

National College of Ireland  
BSc in Computing  
2015/2016

David Heraty  
11401282  
[David.heraty@student.ncirl.ie](mailto:David.heraty@student.ncirl.ie)

# FestFinder



Technical Report



## Declaration Cover Sheet for Project Submission

### SECTION 1 *Student to complete*

<b>Name:</b> David Heraty
<b>Student ID:</b> x11401282
<b>Supervisor:</b> Dominic Carr

### SECTION 2 Confirmation of Authorship

*The acceptance of your work is subject to your signature on the following declaration:*

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

**Complete the sections above and attach it to the front of one of the copies of your assignment,**

**What constitutes plagiarism or cheating?**

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

**Penalties for Plagiarism**

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

# Table of Contents

1.	Executive Summary .....	1
2.	Introduction .....	2
2.1	Background.....	2
2.2	Aims .....	2
2.2.1	Purpose of the Application .....	2
2.2.2	Benefits .....	3
2.3	Technologies.....	3
2.3.1	Language.....	3
2.3.2	Development Environment .....	4
2.3.3	Backend .....	4
3.	System.....	5
3.1	Requirements .....	5
3.1.1	Functional requirements .....	5
3.1.2	Newly Identified Requirements.....	6
3.1.3	Dropped Requirements .....	6
3.1.4	Use cases .....	7
3.1.5	Data requirements.....	10
3.1.6	User requirements.....	10
3.1.7	Environmental requirements .....	10
3.1.8	Usability requirements .....	11
3.2	Design and Architecture .....	11
3.3	Implementation.....	11
3.3.1	Setup.....	11
3.3.2	Project Creation.....	12
3.3.3	Authentication.....	12
3.3.4	JSON data.....	13
3.3.5	SQLite.....	13
3.3.6	Google Maps.....	14
3.4	Graphical User Interface (GUI) .....	15
3.4.1	Authentication.....	15
3.4.2	Home .....	16

---

3.4.3	Profile Pages .....	17
3.4.4	Favourites .....	17
3.4.5	Search .....	18
3.4.6	Map search .....	19
3.4.7	Festival Info .....	20
3.4.8	Other.....	21
3.5	Testing .....	27
3.5.1	Unit Testing.....	27
3.5.2	Persona Testing .....	28
3.5.3	Stress Test.....	28
3.6	Evaluation .....	29
3.6.1	System Usability Scale .....	29
3	Future work .....	31
4	Conclusion.....	32
5	References .....	33
3.6.2	Online References .....	33
3.6.3	Tutorials and Resources .....	35
6	Appendix.....	36
7.1	Project Proposal .....	36
7.2	Online Survey.....	39
7.2.1	Introduction.....	39
7.2.2	Questions and Responses.....	39
7.3	Monthly Journals.....	49

## 1. Executive Summary

In February 2016 the number of available applications on the Google Play Store reached two million (Statista, 2016). There is such a large number of music festival applications already on the Android market such as; The Festival App, Festival Buddy and applications for individual festivals such as The Official Electric Picnic app (Play.google.com, 2016). However, FestFinder offers something different than the rest of these applications. FestFinder holds the locations of multiple music festivals in Europe. Users can search for the desired festival using two search pages. Once the festival is found the user can navigate to the information page where they can find useful material such as general information, line-ups, start and end dates, site maps, ticket information, social media links and promotional multimedia. Not only does the app offer festival information and locations, but it also provides users with a checklist, so they don't forget essential items before heading to their favourite festival. FestFinder can also be used to post and browse tickets for sale.

In order for FestFinder to be different than the other applications available on the market, it needed to have an extra pull towards the app that some of the rest don't. This is why FestFinder includes a Map and list search page, a checklist and a ticket sales page, as mentioned above, along with the current local weather, a festival guide and a twitter search so that users can be up-to-date with festival news and so they can see what festivals might be trending or popular at the moment.

## 2. Introduction

The below section includes the background of why and how I decided to develop FestFinder. It also holds the aims of my application that I set to meet. The purpose of my application will be defined. Some features of FestFinder will be outlined and the benefits this application will bring, are also outlined.

### 2.1 Background

Having talked to some of my target audience to generate some ideas, I discovered that all of them were interested in music and festivals and had been to festivals but have never downloaded a festival themed application before. I personally, have also never downloaded a festival themed application to my device. This is one reason why I wanted to develop such an application, as I have never used or seen such a system with the features that I will outline in the following sections of this document.

As a music lover and a festival goer I noticed from my research that among the 2 million apps available on the Play Store (Statista, 2016), there wasn't one that hosts multiple festival ticket sales and information in the same app. There are plenty of apps that cater for individual festivals but not one that hosts multiple festivals on it. For example, the Tomorrowland app on the Play Store (Play.google.com, 2016) has up to 500,000 installs but it only deals with one specific festival in Belgium. It is one of the most popular festivals in Europe and has been ranked in the top 50 worldwide (Time Out London, 2016), so for my target audience it definitely needs no introduction. For smaller and less popular festivals it can be hard to breakthrough and be like Tomorrowland without a lot of revenue to put into the festival to book better performances, improve sound quality, improve stage designs, advertise, promote and improve the facilities. In my opinion, Festfinder will help smaller festivals become known by providing information about them. Also, it will give them an opportunity to expand by advertising them alongside the larger more popular festivals in Europe.

So, FestFinder will provide links to the official sellers of the festival tickets which will help save people time from searching on Google and it will also save them money and give them trust in the tickets they are about to purchase. FestFinder will also contain a page where users can buy and sell amongst themselves.

### 2.2 Aims

#### 2.2.1 Purpose of the Application

The purpose of my application is to provide a user-friendly interface which could potentially provide a helpful service to the user. The application should be pleasing to the eye while serving the intended purpose - hosting a potentially very large number of festivals on the same application and providing the user important festival information along with other features to help their festival experiences be more pleasant. To do so, the application will store festival data in an online database and display this data in a list view for the user to choose a preferred festival to retrieve information about it. Along with the list of festivals the application will hold festival locations on a google map with custom FestFinder markers and the ability to click on a marker to see the title and date of the festival and also to click again to navigate to the

information page for that festival. Additionally, the application will provide links to the official sellers of tickets for the festivals.

Users can also avail of the in app buy and sell page where users can post tickets for sale and also browse for available tickets.

A festival guide page will be an additional function for the users to view. Often people forget to bring valuable items to a festival with them or setup their tents downhill where they are prone to flooding in heavy rain. So this page will provide users some useful tips to make their festival experience more pleasurable. Along with a checklist for marking off items one may forget to bring, these features are sure to make a user's festival experience more pleasant.

Finally, a twitter feed will be an additional function for the user to check for festival updates like change of venue, date and line-up or to see which festival other people are going to at the moment.

### 2.2.2 Benefits

The application is not only benefitting the users with the above functionality but also the smaller less known festivals too. Providing festival information such as lineup, date, location, ticket sales, multimedia, social media links and a festival map will give them a chance to be among the best in Europe by advertising their information and giving users official links to tickets and presenting them with opportunities to expand.

## 2.3 Technologies

### 2.3.1 Language

Java programming will be used to add functionality to my pages which is the primary role of any application. I chose Java as it is the Android programming language. The reason I chose Android is because I was already familiar with Java and enjoyed writing it and solving problems using this language. There is also a huge amount of documentation and help online and it also has excellent performance (Wikipedia, 2016).

I am using Android XML in Android Studio for defining the layout of all items used in my app. I am using MySQL to create and maintain my database. I decided to use MySQL because of its high availability and performance and because of its scalability and flexibility (Mysql.com, 2016).

I chose PHP to connect to the database. I was not very familiar with PHP but had researched it and found many tutorials on how to use it and found it was very learnable (W3schools.com 2016). JSON (JavaScript Object Notation) data is parsed to Java in order to pull information from my database to the application. JSON is an open-standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs. (Wikipedia, 2016)



### 2.3.2 Development Environment

The app is being developed in Android Studio as the development environment. I chose Android studio, as outline above, because of my familiarity with Java. I also chose it because I was able to find lots of online resources and tutorials on Android Studio.

I am using the Android SDK for simple drag and drop of buttons, widgets and other features. The SDK is very helpful for creating app pages with its easy to use drag and drop functionality of buttons, widgets and views.

I used Cloud 9 (C9.io, 2016) to store my PHP files and also to access my MYSQL database through PhpMyAdmin. Cloud 9 is an open source online integrated development environment (Wikipedia, 2016). I chose to use PhpMyAdmin because it can be integrated into your Cloud 9 workspace and because it generates SQL queries for you using its GUI (Graphical User Interface).

### 2.3.3 Backend

I created a MySql database online using PhpMyAdmin and connected to this using PHP files in Cloud 9 (C9.io, 2016). This will provide remote access to my database whenever I need to get or post information in my app. I found Cloud 9 very flexible as I was able to connect to PhpMyAdmin through the cloud 9 terminal and I found it user friendly as there is plenty of documentation available online which was incredibly helpful when I ran into some difficulties in development. I had trouble setting up PhpMyAdmin at first but soon found help on the Cloud 9 website (C9.io, 2016).

### 3. System

In this section the requirements of the FestFinder application will be outlined and explained in detail. It will include system design and architecture of the application including diagrams where necessary to help illustrate the architecture used. It will also show how the components of the application were implemented using a detailed description, screenshots and code snippets. Finally, testing, GUI layout, customer testing and evaluation will also be covered in this section.

#### 3.1 Requirements

In order to gather the requirements for my application I spoke to around 24 people that included; friends, colleagues, students and family who, like myself, are also music festival enthusiasts. From these discussions I gathered some ideas and requirements for my app. I conducted a free online survey using SurveyMonkey.com(Surveymonkey.com, 2016) and posted it on Facebook. After receiving a substantial amount of feedback I was able to identify a gap in the market for FestFinder. 95.24% of the people who took the survey had not yet seen an app that holds multiple festival information on it. I also found my target audience which is 18-25 year olds. Finally, the scope of my project is to be set to just hold data for festivals located Europe for the time being. *(See appendix 7.2 questions; 1, 3 and 7).*

##### 3.1.1 Functional requirements

From the above I was able to identify functional requirements for my app that will be explained below.

- **Login:** The first function the user will be presented with will be to login to FestFinder. If they have not yet yet registered they can click a TextView widget that redirects them to the registration page. Once logged in the user can avail of the various functions the application has to offer.
- **Register:** If a user has not got a FestFinder account then they can register their details for one and then login after to make use of the various functions in the application.
- **List Search:** The application will provide the user with a simple way of searching for festivals through a list view. By using a list adapter I was able to design how each list item looked and included the name, city and country flag of each festival from an array.
- **Map Search:** The application provides the function of searching for festivals on a Google map. The coordinates of festival locations are stored locally and retrieved as custom markers on the map. A popup states the name of the festival and the dates it commences. The user will be able to tap on the information window and enter the festival information page from here.

A search bar allows the user to search for any locations in the world. There is also a current location feature so users can see exactly where they are on the map at any given time.

If the user does a long click on the map a custom tent marker is dropped.

However, I was not able to save the tent marker when the activity closes but it is definitely something for me to look into for future implementation.

- **News:** The application will provide updates for the user to check and see date changes and other relevant information in the form of a twitter feed using the Twitter API. The user can enter a hashtag and see the results in a list view.
- **Guide:** The application will also provide a festival guide to give people some tips on how to make the most of their festival experience be it their first or tenth time to go to a festival. This data is retrieved from my database in the form of JSON data.
- **Weather:** A final feature I decided to add that can sometimes be handy when out and about was a current weather page using the Yahoo weather API. The user can click to see the current temperature with a different image being pulled for resources for every kind of forecast.

### 3.1.2 Newly Identified Requirements

- **Checklist:** One feature I had not initially intended to implement was a checklist. I decided to include a checklist because whether it be a holiday you are going on or an event you are going to, there has been one time where we have forgot to bring something important and I felt that a checklist would help people to remember what to bring and tick them off as they have them packed.
- **Ticket sales:** Another feature that I decided to add during development was a set of pages where a user could post tickets for sale with their location and contact details and price of the tickets. Users can also browse through a number of available tickets and click to view each post at a time to reveal more details.

### 3.1.3 Dropped Requirements

- **Profile:** I had originally planned on developing a profile page for each user but in the end I decided not to go ahead with this. One reason was because I think there are already many very popular social media sites and applications available to users so trying to make my application social media focused would have taken away from the main purpose of my application.

- **Favourites list:** Along with the profile there would have been a favourites list for users to add their favourite festivals to a list for easy access. Again I decided not to implement this because I thought it would have been a nice feature to include in a profile page but since I have not developed the profiles then I decided to leave the favourites list as well and concentrate on other features of the application.

### 3.1.4 Use cases

In this section I have included my use case diagrams. These diagrams and their descriptions help understand the functional requirements of the application.

#### Requirement 1: Login/Register

##### Use Case

##### **Scope**

The scope of this use case is logging in to the application

##### **Description**

This use case describes the process of logging in or registering

##### **Use Case Diagram**

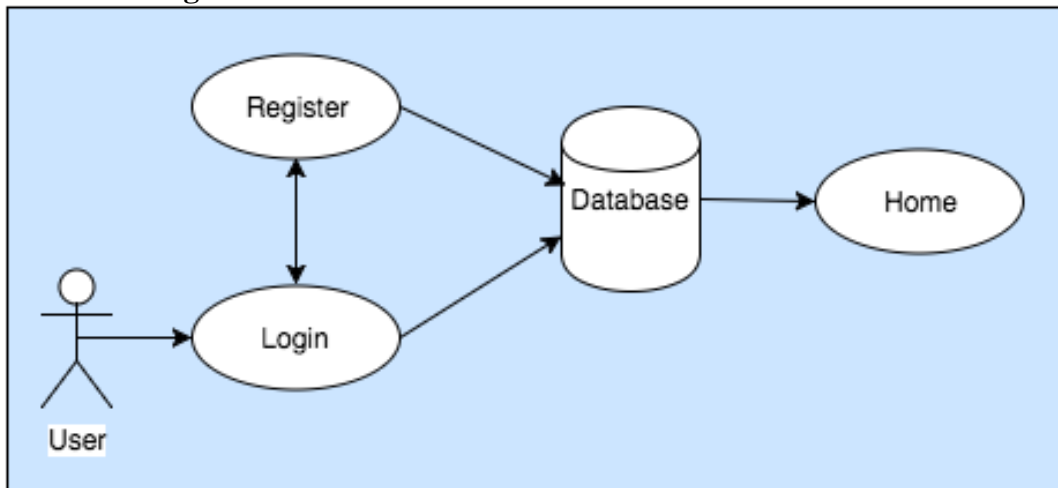


Figure 3.1

##### Flow Description

##### **Precondition**

The system is idle

##### **Activation**

This use case starts when a user tries to log on

##### **Main flow**

1. The system displays a login page
2. If the user has not yet registered they can navigate to register page – see A1
3. The system checks the database for the username
4. The system checks the corresponding password for that username
5. The system verifies the information and logs in
6. The system displays home page

**Alternate flow****A1: Register**

1. The system displays registration page
2. The user enters personal details for registration
3. The database saves registration details
4. The system displays login page
5. The use case continues at position 1 of main flow

**Alternate flow****A2: Details not found**

1. The system cannot find the user information
2. The system asks the user to try again
3. The use case continues at position 1 of main flow

**Termination**

The use case terminates when the user logs in or registers successfully.

**Post condition**

The system goes into a wait state on the home page

**Requirement 2: Search****Use Case****Scope**

The scope of this use case is searching for festivals using a list and Google Maps

**Description**

This use case describes the process of searching for a festival using the Google maps API

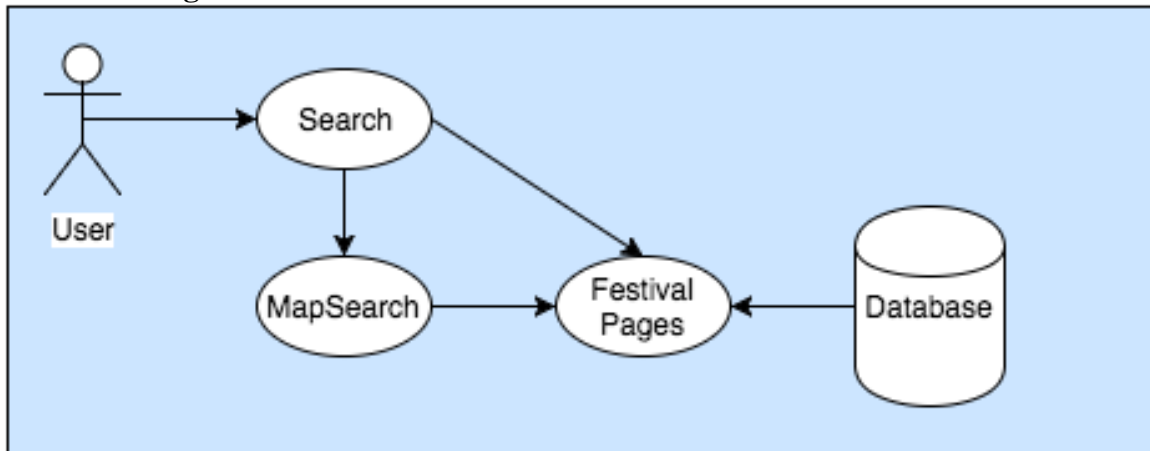
**Use Case Diagram**

Figure 3.2

**Flow Description****Precondition**

The system is in a wait state on the home page

**Activation**

This use case starts when a user clicks on the Search button

**Main flow**

1. The user enters the Search page
2. The user clicks on an item
3. The system displays the festival information page

**Alternate flow****A1: Map Search**

1. The user clicks the floating action button
2. The system displays the Map Search activity
3. The user can click on a custom marker
4. The system displays an info window
5. The user clicks the info window
6. The system displays the festival information page

**Termination**

The use case terminates when the user selects a festival from the list or marker from the maps and is brought to the festival info page or the internet connection is lost.

**Requirement 3: Checklist****Use Case****Scope**

The scope of this use case is using the Checklist

**Description**

This use case describes the process of using the checklist and saving to the SQLite database

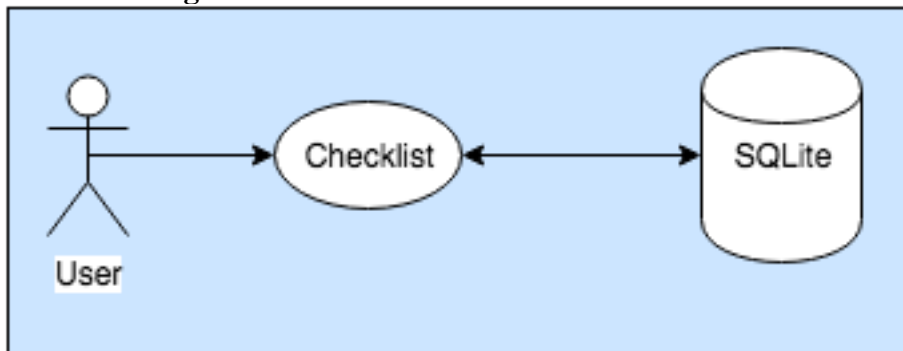
**Use Case Diagram**

Figure 3.3

**Flow Description****Precondition**

The system is in a wait state on the home page

**Activation**

This use case starts when a user clicks on the Checklist button

**Main flow**

1. The user enters the checklist page
2. The user types an item to add to the checklist
3. The user clicks the add button
4. The item is saved to the SQLite database and into the listview
5. The user long clicks an item
6. The SQLite database deletes the item from the table

**Termination**

The use case terminates when the user exits the checklist page.

**3.1.5 Data requirements**

In this section, I will describe the data requirements which are essential for implementing some of the functions already outlined above.

- **SQL Database:** An SQL database was created using PhpMyAdmin through Cloud 9. It used PHP files to connect to and retrieve and save data to my database. I will use this database to store information for users, festivals, ticket sales and the guide.
- **SQLite Database:** A SQLite database is used to store items of the checklist on local storage.

**3.1.6 User requirements**

In this section of the document, the user requirements are described. These are essential requirements the user must have in order to download and use the application.

- **Android device:** The user must possess an Android phone as this is the targeted operating system of the application.
- **Play Store:** The user must have access to the Play Store to download and use the application.
- **Internet access, WIFI:** The user will need internet access, mobile data or be connected to WIFI to download the device from the Play Store. Constant access to the inter is needed to make use of the application's features and different functions.

**3.1.7 Environmental requirements**

In this section, the environmental requirements are described. They are essential requirements that must be present when developing the application.

- **Windows:** The application was developed using the college machines on windows and my own laptop that was connected to Citrix to avail of the college resources and to access files.
- **Android phone:** For testing the functionality and design of the application.
- **Android Studio:** For developing and designing the application.
- **Android SDK:** Software development kit, provides the tools needed to develop an Android application.

- **Internet access:** Required to access the available android documentation online and also to use Cloud 9 to write PHP.
- **Photoshop & Paint:** Used to edit graphics and icons that will be used in my application.

### 3.1.8 Usability requirements

This section will describe the usability requirements. These define objectives to meet during the interface and design process.

- **Understandable:** Interface elements, such as menus, should be easy to understand. Each help button on a given screen must provide at least one 'topic' for each control on the screen.
- **Attractive:** Screen layout, colour and graphics should be appealing. The application should be pleasing to the eye and have a modern look.
- **Learnable:** The system should be easy to learn and come to grips with. It is possible to reach any given system function from the main window in less than 3 clicks.
- **Operable:** The interface actions and elements should be consistent. A new user must be able to use the system without the need for training.

## 3.2 Design and Architecture

The goal of this Android application is to ensure that, when running on a phone, the application doesn't take up too many resources. The space of Android devices must be taken into consideration. To achieve this goal most of the data in the application is stored remotely and accessed via background tasks.

## 3.3 Implementation

This section of the technical report will define the various methods used for implementing the FestFinder application.

### 3.3.1 Setup

In order to start developing my application I first needed to choose a development environment. The options I looked into included Eclipse (Eclipse, 2016), Netbeans (netbeans, 2016) and Android Studio. In the end I chose Android studio because having previously worked with eclipse I found it a bit cumbersome. Some articles online suggested to switch over to Android Studio from eclipse and give it a go. So I decided to use Android Studio first task I completed was to download it. Once it was downloaded I then began to configure it by downloading the Android SDK. The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications. Once configured I then setup the emulator so that I could run my application and test functionality throughout development. As I didn't always have access to an Android device the emulator was a key factor to the development of my application. I used version control through Github (GitHub, 2016) in the event that anything happened and I lost my project. From Android Studio, I was able to send my



project to Github after making substantial changes. This way, my project was backed up online and I could retrieve it if anything went wrong.

### 3.3.2 Project Creation

Once everything was setup and ready for development, I then began to create my project in Android Studio. I selected 'New Project' in the Android Studio file menu, named my project and then proceeded to select the minimum SDK the project will run on, which is level 15, and named my first activity '**MainActivity**'. This activity is the main screen of the application where users can click buttons that will take them to ne activities within the application. For navigating through the application I set onClickListeners for the buttons and then used an Intent to tell the system what activity to start on clicking the button as seen below.

```
public void onClick(View v) {  
    Intent intent = new Intent(MainActivity.this, SearchActivity.class);  
    startActivity(intent);  
}
```

This example will start the **SearchActivity** upon clicking the Search ImageButton located in the **MainActivity** of the application. These intents can be found through my source code and enable users to navigate from activity to activity.

### 3.3.3 Authentication

In order for a user to gain access to the application they must register their details on a register page and then login with the details they have registered. A MYSQL database stores user credentials so they can then login to the application whenever they please. I used PhpMyAdmin to create the database online. Then I connected to the database using PHP files.

```
$con = mysqli_connect($host,$user,$pass,$dbname);  
if(!$con){  
    die("Connection failure". mysqli_connect_error());  
}
```

The above PHP code gets my database variables that I have declared and uses them to connect. If connection is unsuccessful a connection failure error will appear as seen above. This Config.inc.php file is then required in all of my PHP files in order to connect to my database.

When a user enters their credentials in the register Textviews the data is saved to the database. If they do not fill in all the fields or the password and confirm password do not match, an error will appear to the user telling them something went wrong. Successful registration takes the user to the login page where they can now login using their previously registered details. I used password hashing for added security of user details in case anybody gained access to my database.

```
//hashes the passwords using BCRYPT function with a cost of 10  
$storePassword = password_hash($password, PASSWORD_BCRYPT, array('cost' =>  
10));
```

The cost determines how secure the hash is. A higher salt figure means a stronger hash, but this can also slow down software if the cost value is too high. (Wikipedia, 2016)

### 3.3.4 JSON data

To save application resources and space I retrieved most of the data in my application from my database in the form of JSON which then needed to be parsed to the various activities. In my PHP files I used an SQL query to select the rows I needed depending on the activity of the application. A `json_encode` was used to encode the data to JSON.

```
print(json_encode($data));
```

Once the data was in JSON format I then used a `URLConnection` in Android studio in the background task of the class to parse the data.

```
URL url = new URL(params[0]);  
connection = (URLConnection) url.openConnection();  
connection.connect();
```

I used a for loop to iterate through the data and to save it to a String called `jsonobj`. I then returned the `jsonobj` so I could set it to where I wanted the various object to be on the screen.

```
JSONObject jsonObj = jsonarray.getJSONObject(0);  
  
return jsonObj;
```

For my festival information activities, I declared an empty string. I then searched the `jsonobj` for the 'date'. I then declared the date Textview of the festival information layout file. Finally I set the text of the TextView to contain the `date1` string of json data.

```
String date1 = null;  
  
date1 = jsonObj.getString("date");  
  
TextView date = (TextView)findViewById(R.id.dateTV);  
  
date.setText(date1);
```

The above was implemented for a number of Textviews simultaneously containing more information such as; country, city, genre, general information and ticket information.

### 3.3.5 SQLite

I used the SQLite database for saving the checklist items. SQLite is the local database on an Android device. Anything saved to the SQLite db will only be available to a user on that device. The W3Schools site helped me with creating and implementing the SQLite database as this was my first time to do so. When a user enters an item and clicks the add button, the system calls `COL_NAME`, the column for the item to save to, and then gets the text entered and saves it to a string. It is then inserted to the SQLite database and a `SimpleCursorAdapter` is used to put the item into the Listview below. An

OnItemLongClickListener is used on the listview to delete items when they are long clicked. It finds the position of the click, gets the string and deletes it from the SQLite database.

```
myCursor.moveToPosition(position);
String rowId = myCursor.getString(0);
myDb.delete(MyDbHelper.TABLE_NAME, "_id = ?", new String[]{rowId});
```

### 3.3.6 Google Maps

I used the Google Maps API in my application in order to have an activity with a map so I could put the locations of festivals for users to see them. A problem I had was using the info windows of each marker to navigate to the festival information pages for that festival. I had trouble determining which info window was being clicked to set an Intent to navigate to the information page. I found a post on Stackoverflow.com (Stackoverflow.com, 2016) that helped me with this problem. I used a Hashmap to save the markers to. This allowed me to give the markers a unique id and to also give each marker an associated class.

```
private Map<Marker, Class> allMarkersMap = new HashMap<>();
```

```
allMarkersMap.put(marker1, UltraActivity.class);
```

This allowed me to set an onInfoWindowClickListener on the markers which found the Id of that marker and the class allocated to it.

```
Class cls = allMarkersMap.get(marker);
Intent intent = new Intent(MapSearchActivity.this, cls);
startActivity(intent);
```

I also used Geocoding in my **MapSearchActivity**. I included a searchbar at the top of the page that allows users to search for locations and addresses around the world. The geocoder takes in the address and changes it to lat and lng coordinates that the API uses to find the location entered. It adds a marker at the location and takes the entered address and uses it as the title for the marker.

```
LatLng latLng = new LatLng(address.getLatitude(), address.getLongitude());
festMap.addMarker(new MarkerOptions().position(latLng).title(location));
```

### 3.4 Graphical User Interface (GUI)

This section indicates the Graphical User Interface that I mocked up using Balsamiq.com. and the actual GUI of the FestFinder app. Using mock-ups helped me to determine how my application would look and how users could navigate through the app.

#### 3.4.1 Authentication

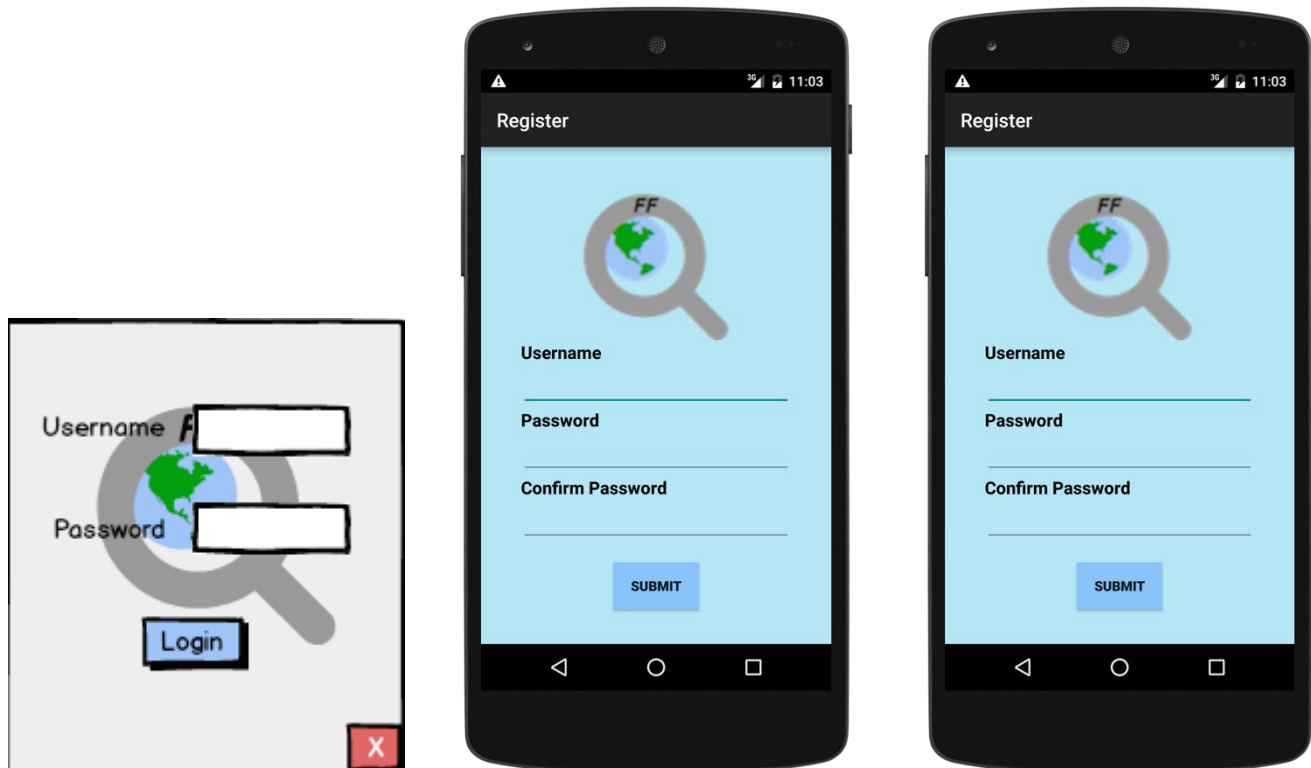


Figure 3.4

These are the first pages a user is presented with when they open the FestFinder app. Users are presented with the login page and if not yet registered they can navigate to register and fill in their credentials to sign up for a FestFinder account. The mockup helped me to make some design decisions. I decided that the logo behind the Textviews and EditTexts didn't look right and changed it in my final design.

### 3.4.2 Home



Figure 3.5

This is the main screen of the application after a user logs in. The layout of this screen has changed too since development has finished. Like the login and register pages I have removed the logo from the background as it didn't look right when I ran it on an emulator. The profile page has been replaced with the checklist. I have added the ticket sales page to this home screen too. The "news" and "guide" buttons can now be found under a "More" logo button along with the current weather page.

### 3.4.3 Profile Pages

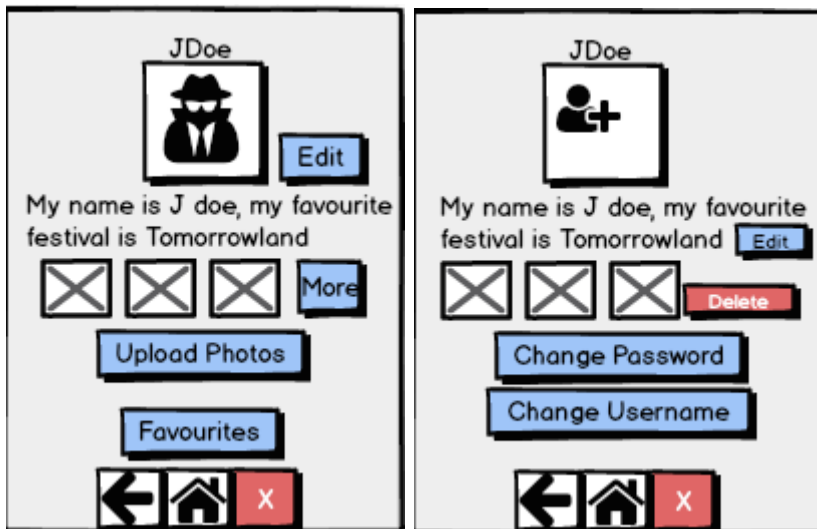


Figure 3.6

This was a mockup for the profile pages I had first planned on including but decided to remove from the application during development.

### 3.4.4 Favourites

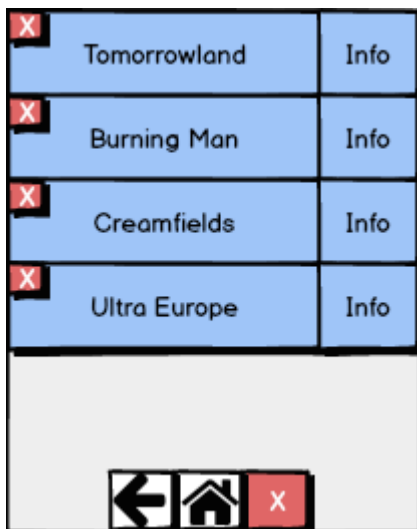


Figure 3.7

I also decided to remove the favourites page, but this was roughly how the page would have looked.

### 3.4.5 Search

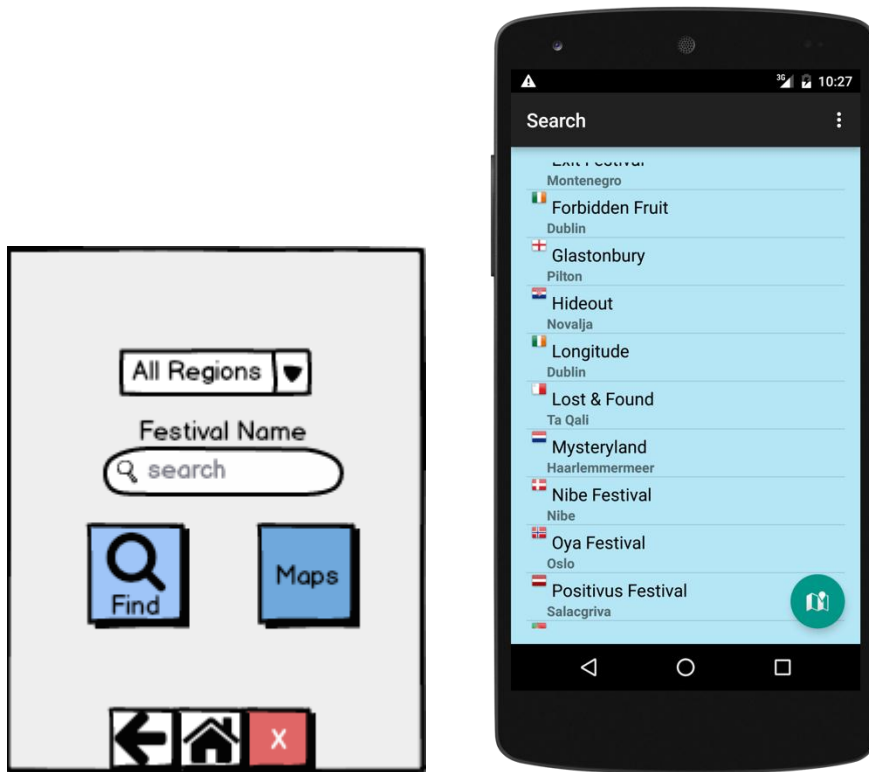


Figure 3.8

The above shows the initial and final design of the Search page. I have now made the search page simpler and more user friendly. A Listview shows the available festivals on the application with their country flag and name of the city they are located in. Users can stroll through the list to find their desired festival. As there are only 23 festivals available for now, I decided to make the design simpler and easier for the user to navigate and find festivals. In the future with more festivals on the application I could incorporate a search bar and a spinner to filter by region. A floating action button is used with a maps icon to navigate to the Map Search page of the app.

### 3.4.6 Map search

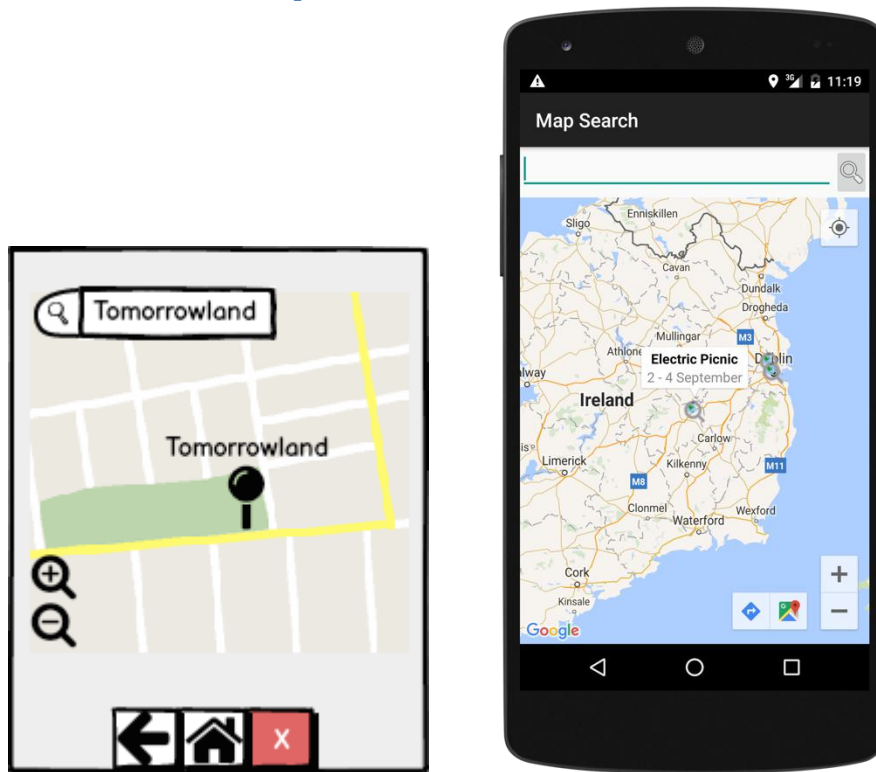


Figure 3.9

The Google map API is used on this page of the application. The above mockup is very similar to the final design of the Map Search page. The coordinates are stored locally. Each marker is put in a Hashmap along with a class for each marker. So when the user clicks on a marker's info window, the application finds the marker that has been clicked and the class allocated to it and then starts that activity using an Intent. Info windows contain the title and dates of festivals.

There is also a search bar where users can search for addresses on the map. Users can also click on the current location button to see where they are at any given time on the map. If the user long clicks on the map, a custom tent pin is dropped with a snippet saying "Tent is here!" This could be implemented so that users can find their current location and then drop a pin to help them remember where they left their tent at busy festivals.



### 3.4.7 Festival Info

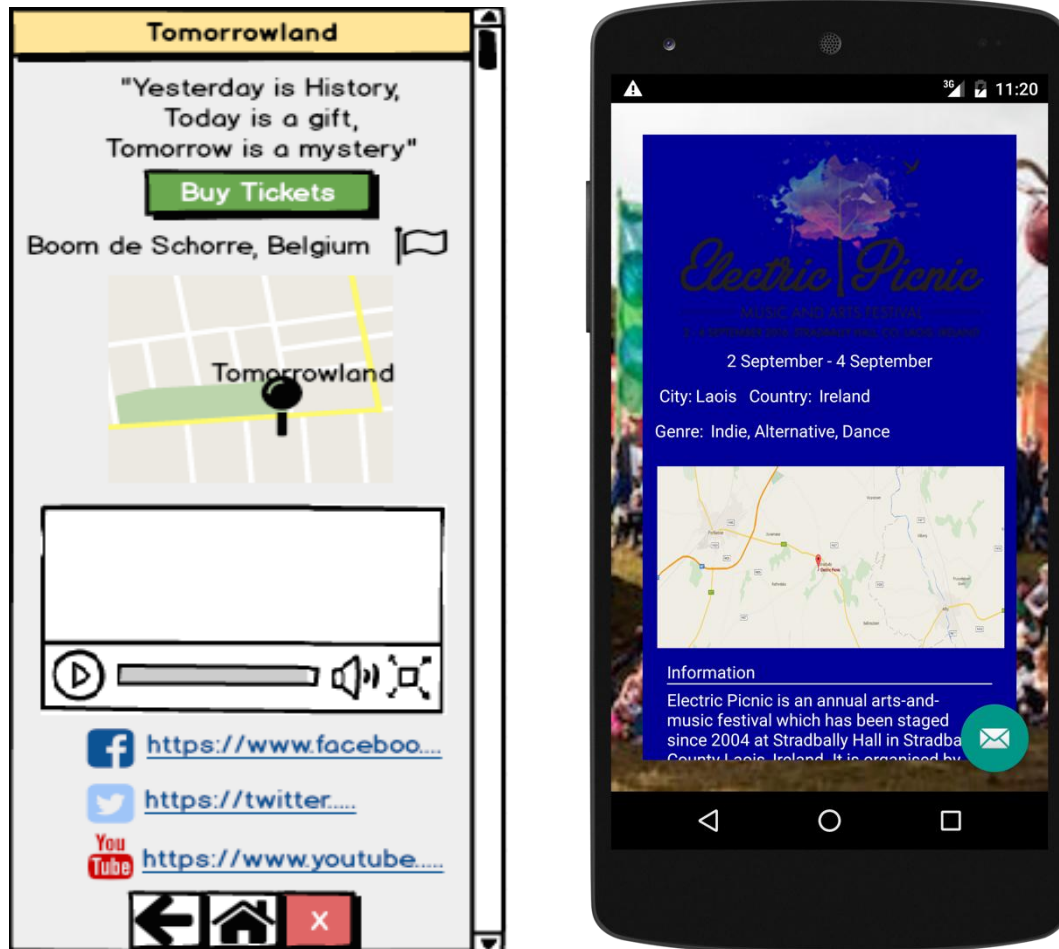


Figure 3.10

The shows shows the layout for the festival information pages. I needed to be careful not to fill this page with too much information and the layout of what I do include will need to look professional and pleasing to the user's eye.

From this page the user can select 'Buy tickets' which will bring them to the official sellers of tickets for that festival. This will mean exiting the application and being brought to a site on their phones browser. The page also includes an area map of the festival, general information, ticket information, promotional multimedia and social media links. The data is retrieved from my database as JSON and parsed into the various widgets on the page.

### 3.4.8 Other

As I decided to implement the other pages mentioned only during development, I have not got any mockups for these pages but the real screenshots can be seen below.

#### 3.4.8.1 Checklist

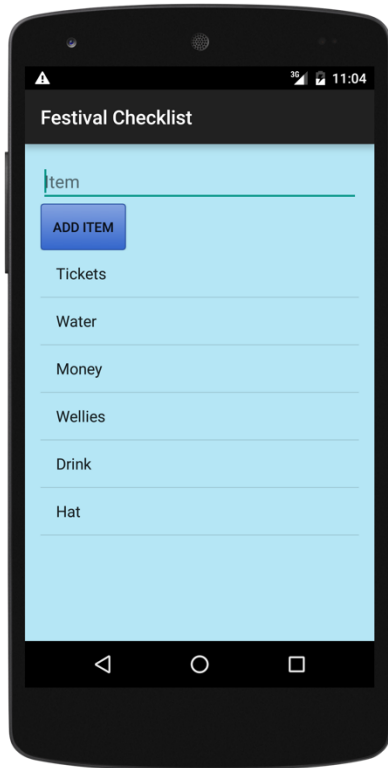


Figure 3.11

This is the festival checklist. Users can add items to the checklist. Once added, SQLite saves them locally on the device. An `onLongItemClickListener` is set so when a user long clicks an item, the system finds its ID in the SQLite database and removes it from the table and Listview.

### 3.4.8.2 Tickets

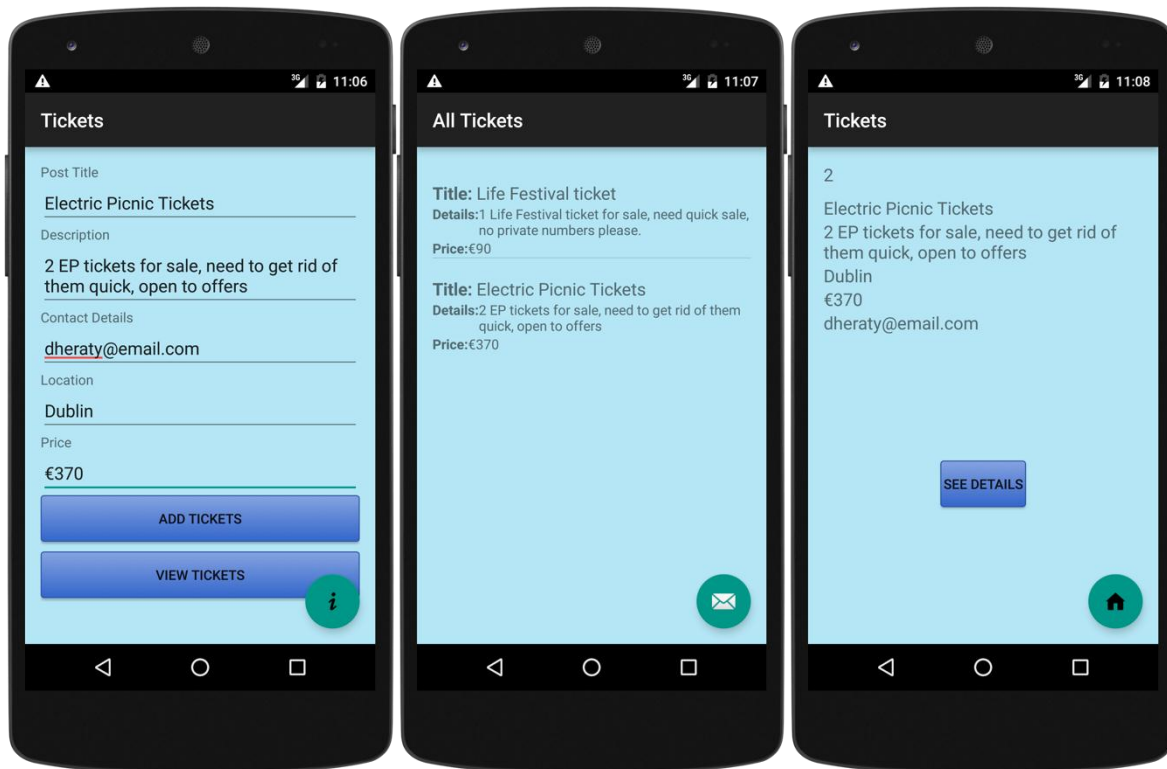
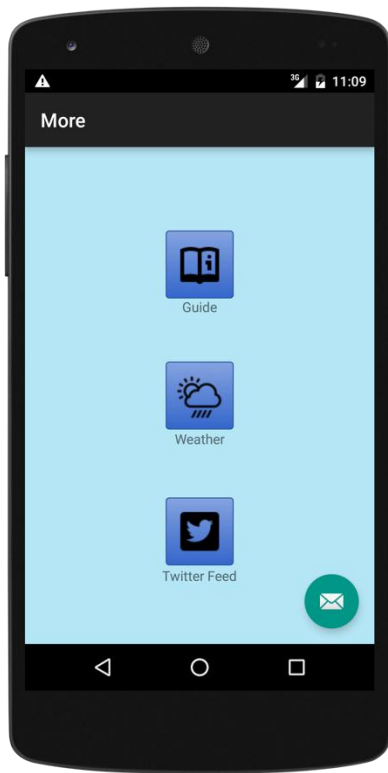


Figure 3.12

The Tickets activities were implemented late into development. I decided I needed some features that would hold the user in the application. The user can enter their details, which are saved remotely. When the user clicks 'Add Tickets', the data is saved to my database and the next **ViewAllTicketsActivity** is started.

Here the user can see all of the ticket posts on the application. If the user clicks a post it will bring them to the **ViewTicketActivity**. This page shows more details about the tickets for sale and a 'See Details' button changes the state of the contact details from INVISIBLE to VISIBLE, revealing the contact details of the advertiser. The following tutorial helped me incredibly to implement this functionality (Khan B. 2016).

### 3.4.8.3 *More*



*Figure 3.13*

I decided to have a **MoreActivity** to hold content that are not my main requirements.

### 3.4.8.4 Guide



Figure 3.14

The festival guide gives advice to festival goers. This data is retrieved from my database online as JSON. A **ListViewAdapter** is used to customise the layout of each listview item. I used a tutorial to help me get started with this (YouTube 2016).

### 3.4.8.5 Weather



Figure 3.15

The Yahoo API was used, with the help of a tutorial (YouTube 2016), to get current weather data. The API searches through JSON data for the channel number (a number with a corresponding weather image saved as a resource file), the temperature, a description and the location of the weather.

### 3.4.8.6 Twitter Feed

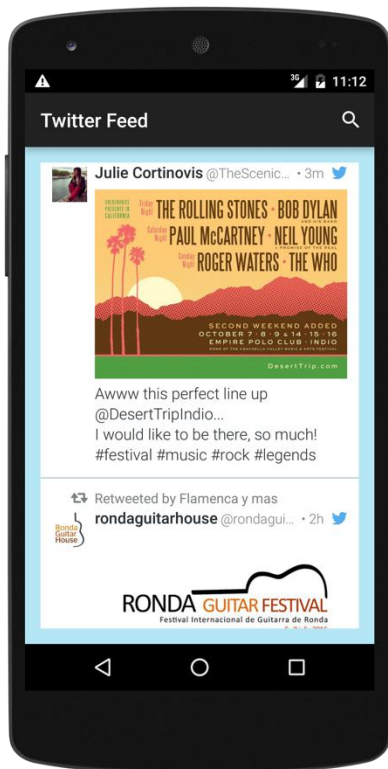


Figure 3.16

The Twitter API was used, with the help of a tutorial (YouTube 2016), to act like a news feed. Users can search by hashtag with the default set to #festivals.

## 3.5 Testing

### 3.5.1 Unit Testing

Each time I implemented a substantial amount of code I ran the application to determine whether the code worked or not. I ran into some errors that I was able to find solutions to. These can be seen below.

Activity	Error	Solution	Status
MapSearchActivity	Searchbar left empty and clicking search button crashes the application	<pre>if (locationText.length() == 0) {     Toast.makeText(this, "Please Enter Address", Toast.LENGTH_SHORT).show();     return; }</pre>	Fixed
ChecklistActivity	Empty EditText and clicking add, adds a null item to the listview	<pre>if (mText.length() == 0){     Toast.makeText(this, "Please enter an item!", Toast.LENGTH_SHORT).show(); }else</pre>	Fixed
TicketsActivity	Empty Edittexts passing over null strings	<pre>if(editTextName.length() == 0    editTextDesg.length() ==0    editTextSal.length() == 0){      Toast.makeText(this, "Please fill in all the above fields",     Toast.LENGTH_SHORT).show(); }else</pre>	Fixed
TicketsActivity	Price Edittext accepting characters instead of numbers	in .xml set <code>android:inputtype="number"</code>	Fixed



### 3.5.2 Persona Testing

My application is aimed for people between the age of 18 up to 30. But one requirement is that the application is user friendly and suitable for all ages. So in order to see whether I met this requirement I gathered results from 4 users sampling across the age range of 20 – 55 years old. This would give better volume and range of feedback.

Name	Age	Profession	Smart Phone User?	Interests	Favourite Application Feature	Comments
Sean	55	Car Salesman	No	Music, Reading, Golf	Checklist	Easy to use application and nice design.
Karl	28	Electrician	Yes	Fitness, Music	Map Search	Track friends locations to see where they are at festivals if you become lost
Anne	44	Receptionist	Yes	Reading, Cooking, Walking	Checklist	Would like to see more pictures
Alan	20	Dj	Yes	Festivals, Concerts, Mucis, Djing	Ticket sales, Map Search	More information on festival pages. Page dedicated to artists and information about them, upcoming gigs, latest music

### 3.5.3 Stress Test

To stress test the application I used Android built in UI/Application Exerciser Monkey (Monkey, 2016). The Monkey is a program that runs on your emulator or device and generates pseudo-random streams of user events such as clicks, touches, or gestures, as well as a number of system-level events. You can use the Monkey to stress-test applications that you are developing, in a random yet repeatable manner. Monkey is pseudo random hence why it is repeatable. This means the same series of clicks can be replicated by using the same seed for the random number generator.

In the command line I changed the directory to the platform-tools folder of my sdk. I then ran monkey using:

```
./adb shell monkey -p com.example.x11401282.festfinderapp -v 1000
```

This command generates 1000 random actions in my application and records the results of the stress test in the terminal.

I ran monkey three times. I started at 500 commands and increase each time by 500. The application didn't crash once. This is down to extensive testing throughout development.

### 3.6 Evaluation

#### 3.6.1 System Usability Scale

I used a System Usability Scale (Wikipedia, 2016) to evaluate my application by asking the following questions along with a likert scale.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

	<b>Strongly Agree</b>	<b>Agree</b>	<b>Neither agree nor disagree</b>	<b>Disagree</b>	<b>Strongly disagree</b>
<b>1.</b>	12	12	2	0	0
<b>2.</b>	0	-1	-2	-2	-8
<b>3.</b>	16	12	0	0	0
<b>4.</b>	0	0	-4	-2	-8
<b>5.</b>	4	18	2	0	0
<b>6.</b>	0	0	-4	2	0
<b>7.</b>	8	18	0	0	0
<b>8.</b>	0	-6	3	-8	0
<b>9.</b>	8	21	0	0	0
<b>10.</b>	-8	0	-2	-4	0

From the above results I calculated a SUS score of 81. According to UsabilityGeek.com (Usabilitygeek.com, 2016) “80.3 or higher is an A. People love your site and will recommend it to their friends”. The average score for systems to accomplish is 68 (Measuringu.com, 2016).

### 3 Future work

In the future the scope of FestFinder could be widened to include festivals worldwide. This would present the application with the opportunity to expand by reaching a bigger audience of users. I would like to save the festivals remotely and pull the coordinates from the database to free up space and resources in the application. I would also like to implement a way to save the custom 'Tent' pin to the Maps activity and then be able to delete it.

I would also like to widen the project scope to not only cater for music festivals, but also to include different categories like film, arts, comedy, culture and food. In turn, this would help reach a larger target audience that would benefit the application in terms of growth as it would no longer just target 18-25 year olds. This would require a lot more time developing and designing the app to do this so this is why I chose to just target music festivals for the purpose of this project.

Another future intention would be to make it multi platformed and to expand into the IOS market as well as the Android market. This could double the amount of potential users to the application.

## 4 Conclusion

The above document outlines my preparation for the FestFinder application. It describes how I researched and came to the decision to develop FestFinder. It details the steps taken to begin development and defines how development was conducted. It also portrays how the system was tested and evaluated post development.

In my opinion the project was a success. I did not meet every requirement I set out to meet, but I feel I have substantial reasons for not meeting these requirements and have found appropriate and suitable substitutes. From my evaluation it is clear that the system is not a failure. Persona testing concludes that people of all ages can use the system and find it attractive and useful.

There were some setbacks along the way but with some research and help online I was able to overcome these setbacks and find solutions to my problems in order to meet the project brief and deadline.

## 5 References

### 3.6.2 Online References

1. Google (2014), Android, the world's most popular mobile platform. Available: <http://developer.android.com/about/index.html>. (Last accessed 01/02/16).
2. Statista. (2016). Number of Google Play Store apps 2016 | Statistic. [online] Available at: <http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/> [Accessed 6 May 2016].
3. Play.google.com. (2016). [online] Available at: <https://play.google.com/store/apps?hl=en> [Accessed 6 May 2016].
4. Play.google.com. (2016). [online] Available at: <https://play.google.com/store/apps/details?id=com.tomorrowland.tomorrowland&hl=en> [Accessed 6 May 2016].
5. Time Out London. (2016). The 50 best music festivals in the world. [online] Available at: <http://www.timeout.com/london/music/the-50-best-music-festivals-in-the-world> [Accessed 6 May 2016].
6. C9.io. (2016). davidh2992. [online] Available at: <https://c9.io/davidh2992> [Accessed 6 May 2016].
7. GitHub. (2016). Build software better, together. [online] Available at: <https://github.com/> [Accessed 8 May 2016].
8. W3schools.com. (2016). W3Schools Online Web Tutorials. [online] Available at: <http://www.w3schools.com/> [Accessed 8 May 2016].
9. Stackoverflow.com. (2016). Stack Overflow. [online] Available at: <http://stackoverflow.com/> [Accessed 8 May 2016].
10. Usabilitygeek.com. (2016). [online] Available at: <http://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/> [Accessed 8 May 2016].

11. Festicket. (2016). The Easiest Way to Book Your Next Music Festival – Festicket. [online] Available at: <https://www.festicket.com/> [Accessed 9 May 2016].
12. Draw.io. (2016). Flowchart Maker & Online Diagram Software. [online] Available at: <https://www.draw.io/> [Accessed 9 May 2016].
13. Duffy, S. (2016). *Fake One Direction and Beyoncé tickets seized by gardai*. [online] TheJournal.ie. Available at: <http://www.thejournal.ie/fake-tickets-1292584-Jan2014/> [Accessed 9 May 2016].
14. Wikipedia. (2016). *Java performance*. [online] Available at: [https://en.wikipedia.org/wiki/Java\\_performance](https://en.wikipedia.org/wiki/Java_performance) [Accessed 9 May 2016].
15. Mysql.com. (2016). *MySQL :: Top Reasons to Use MySQL*. [online] Available at: <https://www.mysql.com/why-mysql/topreasons.html> [Accessed 9 May 2016].
16. Wikipedia. (2016). *JSON*. [online] Available at: <https://en.wikipedia.org/wiki/JSON> [Accessed 9 May 2016].
17. Surveymonkey.com. (2016). *SurveyMonkey - Log in*. [online] Available at: <https://www.surveymonkey.com/home/> [Accessed 9 May 2016].
18. Eclipse, (2016). [online] Available at: <https://eclipse.org/> [Accessed 9 May 2016].
19. Netbeans.org. (2016). *Welcome to NetBeans*. [online] Available at: <https://netbeans.org/> [Accessed 9 May 2016].
20. Wikipedia. (2016). *Bcrypt*. [online] Available at: <https://en.wikipedia.org/wiki/Bcrypt> [Accessed 9 May 2016].
21. Monkey, U. (2016). *UI/Application Exerciser Monkey | Android Developers*. [online] Developer.android.com. Available at: <http://developer.android.com/tools/help/monkey.html> [Accessed 9 May 2016].
22. Wikipedia. (2016). *System usability scale*. [online] Available at: [https://en.wikipedia.org/wiki/System\\_usability\\_scale](https://en.wikipedia.org/wiki/System_usability_scale) [Accessed 9 May 2016].

23. Measuringu.com. (2016). *10 Things to Know About the System Usability Scale (SUS): MeasuringU*. [online] Available at: <http://www.measuringu.com/blog/10-things-SUS.php> [Accessed 9 May 2016].

### 3.6.3 Tutorials and Resources

1. YouTube(2016). How To Integrate Youtube API In Android Application. [online] Available at: <https://www.youtube.com/watch?v=a4NT5iBFuZs>. ( [Accessed 5 Apr. 2016].
2. YouTube. (2016). Android JSON Parsing - Perform parsing & display result in a ListView. [online] Available at: <https://www.youtube.com/watch?v=KSX4zIhiZIM> [Accessed 8 Apr. 2016].
3. Version, M. (2016). *Icons8 - 25,600 icons + Icon Tools*. [online] Icons8. Available at: <https://icons8.com/web-app/for/androidL/guide> [Accessed 8 May 2016].
4. YouTube. (2016). Android Tutorial: Create a Weather App with Yahoo Weather API - Part 3 of 3. [online] Available at: [https://www.youtube.com/watch?v=gJ9Ny\\_J3tcM](https://www.youtube.com/watch?v=gJ9Ny_J3tcM) [Accessed 1 May 2016].
5. Khan, B. (2015). *Android MySQL Tutorial to Perform Basic CRUD Operation*. [online] Simplified Coding. Available at: <https://www.simplifiedcoding.net/android-mysql-tutorial-to-perform-basic-crud-operation/> [Accessed 25 Apr. 2016].
6. YouTube. (2016). *E-04 part2 Twitter Fabric sdk Searching for twitter feeds*. [online] Available at: <https://www.youtube.com/watch?v=kgR-5Sv9bwQ> [Accessed 30 Apr. 2016].



## 6 Appendix

### 7.1 Project Proposal

#### **FestFinder Application**

Name: David Heraty  
Student No: 11401282  
Email: David.heraty@student.ncirl.ie  
Module: BSc (Hons) in Computing  
Specialisation: Networking and Mobile Technologies  
Date: 22/09/2015

#### **1. Objectives**

The FestFinder application will aim to provide a central location of festival information including; ticket sales, line-ups, locations, news updates etc. It will include an area to arrange the sale/swap of tickets. It will also allow users to upload their personal experiences of festivals through photos on personal profiles to share with all the users of the application.

The main objective for FestFinder is to provide a central location of information for music festivals all over the world. Being a festival lover I have yet to come across an application which, in one location, holds information of the festival such as; festival location, line-up, ticket price, date, site map and more.

The second objective of the FestFinder application is to be easy navigable and user friendly. A good design is key for a successful application in a market of high standard applications being developed and designed. Graphically it should be simple in order to avoid design complications and take away from my developing time.

The third objective is to provide legitimate ticket sales to customers. In my experience of buying tickets I have personally been sold counterfeit tickets for more than face value of the original ticket and there is no insurance on these so you lose out on the event and also the extra money spent obtaining the ticket. If ticket sales were provided through links to the official ticket sellers of the festival sites such as festticket.com, then this would rule out the possibility of purchasing counterfeit tickets and result in happy users of my application.

#### **2. Background**

As a DJ, music lover and music festival fanatic, I have found from the results of a free online survey and from searching through the app store and internet that there is no application that holds a lot of information on a variety of festivals worldwide. Usually you would have to search on Google for the festival and retrieve the official website and then you would find out the required information. Sometimes you would have to navigate to a separate site to purchase tickets for the festival. I have yet to find legitimate ticket

sales for more than one festival in the same application. This is where I saw a gap in the market and decided to make my idea a reality for my fourth year software project. There are a lot of festivals that some people wouldn't know about and may be interested in going to, so FestFinder will be the central location for festival managers to advertise their festival by having it alongside the most popular names in the festival industry. Some festivals such as Electric Picnic and Tomorrowland need no introduction as they sell out relatively fast each year. But for smaller less popular festivals, such as Solid Sounds Festival in the US, the application would be of great benefit as the festival management can provide information on their festival including line-up, stages and times of each act, festival map, directions, news, official after movies, sounds and links for legitimate ticket sales. This may help the festival get recognised and become more popular, thus generating more money for the festival to expand and grow in the future. There will be admin login functionality that will allow admins to add their own festivals to the application.

### **3. Technical Approach**

As FestFinder is a central location for festival Information I will need to research what Festivals to put into my application. Having completed a free online survey I found that 54/55 people attend most festivals in Europe so I will concentrate on this region with the option of expanding to other regions.

I will create and connect to an online database using PHP/SQL and the SLIM framework to store and retrieve Festival/Logon/Registration information. I will use the bootstrap framework for layout and designing in the application.

I plan on concentrating on Android development using Android Studio because I have researched it and there seems to be a lot of online help of how to add different features to applications and I am already very familiar with Java from previous years of college. I also chose android because of its popularity among the smartphone market in the world today.

#### **4. Technical Details**

I will use the following implementation languages for my application development; Java, PHP and SQL.

Along with the Android SDK plugin I will use the Google maps API for searching festivals and I have the option of using an RSS feed for news updates.

3G or Wi-Fi will be needed for the application to run properly.

#### **5. Evaluation**

I plan on evaluating my application regularly during development but will leave myself time to thoroughly test it.

I will test the database by entering information and retrieving it to see whether it is showing in the correct table spaces.

I will use an Android device to run my application on to personally test the application and assess the design and user friendliness before having other users test it.

I will system test by asking volunteers to do a list of tasks on my application and ask them to assess each task with regards to ease-of-use. I will also ask them to rate the design of my application using a feedback survey.

## 7.2 Online Survey

### 7.2.1 Introduction

The following survey was created online using SurveyMonkey.com a site for making free surveys. The survey helped to answer some questions I had during my research and helped to identify my target audience, the scope of my application and it also gave me some requirements for the app.

### 7.2.2 Questions and Responses

Question 1:

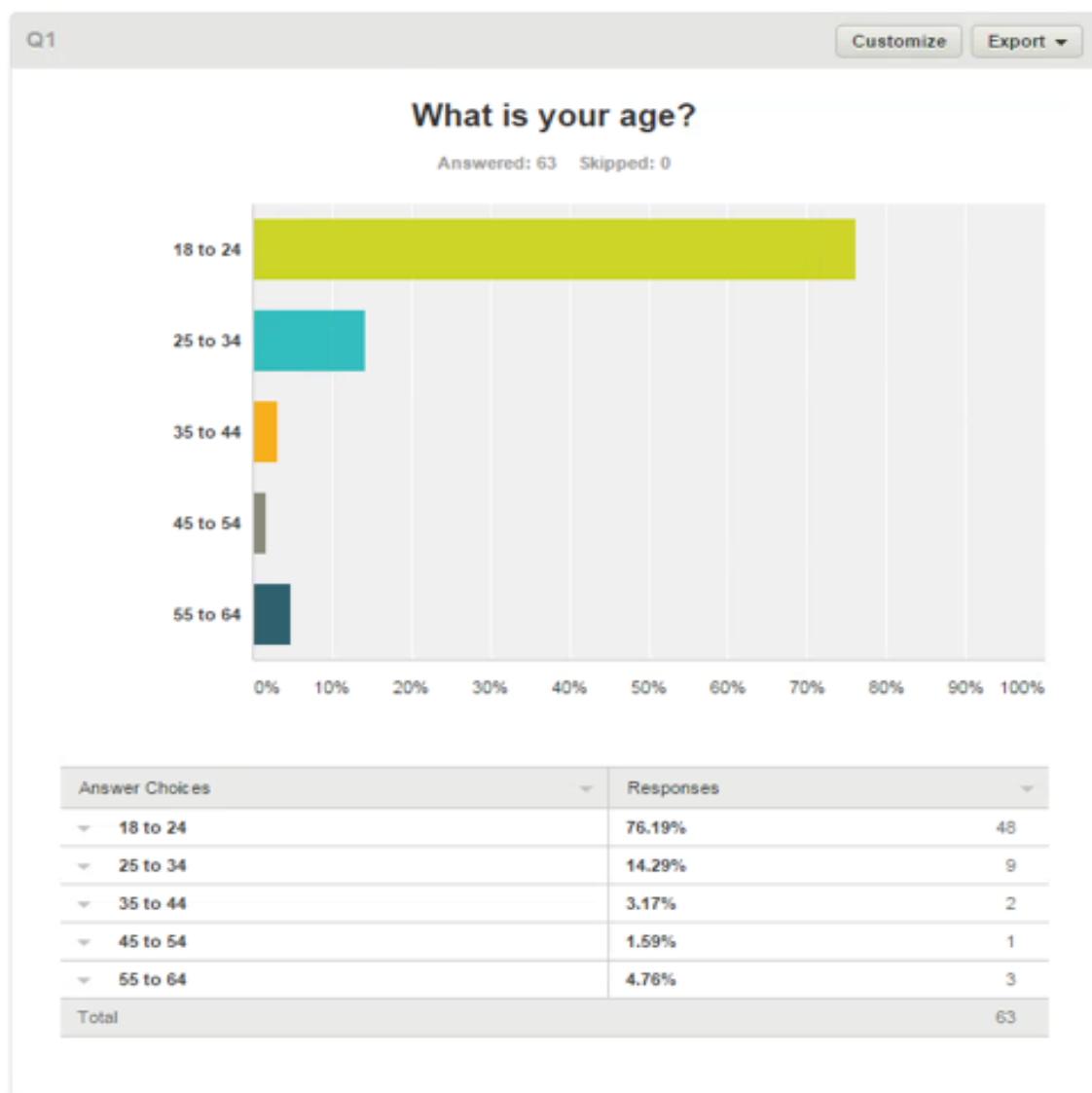


Figure 7.1

## Question 2:

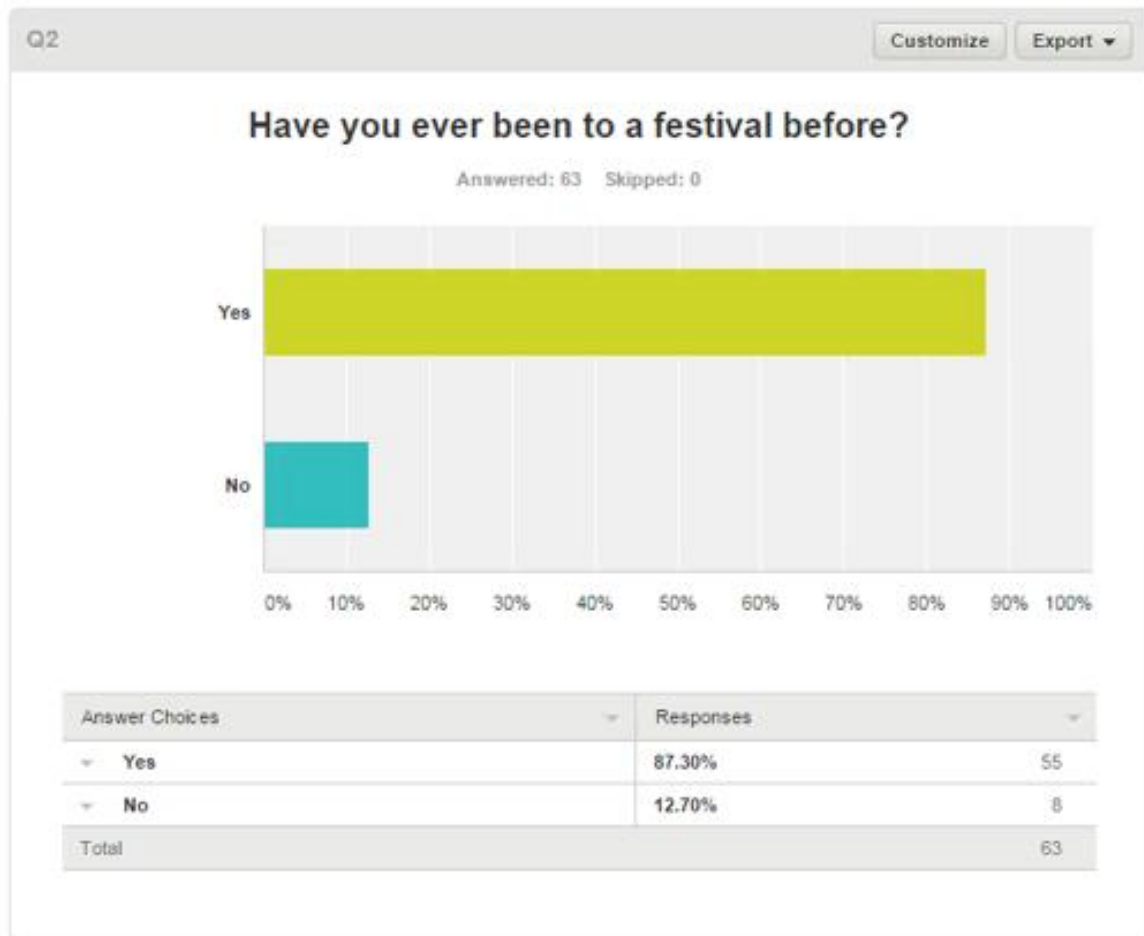


Figure 7.2

## Question 3:

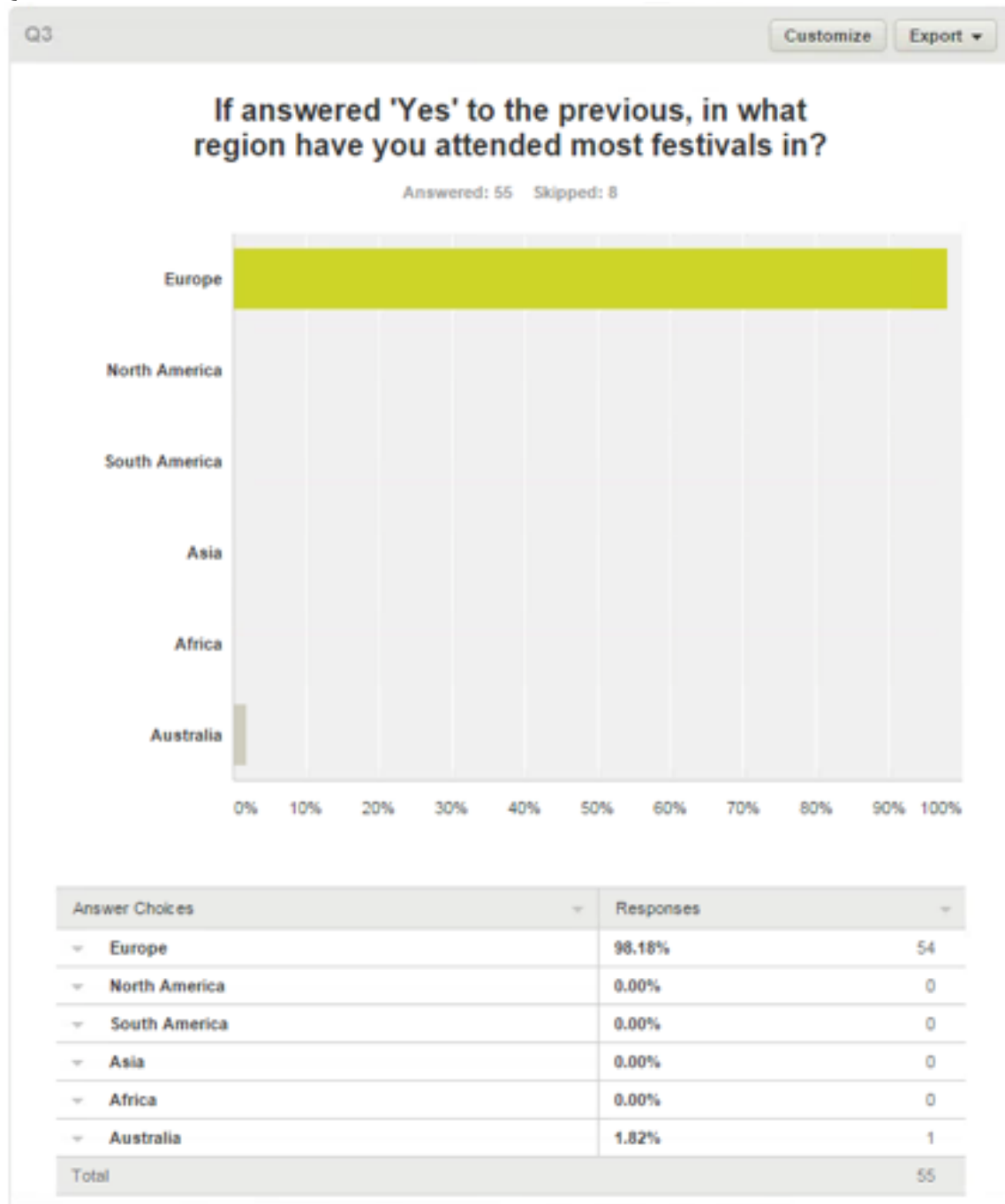


Figure 7.3

Question 4:

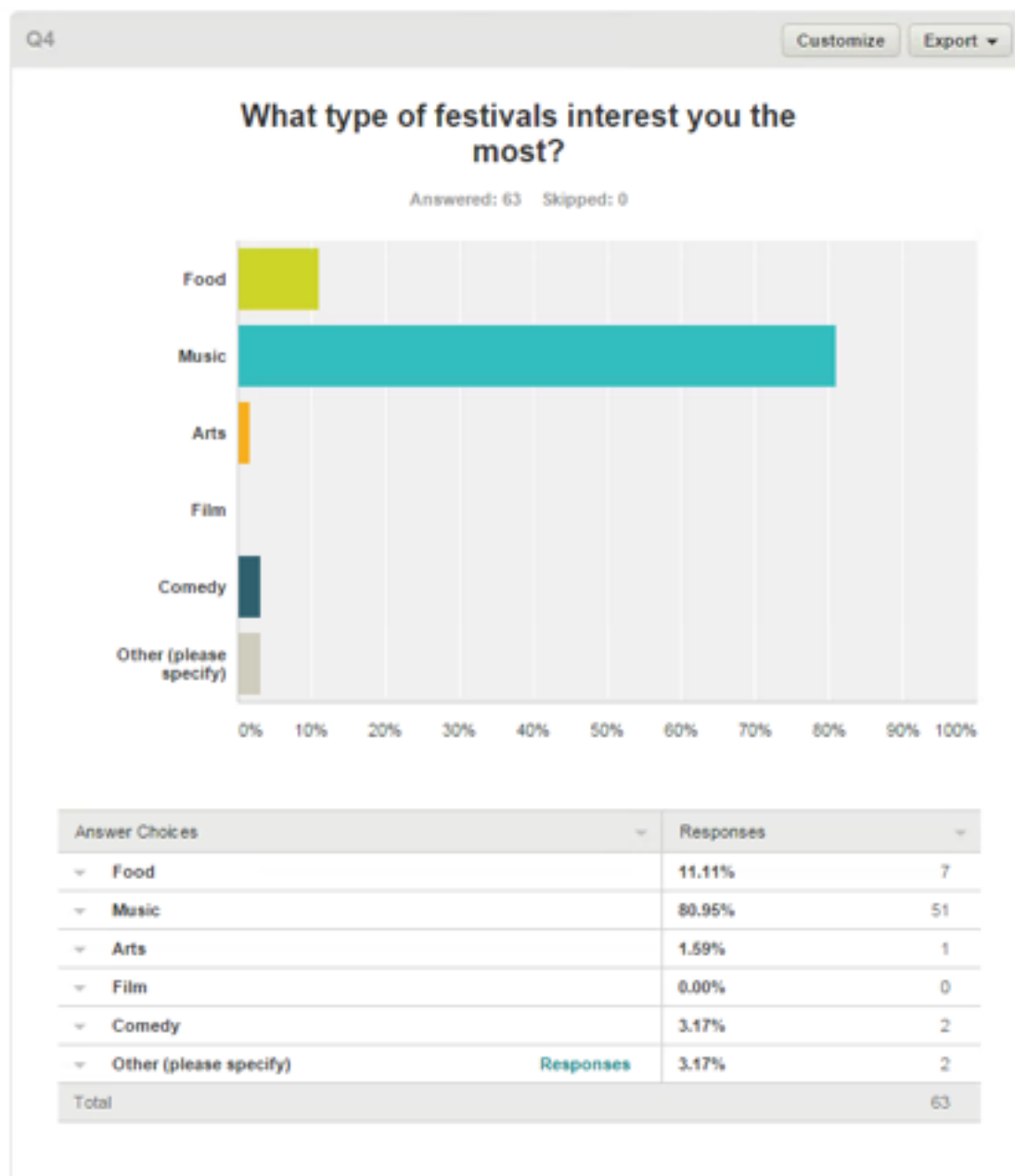


Figure 7.4

## Question 5:

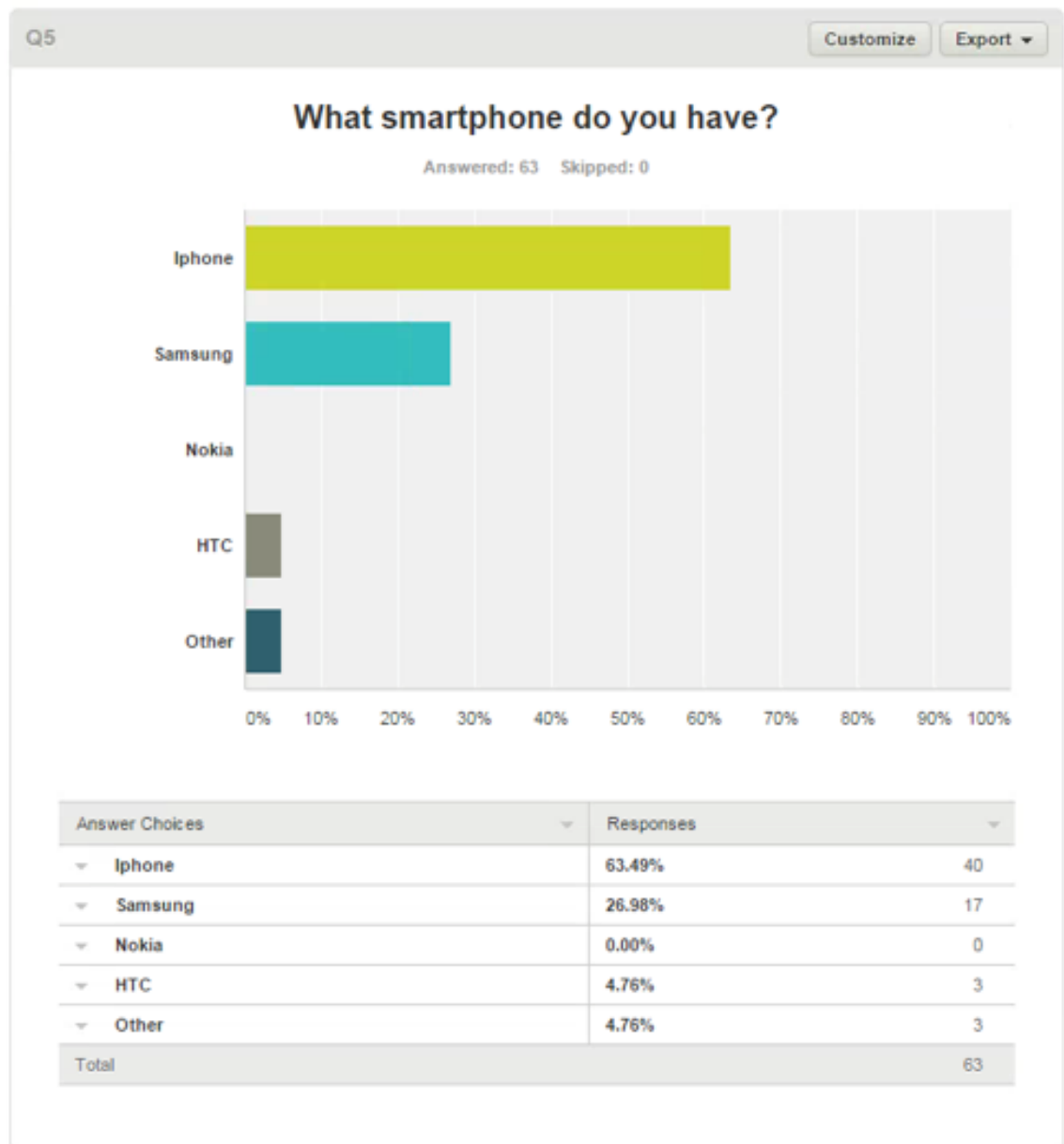


Figure 7.5



## Question 6:

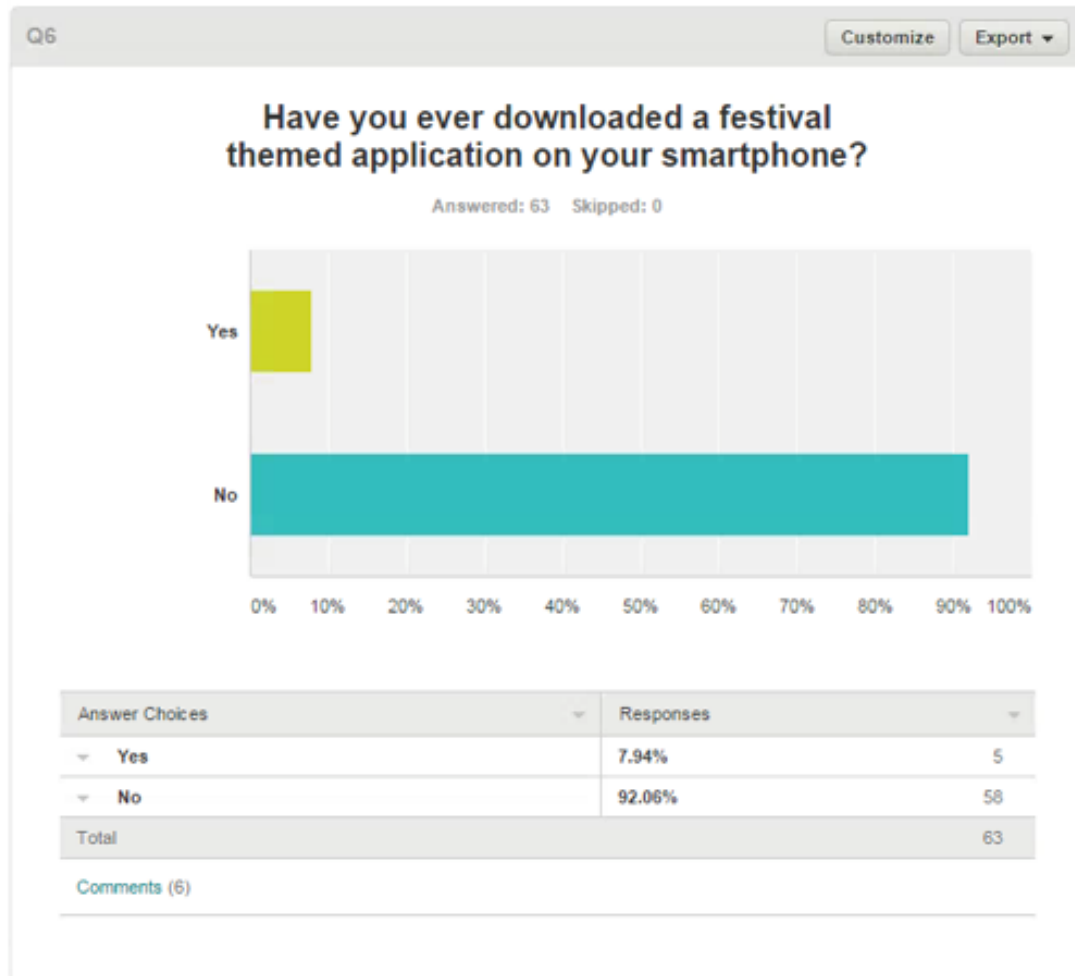


Figure 7.6

## Question 7:

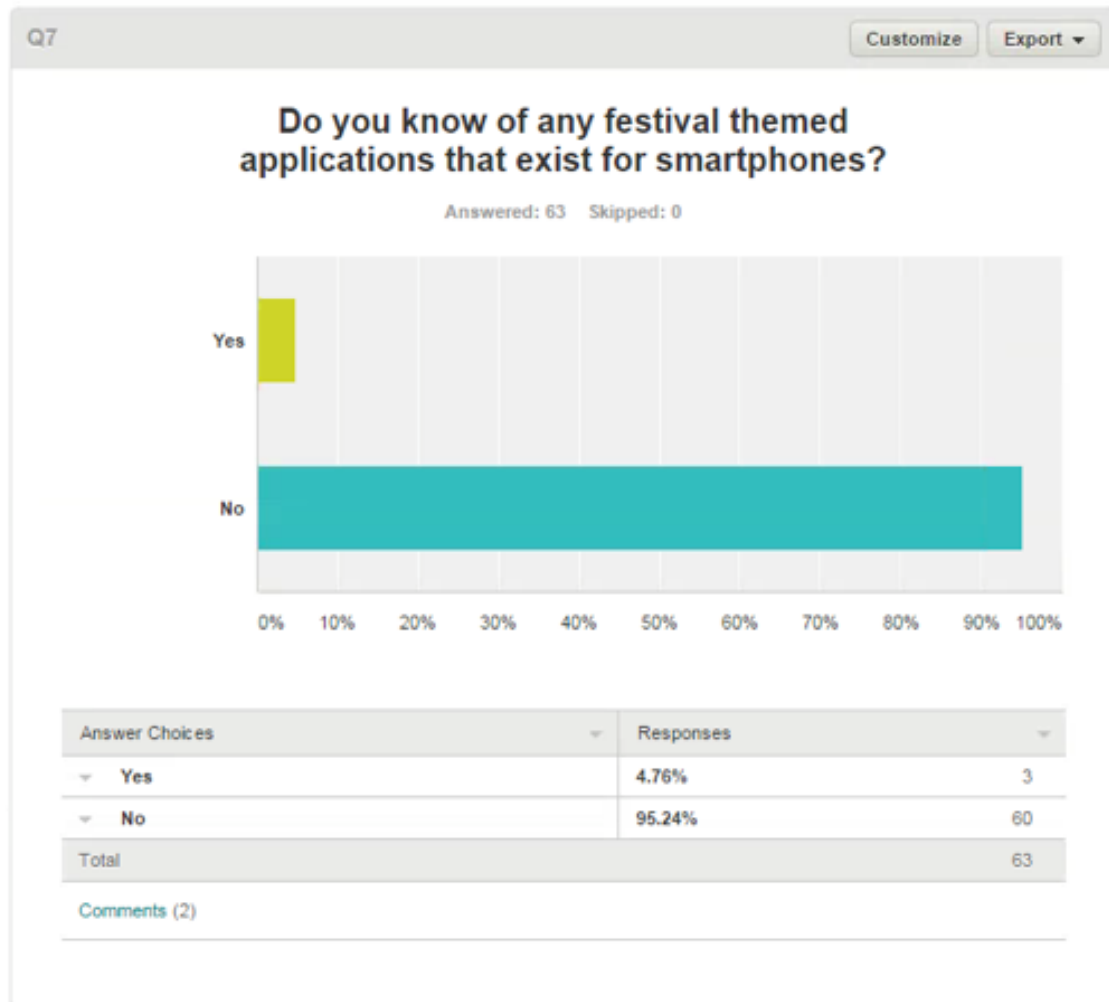


Figure 7.7

## Question 8:

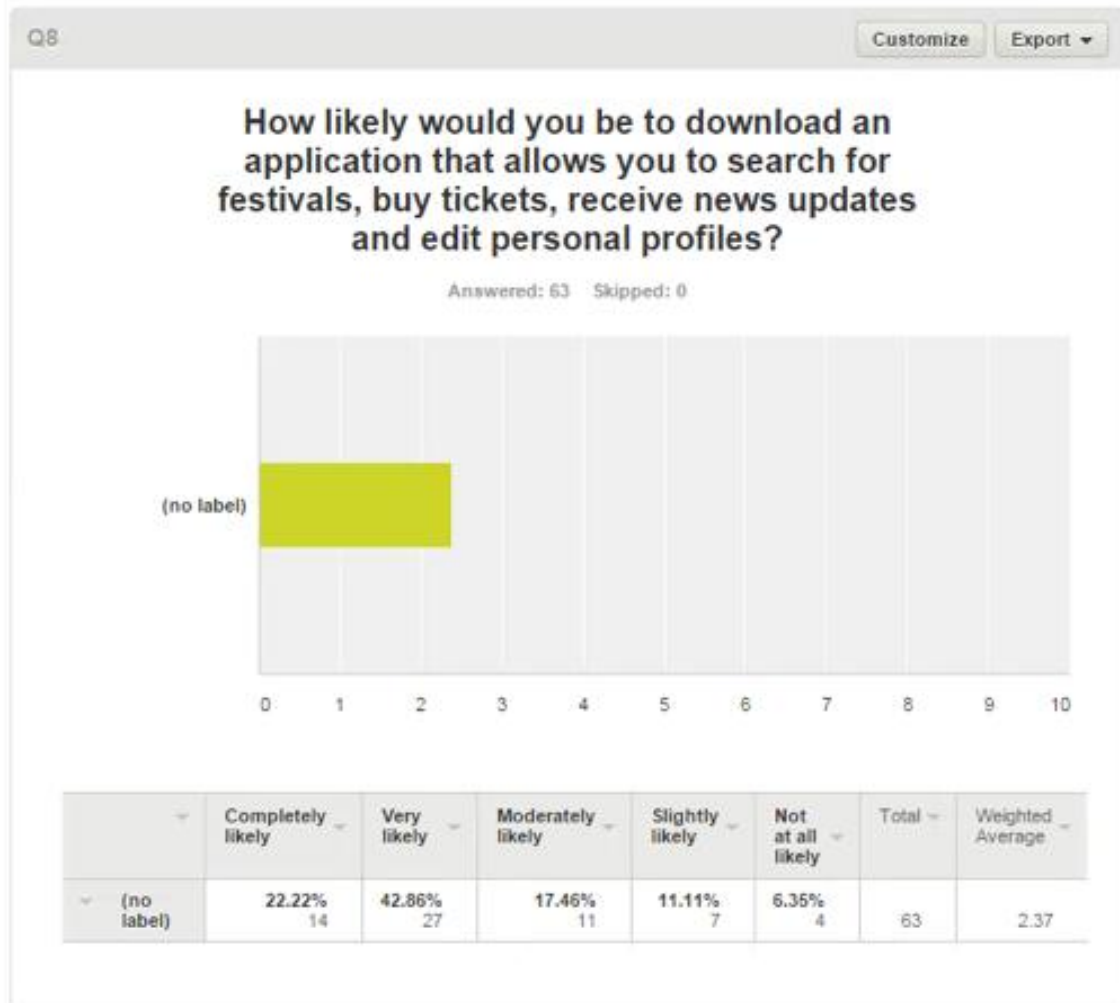
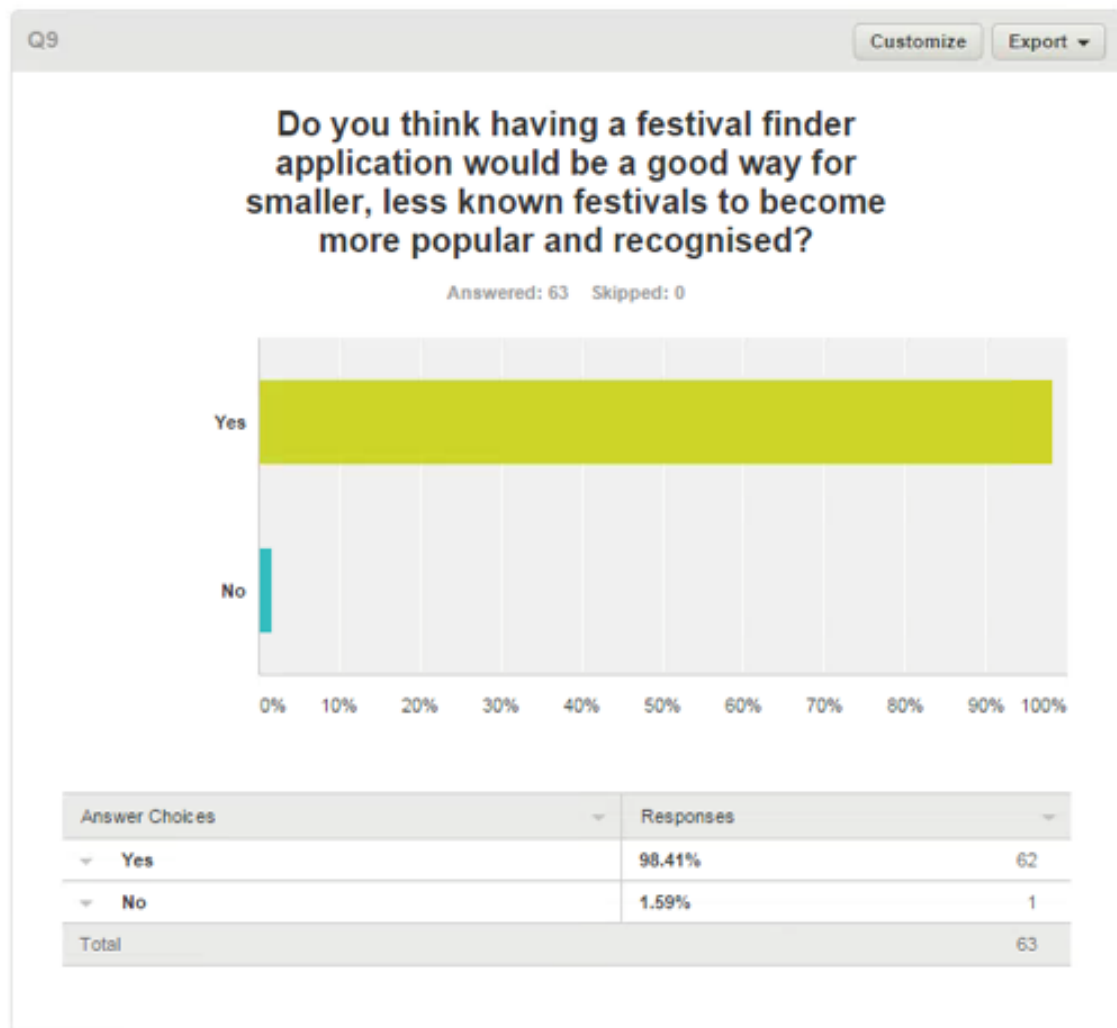


Figure 7.8

## Question 9:

*Figure 7.9*

Question 10:

Q10 Export

**(Optional) What would you like to see in a festival finder application?**

Answered: 19 Skipped: 44

● Responses (19) Text Analysis My Categories

**PRO FEATURE**  
Use text analysis to search and categorize responses; see frequently-used words and phrases. To use Text Analysis, upgrade to a GOLD or PLATINUM plan.  
[Upgrade](#) [Learn more](#)

Categorize as... Filter by Category  Q

Showing 19 responses

- An area where you could swap/buy/sell tickets with other users.  
1/27/2016 5:11 PM [View respondent's answers](#)
- Details of what is on offer, option to buy tickets  
1/27/2016 4:48 PM [View respondent's answers](#)
- Full line up description with accompanying videos to show the genre of music and previous performances would be useful. Best of luck, lisa  
1/27/2016 4:31 PM [View respondent's answers](#)
- Early bird tickets and competitions!  
1/27/2016 4:24 PM [View respondent's answers](#)
- Suggestions on festivals close based on your location Any deals on festivals that are currently on Festivals based on your music taste  
1/27/2016 4:21 PM [View respondent's answers](#)
- Good design, easily navigable.  
1/27/2016 4:14 PM [View respondent's answers](#)
- Time! What food and drinks are served (alcoholic & non-alcoholic)

Figure 7.10

## 7.3 Monthly Journals

### **September Reflective Journal**

#### **My Achievements**

This month I spent most of my time researching for my Software Project. I researched different software that I would use to create my Fest Finder Application. In the end I chose Eclipse as the main IDE. I also refreshed my mind with the different programming languages I will need to use to make my Fest Finder Application work and to look professional. I chose Java and XML as the main programming languages. As my application is an online application connected to a database I also looked over documentation on PHP and SQL so when it comes to connecting to my database I will do so with ease.

My contributions to the project included drawing up each page of the application so when it comes to start coding and designing, I already know how it needs to look. In these drawings I have included the technical features such as Paypal plugin, Google Maps plugin etc.

I also spent some time researching festivals I will include in my Application. I want to include enough festivals spread worldwide to make the App exciting and full of content but I don't want to include too many that will take up valuable coding time.

#### **My Reflection**

I felt, I worked well to design how my application will look when it is finished. I also worked well to make some decisions regarding what plugins I will use and what languages and software I will use to build my Application.

I also collated a number of useful links that will help with various parts of building my project. When it comes the time to add my Paypal plugin, I already have a link to how to do so.

However, I was not successful in time management. I will need to assign more time to my project or else I will be left near deadline with parts left un-built.

#### **Intended Changes**

Next month, I will try to better my time management and assign more time to software project. Even if I am still researching it is better than not doing any at all.

## **October Reflective Journal**

### **My Achievements**

This month, I was assigned Dominic Carr as my supervisor for the project. I met with Dominic and received some really useful tips from him. I found the meetings with Dominic very helpful as I struggled on where to begin with regards to getting my project started and he pointed me in the right direction. I started on my Requirements Specification and drew up Use Case diagrams and mock-up UI's to include in the Specification.

I decided to change my mind and use Android Studio instead of Eclipse. I researched Android Studio and it seemed to be a better option than using Eclipse to develop an Android app. I researched and setup Android Studio by downloading the relevant SDKs.

My contributions to the projects included getting set up with Version Control on Github so that I had something to fall back on should there be any technical errors or loss of data along the way.

### **My Reflection**

I felt, it worked well to manage my time more efficiently this month and allocate more time to my project and to get tasks finished before upcoming deadlines.

However, I was not successful in researching as much as I would have liked to on PHP.

### **Intended Changes**

Next month, I will try to research more into connecting to a database using PHP

I realised that I need to make a start on my project as December and mid-point deadlines are closing in quickly.

### **Supervisor Meetings**

Date of Meeting: 20/10/2015

Items discussed: Version Control, Requirements Specification

Action Items: Dominic discussed Version Control and pointed me in the right direction to maintain this during the Development of my project. He also showed me some sites for mock-ups which I used to design mock-up UI's for my Requirements Specification. He also recommended to break the goal of the project into three sub-goals; Mandatory, Discretionary and Exceptional. This will give my project some structure to scaffold my work.

Date of Meeting: 27/10/2015

Items discussed: Project Proposal

Dominic went through my Project proposal and showed me some things to change. He showed me where I used too much details and parts that weren't detailed enough. This meeting really helped me to improve my documentation and showed me where I need to describe my project in detail and where I need to keep documentation brief.

## **November Reflective Journal**

### **My Achievements**

This month, I was able to do a lot of research on PHP and PHPMyAdmin. This enabled me to get started on my Login/Register system for my FestFinder Application. I found some very useful tutorial videos on YouTube that helped me to get started and to understand some things that I wasn't sure of. I signed up to 000WebHost to create and host my database online with PHPMyAdmin along with my php files to connect to the database. I designed a simple Login/Register UI in Android Studio. In doing so I also learned more about how Android Studio works and how to style and design different pages and also how to connect pages together through buttons.

I also researched how to setup version control in Android Studio so that all of my work will be backed up to Github.

I also completed my first draft of my Analysis and Design Specification which I sent to my supervisor for his feedback before submission.

### **My Reflection**

I felt, I worked well to understand PHP with PHPMyAdmin and how to create and connect to an online database. I also have a better understanding of the functions and layout of Android Studio after playing around with the software and creating my Login system.

However, I was not successful in getting as much work done on my project as I would have liked to due to closing deadlines from various modules and also other project and continuous assessment work.

### **Intended Changes**

Next month, I will try to balance my work better as I have a few deadlines coming to the end of the semester.

I realised that I need to assign specific times to work on my project. If I want to be satisfied with the amount of work I get done then I need to stick to a schedule that allows me to work on my project and complete other projects and continuous assessments.

### **Supervisor Meetings**

Date of Meeting: 23/11/2015

Items discussed: Analysis and Design Specification

Action Items: Dominic opened the template for the Analysis and Design Specification document and went through each heading with me. I couldn't understand some of them and how they would tie in with my project but he helped me to understand that not all of the headings will apply to my project. He asked me to do a draft and to send it to him for review so he could provide some feedback that would help me to fully complete the document.



## **December Reflective Journal**

### **My Achievements**

This month progress on developing my application was slow due to upcoming end of semester deadlines and exam preparation for January. However, I didn't completely forget about my project. I still watched tutorials that gave me some helpful tips that will be useful for when I start developing again. I managed to design other parts of my project by adding activities in Android Studio that will be the layout of my application so I have a skeleton ready to begin development after December.

My contributions to the projects included keeping my mind refreshed on the task ahead by watching tutorials and some minor design input to the project including a logo that I will use for my background and splash screen of the application.

### **My Reflection**

I felt, I worked well to keep my project on my mind during this busy month of deadlines and study.

However, I was not successful I actually getting much development done on my project due to the reasons outlined above.

### **Intended Changes**

Next month, I will try to get back into the swing of things completely and begin development where I left off. I will break down my goals and begin by getting the major tasks completed first leaving the minor design and layout tasks until the end of development.

I realised that I need to prepare for my mid-point presentation in February so I will need to complete documentation for that and have something worth presenting by that stage.

### **My Achievements**

This month I decided to scrap my previous Login/registration system I created before December. The main reason for this was I wasn't happy with the system as a whole. I had a better vision in my head and I decided to start again. I will still use PHP and an online SQL database. This time I created the PHP files on Cloud 9. I will use a landing page for my application that will re-direct users to the Play Store and encourage them to download the app in order to use it. I will use the SLIM framework as it is relatively new and will be good to incorporate new technologies in my app.

My contributions to the projects included getting SLIM and PHP setup on Cloud 9. I also used my previously created database and connected to it on the cloud. As mid-point presentations are nearing I also prepared my Technical report to submit and I started to make my slides and researched presentation tips to help me get the most marks possible.

## **January Reflective Journal**

### **My Reflection**

I felt, it worked well to setup SLIM and PHP on Cloud 9 and get started on developing my new Login/registration system. I also worked well on academically writing my technical report that I will be able to extend and edit a little when it comes to finishing it near project completion. However, I was not successful in making much progress as far as functional development goes. The choice to scrap my old login/registration system has a big role to play in this, but I didn't want to develop something I would later regret.

### **Intended Changes**

Next month, I will try to stay on track and put in a lot of extra hours for college work. I realised that I need to cut out some activities I usually do during the week in my spare time and spend more time doing college work in order to have good quality assignments completed on time.

### **Supervisor Meetings**

Date of Meeting: 02/02/2016

Items discussed: Progress, marking scheme and mid-point presentation.

Action items: Dominic discussed my progress on the application so far and where I was struggling. He reminded me to concentrate on the mandatory requirements of the project and not to worry about features that will be added in the later stages of development if I had time. We went through the mid-point marking scheme and he provided some tips on what to include and what not to include in my presentation.

He also helped answer some questions I had regarding the Technical report.

## **February Reflective Journal**

### **My Achievements**

This month, I was able to get a considerable amount of developing done on my login/registration system. I also started to customise the Google maps search page by implementing the Google maps API in my application and customising markers and locations to include a custom FF icon at the location of festivals. I began loading festivals into my database to pull the coordinates of them into google maps and show them on the map with the custom marker.

### **My Reflection**

I felt, it worked well to divide my time between all of my modules and spend a considerable amount of time each week on each module in order to work on projects and continuous assessments. With my mid point presentation at the start of the month I felt that took away from development a bit but I got working straight after the presentation.

### **Intended Changes**

Next month, I will try to dedicate even more time to my project. This will be hard with other projects and continuous assessments to complete at the same time and exams also coming up. I realised that I need to break up a task/goal to complete each week as I found this helpful in getting tasks done.

### **Supervisor Meetings**

Date of Meeting: 24/02/2016

Items discussed: Mid point Technical Report

Action Items: Dominic walked me through the mid point technical document and showed me where and why I lost marks. He also pointed out areas to change in order to gain marks in the final submission.

I found this very helpful as I wouldn't have spotted some of the errors that he did.

## **March Reflective Journal**

### **My Achievements**

As predicted last month I found it difficult to make and substantial progress on my project this month with lectures coming to an end and trying to complete other projects and CAs. But I still managed to get work on my project finished. My login and registration is now completed with some security elements left to develop like password hashing.

My contributions to the project included finally finishing my authentication system for my project. I have a list of tasks that I will now work on each week to complete. With exams only a week away I think the majority of the work will be completed straight after them when I can fully commit myself just to my project.

### **My Reflection**

I felt, it worked well to finish my authentication system after running into a lot of difficulties. However, I was not successful in making huge progress on my project because of other modules continuous assessments.

### **Intended Changes**

Next month, I will get a lot of time to work on my project and fully dedicate all of my time to developing and finishing my project that meets my brief. I aim to not only finish my project but also be satisfied with the outcome. As this is my first time developing anything of this size and difficulty I am enjoying myself so far and look forward to finishing exams so I can spend all of my time doing it before the deadline!