

National College of Ireland  
BSc in Computing  
2015/2016

Corrina Wynne  
x12342071  
x12342071@student.ncirl.ie

Roadie

Technical Report



---

## Table of Contents

Executive Summary .....	3
1. Introduction .....	4
1.1 Background .....	4
1.2 Aims .....	4
1.3 Technologies .....	5
2. System .....	6
2.1 Requirements .....	6
2.2 Design and Architecture .....	18
2.2.1 Hardware Architecture .....	19
2.2.2 Software Architecture .....	19
2.2.3 Security Architecture .....	19
2.2.4 Communication Architecture .....	19
2.2.5 Performance .....	20
2.3 Database Design .....	20
2.4 Data Conversions .....	22
2.5 Application Program Interfaces .....	23
2.6 Testing .....	23
2.7 Graphical User Interface (GUI) Layout .....	24
2.8 Customer testing .....	28
2.9 Evaluation .....	30
3. Conclusions .....	33
4. Further development or research .....	34
5. References .....	35
6. Appendix .....	37
6.1 Project Proposal - 21/09/2015 .....	37
6.2 Project Plan - 21/09/2015 .....	38
6.3 Actual Outcome of Project Plan .....	39
6.4 Glossary of Terms: .....	40
6.5 Monthly Journals .....	40

---

## **Executive Summary**

Roadie consists of a mobile website and accompanying Android application that is designed to assist musicians and entertainers in creating and managing their acts and setlists.

It allows users to create an account and login, from there they can then create, view, edit and delete setlists.

The site includes a calculator which can be used to calculate the length of their setlists by entering the length of each song.

The site also offers a live feed of news and events which are related to music and entertainment.

---

# **1. Introduction**

The objective for this project is to provide musicians and entertainers with a tool to create and customise setlists. The setlist tool is based on a mobile website, the site also includes a setlist calculator, useful links, news and information. A mobile application will allow the users to view, edit and delete their setlists while on the move. For example, before, or even during a performance.

## ***1.1 Background***

The inspiration for this project came from discussions with friends, including touring and session musicians, who had mentioned that such tools would be of great use to them.

Several, if not most musicians, now incorporate mobile technology into their planning and performances. Sound desks can be controlled by tablets and technical issues can be troubleshooted quickly and without issue. The idea for the mobile application came about after seeing many performers using phone holders in their mic stands to view lyrics, sheet music, tabs, etc. As opposed to the user having to look up the above during a performance, they could instead programme in the lyrics or tabs before the gig, by looking at their setlist and adding the information appropriate to each individual performance.

## ***1.2 Aims***

The aims of the project are to generally create an application and site that have a wide audience with an easy to use, simple interface and responsive design.

Some aims include:

- To create an app and site which allows users to register, login, create, view and edit their setlists, calculate their setlists, as well as stay up to date on music news and events.
- To have a simple, responsive design for the website and application.

- 
- To have a secure database connection for both the site and mobile application which interacts with web services and pages.

### ***1.3 Technologies***

The mobile website has been created using HTML, CSS, JavaScript and PHP in a text editor (mostly Notepad++). The site has also been made using Bootstrap, which results in a responsive layout and caters especially to mobile sites, ensuring a sleek and easy user experience. The application was made using Android Studio and written in Java.

The database was created and integrated/connected to the site and application using PHP MyAdmin and MySQL.

A web service/API is in place in order for the mobile application to send and receive data to and from the database using JSON. A JSON parser is then implemented in the application code in order to read and display the data.

---

## **2. System**

### ***2.1 Requirements***

This section details and lists the requirements involved in the project.

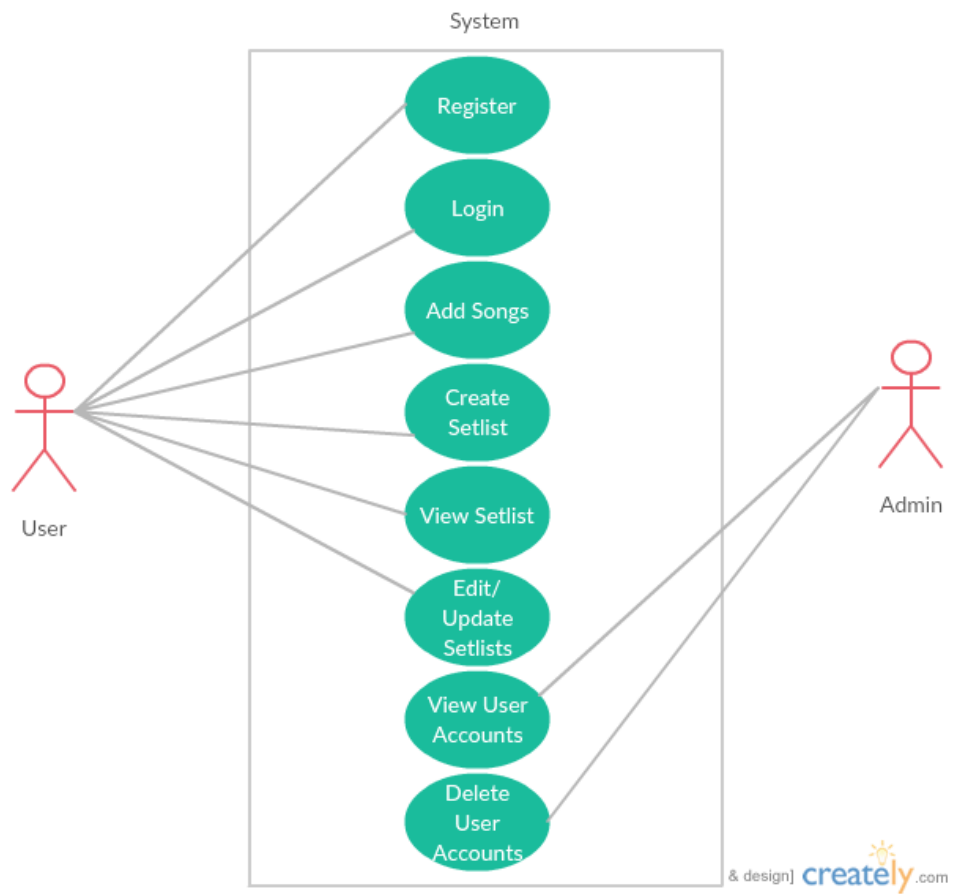
#### **Functional Requirements**

This section lists the functional requirements. The following section will describe each requirement in full detail.

1. The system shall allow the user to create and register an account.
2. The system shall allow the user to log in to the system using the aforementioned account.
3. The system will allow the user to add songs to song database.
4. The system will allow the user to create and save several setlists using the songs mentioned above.
5. The system will allow the user to edit, customise and delete their setlists.
6. The system will allow the user the view their saved setlists.

---

## High Level Use Case Diagram:



---

## **2.1.1 Requirement 1: User Registration**

### **Description & Priority**

This requirement is necessary to allow the user to register an account in the system. It is a high priority requirement as without it the user would not be able to access the other functions of the system.

This function is only available on the website and not the application.

### **Use Case Description:**

#### **Scope**

The scope of this use case is to allow the user to register themselves an account in the system which will be associated with other data and information the user inputs/saves at later dates.

#### **Description**

This use case describes the process of registering an account. It includes the process of the user opening the registration page, inputting their details (email, username and password) and pressing “register”, which will then use a PHP POST method to send the data to the server, saving their details in the database which will be checked when the user chooses to login at a later date.

#### **Flow Description:**

#### **Precondition**

The user has an internet connection and the website open on the registration page.  
The user must not have an account already logged in.

#### **Activation**

This use case starts when a user navigates to the registration page with the intent to create a new user account.



---

## **Main flow**

1. The system identifies that a user is not already logged in.
2. The user navigates to the registration page.
3. The user enters their details and clicks the “register” button.
4. The system send the user information to be stored in a MySQL database.
5. The user is automatically registered and logged in, then redirected to the home page.

## **Alternate flow**

A1: The user must re-enter details Steps 1 & 2 from above remain valid.

1. The user enters their details and clicks the “register” button.
2. The user is prompted to enter different details as the ones originally entered have been taken.
3. The user may now register using their new details. 4. The use case continues at position 4 of the main flow.

## **Exceptional flow**

E1: The user cancels registration

1. The user does not click the “register” button and leaves the page.
2. The user's account will not be created and stored in the database.

## **Post condition**

On successful registration the user will be automatically logged in and redirected to the home page.

---

## **2.1.2 Requirement 2: Logging In**

### **Description & Priority**

This requirement is high priority as it allows the user to log in and access their account, which will then allow them to access the other functions of the site.

This function is available on both the website and the application.

### **Use Case Description:**

#### **Scope**

The scope of this use case is to allow the user to log in to the system and create and view their setlists.

#### **Description**

This use case describes the process by which the user will login to the system. This is achieved on both the site and in the app (facilitated by using HTTP POST & GET methods and JSON) by checking the entered details against those saved in the database. If the details are correct, the user will be logged in.

#### **Flow Description:**

#### **Precondition**

The user has an internet connection and has the login page open on the site, or has the application open on their mobile device.

#### **Activation**

This use case starts when a user navigates to login page with the intention of logging in to the system

#### **Main flow**

1. The system identifies that a user is not already logged in.
2. The user enters their details that they have previously registered.
3. The system verifies that the details entered match the details saved for that user.

- 
4. The user is logged in and redirected to the home page.

**Alternate flow**

A1: The user must re-enter their details

This flow occurs in the event that the user has entered their details incorrectly on their first attempt.

1. The user will be prompted that their details are incorrect and to re-enter their correct details.
2. If the details are entered correctly the use case will continue from step 3 of the main flow.

**Exceptional flow**

E1: The user cancels logging in

1. The user navigates away from the log in page or closes the application.
2. In this case the user will not be logged in to the system and will have to re-open the login page to log in.

**Post condition**

The user is redirected to the home page after log in.

---

### **2.1.3 Requirement 3: Adding & Saving Songs**

#### **Description & Priority**

This requirement is necessary to allow the user to add and save songs, which are then used to create setlists. It is a high priority requirement as it is one of the main functions/selling points of the site.

#### **Use Case Description:**

##### **Scope**

The scope of this use case is to allow the user to add songs and save them to their account for making setlists.

##### **Description**

This use case describes the process of adding and saving songs. There are fields in which a user must enter song names and then click the “save” button in order to save the songs to the MySQL database (achieved using PHP POST methods).

##### **Flow Description:**

##### **Precondition**

The user has an internet connection, is logged in and has the website open on the “Add Songs” page.

##### **Activation**

This use case starts when a user navigates to the “Add Songs” page.

##### **Main flow**

1. The system identifies that the user is logged in.
2. The user enters their chosen songs in the fields provided.
3. The user clicks the “save” button.
4. The data is sent to and saved in the database, providing that not all the fields are null, using PHP.

---

### **Exceptional flow**

E1: The user does not enter any songs.

1. The user does enter values in the fields provided.
2. If all fields are null then the songs will not be saved to the database so as to avoid redundancy.

### **Post condition**

On successful saving of the songs, the user will be redirected to the “Create Setlist” page where they can create a setlists using their newly saved songs.

## **2.1.4 Requirement 4: Create and Save Setlist**

### **Description & Priority**

This requirement is necessary to allow the user to create and save setlists. It is a high priority requirement as it is one of the main functions/selling points of the site.

This function is available on both the website and on the application.

### **Use Case Description:**

#### **Scope**

The scope of this use case is to allow the user to create a setlist and save it to their account for retrieval at a later date.

#### **Description**

This use case describes the process of creating and saving a setlist. There are fields in which a user must enter some setlist information, as well as several dropdown lists which will display the songs they have already saved to the database. The user will then click the “save” button in order to save the setlist to the MySQL database (achieved using PHP POST methods).

---

**Flow Description:****Precondition**

The user has an internet connection, is logged in and has the website open on the “Create Setlist” page, or if using the application – is logged in and selects the “Add New Setlist” option.

**Activation**

This use case starts when a user navigates to the “Create Setlist” page, or selects the “Add New Setlist” option on the application with the intent to create a new setlist.

**Main flow**

1. The system identifies that the user is logged in.
2. The user enters their chosen songs in the fields provided.
3. The user clicks the “save” button.
4. The data is sent to and saved in the database, providing that not all the fields are null, using PHP.
5. The user is redirected to the “View Setlists” page.

**Exceptional flow**

E1: The user does not enter any songs.

1. The user does enter values in the fields provided.
2. If all fields are null then the setlist will not be saved to the database so as to avoid redundancy.

**Post condition**

On successful saving of the setlist, the user will be redirected to the “View Setlists” page where they can view the setlists they have created and saved previously.

---

## **2.1.5 Requirement 5: Editing or Deleting Setlist**

### **Description & Priority**

This requirement is necessary to allow the user to edit or delete their saved setlists. This is a high priority requirement as without it the database would soon become cluttered as well the users account and interface.

This function is available on both the website and on the mobile application.

### **Use Case Description:**

#### **Scope**

The scope of this use case is to allow the user to edit and delete their previously created setlists.

#### **Description**

This use case describes the process of editing or deleting the setlists that the user has created and saved previously. This is achieved by using PHP sessions and HTTP methods. The PHP methods communicate with the server and database to update, return and display the changed setlists to the user, or to delete a setlist from the database.

#### **Flow Description:**

#### **Precondition**

The user has an internet connection, is logged in, has previously created and saved a setlist and has the "View Setlists" page open on the website or the application.

#### **Activation**

This use case starts when a user presses the edit or delete option button.

---

**Main flow**

1. The system identifies that a user is logged in.
2. The system displays the setlists to the user.
3. The user selects a specific setlist to edit or delete.
4. The user makes their changes and saves the altered setlist, or deletes it.

**Exceptional flow**

E1: The user navigates away from the page.

1. The user leaves the page or does not open it.
2. The saved setlist(s) will not be edited or deleted.

**Post condition**

The user may view their newly updated setlist(s) and then navigate away from the page or logout.

**2.1.6 Requirement 6: Viewing Setlist****Description & Priority**

This requirement is necessary to allow the user to view their saved setlists. This is a high priority requirement as without it the process of creating a setlist is redundant.

This function is available on both the website and on the mobile application.

**Use Case Description:****Scope**

The scope of this use case is to allow the user to view their previously created setlists.



---

## **Description**

This use case describes the process of viewing the setlists that the user has created and saved previously. This is achieved by using PHP sessions and GET methods. The PHP methods communicate with the server and database to return and display the relevant setlists to the user

## **Flow Description:**

### **Precondition**

The user has an internet connection, is logged in, has previously created and saved a setlist and has the “View Setlists” page open or selects the “View Setlists” option in the application.

### **Activation**

This use case starts when a user navigates to the “View Setlists” page after logging in and/or saving a setlist, or selects the “View Setlists” option in the application.

### **Main flow**

1. The system identifies that a user is logged in.
2. The “View Setlists” page will automatically display the setlist(s) relevant to the logged in user.

### **Exceptional flow**

E1: The user navigates away from the “View Setlists” page.

1. The user leaves the “View Setlists” page or does not open it.
2. The saved setlist(s) will not be displayed.

### **Post condition**

The user may view their saved setlist(s) and then navigate away from the page or logout.

---

## 2.2 Design and Architecture

The website has been created using HTML, CSS, JavaScript & PHP. The PHP files & portions are used to communicate with the MySQL database.

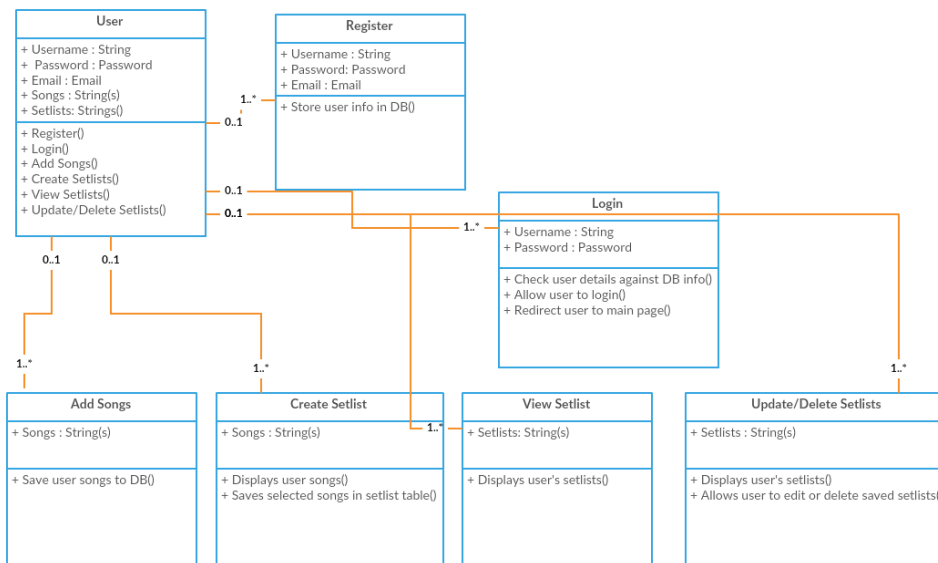
The MySQL database contains three tables – users, songs and setlists. PHP is used to get and send the inputted data to the tables and also to return and display the data to the user.

The songs and setlists tables are connected to the users table using foreign keys (userID) to ensure that the data stored is relative to user which is currently logged in.

The mobile application has been created for Android using Java. It will allow the user to login and view, create, edit and delete their setlists. Any setlists created on the site will also be displayed in this application.

In order for it to communicate with the database, a web service/API has been created and is hosted on the server alongside it. The web service uses PHP, HTTP methods and JSON to send and receive the data to and from the application, which implements a JSON parser in order to interpret the data being received.

### Class Diagram:



---

### **2.2.1 Hardware Architecture**

The site, web service, & database are hosted on a web host/server. The mobile application has been designed & created to be run on an Android device, with Android hardware in mind during development.

The website, being a responsive site, should run on all operating systems and internet enabled devices.

### **2.2.2 Software Architecture**

The scripting language used for the project was PHP, which was used for database interactions and also to create the web service which communicates with the mobile application.

As mentioned above, the mobile application has been created to run on Android OS and was developed to be compatible with Android software.

### **2.2.3 Security Architecture**

Code is included in the PHP files in order to prevent common SQL injections.

### **2.2.4 Communication Architecture**

Communication between the website and the database is achieved using PHP files which are stored on the same host as both the website files and the database files.

Communication between the mobile application and the database is achieved using a web service/API which is stored on the same host as the database, as above.

The web service has been created using PHP & JSON. It takes the inputted data from the application and sends a response using JSON, then on the application side, a JSON parser is used in order to read the response and display it to the user.

The mobile application has internet & network permissions implemented in order for it to communicate effectively with the database.

---

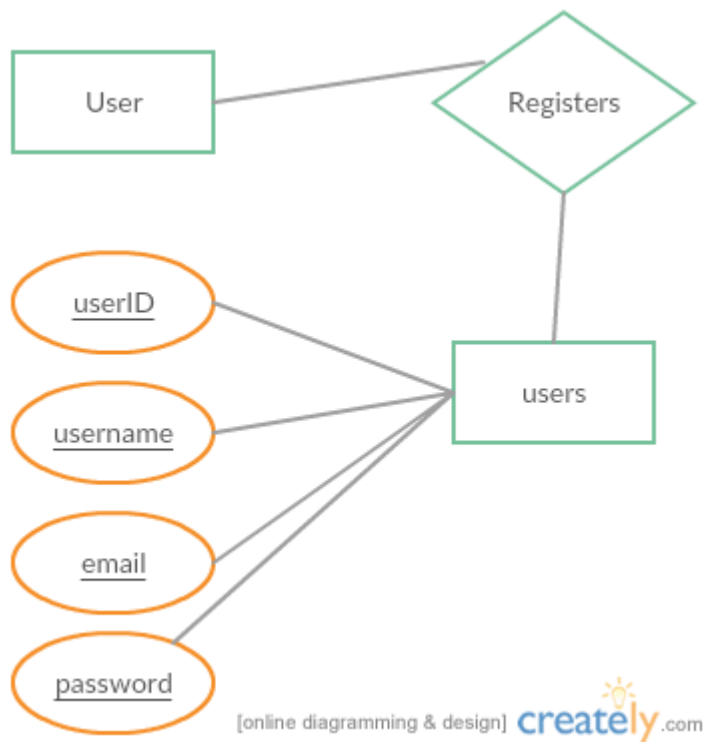
## 2.2.5 Performance

Both the website and the application meet the industry standard for performance & response times. Inputting and retrieving data from the database is done quickly and there is no long wait for the user – thus they won't lose interest or attention when using the website or application.

These quick response times and timely performance contribute to an overall pleasant user experience.

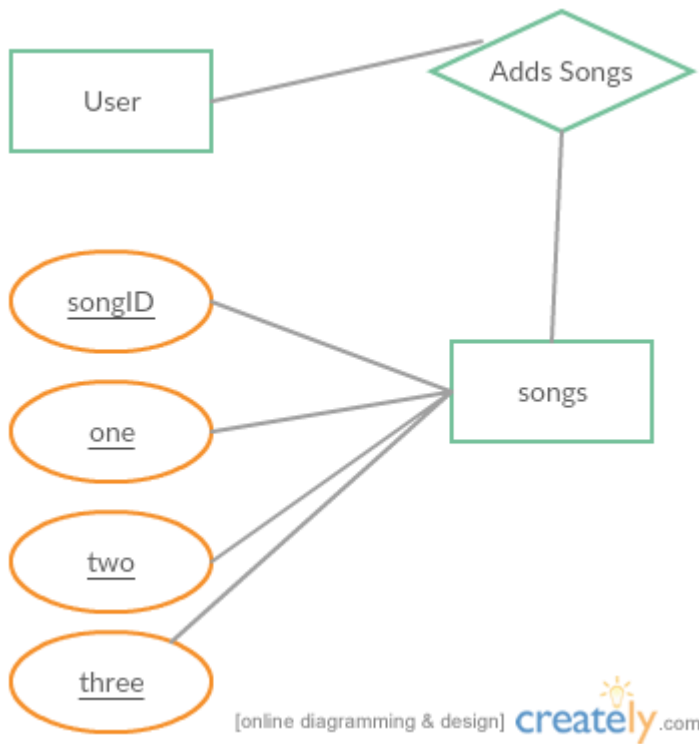
## 2.3 Database Design

ERD for registering a user:



---

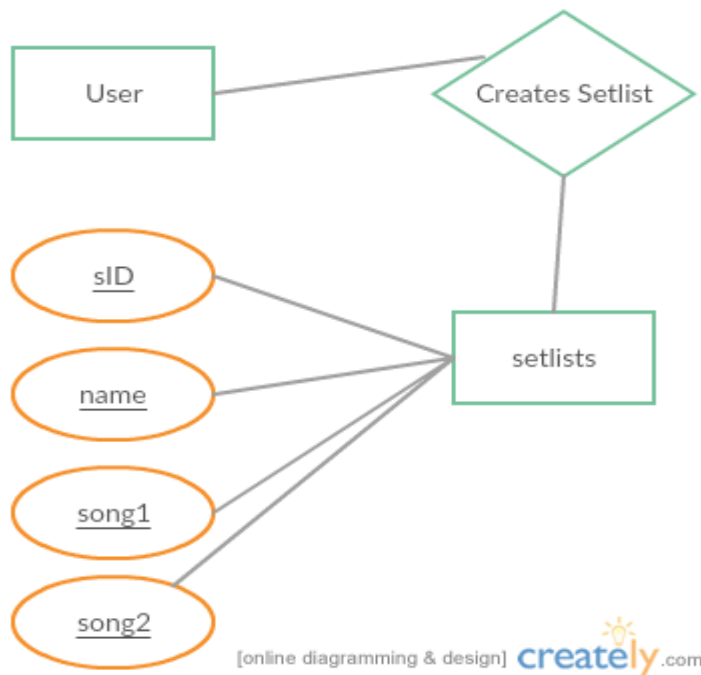
**ERD for adding songs:**



*Note: there are several more song rows in the database – not displayed for simplicity.*

---

## ERD for creating a setlist:



*Note: there are several more song rows in the database – not displayed for simplicity.*

## 2.4 Data Conversions

Any data or information the user enters into the website or application is read in the PHP code via POST.

It is then entered into the database and a JSON response is sent to the user confirming a successful entry or update.

The web service/API created to interact with the mobile application also reads the inputted data using POST and sends a JSON response to the user.

However, there is a JSON parser included in the application which reads the data received from the database/PHP files and displays them to the user in line with the applications design and general view.

---

## **2.5 Application Program Interfaces**

A Web API / Web Service has been created in order for the mobile application to communicate successfully with the server & database.

It has been created using PHP and uses HTTP methods (POST, GET, etc.) to read inputted data and then insert it into the relevant table in the database.

After the data has been inputted/updated successfully, a JSON response is echoed to the user to inform them of the status of their action.

JSON responses are also in place in case of error or failure when inserting or updating the data.

JSON messages sent to the mobile application are read using a JSON parser which has been included in the application.

This parser reads the received JSON responses from the server and displays them to the user through the application – in line with the applications layout and design.

For example, the JSON parser will read the data received and then display it in a listview – which has been chosen as the design method to display the data to the user.

## **2.6 Testing**

To ensure a clean build and high functioning system, testing (including both black box and white box testing) has been done at every stage of development.

Incremental integration testing has featured highly in this project, which consists of testing each section/module of the system to find any issues in each individual section.

After the incremental testing was complete, a whole system test was conducted, which involved using the website as an end user would, trying every area/button/form available on the site to ensure all are working correctly.

There have also been tests to ensure that the data is written to and read from the database correctly, these have been conducted by repeatedly adding/removing

---

data and later retrieving it.

The mobile application has been tested in a similar way, ensuring that the data is written/read correctly each time, that it responds in a timely manner and that it doesn't crash.

## ***2.7 Graphical User Interface (GUI) Layout***

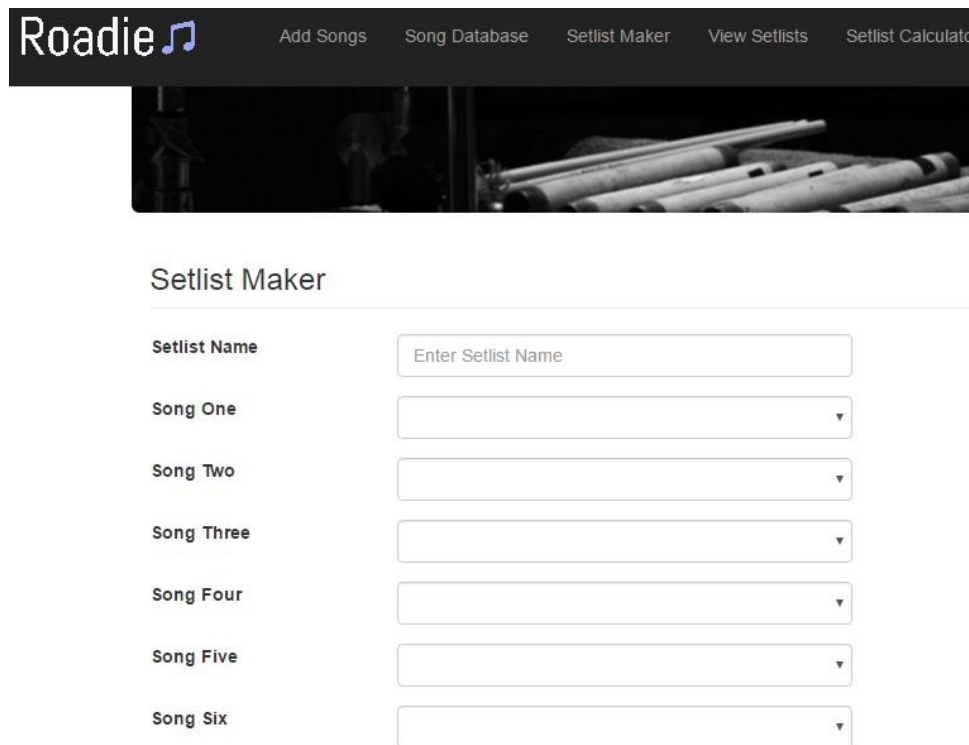
The user interface on the website is based on a responsive design which was created using HTML, CSS, JavaScript and bootstrap.

The responsive nature of the website means that it also works well on mobile devices and there are no limitations when viewing the site on mobile devices.

The design is modern, clean and easy to use - with clear headings and sections to the site.

### **Screenshots:**

Setlist Maker Page:



The screenshot shows the 'Setlist Maker' page of the Roadie website. The header features the 'Roadie' logo with a musical note icon and navigation links for 'Add Songs', 'Song Database', 'Setlist Maker', 'View Setlists', and 'Setlist Calculator'. Below the header is a banner image of drumsticks. The main content area is titled 'Setlist Maker' and contains a form with the following fields:

Setlist Name	<input type="text" value="Enter Setlist Name"/>
Song One	<input type="text"/>
Song Two	<input type="text"/>
Song Three	<input type="text"/>
Song Four	<input type="text"/>
Song Five	<input type="text"/>
Song Six	<input type="text"/>



## View Setlists Page:

Setlist ID	Setlist Name	Song One	Song Two	Song Three	Song Four	Song Five	Song Six	Song Seven	Song Eight	Song Nine	Song Ten	Song Eleven	Song Twelve	Song Thirteen	Song Fourteen	Song Fifteen	Edit	Delete
1	test	test	test2														Edit	Delete
2	newtest	test3	test4														Edit	Delete

*Note: The Song Database page has a similar appearance to the above.*

## Edit Setlist Page:

**Edit Setlist**

**Name**

**Song One**

**Song Two**

**Song Three**

**Song Four**

**Song Five**

**Song Six**

**Song Seven**

*Note: The Edit Songs page has a similar appearance to the above.*

---

## Setlist Calculator Page:

### Calculate Setlist

No.	Song Name	Song Length (Mins)
1	<input type="text"/>	<input type="text"/>
2	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>
6	<input type="text"/>	<input type="text"/>
7	<input type="text"/>	<input type="text"/>
8	<input type="text"/>	<input type="text"/>
9	<input type="text"/>	<input type="text"/>
10	<input type="text"/>	<input type="text"/>
11	<input type="text"/>	<input type="text"/>

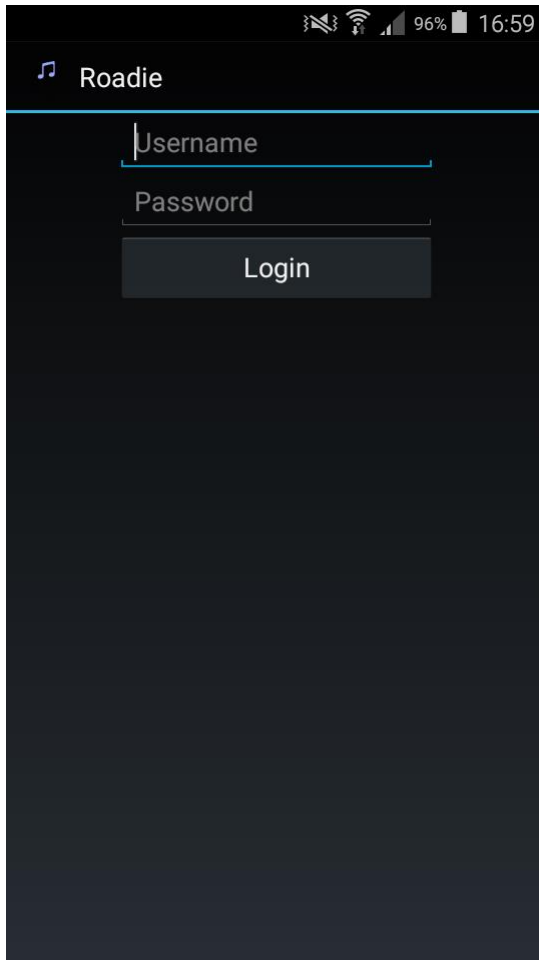
*Note: There is a row at the bottom of the calculator which shows the total length.*

---

The design of the mobile application is also clean and minimal, with a clear set of buttons and headings for each function – which also supports ease of use.

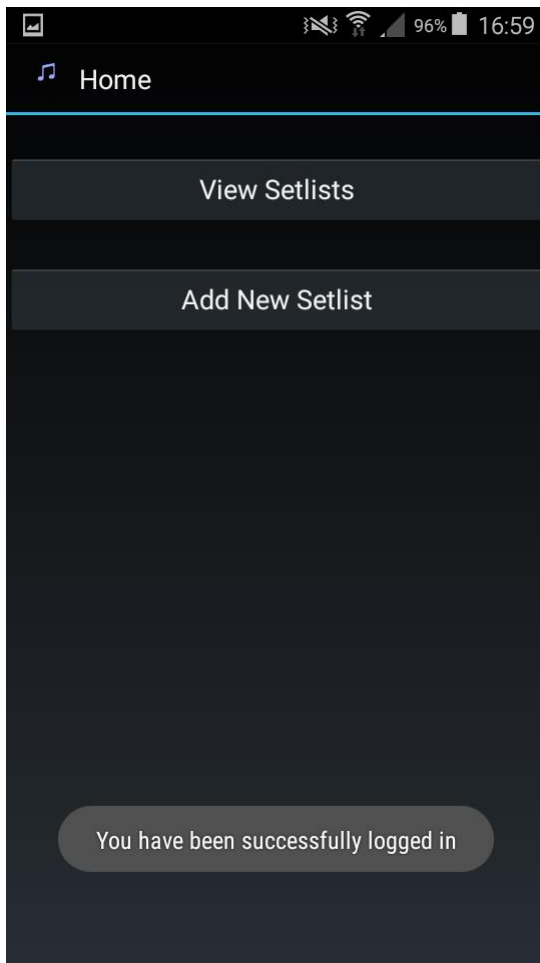
**Screenshots:**

Landing Screen:



---

Home Screen:



*Note: Using the above buttons the user can choose to create a new setlist, or view existing setlists where they will also have the option to edit or delete a setlist.*

## **2.8 Customer testing**

As well as personal testing, throughout development I have approached several musicians and entertainers to review and test the project for me.

They have given feedback and suggestions throughout which I have taken on board, such as:

- 
- Adding a song database page to add and save songs which they frequently play.
  - Adding dropdown menus to the create setlists page which display the songs they have previously saved.
  - Adding the setlist calculator page and improving the layout of it.
  - Adding more suitable/most frequently used twitter and news feeds to the news and events page.

After adding these improvements the customers have since been testing the final product, they found it to be an extremely helpful organisational tool which has improved their planning and organisation for gigs and events.

Most customers I have approached for testing have been using the site to create setlists and then sharing their login with other band members who can then review the setlist and make suggestions. They noted that this was an improvement from their previous system which would usually just be one person making a setlist and then writing it down on a piece of paper which would frequently get lost or eventually become quite messy with corrections or changes being made.

A survey was conducted after customer testing was complete, the results may be viewed in section 2.9 below (Evaluation).

Survey participants included:

Daniel M. – Full time musician and sound engineer

Eoghan D. – Session musician

Liam D. – Session musician

Kris W. – Rapper and MC

Louis F. – Band manager and event coordinator

Johnathan F. – Drummer in the music industry for 20+ years

Philip M. – DJ and entertainer

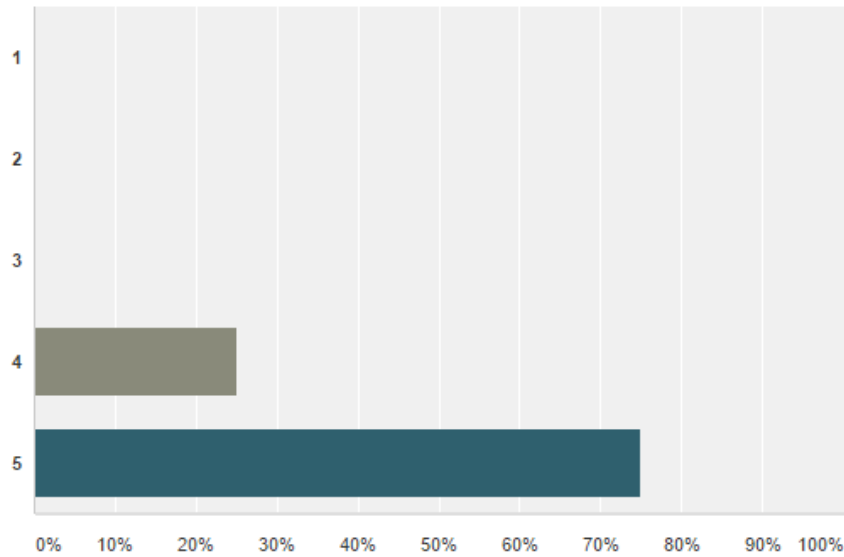
---

## 2.9 Evaluation

As mentioned above, a survey was conducted among customers who tested the project, the results are as follows:

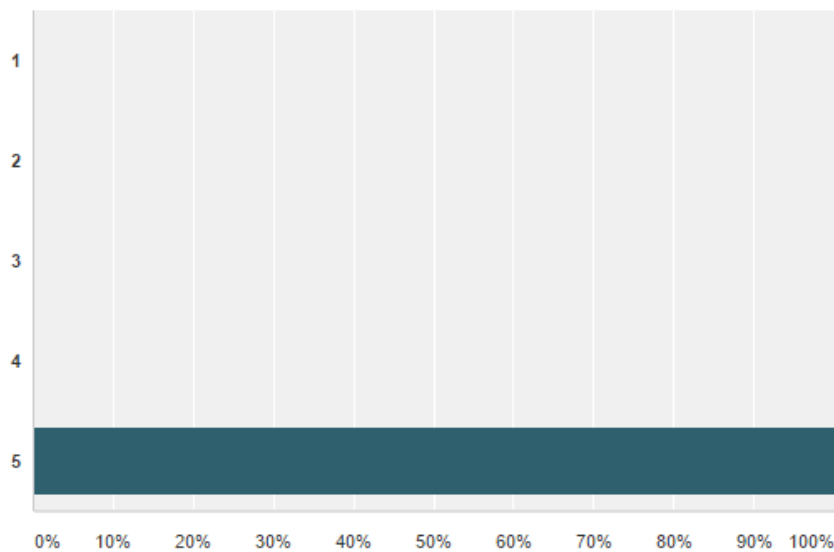
### On a scale of 1-5 how would you rate the UI of the website?

Answered: 8 Skipped: 0



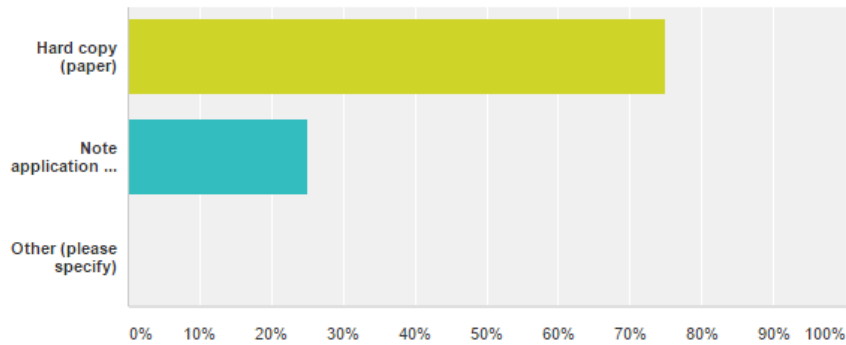
### On a scale of 1-5 how would rate the UI of the mobile application?

Answered: 8 Skipped: 0



## How do you currently organise your setlists and events?

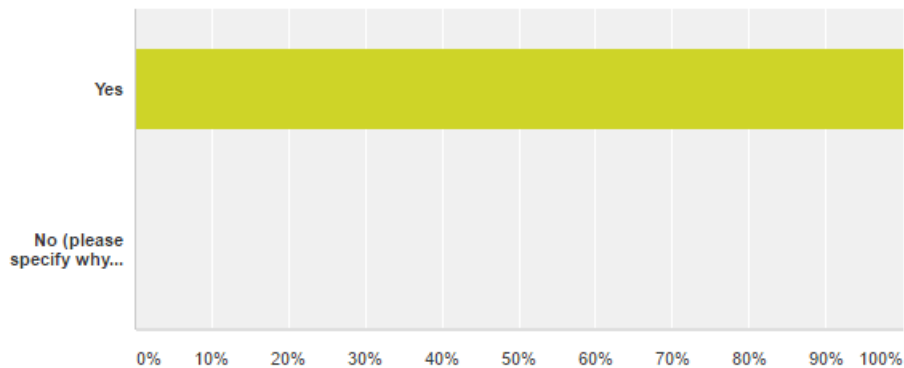
Answered: 8 Skipped: 0



Answer Choices	Responses	
▼ Hard copy (paper)	75.00%	6
▼ Note application on phone	25.00%	2
▼ Other (please specify)	0.00%	0
Total		8

## After using the site and application would you consider using it as your primary way of organising setlists/events?

Answered: 8 Skipped: 0



## Please mention bugs or errors you encountered

Answered: 3 Skipped: 5

● Responses (3)   Text Analysis   My Categories

Categorize as...   Filter by Category   Search responses

Showing 3 responses

---

No bugs were encountered during my use of this app.  
4/16/2016 2:46 PM   [View respondent's answers](#)

---

None  
4/16/2016 2:42 PM   [View respondent's answers](#)

---

None  
4/16/2016 2:42 PM   [View respondent's answers](#)

## General/Extra comments

Answered: 4 Skipped: 4

● Responses (4)   Text Analysis   My Categories

Categorize as...   Filter by Category   Search responses

Showing 4 responses

---

Very cool man  
4/21/2016 8:02 PM   [View respondent's answers](#)

---

The news and events page is a nice touch.  
4/16/2016 3:01 PM   [View respondent's answers](#)

---

Great app.. Very excited to use it live :)  
4/16/2016 2:42 PM   [View respondent's answers](#)

---

None  
4/16/2016 2:42 PM   [View respondent's answers](#)



---

### **3. Conclusions**

Achievements:

- The project has achieved most of the aims I had set out for it.
- Customers have responded well to it and find it to be a very helpful asset.
- The site and application have a good design and are easy to use and navigate.
- The project has been built with customer feedback and recommendations in mind, resulting in a site and application which appeals more to the target audience.

---

## 4. Further development or research

In future it could be possible to extend the website and application in several ways.

Examples include:

- Adding a forum or discussion area to the site.
- Adding an NFC function to the application, which I had originally set out to do but was unfortunately restricted by time.
- Making all setlists available (if the user chooses to make theirs available to the public) for others to view for inspiration, or to comment on.
- Adding location detection to the mobile application which may alert users of gigs/events in their area.

---

## 5. References

Mark Otto, and Bootstrap contributors. Components · Bootstrap [Online]  
Available at *Getbootstrap.com*  
[Accessed Sept. 2015].

Mysql :: Mysql 5.6 Reference Manual :: 13.1.17.3 Using FOREIGN KEY Constraints [Online]  
Available at *Dev.mysql.com*  
[Accessed Oct. 2015].

PHP Connect To Mysql [Online]  
Available at *W3schools.com*  
[Accessed Oct. 2015].

PHP Sessions [Online]  
Available at *tutorialspoint.com*  
[Accessed Oct. 2015].

PHP: Mysql (Original) - Manual [Online]  
Available at *Php.net*  
[Accessed Oct. 2015].

PHP: PHP Manual – Manual [Online]  
Available at *Php.net*  
[Accessed Oct. 2015].

Waterson, Kevin. "Introduction To PHP Sessions" [Online]  
Available at *Phpro.org*  
[Accessed Oct. 2015].

SQL FOREIGN KEY Constraint [Online]  
Available at *W3schools.com*  
[Accessed Oct. 2015].

Android PHP/MYSQL Tutorial [Online]  
Available at *tutorialspoint.com*  
[Accessed Nov. 2015].

Android, Introduction. Introduction To Android | Android Developers [Online]  
Available at *Developer.android.com*  
[Accessed Nov. 2015].

Tamada, Ravi. "How To Connect Android With PHP, Mysql" [Online]  
Available at *AndroidHive*  
[Accessed Dec. 2015].

---

Accessibility - W3C [Online]  
Available at *w3.org*  
[Accessed Jan. 2016].

UML - Class Diagram [Online]  
Available at *tutorialspoint.com*  
[Accessed Jan. 2016].

Javascript Basics [Online]  
Available at *Mozilla Developer Network*  
[Accessed 10 Mar. 2016].

Javascript Variables [Online]  
Available at *W3schools.com*  
[Accessed 11 Mar. 2016].

The Javascript Tutorial | Javascript Tutorial [Online]  
Available at *Javascript.info*  
[Accessed 11 Mar. 2016].

Wynne, Corrina.  
*Mid Point Presentation Report.*  
2016. Print.

Wynne, Corrina.  
*Project Analysis And Design.*  
2015. Print.

---

## 6. Appendix

### 6.1 *Project Proposal - 21/09/2015*

#### **Objectives**

The objective for this project is to provide musicians and entertainers with a tool to create and customise setlists. It will also integrate NFC technology to assist with gigs and shows.

The setlist tool will be based on a mobile website, the site will also include useful links, news, maps and information.

A mobile application will allow users to write to and read from NFC tags.

#### **Background**

The inspiration for this project came from discussions with friends and musicians, who had mentioned that such tools would be of great use to them.

Several, if not most musicians, now incorporate mobile technology into their planning and performances. Sound desks can be controlled by iPads and technical issues can be troubleshooted quickly and without issue.

The idea for the NFC application came about after seeing many performers using phone holders in their mic stands to view lyrics, sheet music, tabs, etc. As opposed to the user having to look up the above during a performance, they could instead programme in the lyrics or tabs before the gig, by looking at their setlist and adding the information appropriate to each individual performance.

#### **Technical Approach**

The mobile website will be created using HTML, CSS, JavaScript and PHP in a text editor (most likely Notepad++). The site will be made using Bootstrap, which includes a responsive layout and caters especially to mobile sites, ensuring a sleek and easy user experience.

The application will be made using eclipse and most likely also written in Java. For the 3<sup>rd</sup> year software project I created an interactive website using bootstrap and also created an NFC app, so I have background in these technologies.

This year however, I wish to expand on the work I completed last and also to complete it to a higher standard with several improvements I have learned over the summer & during my work placement.

For example, my role during my work placement was a junior DBA working with SQL,

---

MySQL, PLSQL & Access, so my database skills have improved greatly, meaning this year there should be little to issues in creating and connecting the database.

### **Special resources required**

The only special resources required for this project will be NFC tags.

## **6.2 Project Plan - 21/09/2015**

### **Technical Details**

The site will be developed using HTTP, CSS, JavaScript & PHP, while the mobile application with developed with Java.

The database will most likely be developed with SQL if possible, as it is the language I am most familiar with.

### **Evaluation**

To ensure a clean build and high functioning system, testing will be done at every stage of development. Incremental integration testing will feature highly in this project, which consists of testing each section/module of the system to find any issues in each individual section.

After the incremental testing is complete, a whole system test will be conducted, which would involve using the website as an end user would, trying every area/button/form available on the site to ensure all are working correctly.

There will also be tests to ensure that the data is written to and read from the database correctly, these will likely be conducted by repeatedly adding/removing data and later retrieving it.

The mobile application will be tested in a similar way, ensuring that the data is written/read correctly each time, that it responds in a timely manner and that it doesn't crash.

There are several musicians and entertainers whom I can approach to test the end product for me. I will request that they also test the prototype and take any feedback on board in order to improve it.

When the project is complete they will again be asked to test the system and report back with any bugs or issues that should be addressed.

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	07 Sep '15	14 Sep '15	21 Sep '15																
							S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T
1																									
2	➤	Research	16 days	Mon 07/09/1	Mon 28/09/1																				
3	➤	Planning	6 days	Mon 14/09/1	Mon 21/09/1																				
4	➤	Beginning Documentation	6 days	Mon 21/09/15	Mon 28/09/15																				
5	➤	Work on site being commenced	5 days	Mon 05/10/15	Fri 09/10/15																				
6	➤	Building main frame and layout of site	5 days	Mon 12/10/15	Fri 16/10/15																				
7	➤	Adding features to site	10 days	Mon 19/10/15	Fri 30/10/15																				
8	➤	Completing work on site	16 days	Mon 02/11/15	Mon 23/11/15																				
9	➤	Creating & connecting database	11 days	Mon 23/11/15	Mon 07/12/15																				
10	➤	Testing site functionality	6 days	Mon 07/12/15	Mon 14/12/15																				
11	➤	Beginning work on application	6 days	Mon 18/01/16	Sat 23/01/16																				
12	➤	Building main layout of application	5 days	Mon 25/01/16	Fri 29/01/16																				

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors	07 Sep '15	14 Sep '15	21 Sep '15																
							S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T
13	➤	Adding functionality	10 days	Mon 01/02/1	Fri 12/02/16																				
14	➤	Connecting to site	5 days	Mon 15/02/1	Fri 19/02/16																				
15	➤	Final testing of all systems	11 days	Mon 22/02/16	Mon 07/03/16																				
16	➤	User acceptance testing	16 days	Mon 07/03/16	Mon 28/03/16																				

### 6.3 Actual Outcome of Project Plan

Overall, the project achieved most goals set out originally, apart from incorporating NFC technology. This was due to both a lack of time and with the NFC technology not meshing well with the new idea and image I had for the mobile application. However, I did add a setlist calculator, which allows musicians to calculate the total length of their setlists.

However, the website and application were completed ahead of schedule (see above), with most of the work being finished before 2016. The time issues came with testing, fixing bugs and other tasks such as documentation and preparing for the presentations.

Fixing errors and expanding the site was an ongoing process, however I only really focused on improvements from January/February 2016. Under guidance of my supervisor, I began taking suggestions from potential customers on changes and improvements that they would like to see.

I took these all on board and implemented them to the best of my ability, including adding dropdown menus and making the calculator more user friendly.

Testing, repairing and improving continued on into mid-April, with the final product being finished around the 15<sup>th</sup>. Final customer testing occurred after, with the results being available above (2.9 - Evaluation Methods).

---

## **6.4 Glossary of Terms:**

- Roadie – Person who tours with band/musician. Manages technical aspects of show such as sound testing and assisting with setting up stage and equipment.
- Setlist – List of songs that a musician/band intends to play.
- Session musician – Musician that is not bound to any band, will play with several other bands when and if they are needed.

## **6.5 Monthly Journals**

**September 2015:**

### **My Achievements**

This month we were introduced to the project and told what was expected of us. I began researching ideas and techniques almost immediately. I feel as though this month I achieved a suitable amount as through my research I found an idea which appealed to me and I now have a solid plan in place for beginning the practical work on my project.

### **My Reflection**

Initially (and still sometimes now) the project was very daunting. There is a lot expected of these projects and they must be of a very high standard as well as being interesting/original.

However, I always find big projects intimidating until I begin working on them and checking milestones off my list, I then find that I'm completing work quite quickly and with ease.

I feel as though my project proposal was a bit rushed as there were also some other big assignments due this week, however, we were told not to worry too much about it and I think it will act as a good guideline regardless.

### **Intended Changes**

Next month I will be beginning work on the project, I have a wide variety of resources saved and have all necessary development applications already installed so that the work can begin straight away.

I expect to achieve a lot next month, at least have the basis of the site complete, without the more complex aspects, but still a functioning, clean website nonetheless.



---

**October 2015:**

**My Achievements**

This month I met with my project supervisor who I found was very supportive and has helped me prioritise tasks. We discussed working on the requirements specification and what research I should be undertaking in order to complete a well written specification. In keeping with my project plan, the website section of my project is almost complete. It is hosted online and each page is built and connected. The only work left on the site itself is working on the member area (where users store their information after creating an account) and some cosmetic work on the setlist calculator. The site has successfully been connected to a MySQL database which stores users' login information. I hope to add another database soon where users can save their setlist information and retrieve it when they log in.

**My Reflection**

I had some issues this month with the database and with using PHP, as I had used it very much previously. However, through extensive research I eventually got it working successfully.

With several other projects being undertaken at the same time I felt I pushed this project aside, however after meeting with my project supervisor I started working on it once again and the work is progressing at a pace I'm happy with.

**Intended Changes**

Next month, I will be looking into mobile application development and connecting the mobile application to the website database. I have never undertaken something like this before so extensive research and practise will be required.

My supervisor suggested I connect the application to the website rather than just having the application be an NFC application in order give the project more depth and functionality.

I realised that I need to stick the project plan and ensure I do not fall behind on my work.

**Supervisor Meetings**

Date of Meeting: 20/10/2015

Items discussed: Architecture of project, requirements specification, general project plan, desired outcomes.

Action Items: Research architecture for requirements specification and begin work on it.

---

**November 2015:****My Achievements**

This month I feel I achieved a lot. I have created a mobile application to accompany the website and it is fully functioning. I have also created a web service which allows the application to communicate with the web server & database.

The app is still in the beginning stages but at the minute it allows the user to view their saved setlists and to create, edit and delete any setlists also.

Not much work was done to the site this month as is it not really needed. The site currently allows the user to register, login and create/view setlists, as well as having a news & events section and a setlist calculator page. I organised the folders and changed the URLs & redirect links slightly just for convenience and simplicity.

**My Reflection**

When creating the app and web service there was a lot of trial and error, mostly small mistakes or errors which were overlooked – mainly due to inexperience in app creation. However, during the process I have learned a lot about android app development and feel confident going forward in completing it.

My workload has lightened this month as most of my other assignments are completed or near completion, which gave me more time to focus on the project and make some progress with it before December, when I assume I will be too focused on exam study to do much work on it.

I have also considered changing the project slightly to make it an open and collaborative website, where all users can view all setlists, I am not certain about this currently but I am thinking about it.

**Intended Changes:**

Next month I will look into adding a login/register to the mobile app. I believe another web service will need to be created for this function and I'm not certain how the development will go.

**Supervisor Meetings**

Date of Meeting: 17/11/2015

Items discussed:

Requirements specification feedback, mobile app development.

Action Items:

Start working on mobile application & web service.

---

Date of Meeting: 26/11/2015

Items discussed:

Project analysis design document, mobile application feedback.

Action Items:

Begin work on design document.

## **December 2015:**

### **My Achievements**

This month I have worked on the project much as I have mostly been studying for exams. I have, however, added a login page to the application – so a user must log in before accessing setlists.

It is currently fully functional and was also created using a web service.

I haven't done any work on the site this month but there is very little work remaining on it.

### **My Reflection**

At the moment I am feeling quite confident in the work I have achieved so far, everything is working and I am ahead of schedule.

I have also received positive feedback from my supervisor throughout the semester so I believe I am on track.

I am quite happy that creating the login feature for the mobile went so smoothly as I was initially worried that I wouldn't be able to extend the application as I had intended, considering my limited experience in android development.

### **Intended Changes**

Next month I will focus on the appearance of the mobile application mostly and perhaps finishing up the website, followed by testing.

### **Supervisor Meetings**

As I met my project supervisor at the end of November (26/11) and considering I was busy finishing up assignments for Christmas and studying, we did not have a meeting this month.

However, at our last meeting there were no issues or urgent tasks brought up, and I have kept him updated on development since.

---

## **January 2016:**

### **My Achievements**

This month I was mainly focusing on exams. I have completed most of my technical report and presentation for my mid-point presentation (9/02/16).

I think I have achieved enough at this stage for the presentation to go well and to be well received.

### **My Reflection**

This month I realised that there is not much time left to devote to working on the project – with various assessments and projects to be done for other modules.

I realised that I should continue working on the project and not leave it to the side, as if I complete it soon I will be able to focus my attention on other work that I need to do.

### **Intended Changes**

After meeting with my project supervisor, he suggested adding drop down menus to the website.

So, rather than manually inputting each song every time, a user can enter their songs and save them in a song database. The setlist maker page will then pull this data and display it for the user to choose from.

### **Supervisor Meetings**

Date of Meeting: 26/01/2016

Items discussed: Current progress, mid-point presentation

Action Items: Create dropdown menus, look into editing the calculator.

## **February 2016:**

### **My Achievements**

This month I gave my mid-point presentation, which went well and I was happy with the grade I received for it.

I was also able to create a simple but functional calculator for the site, as well as updating the "Create Setlist" page to include dropdown menus where users can select saved songs to put in their setlists. Users can also now save their songs in a "Song Database" which they can later choose from when making setlists.

### **My Reflection**

The midpoint presentation gave me a good idea of where I am in the project progress wise and also where to focus my attention for future upgrades.

---

### **Intended Changes**

The main changes I intend to do in the coming months is finishing up the site in general – tidying up, fixing any php issues, etc. As well as any changes my new project supervisor may suggest.

### **Supervisor Meetings**

As I have had a change of supervisor this month, I have not yet had a meeting.

### **March 2016:**

#### **My Achievements**

This month I completed all technical work on the project. This included finishing up the website, extending setlist options, creating a logo and finishing the Android application. I also conducted personal tests as well as customer testing.

#### **My Reflection**

When finalising the project I realised I should have paid more attention to the general layout and format of code while initially developing the site and application as I had quite a lot of tidying up and commenting to do.

I will keep this in mind for future projects.

#### **Intended Changes**

Next month I will be mostly focusing on exams, however I will also be finishing the technical report and working on my presentation and showcase preparation.

#### **Supervisor Meetings**

Date of Meeting: 03/03/2016

Items discussed: Status of project, marking scheme, next steps.

Action Items: Look at grading rubric and apply it to my project, work on technical report.

Date of Meeting: 10/03/2016

Items discussed: Status of project, bugs.

Action Items: Finalise project, fix bugs, work on report.