

National College of Ireland
BSc in Computing
2015/2016

Ciaran Browne
X12375786
Ciaran.browne1@student.ncirl.ie

Music Lounge



Technical Report



Table of Contents

1	Introduction.....	5
1.1	Introduction.....	5
1.2	Executive Summary.....	5
1.3	Background.....	6
1.4	Aims.....	6
1.5	Technologies.....	7
2	System.....	8
2.1	Requirements.....	8
2.1.1	Functional requirements.....	8
2.1.2	Non Functional requirements.....	21
2.1.3	Data requirements.....	23
2.1.4	Data Protection and Security.....	23
2.1.5	User requirements.....	23
2.1.6	Software requirements.....	24
2.2	Design and Architecture.....	24
2.3	Testing.....	26
2.3.1	White BoxTesting.....	27
2.3.2	Black Box Testing.....	27
2.3.3	UAT Testing.....	28
2.3.4	Customer Testing and Analysis.....	28
2.4	Site Layout and Interface.....	32
3	Conclusions.....	37
4	Further development or research.....	38
5	References.....	39
6	<i>Appendix</i>	41
6.1	Project Proposal.....	41
6.2	Monthly Journals.....	41

1 Introduction

1.1 Introduction

This document is the technical report documentation for the final year college project. It investigates features of the project such as project aims, technologies used, user requirements, system architecture etc. All features that are relevant to the project as of the 11th of May are discussed and analysed within this document' The previously uploaded documents that relate to the project, relevant diagrams, as well as the monthly journals kept by myself are attached in the appendices.

1.2 Executive Summary

My fourth year project is a music based web application. It allows a user of the website to register an account, and once logged in, they are brought to a number of features on different pages that are of interest to anyone with a passion for music. As more users register and more profiles build up on the site, more activity and discussion can take place online, therefore my site is heavily reliant on user activity. Features given to users are mainly based around API's, such as the Google Maps API which displays upcoming live events based on the area the user is in. Also contained on the site is a discussion page where users can create posts and generate a discussion in comments on a range of topics such as their favourite bands, previous shows they've been to, instruments they play etc. – much like a forum. They are also given a resource which allows them to search for their favourite artists, and be given a bio of that artist as well as being shown similar artists, recent news/blogs/reviews about their search. Chat rooms are available for online discussion, and users of the site can add each other as friends – an important feature for getting an online community and social website up and running.

1.3 Background

When brainstorming about what ideas to do for my project, I aimed to do something that I have a big passion for as this would increase my overall enjoyment of the project for my final year. With music and sport being my two main interests, I started to consider and take note of potential ideas for my project. I carried out research about what resources were available online, and the music side of things seemed to have a lot more tools to work with and resources such as API's available to me, and generally seemed more flexible. I discussed with my supervisor about what technologies to use and how to lay out system architecture and how to go about structuring everything in development. Once I got the green light from him on my idea and proposal, I was clear to move on and pursue development.

1.4 Aims

By designing this project my number one objective was to create a culture that encourages people in the general public to look for, seek out and be curious about what live shows are upcoming and occurring in their area and for them to be made easily aware of who or what is coming to perform near them. Once people actively seek out live shows to attend, and the attendance of concerts and shows increases, this in turn will encourage artists and musicians to play live and go on tour more often- once they know there is market for them willing and waiting.

I wanted to create a user friendly, and simple website, hosted on a cloud server where users can discuss their passions and favorite artists, share their experiences at live shows and get an online active music community going all in the one place. I was also looking to get people interested in and discovering musicians that may sometimes go under the radar, and ultimately to create an active music based website with a number of useful features.

1.5 Technologies

In my project I used a number of technologies to develop both frontend and backend. I used Bootstrap for the modern interface and mobile friendly framework as well as some custom CSS and JavaScript for Frontend design. For the backend I'm using PHP (no framework) due to PHP working well with databases and a wide range of resources, tutorials and online aid given to me in this area. I also decided to use PHP as it was a language I had previous experience using and enjoyed using as part of my third year project. My confidence with PHP was much higher than other development languages, for example Ruby on Rails, which I have only been learning to use this year. I realised from the beginning the project would be hard work, and I wanted to not waste time learning a new language when I was confident I could achieve my goals using PHP. The use of API's involves working with JSON type data, and using JQuery and AJAX, calls to pull the information I need from the data provided in the external API. Music API's I've used are Google, Spotify, EchoNest and Song Kick for the Google Maps. MYSQL is the relational database I'm using together with PHPMyAdmin. I have used both Notepad++ and the PHP Storm IDE for writing my code.

2 System

2.1 Requirements

This section will investigate different types of requirements relating to the final year project.

2.1.1 Functional requirements

This section lists the functional requirements in ranked order. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish. **NOTE:** For Use Case Diagrams and Class Diagrams relating to each functional requirement please reference the Appendix where a diagram for each is stored.

1.) Requirement: Registration/Log In

Summary

The main requirement for the system, a user must register themselves so they can get access to all available functions

Use Case

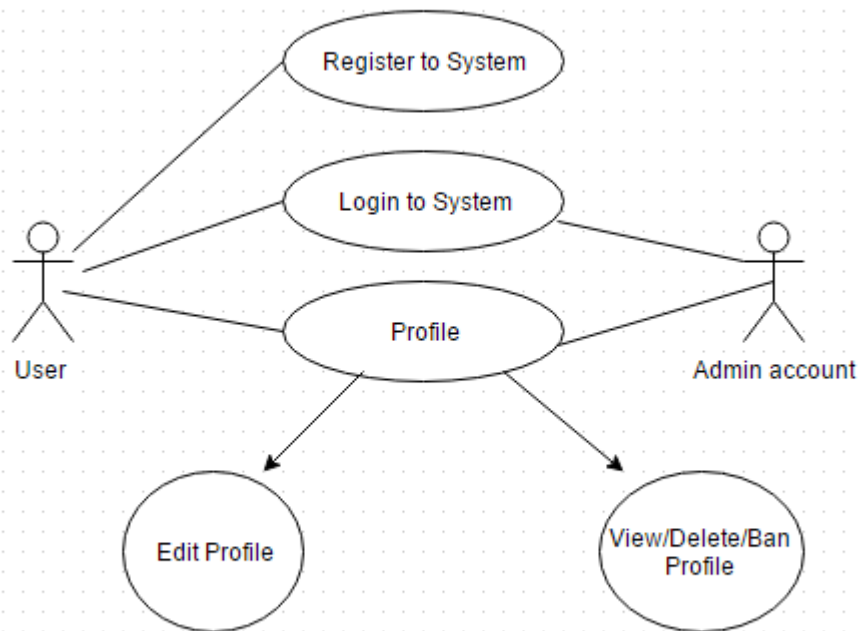
Scope

The scope of this use case is to examine registration of users, and once they are registered, logging in.

Description

Users must register themselves by entering a username, password and valid email address, which they will have to confirm. Once registered successfully they can log into the system

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode

Activation

This use case starts when the user opens the application and is brought to the index page, which will ask them to log in with their registered details or to create a new account if they are not registered

Main flow

1. The system provides a textbox to enter username and password
2. The User enters credentials
3. The system sends entered credentials to the database for verification
4. The User gets logged in to the system

Alternate flow

A1 : <Registration>

1. The System provides details for user to input credentials for new account being created
2. The User enters details to register new account and submits
3. System checks that all credentials entered meet standards
4. User account is created

Exceptional flow

A1 : <Log in fail>

1. System provides textbox for user to log in with email and password
2. User enters credentials
3. System sends entered credentials to database and checks for verification
4. Record does not exist in database and user is asked to log in again.

Termination

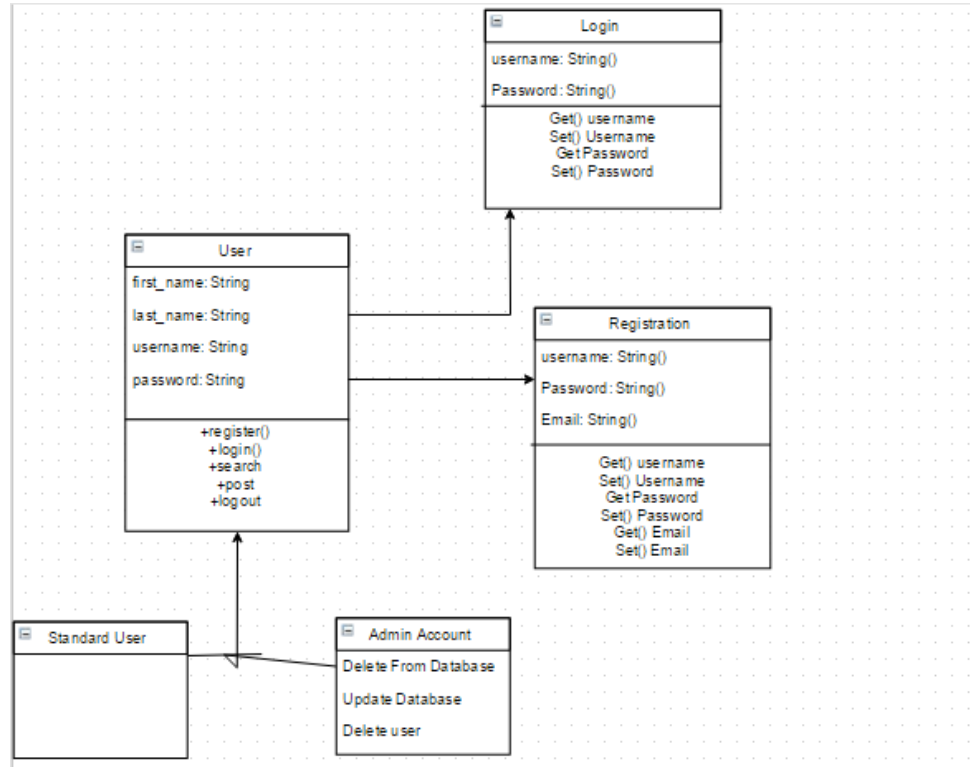
The system presents the home page of the website

Post condition

The system goes into a wait state until user logs out of system

Class Diagram

Class diagram for login and registration system demonstrating the structure of this system



2.) Requirement: Friend System

Summary

Registered users on the site must be able to send friend requests to other users on the system. Once they are sent a request, they can either accept it and become friends with that user, or ignore it.

Use Case

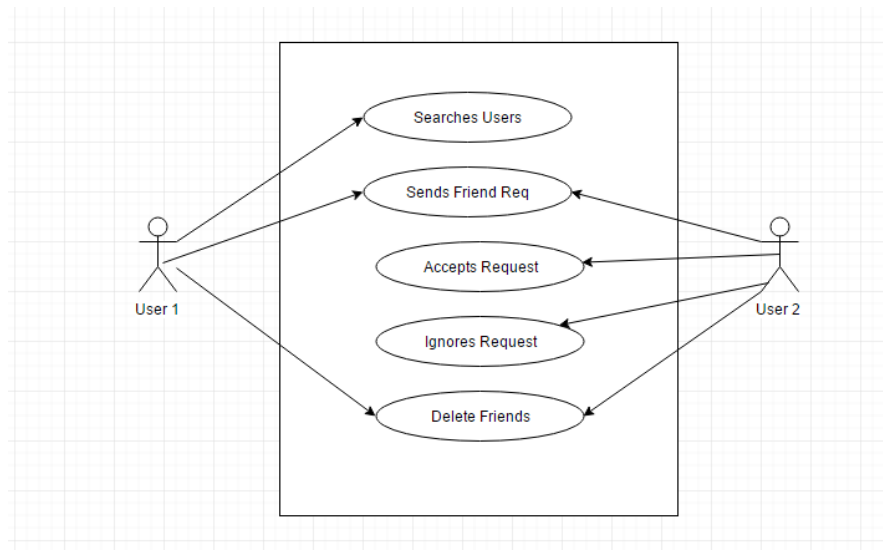
Scope

The scope of this use case is to examine the friend request system on the website and why it is necessary

Description

When a user successfully registers and account they will be given the option on the home page to 'begin adding friends.' When a user navigates here they are show a list of all users who are using the site, clicking their username will bring them to their profile page and give the option to add them is a friend. If a request is sent, the 'user_id' of both sender and receiver is held in a 'friendrequest' database, until the receiver decided to ignore or accept the request. If accepted, the 'user_id' of both users is held in a 'friends' table and they are both listed now as being friends. If ignored, the request that was sent is dropped from the 'friendrequest' table

Use Case Diagram



Flow Description

Precondition

The system is active and the user is logged in

Activation

This use case starts when the user navigates to the member's page and is given the option to begin adding friends

Main flow

1. User 1 clicks into user 2's profile
2. User 1 sends a friend request to user 2
3. User 2 accepts friend request
4. Both Users are now friends
5. User 1 can now see user 2's username under his list of friends

Alternate flow

1. User 1 clicks into user 2's profile
2. User 1 sends user 2 a friend request
3. User 2 choses to ignore friend request
4. Users 1 friend request is dropped from database
5. Both users remain not being friends and option is given to user 1 to add user 2 as a friend once again

Termination

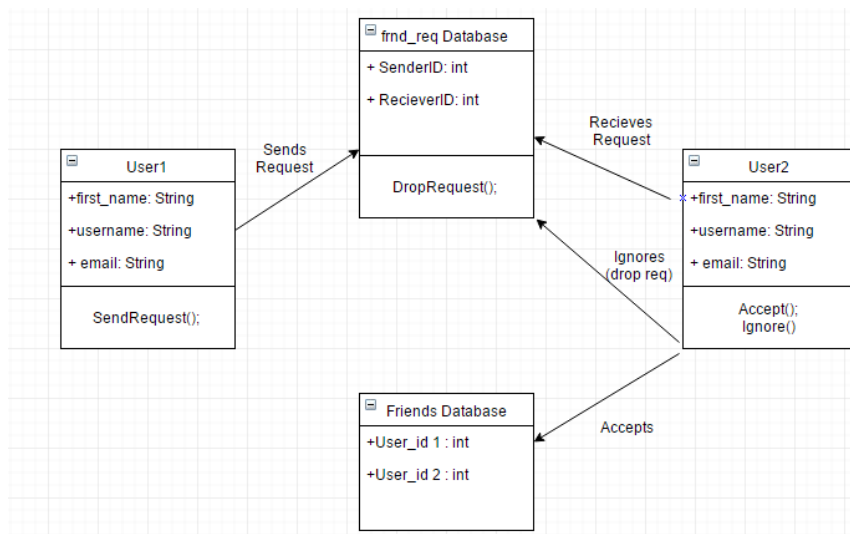
Use case completes when user 2 either ignores or accepts the friend request

Post condition

Both users are now friends and are listed as being friends in the database and on each other's profile page

Class Diagram

Class diagram for the friends system and how it operates.



3.) Requirement: Displaying events

Summary

When on the events page, a google map is displayed centered on the user's location which displays live upcoming events in their area.

Use Case

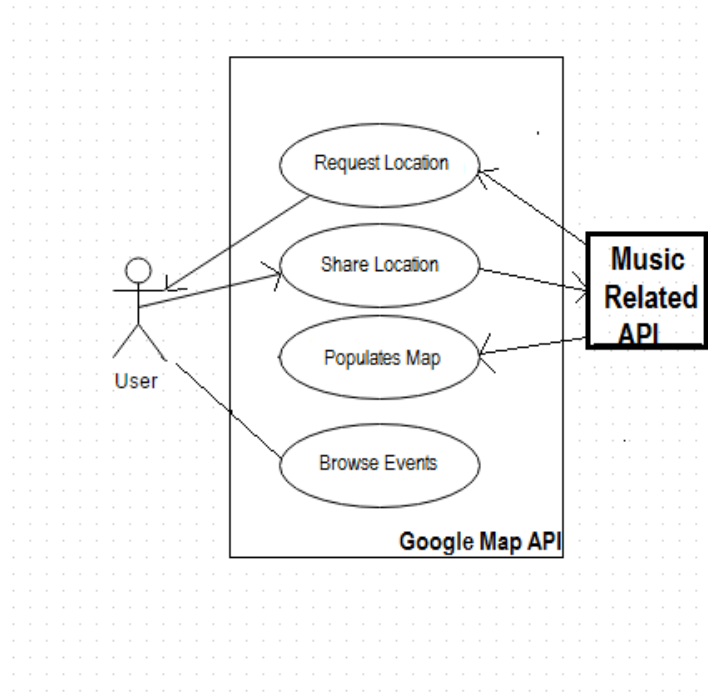
Scope

The scope of this use case is to show events to the user based on their location

Description

When the user navigates to the events page, they will see a google maps div that is centred on their location and pins can be found on the map based on live upcoming events in that area. A user must allow their geolocation for this requirement to take place. Google Maps is combined with the Music Related API so that upcoming events and location can be combined

Use Case Diagram



Flow Description

Precondition

The system is active and the user is logged in

Activation

This use case starts when the user navigates to the events page after log in

Main flow

1. The user is prompted to allow their location to system
2. The system provides a google map centred on users location
3. Music related API which are used to collect all data is crossed with google maps, and upcoming events are shown on the map based on the location

Alternate flow

1. User navigates to events page
2. System prompts user to allow geo location
3. User declines
4. Google map cannot display user's location, so a default map will be shown with no pins

Exceptional flow

1. User allows geolocation and navigates to events page
2. Google map displays but no pins are displayed due to api not working

*This would be considered a technical issue, if one of the API's is not working or down for maintenance, the API mashup will not work correctly

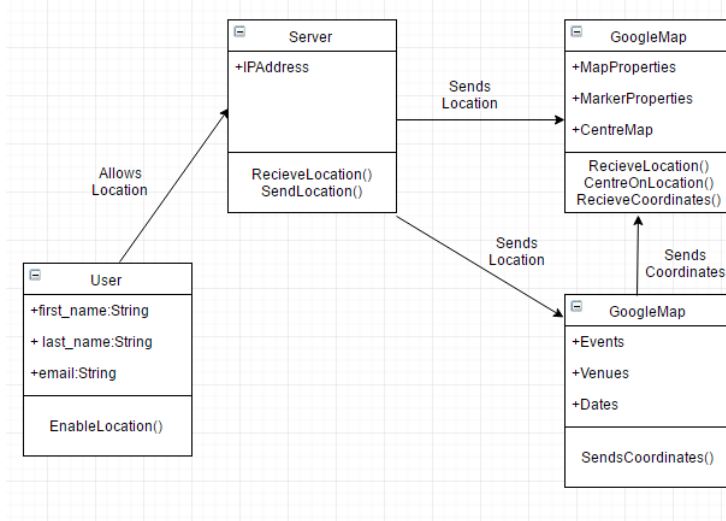
Termination

Once the pins are displayed on the map displaying what event is taking place

Post condition

User is shown google map with their location, pins are stored on the map with upcoming events, who is playing, where and when.

Class Diagram



4.) Requirement: Discover (API call)

Summary

User searches an artist, the search result is pulled from the music related API returning a number of features.

Use Case

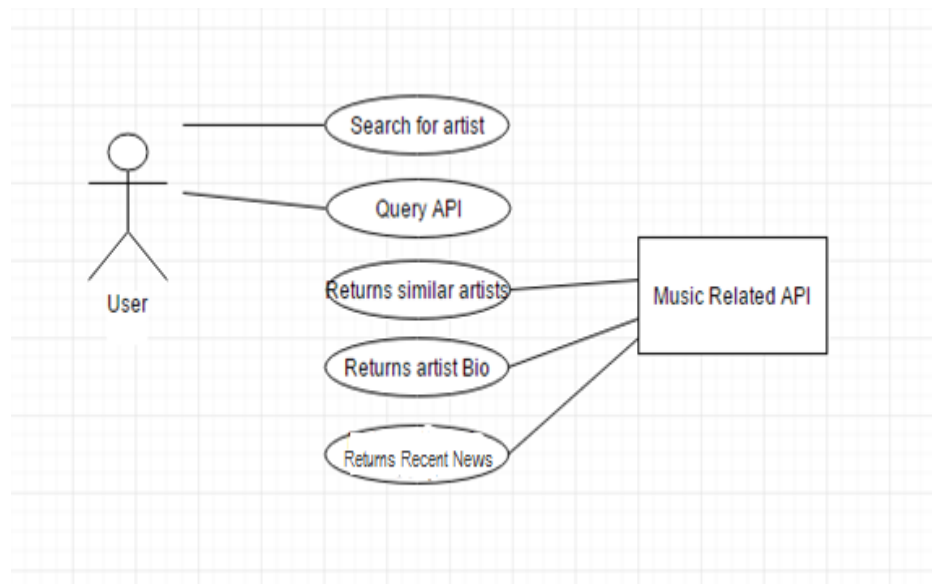
Scope

The scope of this use case is to investigate what happens on the 'discover' page when a user queries the Music API.

Description

When the user navigates to the discover page, they are given a search bar to enter a band/musician they wish to search. When they press enter, the music related API is called and results are shown based on the query. An image of the artist will show, as well as related artists, what genres of Music they fall under, and recent News, Reviews and Blogs about that artist.

Use Case Diagram



Flow Description

Precondition

The system is active and the user is logged in

Activation

This use case starts when the user navigates to the events page after log in

Main flow

1. User navigates to discover page
2. System provides search bar to enter query
3. User searches for something
4. Music related API is called and displays result on the system
5. User is shown artist Bio, Images, Related Artists, News, Reviews, Blogs.

Alternate flow

1. User searches for an artist
2. Music related API is called
3. The API does not recognise the users query
4. System returns error message to user asking them to try search again

Exceptional flow

1. User searches for a lesser known artist who may not have recent news/reviews/blogs or even images
2. System returns Artist bio and whatever information possible there is to return, leaving out that which can't be found

Termination

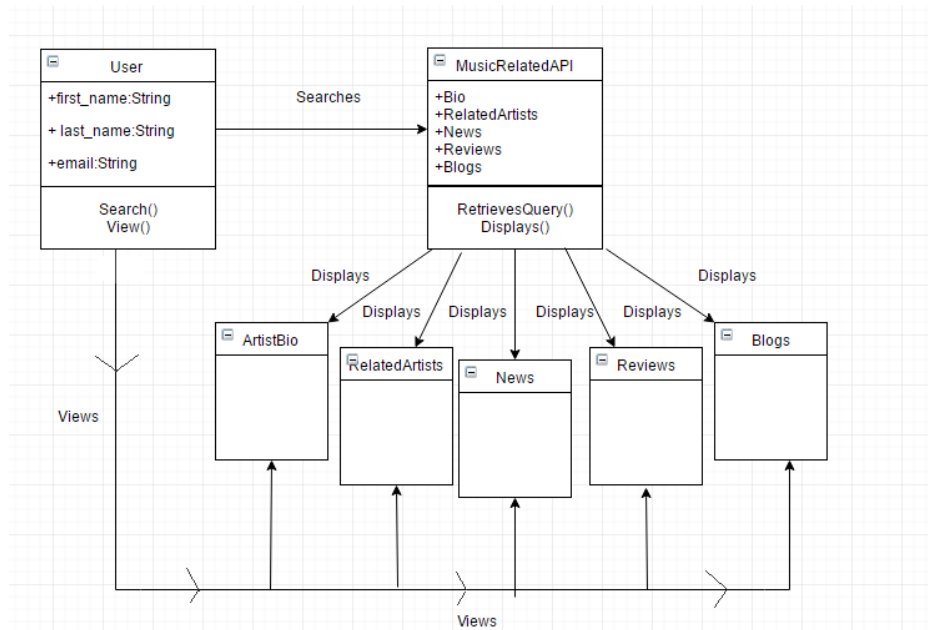
Once search is complete and results are shown

Post condition

User is shown a page with their successful search and the required information based off their query displayed.

Class Diagram

Class diagram for the Discover system demonstrating the structure and how it functions



2.1.2 Non Functional requirements

Physical Requirement – Anyone using the system will need a computer/tablet/phone with an internet connection, a mouse and keyboard for input, and a registered account

Resource Utilization Requirement - The system will make use of at least two API's in google maps and whatever music related API is chosen and will also be reliant on a database that is held on the server

Availability Requirement - The application must be always available to any user with an internet connection on a smart phone, tablet or computer

Security Requirement - The security requirement for the application is a log in system, which password protects the user's profile. When storing the password in the database, the password is encrypted rather than stored in plain text.

Reliability Requirement - Reliability is a key factor and a large amount of this requirement is down to the server the system is hosted on, and the API's functioning correctly. These are two well-known API's that are widely used and have a good reputation.

Accessibility Requirement – The application is available anywhere with an internet connection, and will display neatly on any device because of the bootstrap framework.

Portability Requirement – Because my application is developed in standard PHP and uses MYSQL as the relational database, it is very flexible and can be deployed on all web hosts and cloud providers.

Software Requirement – Required to run my application is a standard internet browser. All features work fine and are tested on Internet Explorer (and Microsoft EDGE on windows 10), Google Chrome, Mozilla Firefox and Safari browsers on a MAC. Required to actually deploy my application is a reliable cloud host. If the site is down for any reason, users leave the website, so a server which can handle web traffic and also makes managing the application from an admin perspective is required.

2.1.3 Data requirements

Data stored by the system includes the first name, last name, username, password and email of the user who is registered to the system. There will also be image files held in the same database – These images coming from the when the user uploads a photo as their profile picture. There is also two Database tables which hold data relating to friend requests being sent out and received throughout the system as well as holding data about every friend a user has on the system in the 'friends' table. PHPMyAdmin makes all data stored on the site easily viewable, manageable and accessible to Admin accounts.

2.1.4 Data Protection and Security

Data being stored within a database relating to users must be held in a database which is reliable and secured with a password that is accessible to an admin only. My registration form that holds users information is well structured and passwords that are entered and stored in the database are encrypted as a form of data security. It also secured against SQL injection which is important to take into account for any data driven website which holds information in a DBMS.

2.1.5 User requirements

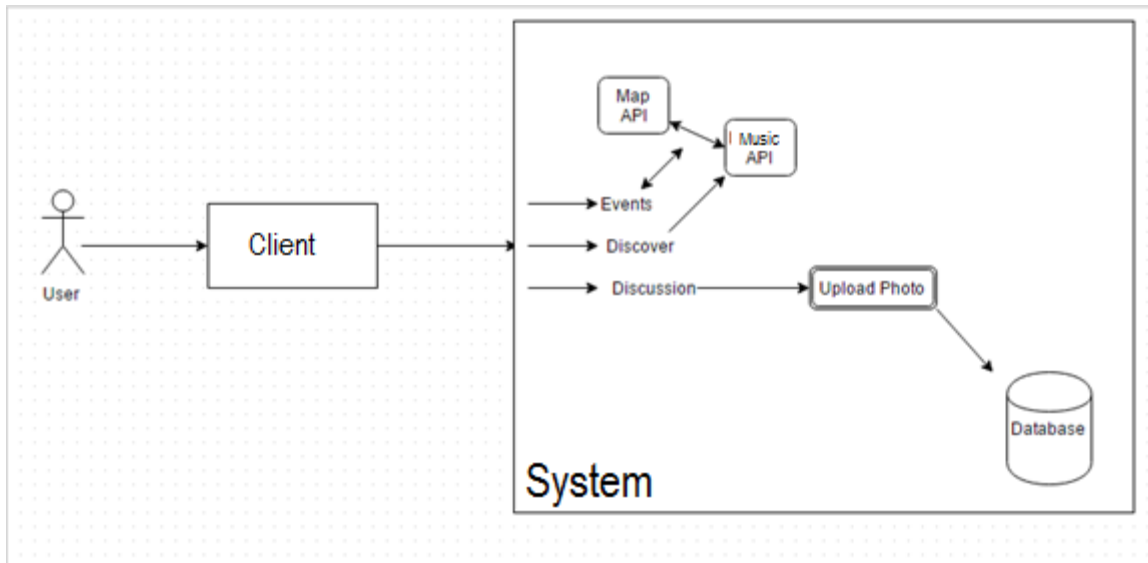
Upon arriving on the website for the first time, the user is prompted to fill out a straightforward registration form to register an account for themselves. They will then be able to proceed to the website to edit their information, upload a profile picture and bio, and access all the sites content on pages that were previously blocked prior to registration

2.1.6 Software requirements

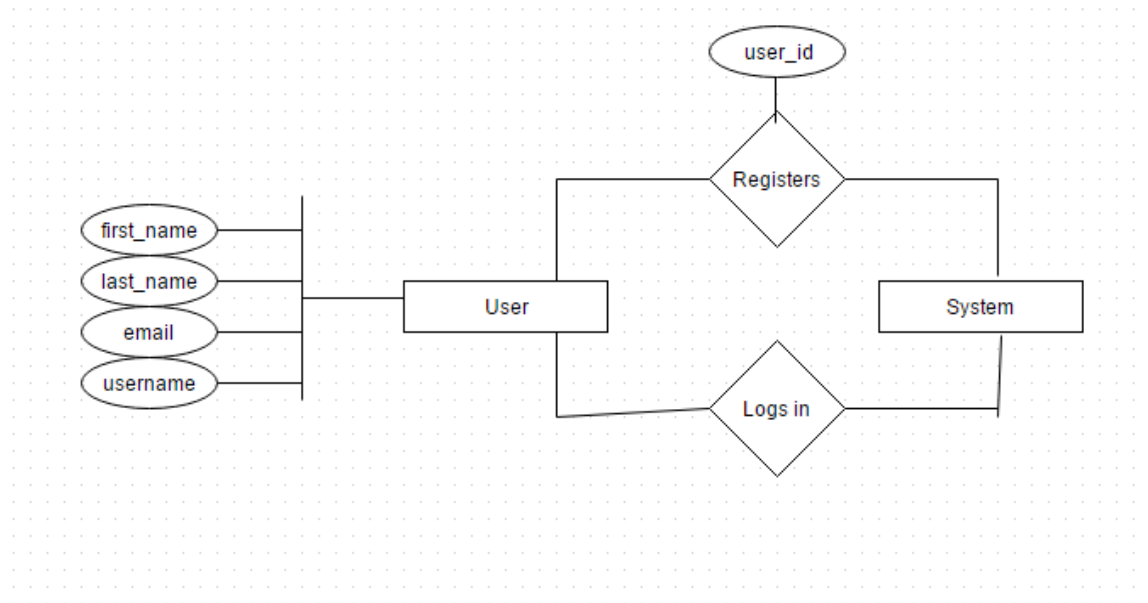
Required to run my application is a standard internet browser. All features work fine and are tested on Internet Explorer (and Microsoft EDGE on windows 10), Google Chrome, Mozilla Firefox and Safari browsers on a MAC.

2.2 Design and Architecture

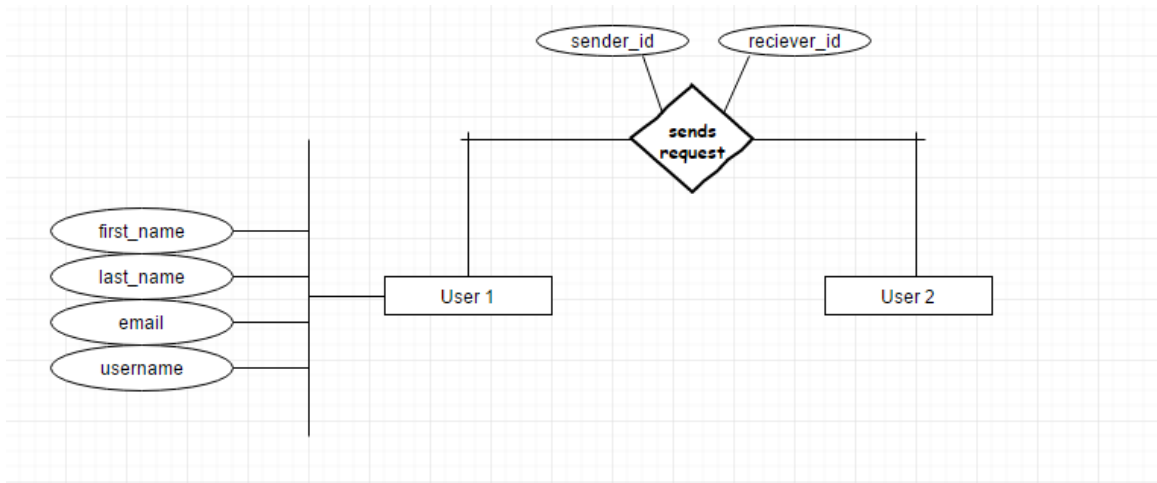
The following diagram outlines the basic **backend** architecture of the system. As discussed previously, the website is built up using PHP, and stored on a cloud host which will provide a database for all the necessary data to be stored on. The system architecture diagram also displays how the external API's interact with the system and what backend roll they have



This ERD diagram outlines the basic relationships between entities held in the database in relation to how a user gains access to the system.



This ERD diagram outlines the process for how a registered user with their details held in the 'users' database, sends a request to another user.



The **frontend** design of the website is developed using the bootstrap framework. One of my main priorities was to give the site a user friendly look that scales down with minimum effort on devices – this website is aimed at Musicians and people with music interests, not IT people, they will not be considering the design so simple, and easy to navigate around are the two key points here.

(See Wireframes in below section for GUI design of Site)

2.3 Testing

Testing proved to be a major part of my project towards the final stages and in the last few weeks and during development I carried out a number of different steps, methods, and tests as part of the process;

2.3.1 White Box Testing

Testing the internal workings of my software, where knowledge of how it works is actually known.

Unit Testing– As I developed my code, piece by piece I would do unit tests as I went along to ensure that each part was doing what it was supposed to do before moving on to the next part and just presuming it was working okay. For example I would echo out the contents of an Array on the page just to see everything was working and it was structured okay. When working with the API, I would consoleLog the information being pulled from the API to make sure that the call and the retrieval of Json was working okay

Integration Testing– Integration testing involved making sure that when two parts of the system came together that they both worked fine together and there were no errors or disagreements. For example, testing that the Google Maps was able to display markers okay when taking coordinates from the SongKick API. Or testing that the tree database tables (users, friend requests and friends table) could all work together to allow a user to add another.

Validation Testing– Involved checking that code I wrote actually did what I intended it to do, without causing any problems or errors. For example, when developing code to allow a user to store a profile picture, I could check my database and ensure it was actually being stored before developing further code to display it on the page. Or again, using the consoleLog function when using an API to ensure the API call was working as I intended it to.

2.3.2 Black Box Testing

Testing the internal workings of my software, where knowledge of how it works is actually known.

Unit Testing– Once I felt that writing code on a specific part was complete I tested it as an end user or client would. Example logging myself out, and in to the system after registering, adding and deleting friends, attempting searches and playing music. I tested on my desktop PC after coding, as well as on my mobile and my tablet device, and used different browsers each time.

2.3.3 UAT Testing

UAT Testing– User acceptance testing involved getting in touch with friends and family, getting them to register an account and navigate around my website to see if they had any problems, queries or feedback. In total, 16 potential users have tested the site and feedback has been mainly positive.

2.3.4 Customer Testing and Analysis

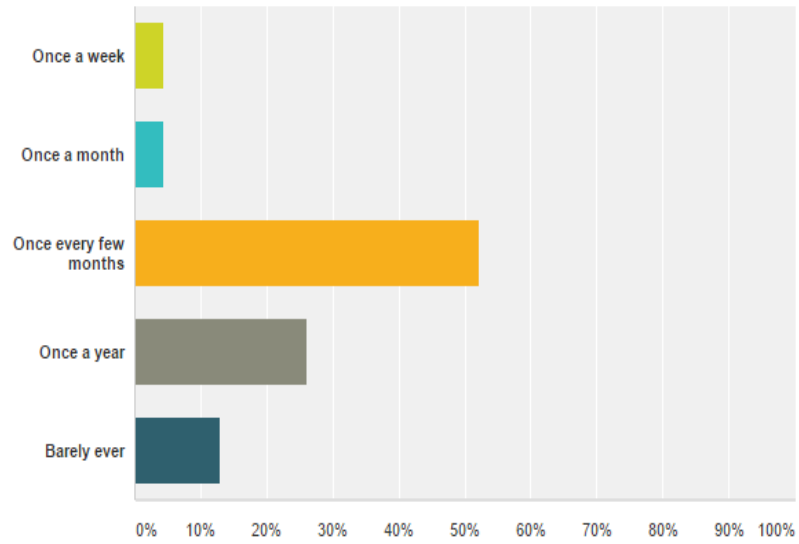
With a number of close friends being musicians, and a family member being in a well-known band around the country, I had people available to me from my target market to survey and discuss my application idea with. I created an online survey and sent it out to a number of friends and across my social media pages to get feedback and suggestions about the general idea and functions of my application. The survey can be found at.

<https://www.surveymonkey.com/r/8BQ8T3F>

I was quite pleased with the overall results and analysis of the survey as a social media website based completely around music seems generally like something people would like to use and that a majority of people who took the survey seem to think there is a gap in the market for this product. An analysis of results can be found below;

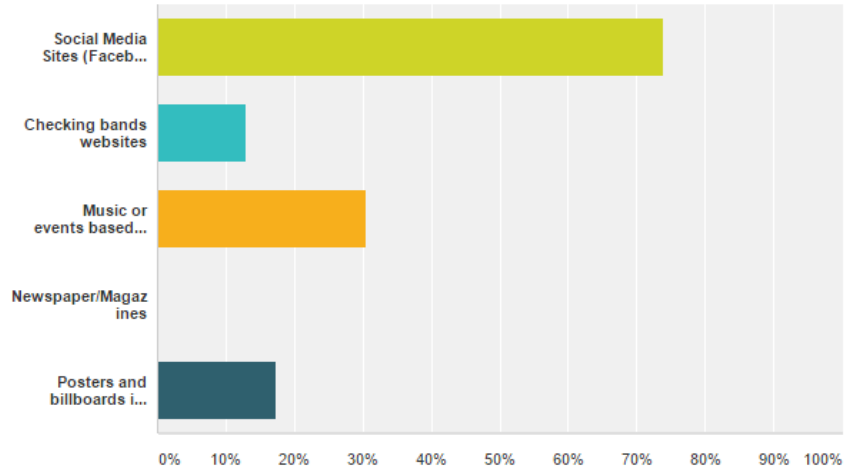
How often do you attend live music events (concerts, festivals, live music in pubs etc.)

Answered: 23 Skipped: 0



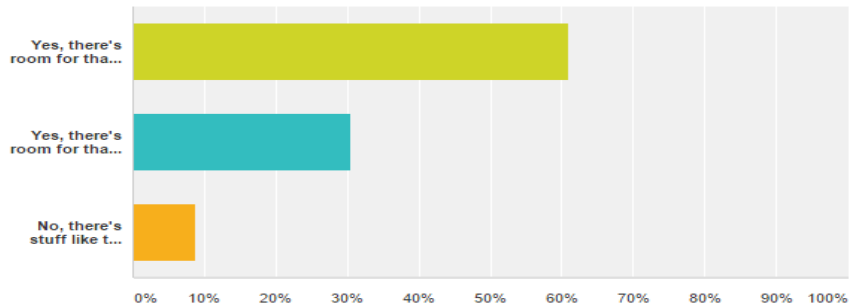
Where would you mainly get information from about upcoming music events in your area and how are you made aware of whos playing?

Answered: 23 Skipped: 0



Do you feel there is room for a new social media website based totally around music and would you use it?

Answered: 23 Skipped: 0



Answer Choices	Responses
Yes, there's room for that and I would use it	60.87% 14
Yes, there's room for that but I wouldn't be interested in using it	30.43% 7
No, there's stuff like that out there already (please provide examples if you can)	8.70% 2
Total	23

	I'd love this feature and would use it for sure	I'd like this feature and would use it sometimes	I could live without this	Pointless feature, would never use it.	Total Respondents
I want upcoming events converts and shows easily displayed to me from my area.	56.52% 13	43.48% 10	0.00% 0	0.00% 0	23
I want to make friends, meet people and chat online with people of similar music tastes	8.70% 2	21.74% 5	69.57% 16	0.00% 0	23
I want to promote my own music or promote my band	4.35% 1	34.78% 8	39.13% 9	21.74% 5	23
I want to be able to discover new music based on genres and artists I already like.	60.87% 14	34.78% 8	4.35% 1	0.00% 0	23
I want to be able to read reviews of my favourite artists recent shows and past albums.	43.48% 10	43.48% 10	17.39% 4	0.00% 0	23
I want to be able to input a location and be made aware of all bands and artists that originate from that area.	43.48% 10	34.78% 8	17.39% 4	4.35% 1	23

This survey was a great method of getting anonymous feedback about the general views and opinions of a handful of potential users in my target market. It allows me to realise what features potential users are really looking for if they were to use an application like mine so for further development I would know the main areas of my website to focus on.

2.4 Site Layout and Interface

Below contains a number of screenshots of how my finished website is looking.

Discover– The discover shows an artist’s bio, general info, similar artists and genres, as well as recent news, reviews and blogs published about the artist the user has searched

David Bowie

For other uses, see [David Bowie \(disambiguation\)](#).

David Robert Jones (8 January 1947 - 10 January 2016), known as David Bowie (/ˈboʊ.i/), was an English singer, songwriter, multi-instrumentalist, record producer, painter, and actor. He was a figure in popular music for over five decades, and was considered by critics and musicians as an innovator, particularly for his work in the 1970s. His career was marked by reinvention and visual presentation, and his music and stagecraft significantly impacted popular music. During his lifetime, he sold an estimated 140 million records worldwide. In the UK, he was awarded nine platinum album certifications, eleven gold and eight silver, and released ten number-one albums. In the US, he received five platinum and seven gold certifications. He was inducted into the Rock and Roll Hall of Fame in 1996. Born and raised in South London, Bowie developed an interest in music as a child, eventually studying art, music and design before embarking on a professional career as a musician in 1963. "Space Oddity" became his first top-five entry on the UK Singles Chart after its release in July 1969. After a period of experimentation, he re-emerged in 1972 during the glam rock era with his flamboyant and androgynous alter ego Ziggy Stardust. The character was spearheaded by the success of his single "Staman" and album *The Rise and Fall of Ziggy Stardust and the Spiders from Mars*, and it won him widespread popularity. [via wikipedia](#)

General Info

Years Active	1962-2016
Location	London, UK
Sites and Social Medias;	twitter mb lastfm official myspace

Music Lounge Stats

Type	Count	Type	Count
reviews	358	biographies	11
video	3641	images	153
news	4971	blogs	21927
audio	153	songs	1244

Similar Artists

[Roxy Music](#), [Bryan Ferry](#), [Tin Machine](#), [T. Rex](#), [Ziggy Stardust](#), [Lou Reed](#), [Sparks](#), [Ultravox](#), [Gary Numan](#), [Japan](#), [Iggy Pop](#), [Adam Ant](#), [Marc Bolan](#), [The Stranglers](#), [Mick Ronson](#), [Adam and the Ants](#), [Soft Cell](#), [Blondie](#), [The B-52's](#), [Steve Harley & Cockney Rebel](#), [The Velvet Underground](#), [Simple Minds](#), [Talking Heads](#), [Talk Talk](#), [The The](#), [The Psychedelic Furs](#), [New Order](#), [Orchestral Manoeuvres in the Dark](#), [Jobriath](#), [Sigue Sigue Sputnik](#)

Top Genres

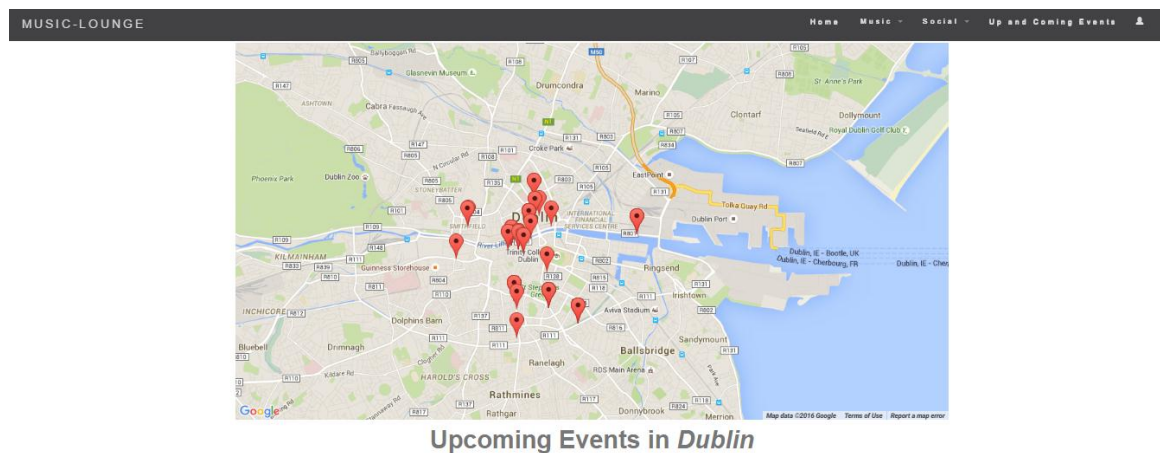
[glam rock](#), [art rock](#), [permanent wave](#), [protopunk](#), [rock](#), [pop christmas](#)

Recent News

Recent Reviews

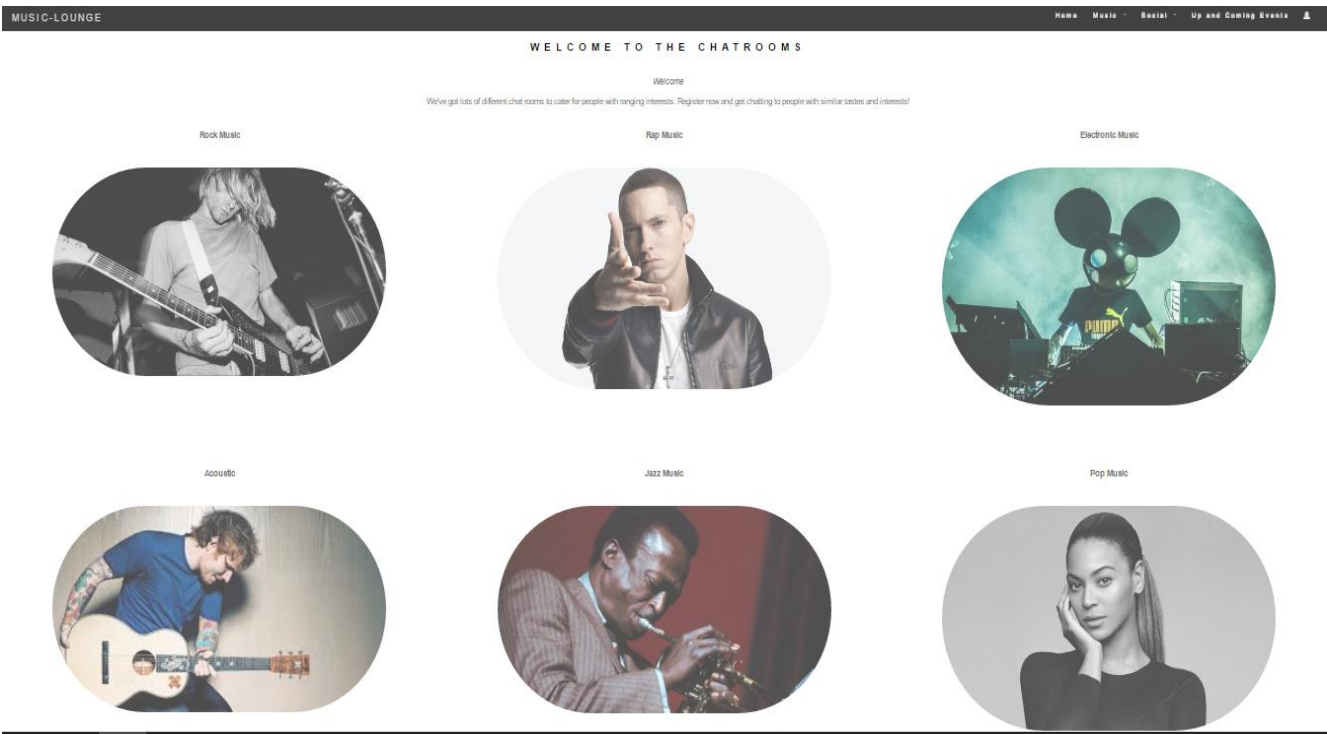
Recent Blogs

Events. – The events page displays a google map with all upcoming shows and music events in Dublin that are being pulled from the SongKick API. Latitude and Longitude is being taken from the JSON data and being placed on the GoogleMap.

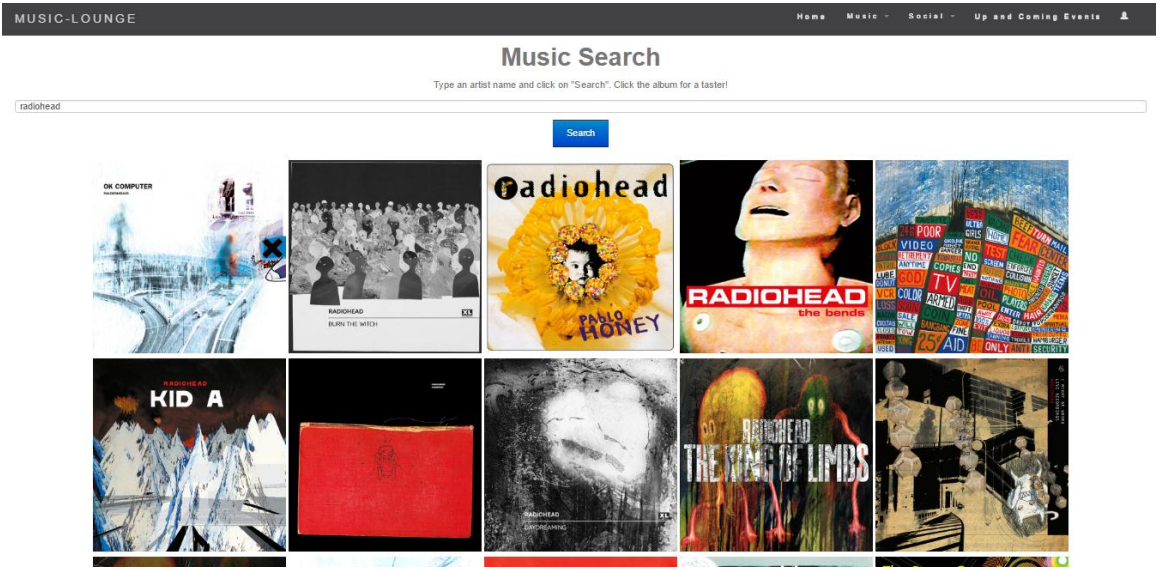


Environment

Chat Rooms - The chat rooms page is the area where users using the site can come together, chat with other users of similar music interests and make friends and discussion take place. Chat rooms are divided into different genres so like minded people with the same interests can converse in the one environment.



Music Search – This is the page where users can search for an artist or band of their choice and be returned music and album art work of the album. When the users clicks the artwork, it will be highlighted and a 30 second preview of that music will be played back to the user.



Pick of the Week- When this project is complete I have every intention of using it as a place where I can review albums or bands myself in my own spare time. The pick of the week page is my area where i'll write a bio about the band, review an old or new album they've released, and display a sample of their music for a user reading it to demo. I was intending on starting a music blog, but with my website now available to me, this is the area i'm going to use for that.

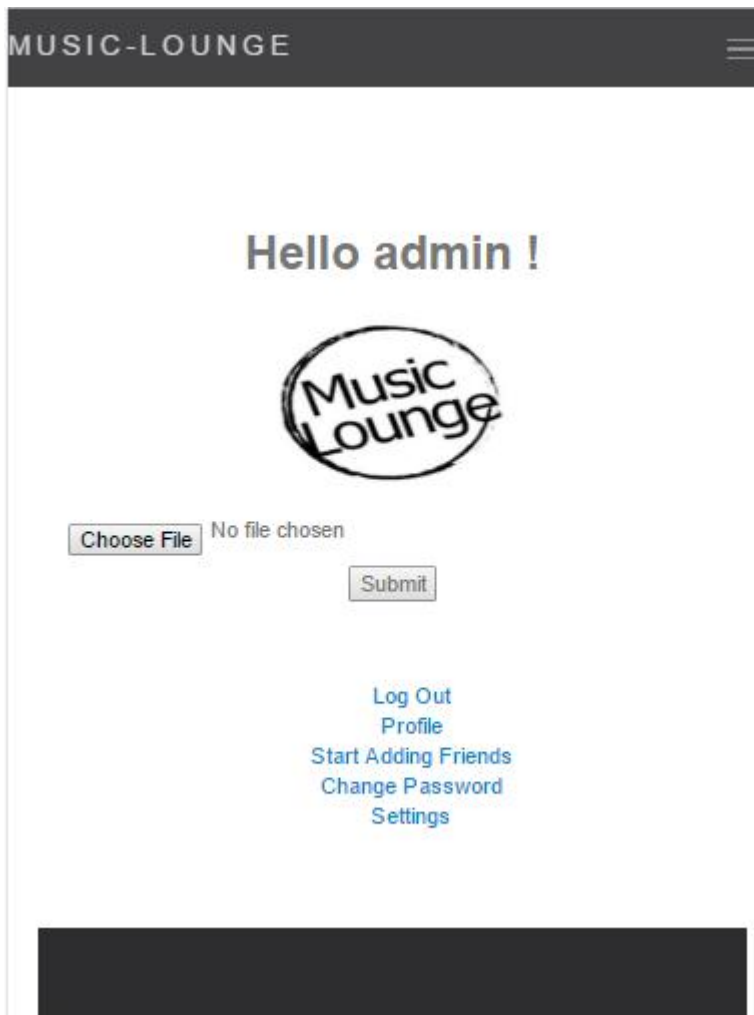
Pick for this week;

B e l l - x 1

Bell X1 are a music group from Dublin, Ireland, known for their wide range of styles, powerful live performances, intelligent and witty lyrics and a dedication to touring. NPR says they deliver "a brilliant co-mingling of electronic music and anthemic pop rock".^[2] Bell X1 are festival and arena headliners in their native Ireland and play to ever growing numbers on their regular North American and European tours. Aside from U2, they are the Irish band with the most airplay in their native country and, according to Billboard, also the second biggest live performers



On Mobile- Thanks to the bootstrap framework being implemented, i'm very happy with how my website appears on mobile and tablet.



3 Conclusions

I believe my project has a lot of potential to attract a wide number of users and become a really active site because it is aimed at such a large target audience – A social media type website focusing on only music is not really existent on the web and there may be a potential gap in the market for this type of thing. There is also no end to how far this idea could be expanded, worked and continued on, and there is huge room for improvement and to add even more features.

In saying that, I do realise that the idea of the project might not be the most original, and at the end of the day one of the main reasons I decided to go with it is because of how flexible it is and how much resources are available to me with it. For example – if One API stops working or is no longer supported, there are many other alternatives for me to fall back on. If I can develop the site to the standards I'm hoping for it's a project I could be potentially take great pride in once finished.

4 Further development or research

One big advantage of this project is its flexibility – there's no end to how far this project could be extended and improved on. There could be a gap for registered users to receive online guitar lessons over video streaming. There could also be a gap for up and coming artists using it to promote themselves by uploading mp3 files of music they have made themselves available as a free download. And if the site was to ever get very big, known musicians could hold Q&A's online, which would encourage more users to register and join the community

5 References

- Code Course. (2012). PHP Tutorials: Register & Login. [Online Video Playlist]. Available from:
<https://www.youtube.com/playlist?list=PLE134D877783367C7>

(Followed tutorial series for login and registration, building up of backend)

- GitHub. 2016. GitHub - echonest/echotron: From the EchoNest API demos page. [ONLINE] Available at: <https://github.com/echonest/Echotron>

(Source Code taken from EchoNest Demos page. Code had exact functionality I was looking for on discover page, found when researching music API's)

- TVAC Studio. (2013). PHP Tutorial - Friend System. [Online Video]. 1 March 2013. Available from:
<https://www.youtube.com/playlist?list=PLGCjwl1RrtcQa44MxSsfP6edUP2Gimhkk>

(Followed tutorial series which helped build my friend system)

- Stackoverflow. 2012. How to parse json data with jquery / javascript. [ONLINE] Available at: <http://stackoverflow.com/questions/8951810/how-to-parse-json-data-with-jquery-javascript>.

(Referenced this page when working with JSON and JQuery)

- iviewsource. 2012. JSON using JavaScript and jQuery. [ONLINE] Available at: <http://iviewsource.com/codingtutorials/getting-started-with-javascript-object-notation-json-for-absolute-beginners/>

(Referenced this page when working with JSON and JQuery)

- Google Developers. 2016. Importing data into google maps. [ONLINE] Available at:
https://developers.google.com/maps/documentation/javascript/tutorials/importing_data#overview.

- Spotify Web API. 2016. Web API - Examples & Libraries. [ONLINE] Available at: <https://developer.spotify.com/web-api/code-examples/>. [Accessed 17 May 2016]

(Used source code and code examples for Music Search page and for displaying an artist's music on the Pick Of The Week page)

- EchoNest 2013. EchoNest Demo Page. [ONLINE] Available at: <http://static.echonest.com/labs/demo.html>. [Accessed 17 May 2016].

(Used source code and code examples from the EchoNest Demos page for artist location search and discover page)

- w3Schools. 2012. Bootstrap Theme. [ONLINE] Available at: http://www.w3schools.com/bootstrap/bootstrap_theme_band.asp.

(Used this page when creating my bootstrap theme and went researching templates)

- TimKippTutorials. (2011). PHP Series - Building A PHP MySQL Forum Tutorial Series. [Online Video]. Available from: <https://www.youtube.com/playlist?list=PLEEE2B57BAE393EF5>.

(Used this tutorial series when looking into building a PHP forum)

- Code Course. (2011). File based image gallery. [Online Video]. 6 February 2014. Available from: <https://www.youtube.com/playlist?list=PLfdtitiRHWG0V4sCGR5a4WmNFgxNcYin>.

(Used this tutorial series when looking into building a nicely styled image gallery)

6 Appendix

6.1 *Project Proposal*



Fourth year project
- proposal.docx

6.2 *Monthly Journals*

Contained below are my monthly journals that I wrote up until this point, each has been uploaded to moodle.



ReflectiveJournal -
Month 1 - CiaranBrc



Reflective Journal -
Month 2 - CiaranBrc



Reflective Journal -
Month 3 - CiaranBrc



Reflective Journal -
Month 4 - CiaranBrc



Reflective Journal -
Month 5 - CiaranBrc



Reflective Journal -
Month 6 - CiaranBrc



Reflective Journal -
Month 7 - CiaranBrc

