**1. Introduction**

**Small breakdown about the application**E-mu is an emotion based music application that allows users to select playlists based on the emotion they want to feel/are feeling. The purpose of this application is give users the opportunity to listen to relevant music to evoke certain emotions that they may be feeling or that they want to be feeling.

**What makes it unique**

It is a unique application that focuses on categorising music into unusual playlists focused around emotions. This can help set the mood for specific occasions for example: social gatherings, intimate meetings or relaxing individual moments.

**Purpose and why it was made**

This application was created to be the first music application solely based on emotions instead of the typical genre of music. By not restricting the playlists to the common categories such as: genres, decades, artists and seasonal it gives the user freedom to listen to multiple genres and artists based on their personal feelings

**Choosing a Framework**

**Using PhoneGap to allow distribution on IOS and Android as apposed to a native app.**

There are two options to choose between when developing a mobile application, create a native application for either Android or iOS, or create a hybrid application that will be accessible on both. Creating a native application means using the native language for the platform, Objective-C on iOS and Java on Android. The main benefits of creating a native application are the specifically catered designs, depending on the platform and improved performance. Native apps get compiled into machine code, allowing them to perform faster and more efficient [[1]](#footnote-1). A hybrid application is developed using common web code (HTML, CSS and JavaScript) to allow fast development times and easy deployment across multiple platforms. When choosing my method to develop a mobile application the ability to not be restricted to one platform (iOS or Android) was very appealing. In addition to this my current skillset as a front-end developer gave me a good understanding of the tools needed to create a hybrid application. Although the performance will not match that of a native application, the ability to distribute across all platforms and improve my current skillset is the reason why I chose to create a hybrid application.

When selecting a framework to build hybrid mobile applications it was important to select one that meets all of the requirements needed for the scope of the project. The three frameworks I analysed for this project were: Adobe PhoneGap [[2]](#footnote-2), Ionic [[3]](#footnote-3) and React Native [[4]](#footnote-4). PhoneGap is a package framework used for releasing applications. It is based on the open source framework Cordova[[5]](#footnote-5), and uses HTML, CSS and JavaScript. Ionic is a hybrid framework that allows the user to also build and package applications using common web languages, such as: HTML, CSS and JavaScript. React Native was originally only developed to support iOS. However recently Facebook announced it was supported and optimised for both iOS and Android.[[6]](#footnote-6) React Native uses JavaScript, CSS elements and Reach. React is a JavaScript library used for building user interfaces. From the research acquired on these three frameworks I gained a better understanding of the level of skill needed for each one. React Native requires intermediate/advanced knowledge in JavaScript and React. Ionic and PhoneGap require intermediate knowledge in HTML, CSS and JavaScript. With PhoneGaps extensive documentation and it’s renowned accessibility to developers with a front-end skillset, it was the correct framework to use for this project.

**2. Research**

**Setting up PhoneGap**

**PhoneGap desktop application**

The PhoneGap desktop application provides a simpler alternative to those who are not confident working on the command line. It is a visual user interface that requires simple tasks to get your first PhoneGap application setup and running. When initially testing my PhoneGap app with the desktop application the setup was fast and simple. However not long after the setup, the application continuously crashed and irregularly updated when prompted to. This is because the application is still in its beta stages.

**PhoneGap CLI**

The command line interface (CLI) states that it is the recommended platform to use if you are comfortable using the command line. Currently the CLI has additional features for building, running and packaging your application that are not yet available with the desktop application. I was not too familiar with the command line prior to this project. However the step-by-step install instructions are easy to follow and once it is setup, you maintain it with a single line of writing each time you begin working. I decided to continue to use the CLI because it took less processing power to run, it is faster to use, it is a good platform to learn and as stated before you gain additional features not available on the desktop application.

**Psychology of colours**

E-mu, which stands for emotional music, is a mobile music application that focuses on the users emotions. When designing the colour scheme for the application it was apparent that colours are widely used to portray emotion. A colours meaning, interpretation and perception are widely varied between different cultures. This makes it difficult to ensure every individual will feel the same emotion when shown a certain colour. The psychology behind colours can be very subjective, however there are proven links to colours and emotions that will be shared among a large percentage of humans. [[7]](#footnote-7)

When selecting the colours for E-mu’s emotion playlists it was important to try and cater for the largest audience possible, this meant finding widely acknowledged emotional colours. By combining colours that across cultures represent the same emotion, can improve the emotional connection towards a colour for a greater number of users.[[8]](#footnote-8)

**User tests**

A user test was created to evaluate the usability of my application to check for any inconsistencies or errors. Each participant was asked to fill in a consent form (see Appendix 2 for the consent form), a demographic questionnaire (see Appendix 3 for the demographic questionnaire) and undertake a feedback session after the test was completed. The users were asked to speak their mind during the analysis of the application, to help gain a better understanding of the thought process behind their decisions. This would help potentially uncover consistency errors the user might expect during the test.

In result the information gained from the user test was informative and helped improve my application’s consistency, usability, accessibility and overall success. By correcting the errors discovered it helped ensure that my application was at a high standard and ready to be packaged and launched *(See appendix 1 for the full user test).*

**3. Design**

**3.1 Colour scheme**

When choosing the colour scheme for my application it was important to use colours that translate an emotion to a large audience. By researching about the most commonly used colours to portray emotions, it helped me potentially engage with the largest possible audience possible.

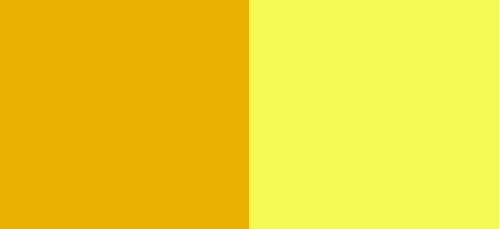
**Anger**

The most common colours used globally to represent anger are: red, black and yellow. In the East, red is used to represent happiness and power, black is the colour commonly used to represent anger and evilness.[[9]](#footnote-9) The combination of red and yellow are more globally recognised as anger. I used black/red and vibrant red together in a radial gradient to create a dark and menacing background.



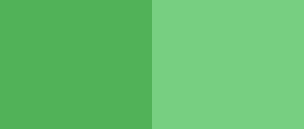
**Joy**

White, yellow and orange are the most commonly used colours to represent the emotion happiness. In the East, white represents death and sadness, whereas orange signifies happiness. The combination of yellow and orange in a radial gradient makes this background seem calm and joyful.

****

**Serenity**

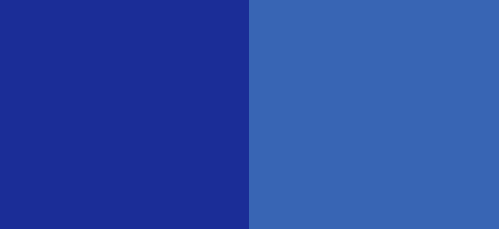
The colour green traditionally represents nature, stability and prosperousness. It is generally considered the most balanced colour, being between the warm colours (red, orange, yellow) and the cold colours (blue and purple), making it the stable, neutral and serene colour in the spectrum.

****

**Sadness**

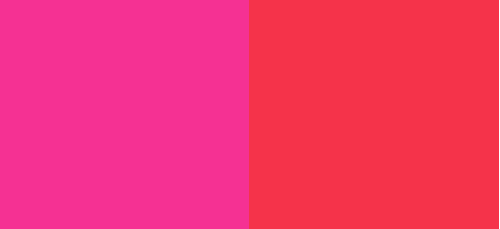
The colours most frequently used for this emotion are blue, purple and white. In the East, blue is used to represent life and immortality. It also is commonly regarded as a feminine colour. Purple is used to signify sorrow and mourning.

The mix of dark blue and light blue in a radial gradient gives the feeling of sombreness and being alone.

****

**Love**

Pink and red are quite globally regarded as colours to represent passion and love. In South Africa, the colour red is used to signify mourning.[[10]](#footnote-10) The combination of red and pink in this radial gradient gives the background a warm and loving feeling.

****

**3.2 Typography**

When choosing the fonts for each emotion in my application, it was important to portray each emotion through the corresponding typography. Once the font was selected it would then be converted into a web-safe font using font squirrels web-font generator [[11]](#footnote-11). The web-font generator allows you to convert standard font file types (.ttf, .eot, .woff) into web-safe fonts and produces the relevant CSS code block to allow for easy use. You must ensure the fonts used are eligible to be converted before doing so.

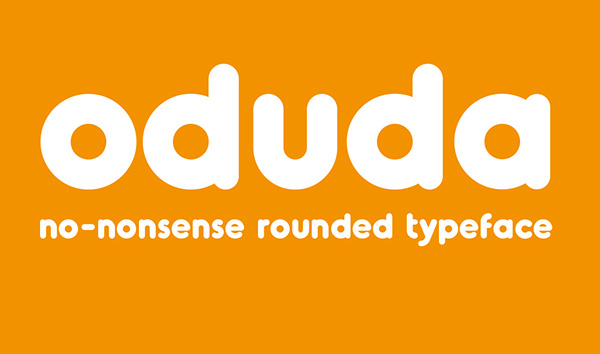
**Anger**

****

Helv children was the font used for the emotion anger. It is a large, bold and aggressive font that when combined with the deep red coloured background conveyed a feeling of anger.

**Joy**

Oduda is the font used for the emotion joy. Oduda is a large, bold, rounded and friendly font that gives the user a feeling of joy when complementing and yellow background.

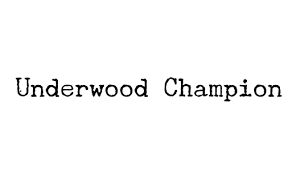
****

**Relaxation**

Moon is the font used for the emotion relaxation. Moon is a clean, thin and simplistic typography that flows well and allows the user to feel comfortable and calm with the security of not having any intrusive text.

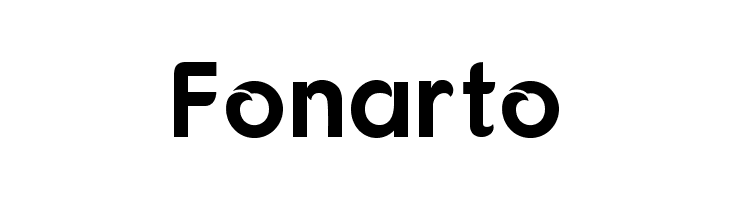
****

**Sadness**

****

Underwood champion is the text used for the emotion sadness. It is a

**Love**

****

Fonarto is the text used for the emotion sadness.

**3.3 Icons**

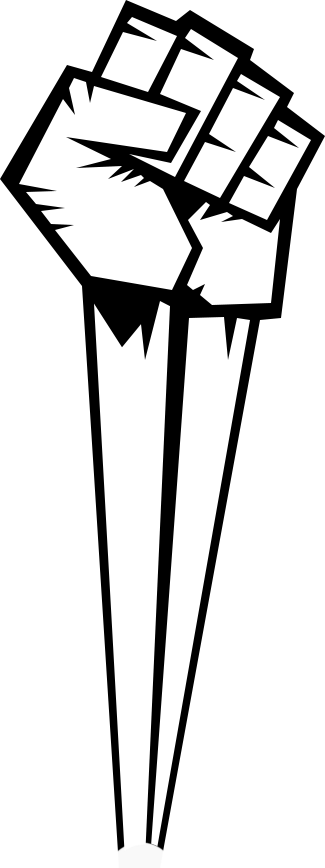
I created all of the bespoke icons in my application. I did this using Adobe Illustrator[[12]](#footnote-12) and Adobe Photoshop[[13]](#footnote-13). Each emotion icon is designed to be a visual representation of the relevant feeling.

**Logo**I chose to illustrate a vector image of an E-mu bird as a memorable and humorous play on words for my application E-mu (Emotional Music). The E-mu’s colourful hair incorporated all of the colour elements within the application to unify the colour emotion theme used throughout.

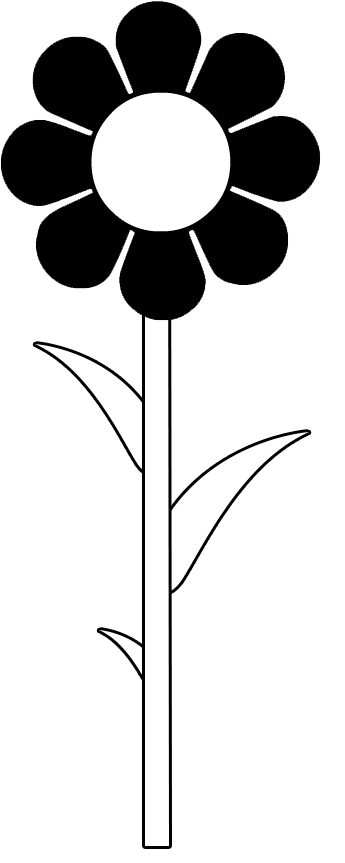
****

**Anger**

The icon used is a clenched fist as anger stereotypically connotes violence and aggression. The fist symbol can also signify rebellion and power.

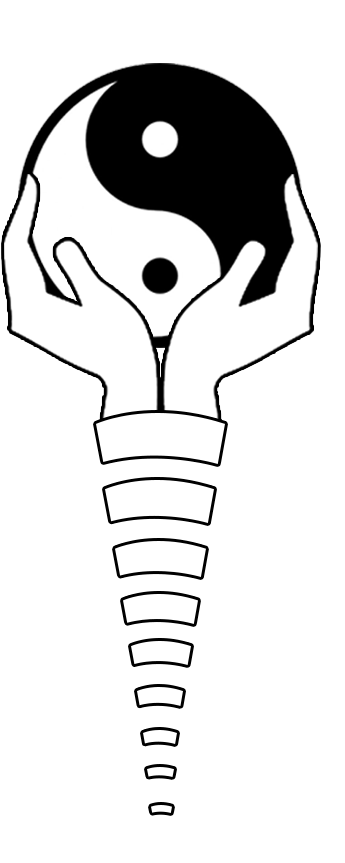
****

**Joy**I chose to use a flower to represent the emotion of joy. Flowers are gifts given on special occasions to evoke happiness and joy to the recipient. Stereotypically joy is also associated with summer, when flowers commonly bloom.

****

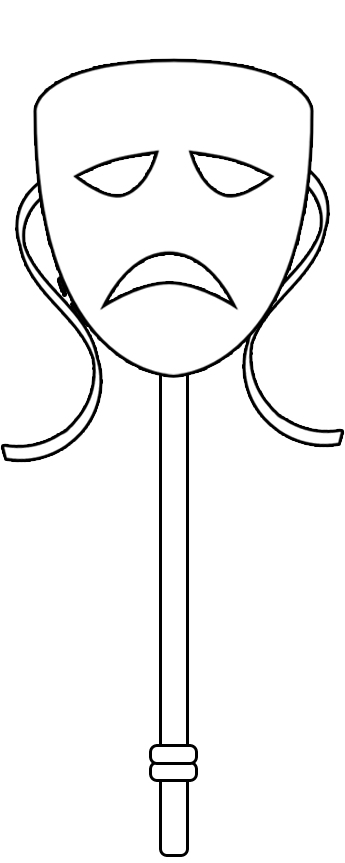
**Serenity**

I selected the ying-yang symbol as the main focus for my serenity icon. This symbol represents calm energy, peace and unity. The flow of energy at the bottom of the icon reminded me of peaceful practises such as yoga and tai chi.

****

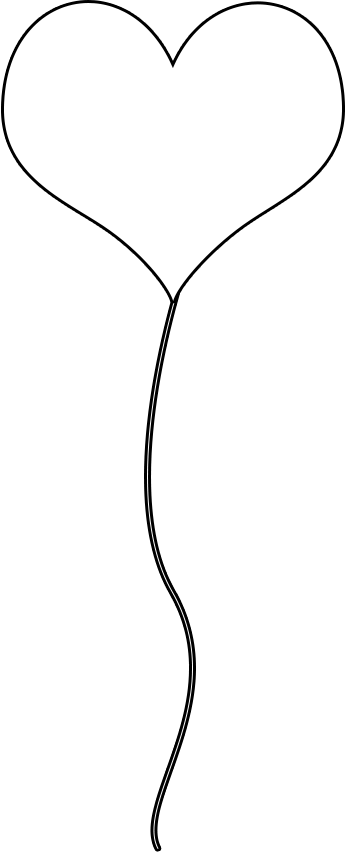
**Sadness**

The tragedy mask commonly used in theatre inspired the icon I created for sadness. It is widely recognised to represent sorrow and despair through a frowning mask.

****

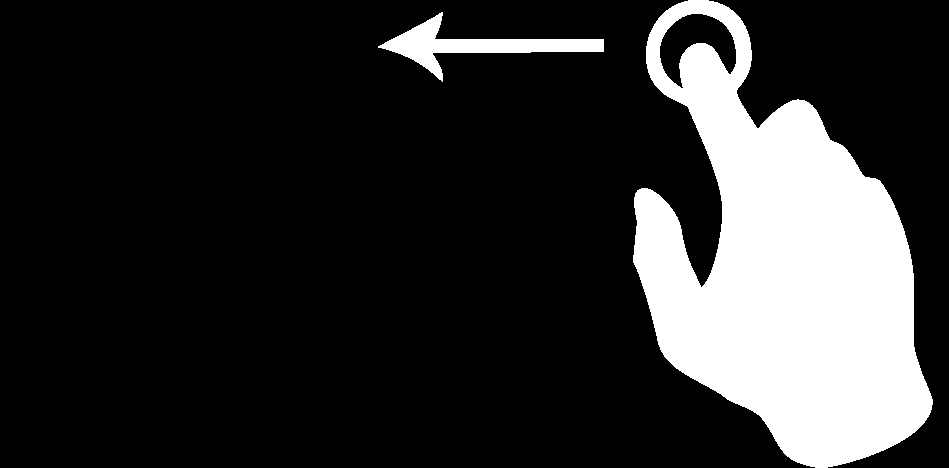
**Love**

I decided to focus my love icon around the traditional symbol to connote love, a heart. To give this icon a unique and personal feel, I decided to illustrate the heart as a balloon.

****

**Swipe gif**

I created a condensed gif image sliding from right to left on the homepage, to help users know how to navigate through my application. This was combined with text to ensure the user made no errors.

****

**Offline**If the user’s internet disconnects during the usage of the application, an offline screen would be displayed informing the user how to continue. The icon used for this screen is a commonly recognised icon for internet connection**.**

****

**3.4 Final page designs**

**4. Development**

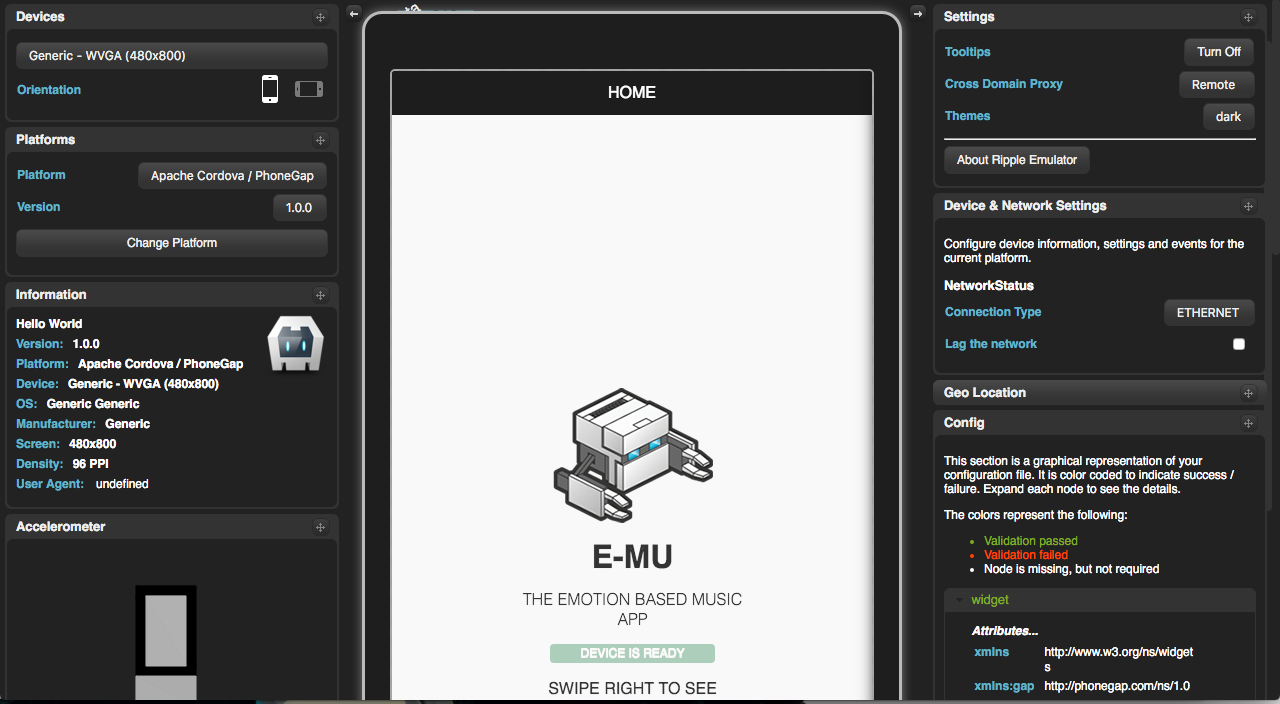
**Debugging the application in real-time**

**Weinre**

Weinre was the tool chosen to debug the mobile application. It is used as a remote web inspector to allow the developer to remotely debug a mobile app through their computer browser.[[14]](#footnote-14) This method of debugging did not allow stepping through with the source files. It was primarily used to analyse console logs and overall styling errors that occurred on the mobile device, but not within the virtual browser.

**Ripple**

Ripple is an open source debugger extension for Google Chrome. It allows the user to test their mobile application in the browser by emulating a mobile device. You can view your mobile application in the browser and fully inspect it using any standard web inspector. Ripple has an informative interface to allow users to get the best experience out of Ripple possible. The emulator treats the mobile application as a standard web application, allowing the developer to debug as thorough as they would a web application.



**Adding touch sensitive swipe gesture to app.**

Jquery mobile was implemented to allow touch sensitive gestures to be used throughout the mobile application. Adding touch sensitive swipe gestures allowed for streamline navigation throughout the application.

**Defensive programming**

Defensive programming is the method used to help ensure your software functions correctly in spite of unforeseeable usage of said software.[[15]](#footnote-15) In the event of a playlist becoming dynamic and allowing users to add to it, a countermeasure must have been implemented to ensure the playlist won’t break due to the number of songs added. This was achieved by adding large testing playlists to the application, to see the point it will begin functioning incorrectly. The maximum number of songs SoundCloud enable users to upload into a playlist is 500. Playlists are treated differently to groups when displayed on mobile. A group can hold thousands of songs, however due to this SoundCloud have limited mobile applications to only able to display 50 songs. I tested a playlist with 500 songs to ensure my app would run correctly if it reached the maximum capacity. The initial loading time was dramatically increased when accessing the 500-song playlist. To prevent this from being an issue for users I made each image get cached in the users local storage to speed up the following times the user accessed the playlists. Displaying a loading icon until the page is fully loaded allows the users experience between transitions smoother and more enjoyable.

Another potential section of E-mu that in time would inevitably evolve are the list of emotions that link to the individual playlists. By copy and pasting the existing emotions I was able to dramatically increase the amount of potential emotions in my application. I tested with 100 emotion pages. The fluidity of E-mu dramatically decreased when cycling through the emotions. I also noticed when swiping it had the tendency to skip emotions and have a reduced frame rate for the animations, this was due to the massive increase in file size. This test helped show the amount of emotions my application could hold before hindering the performance. This will allow countermeasures to be used if the app were to evolve to that stage.

**Offline screen**

E-mu is an application that needs the user to be online for the whole during of its use. Due to this it was imperative to display information to the user if their internet connection was not connected. This was achieved by hiding all application elements and displaying a message once the users internet connection was lost. Once the users internet connection is turned on the message will disappear and the applications interface will return. A JavaScript event listener was used to indentify the users internet connection, it then calls a function you set up to manipulate the application dependant on the internet being on or off.

**Caching images**

To help improve the users experience on the application, image caching was used to help save time when loading previously visited pages. Imgcache.js is a JavaScript library used to locally store images from an application and display them if the image fails to load or takes too long.[[16]](#footnote-16) Imgcache.js can be used for displaying images offline that the user has previous seen. However the user is unable to interact with this application once the internet has been disconnected. Instead Imgcache.js was primarily used to save load the images faster due to slow internet or errors.

**API**

**Reason for not using Deezer AND Reason for choosing Soundcloud SDK**

After further research I decided against using Deezer as my application API. Deezer uses a premium account service that gives the user benefits when paying a monthly subscription. Without a subscription your songs intermittently have adverts and the playlist functions are heavily limited. This ruined the flow of my application and is the main reason I decided to change from Deezer to Soundclouds API. Soundcloud also uses subscription deals to improve the users experience, however the only side effect was certain songs being limited to a premium user. This is a problem that will arise once my application begins to evolve to allow users to add their own songs. At this stage in the project it did not affect the playlists, or API because there were still hundreds of songs that were free to use.

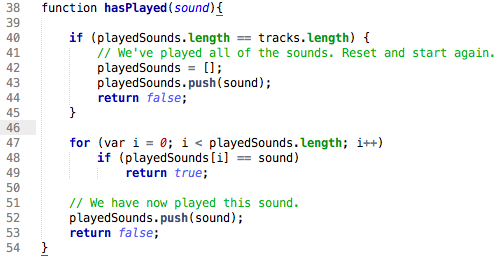
**Using get request to access soundcloud playlists and select the tracks**

Before using a custom Soundcloud player I used the in-built player to display each playlists. This was achieved by using a get request to access the online playlists and then display them using a Soundcloud embed (SC.oEmbed). The tracks were sorted into two arrays, played and not played. This helped randomly select a song without the risk of repeating it for the user.

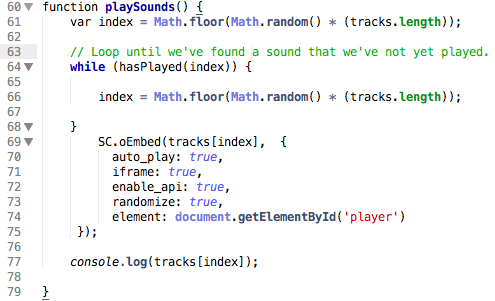
**Get request**

****

**Check if the songs have been played**

****

**Select a song that hasn’t been played and display it with the Soundcloud embed.**

****

**Why I decided to use a custom player and go against sc.oembed.**

Due to this method not allowing the developer to customise the player, along with all of the intrusive content on Soundcloud (section comments, artists page link, song page link etc), I decided to use a custom player that would allow me complete freedom during the development. The custom Soundcloud player is a jQuery plugin that allows the developer to fully customise each aspect to fit their chosen application/website.[[17]](#footnote-17) Each player is individually designed to fit the corresponding emotion playlist. This gives the user a clear visual indication of the playlist he/she is currently viewing. Keeping the player minimal to show: artist, song title, song duration and interactive scrubber (to allow the user to skips through the song) gives the user the main informative content of each track and the ability to skip through the current track or to another. This makes the Soundcloud player user friendly but also retaining the necessary information to fulfil the users needs.

**5 Conclusion**

**5.1 Evaluation of my application**

To conclude, I found the experience extremely beneficial as I gained a lot of knowledge and experience creating a hybrid application with Adobe PhoneGap. I analysed interesting topics such as: the psychology behind colours, the importance of a minimum viable product, defensive programming and the usability for mobile application. During the development I became confident using the CLI (command line interface), HTML CSS and JavaScript. I gained a broad understanding in API’s and SDK’s. My real-time debugging skills dramatically improved along with my knowledge in user-friendly design.

Throughout the process of creating my application I discovered certain aspects that I would have altered if I had the prior knowledge and experience I now have at the end of making my application. I would have selected a different framework to create my application in. PhoneGap is a an easy and fast framework to understand, however you are limited with your development, the debugging tools are very temperamental and code is very bloated, resulting in slow loading times. From the knowledge gained from this project I feel confident to use React Native as my next hybrid mobile framework. I would have conducted two user tests instead of one, as this would have given me a better understanding on what to focus on. I would have liked to user test my application at a very early stage with the bare minimum features to see what parts specifically my audience liked and disliked. This would have been showing my applications minimum viable product to hear feedback about the features I should remove and continue with. By doing this it would have allowed me to focus on the important aspects of my application and cater it specifically for my audience, without wasting time developing unnecessary features.

**TRY AND THINK OF SOME FEATURES I WOULD HAVE DONE DIFFERENTLY!**

**5.2 The future of my application**

There are many ways to improve and expand the features of my application in the future. Including a login system would increase the sophistication and personalisation of E-mu. With a login system users would be able to create their own playlists for each emotion and share it amongst friends and other users. They will also be able to like and dislike songs/playlists to improve the overall reputation/ranking of playlists, displaying the highest rated playlists to users. The creation of global playlists will give users the chance to add their favourite songs to a playlist that is populated by all users on E-mu. By linking E-mu to social media it has the potential to reach a larger target audience through users sharing personalised playlists or inviting other users to join.

**REMEMBER TO REMOVE THE WEINRE TESTING LINE OF SCRIPT AS IT WILL FUCK UP TESTING!**

**Bibliography**

Adobe PhoneGap. (2016). *Debugging.* Available: http://docs.phonegap.com/references/developer-app/debugging/. Last accessed 24th Feb 2016.

Art Therapy. (2016). *Color Psychology: The Emotional Effects of Colors.* Available: http://www.arttherapyblog.com/online/color-psychology-psychologica-effects-of-colors/#.VvFb9hKLSRs. Last accessed 22nd March 2016.

Colour affects. (2008). *Psychological Properties Of Colours.* Available: http://www.colour-affects.co.uk/psychological-properties-of-colours. Last accessed 22nd March 2016.

Changing minds. (2008). *The Meaning of Colors.* Available: http://changingminds.org/disciplines/communication/color\_effect.htm. Last accessed 22nd March 2016.

Font squirrel. (2016). *Webfont generator.* Available: https://www.fontsquirrel.com/tools/webfont-generator. Last accessed 3rd April 2016.

Dorman, S. (2008). *What is “Defensive Programming”?.* Available: https://scottdorman.github.io/2008/07/04/what-is-ldquodefensive-programmingrdquo/. Last accessed 13th May 2016.

Ben, C. (2016). *imgcache.js.* Available: https://github.com/chrisben/imgcache.js/. Last accessed 14th May 2016.

Fisher, N. (2016). *Soundcloud custom player.* Available: https://github.com/soundcloud/soundcloud-custom-player. Last accessed 16th May 2016.

Eric Ries (2011). *The Lean Startup*. USA: Crown Business (USA). p1-336.

Adobe. (2016). *Build amazing mobile apps powered by open web tech..*Available: http://phonegap.com/. Last accessed 17th May 2016.

Ionic. (2016). *Create incredible apps.* Available: http://ionicframework.com/. Last accessed 17th May 2016.

Facebook. (2016). *A FRAMEWORK FOR BUILDING NATIVE APPS USING REACT.* Available: https://facebook.github.io/react-native/. Last accessed 17th May 2016.

Wodehouse, C. (2015). *7 Reasons Why Facebook’s React Native Is the Future of Hybrid App Development.* Available: https://www.upwork.com/hiring/mobile/react-native-hybrid-app-development/. Last accessed 17th May 2016.

Ziflaj, A. (2014). *Native vs Hybrid App Development.* Available: https://www.sitepoint.com/native-vs-hybrid-app-development/. Last accessed 16th May 2016.

Apache. (2016). *Mobile apps with HTML, CSS & JS.* Available: http://cordova.apache.org/. Last accessed 17th May 2016.

Adobe. (2016). *Create beautiful vector art..* Available: http://www.adobe.com/uk/products/illustrator.html. Last accessed 17th May 2016.

Adobe. (2016). *Take your creativity to new places..* Available: http://www.adobe.com/uk/products/photoshop/selector.html. Last accessed 17th May 2016.

**Appendix 1 – User test**

The user test was focused around Jakob Nielsen’s 10 Usability Heuristics for User Interface Design.[[18]](#footnote-18) The 10 principles stated by Jakob Nielsen are important aspects that should be considered and optimised for every interaction design. The principles are:

* Visibility of system status.
* Match between system and the real world.
* User control and freedom.
* Consistency and standards.
* Error prevention.
* Recognition rather than recall.
* Flexibility and efficiency of use.
* Aesthetic and minimalist design.
* Help users recognise, diagnose and recover from errors.
* Help and documentation.

During the test the users were asked to speak their mind to help understand their thought process whilst navigating through the application. After the test each user was asked to give feedback about the application, both positive and negative. From the feedback gained by the users I was able to compile some information to help improve the usability of application, whilst always referring to Jakob Nielsen’s 10 principles.

**Minimum viable product**

When user testing a product it is a good idea to test it at an early stage to ensure the time taken to develop certain features does not go to waste, due to a large percentage of users disliking it. A minimum viable product is a product that is shown to user at an early stage with enough features to gather validated learning about the product. Eric Ries coined it in his book ‘The Lean Startup’. [[19]](#footnote-19) Although my application was at a later stage in the process than a standard ‘minimum viable product’, It was still unfinished and minimal compared to the finished product.

4 users of mixed ages, gender and technical knowledge undertook the test. Each participant was asked to view the application on an android and Apple device. They were asked to speak their mind during the analysis and give positive and negative feedback after the analysis was complete.

**Test administrator tools**

|  |  |
| --- | --- |
| **Hardware** | |
| **Computer** | MacBook Pro (13-inch, Early 2011) |
| **Software** | |
| **Operating system** | Mac OS X El Capitan 10.11.3 (15D21) |
| **Internet connection** | British telecommunication plc |
| **Face recorder** | Photo Booth Version 4.0.2 (288.5) |

**Participants computing environment**

|  |  |
| --- | --- |
| **Hardware** | |
| **Mobile device 1** | Samsung Ace 4 (SM-G357fZ) |
| **Mobile device 2** | Apple iPhone 5 (MD298B/A) |
| **Software** | |
| **Device 1 operating system** | Android OS v4.42 (KitKat) |
| **Device 2 operating system** | iOS 9.3.1 |
| **Internet connection** | British telecommunication plc |

**Results**

The results gained from the user test were beneficial in helping me uncover inconsistencies and errors within the application. Participants noticed valuable issues that needed to be implemented into the application to improve the accessibility and usability of the product.

**Daniela Campitelli**

****

**Current mobile device:** Apple iPhone 5

**Negative feedback**

* When selecting a playlist the user tried clicking the icon to navigate to the page. The title was the only link at that point.
* Wanted there to be a back button on each playlist to take the user back to the emotion select screen, instead of using the navigation.

**Positive feedback**

* Thought the concept was unique and interesting.
* Memorable logo, application name and branding.
* User-friendly design.
* Enjoyed the minimalistic style.

**Dominic Rathbone**

****

**Current mobile device:** Apple iPhone 5

**Negative feedback**

* Tried selecting the emotion icon to navigate to the playlist and mentioned the text needs to be more clear in saying what to click.
* Did not like the colour of the lines specifically against the homepage background.
* Thought the gif image on the homepage didn’t use enough frames and looked slow.

**Positive feedback**

* Liked the logo.
* Found the overall colour scheme pleasant and clean.
* Found the breadcrumb style animation (to signify how far through the application the user was) at the bottom of each emotion page unique and user friendly.
* Overall found the pages user friendly and easy to recognise what page you are on.

**Jemma Ktorou**

****

**Current mobile device:** Samsung Galaxy S4

**Negative feedback**

* Found the text for the emotion ‘Joy’ harder to read than the rest.
* Disliked the swipe gif animation on the homepage. Thought it could be subtler.

**Positive feedback**

* Appropriate images throughout.
* Consistent theme and branding.
* Good user interface to help the user navigate throughout.
* Sophisticated and clean homepage.

**Tom Thorley**



**Current mobile device:** Microsoft Lumia 950

**Negative feedback**

* Disliked the gif image on the home screen.
* Was unsure where to click to navigate to the playlist page.

**Positive feedback**

* Loved the logo.
* Thought the overall application was user friendly and fun.
* Thought it was a unique concept.

**Conclusion**

In result the information gained from the user test was informative and helped improve my application’s consistency, usability, accessibility and overall success. By correcting the errors discovered and implementing feedback it helped ensure that my application was at a high standard and ready to be packaged and launched.

**Appendix 2 – Consent form/Video release form**

I agree to participate in the test conducted by Jamie Millman.

I understand that the participation in this test is entirely voluntary and if at any point I feel uncomfortable or unhappy with the proceeding of the test, I will make it clear to the facilitator.

I agree to be both video and audio recorded during my session.

Please sign and date below to indicate you have read and understood the information on this form and that any questions you may have had has been answered to your satisfaction.

**Signature:**

**Date:**

**Appendix 3 – Demographic questionnaire**

1. Ziflaj, A. (2014). *Native vs Hybrid App Development.* Available: https://www.sitepoint.com/native-vs-hybrid-app-development/. Last accessed 16th May 2016. [↑](#footnote-ref-1)
2. Adobe. (2016). *Build amazing mobile apps powered by open web tech..*Available: http://phonegap.com/. Last accessed 17th May 2016. [↑](#footnote-ref-2)
3. Ionic. (2016). *Create incredible apps.* Available: http://ionicframework.com/. Last accessed 17th May 2016. [↑](#footnote-ref-3)
4. Facebook. (2016). *A FRAMEWORK FOR BUILDING NATIVE APPS USING REACT.* Available: https://facebook.github.io/react-native/. Last accessed 17th May 2016. [↑](#footnote-ref-4)
5. Apache. (2016). *Mobile apps with HTML, CSS & JS.* Available: http://cordova.apache.org/. Last accessed 17th May 2016. [↑](#footnote-ref-5)
6. Wodehouse, C. (2015). *7 Reasons Why Facebook’s React Native Is the Future of Hybrid App Development.* Available: https://www.upwork.com/hiring/mobile/react-native-hybrid-app-development/. Last accessed 17th May 2016. [↑](#footnote-ref-6)
7. Art Therapy. (2016). *Color Psychology: The Emotional Effects of Colors.* Available: http://www.arttherapyblog.com/online/color-psychology-psychologica-effects-of-colors/#.VvFb9hKLSRs. Last accessed 22nd March 2016. [↑](#footnote-ref-7)
8. Colour affects. (2008). *Psychological Properties Of Colours.* Available: http://www.colour-affects.co.uk/psychological-properties-of-colours. Last accessed 22nd March 2016. [↑](#footnote-ref-8)
9. Changing minds. (2008). *The Meaning of Colors.* Available: http://changingminds.org/disciplines/communication/color\_effect.htm. Last accessed 22nd March 2016. [↑](#footnote-ref-9)
10. Changing minds. (2008). *The Meaning of Colors.* Available: http://changingminds.org/disciplines/communication/color\_effect.htm. Last accessed 22nd March 2016. [↑](#footnote-ref-10)
11. Font squirrel. (2016). *Webfont generator.* Available: https://www.fontsquirrel.com/tools/webfont-generator. Last accessed 3rd April 2016. [↑](#footnote-ref-11)
12. Adobe. (2016). *Create beautiful vector art..* Available: http://www.adobe.com/uk/products/illustrator.html. Last accessed 17th May 2016. [↑](#footnote-ref-12)
13. Adobe. (2016). *Take your creativity to new places..* Available: http://www.adobe.com/uk/products/photoshop/selector.html. Last accessed 17th May 2016. [↑](#footnote-ref-13)
14. Adobe PhoneGap. (2016). *Debugging.* Available: http://docs.phonegap.com/references/developer-app/debugging/. Last accessed 24th Feb 2016. [↑](#footnote-ref-14)
15. Dorman, S. (2008). *What is “Defensive Programming”?.* Available: https://scottdorman.github.io/2008/07/04/what-is-ldquodefensive-programmingrdquo/. Last accessed 13th May 2016. [↑](#footnote-ref-15)
16. Ben, C. (2016). *imgcache.js.* Available: https://github.com/chrisben/imgcache.js/. Last accessed 14th May 2016. [↑](#footnote-ref-16)
17. Fisher, N. (2016). *Soundcloud custom player.* Available: https://github.com/soundcloud/soundcloud-custom-player. Last accessed 16th May 2016. [↑](#footnote-ref-17)
18. Nielsen, J. (1995). *10 Usability Heuristics for User Interface Design.*Available: https://www.nngroup.com/articles/ten-usability-heuristics/. Last accessed 17th May 2016. [↑](#footnote-ref-18)
19. Eric Ries (2011). *The Lean Startup*. USA: Crown Business (USA). p1-336. [↑](#footnote-ref-19)