

National College of Ireland

BSc in Computing

2016/2017

Ronan Lafford

X13121651

Ronan.lafford@student.ncirl.ie

Spontaneous

Final Report



DOCUMENT CONTROL

Revision History

Date	Version	Scope of Activity	Prepared	Reviewed	Approved
10/05/2016	1	Complete	RL		

Distribution List

Name	Title	Version
Dr Catherine Mulwa	Project Supervisor	
Mr Eamon Nolan	Project Co-Ordinator	
Mr Michael Broadford	Second Supervisor	

Related Documents

Title	Comments
Project Requirements	
Project Proposal	
Monthly Journals	

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name:	Ronan Lafford
Student ID:	X13121651
Supervisor:	Dr. Catherine Mulwa

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

Document Control	1
1 Executive Summary.....	5
1.1 Background	6
1.2 Aims	6
1.3 Technologies	7
1.4 Purpose.....	12
1.5 Project Scope	12
1.6 Definitions, Acronyms, and Abbreviations	13
1.7 User Requirements Definition - Requirements Specification	13
2 Requirement Specifications for Spontaneous Mobile Application.....	14
2.1 Functional requirements.....	14
2.2 Data Requirements	35
2.3 Non-Functional Requirements	39
3 Implementation of Spontaneous Mobile App.....	41
3.1 Architectural Design.....	41
3.2 Graphical User Interface	42
3.2.1 Application Programming Interfaces (API)	46
3.3 System Evolution.....	46
4 Testing and Evaluation of Spontaneous Mobile App.....	47
4.1 Unit Testing.....	48
4.2 Usability Testing.....	50
5 Conclusions	55
6 Further Development or Research.....	55
7 References	57
8 Appendix.....	58
8.1 Appendix 1: Project Proposal	58
8.2 Appendix 2: Evaluation	66
8.3 Appendix 3: Monthly Journals.....	67

1 EXECUTIVE SUMMARY

Spontaneous is an Android application that is being developed to help people discover and attend events in their vicinity. These events are created by the users of the app and can cover a wide range of activities including, but not limited to, hobbies, education, volunteering and community events.

There are many places where events can be found but they are usually in their own specific sites or forums. The app aims to solve the issue of a lack of a resource that can allow a user to create events based on their interest and display it on a map for every user in the area to view and join if interested. The name Spontaneous reflects the intended usage of the app, to enable a user to open the app, choose an event and go, at any time and in any place.

The report highlights the main functionality that users would require and desire in an application of this nature. It was found that a user wants to be able to open the app and view what is happening at that moment and what is upcoming, so having an always available and real time app is crucial. Safety is essential to encourage users to participate, it was necessary to enable reviews and ratings for each event and allow the upload of media to give a better indication of the event creator, what the event will be like and if it would be worth attending.

Almost everyone has a smartphone and as the majority run Android it was felt this technology would be the best to reach the widest possible range of users. The report covers the tools and technologies used to develop the User Interface and the web services to connect the app to the cloud storage backend.

1.1 BACKGROUND

As a husband, father of two young children, working full time and studying part-time I am fully aware of the small amount of free time available for leisure. I try to make the most of any time I can spare by enjoying it with my family and friends. A common scenario is that there are many times when you may just not know what to do, cannot think of anywhere to go or probably don't have enough money to do certain activities.

The motivation for Spontaneous is to alleviate this frustration by being a gateway to the community and presenting the user with a host of activities that may be currently underway or planned for a later time. Examples of such activities could be a parent creating a parent – toddler meet up where other users could view it and join the event with their children. A hobbyist may enjoy coding and create a Coding event inviting other enthusiasts to join in working on a project. A user may wish to create a litter picking activity and other users could volunteer to help if interested.

I have not found a solution to these types of situations readily available or catered for all in one place and I believe that Spontaneous can satisfy this need and will help develop friendships, build community ties and increase social inclusion by bringing people of all ages together while having fun.

1.2 AIMS

The purpose of this report is give an overview of the development and construction of an Android Application that I will be creating. The name I have chosen for this application is "Spontaneous".

Spontaneous will be a social networking/event app with the core functionality of allowing users to register, view, create and partake in events that may take their interest, at any time. The name Spontaneous reflects the spur of the moment nature of the app. Users will be able to see all "happenings" in their area and beyond at that moment.

With the intention of the app focussing on real time, the user will be able to navigate quickly through the clean well defined screens within a small number of clicks. The aim is to try have all functionality reached with 2 clicks for a better user experience. I will try have an interface that will be as minimalist and intuitive as possible to allow for better performance and ease

of use. The activity and map data showing the events should be available and up to date when accessed. The app needs to be fast to maintain interest and keep with the live theme.

User feedback will be used to rate activities and the ability to upload and view photos to show previous gatherings can encourage future participation. With the collection of user's age, I hope to have a filtering system to restrict users from seeing certain events e.g. an underage person seeing an event at a pub.

As a social app I plan to incorporate the ability to call and message other members including sharing of events. With all events, there is the possibility of too many people turning up, so being able to turn off the event notification when a specified amount of attendees confirm is needed.

If a user attends events regularly they may wish to show others what they were like, therefore, a list of recently attended events should be available to the user. In order to keep the app interesting and engaging. I will not bombard the user with too many features initially. The core functionality will be expanded incrementally with "must-have" features and then "would-like" features depending on time available.

I intend to publish the application on the Google Play Store when complete.

1.3 TECHNOLOGIES

A brief description of the technology used to develop the application:

1.3.1.1 *Android Studio – Release 2.2.2*

Android Studio is the free official integrated development environment (IDE) for developing Android apps. It is based on the excellent JetBrains IDE, IntelliJ, and was released by Google in 2013.

The app will be developed using Android Studio using both the XML and new Visual Design editor that has been built in. It will be synced with the GitHub repository to update and keep track of the application source code.

1.3.1.2 Android – API Level 25+

Android is an operating system owned by Google that is aimed at mobile and touch screen devices. It is almost 9 years old and the latest iteration is Nougat, each major release is named after a dessert in alphabetical order.

I will be using the latest API as this will be compatible with all the newer devices as well as the older releases back to API level 15.

1.3.1.3 SQLite

SQLite is a zero configuration self-contained SQL database. It is built into android devices and allows storage of data to a text file. It is used for structured data such as contact lists etc. It will be used to store user details after they log in to enable them to remain logged in until they sign out.

1.3.1.4 PHP 5

PHP (Hypertext Pre-processor) is a widely used as a server side scripting language that is suited towards web development and can be embedded in HTML. PHP is used to access functionality and resources through Uniform Resource Identifiers (URI).

It allows for the client device to pass messages to the server and receive the response. The app will be using PHP to connect to the server to request data from the database for display in the app.

1.3.1.5 phpMyAdmin

phpMyAdmin is a free and open source tool written in PHP intended to handle the administration of MySQL and other databases with the use of a web browser. It allows the administrator to create, modify or delete databases, tables, fields and rows. It allows for execution of SQL statements and the ability to manage users and permissions.

phpMyAdmin will be located on the server and used to create and maintain the app database and the tables and data within it.

1.3.1.6 JSON

JSON (JavaScript Object Notation) is a human readable format for structuring data, it is also easy for machines to parse and generate. It is used to transmit data between a server and web application, as an alternative to XML.

The application uses the JSON as the data format that is transferred between the application and web services as it is easy to use and works well with the different languages used.

1.3.1.7 MySQL

MySQL is an open source relational database management system owned by Oracle.

MySQL will be used to contain all the user, event and location data used within the application and will be located on a secure remote host.

1.3.1.8 Amazon EC2

Amazon Elastic Cloud Compute (EC2) is an online service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

This is where phpMyAdmin will be located and will store the logic for connecting the web services to the database and retrieving the data to present to the application.

1.3.1.9 Amazon RDS

Amazon Relational Database Service (RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud.

Amazon RDS is where the MySQL database will be hosted containing all user, event and location data related to the application'

1.3.1.10 Google Maps API

Google Maps API is a Web-based service that provides detailed information about geographical regions and sites around the world. Providing the ability to access resources such as street mapping, aerial photography, satellite imagery and also a street view.

The application will be utilising the mapping service in order to drop pins on a map as indicators for event locations and providing directions from the user location to the event address.

1.3.1.11 Google Fused Location Provider API

This is an API that utilizes the technology in your device to find your last known location. It does this by triangulating data from the GPS, Wi-Fi and Data signals to determine the user location to an accuracy of `ACCESS_FINE_LOCATION` or `ACCESS_COARSE_LOCATION` depending on the power management demanded by the application.

1.3.1.12 Google Geocoder

This is a class that allows the geocoding and reverse geocoding of a location. This means that if the co-ordinates are obtained then they can be used to get the address or if the address is obtained it can be used to get the co-ordinates.

This will be used to search the user address and drop a pin on the map according to the obtained Latitude and Longitude co-ordinates.

1.3.1.13 Google Places API

Google Places API is a Web-based service that provides detailed information about local business and other such places around the user location. It helps make the application location aware.

The application will be utilising the Place Picker service in order to find a business etc. on a map and retrieve the location and other details of the business such as telephone number or URL.

1.3.1.14 Picasso 2.5.2

This is a library that can handle the caching, resizing and displaying of images retrieved from the server.

This will be used to populate the RecyclerView card images with the user images stored on the EC2 server.

1.3.1.15 Volley 1.0.0

Volley is a library developed by Google that makes networking easier and faster for Android apps. This is the library used to make the POST and GET requests that will be used to send and retrieve data from the database.

1.3.1.16 Google TalkBack

TalkBack is an accessibility service that helps blind and vision-impaired users interact with their devices. TalkBack adds spoken, audible, and vibration feedback to the device. TalkBack comes pre-installed on most Android devices.

The app aims to be as inclusive as possible and by integrating accessibility it can open up the community to those who may have missed out otherwise.

1.3.1.17 Sinch Messaging API

Sinch is a real time, in-app instant messaging service that allow for better connection between users. Sinch offers Instant Message SDKs that you can integrate with your smartphone to easily add instant messaging based features.

Being a social app it is essential for the users planning on attending an event to communicate with each other or with the creator. Using instant messaging attendees can be kept up to date with any changes in event, venue or any other details as required.

1.3.1.18 JUnit

JUnit is an open source unit testing framework for the Java programming language.

The app will be tested using the JUnit framework. As Android is based on the java language this framework will be able to test the application for any bugs that may occur and will be essential as the application expands and the number of lines of code grows larger.

1.3.1.19 Espresso test recorder

The Espresso Test Recorder allows for the creation of UI tests without writing any code. The tester can record a test scenario, which records the interactions with a device and add assertions to verify UI elements in the app. Espresso Test Recorder then takes the saved recording and automatically generates a corresponding UI test that can be run to test the app.

Espresso will be used to ensure the app UI is functioning correctly and the expected results are what is expected.

1.3.1.20 GitHub

GitHub is a web-based Git repository hosting service. It offers all of the distributed version control and source code management of Git as well as adding its own features.

The app source code will be located on GitHub. This will be essential for developing the app as every time new code is committed to the source it remembers the previous state of the code. If any errors occur the version can be rolled back to the previous state before it was introduced.

1.3.1.21 Circle CI

CircleCI is a hosted continuous integration service that takes care of automated testing, building and deployment of our applications. It can be connected to any project that is hosted on GitHub.

1.4 PURPOSE

- The objective of writing this document is to set out the requirements needed to develop an Android Application and define a design solution based on the criteria outlined in this report. This document contains:
 - The scope of the project for delivery;
 - Analysis of the problems / drivers that lead to the project;
 - A list of features that the system should deliver;
 - Diagrams and Mock Screens to aid understanding.

The content of the document is derived from the input obtained from prospective users and the developer's solution design to achieve the desired outcome.

1.5 PROJECT SCOPE

The scope of the project is to develop an Android Application where the subject matter is social event creation. This Application will allow users to sign up and view all member created events within the immediate vicinity of the device location or in an area of their choosing.

All events can be viewed and accessed through the map or list view. Events can be filtered by location, date, age rating and type. The user will be able to, join, review, rate and even create their own event should they wish to. If a user creates an event, the event will be added to the map, list view and Upcoming events screen for other members to see. The user will be presented with directions to the event if they are attending.

The user will be able to see all Closed, Open and Upcoming events and have a history of events attended. Events can be bookmarked and shared by a user. The user will have the ability to send and receive messages with other members. A user will be able to upload video and photos to the event for others to see.

Data will be stored in a remote MySQL database located on Amazon Web Services (AWS) and will be accessed through PHP web services for CRUD functionality. The format of the data exchanged will be JSON. Google Maps API will be integrated for mapping of User and Events. Accessibility features will allow Users interact in various ways to assist them. An instant messaging API shall be added for Users to contact each other. The User details will be stored in the device SQLite database so they can remain active in the App and not have to re Sign-In unless it is closed out fully.

1.6 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

API	Application Programming Interface
AWS	Amazon Web Services
CRUD	Create, Read, Update, Delete
JSON	JavaScript Object Notation
MySQL	Open Source Database
PHP	Open Source general-purpose scripting language

1.7 USER REQUIREMENTS DEFINITION - REQUIREMENTS SPECIFICATION

The User should be able to download the App from the Google Play Store. Upon launch the User shall be able to Register and Sign-In with the 'Spontaneous' community from within the Application.

When the User signs in they are brought to the main activity screen and the User shall have their current location detected by their device automatically. The User shall be able to create their own event should they wish. They can access the Create New Event from the slide in menu and enter all the required details.

If the User wants to browse all activities, they shall be able to view all events in their area or a specific area of their choosing by typing in the location in a search box. The User shall be able to join an event that they are interested in. They can simply view the details of the event and click the Join button.

After an event the User shall be able to review the event that they attended. The User can view a list of events that they have recently taken part in and add a review by pressing the Review button in the event detail.

Sometimes a User may need more information therefore the User would like to be able to exchange messages with the event creator or other attendees. If the User enjoyed the event, they may have taken some photos or other media. The User would like to upload this media (photos, video) from event to the event detail for others to view. The User shall also be able to view media related to events that were uploaded by other attendees of events.

2 REQUIREMENT SPECIFICATIONS FOR SPONTANEOUS MOBILE APPLICATION

2.1 FUNCTIONAL REQUIREMENTS

2.1.1.1 Use Case Diagram



2.1.1.2 Requirement Table

Use Case	Use Case Description	Importance	Details
UC001	User Registration	Critical	view
UC002	Find Location	Critical	view
UC003	Create Event	Critical	view
UC004	View Event	Critical	view
UC005	Join Event	Critical	view
UC006	Review Event	Important	view
UC007	Share Event	Important	view
UC008	Exchange Messages	Important	view
UC009	Upload Media	Useful	view
UC010	View Media	Useful	view

2.1.1.3 Requirement Priorities are defined as follows:

Critical: Without this capability, the project will be considered a failure - it will not have met its primary objective. This capability must be delivered within this project.

Important: This capability is Important to the customer, however without it the project can still be considered a success based on its primary objective.

Useful: This capability would be nice to have and will be implemented depending on time available.

2.1.1.4 UC001 – User Registration

2.1.1.4.1 User Story & Description

2.1.1.4.2 User story:

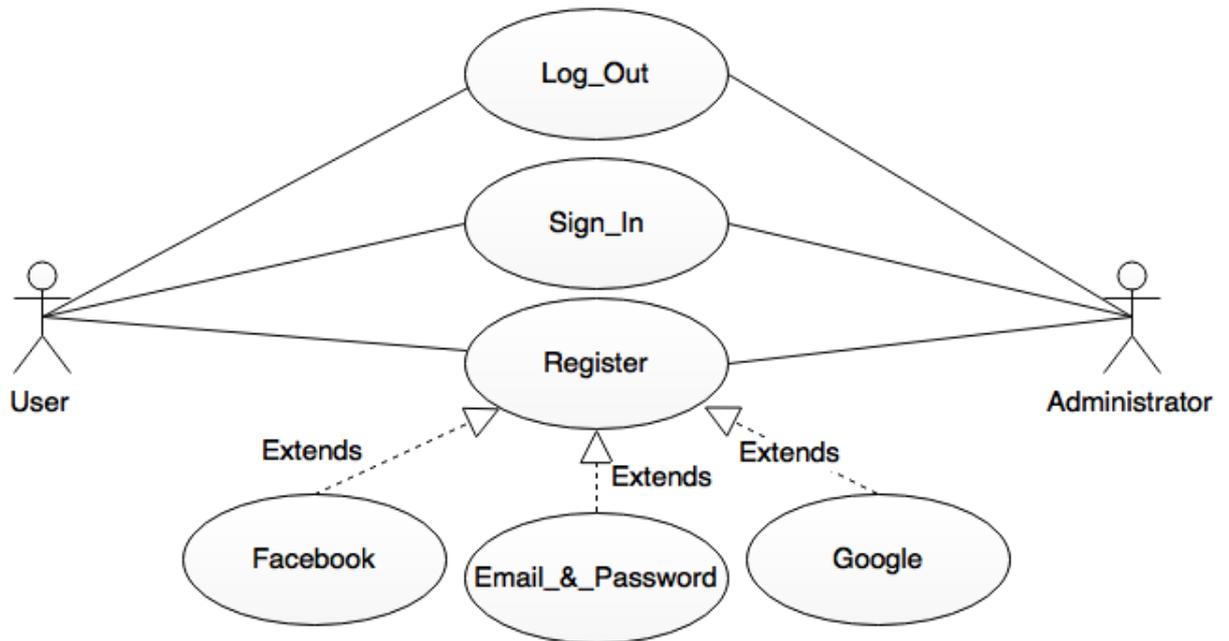
As a User I want to be able to register with the ‘Spontaneous’ community from within the Application.

2.1.1.4.3 Description:

The User can register within the App by pressing the Register Button and then entering their details and saving.

2.1.1.4.4 Use Case

UC001 - Priority: Critical



2.1.1.4.5 Scope

The scope of this use case is to Register a user in the system.

2.1.1.4.6 Description

This use case describes the first interaction that the User has with the Application and how the User becomes a member.

2.1.1.4.7 Flow Description

2.1.1.4.8 Precondition

The App has been launched, is active and has a data connection.

2.1.1.4.9 Activation

This use case starts when a User presses the register button.

2.1.1.4.10 Main flow

1. The User presses the Register Button
2. The App activates the Registration Activity
3. The User enters required fields
4. The User saves their details to complete registration

2.1.1.4.11 Alternate flow

N/a

2.1.1.4.12 Exceptional flow

E1: <Invalid Details>

1. The User enters invalid details
2. The App issues a warning
3. The use case continues at position 2 of the main flow

E2: < Duplicate Details>

1. The User enters duplicate details
2. The App issues a warning
3. The use case continues at position 2 of the main flow

E3: < Cancel Registration>

1. The User cancels registration
2. The User is returned to the main activity

2.1.1.4.13 Termination

The system saves the User data and returns to the main activity.

2.1.1.4.14 Post condition

The App goes into a wait state

2.1.1.5 UC002 – Use Location

2.1.1.5.1 User Story & Description

2.1.1.5.2 *User story:*

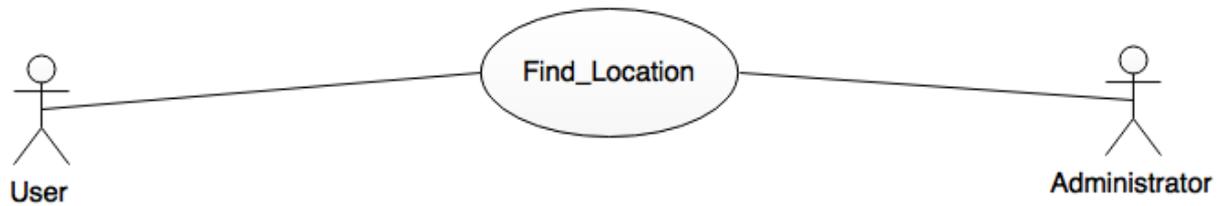
As a Spontaneous user, I must be able to retrieve my current location to display local events.

2.1.1.5.3 *Description:*

When the User launches the App. The device will retrieve the current location. If the user is already within the App they can press the location button to centre their location.

2.1.1.5.4 Use Case

UC002 - Priority: Critical



2.1.1.5.5 Scope

Retrieve the current location of the User to display events around that location.

2.1.1.5.6 Description

This use case starts when the Application launches or when the get location button is pressed.

2.1.1.5.7 FLOW DESCRIPTION

2.1.1.5.8 Precondition

The Application has been launched and the User is on the main activity.

2.1.1.5.9 Activation

This use case starts when a User launches the App or when the get location button is pressed on the main activity.

2.1.1.5.10 Main flow

1. The User launches the App
2. The App initiates a request to find location
3. The App centres the devices position on the map and displays the events around that location.
4. The User can then view the events on the map or on the ListView below it.

2.1.1.5.11 Alternate flow

1. The User launches the App
2. The User presses the location button on the main activity screen.
3. The App initiates a request to find location
4. The App centres the devices position on the map and displays the events around that location.
5. The User can then view the events on the map or on the ListView below it.

2.1.1.5.12 Exceptional flow

E1: <Location not found>

1. The User launches the App
2. The App initiates a request to find location
3. The App cannot find the location due to no connection.
4. The App issues a "Location not found error".

2.1.1.5.13 Termination

The App stops searching when the location has been found successfully or if no connection has been detected.

2.1.1.5.14 Post condition

The system goes into a wait state until it is due to search the location again for an update or the User presses the location button again.

2.1.1.6 UC003 – Create Event

2.1.1.6.1 User Story & Description

User story:

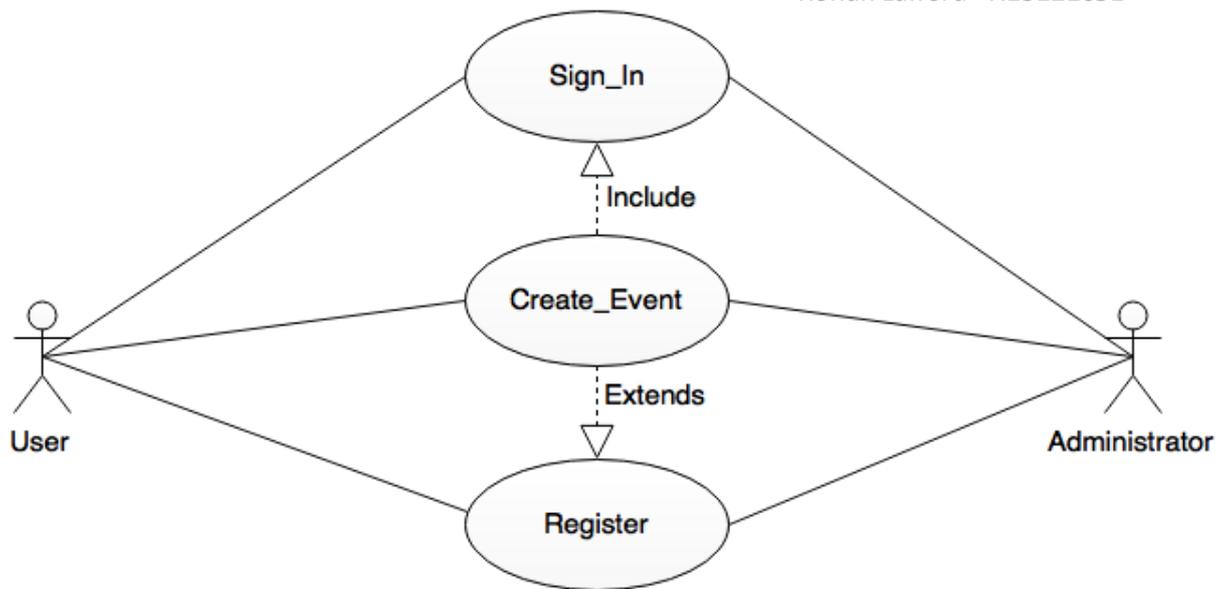
As a User, I want to be able to create my own event and display it on the map.

Description:

A User shall be able to click on Create New Event. This will allow them to enter the event details and save them. This will add the event to the map on a marker pin and as an item in a ListView.

2.1.1.6.2 Use Case

UC003 - Priority: Critical



2.1.1.6.3 Scope

The scope of this use case is to allow for the User to enter details into the Create Event form and have it display on the map for other members to see.

2.1.1.6.4 Description

This use case describes how the User moves to the Create Event screen adds all relevant details about the event such as date, time, location, duration etc.

2.1.1.6.5 Flow Description

2.1.1.6.6 Precondition

The User is logged in and is on the Create Event activity.

2.1.1.6.7 Activation

This use case starts when the App has been launched and the User has signed in and opened the slide-in menu to access the Create Event link.

2.1.1.6.8 Main flow

1. The User signs in to the App
2. The User opens the slide-in menu and presses the Create Event
3. The App displays the Create Event activity
4. The User enters in the required details and saves it.
5. The details are now displayed on the map and ListView.

2.1.1.6.9 Alternate flow

N/a

2.1.1.6.10 Exceptional flow

E1: <Incomplete Details>

1. The User does not enter all fields
2. The App will not allow the event to be created unless all fields are complete
3. The use case continues at position 3 of the main flow until all fields are entered

2.1.1.6.11 Termination

The App stays on the create screen until the details are saved or if the User backs out of the screen.

2.1.1.6.12 Post condition

If completed the App posts the data to the database and then it displays it in the map and ListView.

2.1.1.7 UC004 – View Event

2.1.1.7.1 User Story & Description

User Story:

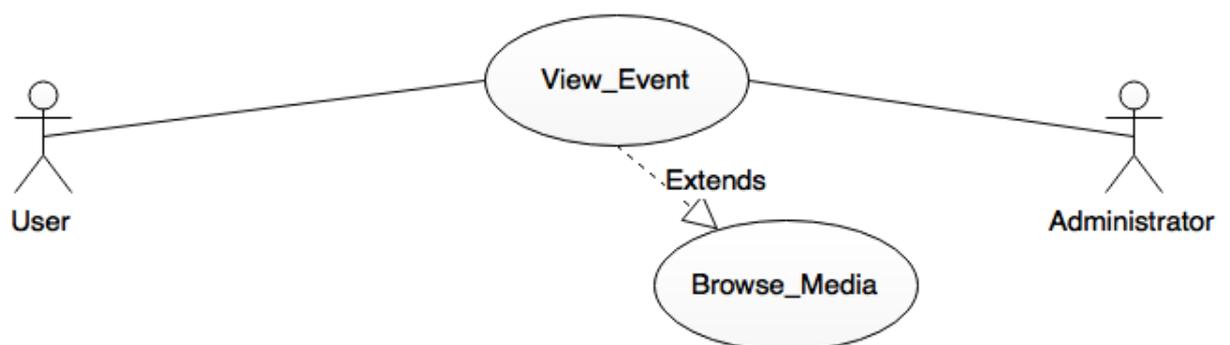
As a User, I want to be able to select an event in my area either through clicking on a map pin or ListView element to view the details.

Description:

A User shall have the ability to browse all events posted to the App. They shall be able to access the event details by either clicking on the event marker on the map or by clicking on the event in the ListView.

2.1.1.7.2 Use Case

UC004 - Priority: Critical Scope



2.1.1.7.3 Scope

The scope of this case is to activate the event details activity displaying all the relevant details needed to attend.

2.1.1.7.4 Description

This use case describes how the User can access the details of any event of interest. The User selects the event by pin or list item and a new View Event activity is displayed with all the details required including uploaded media and buttons to join or share.

2.1.1.7.5 Flow Description

2.1.1.7.6 Precondition

The User is logged in, on the main activity screen and all events are showing for that area.

2.1.1.7.7 Activation

This use case starts when the User selects an event by either pressing the list item or the map pin.

2.1.1.7.8 Main flow

1. The ListView item is pressed by the User.
2. The App launches the View Event activity.
3. The App displays all the details for the event including media and buttons to share or join the event.
4. The User can decide if they want to join/share or not.

2.1.1.7.9 Alternate flow

A1: <Map Pin flow>

1. The User presses the map pin.
2. The App launches the View Event activity.
3. The App displays all the details for the event including media and buttons to share or join the event.
4. The User can decide if they want to join/share or not.

2.1.1.7.10 Exceptional flow

E1: <Event not found>

1. The User presses the map pin or list item.
2. The App cannot access the event details

3. The App will display a warning message if the event details cannot be found due to no connection or no events in that area.

2.1.1.7.11 Termination

If the User has finished viewing the details they can leave the View Event screen by backing out or pressing the join button which will then bring them back to the main activity screen.

2.1.1.7.12 Post condition

The App is in a wait state on the main activity screen.

2.1.1.8 UC005 – Join Event

2.1.1.8.1 User Story & Description

User Story:

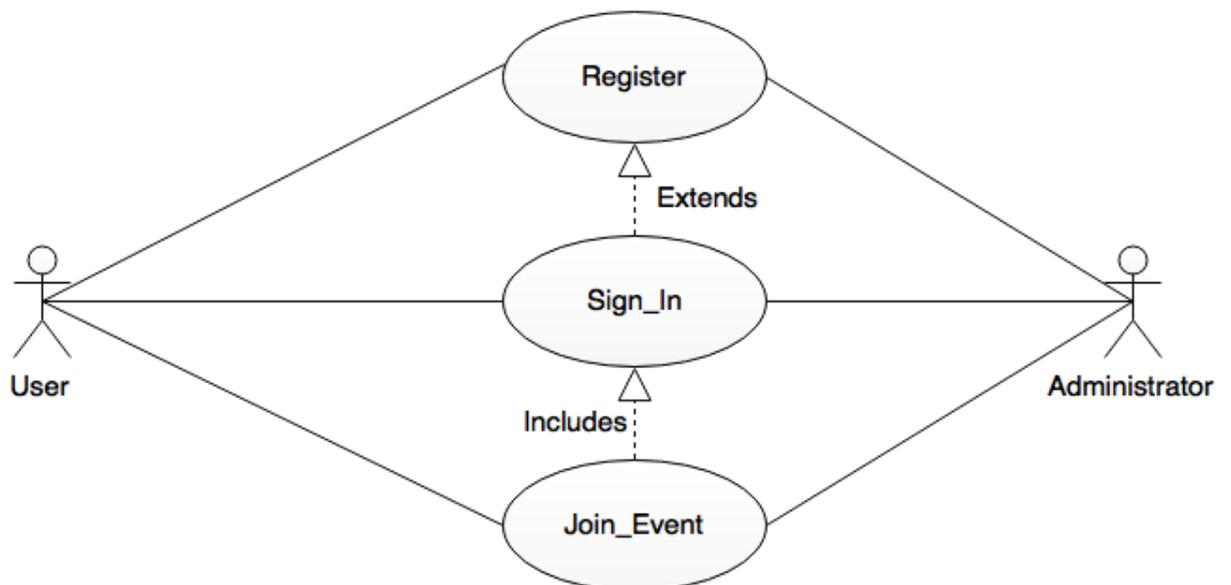
As a User, I want to be able to join an event that interests me.

Description:

When a User browses the events that are planned, they can view the details of the event. The details activity will present a button for the User to press if they should wish to attend the event.

2.1.1.8.2 Use Case

UC005 - Priority: Critical Scope



2.1.1.8.3 Scope

The scope of this use case is to allow a User to join an event by pressing the Join button on the View Event screen.

2.1.1.8.4 Description

This use case describes the situation when a User is viewing the Event details screen and decides to join the event. The User presses the button which in turn will add the event to the Users planned events and the event creator can see another User has joined the event.

2.1.1.8.5 Flow Description

2.1.1.8.6 Precondition

The User is logged in and is on the View Event details screen.

2.1.1.8.7 Activation

This use case starts when the User presses the Join button.

2.1.1.8.8 Main flow

1. The User presses the Join Button on the Event Details screen.
2. The App adds the event to the Users planned events.
3. The App adds the User to a list of Users attending the event.
4. The User receives a toast confirming if successfully joined.

2.1.1.8.9 Alternate flow

N/a

2.1.1.8.10 Exceptional flow

E1: <Cannot Join Event>

1. The User presses the Join Button on the Event Details screen.
2. The App will not allow the User to Join the event as it cannot connect to the database.
3. The App displays a connection error warning of a connection error.

2.1.1.8.11 Termination

The App confirms the success of the Join and returns to the main activity screen.

The App displays connection error and the User can try again or back out of the screen.

2.1.1.8.12 Post condition

The App goes into a wait state back on the main activity screen if the join is complete or the User backs out. It will wait for the User to retry joining if there is a connection error.

2.1.1.9 UC006 - Review Event

2.1.1.9.1 User Story & Description

User Story:

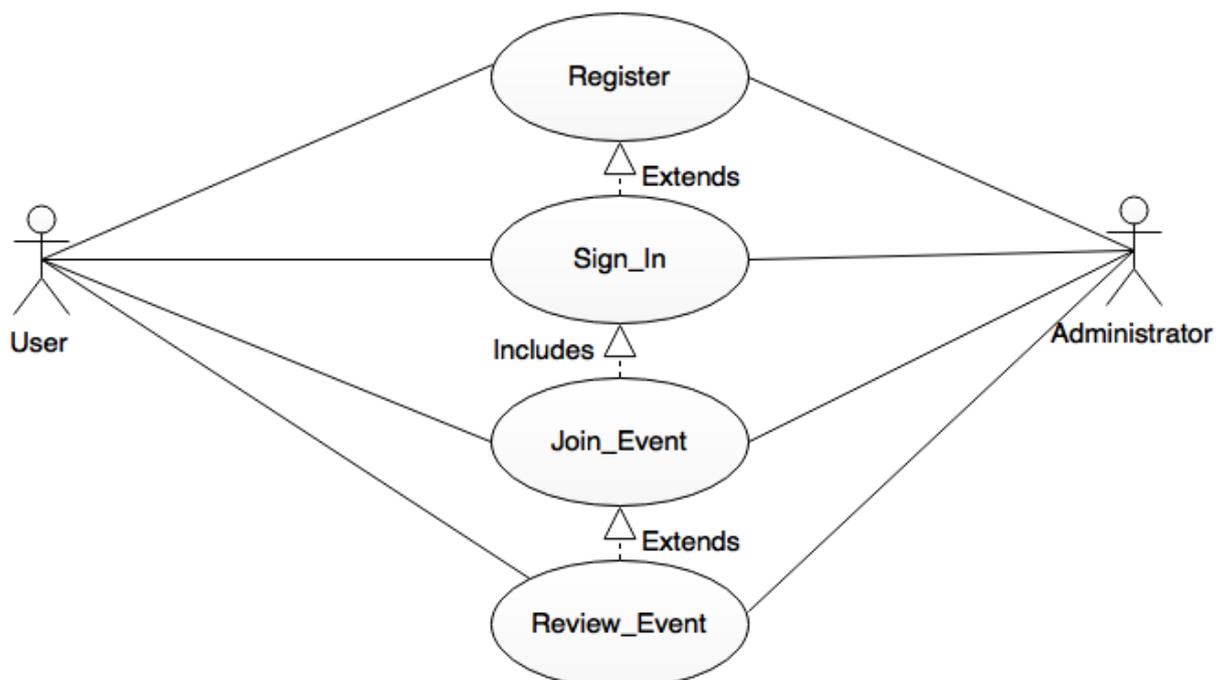
As a User, I want to be able to give a review of any event that I attend to inform others of my opinion of the event.

Description:

To promote regular events and build confidence in a Creator of an event, the App shall provide the User who has attended an event to add their review and rating. This review and rating will assist others in deciding if it is worth attending or not.

2.1.1.9.2 Use Case

UC006 - Priority: Critical Scope



2.1.1.9.3 Scope

The scope of this use case is to allow the User who has attended an event to be able to post a review of the event and give it a star rating.

2.1.1.9.4 Description

This use case describes how the User is given the functionality to add their opinion about an event and give it a rating for others to see. When an event has closed, the Users who had joined are allowed post a review by pressing the Review button on the Event Details screen.

2.1.1.9.5 Flow Description

2.1.1.9.6 Precondition

The User has attended an event and the event has finished. The User is logged in and returns to the Event Details Screen.

2.1.1.9.7 Activation

This use case starts when the User enters the Event Details activity after an event has closed and pressed the Review button.

2.1.1.9.8 Main flow

1. The User presses the Review button.
2. The App presents a dialog that will allow the User to enter some text into a text area and give a star rating.
3. The User adds their review and star rating and presses the save button.
4. The App adds the review to a list of reviews associated with that closed event.

2.1.1.9.9 Alternate flow

N/a

2.1.1.9.10 Exceptional flow

E1: <Cannot Save Review>

1. The User presses the Save button to add the review.
2. The App cannot connect to the database.
3. The App cannot save the review and issues a message to warn of a connection error.

2.1.1.9.11 Termination

The App returns to the main activity screen if the review has been added successfully.

The App will remain on the Review activity if it cannot connect and will remain there until the User either tries again to save or backs out of the screen.

2.1.1.9.12 Post condition

The App goes into a wait state on the main activity screen or on the Review activity if the review was not saved.

2.1.1.10 UC007 - Share Event

2.1.1.10.1 User Story & Description

User Story:

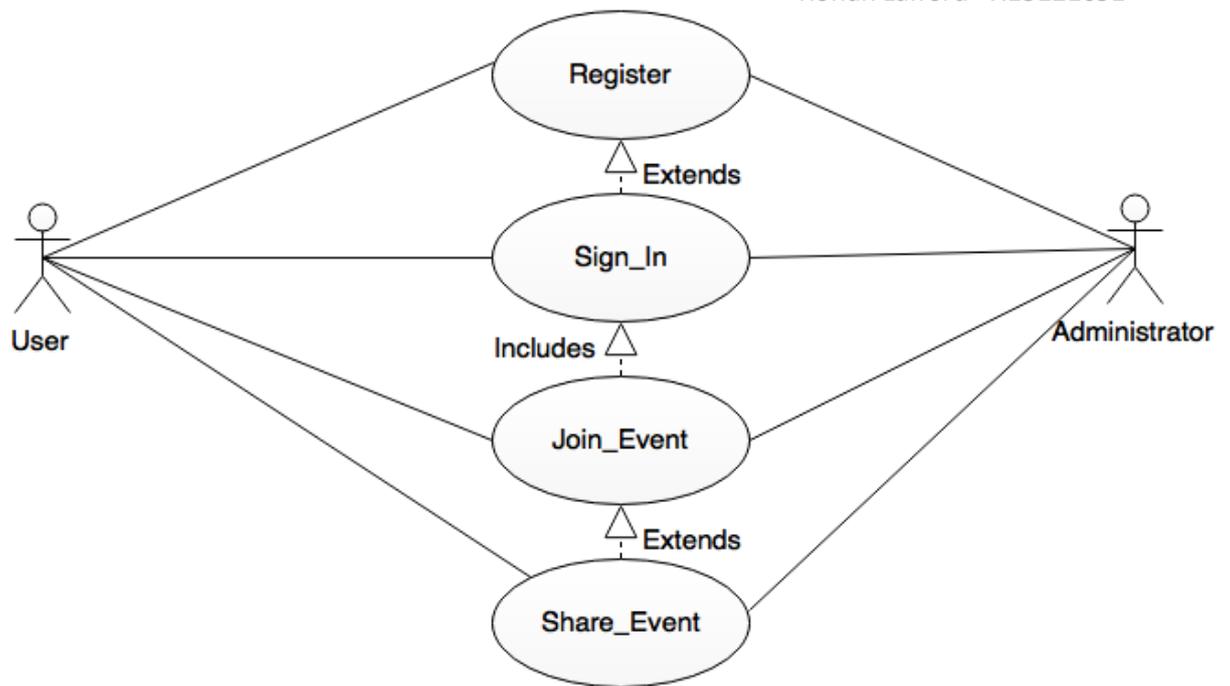
As a User, I want to be able to share an event with my friends if I feel it may be of interest to them.

Description:

A User of the App may be friends with other registered Users. Should they find an event that they would like to attend and would like to perhaps invite their friend then the App will provide the functionality for the User to Share the event by pressing the Share button on the Event details screen.

2.1.1.10.2 Use Case

UC007 - Priority: Important



2.1.1.10.3 Scope

The scope of this use case is to allow the User to share an event with other Users or friends.

2.1.1.10.4 Description

This use case describes the situation where a User is on the View Event details screen and would like to share it with a friend or any other User. The User can press the Share button and the other person will receive the event details for viewing.

2.1.1.10.5 Flow Description

2.1.1.11 Precondition

The User is logged in and is on the View Event details screen.

2.1.1.11.1 Activation

This use case starts when the User presses the Share button.

2.1.1.11.2 Main flow

1. The User presses the Share button.
2. The App launches a dialog window with options for sharing the event.
3. The User selects the required option.
4. The App sends the details to the recipient and they are notified of the event.

2.1.1.11.3 Alternate flow

N/a

2.1.1.11.4 Exceptional flow

E1: <Error Sharing>

1. The User presses the Share button.
2. The App may have no internet connection or data signal.
3. The App displays a cannot share error.

2.1.1.11.5 Termination

The App will display a toast if the event is shared successfully.

The App will display a warning message if the share cannot be completed.

2.1.1.11.6 Post condition

The App will go into a wait state until the User shares it again or backs out of the screen.

2.1.1.12 UC008 - Exchange Messages

2.1.1.12.1 User Story & Description

User Story:

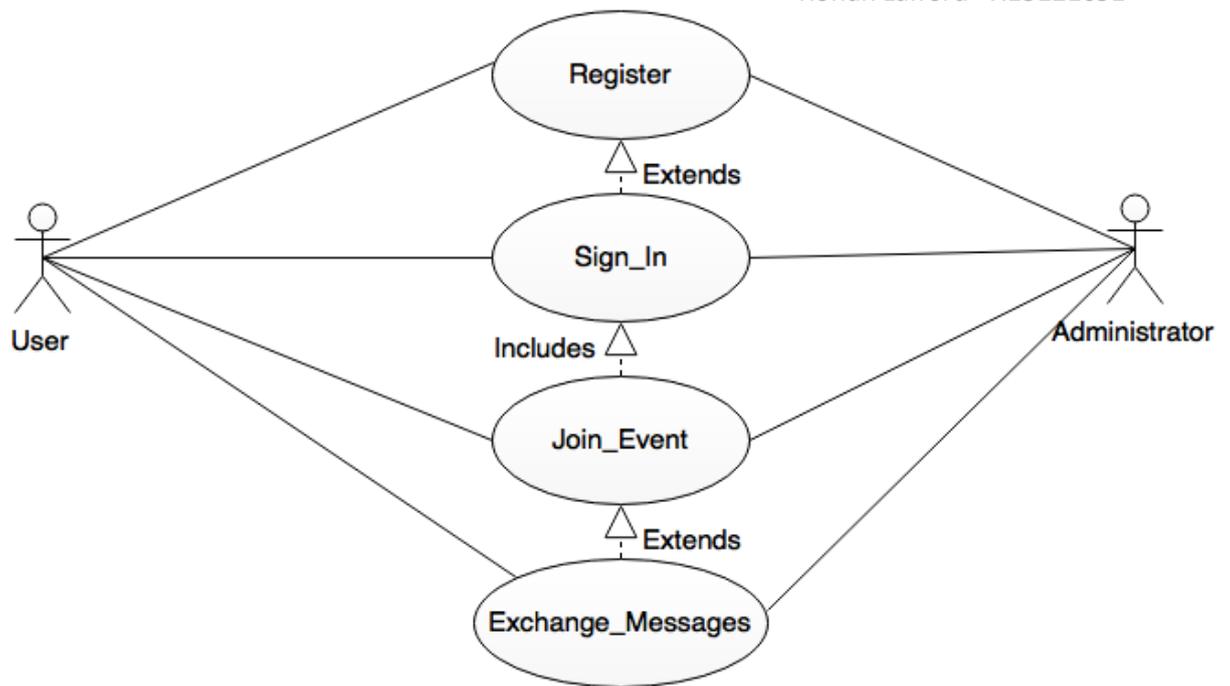
As a User, I want to be able to exchange messages with the creator of an event or other members.

Description:

The User would like a way to be able to obtain more information from the Creator of the event or from other attendees. The App should have the functionality to enable the User to exchange messages with others from within the App.

2.1.1.12.2 Use Case

UC008 - Priority: Important



2.1.1.12.3 Scope

The scope of this use case is to allow a User to contact the creator of an event to get further information or to contact other members who are attending the event.

2.1.1.12.4 Description

This use case describes the situation where a member may need further information about an event before attending. The User should have the ability to contact the creator if needed. The User should also have the ability to contact another member who is attending in the event they could be travelling together to the event or for other reasons. The User should be able to click on the Creator/Member in the attendees list which will display a text area including a send button for the User to send the message.

2.1.1.12.5 Flow Description

2.1.1.12.6 Precondition

The User is logged in and has joined an event and is on the Event Details screen.

2.1.1.12.7 Activation

This use case starts when the User selects a member from the attending list. This will expand the member details and will include a button to send message. The User can click on this button to initiate the sending of a message.

2.1.1.12.8 Main flow

1. The User is on the Event Details screen
2. The User presses the creator/member in the list of attendees and their details expand and displays a send message button.
3. The User presses the send message button and a text area with a send button is displayed for entering the message text.
4. The User completes the message and presses the send button.
5. If message was successfully sent, then a toast will display confirming the message was sent.

2.1.1.12.9 Alternate flow

N/a

2.1.1.12.10 Exceptional flow

E1: <Cannot send message>

1. The User presses the send message button and a text area with a send button is displayed for entering the message text.
2. The App cannot send the message
3. The App will display a toast warning of a possible connection error and the message could not be sent.

2.1.1.12.11 Termination

If the message was sent successfully then the App will remain on the Event Details screen as the User may wish to send another message.

If the message was not sent the App will keep the text area still displaying in the event the User wants to try re-sending the message.

2.1.1.12.12 Post condition

The App will go into a wait state

2.1.1.13 UC009 - Upload Media

2.1.1.13.1 User Story & Description

User Story:

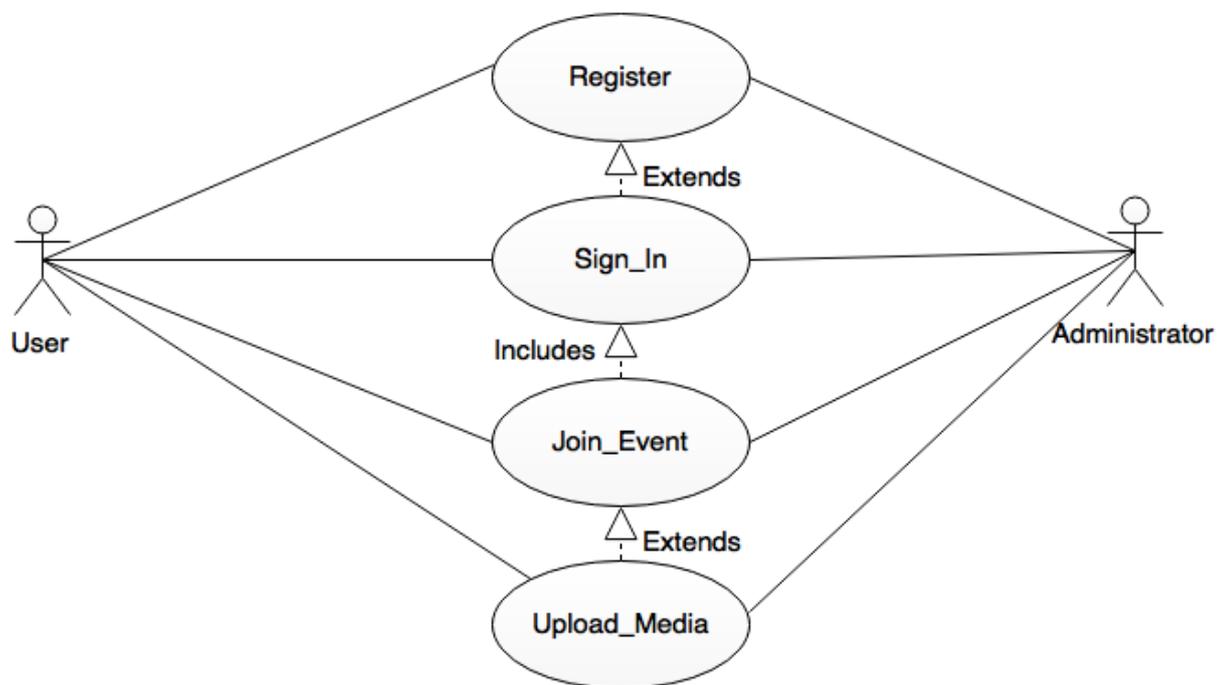
As a User, I have attended a recent event and took some great photos. I want to be able to upload these photos for everyone to see how great the event was.

Description:

The User who has attended an event would like the ability to upload any media taken at the event for others to see. The creator of an event should be able to add media to the details of the event if required to help promote the event.

2.1.1.13.2 Use Case

UC009 - Priority: Useful



2.1.1.13.3 Scope

The scope of this use case is to allow a User who has attended an event to login to the App and upload media relating to the event they attended for others to see.

2.1.1.13.4 Description

This use case describes how the User can log into the App and select the event from their recently attended list. They shall have the option to upload their media for that event.

2.1.1.13.5 Flow Description

2.1.1.13.6 Precondition

The User is logged in to the App and has clicked on one of their attended events to display the Event Details which includes a button to upload media.

2.1.1.13.7 Activation

This use case starts when the User clicks on the upload button on the Event Details screen.

2.1.1.13.8 Main flow

1. The User presses the Upload button on the Event Details screen.
2. The App launches the photo gallery activity.
3. The User selects which media to upload.
4. The App adds the media to the Event Details for display.

2.1.1.13.9 Alternate flow

N/a

2.1.1.13.10 Exceptional flow

E1: <Error Uploading>

1. The User selects which media to upload.
2. The App is unable to upload the media.
3. The App issues a warning for a possible connection error when the media fails to upload.

2.1.1.13.11 Termination

The App updates the Event details gallery with the new media available to view.

2.1.1.13.12 Post condition

The App goes into a wait state for the User's next action.

2.1.1.14 UC010 - View Media

2.1.1.14.1 User Story & Description

User Story:

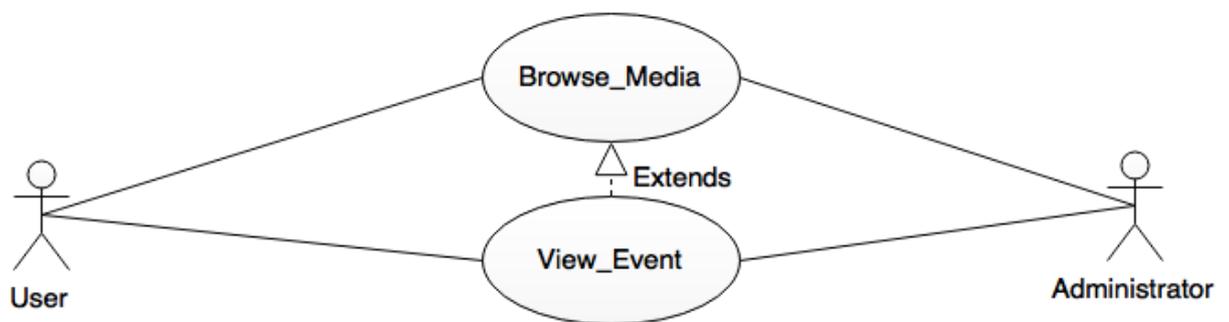
A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

Description:

The User should be able to view any media uploaded to all events. The User should be able to enter the Event details screen and see a gallery of all the media linked to that particular event.

2.1.1.14.2 Use Case

UC010 - Priority: Useful



2.1.1.14.3 Scope

The scope of this use case is to allow the User of the App to view all media associated with all the events posted on the App.

2.1.1.14.4 Description

This use case describes how the User can have access to view the media that is linked to any event that has been uploaded to the App by either the creator of the event or other event attendees.

2.1.1.14.5 Flow Description

2.1.1.14.6 Precondition

The User has launched the app and has entered the Event details screen of any event they choose.

2.1.1.14.7 Activation

This use case starts when a User clicks on an event to launch the Event Details screen.

2.1.1.14.8 Main flow

1. The User clicks on an event.
2. The App launches the Event details screen.

3. The User should be able to view the media that is being displayed at the top of the Event details screen.

2.1.1.14.9 Alternate flow

N/a

2.1.1.14.10 Exceptional flow

E1: <Cannot View Media>

1. The User clicks on an event.
2. The App launches the Event details screen.
3. The App fails to load the media into the Event details screen.
4. The App issues a warning message that the media could not be loaded.

2.1.1.14.11 Termination

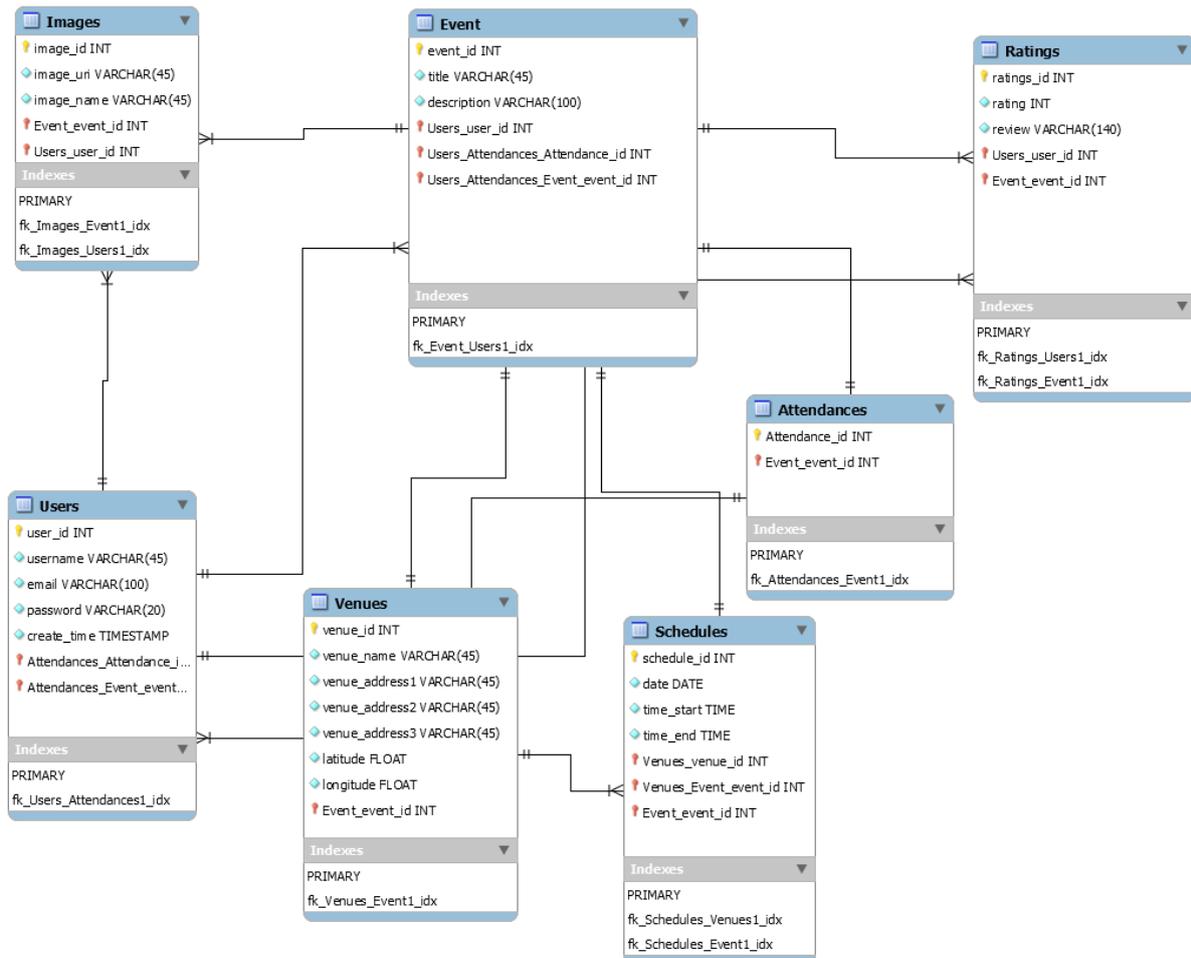
The User is on the Event details screen with the media at the top of the screen ready for the User to view.

2.1.1.14.12 Post condition

The App goes into a wait state waiting for the User's next action.

2.2 DATA REQUIREMENTS

As the application is required to store information about various events and users and media it was essential to try and build a database that was efficient and concise. For this to happen the objects that were being used within the app had to be identified and then broken down into their simplest forms where the data in each table was normalised and wholly dependent on the Primary Key. The aim was to use constraints to enforce referential integrity between the tables linked by foreign keys and use a junction table between any many to many relationships.



```
-- Schema Event
```

```
CREATE SCHEMA IF NOT EXISTS `Event` ;
USE `Event` ;
```

```
-- Table `Event`.`Attendances`
```

```
CREATE TABLE IF NOT EXISTS `Event`.`Attendances` (
  `Attendance_id` INT NOT NULL AUTO_INCREMENT,
  `Event_event_id` INT NOT NULL,
  PRIMARY KEY (`Attendance_id`, `Event_event_id`),
  INDEX `fk_Attendances_Event1_idx` (`Event_event_id` ASC),
  CONSTRAINT `fk_Attendances_Event1`
    FOREIGN KEY (`Event_event_id`)
    REFERENCES `Event`.`Event` (`event_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

-- Table `Event`.`Users`

```
CREATE TABLE IF NOT EXISTS `Event`.`Users` (  
  `user_id` INT NOT NULL AUTO_INCREMENT,  
  `username` VARCHAR(45) NOT NULL,  
  `email` VARCHAR(100) NOT NULL,  
  `password` VARCHAR(20) NOT NULL,  
  `create_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `Attendances_Attendance_id` INT NOT NULL,  
  `Attendances_Event_event_id` INT NOT NULL,  
  PRIMARY KEY (`user_id`, `Attendances_Attendance_id`,  
  `Attendances_Event_event_id`),  
  INDEX `fk_Users_Attendances1_idx` (`Attendances_Attendance_id`  
ASC, `Attendances_Event_event_id` ASC),  
  CONSTRAINT `fk_Users_Attendances1`  
  FOREIGN KEY (`Attendances_Attendance_id` ,  
  `Attendances_Event_event_id` )  
  REFERENCES `Event`.`Attendances` (`Attendance_id` ,  
  `Event_event_id` )  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `Event`.`Event`

```
CREATE TABLE IF NOT EXISTS `Event`.`Event` (  
  `event_id` INT NOT NULL,  
  `title` VARCHAR(45) NOT NULL,  
  `description` VARCHAR(100) NOT NULL,  
  `Users_user_id` INT NOT NULL,  
  `Users_Attendances_Attendance_id` INT NOT NULL,  
  `Users_Attendances_Event_event_id` INT NOT NULL,  
  PRIMARY KEY (`event_id`, `Users_user_id`,  
  `Users_Attendances_Attendance_id`,  
  `Users_Attendances_Event_event_id`),  
  INDEX `fk_Event_Users1_idx` (`Users_user_id` ASC,  
  `Users_Attendances_Attendance_id` ASC,  
  `Users_Attendances_Event_event_id` ASC),  
  CONSTRAINT `fk_Event_Users1`  
  FOREIGN KEY (`Users_user_id` , `Users_Attendances_Attendance_id`  
  , `Users_Attendances_Event_event_id` )  
  REFERENCES `Event`.`Users` (`user_id` ,  
  `Attendances_Attendance_id` , `Attendances_Event_event_id` )  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `Event`.`Images`
```

```
-----  
CREATE TABLE IF NOT EXISTS `Event`.`Images` (  
  `image_id` INT NOT NULL AUTO_INCREMENT,  
  `image_uri` VARCHAR(45) NOT NULL,  
  `image_name` VARCHAR(45) NOT NULL,  
  `Event_event_id` INT NOT NULL,  
  `Users_user_id` INT NOT NULL,  
  PRIMARY KEY (`image_id`, `Event_event_id`, `Users_user_id`),  
  INDEX `fk_Images_Event1_idx` (`Event_event_id` ASC),  
  INDEX `fk_Images_Users1_idx` (`Users_user_id` ASC),  
  CONSTRAINT `fk_Images_Event1`  
    FOREIGN KEY (`Event_event_id`)  
    REFERENCES `Event`.`Event` (`event_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Images_Users1`  
    FOREIGN KEY (`Users_user_id`)  
    REFERENCES `Event`.`Users` (`user_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `Event`.`Ratings`
```

```
-----  
CREATE TABLE IF NOT EXISTS `Event`.`Ratings` (  
  `ratings_id` INT NOT NULL AUTO_INCREMENT,  
  `rating` INT NOT NULL,  
  `review` VARCHAR(140) NOT NULL,  
  `Users_user_id` INT NOT NULL,  
  `Event_event_id` INT NOT NULL,  
  PRIMARY KEY (`ratings_id`, `Users_user_id`, `Event_event_id`),  
  INDEX `fk_Ratings_Users1_idx` (`Users_user_id` ASC),  
  INDEX `fk_Ratings_Event1_idx` (`Event_event_id` ASC),  
  CONSTRAINT `fk_Ratings_Users1`  
    FOREIGN KEY (`Users_user_id`)  
    REFERENCES `Event`.`Users` (`user_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Ratings_Event1`  
    FOREIGN KEY (`Event_event_id`)  
    REFERENCES `Event`.`Event` (`event_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `Event`.`Venues`
```

```
-----  
CREATE TABLE IF NOT EXISTS `Event`.`Venues` (  
  `venue_id` INT NOT NULL AUTO_INCREMENT,  
  `venue_name` VARCHAR(45) NOT NULL,
```

```
`venue_address1` VARCHAR(45) NOT NULL,  
`venue_address2` VARCHAR(45) NOT NULL,  
`venue_address3` VARCHAR(45) NOT NULL,  
`latitude` FLOAT NOT NULL,  
`longitude` FLOAT NOT NULL,  
`Event_event_id` INT NOT NULL,  
PRIMARY KEY (`venue_id`, `Event_event_id`),  
INDEX `fk_Venues_Event1_idx` (`Event_event_id` ASC),  
CONSTRAINT `fk_Venues_Event1`  
FOREIGN KEY (`Event_event_id`)  
REFERENCES `Event`.`Event` (`event_id`)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
-----  
-- Table `Event`.`Schedules`  
-----  
CREATE TABLE IF NOT EXISTS `Event`.`Schedules` (  
  `schedule_id` INT NOT NULL AUTO_INCREMENT,  
  `date` DATE NOT NULL,  
  `time_start` TIME NOT NULL,  
  `time_end` TIME NOT NULL,  
  `Venues_venue_id` INT NOT NULL,  
  `Venues_Event_event_id` INT NOT NULL,  
  `Event_event_id` INT NOT NULL,  
  PRIMARY KEY (`schedule_id`, `Venues_venue_id`,  
  `Venues_Event_event_id`, `Event_event_id`),  
  INDEX `fk_Schedules_Venues1_idx` (`Venues_venue_id` ASC,  
  `Venues_Event_event_id` ASC),  
  INDEX `fk_Schedules_Event1_idx` (`Event_event_id` ASC),  
  CONSTRAINT `fk_Schedules_Venues1`  
  FOREIGN KEY (`Venues_venue_id`, `Venues_Event_event_id`)  
  REFERENCES `Event`.`Venues` (`venue_id`, `Event_event_id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Schedules_Event1`  
  FOREIGN KEY (`Event_event_id`)  
  REFERENCES `Event`.`Event` (`event_id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

2.3 NON-FUNCTIONAL REQUIREMENTS

2.3.1.1 Design

Design is an important aspect to the App, as a good design allows for better performance and User satisfaction. The screens should be well planned and laid out with an aesthetically pleasing and consistent theme.

2.3.1.2 Performance/Response time requirement

It is important to keep the App running quickly and prevent and App Not Responding errors for a good User experience. The App will be using AsyncTask which is a way to create new threads to process actions such as request data and wait for a response from the server. The new thread is separate from the UI thread so it keeps the UI responsive while performing the actions in the background.

2.3.1.3 Availability requirement

The App is required to be available to Users at all times and by using AWS it will be highly available.

2.3.1.4 Recover requirement

Recovery is important to be able to rectify database in case of failure and as part of the AWS service recovery is automatic.

2.3.1.5 Security requirement

The App requires Users to register before accessing all its functionality. The User data used to register such as the Username and Password is used to verify valid Users. The User data is securely stored and encrypted on AWS servers.

2.3.1.6 Reliability requirement

The App should be able to connect to the data stored on AWS through PHP web services at all times.

2.3.1.7 Maintainability requirement

The App should be designed in a way that is modular allowing for easy maintenance and any changes to one part should not affect the whole system.

2.3.1.8 Portability requirement

The App should be available to the User at all times wherever they have access to a data signal/Wi-Fi connection.

2.3.1.9 Extendibility requirement

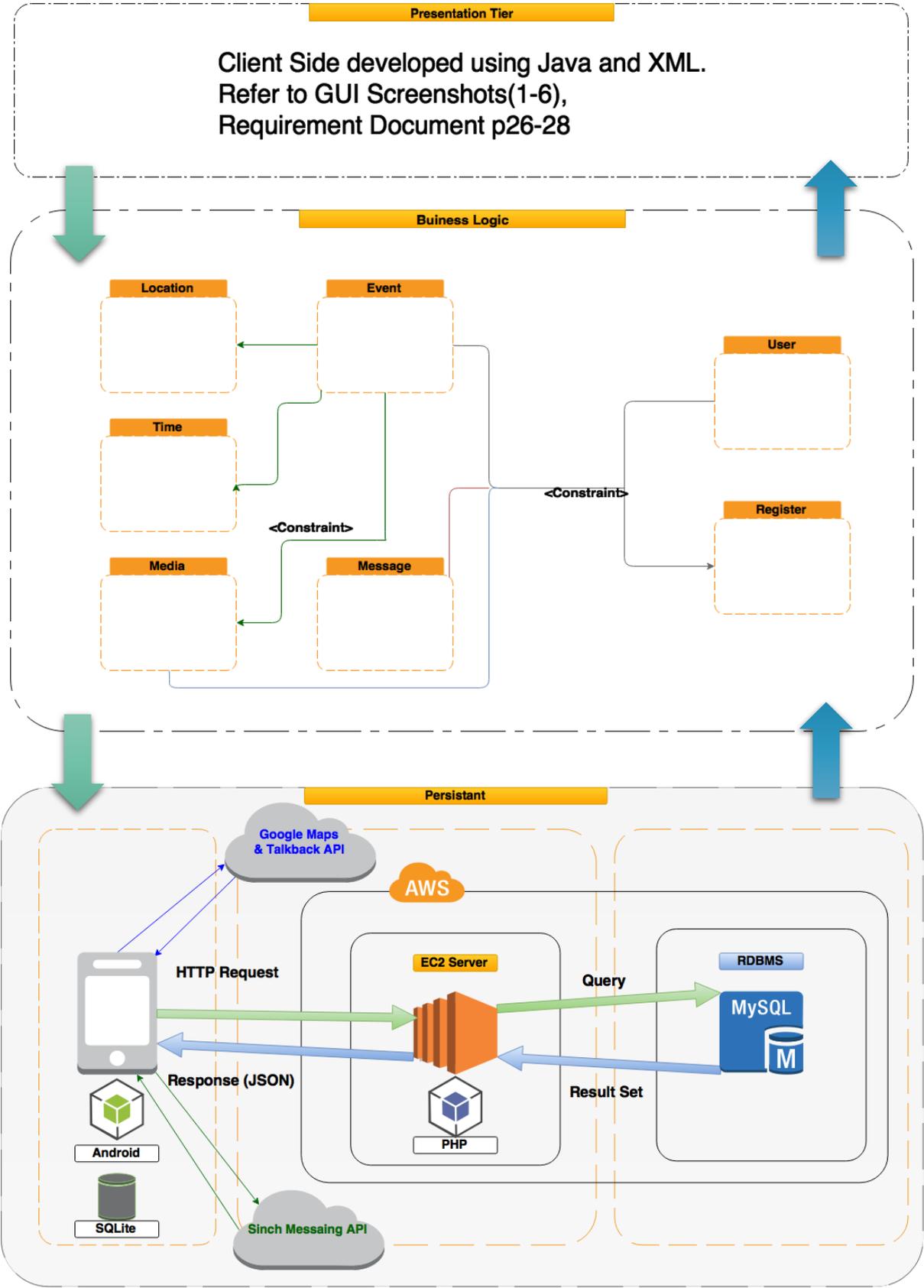
The App has a wide scope for extending functionality and it must be easy to implement each new feature without too much effort or refactoring.

2.3.1.10 Reusability requirement

If the App is designed in a modular way, then some of this code can be re-used for other new features when extending at a future date instead of rebuilding parts from scratch. This would allow for quicker iteration and faster deployment.

3 IMPLEMENTATION OF SPONTANEOUS MOBILE APP

3.1 ARCHITECTURAL DESIGN



The reason I have chosen this architecture is because I want to have my data served from a remote host and I have chosen Amazon as my platform. PHP is a scripting language that has a big community behind it and it is known to work very well with MySQL. MySQL is the world's number 1 open source database and has proven reliability, performance and ease of use. I want to use PHP, phpMyAdmin and MySQL, and Amazon allows for installation of this software in its services. The benefits of using Amazon are flexible, reliable, secure, scalable and cost effective.

Amazon needs to be reliable as it is the platform Netflix use for its business. It is very secure as that is the main priority of a cloud service and it stores all data in highly secure AWS data centres. Amazon automatically scales depending on the usage of your Application and because you only pay for what you use this makes it cost effective.

I am Using Google Maps API to integrate the map and location services required to display where the User is and where the created events are.

I want to keep the app as open to as many people as possible and have decided to try implement accessibility features that may help impaired users with API such as Google Talkback.

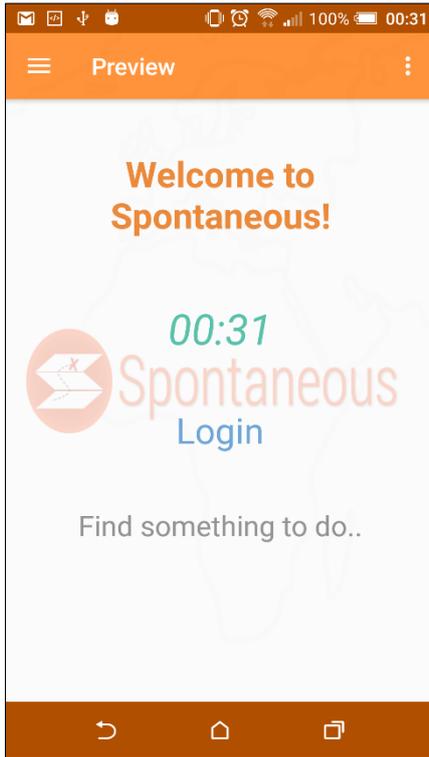
It would be useful to enable messaging within my app but I have yet to finalise which API would be most suitable. I want to have the functionality to send alerts to users as well as messaging between users. For alerts I believe Firebase Cloud Messaging may be best and Sinch messaging for instant messaging between users.

3.2 GRAPHICAL USER INTERFACE

When designing the GUI, it was essential that the application set the theme for what the app was about. The name Spontaneous was due to being able to get up and go at any time. To try keep the look matching this idea was to have it fun and exciting looking through the use of bright colours with the main colour being Orange recognised as a colour of energy enthusiasm and happiness.

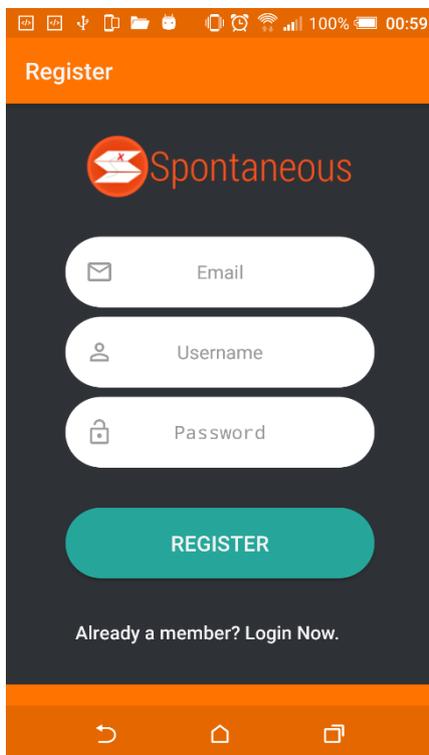
The design was about usability, so keeping the screens simple and to a minimum would mean that it could be easily learned without a manual or instructions and that the interface would be intuitive.

Launch Screen



This is the initial screen presented when the App has been launched. It shows a welcome message with the time and a call to login and find something to do. It Provides the options of Register or Sign-In, included is a Hamburger to allow a view of the Event Screen.

The Hamburger allows for the User to sample the App and view the events in both map and list form but there will be no other functionality unless Registered and Signed In.



Registration Screen

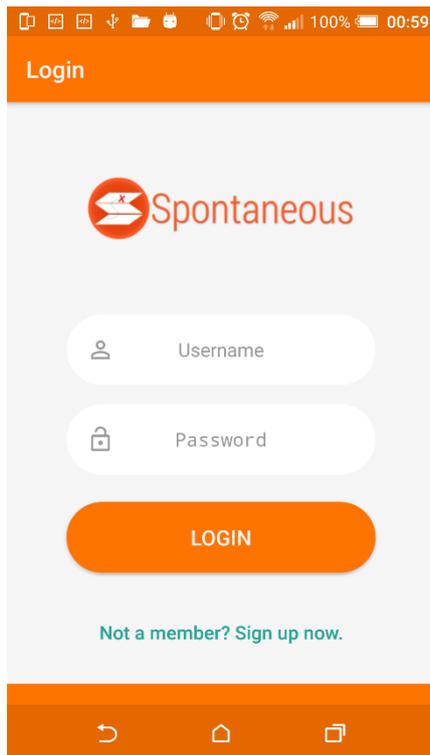
This is the Registration screen that is displayed when the User presses the Registration Button.

The User will have to enter the details required in each of the fields.

If all fields are not completed, then an error message will be displayed stating all fields must be completed.

If the details are deemed duplicated an error message will be displayed stating User already exists.

Pressing the button will add the Users details to the database and they will be registered.



Sign-In Screen

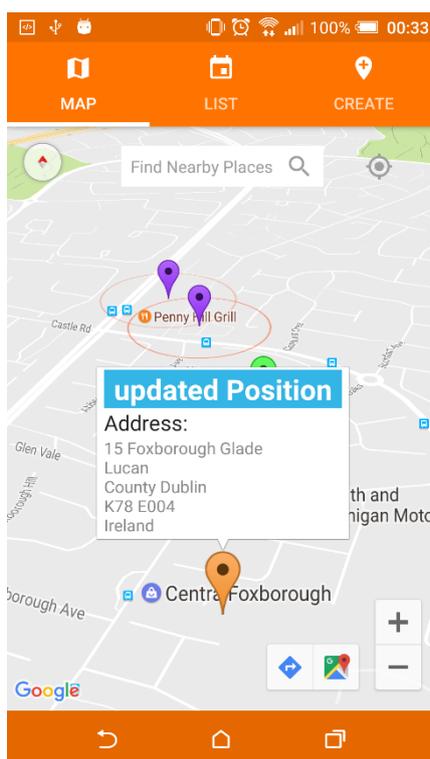
The Sign-In Screen allows the User to gain access to all functionality of the App.

The User will have to enter the details required in each of the fields.

If all fields are not complete then an error message will be displayed stating all fields must be completed.

They can also go back to the Register Screen if they wish.

If the correct details are entered, pressing the button will login the User.



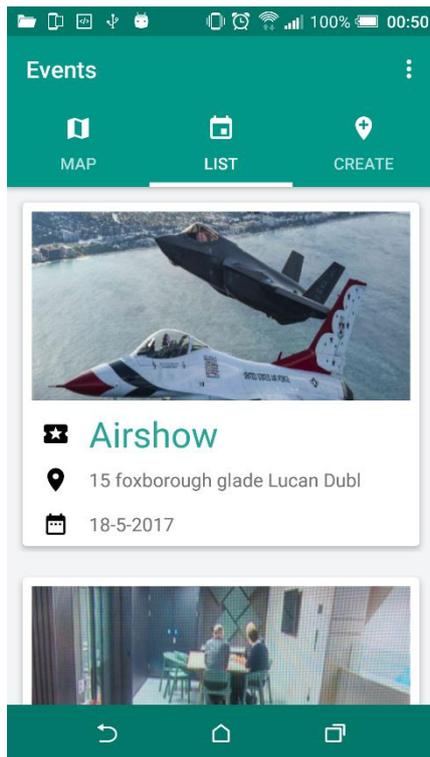
The Main View Screen

The Main View Screen is a Tab Layout that displays the Create, Map and List view of the events in the location of the User.

The events will be displayed on the map using Pins to show their location.

The events will also be shown in a scrollable RecyclerView that will be ordered by newest in descending order.

The Create screen is also available if the user wants to create a new event.

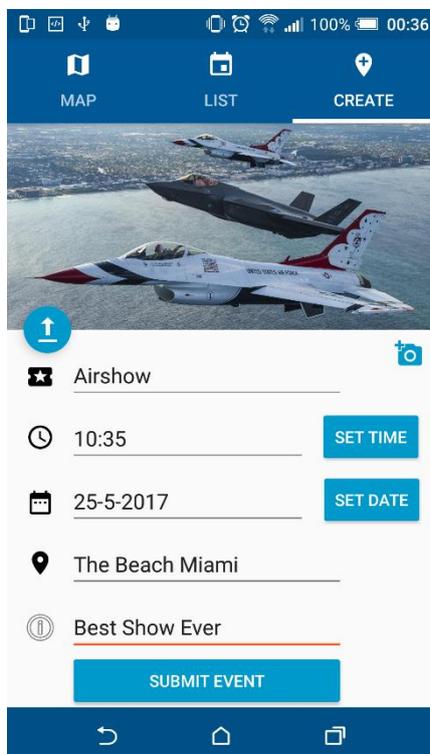


View Event Details Screen

The Event details screen is where the User can get to see the media that was uploaded for the event. The details of the event are listed on cards in a RecyclerView and is scrollable depending on the amount of details entered.

There will be buttons that will allow the User to share or favourite the event should they choose to do so.

When a card is clicked, an activity is launched with more detail about the event. At the end is a large button that will allow them to join up and attend the event.



The Create Event Screen

The Create Event screen presents the User with the form that allows them to enter the details that they wish to provide for their event.

They will also have a button that will allow them to open their gallery and upload media to help promote their event.

At the bottom of the screen will be a large button that will allow them to create the event and add it to the list of events and have it appear as a card in the RecyclerView and as a pin on the map in the Map fragment.

These are only a few of the screens that will be implemented but they are the main ones that will allow the User to get started with the basic functionality.

3.2.1 Application Programming Interfaces (API)

The API my App will be making use of are:

Android API Level 25 – Android Nougat

PHP Web Service API

Google Maps API

Google Fused Location Provider API

Google Geocoder API

Google Place Picker API

Android Accessibility API (Including Talkback)

3.3 SYSTEM EVOLUTION

The Spontaneous App is aimed at being free and available to all, however, depending on popularity, I could monetize the App through adding additional Premium Features as either in App purchases or make a separate Premium version.

I could also monetize the App by allowing commercial event advertisement and perhaps ads for business in the area surrounding the events such as shops, restaurants, bars nearby etc.

The App has a wide scope for upgrading and expanding with some features being identifiable now and others over time as users become more familiar with it and making recommendations. Events can be wide ranging so it may get unwieldy over time in this situation I would have to use some analytics to determine the main usage and frequency to better categorise events for easier searching. As the application will be hosted on AWS the scalability will not be an issue as that is taken care of automatically. The web services and database would have to be updated for any new data collection that is implemented.

4 TESTING AND EVALUATION OF SPONTANEOUS MOBILE APP

The app will require a lot of testing before it will be suitable for public use. The app will be tested to check the different features to confirm if it meets expectations for functional and non-functional requirements and produces the correct inputs and outputs. The areas that will be tested are listed below:

Usability - The mobile app should be easy to use and satisfy the expected user experience to of being simple intuitive and easy to navigate. A select few users will also be tasked to check the flow of the app and give feedback in relation to the Usability.

Compatibility – The app will be tested on a number of devices to ensure it will work correctly on devices of different screen size and hardware. This will mostly be done using the emulator on Android Studio and a couple of Android devices available to me.

GUI – The GUI will need to be tested to ensure all buttons, activities, listviews, images, text, inputs and outputs are working as expected. Espresso Test Recorder will be used to create and carry out the test of GUI features. A select few users will also be tasked to check the flow of the app.

Web Services – The app services will be tested both online and offline to ensure the tasks are fully and correctly carried out while online and to confirm that offline status is handled correctly and the user is notified of connection issues.

Performance – The performance of the app will have to tested to identify if there will be any issues between using the app on Wi-Fi or mobile data (3G) and how long it takes to request and load the data from the remote host.

Operational – The app will be tested to ensure that it will work correctly if something unexpected happens such as a loss of signal or battery running out if in the middle of transfer of data between the app and the server.

Installation – The app will be packaged as an apk file and sent to a couple of devices to ensure full and correct installation.

Security – The app will be tested to ensure that only the required data can be entered and the user only gets to see data related to their unique id.

4.1 UNIT TESTING

An android mobile device was connected to allow for testing of the application at each stage when some functionality was added. It was better to stick with a mobile device as this was more real world testing than using a Virtual device and seeing how it reacts with all the new permissions required in the newer versions of Android (API 25+). Having the device connected as the application was installed and run was essential to be able to view the Android Monitor and Logs within android studio. These showed what exactly was happening as each part of the application was used.

The type of feedback gained through this was useful to see the performance of the user interface as it would show that there were many frames skipped if there was too much work being done on the UI thread. If the application crashed or produced an App Not Responding (ANR) it would show in the monitor the error and the associated class and line that caused it.

Testing was also carried out on the other technologies used to ensure that they were all working and linking up fine. This was assisted through using the application and checking to see if the connections were being made properly and the correct data was being retrieved. An example of this was the uploading of images to the server and storing the URI to the database for later retrieval. The image that was being uploaded to the server was encoded to a base64 string and the location of this string was stored in the database. The string was saved correctly but the image did not. The PHP files had to be examined to see if there were any errors, of which there were none, however, by changing the image filetype from .jpeg to .png allowed for the successful upload to happen and the images were then visible on the server and through their URL.

The Amazon Web Services was tested to ensure that both the server and the database were reachable but secure. The host IP Address changed which did not allow for the application to connect to it properly after the change. Through examining the route tables, sub nets and security groups that separated the server from the database the issue was discovered. In order to prevent this from happening again the IP was changed to a static IP address and has worked well since.

Espresso

This is a testing framework that helps to test the User Interface automatically. The Espresso tests were run on an actual android mobile device. The framework is designed to automate tests in a way that imitates and behaves as if a user is interacting with the application frontend.

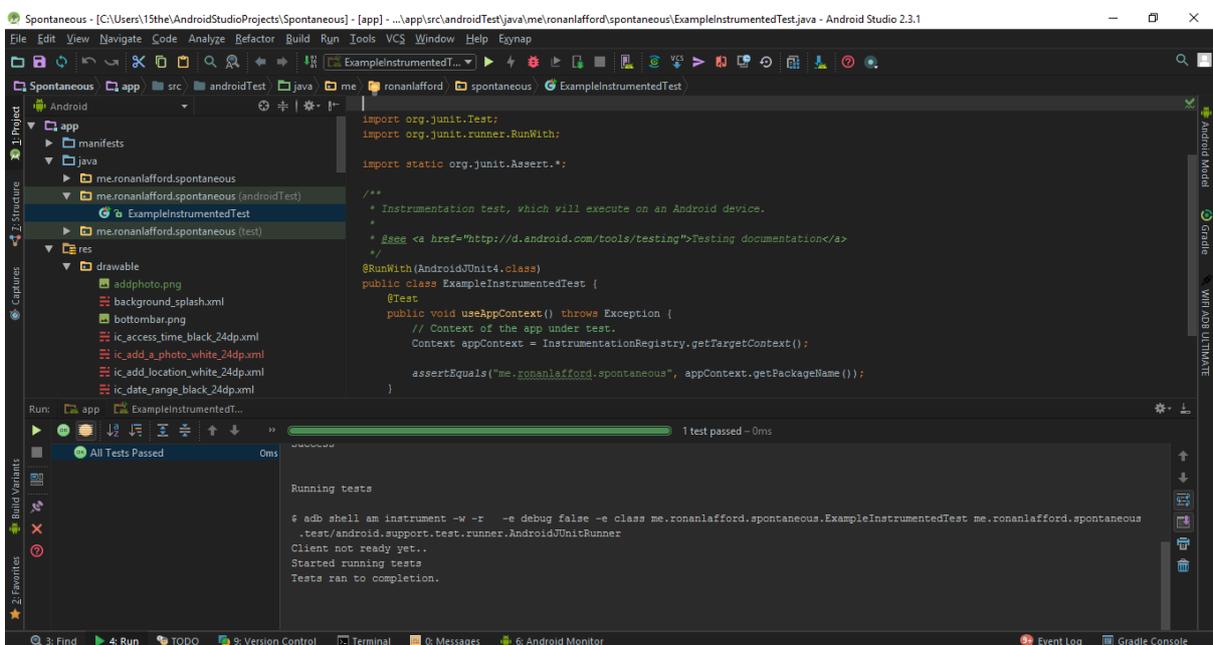
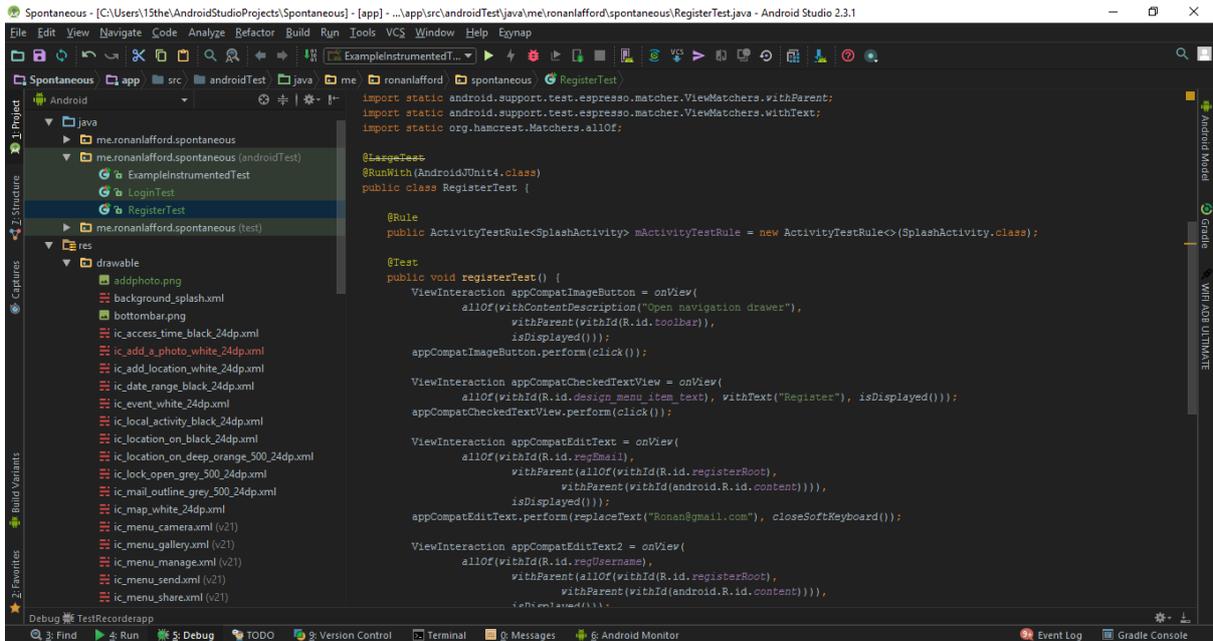
It has a simple API with synchronization of test actions with the app UI and great error information upon any test failures. To set up Android Studio for Espresso there were a number of settings had to be configured.

- Change to the Project Window
- Ensure Android Support Repository greater than 15 is available
- Check to make sure there is an androidtest/Java folder
- Turn off device animations as this may cause issues in the tests.
- Add Espresso dependencies to the build.gradle file

```
// build.gradle
android {
    defaultConfig {
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
}
dependencies {
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2') {
        // Is needed as the test runner hasn't moved to Marshmallow yet
        exclude group: 'com.android.support', module: 'support-annotations'
    }
    androidTestCompile('com.android.support.test:runner:0.3') {
        // Is needed as the test runner hasn't moved to Marshmallow yet
        exclude group: 'com.android.support', module: 'support-annotations' }}
```

- Create a Test
 - Create a test class in the location

- /androidTest/java/me.laffordronan.spontaneous.MainActivityUITest
- Run the test
 - Click run
- View result in console
- Enable show passed



4.2 USABILITY TESTING

As part of the Usability testing, a survey was created using the pre questionnaire to assist with the selection process of individuals for User Testing.

The main criteria for selection was to meet the requirements listed below

- An individual matching target audience (Matching Personas)
- An Android device user
- Willing to test the application
- Agree to Sign the consent form for testing

4.2.1.1 *Personas*

4 matching personas were found for testing that covered an age range from 16 – 58 and from whom a good range of feedback could be expected.

Beautician: Lisa (Age 36) Married with 2 young children. Likes to spend her free time with her children and bringing them on different activities. She loved the idea of being able to find something local that she could do with the kids or also with other parents. She thought the design was nice and wasn't complicated.

Student: John (Age 17) is single and still lives at home. He was excited to try a new application as he enjoys using his phone for social media and interacting with friends. He loved the idea of being able to create events like football matches for himself, his friends and others to play during the summer. He liked the simple flow of the application and thought the ability to upload photos was essential so he could show them to others.

Office Manager: David (Age 55) wouldn't use his android phone too much as he has the use of an iPhone for work, however, his personal phone is android and he uses it on weekend to organise walks with his hillwalking club. The idea was great as it would save him ringing around everyone to see if they would like to go on a walk, when he could just add it as an event for the others to see and join. He thought the design was nice and easy to use. He was amazed at the ability to get the co-ordinates from typing an address.

Nurse: Catherine (Age 26) has very awkward working hours due to her long and often changing shift times. There are many times she is home when everyone else is in work and she can get bored but she believes this may be a way to change that and make new friends in the area. Catherine uses her phone a lot but is limited to a small number of apps, mainly

YouTube, Facebook, WhatsApp messaging and Netflix. She believed the look to be similar to YouTube with the sliding tabs interface, so immediately she felt comfortable using it. She liked the addition of the Place Picker functionality as if she was at an event she could find out if there were any restaurants or shops nearby if needed.

4.2.1.2 Questionnaires

A simple Pre – Questionnaire was created to help narrow down the possible testers to those who were most suited and useful for testing.

4.2.1.2.1 Pre-Questionnaire

1. What age group?
2. Do you have a smartphone?
3. Name 3 apps you like using?
4. Name 3 apps you don't like using?
5. Would you use an app to find recreational events at your location?
6. Would you like to see events on a map?
7. Would you like to create your own event for others to join?
8. Would you like to review events attended?
9. Would you like to take part in testing for such an app?

The System Usability Scale (SUS) is a standard tool used to assess usability. It was decided to use this scale for a number of reasons as it is an industry standard which can use a small sample size while still achieving reliable results and being effective at determining if the app is usable or not.

The SUS is based on a scoring system between 0 and 100. The user must give a score to each of the 10 questions by choosing 1 of 5 responses that range from Strongly Disagree to Strongly Agree. The questions below were asked.

Link to Survey: <https://www.surveymonkey.com/r/3TMMGCM>

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.

4.2.1.2.2 The scoring

This was a complicated procedure as there were some unusual ways of scoring. The rules for scoring are as follows:

- For each odd number question subtract 1 from the rating.
- For each even number question subtract the rating from 5.
- Get each of these new values and then multiply it by 2.5.
- The Final score is a result rating out of 100

The average System Usability Score is around 68.

4.2.1.2.3 The Testing Process

Step	Time
1. Welcome / Signing of the consent form	5 min
2. Pre-test interview	5 min
3. Carrying out the test tasks	35 min
4. Post-test questionnaire	5 min
5. Post-test interview	5 min
6. Debriefing	5 min

4.2.1.3 Consent Forms

Consent Form (Adult)

I agree to participate in the study conducted by Ronan Lafford.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date:

Please print your name:

Please sign your name:

Thank you!

We appreciate your participation.

Consent Form (Child)

I agree to allow my child to participate in the usability study conducted by Ronan Lafford.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort my child or I might have with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date:

Child's name:

Please print your name:

Please sign your name:

Thank you!

We appreciate your participation.

5 CONCLUSIONS

The advantages of this project is that the idea is simple yet can reach out to a wide range of users. The great thing about the idea is that it is workable with a small amount of functionality so it should be relatively quick to get a barebones version developed initially. The ability to target a specific range of devices by API level helps to identify the potential percentage of Android users reachable. The unique selling point of this app is that it is a platform to showcase events that are all user created. Users can present their events and manage the details and attendees through an intuitive interface.

The disadvantages of the application are that it relies on a data connection for the user to be able to view all the events created. The application communicates with Google Maps API and web services to connect to the remote server to update all user, event and location details. If there is no internet connection, then the app is not fully functional. With minimum resources available for advertising, it will take a lot of exposure through word of mouth to get the app known and hoping that it may be discovered and reviewed on the Google Play store may hinder its take-up.

The project is limited in what may be achieved with the idea in the short term, mainly due to a lack of sufficient time and knowledge of the technology to create a professional and highly polished product. With the time constraints due to college, work and other commitments it is not looking possible to maximise the app to its fullest potential within the specified timeframe for the college deadline, however, I will be continuing to update and refine the app and my Android skills after that point to eventually get the app to a point where most user needs are met and all functionality required is implemented.

6 FURTHER DEVELOPMENT OR RESEARCH

As this is a new technology that I am only beginning to learn, I know that there will be unexpected results and issues that will occur and will have to research to figure out why. This

is all part of the iterative nature. For every feature that I add, I may discover newer, better ways to implement it. It will present difficult decisions as adding new functionality may reduce compatibility with some devices and limit the experience for those users affected. User needs change frequently so part of the development of the application is to keep up to date with the latest mobile trends, finding out what apps are popular and why so I can keep the app fresh and interesting. I may need to use surveys to gauge user feedback so I can refine what is most useful and what I should consider changing.

The app could turn out to be a very useful and popular piece of software as it is aimed at a wide range of users. It can be seen using Facebook that people love to share what they are doing including their photos and whereabouts at all times. It is part of human nature to be curious and to see what is going on around in the hope that they are not missing out on anything. These are factors that play a part in user retention and maintaining interest. With further resources the application could be improved not only technically but designed in such a way to appeal more to human behaviour and use psychological cues to reinforce. The use of colours, layout and imagery all add to the theme and branding of the application.

The app will be free but advanced features that may be included could lead to a premium version or perhaps the addition of specific type of ads could help to monetize the app and allow it to become more commercialised.

7 REFERENCES

Baddam, R. (2016). SQLite Tutorial for beginners - Udemy. [online] Udemy. Available at: <https://www.udemy.com/sqlite-tutorial/learn/v4/overview> [Accessed 10 Nov. 2016].

eLearning, S. (2016). (New) The Complete PHP with MySQL Developer Course. [online] Udemy. Available at: <https://www.udemy.com/php-with-mysql-developer-course/learn/v4/overview> [Accessed 28 Oct. 2016].

Google, (2016). *Developing Android Apps by Google*. [video] Available at: <https://www.udacity.com/course/new-android-fundamentals--ud851> [Accessed 1 Dec. 2016].

<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036670.pdf>, (2016). *Error Cost Escalation Through the Project Life Cycle*. Heuston: Nasa, p.1.

Lars Vogel, 2. (2016). Using lists in Android with ListView - Tutorial. [online] Vogella.com. Available at: <http://www.vogella.com/tutorials/AndroidListView/article.html> [Accessed 15 Oct. 2016].

Nielsen, J. (2016). *Why You Only Need to Test with 5 Users*. [online] Nngroup.com. Available at: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> [Accessed 21 Oct. 2016].

Official, I. (2016). JUnit Tutorial for Beginners - Learn Java Unit Testing. [online] Udemy. Available at: <https://www.udemy.com/junit-tutorial-for-beginners-with-java-examples/learn/v4/overview> [Accessed 19 Nov. 2016].

ProgrammingKnowledge, (2016). *Android Studio Tutorial for Beginners (Step by Step tutorial)*. [video] Available at: https://www.youtube.com/playlist?list=PLS1QuIWo1RIbb1cYyzZpLFCKvdYV_yJ-E [Accessed 10 Nov. 2016].

Regmi, P. (2016). *Android JSON Data Parsing Tutorial with Example*. [online] Viral Android – Tutorials, Examples, UX/UI Design. Available at: <http://www.viralandroid.com/2016/12/android-json-data-parsing-tutorial-with-example.html> [Accessed 5 Nov. 2016].

Support.google.com. (2016). Turn on TalkBack - Android Accessibility Help. [online] Available at: <https://support.google.com/accessibility/android/answer/6007100?hl=en> [Accessed 28 Nov. 2016].

Sinch. (2016). Voice API and SDK Documentation | Sinch. [online] Available at: <https://www.sinch.com/docs/voice/> [Accessed 10 Dec. 2016].

Svekis, L. (2016). Github introduction to version control and remote files. [online] Udemy. Available at: <https://www.udemy.com/github-introduction-to-version-control-and-remote-files/learn/v4/content> [Accessed 3 Nov. 2016].

Tamada, R. (2016). How to connect Android with PHP, MySQL. [online] androidhive. Available at: <http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/> [Accessed 6 Nov. 2016].

Waters, K. (2016). *What Is Agile? (10 Key Principles of Agile) | All About Agile*. [online] Allaboutagile.com. Available at: <http://www.allaboutagile.com/what-is-agile-10-key-principles/> [Accessed 21 Oct. 2016].

8 APPENDIX

8.1 APPENDIX 1: PROJECT PROPOSAL

8.1.1.1 Objectives

The purpose of this report is give an overview of the development and construction of an Android Application that I will be creating. The name I have chosen for this application is “Spontaneous”.

Spontaneous will be a social networking/event app with the core functionality of allowing users to register, view, create and partake in events that may take their interest, at any time. The name Spontaneous reflects the spur of the moment nature of the app. Users will be able to see all “happenings” in their area and beyond at that moment.

With the intention of the app focusing on real time, the user will be able to navigate quickly through the clean well defined screens within a small number of clicks. The aim is to try have all functionality reached with 2 clicks for a better user experience. I will try have an interface that will be as minimalist and intuitive as possible to allow for better performance and ease of use. The activity and map data showing the events should be available and up to date when accessed. The app needs to be fast to maintain interest and keep with the live theme.

User feedback will be used to rate activities and the ability to upload and view photos to show previous gatherings can encourage future participation. With the collection of user's age, I hope to have a filtering system to restrict users from seeing certain events e.g. an underage person seeing an event at a pub.

As a social app I plan to incorporate the ability to call and message other members including sharing of events. With all events, there is the possibility of too many people turning up, so being able to turn off the event notification when a specified amount of attendees confirm is needed.

If a user attends events regularly they may wish to show others what they were like, therefore, a list of recently attended events should be available to the user. In order to keep the app interesting and engaging. I will not bombard the user with too many features initially. The core functionality will be expanded incrementally with "must -have" features and then "would-like" features depending on time available.

I intend to publish the application on the Google Play Store when complete.

8.1.1.2 Background

As a husband, father of two young children, working full time and studying part-time I am fully aware of the small amount of free time available for leisure. I try to make the most of any time I can spare by enjoying it with my family and friends. A common scenario is that there are many times when you may just not know what to do, cannot think of anywhere to go or probably don't have enough money to do certain activities.

The motivation for Spontaneous is to alleviate this frustration by being a gateway to the community and presenting the user with a host of activities that may be currently underway or planned for a later time. Examples of such activities could be a parent creating a parent – toddler meet up where other users could view it and join the event with their children. A

hobbyist may enjoy coding and create a Coding event inviting other enthusiasts to join in working on a project. A user may wish to create a litter picking activity and other users could volunteer to help if interested.

I have not found a solution to these types of situations readily available or catered for all in one place and I believe that Spontaneous can satisfy this need and will help develop friendships, build community ties and increase social inclusion by bringing people of all ages together while having fun.

8.1.1.3 Technical Approach

Creating a mobile application requires continuous feedback and adjustments throughout all stages of its lifecycle. Android is a mobile operating system that is used on many different devices all around the world. These devices are evolving extremely quickly with new hardware and specifications almost every year. It is very difficult to cater for all devices, screen sizes, and user requirements.

This can be seen with the constant flow of updates and improvements for the majority of apps on the Google Play store and even the Android operating system itself now into its 7th iteration called Nougat which has the Material Design standard that is not available in some older versions. Working with an Agile approach would probably make the most sense given the scope of the project and having the application as widely available and user friendly as possible within the completion deadline.

I aim to incrementally develop my application in as modular, low coupling and high cohesion manner as possible. I want to be able to deliver the core functionality at the earliest in order to have a functioning prototype. This will ensure that I will at least have a working application in the event of any unforeseen delays or obstacles that may hinder the desired progress. Working in this manner will allow for easier identification and correcting of bugs as I will be focusing on specific features on each lifecycle iteration.

I have some experience with Java but have never developed an Android Application before. This is a challenge but I have set about researching and learning the best tools and processes to use. With limited time I have to be smart about my strategy, the temptation to open the floodgates to Android Tutorials is strong but I have to resist as I would only be overwhelmed with the amount available. I have decided to learn as I build, what component do I need? how

do I make it work. This reduces time spent on the unnecessary and allows concentration on what is needed according to the Specification Requirements.

8.1.1.4 Research

I searched for any apps that may be similar to Spontaneous and useable within Ireland in order to see how they were implemented and how good the user interface and flow of the app may be. The app that I found to be the most similar to my app is called Eventbrite, however, it is very limited for Irish users, the events are for more commercial events being advertised than actual user events and the main difference is that a user cannot create events.

Being an Android smartphone user, having installed and used many apps, I can take cues from the ones that I have kept and those I have deleted by looking deeper at the reasons why some made more of an impression than others. I can use this knowledge to assist with the design of my application.

Rather than just rely on my own ideas for the app I decided to try gain some opinions and insights from other Android users from across the spectrum of technical ability. I interviewed two friends James and Stephen who would be of similar age and use social apps fairly regularly. Stephen, who is a single man, would attend social events frequently believed my idea would be fairly good as he likes to be able to head out at the drop of a hat and to have the option to view and join events at any time is attractive. He would rather attend than create events and thinks having a list of attendees would be good to see what kind of crowd would be there before going. He didn't have much else to offer but just wanted a simple user Interface to see all details clearly with minimum searching. James being a family man would like to be able to bring the kids out and thinks having the option to make re-occurring events would be good to help the children make new friends. James works in his family business and travels a lot around the country as part of his role. He thinks he could use the app for advertising product demonstrations or information sessions and would like to see the ability to invite clients to these events.

Desmond is a middle aged man who works voluntarily within the community as part of a kid's club for afterschool activities who teaches technology. He loves the idea of being able to create an event for the children in the area to see and who can then attend. The children all have access to smartphones so he thinks be able to set up an alert to notify them of an event

would be great and he would be able to prepare better knowing the numbers that will be attending.

Ann is a retired widow who has more time to do things and likes to meet up with friends on a regular basis and likes to spend more time doing hobbies. She would not be the most technologically literate so her advice is to make it very intuitive and simple and not have too many screens or buttons to navigate. She believes that it would be nice if the text is clear and easy to read. She wants to be able to show the family and friends, photos of events and people that were there.

I will be relying on Google's Android Developers site to keep informed of the main principles and best practice in Android development and design. I will be making use of our college modules that cover Web Services and Restful API's, Mobile Application Development, Databases and many external sources from the internet including YouTube, Pluralsight, Codecademy, Udacity, Udemy and various books available from the library.

8.1.1.5 Requirements Capture

As I am developing the app as something that I would like to use myself, I know a lot of features that I want to incorporate from the start and many that can be added as updates at a later date to keep it fresh and maintain user interest.

I know that there may be features that I would not think of or feel I even need, but for completeness and to cater for a wider variety of users I need to get other perspectives. I will be asking friends and colleagues of various ages if they would use it and if so what they would like to see in this type of application.

I will write User Stories to make it easier to identify all the use cases and requirements needed to fulfil my objective.

With my own research of similar applications, I will take note of all the common features between them. This will be a good baseline to see if I have left anything out or if I am going in the right direction.

As I will be taking an Agile approach I will try keep to the 10 Key Principles:

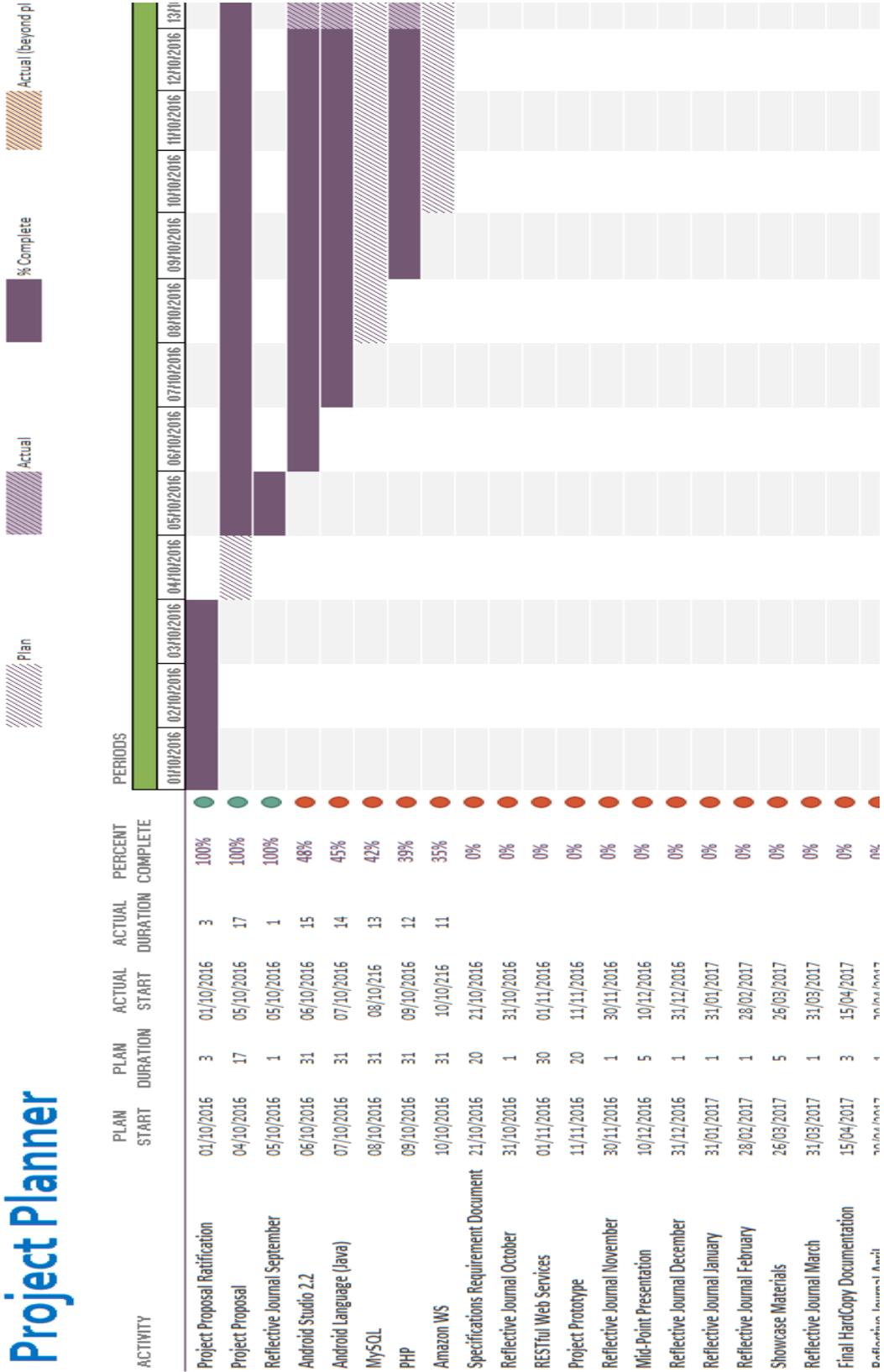
1. Active user involvement is imperative
2. The team must be empowered to make decisions

3. Requirements evolve but the timescale is fixed
4. Capture requirements at a high level; lightweight & visual
5. Develop small, incremental releases and iterate
6. Focus on frequent delivery of products
7. Complete each feature before moving on to the next
8. Apply the 80/20 rule
9. Testing is integrated throughout the project lifecycle – test early and often
10. A collaborative & cooperative approach between all stakeholders is essential
11. (Waters, 2016)

8.1.1.6 Special Resources Required

Android Studio 2.2 will be the main IDE for the development of the application. I have signed up to the GitHub Student Pack to make use of the excellent offers available. I will have access to free Private GitHub Repositories for use in this project. The subscription to Amazon Web Services is essential for hosting my remote database and the bonus of \$150 dollars in credit is more than enough to cover the usage during testing and for the period up to examination. Additional resources and expenses will be added to this section before final submission if required. I aim to publish the app on Google Play store so this will incur a future cost of about \$25.

8.1.1.7 Project Plan – Gantt Chart



(Gantt Chart Zipped with Proposal)

8.1.1.8 Technical Details

Android Studio:	The main IDE for developing the application.
Android:	The primary language used will be Java as Android is based on the Java language.
PHP:	Used to create the web services that will connect my app to the remote database.
MySQL:	Backend to store all data including user details, locations and events etc.
SQLite:	The device storage to store some details on the mobile device such as recently attended events.
JSON:	The format of the data exchanged between client and server.
CircleCI:	Continuous Integration tool for running automated tests.
JUnit:	Used for testing the Android application.
Espresso:	For testing the UI of the Android app.
Google Maps API	This will be used for adding event locations to maps.
Google Place Picker	This is to view business details within the area of an event.
Google Geocoder	This is to allow for geocoding and reverse geocoding of locations.
Google Fused Location Provider	Better location finding and power management
Google TalkBack	This will be used for adding accessibility to the app.
Sinch Messaging	This will be used for in-app instant messaging.

8.2 APPENDIX 2: EVALUATION

The planned workflow is to have my Android studio linked to my private GitHub Repository. I aim to integrate a continuous integration tool such as CircleCI. This means that when I make a change and push to GitHub it triggers the CI tool which runs JUnit Instrumented tests and checks the app UI with the Espresso APIs to perform click actions on UI elements for integration testing and to ensure there are no errors in the new build and will deploy the updated project to my remote host if successful.

It is important to gather feedback in order to detect errors or defects as early as possible in projects, according to Nasa, at each stage of the lifecycle, changing any requirements later and later increases costs (my costs would be reduction of time) exponentially.

(<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100036670.pdf>, 2016)

“Summary: Elaborate usability tests are a waste of resources. The best results come from testing no more than 5 users and running as many small tests as you can afford.” (Nielsen, 2016)

For my application I will follow Nielsen’s view that only a small numbers of users are needed to test, therefore, I will have a maximum of 5 users who will be tasked to use all the functionality of the app and provide feedback on

- Usability
- Functionality
- Speed
- Features

I will implement any changes that **all users agree** would make an improvement to the application and re-issue the app for testing after each iteration (depending on time available).

8.3 APPENDIX 3: MONTHLY JOURNALS

8.3.1.1 *September Reflective Journal*

I finally decided on my project after many weeks trying to decide between a couple of ideas. I was trying to find a good balance between having a project that was interesting to me, having it sufficiently complex enough to be approved and to appeal to a wide range of people.

I felt creating an Android app would cover all three criteria. I had been inspired by colleague's inability to get tickets for events or even sell them at the last minute. My first idea involved real time buying and selling tickets so one could see who had tickets for sale and where they were in the local vicinity on Google Maps. At my presentation the panelists liked my idea but were unsure of the legality of re-selling tickets within an app. I changed the idea to one of a social networking, event finding, event creation and sharing using all the same technologies involved. I think that this worked well as it actually improves upon my criteria for a wider appeal and greater complexity.

I have decided on a name for the application and set about researching similar applications and what I believe I would like to see in such an app. The difficulty now is that deciding so late means that I have a lot to figure out in a short time.

The workload will be heavy with other modules CA's and trying to have the project proposal ready in time.

Supervisor Meetings

No supervisor assigned yet.

8.3.1.2 *October Reflective Journal*

This month was tough as having to change the idea for my project I had to come up with a way of fitting it to the technology of the original idea and expand upon the functionality for it to be a fit for purpose event application.

I am not the most visually creative person, so trying to decide how I want it to look is frustrating as I am never fully happy with my styles. This is an issue for me as I tend to get hung up on it which wastes time. I have decided I will forget about how well it looks for the

moment and just create the screens for the intended activities and try make it basic but functional for now to assist with ironing out my requirements.

Android Studio can be extremely slow to use. I have tried to find tips on how to speed up the Gradle build and use my phone instead of an emulator. This has helped but the IDE itself can be slow to respond, however, I have gained a bit of experience finding my way around it through experimenting with different settings etc.

I want to host my database on Amazon but I didn't realise how much of a maze Amazon can be with all the different services. I had to try discover what was best for me, including what would be suitable for adding PHP files for web services. At present, I believe that it would be best to use Amazon EC2 as the server with an installation of phpMyAdmin and link it to an Amazon RDS with a MySQL database.

My biggest focus is on the interaction with the back end so I have tried to watch a few tutorials on how to use PHP and MySQL and have the app make requests through the AsyncTask and HttpURLConnection. I am happy that this is becoming a bit clearer now and I am understanding what is happening. I have installed WAMP server to experiment and do some trial runs before moving onto Amazon. I have made a bit of progress in becoming clearer in what I want and understanding what I need but it is all a bit slow as finding the time to get done what I want is difficult with other modules and external factors taking my attention. I will need to up the progress next month and try and keep on top of everything.

Supervisor Meetings

Dr Catherine Mulwa

Date of Meeting: 08 November 2016 (1st Meeting)

Items discussed: Project Proposal and app functionality

8.3.1.3 November Reflective Journal

I have made a decent bit of progress this month as I have dedicated more time to the project. The requirements document was a good task as it allowed me to flesh out the project and get thinking more thoroughly about what I intend to achieve and how I expected to approach each requirement. I had a good meeting with Dr Mulwa and discussed my use cases and architectural diagram which helped clear a few things up in my mind which in turn led me to

change a few things. I wanted to put as much as I could into the document when I began writing it but I did not realise how much I was actually going to have to write. The more I explored each objective, the more I found myself having to expand them which resulted in the document growing ever larger. I finally completed the document but did not have a chance to work on any prototype.

Having the Mobile and Multimedia module is a great help in relation to my project as it is focused on Android and some of the functionality that felt I could use for the project. As part of the module we had to create an android app for one of the CAs, so I used this as way of killing two birds with one stone. I developed an app incorporating functionality that I wanted to learn and use for the project such as SQLite database, AsyncTask, connecting to web services etc. I'm happy with the app I created for that CA and I now feel more prepared to attempt to create a prototype for the midpoint presentation and becoming more confident that I can continue to progress my learning and be able to complete the project to a decent standard.

Supervisor Meetings

Dr Catherine Mulwa

Date of Meeting: 15 November 2016 (2nd Meeting)

Items discussed: Project Requirements, Use Cases and Architectural Diagrams

8.3.1.4 December Reflective Journal

December wasn't a great month for progressing on the project due to CA's and large workload. I wasn't happy at not being able to properly work on the app as I know that it will be difficult to keep up and with exams there will even less time to work on it. I'm hoping that after the exams I will be able to do some more tutorials and try and work on the backend and the PHP web services as I think this may be the most difficult part. I have the Amazon EC2 web server set up with phpMyAdmin installed and linked to an Amazon RDS with MySQL installed. I have also created the security groups to enable communication between the server and the database. This restricts the traffic to connect only with the database via the web server.

I pretty much know how I want it to look now after many ideas and layouts I created did not suit the app. I have the layouts created now so it will be mainly working on linking the screens up and getting the functionality working in order to connect to the web services.

I intend to work on the project as much as possible after the exams, hoping to do at least an hour or two an evening to try and get to where I want to be with it.

Supervisor Meetings

Dr Catherine Mulwa

Date of Meeting:

Items discussed:

8.3.1.5 January Reflective Journal

I felt January was a good month as I managed to get the connection between my application, the PHP web service and the MySQL database working properly. This was a major issue before Christmas as it was not working and I spent a lot of time trying to figure out why. Fixing the issue led to me being able to progress and get back to working on the UI of the app. I was able to implement the Google Map and Google Places API which allows the user to find their location and view their position on the map. For the user to see the actual address, I added a geocoder to return the detailed address from the co-ordinates found by each marker and presented them in an info window when the marker is clicked.

It was essential that I had this working as this is what the main objective of the app is based on. I added the Google Places API as I believe this would be useful to find services such as restaurants, transport etc. in the area a user is attending an event. I am a bit happier now as a major obstacle was overcome. I feel I am getting more comfortable with android and getting better at adding functionality to the app using different layouts and views such as Tab Layout and the advanced list type of Recycler View with Card View items.

My next step is to try and get the UI completed in February and to start finalising the database tables. Once I get the tables finalised then I will be able to work on the PHP logic to carry out the CRUD operations needed.

Supervisor Meetings

Dr Catherine Mulwa

Date of Meeting: 31/01/2017

Items discussed: Mid-Point Feedback

ACTION ITEMS: UPDATE SYSTEM ARCHITECTURE DIAGRAM FOR BETTER STRUCTURE AND VISIBILITY. UPDATE TECH REPORT FOR ANY CHANGES MADE.

8.3.1.6 February Reflective Journal

February was ok progress wise, it was fairly tough to do what I wanted within the app as I am working with fragments which are not as straight forward as activities. This was an issue as there are not many tutorials or examples that deal with fragments, they are all geared toward the single activity. As my aim is to try and keep the app as simple and usable as possible with the minimum amount of clicking, I wanted to keep working with the tabbed fragments that would allow a simple swipe to change screens.

I tried to style each of the fragments so the user would have some feedback when swiping through the screens. I coded the status bar, action bar and navigation bar to change colour as the tabbed was swiped. For extra functionality for the map I wanted to be able to alert the user to any events that might be happening in their location. To do this I implemented Geofencing, which is using the event marker to find the location and then creating a ring or fence of a certain radius around the event. This meant if the user walked inside this fenced area then they would be notified that they have entered the zone of an event and if they click on the notification the app would open to show it.

With the event list Tab I have connected the database to the Recycler View and can show the events that are created. To keep the Event list up to date, the pull to refresh functionality was added which brought the newest event in at the top of the list. I feel that I have been able to figure out some important features that I will need to properly implement and tweak for the final version of the app.

Supervisor Meetings

Dr Catherine Mulwa

Date of Meeting:

Items discussed:

8.3.1.7 March Reflective Journal

March was a month where I did not manage to get much done between family commitments and CA projects for other modules this semester. I tried to move onto additional functionality that I wanted and felt the user would like to have, the ability to take photos of the events and upload them to the server for viewing within the app. I decided against creating an Amazon S3 bucket for this version of the app and instead just decided to create a photo folder on my current EC2 server. The aim was to upload the photo to the server folder and save the address to the MySQL database so the app could reference the address instead of saving the image on the mobile device. I have managed to have a button open the camera gallery in order to select a photo and once selected I can then upload the photo and save the address, however I have issues that the address is saved in the database but the photo has not transferred to the server and gives a timeout error instead.

I am trying to rectify the issue but having no luck so far with various fixes such as extending the timeout time. I may have to look at using a library that can handle this task without error. When this is fixed the next step is to link the event list to the map so a marker is added every time the list is updated and if the marker is clicked it should open the event details screen. If I can get this working, then I think I will be almost there with only a few smaller tasks to add such as creating a rating and review section which should not be too difficult with what I have learned to date. I aim to then use April for finalising the app and technical document with any changes and including testing (including Usability) that I have carried out on it.

Supervisor Meetings

Dr Catherine Mulwa

Date of Meeting:

Items discussed: