

Abstract

“One in four people are now renting their home in Dublin, the highest figure since quarterly records began” (1). This figure emphasises the volatility of the Irish housing market, with more and more people moving into new areas for accommodation.

With so many people now renting, it is unlikely that they know which tradesmen operate within their area. Tradesmen Near Me will be a user friendly application making it easier than ever to connect consumers with their desired tradesman. The aim of this application is to eliminate the stress associated with finding a tradesman to carry out an urgent job. This web application will then act as a contacts directory for future use while providing for offline functionality.

Tradesmen Near Me will also be in the best interest of local tradesmen as it will act as a free advertisement for their services. In an age where very little is free, tradesmen should leap at the chance to have their own page displayed in this application as it will only help to increase business for tradesmen.

The continuous demands from consumers to employ tradesmen will see this application grow in popularity once deployed.

Table of Contents

| | |
|---|----|
| 1. Introduction | 8 |
| 1.1. Project Background..... | 8 |
| 1.2. Project Description..... | 9 |
| 1.3. Project Aims and Objectives | 10 |
| 1.4. Project Scope | 11 |
| 1.5. Thesis Roadmap | 13 |
| 2. Literature Review | 14 |
| 2.1. Introduction | 14 |
| 2.2. Alternative Existing Solutions to Your Problem | 14 |
| 2.2.1 Tradesconnect.ie..... | 14 |
| 2.2.2 Daft.ie..... | 15 |
| 2.2.3 Airbnb..... | 16 |
| 2.3. Technology Review | 16 |
| 2.3.1 Programming Languages..... | 17 |
| 2.3.2 Client-Side Scripting | 18 |
| 2.3.3 Architectural Patterns for Web Applications | 19 |
| 2.3.4 Framework | 20 |
| 2.3.5 Client-Side Language | 21 |
| 2.3.6 Data Persistence | 23 |
| 2.3.7 Cloud Computing Services | 24 |
| 2.4. Geo-fencing Research | 25 |
| 2.5. Existing Final Year Projects | 31 |
| 2.5.2 Project 1: Crime Explorer | 31 |
| 2.5.1 Project 2: FindMyDoggo..... | 32 |
| 2.6. Conclusions | 33 |
| 3. Prototype Design..... | 34 |
| 3.1 Introduction | 34 |
| 3.2. Design Methodology | 34 |
| 3.2.1 Agile..... | 34 |
| 3.2.2 Scrum | 35 |
| 3.2.3 Kanban | 35 |
| 3.3. Overview of System | 36 |
| 3.4. Front-End | 38 |
| 3.4.1 Activity Diagram | 38 |

| | |
|---|----|
| 3.4.2 Use Case Diagram | 39 |
| 3.4.3 Class Diagram | 40 |
| 3.4.4 Front-end Architectural Diagram | 42 |
| 3.4.4 Front-end Prototype Design | 43 |
| 3.5. Middle-Tier | 44 |
| 3.5.1 Consumer Middle-Tier | 44 |
| 3.5.2 Tradesman Middle-Tier | 45 |
| 3.6. Back-End | 45 |
| 3.6.1 Database Design ERD | 45 |
| 3.7. Conclusions | 46 |
| 4. Prototype Development | 47 |
| 4.1. Introduction | 47 |
| 4.2. Front-End | 47 |
| 4.3. Middle-Tier | 49 |
| 4.4. Back-End | 50 |
| 4.4.1 Docker | 50 |
| 4.4.2 Digital Ocean | 51 |
| 4.5. Conclusions | 52 |
| 5. Testing and Evaluation | 53 |
| 5.1. Introduction | 53 |
| 5.2. Plan for Testing | 53 |
| 5.3. Plan for Evaluation | 54 |
| 5.4. Conclusions | 55 |
| 6. Issues and Future Work | 56 |
| 6.1. Introduction | 56 |
| 6.2. Issues and Risks | 56 |
| 6.3. Plans and Future Work | 57 |
| 6.3.1. GANTT Chart | 58 |
| Bibliography | 59 |
| Appendix | 62 |

Table of Figures

| | |
|--|----|
| Figure 1: Screenshot of TradesConnect (4)..... | 15 |
| Figure 2: Screenshot of Daft Search Results (6) | 15 |
| Figure 3: Screenshot of Airbnb Property Location (7) | 16 |
| Figure 4: Great-circle formula | 29 |
| Figure 5: Geofencing Algorithm | 30 |
| Figure 6: Kanban Board Life Cycle (38) | 36 |
| Figure 7: Architecture Diagram for Tradesmen Near Me | 37 |
| Figure 8: Activity Diagram for Consumer Search | 38 |
| Figure 9: Use Case Diagram - Overview of Application..... | 39 |
| Figure 10: Class Diagram of Tradesmen Near Me Front-end | 41 |
| Figure 11: User Interaction with the MVC Architectural Pattern (41)..... | 42 |
| Figure 12: Front-End Prototype Design | 43 |
| Figure 13: Sequence Diagram for consumers querying tradesmen | 44 |
| Figure 14: Sequence Diagram for tradesmen adding data | 45 |
| Figure 15: Database ERD for Tradesmen Near Me | 46 |
| Figure 16: Application's Directory..... | 47 |
| Figure 17: Front-End User Interface | 48 |
| Figure 18: SSL certificate | 49 |
| Figure 19: Implementation of a service worker..... | 50 |
| Figure 20: Back-end configuration files | 50 |
| Figure 21: Docker containers | 51 |
| Figure 22: Created Digital Ocean Droplet | 51 |
| Figure 23: GANTT Chart | 58 |
| Figure 24: Screenshot of Daft Property Location (46) | 62 |
| Figure 25: Front-End Prototype Design - Low Fidelity Prototype | 63 |
| Figure 26: Front-End Prototype Design - Low Fidelity Prototype | 63 |
| Figure 27: Implementation of a Service Worker..... | 65 |

1. Introduction

1.1. Project Background

Tradesmen Near Me will be a web application which enables people to locate various professionals in their area for their particular problem.

As the property crisis continues to grow in Ireland, so does the need for property owners to employ tradesmen in their area who they are not familiar with“ (2). This continuous growth in the number of people living in rented accommodation has resulted in a greater demand for a platform where individuals can search for tradesmen. A platform to facilitate this need would also benefit landlords as they cannot simply know a tradesman for each property location. Despite the advancements in technology in modern times, most people still turn to a friend or family member to recommend a tradesman. Therefore, the standard of work carried out by the suggested tradesman partially depends on your connections. Tradesmen Near Me aims to eradicate this issue as it will provide consumers with the necessary information so that they can make an educated decision as to who to contact. Consumers will be capable of viewing pictures of each tradesman’s previous work, as well as being able to contact fellow consumers who have had work carried out by that same tradesman. Thus, enabling consumers to make an informed decision when contacting their desired tradesman.

The inspiration for this project came from my own experience renting in Dublin. Our apartment developed a serious leak in the summer of this year and after contacting our landlord, he informed us that he and a plumber would arrive first thing in the morning. Upon arrival, I was amazed to see that the landlord had brought a plumber he had known from his own county, Cavan. After completing the job, the plumber was then entitled to the cost the job itself, as well as the cost of the 3-hour round commute. It was obvious from this experience and many others, that people tend to not sway away from their tried and trusted tradesmen. Some people would prefer to pay the extra travelling fee for that added peace of mind, got from dealing with a professional who they have dealt with before. One of

the aims of this application is to provide a safe and reliable option for consumers who would rather employ a local tradesman and benefit financially in the meantime.

Another issue of existing solutions for contacting and employing a tradesman is the response time between initially contacting a tradesman, and them arriving on the scene. For those in need of immediate aid from a tradesman such as a plumber, they will have to fill out a form detailing their job to be carried out. Once completed, they must then wait for an available tradesman to contact them. This response delay could prove detrimental, depending on what job needs to be carried out. Another aim of this application is to improve the efficiency of connecting a client and a tradesman.

1.2. Project Description

A simple, user-friendly web application will act as a medium between tradesmen and customers. A user will be able to login as either, a consumer or a tradesman. It will allow tradesmen to advertise their services through a website which will only help to increase their business. These professionals will have the opportunity to advertise their services in the form of a web page, containing any other helpful information to the user. Tradesmen Near Me will display to the user, the diameter in which each individual tradesman is willing to travel to, to carry out a job. Not only this but it will give them an opportunity to display pictures of their previous work as well as their contact information. Users will have the opportunity to save a tradesman into their favourites tab in order to be easily accessed again for offline use. This project will take the form of a progressive web app (PWA) and will contain various web page headers, allowing the consumer to query the tradesmen near them as well as retrieving their favored tradesmen. Favored tradesmen will then be stored in the appropriate tab for future use. Storing a tradesman in favourite results in it being stored as cache on the device being used. This will be accomplish using a service worker and will allow for offline use.

Tradesmen Near Me will provide the consumer with the option to filter the results based on the attributes they would like prioritized. A user will have the option to login as a customer or as a tradesperson while the administrator will monitor the data displayed.

This project will be implemented as a progressive web app, allowing the user to access it cross platform, therefore, enabling reaching the widest possible user base. Users will be able to acquire the web app on any device, mobile or otherwise. To cater for the largest feasible target market, Tradesmen Near Me will incorporate the use of a service worker which will let the user access their desired contact information offline.

This application will be launched in the coming year so that its widely used by both, tradesmen and consumers. The application will then be evaluated by applying comprehensive usability testing. This involves users continuously digging into the UI of the app to identify any usability issues.

1.3. Project Aims and Objectives

The overall aim of this project is to build a secure application which connects consumers and tradesmen. This will be achieved by developing a user interface as the application's front-end such that it connects to the necessary components of the application. Once the project is completed, the following objectives will be present in the application.

1. The overall objective of this application is to act as a medium between consumers and tradesmen so that it displays the necessary contact information from tradesmen to consumers.
2. A back-end will be developed in order to store the necessary tradesmen attributes such as their maximum travel distance and contact information.
3. A scalable front-end will be implemented with consumer satisfaction in mind. The front-end will supply a login option for either, a consumer or a tradesman.
4. Middleware will take the form of Fetch API calls in order to query the required data while also connecting the front-end and back-end of the application.
5. A cloud provider will be implemented alongside container software in order to make Tradesmen Near Me deployable from one computing environment to another.

1.4. Project Scope

There are two main areas of development within this application, these include the application's front-end and its back-end.

1.4.1 Front-end

The front-end of this application will take the form of a progressive web application (PWA), providing users with the option to install the web app on their home screen. Building an application as a PWA gives it the look and feel of a native app while also providing offline functionality.

Login & Register

Upon visiting this web application, users will be able to search for tradesmen without needing to register or login. Users can register an account as either a consumer or tradesman, depending on the preference. Registering an account will be needed if a consumer decides to favourite a tradesman, or if a tradesman decides to add information to their page.

Searching for Tradesmen

A user will have the ability to search for their required tradesman. After the user has selected their desired tradesman, they will have the option to filter their search based on their requirements. Users will be given the option to filter their search based on the tradesman's working hours, pictures uploaded and whether their business has a website or not. The objective of this process is to be as user friendly as possible.

Favouriting a Tradesman

Once a consumer has searched for their desired tradesman, they will have the ability to favourite the returned tradesmen. Before favouriting a tradesman, the consumer must login or register. Favouriting a tradesman will then see it appear in the appropriate favourites tab within the application. The role of the favourites tab is to provide consumers with their desired tradesmen information without the need to search again. The provided information will also be available for offline use.

Adding Tradesmen Information

Once a user has logged in as a tradesman, they will have the opportunity to add relevant information to be displayed on their page. This information will be sent for approval from the site's administrator and once approved, it will be displayed on their page for consumers to view.

1.4.2 Back-end

The back-end of this application will centre around the maintaining of database entries. It will involve handling the verification of a user, searching for tradesmen and adding tradesmen information.

Credential Verification

When a consumer attempts to favourite a tradesman or a tradesman attempts to add information to their page, there will be a verification process in place that ensures only registered users may perform such actions. This guarantees that only valid users will be able to persist changes within the application.

Searching for Tradesmen

When a consumer searches for a tradesman, the relevant query will be made to this application's database using the consumer's selected filters. This will see the search iterate over the application's stored tradesmen so that it returns the correct results.

Adding Tradesmen Information

If a tradesman adds to their page's information, it will be sent for administrator approval. This process will not be immediate, but it is essential to the providing of accurate information to the consumer. After the admin has approved the information, it will be instantly displayed on their page. The more information a tradesman adds to their page, the more chance they will have of being viewed by consumers.

1.5. Thesis Roadmap

This section will provide an overview of what the following chapters are about.

2. *Literature Review*

The purpose of this project's Literature Review is to investigate existing solutions to the problem that this application attempts to overcome. It also explores how functions of these existing solutions can be incorporated into this application using the researched technologies.

3. Prototype Design

This chapter outlines the development approach to be taken during the project. It accomplishes this by explaining the chosen methodology, and through the designing of various UML diagrams.

4. Prototype Development

This section provides a detailed overview of the development process to date, as well as the issues faced throughout the process.

5. Testing and Evaluation

The aim of this chapter is to define the plan for the application's testing and evaluation going forward.

6. Issues and Future Work

Within this segment, the issues faced during the development of the application are discussed and the future work to be carried out on the project is outlined.

2. Literature Review

2.1. Introduction

The following section expands on the research carried out on other existing solutions to the problem that this application helps solve. It also discusses the possible technologies used to implement the desired application.

2.2. Alternative Existing Solutions to Your Problem

This section will cover the key topics of research that were instrumental to this project. The studies carried out involve numerous existing applications that currently help tackle the problem that my project aims to overcome. In addition to this, web applications which produce a similar end product to that of my desired project were studied. Individually, each of the researched applications offer partial functionality towards my project domain, acting as an efficient medium between tradesmen and customers. Three useful existing solutions evaluated include Tradesconnect, Daft and Airbnb.

2.2.1 Tradesconnect.ie

Tradesconnect.ie is a web application which enables the consumer to contact a tradesman based on the company's location, reviews and photos. After the user fills out a form, detailing their job requirements, Tradesconnect "will automatically send it on to our relevant verified Tradesmen" (3). Although, this method of communication between the customer and tradesman is inefficient, it allows the user to get a quote from the tradesman before deciding to proceed with the job. This product provides reviews, photos and a rough graphical location of the tradesman's company to user. It does however, fail to incorporate tradesmen who are not associated to any company. This application works well for people who do not require an urgent job to be carried out. The form and geolocation functionality implemented in Tradesconnect can be seen in figure 4.

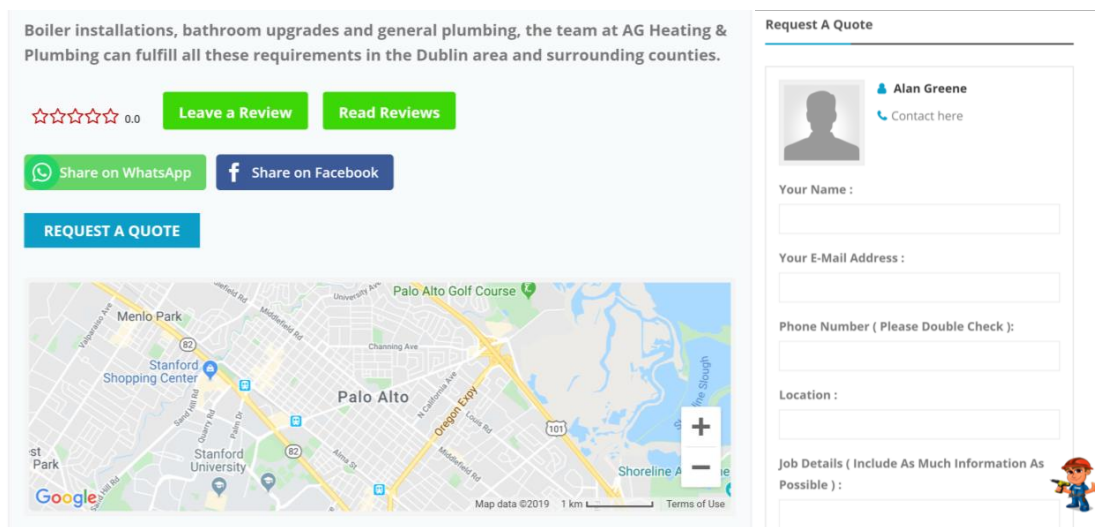


Figure 1: Screenshot of TradesConnect (4)

2.2.2 Daft.ie

Daft is web application, founded in 1997 and is aimed at connecting professionals and property owners. Daft has risen substantially since 1997, boasting an “audience of over 2.5 million users each month” (5). Daft is a complex app, combining both, geo-boundaries of properties and a recommendation sorting algorithm to display a user interface. Tradesmen Near Me aims to incorporate a similar process to display the tradesmen near a particular location, based on the consumers inputted filters. A sample of Daft’s returned results and filtering system can be seen in the figure below as well as figure 7.1 in the appendix.

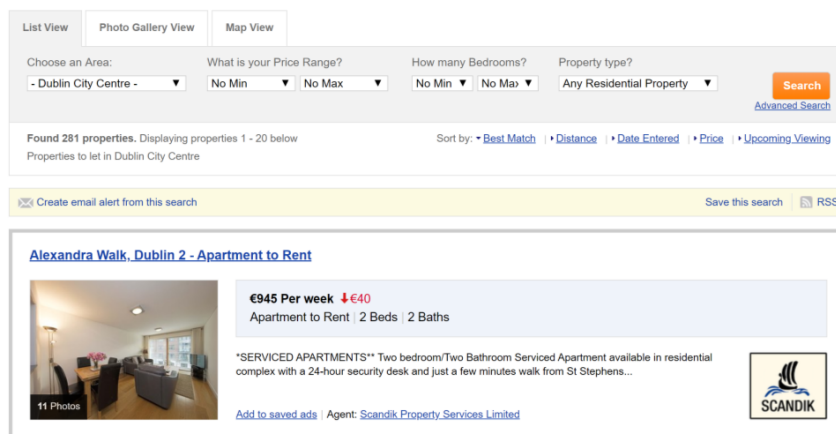
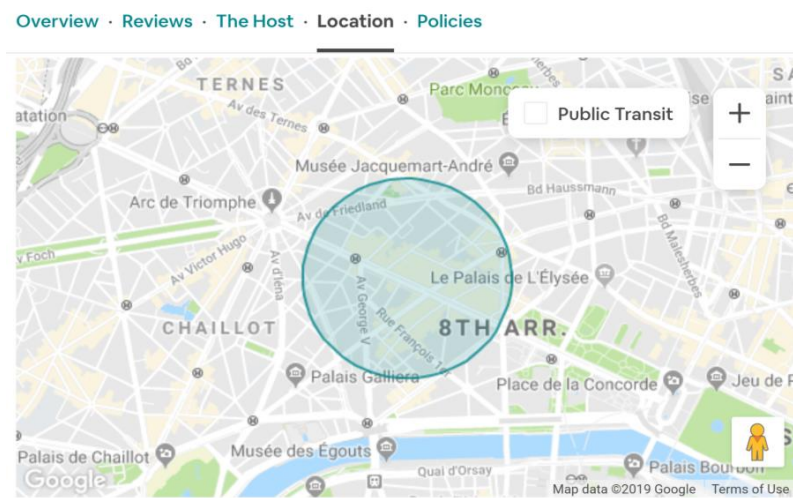


Figure 2: Screenshot of Daft Search Results (6)

2.2.3 Airbnb

Airbnb is an online marketplace, connecting users and home owners across the world. Airbnb acts as a medium for arranging accommodation and tourism experiences. “It provides access to 7 million unique places to stay in more than 100,000 cities and 191 countries and regions” (7). Similar to daft, Airbnb allows the user to simultaneously view the details and the location of the property through a graphical interface. As well as this, they implement a recommendation search algorithm to return properties best suited to each individual user. Upon selecting a property to view, the user is provided with an area as to where the property is based. It is this, location-based user interface that this application aims to achieve.



Exact location information is provided after a booking is confirmed.

Figure 3: Screenshot of Airbnb Property Location (7)

2.3. Technology Review

There are a multitude of technologies to implement a progressive web application. Below are a sample of the technologies researched and selected for this project.

2.3.1 Programming Languages

There are numerous programming languages ideal for web application development. For this project, PHP, Node.js and Python were researched, each of which make valid arguments to implement Tradesmen Near Me. The subsequent passages will discuss these three languages in depth.

PHP is a server-side programming language, used to develop web applications. PHP provides the foundations to enable a user to build a high-performance reliable website while maintaining relatively low development costs. PHP is very fast loading language which is compatible on various platforms. PHP is one of the most popular programming languages among website development, but unfortunately it was not chosen as this application's programming language.

Node.js is a runtime environment, written in JavaScript. It was developed in 2009 and has grown in popularity year on year. Node.js offers a higher performance than most other programming languages as it executes its code using Google's V8 JavaScript engine. Node.js allows the user to access "a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent" (8). Another key advantage of using Node.js is that it can be implemented for both, the front-end and back-end of the project, saving valuable time needed to learn another language as a result. However, a major flaw of using Node.js is that it's API is constantly changing which in turn creates huge problems for developers as the changes are not backward-compatible. Despite the initial time management benefits, Node.js will lose developers time when the API changes. As time management plays such a pivotal role in this project, Node.js was dismissed as a potential technology to implement.

One of the main factors considered when choosing a programming language is the user's prior experience in each language. Having previous experience in Python played a central role in the selecting of it as this project's programming language. Python has been one of

the most popular programming languages since its first version was published in 1991. It's consistent popularity over the last decades is thanks in many ways to its extensive features made available to the user. In addition to supplying a large standard library to the user, "the Python Package Index (PyPI) contains numerous third-party modules that make Python capable of interacting with most of the other languages and platforms" (9). The vast advantages affiliated with Python make it the ideal programming language for this project.

2.3.2 Client-Side Scripting

Vanilla JS "is the most lightweight framework available" (9), and involves the use of JavaScript without any additional libraries made available by JavaScript. These libraries include jQuery which is aimed at making JavaScript more efficient. Vanilla JS is substantially faster than any other scripting framework, allowing the user's browser to load Vanilla JS into memory before it even requests the website. Companies who incorporate Vanilla JS into their technologies comprise of Facebook, Google and Netflix.

A scripting framework however is a platform to aid in the development of software applications. It provides the structure on which software developers can build their application on. This pre-defined structure enables developers to streamline the development cycle of a project. "You can expect to build a project in much less time than would be achieved writing code without a framework" (10). There are endless advantages to implementing a scripting framework but the role it plays in integrating the front-end of an application to its back-end, cannot be overlooked. Any large web application relies on an efficient connection between stored data, typically stored in a database, and the user's interface. It is this efficiency and reliability which makes a scripting language essential for this project. After deciding upon the use of a scripting framework for this project, the next logical step was to choose an architectural pattern in which to base the framework of.

2.3.3 Architectural Patterns for Web Applications

MVP, MVVM and MVC are a sample of the most used architectural patterns in application development. They are all patterns which provide a “reusable solution to a commonly occurring problem in software architecture within a given context” (11). Below I will cover the benefits of each pattern and which best satisfied the demands of this project.

The Model-View-Controller (MVC) pattern “divides an application into three major aspects: Model, View and Controller” (12), each component, being built to handle specific development functionality. The Model represents the user interface, consisting of a collection of classes which display the view. Its controller behaves like an interface which allows the Model component to interact with the View component. The controller is an essential part of the MVC pattern as it handles the incoming of requests and the manipulation of data in order to display the View. Implementing the MVC pattern offers a faster development process as well as faster debugging times. Combining this with the fact that the Model limits the need for formatting data, makes it very hard to look past MVC as this project’s architectural pattern.

Model-View-ViewModel is a software architectural pattern used to implement bi-directional data within views. This enables the automatic syncing of changes of properties in the View-Model. This is executed through the use of binding data between View and View-Model. “It is responsible for exposing methods, commands, and other properties that help to maintain the state of the view, manipulate the model as the result of actions on the view, and trigger events in the view itself” (12). MVVC is focused towards separating the user interface part of the application from the related code as well as allowing different layers to be reused. It also provides easier testing through separating the user interface and application logic. These attractive benefits make the MVVM an appealing choice as this project’s architectural pattern.

The Model-View-Presenter (MVP) pattern lets you “decouple business logic (Model) from view logic (Activity/Fragment) by introducing an intermediary called as Presenter” (13). It is derived from the foundation of the MVC pattern and serves to divide the patterns responsibilities over three components. The view oversees the displaying of the user interface, the presenter is in charge of communicating between the view and the model, while the model is responsible for the handling, changing and manipulating of the business logic. With MVP, the view and presenter communicate with each other via an interface, making the implementation of unit testing easier.

After much deliberation, MVC was decided upon as the chosen architectural pattern for this project as it is an ideal pattern to implement a large scalable application. As well as this, the easy separation of business logic promotes clean, maintainable code. These two benefits to using the MVC pattern make it the perfect architecture on which to implement Tradesmen Near Me.

2.3.4 Framework

For this application, frameworks such as Express, Laravel and Django were studied, each being viable options on their own accord.

Laravel is an open sourced PHP framework designed to make web application development more efficient. This is achieved by containing many built-in libraries. A secure authorization system is integral to any modern web application and as Laravel “provides a simple way to organize authorization logic and control access to resources” (14), it presents an appealing case as the framework to be implemented in this project. On top of this, Laravel also provides pre-configured error handling for any new project. Therefore, Laravel makes a compelling argument to be the framework of choice for this project.

Express is a Node.js web application framework aimed at minimalizing the code needed to execute tasks. It enables the user to shorten the development time of a project in several

ways including that Node.js can be implemented for both, the front-end and back-end of a web application. Express' benefits can be utilized in applications requiring a great amount of user input handling, and it is precisely this reason that companies such as Uber chose Node.js for their runtime environment. As time management is such a crucial aspect of this project, it is hard to look past the many benefits associated with the implementation of the Express framework.

Django was ultimately chosen as the framework to deploy this application. This choice was based on the severe multitude of advantages associated with it. Django is written in Python and “provides code for common operations like database manipulation, HTML templating, URL routing, session management, and security” (15), thus improving development time. Django is one of the most compatible frameworks available, allowing users to develop using most popular operating systems such as Windows, MacOS and Linux. In addition to this, Django contains built-in security features to help protect against unwanted security attacks. Django adheres to the model-view-controller (MVC) design, often referred to as MVT as its main components are the layers, model, view and template. Having a detailed structure for a framework, allows the developer to easily separate the different components of an application. GeoDjango is a Django application, essential for developing a Geographic Information System (GIS), and it was this additional functionality which made Django the best suited framework for this project.

2.3.5 Client-Side Language

A client-side programming language refers to the language implemented on the client's browser to make up the front-end of a web application. There's a huge selection of client-side languages to pick from but for this section, AngularJS, React and JavaScript will be discussed.

AngularJS is a client-side language which has seen itself grow significantly in popularity in the last number of years. “AngularJS lets you extend HTML vocabulary for your application.

The resulting environment is extraordinarily expressive, readable, and quick to develop” (16). It also enables the user to create a clean and maintainable single page application using reusable components and unit testable code. Angular is compatible with all popular browsers, making it a safe choice for a client-side language. However, Angular would fail to satisfy the requirements of this project as it not as secure as its competitors and is predominately used for the development of large single page applications.

React is a flexible JavaScript library implemented in order to speed up the development cycle of a user interface. “It lets you compose complex UIs from small and isolated pieces of code called ‘components’ ” (17). React offers the user a chance to design simple views for an application which it will in turn update and render the appropriate components when the data changes. On one hand, the continuous development of ReactJS results in a highly efficient language but one in which the speed of development forces developers to continuously relearn new methods of implementing things. Combined with poor documentation for the rapidly growing technologies, React was not chosen as a client-side language.

JavaScript is one of the most popular client-side scripts as “nearly every site’s front end is a combination of JavaScript and HTML and CSS” (18). The advantages of implementing JavaScript are endless, with it offering a fast scripting language which can be run immediately within the client’s browser. Furthermore, its simple syntax is easy to learn and implement. These factors has seen it become more popular in the industry in previous years. Additional JavaScript libraries such as jQuery and RxJS will also be incorporated within this project, allowing for the development of more efficient code. jQuery is designed to simplify event handling as well as allow for HTML DOM tree manipulation, while RxJS “is a library for reactive programming using observables that makes it easier to compose asynchronous or callback-based code” (19). Both, jQuery and RxJS are lightweight libraries which enable the developer to condense many lines of JavaScript code into methods that can be called using a single line of code. Having previous experience in JavaScript, as well as

the benefits mentioned, made it the only logical option as this project's client-side programming language.

2.3.6 Data Persistence

With the idea of implementing a progressive web app in mind, choosing a reliable database is extremely important in the running and maintaining of the system. There is a vast array of database technology to choose from when implementing a web application, a sample of which being Oracle, MySQL, Firebase and PostgreSQL.

Oracle Database is a collection of data, used to store, access and retrieve information through the use of a database server. "In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data" (20). Despite the numerous advantages associated with the use of Oracle, it was not selected as this project's database. This was due to two factors, one being the cost of running Oracle and the other being the complexity that it brings with it. Oracle was the most expensive data persistence option researched and therefore was not best suited to this particular project.

Another database studied before the implementation of this project was MariaDB. "MariaDB was initially developed in direct response to Oracle acquiring MySQL. MariaDB offers you an opportunity to "break free from the costs, constraints and complexity of proprietary databases and reinvest in what matters most, developing innovative applications and services as fast as possible" (16). MariaDB offers frequent security releases as well as being open sourced and easily compatible with MySQL. MariaDB is a solid choice to use for this project but it ultimately was not chosen because of its difficulties with fast reliable caching.

After extensive research, PostgreSQL was decided upon as this projects database. PostgreSQL supports several programming languages such as Python, C, Java and Perl and as

well as this, it “supports different kinds of techniques for geographical data storage such as PostGIS, Key-Value Store and DBLink.” (21). PostgreSQL is a free, open-sourced database which enables you to use, manipulate and implement it as you see fit. As well as this, PostgreSQL supports the storage of both, geographical and JSON objects, making it ideal for a location-based web application. In addition to this, PostgreSQL provides a geographical extension called ‘GIS’ which supplies built-in functions to handle geographic data in different formats. Tradesmen Near Me plans to be implemented using both, the programming language, Python, and the geographical data storage, PostGIS. Therefore, choosing PostgreSQL became an obvious choice.

2.3.7 Cloud Computing Services

Before deploying this project, a cloud provider must be chosen on which to host the application. The researched cloud computing services include Amazon Web Services (AWS), Google Cloud Platform and Digital Ocean.

“Amazon Web Services (AWS) is the world’s most comprehensive and broadly adopted cloud platform” (22). AWS Elastic Beanstalk is a service offered by AWS, providing a platform for deploying and scaling web applications. The documentation for the implementation of an AWS service is very clear and provides detailed tutorials to get started. There are many advantages associated with the implementation of AWS services which include having no commitment to any contract, having built-in security infrastructure and allowing your web app to be completely flexible. There are few issues associated with the use of AWS, one being that service limits are set by the platform so that it can prevent uncontrolled resource usage. Although, AWS was eventually dismissed as a potential cloud provider for this application due to the investment in time needed to fully understand and implement the extensive features provided.

Google Cloud Platform includes a range of cloud computing services which provide “a web-based, graphical user interface that you can use to manage your GCP projects and

resources” (23). As well as supplying useful documentation, Google provides the user with the same security model for their application as they have currently securing YouTube and several other products. These extensive features don’t come cheap however, with fees for both, support and downloading data, thus making it unsuitable for this project.

Digital Ocean was this application’s eventual choice as a cloud platform. Digital Ocean provides many advantages such as daily data backups and end-to-end security. “Managed Databases run on enterprise-class hardware and SSD storage, giving you lightning-fast performance” (24). The only compromise with Digital Ocean is that the developer must source their own SSL certificate and domain name, but these can be obtained using other methods. Digital Ocean also provides students with \$50 of free credit so that they can start deploying their application as soon as possible. The previously outlined benefits make Digital Ocean an smart choice as this application’s cloud provider.

2.4. Geo-fencing Research

The complex features of this application arise from its filtering-based search algorithm and through its use of geo-fencing. Additionally, this project will be developed as a progressive web app. To ensure that these features are integrated seamlessly into the web app, the following research was carried out.

A filtering tool in any given website has traditionally been a left-hand vertical sidebar. However, over the last number of years, a horizontal sorting toolbar has slowly grown in popularity. This is because a horizontal filtering toolbar can limit two issues frequently caused by a traditional sidebar. One being that users fail to notice the filtering sidebar and the other being that users can mis interrupt the site’s sorting tool as the site’s filtering toolbar. “Both issues are severe as it prevents users from getting a well-defined product list which match their purchasing criteria - and instead leaves them navigating overly broad product lists” (25). This would then cause a tired user who has been searching an unnecessary broad list of results, to navigate to a competitor site. This being the very thing

that a clean and efficient user interface is designed to prevent. “Despite the initial plaudits for designing a horizontal sorting toolbar, it does have its drawbacks. The main drawback being that it can quickly become cluttered if there are too many filtering options. A horizontal toolbar is the optimum choice for a filtering tool when the application has a small number of filters and that the number of these filters will not change as the complexity of the application grows. As this application requires only four filtering options, it is within its best interest to implement a horizontal toolbar used for both, filtering, and sorting.

Many types of websites share searching functionality, with each being tailored to that specific website. This searching functionality is referred to as a ‘full text search’ and involves storing data as an array of objects in a JSON file which can be iterated over to find the appropriate matches. As the need for an efficient, accurate full text search grows, so does the available options for implementing it. There are now many ways to carry out a search, with this section concentrating on the use of a back-end search engine, commercial search engine and a database with a built-in search engine.

Benefits of using a back-end search engine such as Elasticsearch include that it can handle queries very fast thanks to its process of data ingestion, in which “raw data is parsed, normalized, and enriched before it is *indexed* in Elasticsearch” (26). In addition to this, Elasticsearch can be easily scaled, allowing users to expand resources. In spite of these initial advantages, back-end search engines were deemed unsuitable for this project due the expensive cost of setup and maintenance.

Another possible way to perform a full text search is by using a commercial engine such as Alogia or Amazon Cloud Search. Commercial engines can simplify the management process of setup, maintenance and updates, with Amazon Cloud Search supporting “features such as highlighting, autocomplete, and geospatial search (27). Although Amazon offer an extensive list of features, these features can result in additional costs, giving it the potential to become expensive. As well as the financial drawbacks, Cloud Search’s server-side relies on

connectivity and battery usage to operate. Despite this, commercial engines such as Amazon's Cloud Search present a solid case as this projects method of implementing a full text search.

The chosen method to implement this project's searching functionality was to select a database with a built-in search. Databases which support full text search comprise of MongoDB, CouchDB, MySQL and PostgreSQL. Selecting a database with full text search saves a substantial amount of time and money incorporated in the setting up and maintaining of a separate search engine. Thus, shortening the developing time needed to execute a search.

After a user searches for tradesmen near them, they will be greeted with a graphical representation of available tradesmen in their area. This graphical representation will be performed using geo-fencing, which is a "location-based service software that uses GPS, Wi-Fi, Cellular Data, RFID (Radio Frequency Identification) to trigger a precoded action when an RFID tag or mobile device enters the geofence or the virtual boundary that is set up around a geographical location" (28). The software implemented to measure the user's location depends on the positioning of the user. If the user is outdoors, GPS will be used where available, while indoors will involve the use of cellular and Wi-Fi data. The Global Positioning System (GPS) is a network of satellites orbiting Earth whereby the three nearest satellites to the requested GPS position send messages to each other in order to calculate exactly how far each satellite is away so that they can isolate the needed location.

In previous years, the accuracy of GPS has been continuously improved. This can be seen with the EU project, Galileo, "which is only partially complete is intended to be more advanced than any other system in the sky at the moment with position measurements within 1-metre precision and better positioning services at higher latitudes than other positioning systems" (29). Despite this promising information, Galileo's new system is not scheduled for completion until 2020. In addition to Galileo, GPS chip manufacture,

Broadcom, are developing a new chip for smart phones which will see it get a location within an accuracy of 30 centimeters using the existing US GPS network. Despite the development of this instrumental GPS technology, it will not become commercially available until 2022.

GPS is the most accurate method of retrieving a location, however, for indoor environments, buildings and walls can make for an inaccurate reading. As a result, geofencing uses a combination of cellular and Wi-Fi data for indoor use. The accuracy of the retrieved user location depends on the type of environment they are in, with urban environments reaching an accuracy of 100-200 meters and rural environments spanning an accuracy of several hundred meters.

This application will also pose an alternative method for users to retrieve their location which does not involve the use of GPS, cellular or Wi-Fi data. It will take the form of a text field in which users will be able to type or select their Eircode. This will in turn query an Eircode API to return the users location. An appropriate link to an Eircode finder will be presented to any user unable to recall their Eircode. Therefore, this application will cater to all users regardless of their GPS technology.

Tradesmen Near Me will involve the storing of tradesmen information as an array of JSON objects which will then be converted into a specific location and displayed to the screen. Following a user search, each tradesman returned, will first have its longitude and latitude stored, then using these attributes, they will have their specific location marked. After this, each tradesman's location, as well as the distance they're willing to travel to, will be used to display each appropriate circle diameter in which each tradesman is available.

This application will implement an algorithm to check if the user's location falls into the specified maximum distance each tradesman is willing to travel to, and based on these results, Tradesmen Near Me will display the relevant tradesmen to the consumer. This will

be accomplished by following a defined number of steps within a selected geo-fencing formula.

The formula that will be implemented in this project will be The Great Circle Distance formula. “The Great Circle Distance formula computes the shortest distance path of two points on the surface of the sphere” (30). This formula will involve using the latitude and longitude in radians of two GPS coordinates, their absolute differences and the angle between them. The formula can be expressed in Python code as shown in figure 4. This formula is explained in greater depth in the article, ‘Great Circle Distances and Bearings Between Two Locations’ by R.Bullock (31).

```
def great_circle(lon1, lat1, lon2, lat2):  
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])  
  
    return 6371 * (  
        acos(sin(lat1) * sin(lat2) + cos(lat1) * cos(lat2) * cos(lon1 - lon2))  
    )  
  
great_circle(59.9, -30.3, 37.8, 122.4)
```

Figure 4: Great-circle formula

After the consumers location has been obtained, the next step is to manipulate a county dataset for the whole of Ireland in order to find what county the user is located in. The isolating of the desired county, allows the process to be executed faster, thus, improving the overall speed of the application. After this, the remaining of the algorithm will be iterated over for each individual tradesman within the consumers acquired county. This iteration over stored tradesmen enables multiple tradesmen to be found within the given boundary. In order to compare the distance to the consumer, with a tradesman’s pre-defined travel distance, the application reads in the tradesman’s location as well as their desired travel distance. These server-side values will be derived from querying an established Postgres database for the relevant data. Using these values, the distance between the tradesman and the consumer can now be measured and compared to that of the tradesman’s desired distance of travel. Once compared, the tradesman can now be displayed if the consumer is within the identified diameter, as seen in the figure below.

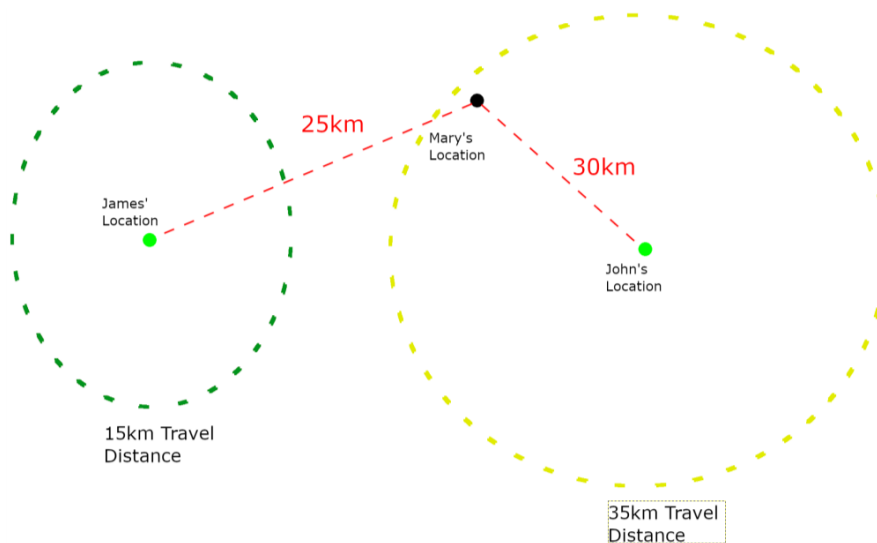


Figure 5: Geofencing Algorithm

The geo-fencing algorithm implemented in this application can be explained through the diagram illustrated above. In this example, Mary is the consumer searching for a tradesman in her area while James and John are both tradesmen in her county. Here, it can be seen how the distance between Mary and John, is first measured, then compared against the distance that John is willing to travel to. In this case, the consumer's location lies within John's specified maximum distance of 35km but fails to lie within James' desired distance of 15km. Subsequently, John would be the only tradesman displayed to the user.

In order to present a clear, user friendly interface, this project will incorporate the use of Leaflet and Bootstrap. "Leaflet is the leading open source JavaScript library for mobile-friendly interactive maps" (32), allowing for the creation of a map and location markers. Despite only weighing around 38KB, Leaflet provides an extensive array of mapping features, making it the ideal mapping technology to implement Tradesmen Near Me's intricate geo-fencing. Bootstrap will also be deployed along with it, in order to help tailor the application towards mobile use. Bootstrap enables the fast development of a project as well as being "equipped with responsive layout and 12-column grid system that help dynamically adjust the website to a suitable screen resolution" (33). It is extremely

important that Tradesman Near Me is in fact responsive so that it allows for the widest user base possible.

When a user visits Tradesmen Near Me, they will be met with the option to install the web app on their home screen. Thus, giving this web application the appearance of a native app, traditionally downloaded through an app store. To give the overall look and feel of a native app, this application will be developed as a progressive web application (PWA), providing users with offline functionality. To allow for this, a service worker will be integrated into the application, which enables control over network requests. A service worker lets the developer control what data to cache when online and provides the user access to this same content once offline. In this project's case, the users favoured tradesmen's information will be the cache data. Therefore, permitting users to access this information regardless of their network status. The popularity of progressive web applications has steadily grown over the last several years and despite being in their infancy "they have the potential to create a shift in the way the web works" (34).

Each of the technologies described above, are individually, comprehensive components to implement, but the real challenge will see each technology being incorporated together into one user-friendly, responsive progressive web application.

2.5. Existing Final Year Projects

2.5.2 Project 1: Crime Explorer

Student: Deividas Savickas

Description (brief):

Crime Explorer is a web application which aids to allow users to view crimes instantly. The user will be able to filter the crimes based on a region, time and crime type. This returned information is then displaying using a geographical visualization.

3. Prototype Design

3.1 Introduction

This following chapter provides detailed research into the software methodologies implemented within this application. It will also cover the early designing stages of the project along with high fidelity prototypes of the user interfaces.

3.2. Design Methodology

A design methodology outlines the framework implemented for the development lifecycle of a particular system. Implementing a design methodology ensures that the developer or team follow a set defined structure to provide an efficient project delivery. Methodologies are more advantageous for large scale development but also can be used in any project, large or small to maintain quality throughout the development process. To ensure that chosen methodology was best suited to this project, the following research on existing methodologies was carried out on Agile, Scrum and Kanban.

3.2.1 Agile

"Agile methodology is a practice that helps continuous iteration of development and testing in the SDLC process" (35). Agile enables developers to split the development stages of a project into smaller builds called sprints so that the development is concurrent. Sprints are short time periods whereby the development team work together to complete a defined amount of work. These short sprint cycles allow developers to continuously refine the project while maintaining quality. For Agile, customer engagement is a high priority with emphasis on continuous delivery between the development team and the customer. This continuous customer development helps in the development of a high-quality product which adheres directly to the customers vision. Within Agile, there are various forms of methodologies that can be applied to a project, two of which are Scrum and Kanban.

3.2.2 Scrum

“Scrum is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks” (36). Scrum is the most popular agile methodology implemented as “70% of software teams use Scrum or a Scrum hybrid” (36). With Scrum, the features of a given project are broken down into sprints lasting for one to four weeks. At the beginning and end of these sprints, there are meetings with the scrum master and development team, during which tasks are first assessed from the previous week, then assigned for the coming week. During these meetings, any issues obtained during the sprint are brought to the team’s attention. The sprints carried out during a project help ensure that customer deadlines are met. The timeboxed iterations associated with Scrum improve the overall quality of the project, with the continuous iterations of the project improving debugging times. Despite the numerous plaudits for the use of Scrum as a software methodology, the reason it was not chosen as this applications methodology was because it works best within a small team environment consisting of at least three people. This application however, is to be developed individually and therefore is not best suited to applying Scrum methodology.

3.2.3 Kanban

“Kanban is a method for managing the creation of products with an emphasis on continual delivery while not overburdening the development team” (37). In a Kanban process, there are cards called Kanban that move through the process depending on the stage of development. These cards are assigned to a Kanban Board, moving from start to finish as the task it refers to gets implemented. An example of a Kanban Board can be seen in the figure below.

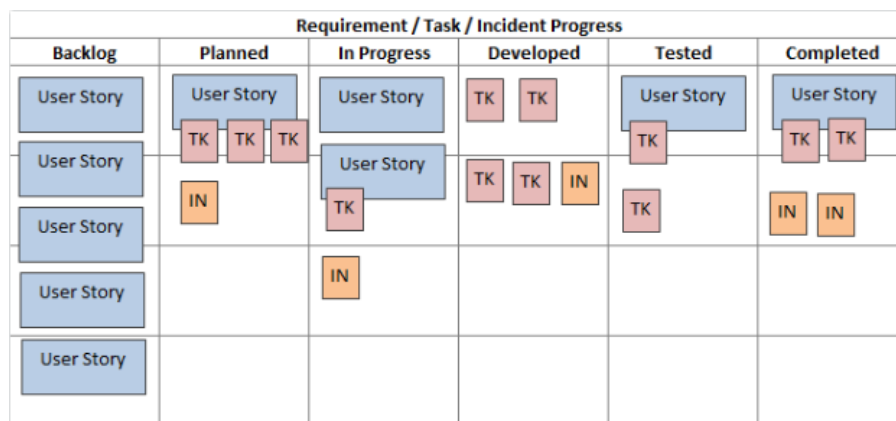


Figure 6: Kanban Board Life Cycle (38)

Here, it is clear to see the development stages of the various tasks within a project. Once assigned, each task makes it way from the 'Planned' stage, to the 'Completed' stage of the Kanban Board. The speed at which each task moves depends on how complex it is, with some tasks taking longer than others.

Kanban differs to Scrum in that Kanban is event-driven, with timeboxed iterations being optional. Event-driven tasks are ideal for the deployment of this application as sometimes it is extremely hard to estimate the amount of time and required to carry out a task. Thus, making Kanban a perfect methodology to implement as part of this project. Kanban allows users to add new tasks once the capacity is available. In contrast to Scrum, a Kanban Board can be implemented as part of a team or individually. Kanban's ability to continuously add new tasks is ideal for this project as its functionalities will continue to grow.

3.3. Overview of System

The application to be deployed will be a progressive web application and will follow the system architecture shown in the figure below and in the appendix at figure 24.

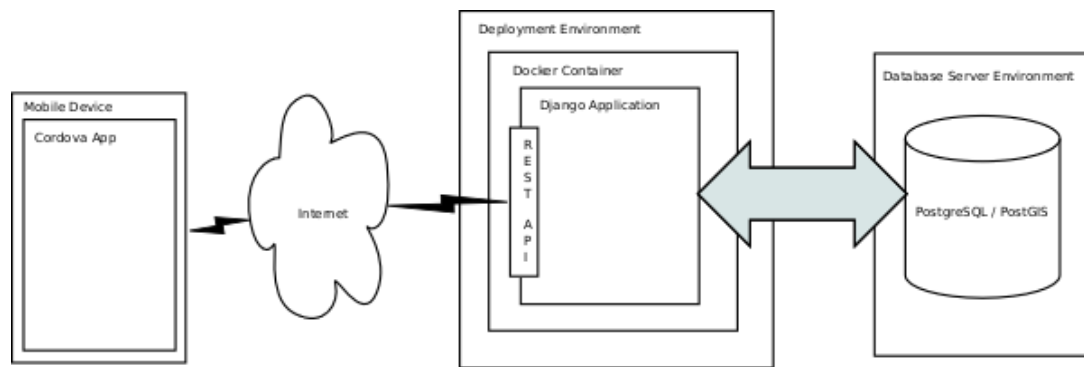


Figure 7: Architecture Diagram for Tradesmen Near Me

The front-end of this web application will take the form of a progressive web application and will be implemented using Python, HTML, CSS and JavaScript. They will be accompanied with additional JavaScript libraries as well as Bootstrap to shorten this projects development life cycle. Incorporating Bootstrap into this project enables Tradesmen Near Me to be scaled according to each user’s device. Upon visiting the site, users will be provided with the option to download it to their home screen. This will be accomplished using Apache Cordova which “allows you to use standard web technologies - HTML5, CSS3, and JavaScript for cross-platform development” (39). This, combined with offline functionality, will present a very practical application, equipped for daily use. Its front-end will be developed using the framework, Django, and will subsequently follow the MVC design pattern.

The front-end of this application will interact with its back-end from using REST API. REST stands for Representational State Transfer which “means when a RESTful API is called, the server will transfer to the client a representation of the state of the requested resource” (40). REST API follows a request from the user to query data and returns the state of that query. The representation of the state in this application will be in a JSON format. The request the user selects will take the form of a HTML verb, with the most common being GET, POST, PUT and DELETE. These then are passed to the back-end of the application in order to perform the relevant tasks.

This project's back-end will consist of PostgreSQL as the database, Digital Ocean as the cloud provider and Docker as the container software. PostgreSQL will communicate to the front-end using the REST API to interrupt user requests and will then search the database for any relevant results. Docker will be deployed to package all the dependencies of this project together in a docker container, therefore, allowing the application to work effortlessly in any environment. Digital Ocean will subsequently use the deployed Docker container to host this application in the cloud.

3.4. Front-End

3.4.1 Activity Diagram

An activity diagram is an extremely useful tool in UML as it helps describe the operation of a system. It details the flow of one activity to another within a system. The activity diagram outlined below illustrates the flow of events in this application following a search from a consumer.

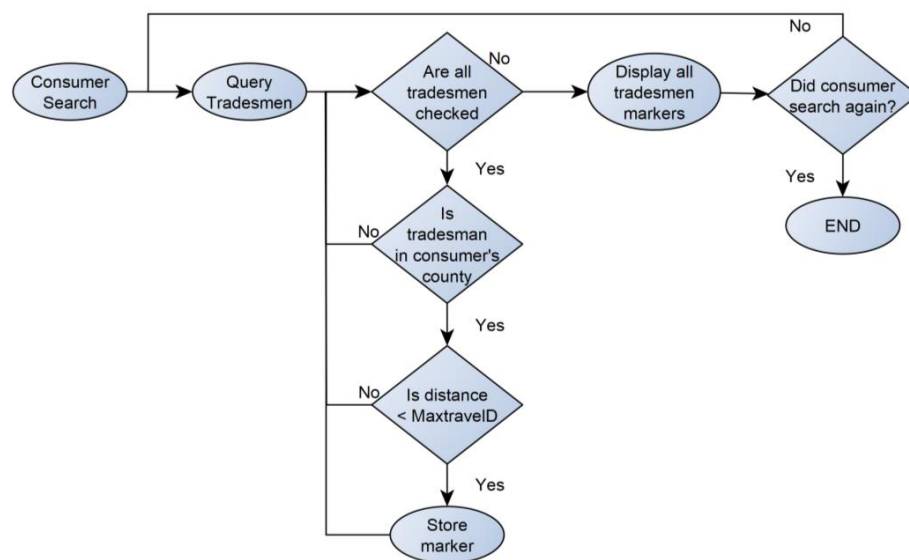


Figure 8: Activity Diagram for Consumer Search

The events depicted in figure 8 represent the flow from one activity to another following a search from a consumer. After the consumer searches for their desired tradesman, the application will retrieve the relevant data using the appropriate APIs and database queries until all possible tradesmen have been checked. Each tradesman is then iterated over to check whether they are in the consumer's required county. Each tradesman in the desired county are then passed into the specified geofencing algorithm so that it compares the distance from the consumer to each tradesman, with each tradesman's maximum travel distance. If the consumer falls within the tradesman's desired distance, it then stores its location as a marker. This procedure is then repeated for each tradesman until complete. When the application has finally iterated over every possible tradesman, it will display all the stored markers to the consumer in the form of a map.

3.4.2 Use Case Diagram

The use case diagram below in Figure 9 details the functionality interactions both, the user and consumer can have with the application. It also illustrates how the admin has exclusive privileges to maintain the data uploaded by each tradesman.

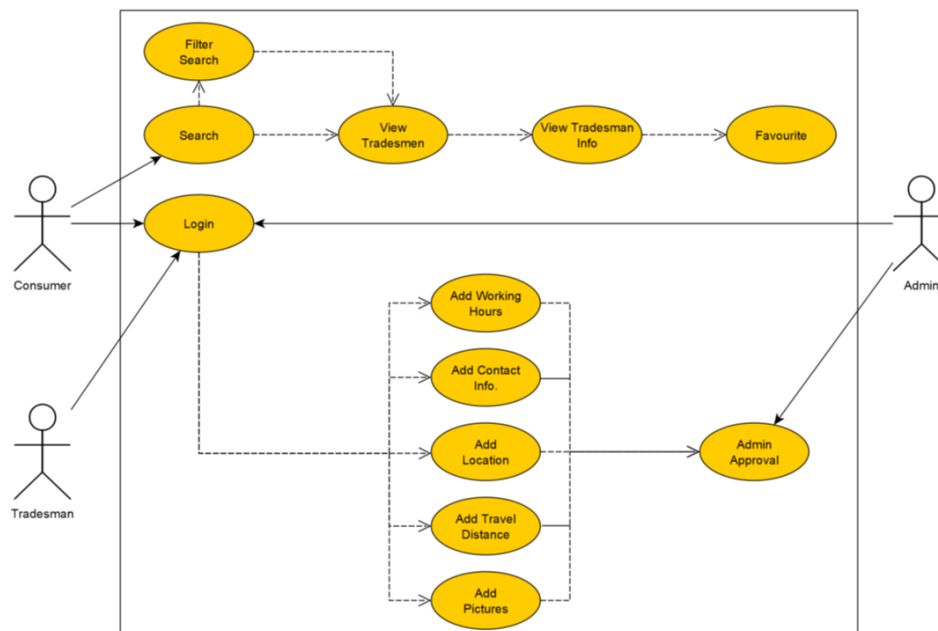


Figure 9: Use Case Diagram - Overview of Application

As seen within the use case diagram above, there are an array of features on offer for both, consumers, and tradesmen. When a tradesman initially visits this web application, they will be asked to login or proceed to sign up before they are provided the capabilities to add their own professional information. This information ranges from working hours to the distance they are willing to travel. After their desired information has been added, it will be sent for admin approval, meaning they will have to wait for the site admin to approve their submitted data. This form of error checking involves a short wait but is critical for ensuring the reliability of the information provided to consumers. A consumer on the other hand can query tradesmen in their area without the need to login. Providing the consumer with this functionality without the need to login is key as it keeps the consumer using the application for longer. From here the user can also filter their search to best suit their criteria. From the returned search, the consumer can view each supplied tradesman's page, providing them with helpful information such as their contact information. Upon logging in, the user can take advantage of the application's favourites tab by saving any tradesman they wish when viewing their page. This favourites tab will allow the consumer to save their preferred tradesmen for future and offline use.

3.4.3 Class Diagram

The following diagram depicts the classes and interfaces identified for the front-end of this application. This class diagram was compiled during the design stage of this application, leaving some scope for features to be added later in the development process.

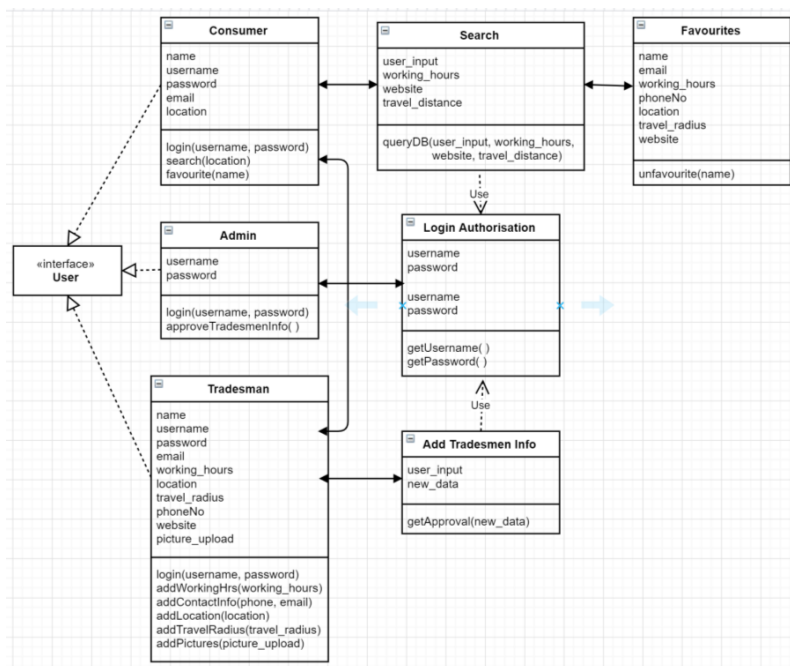


Figure 10: Class Diagram of Tradesmen Near Me Front-end

Figure 10 illustrates the class structure of this application and the relationships they share with each other. The three users of the system, admin, consumer and tradesman contain many attributes and methods, some more so than others. Tradesmen accessing the system have the most potential information to upload, should they choose to. Each of the three possible users of the system can login using the provided login authorization class. After doing so, each user branches off as they all have their own functionality. Tradesmen have the option to add their information for consumers to view, consumers can query stored tradesmen while the admin can approve submitted tradesmen information. All users share the same interface called user containing some shared attributes. The designing of a class diagram provides developers with a great overview of the required system as well as preparing them for the work that lies ahead when implementing such system.

3.4.4 Front-end Architectural Diagram

The front-end of this application will be developed using Python in conjunction with JavaScript, HTML and CSS whilst also making use of third-party JavaScript libraries. Tradesmen Near Me will utilize the UI library Bootstrap, so that the application can be used by any device without the need for developing separate native applications for IOS and Android. This application will follow the front-end architectural pattern, MVC as shown in figure 11 below.

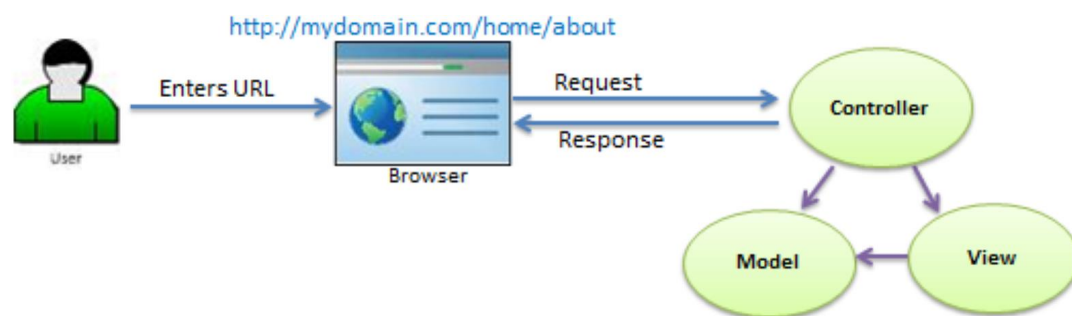


Figure 11: User Interaction with the MVC Architectural Pattern (41)

Here, we can see how the MVC pattern's components interact with each other, as well as how the MVC pattern interacts with the client accessing the browser. The MVC pattern separates an application into three components, model, view and controller, with each having their own specific function. The Model component stores retrieved data and as its related business logic which is then in turn requested by the View in order to display the user interface. The controller's role on the other hand is to manage the user's requests as it "renders the appropriate view with the model data as a response" (41). As depicted in the figure above, the user requests data from the controller and after interacting with the Model and View components, it returns the appropriate response. Implementing the MVC architectural pattern allows for easier code maintenance which enables faster debugging of the system. As well as this, the MVC pattern enables parallel development of the various components of the system, meaning that the different functionalities of the system can be developed simultaneously. The implementation of an MVC pattern provides numerous

advantages catered towards making the system development cycle more efficient and therefore is an ideal architectural pattern for this application, with time being such a valuable commodity.

3.4.4 Front-end Prototype Design

The high-fidelity prototype below was developed after extensive research and design. Although it requires a significant investment of time, a high-fidelity prototype allows developers to design a model closer to that of the final product. For this project, it helps to separate the various components of the application. Before developing this project, a low-fidelity prototype was first made as seen at figure 25 the appendix.

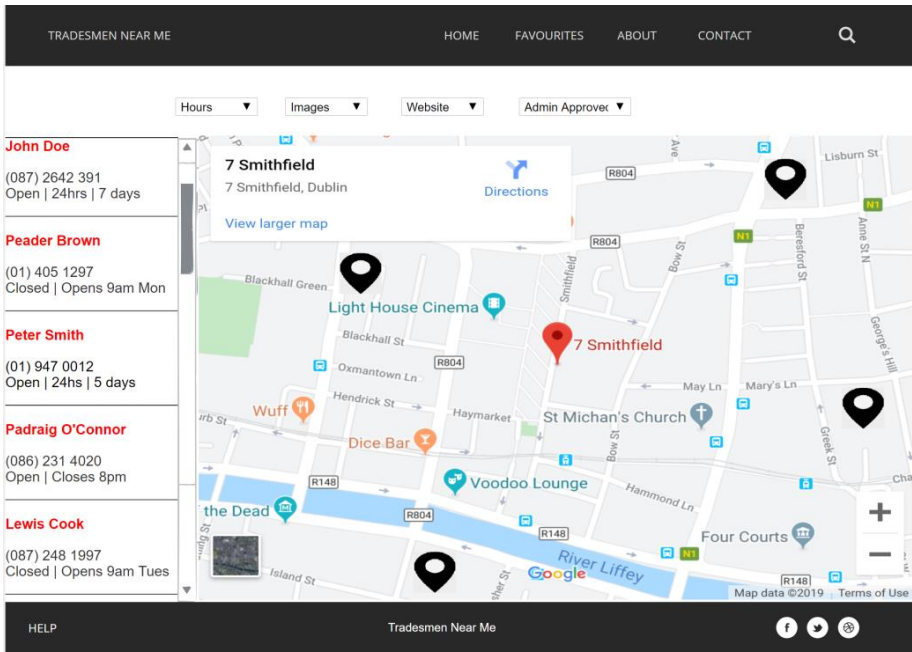


Figure 12: Front-End Prototype Design

Figure 12 represents the desired front-end of this application. Here, the consumer searching for a particular tradesman can see their location as displayed in red while the locations of tradesmen in their area are highlighted in black. The consumer will also be able to filter their search based on their requirements using the supplied filtering options. Upon searching, the

consumer will be provided with a graphical representation of the returned tradesmen alongside a concise scrollable list showing a brief description of their page.

3.5. Middle-Tier

3.5.1 Consumer Middle-Tier

The middle-tier of any web application refers to the handling of user requests such as submitting a form. In the case of a consumer in this application, it alludes to the searching function as performed by consumers and shown in figure 12.

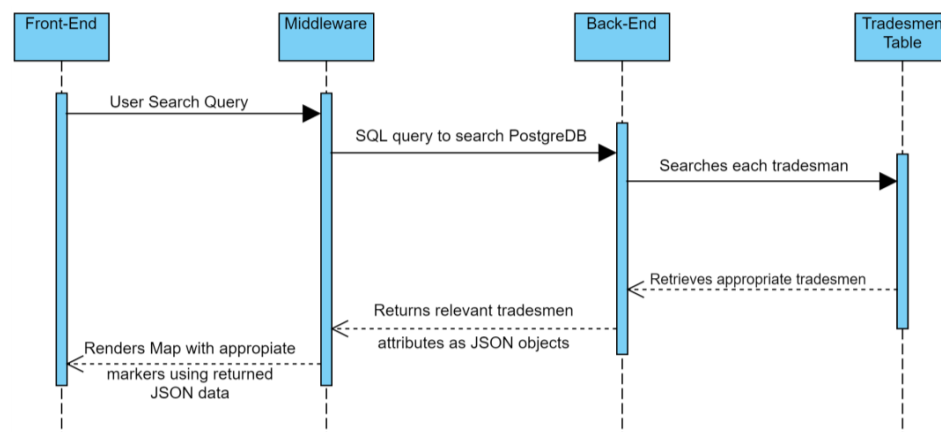


Figure 13: Sequence Diagram for consumers querying tradesmen

Here, it is clear to see the different operations assigned to each layer of the application. Following a consumer search, the appropriate SQL statement queries the PostgreSQL database. It then navigates to the previously created Tradesmen table where it iterates over each row in order to return the necessary tradesmen. After this, the relevant tradesmen are returned to the front-end of the application so that they can be rendered as markers on a map and subsequently displayed to the user.

3.5.2 Tradesman Middle-Tier

In contrast, for tradesmen accessing this web application, the middle-tier performs a different operation despite consisting of a similar form. It can be seen when a tradesman adds their own information for their page, as shown below.

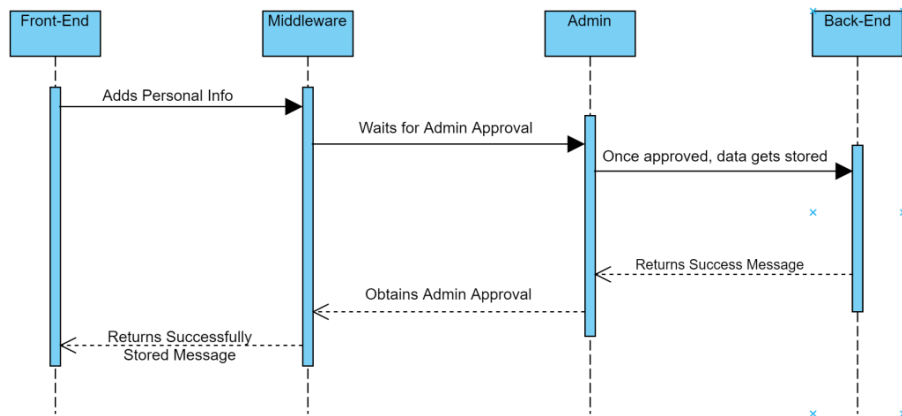


Figure 14: Sequence Diagram for tradesmen adding data

Sequence diagrams such as Figure 14 allow developers to easily document a system's requirements during the designing stages of a project. When a tradesman attempts to add their own information to their page, the request will be sent to the middle-tier of the application where it will wait for approval from the system administrator. Once approved, the new data will then be stored in the pre-defined database. Therefore, consumers are now able to query the newly added data.

3.6. Back-End

3.6.1 Database Design ERD

The figure below features an Entity Relationship Diagram (ERD) which illustrates the database entities needed to implement this application, as well as their relationships between one another.

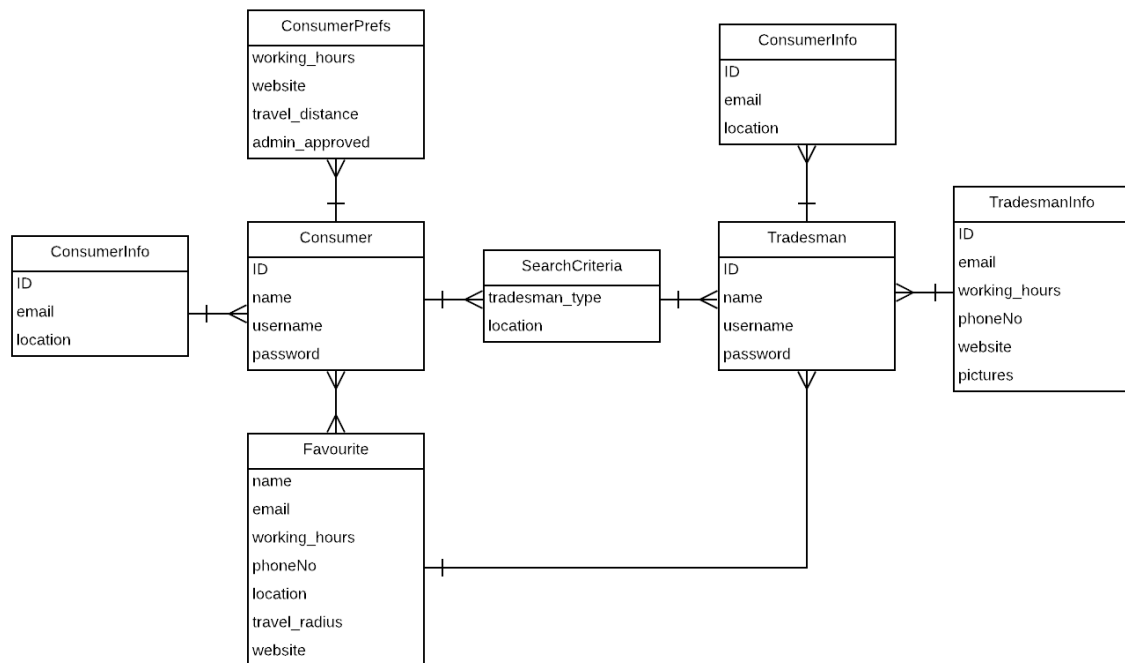


Figure 15: Database ERD for Tradesmen Near Me

The diagram above details the relationship diagram of this system's database schema which will be implemented using PostgreSQL. It emphasises the functionality the consumer has in being able to group multiple favourite tradesmen as well as being able to filter the search based on their requirements. Both, consumers, and tradesmen have the option to add many attributes to their profile with it also being in the tradesmen's interest to do so as it will further attract consumers to their page. An ERD is an essential in the designing of any application back-end as it gives a visual presentation of the database layout.

3.7. Conclusions

With a chosen methodology, design and initial prototype developed, the development of this application's core functionality can take place. The next development stages of the project will see the project being developed in accordance to the diagrams defined in this section.

4. Prototype Development

4.1. Introduction

This project's prototype development has been centred around the configuration of its various components. The main focus of the early development of this application involved the connecting of this application's chosen technologies. The implemented technologies, Django, Digital Ocean and Docker are explained in greater detail in the following section.

4.2. Front-End

The development of the front-end of this application concentrated on the use of Django as a framework. Django is a Python-based framework which follows its own variation of the Model-View-Controller architectural pattern, called Model-View-Template. Django enables developers to divide their project into three aspects, Model, View and Controller. This separation of functionality helps to deliver a fast solution using high quality code

PyCharm was chosen as this application's IDE (Integrated Development Environment) in which to implement Django. Tradesmen Near Me combines the front-end technologies, Python, JavaScript, CSS and HTML in order to display a clear, event-driven user interface. The structure of this applications front-end components within PyCharm can be seen in figure 15.

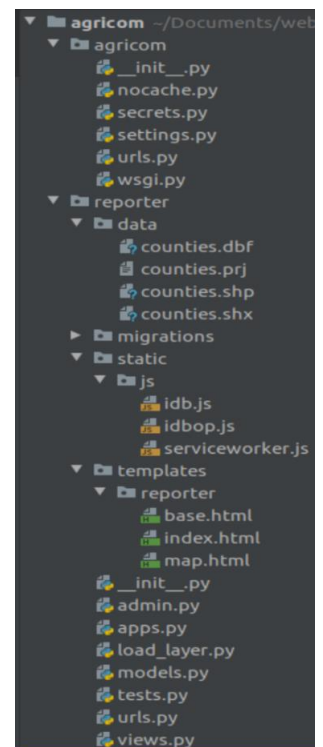


Figure 16: Application's Directory

The files associated with the technologies outlined above can be seen clearly within the application's directory. The HTML files have their necessary CSS code within them as the CSS is yet to go into excessive detail. The JavaScript files as shown within the static folder are responsible for the handling of Tradesmen Near Me's service worker.

The various Python files incorporated into this application include models, views, urls and settings. These components have their own individual purpose for ensuring the successful deployment of this Django application. Settings.py is responsible for the central configuration of a Django project, allowing for the integration of installed applications, middleware, database and Docker. A model as defined in models.py represents a collection in the application's Postgre database while views.py contains functions so that the application can read web requests and issue a response. The role of urls.py is then to define the mapping between URLs and views. The clear separation of application logic allows for faster development as the different components of the application can be worked on synchronously. Figure 17 illustrates this project's current front-end development.

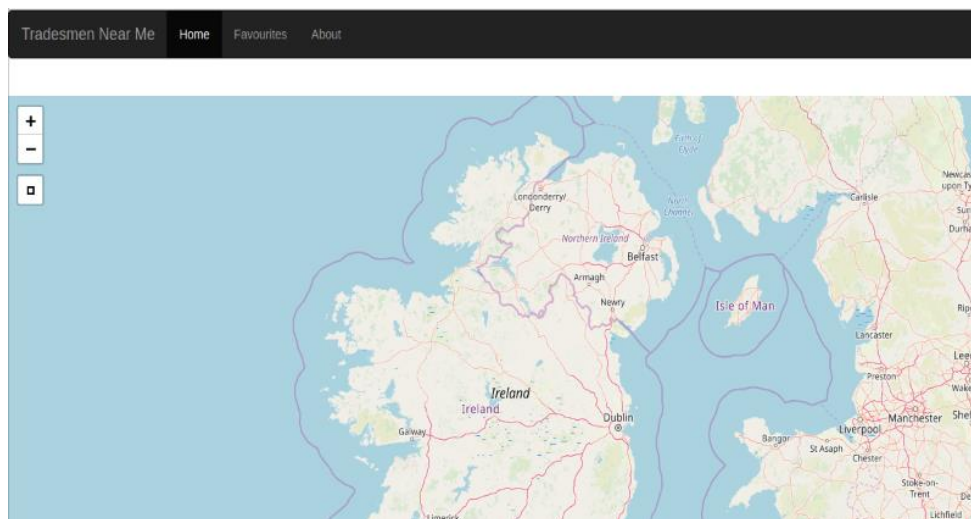


Figure 17: Front-End User Interface

Despite focusing the majority of the project's efforts on the internal structure of the application, the development of its front-end to date has seen the incorporation of a GeoDjango map into the display. The output of its Fetch API call will see the individual counties initially outlined with borders. This will be covered in more depth in the following section.

```

//collect latest post from server and store in idb
fetch('http://127.0.0.1:5006/getdata').then(function(response){
  return response.json();
}).then(function(jsondata){
  console.log(jsondata);
  dbPromise.then(function(db){
    var tx = db.transaction('feeds', 'readwrite');
    var feedsStore = tx.objectStore('feeds');
    for(var key in jsondata){
      if (jsondata.hasOwnProperty(key)) {
        feedsStore.put(jsondata[key]);
      }
    }
  });
});
});

```

Figure 19: Implementation of a service worker

In the figure above, it can be seen how the service worker fetches the appropriate API which in turn queries the getdata method, as defined in this application's views.py file. The received API response then returns the requested counties as JSON objects. The returned objects can subsequently be used to include a service worker with the application as outlined in figure 26 in the appendix.

4.4. Back-End

The development of this projects back-end included the use of Postgre as its database and Digital Ocean as its cloud provider. These, along with the incorporation of Docker accounted for the vast majority of this application's development time. Postgre is configured in secrets.py so that it can define database attributes such as engine, name and host. The layout of the directory files associated with this application's back-end technologies can be seen in figure 20.

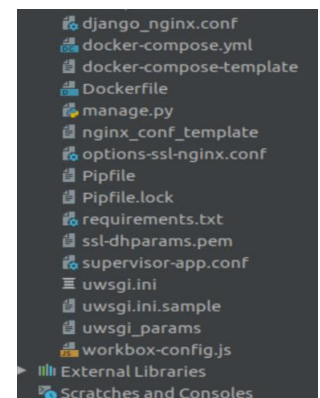


Figure 20: Back-end configuration files

4.4.1 Docker

Docker is “the only independent container platform that enables organizations to seamlessly build, share and run any application, anywhere” (43). Docker allows developers to package their code and its dependencies in the form of a docker container. A Docker container is simply an instance of a docker image. Implementing Docker as part of this

application Tradesmen Near Me to run reliably from one computing environment to another. The containers created as part of this application can be seen on a virtual machine using the 'docker ps' command, as shown in the figure below.

```
root@ubuntu-s-1vcpu-1gb-lon1-01:~# docker ps -a
```

| CONTAINER ID | IMAGE | COMMAND | CREATED | NAME |
|------------------------|---|--------------------------|-------------|------|
| STATUS | PORTS | | | |
| S | | | | |
| bb4f091f8752 | 225d8173801e | "supervisord -n" | 2 days ago | aaro |
| Up 2 days | 0.0.0.0:8000->80/tcp, 0.0.0.0:8443->443/tcp | | | |
| n_app_1 | | | | |
| 11fc634d28e5 | certbot/certbot | "/bin/sh -c 'trap ex..." | 2 days ago | aaro |
| Up 2 days | 80/tcp, 443/tcp | | | |
| n_certbot_1 | | | | |
| 58523f1c9189 | hello-world | "/hello" | 11 days ago | ferv |
| Exited (0) 11 days ago | | | | |
| ent_pare | | | | |
| 91a919061329 | kartoza/postgis | "/bin/sh -c /docker-..." | 13 days ago | dock |
| Up 2 days | 0.0.0.0:5433->5432/tcp | | | |
| er_container | | | | |

Figure 21: Docker containers

In order to build a Django application, the test environment must build a docker image from a Dockerfile. After doing so correctly, Docker push must be executed so that it publishes the newest image to the specified Docker Hub account. The next step in deploying a Docker container is to pull the latest Docker image from Docker Hub on a virtual machine then using Docker compose, stop or restart the Docker container, depending on your preference.

4.4.2 Digital Ocean

Digital Ocean is this application's cloud provider and was incorporated into the early development stages of the project. Upon obtaining a domain name, a droplet was created with a specified IP address. This IP address is necessary for getting the required SSL certificate. The created droplet can be seen in below in figure 22.

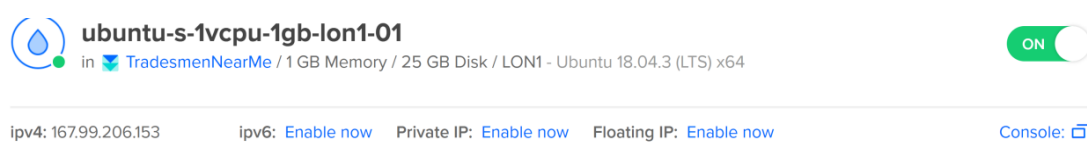


Figure 22: Created Digital Ocean Droplet

Digital Ocean aims to be designed for developers, enabling them to “build more and spend less time managing your infrastructure with our easy-to-use control panel and API” (44). The droplet used for this application consists of an Ubuntu virtual machine and allows the developer to gain access to the server’s console. From here they can assign a droplet to point to the project’s domain name. Digital Ocean’s servers only use solid state disks, making it very efficient. Not only this, but Digital Ocean’s control panel makes configuring cloud servers simple.

4.5. Conclusions

Upon reflection on the current working prototype, it is obvious that the internal structure of this application is in place. Therefore, allowing the application’s domain name to show the expected output. In future, changes to this application will require each Docker command mentioned above to be executed again. The current prototype has met its early development goals as it has all the required technologies implemented and connected. The subsequent section will now cover the various testing methods carried out on the existing prototype.

5. Testing and Evaluation

5.1. Introduction

Testing is a vital part of any software development project, with various testing techniques used throughout the development cycle of this project. Testing is hugely beneficial in pinpointing defects in software so that the appropriate corrections can be made. This can therefore prevent potential critical development errors from occurring. The testing techniques carried out on this project include white and black box testing.

5.2. Plan for Testing

One form of testing carried out has been white box testing, and more specially, unit testing. Unit testing has been performed on each developed component of this application. These tests ensure components are not throwing unexpected errors. “Unit testing helps identify a majority of bugs, early in the software development lifecycle” (45). Bugs identified at this early stage are easier to fix, thus, improving development time as a result.

Black box testing is also another form of testing being implemented in this project. The majority of black box testing carried out thus far has centred around API testing. These tests have been focused on the returning of the correct data from the necessary endpoints. The purpose of these tests is to ensure complete accuracy in returning the desired data the user requests.

Both, black box testing, and white box testing will continue to be implemented throughout the ongoing development of this project.

5.3. Plan for Evaluation

Before determining this applications type of evaluation, one must select the components that should be evaluated. Once these components have been chosen, the following research into the various evaluation types has been carried out. These include formative evaluation, process evaluation and outcome evaluation.

Formative evaluation is implemented during the development of a new program or when an existing program is modified. It “ensures that a program or program activity is feasible, appropriate, and acceptable before it is fully implemented” (46). Formative evaluation enables changes to be made to the product before starting the final stages of the development cycle. This allows developers to continuously modify their project, helping to further improve the quality of the product.

“Process Evaluation focuses on the implementation process and attempts to determine how successfully the project followed the strategy laid out in the logic mode” (47). Process evaluation can be carried out throughout the life cycle of a project. This is conducted by assessing the activities and components of the application.

“Outcome Evaluation measures program effects in the target population by assessing the progress in the outcomes that the program is to address” (46). Its process of evaluation begins as soon as one person or group in the target population has used the program. Outcome Evaluation clearly outlines how the program effects the target population’s behaviour while also communicating how effective the program is in meeting, both its deadlines and objectives.

The plan for evaluation in the next stage of this application is to continue using Process Evaluation. Process Evaluation has been conducted throughout the early development stages of this application. Process Evaluation aims to show the extent at which the program

is being implemented as initially designed. The quality of this application's designs makes the Evaluation Process an obvious choice as an evaluation type.

5.4. Conclusions

The testing and evaluating of this application is extremely important with it carrying greater importance as the development of the application continues. Upon entering the final stages of development, a detailed plan for testing will be outlined before being carried out.

Evaluating the application will take on a significant role in the future, with user satisfaction being a top priority.

6. Issues and Future Work

6.1. Introduction

There were some issues in the initial development of this project. The issues associated with the early stage of this application involved the use of Digital Ocean as a cloud provider and the integration of an API into the applications' front-end. After debugging these errors, a list of future work was composed. Both, the issues of this application, and its future work to be implemented are outlined in greater detail below.

6.2. Issues and Risks

Issues and risks encountered so far in the development of this project have centred around the management of version control. During the initial development stages of this application, I encountered an obscure problem with the downloading and installing of PostgreSQL as a back-end technology using Windows.

The main issue experienced in the early development stages of this application was the managing of its version control. I began by using GIT as well Google Docs as a secondary backup. Even at the beginning of this project, the importance of version control became apparent. After the difficulties encountered working with Postgre and Windows, I installed Ubuntu using a VirtualBox. Since then, Tradesmen Near Me has been developed on Linux Ubuntu 19.10 and PostgreSQL has been integrated seamlessly.

Another problem encountered with the application included the development of its architecture. This ranged from the front-end of the application to the incorporation of Digital Ocean as the applications back-end. In order to correctly implement Digital Ocean with Docker, research into the various Docker commands was carried out. However, after debugging some errors, both, Docker and Digital Ocean were successfully incorporated into this application.

Developing and implementing the complex features of this application could pose a potential issue with the overall efficiency of the app. After the application is deployed, the number of tradesmen stored within the Postgre database will only increase, resulting in more tradesmen to query when searching. This has the potential to have a severe effect on the speed of the returned search. If this does pose an issue in the future development of this project, measures will have to be taken to increase the speed of the search. Such measures could see the altering of the query used when querying the database.

Tradesmen Near Me has ultimately benefitted from the issues and risks experienced to date which has enabled the internal foundations of the application to be developed. Despite some development setbacks, the issues experienced so far will ultimately help solve future bugs in the system.

6.3. Plans and Future Work

Future work to be carried out include the incorporation of querying neighbouring counties as part of a consumers search. Therefore, benefiting consumers beside bordering counties. This can be seen in cases where a tradesman is close to a consumer but not within their given county, thus not including them within their search. The plan for the next development stage of this application will see neighbouring counties included in the full text search functionality.

Another feature to be added to this project includes a rating system whereby consumers can rate tradesmen who they have previously employed. The main task associated with implementing this feature is the authenticity of the reviews posted. To approve a consumer review for posting, the application must verify that particular consumer had work carried out by the claimed tradesman.

A fundamental component of the web application is yet to be implemented within this project. The favourites tab will be fully implemented and deployed within the coming

months alongside the existing service worker so that consumers can access their preferred tradesmen offline. The existing service worker implemented in this project is deployed correctly and it currently shows the style of the progressive web app once offline.

6.3.1. GANTT Chart

The GANTT chart below depicts an accurate picture as to how much time and commitment will be required over the coming months in order to deploy the application on schedule in March.

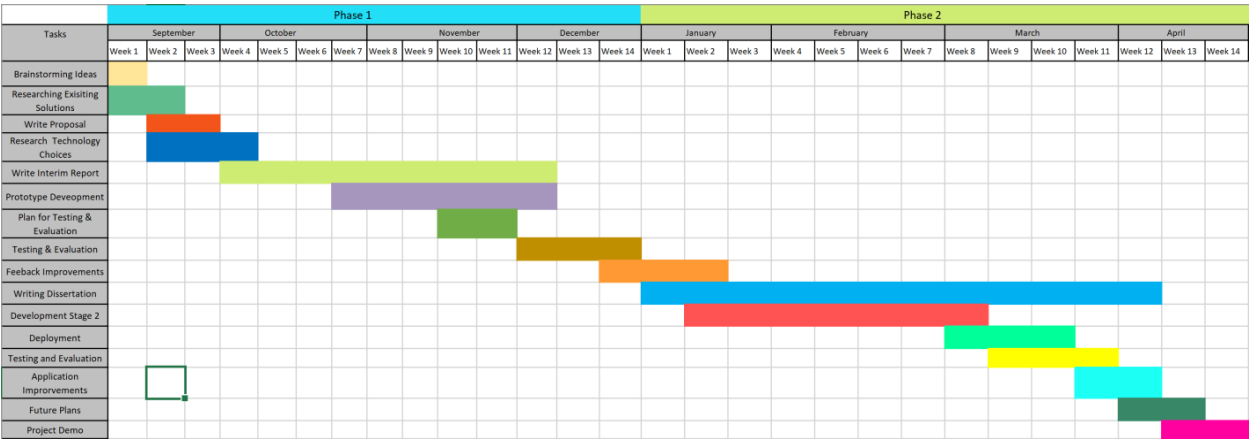


Figure 23: GANTT Chart

The GANTT chart’s objectives as defined in the left-hand column emphasises the importance of research in this project. Not only this but it highlights the significance of testing and evaluating the application before and after deployment. Despite making a solid start to this project, there is a substantial amount of work yet to do to meet this application’s required standards.

Bibliography

1. Reddan F. Rents in Dublin to rise by 5-6% a year and surpass €2,000 by 2019 [Internet]. The Irish Times. [cited 2019 Dec 7]. Available from: <https://www.irishtimes.com/life-and-style/homes-and-property/rents-in-dublin-to-rise-by-5-6-a-year-and-surpass-2-000-by-2019-1.3314199>
2. Savills. Number of people renting in Ireland at record high. 2017 [Internet]. [cited 2019 Mar 11]; Available from: <https://www.rte.ie/news/business/2017/1204/924777-number-of-people-renting-at-record-high-savills/>
3. About Us - Find Online Tradesmen [Internet]. [cited 2019 Nov 3]. Available from: https://tradesconnect.ie/page/about_us
4. Security Specialists | Dublin 24 | Dublin | Advance Access - Access Control, Automatic Barriers, Car Park Solutions, Revolving Door Providers | Trades Connect [Internet]. [cited 2019 Nov 3]. Available from: https://tradesconnect.ie/security-specialists/advance-access-access-control-automatic-barriers-car-park-solutions-revolving-door-providers_i720
5. Daft.ie : About us at Daft.ie - Ireland's No.1 Property Website [Internet]. [cited 2019 Nov 3]. Available from: <https://www.daft.ie/about/>
6. Property to rent in Dublin City Centre, Dublin City | Daft.ie [Internet]. [cited 2019 Nov 3]. Available from: https://www.daft.ie/dublin-city/residential-property-for-rent/dublin-city-centre/?s%5Bignored_agents%5D%5B0%5D=1551&searchSource=rental&offset=20
7. About Us [Internet]. Airbnb Newsroom. [cited 2019 Nov 3]. Available from: <https://news.airbnb.com/about-us/>
8. Node.js - Introduction - Tutorialspoint [Internet]. [cited 2019 Nov 4]. Available from: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm
9. Benefits of Python over Other Programming Languages - Invensis Technologies [Internet]. [cited 2019 Nov 4]. Available from: <https://www.invensis.net/blog/it/benefits-of-python-over-other-programming-languages/>
10. 5 software frameworks advantages for web app development [Internet]. Syndicode - Ruby on Rails development agency. 2017 [cited 2019 Nov 9]. Available from: <https://syndicode.com/2017/08/21/5-open-frameworks-advantages-for-web-app-development/>
11. Mallawaarachchi V. 10 Common Software Architectural Patterns in a nutshell [Internet]. Medium. 2018 [cited 2019 Nov 9]. Available from: <https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013>

12. Sinhal A. MVC, MVP and MVVM Design Pattern [Internet]. Medium. 2017 [cited 2019 Nov 9]. Available from: <https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad>
13. Architectural Guidelines to follow for MVP pattern in Android [Internet]. [cited 2019 Nov 11]. Available from: <https://android.jlelse.eu/architectural-guidelines-to-follow-for-mvp-pattern-in-android-2374848a0157>
14. The Top 10 Advantages Of Using Laravel PHP Framework [Internet]. Belitsoft. 16:50:00+00:00 [cited 2019 Nov 4]. Available from: <https://belitsoft.com/blog/10-benefits-using-laravel-php-framework>
15. Advantages and Disadvantages of Django | Blogs @ Mindfire Solutions [Internet]. [cited 2019 Nov 4]. Available from: <http://www.mindfiresolutions.com/blog/2018/04/advantages-and-disadvantages-of-django/>
16. AngularJS — Superheroic JavaScript MVW Framework [Internet]. [cited 2019 Nov 14]. Available from: <https://angularjs.org/>
17. Tutorial: Intro to React – React [Internet]. [cited 2019 Nov 14]. Available from: <https://reactjs.org/tutorial/tutorial.html>
18. What is Client-Side Scripting? Choosing the Scripting Languages for your Web Application [Internet]. Hiring Headquarters. 2015 [cited 2019 Nov 14]. Available from: <https://www.upwork.com/hiring/development/how-scripting-languages-work/>
19. Angular - The RxJS library [Internet]. [cited 2019 Dec 9]. Available from: <https://angular.io/guide/rx-library>
20. Database Concepts [Internet]. [cited 2019 Nov 11]. Available from: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm
21. Advantages of PostgreSQL ★ Bitnine Global Inc. [Internet]. [cited 2019 Nov 3]. Available from: <https://bitnine.net/blog-postgresql/advantages-of-postgresql/>
22. What is AWS [Internet]. Amazon Web Services, Inc. [cited 2019 Dec 7]. Available from: <https://aws.amazon.com/what-is-aws/>
23. Google Cloud Platform Overview | Overview | Google Cloud [Internet]. [cited 2019 Nov 11]. Available from: <https://cloud.google.com/docs/overview/>
24. Managed Databases (DBaaS) on DigitalOcean [Internet]. DigitalOcean. [cited 2019 Nov 11]. Available from: <https://www.digitalocean.com/products/managed-databases/>
25. Filtering UI: A Horizontal Toolbar Can Outperform the Traditional Sidebar - Articles - Baymard Institute [Internet]. [cited 2019 Nov 21]. Available from: https://baymard.com/blog/horizontal-filtering-sorting-design?source=post_page-----beea1798d64b-----

26. What is Elasticsearch | Elastic [Internet]. [cited 2019 Nov 23]. Available from: <https://www.elastic.co/what-is/elasticsearch>
27. AWS | Amazon CloudSearch - Search Service in the Cloud [Internet]. Amazon Web Services, Inc. [cited 2019 Nov 23]. Available from: <https://aws.amazon.com/cloudsearch/>
28. KRISHNA S. Code for a Cause: Geofence Technology for Priority Vehicles [Internet]. Medium. 2019 [cited 2019 Nov 23]. Available from: <https://medium.com/progate/code-for-a-cause-geofence-technology-for-priority-vehicles-be0e018df0f0>
29. Just How Accurate is GPS? - Microlise [Internet]. [cited 2019 Nov 27]. Available from: <https://www.microlise.com/blog/just-how-accurate-is-gps/>
30. Houston P. Calculate distance of two locations on Earth using Python [Internet]. Medium. 2019 [cited 2019 Dec 7]. Available from: <https://medium.com/@petehouston/calculate-distance-of-two-locations-on-earth-using-python-1501b1944d97>
31. Bullock R. Great circle distances and bearings between two locations. MDT,. 2007 Jun; Available from: https://dtcenter.org/sites/default/files/community-code/met/docs/write-ups/gc_simple.pdf
32. Leaflet — an open-source JavaScript library for interactive maps [Internet]. [cited 2019 Nov 24]. Available from: <https://leafletjs.com/>
33. Advantages of Bootstrap framework | Vmoksha [Internet]. [cited 2019 Nov 24]. Available from: <https://vmokshagroup.com/blog/bootstrap-advantages/>
34. Nath DS. 4 important points to know about Progressive Web Apps (PWA) [Internet]. Medium. 2018 [cited 2019 Nov 24]. Available from: <https://medium.com/@deepusnath/4-points-to-keep-in-mind-before-introducing-progressive-web-apps-pwa-to-your-team-8dc66bcf6011>
35. Agile Vs Scrum: Know the Difference [Internet]. [cited 2019 Dec 5]. Available from: <https://www.guru99.com/agile-vs-scrum.html>
36. What Is Scrum Methodology? [Internet]. [cited 2019 Dec 5]. Available from: <https://resources.collab.net/agile-101/what-is-scrum>
37. What Is Kanban? An Introduction to Kanban Methodology [Internet]. [cited 2019 Dec 5]. Available from: <https://resources.collab.net/agile-101/what-is-kanban>
38. What is Agile Kanban Methodology? Learn the Methods & Tools [Internet]. [cited 2019 Dec 5]. Available from: <https://www.inflectra.com/methodologies/kanban.aspx>
39. Architectural overview of Cordova platform - Apache Cordova [Internet]. [cited 2019 Nov 25]. Available from: <https://cordova.apache.org/docs/en/latest/guide/overview/>
40. Avraham SB. What is REST — A Simple Explanation for Beginners, Part 1: Introduction [Internet]. Medium. 2017 [cited 2019 Nov 25]. Available from:

<https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-1-introduction-b4a072f8740f>

41. MVC Architecture [Internet]. [cited 2019 Nov 17]. Available from: <https://www.tutorialsteacher.com/mvc/mvc-architecture>
42. Introduction to Service Worker | Web [Internet]. Google Developers. [cited 2019 Dec 5]. Available from: <https://developers.google.com/web/ilt/pwa/introduction-to-service-worker>
43. Why Docker? [Internet]. Docker. [cited 2019 Dec 6]. Available from: <https://www.docker.com/why-docker>
44. DigitalOcean – The developer cloud [Internet]. DigitalOcean. [cited 2019 Dec 6]. Available from: <https://www.digitalocean.com/>
45. What is WHITE Box Testing? Techniques, Example, Types & Tools [Internet]. [cited 2019 Nov 26]. Available from: <https://www.guru99.com/white-box-testing.html>
46. Types of Evaluation. :2.
47. Process Evaluation [Internet]. [cited 2019 Dec 5]. Available from: http://www.uniteforsight.org/evaluation-course/module5#_ftn1

Appendix

Figure 24: Screenshot of Daft Property Location (46)

