

Good improvements over initial draft.

G 4

F 8

D 6

U 4

I 4

S 4

A 4

P 6

40

Bandmaster

David Lordan

Paul Karcher

Dean Marsinelli

91.624 GUI Programming II

Professor Heines

February 13, 2015

Table of Contents

Project Goal.....	2
Program Features.....	2
Admin Page.....	3
Artist/Band Page.....	4
Software Components.....	5
Intended Audience.....	7
Issues To Be Addressed.....	8
Development Schedule.....	9
Minimum Release Criteria.....	10
Additional Criteria	11
Long Term Features.....	11
References.....	12

Project Goal

Our project provides a **means by which** musicians can more easily and effectively collaborate with one another through the Internet. Artists and bands will be able to publicly or privately share recordings, sheet music, images, calendar events, and to-do lists. Groups of **musicians** will have the option of sharing a single account so that all members of a band will have instant access to the content and ideas provided by other users within their group.

Program Features

Bandmaster will be a completely web-based application that will allow users to add, view, share, and **remove content** related to musical projects. This will be achieved by using a two-page system. The first of these will be an admin page that will be for private use, accessible only to the account holder. This may be an individual user or a **group of users in a band**. This page will allow users to manage all of their account's content, the details of which are outlined below.

The other half of the site will be a **public band page**. This may be seen by anyone and will only display **content that has been** specifically chosen on the admin page. This public page may be used for band promotion, to showcase songs the users may wish to share with potential collaborators, or for any other purpose that a user may think of.

Admin Page

Our application's **admin page** will give users complete control of their account's content. The central feature of this page is uploading and organizing files. The admin page will allow users to simply drag and drop files into their appropriate location, or to use a **conventional upload** system whereby users browse their local drive and select a file to upload. After files have been uploaded, they may be categorized as songs, sheet music, images, or any other custom category created by the user.

Songs may be organized into playlists that can be edited at any time. The page will also feature a **built-in audio player** that will support all common audio file types, such as mp3, mp4, ogg, wav, and so forth. When a song is selected, information regarding that file will be displayed. This information may include associated sheet music, user comments, or references to other audio tracks that may be an alternative version of the same song. Different versions of sheet music written for different instruments may also be added for a single song. Any uploaded user content may also be downloaded from this page.

The admin page will also include a calendar, comment board, and to-do list. These three features will allow for the organization of group responsibilities and goals. The dates of **performances, rehearsals, and recording sessions** will be conveniently accessible to all. Thoughts and ideas may be easily shared and stored, while important band tasks can be delegated. In addition, events and tasks

may be set up by the user to be associated with a particular song. This will enable an additional feature of the public page that is described below. This admin page will be designed for non-technically minded users with a simple interface allowing selected content to be displayed or hidden on the public page.

Artist/Band Page

Each account will also have a public band or artist page that will be accessible and viewable by anyone (see Figure 1). The content of this page will be generated by the file structure created by the admin page. The layout for this page will have three main panes. On the left pane, a list of all public songs will be displayed. By selecting a song, it will become active and all associated public information will be shown in the center pane. Again, this may include links to view sheet music, comments, links to download the song, or references to alternative versions of the song. When clicked, these song references will become active and their information will take over the center pane. The left pane may be set to display a list of all songs or a list of playlists.

The right pane will display the calendar and to-do list. Each entry in either the calendar or the to-do list may be set by the user to reference a song or playlist. When selected, the song or playlist will become active and its associated information displayed.

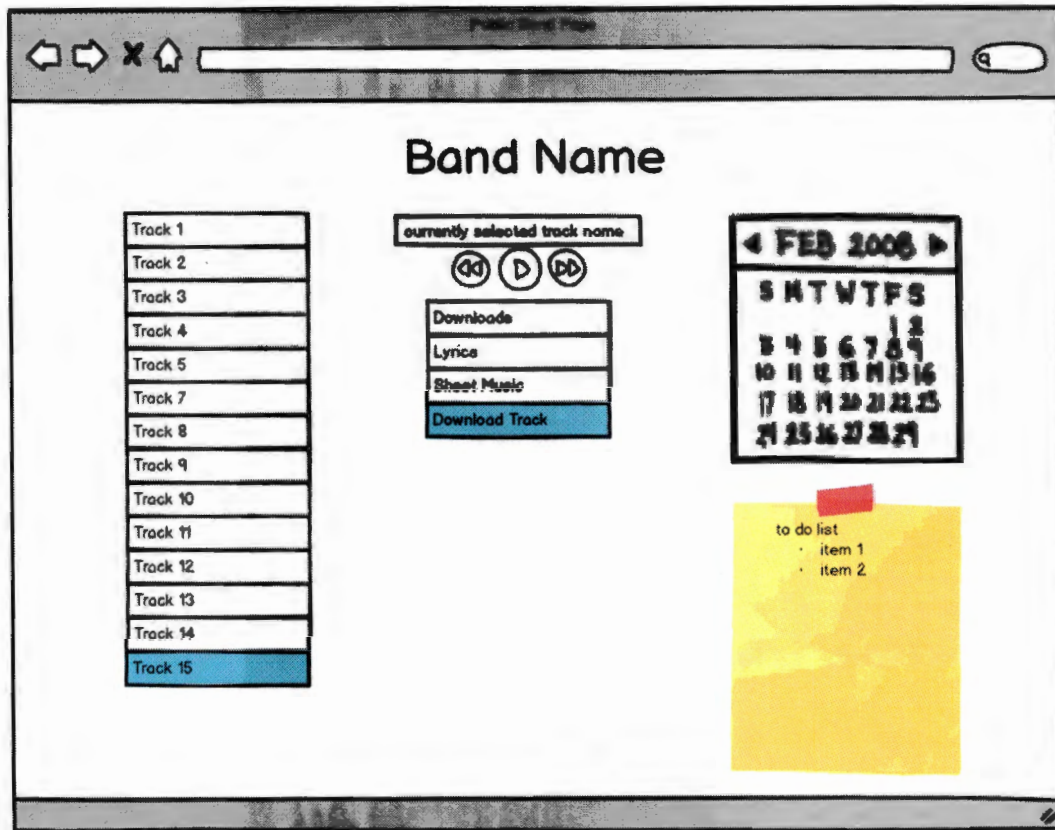


Figure 1. - Band Page mock-up

Software Components

This project will make use of client-side scripting along with a server-side component and a database. We plan to use AngularJS and its MVC pattern to construct the user interface for the admin dashboard page and to arrange and organize the public facing band page. Much of the client-side functionality will also be implemented using jQuery. This includes adding event-listeners, controlling CSS animations, and using the jQuery UI library. Specifically, the UI tabs and accordion widgets will be used to display lists of songs. We will use a

PHP server component to handle account creation, logging in, and controlling how the admin dashboard controls the user's account. The database component will be used to store usernames and passwords to allow for user authentication at login. DropzoneJS will be linked with the PHP server component allowing users to upload audio files and store them in their band's directory on the server. After each upload, a script will append the audio track into a JSON file used to store and organize the current available tracks. This data will be used to populate the public facing band page and allow for music playback. Chrome-Logger (formerly ChromePHP) will be used to troubleshoot and test the server side functionality.

As HTML audio controls are rendered inconsistently between browsers, we will implement an already built custom audio controller that will display appropriately on all platforms. This custom controller will include a dynamic play-head showing the song's current position. The play-head will be draggable allowing users to scroll to any part of a song. This will be achieved using jQuery, AngularJS and CSS animations. We will also be making use of the Bootstrap framework so that the application will be responsive to various platforms and browser window sizes. This will ensure that the program is always displayed in a practical and organized manner.

Intended Audience

Bandmaster is intended for musicians or groups of musicians who wish to keep their creative output organized and easily accessible. The only technical experience required by users is basic web skills, on par with browser navigation and using email. One of the main focuses of this project is creating a simple, clean, and approachable interface so that non-technical users will not be discouraged from using this service.

This application is perfect for groups of musicians that have several rotating members who wish to stay current with the group's activities. For many large bands, it is common to have fill-ins and auxiliary musicians. Often these musicians do not have updated versions of songs, access to new recordings, nor updated sheet music. Also, scheduling mishaps and miscommunications are common. This application will allow everyone in the project to stay updated and informed by having instant access to all of the band's material and schedule. If a band wishes to add a new member to the group, rather than send an email with several recordings and transcriptions, the group can simply send a single URL, giving new members access to all of the band's content. It is also very common for solo artists to hire studio or touring musicians to assist them. Again, rather than send an email or give out a CD with potentially dozens of files, the artist can refer the musician to their *Bandmaster* page.

Issues To Be Addressed

One potential issue that may come up will be displaying an updated file list.

When a file is uploaded, a PHP function will be called that updates a JSON file containing all of the account's content. After the upload has completed

successfully, a client-side AJAX request must be sent to the server to retrieve the

updated list. We would like the server to notify the client that an upload has

occurred and have the client page auto-update. However, this will require the

server sending a push-notification to the client *after* the JSON file has been

updated, which may be difficult to implement. Should we fail to create such

behavior, the user^g will be required to refresh their page^s manually after each file

upload to see the updated list. This can potentially have a negative effect on the

user experience. A possible solution would be to have a JavaScript function

sending repeated AJAX requests over a short interval of a few seconds, constantly

checking for updates. The updated list would simply be retrieved and displayed

after the next request. This would also ensure that the content being displayed

would be consistent for multiple users viewing the page simultaneously on

separate machines.

Consistent display and behavior on multiple platforms including mobile devices

are another potential concern. We will be using the Bootstrap framework to make

the pages responsive to different screen sizes. Unfortunately, Bootstrap has been

known to be problematic when detecting the size of mobile browser windows,

*I have apps
that do
this, too.*

*Yes, but I
have apps
that work
just as you
describe.*

particularly devices using Apple's Retina-Display. We will need to implement a function that determines if the user is on a mobile or stationary device, adjusting the display appropriately. This can be achieved with the JavaScript 'userAgent' function.

Development Schedule

I see you've added who's doing what — Good.

Date	Goals
Feb 15	Upload System - Dave ✓ Develop a working upload system that allows a user to upload a file to a server, append that file data to a JSON file, and use that JSON file to display uploaded files in a public facing page.
Feb 22	Basic Login System - Dean ✓ Develop a working account system where users can create an account and log in to an administrator dashboard. From the dashboard, users should be able to change various settings and these changes will appear on their public band page.
Feb 26	Alpha Version Due – Dave, Dean, & Paul ✓ Using previously developed components, we will have a full user experience where users can create an account, log-in, upload their files and make changes. These changes will propagate to their band's public facing page which will allow for music playback and download. The system should handle multiple accounts at this point.
March 8	Additional Features - Paul We will add additional features to the admin dashboard that users to create and edit a calendar for their band and to add additional information to each audio tracks such as sheet music, guitar tabs, lyrics, etc.
March 22	Improve the Band Page – Dave & Dean We will investigate how to improve the band page, possibly by separating sections into multiple pages. This will largely be

	dictated by user feedback.
April 5	Polish User Experience – Dave & Paul We will continue to polish both the admin dashboard and band page to make our application as easy-to-use as possible.
April 14	Beta Version Due – Dave, Dean, & Paul Our application will be feature complete at this point and we will work on fixing any leftover bugs that may appear. We will continue to take in feedback from users and polish our design.
May 1	Project Completion

Minimum Release Criteria

At the very least, our application will give users the ability to create an account, upload music or transcriptions, and play or view the uploaded files. A list of these files will be displayed on a public facing band page where, again, the files may be played or viewed. These basic features will be implemented with PHP to handle the uploading, AngularJS for the listing of files, and our custom built audio player. The files will also be downloadable from both the admin page and the public page. Additionally, users will be able to view and change both a calendar and to-do list. ^{go}While ^gon the admin page, the user ^smay choose whether ^gor not the calendar and to-do list will be displayed or hidden on the public page. In its simplest form, the page's calendar will contain links to external Google Calendar events. More sophisticated behavior will be implemented as time allows, the details of which are described below.

Additional Criteria

Assuming the minimum release criteria have been achieved, there are several additional features we would like to add to our application. Firstly, we would like ^gfor the user to have the option of creating playlists. The song list will have a button that when clicked will allow the user to enter the new playlist's name. Individual songs will also have a button that will give users the option of adding a song to any of the currently available playlists. This can be achieved with a PHP script that edits a JSON file containing the lists of songs. We would also like to add a more sophisticated file display system. Rather than show all of an account's content in a single list, the list will be split into three tabs. The first tab will display all of the music files, the second will display playlists, and the third will display transcriptions. This will be implemented with the jQuery UI tabs widget. Additionally, we would like to implement a way to display details about a calendar event within the page itself, rather than navigating to an external site. This can be achieved with AngularJS. Finally, the user will have the option of keeping individual songs hidden from the public page.

Long Term Features

If we have remaining time on the project and all of the above features have been successfully implemented, there are further components and behaviors we hope to add. The largest of these is the ability to control privacy rights and privileges for

Good additions here.

all content. Users will have the option of allowing only accounts they have approved to view all or a selected portion of their files. Finally, when a user selects a song, any **important** information or other files associated with that song would be displayed, such as transcriptions or comments the user has added.

References

AngularJS:

<https://angularjs.org/>

BOOTSTRAP:

<http://getbootstrap.com/>

ChromePHP

<https://github.com/ccampbell/chromephp>

DropzoneJS:

<http://www.dropzonejs.com/>

jQuery:

<http://jquery.com/>

jQuery UI:

<http://jqueryui.com/>

PHP:

<http://php.net/>