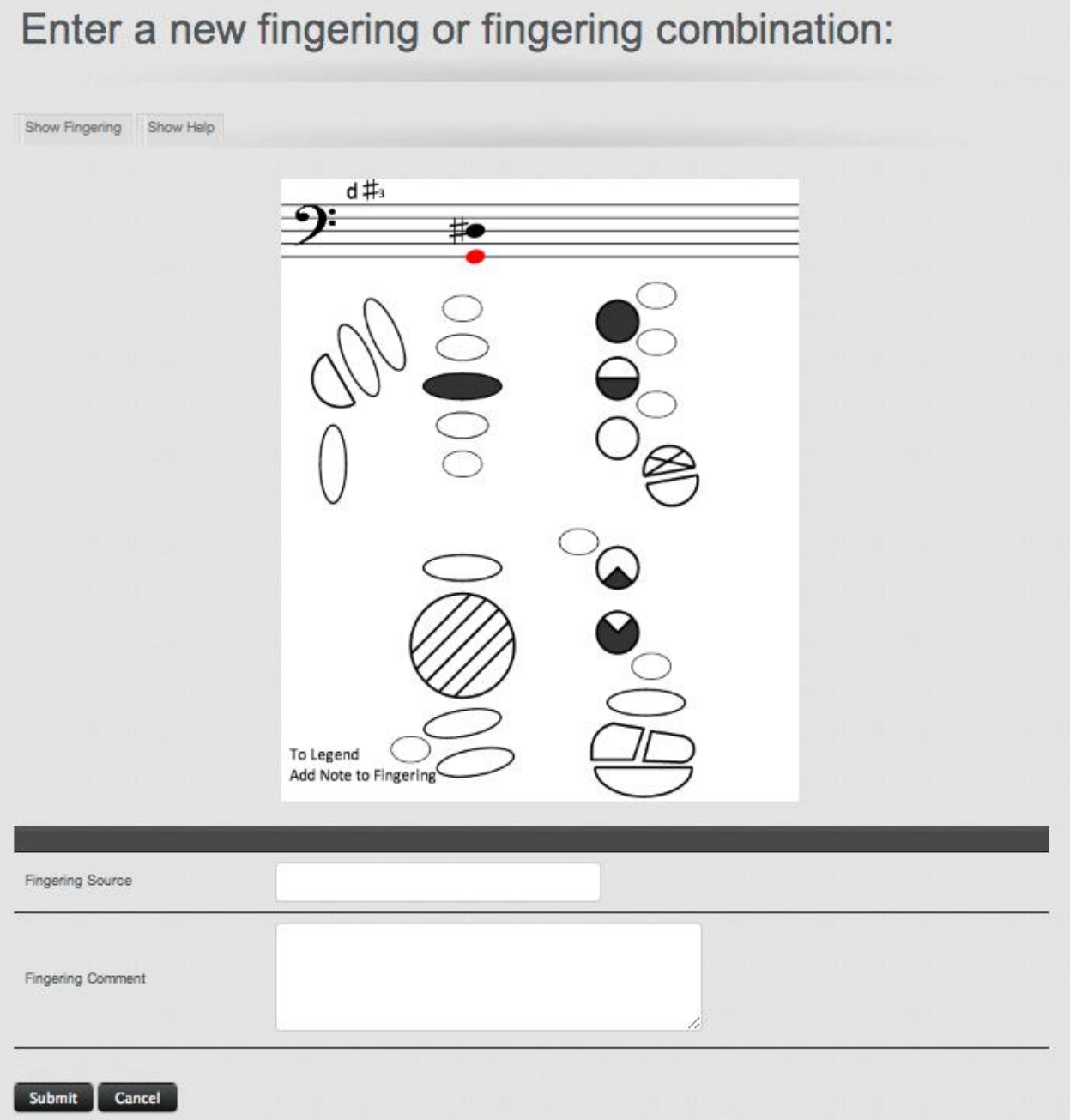


Figure 7: Add Fingering Page

5.5 Edit Fingering Page

This page is equivalent to the Add Fingering Page, except that the fingering displayed for modification is already in the database. Users are able to edit fingerings that they have entered, though if the fingering has already been approved before they edit it, it will need to be approved again before being visible to the general user. Admins can edit any fingering.

5.6 All Fingerings Page

This page displays a hierarchical listing of all fingerings available in the application. The list is ordered first by octave (low to high), then by note name (from a-g) then by accidental (flat, natural, then sharp), producing a list in frequency order as is standard in the musical world. Each note row can be clicked on, expanding into a selection of sub-categories of available fingerings for that note. These categories can then be clicked on, allowing a user to easily sort through hundreds of fingerings to find a fingering that suits their needs. Each of the blue links can be clicked on to display all results for that note or note/category combination contained within the submenu below the row. This table provides an alternate way of searching for a fingering for all users, and also adds a way for an admin to easily find and review fingerings needing approval.

Note: The elements in red will not be visible to the average user, as only admins will be presented with fingerings that have not yet been approved.

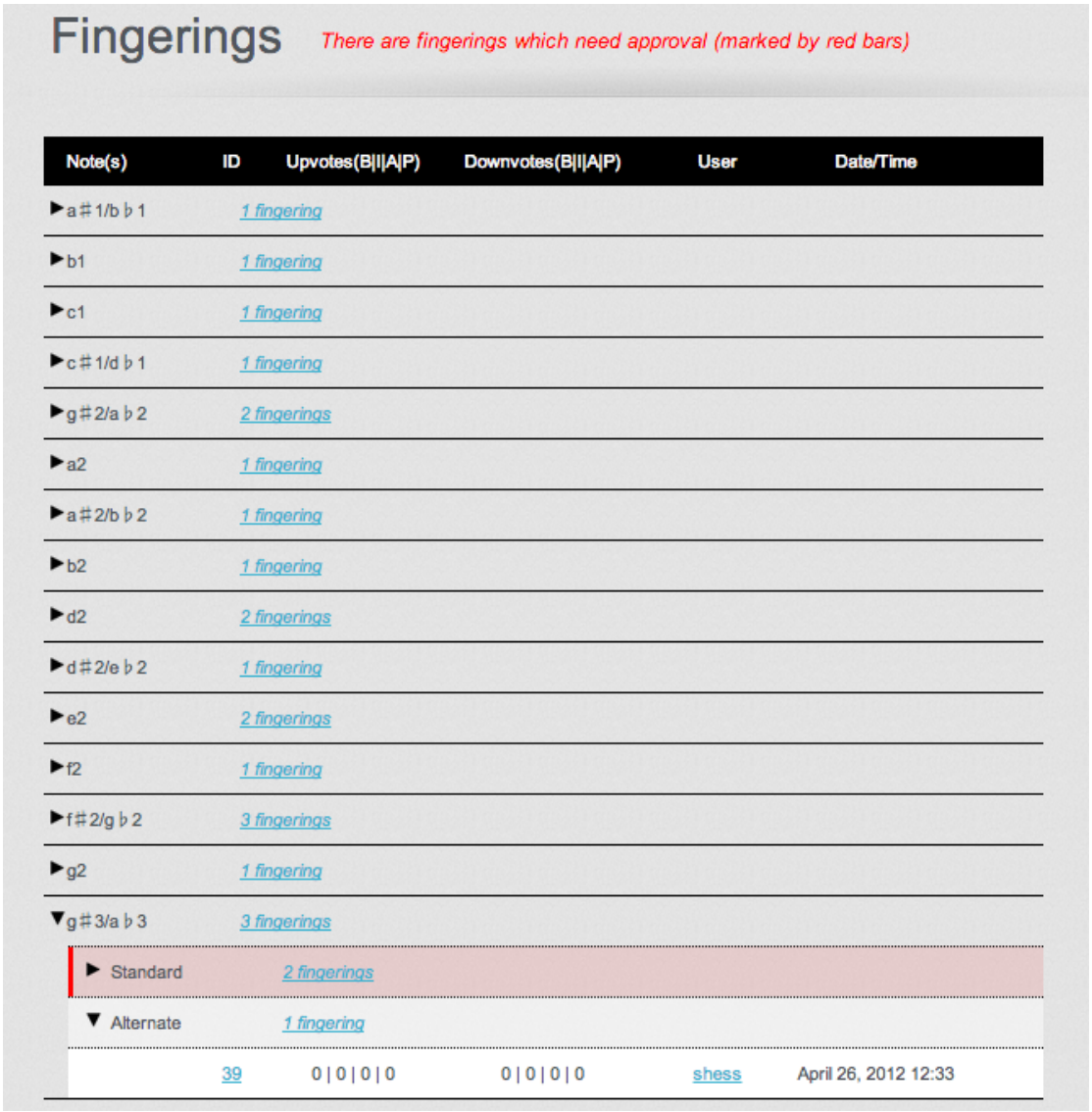


Figure 8: All Fingerings Page

5.7 Search Results Page

This page displays the fingerings associated with a user-performed search. This page contains links to pages for all fingerings returned by a search (whether from the Search Fingering page or from a link clicked on the All Fingerings page). The diagram for the current fingering is displayed and information about the fingering is displayed below the fingering diagram. If the fingering is an enharmonic note, then there is a notice at the top of the page notifying the user that this is the case, informing them that results for both the requested note and the enharmonic equivalent are being displayed.

Note: Several rows ("Fingering has been approved/Fingering has not been approved," "Display Order," "Keys String," and Notes String are only visible to admin users.

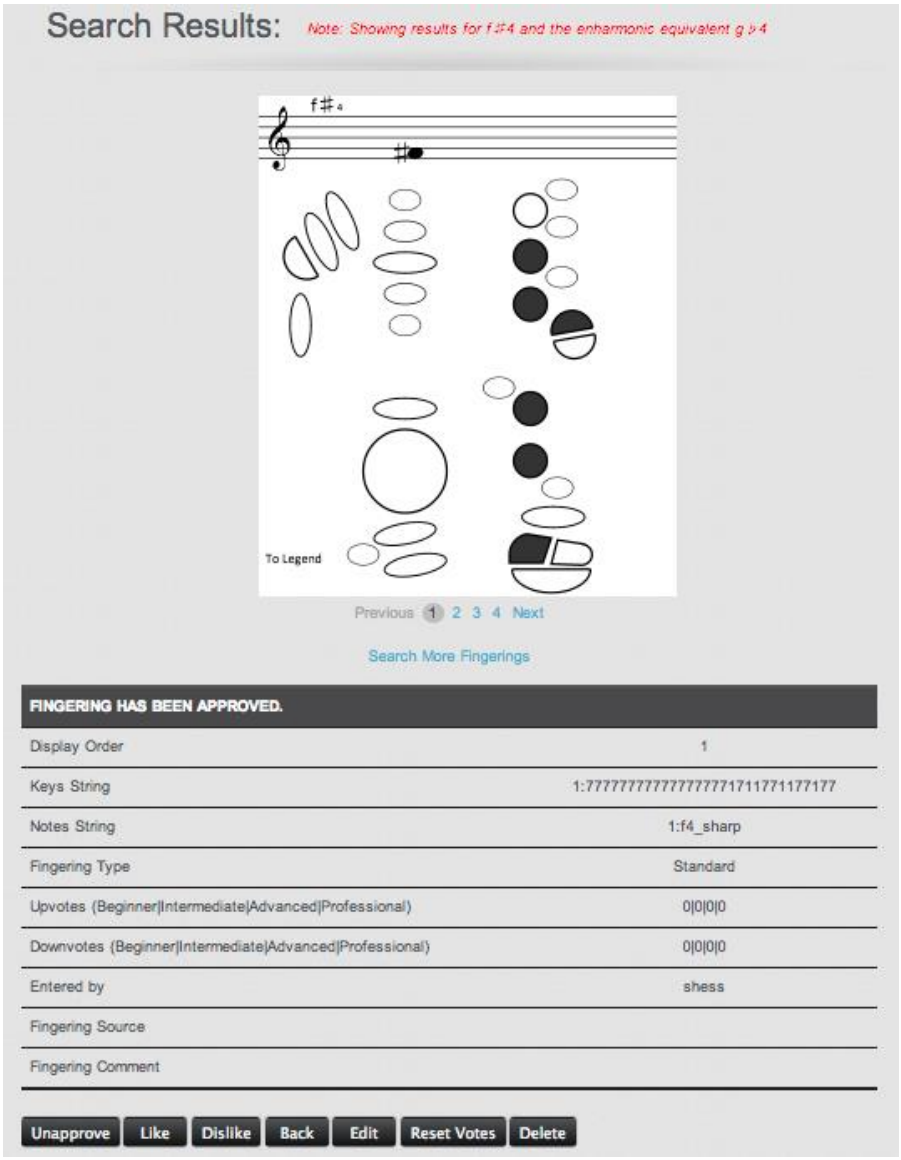


Figure 9: Search Results Page

5.8 Contact Page

This page displays simple information on how to contact the current development team of the BFF application.

5.9 Edit User Page

This page displays information pertaining to a user and allows them to edit information. Admins are able to access this page for all users, while standard users are only able to access this page for their own account.

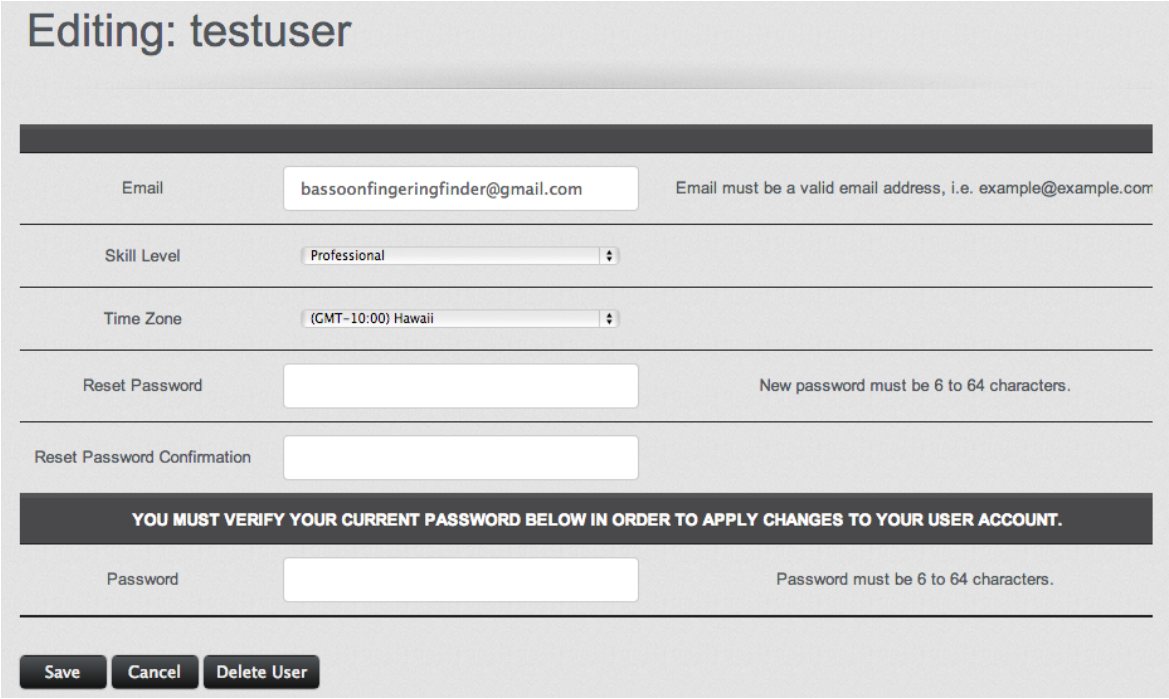


Figure 10: Edit User Page

6.0 Help System Design

The software's simplicity of user interface should eliminate the need for an extensive help system. There are two reference pages on the site. One is a key that describes the fingering diagram and is available via a link inside the fingering diagram wherever the fingering diagram is displayed. The other is a quick reference guide that is available under a tab wherever an editable fingering diagram is present to inform users on how to interact with the diagram and create/edit a fingering. The other features of the application are fairly straightforward and require no user manual. In addition, there are numerous informational messages that inform the user whether or not their action was successful.

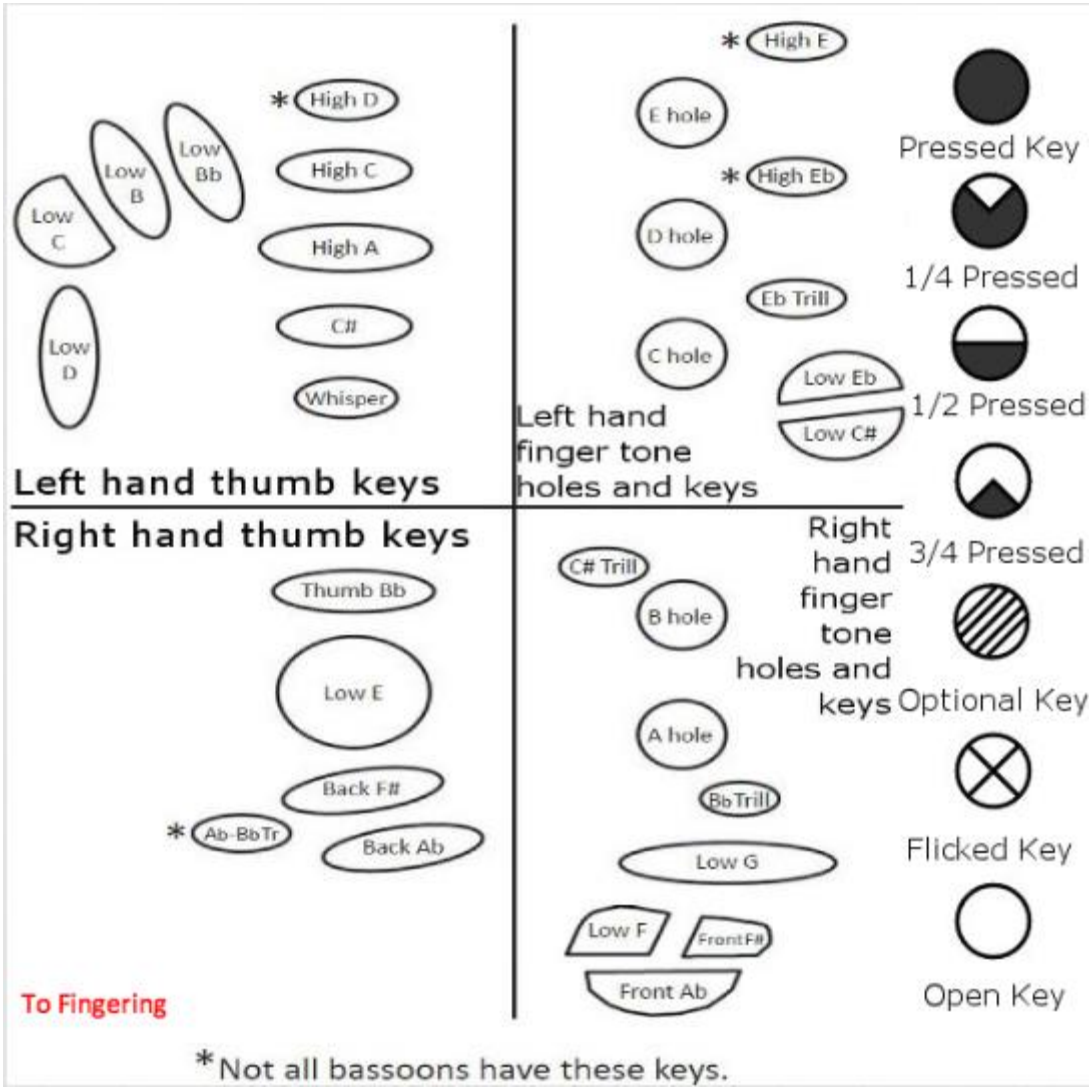


Figure 11: Fingering Diagram Key