

**System Overview**

Project Name: **Wishbone**

**Prepared for:** Daniel Dogbey, Wishbone

**Prepared by Axiom:**

**Team Leader:** Andrew Archibald

Zeyang Hu, Ksenia Lopukhina, Svetlana Netchaeva, Oksana Shapoval, Minyi Yang

**Algonquin College Professor:** Patricia Murphy

2019-04-04

# Revision History

|  |  |  |
| --- | --- | --- |
| **Description** | **Author** | **Date** |
| Discussion of report contents during meeting | All members | March 30th |
| Initial Edit from meeting published | Andrew | March 31st |
| Edit/Review of content | All members | April 1st |
| Additional Information added to document directly or emailed to Andrew/Team email | All members | April 1-2nd |
| Merge information | All members | April 2nd |
| Complete Document – Review | Andrew, Svetlana | April 3rd |

Table of Contents

[Revision History iii](#_Toc5239512)

[System Overview 1](#_Toc5239513)

[Client and Project Information 1](#_Toc5239514)

[System Request 1](#_Toc5239515)

[Purpose of the Project 1](#_Toc5239516)

[Scope of the Project 2](#_Toc5239517)

[End User of the Product 2](#_Toc5239518)

[Sustainability 2](#_Toc5239519)

[Completion Date 3](#_Toc5239520)

[Data Design 5](#_Toc5239521)

[Architectural Design 8](#_Toc5239522)

[Error handling architectural design 8](#_Toc5239523)

[MVC pattern Architecture 9](#_Toc5239524)

[Deployment Plan 8](#_Toc5239525)

[How to deploy 8](#_Toc5239526)

[Program Design 13](#_Toc5239527)

[Log in 17](#_Toc5239528)

[My Page (User Home) 17](#_Toc5239529)

[MyNetwork 18](#_Toc5239530)

[Messages 19](#_Toc5239531)

[My Profile 20](#_Toc5239532)

[Search 21](#_Toc5239533)

[Conclusions 21](#_Toc5239534)

[Appendices 23](#_Toc5239535)

[Appendix A – Schedule Gantt Chart 23](#_Toc5239536)

[Appendix B – Source Code 24](#_Toc5239537)

# System Overview

## Client and Project Information

“You can think of Wishbone as the high-tech equivalent of going to a traditional talent event where you go and meet other entertainers and creative [people] in person, talk a little bit about what you do and exchange numbers and instagrams.” “[…] [O]n Wishbone, you add connections similar[ly] to how you’d make a friend request on Facebook, you converse via private message (or available contact information) and you have all your professional experience and achievements laid out in a neatly organized profile to show off to other users.”

Wishbone was started by a group who worked together and found that services like The Special Guest App do not operate in Canada. Wishbone wants to compete with sites like GigSalad. Their goal is to provide a similar service connecting Event planners, Artists, and Venues to each other in the Ottawa area and expand to Canada more broadly. They would also like their platform to connect artists together to share and communicate with each other, acting as a professional social media and collaboration platform.

## System Request

The goal of the system is to allow people to connect with each other to plan events. Examples of this might range from a local small business that seeks to hire a local talented artist to play music during dinner. It could be a club looking for a DJ or a party looking for an entertainer. Perhaps a group of artists would see value in booking a venue and organizing their own art show. A theatre is looking for actors and actresses. Or it could be a large-scale event, where a band or performer is looking for additional local talent to lead in to their show.

We see the project as having two major components. The social networking and the event planning ‘Gigs’. The front end of the system is partially completed, primarily for information about the site, so likely most pages for functionality will need to be produced. The entire back-end needs to be produced. We have two proposed designs with preference for a layered design described later.

## Purpose of the Project

The purpose of the project is to build a web site where artists can collaborate and that can facilitate event planning. Our initial plan for the system is to develop some social connection features.

## Scope of the Project

Due to limitation of experience, resource, and time, this project will be mainly focusing on developing the following key features for the social media side:

* Use their given *Home* page which is the main page when user visits Wishbone website,
* designing *User Profile* page which contains user’s personal information,
* creating *My Network* page which shows all the social connections which will include user interests for connecting to other users.,
* developing a *Messenger* chatting tool,
* generating *Notification* page which is used for notifying invitations, posts, and other interests – Also used as login landing page,
* *User Search* function related to *My Network* page,
* The ability to log on using Facebook,
* The ability to invite users to the platform.

## End User of the Product

A target audience is defined by the following criteria:

* age (young adults, primarily 17-25, extending to 35)
* interests (artists, venues, entrepreneurs, and event planners)

## Sustainability

Social media is a competitive space with a lot of entrenched and wealthy players. Our major goals in this project were to keep the costs low to the client, using free hosting limits the potential user-base, but increases the sustainability of the project as it is very low/no overhead. We also try to have a flexible data model. We believe our database is compatible with the clients goals for the next phase of the project, so we believe our data model is sustainable.

## Completion Date

Based on the analysis of the client requirements, we believe we can meet the target deadline and complete the social media side of the platform by April 11, 2019.

# Data Design

This design shows the overall data structure for the Wishbone project. This is the current implementation of the Database. Authentication is separated from the User table. Authentication uses AuthID since the authentication table is more exposed as it is the first table accessed, if information is taken it doesn’t contain user ID’s.

Our goal with the Database was to try to match business logic so that changes to the database could be done without being too fragile. If the database is built to implementation it seems likely it will be more difficult to modify for dramatic changes in the future. The long term plan for the system is to include the ability to set up events and connect to artists on the platform.

The client specified that a user may have multiple addresses as many artists work out of multiple cities. We currently only use 1 address but want to enable the database to support the possibility of multiple addresses (address table) and contacts(contact table). It’s possible that a user could be a business supporting/managing multiple artists (artists id), and it’s possible for an artist to have multiple artforms (artprofile). For example, a person could be a guitarist and a drummer, or Painter, etc. Currently they have 1 profile, and 1 artist per user, but splitting the database seems more consistent with business logic. So the database is built more to business logic, rather than tied to implementation. Experience is intended to be connected/related to the planning side of the future implementation, where events that are hosted can populate or count as experience.

The database is running on MySQL Version: 8.0.14.

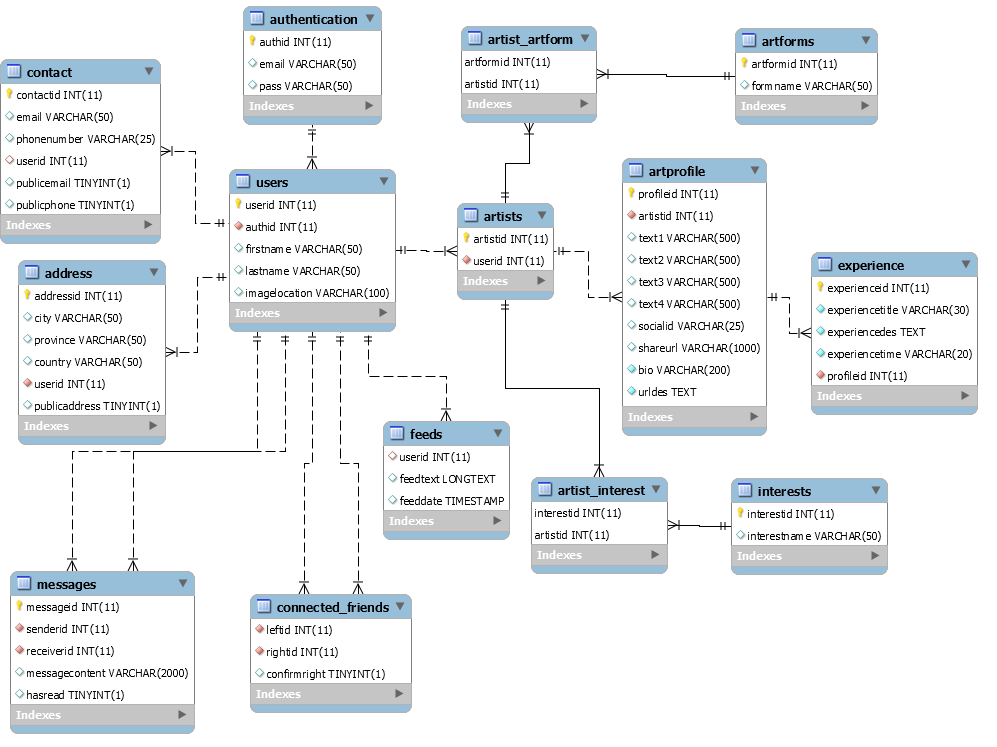


Figure 1. Wishbone ERD.

# Architectural Design

The system is a web site running on PHP with a MySQL database. The system has 2 physical layers, the system is running with the business logic running on the PHP server on the front end. This is because we can deploy it to a free hosting service that supports PHP and MySQL. The layout is a frontend/backend 2 tier layout. Php 7.2.14, MySQL Version: 8.0.14.

This project is new. So, we worked on both the Database and the Front-End.

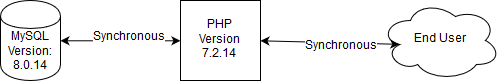


Figure 2. High Level Architecture.

## Error handling architectural design

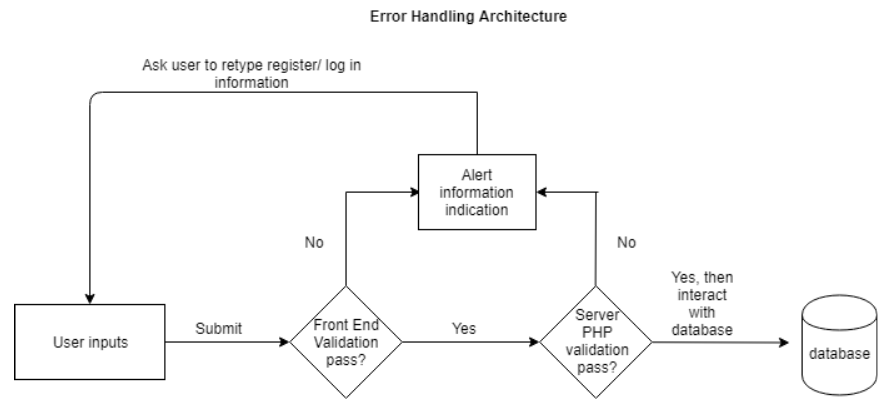
This error handling architecture design was created to serve the purpose of validating new users when registering and logging in. This design is separate into two parts. One is the front-end validation part which is to ensure that all required parts are filled and followed the pattern as needed. The second part is server side validation part which serves the purpose to make sure that all information inputs are in valid format and filter out unnecessary special characters, which could cause the potential security leak. Input data has to pass these two validation procedures to interact with the MySQL database. Otherwise, notice will be shown and indicate that user didn’t pass the validation and need to recheck what error caused it. 

Figure Error Handling

## MVC pattern Architecture

In our design, we mainly used the DAO pattern structure as our way of building up this application. Every piece of information submitted from the front-end layer will firstly be converted as an object model and then sent to DAO php class. The DAO class is responsible to connect with the MySQL database and invoke some basic create, read, update, and delete functions. MySQL database will retrieve these commands and perform corresponding tasks.

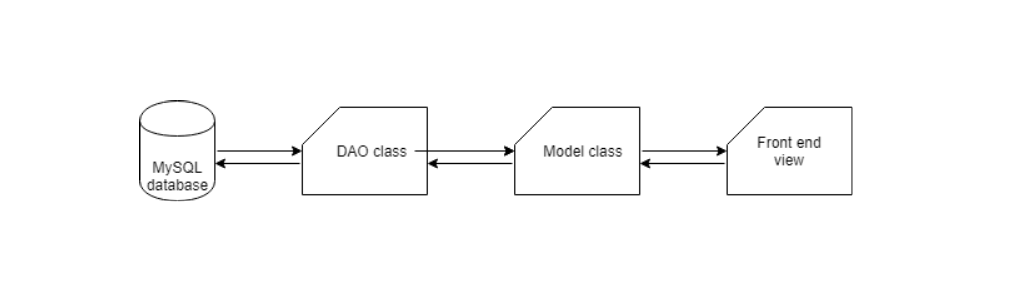


Figure Data Flow

# Deployment Plan

We also will deploy it to a free web hosting service infinityfree.com. This site will host a php site and a mysql database. To do so they will requires the zipped project, an ftp program like filezilla, and an infinityfree account.

Our deployment plan is to set the infinity account to a newly created gmail account, deploy the service for the client, then to hand over the Infinityfree account and the gmail account. This way requires no setup for the client. We will also include the zip file and setup information for the client so they have instructions for the future.

## How to deploy

After you setup the infinityfree account its easy to request a sub-domain. Then it autogenerates a username and a password. The FTP screen show also shows the ftp URL.

To setup the server you need the automatically generated username and password. This is used for the FTP and MySQL.

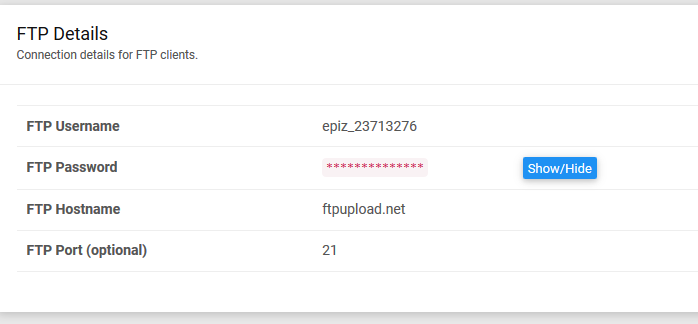


Figure FTP information

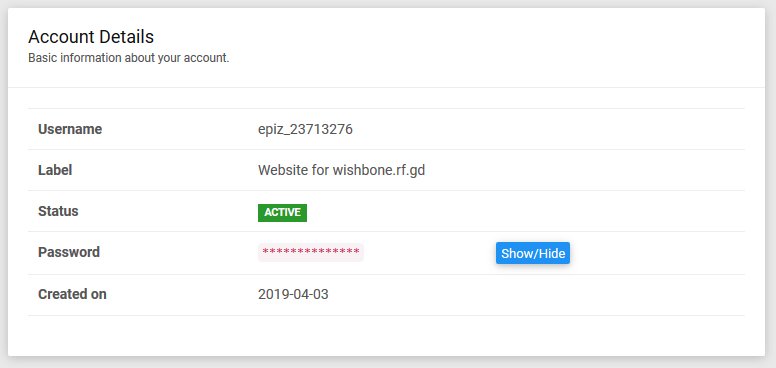


Figure Account info

With FileZilla you can connect and upload the project. Note that inside the DAO folder, the AbstractDAO must be updated with the information for the MySQL database created later.

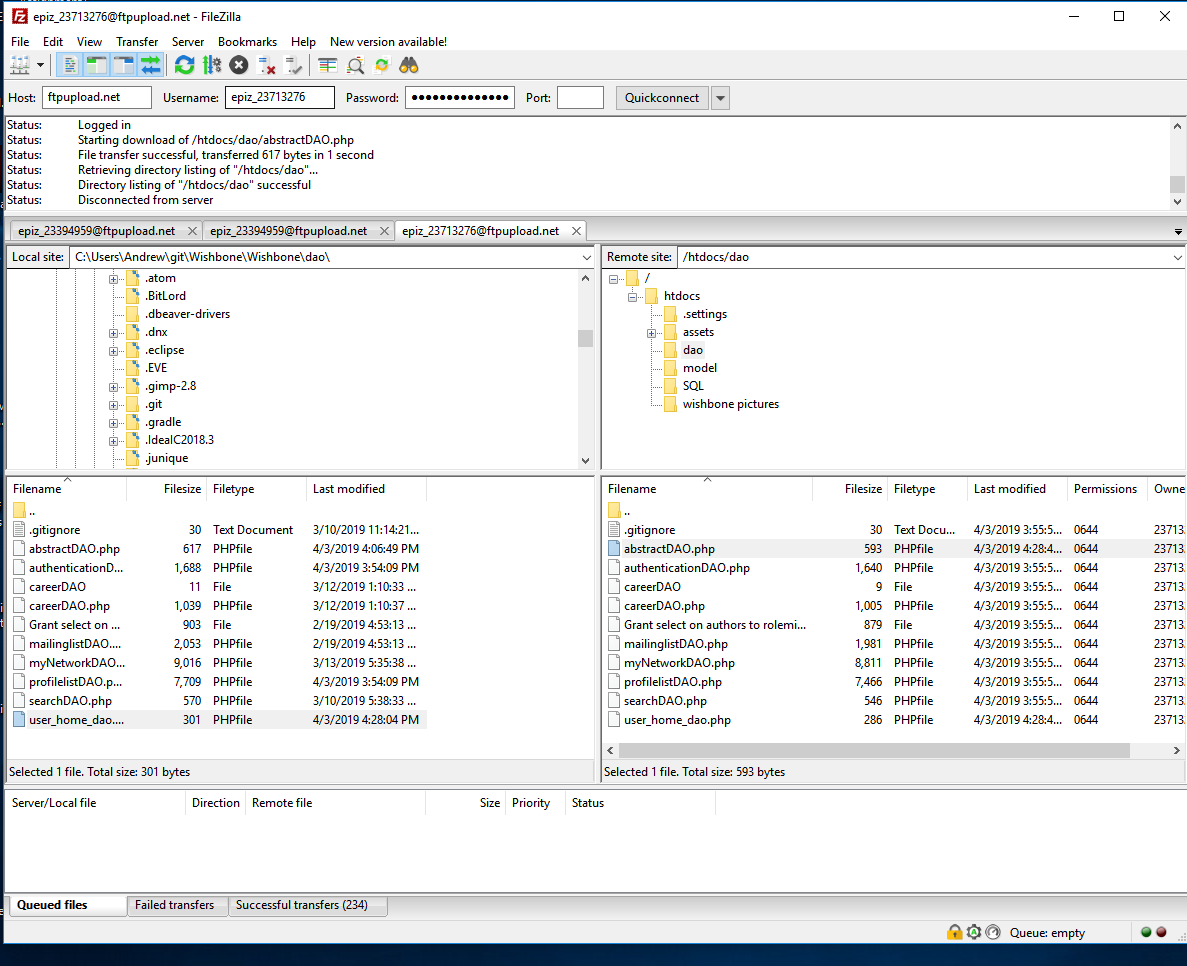


Figure Transferring by FTP

When you go back to your accounts you can go to the control panel.

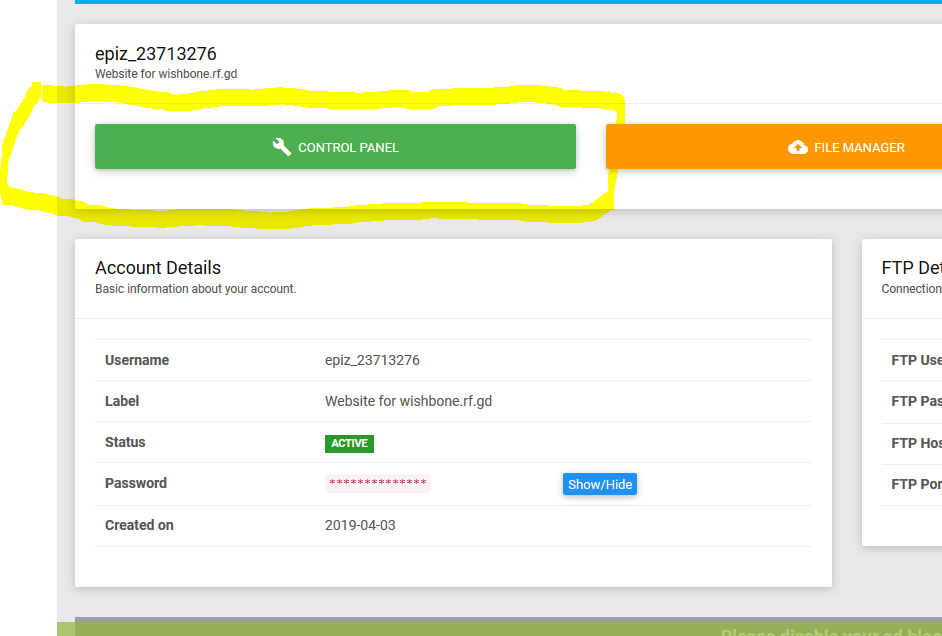


Figure Navigating to control panel

The control panel has a MySQL button.

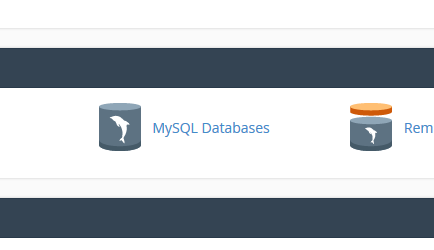


Figure SQL button

Which will allow you to create a new database.

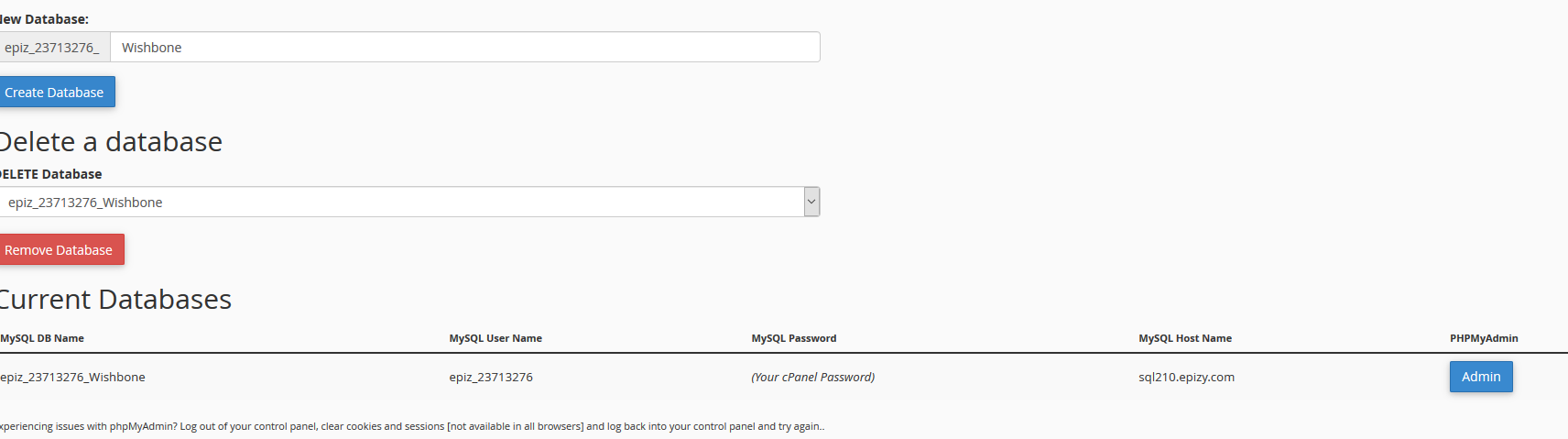


Figure Creating a new Database on InfinityFree

It’s difficult to read, but the created database has the SQL database full schema name, URL, and UserName. Premium is required for additional SQL accounts. Once the database is created you can click admin, which brings you to a phpmyadmin console. From there you can use the import tab to import SQL.

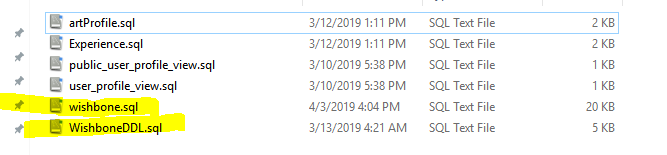


Figure Loading SQL to initialize DB

Inside the project the Wishbone DDL is critical and must be run. Wishbone.sql has the test/sample data.

Note that you can see the database, but the PHP side may require some time to connect to the SQL. The SQL database cannot be connected to from outside, only the PHP inside the service, but it takes time for the database to become visible to the PHP.

We plan to give the instructions to the Client on how to setup the database and website, but Primarily we will simply hand over the account usernames and passwords so they don’t have to.

# Program Design

The Database is running MySQL. The front end is in PHP, with HTML and JavaScript. The Table I lists functional requirements and identifies associated use case and sequence diagrams.

*Table I.* Cross Reference Table.

| Functional  Requirement  Number | Functional  Requirement  Narrative | Use  Cases | Sequence  Diagrams | Design |
| --- | --- | --- | --- | --- |
| SWF1 | provide artists and entertainers a method of  internally communicating with each other. (Messaging function) |  | Figure 14 | Figure 18 |
| SWF2 | provide a “My Network” function which allow  artist to create social networks with each other. |  | Figure 13 | Figure 17 |
| SWF3 | allow a user to view other users’ profiles in their network |  |  | Figure 19 |
| SWF4 | allow user to create their personal profile |  |  | Figure 15 |
| SWF5 | allow user to invite another person to the platform. |  |  |  |
| SWF6 | allow user to connect to/find other artists on the platform |  |  | Figure 20 |
| SWF7 | show updates or relevant new information to a user when they log in – User Feed |  |  | Figure 16 |
| SWF8 | allow user to log in using other social media credentials. |  |  |  |

We have two different types of diagrams used, a Use Case Diagram and a Sequence Diagram. The Use Case Diagram shows the basic interactions of the User with the system more of a front-end logical diagram, while the Sequence Diagram show the basic process flow of the chat system.

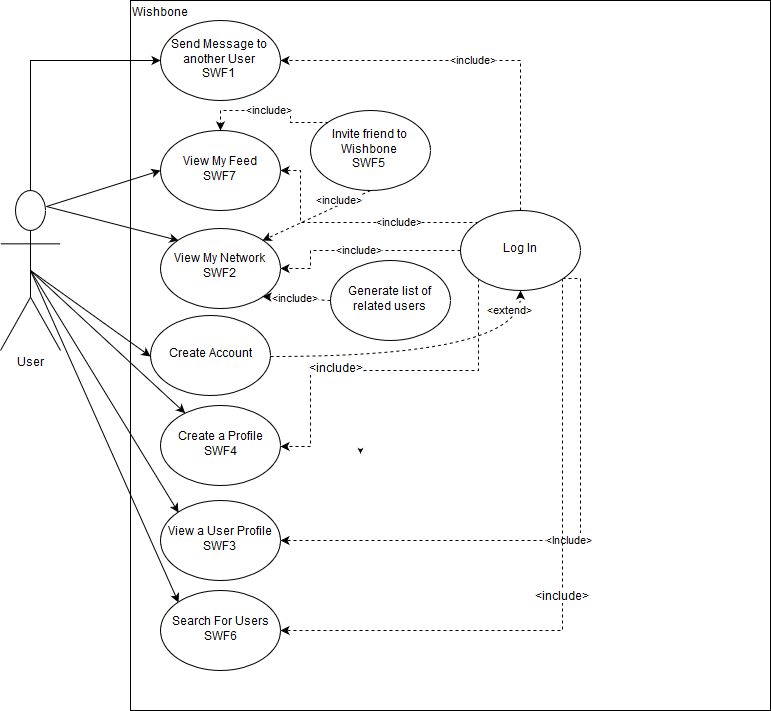


Figure High Level Use Case Diagram

This use case Diagram shows the high level use case diagram for the major components of the Wishbone system.

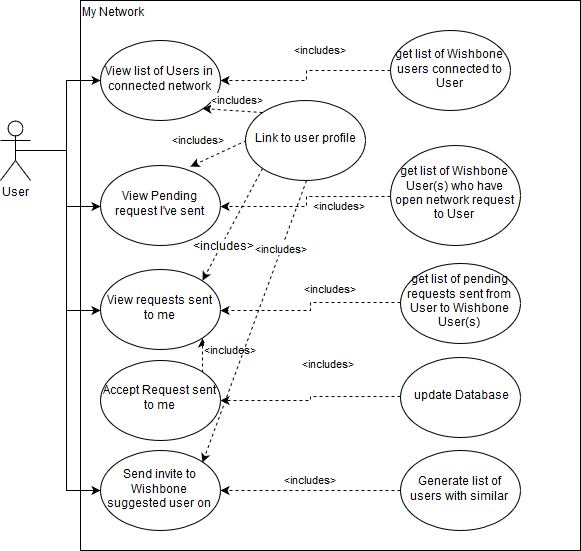


Figure 13 MyNetwork Use Case Diagram

MyNetwork use case shows the use cases for users to connect to other users on the platform, allowing them to accept and offer requests to join their network, or view profiles and send messages.

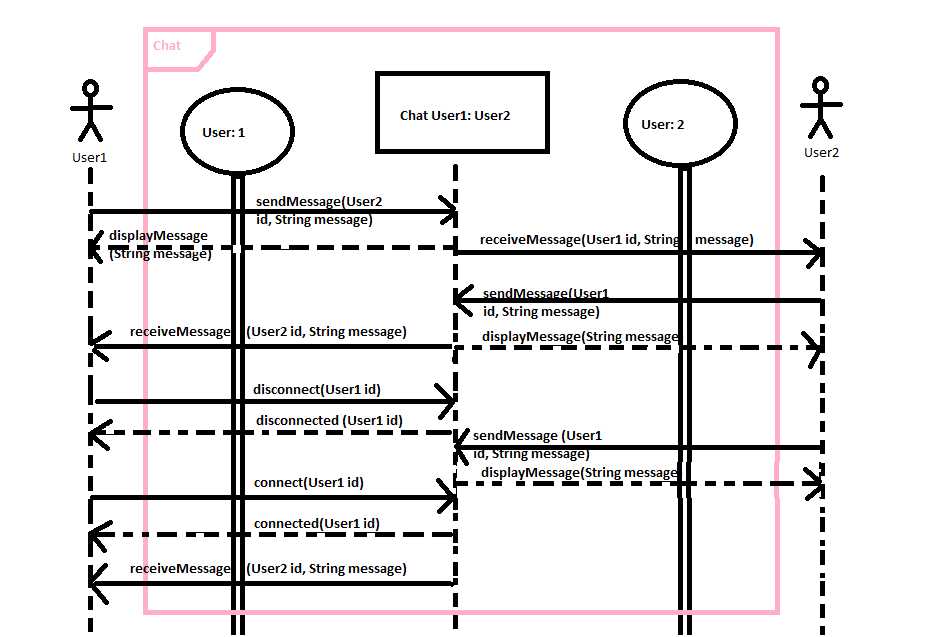


Figure Chat Sequence Diagram

This diagram shows the sequence of communication between users in when chatting. The initial user sends a message to the system, the system then updates the other user. That user can then also send message(s) back to user 1, and vice versa back and forth.

## Log in

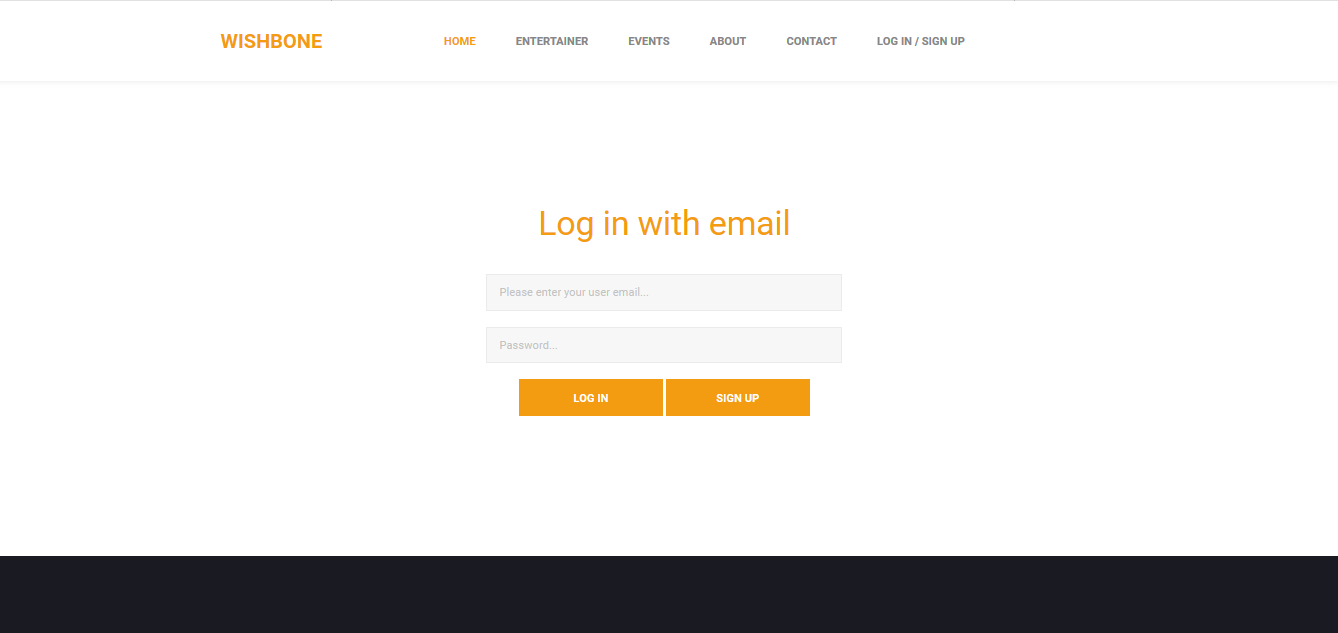


Figure Login Page

The log in page also has the sign up page. Users enter their username and password to sign in.

Login was done by Zeyang, referenced in Appendix B

## My Page (User Home)

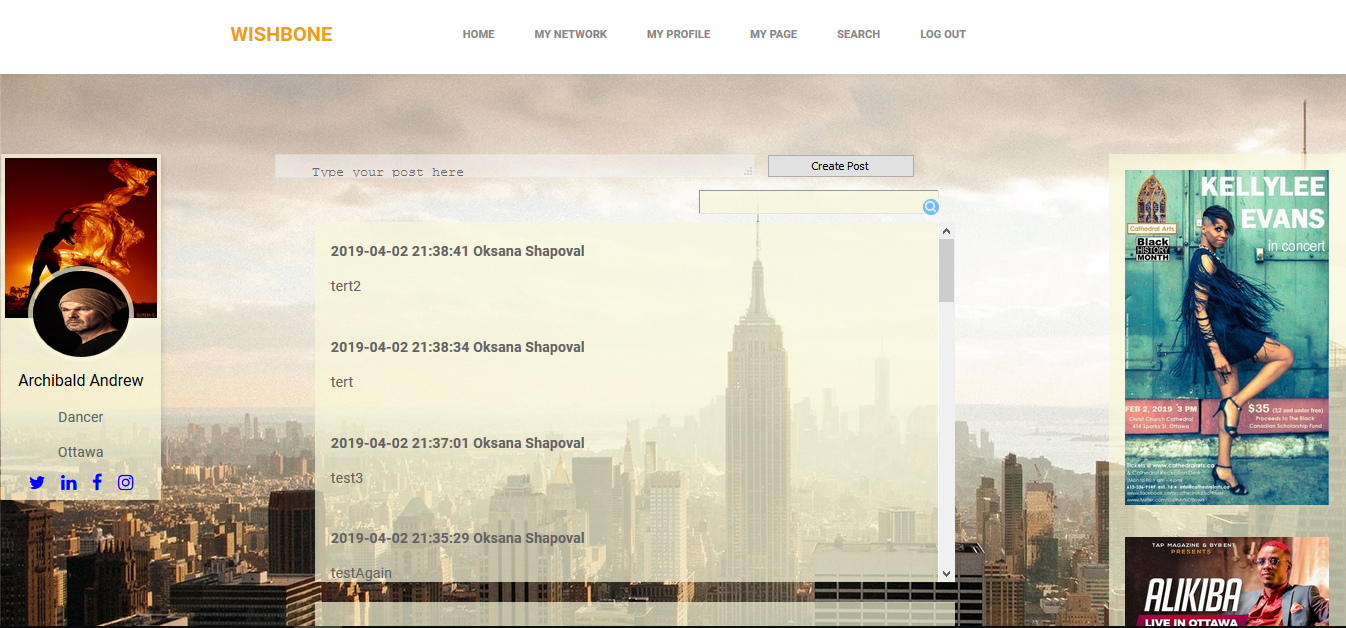


Figure My Page (User Homepage)

The user homepage allows users to view a feed of user posts. It includes the users name and a placeholder for image (to minimize cost we use select images on platform instead of hosting service for images). The User can add their own posts, or search through existing posts.

This page was primarily worked on by Oksana, with minor contributions from other members. More information in Appendix B

## MyNetwork

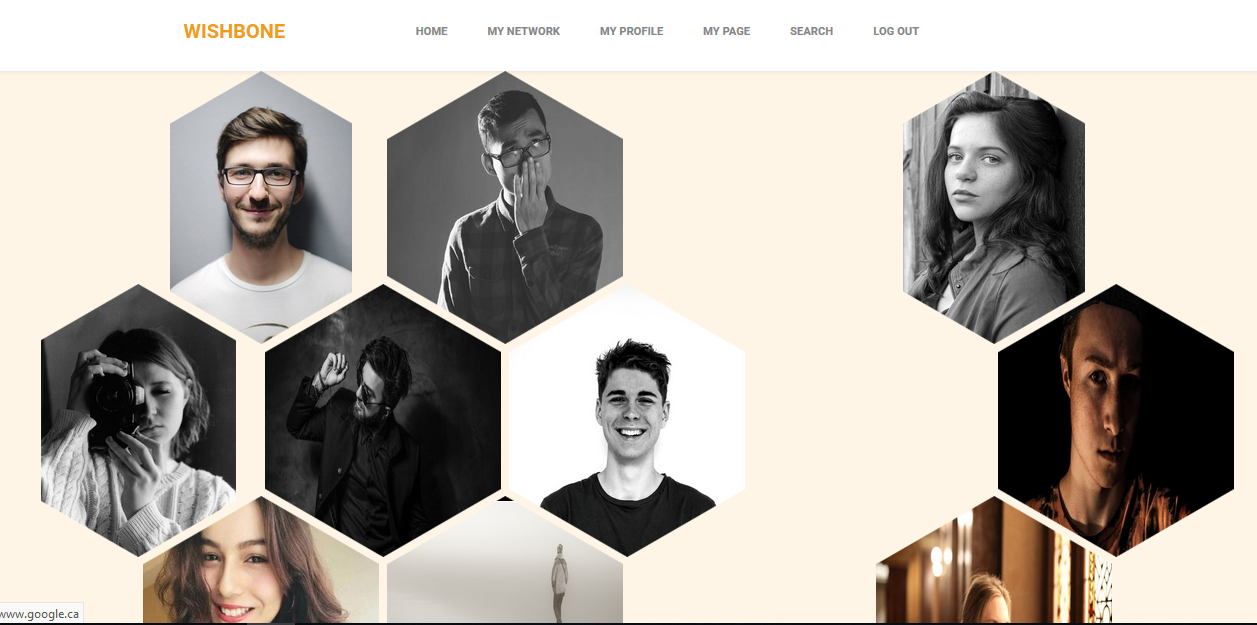


Figure MyNetwork Page

My Network shows users on the left that are connected – allowing you to send messages and view profile, it also has pending outgoing requests, and pending incoming requests. On the right side are suggested users to add to your network.

Primarily worked on by Andrew. Minor contributions from other members. More information in Appendix B.

## Messages

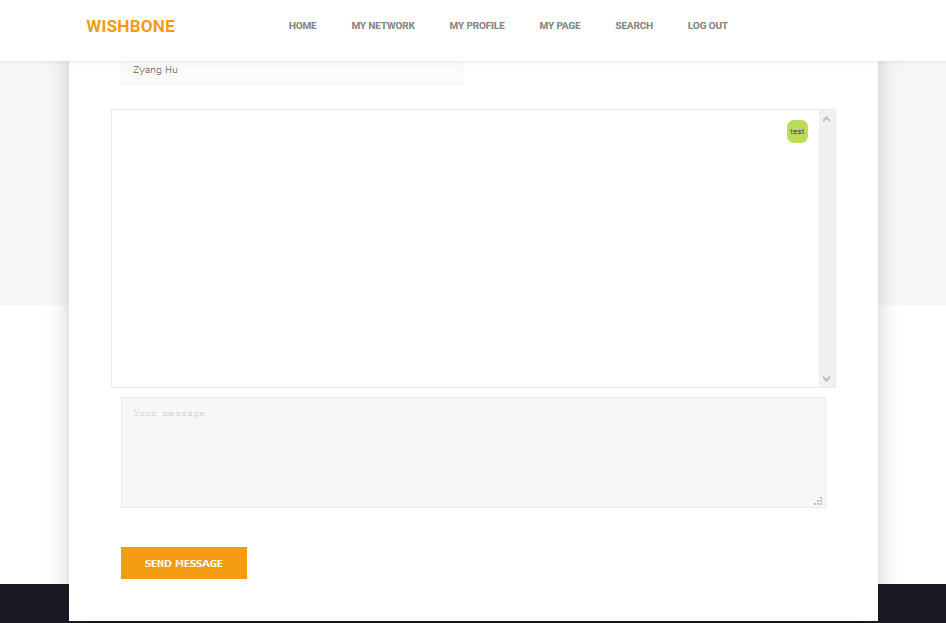


Figure User to User Messaging

Messages is linked to by MyNetwork, and allows a user to communicate with another user in their ‘MyNetwork. Real time text communication between users.

Primarily worked on by Ksenia, with minor contributions from other team members. More information in Appendix B.

## My Profile

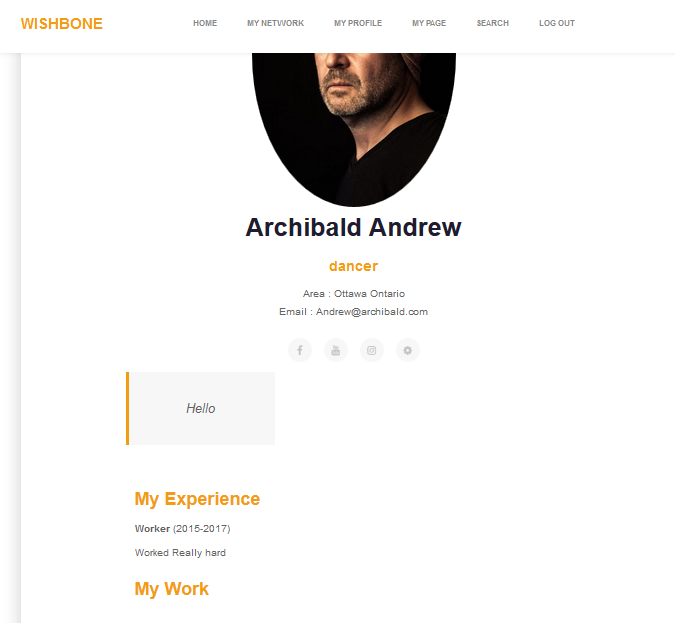


Figure My Profile

My Profile shows user profile information, this is also re-used to load other users. Contains contact information, Description, previous experience and previous work.

Primarily worked on by Minyi. Minor contributions from other members. More information in appendix B

## Search

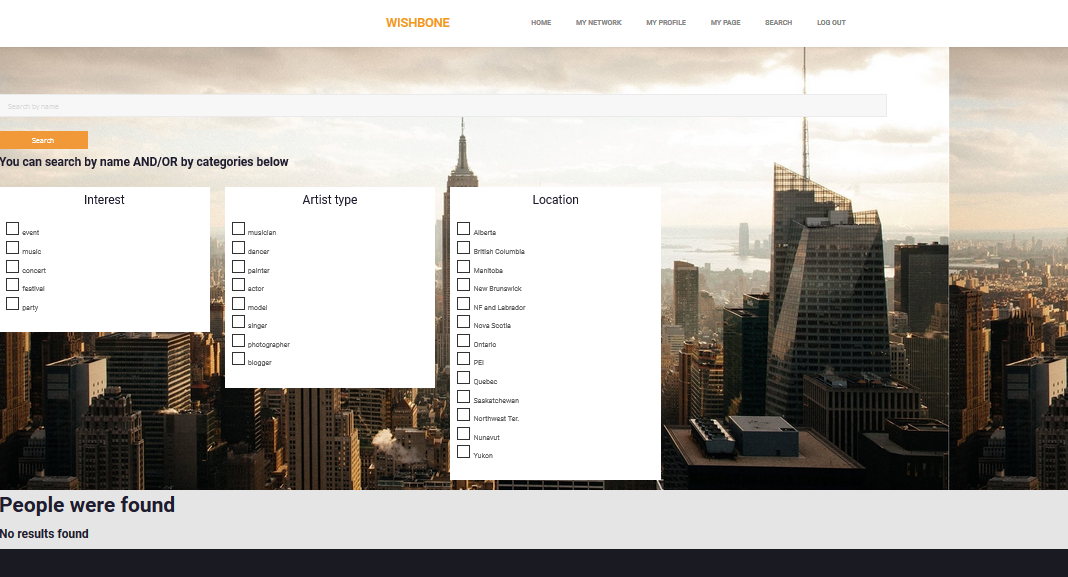


Figure Search

The search function allows users to find other users based on artist type, interests, location, and by name.

Primarily worked on by Svetlana. Minor contributions from other Members, more information in Appendix B.

# Conclusions

The data design. One of the major considerations for the Data design is that we knew from the start that the system would be incomplete, and that they would have to continue on the project to get it completed. We know some information about the future design, namely that the goal is for organizations or individuals to be able to offer jobs to artists, and potentially to schedule events. The goal with the database was to try to develop the database such that it would be structured enough to be coherent, but flexible enough to not be rendered obsolete when further additions are made.

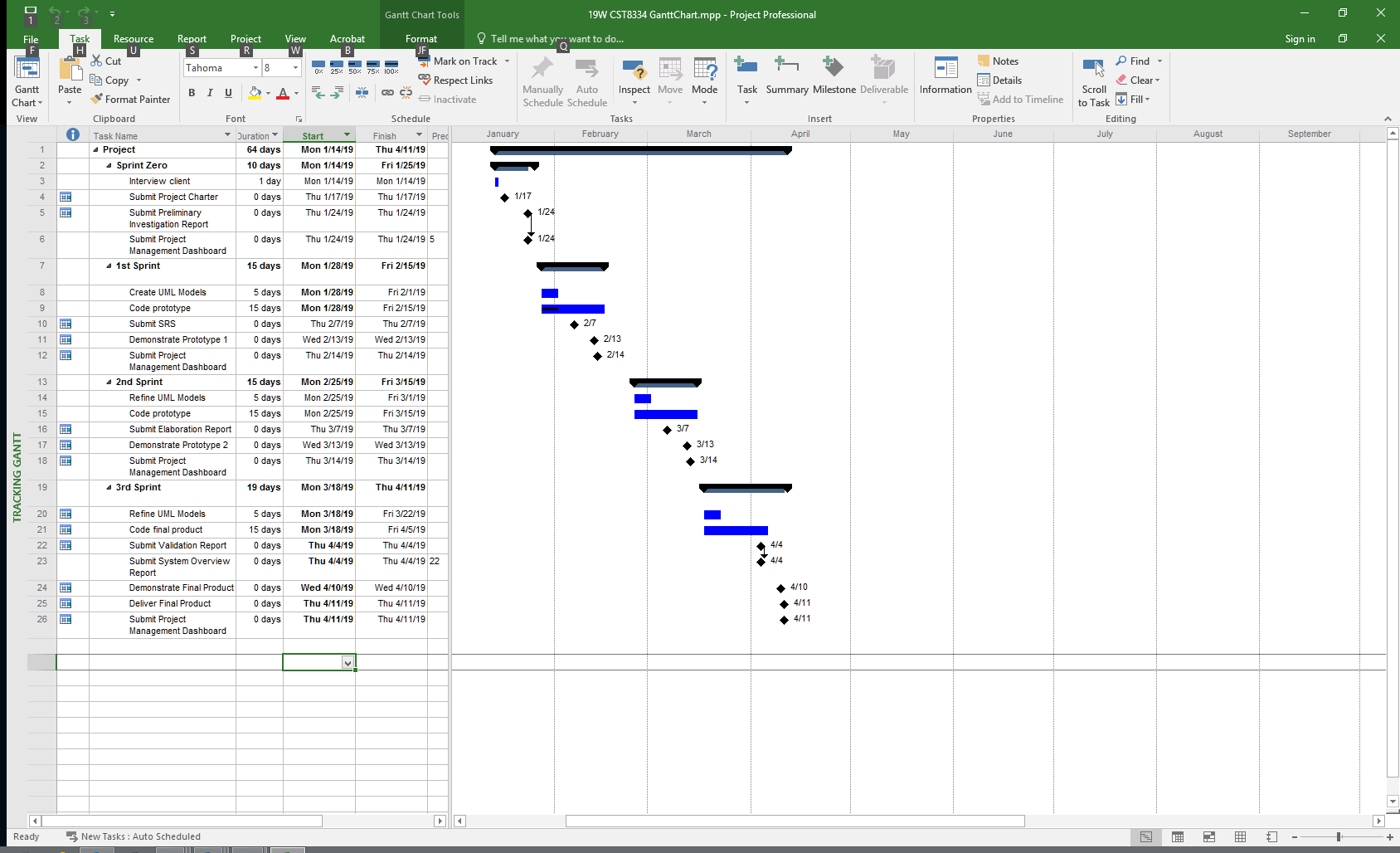
Architecture Design. We believe that the architectural design works well for the client, as it can be deployed to a low/no cost service. The architectural design is suited for the clients more immediate and short term goals, than long term. Long term the architecture would likely be better suited to run on 3 tiered system with separate database, business logic, and presentation layers. The 3 tiers however increase the complexity of the system, and would increase the immediate cost. The long term costs may be less, as the 3 tier architecture has more flexibility for cost savings. For example a middle tier could minimize large data transfers from the database by caching, something that is more difficult to do with PHP. The startup costs would potentially be higher, as we found a free php/mysql host, but most server hosting comes at a premium. Also with 3 tiers there is potential for more expense as data flows between an additional layer, meaning potentially more bandwidth costs, i.e caching may reduce bandwidth, but without caching sending the data through another tier is just more bandwidth.

Program Design. We believe that our project has covered the basic elements of a social networking site. Established social networking platforms are quite large and complex, our system is relatively simple and minimal by comparison. Our hope is that it is useful for continued development and expansion of the project.

# Appendices

## Appendix A – Schedule Gantt Chart

The Appendix A was provided as an example by Patricia Murphy (January 2019) Software Development Project. It was used here with no modifications.



## Appendix B – Source Code

| **Source Code**  **(filename of each program**  **module)** | **Author** | **Created (C)**  **Updated (U)** | **License** |
| --- | --- | --- | --- |
| userHome.php  ~~out.php~~  assets/css/userHome.css  SQL/interest\_view.sql  SQL/artform\_view.sql  SQL/artProfile.sql  insertPost.php  searchFeed.php | Oksana | Jan 2019 (C)  Apr 2019, (U) |  |
| Search.php  dao/searchDAO.php  search\_action.php  assets/css/search.css  SQL/message\_view.sql  SQL/friends\_view.sql | Svetlana | Jan 2019 (C)  Apr 2019, (U) |  |
| Profile.php  model/proflielist.php  Authentication.php  dao/mailinglistDAO.php  assets/css/profile.css  profilesetting.php  successful.php  SQL/Experience.sql  model/experience.php | Minyi | Jan 2019 (C)  Apr 2019, (U) |  |
| myNetwork.php  model/myNetworkModel.php  SQL/WishboneDDL.sql  dao/myNetworkDAO.php  header.php  assets/css/myNetwork.css  dao/server.php | Andrew | Jan 2019 (C)  Apr 2019, (U) |  |
| Signup\_user.php  Login.html  dao/AbstractDAO.php  dao/AuthenticationDAO.php  model/Authentication.php  Logout.php  Login.css  Login.php | Zeyang | Jan 2019 (C)  Apr 2019, (U) |  |
| Chat.php  Ajax.php  SQL/User\_profile\_view.sql  SQL/Public\_user\_profile.sql  Database.class.php  Chat.css | Ksenia | Jan 2019 (C)  Apr 2019, (U) |  |
| Wishbone.sql | Oksana, Svetlana | Feb 2019 (C) |  |
| assets/css/myNetwork.css | <https://github.com/web-tiki/responsive-grid-of-hexagons>  Modified by Andrew | Dec 18, 2015  (C)  April (U) | Apache 2.0 |
| SQL/Wishbone.sql  SQL/Wishbone.DDL | All Members | April 2019(U) |  |
| Remaining material | Unknown/Various  From Wishbone |  | Various |

Depreciated/removed files are ~~strikeout~~