Applied Project and Minor Dissertation

Eoghan O'Connor, Software Development

Galway Mayo Institute of Technology

Project Supervisor Dominic Carr

GMIT
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Table of Contents

Project Overview

Abstract FuelMe is a community lead initiative to get the most recent fuel in a 12km radius. Users are encouraged to enter fuel prices where possible and where safe to do so. In the current climate saving money where possible is essential and this app allows a community of people to make the decision of where to get fuel. The app also gives information about the station such as whether or not the station is open. This application is an android application using the Parse[13] Platform as a backend.

Introduction

The Introduction serves as introduction to the project. I will cover various aspects of the project and the main objectives of the project. all chapters will also be discussed and a small description of what each chapter will contain.

1 Chapter Summary

- **Chapter 1 - Introduction** - This Chapter on will discuss the aims of the project.
- **Chapter 2 - Methodology** - This Chapter will discuss the methodologies used in research and and development.
- **Chapter 3 - Technology Review** - This Chapter will discuss technologies, languages and tools used.
- **Chapter 4 - System Design** - This chapter will discuss the design of the system.
- **Chapter 5 - System Evaluation** - This chapter will evaluate the system as a whole
- **Chapter 6 - Conclusion** - This chapter will contain all conclusions of the project.

2 Aims and Goals

The aim of this project was to create an Android app that allows users to input fuel prices within a 12km radius of their current location. The aim is to have users make and actively add prices, have those prices update a Parse database so they're retrievable by other users on the platform so the community as a whole can see prices nearby and make more informed decisions.

3 Fuel Prices in Ireland

Irish fuel prices rank 23rd highest for petrol[14] and 22nd highest in diesel prices[4] at the time of writing. Irish people are also required to commute further than ever before due to the on going housing crisis[3], Speaking from personal experience I have commuted to GMIT for five years and on average it takes me 1.5 hours to travel the 24km from my home to GMIT, over the years my fuel bill has stayed between 50 - 60 euro a week, as a full-time student and part-time shop assistant this is a lot of money to take out of an already tight weekly budget.

4 Objective of FuelMe

- Create service the helps users make more informed decisions regarding fuel purchases.
- Create and easy to use android application accessible to people of all abilities.
- Allows users to register and few all fuel stations within 12km radius.
- Encourage users to take part in the community driven initiative.

5 Project Scope

When initially considering what to do my project on I had two ideas, the first was stock management system to allow shops to minimise waste by tracking stock sold. After some investigation it turned out the idea had too many wholes and would require a lot of hardware and expense. FuelMe was my second idea, during my final year I started working in a filling station and would offend have customers complaining about fuel prices, this spawned the idea for FuelMe. After discussing the idea with my project supervision I began research in the area, after this I moved into learning to use the technologies that were to be used in the project. The register and login systems where implemented by Christmas. After this Maps were added as well as map markers for filling stations within the given radius, then user profile editing and viewing were added. Finally adding prices and updating prices were implemented.

5.1 Requirements

I set out a series of requirements for the project that I would use to guide me in the development of the application.

- Allow users to register to the platform.
- Show users all filling Stations within a 12km radius.
- Allow users to enter prices for stations.
- Allow users to update prices for stations.
- Allow users see all information stored about them.

Methodology

This Chapter will give detail the various methodologies that were implemented in this project. It takes a look at the different types of research methodologies that were used such as Quantitative Research, Qualitative Research, and Mixed Methods Research. Furthermore I will outline the different development methodologies used in this project such as Continuous Delivery and Agile based methodologies. This Chapter will also cover meetings, development tools and project management.

6 Research

Throughout this project I used many different Research methods. Qualitative Research is primarily exploratory research. It is used to gain an understanding of underlying reasons, opinions, and motivations. Qualitative Research is an informal and is used to help the researcher in performing Quantitative research. Quantitative Research is used to quantify the problem by way of generating numerical data or data that can be transformed into usable statistics.
Research began with looking for different places consumers can see actuate fuel prices. In my research I came to the conclusion that nothing exactly like fuel me is readily available to the Irish public the two closed were GlobalPetrolPrices[8] and The AA motoring advice page[18]

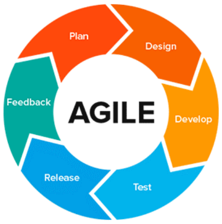
6.1 GlobalPetrolPrices

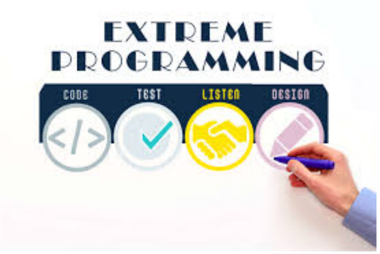
GlobalPetrolPrices is a website that displays natural resources prices for 135 countries. In their Sources and Methods pdf[16] they state they do not use automation and they collect the data by hand. The prices are updated weekly.

6.2 The AA

The AA are an insurance company founded in the UK in 1905[19]. Their fuel prices are similarly set up as GlobalPetrolPrices they update prices on a weekly basis and dont show station by station info. They provide a break down of how the tax is applied.

7 Software Development Methodologies





I adopted an a Extreme programming(XP) approach to this project.XP is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development[20]. XP as developed due growing frustration with the waterfall methodology. Waterfall is a highly structured me methodology and doesn't account for iterative changes. XP allows the developer to change and produce software faster and allows for small iterative changes. XP allows developers and customers to make changes as the project progress's through the Software Development Life Cycle.

It became clear that XP was the best methodology for this project as it allowed for small changes to be made at the developers discretion, incremental development, prototyping and although there was no customer I decided to apply this methodology.

8 Meetings

Meeting took place on a weekly basis for the duration of the project. The meeting consisted of the following

- Project updates.
- Feedback from supervisor.
- Discussing features and the scope of the project.
- Possible additional features.
- Planning next step in development.

Meetings initially were used for brainstorming ideas that were to be researched. The following weeks was used to discuss research and decide on a final project idea. Following this the meetings took place on a weekly basis and discussed the progress over the following week, problems experienced during the week and the solution to those problems, problems current affecting the project and possible solutions to those problems.

9 Development Tools

The main development environment I used was Android studio. Android Studio is built on the IntelliJ IDE and is enhanced by Google for use with Android. Android Studio allows for in built Git controls and allows users to design the aesthetic of the app and develop the backend in one IDE.

10 Source Contorl

All the code for this project was regularly committed to a remote github repository. GitHub provides software version controls using Git[7].

Advantages of GitHub

- **Distributed Development** - Git is a distributed version control system. Instead of a working copy, each developer gets their own local repository, complete with a full history of commits. Having a full local history makes Git fast, since it means you don't need a network connection to create commits, inspect previous versions of a file, or perform diffs between commits.
- **Faster Release Cycle** - As a result of feature branches, distributed development, pull requests, etc. is a faster release cycle. This facilitates an agile workflow, encouraging developers to share smaller changes more frequently. This results in changes get pushed down the development pipeline faster

Disadvantages of GitHub

- **Cost -** some of GitHub features require users to pay for the service. This didn't impact me personal however if a large team all need a subscription and private repositories.
- **Security -** GitHub offers private repositories however like all data storage companies they're under a constant threat of a security breach. Companies with private repositories and high value intellectual property this may not ot a risk they're willing to take.

Technology Review

This chapter will take an in depth look at all the technology used throughout the project. I will compare other systems that could have been used and why I choose to not use them. I will also discuss the advantages and disadvantages of each technology used in the project.

11 Overview

This project is an android app using native android xml as a frontend, Parse as a back end hosted on Amazon Web Services which deals with user authentication as well as storing all information about all the filling stations add to the platform and Google Maps and Google Places APIs are used to get information about the service stations. Before I decided to use these I researched other alternatives such as.

- Ionic/ Hybrid app
- Firebase
- mySql
- Google Cloud Services

12 Main Technologies

This section I will take an in depth look at each technology used and some of the advantages and disadvantages, as well as other technologies that could have been used and why they weren't.

12.1 Android

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google[2]. Android was at the time of writing has 70% of the market share[15]. This would suggest the there are roughly 2.5 billion active Android users. Developing on IOS was disregarded straight away due to the limitations of my hardware, Apple requires developers to use Apple products to develop applications for their platform. Secondly I considered developing a Hybrid application, Hybrid application are installed on a phone like any other app however they're deployed in a native container that uses a mobile WebView object. When the app is used, this object displays web content thanks to the use of web technologies, Hybrid apps can access a devices hardware and use the camera or accelerometer. Hybrid apps lack the feel of a native app, Apps developed natively have a more familiar feel and can encourage a user they by making them more comfortable. Ionic is one of the most notable hybrid app frameworks. Ionic uses HTML, Javascript and CSS for styling however they also support React and Angular. Although hybrid apps offer a cost effect way to desgin and develop apps for all platforms and make updates very easy they do have limitations. Hybrid apps can't fully take full advantages of device hardware, in my circumstance Hybrid would not work as I would need GPS to be fast and get user location effectively. After a process of elimination I was left with Android as my platform of choice, I was happy with this as app development is a big sector in the software industry and it is something I was wanted to try since I started in GMIT. Furthermore I wanted to take this time to improve my skills in Java.

Advantages and Disadvantages of Android Development. The Android platform has Advantages and Disadvantages like all ecosystems.

Advantages

- Android has the Market share in terms of users, this allows all apps to get the most users, and in a community initiative that's a big advantage.
- Android is a very accessible platform to develop on. Android Studio is compatible with both Windows, Mac, Linux and ChromeOS, this helps newer developers as they're a lot of tutorials online as well as online courses.

Disadvantages

- The number of mobile manufacturers may be big but it's still manageable compared to the number of devices these providers release – each with their own screen size, sensors, performance issues, and graphics drivers.
- The range of operating systems is large and one of the challenges related to app customization. According to Google's own 2018 report, most Android devices are still running on Lollipop, Marshmallow, and Nougat – operating systems from 2014, 2015, and 2016 respectively.

12.2 Parse

Parse was a mobile backend as a service platform originally developed by the provider Parse, Inc. The company was acquired by Facebook in 2013 and shut down in January 2017. Following the announcement in 2016 of the impending shutdown, the platform was subsequently open sourced[13]. Parse was founded in 2011 by Tikhon Bernstam, Ilya Sukhar, James Yu, and Kevin Lacker, previously at Google and Y Combinator. The firm produces back-end tool for mobile developers that help developers store data in the cloud, manage push notifications and manage log-in systems. In 2014 it was reported that parse provided the backend services for 500,000 apps[12]. Parse offer SDKs for Android, IOS and Unity. I decided to use parse as it continually appeared during my research into how to manage the back-end of an application. Firebase is similar to Parse in its ability to offer developers data storage in the cloud, manage push notifications and manage log-in systems. It was acquired by Google in 2014[6]. I have previous experience with Firebase from a Mobile Development module in second year and the experience was positive however I felt learning to use a different back-end system would benefit in the long run. Parse was very easy to set-up and easy to use on the Android platform, The android SDK was easy to implement.

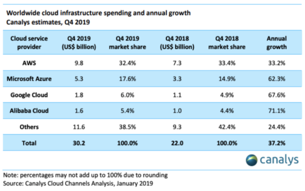
Advantages and Disadvantages of Parse Server Like all back-end platform Parse server has it's advantages and disadvantages. **Advantages**

- Parse is very simple to use. It took me about 15 minutes to set up a user registration system.
- Very active community with plenty of resources online.
- Documentation is very readable and explains everything well.

Disadvantages Parse is a very rounded service and I am very happy with its implementation in my application, however linking classes or tables of data can be tricky.

13 Amazon Web Services

Amazon Web Services (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. In aggregate, these cloud computing web services provide a set of primitive abstract technical infrastructure and distributed computing building blocks and tools[1]. As of Janurary 2019 AWS has 32.9% of the market share, partially due to its feature rich suite of cloud based services but mostly due to its seven year head start on like minded competitors.



Worldwide cloud infrastructure spending and annual growth
Canalys estimates, Q4 2019

Cloud service provider	Q4 2019 (US$ billion)	Q4 2019 market share	Q4 2018 (US$ billion)	Q4 2018 market share	Annual growth
AWS	9.8	32.4%	7.3	33.4%	33.2%
Microsoft Azure	5.3	17.6%	3.3	14.9%	62.3%
Google Cloud	1.8	6.0%	1.1	4.9%	67.6%
Alibaba Cloud	1.6	5.4%	1.0	4.4%	71.1%
Others	11.6	38.5%	9.3	42.4%	24.4%
Total	**30.2**	**100.0%**	**22.0**	**100.0%**	**37.2%**

canalys

Note: percentages may not add up to 100% due to rounding
Source: Canalys Cloud Channels Analysis, January 2019

Google Cloud Services offer many of the same services as AWS. Google Cloud has roughly 6% of the market share. Similar to my choice in using Parse I decided to use AWS as I wasn't familiar with the platform and felt that I should use something different. During my time at GMIT I have used Google Cloud platform and a number of occasions so felt I have a good gasp on the service.

Advantages and Disadvantages of AWS Advantages

- AWS is very easy to use and to setup an instance of any sort. The instance that runs the Parse server took roughly 30 minutes to set up and deploy for the first time.
- AWS is a very reliable service. During the development of FuelMe the user authentication never wnet offline

Disadvantages

- Although the Parse server is running on the free tier I have received a number of bills of usage.

14 Google Maps and Google Places APIs

””

Google launched the Google Maps API in June 2005 to allow developers to integrate Google Maps into their websites. It was a free service that didn’t require an API key until June 2018 (changes went into effect on July 16), when it was announced that an API key linked to a Google Cloud account with billing enabled would be required to access the API[9].The Places API is a service that returns information about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points of interest[10]. Maps were an integral to the success of FuelMe I decided Google Maps and Google’s Places API were the only to consider due to there ease of use and massive amounts of data available.

Advantages of Google Maps and Google Places APIs Advantages

– Location data is pulled in real time from the Google Map’s location database, meaning your locations always use the latest location data available from Google.
– Location searches are automatically tailored to your users’ likely behavior. For example, if a user is located in India, the system will read their IP address and provide them with location suggestions that are most likely to fit their needs

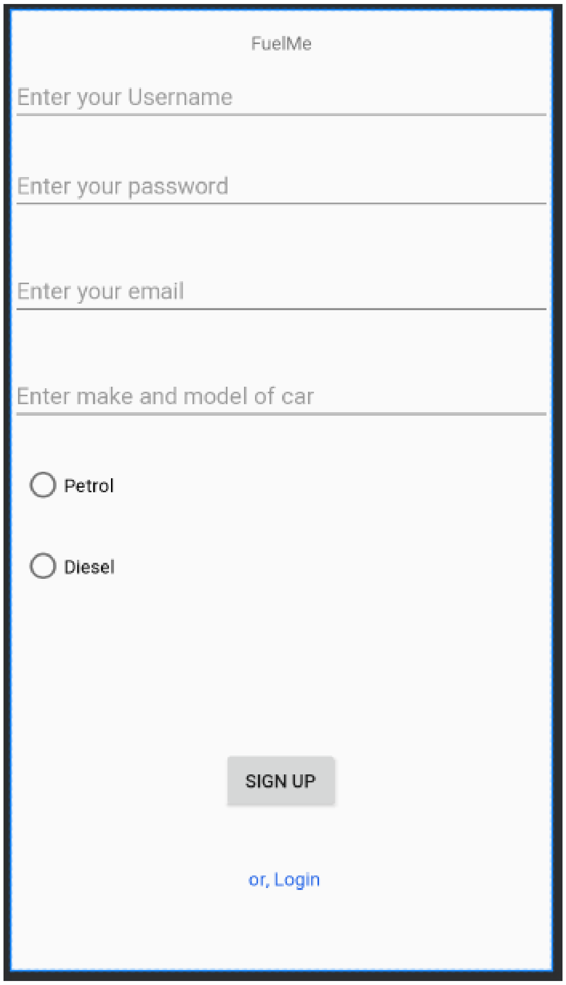
Personally I don’t feel Google Maps API and by extension Google Places API has any Disadvantages.

System Design

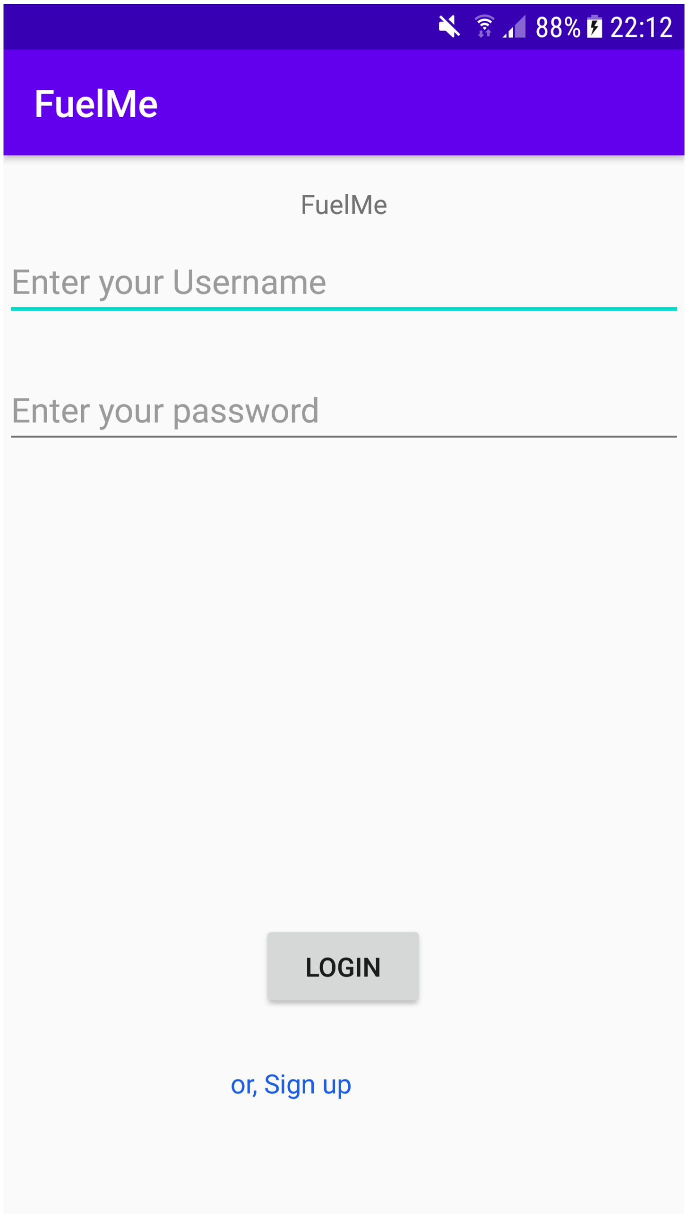
This Chapter will be a detailed through each element of the project. What each class does and why it does. Certain elements of this project are dictated by the platform and I wasn't able to change.

15 Login

The Login page is the first page of the that greats the user.



I decided that a simple layout will help less "techsavy" users feel they can use the application, On launch the page is set up as Registration page, when the user presses the "Or, Login" TextView the Page to a Login Page.



””

The Login page consists of user name and password field and a login button. as well as a ”or, Sign Up” TextView that will change the view back to a registration page.

The coding behind the Login/Registration is very simply and highlights the many of the reasons I chose to use Parse to manage the back-end of this Project.

```
public void onClick(View view){
        if(view.getId() == R.id.ChangeSignUpMode){
            if(signUpMode){

                signUpMode = false;
                signUpButton.setText(”Login”);
                changeSignUp.setText(”or, Sign up”);
                email.setVisibility(View.INVISIBLE);
                makeModel.setVisibility(View.INVISIBLE);
                diesel.setVisibility(View.INVISIBLE);
                petrol.setVisibility(View.INVISIBLE);
            } else {
                signUpMode = true;
                signUpButton.setText(”Sign Up”);
                changeSignUp.setText(”or, Login”);
                email.setVisibility(View.VISIBLE);
                makeModel.setVisibility(View.VISIBLE);
                diesel.setVisibility(View.VISIBLE);
                petrol.setVisibility(View.VISIBLE);
            }

        }
    }
```

This OnClick Method deals with changing the Registration page to a Login page. Simply by setting the visibility of the different UI elements. The Boolean signUpMode also chnages the function of the Button.

The OnClick Method assigned to the button is display beneath

```
 EditText username  = (EditText) findViewById(R.id.usernameEditText);
 EditText password  = (EditText) findViewById(R.id.passwordEditText);

if (username.getText().toString().matches(””)||
        password.getText().toString().matches(””)){
    Toast.makeText(this, ”Please Enter a username and a password”,
                    Toast.LENGTH_SHORT).show();
} else {
    if (signUpMode) {
        ParseUser user = new ParseUser();
        user.setUsername(username.getText().toString());
        user.setPassword(password.getText().toString());
        user.setEmail(email.getText().toString());
        user.put(”makeModel”, makeModel.getText().toString());

        if(petrol.isChecked()){
            diesel.setChecked(false);
            user.put(”fuel”, ”Petrol”);
        } else {
            petrol.setChecked(false);
            user.put(”fuel”,”Diesel”);
        }
        user.signUpInBackground(new SignUpCallback() {
            @Override
            public void done(ParseException e) {
                if (e == null) {
                    Log.i(”Signup”, ”Successful”);
                    login();
                } else {
                    Toast.makeText(LoginActivity.this,
                                    e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        });
```

First the application checks what operation the user wishes to perform, from there a new ParseUser is created. Parse provide a predefined User class that deals with user authentication and user registration. Elements such as Username, Password and Email was predefined attributes as seen above. For developer created columns you must use the .put attribute. the signUpInBackground signs the user up and returns an error if it's unsuccessful, if there is no error the user can be brought to the next page.

```
else {
    ParseUser.logInInBackground(username.getText().toString(),
                                password.getText().toString(),
                                    new LogInCallback() {
        @Override
        public void done(ParseUser user, ParseException e) {

            if (user !=null){
                Log.i(”Appinfo”,”Successful”);
                login();
            } else {
                Toast.makeText(LoginActivity.this, e.getMessage(),
                                Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

Logging in using Parse is also a very simple process, similar to signing up Parse makes this easy to the developer and by simply checking the the response from the server and allowing the user through to the next page.

```
public void login(){
        // opens App to user
        Intent intent = new Intent(this, MainActivity.class);
        startActivity(intent);
    }
```

The Login method simply sets up an intent and puts the user into the main activity.

16 Main Activity

The Main Activity sets up the tabbed navigation style the app follows. I didn't edit or change in this class.

```
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        BottomNavigationView navView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        AppBarConfiguration appBarConfiguration = new AppBarConfiguration
        .Builder(
                R.id.navigation_home,
                R.id.navigation_dashboard,
                R.id.navigation_notifications)
                .build();
        NavController navController = Navigation.findNavController(this,
                                                     R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
                                                        appBarConfiguration);
        NavigationUI.setupWithNavController(navView, navController);
    }
```

17 Dashboard Fragment

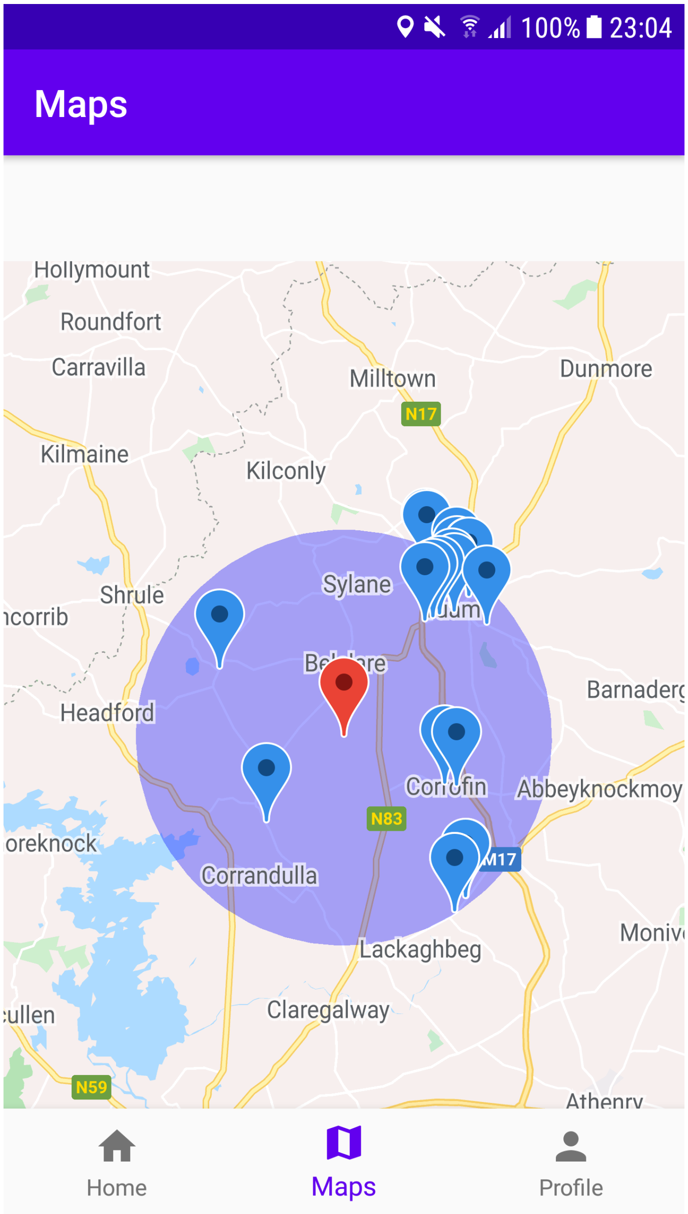
The Dashboard Fragment contains all location based activity here. The user location and setting up the HTTP request to get the filling stations nearby.

```
public void onRequestPermissionsResult(int requestCode,
                                        @NonNull String[] permissions,
                                        @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
                                            grantResults);
        if (requestCode == 1) {
            if (grantResults.length > 0 && grantResults[0] ==
                                    PackageManager.PERMISSION_GRANTED) {
                if (ContextCompat.checkSelfPermission(getContext(),
Manifest.permission.ACCESS_FINE_LOCATION) ==
                        PackageManager.PERMISSION_GRANTED) {
                    locationManager.requestLocationUpdates(
                                            LocationManager.GPS_PROVIDER, 0, 0,
                                            locationListener);
                }
            }
        }
```

```
    }
```

Here I check for the Permissions, making sure the user has accepted the permissions and I have permission to access their location.

Maps The implementation of Google Maps is following the google tutorials. Once the Map has rendered the application gets the user location. Once the Location is got a marker is put on the map and a circle of a radius of 12km is painted on the map. A markerOnClickListener is assigned to all markers that will be put on the map, This will be used later to add fuel prices to the individual Filling Stations.


100% 23:04
Maps
Hollymount
Roundfort
Carravilla
Milltown
N17
Dunmore
Kilmaine
Kilconly
Sylane
Shrule
ncorrib
uam
Be lare
Headford
Barnaderg
Corrofin
Abbeyknockmoy
N83
oreknock
M17
Corrandulla
Moniv
Lackaghbeg
Claregalway
cullen
N59
Athenry
Home
Maps
Profile

” ”

Above is all the filling stations are highlighter in on the map. Once the user location is acquired a method getNearByGasStations is called.

```
 public void getNearByGasStations()
 {

    StringBuilder stringBuilder =
            new StringBuilder
            ("https://maps.googleapis.com/maps/api/place/nearbysearch/json?");
            stringBuilder
                .append("location="+String.valueOf(currentLat)+"
                       ,"+String.valueOf(currentLong));
            stringBuilder
                .append("&radius=10000");
            stringBuilder
                .append("&type=gas_station");
             stringBuilder
                .append("&key="+mContext.getResources()
                    .getString(R.string.googlemapskey));

    String url = stringBuilder.toString();
    Log.i("Url:", "NEarBy " + url);

    Object dataTransfer[] = new Object[2];
    dataTransfer[0] = mMap;
    dataTransfer[1] = url;

    getNearbyPlacesData.execute(dataTransfer);

}
```

This method builder a string for use as a url for the Google Places API. The response is handled in the GetNearbyPlaceData class.

GetNearbyPlacesData This Class handles the JSON response from the API response and implements an Async task.

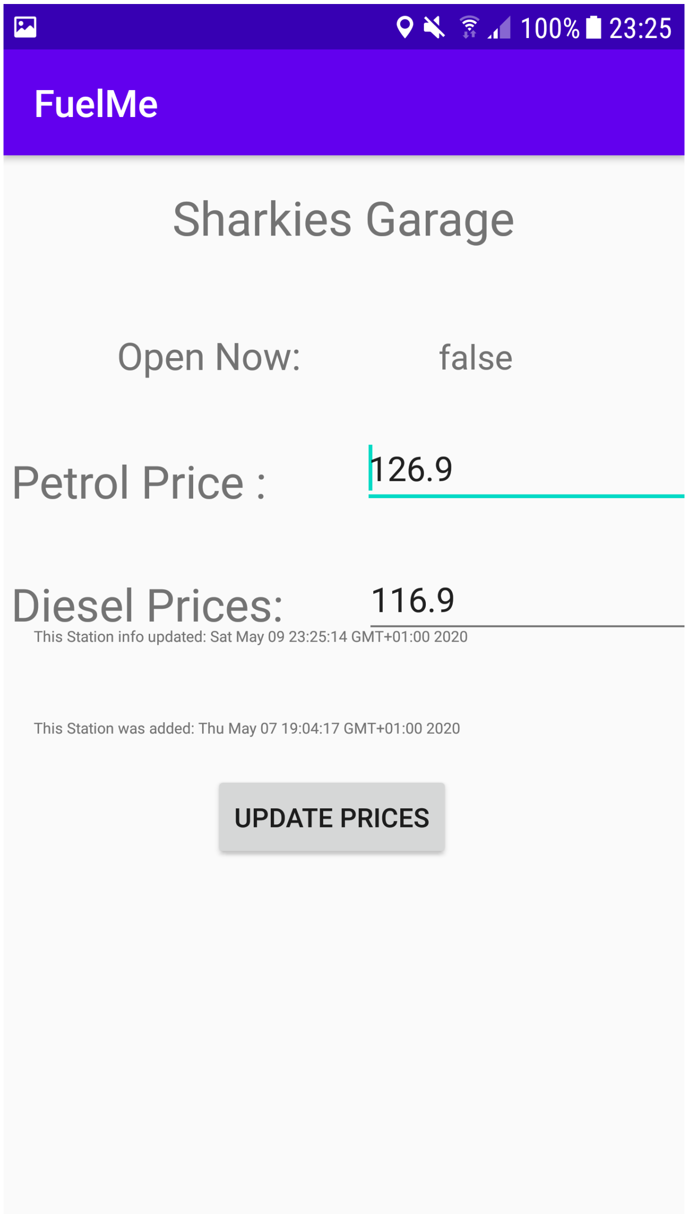
```
 protected String doInBackground(Object... objects) {

        Log.i(”doInBACK”, ”*****”);

        map = (GoogleMap) objects[0];
        url =(String) objects[1];

        DownloadUrl downloadUrl = new DownloadUrl();
        try{
            googlePlaceData = downloadUrl.readUrl(url);
        } catch (Exception e){
            e.printStackTrace();
        }
        return googlePlaceData;
    }
```

This method downloads the JSON response and makes in usable within the class. The onPostExecute method pulls all relevant info from the JSON response and adds to the FuelPrice Parse class. In this response there is a unique place Id, This ID is unique to every business on google maps. This Place ID is how I identify each filling station in the Parse class. After that information is pulled from the response markers are added to the map as pictured above. Once a marker is made clickable by the OnMarkerClickListener in the Dashboard fragment. Once the marker is pressed a second HTTP request is made using the unique place ID to get more information about the selected stations.



FuelMe

Sharkies Garage

Open Now: false

Petrol Price : 126.9

Diesel Prices: 116.9

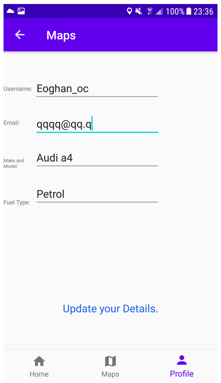
This Station info updated: Sat May 09 23:25:14 GMT+01:00 2020

This Station was added: Thu May 07 19:04:17 GMT+01:00 2020

””

Here is an example of what a Filling Stations profile looks like. The user can update the prices here and view if the filling stations is open. It also shows the user when the station was added to the service and when the information was last updated.

User Profile The user profile allows the user to see all information FuelMe has on on them.

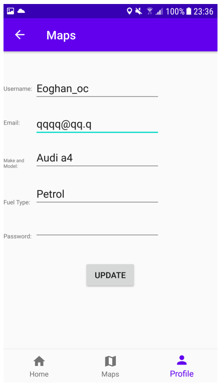
”” 

Username: Eoghan_oc

Email: qqqq@qq.q

Make and Model: Audi a4

Fuel Type: Petrol

”” 

Username: Eoghan_oc

Email: qqqq@qq.q

Make and Model: Audi a4

Fuel Type: Petrol

Password:

Similar to the login page, the Profile page has a dual function it also allows the user to update all details.

System Evaluation

This Chapter will evaluate the project in relation to the requirements set out initially in the early stages of the project. I will also discuss the various types of testing undertaken during the Software Development Life cycle.

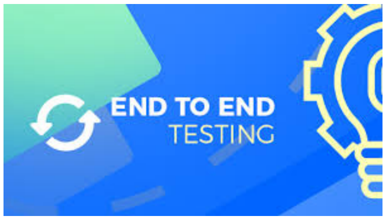
18 Overview

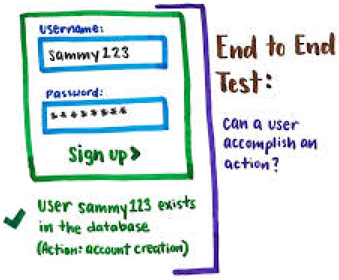
The application was designed and developed using a Test Driven Development methodology. As each part of the project was being written it was tested first out side the application to ensure it worked correctly and if not was adjusted and fixed as required.

Testing was carried out as follow

- End-to-End testing
- Graphical User Interface Testing
- System Testing

19 End to End Testing

””


””

Username:
Sammy123
Password:
****
Sign up >
End to End Test:
Can a user accomplish an action?
User sammy123 exists in the database (Action: account creation)

End-to-end testing is a technique used to test whether the flow of an application right from start to finish is behaving as expected. The purpose of performing end-to-end testing is to identify system dependencies and to ensure that the data integrity is maintained between various system components and systems. The entire application is tested for critical functionalities such as communicating with the other systems, interfaces, database, network, and other applications[5]. Every new feature was tested using this methodology before being committed to the GitHub repository.

20 Graphical User Interface Testing.

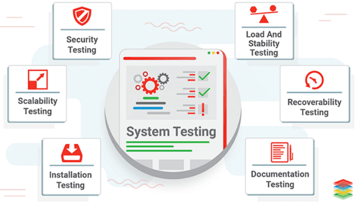
””

””

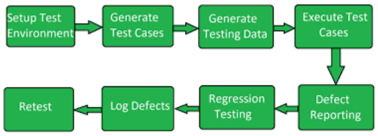
Graphical User Interface (GUI) testing is a software testing type that checks the Graphical User Interface of the Application Under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars - toolbar, menu bar, dialog boxes, and windows, etc. The purpose of Graphical User Interface (GUI) Testing is to ensure UI functionality works as per the specification[11].This form of testing is intended to show many GUI related bugs and allowed me to make the user interface as user friendly as possible.It also changed the navigation style of the app as during development I decided to change to a Tabbed layout rather than the original ”Hamburger Style” side menu.

21 System Testing

””



””



System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems[17].

Conclusion

This Chapter will be serve as the conclusion of this project. Here I will analysis the project objectives and evaluate whether or not I achieved what I set out to do. I will also suggest what I would do differently if i was to do the project again.

22 Objectives

One of the objects of this project was to create an application that allows users to add fuel prices to there nearby filling stations and help consumers make more informed decisions maybe where their money goes.

The application also allows the user to see filling stations in a 12km of their current location and it also allows the user to make an account with the service.

Reflections After spending some time thinking about the project I believe the project standards were met to an acceptable level and I am happy with FuelMe its current form. However all things there are areas of the project I wish I had done differently and ares i wish I had used different technologies.

The positives and the negatives. If I was to start FuelMe again I would change a number of things, I would have used a different type of navigation as I had lots of issues using the fragments as they do not behave the same as a activity classes. I would like to have implemented an optical character recognition to the app which would streamline the adding of fuel prices by simply pointing a camera at the price sign.

Overall With all things considered I am very happy with FuelMe and I enjoyed the experience of developing an application where all ideas and all the work came from myself. I feel a great sense of pride in what I achieved and I am confident what I have learned here will be applicable throughout career.

References

[1] AWS Amazon. *Amazon.* 2014.

[2] *Android.* URL: https://en.wikipedia.org/wiki/Android_(operating_system).

[3] *Commuters.* URL: https://www.irishtimes.com/business/economy/priced-out-home-buyers-drifting-to-dublin-s-commuter-belt-1.4104686.

[4] *Diesel Prices in ireland.* URL: https://www.globalpetrolprices.com/diesel_prices/.

[5] *End to End Testing.* URL: https://www.tutorialspoint.com/software_testing_dictionary/end_to_end_testing.htm.

[6] *Firbase.* URL: https://firebase.googleblog.com/2014/10/firebase-is-joining-google.html.

[7] *Git.* URL: https://en.wikipedia.org/wiki/Git.

[8] *GlobalPetrolPrices.* URL: https://www.globalpetrolprices.com/Ireland/.

[9] *Google Maps.* URL: https://developers.google.com/maps/documentation/android-sdk/intro.

[10] *Google Places.* URL: https://developers.google.com/places/web-service/intro.

[11] *GUI.* URL: https://www.guru99.com/gui-testing.html.

[12] Jordan Novet. *Devs have built 500K apps on Facebook's Parse, with Asia seeing 90% growth in first half of 2014.* Nov. 2015. URL: https://venturebeat.com/2014/12/01/facebook-parse-asia/.

[13] *Parse.* URL: https://parseplatform.org/.

[14] *Petrol Prices in ireland.* URL: https://www.globalpetrolprices.com/gasoline_prices/.

[15] *Platform market share.* URL: https://gs.statcounter.com/os-market-share/mobile/worldwide.

[16] *Sources and methods pdf.* URL: https://www.globalpetrolprices.com/documents/Sources_and_methods_GPP.pdf.

[17] *System Testing.* URL: https://www.guru99.com/system-testing.html.

[18] *The AA.* URL: http://www.theaa.ie/aa/motoring-advice/petrol-prices.aspx.

[19] *The AA Info.* URL: https://en.wikipedia.org/wiki/AA_plc.

[20] *XP.* URL: https://www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1).