Process Plan Document V1.2

Good Vibes

1. Project Summary

1.1 Project Overview

The social networking service, Good Vibes, intends to connect musicians in a local area to create various kinds of performance groups and allow for posting of group events.

1.2 Project Scope

The project's scope is to allow musicians to create user accounts to advertise their skills and musical preferences for others to view. Independent musicians will be able to create profiles, similar to *Linked-In*, advertising their experience in various musical styles to music group directors and the directors can request these musicians to join their group. When accounts are created, the website will also provide a list/member directory of musicians that are also registered on the site. The member directory is in order of closest to the user in terms of location. There is a band directory which is also in order of closest to the user in terms of location. This is done by using the latitude and longitude coordinates of the users.

1.3 Development Process

Our group will follow the rapid prototyping. We will make a version of the site quickly and test it for bugs. We will fix those bugs and add more features to the next prototype.

2. Process Description

2.1 Project Lifecycle

The lifecycle for the project will implement the waterfall model. We developed the program piece by piece and performed testing as each portion is completed.

2.1.1 Reflection

Due to time constrictions, the amount of time that our group collectively put into the project was insufficient in terms of achieving our goals. Some of our desired functions were not implemented because either they were unnecessary or we didn't have time. We should have done automated testing throughout the lifecycle of the project instead of coding and adding features, then manually going through the site and testing different things. For what we would do the same, we would use PHP and mySQLi again since they're key aspects of web page development.

2.2 Process Activities

Activity	Description	Input	Output
Requirements	The requirements and details of the project are established	Brainstorming on necessary aspects of the project so the correct software is created	SRS
Analysis	Specifics to the criteria laid out by the requirements	Duration, project deliverables, project milestones	The process plan document
Design	How the project will actually will be implemented by code	Architecture, high-level design, and detailed design of the project	Design document
Implementation	The actual creation of the project	Simple, effective code	Testable and accurate code
'Testing'	Major troubleshooting and correction of bugs. Done throughout the project anyway	Time spent debugging and troubleshooting	A finished product

3. Roles

3.1 Team member names

Aukshunas, Max
Chen, David
Dester, Brandon
Shea, Liam

3.2 Roles Table

1. Project Manager	Understand the overall design of the software to ensure that the members working together and to communicate with the members about the development process, the project schedule, and the software requirements. Maintaining, revising, and ensuring that the team members adhere to the developed schedule.
2. Quality Assurance	Take the written code and perform test to confirm that the software meets the requirements specified.
3. Documentation Lead	Work with the developer lead in order to provide accurate documentation for the code so other people can quickly understand what the code is designed to do.
4. Development Lead	Write and understand the coding within the software. Will work hand to hand with the documentation lead.
5. UI Lead	Create a user interface that allows for the consumer to easily navigate throughout the software.
6. Requirements Lead	Understand what the software is required to do and to determine additional requirements that may contribute to the performance/quality of the software.

3.3 Roles Assignment Table

Team Member Name	**Role(s)**
Max Aukshunas	Development Lead, Requirements Lead, Project Manager
Liam Shea	Quality Assurance
David Chen	Documentation Lead
Brandon Dester	Development Lead, UI Lead, Project Manager

4. Estimates

4.1 Effort Estimate

- Effort hours: 6 hours a week per member, 4 group members, 10 weeks, 60 hours per person, 240 total team-members hours
- Lines of source code: ~ 1,000 LOC
- Defects: 10

4.2 Schedule

Task Name	Description	Start	End	Dependent task	Roles
Requirement	Determine what the software will be designed to do.	9/15	9/30		1, 3, 6
Coding and documentation	Implementation through PHP, CSS.	10/3	12/7	Learning the basics for PHP	1, 4, 6
Geolocation	Learn to use and incorporate geolocation into our software.	12/12	12/14		1, 4, 6
User Account System	Create a system that allow users to create accounts and profiles and store the required information within a database.	10/3	10/10		1, 4, 6
Group Systems	Allow users to create and organize groups	10/10	10/28		1, 4, 6
Continuous Testing	Manually review the code and confirm that the software functions to the written specifications.	10/3	12/7	Complete or partially complete code.	1, 2, 4, 6
Create/Edit UI	Create the program interface that the consumers will interact with. Implemented using CSS and HTML.	10/3	12/7	Design the website and various webpages in PHP	1, 4, 5, 6

Max Aukshunas
David Chen
Brandon Dester
Liam Shea

Software Design Specification V1.2

1. Introduction

Sl	Class	Principle Responsibility
1	createAccount	Create a new user profile
2	profile	Display/edit user/band data
3	messaging	Manage/send user messages
4	bands	Create/edit/add members to bands.

Good Vibes is the name of the software being developed. The goal of the functionality of this software is to allow users to access the browser based client service locally from any location on a computer in order to meet musicians. It will contain independent web pages that are generated for each user which will allow for information about the user to be displayed (including instruments played, experience, genre etc.). Users can look on the member and band directory to find members and bands closest to them. From there, the user may access other users' web pages to view information about them and message them or send a band request. Users can form bands and invite other users to the band by going to the member directory and going to their profile.

2. Design Considerations

With most the functionality of the site relying on CSS and PHP, one issue is that no one has experience coding in either. We need to resolve this by looking up tutorials and guides on how to code things like account systems, messaging systems and other operations. Another similar issue would actually be setting up the PHP web server and then finding out how to interact with the mySQL database. Another issue is that we need to learn how to code in HTML for the user interface. This is not a big issue as the user interface doesn't matter as much as the functionality does, but we will need to create web pages that allow users to actually use the site to its full potential.

2.1 Assumptions

We are assuming we are going to be able to write HTML code within the PHP built in web server. We are assuming the client for our product is just the web browser that all users must have access to. In order to use the website/software, all users must have internet access. The mySQL database must be online at all times in order for our PHP functions to access the data and return data to the web pages. We are assuming most users will be using the site from the desktop version of Google Chrome. Our UI will be custom made to that specific browser so it will most likely not look good on other browsers especially mobile.

2.2 Constraints

Development time is one of our biggest constraints. With little to no prior experience in server/web design, we are unable to determine the amount of resources required for the development of each aspect of our project.

2.3 System Environment

The website is functional in all web browsers. The website will be accessible from mobile browsers like the Google Chrome app but a mobile-friendly version of the site will not be created. We will use the mySQLi database to store all information pertaining to users, bands, profiles, messages, and requests. We will be using HTML, as well as some CSS to create the UI and javascript for the messaging system, with all the functionality of the site coded in PHP within that HTML. It will be coded using the PHP built in web server which would then interact with the mySQL database to get information and then display it within the web browser.

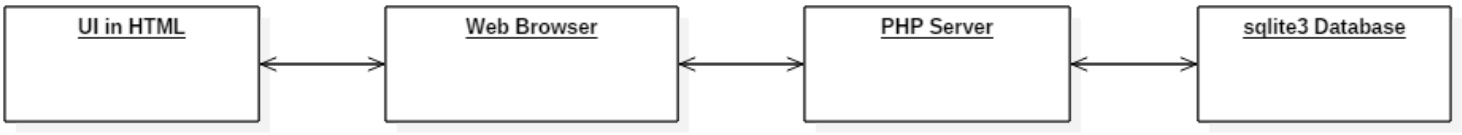
3. Architectural Design

3.1 Overview

Bandprofile.php - displays all band info like profile picture, bandleader, members, genre etc.
bandRequests.php - handles the sending and receiving requests to join bands
Bands.php - band directory in order of closest to user, allows users to click on band profile
createAccount.php - primarily html for the create account form
createAccount_script.php - adds account to database with all user info
createBand.php - adds band to database with band info
deleteAccount.php - deletes band from database
Editbandprofile.php - primarily html for editing band profile
Editbandprofile_script.php - changes values for band info in database
Editprofile.php - primarily html for editing user profile
Editprofile_script.php - changes values for user info in database
Header.php - header that is displayed on all pages that includes logo, menu and username/picture
Loginpage.php - home page that allows users to log in or create an account
Logout.php - allows users to log out of their account
Members.php - member directory in order of closest to user, allows users to click on member profile
My_messages.php - displays all unread/read messages from other users

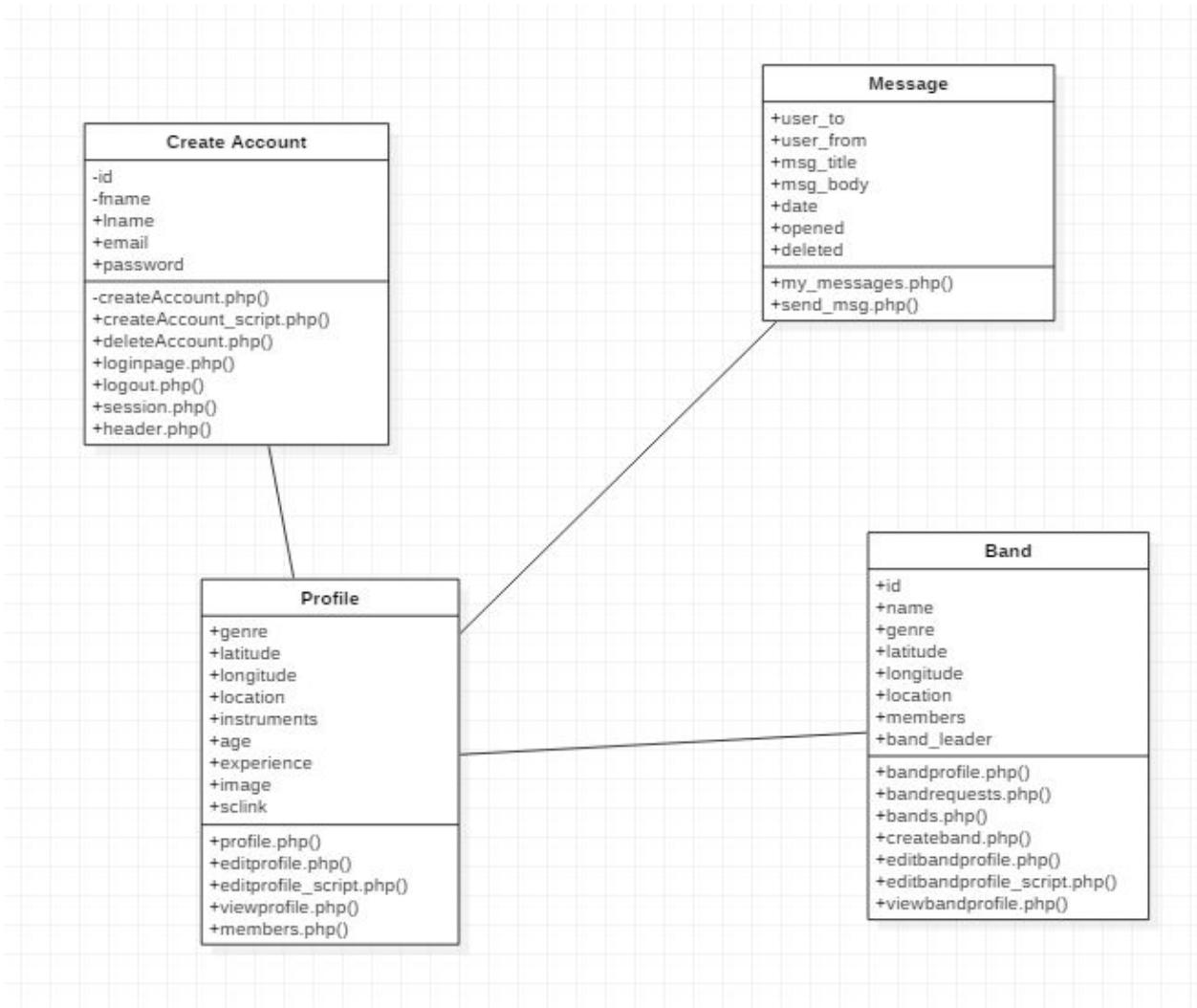
Profile.php - primarily html that displays the user's profile/info
Send_msg.php - allows users to send messages to other users as long as they're in database
Session.php - starts a session for a logged in user
Viewbandprofile.php - allows users to see a band's profile/info
Viewprofile.php - allows users to view other user's profiles

3.2 Conceptual (or Logical) View

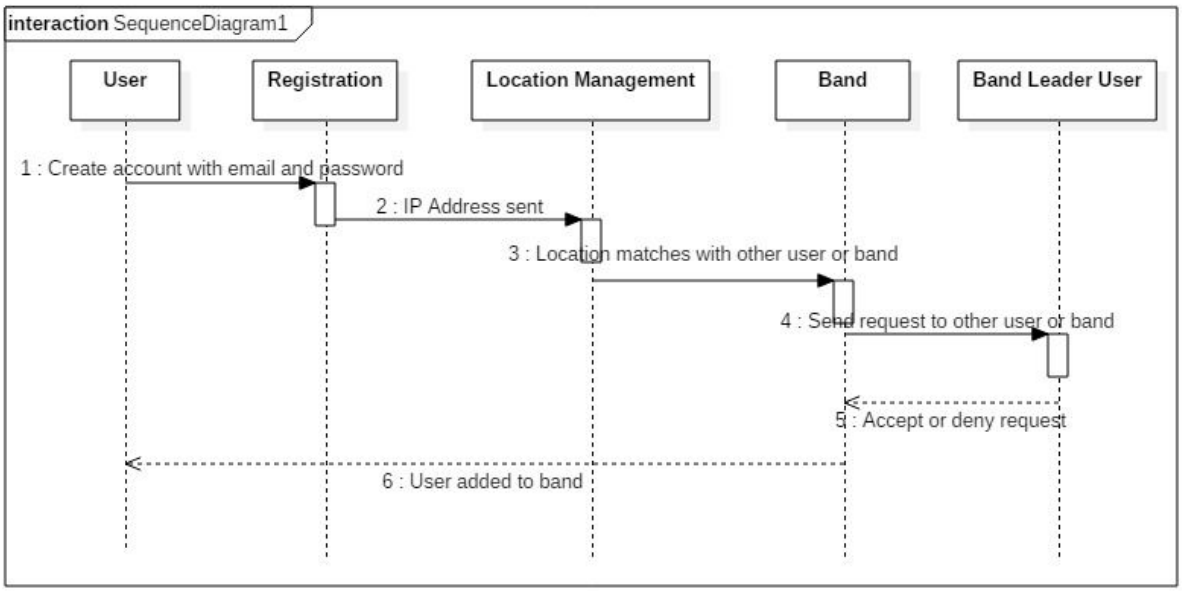


4. Low Level Design

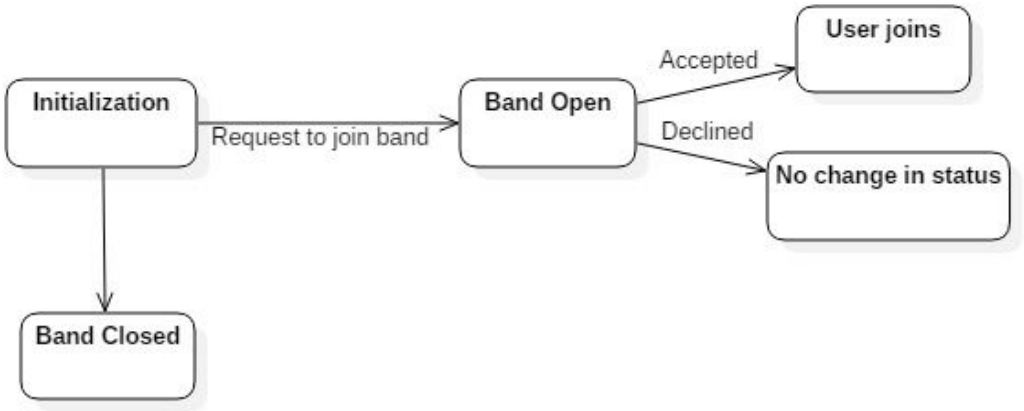
4.1 Class Diagram



4.2 Sequence Diagram



4.3 State Diagram



Illustrates the status of users when they attempt to join bands

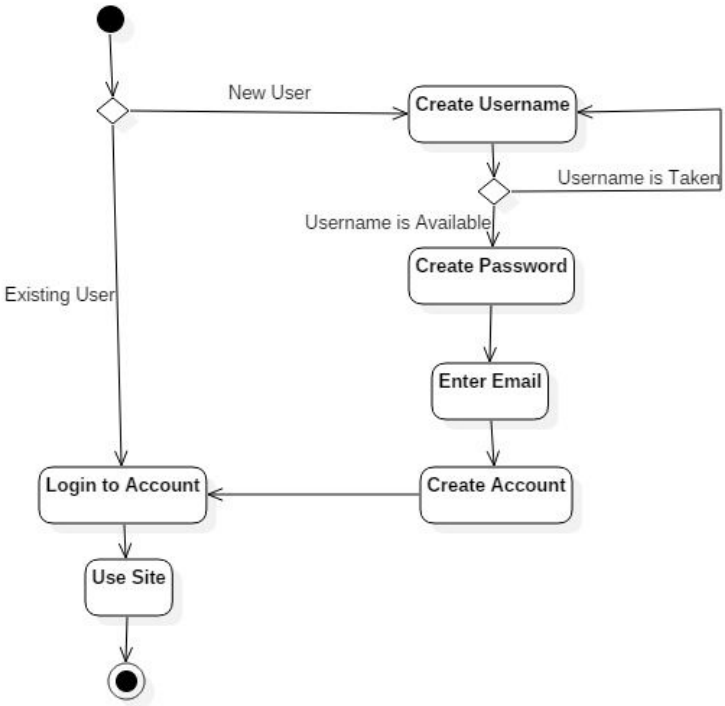


Diagram that describes the process of username availability and user account creation

5. User Interface Design

A user friendly system has been implemented. The first page the user will see is the home page, which is essentially just a login screen. The user can login or redirect to a create account page in order to gain access to logging in to the website. Once logged in, the user is redirected to their profile page which is unique and generated with the information the user provided when creating an account. On this and every page is a header which allows the user to navigate the website by clicking on the redirection links to reach certain pages and functionalities. These functionalities include an inbox, a band request query page, a view profile page, a member directory page, a band directory page, and a logout function.

From the user's profile page, accessed by clicking on the "View Profile" link, the user can edit their profile or deactivate their profile. Within the edit profile page, the user can change various information, including genres they're interested in, instruments they play, their age, as well as their profile picture, which will be displayed along with their name in the member directory and in the top right corner of all pages. The user's location may also be edited, which will automatically find their location if they just type in the city they wish to enter. A user may also enter a link to their SoundCloud account page, which will display an icon on their profile to redirect there for other users viewing their profile.

The member directory page displays all current users that have account on GoodVibes. They are ordered by location closest to the logged in user. The band directory page has a

similar function, displaying all bands that are in close proximity to the user. From these pages, the user can view other users profiles and band pages. The user profiles are similar to the own users profile, just without the ability to edit or deactivate the account. From the band directory page, users have access to creating their own bands. Once the user has created a band, the user may invite other users to this band by going to a user's profile within the member directory page, selecting the band they want them to join from a drop down menu, and clicking "Invite". The user is notified that their invite has been sent. Users may also send messages to other users from the user profiles by selecting "Send Message". The message composing page is simple, allowing the user to enter the title of a message and a body and send a message to the desired user by clicking "Send Message".

Band pages contain information about the bands that have been created. Only the band leader may edit the band profile page, which will be accessible for them by clicking on their own bands within the band directory page. The band pages display the band picture, the band leader, and band members, as well as the band's location and the and the genre of music the band plays.

The inbox functionality shows to the right of it how many messages the user currently has in their unread message inbox. Once clicked users may view messages that have been sent to them from other users. Within the inbox, the user may view the message, reply to the message, mark the message as read/unread (changing the count of messages in the inbox), and delete the messages.

Finally, the user may view the band requests that have been sent to them by selecting "Band Requests" in the header of the page. If there are invitations for the user, they will display, showing who sent the request and the name of the band. The user may either accept or decline the request.

Liam Shea
Brandon Dester
Max Aukshunas
David Chen

Software Requirements Specification V1.2
Good Vibes

1. Introduction

Good Vibes is a social networking service that provides a way for musicians to connect with each other to form local performance groups. The website will be the user interface for the musicians to perform actions such as making an account, joining a group, and posting samples of music. The website and all of its backend software will run on HTML, CSS, and PHP. The software will implement a database for storing various things like user accounts and information, messages, information for bands.

2. User Requirements

2.1 Product Perspective

Good Vibes is aimed towards musicians and people that like to listen to music. Good Vibes is similar to other social networking sites like Facebook or Linked-In and acts like a combination of the two. The primary focus is for musicians to be able to meet each other and form bands. The website will be easy to use yet helpful in finding other musicians. Good Vibes is a stand-alone website that doesn't require any software to be downloaded. It can be accessed by going to goodvibes.rocks. The website will be accessible from any browser and will also function on any mobile device that utilizes a browser, but a specialized app will not be available to access this software.

2.1.1 Software Interfaces

The software will have to interact with PhPMyAdmin to implement MySQLi so that it can store user account data. The software will also have to interact with the Internet/web browsers such as Google Chrome, Firefox or Safari which involves creating a network between the database, our back-end code, and the HTML/PHP coded website.

2.2 User Interfaces

The software will have a user friendly interface. It will provide the ability to create and edit user accounts that can describe the user in many ways with fields that the user can fill in, i.e. name, music taste, and instruments learned. The software will automatically recommend other users based on location via the member directory. The software will also automatically recommend bands based on location via the band directory. Users can edit all aspects of their profile including the link to their soundcloud profile. Once users enter the link to their soundcloud profile, our site automatically embeds a soundcloud button that once clicked, brings users to the specified soundcloud profile. Users can edit their location by entering the city that you're located in which then will add the coordinates of that city to the database so the location recommendations work properly.

Users will be able to form bands with a custom name and genre. They can then send invites to other users and also edit the band profile page. If users see a band they'd like to join, they can send a message to the band leader using our messaging system.

2.3 User Characteristics

The general users of the product are high school students and older. Beginning at high school, students would have sufficient experience in music and knowledge to participate in serious organized musical events as performers. As participants, the age of the users being 18+ would generally imply that the users have enough freedom to go participate in organized musical events. There will be minimal technical experience required as they will only be interacting with the UI of our website and the musical skill of the users may vary as well as participants of musical events do not need to perform to attend events.

2.4 Assumptions and Dependencies

All users must have connection to the Internet to use our software. We also need users to enter a valid city where either themselves or a band is located. Once that city is entered, our system will get the coordinates of that city in order to recommend other users closest to them. We also assume that people are creating accounts and trying to form groups within decently populated areas. If a user wants to form a group and there is barely anyone in the area, it wouldn't work well.

3. System Requirements

3.1 Functional Requirements

The software will need to be able to take the location that the user enters and find the latitude and longitude coordinates of that location. It will then need to use that location to show other musicians and bands that are nearby.

The software will need to be able to store user data including username, password, profile information, location. etc. Data will be saved through PHPMyAdmin and MySQL. A messaging system will be implemented that will allow any user to send a message to any other user. Messages will be stored in the database so users can refer to past messages later.

3.2 Non-Functional Requirements

Our software must be compatible with all major web browsers, i.e. Google Chrome, Mozilla Firefox, Internet Explorer, and Safari. Our software will be able to obtain maximum load or response time of 0.1 seconds. The software will stay stable more than 99% of the time. User information and messages will also need to be backed up on the network in order to avoid loss of information in case of a server crash. The users should be able to navigate through the site easily.

3.2.1 Software Quality Attributes

- The website itself will be very reliable in that if the user has connection to the Internet, they will be able to access our site and use it. If the users can't access the website, then they can't do anything.

- The website should be very simple and easy to use but also helpful at the same time. We don't want users to be confused on how to do certain things on the site. In the future, we can have users take surveys to find out how they feel about the user interface and overall functionality.

4 Reflections

For the end product, we were unable to implement the ability to match users through similar musical tastes, the ability to post listings and events, encryption, and automatically obtaining the user's location via ip address.

Liam Shea

Max Aukshunas

Brandon Dester

David Chen

Test Plan Document

Testing for Goodvibes

1)

The basic operations of this software are:

- The ability to create, edit, and delete user profiles
- The ability to log in and out.
- The ability to create, edit, and delete band pages
- The ability to add and remove members from the band
- The ability to share music via SoundCloud
- The ability to message other users and invite them to bands.

2)

Most of the testing was done via static inspection by the lead developer. This is not an ideal test plan but the code for Good Vibes isn't very conducive to automated testing. Ideally, individual units or objects in the code would be tested one at a time and then integrated together but the code in Good Vibes is not easily broken down into units. This leads to having to design large tests, effectively skipping unit testing and diving straight into component testing. This left the Good Vibes team between a rock and a hard place. On one hand, they did not have the expertise to design and execute these large tests but on the other hand, it would be a massive and risky undertaking to try to divide the already working code into smaller and more testable units.

3) The strategy for testing includes thinking of test cases involving inputs that may be limited by variable type or by size and seeing the outputs. Many of these include if and else statements that echo out success or error messages depending.

4) **Test Cases**

Test Case	**Purpose**	**Steps**	**Expected Results**
Database	To see if connection to database is successful when a dummy database location is entered	Put in a dummy database location. Pass this through the if else statement that echoes the result.	Fail
EditProfile	To check if the software will allow a value of incorrect data type to be inserted into the database	Enter an integer for a VARCHAR variable field ‘genre’. Pass this to the database.	Pass
Login	To check if users can log in when to values are entered into the fields	Leave username and password values empty for when $_POST is called and run it through the $required variable condition.	Pass
CreateAccount	Checks to see if a user can create an account if they do not enter values for one of the fields during the account creation process	Leave all values for create account empty and attempt to pass them to the database. Have the $required conditional check to see the error.	Pass

Brandon Dester
Max Aukshunas
Liam Shea
David Chen

User Documentation

Good Vibes

How to use our software:

The first step in using the Good Vibes website is to make an account. To do this, navigate to goodvibes.rocks in your internet browser. This will bring you to the login page. At the bottom of the page there will be the text “New User? Create Account”. Click the “Create Account”. This will bring you to the account creation page. Please fill in all the required fields and click the “submit” button. If the requested username is unavailable then please chose another one.

Now that your account is created, return to the login page and enter your newly chosen username and password into the correct text boxes and press “submit”. Now you will be directed to your profile page. Now that you’re on your profile page, let’s talk about navigating the rest of the website. Towards the top on the page there will be a six links. These are “View Profile”, “Member Directory”, “Band Directory”, “Inbox”, “Band Requests”, and “Log Out”. “View Profile” will bring you to your own profile page. “Member Directory” and “Band Directory” are lists of all the active users and bands respectively. The “Inbox” is where your messages from other users are stored and “Band Requests” are where requests from other users for you to join their band are stored. Lastly, “Log Out” is what you click to exit the website.

Most of the information fields on your profile page will start off blank. To solve this, scroll to the bottom of the page and click “Edit Profile” to edit your page. To edit the page, please enter all the required fields and click “Submit” if you are satisfied with the changes or “Cancel” if you are not.

Most of the fields on the edit profile page are self-explanatory. One possibly confusing field is “Soundcloud Profile Link”. Here, place a link to your Soundcloud profile (if you have one)

and then a link to your Soundcloud will be embedded in your profile page. Some other tricky fields are “Choose File” and “Upload Image”. “Choose File” is used to select an image you would like to use for a profile picture and “Upload Image” will set the image you have chosen as your profile picture.

Let’s move to the “Member Directory” page. This is simply a list of all the users on the website. Usernames can be clicked to take you to another user’s profile page.

While on another users profile page you can see all there information. Right below their profile picture there are 2 buttons and a dropdown menu. The first button “Send Message” is used to send that specific user a message. The drop down menu is used to select a band you are in charge of. Once a band is selected, you can press “Send Invite” to invite them to the band.

Next is the “Band Directory”. Here you can select to “Create a New Band” or you can view other band’s band pages. A band page consists of the name of the band at the top of the page and the list of band members underneath. The members’ names can be clicked to take you to their profile pages.

Next is your inbox. If there is a number besides zero next to your inbox that means you have unread messages. Unread messages are stored towards the top of the page while read messages are stored near the bottom. A single message has 4 buttons you can click. The first one is the subject of the message received. The second, “Reply”, is used to message back the person who messaged you. The third, “Mark As Read”, will move the message from the unread section to the read section and it will lower the number next to the “Inbox” link by one. The final box, “Delete”, will delete the message.

The penultimate link to click is the “Band Requests” page. Here is where you find all the requests to join other people’s bands. There will be a sentence saying who invited you to which band. There will be two buttons as well. The first is “Accept Request” and this lets you join the band. The second button, “Ignore Request” will reject the invitation to join.

The ultimate link to click is “Log Out”. Clicking this will bring you to the original login page.

Bugs:

Some bugs within our software are the location must be a valid city so that way our system can get the coordinates. Also, the coordinates for some cities can't be retrieved. Another bug is if a user doesn't enter their location, their coordinates are set to 0,0 which throws off the calculation for nearest users. Once you create a new band, an error message will pop up saying "Cannot modify header information - headers already sent by...". This is bug that we are having a lot of trouble fixing but it doesn't affect the functionality of creating a band. Another bug is in your inbox. You have to click mark as read twice. For band requests, you also have to hit accept or decline request twice in order for it to go through. You can also invite the same user to the same band multiple times.

Brandon Dester
Liam Shea
Max Aukshunas
David Chen

Developer Documentation
Good Vibes

Our code doesn't need to be compiled because it is just php files connected to a mySQL database. The database and all php files have been uploaded to our hosting service, HostWinds, and is live at http://www.goodvibes.rocks. Going to goodvibes.rocks redirects you to the home/login page which is loginpage.php.