2017

GigMe

TECHNICAL SPECIFICATION

JORDAN MULVANEY & EMER HENNBRY

Table of Contents

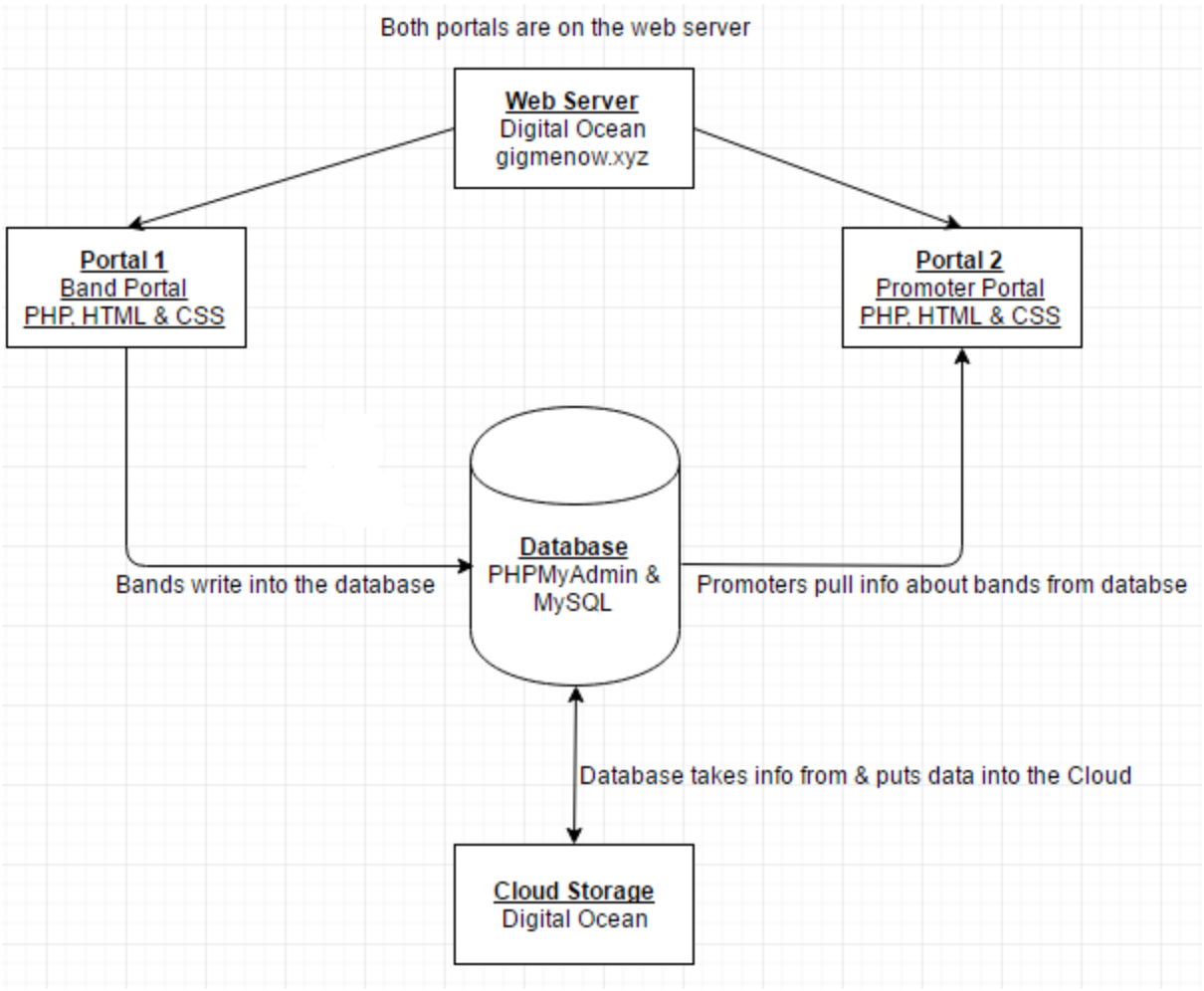
Introduction

We have developed a band booking system called GigMe. We have made it as easy as possible for bands to get themselves out there by making them more accessible to promoters. We have also made the site very straightforward for promoters to book bands. The web application is now live at gigmenow.xyz.
Our system has two portals, a band portal and a promoter portal. The band side of the site is where the bands information is stored and then displayed on their unique profile page. In registration we ask users to tell us all about their band, add their social media information, their Gmail and to upload photos to our server. Users should take as much time as possible in order to make their profile attractive to potential bookers. Each bands page displays their photos in a slideshow, buttons with links to their social media accounts, their Google calendar and a link to their about section which has all their information in one place.
The promoter side shows their Google calendar along with a nice search function that allows them to search and attempt to book bands. The search function lets a user specify the type of band they need by choosing the desired age, genre and location. Upon finding a band that suits their needs, they can click the “BOOK NOW” button, this brings a promoter to a page where they can outline the details about the event they want to book them for. After clicking submit, an email is sent directly to the band specifying all the details about the event. If a band chooses to reply to the promoter, a shortcut link is in the email which allows them to respond quickly. A link is also provided within the email that redirects the band to their google calendar and gives them the option to save the event.

System Architecture

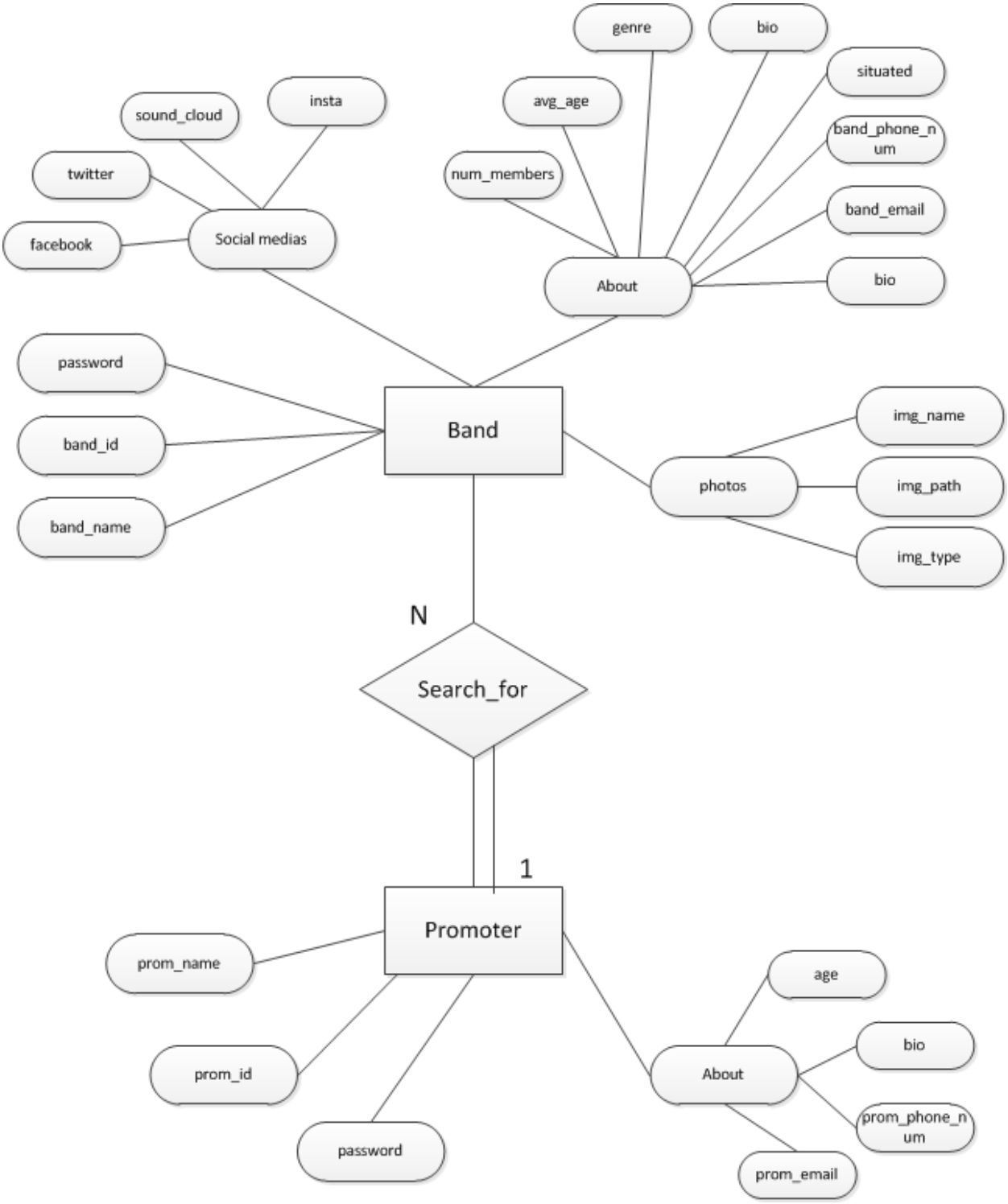
Connected System

As you can see, the system architecture within our system, is simple. We store all our data and files on a cloud server provided by Digital Ocean and use WinSCP to access and edit them. Digital Oceans then communicates with our SQL and PHPMyAdmin. The database reads information from our cloud storage to display in our portals but it also updates or adds new information to it. Our database now connects to our portals in two very different ways. When a user registers, it writes information to the database which then gets added into the cloud. The user can also update information that they have previously provided. A promoter interacts with the database in the exact same way. However, promoters also have access to the entire database of bands. They can search for a specific type of band and their information pulled from the database and displayed. Both portal user interfaces are designed using HTML and CSS. As well as these languages, we also used PHP to connect our user interfaces to the database. This enabled our web pages to read information from the database and display it and also to write to the database when registering a user or updating information. Both portals are accessed through the web server, we used Digital Ocean to host our cloud server and GoDaddy.com to buy an appropriate domain name, gigmenow.xyz.

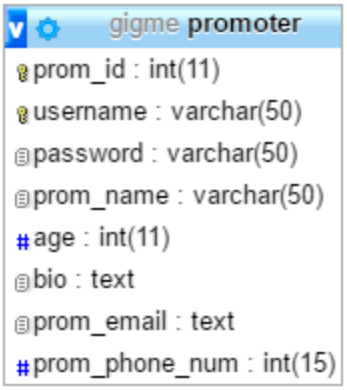


ER-Model & Database structure

Below is the ER-model for our Database and the structure of our Database. The band and promoter sides are clearly laid out in two different tables. We have one relationship involved, where one promoter searches for N number of band.


genre
bio
situated
avg_age
band_phone_num
num_members
band_email
bio
About
insta
sound_cloud
twitter
facebook
Social medias
password
band_id
band_name
Band
img_name
photos
img_path
img_type
N
Search_for
1
Promoter
prom_name
prom_id
password
age
bio
About
prom_phone_num
prom_email

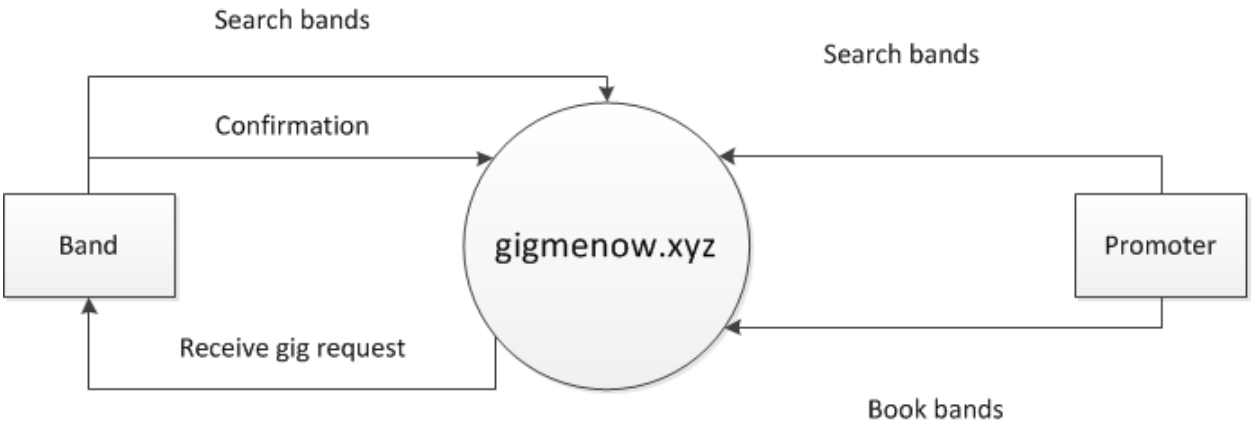

gigme bands
band_id : int(11)
username : varchar(50)
password : varchar(50)
band_name : varchar(50)
avg_age : varchar(50)
num_members : int(11)
genre : varchar(40)
situated : varchar(50)
band_phone_num : int(50)
band_email : varchar(100)
bio : text
facebook : text
twitter : text
sound_cloud : text
insta : text
img_name1 : varchar(300)
img_path1 : varchar(300)
img_type1 : varchar(100)
img_name2 : varchar(300)
img_path2 : varchar(300)
img_type2 : varchar(100)
img_name3 : varchar(300)
img_path3 : varchar(300)
img_type3 : varchar(100)
img_name4 : varchar(300)
img_path4 : varchar(300)
img_type4 : varchar(100)


gigme promoter
prom_id : int(11)
username : varchar(50)
password : varchar(50)
prom_name : varchar(50)
age : int(11)
bio : text
prom_email : text
prom_phone_num : int(15)

High-Level Design

Context Diagram

This is a context diagram. It doesn't show the process but shows the interface and the boundaries of gigmenow.xyz.



Band Data Flow Diagram

Step 1: Register
The bands will set up an account using their band name, username and password. Then they will fill in more details about their band, including email address and photos to complete their band profile.

Step 2: Log In
Log In using username and password.

Step 3: Edit Profile Page
Edit profile page to update information and photos.

Step 4: Search Bands
Search and look at other bands profiles.

Step 5: Accept/Reject Gigs
Respond to emails from promoters and add the gig to google calendar.

Promoter Data Flow Diagram

Step 1: Register/Set Up:
Set up an account using their name, username and password. Then a few more boxes about themselves.

Step 2: Log In
Log In using email address and password.

Step 3: Edit Profile Page
Edit profile page to update information.

Step 4: Search bands
Search for bands with their band name which is a fuzzy search, the bands age, genre and/or the location the band is situated.

Step 5: View Band Profiles
With the search function can explore a number of bands pages.

Step 6: Book Bands
Fill in an email form with gig details to get in contact with bands.

Step 7: Confirmation
Bands we receive feedback from the bands through email.

Problems and Resolutions

AWS: One of the big problems we were faced with was finding a cloud server that suited our site. AWS (Amazon Web Services) was recommended to us from the very beginning. After setting up our account and spending ten days trying to configure it to work online we gave up and decided to look for a new server. We had heard of other groups using Digital Ocean so looked into it, we soon realized that Digital Ocean is partnered with Gitlab's student pack offering a $50 coupon to use their services. After looking into Digital Ocean's services we realised the large community that use DO and help other users. With countless in detail tutorials on how to create each aspect of our server, we jumped at the opportunity and created our cloud server.

Calendar: Initially, we planned on gaining access to a user's google calendar through their Gmail and have it displayed on their profile for other users to see. When it came to implementing this we found that a site must have high security and required authorisation from Google. We realised this too late and felt that it would be more beneficial to the project to focus on testing other aspects of the site rather than waste our time trying to figure out something we weren't sure was possible. Instead we found that a user's calendar would be made visible to other users if they made their calendar public. When a user registers, we make this known to them and provide a link which clearly outlines how to do so.

Booking: Another problem that we became aware of too late was our booking system. We planned on when a promoter submits a booking request to a band that upon the band confirming, the promoter would receive a confirmation email and the event would instantly be saved within both their google calendars. This proved quite tricky. We were able to send an email to the band with details of the event but unable to format the email the way we had planned. Instead, we provide a link which allows the band to quickly respond to the booker either accepting, declining or negotiating the terms of the gig. We also implemented a link that shortcuts to the bands Google calendar and upon clicking save, the event is saved within their calendar.

Gmail: Our final problem was that sending an email to book a band always send to spam, this is because we send the email from our Digital Ocean server, change the name of the sender and send it. Gmail recognises that we change the name of the sender so the emails goes into spam.

Installation guide

A piece of software that we installed was a local WAMP server. This is a Windows version of a LAMP server. WAMP stands for Windows, Apache, MySQL and PHP and it allows all these different languages and servers to work as one to create a web application. We used wampserver.com and installed the X64 bit version of it. We used this locally while our cloud server was being set up. It allowed us to gain a good understanding of how to use PHP alongside HTML and CSS. We also used it for the majority of our testing, we were able to break our code down piece by piece and not worry about effecting our live version.

To set up server we used Digital Ocean. The first step we took was that we downloaded and installed PuTTY and PuTTYgen, which is an open-source Telnet and SSH client for Windows. It is possible to manage a server with just using username and password based logins, however, we felt it was a better for security reasons to set up and utilize SSH key pairs. With PuTTYgen we were able to generate the key pair. Next we made a Droplet, which is the server, on the Digital Ocean website. While we were setting up the droplet we were able to upload our public key so that it will be embedded in our server. Also when creating our droplet we choose Ubuntu and we choose 512MB of RAM with 1 CPU and 20GB of SSD storage. After the droplet was made we were able to log into our server using PuTTY using our private key we generated so we were able to log into our server without using a password but still being very secure. The next big step we took was Installing Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu using PuTTY. This step was necessary to serve our content on the internet. Next we bought a domain name on GoDaddy and we set it up to our IP address on Digital Ocean. The domain name we bought is called “gigmenow.xyz”. You don’t need a domain name but we thought it would be more professional. We tested to make sure our domain name was set up with our ip address by pinging our domain name on our command prompt. Then we installed phpmyadmin on your server. This is a free web software to work with MySQL. We were able to test if this was running by typing www.gigmenow/phpmyadmin/index.php. We then connected our server to the front end of the project by adding some of the pages of code to our server using WinSCP and our SSH private key.

To access our web application you will need an internet connection and a web browser.

References

- www.w3schools.com
- http://help.typepad.com/link-to-an-email-address.html
- http://useroffline.blogspot.ie/2009/06/making-google-calendar-link.html
- www.css-tricks.com
- http://www.broculos.net/2008/03/how-to-make-simple-html-template-engine.html#.WMHnIDuLTIW
- www.myhtmltutorials.com
- www.php.net
- www.tutorialspoint.com
- www.digitalocean.com
- https://dev.mysql.com/doc/refman/5.7/en/string-functions.html
- https://www.youtube.com/watch?v=PBLuP2JZcEg
- https://www.youtube.com/watch?v=e8TP2FERKls&t=1857s
- https://www.youtube.com/watch?v=HdIo6-PWdkw&t=1227s
- https://www.youtube.com/watch?v=t68aj7M3mPQ
- https://www.youtube.com/watch?v=yinO3mjJLyk&t=14s
- https://www.youtube.com/watch?v=IYmJeri6r0Y&t=999s
- https://www.youtube.com/watch?v=dMr66xhj42I