Brandon Dester

Max Aukshunas

David Chen

Liam Shea

**Software Requirements Specification**

**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. It also allows for posting/inviting to events. The website will be the user interface for the musicians to perform actions such as making an account, posting a request for a group member, joining a group, posting samples of music and listing events. Also, non-musicians can visit the website as guests without account to search for musicians and musical events, but they will be limited in terms of interacting with other users. Some actions on the website that you won’t be able to do are update your status or like or comment on other users pages but users will be able to edit and customize their profiles as well as message each other. The back end of the software will be coded primarily in Java, with some in C++. The webpages will be coded in either HTML or PHP. The software will implement a database for storing various things like user accounts and information, messages, and local listings to find musicians.

**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, however, it will be accessible to non-musicians if they wish to discover bands or attend performances. The website will be easy to use yet helpful in finding other musicians or events to go to. Good Vibes is a stand-alone website that doesn’t require any software to be downloaded. The website will be accessible from any browser and also there will be a mobile friendly version of the website, but an app will not be available to access this software.

**2.1.1 Software Interfaces**

The software will have to interact with Amazon/Google Cloud Servers so that it can store account details and posts/listings. 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 predetermined fields the user can fill in, i.e. name, music taste, and instruments learned. The software will automatically recommend other users that share commonalities with the user based on music taste and location. Users may select from a variety of choices a maximum of five different genres they are interested in from drop down menus. Users may then specify below each choice any sub-genres they are interested in. This is to allow for recommendations based on music to be possible. On top of this, users will also have a free-form section where they can write anything of their choice, but the inputs in the free-form section will have no influence on the recommendation algorithm. User pages will also display “bands” that they are currently in, as well as manually input bands that the user may have been part of that is not currently recognized as a band by the website.

Users will be able to form “bands” which allows them to broadcast whether they are searching for new members or not and can include various information on the band such as events that they will be playing at, what positions in the band that are filled, and what positions they are currently looking for. There will be a text section that allows the band leader to post additional info as they see necessary.

The software also provides a basic messaging feature for users that do not want to create an account. Musician requests can be posted for anybody to view if a user has a show coming up that requires a musician to stand in for. Any user can send messages to the one who posted the request and they will be assigned a temporary name for a set period of time. The user who is posting a request for another musician will also indicate the duration of the listing. The listing duration may be readjusted by the user and after the request expires, it will be removed from the database to reduce memory consumption.

The users will also have the ability to contact the developers through a messaging system on the website that sends the text to a designated e-mail as well as any attachments, such as images or files for account recovery. Users may also “report” any user page on the website. Users must include why they are reporting the page, i.e. inappropriate content.

**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 are assuming that people are going to be accessing the website and creating their account from the location of which they want to form a group. Users will not be using a VPN or a proxy to access the site, otherwise the location system would not work. 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 access the IP addresses of users in order to approximate the location of the user. It will then need to use that location to map out other musicians nearby that are seeking and automatically recommend users to each other based on the location. SInce recommendations will also be based around music style and genre, an algorithm will need to be written to create a sequence of importance, with location being most important and the other factors being less important.

The software will need to be able to store user data, including username, password, profile information, and location. Data will be saved through the Amazon/Google Cloud Service. A friends list integrated with a messaging system will be implemented. 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. Encryption will need to be implemented to protect user account information. User information and messages will also need to backed up on the network in order to avoid loss of information in case of a server crash. All listings will also need to be backed up until they expire. 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 check up on their listings, look for events or do anything else.
* The actual system for finding similar band mates should be pretty accurate. We want musicians to be able to find good matches for their potential groups. We will implement a feedback system where we measure the amount of potential matches someone gets versus how many they actually think would be a good fit for their group.
* 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. We can have users take surveys to find out how they feel about the user interface.
* We will encrypt every one of the user accounts that gets created so that if there was a data leak, no one would be able to see everyone’s username and password. This is extremely important because we only want the actual account owners accessing their own accounts. When a user logs into their account, we will document their IP/location and compare to what their IP/location was when they created their account.