

National College of Ireland
BSc in Computing
2015/2016

Fiachra Devitt
x12317886
x12317886@student.ncirl.ie

Supervisor:
Arghir Moldovan

TapReference

Technical Report



Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name:	Fiachra Devitt
Student ID:	x12317886
Supervisor:	Arghir Moldovan

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: **Fiachra Devitt**

Date: **11/05/2016**

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,

Table of Contents

1	Executive Summary	5
2	Introduction	6
2.1	Background.....	6
2.2	Aims.....	7
2.3	Technologies	7
2.4	Structure	8
3	System.....	9
3.1	Requirements	9
3.1.1	Functional requirements.....	9
3.1.2	Data Requirements	11
3.1.3	User Requirements	11
3.1.4	Environmental Requirements.....	11
3.1.5	Usability Requirements	12
3.2	Design and Architecture.....	13
3.3	Implementation	15
3.3.1	Technology overview.....	15
3.3.2	Technologies.....	15
3.3.3	Procedures.....	16
3.4	Testing.....	20
3.4.1	Unit Testing	20
3.4.2	Usability Testing.....	21
3.4.3	Security Testing.....	21
3.5	Graphical User Interface (GUI) Layout.....	21
4	Conclusions	25
5	Appendices.....	26
5.1	Project Proposal	26
5.1.1	Objectives	26
5.1.2	Technical Approach.....	26
5.1.3	Special resources required.....	27
5.1.4	Technical Details.....	27

5.1.5	Evaluation	28
5.2	Project Plan	29
5.3	Monthly Journals.....	29
5.4	Requirement Specification	39
5.4.1	Introduction	40
5.4.2	User Requirements Definition	40
5.4.3	Requirements Specification.....	41
5.4.4	System Evolution.....	52
6	Bibliography	53

1 Executive Summary

The core idea behind TapReference is to help users promote their work experience and showcase the impact they had upon their fellow co-workers, bosses, managers etc. through a medium that makes gathering that information fast and simple. To achieve this, users must first register themselves to create a personal account which then produces a profile page for that specific user. Other users of the service are then able to create a reference for that specific user. The user who owns the profile can then choose whether or not to allow a referral that has been sent to them, to be displayed on their profile page. This is to prevent bogus or malicious “*referrals*” from appearing on their profile.

To give potential employers a better idea of what kind of perspective this user’s talents are being viewed from, users who submit referrals will have their current area of employment displayed next to their referral. This will help give other users an idea of what kind of position this person has in the organization which they work for such as the position of CEO, a manager or a developer.

I intend to implement this solution using a combination of Ionic framework, AngularJS and Firebase to create a hybrid mobile application that is capable of storing and retrieving information upon request.

2 Introduction

2.1 Background

The idea came from the thought that employers would like information about how well an employee was received in their prior work experiences. To achieve this I considered the possibility of an app that allows past employers or fellow employees to post informative data for future employers to use to help them decide whether or not this potential employee has the skills (i.e. Leadership, Social, Planning, Charismatic skills etc.) that they are looking for. Additionally this app could prove useful for users who would like to have the impression they left on their colleagues and what skills they displayed to them put onto their profile.

I would like to create this idea such that users can post referrals on each other's profiles for the purpose of increasing their chances of employment.

I have considered the possibility of using a ranked user system as a means of displaying a user's status within a company (e.g. whereby a delivery man would be a rank 10 user and a CEO of an enterprise corporation would be rank 1). This has the potential of conveying the importance of a user's referral. In a rather extreme scenario, if a rank 1 user were to post a referral of the user then it would show potential future employers that the employee had enough impact in that company that their reputation was good enough to be recognized by its CEO.

With that said however, I also feel that it is possible that the idea previously mentioned may simply be overcomplicating something very simple. Another way of addressing this would simply be to explicitly list the job title of the user making the referral.

I have also taken into consideration the idea of providing the user with the ability of allowing or disallowing referrals onto their profile should they believe its content to be distasteful or inflammatory.

2.2 Aims

The main aim of the project is to create a mobile application that allows users to send and display referrals so that clients interested in viewing them may do so to better their understanding of an individual. This in turn should assist in their decision as to whether or not they are interested in giving the individual an opportunity to have an interview for a job position.

2.3 Technologies

Ionic – an open source framework used primarily for developing hybrid phone apps which utilise HTML, CSS, SASS and AngularJS(Drifty, n.d.)

It will assist in the creation of my project by greatly simplifying the process of creating a mobile application.

AngularJS - a structural framework that utilises and extends HTML's syntax to define an applications components. It is used to assist in the development of dynamic web applications.(N/A, n.d.-b)

I am currently using it in my application to handle the services and controllers to be able to send data retrieved to Firebase.

HTML – stands for “Hyper Text Markup Language”. As its name implies, it is a markup language and is used to describe web pages to browsers. It utilises standardised semantics to help different browsers display the same web page as closely to each other as possible.

I will be using HTML to handle the elements present on each page of the app.

CSS – Stands for “Cascading Style Sheets”. It is used in conjunction with HTML to assist in stylising the elements present on a HTML page.

I will be using it in my project to create a graphical user interface for my app.

SASS – Stands for “Syntactically Awesome Stylesheets”. It is a scripting language which uses an interpreter to become translated into CSS code.

In essence it can be used similarly to bootstrap which is mainly the purpose for which I will be using it in my project. This will essentially make creating the GUI faster.

Firestore – A backend as a service company that provides its users with free cloud hosting and provides developers with support in the creation of web-based applications. (N/A, n.d.)

In my project it is being used to store user data upon their registration. This data can then be recalled when necessary such as when a user wishes to log in again or query the search tool to locate other user profiles.

2.4 Structure

Chapter 1: The introduction to the project which details the concepts behind the core ideas of the project.

Chapter 2: Describes the project's system, what it requires to function and its design and architecture

Chapter 3: Lists all resources referred to inside this document

Chapter 4: Includes the project's initial proposal, how it is intended to be carried out and the journals detailing my progress in creating the project on a monthly basis.

3 System

3.1 Requirements

3.1.1 Functional requirements

Listed below are the functional requirements for each screen of the app.

The Login Menu

Enter Email (Input field): The app will have an input field which requires the user to input an email which has already been registered on the database.

Enter Password (Input field): The app will have a second input field which requires the user to enter the correct password associated with the registered email typed in the previous input field.

Login (Button): Once users have entered valid data into the fields mentioned above, they may use the login button to authorize themselves to access their account. However in the event that one or both of the required input fields have invalid data inside them, the user will be denied access to the service and will have to try logging in again.

Register (Button): Allows users who have not created an account yet, to do so by taking them to the registration screen.

The Registration Menu:

Enter Email (Input field): Allows the user to submit their email to the database to be used for account authorization purposes.

Enter First Name (Input field): Allows the user to submit their first name to the database and have it associated with their account

Enter Second Name (Input field): Allows the user to submit their second name to the database and have it associated with their account

Enter Password (Input field): Allows the user to set a password for the account in the database to be used for account authorization purposes.

Confirm Password (Input field): Requires the user to confirm that they have entered their desired password correctly. If both passwords match, the password is assigned to the account, if they don't, the user must retype the password.

Register (Button): Registers the user's account and saves the entered data from the input fields to the database, allowing the user to log back into their account if they return to the service after exiting.

Profile Page

User Details (HTML & AngularJS): Will call information from the database based on who owns the profile being viewed and will display details such as the user's full name.

Referrals (HTML & AngularJS): Will call the text data of a referral submitted to that particular profile from the database and display it using HTML tags.

Reference Details (HTML & AngularJS): Displays the information of the user who submitted the referral.

Create Referral (Text Area): Allows a user browsing the profile to enter text into the text area.

Submit Referral (Button): Collects the text data stored in the text area for creating a referral and submits it to the database.

Search Page

Search (Input Field): Allows users to type the name of the person who's profile they are looking for and queries the database for account possessing the same or similar names.

Search Results (HTML & Angular): Displays the search results which upon click, take the user to the profile the result represents.

3.1.2 Data Requirements

Firestore Database: All data submitted through the app by users can be found on the Firestore database. This information can only be accessed by querying the database through the use of AngularFire, an open source library developed by the Firebase staff that provides developers with a three way communication system between the HTML, JavaScript and Firestore database of a given project or application (N/A, n.d.-a).

3.1.3 User Requirements

Android phone: To access and use the app, the user must be using an Android Phone.

Google Play Store: To download the app, the viewer must first download Ionic View, the official Ionic app which is used to access apps made with Ionic framework.

Android Version: To use Ionic View users must have an Android version of 4.1 Jellybean or higher.

Internet Access: The user must have access to the internet to use the app, otherwise the app will be unable to make queries or send data to the database.

3.1.4 Environmental Requirements

Computer (Desktop/Laptop): Ionic is fairly lightweight in terms of resource usage so any standard desktop or laptop running windows 7 or higher should be able to run it.

NodeJS: To install correctly, Ionic Framework requires that NodeJS be installed onto the windows machine being used for development in order to help create, manage and serve applications using JavaScript on the V8 virtual machine.

Cordova: Used by Ionic to build on top of, by using Cordova to package the HTML5 app for Android, Ionic is able to then provide the “front-end” components such as the HTML, CSS and AngularJS used to construct the app by the developer.

Android SDK: To develop for android first the android sdk tools must be installed onto the computer.

Ionic Framework: A mobile app development framework that uses a combination of HTML5, CSS, AngularJS to build hybrid apps for smartphones. Ionic can be installed onto any Windows computer by entering a few short commands into the command prompt.

Text Editor: Such as Notepad++, Atom or Sublime text that can be used to view and edit multiple files of different types and format them in a way that is logical.

Pixlr: A free online graphics editor which I have prior experience in using from my past work experience, I will use it for editing any graphics I want to add to the app.

Internet Access: To manage and Access the Firebase database I must have a connection to the internet.

Firebase Account: To use Firebase I must create an account on their site in order to generate a database.

3.1.5 Usability Requirements

Interface simplicity: I want users of the app to be able to understand what they need to do to get the desired result of the action they wish to accomplish by providing a simple, easily understood UI.

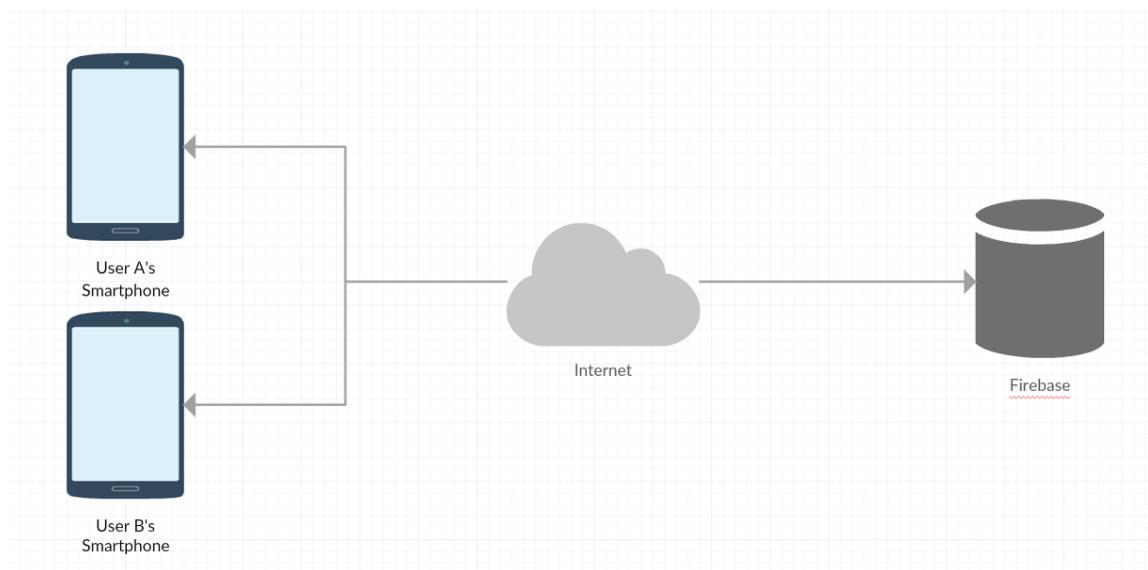
Operability: Actions performed in the app should be consistent in terms of their functionality. Any errors that occur should be explained to the user using an error alert box when the system detects that an error has occurred.

Learnability: From the user's perspective, the app should be relatively straightforward in terms of how to use it. Users should be able to learn how to use the app's functions quickly.

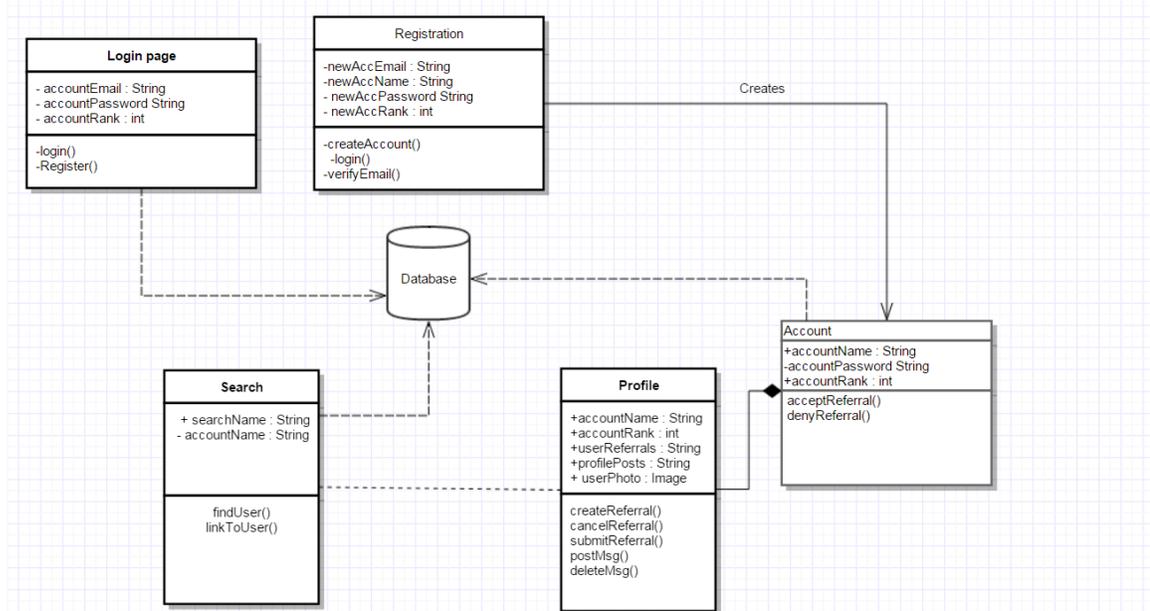
Attractiveness: The application should use colours that are easy on the eyes and utilize a layout that separates content into different areas to ensure that the interface is not convoluted.

3.2 Design and Architecture

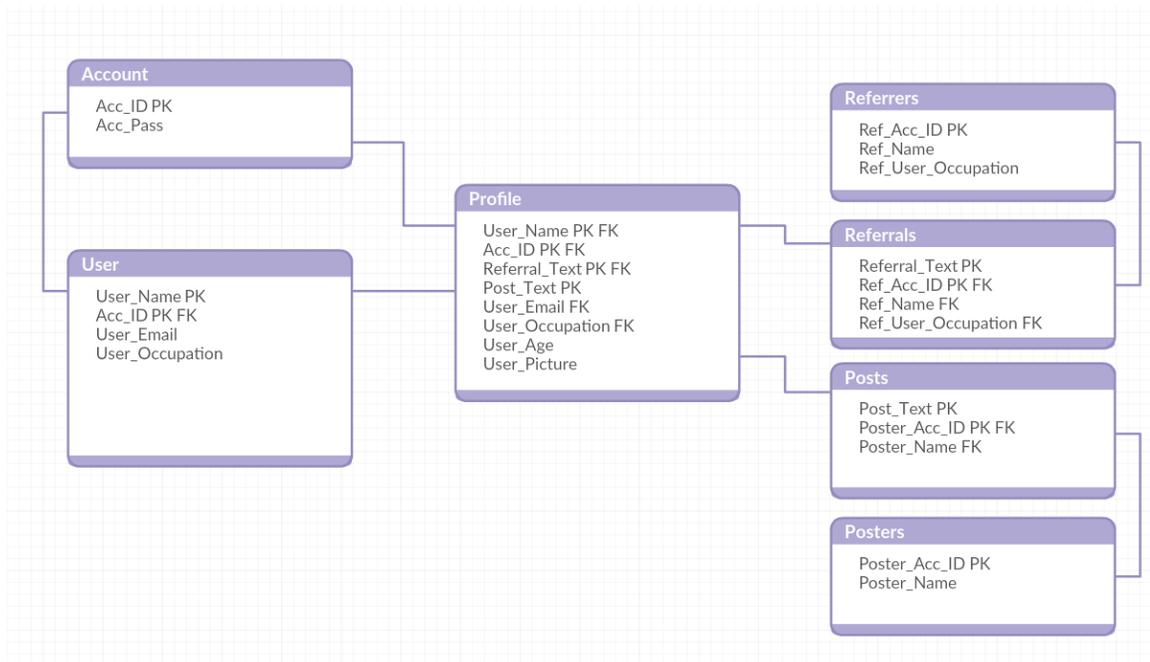
Architecture Diagram



Class Diagram



Database Diagram



3.3 Implementation

3.3.1 Technology overview

The main technologies I have used to develop the project were HTML5, CSS and AngularJS. Together the three of these technologies were able to be used to work together to provide a coherent, easily understood structured UI with added functionality and some much needed styling.

3.3.2 Technologies

HTML5:

HTML5 is largely tasked with the job of structuring the “screens” in the app. Its advantages are that it is very easy to learn and there is a plethora of resources to choose from should I get confused and need help with something. With that said, I already had a lot of experience with HTML5 before starting this project which made using it very easy for me.

CSS:

The main task of CSS is to ensure that the interface of each screen is styled well and to prevent any convolution. CSS is advantaged in the same way HTML is, given its popularity and widespread adoption, finding any resources for learning purposes is very easy. To add to that, the rules of CSS are simple and easy to master.

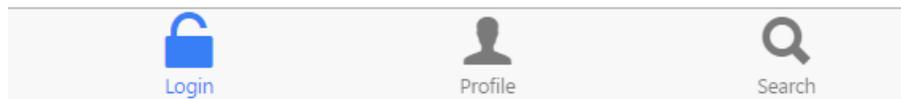
AngularJS:

AngularJS handles to main functionalities of the app and is used to bind data to the HTML. It is responsible for what the app *does*. It is a lot more technical than the previous two languages and a lot less used, but with that said it is *still* a popular language and as a result it is easy to find plenty of resources for help in trying to understand it's intricacies.

3.3.3 Procedures

Creating the project

After installing the necessary software that I needed for creating the project, I began by generating a template app using several Ionic commands in the command prompt to build one of Ionic's presets. After which I used additional Ionic commands for the project to "build" in relation to android's libraries. Unfortunately I could not choose to build for iOS as well as that requires a Mac computer which I do not have access to. In my case I chose to use the "tabs" preset since I would be using tabs inside my project. A "tab" in this instance is represented by an icon located at the bottom of the screen and once clicked, the icon will take you to a different page. At the bottom of my app's screen I have 3 tabs in total, all of which are line together horizontally. They are used to navigate between the login, profile and search pages.



All pages have their tabs generated in the tabs.html file, with one exception, the registration page. Since in theory, most users would only use the registration page once, I have the attribute associated with displaying the register page's tab set to hidden

```
12 <!-- Registration Tab -->
13 <ion-tab title="Register" hidden="true" href="#/tab/register">
14   <ion-nav-view name="register"></ion-nav-view>
15 </ion-tab>
```

Once I finished that, I began writing the HTML and CSS of my pages. Finally, I then began work with AngularJS to incorporate the core functionalities of my project.

Registration

To use the app, I needed to somehow distinguish between different user profiles, to do this, I used Firebase and AngularFire to help me identify who the user was by creating a means of authenticating users and registering them to the database.

This required me to set up an authentication system on my firebase account and to set the security rules required to allow access to the service. I was able to successfully create a rule which allowed users to write to my “users” node on the database thereby allowing them access to create an account.

```
2  |   "rules": {
3  |     "users": {
4  |       ".write": true,
5  |       "$uid": {
6  |         //allow user write if they meet the following criteria
7  |         ".write": "auth != null && auth.uid == $uid",
8  |         ".read": true

```

Next I needed to create a register controller which was directly linked to my Firebase account. To do this I created a variable in the app.js file which could be used by the app to directly communicate with my Firebase account.

(Note: I will not put the code for that in this document as I'd rather not have others use my account links. The code can however be seen on line 25 of my app.js file by those authorized to look at my submissions).

Once I was able to communicate with my Firebase database through the app I began to implement the code needed to save the data in my registration form to the database.

```

22 .controller('RegisterCtrl', function($scope, $firebaseAuth, $state, $stateParams) {
23     $scope.register = function() {
24         var fbAuth = $firebaseAuth(fb);
25         with ($scope) {
26             if (password != confirmPassword)
27                 throw "Passwords don't match"
28             fbAuth.$createUser({email: email, password: password}).then(function() {
29                 return fbAuth.$authWithPassword({
30                     email: email,
31                     password: password
32                 });
33             }).then(function(authData) {
34
35                 var user = fb.child("users").child(authData.uid);
36                 user.child("Email").set(email);
37                 user.child("First Name").set(firstName);
38                 user.child("Second Name").set(secondName);
39                 fb.child("referrals").child(authData.uid);
40                 //var referrals = fb.child("referrals").child(authData.uid);
41                 //referrals.child("Anything").set("words");
42
43                 $state.go("tab.profile", { uid: authData.uid });
44             }).catch(function(error) {
45                 console.error("ERROR " + error);
46             });
47         }
48     }
49 }
50 $scope.email = $stateParams['mail']

```

Login

The login screen works by querying the database for a user profile that meets the criteria the user has entered into the email and password input fields. The page uses an ng-click directive on the login button to execute the following code:

```

5     $scope.login = function(email, password) {
6         var fbAuth = $firebaseAuth(fb);
7         fbAuth.$authWithPassword({
8             email: email,
9             password: password
10        }).then(function(authData) {
11            $state.go("tab.profile", { uid: authData.uid })
12        }).catch(function(error) {
13            console.error("ERROR: " + error);
14        });
15    }

```

Which will use the variable I mentioned earlier to query the database for the user that matches that exact criteria (if any) that user entered into the email and password input fields on the page and to take the user to the profile tab automatically if a match is found and they are authenticated.

The register button simply functions as a means of redirecting the user to the registration page so that they may create an account.

Profile

The profile page uses AngularJS and Firebase to find the authorization ID of the user who owns the page, so that it can populate the elements on the page with relevant data such as user's name and their referrals.

```
var user = $firebaseObject(fb.child("users").child($scope.uid));
user.$loaded().then(function() {
    $scope.fullName = user["First Name"] + " " + user["Second Name"];
    $ionicLoading.hide();
    $scope.profileClass = "";
})
```

```
$scope.refer = function() {
    if (fb.getAuth() == null)
        return;

    var referral = fb.child("referrals").child($scope.uid).child(fb.getAuth().uid);
    referral.child("body").set(referralText);
```

Search

The search screen provides an input field for users to type in the name of the user whose profile they want to visit. Using the data entered into the input field, AngularJS queries the database for users whose name fields are close to or match that criteria. Any users brought from the database are shown below the search box displaying their names so that the user can determine which one they wish to visit. Upon clicking one of the results, the user will be directed to that user's profile.

3.4 Testing

3.4.1 Unit Testing

While developing the app I performed unit testing on the various components of the app and their interactions with each other. Some examples of this would be, making sure that users who registered, had their data sent to the correct node(s) or that the native code of the app actually *authenticated* the user after logging in or registering. In many cases I simply used a simple console log command to test if data was being sent/received/used as intended. In some cases I had to perform more elaborate testing by using the developer tools of the browser I was using to monitor the network activity to troubleshoot some problems.

Rather than constantly re-uploading the app whenever I made an update and sending it to my Ionic account and downloading it onto a mobile device repeatedly I simply used the ionic serve command inside the command prompt of my computer to provide an accurate emulation of the app in my different browsers (Chrome and Firefox) instead. The emulation is also updated in real-time, meaning any changes made to the application after it has been “served” are immediately seen on the emulation without having to reload which proved very useful.

3.4.2 Usability Testing

Usability testing was performed by asking some volunteers at random to test out the application and see if they were able to navigate through the app and understand what they were presented with. The feedback I received was generally positive and I was even given a few helpful tips in some cases and made aware of potential issues I wasn't previously aware of.

3.4.3 Security Testing

I checked to see if a user was able to access the service without authorizing themselves through the login process first. This is impossible and access to the profile page is denied when an unauthorised user attempts to access it.

3.5 Graphical User Interface (GUI) Layout

The following pictures displayed are simply mock ups of what my GUIs are intended to look like.

Mockup of the TapReference login page. The page features a white background with a light gray header containing the word "Login". Below the header is the TapReference logo, which consists of the text "TapReference" in blue with a blue circular graphic element. Underneath the logo is the text "Welcome to TapReference!" followed by "Please login or register your account." There are two input fields: "Email" and "Password". Below these fields are two buttons: a blue "Log in" button and a green "Register" button. At the bottom of the page is a navigation bar with three icons: a blue padlock for "Login", a gray person icon for "Profile", and a gray magnifying glass for "Search".

The login page is the first page the user will see when they load the application on their smartphone device. To login, the user must first create an account with the service. The user is able to enter this process by simply clicking the "create account" button which will be located nearest to their thumb. This will take them to the registration page of the app which provides the user with the relevant forms and required for account creation. Once this process is complete the user will be able to log into their new account by

entering their email followed by their password into their respective fields in the login page followed by pressing the login button. After this is done the user will then be taken to their profile page.

Register

TapReference

Email Address

First Name

Second Name

Password

Confirm password

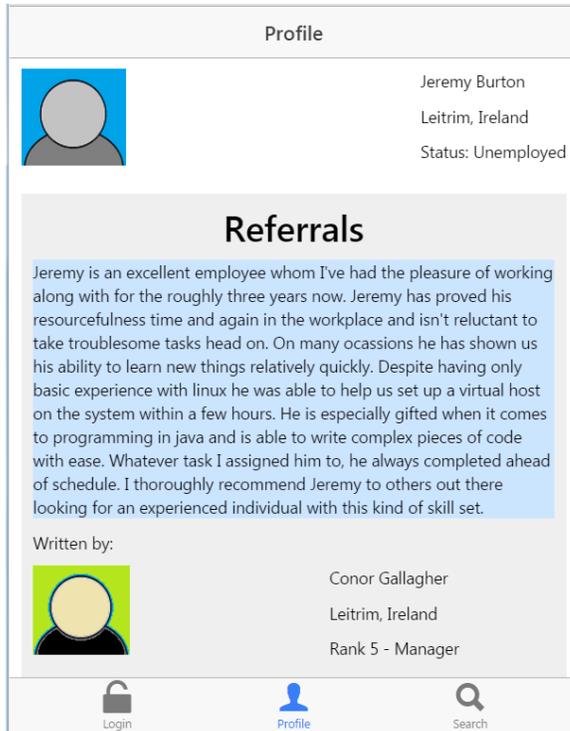
Register

Login Profile Search

The register page's main function is to receive information from the user in the form of their email address, password and desired account name followed by their occupation. The occupation dropdown menu will not specify a list of every possible job that exists as creating such a list and ranking them all individually would be too unrealistic, instead this process will be streamlined by asking for something more abstract that many types of job can fit into (i.e. "stock manager" would simply be categorized as "manager"). The

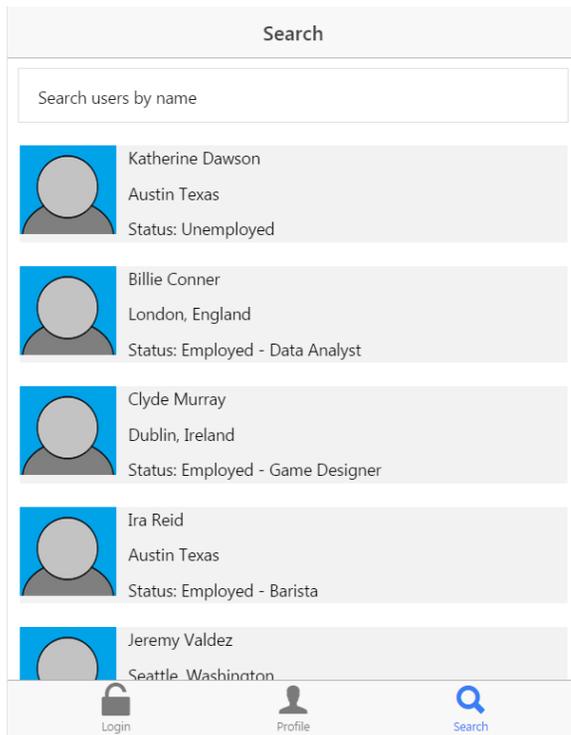
information gained from this will be used to determine user's current rank value.

Once all fields have been filled out the user will be able to click the register button. This information will then be sent and stored to the database and be used to generate the user's new profile page.



The profile page will feature information relating to the user and will display the referrals which they have received and accepted from other users. Any referrals shown will also feature the user(s) who wrote them and will display that user's rank to give people an idea of what position this person holds and just how relevant that user's referral is in terms of identifying the profile owner's skillset. For example, a user whose rank is in close proximity may have a more comprehensive understanding of the

profile owner's skills and would be able to describe them with greater accuracy. A different user who may have a much higher rank, would not be able to give as much of an in depth description of the profile owner's skills but may instead be able to provide alternate information such as what kind of impact that user had at the company they were working for, how they were able to handle assignments given to them and what their communication skills were like.



The search page can be accessed by the user through pressing a link on their profile page. The search page will initially appear blank with a single text field at the top. Once text is entered into this field this information will be cross referenced with all user name data in the database until a name or set of names that closely resembles the text entered into the search field are found and displayed beneath the search field. If there are no matches to the text entered, then the area below the search field will remain blank and

will display a message to say that no results were found. In the event that a user finds the person they are looking for, they will be able to visit this person's profile simply by clicking on them in the list of results.

4 Conclusions

In closing, I believe this project has been a very important experience that has helped me learn how to understand and use many new technologies which in some cases I had no prior experience in using before. Overall I am content with how the project has turned out and look forward to showcasing it at a later date.

There is one issue that has been on my mind since the project started however, and that is, why is it that the modules most relevant to our specialisations are present in the second semester rather than the first semester of 4th year? The Multimedia and Mobile Application Development module could have proved extremely useful and very likely would have provided me with a better informed approach to laying the foundation for my project at the beginning. However since this module was not present in the first semester, I could not avail of the information that it would provide me with later. I know I'm far from the only student who feels this way and believe that it should be addressed.

5 Appendices

5.1 *Project Proposal*

5.1.1 Objectives

- Create a mobile app that allows users to create accounts and sign in.
- Provide users with access to modify their own profile.
- Allow users to visit each other's profiles and view data.
- Assign ranks (likely during account creation) to user accounts as a means of approximating a user's position in a company.
- Provide users with the ability to recommend/refer other users.
- Display referrals that a user has received on their profile.
- Create a database that stores user accounts and their information.
- Allow users to post comments to each other.
- Allow users to hide their identity when recommending other users and display their rank value instead.
- Create a smoke test for the app.
- Refine interfaces and remove bugs as needed.

5.1.2 Technical Approach

Research and Requirements:

To achieve my desired goals I will need to research Ionic framework and how to develop robust complex applications when using it in conjunction with Firebase. Based on what I have already researched on it, there appear to be a plentiful amount of quality resources for learning how to use it online, including a three hour training course on Pluralsight.

I will need to research what how to use Firebase to send and receive values from an Ionic app to allow users to access the core functions of the service when connected to the internet.

Thanks to a group project during my second year in NCI I already have some experience with creating a login and account system with java which may at the

very least assist me in having a better understanding of what I should take into consideration when creating these systems. Unfortunately however that project did not avail of a database and some research and analysis will still be required as a result.

Implementation:

1. Using Ionic, AngularJS and Firebase, create a functional account creation system with data sent to and stored inside a database.
2. Assign a rank to the user, based on the pre-set job position they select during account creation
3. Create a sign in system which will compare values in the required fields to account name and password data in the database.
4. Create a search functionality that allows users to visit other profiles.
5. Allow users to add referrals to each other.
6. Add option to allow users to remain anonymous but instead display their rank value when submitting a referral.
7. Display received referrals on the user's profile.
8. Create a means of sending and receiving comments to or from other users.

5.1.3 Special resources required

- Pluralsight for learning purposes.
- A text editor to code in (e.g. Sublime Text 2 or Atom).
- Internet access for research purposes and for sending and receiving data from the database.
- A phone, tablet or emulator running Android to test the app on different devices.

5.1.4 Technical Details

Implementation languages:

HTML

CSS

SASS

JSON

AngularJS

Principal libraries:

Android API

Ionic

Services:

Firebase cloud services

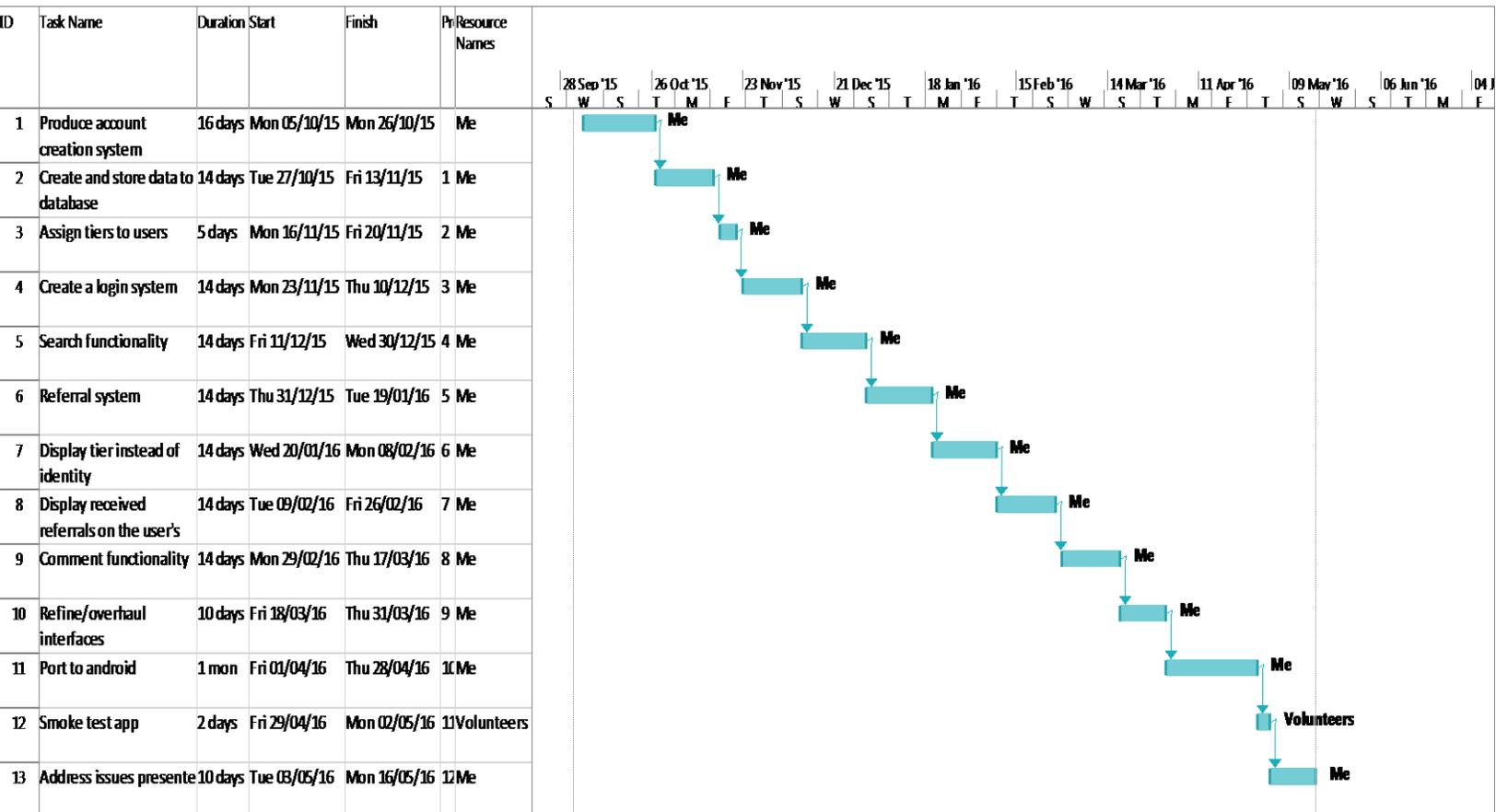
5.1.5 Evaluation

I will begin evaluating the system by creating data and testing whether or not it has successfully been stored to the database and can be called to again by code that requires that particular data. I will then test as to whether or not other users can then call to and receive that data successfully.

I will also carry out a smoke test whereby two or more users will use the app without prior knowledge of it and attempt to use all functionalities that are available. I will likely create a set of very brief goals (e.g. find user John Smith's profile or write a referral for John Smith) for my volunteers to use to know what functionalities there are and to see if they could realistically use them without need for assistance.

The volunteers that I use will likely be selected in terms of how familiar they are with using smart phone technology, for example, I would select two volunteers that are familiar with smart phones to use the app to get feedback from one viewpoint. After that, I would select an additional two volunteers who are not familiar with using smart phones to see how easy it is for them to navigate and carry out the goals I have given to them.

5.2 Project Plan



5.3 Monthly Journals

Student name: Fiachra Devitt

Programme (e.g., BSc in Computing):

Month: September

My Achievements

This month I mainly spent my time deciding on what my project was going to be, identifying its requirements and researching the best approaches I could take to creating my idea.

I started by voicing my ideas to others until eventually there was one that most of the people I asked either liked or thought could be useful in their daily lives. I decided to stick with the idea of an App that could facilitate employees and employers to give each other referrals to help people with whom they have enjoyed working with in progressing their career.

Given that mobile technologies is a part of the specialisation I selected for my fourth year in NCI, I decided to make my project in the form of an App for use on a range of devices running android. This will require me to learn some new technologies such as Android Studio (an IDE which caters to android Apps development).

This month I also scheduled myself some time in advance to go to the library and finish my project proposal which I did and submitted it on time as intended.

My Reflection

I will likely have to spend a considerable time in the coming weeks, studying and familiarising myself with the new applications and technologies that I will be using to create the app. But I hope to at least make some progress in creating the account creation system either this week or next. Once I finish the account creation I will likely try to create and set up a database to store information collected.

Another task I must complete for the following month is finding a supervisor for my project who is familiar with the technologies that I will be using so that they can give me some much needed feedback for my work. I will send out some emails this week and next to try and get in contact with some NCI staff that may be able to help me.

Student name: Fiachra Devitt

Programme (e.g., BSc in Computing):

Month: October

My Achievements

This month I organised a meeting with my supervisor where I discussed with him the idea behind my project, what I aimed to do and how I was going to do it. He gave me advice in regards to possible alternative ways I could go about implementing what I wanted in case there may be easier ways of doing such.

I started into and finished my requirements specification which took a considerable amount of time and research to complete along with creating the use case diagrams that were included inside it.

This week I gave PhoneGap a try, however I have not used it extensively enough to decide whether or not I would like to use it for my project yet.

My Reflection

I will likely modify my requirement specification over time as I make modifications to the project itself, it was advised to us in our software project class that when we do this we should add the dates of modification in a designated section of the file. I will keep this in mind when doing so.

I would have liked to have had some time set aside to get started with my project over the reading week at the end of the month, however the sheer amount of assignments and the amount of work and research involved with each of them greatly hindered this. I hope that since I have most of them out of the way now that I will be able to spend more time on my project this month without more distractions getting in my way.

I suppose that since I now that have my requirements specification out of the way I can at least refer to that to help me structure the project and keep some of my core ideas and their implementations in mind.

Supervisor Meetings

Date of Meeting: 30/10/2015

Items discussed: We discussed my project as a whole and what features I would like to have in it and what I would require to create to create the feature. It was also recommended to me that I look into PhoneGap, an open source framework used for creating apps for smartphones using HTML5, JavaScript and CSS, all of which I am familiar with which is convenient. Admittedly this was not the first time PhoneGap had been mentioned to me and I had already knew about its existence but not exactly what it did.

Student name: Fiachra Devitt

Programme (e.g., BSc in Computing): BSHCNMT4

Month: November

My Achievements

This month I was extremely busy with my other modules, I'd go as far as to say overburdened by too many assignments occurring around or even having exactly the same deadlines. Aside from that, I looked into the Analysis & Design Specification that needed to be submitted for the end of the month only to find a severe lack of information regarding it. Apparently I was not the only one who felt this way as when I asked other students about it they seemed equally clueless as to what to do. For reference, the Requirement Specification that we submitted as last month's assignment had seven materials on Moodle in the form of Microsoft Word documents, a PowerPoint presentation, a PDF, some links to online sources etc. whereas this Analysis & Design Specification had a meagre two materials covering it, which included a presentation and a template Word document which was very non-descript in what it actually wanted in the particular sections provided. Because of all of this I decided to ask my Supervisor during our meeting about it, he seemed more concerned about my Requirement Specification because he wanted me to get the Use Case diagrams done since despite that I was able to get

the Use Cases themselves done for the upload date, I was unable to find the time in my schedule then for their diagrams. I told him that I would get them done before I tackled the Analysis and Design Specification. After that my supervisor gave me some tips on what I should consider doing when tackling the Database Design section which was very helpful. Once I had finished my Use Case diagrams I added them to my section in the Requirement Specification and added the entire section to the Analysis and Design Specification because it asked for them as well for some reason. I asked around the class for what the other students were doing for the other sections of the document outside the Use Cases and the Database Design however, most replied that they had no idea what to do because there was too little information to go by. Although I'm not sure where this information came from but several students decided that the following should be added to the document: A collaboration diagram, for each Use Case, a Class diagram for each Use Case, a path diagram for each Use Case, a State chart diagram for each Use case and a four stage DFD diagram.

I am unsure as to where this idea came from because nowhere in the Analysis and Design Specification that we were given were these things mentioned at all so I am unsure as to whether or not I should be including them myself as it sounds like a very large portion of work to get done in such a small timeframe. I suppose it depends on how detailed one intends to make each diagram however this would mean that at absolute minimum 21 additional diagrams were required but unmentioned in the Analysis and Design Specification which is mind boggling.

My Reflection

I was able to do a Database design diagram thanks to the help given to me by my supervisor. I also got the Use Case diagrams done as requested.

In regards to the Monday lectures for the software project module, I feel that there has been a severe lack of communication regarding this Analysis and Design Specification and would like this to be addressed in the future. The only information that I seemed to get from the Monday lectures was "do whatever". I find it very

strange that this has been the case considering that there were so few problems regarding the Requirement Specification which came the month before. I feel as though students such as myself are very much in the dark regarding this assignment and would appreciate it if this content could be addressed in more detail in the future.

Furthermore I believe that a revision session regarding the different types of diagrams would have been greatly appreciated by students as it has been a long time since we last did them. I know for a fact that even the students who I know are good at programming had a tough time understanding them back when we were doing them two years ago in our software engineering module. While there are indeed a fair amount of materials explaining them online, it is somewhat difficult to distinguish between good sources and bad considering that many sources seem to approach them differently.

Supervisor Meetings

Date of Meeting: 27/11/2015

Items discussed:

1. Analysis and Design Specification
2. Use Case diagrams
3. Database Design diagram
4. The assignment's actual value in terms of marks for the module

Student name: Fiachra Devitt

Programme (e.g., BSc in Computing):

Month: December

My Achievements

This month I began coding my app and was eventually able to get it hosted on my computer using Ionic, an open source tool used for creating mobile apps for both android and iOS. In addition I was also able to get the current state of my application working on an emulated Android Nexus 7 running on my computer although the software used for emulating the devices takes a very long time to start emulating.

My Reflection

Setting up Ionic was initially easy, however I did run into issues with the technologies which I intended to use alongside it. Using a combination of Ionic's and Chrome's Android emulation features, I intended to test out how my app would look on an Android device. Unfortunately the guides on the official Ionic website were lacking in detail along with several steps it would seem. This in turn had caused my Android SDK installation to go awry. Although my SDK was installed correctly, the guide failed to explain what packages from the Android SDK Manager were needed to run the service, nor did it explain how to create and configure an AVD or "Android Virtual Device". Through much time and effort I was able to fix these issues myself after searching and querying several forums online. Once I was able to get it working I finally got the program to start emulating the Nexus 7 AVD I created. The process of creating the emulated device took several hours regardless of the fact that the computer I am running it from at home would be considered high spec. Given that during the emulation process, my computer's CPU usage was only 10% I thought that something was wrong. When I looked it up online however, I found out that this was actually considered normal and was informed that creating a "snapshot" of the device would be more optimal as it will load significantly quicker that way.

Intended Changes

For the rest of this month I will be working on the app's GUI elements and likely a login system.

Student name: Fiachra Devitt

Programme (e.g., BSc in Computing): BSHCNMT4

Month: January

My Achievements

This month I was successfully able resolve several of the issues I was having with Ionic. I am now able to create my own web pages which display correctly however, I have noticed that AngularJS's MVC can be extremely volatile, often times removing all elements present on a given page over the smallest of errors. This makes developing pages of higher complexity such as the login and registration pages a headache to work with. I originally performed a database test using WAMP server on my own computer to test basic database functionalities which I was able to get working relatively easily however, for my project I needed my back end to be hosted online with constant uptime which my computer would be unable to do and as a result of this, I decided to move my work to Firebase. At this current moment in time it would appear that I have been able to get Firebase to save form data that I pass to it through my Ionic app although I feel that more testing is necessary.

Intended Changes

Next month I will further develop the registration section of the app and begin working on either the search or user profile pages depending on my progress with Firebase. While their GUIs should be relatively easy to do, I anticipate their functions being more difficult to carry out. Though with that said, I feel that the search page should be easier for me to work with first since I am already working with the values that it will be utilising.

Supervisor Meetings

Date of Meeting: 1/02/2016

Items discussed: During my meeting with my supervisor we discussed the mid-point presentation and what I should do to better prepare myself for it. My supervisor also helped clarify each section of the mid-point technical report to me and also informed me of a referencing service called Zotero which I could use to make referencing sections of the technical report faster and easier.

Action Items: Mid-point technical report

Student name: Fiachra Devitt

Programme (e.g., BSc in Computing): BSHCNMT4

Month: February

My Achievements

This month I completed my mid-point presentation which involved creating some slides in Microsoft PowerPoint along with creating a rough script of what to talk about during the limited time for which I would be presenting for. I felt that I handled the presentation fairly well and was overall happy with how it went.

I created GUIs for the Login, Profile and Search pages for my app over this month using a combination HTML, CSS and a bit of AngularJS. The GUIs are more or less how I want them to look in the final version although I had to use some placeholder elements for functions that weren't currently implemented for the purpose of displaying something more coherent for the mid-point presentation.

Intended Changes

Next month I aim to work on the issues I am currently running in to regarding Ionic and its AngularJS controllers. While I seem to have modified my functions to adhere to the new way in which Ionic handles its controllers, I am currently having a problem whereby Ionic doesn't seem to read one particular section of code.

Supervisor Meetings

Date of Meeting: 29/02/2016

Items discussed: I informed my supervisor that while I was successfully able to use Zotero to acquire information relating to referencing from the websites I had been accessing, I was unable to use its Microsoft Word plugin to apply all of the referencing info that had been acquired. Given that the Harvard standard was specifically required for referencing in our final project specifications, this posed a problem that needed to be addressed. Luckily my supervisor was able to identify the problem that I was running into and showed me how to resolve it and I can confirm that it works perfectly now. We also talked about what would be expected of my project during the final year presentation and how marks would be distributed amongst the different requirements. After that we discussed how to further improve my future presentations.

Action Items: Project final year report

Student name: Fiachra Devitt

Programme (e.g., BSc in Computing): BSHCNMT4

Month: March

My Achievements

This month in particular has been fairly unproductive for my software project due to the amount of complex assignments I've had to deal with simultaneously during this period. As a result, I have been unable to find much time to work on my project although with that said, I was able to find the time to create and finalize my project poster for the presentations in May.

Intended Changes

Next month I will be studying for and finishing my exams. After that period is over I will have around 3 weeks to implement the rest of the required functions which

my project needs followed by performing tests on it. Once this is finished I will update the final report accordingly.

Supervisor Meetings

Date of Meeting: 04/04/2016

Items discussed: During my meeting with my supervisor we discussed some of the modifications that I needed to make with my project's final year report before submitting and how I should allow him to have a look at it once I have made the necessary changes after my exams are over. We also talked about the project submission cover sheet as, despite it being mentioned during a software project lecture, I was unable to find an example or template of it on Moodle. My supervisor then showed to me where the submission sheet's template could be found on the MyNCISStudent website. After this we then discussed the details of the project poster which was needed for the project showcase in May. Although my poster was finished I didn't know how large exactly it had to be as no dimensions were given, all I knew was that A4 was too small. We eventually came to the conclusion that it was likely A3 that would be needed but that I should ask the lecturer in charge of the module for clarification before printing.

Action Items: Project final year report, Project submission sheet, Project poster

5.4 Requirement Specification

5.4.1 Introduction

5.4.1.1 Purpose

The purpose of this document is to provide a description of my project, how it is to be implemented, what its requirements are, to display my project's use cases and what its general purpose is. The purpose of the project itself is to provide its users with a means of posting recommendations to people whom they may have worked with in the past in order to assist them in furthering their career. The recommendations should go into detail about the skills that their co-workers perceived the user to possess, how said skills were applied and their overall impact.

5.4.1.2 Project Scope

The scope of the project is to create a system that allows users to create their own profiles and send referrals to their colleagues. Referrals that have been accepted become visible to other users on the site via their profile page. The job of the user who wrote the referral can also be seen. The purpose of this is to give any user potentially seeking to hire a client whom they feel is qualified for a position in their organisation, can get a grasp of the importance of a user's referral based on their position in a company. For example, someone who works as a manager may be able to provide detailed accurate input of the user in question's skillsets, how they handled tasks that were assigned to them and what kind of work they excelled at.

5.4.1.3 Definitions, Acronyms, and Abbreviations

GUI = Graphical User Interface

5.4.2 User Requirements Definition

From a user's perspective, the system is to provide them with a means of creating and signing into their personal account and profile. The profile which they create should be made viewable to other users after creation. The account creation should include a means of defining what their current status of employment is and what their relative position in the company they work for is (delivery man, Manager, CEO etc.). It should allow them to search and find other users and view their

profiles. The service should also give them the ability to submit referrals to other users. At the same time it should allow the user in question to receive referrals from others as well. In the event that a referral is received, an option should be sent to the user to allow or deny it from being posted on their profile. Once a referral has been accepted it should be posted on their profile and be viewable to other users. Lastly the systems should allow users to post comments to each other's profiles.

5.4.3 Requirements Specification

After a user has created an account, the data related to that account should be stored in a database. After account creation, an employment status should be assigned to the user. Once this is done, the user should then have the ability to sign into the account and granted access to its features. The user's account will need to be generated with a default layout upon creation. A search functionality will need to be incorporated to allow for users to visit each other's profiles. The search tool should compare what has been entered inside the search field with account names which are stored into the database. A prediction of the account or a list of accounts that have a semblance to what the user has typed into the field should be presented and be clickable by the user to visit the profile they represent. When visiting an account, a user must be able to have an option to write a referral by pressing a button. Once clicked, the button should provide the user with a text field to write in. Once they are finished, a second and third button should be located at the bottom of the text field. One button should allow the user to discard their message and return to the profile page. The other should save and store their referral to the database. The database must then send the referral to the recipient. The recipient is given the option to reject or allow the referral. Should the user reject it, the referral will be permanently deleted from the database. In the event

that the user chooses to allow it, the referral remains on the database and is displayed on the user's profile.

5.4.3.1 Requirement 1 <User registration>

5.4.3.1.1 Description & Priority

Before the full system can be used, the user must have an account to access it with. During their first visit to the site, the first thing the user must do is register.

5.4.3.1.2 Use Case

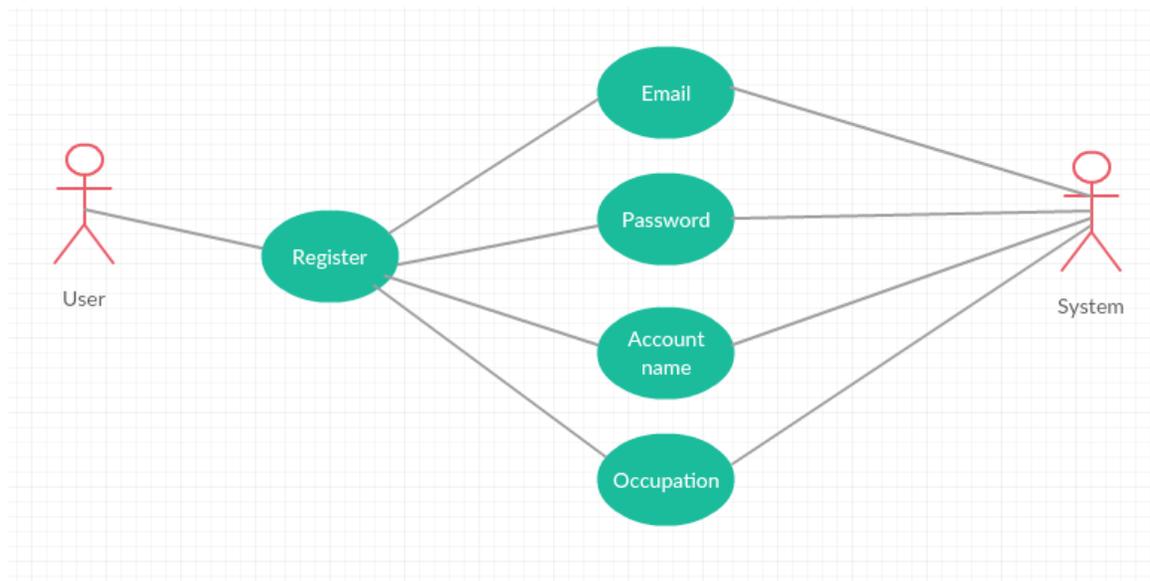
Scope

The scope of this use case is to create an account for the user to access the site

Description

This use case describes how the user registers themselves on the system

Use Case Diagram



Flow Description

Precondition

The user has already arrived at the login page

Activation

This use case starts when the user clicks the link at the bottom of the page that bring them to the registration page.

Main flow

1. The system identifies the user's input and triggers an on-click event which brings them to the registration page
2. The User enters their email (see A1)
3. The User enters their password
4. The User enters their account name
5. The User chooses their occupation
6. The User clicks the register button at the bottom of the page
7. The System saves the user's account data to the database
8. The system generates a user account with the data provided

Alternate flow

A1 : <Email verification error>

1. The system detects an invalid email
2. The system denies account creation until all required fields are correct
3. The System alerts the User of the incorrect email address.
4. The Use case continues at position 3 of the main flow

Exceptional flow

Termination

The system brings the user back to the login screen

Post condition

The system goes into a wait state

5.4.3.2 Requirement 2 <User Login>

5.4.3.2.1 Description & Priority

To use the system, the user must first sign into their account.

5.4.3.2.2 Use Case

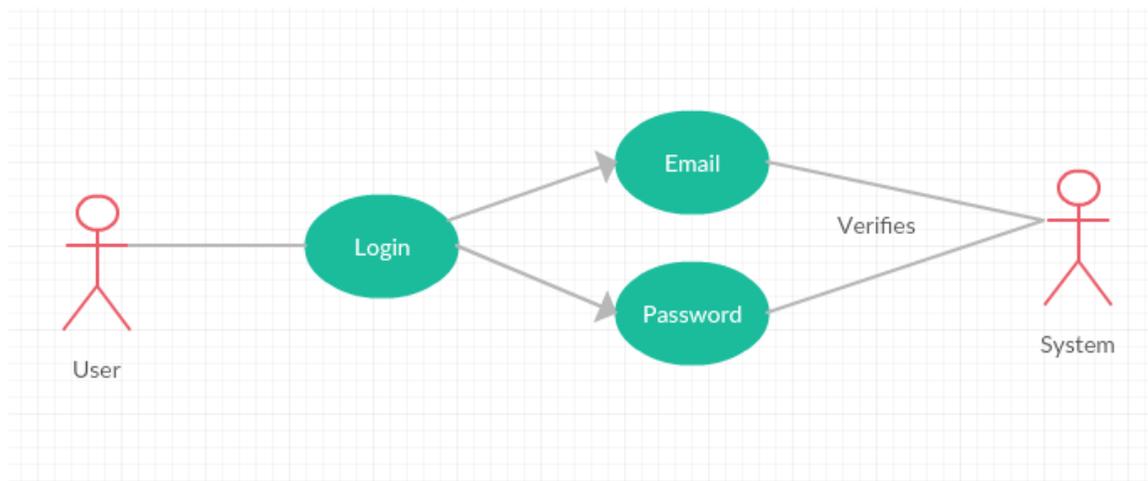
Scope

The scope of this use case is to sign the user into their account and allow them to access its functions.

Description

This use case describes the process of logging into the system

Use Case Diagram



Flow Description

Precondition

The user has already created an account

Activation

The user arrives at the login page

Main flow

1. The User enters their email
2. The User enters their password
3. The User presses the login button
4. The System compares the results in the email and password fields with those of the accounts stored in the database (see A1)

5. The System verifies that the account details match, allows them to access their account and directs the user to their profile

Alternate flow

A1 : <Incorrect password>

5. The System detects that the data entered into the password field does not match the password in the database
6. The system denies login until all required fields are correct
7. The System alerts the User of the incorrect password.
8. The Use case continues at position 2 of the main flow

Exceptional flow

E1 : <invalid email>

1. The System detects that the data entered into the email field does not match any of the emails stored in the database
2. The System alerts the user of an invalid email address
3. The use case continues at position 1 of the main flow

Termination

Post condition

The system goes into a wait state

5.4.3.3 Requirement 3 <User search>

5.4.3.3.1 Description & Priority

To give a referral to another User B, User A must first find User B's profile. This can be achieved by using the search function.

5.4.3.3.2 Use Case

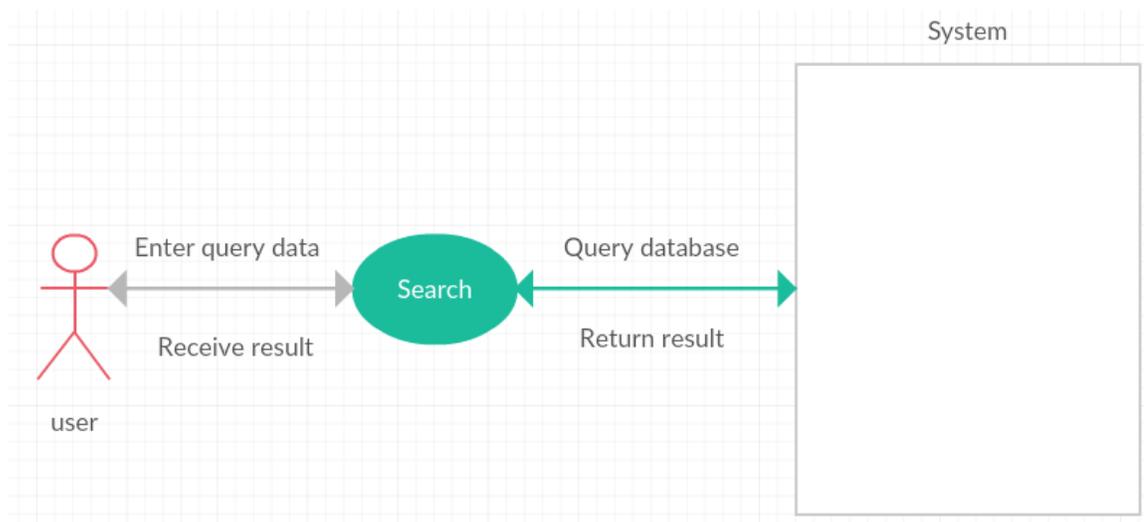
Scope

The scope of this use case is to get User A to User B's profile page

Description

This use case describes how User A is going to navigate to User B's profile page

Use Case Diagram



Flow Description

Precondition

The user has already arrived at the search page

Activation

This use case starts when the user begins typing into the search field

Main flow

1. The User types a name into the search field
2. The System compares the string in the search field with the account names stored on the database (see E1)
3. The System produces a list of accounts which match the search field's string
4. The System lists the profile picture, account name and employment status of the users in its list
5. The User clicks on the account they are looking for
6. The System directs the user the account profile of User B

Alternate flow

Exceptional flow

E1 : <Invalid User>

1. The System finds no matches to the string
2. The System notifies that the user does not exist
3. The use case ends.

Termination**Post condition**

The system goes into a wait state

5.4.3.4 Requirement 4 <User Submits Referral>**5.4.3.4.1 Description & Priority**

User A wants to provide User B with a referral. To do this, the user must create and send their referral

5.4.3.4.2 Use Case**Scope**

The scope of this use case is to allow User A to create and send a referral to User B

Description

This use case describes how User A is going to create and submit a referral to user B

Use Case Diagram



Flow Description

Precondition

User A is already on User B's profile.

Activation

This use case starts when User A presses the "add referral" button on User B's profile.

Main flow

1. The System presents User A with a text field (see E1)
2. User A types their referral
3. User A presses the "submit" button
4. The System saves the referral to the database
5. The referral text field and buttons are dismissed, the user is returned to the profile of user B

Alternate flow

Exceptional flow

E1 : <Cancel referral>

1. The user presses the "cancel" button
2. The system dismisses the referral text field and buttons, the user is returned to the profile of user B
3. Use case ends

Termination

Post condition

The system goes into a wait state

5.4.3.5 Requirement 5 <Accepting/Denying a referral>

5.4.3.5.1 Description & Priority

For user A's referral to user B to appear on User B's profile, User B must first accept or deny the referral.

5.4.3.5.2 Use Case

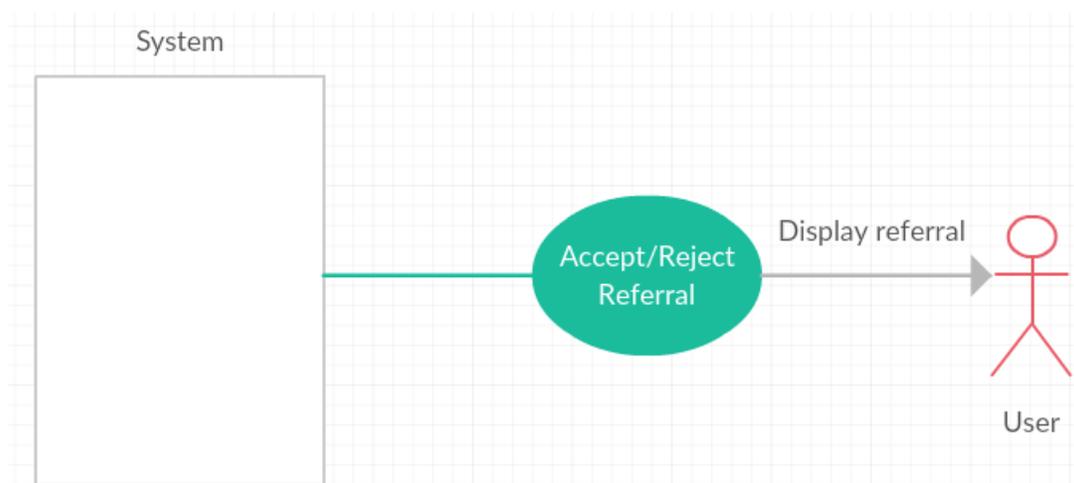
Scope

The scope of this use case is to show what actions User B can take with the referral he/she has received.

Description

This use case describes how User B accepts or denies the referral User A has sent

Use Case Diagram



Flow Description

Precondition

User B has received a referral from User A

Activation

This use case starts when the system presents User B with the referral that is to be posted onto his/her profile

Main flow

1. User B views the referral User A has written (see E1)
2. User B presses the “accept” button for the referral
3. The System posts the referral to User B’s profile such that other Users can view it upon visit

Alternate flow

Exceptional flow

A1 : <Deny referral>

- 1 User B presses the “deny” button
- 2 The System expunges all data of User A’s referral of user B from the database
- 3 The use case ends.

Termination

Post condition

The system goes into a wait state

5.4.3.6 Recover requirement

Backups will be made whenever the database has been updated to ensure that in the event of a failure its data can still be recovered.

5.4.3.7 Robustness requirement

If an invalid input has been received from the user, (e.g. an incorrect password or account name) the system will continue normally and alert the user of the error and help them correct it.

5.4.3.8 Security requirement

User accounts will be password protected to prevent unauthorized access.

5.4.3.9 Reusability requirement

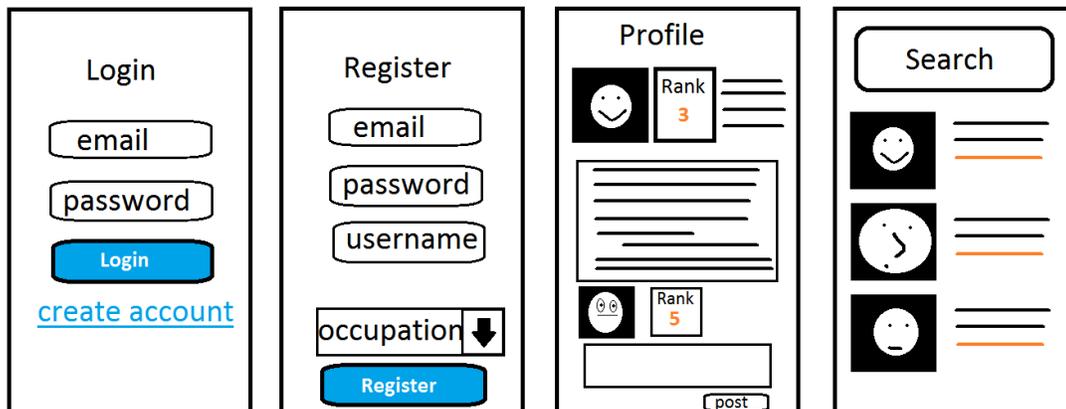
Once a user has logged back in after logging out, their account and profile data will remain as it will be saved to the database.

5.4.3.10 Interface requirements

The login page's GUI requires the user to enter in text, a valid user name and password in the appropriate fields. In the event that the user is new to the site, they may choose to click a link that takes them to the registration page. In the registration page the user can enter their desired name email and password in their relevant fields followed by their occupation (or an occupation that is relatively close to) their current occupation. Once this is done they may click a button to bring them to their generated profile. Inside the profile, the user can view the referrals they have received from other users and are also able to search for other users profiles.

5.4.3.11 GUI

Include mock-ups of the key pages or stages of the system. Explain how they are linked. Explain how you addressed above requirements in the design. It is important that the mock-ups are in line with the functional requirements above, e.g., if one of your requirements is "user registration" then one of the screens listed in this section should show a registration page.



The above GUI mock-ups represent how the Login, Registration, User Profile and Search pages should relatively look in the finished product.

5.4.4 System Evolution

It is possible that in a future iteration of the system that a browser-based client could be created to allow Windows, Macintosh and Linux users to view and use the system as well. Furthermore the profile system would be changed to include additional content such as links to personal projects that would not have been seen in the work environment. The system may also avail of a friend system, allowing users to add each other to their friends list and live chat. It would also be useful to provide a means for users to message each other privately, which could perhaps be done by implementing an inbox for messages to be received.

Furthermore quality life improvements such as providing the user an option to remember their credentials for the next time they visit so that they don't have to repeatedly sign in.

6 Bibliography

Drifty. (n.d.). Ionic Documentation Overview - Ionic Framework. Retrieved February 3, 2016, from <http://ionicframework.com/docs/overview/>

N/A. (n.d.). Features - Firebase. Retrieved May 9, 2016, from <https://www.firebase.com/features.html>

N/A. (n.d.-a). AngularFire Development Guide - Firebase. Retrieved May 8, 2016, from <https://www.firebase.com/docs/web/libraries/angular/guide/>

N/A. (n.d.-b). AngularJS: Developer Guide: Introduction. Retrieved May 9, 2016, from <https://docs.angularjs.org/guide/introduction>