

# WebApps Project - Music Collaboration (change name?)

Ashley Hemingway, Charchris Sloan,  
David Liskevich, Edwin Kamulegeya  
Group 9 (g1327109)

June 9, 2014

## 1 Introduction

### 1.1 Overview

During discussions to find a suitable web app, it was discovered that a few members of the group had problems when trying to development music. The main reason for this problem was that they were only good a certain part of the music development process, needing help with other parts. This is where the idea for our web app came from, allowing people from all over the world to help each other to make music together.

We also wanted to make a unique app, which either was a brand new original idea or one which improved on limited existing apps. With further research into our proposed solution it was discovered that a similar app already existed, although it was below our expectations and lacked a clean user friendly interface with good functionality. This is why we decided to continue with our original app idea, but to make it much better than the existing app.

The solution for this problem will be discussed in much further depth in later sections.

### 1.2 Requirements

Considering our web app deals with music collaboration, below are the required functionality of the web app.

- Home Page which is simple and straight to the point, allowing for easy access to sign up or to login in to an already existing account.
- Profile pages which allow for users to keep track of their own songs and album artwork created via the web app. It will allow for easy uploading and downloading of files while giving the ability for the user to change the information about themselves.
- The ability to create a project, specifying the type of help required and the genre of the music associated with the project.
- Users must be invited by the project owner to join and collaborate within a specific project.
- Within a project, it will allow members of the project to add new music and album art to be upvoted/downvoted by members.
- A top featured page showing the top rated music and best collaborators.
- The ability to search via users, projects, genres and music files.
- Group chat between all members of each project to discuss the way the project is developing.
- Ability to rate the help and contribution of users.

### 1.3 Targets

Who the fuck knows???

## 1.4 Alternative Apps

During our discussions, as a group we also came up with other numerous ideas which shall be mentioned.

We firstly discussed the idea of a lending app which would allow people to keep track of items which they have lent to other people, while being a place to ask to borrow items from other people. We thought that this idea didn't have enough scope or feasibility as it lacked a decent amount of features and we thought it wouldn't be used as much as desired.

Secondly, we thought of making a game which would allow interaction for multiple users via handheld devices, but the overall game would be played on a computer. Basically the handheld device would act as your own personal game management screen while the computer would keep track of the overall game. We liked the initial idea, however, we thought that the overall concept would be too hard to implement due to our time frame and the lack of experience in the team.

Finally, we wanted to build a web app which would allow people to basically build their own website containing of their customised RSS feeds. We liked this proposed idea as it would allow us to extend it with features when needed, although we diverged away from this idea in the end as we thought it lacked a full user interaction experience as required, while also being pretty easy to implement.

## 2 Project Management

### 2.1 Group Structure

Obviously, it would have been ideal for every member of the group to sample every part of the development stage, whether that was the back-end or front-end implementation. In practice, we decided to split the group up into the front-end and back-end sections, mainly because none of us had experience in this type of development, so we would have to learn everything from the start. Splitting the group up meant we could focus on one section and become an expert in this section, allowing for a better end product.

The back-end was mainly handled by Ashley Hemingway, who will be designing the database structure and writing the database interaction code in PHP and handling the constant data pulling with Ajax and JavaScript.

The front-end was handled by the other group members (Charchris Sloan, David Liskevich & Edwin Kamulegeya). Their main job was to design the user interface and carry out the implementation, using multiple languages like HTML, CSS and JavaScript/jQuery.

The group was split up like this, as we decided that the web app design and user interaction should be the main feature, allowing the web app to be easy and intuitive to use. This meant that 3 people focussed on this area. The main problem with this decision was the obvious disparity between the front-end and back-end development, meaning that we somehow had to bring these differences back together for the final product. This was done via constant interaction between the group members, meaning that the front-end and back-end could be constantly be linked together during the development stage once certain sections were completed. For example, once the homepage design had been implemented by Edwin, David and Charchris, this allowed the homepage to be passed to Ashley in order to complete the form validation and the interaction with the database.

### 2.2 Implementation Languages

Below are the choices of coding languages we chose to implement our web app in. Although there are many to pick from, we thought that the combination of all of these, combined with our knowledge of web app design, would give us the best chance of producing a highly polished and functional website.

### 2.2.1 HTML/CSS

HTML and CSS were an obvious choice. Their popularity is down to the fact that they are the best basis to design a website on. HTML offers all of the functionality required, and combined with CSS to easily change the look of the website, it is their perfect choice. HTML also neatly interacts with JavaScript/jQuery and PHP which is a massive advantage. Furthermore, the newly updated HTML5 offers even more functionality like drag and drop and embedded audio which are both requirements of the web app. DEVELOP FURTHER

### 2.2.2 Bootstrap

Bootstrap was chosen because it basically allows us to develop a great looking website from scratch. As a group we were wary of using a website template as it wouldn't give us the best learning experience, so we decided to use the Bootstrap components when required. This allows us to keep a constant design throughout the whole of the website while making the trivial component coding not necessary, meaning we can focus more on the user experience and functionality.

### 2.2.3 JavaScript/jQuery

JavaScript enables us to add better usability to the website while also keeping things simple. It is widely supported and offers a vast array of frameworks to help us, most noticeably jQuery. It will become very useful when implementing such features like form validation and easing effects to provide the website with an enhanced user experience. Furthermore, JavaScript is a requirement when using Ajax, which we will discuss later.

### 2.2.4 PHP

Although there are many options to code the back-end in like Node, Django, Ruby, Python and Java we decided to choose PHP. The main reason behind this was that PHP is widely used and it is therefore very stable, with many tutorials and supporting libraries. Since none of us have any experience with developing the back-end of a web app, we decided that this would give us the best learning experience. It also offers an easy to use interaction with the Postgresql database and means we can push and pull from the database when required via the use of Ajax. It also had great functionality when it came to data encryption and JSON (JavaScript Object Notation) which benefitted us when transferring data between the server and client.

### 2.2.5 Ajax

Ajax became an important part of the app as it allowed us to pull and store data from the database, without the need for a page refresh. As a major part of our app is to do with user experience this became very useful. Since we were implementing our app in JavaScript/jQuery, using Ajax was an obvious choice as the major functionality is already included in those frameworks, while the server side data handling was relatively easy as well. Ajax became useful for features like the user chat and the constant updating of the top featured music. There aren't many alternatives to Ajax available, and since our app doesn't require an immediate update (i.e. it can wait 0.5 seconds to get the latest data) it is the best technology at our disposal.

## 2.3 Design Processes

## 2.4 Back-up Systems

Due to the fact that we are using the provided Virtual Machine to run our server and Postgresql database on, backing up the data is imperative as the VM isn't backed up automatically. In order to back-up the web page data/scripts (e.g. PHP scripts, css, html, JavaScript), we have been using GitHub. This was a great platform to use, as it allowed

is to back-up our work, roll back to a previous stage of the design process and to easily share our work throughout the whole group on different devices. The choice to use GitHub over GitLab didn't really cross our mind as they both provide the functionality of git, while been both easily accessible.

As we were using a large database storing multiple values for all aspects of the app, it was also important that the database didn't become corrupt or lost. In order to stop this from happening, we implemented an auto-back-up feature on the VM so the database would be automatically back-up every day, allowing us to restore to previous points when required. This became very useful when playing around with data inserts, as on occasions this didn't work as expected.

## **3 Actual App**

### **3.1 Program Description**

### **3.2 Implementation**

Design patterns, diagrams, screen shots

## **4 Acknowledgements**

### **4.1 Libraries Used**

### **4.2 Code/Pictures used?**

### **4.3 Legal Issues**

## **5 Conclusion**

### **5.1 What We Have Done**

### **5.2 What We Have Not Done**

### **5.3 What We Have Learned**

### **5.4 What We Would Have Done Differently**