



Dublin Coffee Shop Finder

This Report is submitted in partial fulfilment of the requirements for the
BSc (Honours) Information Systems and Information Technology (DT249) to the
School of Computing, College of Sciences & Health, Dublin Institute of Technology.

Author: Keith ÓMuirí, D11124160

Supervisor: Patrick Browne, School of Computing

Date: 4th January 2016

1.0 Abstract

The project will take a spatial analysis approach to review the location of coffeeshops in the Dublin region. To provide a responsive website which users can use on their desktops and mobile devices to enable them find speciality coffee shops in Dublin. Shop details are stored in a postgres database, allowing users to filter specific requirements via the menu system. The server will then return location data and display on a map the nearest coffee shop based on the users search criteria.

The implementation is a full stack development from building out a debian virtualmachine on an ISP to building and configuring the servers needed to serve the individual elements required. Developing and configuring the database back end and designing and building the responsive front end. All aspects of the project are opensource in nature and rely on opensource software and tools to bring the project to fruition.

Keywords: OpenSource, GIS, Spatial Database, GeoCoding, Leaflet, Node.js, OpenLayers, Responsive, full stack, Postgres.

2.0 Acknowledgements

Firstly, I would like to express my gratitude to Patrick Browne for the continuous support, patience and motivation given during the preparation of this dissertation. I cannot imagine having a better adviser and supervisor for this.

Besides my supervisor, I wish to thank the entire school of computing staff in the Dublin Institute of Technology who have provided an amazing experience during the course of my studies in Kevin Street.

My sincere thanks also goes to a number of people who answered silly questions, gave freely their expertise and experience in many aspects of my project, especially in the run up to submission with code problems I was experiencing, none more than Nigel Hanlon, and Donovan Hutchinson for his freely available course on CSS animation and his advice throughout.

I also would like to express my sincere gratitude to friends for the proofreading provided.

Last and certainly not least, I would like to thank my wife Siobhán, my children Cóil, Clíodhna, Doireann and Fiadh and my extended family particularly my Father in law Tom O'Neill for supporting me throughout my return to college and encouragement during the entire period and not just in the writing of this dissertation.

3.0 Declaration

DT249 BSc in Information Systems and Information Technology

DECLARATION OF OWN WORK.

Student Name : Keith ÓMuirí

Student Number: D11124160

DECLARATION

I Keith ÓMuirí hereby confirm that the project I now submit for Honours Project Module.

Titled: Dublin Speciality Coffee Finder

is my own research work and was written by me following my research as cited therein. The work is new and has not been submitted for any previous award. I also confirm that the slides I use/will use during the presentation that I made/will make are also my own work and are based on the content of my proposal. I further confirm that the work has not been taken from the work of others save and to the extent that such work has been identified, cited and acknowledged within the text of my proposal and presentation.

Signed _____

Date _____

4.0 Contents

1.0 Abstract.....	2
2.0 Acknowledgements.....	3
3.0 Declaration.....	4
4.0 Contents.....	6
5.0 Glossary of Terms.....	8
6.0 Table of Figures.....	10
7.0 Introduction.....	11
Goal & Aims.....	11
8.0 Motivation.....	12
9.1 Research Delivery Environment.....	14
9.2 Research Platform.....	15
Django.....	15
Node.js.....	16
9.3 Research Maps.....	18
Tilemill maps.....	19
Feature Server.....	19
9.4 Data Collection.....	20
10.0 Design.....	22
Custom Artwork.....	24
Mockups.....	25
My visceral reaction study.....	28
Mobile First versus Responsive.....	29

12.0 How the application works.....	30
Searching.....	32
13.0 What the map shows:.....	37
14.0 Testing.....	41
Functional Acceptance Testing (FAT).....	41
Infrastructural testing.....	42
Software Testing.....	43
Browser Environment Testing.....	44
User Acceptance Testing.....	44
Load / Stress testing.....	45
15.0 Analysis.....	46
16.0 Conclusion.....	47
17.0 Recommendations.....	49
18.0 References.....	50
19.0 Appendix 1 photos of Book and App referenced in [8.1]&[8.2].....	
20.0 Appendix 2 copy of survey sent to shops	
21.0 Appendix 3 list of shops used for purposes of application.....	
22.0 Appendix 4 – attached as a zipped copy of the entire folder structure (the code)	

5.0 Glossary of Terms

DIT	Dublin Institute of Technology
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language.
SVG	Scalable Vector Graphics
Javascript	JavaScript (JS) is a programming language mostly used client-side to dynamically script webpages, but in this case also server-side
Django	is a free and open source web application framework, written in Python.
Framework	A web framework is a set of components that helps you to develop websites faster and easier.
Node.js	Node.js is an open-source, cross-platform runtime environment for developing server-side web applications.
Tilemill	is a design environment developed by MapBox for cartography, constituting Mapnik as a renderer, CartoCSS as a stylesheet language
Mapnik	Mapnik is an open source mapping toolkit for desktop- and server-based map rendering
CartoCSS	is a Mapnik stylesheet pre-processor developed by MapBox and inspired by Cascadenik.
Mapbox	Mapbox is a large provider of custom online maps for websites
Server	a computer or computer program which manages access to a centralized resource or service in a network.
Linode	A Virtual Private Server Provider (my data centre is based in Frankfurt)
Postgres	A free, open source relational database management system
PostGIS	an open source, freely available spatial database extender for the PostgreSQL Database Management System
Linux	an open-source operating system modelled on UNIX.
PGRouting	An open source geospatial database extension to provide geospatial routing functionality.
Debian	Debian is a popular and freely-available computer operating system that uses the Linux kernel and other program components obtained from the GNU project
FOSS	Free and Open Source Software

Open Source	In general, open source refers to any program whose source code is made available for use or modification as users or other developers see fit. Open source software is usually developed as a public collaboration and made freely available.
Phonegap	An open source framework for quickly building cross-platform mobile apps using HTML5, Javascript and CSS.
OSM	Open Street Maps
CMS	Content Management System
QGIS	Open Source GIS Software
DIV	This is a HTML instruction for dividing the layout of a web page
AJAX	Asynchronous JavaScript and XML
GIS	Geographic Information Systems
GML	Geographic Markup Language
Layer	Typically a layer of information (the results of a search query)we wish to display over a map
WMS	Web Map Server
WFS	Web Feature Server
OpenLayers	This is a Javascript product used in accessing and displaying map information
OSI	Open Source Initiative
Projection	This is an agreed representation of a section of the earth in two dimensional environment
Route	a way or course taken in getting from a starting point to a destination.
API	a set of functions and procedures that allow access to features or data of an operating system or application.
Heroku	Heroku is a cloud Platform-as-a-Service
Twitter	Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets".
Vitual machine	a software computer that, like a physical computer, runs an operating system and applications.

6.0 Table of Figures

Illustration 1: Custom Logo.....	21
Illustration 2: Map showing custom styling.....	21
Illustration 3: Zoomed in map showing custom styling.....	22
Illustration 4: Sample of Custom Marker Pins.....	23
Illustration 5: Custom "on brand" 404 Page.....	23
Illustration 6: Mouse over action mockup.....	24
Illustration 7: 'on click' action mockup.....	24
Illustration 8: Showing mockup of shop details layout.....	25
Illustration 9: Hamburger menu icon illustration.....	25
Illustration 10: How the different elements of web mapping work.....	30
Illustration 11: Map showing selection of shops.....	36
Illustration 12: mockup showing map selection by business friendly status.....	37

7.0 Introduction

Goal & Aims

The principal aim of the project is to produce a web based product, which would display a map for the user of curated speciality coffee shop locations within the Greater Dublin region. By allowing the user to use their location, they should be able to find the closest store on a map displayed within a browser. A high level investigation into Geographical Information Systems and Spatial Database Systems will be undertaken in as far as required to implement the project. All of the tools used throughout the project, save the graphic design software for generating icons, branding identity will be open sourced in nature and freely available. The use of Open Source software for the generation and publishing of the data to be used is fundamental to the project. From a deployment point of view this includes the use of Debian-Linux as an Operation System, PostgreSQL with PostGIS, a custom web feature server, custom map tile server, serving custom map tiles, OpenLayers, leaflet, CSS3, Javascript, Node.js and numerous open sourced add on libraries (ie Bootstrap, jquery, mustache, handlbar) The delivery is aimed at the specialist coffee community for the most part. The searches available are designed with those in mind.

8.0 Motivation

Apart from the obvious motivation of having to write a dissertation, my motivation was to fill a need I myself had.

I am a coffee geek. I started normally like anyone else, but my morning habit quickly became an obsession. I had been to the big american chain when it opened in Dublin and one opened just around the corner from my office in Waterloo Road. I was an addict. I moved offices to Shelbourne Road and my coffee haunt was just a little too far to walk for my fix. I tried a few other places, but the coffee was sub standard. On the course of my daily commute to the office, and being then in the property development world, I kept an eye out on retail and office refurbishments along my route.

I noticed an old cafe being renovated over the course of a few weeks. On the day the signage went up I recognised the branding and excitedly pulled over and went into the building site to enquire. It was a very highly regarded coffeeshop, the owner of which had already achieved legendary status in the Irish and international coffee circuit. The shop was 3fe (Third Floor Espresso) and the owner was Colin Harmon, multiple Irish Barista champion and then ranked 4th in the world championships. I was aware of his shop on Abbey Street but had only ever been once. I was soon visiting every morning and converting everyone I met to stop using the large chains and to visit 3fe. The staff were very happy to discuss the coffees and methods of preparation on offer and would educate anyone who was willing to listen.

As part of my work, I spent a number of days in central London and was offered a loan of a book by one of the Baristas (Pete Williams) in 3fe before I left “The London Coffee Guide” [8.1 The London Coffee Guide] I was amazed that this book was not in production for Dublin. It gave a curated listing of 100 speciality coffee shops in London, details on locations and a two page spread per shop giving all sorts of information, allowing me to plan my lunch and pre work coffees without

ever having been there before. (sample pages in Appendix 1) The book also included a fold out map showing central London and the shop locations.

I also then thought that this map would be ideally delivered online and the book could be replicated quite easily allowing for searching of attributes and filtering of results, not to mention being able to keep it up to date. The seed was sown. (the publishers have since developed an app for London) [8.2 London Coffee Guide App] (screen shots in Appendix 1)

Time passed and circumstances had me take up a new role in the software development industry. I moved offices over to Marlborough St and away from my latest haunt. I didn't know of any shop close enough to fill my daily requirements. I walked quite a distance to a shop I did know and enquired there as to any shop they were familiar with, which might be closer to my office. They told me about a shop about 200 metres from my office which I had never seen fit to visit. Insider knowledge passed along. Some six months passed and our offices moved to a new location, once again, too far from my familiar haunt, I was going through the same process asking at a shop I was too far from for a daily visit when the idea of the coffee guide resurfaced. I was coincidentally thinking of topics for my dissertation and the idea stuck.

I read the guidance notes and spoke with the projects coordinator and formalised my decision to base my dissertation on a topic I enjoyed and would happily spend time working on.

My initial concept was for my own use only. It was suggested by a friend that I make it available as the World Barista Championships 2016 would be taking place in Dublin with over 10,000 visitors from 100 countries flying in and the project would be ideal for visitors to the city to experience some of our flourishing local coffee scene.

9.1 Research Delivery Environment

Having planned my application at a high level, I set about filling in details on how I might deliver the finished product and what environment I might deliver it in.

Delivery Environment

I initially presumed that I would make a smartphone app, considering that I felt most people would use the application on their smartphone. I carried out some research and multiple sources pointed to a piece of software called Phonegap [8.1 Phonegap]. I read the documentation and installed a trial version for prototype testing. This application allowed you to basically turn your web delivered application into an encapsulated smart phone app. Initial research showed that it seemed to suit my requirements. However, having spoken with a number of software developers I work with, I was told that phonegap had specific requirements which would not particularly suit my application, the app would still require live connections to my database unless users were only to receive updates to new shop locations or details at upgrade releases of the app. Furthermore, my routing proposal would also require connection to a server to enable the location information to be passed into the software required to perform the route analysis.

I also came to the conclusion that people would not necessarily install an app to find a coffeeshop as a once off exercise and that a website which looked and performed well on a smart phone might be better suited. I also considered that people visiting Dublin might not necessarily wish to install an app specifically for this purpose and that a link, advertised in a coffeeshop or online with a memorable URL listed might gain more traffic.

As far as delivery environment was concerned, my mind was made up, a web application it was.

9.2 Research Platform

My initial prototyping was carried out using basic html written in a text editor.

I undertook a number of online tutorials for the mapping interface from *Openlayers / Google maps / Leaflet / Mapbox*. I had hoped that this would be sufficiently complex to undertake my project. I started to notice gaps in my work where each of the tutorials were relying on system dependencies which I did not have in place. I researched this and soon found that most professional web mapping developments were based on a html / javascript framework.

The company I work for use a number of frameworks as standard and I spoke with some of the software engineers and reduced my list of options to two standard frameworks in predominant use in my office. This made sense as I would have access to professional software engineers on a daily basis and any questions I might have could be answered immediately and not require extensive research.

The choice to be made was between *Django* and *Node.js*.

Django

Django has wide support community and comes with an extensive range of ready to go bolt-on libraries which seemed to cover all the elements I wished to implement in my project. *Django* required some configuration and planning at the start to begin my project correctly which I was assured would be covered by one of the numerous online courses freely available. Server side calculations and functions would be carried out using *Python* which might prove beneficial to me in the long term as my office used *Python* extensively.

Node.js

Node.js was a less well supported but somewhat lighter weight approach. I was warned that some of the elements I had discussed would require my writing javascript functions myself as the community support was not in place as with *Django*. I am not adept at writing javascript programs apart from some smaller functions adapted from online tutorials.

There was a positive element to using node, the server and the application would use the same language, both *Javascript* client and serve side.

The objects I would be using by default would be JSON, which is particularly useful for web APIs.

I felt the arguments pushing for either choice over another contained a certain element of religious belief. I found it difficult to find an unbiased piece of research to help me choose. Part of the issue I now understand having carried out development using both is that I might as well have been comparing apples and oranges. They can both be used to deliver on the surface, all of the requirements for my project, however, both capabilities and approaches are entirely different and the choice to go with *Django* was based on the fact that the company I work for are a *Python* development house, despite my not having any *Python* knowledge or experience.

I chose to develop my application using *Django*. I enrolled on an online course for *Python*, [9.1 Shaw 2013] bought the associated book and began working through the tutorials. I also undertook a course [9.2 *Django*] on *Django* development at the same time. I developed the second prototype for my project using *Django*. A decision biased by the software engineers I work alongside. I did reach a point where I wasn't entirely sure how to change a piece of text on one of the pages I had written, having carried out a number of builds and revisions locally. I asked for help, which was immediately given, but I still didn't understand where I was going wrong.

I felt that I did not really understand the project and could not draw on a piece of paper, where all of the elements were stored. I was drowning in code. I pulled back from writing code and went back to drawing diagrams of the workflows for my application.

After a prolonged period of frustration with *Django* I decided to take a small detour and try *Node.js*. I enrolled on a free online course for writing *Node.js* [9.3 Node]. Within minutes I had a running server, within a few days I had gotten myself to the same stage as I was after weeks of *Django*. Whilst I accept that many of the decisions made during my initial development of my application had been made for the *Django* prototype and I could simply replicate them, I did find *Node.js* to be more flexible.

I started a new git repository and began again after almost 6 months of hit and miss development.

Within one month I was in the same place as I had been before I restarted but I understood where everything lived and could change elements of the site comfortably.

The indoctrination was complete, *Django* was now the devil, all praise beautiful *Node.js*. (until something easier comes along) My initial development with *Node.js* was deployed live within a few weeks using Heroku [9.4 Heroku] which allowed me to have a running server and database on a free tier. I soon realised that the routing and specific requirements I had for my application meant that whilst *Heroku* would be great for early development that I would eventually have to choose a 'grown up' solution. I based my decision on which provider to go on based on a quick *Twitter* poll (like too many things these days, instant gratification) and I purchased a Virtual Machine from *Linode*. I also purchased a domain name for my application **www.coffeecoфее.ме** and had a live system up and running after a few days of configuration and installation.

9.3 Research Maps

My initial prototyping used map tiles provided by *Mapbox* [9.5 Mapbox]. The interface on their website was simple to use and they generated the code required along with the security keys required to insert into your application to gain access. *Mapbox* also enabled me to upload features from a csv file and serve them directly on the map they provided. When I first developed a map for testing and prototyping purposes I had an issue with the fact that *Mapbox* only allowed for minimal customisation of appearance and I had planned to customise my maps entirely. I had experience developing custom mapping solutions for a number of clients and chose to use *Tilemill* to develop the map tiles for this application. *Mapbox* was also at a disadvantage as far as dependency was concerned, if *Mapbox* experienced some technical issues my application would be without mapping, whereas if I served map tiles myself, I would be in control of this element of the application.

Mapbox have since entirely revamped their mapping offering and a user can customise each and every facet of their map styles. I did carry out a trial to gauge if I could get up to speed but the sheer volume of choices available meant that I found myself spending far too much time tweaking maps which I already had almost completely prepared in *Tilemill*.

Tilemill maps

I downloaded the open street map dataset covering Dublin region and imported them into *Tilemill*.

In *Tilemill* I was able to format the styling of the layers using *CartoCSS* or 'map style sheets' similar in nature to CSS. I could reduce the number of available zoom levels and restrict the overall map size, thus making the export quicker and optimise the amount of detail for my application.

I have enclosed the very simple MSS file in the code of Appendix 3. The amount of control possible is staggering and given the chance to revisit, I believe I would spend more time customising the maps to appear at their optimal for each of the 7 zoom levels I accommodated in my application.

In order to serve the map tiles I needed a tileserver. This was relatively straight forward to develop and required minimal configuration to start serving up my custom tiles having combining it with some of my existing web mapping experience.

Feature Server

The feature server needed to serve details on top of this map was written, once again using a number of online tutorials, from a number of courses, one in particular on *Openlayers*. [9.6 Openlayers] I also went through a couple of tutorials on web mapping with *Leaflet* [9.7 Leaflet] [9.8 Maptimeboston][9.9 Andy Moloney] I also adapted some of the code available from *Esri* [9.10 Esri] I also used code that I had developed on smaller projects previously. The feature server took longer to configure correctly due to the fact that I changed my direction on a number of occasions and there was an element of short term hackery to have parts work in the midst of changes only to find that the effort required to fix them once I had moved the site to a live server was more than originally anticipated. I did accomplish this and its running well and serving layers as requested

9.4 Data Collection

Having interviewed a number of friends who were owners of coffee shops, and following their reassurance that they would happily fill in a '*survey-monkey*' style survey as a way of my collecting data for their shops listing, I set about drawing up a survey. (Appendix 1) I chose two alternative survey providers, *survey monkey* and *google forms*. The final decision to choose *google forms* was simply one of ease of access, and functionality. I duly sent this survey out to 50 of the shops I deemed to be both well spread around Dublin and those with a selection of services and suppliers.

Much to my dismay, I found that only 6 shops responded fully, I had a number of email responses asking what purposes the data was being collected for, a sure sign of our new found data awareness. I did explain in the introduction email the purpose, data usage and retention information of the returned data but the people choosing to open the email and taken the time to respond to me with a question must surely have ignored this opening paragraph. I received one further completed response following my answer to the data question.

As time moved along I realised that I was holding off on some elements of the project while I awaited survey responses, a situation which was not sustainable. I took the decision to mock up a set of test data, using the shop names and locations as they exist and filling in the remainder myself, with a view to correcting all inaccuracies when the finalised returns from the actual coffeeshop owners eventually filled my inbox. To date I have spoken with 6 shop owners and I have 5 completed data surveys, one of the owners did not wish for their shop to be listed on my site, I queried the reason and their response was a mumbled data protection issue. I have since removed their shop.

The data as displayed on the map and database is now almost entirely mock data. If the site were to be pushed out as a commercial enterprise, I feel a meeting with each owner would be required to ensure all fields are filled in and that the owner is satisfied that the data contained therein was correct. I have chosen to retain the larger number of shops and keep my mocked up dataset to show proof of concept for my dissertation rather than a very restricted accurate set of data. I trust this is in order for the purposes of this project.

I also planned on collecting more data at the onset of the project, some of the elements I planned on collecting would not be used for the purposes of this project. This was not personal data, but attributes like number of seats, or take out service, or availability of retail bags of coffee, some of which was originally intended for use, but other elements proved too difficult to implement.

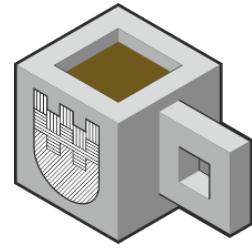
There were many other reasons behind collecting more data than strictly required, it would be easier to have someone sit down in one session and input all of the desired data rather than going back again and again as I chose to implement additional features. Furthermore, the original plan to allow shop owners administer their own data fell by the wayside as time progressed. I would therefore need to match returns from further data surveys manually, I therefore chose only to set up one administrator user responsible for all shops. I also realised that I might be delivering a mid development project to a potential customer and they may choose to no longer participate, leaving me with half filled forms or no database content at all.

10.0 Design

I know design, like art, is entirely subjective. An interface for my application that one person might trumpet as the best thing since sliced bread might have another's teeth on edge. I hope I have designed an interface that is not too intrusive and contains sufficient user feedback to enable intuitive use of the application.

I designed the logo based on the Dublin crest with its 3 castles, not dissimilar to the millennium logo in use on everything from milk bottles to our commemorative coins for the 1988 celebrations. The isometric coffee cup is a nod to my former career and the burnt orange colour is my choice to offset the grey used throughout the rest of the site.

The map tiles I baked in *Tilemill* use the same colour scheme, albeit more washed out and hopefully do not detract from the main function of the application.



*Illustration 1:
Custom Logo*

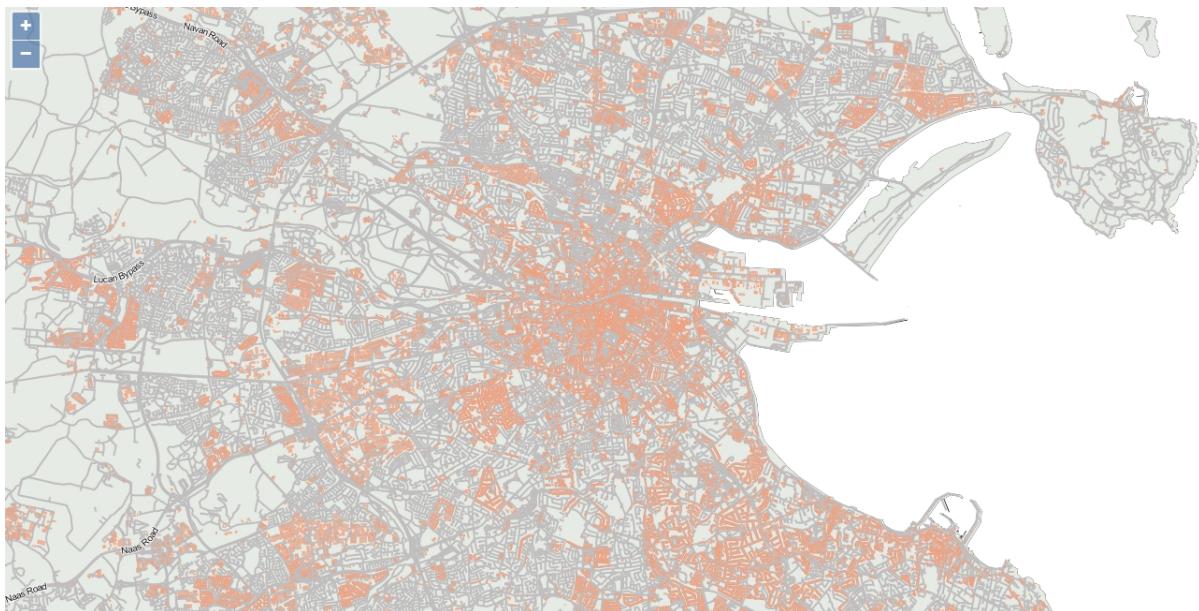


Illustration 2: Map showing custom styling

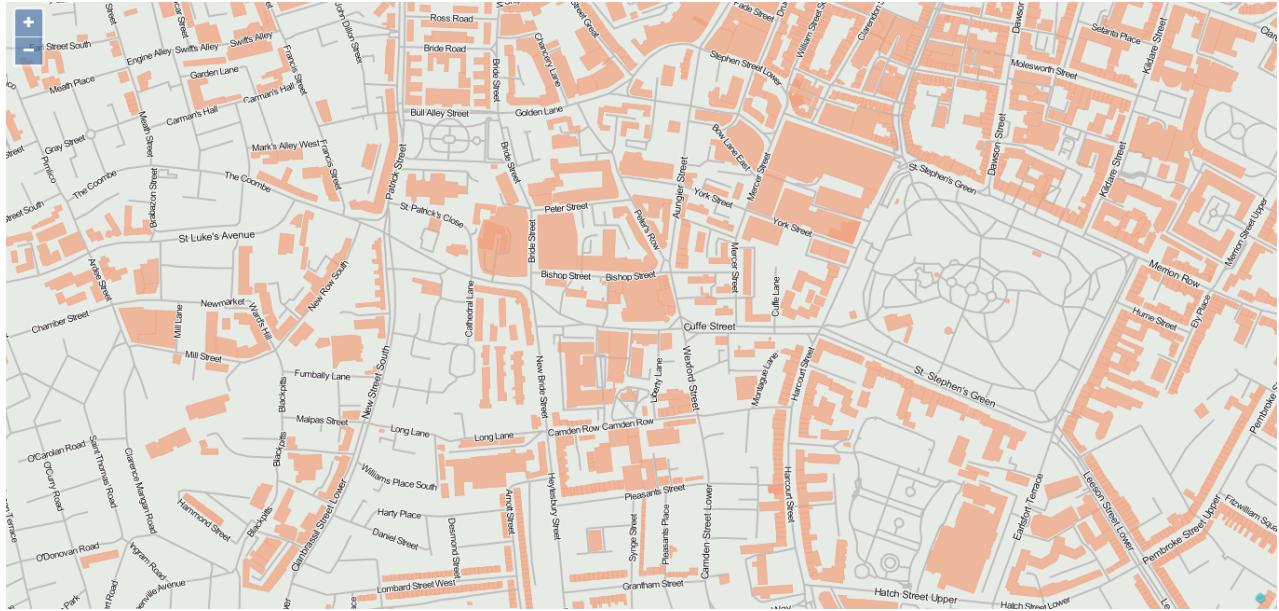


Illustration 3: Zoomed in map showing custom styling

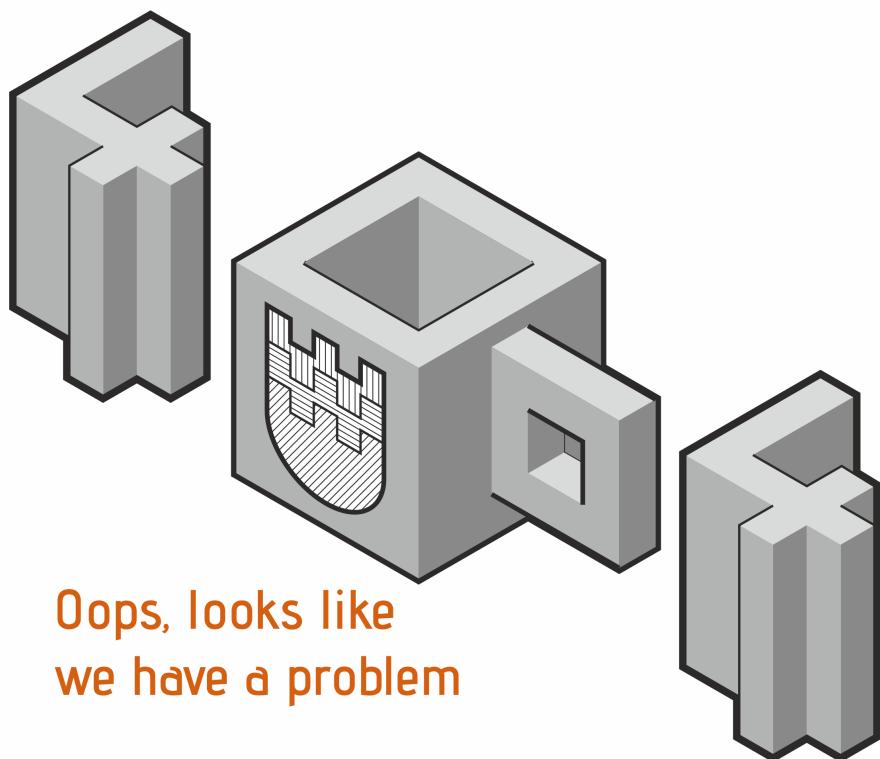
The whole world of cartography is a science unto itself and considering this was not a cartography project, I felt the light touch I gave it was sufficient. I would like to get more in depth with the process of preparing the maps but was very conscious of the time I would find myself sinking into it for minimal return. I had prepared a number of animated svg images animated using css only for use on the site, but my ambition and artistic leaning had to be curbed as I was struggling to find somewhere to use them. I did host a “holding page” on the domain which was svg animated using css, however once the site started to come together I had no longer any use for it.

Custom Artwork

For each shop I designed a custom marker pin, based on their logo / brand. I believe this allows a browser, familiar with the shops in question to zoom directly into their desired location without having to pan around the map clicking on generic marker pins in order to find the shop they desire.



Illustration 4: Sample of Custom Marker Pins



Mockups

I mocked up almost every element of the site prior to attempting it in code.

When a user hovers over a marker pin, the name and address are shown akin to an ALT text in html.

When the user clicks on a marker pin a brief pop up 'bubble' is shown, controlled by the feature server this shows an external photograph of the shop and the name and address with a "...more" option.

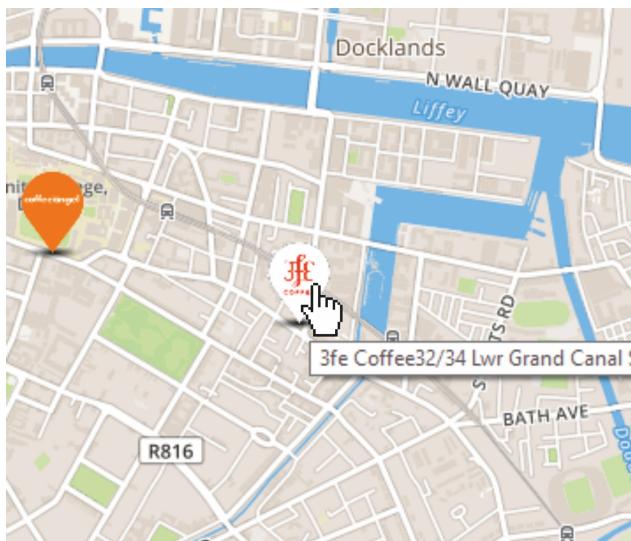


Illustration 6: Mouse over action mockup

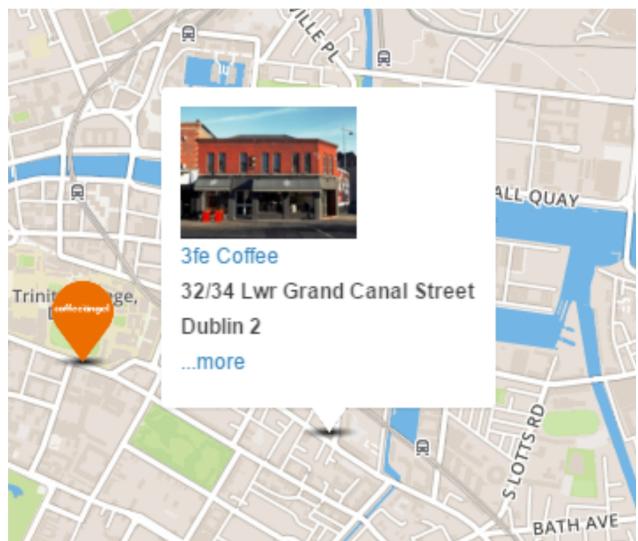


Illustration 7: 'on click' action mockup

I had planned that when a user clicks on the marker they would see information gleaned from the database for that shop in a formatted fashion similar to the mock-up below.

The mockup shows a mobile application interface for a coffee shop named '3fe'. At the top, it displays the shop's name, address ('32 Grand Canal Street Lwr, Dublin 2, Ireland'), and a small map icon. Below this is a photograph of a coffee setup on a table. The main content area contains several sections of text and tables:

- 3fe.ie**
- About us**: A paragraph detailing the shop's history, mentioning its opening in September 2011, its three-story premises, and various coffee-related achievements and services.
- As well as running the shops, we've managed to keep fairly involved in coffee competitions of all kinds. I've been fortunate enough to win the Irish Barista Championships on three occasions and have finished 4th, 4th and 3rd in the World Barista Championships. We've also had latte art champions, a brewers cup champion, and even a Slovakian barista champion amongst our staff and although these are individual awards, there's always a collective effort behind them.**
- In January 2014 we began roasting our own coffee here in Dublin from our brand new roastery in Dublin 1. We sourced a refurbed 50-year-old Probat-UG22 and with the help of Steve from Hasbean, set about training our staff and implementing systems so that we could keep the quality of the coffee to the same level that we had become accustomed to over the years. Today our roastery is a full time operation that sends coffee all over the country**
- pastry**, **breakfast**, **lunch**, **hot food**
- wifi**, **seating**, **service**, **plugs**
- coffee**, **methods**, **machine**, **grinder**

	Mon	Tues	Wed	Thurs	Fri	Sat	Sun	Bank hol
Open	7:30	7:30	7:30	7:30	7:30	9:00	9:00	
Close	17:30	17:30	17:30	17:30	17:30	18:00	18:00	

Illustration 8: Showing mockup of shop details layout

This is one example of many where my artistic sensibilities were forced to take a back seat to my coding ability. I could not make the above appear in any responsive manner apart from simply scaling the information for a smaller screen. I chose to revisit the design for this element simplifying the information I wished to show and also keeping it in line with my technical abilities. The design of the application has taken more than one turn during the course of development. My initial design was a single page application with a map as the main navigation, hiding and showing menus using javascript. I developed an early html prototype for this but found the interface to be less than optimal with interaction not being very intuitive.

I went through three or four different hamburger menu versions before I found one that didn't look atrocious on both a large and small screen. But I still felt that I was restricting myself. The menu system was hidden off screen and the instructions on use themselves required a person to experimentally click to see if it did anything.



Illustration 9: Hamburger menu icon illustration

I then went back to the drawing board so to speak and started pen and paper and then on to graphics software, forgetting about my code for the time being.

I needed a starting point. I researched how others had decided on which path to follow when starting out with a new project. I stumbled upon an article by a much revered front end designer I follow on twitter named Luke Wroblewski [10.2 Wroblewski] I duly bought his book in July [10.3 Wroblewski2] and recognised some of the issues I was already coming up against. I found through a discount code delivered with the book another book by the same publisher on responsive design by Ethan Marcotte [10.4 Marcotte] which I duly bought and between reading and absorbing both books I procrastinated for quite a long time. I found these books to be both helpful and a hindrance. Providing me with the means to work around most problems but the client relationships and design led development environment described was far beyond my abilities and expectations.

In one of Wroblewski's articles he suggests that you carry out a study of existing websites with the client, having them rate their competition on a completely visceral reaction. Considering for the purposes of this project, I was both developer and client, I set about carrying out a similar study.

My visceral reaction study

I carried out a review of 10 map based location websites and marked them out of 10 for each of the following criteria:

- Appearance
- Ease of Use
- Functionality
- Similarity (to my planned design)

The above were less than scientific in nature, however I was basing my opinion on my initial twenty seconds or so interaction with each site which is more than most people allow before making a decision on a website.[10.4 Soskey 2014]

This small study lead me to change direction of how the work flow on my site would operate. I found, that I preferred the sites which introduced the topic to be mapped first and also pointed out more than one piece of functionality which I had left out of my own application. For instance, I had neglected to include a restart button, which I find hugely helpful when dealing with mapping applications.

I also left out the idea of a modal / pop-up for further details of each shop. I had originally envisioned this as being a click through to another page but the modal is ubiquitous in today's internet and once I had incorporated it, even my children understood its use and had inherent knowledge on how to make it go away.

I then produced some mock-ups and screen shots and ran them by a number of people whose opinion I trusted. Once I had tweaked these to the point where I was ready to lock the details down, I began to flesh out the ideas in code.

This is where the design decisions made by pushing pixels around in a graphics editor sought to drive me insane.

Mobile First versus Responsive

I chose not to base my design on the mobile first rationale. After an initial trial run I soon found myself restricted in the usability of a site for desktop too. Whilst I do agree that mobile first ought to be the way an application such as mine should be developed, I felt that I did not possess the requisite skills to carry it out without extensive training.

There are a plethora of books and website alike for championing either one side or the other. I chose to borrow a little of both and built the main grid layout using twitter's bootstrap layout, but did not stick to this as a strict design decision. The fact that I wished to incorporate full screen maps automatically broke this and I ended up picking and choosing elements which delivered functionality I required. The end result is not what my software development colleagues would describe as being elegant or easily maintained, however time and experience constraints meant that choices were made and stuck to. The further I changed the site and developed solutions to issues I was experiencing, the less regard to bandwidth consumption of the resulting application was given. Given more time and a paying client, I feel I could improve both processing and network bandwidth requirements and improve the overall maintainability of the application.

The end result does, however, hold true more or less to my graphic screen mock-ups which I tried to use as my guide throughout. They are based on personal preference and taste.

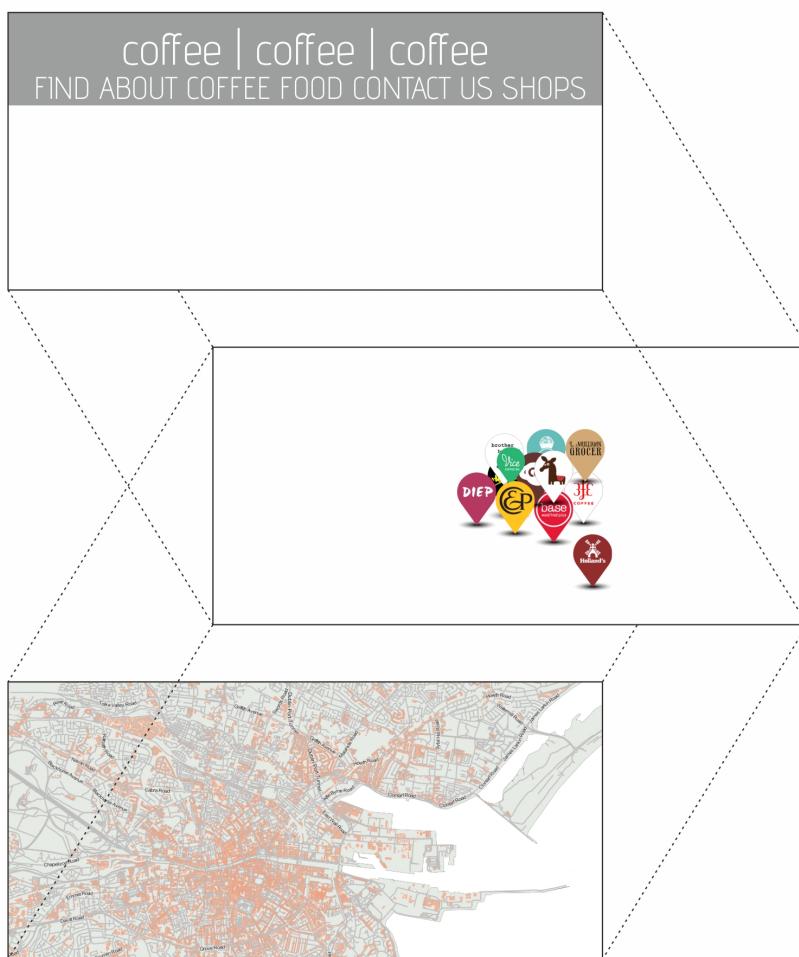
12.0 How the application works

The application is a web delivered site, that is to say, a modern *HTML* website, formatted using *CSS* and relying heavily on *Javascript* both client and server side to accommodate database interface and rendering of maps.

There is a map server serving tiles at the bottom most layer, with a variety of features being served, via *GeoJSON* files above this with *html* being served above this again. The layers are not necessarily in that chronological order but this is a good analogy. The use if the term layers is loose too but seems to be used throughout the industry. In order to allow selections from filtered lists, I have chosen to use static searches.

Each search query results in a layer which can either be displayed or hidden.

The initial database interaction retrieves all of these layers as *GeoJSON* objects and sets their respective visibilities to be hidden. Upon click of any one of the filters the visibility is set to show and the layer is displayed on the map. Whilst this approach is front heavy, it does mean all further interaction is without any bandwidth requirement.

**html & javascript**

responsive html, custom css3 elements for styling and animations to provide responsive feedback to user, javascript elements (some custom, some jquery) served by node.js service running on custom node server. Database connections are handled server side by node.js.

map features

custom markers, coordinates and all of the relevant database entries stored in a postgres database with PostGIS add on installed to allow for use as a routing pathfinder and to allow users to filter entire dataset by specific requirements. All served via feature server running on a separate server.

map tiles

Open Source OpenStreetMap geometry downloaded and trimmed to restrict database search size and file storage requirements. Initially served via mapbox with their custom mss toolkit, subsequently baked into bespoke tiles and styled in tilemill studio using CartoCSS. All served via a simple node.js tilesserver running on the same server as the html node server.

resultant webpage

resulting in fully functioning webpage, with mapping background by way of leaflet lightweight web mapping javascript framework



Illustration 10: How the different elements of web mapping work

Searching

I have selected a number of standardised searches which I believe users of the webpage may find most useful. Having considered the most likely functionality and considering the data I have collected on each shop I have chosen to include, I have drawn up a number of plain English searches which I believe will be most useful.

I have written same in the style of user stories

(this enables me to build SQL queries far easier by forcing me to specify the criteria from the outset and also forms the basis for my test criteria for each search)

The matrix of SQL searches becomes more complex when a query is built up using multiple filtering

Specific to shop owners (this was subsequently removed as I determined that for the purposes of the project, I would be the sole Administrator

- As a **shop owner** I should be able to **amend my details** < required me to develop a CMS system whereby shop owners would be given user names and passwords and allowed change details for their individual shop, but not of others.
- As a **user** I should be able to **log in** (until a user logs in, the system cannot know that they are a shop owner)
- As a **shop owner** I should be able to **log out**
- As a **shop owner** I should be able to **upload** a different **photo** (using web link locations rather than file uploads due to nature and size of variation of photographs possible and restricted size of free db servers) this is accomplished by simply linking to the url for the photo, be it locally uploaded via Git or static path to a web resource.

Specific to users

- As a **user** I should be able to find the **nearest** coffeeshop

There are a number of approaches I tried with this.

Voronoi approach: divide the map into voronoi diagram using shop locations as centroids, when a users location is first found, we already know which polygon they are in and can thus point them immediately to the closest shop) this can be achieved using simple sql queries without the need to route to any individual coffeeshop, however, the nearest shop is not always the shortest travel, take the case of a shop only 50 metres away, but on a different street, requiring the user to walk 200m to the end of the street and 50m along a traversing street and then 200m back to the shop location, only 50m from the starting point, yet the user will have travelled 450m, if routing is used correctly, there may have been a shop far closer for the user that cost less than 450m of travel.

Buffer approach: run a buffer from the users location to all shops, determine this distance when the buffer 'touches' the shop, list them in order and limit to a desired number. For my site I chose twenty, but in reality I believe five might be more useful.

- As a **user** I should be able to **see all shops** on a **map**
- As a **user** I should be able to **select a shop** from **map** and **see details**
- As a **user** I should be able to **go back to the start**
- As a **user** I should be able to see **my location** on a **map**
- As a **user** I should be able to zoom in to map
- As a **user** I should be able to search by different criteria

Search individual criteria Food

- As a **user** I should be able to find the nearest 20 **coffeeshops** that **serve breakfast**
- As a **user** I should be able to find the **nearest 20 coffeeshops** that **sell pastry**
- As a **user** I should be able to find the **nearest 20 coffeeshops** that **serve lunch**
- As a **user** I should be able to find the **nearest 20 coffeeshops** that **serve hot food**

Search by unique criteria

- As a **user** I should be able to find the **nearest coffeeshop** which has **seating**
- As a **user** I should be able to find the **nearest coffeeshop** that **accept credit cards**
- As a **user** I should be able to find the **nearest coffeeshop** that **stocks A or B or C coffee**
- As a **user** I should be able to find the **nearest coffeeshop** that have a **loyalty card plan**

(I'm not convinced that this is a viable selection criteria, but one or two of the owners I interviewed were promoting their business and felt this was a factor which put them above others)

Select by multiple criteria or grouped criteria (not implemented but data has been collected in order to do so)

- As a **user** I should be able to find the **nearest coffeeshop** that is **child friendly**
 - child friendly means that the shop have high chairs and room to wheel a buggy in, its a yes|no criteria (this is self reported by the shops unless an administrator were to visit each shop)
- As a **user** I should be able to find the **nearest coffeeshop** that is **business friendly**
 - business friendly means the shop have **wifi** and **seating** and **table service** and **plug points** for charging computers (these criteria may meet some criticism, but once its explained, I may need to introduce levels of business friendliness) perhaps a 4 * system. For this each 1|0 will be added up and the shop given a criteria in the form of another column in the database table.

Select by opening hours alone (not implemented)

- As a user I will be able to find the nearest coffeeshop which is open now and will be open for the next 30 minutes (assuming that 30 minutes is more than enough time to travel to a selected shop on foot and drink a coffee) (this requires a database query using the current time and day of the week to determine if the shop is open and how long the shop will remain open for.) (if closing time - travel time >30minutes = true, sorted by distance)

Having attempted the above functionality, I chose to remove the option from the menu as I could not get it to work correctly, too much depended on the input method and I would really need to add in an additional input method into the CMS in order to capture opening hours correctly and uniformly.

Query building (these were not implemented)

- As a user I should be able to find the nearest shop which is open, and is child friendly and serves hot food, this will combine multiple SQL queries in order from the above queries and allow for complex filtering and hopefully a useful system.

The above was my wish list of search capabilities for the application, for the most part I think the searches would prove useful to users. However, due to time and technical abilities I reduced the search layers available to be displayed on the map. The searches which made the final cut are

By Shop:

- All coffee shops just displayed on map
- the closest 20 shops (using the users location)
- All shops with wifi
- All shops with seating
- By Coffee (these are similar layers which are associated with a single SQL searches)

- Shops selling each of the food criteria
 - breakfast
 - lunch
 - hot food
 - Pastry

I would like to have been able to implement all of the searches I had planned but time was against me and I chose to display the ability to develop a smaller number of working searches rather than a large number of unfinished ones.

13.0 What the map shows:

The initial map view shows all coffee shop locations contained in the database.

Each of the points on the map is a database record. The front end requests via an SQL statement, “the geom” and additional information for each of the points and render them on the map. “The Geom” is the geometry (X coordinate, Y coordinate) in other words, the geographic reference to where our point exists on the map. These points are also recorded in a manner which is associated with our chosen reference system. (WSG84 epsg:3857 CRS)

With each filter button the the server set invisible to the markers on the map for which the search string doesn't match.

for example : by coffee> 3fe

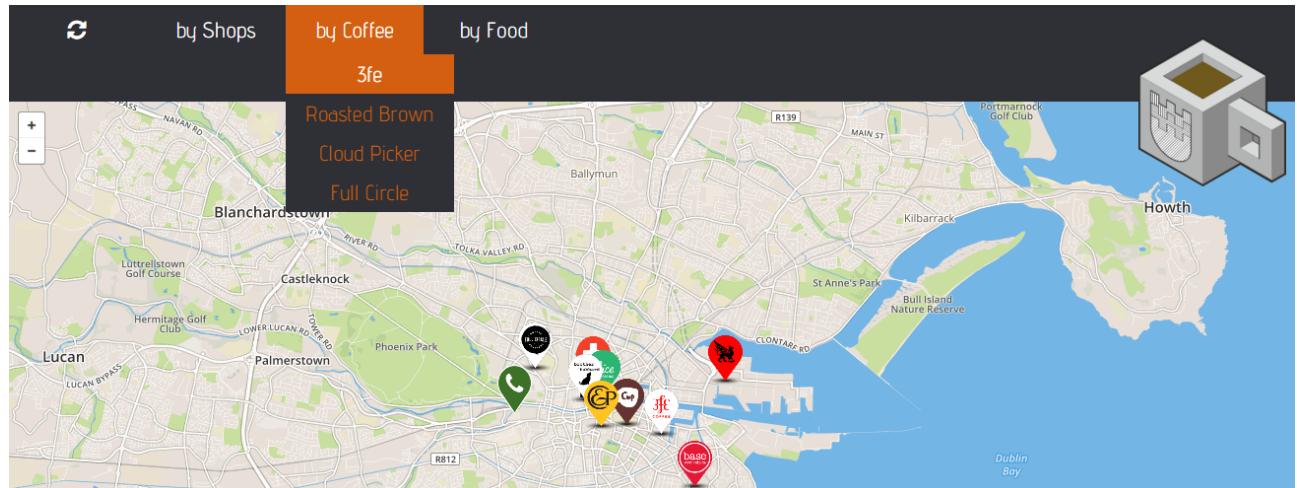


Illustration 11: Map showing selection of shops

The site sets all those shops for whom the 'coffee1' or 'coffee2' or 'coffee3' or 'coffee4' field contains “3fe” as visible, thus effectively removing other shops from the map.

Some of the other search criteria are more complex. One such planned search was, for example:

by shops>business friendly

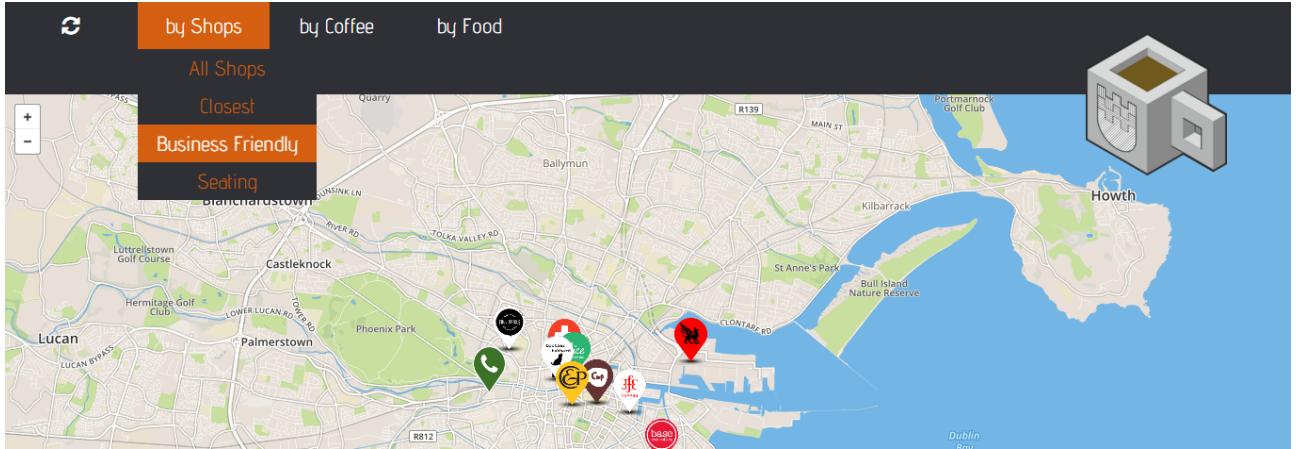


Illustration 12: mockup showing map selection by business friendly status

Having interviewed a number of coffee shop owners, I determined that a shop might be deemed business friendly if it had a number of attributes:

seating	Can someone sit down with their laptop?
wifi	Is wifi available to allow someone to work?
power	Are power sockets available to charge laptops / phones?
Service	Is table service available? its not conducive to staying in a place if you have to either pack up your laptop / notes and take your bag with you to get a refill of coffee or leave them at the table unattended.

In this instance, I had planned to perform a SQL search something like

```
SELECT all FROM 'coffee.shops'
```

```
WHERE seating &&
```

```
WHERE wifi &&
```

```
WHERE power &&
```

```
WHERE service
```

```
ORDER by 'distance'
```

LIMIT to 5;

The site having gone through a number if design iterations and the decision to show the twenty closest shops was taken rather late in the day, having tested the site with live data. I had originally chosen to show only the closest shop by Euclidean distance. If I were to specify the closest shop only, I would add (ORDER BY 'Distance' LIMIT to '1')

I found that the search would sending me 2 miles across the city to find a shop that matched criteria when a similar shop, say, without power sockets, might be within a few metres distance. This is a flaw, which I have a solution to but did not have time enough to implement and test.

The solution involves a trigger upon the change of any record involving the seating, wifi, power or service columns. The trigger adds up the values in the cells and writes this value to the business column, this would allow me a star rating system not giving preference to any one attribute over another.

I could then choose to order my serch results by business rating, returning those shops with a 4 star business rating first, then ordering by distance, meaning that if there are 6 shops matching the criteria, then the closest five are shown, whereas if there is only one 4 star shop, but seven or eight more which are 3 star, then the matching 4 star shop is automatically part of the results table returned, and the four closest 3 star shops make up the returns. I think this is a far more robust solution, but time was against me and I reverted my site to an earlier 'working' version for submission.

I am aware that the complexity of the filtering criteria being used could be increased, but with that comes an increased complexity for the user and considering I anticipated this website being used primarily on a phone sized handset, I felt that increased text and complexity of a menu system might restrict the usage and uptake of the site.

14.0 Testing

Testing was carried out in a number of ways and throughout the development of the application.

Functional Acceptance Testing (FAT)

FAT was carried out as and when I felt I had a piece of functionality ready to release.

FAT involved using the application in the manner as designed. It did not in my case mean extensive regression testing nor did I assess the resultant returned values based on a particular search result.

I drew up test plans based on my user stories with a pass fail for each story. The user stories and the scope changed throughout the course of the project, not unlike a real life project, however I stuck to the user stories which were still applicable and, for the most part, didnt formally test until i knew the feature was functioning correctly.

Infrastructural testing

General testing of the infrastructure was both straight forward and linear in nature as almost each phase required the previous phase to have passed the test.

purchase a linode Virtual machine.

- Can I log onto my web control panel for new server? Yes / No
- Setting up of my Debian system. Did it complete successfully with no errors? Yes / No
- Setting up of new user accounts. Can I log in? Yes / No
- Setting up of Postgres. Can I make a new table? Yes / No
- Setting up of PostGIS. Can I query geom? Yes / No
- Importing data to database. Can I see the data? Yes / No
- Purchasing of a new domain name, setting DNS records to point at new server. Has DNS resolved? Yes / No
- Setting up of mail server on new server. Can I send / Receive mail? Yes / No
- Can I ssh onto new server? Yes / No
- Installed git on new server, performed a pull to new folder. Can I see files Yes / No
- Set up and configured new node webserver. Can I see holding index.html page? Yes / No
- Set up and configuration of feature server. Tested following majority of software development. Does it serve web features? Yes / No
- Locking down of access to new server. Can I ssh in as root? No / Yes
- Can i still ssh in as the correct user and 'su' onto root? Yes / No

Software Testing

The testing of the software was carried out continually on my local development environment My laptop. I ran a local copy of all servers and elements. When I had passed all of the tests locally, I pushed to a git repository I had set up for the purposes of the project using github as the service. I then ssh'd onto the development copy of my server and performed a 'git pull' from the github repository.

I then tested the software in a live environment (albeit staging) There were a number of occurrences where I needed to change paths from absolute to relative in my code so I wasn't clicking a link and returning to the local copy I held on my C:\ This was frustrating after one or two repeats and extremely so after an hour spent 'debugging' my software down a rabbit hole which I had dug for myself.

Overall the software testing was carried out on an almost constant basis.

Change some code, click save, toggle into the browser, click F5:

- Does it look right?
- Does it break anything else?
- Does it have an activity associated?
- If so, does it perform correctly?
- repeat

Browser Environment Testing

I carried out multiple browser tests on:

- Windows 10 (Chrome, Firefox, Edge)
- Windows 7 (Chrome, Firefox)
- Windows XP using Virtualbox (Chrome, Firefox)
- Osx (Safari , Firefox)

Having had to rewrite a lot of my earlier CSS for Internet Explorer and the advent of Windows 10 and its own new 'Edge' browser, I chose to stop development with IE in mind and restricted my testing to Chrome and Firefox for the most part with some exceptions for safari on Apple mobile devices (with mixed successs throughout)

- Apple iphone 4, 4s, 5, 5s (using safari & Chrome) (on latest possible iOS dstrubutions)
- Samsung Galaxy 6 (using default broswer)
- Apple ipad mini2, ipad3 (with safari & chrome)(on latest possible iOS dstrubutions)

User Acceptance Testing

This testing was carried out by handing my phone to one of a number of coffeeshop owners and asking if they would 'play with' the software. I also walked around on some lunch times in Dublin and testing routing on the database, using my handsets location as the starting point and selecting a particular shop as my destination.

Testing that the route determined by the software was 'correct' was carried out by routing from the same origin to the same destination using googles own mapping and routing software on the same handsets. This was not always the case. I spent an amount of time studying if there were an open source option for routing methods more suited to walking routes but was unable to identify one that gave results similar to googles own proprietary software.

Load / Stress testing

I carried out some rudimentary monkey testing of the site itself for reliability. Monkey involves loading the site with a piece of javascript which mimics a user clicking multiple times on various locations at the same time, similar I presume, to how a monkey might play with your site.

I did not carry out any Gremlin testing, this performs a similar role with a little more malice, the site held up well and passed the monkey test under five consecutive users. This was not carried out extensively nor did I load the server with more users than this.

I do understand that the purposes of these tests are to put the application under sufficient stress to determine the breaking point, but I was not willing to do so. These tests can mimic thousands of consecutive users all acting like gremlins and clicking multiple times on links and dragging features and restarting animations etc.

For a live system, further load testing should be carried out.

15.0 Analysis

Having re-read my aims and goals and my earlier IRP1 & IRP2 submissions, I am aware of an elephant in the room which I did not cover in the body of this document. One of my original goals was to implement routing using the open street map vector data, using PGRouting, which I have installed on the server. I had satisfactorily achieved routing using a hobbled together dataset based on dubroads1.sql and dubhighways.sql both datasets which were made available to me for the purposes of my spatial databases module of this course. In disregarding this earlier development work and leaving it aside to be brought back in for a grand finale I left myself without sufficient time to implement it. I tried and failed. I went to my fall back routing solution which entailed sending the user on a round trip to google maps with the pre determined destination based on their selection on my map and their current location as the origin point. I selected pedestrian routing, as I had done so for my pgrouting, thus escaping the issue of having to negotiate one way streets or pedestrianised zones. I believe this to be a sub par solution to my requirement. The fact that I custom developed tiles specifically using all open source data and was in control of what any user might see within my application (ie not allowing google to advertise to my users) made it even more frustrating that I could not implement an open source and clean routing solution.

Beyond that I believe I achieved what I had set out to do, I ticked all of the other boxes, I now have a functional product which delivers as I set out to do. I have also completed this project as if it were a commercial product, adding a facebook page, twitter profile and content management backend system.

16.0 Conclusion

In conclusion, notwithstanding the period from July to October when little was done on the project, during which I still spent a great deal of time sketching and designing the end product, I have delivered what I believe to be a well designed and aesthetically pleasing product.

I have discussed it at large with a wide variety of people involved in both the speciality coffee scene in Dublin and software developers and front end designers. I have learned a huge amount, more I believe, on this module than on any other individual module during the term of my Hons Degree programme.

I did enjoy the process. I do think however that in many aspects, I bit off more than I could chew and the functionality contained is lower than my original aspirations. This was the first time I had set up a server in total from purchase through to production delivery. This was the largest web development project I have undertaken, but now having done so, there are elements that I will re-use, rework and learn from for the future.

I changed my mind too often, or at least I made my mind up too soon and started running towards my end goal before the path was fixed in place. As a result I changed direction more than once, costing me time and energy. I would hope not to do that again to any similar level.

I also learned a great deal along the way. The amount of code I wrote for the application is more than I have written for any other single application. My code has changed over the course of the project, as I changed tack, I adapted earlier code to re-use. I also took on board recommendations and tips from more experienced colleagues with regard to my development. This project represents my first 'proper' coding application. Hitherto my role has been that of Designer and product manager, whilst I did thoroughly enjoy the task, I appreciate now that there is a reason there is usually a team effort involved in an application such as this. Many of the tasks I undertook caused

numerous google searches and trawlings of stack overflow and various other tutorial and help forums. These could have been achieved within minutes by a number of my colleagues for whom these are daily routine tasks. Although having said that, I did it, a little more convoluted in delivery but delivery has been achieved.

Next time, though.....a little more research and planning before implementation.

17.0 Recommendations

Having completed the application using a different delivery methodology from my earliest plans, I would urge caution and recommend more time be spent researching and testing the mechanisms available to you prior to setting out on one path.

18.0 References

(8.1 The London Coffee Guide 2014) Allegra Publications 2014,

ISBN-13: 978-0956775931

(8.2 The London Coffee App) Allegra Publications 2015,

<https://itunes.apple.com/gb/app/the-london-coffee-guide/id632970327?mt=8>

(9.0 Phonegap)<http://www.phonegap.com>

(9.1 Shaw 2014) Learn Python the hard way, Addison-Wesley Professional; 3 edition

(October 11, 2013. ISBN 978-0321884916

and accompanying online course <http://learnpythonthehardway.org/>

(9.2 Django) Django online tutorial supplied by django foundation

<https://docs.djangoproject.com/en/1.9/intro/>

(9.3 Node) <http://howtonode.org/> an free online platform supported by Tim Caswell and the Node community

(9.4 Heroku) www.heroku.com

(9.5 Mapbox) www.mapbox.com

(9.6 OpenLayers) <http://openlayers.org/en/v3.12.1/doc/tutorials/> accessed from January – December 2015

(9.7 Leaflet) <http://leafletjs.com/examples.html> accessed from January – December 2015

(9.8 maptime boston) <http://maptimeboston.github.io/leaflet-intro/> accessed September-December 2015

(9.9 Andy Moloney) <http://asmalonev.com/2014/01/code/creating-an-interactive-map-with-leaflet-and-openstreetmap/> accessed September-December 2015

(9.10 Esri) <http://esri.github.io/esri-leaflet/> accessed from January – December 2015

(9.11 Tyler Mitchell 2005) Web Mapping Illustrated: Using Open Source GIS Toolkits 1st (first) Edition by Tyler Mitchell published by O'Reilly Media (2005) ISBN: 0-596-00865-1

(10.1 Sosky 2014) Hubspot, 2014, author Ginny Soskey
<http://blog.hubspot.com/marketing/chartbeat-website-engagement-data-nj> accessed November 20th 2015

(10.2 Wroblewski) www.lukew.com
accessed July 2015

(10.3 Wroblewski 2) Mobile First, Luke Wroblewski, Ingram (2011),
ISBN: 978-1-937557-02-7

(10.4 Marcotte) Responsive Design, Ethan Marcotte, Jeremy Keith, Ingram (2014),
ISBN:978-1- 9375571-8-8

19.0 Appendix 1

20.0 Appendix 2

21.0 Appendix 3

Shop details used in the final database:

Name & Address:

3fe	Hollands
32/34 Lwr Grand Canal Street Dublin 2	80 Main St, Bray, Co. Wicklow
All City Records	Indigo and Cloth
4 Crow Street, Temple Bar, Dublin 2	9 Essex Street East, Temple Bar, Dublin 2.
Base Merrion	Kaph
180 Merrion road, Ballsbridge, Dublin 4	31 Drury St, Dublin 2, Ireland
Base Terenure	L Mulligan Grocer
92 Terenure Road East, Dublin 6	18 Stoneybatter, Dublin 7
Blahs Cafe	Little Ass Burrito
The Chocolate Factory, 26 King's Inns St, D1	32A Dawson Street, Dublin 2
Brioche	Roasted Brown
51 Elmwood Avenue, Ranelagh, D6	PROJECT ARTS CENTRE, 39 Essex Street
Brother Hubbard	East, Dublin 2
153 Capel Street, Dublin 1,	Science Gallery
Clement and Pekoe	Naughton Institute, Dublin, Trinity Technology &
50 South William St, Dublin 2, Ireland	Enterprise Campus, Pearse St, Dublin 2
Coffeeangel CHQ	Sip & Slurp
Sean O'Casey Bridge, Custom House Quay,	67 Charlemont St, Dublin 2
Dublin, D01 KF84	Sister Sadie
Coffeeangel NWQ	46a Harrington St, Dublin 8
North Wall Quay, facing the new National	Tamp & Stitch
Conference Centre	Unit 2 Scarlett Row, Essex Street, Dublin 2
Coffeeangel PSL	The Fumbally
27 Pembroke Street Lower Dublin D02 T922	Fumbally Lane, Dublin 8
Coffeeangel SAS	Third space
16 South Anne Street Dublin D02 VF29	14, 7 Smithfield, Dublin 7
Coffeeangel TCD	Twofiftysquare
15 Leinster Street Sth, Dublin 2, D02 CY95	Williams Park, Rathmines Rd Lower, Rathmines,
Diep Le Shaker	Dublin, Ireland
55 Pembroke Ln, Dublin 2	Urbun
Fallon & Byrne	Old Bray Road Cabinteely, Dublin 18
11-17 Exchequer St, Dublin 2	Vice Coffee Inc
Foodgame	54 Middle Abbey street, Dublin 1
10 South Lotts Road, Ringsend, Dublin 4	WJ Kavanagh
Full Circle Coffee	4 & 5 Lower Dorset St, Dublin 1
46 Manor Street, Stoneybatter, Dublin 7	Yogism
	23 Dawson Street, Dublin 2