Final Project Report

Wildfire Location & Information App

29th April 2021

Name | Jack McNally

Student No | C00228758

Supervisor | Chris Meudec

Abstract

The purpose of this project is to create a mobile application for the General public and semiprofessional and professional firefighters. It should help people who fight fires as well as protect people who don’t fight them. The application should be used as a tool to report fires, search for fires and notify people of nearby fires. It also should be able to provide the general public and the firefighters with useful information about fires.

Table of Contents

Table of Figures

1. Introduction

This document describes the final report of the Wildfire project based on providing a quick and easy way to Report fires, stay safe and access important information. This report will cover creating the project from beginning to end and will also contain a personal reflection.

The first section of this document will describe the project itself, it will deep dive into all of the front-end and backend technologies that were used to construct this application. Each of these subsections will all contain a small introduction, their functionality, how they helped this project and the successes and challenges I faced when using them.

The next section will discuss the issues that I encountered while constructing this application. As I was a novice user of xamarin and nearly all of the technologies I used were new to me, there were quite a few obstacles to overcome before I fully understood the tools. Each subsection will contain a small introduction about the error I encountered and also the method in which I went about fixing it.

The following section will describe how I adhered to the original specification and design. This will go into detail how I followed the specification and what I changed while undertaking the project.

Finally the last section will discuss my personal learning outcomes of the project and a final project review. This section is important as it will describe how I felt the project went as well as the things I could have done better. There will also be a subsection that discusses the possible future development of this application and what direction I would like this project to go in.

2. Project Description

2.1. Wildfire Application

The wildfire mobile application is the main section of the project. The development of the wildfire application was aided strongly by the documentation that was constructed at the early stages of the project.

The Wildfire application was developed using Xamarin forms, which uses C# (C-Sharp) and XAML(Extensible Application Markup Language). The idea of developing using Xamarin forms means that the application can be developed for both Android and iOS while working off one shared code base. The wildfire application is fully developed on Android but due to limitations the iOS development could not be developed, but would be a key part of the future development of the application.

Xamarin forms was used mainly for the front-end user interface and the map functionality. The NuGet packages that were offered by Xamarin made the implementation of the map much easier. The use of the NuGet packages made it easier to implement pieces of already developed code. Some of the NuGet packages that I used through-out the implementation of the wildfire application were Newtonsoft.json, Xamarin.forms.GoogleMaps and

Xam.Media.Plugin. The use of these packages is much easier and more efficient as that code is only used when it is needed and no code is repeated.

The backend section of the Wildfire application was made using Firebase by Google. The use of firebase which is a Backend as a Service(BaaS) made the backend implementation quick. Firebase offers many different services that helped the wildfire application come to life, these services included Firebase Authentication, Firebase Storage and Firebase Database. All of these services were used together to create a reliable and safe backend that is constantly being updated and maintained by Google.

Lastly, the Google Maps API was a key technology used by the wildfire application. The google maps API provided the map feature for the application. This API also provided the places API which contains a list of places that aided in the autocomplete for the search function. The google maps API is a simple and effective API that is cheap in terms of price and reliable as like firebase it is constantly being updated and maintained by Google.

2.2. Application Screens

Below is a very high level view of the application screens, this will demonstrate the basic process of using the application. This section is explained in much more detail in the technical and design manuals.

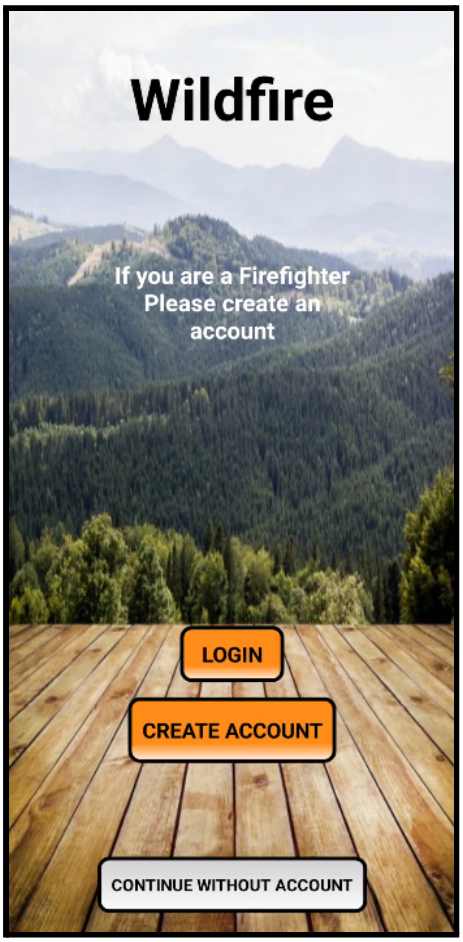


Figure 1 - First Screen

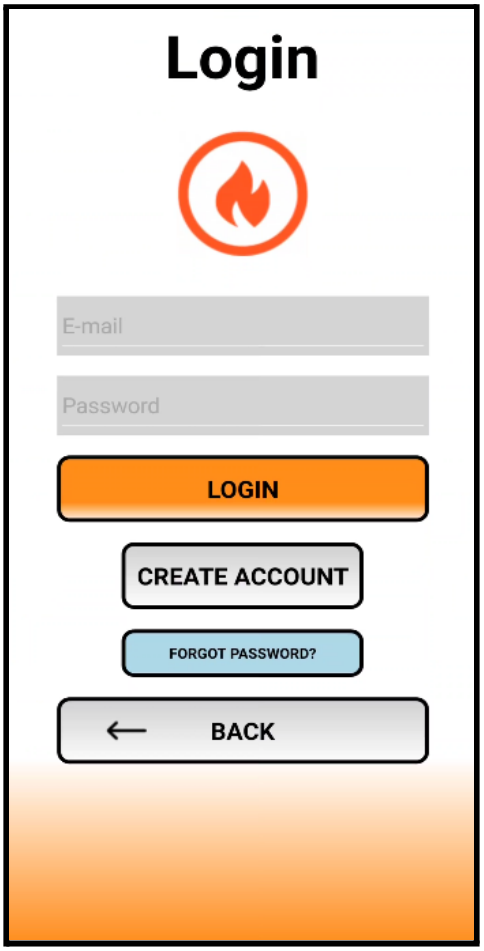


Figure 2 - Login Screen

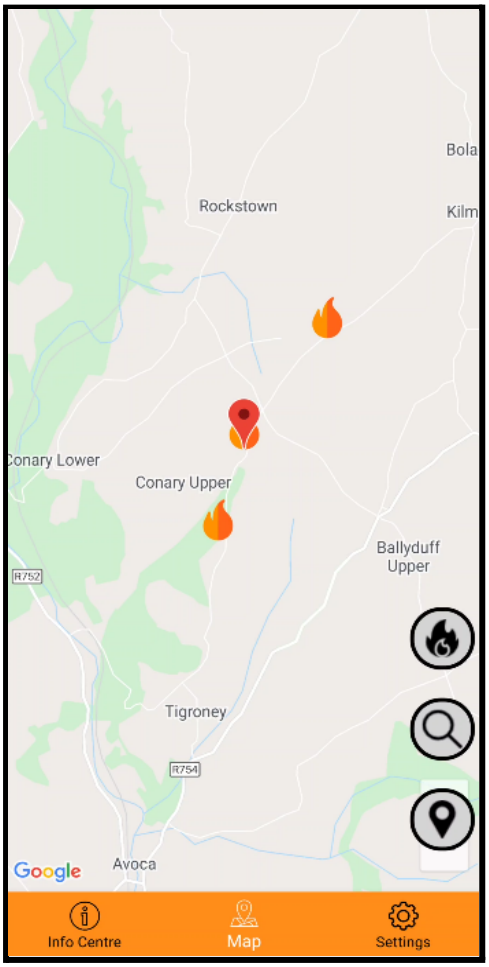


Figure 3 - View Map.

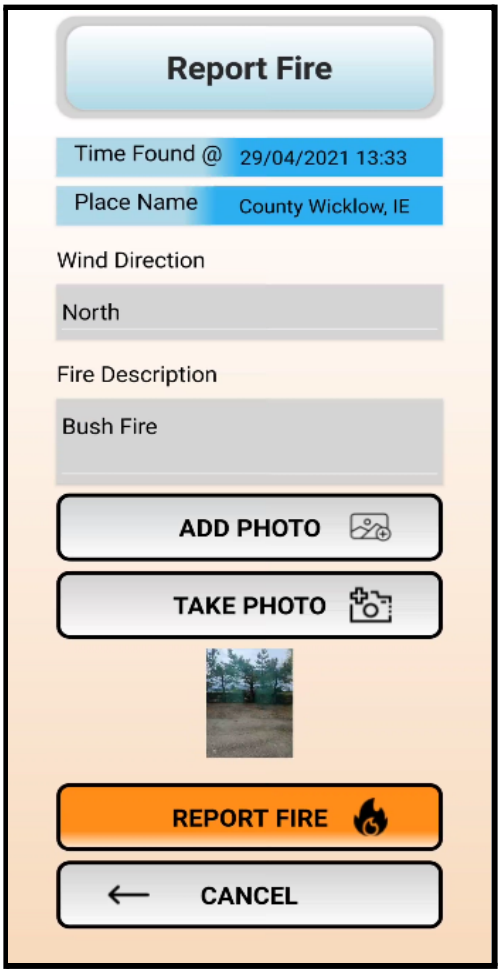


Figure 4 - Report Fire.

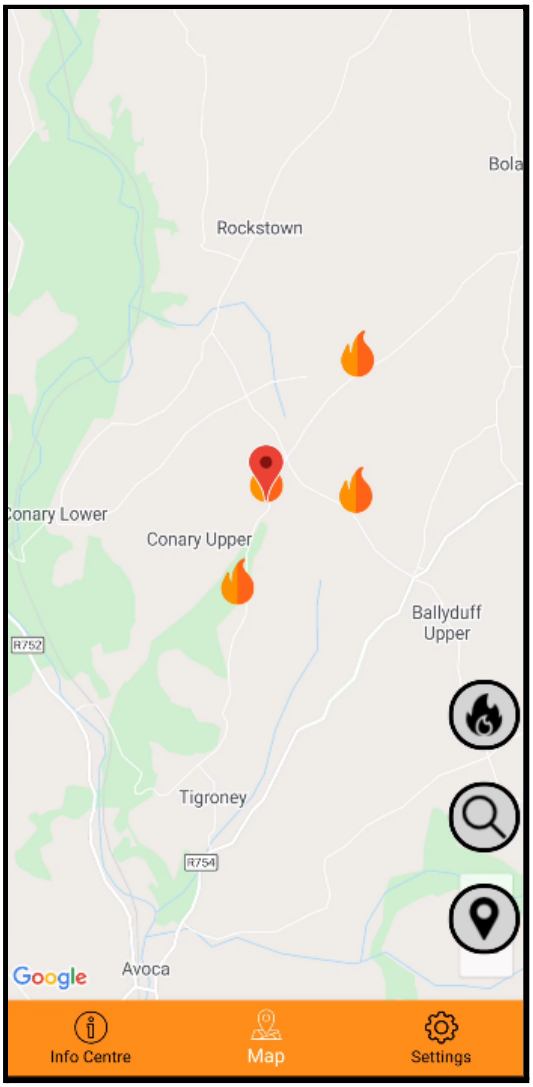


Figure 5 - Updated View Map.

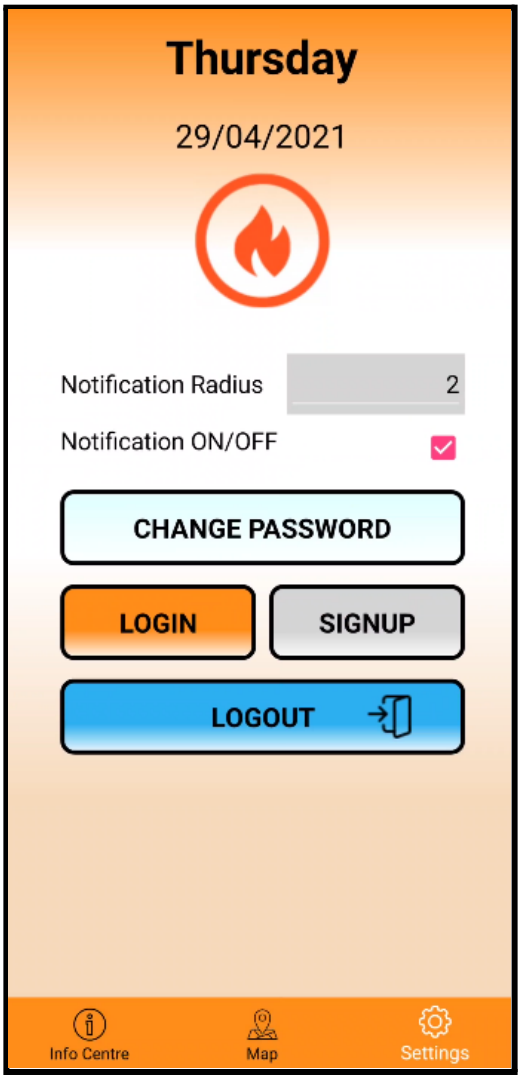


Figure 6 - Settings View

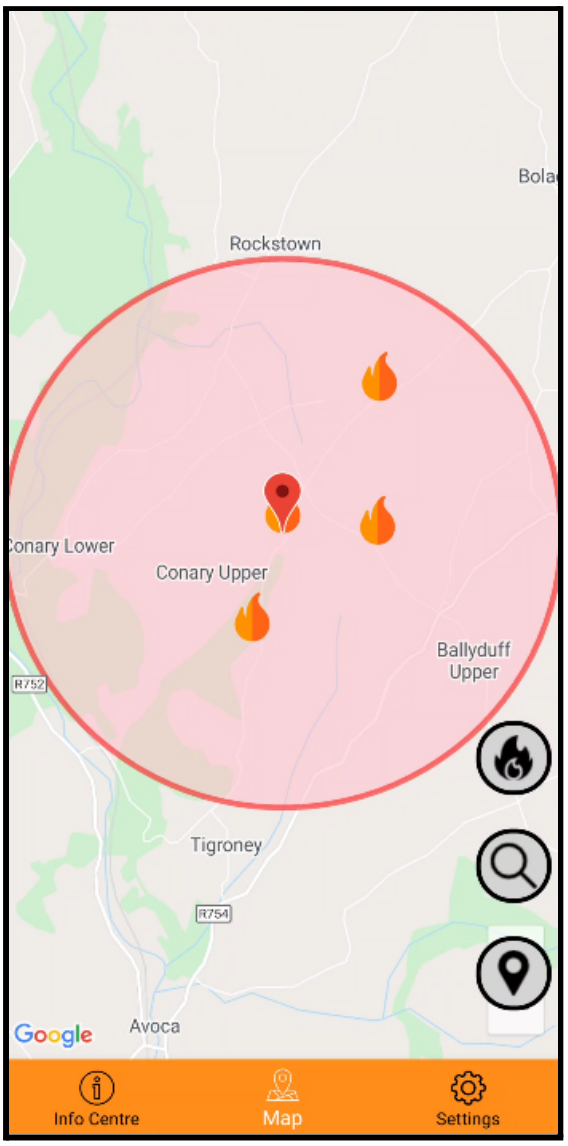


Figure 7 - Update View map with Radius.

Figure 8 - Settings View

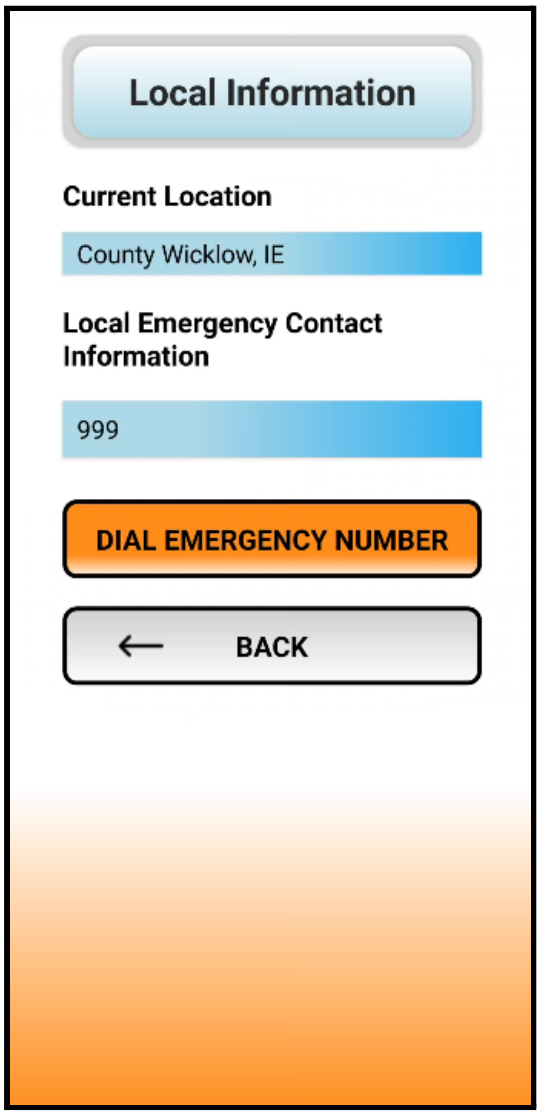


Figure 9 - Local Information.

Figure 10 - Current Fires.

2.3. Xamarin Forms

The use of Xamarin forms was majorly a good thing but it did have its downfalls. When choosing my development platform, I was leaning more towards C# and Xamarin forms is heavily implemented using this language. After much consideration I eventually choose Xamarin forms. I will talk about the successes and challenges that I encountered while developing with Xamarin forms. I used the MVVM structure to structure my project.

2.3.1. Successes - Xamarin

Overall my experience using Xamarin was very successful, one of the reasons for my success with xamarin was the community that in term came with Xamarin. The xamarin

community is large. The dedicated forms from Microsoft for xamarin are full to the brim with so much useful information about every different topic. Also the xamarin documentation is easy to use and is very easy on the eye, this was particularly important as many of the code documentation online is very poorly put together.

Having a very nice documentation website and a strong ever growing community was a big advantage as if I had a problem 95% of the time it has already been solved on either of these platforms.

2.3.2. Challenges - Xamarin

As stated above my experience with xamarin was overwhelmingly positive but like everything it did have its shortcomings. One of the main challenges I faced with xamarin forms was that it was difficult to create nice user interfaces. The use of the hot reload system in place was good to quickly see the front end changes that were made but the fact that if you wanted to make a very appealing and attractive user interface you needed to create your own renderer which took too much time.

The final challenge I faced while using xamarin was the compile times. To compile the code and deploy the app to an emulator would sometimes take in excess of 5 minutes. This sometimes did lead to frustrations as if anything code wise had to be changed on the backend the code had to be recompiled and deployed to the emulator again.

2.3.3. Retrospective - Xamarin

All in all my experience with xamarin was overwhelmingly positive and the advantages heavily outweigh the disadvantages. If I was to do this project again, I would use xamarin but I would definitely begin using a much more powerful PC to try combat the compile times and also do some more research to create custom graphic renderers.

2.4. Firebase

The use of firebase was definitely one of the best decisions I made when deciding what technologies I was going to use to construct the wildfire application. The other options I considered were mySQL and Amazon azure but firebase was undoubtedly the best decision. Firebase is a (BaaS) tool and was used to implement much of the backend of my application

2.4.1. Successes - Firebase

The first main success I encountered when using firebase was the discovery of the other services that it offered. These features include;

- **Firebase Authentication** - This service made the account side of my application extremely secure. The firebase authentication system can automatically use google sign and accounts as its a google service. But it also had the manual option that in conjunction with a NuGet package made my application secure and authenticated very easy.

- **Firebase Storage** - This service is a media storage, so it made the photo upload of my application extremely easy after the initial configuration. This was an extremely beneficial service as I would have had to outsource to yet another 3rd party service to store images.
- **Firebase Database** - This service was the main reason I chose to use Firebase as it is a cheap data store for small-scale applications that was very easy to set up and connect to my wildfire application.

2.4.2. Challenges - Firebase

As I did have many successes with Firebase I did have some issues. The first was the lack of a community, and this was not a massive issue but many of the coding issues I did face seemed like it was the first time anyone had gotten this error. This led to increased problem solving times.

The next challenge I faced with firebase was the initial setup. It was tricky to link firebase to the wildfire application as there were many complex steps that needed to be completed to ensure that the connection was secure and stable. After some time I eventually secured the connection.

The final challenge I faced with firebase was the method in which the data is displayed on the website. Firebase does not use the traditional table but instead it uses a tree type structure which I personally found very tedious. This system was an adequate solution if the application had less than 10 rows of data but once the data started coming in the navigation got much harder.

2.4.3. Retrospective - Firebase

Overall firebase was a genuine success for the wildfire application, and I was able to implement a quick and secure backend service. After the initial challenges were faced the success just kept coming with firebase.

Please be notified if you would like access to the Firebase connection please send me an email on my personal or student email to request permission.

Personal Email: jackjjmcnally@gmail.com

Student Email: c00228758@itcarlow.ie

The API keys have been replaced with the “FB_CONN” & “FB_CONN_1 ”keyword.

2.5. Google Maps API

Similar to firebase the usage of the Google Maps API was a success. Even when I was researching the maps technology I was going to use for the wildfire application I was leaning heavily to the Google Maps API as I knew that it was a very finished product and had many

features to offer to the wildfire application. The Google maps API offers a whole variety of different API's that are aimed towards map development.

2.5.1. Successes - Google Maps API

Overall the use of the Google Map API was a genuine success. The first success I encountered when using the Google Maps API was the community behind this tool and especially the xamarin side of this community. Similar to the other tools I worked with, this API offers a lot of support in terms of forums, documentation and even online video tutorials. This was extremely helpful as I never worked with maps before in the past.

Another key success was the discovery of the Places API that was included in the package. I discovered I had access to this API when I was struggling to find a method to display an adequate autocomplete function in my search functionality. The use of the google places API allowed me to integrate this with the map, this API allowed me to select a location and it returned the latitude and longitude which I could then use to go to the desired location.

Overall my successes with this API are fantastic but like everything it has its downfalls.

2.5.2. Challenges - Google Maps API

The challenges that I faced with the Google Maps API were rare, but the key challenge was price. At the beginning when I made the account to receive this API I received a free trial and after that free trial my billing account would be charged on a "pay as you go basis". The idea of this system had me worrying if I would over use the features and would be charged large amounts of money.

The final challenge I faced with this API was the initial set up in Xamarin. After the initial set-up there was all of the forums and videos to harness but before this the API took a considerable amount of time to set up, issues such as API key implementation in xamarin and other technical issues made the initial set up of the Google maps API difficult.

2.5.3. Retrospective - Google Maps API

Overall the Google Maps API was a success once the initial challenges were faced. This API was a breeze to work with and if I was to need map technology in the future I would turn to the Google Maps API once again.

Please be notified if you would like access to the API key please send me an email on my personal or student email to request permission.

Personal Email: jackjjmcnally@gmail.com

Student Email: c00228758@itcarlow.ie

The API keys have been replaced with the "API_KEY" keyword.

2.6. Testing

The testing side of the application was very interesting. I faced many successes and challenges, testing the application was a very key step to take when creating the wildfire application. Testing the app verified if the logic I wrote actually functioned. The testing side of this application was 50% of the work. Overall my testing experience was overwhelmingly positive, but there was time when I faced challenges.

2.6.1. Successes - Testing

The main success I encountered when testing my application was the uncovering of hidden bugs that I have not encountered yet. These hidden bugs spanned from flawed logic that I had not tested yet, testing my application on different devices and also UI bugs. It was extremely useful to send the wildfire application to my supervisor or one of my peers and in return I would receive a lot of very valuable feedback that would help me test for bugs they encountered. This was helpful as these were bugs that I had not encountered and might not have encountered if it wasn't for testing.

Another main success of testing was the debugging aspect, the process I took when a bug was reported was to try to recreate that error. Once I am able to consistently replicate the error, I would then debug the code. This involved using an android emulator that is on my PC, I then would attempt to pinpoint the error in my code and proceed to set breakpoints and set through the code to eventually find and fix the error. I found this process very beneficial as while I was debugging I would find ways to make my code faster and more efficient.

2.6.2. Challenges - Testing

The challenges I faced when testing my code were limited. The only main challenge that comes to mind is the problem trying to recreate bugs that only occurred on specific devices. At one point near the end of the development of the application I faced a bug that only occurred on Samsung devices. While trying to replicate this issue I realised that this bug was device specific as the two previous devices I tested on could not produce this error. My solution was to reach out to one of my peers who had a Samsung device and ask them to pinpoint the exact moment the application crashes. After some time I eventually overcame this issue, this issue will be mentioned later in this document.

2.6.3. Retrospective - Testing

Overall my testing experience was positive, it really helped me overcome the hidden errors and issues that were present in my code. This step took the wildfire application from a functioning application to a fully functioning application.

3. Issues

This section will contain all of the significant issues I encountered while completing the wildfire application. These issues range from coding issues to the tools I was using to construct the application. Each issue will contain a section explaining the issue and also a

section containing the remediation. The idea of this is to give the reader a greater understanding of the issues I encountered and my thought process when fixing them.

3.1. GPS

3.1.1. Issue

The GPS issue was one of the first important issues I encountered in the wildfire application. This issue was focused centrally around the permissions that the users asked to grant in order for the application to access the user current location.

The main bug this issue was causing was the application not centering the map on the user's current location. This was a very important bug to fix and took some time to remediate.

3.1.2. Remediation

After some time debugging the code and researching the error message I was receiving online, I landed upon the order in which the code was iterating. Initially, once the wildfire application was opened, the code would attempt to get the users current location without requesting the permission.

This error despite taking a long time to find the source of the issue was a simple fix, the fix was to request the permissions before the code attempted to get the users current location. This error also allowed me to implement some error checking so if he permission was not granted upon the attempt to get the current location the application would once again request the user permission to use the device's location features.

```
async Task LoadCurrentPosition()
{
    try
    {
        // Permissions
        var phonePermissions = await Permissions.CheckStatusAsync<Permissions.Phone>();
        phonePermissions = await Permissions.CheckStatusAsync<Permissions.LocationWhenInUse>();

        if (phonePermissions != PermissionStatus.Granted)
        {
            phonePermissions = await Permissions.RequestAsync<Permissions.Phone>();
            phonePermissions  = await Permissions.RequestAsync<Permissions.LocationWhenInUse>();
        }

        if (phonePermissions != PermissionStatus.Granted)
        {
            return;
        }
```

Figure 11 - GPS Permissions remediation.

3.2. Notifications

3.2.1. Issue

The issue with the notification on the wildfire application stemmed mostly from not knowing what solution to implement. Initially, I discovered that firebase had a built-in notification system, but after some research I realised this solution would not work from the wildfire application. The reasoning for this was the firebase solution is a manual solution that meant that the notification needed to be sent manually by an admin user and would also be sent to all the members of the application.

3.2.2. Remediation

The remediation for this was to use the built in xamarin local notification builder. This meant that if a user's condition meant the need for a notification could be built and sent to that single user's device. This solution resulted in a reliable and built in service that did not depend on any third party technologies.
The code for the notifications is built into the main LoadCurrentLocation(), now when the current position is loaded if a user has enabled notifications and a fire appears within the users radius a notification will be pushed to the users device.

```
if (fireNotCount == 1)
{
    DependencyService.Get<INotification>().CreateNotification("Wildfire", "A fire is active in your area.");
    notificationCount++;
}
```

Figure 12 - Sending Notification

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Wildfire.Services
{
    public interface INotification
    {
        void CreateNotification(String title, String message);
    }
}
```

Figure 13 - Create Notification service.

3.3. Search functionality

3.3.1. Issue

This issue was one of the earliest issues I encountered while making the wildfire application. The main issue behind this functionality was displaying the actual location that was searched on the map. This issue stuck around for quite awhile, but after leaving it aside for a few weeks I had learned more about geocoding and geolocation. I had learned how to convert a place name to coordinates and then to go to the specific coordinates on the map.

3.3.2. Remediation

This remediation did take quite a long to come to fruition as I did leave it aside to do some other work. The initial idea was to take the text that was imputed into the search bar and pass it into the Geocoding.getlocation function. Then if the location existed with this function the coordinates could now be used to set a pin on the map in the position that the user has searched. The following snippet of code shows the coding logic of how this was achieved.

```
var search = originEntry.Text;
var searchLocation = await Geocoding.GetLocationsAsync(search);

var sourceLocations = searchLocation?.FirstOrDefault();
if (sourceLocations != null)
{
    Location sourceCoordinates = new Location(sourceLocations.Latitude, sourceLocations.

    Pin pin = new Pin()
    {
        Icon = (Device.RuntimePlatform == Device.Android) ? BitmapDescriptorFactory.From
        Type = PinType.Place,
        Label = originEntry.Text,
        Position = new Position(sourceCoordinates.Latitude, sourceCoordinates.Longitude)
    };
    map.Pins.Add(pin);
```

Figure 14 - Search functionality remediation.

After the remediation was complete I discovered an issue that could not be solved involving the search functionality. This issue was caused only when using the final APK of the wildfire application, when a user attempted to search for a location the autocomplete functionality would not work, but the user is still able to search for a location. In the final demo it is evident that the autocomplete does work when on the emulator environment.

3.4. Firebase Accounts

3.4.1. Issue

This issue heavily revolved around the updating on the professional firefighters accounts on the wildfire application. The issue was simple but the solution was complex, the issue referred to the resetting and changing of the user's accounts. While it sounds as simple as a

few queries to the database and verifying the user, firebase has a totally different approach to this issue.

3.4.2. Remediation

The route that I had to take to complete this remediation was to write the functions myself using the limited resources that were available. This issue does not seem to have a high response rate on the forums so I had to take the manual approach. The first step was to add a forgot password and change password to my IAuth service. Once this was done, I had to implement the methods in the android code. This ended up not being as tedious as I first expected but did end up taking awhile to correctly implement these features.

```
// Forgot Password
public async Task ForgotPassword(string email)
{
    try
    {
        await FirebaseAuth.Instance.SendPasswordResetEmailAsync(email);
    }
    catch(Exception ex)
    {
        ex.Message.ToString();
    }
}

// Chnage Password
public async Task ChangePassword(string newPassword)
{
    try
    {
        var currentUser = FirebaseAuth.Instance.CurrentUser;

        await currentUser.UpdatePasswordAsync(newPassword);
    }
    catch(Exception ex)
    {
        ex.Message.ToString();
    }
}
```

Figure 15 - Firebase Account remediation.

3.5. NuGet Packages

3.5.1. Issue

This issue was not a major issue but helped me understand the issue that I could have encountered if I did not set up my project correctly on GitHub. This issue arose when I attempted to update one of the NuGet packages that was being used by the wildfire application. After seeing that this update was available I immediately selected the update option without reading the update description. This update ended up being a Beta update that should only be updated if needed. This update ended up breaking my application for 3 days.

3.5.2. Remediation

The only remediation for this issue was to revert to one of my previous Git commits. This was a straightforward and simple process and fixed my application. This taught me two main lessons: always commit to Git regularly and never update a NuGet package without reading the update notes first.

3.6. Image Retrieval

3.6.1. Issue

This issue arose when I was implementing the image section of the report fire functionality. The initial uploading of the image was simple enough, the image was then stored on firebase in the firebase storage section. The issues then began to arise when I attempted to display them on the application for the users to view. The problem was when an image is uploaded to the firebase storage service the only way to retrieve the image is to download it. I now understood that it wasn't as simple as querying firebase storage and receiving the image. I knew what I had to do to overcome this issue.

3.6.2. Remediation

After finding out what I had to do to display the image on the screen I set out trying to find a way to implement it. After trial and error I eventually found the solution, I ended up downloading the link from firebase storage and then storing the image in terms of bytes. Once this was completed I could now display the image on the screen.

```
public async Task LoadImage()
{
    try
    {
        var filename = fireTag.Text;
        var webClient = new WebClient();
        var storageImage = await new FirebaseStorage("driven-bulwark-297919.appspot.com")
            .Child("Fires")
            .Child(filename + ".jpeg")
            .GetDownloadUrlAsync();
        string imgurl = storageImage;
        byte[] imgbytes = webClient.DownloadData(imgurl);
        imgChoose.Source = ImageSource.FromStream(() => new MemoryStream(imgbytes));
```

Figure 16 - Image Retrieval remediation.

3.7. Local Information

3.7.1. Issue

The local information issue refers to the local information section in the information center. The initial idea was to harness the power of a reliable external API that contains all of the emergency contact information around the world. After scouring the internet for an API that provides this solution I found a possible solution. But after some initial tests I was unable to use this API because of its lack of documentation.

3.7.2. Remediation

After some time trying to think of a good solution to this issue, I ended up going with a hard coded approach. The Idea was to take the user's current location then find their country code. Once the application has the country code the application can then find the countries emergency contact information and display it to the user.

3.8. Map recentering.

3.8.1. Issue

This was a massive issue that first appeared during the final iteration of the application. This bug was based around the map centering on the user’s current location. The idea was that when a user selected the view map screen the application would get the user’s current location and center the map on their position. But after dealing with an issue that will be mentioned later this issue was born.

This error was uncovered during the external testing of the application and was very important that this error was found as it could have majorly impacted the usability of this application.

3.8.2. Remediation

The remediation to this issue involved a lot of logic, the main problem was the application was going into a loop to only load the fires of the application and not the current location. The default location that is set by the Google Maps API is Rome, and whenever the wildfire application encountered this error the map was centered to Rome.

The remediation involved creating a variable that whenever a section of the application was changed the variable would be set to zero. When this variable was set to zero, the application would load the current fires and the current location of the user on the map. When this variable was greater than 0 the application would only load the fires. This functionality was added to prevent the radius from overlapping itself.

```
if (locationCount == 0)
{
    await LoadCurrentPosition();
    await LoadFires();
    overlay.IsVisible = false;
    loading.IsVisible = false;
    map.IsVisible = true;
    Report_Clicked.IsVisible = true;
    searchPopup.IsVisible = true;
    Location_Clicked.IsVisible = true;
    locationCount++;
}
```

Figure 17 - Map Centring.

3.9. Radius overlapping.

3.9.1. Issue

This issue was caused when a user of the application had a radius set and they kept returning to the view map screen. The radius is displayed in the form of a circle with a solid red border and a translucent red filling. The problem was that this radius circle was being redrawn over the original radius circle resulting in the circle not being translucent anymore.

This issue ended causing other issues but it was a very important find and really helped improve the logic behind the application as well as the usability of the application.

3.9.2. Remediation

The remediation to this issue took quite a long time to get correct. After the initial fix was implemented I sent the application for some external testing. After receiving comprehensive reports I was then able to return to the application and simply implement the further fixes. The images below display where the circle radius is drawn and removed upon every load of the view map screen. It is important to do this every time as a user's location may change.

```
Circle circle = new Circle()
{
    Center = new Position(location.Latitude, location.Longitude),
    Radius = new Distance(Convert.ToDouble(SettingsView.radius) * 1000),
    StrokeColor = Color.FromHex("#88FF0000"),
    StrokeWidth = 4,
    FillColor = Color.FromHex("#88FFC0CB"),
    IsClickable = true
};
map.Circles.Clear();
```

Figure 18 - Initial Circle.

```
map.Circles.Add(circle);
```

Figure 19 - If Radius is present, circle added to map.

3.10. Samsung Testing

3.10.1. Issue

I was only made aware of this issue upon the first round of external testing of the wildfire application. This issue is split up into two main problems: the first issue regards permission and is similar to the GPS issue.

The permission issue only occurred on Samsung devices, when a user wished to report a fire on the map the application would crash and this was due to the application needing a certain permission prior to loading this screen. The second error once again involved reporting fires, this error crashed the application because the device ID that is stored during the process of reporting a fire did not exist on Samsung devices.

3.10.2. Remediation

The remediation to the first issue was inspired by the GPS issue and was to simply request all of the permissions needed to report a fire upon loading the view map screen. The second issue was move complex and not involved getting the device fingerprint which is available on all devices instead of the serial number.

```
// deviceID.Text = Android.OS.Build.GetSerial().ToString(); Optimal solution

deviceID.Text = Android.OS.Build.Fingerprint;
```

Figure 20 - Samsung issue remediation.

3.11. API Key & Firebase Connection

3.11.1 Issue

During the final stages of my project I face an issue with my API keys. This issue involved making my final project public on GitHub. If I made my repository public my API keys would be leaked and could end up having my API key stolen. This is the same for the firebase connection.

3.11.2 Remediation

The solution to this problem was to create a file not tracked by Git that contained the API keys and firebase connection. This now means if my project needs to be cloned the individual cloning the project must email my personal account with verification requesting the API key and the firebase connection .

My Personal email is: jackjjmcnally@gmail.com
alternatively my college email: C00228758@itcarlow.ie

4. Adherence to Specification & Design

This section will contain information of how I adhered to the specification and design manuals. This is an important section to consider as it will allow me to reflect on how things changed through the construction of the wildfire application.

4.1. Adherence to Main Features

The main features of the application that were set out in the requirements have not changed. These features were set out to be the main and most important features of the application. It was important that these features were adhered as they are the core section of the wildfire application.

4.1.1. View Map

The **View map** functionality was the first main feature mentioned and this has kept its place. It was important to consider the view map feature as one of the main features if not the most important as this is where all of the features are housed. The view map is where the fires are reported and resolved. This was the first feature that got worked on, and it was very important to testing the view map feature to make sure it functioned to its best ability.

The view map functionality also included one of the secondary functionalities. The search location feature is an important feature that allows users to search different locations. Altogether this feature was very important to be adhered to correctly.

4.1.2. Report Fire

The **Report fire** functionality was the next main functionality mentioned and it has also kept its place as one of the most important features of this application. This application is built on the reporting of fires and keeping safe from fires and this is the feature that allows the users to actually report the fires.

The development of the report fire functionality was the next significant piece of coding that was carried out on the wildfire application. It was very important to adhere to this feature and make sure that it was developed properly. While developing this feature some features did improve initially I did not plan on adding photos to the report fire screen and after some consideration it was decided to add photos as it would be a useful and beneficial feature to the users of the wildfire application.

4.1.3. Resolve Fire

The final main functionality of the application has also kept its spot as one of the main features of the application. It was very important to make sure that this feature was developed correctly as it is a feature that is only accessible by the professional firefighters that use this application. This feature allows the professional users to resolve the fires, this feature is very important as the responsibility to resolve fires could be the difference between panic and relaxation.

This functionality also had the add photo feature added to the original features. This feature was added as it is considered useful and beneficial to the professional user and it will help the user better visualise the resolved fire.

4.2. Adherence to Secondary Features

The secondary features of the application that were set out in the requirements have not changed. These features were set out to be the second most important features of the application. It was important that these features were adhered as they are the core section of the wildfire application.

4.2.1. Search Information Center

This feature was first secondary functionality mentioned and has kept its place as a secondary feature. There have been features added to this functionality, originally it was planned to only have three sections to this feature.

The firefighter section was added as an exclusive feature only available to professional users and contains lists of current and resolved fires. The main aim of this feature is to

provide all of the user's of the application with helpful information that is aimed at keeping the users of the application safe in the event of a fire.

4.2.2. Set Notification Preferences

This feature was the next secondary functionality mentioned and has also kept its place as a secondary feature. This functionality works with the radius functionality that will be mentioned next.

The main aim of the functionality is to notify users if a fire has been reported within the radius that the user has set, if the user enabled notifications and has not set a radius the application has a default radius of five kilometers. This was a very important feature to develop correctly as it may be the only thing that notifies the users if a fire has been reported in their area.

4.2.3. Set Radius Settings

This is the final functionality that was mentioned in the secondary functionalities and has also kept its place. This functionality works with the set notifications preference functionality. The main aim of this functionality is to allow the user to set a radius from their current location. This feature then works with set notification preference and if the user has allowed notifications the user will be pushed a notification alerting them of a fire within their radius.

4.3. Adherence to Application Users

4.3.1. General Public

The roles of the general public user throughout this application have changed, originally the general public user had the option to create an account. Now the general public user is not expected to create an account and needs no verification. The original idea behind making the general public user create an account was down to verification. By making the user create an account they would have to enter an email and password, this email could then be used to track the users reports.

After some thought and consideration I decided to only allow accounts for professional users. This now means that the general public users now do not have to waste time creating an account and enter details whenever they want to use the application. This was a very important decision as I believe this resulted in a greater usability of the application.

The features for the general public users have not changed as they can still access all of the key parts of the application that they may need to access. The only feature the general public users do not have access to is the exclusive features to the professional firefighters. By allowing the general public users to report a fire without having to create an account it is vital as it could eventually lead to the prevention of loss of homes or life.

4.3.2. Professional Firefighters

The roles of the professional firefighters has not changed much throughout the creation of the wildfire application. The professional firefighter role is still to respond and resolve wildland fires.

Professional users are still required to make an account, this process will also require the professional fire fighter to verify their status as a professional firefighter. The original idea to provide verification was to supply a documentation or badge of some sort via image. This method of verification would then be reviewed by the admin of the wildfire application who then confirms or denies the account. Due to time constraints and limited information the simplest way to implement this was to ask the user to check a checkbox to confirm if the user is a professional or not.

All of the features of the application are still available to the professional user as well as the exclusive features. It was important to give the professional users of this application these high responsibility features so the professional users have a reason to use this application.

4.4. Changes to Design

The design of this has changed a considerable amount of the duration of this wildfire project. The main section of the design that has changed is the screen structure of the application.

4.4.1. Report Fire

In the initial design of the application the report fire screen had an overview screen. The idea of this screen was to display back to the user all of the information that they had entered to report a fire. After some consideration while coding the screens it was decided that the report fire process would take much less time and how encourage the user to report more fires in the future if they had to.

After the report fire overview screen was removed the design of the report fire screen was improved to display the information entered much clearer. Now when a fire is reported the user must enter the information on the report fire screen and once they are ready select the report fire option process would take much less time and how encourage the user to report more fires in the future if they had to.

4.4.2. Resolve Fire

In the initial design of the application the resolve fire screen had an overview screen. The idea of this screen was to display back to the professional users all of the information that they entered to resolve a wildland fire. After some consideration while coding the screen it was decided that the resolve fire process would take much less time and how encourage the user to report more fires in the future if they had to.

After the resolve fire overview screen was removed the design of the resolve fire screen was improved to display the information entered much clearer. Now when a fire is resolved the

professional user must enter the information on the resolve fire screen and once they are ready select the resolve fire option process would take much less time and how encourage the user to resolve more fires in the future if they had to.

5. Learning Outcomes

During the completion of this project, I have learned a lot of useful information that I will be able to bring with me in the future. I have separated this section into main parts, the first part will discuss the technical skills I have learned and how they have benefited me. The second part will contain the skills that I have learned in the project management side of the wildfire application and how I can use them in the future.

5.1. Technical Learning

The technical learning that I undertook was tremendous. I believe my coding ability has taken a step up the skill ladder. In this section I will discuss Xamarin and firebase as the two main sources of technical learning in this project.

5.1.1. Xamarin

Xamarin is the tool that I used to develop my whole application. Xamarin is an extension of the .NET platform that specialises in the development of Android, iOS and windows applications using the same shared code base. I used a combination of both Xamarin forms and visual studio which was used as my main IDE.

Xamarin uses C# as the main language to build the backend side of the application while on the front end XAML is used. Xamarin also uses an extensible markup language which is used to build the front end section of the xamarin application. The final key factor to point out is that xamarin can access an online code library known as NuGet packages. These packages are fully functionally code that has been created and maintained by other developers for free usage. These NuGet packages are extremely useful and allowed me to implement API's, Map and many more features by downloading a simple package.

Overall my experience with xamarin was very positive I leaned how to code in C# which was a language that I was very inexperienced in. I can now use XAML to build user interfaces, a skill that I only knew how to achieve with HTML and CSS prior to this experience. The usage of xamarin also taught me how to implement android specific features, I know how to implement a local notification service on android.

I believe my coding ability has immensely improved throughout the implementation of this project. I believe that this will help me immensely in the future and has also taught me how to approach complex coding problems and break them down into manageable steps.

5.1.2. Firebase

Firebase is a realtime database service that is provided by Google. Firebase is a (BaaS) Service. Firebase also offers a wide variety of different services such as firebase storage and firebase authentication. The initial decision to use firebase over the other solutions available on market was a key decision.

Firebase uses a JSON like structure to display its database service, as my application is only a small scale application this structure is handy. I am able to quickly search through the entries and find what I am looking for quickly. The database service firebase offers a noSQL database this means no queries are needed to pull the data to the application. This is extremely helpful as it speeds up development time.

Overall my experience has been overwhelmingly positive, when researching all of the potential database technologies that I could have used for this application I was dreading that fact of having to work with the old school SQL language. But then I stumbled upon Firebase and after some research I was excited to get started using it and seeing what it offers. Now after using firebase for the best part of 7 months, I have learned a lot from securely storing users to storing and retrieving images. Firebase has turned the database service into an enjoyable learning process.

5.2. Project Management

The project management side of the project was another large area of learning. This area was a self learning area as I learned how to plan and manage the project much better as time went on. I will discuss the tool Git and how it helped me project management. Finally I will discuss how my timing and planning skills increase and how I kept track of them.

5.2.1. Git

Git is the version control that I used to keep track of my code throughout the project. This tool allowed me to manage the different versions of my code. This was very helpful as at one stage of the project around the halfway point I had to revert an older commit. The use of GIt became a very important step as the project went on as it was implementing fixes and complex sections of code every time I would implement one of these features I would commit a version of my code to Git.

This ensured that if one of these complex sections of code caused issues to the application I was able to revert to an older commit and start that section again.

The main thing I learned from using Git is to use it regularly; this ensures you always have an up to date version of your code and the overall management of the project improves as well.

5.2.2. Timing & Planning

The timing and planning aspects of this project were absolutely vital. I was helped by my supervisor Chris to plan the project from day one. In our first meeting we discussed splitting up the project into three iterations. These iterations were split evenly throughout the year.

In each iteration I decided what sections of the project I wanted to have completed, what functionality I wanted to have functioning and how much documentation I should have completed. This allows me to plan each section of the project to my own workload and my own pace. I have learned a lot about planning and timing as the project went on, at the beginning I was trying to do too much work. In the middle I wasn't doing enough work now at the final section of the year I know exactly how much work I can handle at one time.

Overall I learned a lot about timing and planning. I believe that I am now comfortable to give time estimations on how long it should take me to complete a task based on how complex the task is. Also I have learned to keep visual lists that after every task I can mark off and If a new task appears I can add that to the list. These lists gave a reference point to how much work I had completed and how much work still needed to be done.

I believe I have gained an invaluable skill that will greatly benefit me in the future as in every company there are time limits and deadlines. I believe I now have a base to learn from and start ascending the skill ladder.

6. Project Review

This section will contain an overall project review, I will discuss the work I achieved throughout the implementation of this application, the work I did not achieve and final if I had more time what feature I would focus on improving.

6.1. Achieved

Throughout the implementation I have achieved most of my original goals for this project. In the beginning when setting out to create this project I set a goal to complete all of the functionalities mentioned in the use case diagram.

I have created an Android application that was the ability to report and resolve wildland fires. This is all housed on a view map screen which displays the user's current position and a radius if set. The view map screen also displays all of the current fires on the map. Lastly the view map screen offers three more options in terms of reporting a fire on your current location, searching the map for a location and finally a button to return to your current position. All of these features make up the view map screen have all been achieved.

Moving onto the information center section of the project, the Information center screen houses four main screens, first the home fire safety screen that includes information that can be used by any user of the application. Next the wildland fire safety section similar to the home fire safety section this section contains information accessible by all users aimed at keeping you safe from wildland fires. The next section of the information center screen is

accessible by all users of the application and allows the user to access the information about the current location they are at. Finally the last feature on the information center that was achieved was the firefighter section, this is an exclusive section only available to firefighters and contains lists of the current and resolved fires. All of the features that were mentioned above have all been achieved.

The final section of the app that I have achieved multiple features in is the settings page, this page holds the notification preferences and the radius settings. These two features are both integrated with the view map screen and are also stored in the cache memory of the application. This means if the application is shut down the preferences are still remembered. This screen also holds a link to the change password screen and is only accessible by professional users.

Some additional features that I have not mentioned include the Login/ Logout functionality, the create an account functionality, forgot password functionality and finally the dial emergency number on the local information screen.

The list below contains all of the features that I have achieved to create and make fully functional.

- Android Application.
- Report Fire.
- Resolve Fire.
- View Map.
- Report Fire on current location.
- Search a location.
- Move to the user's current location.
- Information center - Home,Wildland,Local and Firefighters
- Settings View - Notifications preferences, Radius setting and change password
- Login
- Logout
- Create an Account.
- Forgot password.
- Dial local emergency number.
- Push notification

6.2. Not Achieved

The section of the project I did not achieve was very small but significant. I achieved most of the functionality I set out to achieve. The main section I did not achieve was creating an iOS version of my application. This initially boiled down to the resources that I had available at my disposal.

The second piece of work I did not get to achieve was the use of an emergency API that had access to all of the emergency contact information across the world. I could not achieve this

because there was no API available to use and the only solution that I found near viable did not contain the information that I needed.

The list below contains all of the features that I did not achieve and make fully functional.

- iOS.
- Emergency Information API implementation.
- Search functionality autocomplete(APK).

6.3. Future Development

In this section I will discuss the future development of my application if I had the time and resources available. The first step I would take would be to make the iOS version of the application. This would include getting an iOS developers license and also purchasing an iPhone for testing. This would be a very important step as 28% of the mobile device market are iOS as of march 2020[1].

The next step I would take would be to improve the applications user interface, this would involve getting the help of a UX designer and getting them to redesign the UI of the application. In this step I would focus on implementing a fix for the autocomplete functionality, this is an important feature that will help with the overall usability of the wildfire application.

The next step would be to improve some of the functionalities of the application, I would first improve the notification setup so you can receive them offline, I then would allow the user to upload more photos. To the report and resolve fire screens.

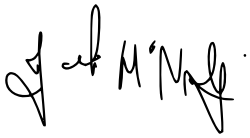
The final step I would make to my application would be to create an admin dashboard so that the future admin users of the application would be able to have a wider display and also some metrics on the reporting and resolving of fires.

7. Declaration

- I declare that all material in this submission, e.g. thesis/essay/project/assignment, is entirely my own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams, or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in the preparation of this submission.
- I understand that failure to comply with the Institute’s regulations governing plagiarism constitutes a serious offense.

Student Name : Jack McNally

Student Number : C00228768

Student Signature : 

Date: 30/04/21

8. References

[1]. StatCounter Global Stats. 2020. *Mobile Operating System Market Share Worldwide | StatCounter Global Stats*. [online] Available at: <https://gs.statcounter.com/os-market-share/mobile/worldwide> [Accessed 28 April 2021].