

WebApps Project - Music Collaboration

Ashley Hemingway, Charchris Sloan,
David Liskevich, Edwin Kamulegeya
g1327109

May 28, 2014

1 App Description

The main objective of the application is to allow people to easily collaborate with musicians who may have expertise in separate fields. For example a lyricist, may want someone to sing or even remix his lyrics and this application would allow said lyricist to publicise his project, and ask for help from users with the requirements he needs. So the lyricist may need a sound engineer with expertise in software such as 'Massive', 'Pro Tools' or 'Ableton Live'(software where many producers and dj's produce their sounds). Publishing requests for help will be done in a specialised area where it will be able to match users with projects. The project owner will then be able to invite the correct people to the project in order to collaborate, which will be done via user requests. Users will be able to add skills to their profile, which would make it easier for someone starting a project to find people with a certain skill set.

The project page will be a multi-function but easy to use interface which will allow the project owner certain administrative rights in order to customise the project development how he/she wants. The project interface will have an efficient organisation structure which will allow project collaborators to upload their own music snippets into the relevant project sections, such as 'intro', 'melody' etc. Obviously, the web app will store all of the music snippets on the database so the project will become a workspace for all of the users, allowing them to upload and download the relevant sections so they can mix on their own computers. The project will also accommodate people who aren't musicians, allowing people who specialise in other areas like album artwork to contribute to the overall finished product.

The web app will also be a place where bands and individual musicians can show off their work. This will be optimised by the ability to follow certain users and to share music within the web app with other users. These pieces of work will be showed on a global stage as people can search for specific music genres, while also utilising a 'top rated' section for the best work out of the web app.

2 User Interactions

Our proposed web app has multiple user interactions which are both vital to the functionality of the website and the user experience. Obviously, the main user interaction comes with the ability for users to collaborate and share in order to compose music together in the project workspace. These users will only be invited to the project at the owners discretion once convinced the user can add a positive effect to the project. This workspace will have the ability for the users to communicate with each other via a chat function. During the development stage, users within the project can positively rate sound clips to indicate the best work, and once the project has been finished, the global user base can also rate the finished work. Users will also be able to follow their favourite musician in order to find the latest work being produced, while further features will include the ability to share music among users directly and to publish their work on other popular websites (including soundcloud, Facebook and twitter).

3 Implementation Languages

3.1 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 is offers all of the functionality required, and combined with CSS to easily to change the look

of the website, it is there perfect choice. HTML also neatly interacts with JavaScript/jQuery and PHP which is a massive advantage. Furthermore, the newly updated HTML5 offers even more functionally like 'drag and drop' and embedded audio which are both requirements of the web app.

3.2 Bootstrap

Bootstrap was chosen because it basically allows us to development a very nice 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.

3.3 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 make the website more user friendly. Furthermore, JavaScript is a requirement when using Ajax, which we will discuss later.

3.4 PHP

Although there are many options to code the back-end in like Node, Django and Ruby, 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.

3.5 Ajax

Ajax is a necessary component to our web app since it requires content to be pulled from the database once it arrives. This is 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) it is the best technology at our disposal.

4 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 is 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. Ashley will be partly supported by Charchris Sloan, who will be predominately designing the user interface, but will also help with the back-end when required. The front-end will be developed by David Liskevich, Edwin Kamulegeya and helped by Charchris. Their main job will be to design the webpages and implement them, using multiple languages like HTML, CSS and JavaScript/jQuery.