



RONAN RATTY

X13119702

BSC (HONS) IN COMPUTING

PROJECT: CLEVADEALS

FINAL REPORT

CONTENTS

Introduction	6
Executive Summary	6
Aims	7
Technologies	8
C#	8
ASP.net	8
JavaScript	8
Microsoft SQL Server	8
Google Maps API	8
Microsoft Azure	8
Requirements Elicitation and Analysis	9
Introduction	9
Stakeholders List	9
Business Need	10
Requirements Elicitation Techniques	10
Interviews	10
Interview Results	11
Summary	12
System	13
Requirements Specification	13
Functional Requirements	14
<i>Requirement 1: User Registration</i>	14
<i>Requirement 2: User Sign In</i>	15
<i>Requirement 3: Edit User Details</i>	16
<i>Requirement 4: Register New Deal</i>	17
<i>Requirement 5: Edit Existing Deal</i>	18
<i>Requirement 6: Search for Deal</i>	19
<i>Requirement 7: Retrieve navigation instructions</i>	20
<i>Requirement 8: Obtain Current User Location</i>	21
<i>Requirement 9: Receive Alerts When New Deals Are Submitted</i>	22
<i>Requirement 11: Sign Out Of Application</i>	23
Non-Functional Requirements	24
Performance / Response	24
Availability	24
Accessibility	24

Data requirements	24
User requirements	24
Security	25
Maintainability	25
Portability	25
Error checking	25
Concurrency	25
Design / Architecture	26
Class diagram	26
System architecture.....	26
Entity relationship diagram.....	27
Implementation.....	28
Microsoft ASP.net MVC.....	28
Database design and implementation	28
Createing the category database table	28
Creating the offer table	29
User Login.....	31
Obtaining a users location	34
Identifying deal address from postal address	35
Searching by distance.....	37
Email alerts.....	39
Display deal location on google maps.....	41
Adding photos and images.....	42
Publishing database to azure.....	43
Publishing application to azure	45
GUI.....	47
Home Page	47
Category search options	47
Distance search.....	48
Search results.....	48
View deal.....	49
Add Deals	50
Add new category.....	51
Login	51
Registration.....	52
Testing.....	53

Functional Testing – Black Box Testing	53
Usability.....	64
Site overview	65
Competitive websites / applications	65
Primary Needs and objectives.....	65
Targetted site users	65
Application Design	66
Mobile usability.....	66
User Testing	67
User Tests - Business User.....	69
User Tests – End User	71
Summary.....	72
Further Development and Research	73
Commercialisation.....	74
Appendix.....	75
Project Proposal	75
Project Plan.....	76
Requirements specification.....	77
Functional requirements.....	77
Requirement 1: business user regsitration.....	78
Requirement 2: business user sign in.....	79
Requirement 3: edit business user details.....	80
Requirement 4: delete business user	81
Requirement 5: register new promotion.....	82
Requirement 6: edit existing promotion.....	83
Requirement 7: search for promotions	84
Requirement 8: retrieve navigation instructions	85
Requirement 9: sign out of application.....	86
Interview documents	87
End user.....	87
Business user.....	88
Monthly journal.....	89
September	89
October.....	90
November.....	91
December	91

January	92
February	92
March	93
April	94
Additional resources used	95
Leankit kanban	95
GitHub	95
Adobe illustrator	95
GoDaddy	96
References	97

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Ronan Ratty
Student ID: x13119702
Supervisor: Ralf Bierig

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: Ronan Ratty

Date: 09-05-2014

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

INTRODUCTION

EXECUTIVE SUMMARY

ClevaDeals is a web application that is primarily aimed at helping local businesses to attract more customers. Over the past 20 years' internet shopping has grown exponentially and by 2021 it is expected that Irish consumers will spend €14.1 billion online (The Irish Times, 2017). While the growth of online shopping has been hugely beneficial to the consumer and has resulted in much greater ranges of choices of products at times lower prices this has not been a wholly advantageous development.

Of the €14.1 billion that is expected to be spent only half of this is expected to be spent purchasing goods from Irish businesses with the other 50% going to overseas companies (The Irish Times, 2017). Indeed, by 2013 high street retail sales had fallen by 25% from their pre-recession levels and this trend has continued in recent years (The Irish Times, 2017). This not only has a negative effect on the retailers, with that amount of money leaving the local economy it has a knock-on effect to entire locality.

However, there are a number of advantages that local retailers and businesses have over online stores such as the fact that being local customers can call into the store and physically see the product before they purchase it. Also, some products and services are by their nature more suitable to local sales, such as food and drink or items that are large and bulky where delivery costs may be expensive.

To try and attract customer's businesses often try to offer special promotions where for a limited time or at specific periods they make their products and services available at reduced prices. While this is a tried and tested way to attract customers one of the drawbacks of this is while the special offers may be attractive to customers they can be in-effective if the customer does not know about them.

While technology and the internet in particular may appear to be the main protagonist for the continued increase of online sales they can also be utilized by local shops and businesses to counteract the negative effects of the e-commerce revolution. In fact, there is already a cross-over between local and online stores. 42% of consumers while shopping in store use their smart-phones to research products (KYROS, 2017)). This can be to check availability and price in other local stores or to check for other customer reviews before purchasing.

The aim of ClevaDeals is to create a platform that will allow local shops and businesses to promote these special offers while allowing customers to search for them. The challenge, as is the case for any new online application, is to not only create the service but to also make it attractive so people want to use it. To this extent ClevaDeals is not a simple website but it also includes features that make sure it will continually engage with users to ensure they will want to use the application on a regular basis.

Part of this process was also determining that there was a real need for this type of service and to fulfil this requirement market research was carried out. This market research involved identifying a number of questions that were then put to local businesses and consumers. The results of this research confirmed that there is an appetite for this type of product and also helped identify the design and requirements that would need to be included.

The aim of the project was to create a web-based application which allows shops and businesses to advertise their products and services and to also allow customers to search for them. This application was developed to demonstrate the benefits that it could provide to both businesses and consumers. One of these benefits is the wide range of products that can be advertised which means the final product has the potential to be used to a large number of users.

Given the user base that the application is aimed at, which is consumers who are potentially on the move at the time they are accessing the application, there were a number of particular considerations taken. Users could be accessing the application on a number of devices such as smart phones, tablets or desktop computers so it was important that the application was developed to respond to whatever type of device was being used. Also, given that users are willing to wait very short times (recommended no more than 10 seconds)(Nngroup.com, 2017) before they get responses from applications it was imperative that the performance of the application sufficient to meet these requirements.

The application will be aimed at two separate types of users, businesses and end-users, with both having different functionality requirements.

The business user will be able to create an account which enables them to login to the application. After the business user has logged in they will be able to submit new promotions. The end user also be able to register an account and then search for the products and services that have been added by the business user. When adding new promotions, the business user will be required to submit a number of details about the product such as the name, description, cost, location, expiry and contact details as well as submitting a photo or image related to the promotion.

Users, if not registered / logged in to the application will be able to perform an initial search and have limited results returned however they will not be able to view all of the details until they are fully authenticated.

For a better user experience, users will be able to perform different types of searches when searching for products and services. The default search will be a search of all products from a specific category such as "Food & Drink" or "Travel". To allow for more specific searches users will be also be able to refine the search based on distance from where the user currently is. The distance search options will be pre-defined choices such as 1,5,10 kms. The location of the product will also be displayed on Google Maps to provide navigation functionality.

To facilitate this distance based searching the application will use a combination of JavaScript functions and .net classes to identify the products location and also the devices current location. Also an additional .net class "DbGeography" will be used to calculate the distance between both locations.

Finally, users who after searching for and finding products that they want will be able to subscribe to an alert process that will send them an email alert whenever a new deal from that category is added. The application will be developed in C# and ASP.net and published to Azure to ensure it can be accessed online. Also, to provide an easy to remember web URL the domain name

<http://www.clevadeals.com/> has been purchased from GoDaddy and will be used as the website address where users can access the application.

TECHNOLOGIES

Below is an overview of the different technologies that have been used to develop this application.

C#

ClevaDeals is developed using the C# programming language. C# is a type safe object oriented language that enables developers to build a variety of an applications including Windows client application, web services, database application and web applications

ASP.NET

ASP.net is a web development framework from Microsoft that is used to produce dynamic web pages. ASP.net includes an extensive class library that developers who write applications in any language that is compatible with the Common Language Runtime (CLR) can utilise these libraries

JAVASCRIPT

JavaScript is a dynamic computer programming language which most commonly is included in the HTML code of a website or web-application. The main advantage of this is a web page that contains otherwise static HTML code can interact with the browser and perform functions to make dynamic web pages.

MICROSOFT SQL SERVER

Microsoft SQL Server is Microsoft's Relational Database Management System (RBDMS) product. MS SQL is a product that provides the basic function of storing and retrieving data as required by other compatible applications. MS SQL is available as a number of different releases; SQL Standard 2014 has been used for this project.

GOOGLE MAPS API

Google Maps is an Application Programmable Interface is coded in JavaScript and XML. Google Maps can be integrated into websites and applications and enables developers to display maps of any location around the world. Using co-ordinates developers can also identify any point on map and display it on the webpage to be used for services such as navigation

An additional Google API named Geocoding has also been used with this application. Geocoding allows a locations co-ordinates to be obtained by the locations postal address.

MICROSOFT AZURE

Azure from Microsoft is a Cloud Services Platform that has support for a wide number of services such as application hosting, mobile services, storage and database services plus many more. For this project the completed application has been published to Azure to make it available via the internet.

INTRODUCTION

The requirements elicitation and analysis process is vital to the overall project as it can be used to investigate whether there is a valid business need for the application to be developed. This business need can only be validated by involving relevant stakeholders in the process and obtaining their recommendations as to what, if any, functionality is needed or can add value to the final product.

As this project is not being developed with one single customer in mind this added a level of complexity to the process of eliciting the requirements. If this project was being developed on the request of a customer, they would be expected to provide resources to answer questions and take part in any analysis that is required.

However, as no single customer is available multiple stakeholders had to be identified and approached.

STAKEHOLDERS LIST

As a result, the following stakeholders have been identified:

1. Retail / Business Owners

The business owners will play an important role in this. The initial questionnaire which will investigate how the businesses are currently targeting local consumers. They can also validate that they see a use for the application and if they would be willing to use the service.

2. Consumers

Consumers will also play an important role in this as they will be able to provide feedback around areas such as do you have a preference for local or online shopping and what influences those preferences.

3. Application Developers

The application developer will be required to analyse the results of this process to help him to create a set of functional and non-functional requirements that will enable him to produce an application that meets these requirements.

4. Project Supervisor

The project supervisor must provide on-going support and advice as to how the project is progressing and suggest areas where it can be improved

BUSINESS NEED

As previously mentioned, there are a number of statistics to show how local business and retailers are being negatively affected by the competition posed by online stores. While there is an obvious need for these businesses to attempt to counteract these affects it is up to this requirement and elicitation process to try and identify what can be done and try to produce a solution that can be used.

REQUIREMENTS ELICITATION TECHNIQUES

The first step in this process was to determine what techniques would be used to gather the required information.

INTERVIEWS

The first technique used was asking businesses to complete a Research Interview. This involved identifying 10 questions that were used to obtain an overall view of how a section of local businesses were currently using online services to promote their products, were they being successful and what they would be willing to do to improve the service.

The questions that the businesses were asked are as below:

1. Do you have an online presence? If Yes: what type: Facebook / own website / etc.
2. Who manages the website site / app for you?
3. Have you ever sold goods via an online “Deals” website such as “GroupOn”? Yes: No?
4. Has your online advertising been successful? Yes: Why? No: Why Not?
5. Do you think you need more local visibility to attract local customers? Yes: Why? No: Not?
6. Do you ever use special promotions / sales to try and attract more customers? If so, have they been successful? Yes: Why? No: Why Not?
7. How do you promote these offers / sales?
8. Would you be willing to use an online app to advertise these promotions aimed specially at local customers?
9. Would you be willing to pay for this type of service?
10. If Yes, would you prefer to pay a monthly subscription or a single yearly fee?

Next a similar exercise was carried out aimed at consumers. The aim of this procedure was to obtain information related to their approach to shopping online versus shopping locally and what would influence their decisions for choosing one over the other.

The questions that the consumers were asked are as below:

1. Have you ever made an online purchase?
2. Yes: What have you purchased online? What, if anything, would you not purchase online?
3. If you could find the same products / services local would you shop prefer to shop locally?
4. Do you ever search online for local retailers or do you simply browse local stores?
5. Generally, how do you find the experience of shopping locally?
6. How do you shop online? Direct with retailer’s websites? eBay / Amazon? Deal Site such as GroupOn?

7. Do you ever “Impulse Buy” online? If so: what factors would make you make an impulse purchase?
8. What device do you use to shop online? PC / Laptop / Phone / Tablet / Other?
9. How do you find the experience of online shopping? Why?
10. How long are you willing to wait for delivery or an online purchase?
11. How much delivery charge are you willing to pay for delivery of online items?

INTERVIEW RESULTS

In total 20 consumers and 10 businesses completed the questionnaires. The appendix section of this report contains completed questions from both target groups.

The results have been analysed and arranged into three separate categories that can be used to determine what functionality is required in the application.

Results Set 1:

The first set of results are grouped under the heading **“Why would you prefer to shop locally?”**

This analysis resulted in extremely important information as it detailed the main factors as to why people would prefer to shop locally. These are the main points that the application will need to capitalise on and to ensure that it pushes these features to consumers to ensure they can see the value of using the application



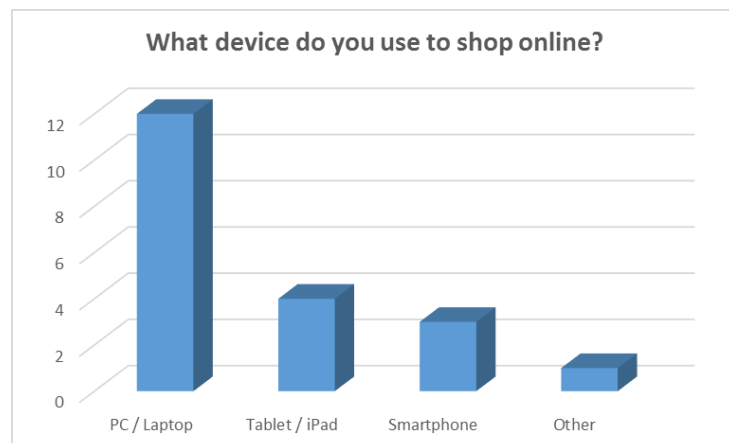
Results Set 2:

The next set of results are equally important as they show what people see as barriers to shopping locally and these are exactly what we want to overcome. By using the ClevaDeals application to address these concerns we can make shopping locally more attractive



Results Sets 3:

The last set of results provide an insight as to what type of device people tend to use when shopping online. It was surprising to see PC/Laptop weighted so highly.



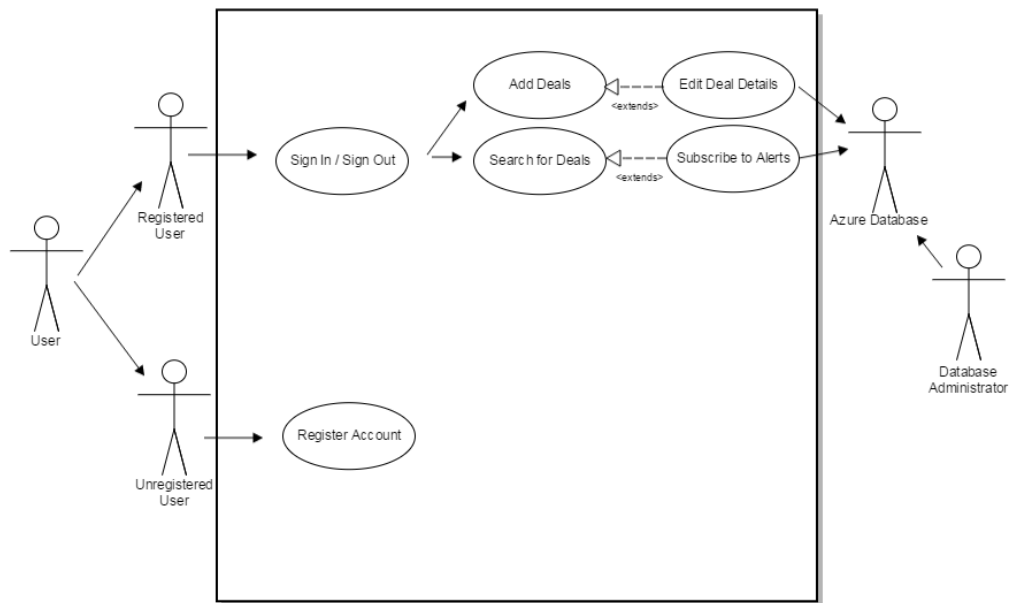
SUMMARY

Overall the requirements and elicitation process proved to a very valuable process and highlighted a number of factors that backup the business case for this type of application. The feedback from the "Results Set #2" is especially important as these are factors that ClevaDeals can address. It is important that the requirements specifications include these.

Also, although the interviews also suggest that PC/Laptop are what people mostly used to complete their on-line purchases the ClevaDeals app will be aimed equally at users of mobile devices.

The existing requirements specification that were determined in the early stages of this project have now been reviewed. The reason for this is as the project progressed it became clear that while some of the requirements that had been identified were still valid others were not. Also, some additional requirements have been added.

Below is an updated use case diagram for the ClevaDeals application:



FUNCTIONAL REQUIREMENTS

REQUIREMENT 1: USER REGISTRATION

- Description

This requirement is related to an unregistered user who needs to register an account to allow them to advertise and search for promotions on the web application. Users also need to log in to the application to be able to subscribe to the “New Deal” alerts.

- Use Case

Below is the user case for the User Registration Process

Description / Scope	Allows and unregistered user to register a new account
Pre-Condition	The user does not currently have an account registered with the application
Activation	When the user click the “Register” button on the main home page
Main Flow	<ul style="list-style-type: none">- Unregistered user clicks on the “Register” button.- User then completed the registration form entering all required details- Application verifies all required information has been provided- User is then added as a “Registered User” and can now login and access features of the application such as submitting new deals and subscribing to deal alerts
Alternative Flow	<ul style="list-style-type: none">- An “Unregistered” user does not supply all required details to complete the registration process- Application displays a friendly error message detailing what the error is.- User re-enters the required details.
Termination	<ul style="list-style-type: none">- Main Flow: User is successfully registered.- Alternative Flow: User registration fails.
Post Condition	<ul style="list-style-type: none">- Main Flow: Business User can now login, search for and submit new deals.- Alternative Flow: User is returned to home page.

REQUIREMENT 2: USER SIGN IN

- Description

This requirement is related to user who has previously completed the registration process and can login to the application to submit new or manage existing deals that have been posted and also search for deals added by other users.

- Use Case

Below is the Use Case for the User Sign-In Process.

Description / Scope	Allows registered users to sign-in to the application
Pre-Condition	The user has already completed the registration process
Activation	When the user clicks on the "Login" button
Main Flow	<ul style="list-style-type: none">- Registered user clicks on the "Login Button"- User then enters their username and password- Applications validates the username and password against the details stored in the database- User can then login and access the application
Alternative Flow	<ul style="list-style-type: none">- Registered user does not enter the correct username or password- Application will present an error message saying the "details entered were incorrect"- User reenters the login details- Access is granted to the application
Termination	<ul style="list-style-type: none">- Main Flow: User is successfully authenticated- Alternative Flow: Login process fails
Post Condition	<ul style="list-style-type: none">- Main Flow: User can login and submit and manage promotions.- Alternative Flow: User is returned back to the home page.

REQUIREMENT 3: EDIT USER DETAILS

- **Description**
This requirement is related to a user who has previously resisted and logged into the application and they need to edit their previously created profile.
- **Use Case**
Below is the use case for the “Edit User Details” requirement.

Description / Scope	Allows users to edit their previously created profile.
Pre-Condition	The user has successfully logged into the application.
Activation	When the user clicks on the “Edit Profile” option
Main Flow	<ul style="list-style-type: none">- User clicks on the “Edit Profile” button”.- User updated profile details ensuring the details being entered match the data type requirements for validation.- User clicks “Save Details”.- User Profile is saved to the database.
Alternative Flow	<ul style="list-style-type: none">- User does not enter the correct data type to update the profile.- Applications will present an error message asking to re-enter the data.
Termination	<ul style="list-style-type: none">- Main Flow: User profile is updated.- Alternative Flow: User profile is not updated
Post Condition	<ul style="list-style-type: none">- Main Flow: User profile displays updated user details.- Alternative Flow: User profile remains as previously saved.

REQUIREMENT 4: REGISTER NEW DEAL

- **Description**
This requirement is related to an existing user who wants to submit a new deal to be advertised on the application.
- **Use Case**
Below is the Use Case for the Register New Deal Process requirement.

Description / Scope	Allows existing users to submit new deals to the application.
Pre-Condition	The user has registered with the application
Activation	When the user clicks the “Add New ClevaDeal” button.
Main Flow	<ul style="list-style-type: none"> - User clicks on the “Add New ClevaDeal” button. - User enters a title for the new deal”. - User chooses deal type from the cascading menu. - User enters a prices for the deal. - User enters an end date for the deal. - User add postal address. - User adds Facebook address - The use adds a photo or image related to the deal - User clicks “Create Button. - The application determines the longitude and latitude coordinates based on the postal address
Alternative Flow	<ul style="list-style-type: none"> - User chooses an incorrect deal type - User clicks “Create” button.
Termination	<ul style="list-style-type: none"> - Main Flow: New deal is successfully submitted under the correct category. - Alternative Flow: New deal is not submitted under the correct category.
Post Condition	<ul style="list-style-type: none"> - Main Flow: End users can search for the deal that has been advertised. - Alternative Flow: Users cannot search for the new deal as it has not been advertised correctly.

REQUIREMENT 5: EDIT EXISTING DEAL

- Description
This requirement is related to editing or deleting a previously submitted deal
- Use Case
Below is the Use Case for the “Edit Existing Deal” requirement

Description / Scope	Allows users to edit or delete a previously submitted deal.
Pre-Condition	A deal has already been submitted by a registered user.
Activation	The user clicks on the “Edit” button.
Main Flow	<ul style="list-style-type: none">- User clicks on the “edit” button- The application will display the previously entered details- The user then modifies the existing details ensuring they format and data type matches the requirements.- User clicks “ok” to submit the new changes.- Application displays a confirmation message asking the user to confirm they want to save the changes.- New details are saved to the database.
Alternative Flow	<ul style="list-style-type: none">- Updated details do not meet the required data format as required by the application.- On clicking save the application will display an error asking the user to review the updated data.
Termination	<ul style="list-style-type: none">- Main Flow: User clicks on submit.- Alternative Flow: Application rejects the new changes.
Post Condition	<ul style="list-style-type: none">- Main Flow: Deal details are updated or deleted.- Alternative Flow: User is returned to the main home screen.

REQUIREMENT 6: SEARCH FOR DEAL

- **Description**
The requirement is related to an end user performing a search for deals that have been previously submitted by users.
- **Use Case**
Below is the Use Case for the “Search for Deals” requirement.

Description / Scope	User searching for deals that have been previously submitted by a user
Pre-Condition	A deal has been submitted by a user.
Activation	A user launches the application and opens the search option
Main Flow	<ul style="list-style-type: none">- User opens the application- User navigates to “Search for Deals”- User chooses search from the category menu- User choose the distance to limit the search to- User presses the search button- Application returns list of matching deals- User chooses individual deal they want to view.
Alternative Flow	<ul style="list-style-type: none">- User opens the application- User navigates to “Search for Deals”- User does not correctly choose search options from search menus- User presses the search button- The application will return an error message notifying the user they must choose all required options.
Termination	<ul style="list-style-type: none">- Main Flow: User clicks on selected item from the search results to view information about deal.- Alternative Flow: User does not choose all required fields from the search menu
Post Condition	<ul style="list-style-type: none">- Main Flow: The user is presented with a view containing all information about the selected deals including price, location, availability and contact details- Alternative Flow: User is returned to the search menu.

REQUIREMENT 7: RETRIEVE NAVIGATION INSTRUCTIONS

- **Description**
This requirement is related to an end user retrieving directions from the Google Maps API feature on how to get to the location of the business advertising the product or service.
- **Use Case**
Below is the Use Case for the Retrieve Navigation Instructions Requirement.

Description / Scope	User gets directions to the location of the business advertising a particular deal when accessing the applications on a mobile device.
Pre-Condition	User has searched for available deals and retrieved a list of search results.
Activation	User clicks on “Details” from the “Deal View” section.
Main Flow	<ul style="list-style-type: none"> - The user logs into to the application and chooses search criteria from the drop down menus. - The user is then presented with a list of search results. - The user clicks on an individual deal to view more information about it. - The user then clicks on the on-screen map to be presented with directions via Google Maps.
Alternative Flow	<ul style="list-style-type: none"> - The user logs into to the application and chooses search criteria from the cascading drop down menu. - The user is then presented with a list of search results. - The user does not click on an individual deal to view more information about it.
Termination	<ul style="list-style-type: none"> - Main Flow: The user clicks on the on-screen map - Alternative Flow: “The user does not choose to view a result from the returned search list.
Post Condition	<ul style="list-style-type: none"> - Main Flow: The user is presented with directions to the location of the business. - Alternative Flow: The user gets returned to the main search screen.

REQUIREMENT 8: OBTAIN CURRENT USER LOCATION

- Description
This requirement is related to the current user’s location being obtained by the application to allow them to search for deals based on the distance from their location
- Use Case
Below is the Use Case for the Obtain Current User Location Requirement.

Description / Scope	On first navigating to the home page of the application the user’s current location is obtained. The location, determined by the user’s longitude and latitude, is stored for this session and used to calculate the distance between the user and a specific business who has advertised a deal.
Pre-Condition	User has navigated to the web applications URL http://www.clevadeals.com
Activation	User accesses the web application via a web browser
Main Flow	<ul style="list-style-type: none"> - The user opens a web browser and navigates to the application URL - As the application loads it, using client side JavaScript, obtains the current location - The location is stored as string variables
Alternative Flow	<ul style="list-style-type: none"> - The user does not enter the correct URL to access the web application
Termination	<ul style="list-style-type: none"> - Main Flow: The application opens - Alternative Flow: The browser does not open the correct web address and returns a friendly error to the user informing them the address is incorrect.
Post Condition	<ul style="list-style-type: none"> - Main Flow: The application has obtained the current location - Alternative Flow: The user re-enters the URL into the browser

REQUIREMENT 9: RECEIVE ALERTS WHEN NEW DEALS ARE SUBMITTED

- **Description**
This requirement is related to an end user, who after searching for and locating deals that they are interested in, can subscribe to receive alerts when new deals from the same category are added to the application.
- **Use Case**
Below is the use case for the Receive Alerts When New Deals Are Submitted Requirement.

Description / Scope	Users can subscribe to receive alerts for new deals that are added to specific categories
Pre-Condition	User has searched for available deals and retrieved a list of search results.
Activation	User selects "Receive Notifications" tick box.
Main Flow	<ul style="list-style-type: none"> - User searches for deals based on their category and location - A list of deals matching the search is returned - The user clicks on "Details" to view more information about the deal - The user clicks on the "Receive Notifications" tick box - When future deals are added the user receive an email alert.
Alternative Flow	<ul style="list-style-type: none"> - The user does not click in the "Receive Notifications" tick box.
Termination	<ul style="list-style-type: none"> - Main Flow: The user clicks on "Receive Notifications" tick box. - Alternative Flow: "The user does not select the "Retrieve Notifications" tick box
Post Condition	<ul style="list-style-type: none"> - Main Flow: The user receives alerts when new deals are added. - Alternative Flow: The user gets returned to the main search screen.

REQUIREMENT 11: SIGN OUT OF APPLICATION

- **Description**
This requirement is related to a business user signing out of the applications after they have completed their tasks.
- **Use Case**
Below is the Use Case for the “Sign-Out of application functionality.

Description / Scope	The user has completed their session in the application and they can then sign-out.
Pre-Condition	The user has previously registered an account and signed in as an authenticated user
Activation	The business user clicks on the “Log Off” button.
Main Flow	<ul style="list-style-type: none">- The user logs in to the application and performs some tasks.- The user then clicks on the “Log Off” button.- The applications ends this session and logs out the user.
Alternative Flow	<ul style="list-style-type: none">- The user does not click on the “Log Off” button.- The user remains logged in and authenticated in the application.
Termination	<ul style="list-style-type: none">- Main Flow: The user clicks on the “Log Off” button”- Alternative Flow: The user does not click on the “Log Off” button.
Post Condition	<ul style="list-style-type: none">- Main Flow: The user is returned to the main “Sign-In” page- Alternative Flow: The user stays in the main home page as an authenticated user.

NON-FUNCTIONAL REQUIREMENTS

PERFORMANCE / RESPONSE

The application must be responsive in order to promote usability. If there is significant lag in response to user interaction this will cause issues and could potentially result in users not returning to the application. To promote an experience that the application responds instantaneously ideally it would respond in 0.1 seconds, a response time of 1 seconds will stay in time with a user's thought flow and 10 seconds is the absolute max amount of time to be able to keep a user's attention and ensure they stay interested in the application (Nngroup.com, 2017). It is important that the application is hosted on the correct hardware and software infrastructure to ensure this requirement is met. To ensure this requirement is met the application will be fully hosted in Azure.

AVAILABILITY

As this is an online resource that is targeted at users in multiple locations across the country and ultimately the aim is that will be accessible on a global scale it is important for the application to be highly available Microsoft Azure hosting platform will deliver this level of availability and resilience as they are bound by a Service Level Agreement of 99.5% (Azure.microsoft.com, 2017)

ACCESSIBILITY

The application should be easy to understand and navigate through the pages. This will include all key functionality such as the Search Function and Registration & Sign-In. The application must behave in a fluid and responsive manner in order to deliver a high level of service to all users.

DATA REQUIREMENTS

The application data will be stored in a Microsoft SQL Server database. This will provide the basic access to support the CRUD functionality (Create, Read, and Update & Delete) as required by the application. The data will initially be stored in a local instance SQL and when development is complete it will be published to a SQL Server database stored in the Microsoft Azure cloud service.

To access the data, the application will need to have the "web config" connection string modified to ensure it can connect to the Azure database instance

USER REQUIREMENTS

- Hosted application that is accessible from the web
- Retailers / Business are able to register an account
- Ability to Sign-In & Sign-Out (Registered Users)
- Ability to update registered profiles
- Ability to submit new promotions
- Ability to modify / edit / delete promotions
- Granular Search functionality by distance for end users
- Ability to view search results
- Ability for end user to contact vendors once a product / promotion has been identified

- Ability to retrieve directions to location of retailer / business
- Ability for application to send notifications to end users

To access the application users will simply need to use a device that runs a web browser. The browser will need to have access to an active internet connection and be able to make requests via HTML port 80

SECURITY

The application will need to be fully secure as it will be holding personal details such as business names and contact details. As such the application should provide a level of security to protect the personal details of the applications registered users from threats such as SQL Injections. Password hashing will also provide an additional layer of security against packet sniffing threats.

MAINTAINABILITY

As the application develops and matures it is expected that it will require a dedicated support team to ensure that all aspects are maintained i.e. Web Developer, Database Admin etc. There should be an option to contact the application administrator in the event a user has problems with their access. This should be provided by a support email address. In addition, there shall be an option that allows registered users to request a password reminder in the event they are unable to access the application.

PORTABILITY

The web application should be accessible across multiple devices and multiple browsers. The application should resize based on screen size of the device i.e. should operate in a fully responsive fashion. As the application resizes to each screen size it is important that the navigation and layout is maintained to be consistent with the layout from larger screens.

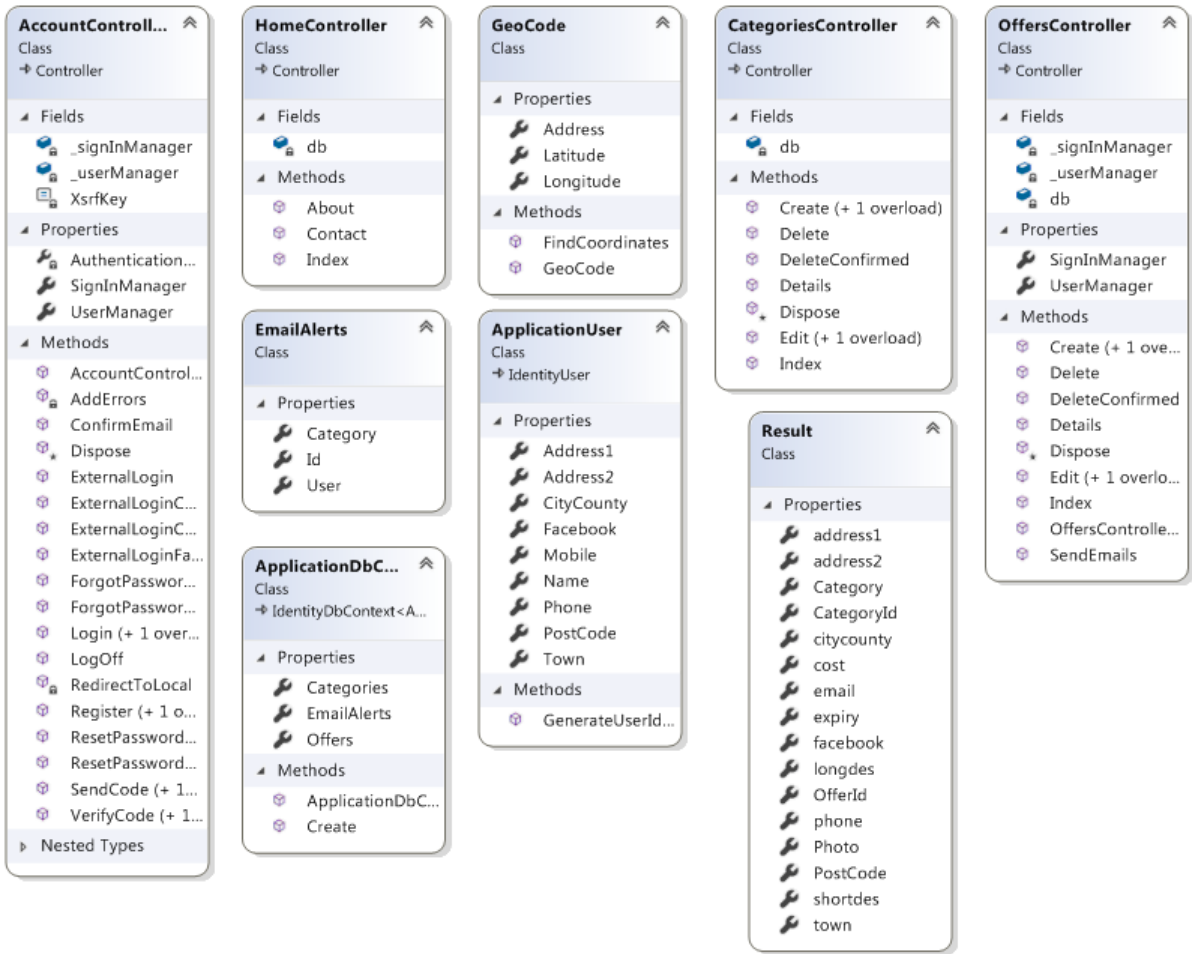
ERROR CHECKING

Error checking will be key in terms of the registration process. The application must recognize that only specific information in specific formats will be accepted e.g. when requesting an email address, it must be entered in the format 'person@domainname'. This will be important for ensuring that the database is populated with only acceptable information. If incorrect information is entered or information is in the wrong syntax (name, promotion description, location, etc.), this will impact the data quality and consequently the accuracy of the search tool.

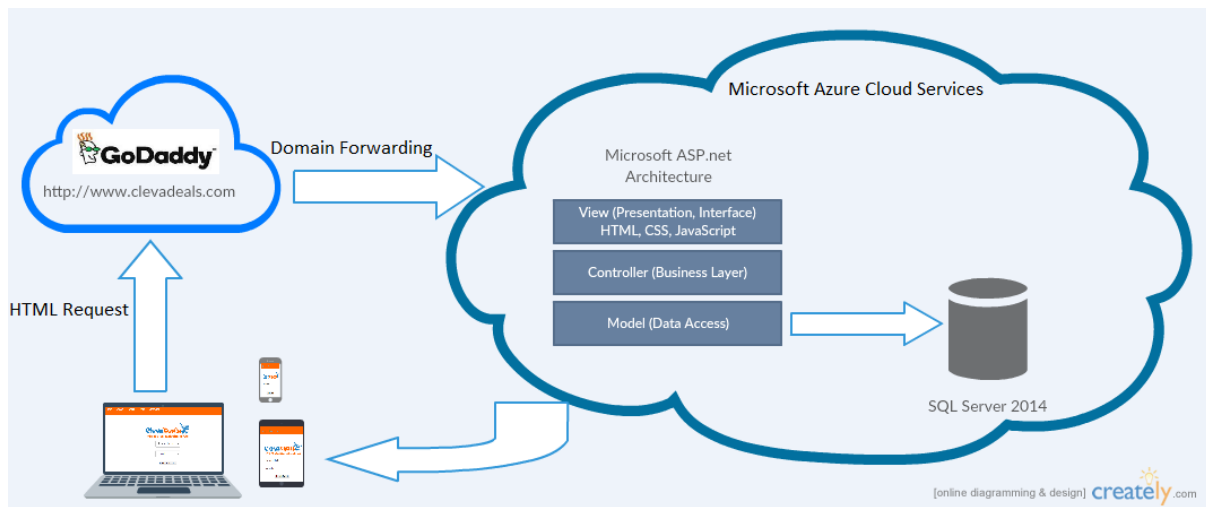
CONCURRENCY

Any successful web application has a fundamental ability to accept multiple users at any one time. This maximizes potential profits and grows the use of the application. The application must be able to accept multiple concurrent registrations, log-ins, searches etc. without user experience being degraded.

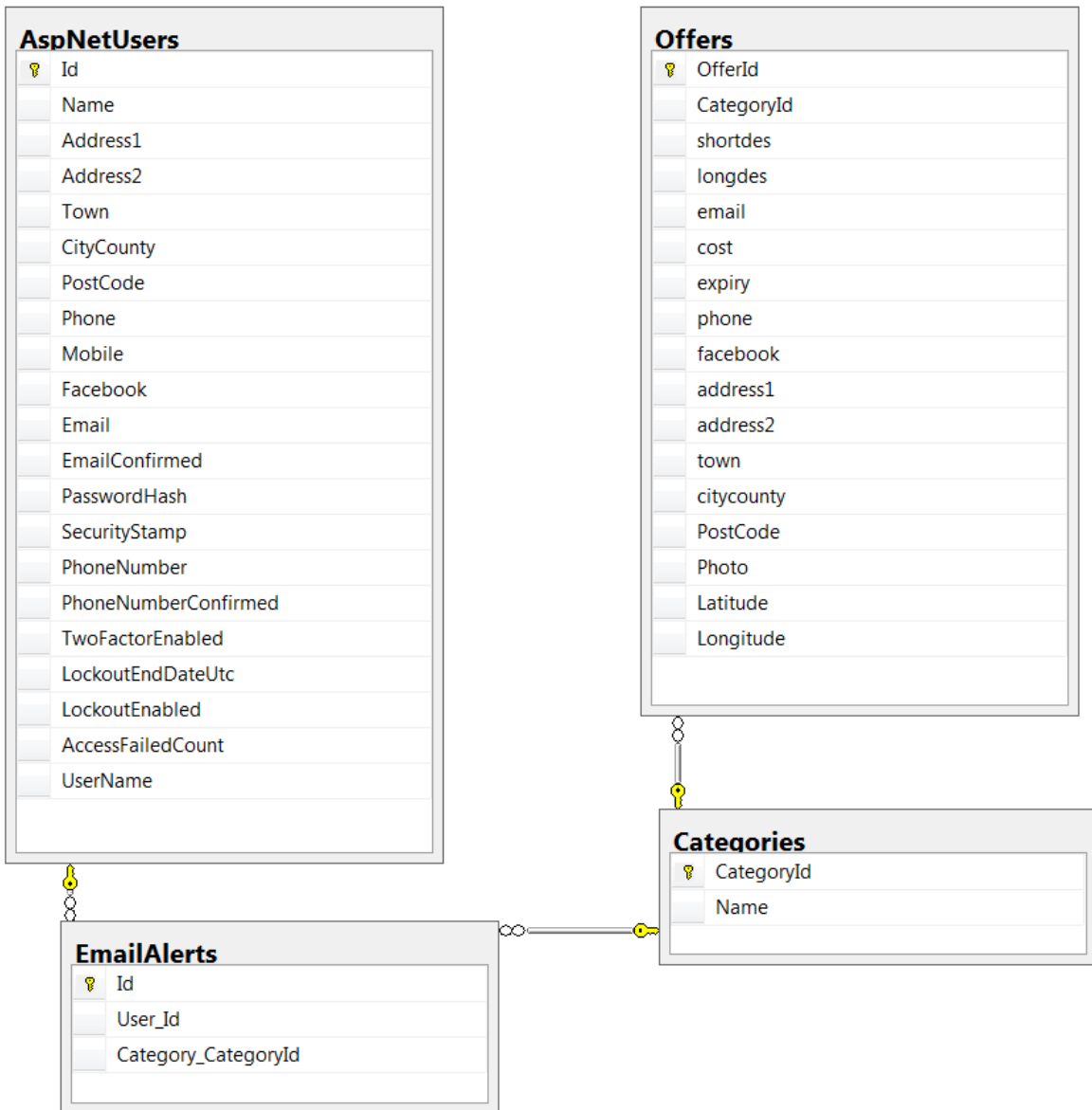
CLASS DIAGRAM



SYSTEM ARCHITECTURE



ENTITY RELATIONSHIP DIAGRAM



IMPLEMENTATION

The implementation process is outlined in the following section, specifically it will detail the relationship between the different technologies used and the order in which the features were developed. The relevant code has been included for each of these functions:

MICROSOFT ASP.NET MVC

This application has been developed in Microsoft's C# development language utilising the ASP.net framework. Microsoft Visual Studio Community Edition 2015 has been used as the IDE.

MVC is a software design pattern used to develop web applications. This pattern isolates the user interface layer from the application logic into three "concerns", namely the Model, View and Controller

The View is the interface that the user will use to interact with the application. The Model is responsible for managing the data of the application, for the ClevaDeals application the model manages the connection to the SQL Server database. Finally, Controller takes user input from the view and after performing any required operations it then passes the request to the Model who modifies the data accordingly.

DATABASE DESIGN AND IMPLEMENTATION

Using the ASP.net Entity Framework as the method of data access for this application a "Code First" approach has been taken to create the database tables. This involves defining what data and structure our database will need to contain.

To complete this a corresponding model class has been created for each table with the relevant relationship between each table defined in the model code also.

Below are snippets of the code used to create the main database tables:

CREATING THE CATEGORY DATABASE TABLE

```
namespace CodeFirst.Models
{
    public class Category
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int CategoryId { get; set; }

        [Required]
        [Display(Name = "Category")]
        public string Name { get; set; }

        public IEnumerable<Offer> Offers { get; set; }
    }
}
```

```
namespace CodeFirst.Models
{
    public class Offer
    {
        public int CategoryId { get; set; }
        public virtual Category Category { get; set; }

        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int OfferId { get; set; }

        [Required]
        [Display(Name="Provider")]
        public string shortdes { get; set; }

        [Required]
        [Display(Name = "Deal Information")]
        public string longdes { get; set; }

        [Required]
        [Display(Name = "Email Address")]
        [DataType(DataType.EmailAddress)]
        public string email { get; set; }

        [Required]
        [Display(Name = "Cost")]
        [DataType(DataType.Currency)]
        public double cost { get; set; }

        [Required]
        [Display(Name = "Expiry")]
        public DateTime expiry { get; set; }

        [Required]
        [Display(Name = "Phone Number")]
        public string phone { get; set; }

        [Required]
        [Display(Name = "Facebook")]
        public string facebook { get; set; }

        [Required]
        [Display(Name = "Address1")]
        public string address1 { get; set; }

        [Display(Name = "Address2")]
        public string address2 { get; set; }
    }
}
```

```

[Required]
[Display(Name = "Town")]
public string town { get; set; }

[Required]
[Display(Name = "City/County")]
public string citycounty { get; set; }

[Display(Name = "PostCode")]
public string PostCode { get; set; }

[DataType(DataType.ImageUrl)]
[Display(Name = "Photo")]
public string Photo { get; set; }

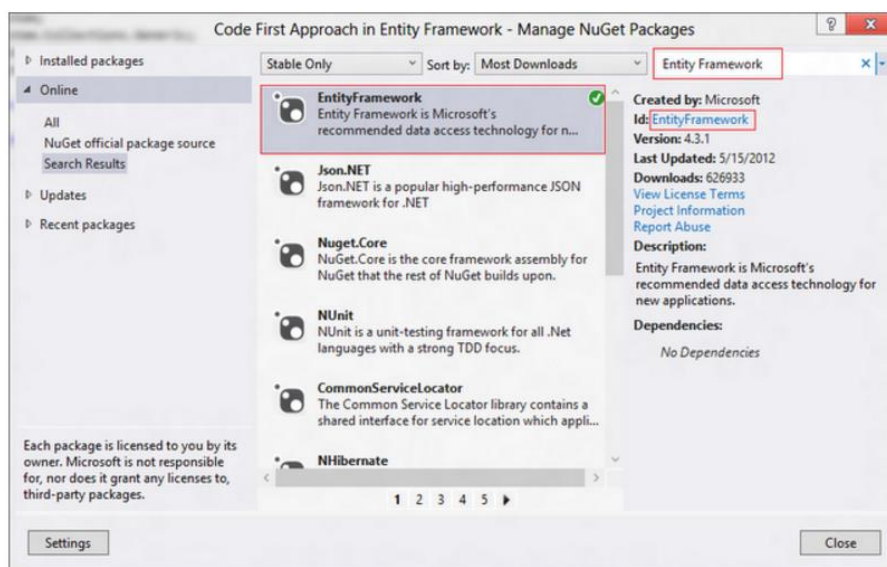
public string Latitude { get; set; }

public string Longitude { get; set; }

}

```

Once the model class has been created it then needs to be “Migrated” using the Entity Frameworks “Nuget” package that is available as part of Visual Studio as below:



To complete the migration there a number of commands to be executed

1. Add-Migration “1” (This creates a migration file)
2. Update-Database (This executes the most recent migration file that was created.)

USER LOGIN

When creating a new MVC application a basic user login system get created by default. This includes a simple form allowing the user to enter their email address and choose a password.

This basic login has been extended to capture additional information such as the users name, address, phone number and Facebook address.

The code snippet below shows the code needed to extend this:

```
namespace CodeFirst.Models
{
    // You can add profile data for the user by adding more properties to your ApplicationUser class,
    please visit http://go.microsoft.com/fwlink/?LinkID=317594 to learn more.
    public class ApplicationUser : IdentityUser
    {
        public string Name { get; set; }
        public string Address1 { get; set; }
        public string Address2 { get; set; }
        public string Town { get; set; }
        public string CityCounty { get; set; }
        public string PostCode { get; set; }
        public string Phone { get; set; }
        public string Mobile { get; set; }
        public string Facebook { get; set; }
    }
}
```

The corresponding HTML view also needed to be updated to present the new options to the user.

```
@using (Html.BeginForm("Register", "Account", FormMethod.Post, new { @class = "form-
horizontal", role = "form" }))
{
    @Html.AntiForgeryToken()
    <h4>Create a new account.</h4>
    <hr />

    <div class="form-group">
        @Html.LabelFor(m => m.Name, new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Name, new { @class = "form-control" })
        </div>
    </div>

    @Html.ValidationSummary("", new { @class = "text-danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.Email, new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Email, new { @class = "form-control" })
        </div>
    </div>
}
```



```

</div>
<div class="form-group">
  @Html.LabelFor(m => m.Password, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.PasswordFor(m => m.Password, new { @class = "form-control" })
  </div>
</div>
<div class="form-group">
  @Html.LabelFor(m => m.Password, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.PasswordFor(m => m.Password, new { @class = "form-control" })
  </div>
</div>
<div class="form-group">
  @Html.LabelFor(m => m.Address1, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.Address1, new { @class = "form-control" })
  </div>
</div>
<div class="form-group">
  @Html.LabelFor(m => m.Address2, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.Address2, new { @class = "form-control" })
  </div>
</div>
<div class="form-group">
  @Html.LabelFor(m => m.Town, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.Town, new { @class = "form-control" })
  </div>
</div>
<div class="form-group">
  @Html.LabelFor(m => m.CityCounty, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.CityCounty, new { @class = "form-control" })
  </div>
</div>
<div class="form-group">
  @Html.LabelFor(m => m.PostCode, new { @class = "col-md-2 control-label" })
  <div class="col-md-10">
    @Html.TextBoxFor(m => m.PostCode, new { @class = "form-control" })
  </div>
</div>
<div class="form-group">
  @Html.LabelFor(m => m.Phone, new { @class = "col-md-2 control-label" })

```

```

        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Phone, new { @class = "form-control" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(m => m.Mobile, new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Mobile, new { @class = "form-control" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(m => m.Facebook, new { @class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Facebook, new { @class = "form-control" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" class="btn btn-default" value="Register" />
        </div>
    </div>
}

</div>

```

OBTAINING A USERS LOCATION

When a user first browses to the applications home page a JavaScript client-side function gets executed that determines the current longitude and latitude co-ordinates which is used to calculate the distance between the user and a selected deal that has been added by another user.

This JavaScript code is below:

```
<script type="text/javascript">

function showLocation(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;
    //alert("longitude : " + longitude + " latitude: " + latitude);
    $("#Latitude").val(latitude);
    $("#Longitude").val(longitude);
}

function errorHandler(err) {
    if (err.code == 1) {
        alert("Error: Access is denied!");
    }

    else if (err.code == 2) {
        ForChrome();
    }
}

function getLocation() {

    if (navigator.geolocation) {
        // timeout at 60000 milliseconds (60 seconds)
        var options = { timeout: 60000 };
        navigator.geolocation.getCurrentPosition(showLocation, errorHandler, options);
    }
    else {
        ForChrome();
    }
}

getLocation();

function ForChrome() {
    $.getJSON("http://freegeoip.net/json/", function (data) {
        var latitude = data.latitude;
        var longitude = data.longitude;
        $("#Latitude").val(latitude);
        $("#Longitude").val(longitude);
    });
}
</script>
```

IDENTIFYING DEAL ADDRESS FROM POSTAL ADDRESS

After a user enters a new deal to be advertised on the application the postal address that has been included is used to determine the addresses longitude and latitude co-ordinates. These co-ordinates are then added to the database record for the deal.

To enable this the C# GeoCode class has been used.

In the OffersController.cs class the postal address is concatenated into a string value "Address" as below:

```
GeoCode oGeo = new GeoCode();
    string Address = offer.address1 + "," + offer.town + "," + offer.citycounty;
    if (!string.IsNullOrEmpty(offer.PostCode))
        Address += "," + offer.PostCode;
    oGeo.Address = Address;
    oGeo.FindCoordinates();
    offer.Latitude = oGeo.Latitude;
    offer.Longitude = oGeo.Longitude;
```

This string is then passed to the GeoCode class who takes the address and using the Google Maps API returns the coordinated to string variables named "Longitude" & "Latitude"

```
public string Latitude { get; set; }
    public string Longitude { get; set; }
    public string Address { get; set; }

    public void FindCoordinates()
    {
        //to Read the Stream
        StreamReader sr = null;

        //Saving the url of the Google API
        string url = String.Format("http://maps.googleapis.com/maps/api/geocode/xml?address=" +
            this.Address + "&sensor=false");

        //Sending the request to Web Client
        WebClient wc = new WebClient();
        try
        {
            sr = new StreamReader(wc.OpenRead(url));
        }
        catch (Exception ex)
        {
            // throw new Exception("The Error Occured" + ex.Message);
        }

        try
        {
            XmlTextReader xmlReader = new XmlTextReader(sr);
```

```
bool latread = false;
bool longread = false;

while (xmlReader.Read())
{
    xmlReader.MoveToElement();
    switch (xmlReader.Name)
    {
        case "lat":

            if (!latread)
            {
                xmlReader.Read();
                this.Latitude = xmlReader.Value.ToString();
                latread = true;
            }
            break;
        case "lng":
            if (!longread)
            {
                xmlReader.Read();
                this.Longitude = xmlReader.Value.ToString();
                longread = true;
            }
            break;
    }
}
}
catch (Exception ex)
{
```

SEARCHING BY DISTANCE

At this point a user has created and submitted a new deal and another user has also navigated to the application, so there are two separate sets of coordinates available for the application to use as reference points to perform the search.

Another C# class called “DbGeography” is used to calculate the distance. The process for this is as follows.

1. The application performs a search to return the details of the deal as requested by the user i.e. by choosing the category such as “Food”.

The user also chooses a distance, the distance values have been hard coded in a Dictionary named “Dis”

```
SearchViewModel model = new SearchViewModel();
    model.Categories = new SelectList(db.Categories, "CategoryId", "Name");
    Dictionary<string, string> dis = new Dictionary<string, string>();
    dis.Add("1", "1km");
    dis.Add("5", "5km");
    dis.Add("10", "10km");
    dis.Add("100", "Anywhere");
    model.Distance = dis;
    return View(model);
}
```

2. This search result returns all details such name, cost, etc... and also the longitude and latitude co-ordinates that has been added to the database record for that user.
3. The application also takes the current user co-ordinates that have been passed from the JavaScript function that has executed when the home page was first accessed.

The code for the above functionality is as below:

```
double longitude = Convert.ToDouble(model.Longitude);
double latitude = Convert.ToDouble(model.Latitude);
DbGeography searchLocation = DbGeography.PointFromText(String.Format("POINT({0} {1})",
longitude, latitude), 4326);
int CategoryId = Convert.ToInt32(model.Category);
double kms = Convert.ToDouble(model.Kms);
/
if (kms == 0) kms = 100;
var nearbyLocations =
    (from product in db.Offers.AsNoTracking()
     where CategoryId == 0 || product.CategoryId == CategoryId &&
     product.expiry.CompareTo(DateTime.Now) >= 0
     select new
     {
         OfferId = product.OfferId,
         CategoryName = product.Category.Name,
         Cost = product.cost,
```

```

    Expiry = product.expiry,
    shortdes = product.shortdes,
    longdes = product.longdes,
    email = product.email,
    phone = product.phone,
    facebook = product.facebook,
    address1 = product.address1,
    address2 = product.address2,
    town = product.town,
    citycounty = product.citycounty,
    photo = "~/Deals" + product.Photo,
    Distance = string.IsNullOrEmpty(product.Longitude) ? 100 :
        searchLocation.Distance(DbGeography.PointFromText("POINT(" + product.Longitude + " " + product.Latitude + ")", 4326))
        / 10000 })
    .OrderBy(location => location.Distance);

```

The above code returns all of the deals that have matched the category search and has also determined the distance between the user's current location and each deal.

Next, the search uses the following code to filter based results based on the option that was chosen from the "Distance" drop down menu i.e. 1km,5km, 10kms or Anywhere.

```

model.list = (from product in nearbyLocations

```

```

    where kms == 100 || product.Distance <= kms
    select new Result
    {
        OfferId = product.OfferId,
        Category = product.CategoryName,
        cost = product.Cost,
        expiry = product.Expiry,
        shortdes = product.shortdes,
        longdes = product.longdes,
        email = product.email,
        phone = product.phone,
        facebook = product.facebook,
        address1 = product.address1,
        address2 = product.address2,
        town = product.town,
        citycounty = product.citycounty,
        Photo = product.photo
    }).ToList();

```

```

return View(model.list.ToList());

```

EMAIL ALERTS

To improve the usability of the application users can, once they find a deal that they are interested in, subscribe to receive alerts whenever any deals get added that match the same category.

To implement this feature a simple tick box has been placed on the “Deal Details” page which only displays if the user has logged in. The code below shows the HTML code and the JavaScript from the Detail view. The JavaScript checks when the box is ticked, it then uses an Ajax function so the page does not have to be fully reloaded and it then posts the data to the “Offers/SendEmails” controller

```
@if (Request.IsAuthenticated)
{
    <p>
        <div class="col-md-6" style="text-align:left">
            @Html.CheckBox("alertForCategory", (bool)ViewBag.IsAlert)
            @Html.Hidden("pkId", (int)ViewBag.Id)
        </div>
    </p>
}
```

The controller then adds the user id and the category id to the Email Alerts table.

```
[Authorize]
public JsonResult SendEmails(int CategoryId, int Id, bool check)
{
    try
    {
        var obj = new EmailAlerts();
        var user = UserManager.FindById(User.Identity.GetUserId()).Id;
        obj.User = db.Users.FirstOrDefault(x => x.Id == user);
        obj.Category = db.Categories.FirstOrDefault(x => x.CategoryId ==
CategoryId);

        if (check == true)
        {
            db.EmailAlerts.Add(obj);
        }
        else
        {
            EmailAlerts e = db.EmailAlerts.Find(Id);
            db.EmailAlerts.Remove(e);
        }
        db.SaveChanges();
    }
    catch (Exception ex)
    {
        throw ex;
    }
    return Json("c", JsonRequestBehavior.AllowGet);
}
```


When a new deal is added that matches the category ID another class named "SendEmail" is called which is then used to send the email using the SMTP details that have been set in the class variables.

```
namespace CodeFirst.Models
{
    public static class SendEmail
    {
        public static void fSendEmail(string ToEmail,string Subject,string Body)
        {
            string FromEmail = "clevadeals@gmail.com";
            string Password = "March2017!";
            using (MailMessage mm = new MailMessage(FromEmail, ToEmail))
            {
                mm.Subject = Subject;
                mm.Body = Body;
                mm.IsBodyHtml = true;
                using (SmtpClient smtp = new SmtpClient())
                {
                    smtp.UseDefaultCredentials = false;
                    smtp.Port = 587;
                    smtp.Host = "smtp.gmail.com";
                    smtp.DeliveryFormat = SmtpDeliveryFormat.SevenBit;
                    smtp.DeliveryMethod = SmtpDeliveryMethod.Network;
                    NetworkCredential NetworkCred = new NetworkCredential(FromEmail,
Password);

                    smtp.Credentials = NetworkCred;
                    smtp.EnableSsl = true;
                    smtp.Send(mm);
                }
            }
        }
    }
}
```

DISPLAY DEAL LOCATION ON GOOGLE MAPS

Once the co-ordinates for the provider offering a selected deal have been located they are passed to a JavaScript who, using the Google Maps API, displays the location which can be used to provide navigation instructions to users.

```
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=AlzaSyCED1rL5spWkgUjjPc5Y5POTqYa3dl31PY
&sensor=false">

    var lat = $("#Latitude").val();
    var lon = $("#Longitude").val();

    var myLatLng = new google.maps.LatLng(lat, lon)
// This is used to center the map to show our markers
    var mapOptions = {
        center: myLatLng,
        zoom: 12,
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        marker: true
    };
    var map = new google.maps.Map(document.getElementById("map_canvas"), mapOptions);
    var marker = new google.maps.Marker({
        position: myLatLng
    });
    marker.setMap(map);
});
```

ADDING PHOTOS AND IMAGES

When adding new deals users can optionally submit photos and images of the deal. To provide this functionality the following code has been developed.

First we set a default image that will display if not custom image is set. Next a unique name is generated from the Guid and it appended to the image file extension type to create a new name for the image (guidstring.file_extension). This new image name is added to the database record for the deal and the actual image is saved in a folder name "deals" located at the application root.

```
var fileName = "no-thumb.jpg";
    if (Photo != null)
    {
        if (Photo.ContentLength > 0)
        {
            fileName = Guid.NewGuid().ToString() + System.IO.Path.GetExtension(Photo.FileName);
            Photo.SaveAs(Path.Combine(Server.MapPath("~/Deals"), fileName));
        }
    }

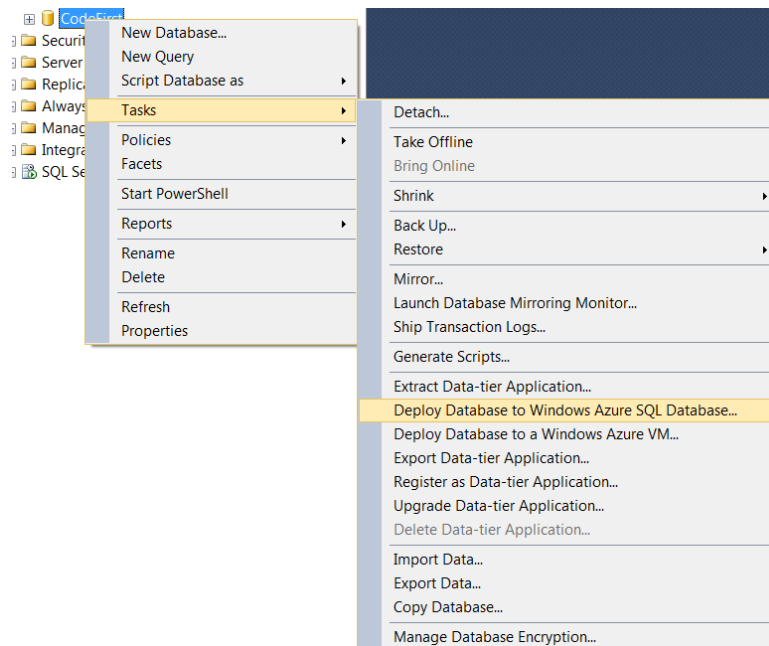
offer.Photo = fileName;
db.Offers.Add(offer);
```

PUBLISHING DATABASE TO AZURE

The application has been published to Microsoft Azure, this will allow users to access the application via the internet. As the application is published to Azure the database also needs to be published to an instance of SQL Server in the Azure Cloud.

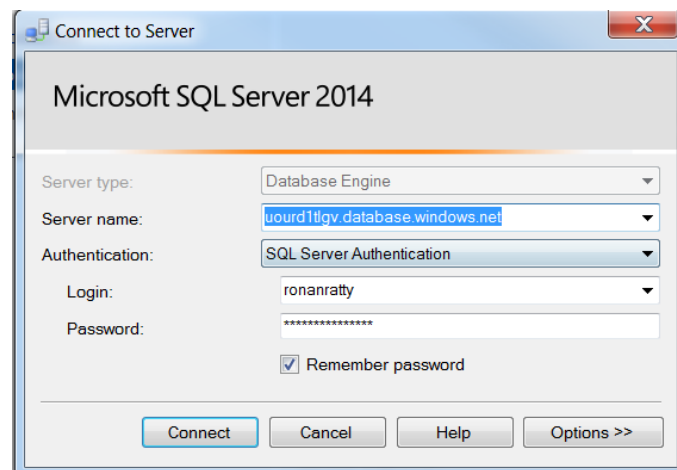
The first step was to register an account with Azure, there are a number of different account types available depending on how much storage or traffic the application will be expected to consume or require. For this project the basic plan was chosen which costs €4 per month.

Once the subscription is active publishing the SQL database is quite straightforward. Using the hosed server name and credentials that were chosen during the sign-up process we can connect to Azure via the local SQL Server application.

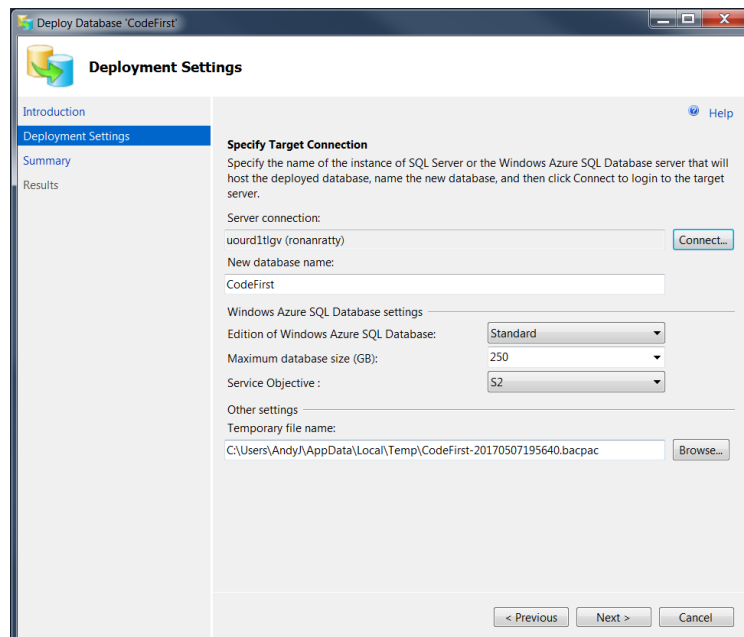


One requirement to note is that the public IP address that we are currently using to connect to Azure on needs to be whitelisted on the Firewall application that is included in the subscription.

Next, we enter the connection details as below:



Finally we set the name that we want to give the new database. After this SQL Server will simply publish the existing database to Azure.

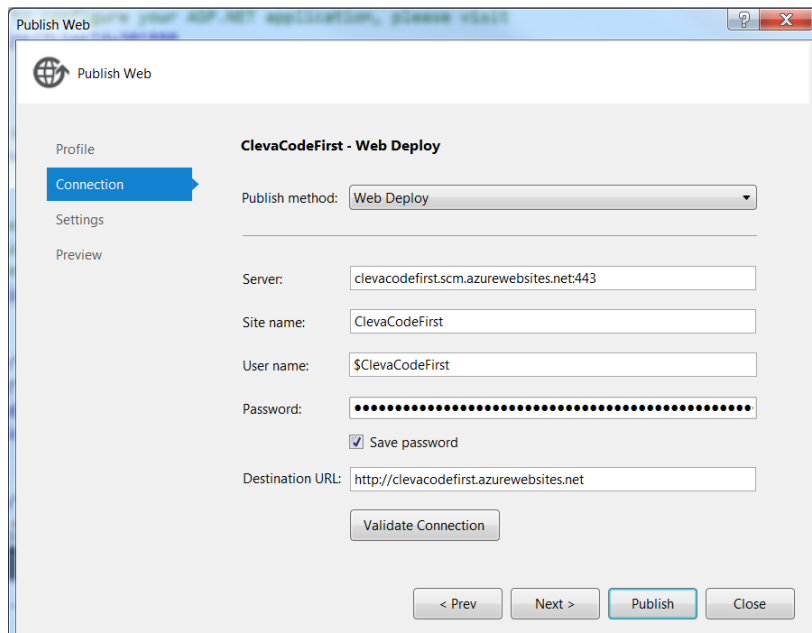
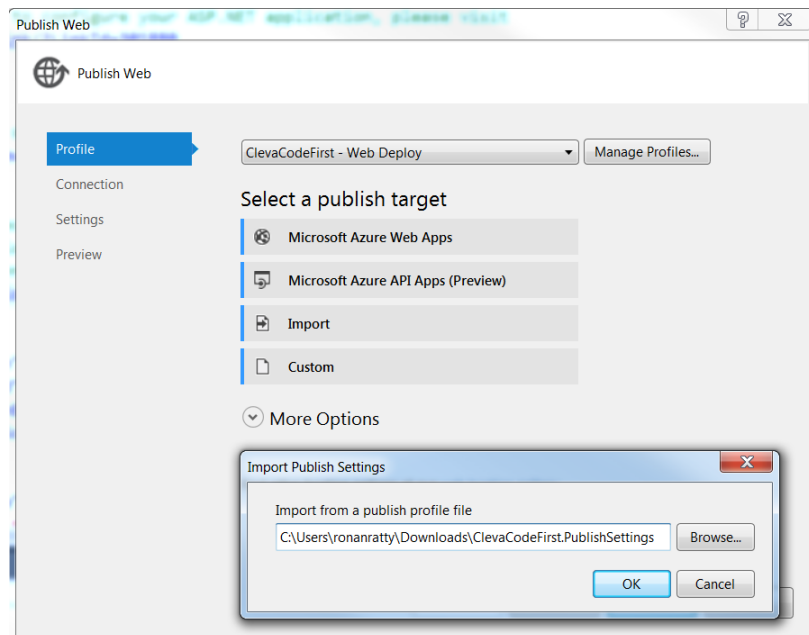


PUBLISHING APPLICATION TO AZURE

Similar to publishing the database to Azure the ClevaDeals application has also been published to the Azure App Service. This uses the same account that was created as part of the SQL Server process. However, making the connection to initiate the push process is more straightforward.

Azure provides a facility which allows the “publish settings” to be downloaded in the format of a .PUBLISHSETTINGS file.

When initiating the publish Visual Studio uses this file to open the connection to the Azure App Service.



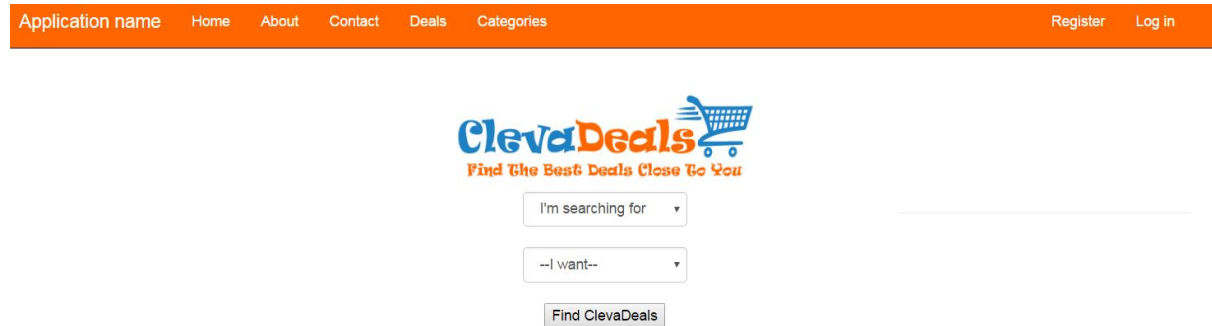
When the application and database are stored locally on the same development server the application is configured to connect to the database using a local connection string. However, when both have been published to Azure the connection string needs to be change to tell the application where the database is located as below:

```
<add name="DefaultConnection"
connectionString="Server=tcp:uourd1tlgv.database.windows.net,1433;Initial
Catalog=CodeFirst;Persist Security Info=False;User
ID=ronanratty;Password=125Blackcastle;MultipleActiveResultSets=False;Encrypt=True;TrustServerC
ertificate=False;Connection Timeout=30;" providerName="System.Data.SqlClient" />
```

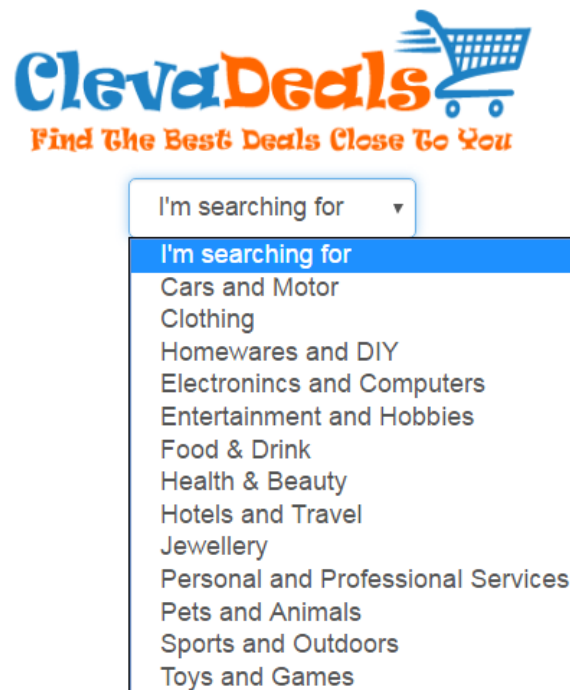
GUI

The section below includes screenshots of all of the key screens in the application.

HOME PAGE



CATEGORY SEARCH OPTIONS



DISTANCE SEARCH



I'm searching for ▾

- I want-- ▾
- I want--
- 1km
- 5km
- 10km
- Anywhere

SEARCH RESULTS



shortdes	longdes	email	cost	expiry	phone	facebook	address1	town	citycounty	
RightPrice	49" Samsung TV for sale	rv@rp.com	1000	10/10/2019 12:00:00 AM	1	1	19 Watergate Street	Navan	Meath	Edit Details Delete
Dominos Pizza	Lunch Meal for One Person	clevadeals@dominos.ie	6	10/10/2017 12:00:00 AM	046908784741	http://fb.com/dominos	Unit 3 Navan Shopping Center	Navan	Meath	Edit Details Delete

ADD NEW CLEVADEAL

VIEW DEAL

Category Food & Drink
Provider Dominos Pizza
Deal Information Lunch Meal for One Person
Email Address clevadeals@dominos.ie
Cost \$6.00
Expiry 10/10/2017 12:00:00 AM
Phone Number 046908784741
Facebook <http://fb.com/dominos>
Address1 Unit 3 Navan Shopping Center
Address2
Town Navan
City/County Meath
PostCode
Photo



Where to Find Us



Receive notifications from any business of this type when they add a new product

[Edit](#) | [Back to List](#)



Offer

CategoryId	<input type="text" value="Food & Drink"/>
Provider	<input type="text" value="Dominos Pizza"/>
Deal Information	<input type="text" value="Lunch Meal for One Person"/>
Email Address	<input type="text" value="clevadeals@dominos.ie"/>
Cost	<input type="text" value="6"/>
Expiry	<input type="text" value="10/10/2017"/>
Phone Number	<input type="text" value="046908784741"/>
Facebook	<input type="text" value="http://fb.com/dominos"/>
Address1	<input type="text" value="Unit 3 Navan Shopping Center"/>
Address2	<input type="text"/>
Town	<input type="text" value="Navan"/>
City/County	<input type="text" value="Meath"/>
PostCode	<input type="text"/>
Photo	<input type="text" value="Choose File pizza3.jpg"/>

Create

ADD NEW CATEGORY



Category	
Cars and Motor	Edit Details Delete
Clothing	Edit Details Delete
Homewares and DIY	Edit Details Delete
Electronics and Computers	Edit Details Delete
Entertainment and Hobbies	Edit Details Delete
Food & Drink	Edit Details Delete
Health & Beauty	Edit Details Delete
Hotels and Travel	Edit Details Delete
Jewellery	Edit Details Delete
Personal and Professional Services	Edit Details Delete
Pets and Animals	Edit Details Delete
Sports and Outdoors	Edit Details Delete
Toys and Games	Edit Details Delete

[ADD NEW CATEGORY](#)

LOGIN



Use a local account to log in.

Email

Password

Remember me?

[Log in](#)

[Register as a new user](#)



Create a new account.

Name	<input type="text"/>
Email	<input type="text"/>
Password	<input type="text"/>
Confirm password	<input type="text"/>
Street Address 1	<input type="text"/>
Street Address 2	<input type="text"/>
Town	<input type="text"/>
City / County	<input type="text"/>
Postcode	<input type="text"/>
Phone	<input type="text"/>
Mobile	<input type="text"/>
Facebook	<input type="text"/>

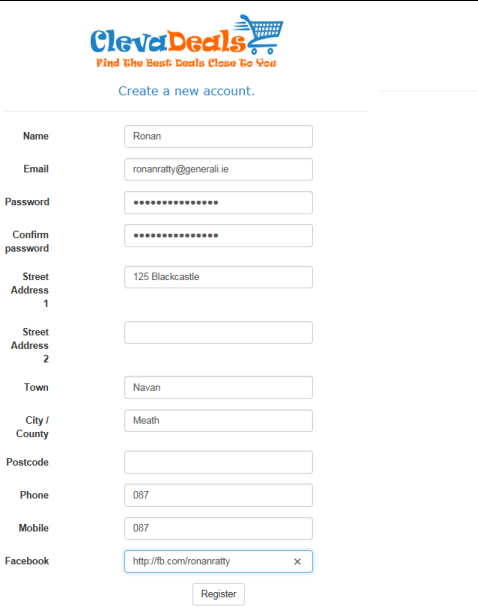
Register

FUNCTIONAL TESTING – BLACK BOX TESTING

The main functions of the application as defined earlier have been tested. Each of these tests are detailed below with each being evaluated based on the expected and actual results of the test.

USER REGISTRATION

Test Description	User Registration
Input Data / Action Performed	<ul style="list-style-type: none"> - Email Address - Password - Name - Address - Phone Number - Facebook Address
Steps Involved	1. Click on the “Sign-Up” link
Expected Result	- User gets added to the application and automatically logged in.
Actual Result	- User has been registered with the application and it then automatically logged in.



ClevaDeals
Find The Best Deals Close To You

[Create a new account.](#)

Name:

Email:

Password:

Confirm password:

Street Address 1:

Street Address 2:

Town:

City / County:

Postcode:


Phone:

Mobile:

Facebook:

Test Description	User Sign-In
Input Data / Action Performed	<ul style="list-style-type: none"> - Email Address - Password
Steps Involved	<ul style="list-style-type: none"> - Click on the "Sign-In" link - Enter email address and password
Expected Result	<ul style="list-style-type: none"> - Users gets logged in and returned to the main search screen
Actual Result	<ul style="list-style-type: none"> - User has been logged in and returned to the main screen

Deals Categories
Hello ronarratty@general.ie! Log off



ClevaDeals
Find The Best Deals Close To You


I'm searching for

-I want-

Test Description	Edit User Details
Input Data / Action Performed	<ul style="list-style-type: none"> - Email Address - Password - Name - Address - Phone Number - Facebook Address
Steps Involved	<ul style="list-style-type: none"> - Sign in to the application - Click on the username displayed - Choose the edit option
Expected Result	<ul style="list-style-type: none"> - User should be able to edit their details and save the changes
Actual Result	<ul style="list-style-type: none"> - User has been able to change their email address and save the updated email address

Test Description	Register New Deal
Input Data / Action Performed	<ul style="list-style-type: none"> - Category type - Provider / Retailer - Deal Information - Email Address - Cost - Expiry - Phone Number - Facebook - Address - Photo
Steps Involved	<ul style="list-style-type: none"> - Login to the Application - Enter the required details - Add a photo / image - Click "Create"
Expected Result	<ul style="list-style-type: none"> - New deal gets added to the application and users can search for the deal
Actual Result	<ul style="list-style-type: none"> - A new deal has been added and user can also search for and return details about the deal
<div style="display: flex; justify-content: space-between; font-size: 0.8em;"> Dominos Pizza Lunch Meal for One Person clevadeals@dominos.ie 6 10/10/2017 12:00:00 AM 046908784741 http://fb.com/dominos Unit 3 Navan Shopping Center Navan Meath Logout Edit Details Delete </div> <div style="text-align: center; margin-top: 10px;"> ADD NEW CLEVADEAL </div>	

Test Description	Edit Existing Deal
Input Data / Action Performed	<ul style="list-style-type: none"> - Category type - Provider / Retailer - Deal Information - Email Address - Cost - Expiry - Phone Number - Facebook - Address - Photo
Steps Involved	<ul style="list-style-type: none"> - Login to application - Search for existing deal - Click on the edit deal option - Edit deal details - Click "Save"
Expected Result	<ul style="list-style-type: none"> - Existing deal details are changed
Actual Result	<ul style="list-style-type: none"> - The deal details (cost) have been changed
<div style="display: flex; justify-content: space-between; font-size: small;"> Dominos Pizza Lunch Meal for Two People cleveadeals@dominos.ie 6 10/10/2017 12:00:00 AM 046908784741 http://fb.com/dominos Unit 3 Navan Shopping Center Navan Meath Edit Details Delete </div> <div style="text-align: center; margin-top: 10px;"> ADD NEW CLEVADEAL </div>	

Test Description	Search For Deal
Input Data / Action Performed	<ul style="list-style-type: none"> - Choose Category
Steps Involved	<ul style="list-style-type: none"> - Login to the application - Choose category from drop down menu - Choose distance to limit the search to - Click "Find ClevaDeals"
Expected Result	<ul style="list-style-type: none"> - A list of matching deals gets returned
Actual Result	<ul style="list-style-type: none"> - A list of deals matching the search options has been returned
	

Test Description	Search For Deal by Distance
Input Data / Action Performed	<ul style="list-style-type: none">- Deal Provider Postal Address- Current User Address- Deal Details
Steps Involved	<ul style="list-style-type: none">- Create a new deal- Observe the address obtained from the GeoCode class- Login as an end-user- Observe the co-ordinates obtained for the user- Note the distance between the two points- Point 1 (Deal Location) has been set as Cork City- Point 2 (User Location) has been set as Navan, Meath- Search using distance filter
Expected Result	<ul style="list-style-type: none">- When choosing 10k the deal should not be returned- When choosing "Anywhere" the deal should be returned
Actual Result	<ul style="list-style-type: none">- For the 10k search no deal has been returned- For the "Anywhere" search the deal from the Cork location has been returned

Test Description	Retrieve Navigation Instructions
Input Data / Action Performed	<ul style="list-style-type: none"> - Perform search for deals - Click “View Deal” from results - View on-screen map
Steps Involved	<ul style="list-style-type: none"> - Perform search for deals - Click “View Deal” from results - View on-screen map
Expected Result	<ul style="list-style-type: none"> - Location of the deal provider gets displayed on the Google Map on the screen
Actual Result	<ul style="list-style-type: none"> - The location of the Provider has been displayed on the screen




OBTAIN CURRENT LOCATION

Test Description	Obtain Current Location
Input Data / Action Performed	<ul style="list-style-type: none">- Launch the application home page
Steps Involved	<ul style="list-style-type: none">- Open an internet browser- Navigate to http://www.clevadeals.com
Expected Result	<ul style="list-style-type: none">- The application should be able to, using the client side JavaScript obtains the current users location
Actual Result	<ul style="list-style-type: none">- The application has obtained the current users location.

Test Description	Subscribe to Alerts for New Deals
Input Data / Action Performed	<ul style="list-style-type: none"> - User chooses to receive alerts when a new deal gets added to a specific category
Steps Involved	<ul style="list-style-type: none"> - User registers and logs into the application - User searches for a deal - User displays the deal and chooses "View Deal" - User selects the "Receive Notifications" tick box
Expected Result	<ul style="list-style-type: none"> - User receives email alerts when a new notification gets added
Actual Result	<ul style="list-style-type: none"> - User has received an email notification.

New clevadeal added Inbox x

 **clevadeals@gmail.com**
to me ▾

Hi,

Act fast! New clevadeal **Dominos Pizza** has been added in Food & Drink.

Click [here to view deal](#)

Regards,
Team Clevadeals

Test Description	Sign Out of the Application
Input Data / Action Performed	<ul style="list-style-type: none">- User clicks on the log out link
Steps Involved	<ul style="list-style-type: none">- User signs in to the application- When finished the session they click on logout
Expected Result	<ul style="list-style-type: none">- User session is finished and user is logged out of the application
Actual Result	<ul style="list-style-type: none">- The user has been logged out of the application

USABILITY

Usability testing is a critical stage in development of any web application. This section details what approaches have been taken to test the final ClevaDeals application.

The application has been evaluated and tested from a usability perspective. Usability is important as it makes the application easier to use and more intuitive while ensuring the requirements and needs of the end users are met. To ensure the test results were as accurate as possible external candidates (1 x end user & 1 x business user) were involved.

The two users involved are:

1. **Business User:** David Gould (Owner at David Gould Car Valeting Limited)

As the owner of David Gould Car Valeting, David was one of the business owners to complete the interview as part of the requirement elicitation process in the early stages of this project. As David had agreed that I could contact him again he was asked to take part in this testing process

Profile:

David Gould Valeting is a successful car valeting company located in Navan, Co Meath. As a local business David has stated that he relies on local trade for nearly 100% of his business. David has also stated that he faces competition from a number of other similar types of businesses and very often uses special offer type promotions to try and promote his company meaning this business is a perfect candidate for the testing exercise.

2. **End User:** Nichola Donagh

Nichola Donagh also completed the interview in the earlier stages of this project and also agreed to be contacted to take part in this testing process

Profile:

Nichola Donagh is married with two children. Nichola lives in Navan and has stated that as she has two young children most of her shopping for any types of products or services are performed in either the local community or online. For this reason, it was felt that Nichola was a perfect candidate for this testing exercise.

The purpose of the usability tests is to provide an analysis of the ClevaDeals application. It will cover aspects such as what are the needs and objectives of the website and have these been fulfilled. This usability test will conclude with both the end user and business user making recommendations as to what, if any, changes could be made.

SITE OVERVIEW

ClevaDeals is a web application that allows businesses and retailers to advertise special promotions that they may be using to try to attract more customers.

COMPETITIVE WEBSITES / APPLICATIONS

As ClevaDeals is unique in its approach it is difficult to identify any particular competitive applications. However, there are some websites that although not targeting the same type of user or providing the same functionality they can be useful to compare against.

1. **GroupOn:**

GroupOn is an e-commerce marketplace where companies can sell products and services at discounted levels. The main difference between ClevaDeals and GroupOn is the flexibility that ClevaDeals offers to businesses. While trading on GroupOn requires businesses to sell high volumes of goods, ClevaDeals can be used for any amount of products. Also, while GroupOn takes up to 30% of the sales price, ClevaDeals is free to use for the business.

2. **DealRush**

DealRush is a similar site to GroupOn but it is aimed solely at the Irish market

PRIMARY NEEDS AND OBJECTIVES

The primary needs and objectives of ClevaDeals are outlined below:

- Hosted application that is accessible from the web
- Retailers / Business are able to register an account
- Ability to Sign-In & Sign-Out (Registered Users)
- Ability to update registered profiles
- Ability to submit new promotions
- Ability to modify / edit / delete promotions
- Granular Search functionality by distance for end users
- Ability to view search results
- Ability for end user to contact vendors once a product / promotion has been identified
- Ability to retrieve directions to location of retailer / business
- Ability for application to send notifications to end users

TARGETTED SITE USERS

There are two main users targeted by this web application

- **Business Users**

Business Users will be the people who use the application to advertise their special promotions.

- **End-Users (Consumers)**

End-Users are the people who the business users are targeting.

SCENARIO 1

David Gould Car Valeting is local business who provides quality car valeting services to customers 7 days per week. David employs three people and mostly they are all very busy. However, at certain times during the week David has noticed that business tends to be quieter and as a result some of his staff are not busy. To try to improve this situation David uses ClevaDeals to run short term promotions where he can simply create a new promotion and offer discounts on his services at specific times. By setting a short expiry date on the promotion he can easily manage how effective the promotion is.

SCENARIO 2

Nichola Donagh is a married mother with two children. Nichola has a busy week scheduled where one of her children is making their First Holy Communion. As a result, Nichola wants to get her car valeted however cost is an issue as, given how expensive this week will be, she does not want to spend much money on the car valet.

Nichola logs on to www.clevadeals.com and searches for “Car and Motor Services” within a limit of 5kms. The search result returns a list of deals and Nichola can see that “David Gould Valeting” is currently offering car valeting services at half price for the rest of today. Nichola used the contact details provided by ClevaDeals to make an appointment with David’s company.

APPLICATION DESIGN

The ClevaDeals application has been designed with ease of use one of the primary considerations. The architecture can be described as “Top Down” where the home page provides a broad overview of what the function of the application however the functionality increases as the user navigates through each area of the application

The colour scheme of the application has also been carefully identified. The two colours (orange and blue) have been chosen as blue has been shown to make users feel more secure (think Citibank or Facebook) while orange invites people to do things as opposed to trying to force them to do it (think Amazon) (www.blog.kissmetrics.com, 2017)

MOBILE USABILITY

As this web application will be accessed by people on mobile devices it is important that it is accessible and responds to different types of screens. As a result, ASP.net MVC utilizes CSS to ensure the applications views respond as expected.

USER TESTING

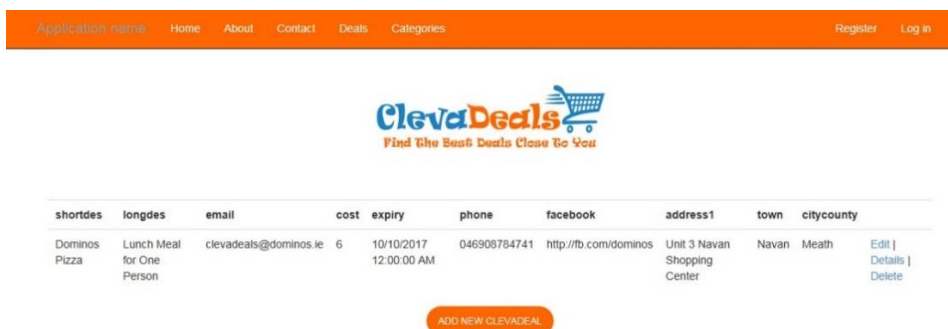
To test the overall design and functionality of the application a number of tests were performed to test the design and functionality from a user's perspective. Both users introduced previously agreed to participate in this process also.

In total there were 4 different techniques employed to perform these user tests, these were:

1. 5 Second Test

"The 5 Second Test" involves showing a user an image of the webpage for 5 seconds and then taking the image away. A number of pre-set questions are then asked to the user and they need to answer based on the image they were shown.

This is a good way to determine the initial impression the application makes on the user. To perform the "5 Second Test" a free online resource was used which is availability from <http://usabilityhub.com/>. This website allows a static image of the website to be displayed and 5 pre-set questions can then be presented to the user, an image of the "Deal Details" screen was used for this.



The screenshot shows the ClevaDeals website interface. At the top is an orange navigation bar with links for Home, About, Contact, Deals, and Categories, along with Register and Log In buttons. Below the navigation bar is the ClevaDeals logo with the tagline "Find The Best Deals Close To You". The main content area displays a table of deals with columns for shortdes, longdes, email, cost, expiry, phone, facebook, address1, town, citycounty, and actions (Edit, Details, Delete). A red button labeled "ADD NEW CLEVADEAL" is positioned below the table.

shortdes	longdes	email	cost	expiry	phone	facebook	address1	town	citycounty	
Dominos Pizza	Lunch Meal for One Person	clevadeals@dominos.ie	6	10/10/2017 12:00:00 AM	046908784741	http://fb.com/dominos	Unit 3 Navan Shopping Center	Navan	Meath	Edit Details Delete

The 5 questions as to the users were:

1. What company does the website belong to?
2. What did you most like about the website?
3. What did you like least about the website?
4. What product or service is being provided?
5. What is the main benefit to you as a customer?

2. Trunk Test

Developed by Steve Krug, the trunk test places a user at any random page on the web application. From here the user should be able to answer the following questions, if they can answer them then the site can be viewed as being designed for optimal navigation:

1. What page am I on?
2. What are the major sections of this site?
3. What are my option at this level?
4. Where am I in the scheme of things?
5. How can I search from here?

3. Think Aloud Test

The think aloud test involves giving a user a particular task to complete on the application. As the user completes the task they are required to talk about how they find the experience and commenting on what they find good or bad. As the user completed this test the session is recorded.

4. Heuristic Evaluation

Heuristic Evaluation is a technique where the application is evaluated against 8 recognised usability principles as opposed to simply asking the user to perform standard tests.

8 recognised usability principles used in this test are:

1. Visibility of System Status
2. Match between the system and the real world
3. User control and freedom
4. Consistency and Standards
5. Error Prevention
6. Recognition rather than recall
7. Flexibility and ease of use
8. Aesthetic and minimalistic design

USER TESTS - BUSINESS USER

5 SECOND TEST – BUSINESS USER

Question	Response
What company does this website belong to?	“A company called ClevaDeals”
What did you most like about the website?	“The colour scheme”
What did you like least about the website?	“The layout is very basic”
What product or service is being provided?	“Pizza”
What is the main benefit to you as a customer of the company?	“What is the main benefit to you as a customer of the company”

TRUNK TEST

For the business user trunk test the “Add New Deal” screen was presented:

ClevaDeals
Find The Best Deals Close To You

Offer

CategoryId:

Provider:

Deal Information:

Email Address:

Cost:

Expiry:

Phone Number:

Facebook:

Address1:

Address2:

Town:

City/County:

PostCode:

Photo:

Question	Response
What page am I on?	The page that allows me to add a new deal
What are the major sections of this site?	Inset details related to the deal I want to add
What are my options at this level?	Add a new deal or use the navigation bar to login or register
Where am I in the scheme of things?	I am at the stage when I need to add a new deal to be added to the application
How can I search?	There is no search functionality on this page

HEURISTIC EVALUATION

Below are the comments from the business user in relation to the heuristic evaluation principles:

Principle	Evaluation
Visibility of system status	It is not very clear where I need to go next, it is assumed that I know where to go so some on screen prompts would be useful
Match between the system and real world	The text and colures are very clear so it is easy to see each part of the page
User control and freedom	There are not many options to choose on each page. It would be used to have more navigation options on each page
Consistency and standards	The standard layout is consistent if however very basic
Error prevention	There are on-screen prompts if required information is not entered as expected or required.
Recognition rather than recall	The consistent use of colours so similar functions – such as the same style and colour for buttons makes each item easily identifiable
Flexibility and ease of user	There are not many options to choose on the application so it is by default very easy to use
Aesthetic and minimalist design	The colours and logo look very good and it creates a very enticing mood when using the website.

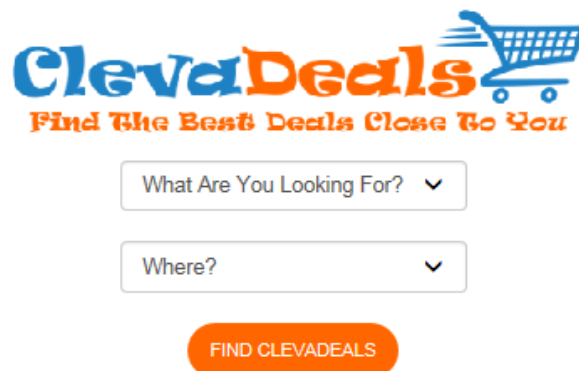
USER TESTS – END USER

5 SECOND TEST – END USER

Question	Response
What company does this website belong to?	ClevaDeals
What did you most like about the website?	The logo
What did you like least about the website?	It's not very appealing, I am not really interested in clicking on other parts of the website
What product or service is being provided?	An online shopping service
What is the main benefit to you as a customer of the company?	I can find products being sold at discounts?

TRUNK TEST

For the end-user trunk test the “Search for new Deals” screen was presented:



Question	Response
What page am I on?	The main search page
What are the major sections of this site?	There are two sections, one where I can select what I want to search for and another to define where I want to perform that search for.
What are my options at this level?	Search for products and register or login to the application
Where am I in the scheme of things?	At the part where I search for deals for at the beginning of the process
How can I search?	By choosing the options from the drop down menus and clicking search

THINK ALOUD TEST

****The think aloud test for this application has been submitted with this Project Document****

HEURISTIC EVALUATION

Below are the comments from the end-user in relation to the heuristic evaluation principles:

Principle	Evaluation
Visibility of system status	From the home page it is clear that I need to perform a search however after that it is not very clear what I need to do.
Match between the system and real world	The text on the pages are informative and easy to read
User control and freedom	It could be easier to move forward and back between pages
Consistency and standards	Each page appears to be a similar layout with similar colours and logos
Error prevention	When searching for products if I don't not select the correct option I am prompted to choose the correct one. The same occurs during the registration process.
Recognition rather than recall	I found it was easy to navigate the application without having to check what options were available.
Flexibility and ease of user	I found the process of logging on and searching for products very easy
Aesthetic and minimalist design	There was no clutter on any of the pages and it was clear as to what I needed to do

SUMMARY

It is very clear from the user's responses to each of the above tests that there is quite a lot of work to be done with the usability aspect of the application however as far as core functionality the black box tests has shown that the functions identified in the earlier stages have been implemented.

From the beginning of this project there were two criteria that defined what the project being developed was going to be and the technologies to be used, these were:

1. To meet the requirements to the BSc in Computing module to produce a final project and its supporting documentation
2. To use this as an opportunity to gain experience in developing an ASP.net MVC application including covering every part of the software development life cycle. The reason for this is will be changing from my current role as a helpdesk technician to a software developer on completion of this course. As my employer uses exclusively .net applications this has been a good opportunity to gain some relevant experience in this.

This project has allowed me to satisfy both of the above criteria however, as this project has progressed and I have received feedback from different people the future development of this project will need to be re-accessed. I have always believed that the functionality of the application, namely allowing retailers and businesses to advertise special promotions and also allow people to search for these, was a very good idea that it has the potential to be commercialised.

On review however it would appear that using ASP.net solely may not be the most appropriate technology to use for this type of application. Although ASP.net is sufficient for accessing the application from a desktop or laptop computer the application runs from a web browser which means that it cannot utilize some of the native features of mobile devices. The two main features that would be most beneficial to access on mobile devices are GPS and Notifications. This would allow for more seamless and accurate identification of location and smarter notifications based on the improved notification options available.

With this in mind the future development of the application will include the following:

1. Development of mobile apps for both Android and IOS.

Some of the benefits that this would bring would be improved GPS location services and notifications. Also, as the app would be installed on the device it would mean that users would only need to authenticate once as opposed to every time they launch the app.

2. Improving the current alerting options of the MVC app.

Currently the only option users have been to subscribe to alerts by manually selecting a tick box on the "Deals Details" screen. This can be improved by implementing more targeted notifications.

This will be achieved by storing more information about how the users are using the app. This will have to include what the user has searched for and also what time they have performed the search. The application will also include a rating system so by performing a search of the database we could identify what deals the users has rated highly and when they performed this search.

We could then use this information to send timed alerts to users related to similar deals around the same time that the previous search was completed.

3. **Developing the GUI**

The GUI was not a primary concern for this project as it was more focused on showing what the potential functionality could be. In future developments it needs to be a priority to increase the appeal to users.

COMMERCIALISATION

As mentioned previously I feel this is an application that could be appeal to and can be used by a wide range of the user. To this extent I have demonstrated the current idea and functionality to a member of the “Meath Enterprise Board”. The idea was well received and they agreed that it does have potential and they are interested in investigating this potential further. Once this module has finished I will be meeting a number of people from both the Meath Enterprise Board plus a member of the Meath County Council who is responsible for supporting local businesses to investigate any supports available.

PROJECT PROPOSAL

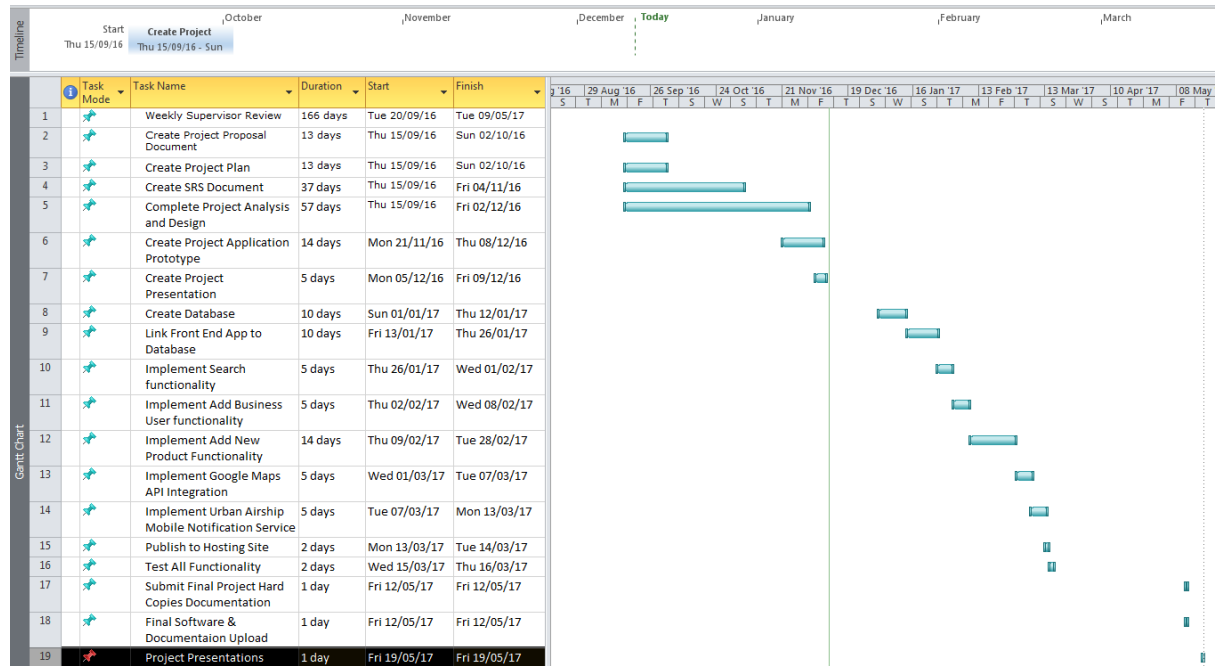
The purpose of this document is set out the reasons why I have chosen to create this application for my 4th year software project and also to document the requirements needed to design and implement the “ClevaDeals.com” web application.

This web application will allow local retailers and businesses to advertise their products and promotions and also provide search functionality to allow users of the application to search for what has been advertised. This is an idea that I have had for a number of years and one that I have always felt that if developed correctly could provide an extremely useful service that consumers and businesses alike will want to use.

There are a number of advantages to shopping locally for products. Later in this document I will cover these in more detail when presenting the results of the market research interview exercise that was completed. Essentially however the two main reasons people like to shop locally is they can view the product before they buy and they can also receive the product immediately at the time of purchase, these were two surprisingly strong results that the interviews returned. Retailers and businesses also can gain an advantage when promoting to local customers as while the customer base is indeed smaller when compared to an online marketplace this type of promotion is not suited to every business type and they can target a customer base that is more suited to what they are trying to sell.

This document will detail the intended purpose and features of the application, the constraints against which the application will operate, the user interfaces as well as the key application functionality. Initially this document is intended for ourselves – the project team – as well as our lecturer so as to set out a clear guideline on what we are looking to achieve.

PROJECT PLAN

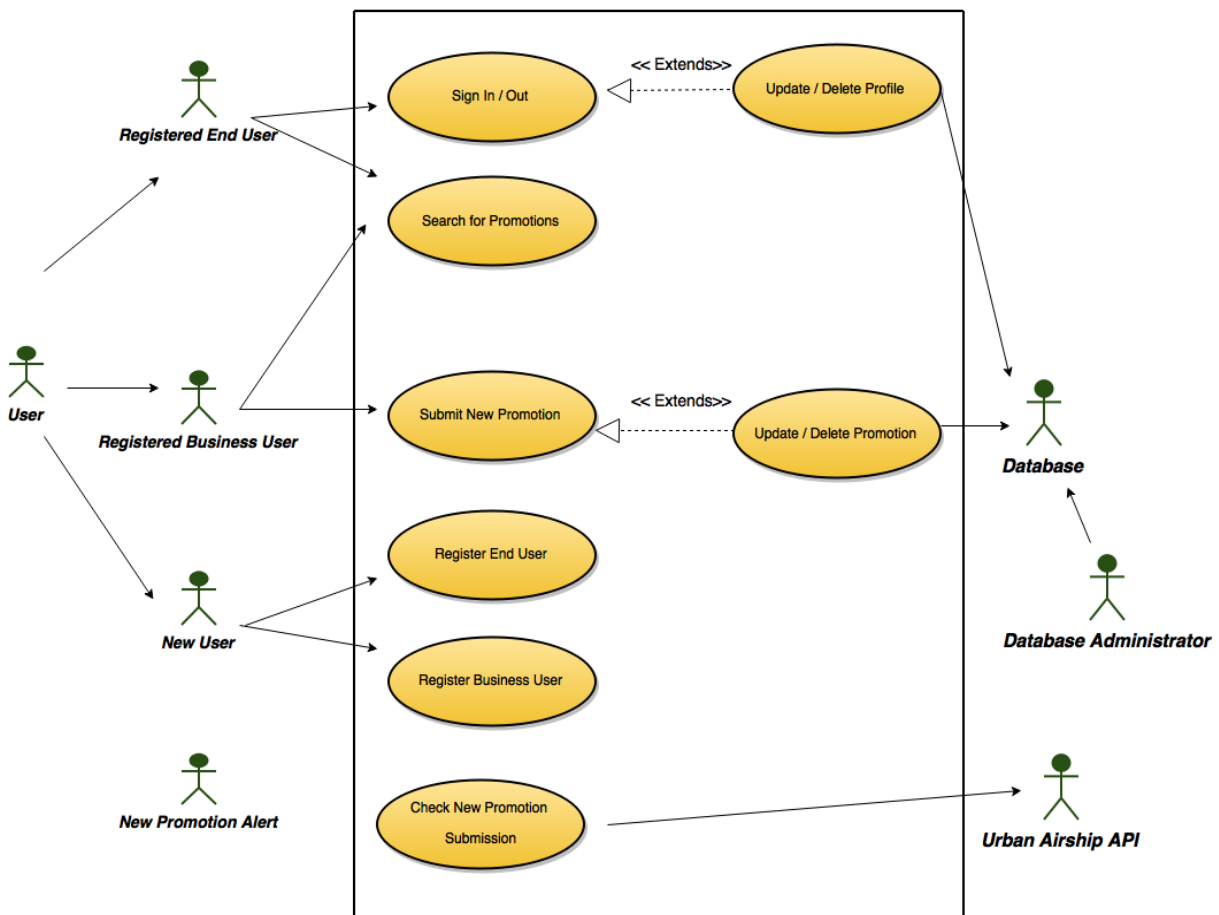


REQUIREMENTS SPECIFICATION

Requirements, both functional & non-functional will be detailed on the below section. Collectively this should build a solid picture of the overall requirements that will determine the applications functionality and performance in production.

FUNCTIONAL REQUIREMENTS

The functional requirements in this section will highlight all of the fundamental functionality that we need to have operational in order for the application to run in the desired fashion and to interact with our users in the way we require. Firstly, there is a use case diagram for the complete system shown below this paragraph.



REQUIREMENT 1: BUSINESS USER REGSITRATION

- Description
This requirement is related to an unregistered business user who needs to register an account to allow them to advertise promotions on the web application
- Use Case
Below is the Use Case for the Business User Registration Process

Description / Scope	Allows and unregistered business user to register a new account
Pre-Condition	The user does not currently have an account registered with the application
Activation	When the user click the "Register" button on the main home page
Main Flow	<ul style="list-style-type: none">- Unregistered user clicks on the "Register" button.- User then completed the registration form entering all required details- Application verifies all required information has been provided- User is then added as a "Registered Users" and can now login and access features of the application such as submitting new promotions.
Alternative Flow	<ul style="list-style-type: none">- "Unregister" user does not supply all required details to complete the registration process- Application displays a friendly error message detailing what the error is.- User re-enters the required details.
Termination	<ul style="list-style-type: none">- Main Flow: User is successfully registered.- Alternative Flow: User registration fails.
Post Condition	<ul style="list-style-type: none">- Main Flow: Business User can now login and submit new promotions.- Alternative Flow: User is returned to home page.

REQUIREMENT 2: BUSINESS USER SIGN IN

- **Description**
This requirement is related to Business User who has previously completed the registration process and can login to the application to submit new or manage existing promotions that have been posted.
- **Use Case**
Below is the Use Case for the Business User Sign-In Process

Description / Scope	Allows registered users to sign-in to the application
Pre-Condition	The user has already completed the registration process
Activation	When the user clicks on the “Login” button
Main Flow	<ul style="list-style-type: none"> - Registered user clicks on the “Login Button” - User then enters their username and password - Applications validates the username and password against the details stored in the database - User can then login and access the application
Alternative Flow	<ul style="list-style-type: none"> - Registered user does not enter the correct username or password - Application will present an error message saying the “details entered were incorrect” - User reenters the login details - Access is granted to the application
Termination	<ul style="list-style-type: none"> - Main Flow: User is successfully authenticated - Alternative Flow: Login process fails
Post Condition	<ul style="list-style-type: none"> - Main Flow: Business users can login and submit and manage promotions. - User is returned back to the home page.

REQUIREMENT 3: EDIT BUSINESS USER DETAILS

- **Description**
This requirement is related to a business user who has previously resisted and logged into the application and they need to edit their previously created profile.
- **Use Case**
Below is the use case for the “Edit Business User Details”

Description / Scope	Allows business users to edit their previously created profile.
Pre-Condition	The user has successfully logged into the application.
Activation	When the user clicks on the “Edit Profile” option
Main Flow	<ul style="list-style-type: none">- User clicks on the “Edit Profile” button”.- User updated profile details ensuring the details being entered match the data type requirements for validation.- User clicks “Save Details”.- User Profile is saved to the database.
Alternative Flow	<ul style="list-style-type: none">- User does not enter the correct data type to update the profile.- Applications will present an error message asking to re-enter the data.
Termination	<ul style="list-style-type: none">- Main Flow: User profile is updated.- Alternative Flow: User profile is not updated
Post Condition	<ul style="list-style-type: none">- Main Flow: User profile displays updated user details.- Alternative Flow: User profile remains as previously saved.

REQUIREMENT 4: DELETE BUSINESS USER

- **Description**
This requirement is related to a business user who wishes to delete their account from the application.
- **Use Case**
Below is the Use Case for the Delete Business User Process

Description / Scope	Allows a user to delete their account from the application
Pre-Condition	User has previously registered an account with the application
Activation	When the user clicks on the "Delete Account" option from the "Profile" section of the application.
Main Flow	<ul style="list-style-type: none">- User click on the "Delete Account" option- The application presents a message asking the user to confirm the request.- The user clicks "ok" to confirm the request- The user account is deleted from the application database
Alternative Flow	<ul style="list-style-type: none">- The user does not click "ok" to confirm they wish to delete their account- The deletion process fails.
Termination	<ul style="list-style-type: none">- Main Flow: User account is deleted.- Alternative Flow: Deletion process fails.
Post Condition	<ul style="list-style-type: none">- Main Flow: The user can no longer login to the application.- Alternative Flow: The user is returned back to the home page

REQUIREMENT 5: REGISTER NEW PROMOTION

- Description
This requirement is related to an existing business user who want to submit a new promotion to be advertised on the application.
- Use Case
Below is the Use Case for the Register New Promotion Process

Description / Scope	Allows existing users to submit new promotions to the application.
Pre-Condition	The user has registered with the application
Activation	When the user clicks the “Submit New Promotion” button.
Main Flow	<ul style="list-style-type: none"> - User clicks on the “Submit New Promotion” button. - User enters a title for the new promotion”. - User chooses promotion type from the cascading menu. - User enters a prices for the promotion. - User enters an end date for the promotion. - User clicks submit new promotion.
Alternative Flow	<ul style="list-style-type: none"> - User chooses an incorrect promotion type - User click submit.
Termination	<ul style="list-style-type: none"> - Main Flow: New promotion is successfully submitted under the correct category. - Alternative Flow: New promotion is not submitted under the correct category.
Post Condition	<ul style="list-style-type: none"> - Main Flow: End users can search for the promotion that has been advertised. - Alternative Flow: Users cannot search for the new promotion as it has not been advertised correctly.

REQUIREMENT 6: EDIT EXISTING PROMOTION

- Description
This requirement is related to editing or deleting a previously submitted promotion
- Use Case
Below is the Use Case for the “Edit Existing Promotion” requirement

Description / Scope	Allows business users to edit or delete a previously submitted promotion.
Pre-Condition	A promotion has already been submitted by a registered business user.
Activation	The user clicks on the “Modify Promotion” button.
Main Flow	<ul style="list-style-type: none">- User clicks on the “edit promotion” button- The application will display the previously entered details- The user the modifies the existing details ensuring they format and data type matches the requirements.- User clicks “ok” to submit the new changes.- Application displays a confirmation message asking the user to confirm they want to save the changes.- New details are saved to the database.
Alternative Flow	<ul style="list-style-type: none">- Updated details do not meet the required data format as required by the application.- On clicking save the application will display an error asking the user to review the updated data.
Termination	<ul style="list-style-type: none">- Main Flow: User clicks on submit.- Alternative Flow: Application rejects the new changes.
Post Condition	<ul style="list-style-type: none">- Main Flow: Promotion details are updated or deleted.- Alternative Flow: User is returned to the main home screen.

REQUIREMENT 7: SEARCH FOR PROMOTIONS

- Description
The requirement is related to an end user performing a search for promotions that have been previously submitted by business users.
- Use Case
Below is the Use Case for the “Search for Promotions” requirement.

Description / Scope	User searching for promotions that have been previously submitted by a business user
Pre-Condition	A promotion has been submitted by a business user.
Activation	A user launches the application and opens the search option
Main Flow	<ul style="list-style-type: none"> - User opens the application - User navigates to “Search for Promotions” - User chooses search criteria from cascading menu - User presses the search button - Application returns list of matching promotions - User chooses individual promotion they want to view.
Alternative Flow	<ul style="list-style-type: none"> - User opens the application - User navigates to “Search for Promotions” - User does not correctly choose search options from cascading search menu - User presses the search button - The application will return an error message notifying the user they must choose all required options.
Termination	<ul style="list-style-type: none"> - Main Flow: User clicks on selected item from the search results to view information about promotion. - Alternative Flow: User does not choose all required fields from the search menu
Post Condition	<ul style="list-style-type: none"> - Main Flow: The user is presented with a view containing all information about the selected promotions including price, location, availability and contact details - Alternative Flow: User is returned to the search menu.

REQUIREMENT 8: RETREIVE NAVIGATION INSTRUCTIONS

- Description
This requirement is related to an end user retrieving directions from the Google Maps API feature on how to get to the location of the business advertising the product and service.
- Use Case
Below is the Use Case for the Retrieve Navigation Instructions Requirement.

Description / Scope	User gets directions to the location of the business advertising a particular promotion when accessing the applications on a mobile device.
Pre-Condition	User has searched for available promotions and retrieved a list of search results.
Activation	User clicks on “Get Directions” from the “Promotion View” section.
Main Flow	<ul style="list-style-type: none"> - The user logs into to the application and chooses search criteria from the cascading drop down menu. - The user is then presented with a list of search results. - The user clicks on an individual promotion to view more information about it. - The user then clicks on the “Get Directions” to be presented with directions via Google Maps.
Alternative Flow	<ul style="list-style-type: none"> - The user logs into to the application and chooses search criteria from the cascading drop down menu. - The user is then presented with a list of search results. - The user does not click on an individual promotion to view more information about it.
Termination	<ul style="list-style-type: none"> - Main Flow: The user clicks on “Get Directions” - Alternative Flow: “The user does not choose to view a result from the returned search list.
Post Condition	<ul style="list-style-type: none"> - Main Flow: The user is presented with directions to the location of the business. - Alternative Flow: The user gets returned to the main search screen.

REQUIREMENT 9: SIGN OUT OF APPLICATION

- **Description**
This requirement is related to a business user signing out of the applications after they have completed their tasks.
- **Use Case**
Below is the Use Case for the “Sign-Out of application functionality.

Description / Scope	The business user has completed their session in the application and they then sign-out.
Pre-Condition	The business user has previously registered an account and signed in as an authenticated user
Activation	The business user clicks on the “Sign-Out” button.
Main Flow	<ul style="list-style-type: none">- The user logs in to the application and performs some tasks.- The user then clicks on the “Sign-Out” button.- The applications ends this session and logs out the user.
Alternative Flow	<ul style="list-style-type: none">- The user does not click on the “Sign- Out” button.- The user remains logged in and authenticated in the application.
Termination	<ul style="list-style-type: none">- Main Flow: The user clicks on the “Sign Out” button”- Alternative Flow: The user does not click on the “Sign-Out” button.
Post Condition	<ul style="list-style-type: none">- Main Flow: The user is returned to the main “Sign-In” page- Alternative Flow: The user stays in the main home page as an authenticated user.

END USER

ClevaDeals.com: Market Research Interview End User

Participant Name: *Nichola Danagh* Age: *32* Location: *Neven - Moeth*

Have you ever made an online purchase?	Y/N: <i>Yes</i>
Yes: What have you purchased online? No: What not?	<i>Food from Tesco Products like Puffs from Amazon</i>
If you could find the same products / services local would you shop prefer to shop locally?	<i>Yes</i>
Yes: Why? No: Why Not?	<i>It is easier to buy the products Easier to return Support the local economy</i>
Do you ever search online for local retailers or do you simply browse local stores?	<i>Normally search online first</i>
Yes: Why? No: Why Not?	<i>Easier + save money</i>
Generally how do you find the experience of shopping locally?	<i>Stressful as I mostly have to bring the children with me :-)</i>
How do you shop online? Direct with retailer's websites? eBay / Amazon? Deal Site?	<i>Both retailers websites + amazon + e-bay</i>
Do you ever "Impulse Buy" online? If so: what factors would make you make an impulse purchase?	<i>Yes - if it is a cheap or Reduced offer</i>
What device do you use to shop online? PC / Laptop / Phone / Tablet / Other?	<i>PC mostly - sometimes the phone</i>
How do you find the experience of online shopping? Why?	<i>it can sometimes be confusing - I also do be wary of using my credit card online</i>
How long are you willing to wait for delivery or an online purchase?	<i>Depends</i>
How much delivery charge are you willing to pay for delivery of online items?	<i>Depends also on size of product</i>

ClevaDeals.com: Market Research Interview Business Users

Participant Name: *David Gault Volating* No of Employees *4* Location: *Naxon, Meath*

Do you have an online presence? Yes: what type: Facebook / own website / etc..	<i>Yes Facebook</i>
Who manages the website site / app for you?	<i>We manage it ourselves</i>
Have you ever sold goods via an online "Deals" website such as "GroupOn"? Yes: Why: No: Why Not:	<i>No.</i>
Has your online advertising been successful? Yes: Why? No: Why Not?	<i>Never Used</i>
Do you think you need more local visibility to attract local customers? Yes: Why: No: Why Not:	<i>Yes - most of our custom comes from local users</i>
Do you ever use special promotions / sales to try and attract more customers? If so, have they been successful. Yes: Why: No: Why Not:	<i>Yes - we offer a lot of different offers They have been moderately successful - Not all of our customers use facebook to see the offers</i>
How do you promote these offers / sales?	<i>facebook / local advertising</i>
Would you be willing to use an online app to advertise these promotions aimed specially at local customers?	<i>Yes</i>
Would you be willing to pay for this type of service.	<i>Yes - as long as it is not too expensive</i>
Would you prefer to pay a monthly subscription or a single yearly fee?	<i>Yearly</i>

MONTHLY JOURNAL

This series of journals will record my progress over the next 6 or so months. In this journal I will record every aspect of my project such as how I chose my project subject, challenges I face as the project progresses and details of my project sponsor meetings.

I will make entries to this journal on a monthly basis however and I feel at the end I will enjoy looking back over what I hope is a project I have completed fully and one that I feel I have made my very best effort on.

SEPTEMBER

This is my journal entry for September. This will be a short submission as honestly my project has not really started yet. I am somewhat disappointed by that as I was planning on hitting the ground running with this. I put some thought into this over the summer and even had the project proposal completed before we returned for the college term.

However, my project proposal presentation did not go as planned and at this point I am considering alternatives. My proposal is to create a web application in ASP.net that will allow local retailers to advertise their projects and provide functionality for users to search for these projects.

The feedback that I received was positive as far as the idea goes but the technical aspects of the project are a bit light as far as a 4th year final project goes. The panel has provided some feedback and made some recommendations on what I can add to application, specifically they recommended I add some Geolocation services to the app where if a user is in the same area as a retailer that has advertised on the app they would receive a notification to their phone. This would add a dimension to the app where I am not relying on users opening the app and searching but instead they would have the product pushed to them.

While I feel it is a very good suggestion, and honestly it is an idea I did think of earlier, and I can see where it would add a great deal of functionality and value to the idea I have some reservations about how it would work. My concerns are the app is a web based app and I am not sure if a web based app can access the notifications services on mobile devices to enable the push alerts. I have limited knowledge around this area so at the moment I am looking at the possibilities.

If I choose to scrap this idea my alternative is to create an application that will allow me to visualise information from a database. This is an idea I have taken from the company I work for. In work we have a helpdesk application that records a huge amount of information from day to day support tickets to an asset inventory. The base reports available from this are quite basic and we are currently looking at solutions to provide more detailed information. I am going to check if I would be allowed to use this database for the project

Over the next few days I aim to investigate more about the Geolocation possibilities of the web app and decide what the next steps are.

OCTOBER

This is my journal entry for October. For the past month I had been mulling over my final project choice, not the best time I know!! However, I have now investigated all the options I have regarding adding a notification service to a mobile web application.

There are a number of applications which provide the functionality to send notifications to web applications on mobile devices. Google for example have a Web API that can be used, there are also a number of third party apps such as "Urban Airship". Depending on what features I want to use the cost can range from free to \$89 per month.

So at this point I am very happy with my project choice and I feel I can create an application that provides a very useful service. I have some UI designs in mind, I know the project brief stated that this is a Software Development project so not to spend too much time on the interface but the reality is this is a Web App so in my view the interface is as important as the application code.

I also had my first supervisor meeting this month, I have been assigned Dr Ralf Bierig. I had not met Ralf before but I must say I was very happy with how our first discussion went. Ralf had some great initial feedback and pointed out a few areas where I should be concentrating on. I feel Ralf will be a great support as this project progresses.

I have also started my Project Requirements Specifications document this month. I forgot how comprehensive this document is and it is taking much more time than I thought. On Ralf's recommendation I have included a short questionnaire that I have at this point put to 20 people, some of these people were from work but 12 of these were people who I have randomly stopped in the street. This was an awkward experience as initially I think people assume I am trying to sell them something and they try to walk away but once I got their attention most were surprisingly co-operative.

I tried to use the questionnaire to discover how and why people prefer to shop, either online or via local retailers. Based on their response I wanted to see how did they search for local products and if they have ever used an online shopping app what was their experience.

Following on from my "Questionnaire Process" there is one aspect of this project that I need to refine and that is how I initially describe what exactly the application will do. Up to this point I have been describing it as a "type of GroupOn" app. This leads people to form an opinion as to what it should do and after describing it further some people are slightly confused by the idea. The reason for this is my project is inherently different to apps like GroupOn. The real advantage to this app is felt by local businesses and local shoppers and as such this is what I need to stress to people when they are first hearing about it.

My next steps are to start building the prototype, for this I would like to have the search functionality completed where I can show how the cascading search options will work.

NOVEMBER

This is my journal entry for November. This month has by far been the busiest month I have faced in college. There were multiple submissions, CAs and other project work to be completed. These all also coincided with a major project I have been involved with in work nearing completion which meant I have had to spend a good few evenings and Saturdays in work too. This month my family time has been affected in a big way so December will have to be spent making this up to my wife and two kids.

At this stage I have now submitted my mid-point documentation and have also completed my presentation which I will give to Ralf and Eamon this Tuesday 13th December. My aim for this is to really show the application functionality, why I think there is a real-world use for this project and also present the results of the market research I have carried out to prove this.

In developing the prototype, itself I created an ASP.net application that has a set workflow. This is to show how the application will be used. There is no database integration at present as I have not even started to create the backend. My only concern is that I will not be able to really show how the notification service will work and this is an integral part of the project. I can however show how the notifications will appear on the user's devices so this may be sufficient.

DECEMBER

This is my journal entry for December. This month followed on from November in terms of workload. We have had so much to deliver in terms to CAs and Project submissions however they have all been completed and I am very happy with how they finished up.

This past month was very important in regards to the Project module. There have been a number of deliverables due. The mid-point project report has been completed, I have also completed the project prototype and presented this to Eamon and Padraig de Burca.

Initial feedback from both lecturers was promising and both raised some very important areas that I need to concentrate on such as the mobile aspects of the projects. Padraig also highlighted the necessity to ensure the actual presentation is formatted correctly and checked for grammar, this is not something I would normally need to be corrected on but in fairness there were some mistakes with how I prepared to PowerPoint presentation.

For the prototype, I have created a 7 page MVC application which is hosted in Azure. The prototype demonstrated how the user will navigate the application from initial login to searching for and displaying products. It also demonstrated how business users will use the application to advertise their products.

This prototype provides a good base for the application for me to build on. The interface and design used in the prototype will be maintained and I have also created a GitHub account to manage the code.

The next stage in this project is to complete the database development and then move on to building the main functionality into the application.

JANUARY

This is my journal entry for January. This month we have returned to college with the news that the exams are now going to be April instead of May meaning we will be finished in 11 weeks. This is a great help to students as it means we will have a month free to work purely on the project from April to May.

At this stage I have created the database and the ASP.net front end. By the end of the month I want to have the cascading search option created as this is a complicated part of the application. The challenges I face at this point are more around time management and scheduling face to face meetings with Ralf. I am due to meet him next Monday morning @ 7.45am which means I will need to drive to college before work, meet Ralf and then drive back to work for 9am. Hopefully we will be able to come to a more suitable agreement as far as the future meeting times are concerned.

I also plan on starting to use GIT Hub from this point plus some time/project management applications such as Kanban to manage this project more easily

FEBRUARY

This is my journal entry for February. This whole process is now becoming very difficult to manage and I have to juggle college, work and my family commitments all at the same time. The other modules in semester two each have a number of CAs and exams that all have to be completed so the workload is immense.

To try and make time for my project I have started to stay late in work on Monday and Friday evenings meaning I am now committing Mon, Tues, Thurs, Fri and Saturdays to try and get through the workload. I am also working on my project during lunchtimes and even at times during the day when work gets quiet. This is not ideal but it's the only time I can work on the project as if I try at home in the evenings I have two kids interrupting me every 10 minutes!! One problem I have is GitHub is blocked from our work network so for the moment I have been using Team Foundation Server as version control and am transferring the code to my home PC every few days.

As far as the project goes I have completed the registration and login processes. Ralf has stated that this is not a feature he thinks I should spend much time on however they are vital to my application being a fully complete and functioning entity as the logins will create different levels of access – specifically admin, business and end-user roles.

I have also started some work on other features, specifically the Google Maps API and the add product and service options.

I will also try to arrange another meeting with Ralf over the next few weeks to keep him up to speed with the process.

MARCH

This is my journal entry for March. Following on from last month's journal this entry will follow a similar theme.... the three biggest challenges I am facing are time, time and time!!

I find the other module deliverables need to be prioritised at this point as they all are approaching submission dates however I am making some time for the project module too.

On one of my first meetings with Ralf he pointed out to me that I will need to look at some features of the application and decide if they are needed, as in do they add value to the application. Ralf suggested I get the core functionality completed first and IF there is time left at the end then I should then look at adding in the "nice to have's". This, almost prophetic, is exactly the situation I have found myself in.

I have looked at the time remaining and I have looked at the amount of time that each feature will require, normally I would try to continue with the original plan but with Ralf's words ringing in my ear I have decided some features will be removed.

The first feature to be removed is the Role Based Authorisation service. Although I have completed a good bit of work on every other feature as it gets developed would need to take the roles into account which would add more time. So for the moment I have implemented a basic two-level login system. This is needed as I will require users to login for them to subscribe to email alerts.

The second feature I am removing is the "Advanced Search" option. This is where users, who after selecting the top level search choice such as "Food" could then further refine the search by selecting a type of food such as "Breakfast" or "Restaurant". The advanced search can be removed as performing a search at a higher level such as "Food" can demonstrate the location based search just as well as if I was refining the search.

All other features will be left as is and as such I have started looking at what is the best way to implement each function. One issue with this application is that it is being developed using ASP.net. As such, the application will be accessed using the browser and will not be able to access the native features of the mobile device. This means that the GPS function cannot be used to obtain the location.

APRIL

This has been the month that has really kick started the development of the application. I have been complete the base functionality such as the ability to register a new account, add deals and perform basic searches. I have had to take a total of 8 days' annual leave from work to facilitate this work but it has been hugely beneficial.

I have also been testing the additional functionality that will make this application unique compared to a standard website what just provides login and CRUD functionality. I have been able to add Google Maps API functionality which allows for the navigation instructions to be returned to the user. I have also investigated the JavaScript functions that will allow the user location to be obtained when they first launch the application.

Although the application development is moving along I am concerned about the final report as this is quite a large document. I also need to start considering the usability and functionality testing which will all require a considerable amount of time. This, along with the last few CAs and terminal exams that we have are adding a lot of pressure to the timescales.

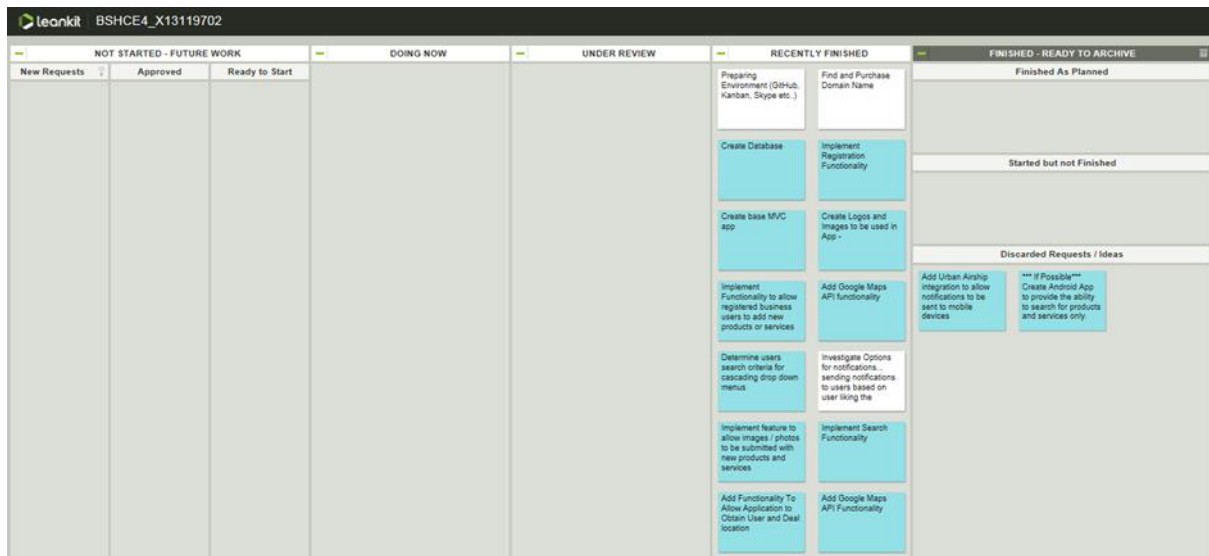
ADDITIONAL RESOURCES USED

LEANKIT KANBAN

Leankit Kanban is an online tool that is used to create a dashboard board of a project. The project can then be broken into individual tasks which can be placed at different points of the dashboard to give a graphical representation of the overall progress of the project.

This is a very useful tool as it simplifies the process of identifying the individual tasks associated with the overall project. The Kanban chart for the ClevaDeals project can be accessed using the link below:

<https://clevadeals.leankit.com/>



GITHUB

GitHub is one of the leading solutions used by developers to implement version control for applications. All of the source code for the ClevaDeals application is available on the public GIT repository as the link below:

<https://github.com/ronanncirl/FinalProject>

ADOBE ILLUSTRATOR

The main logo as displayed on each page of the application has been developed using Adobe Illustrator.



GODADDY

GoDaddy are an Internet Domain Registrar and Web Hosting Company. For this project the domain www.clevadeals.com has been purchased to provide a friendlier URL to access the website as the default address from azure is <http://clevacodefirst.azurewebsites.net/>

REFERENCES

The Irish Times. (2017). *'Shockingly poor' retail sales dampen hopes of recovery on high street*. [online] Available at: <http://www.irishtimes.com/business/retail-and-services/shockingly-poor-retail-sales-dampen-hopes-of-recovery-on-high-street-1.1373836> [Accessed 9 May 2017].

Anon, (2017). How do colors affect purchases? *Infographic*. [online] Available at: <https://blog.kissmetrics.com/color-psychology/> [Accessed 9 May 2017].

The Irish Times. (2017). *Irish consumers to double online spend by 2021, says study*. [online] Available at: <http://www.irishtimes.com/business/technology/irish-consumers-to-double-online-spend-by-2021-says-study-1.2799411> [Accessed 9 May 2017].

KYROS. (2017). *The 3 New Realities for Local Retail | KYROS*. [online] Available at: <http://gokyros.com/3-new-realities-for-local-retail/> [Accessed 9 May 2017].

Azure.microsoft.com. (2017). *SLA for App Service*. [online] Available at: https://azure.microsoft.com/en-us/support/legal/sla/app-service/v1_4/ [Accessed 9 May 2017].

Nngroup.com. (2017). *Usability Engineering: Book by Jakob Nielsen*. [online] Available at: <https://www.nngroup.com/books/usability-engineering/> [Accessed 9 May 2017].

Pidoco.com. (2017). *Pidoco - The Rapid Prototyping Tool*. [online] Available at: <https://pidoco.com/en/help/ux/5-second-test> [Accessed 9 May 2017].

Nngroup.com. (2017). *Website Response Times*. [online] Available at: <http://www.nngroup.com/articles/website-response-times/> [Accessed 9 May 2017].