

Course: Interactive Multimedia & Design

Year: 4

Name: Sam Crowe

Student No: B00535676

Peer Group Number: 2

Mentor: Peter Nicholl

Assignment for Major Project:

I'm Hungry

Acknowledgements

There was a series of people and communities that helped during the development of this project. For all that helped I would like to thank them for their input, patience and inspiration. Firstly I would like to thank group peer mentor, Peter Nicholl. Peter was there from the start of the project and always helped steer me in the right direction. I would also like to thank Peter for everything that was provided during his COM601 class the information gained from that class had a major factor in the final outcome of the application.

I would like to thank the expressionengine.stackexchange community, this group of Expression Engine developers helped on numerous occasions when I was lost for solutions. I would like to thank Tim Whiteside for his professional opinion and support of the product. I would like to thank Owen Kavanagh an EE web developer that helped problem solve many issues. I would like to thank Giuseppe Trombino for all his help during the programming help classes, Johnny Wallace for his kinds words about my progress and I would also like to thank Alison Duff for her research and industrial connections.

Finally I would like to thank all the developers that created the plugins and add-ons that helped piece this application together.

Contents

CHAPTER 1	6
Concept definition and testing.....	6
1. 1 – Idea generation.....	6
1. 2 – Requirements specification	7
1. 3 – Paper Prototyping.....	8
Feasibility Testing	9
Methodology Selection	14
CHAPTER 2	15
Design.....	15
UX design evolution	16
Inspiration & design	16
Initial ideas and wireframes	17
Final Designs	18
System design.....	22
Logic design	23
Expression Engine design	25
Data design.....	28
chapt. 3.....	29
Implementation.....	29
Creating multiple environments	29
Building Expression Engine Channels.....	30
Building Expression Engine Custom fields.....	31
Building Expression Engine Templates	33
Building Expression Engine Upload destinations.....	33
Channel:entries loops	34
Favourites.....	35
Comment:entries loops	38
Registration and Login	40

Flip Animation	42
Geo location	44
Way points, bells & whistles.....	45
Chapt 4.....	49
Testing	49
Functional and system testing.....	48
Verification & functional tests.....	49
Performance testing.....	50
Usability testing	50
Validation & user reflection.....	51
Chapt 5.	55
Evaluation.....	55
Chapt 6.	61
Conclusion	61

Introduction

I'm Hungry is an online directory that is populated with simulated take-away restaurants in Belfast. This project aims to provide users with useful details about each business in the database. Users will have various methods of search, including geolocation, voice input and simple search via a search box.

The objective of this project is to offer users a method of finding take-away restaurants located close to their geolocation. The application will allow users to filter menus and observe the items that are within their price range, therefore creating a cycle of usage that will objectify the ideology of the application.

A further objective is to increase marketability through strong branding, this factor will aid users with the connection between the brand and the concept. Aesthetically branding and design can lead to an enriched user experience. This objective will help users access information quickly without confusion.

To ensure the aims and objectives of this project have been met, a methodology was chosen to manage the each phase of the project. Detailed research was required to truly understand the needs of the users. This strategy applied a respective project plan that will meet the needs of the user and enforce security of the application.

The documentation of this report will cover areas including, project management, design and development rationale. The rationale described will underline the strategy for bringing this project from start to finish. This report will also discuss the integration of the software, describing how each of the algorithms for code execution were concluded and implemented.

CHAPTER 1

Concept definition and testing

The concept definition and testing of this project helped define the direction the project would take and provided the requirements to build it. To gain a broader perspective ideas were fleshed out with a group of peers, this session lead to a wider range of possibilities that had not been previously considered. Receiving feedback aided the agreed deliverables as well as expanding fresh and proposed requirements. The segments covered in this section will include the idea generation, requirement specification, paper prototyping, feasibility testing and selection of methodology.

1.1 - Idea generation

The basic concept for this project had originally been considered a few months prior to any initial deliverables were submitted. To gain a different perspective about the concept the idea was presented at an idea generation workshop.

The purpose of the workshop was to create an environment that would allow each developer to generate ideas amongst a group of peers. This workshop delivered a positive response with different opinions and insights. This helped provide clarity on the proposed objectives. During the workshop a mind map was created alongside a group of peers that linked words relative to the product.

Developing a mind map helped understand how users related to the concept, food was primarily the association with this application (see appendix 1.1). The decision that needed to be made after this workshop was whether or not to use food as a factor in the design. The group felt that a decision would need to be made on whether to promote the brand relating to a particular food. The issue that could occur with this method of branding is that users could associate it the product with a particular type of food. eg. If a pizza slice was used as part of the

logo or main imagery would users think of the application solely as a pizza orientated product.

1.2 - Requirements specification

To deliver the requirements of the project it was decided that the Volere Requirements Specification Template would be used to approach a list of targets. This template provides a method of structure by grouping the requirements into nested deliverables. The requirements include the description, type, priority and dependancies of each task at hand.

Each requirement of the project has two specific elements that are associated with it, these elements are the requirement type and priority level of the task. In appendix 1.2 you can see the scale of requirements types and priority levels. The scale of these two factors help to weigh out the needs and concerns of the project. This process identified requirements that would were high level in regards to the success of the project. Requirements were assessed based on the priority level and the dependancies of other requirements. A full list of requirements can be found in **appendix 1.3**.

Many requirements caused issues during the implementation phase of the project. Requirement #019 declared that relative facts about each business should be entered into the directory. In the later stages of the project their was no time to market the product to commercial users, this left a high level requirement needing reassessment.

It was decided that the project needed to branch off to find a new source of information. This matter resulted in researching the contents of take-away menus and generating simulated accounts to show that the application worked. This was a time consuming factor as logos needed to be designed for some businesses. This measure assisted the requirements as no data protection was infringed.

Previously it was considered important to ensure the usability and maintenance of the application was up to date, allowing correct information to represent the application and business user respectively. To ensure this requirement was met each business user would be allocated their own unique login for the CMS. This requirement raised a security issue as it was felt that the application would be more of risk of hacking and scamming if a secure path was made public. It was decided that business users would have to enquire in order to have their business added to the directory.

To gain a more experienced perspective on how to deal with this scenario it was concluded that the best solution would be to enquire to Hungry House and Just Eat as a business user. The idea was to propose to these two outfits that a new business was looking to sign up to their service and find out how a menu would be added to their database. The impression received from this tactic was that Just Eat have a data entry team that enters each menu into the database for the user as part of their service (see **appendix 1.4**).

1.3 - Paper Prototyping

The paper prototyping process was useful when modeling a user-interface in initial design phase of the project. This method included mind-mapping, 6-ups and 1-up prototyping. These methods helped illustrate the requirements of early concepts related to the pages sketched during this phase of research and design (see **appendix 1.5**).

To gain an understanding of layout and functionality an article was researched about the mistakes that are often made when developing a web based product. The article informed how to present information to gain the best results for usability from a user perspective (see **appendix 1.6**).

In usability studies using our eye-tracking equipment, we commonly find that the far right of the page is one of the last places people look (along with the very bottom of the page). Even more strongly, users don't expect to have to scroll

horizontally to see anything important beyond the right-hand border of the window.

Thomas S. Tullis . 2005. Web-based Presentation of Information: The Top Ten Mistakes and Why They Are Mistakes. [ONLINE] Available at: <http://citeserx.ist.psu.edu/view/doc/download?doi=10.1.1.107.9769&rep=rep1&type=pdf>. [Accessed 09 April 14].

Feasibility Testing

The feasibility testing aided the consideration of risks that this project may come up against in future proceedings. The evaluated risks assess scenarios from the development and planning phase, leading to the consideration of how these risks will impact the project. The likelihood of these risks occurring require assessment, as a solution for mitigating the challenge will ease the implementation phase of the project.

“Defined as a controlled process identifying problems and opportunities, determining objectives, describing situations, defining successful outcomes and benefits associated with several alternatives for solving a problem.”

PWC. 2013. PWC Feasibility Report.[ONLINE] Available at: <http://tinyurl.com/pnx76at> [Accessed 20 November 13].

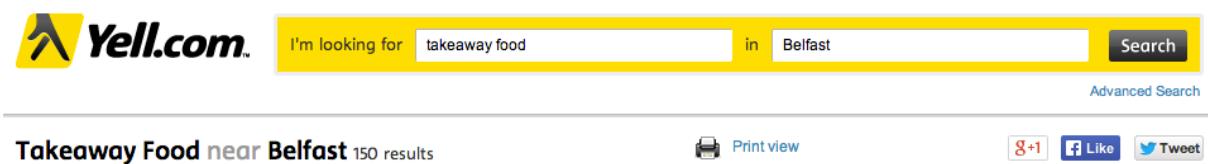
Project Risk One

When delivering this product the issue of finding and relating to the target audience was discussed, without a market the product would become ineffective. A solution was required to reach and relate to the target audience. In order to understand what information would be useful to the user group, a survey would be taken to see what information would be most useful about each business. The initial target audience for this application was students, with the prospect of growing popularity, the application could reach a broader audience.

The survey results projected that price and location were the main concern of users when they were choosing a restaurant and all most all users found that it would be helpful to search by location ([appendix 1.7](#)).

Project Risk Two

The application will need to have businesses in the majority of post codes in Belfast, this will allow users across the city to view relative nearby results. The concern with this risk is that by covering take-away restaurants in all postcodes a method for inputting data would need to be resolved. Each business will have detailed documentation to support their entry. This feature could be very time consuming depending on the number of take away restaurants chosen and the depth of their menu. (*figure 1.1*)



The initial concept was to webscrape menus from various online sources such as Just-Eat. This method of extraction would allow for a quick technique to gather information such as prices, descriptions and details about take away businesses.

If the information was web scraped it would allow for a simple CSV import to the database, quickly populating the application with businesses across the city. To make this plan successful an excel sheet would need to be set up and each column of information would be headed as the relative table name in the database. After researching further it was realised that web scraping content was against Just Eat's policy. (*figure 1.2*)

7.LICENCE

7.1.Terms of permitted use: You are permitted to use the Website and print and download extracts from the Website for your own personal non-commercial use on the following basis:
7.1.1.You must not misuse the Website (including by hacking or "[scraping](#)").

To enable easy management of the application, a CMS with a user-friendly control panel will need to be selected. It was felt that the best way to input data would be to give an administrator access to the control panel. Developing with a

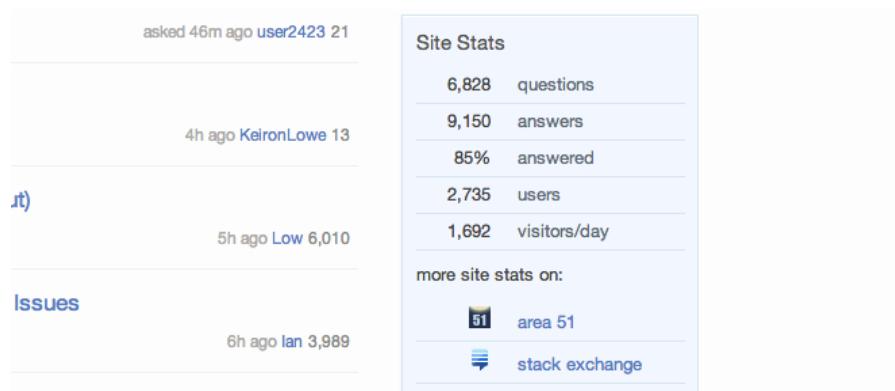
CMS would be useful for the longevity and future of the project. Perceiving that businesses will want to sign up, additional content will need to be managed. This risk helped emphasis and regulate a host of requirements described in **appendix 1.3**.

Project Risk Three

Building a project with a new software compilation requires risk assessment. A php framework and content management system will need to be integrated to perform the heavy lifting of backend development. Research was undertaken to fully understand the strengths and weaknesses of the various platforms. The software researched included Laravel, Code Igniter and Phalcon.

It was decided that Code Igniter would be used for the php framework and Expression Engine for the CMS. EllisLab have built the php framework Code Igniter and they use it to power Expression Engine. This decision was endorsed due to the strength and reliability of the online community that supports users of EllisLab software.

The risk factor that comes with using this framework is that research needs to be conducted into building a sandbox of tools and API's. The sandbox will include different plugins that are used to access information from the database in different ways. Add-ons(Expression Engine API's) are essential in regards to how Expression Engine will functionally work. The results from the research concluded that the Expression Engine stackexchange has a vast community that has an answer rate of 85%. This figure strengthens the rationale for choosing an EllisLab framework. (*figure 1.3*)



ExpressionEngine is the ultimate in flexibility, designed to be a platform for developers to easily extend to enable rich functionality. Like an operating system, it ships with a set of “apps” to cover the basic necessities right out of the box. We call those apps Add-Ons.

Add-Ons — ExpressionEngine 2.8.1 documentation. 2014. Add-Ons — ExpressionEngine 2.8.1 documentation. [ONLINE] Available at:<http://ellislab.com/expressionengine/user-guide/general/add-ons.html>. [Accessed 06 April 2014].

Project Risk Four

Initially this project aimed to give business users access to the content management system. This concept would allow users to sign up their business through an outlet in the application. When a business user has their own username and password they would be able to update the menu and business details themselves.

The risk involved with giving countless business users access to the control panel of the CMS would be that the admin side of the application could be at risk of foul play. Generally the path to the administrator login is limited to a single admin user as this reduces the risk of hacking. It was felt that the application could become vulnerable by giving this login address out to various parties.

The best method to keep this application secure would be to have users email the admin through an outlet in the application. This email would be an enquiry about having their business added to the directory. To make this successful a clearly designated area will inform users how to do so (see **appendix 2.9**).

The decision to enter businesses manually to the directory has changed the scope of the project slightly. It has been decided that the removal of a functional requirement would benefit the projects security.

Project Risk Five

Expression Engine development relies heavily on third-party plugins to aid the integration of the CMS, providing necessary functionality. The majority of third party plugins are free to download. Although there are number of highly regarded add-ons that may require a commercial license to produce this product. A commercial license would add cost to the project and rationale would need to be considered of whether adding cost to development would be worthwhile. Commercial interest in this produce would support that decision.

Referring to appendix 1.3 requirement #20. It was decided that using a stakeholder with expertise in marketing would help to determine the risk and reward of this decision. Alison Duff, International Marketing Executive for The Creativity Hub would apply her insight into whether the product was strong enough to justify commercial plugins.

The stakeholder was able to introduce the application concept to industry related companies. This form of feedback provided invaluable observation in regards to the products marketable value. Alison introduced the project to AllState and Bluegator Creative. The technical director of Bluegator Creative declared that if the project was built with Expression Engine and used commercial add-ons that they would be interested in developing the product further. Between the two stakeholders rationale was conceived.

The screenshot shows an email conversation. The header includes the recipient's name (Tim Whiteside) and email (tim@bluegatorcreative.com). The message body starts with "Hi Sam". Below this, a block of text discusses the benefits of using commercial add-ons for Expression Engine websites, mentioning robust security and professional support. The message continues with a statement about commercial plug-ins being easier to train users and having more documentation. The sender expresses interest in the project and desire for updates. The message concludes with "Best Wishes" and the sender's name, Tim.

Tim Whiteside tim@bluegatorcreative.com

Hi Sam

When building websites with Expression Engine, I would recommend that you use commercial add-ons where necessary. The benefit of using commercial add-ons is the security and support that comes with them. Our experience with commercial versus non-commercial, is that commercial plug-ins tend to be more robust and the support provided is professional. The non-commercial plug-ins tend to be less secure and updates and support enquires tend to be very slow.

We have found that commercial plug-ins are easier to train users on and there is plenty of documentation online to support them.

I like the concept for your product and have a client that this would be of interest to in the US. Keep us up to date with the development of your project.

I would like to meet with you when it is finished.

Best Wishes
Tim

The information provided about using Expression Engine declared that EE was a more elegant solution than most content management systems. This system is more purpose built compared to Wordpress and it would be easier to add stuff onto later. When developing with Wordpress, developers build to Wordpress restrictions. EE allows for custom, flexible websites to be integrated without restrictions.

The alternative to using commercial plugins would be to build bespoke add-ons that replicate performance. Although it was decided that the commercial value could be devalued by attempting to replicate third-party plugins. The reason behind this was that bespoke plugins are less secure unless they have been developed by experienced EE developers. Building all the third party plugins that are required to develop this project would be as time consuming as building the planned project in itself with the resources that are available online. A benefit of using the commercial add-ons would be that there is a stakeholder willing to invest in taking the product to the next stage of development.

Methodology Selection

When the concept was created for this project development, testing and functional scripting began ahead of any recommended milestones. The decision to test various techniques helped determine if the project development was feasible before committing to it. The initial approach was agile, this methodology was useful at a basic level. This method helped raise awareness of concerns such as building user registration and integrating the framework.

The final methodology chosen was the waterfall method. This methodology was selected due to its sequential process, allowing the project to be managed in stages. This methodology is recommended for long projects and its downward rigid, spiral of phases will be helpful in regards to managing a project. This method has been tested against longevity (**appendix 1.8**).

To adhere with this model each process needs to be completed before another has started. This will be beneficial to this application as the development team

consists of one person. The attributes of this rigid model will help show deliverables and review the progress of each phase. The project will be much easier to manage if it is managed and developed in stages. The gantt chart in appendix 1.9 shows the various tasks that were projected for deliverables to ensure the deadline was met.

CHAPTER 2

Design

The design section of the project consists of the UX design evolution, system design, logic design and data design. These design phases include the aesthetic look and feel of the project as well as the functional design of the system.

UX design evolution

The team went through three stages of UX design. To attain the ideas and concepts for an agreed user friendly design, an approach that included sketches and illustrations were produced. As there were three stages of design, this ultimately permitted three rounds of snags to perfect the design and concept.

Inspiration & design

The search for inspiration and design was set off with the intent of finding a style that would uniquely represent the application. When researching this section of the project, exploration into branding, layout, colour scheme and design trends were researched to gain objectivity of the market.

When researching relative inspiration, design communities such as Dribbble, Pinterest and Behance were approached. Mood boards were created to show the useful inspiration gathered for brand and web design (**appendix 2.1**).

The competitors that were researched included ‘hungryhouse.co.uk, just-eat.co.uk and niftynosh.co.uk’. These are the three leading competitors in the online market related to this application, it was felt that inspiration and knowledge could be gained from the success of these companies.

Generally hot food related sites are linked to warm colours, this colour scheme has correlation with the product, warm colours – hot food. The use of red and orange in these type of sites are consistent and it was felt that it was difficult to separate them as they are using similar colour palettes. (**appendix 2.2**)

The style of design intended for this application will be to promote a positive business opportunity that doesn't follow trends from the market. This concept was aimed at creating a unique product that will be memorable amongst competition.

Initial ideas and wireframes

As the research phase had been conducted, a sufficient amount of information had been gathered and this allowed the wireframing and interim design stage to take place. Important decisions in regards to which web pages were required and which functionality would be applied to them was met at this stage.

It was decided that wireframes would be designed for the home page, nearby results page, search page, business profile page and contact page. Additionally users will be able to login and register with the application, therefore two additional wireframes will be drawn to represent these areas of functionality (see **appendix 2.5**).

The first stages of the design section involved creating a colour palette and choosing a font that would help the flow of the design. Appendix 2.3 shows the initial colour palette and a sample font book that was designed. The chosen font is Signika, this is a Google font.

When designing the brand for I'm Hungry the aim was to create a word mark and logo relevant to the name of the project. Users need to relate to the project without knowing what it does, the branding should speak for the project.

Appendix 2.4 shows the concepts created on the path to this solution.

The brand rationale determines the main factors of the application. The brand represents location and hunger by using a location symbol as the main entity in the design. The first instinct when looking at the symbol was that the circular area in the middle of each pin could represent a plate. The next step was to apply a knife and fork inside the pin, enabling the negative space to represent a

plate with a knife and fork wrapped inside a location symbol. (figure1.4)



When creating a wireframe for the home page the initial paper prototype ([appendix 1.5](#)) was reflected upon. This prototype helped define the layout for the home page. It was decided that rather than having an about page, the design would host two separate areas on the home page that provide information that focuses on the purpose of sign up aimed at the demographic. In [appendix 2.5](#) the initial wireframes for the application can be seen. When the wireframes were created some creative experimentation was implemented.

The area that the team faced the most difficulty with was the navigation. The navigation area was changed quite a few times during this process. The original concept was to keep the navigation design mobile ready from the start. The application would have hidden navigation and users could click an icon that represents navigation and this would display the hidden nav. The problem with this method was that not all users could identify with the icon and it lead to a horizontal, traditional navigation being designed to aid usability. [Appendix 2.6](#) shows variations of the navigation.

Final Designs

When creating the final designs a few aspects were considered to ensure that usability would enrich the user experience. The aim was to ensure the flow of design would remain professional and reflective of the project ideology. Brand guidelines were created to apply a style to the project. To ensure the brand

guidelines were at a professional standard, information from an article at <http://www.logodesignlove.com/brand-identity-style-guides> was used for guidance.

The application doesn't directly sell food and it was felt that technically this provided a loop hole for the design team in regards to using stereotyped colours related to this market. It was decided that secondary colours would be used to create a complimentary colour scheme.

COLOR PALETTE



Web page design

The home page of the application is split into four distinct sections, those sections are the header, our mission, featured business and application features. It should be noted that there is a global footer that is on this page, this will be used on all pages. Each of the images supporting this section of the report have been annotated using the online tool ink2go. The homepage design, including all sections can be found in **appendix 2.7**.

The our mission section displays key word messaging, that should inform users about the ideology and intentions of the application. This section was chose instead of an about us page, similarly the features section at the bottom informs

users about the benefits of using the product. It was felt that keeping the about us information on the home page would decrease page to page navigation and would keep users closer to the fundamental usage of the application (see appendix 2.8).

The business section shows a grid with four accounts from the directory. Each account will have information such as the logo, address and a relative background image relating to the category of the business. Each account will have information on the front and back of the entry. The reverse side of the entry is split in half, one half shows the business name, logo, address, telephone number, website and social media. The other half of the card shows a review form, this will allow users to review each account in the directory.

The screenshot displays a section titled "Recommended Businesses" with the sub-instruction: "These businesses are the highest rated in the directory, they have been approved by other users".

Two business cards are shown on the left:

- Bengal Brasserie**
455 Ormeau Road, Belfast BT7 3GQ
[Read More](#)
- Mario's Pizza**
410 Woodstock Road, Belfast BT6 9DQ
[Read More](#)

A detailed view of a business card for **Zeera** is shown on the right:

- Zeera** logo
- Address: 280 Ormeau Road, Ormeau Belfast BT7 2GB
- Website: [Website](#)
- Phone: 07595371316
- Social media icons: Facebook, Twitter, Google+
- [View Menu](#)
- Form fields for Name and Review, with a "Submit Rating" button.

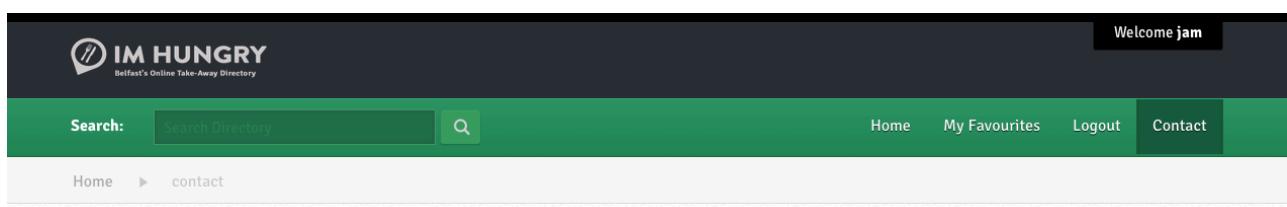
URL: www.majorproject.jamcrowe.com/browse/profile/zeera

The footer section of the application is global, therefore it will remain consistent across all pages. The footer section design took longer than any other area of the application. The footer was condensed with information and had another mobile device image, it was felt that pushing this much content towards the user could be considered overkill and the thought process for the section was reevaluated. It was decided that as the application would require an

enquiry from business users to have their business added to the directory the footer would be a suitable area to place this functionality (see **appendix 2.9**).

The login and registration section of the application required two separate interfaces. The functionality for this section will apply a modal or lightbox window. The design for this section includes a dark colour wrapped around the form, the modal template will have a white background and a shadow below each form. This effect shows depth in the design and presents the form with an individual style (see **appendix 2.10**).

The inner pages of the application remain consistent in regards to the header and footer. A specific header has been designed for the inner pages. The header area includes navigation, logo, search box, username and a breadcrumbs section. (*figure 1.6*)



The intended style of this application was to produce a realism effect on each of the pages. This effect is implemented by adding small details to the borders of each section. The detail to the design is very fine, the light borders add depth to each area and give it a more realistic and professional feel.

The profile page for each business has a few key elements that aid the usability of the application. This page shows the method for styling each account as though they are presenting the user with a business card. The profile page has three tabs that include menu, info and reviews. The menu tab shows a lists of categories that each menu consists of, this feature will help users quickly get to the area of food that would like to see. Users will be able to filter the menu using the input range element and each category is highlighted with a green banner. This banner works as a heading for each category and clearly represents it each section of the menu (see **appendix 2.11**).

The other design area that was added to the inner pages was a global sidebar, the purpose of the sidebar is to add additional information to the desktop version of the application. It was decided that two sidebar items would be added, one section would display the 8 closest take-aways to users geolocation. The other sidebar section will show the top 5 favoured accounts in the application. To aid the usability and add translation to the sidebars functionality font icons have been added to aid usability. (see [appendix 2.12](#))

System design

To develop a complex system to support this project, various technologies were required in order to secure the front and backend structure. To determine the most effective software to develop this project, various methods of planning have been applied. The process for selecting the chosen technologies aided the functionality on different levels of development. This section will describe the rationale behind the technologies, providing information of the duties required from them.

The application will require data entry to add each business account into the directory, a content management site will be used for this feature. The project will require php to produce the heavy lifting of backend development. It was decided that a framework will be used for this functionality as it will reduce development time. The project will use w3 standard coding, therefore the markup will be written with HTML5. Expanding on the front-end technologies jQuery, AJAX and JSON will be used to pass information through the application.

Expression Engine was chosen as a software package for this application as it is renowned for its flexible and secure system. This framework will enable less complex development when adding heavy duty functions to the system.

This report will include references to Expression Engine terminology throughout the remainder of the report. In order to understand how Expression Engine works and how it will be integrated into this application refer to [Appendix 3.1](#) for a user guide.

This project will use a responsive framework in order to speed up the development of the front end application. The chosen framework is ResponsableCSS, this framework has been tried and tested and proved useful each time. The CSS compiler LESS will also be used, the benefit of using LESS is that it enables quicker and leaner code, this will simplify writing code.

Third party plugins will be used heavily during the development of this project. Plugins will be required to perform tasks such as finding the geolocation of the user, providing lightbox functionality and simplifying ajax form submits. A full list of plugins can be found in [appendix 3.2](#).

Logic design

To create a spec for the project Google Drive was used to create an Excel sheet. In the Excel sheet a list of Expression Engine functionalities were listed, ranging from site map, channels, templates and functionality. This method of planning makes it easier to gather the range of add-ons required to develop the project (see [appendix 3.3](#)). The site map logic design decided which pages should be included in the application. The pages are homepage, search results page, nearby results, business profile and contact page. These pages have been agreed that they will provide the relevant information with the users, this site map was designed to use the least amount of pages possible to provide the correct information to users.

The channels that will be developed include pages, accounts and members. The pages channel will be used for non dynamic information. Although the content can be edited, as the text will be placed in global variables, this will allow the information to be edited from the control panel.

The accounts channel will hold information about each business account that has been published in the database. The accounts channel will have 25 custom fields. The noted fields from this channel will require Google Maps for Expression Engine, file uploads and text inputs. The file uploads will be used for any images that need to be added for each business.

EE - I'm Hungry

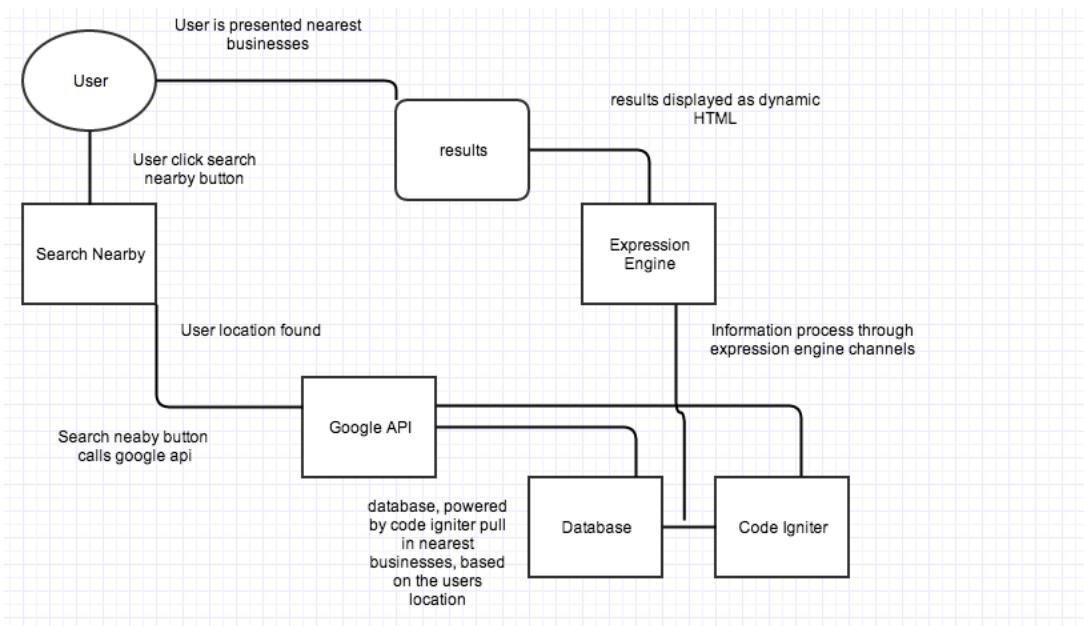
File Edit View Insert Format Data Tools Help All changes saved in Drive

Comments Share

A	B	C	D	E	F
1	Channel / Field	Field Type	Notes		
3	Accounts				
4	Title	Text Input			
5	Username	Text Input			
6	Address	Text Input			
7	Phone	Text Input			
8	Email	Text Input			
9	Location	Google Maps			
10	Latitude	Text Input			
11	Longitude	Text Input			
12	Logo	File			
13	Main Image	File			
14	Menu	Matrix			
15	Description	Text Input			
16					

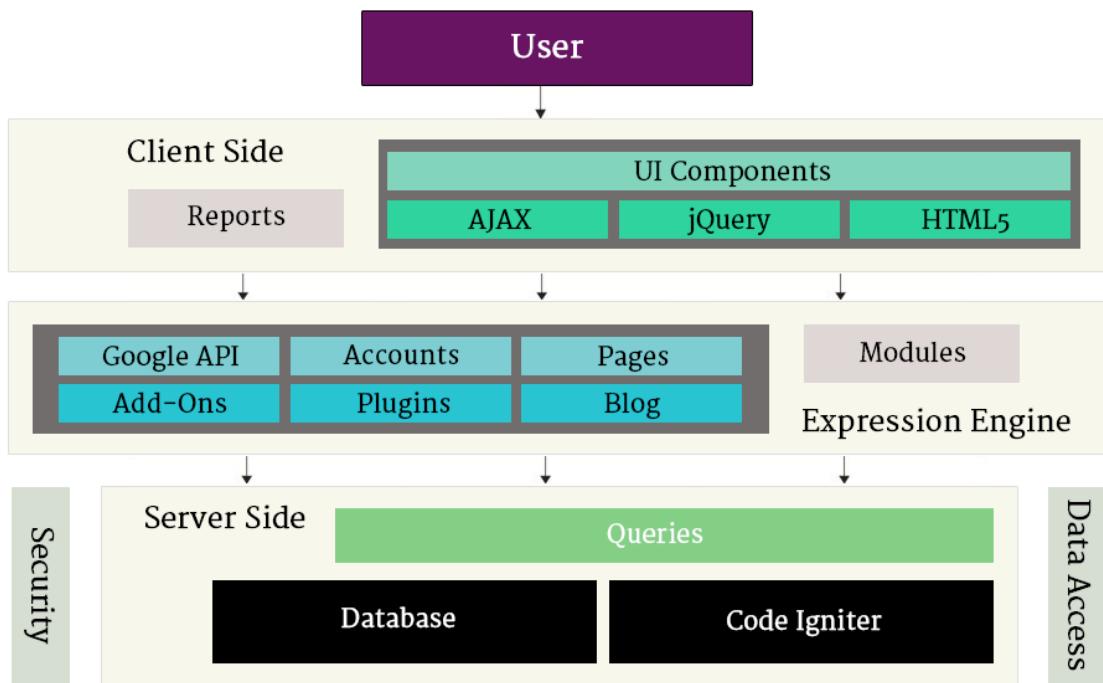
+ Site Map Channels Templates Functionality

The most complicated set of fields that will be created for this channel is a combination of the location, latitude and longitude fields. The location field relies on the latitude and longitude fields to pass the lat and lon values of each location into these fields. When the admin enters a new location for a business the lat and lon values of that location will be passed into the relative fields, this will enable the business to be located through the nearby results feature. The process the system goes through to deliver the user requested content can be seen below. (figure 2.1)



Platform Architecture

The platform architecture needs to have a comprehensive design, this will work as the structure and foundation for delivering content and developing the system. The platform architecture is designed to analyse any needs that are required for the application to be successful, as well as to promote a richer user experience and improve on application performance. The diagram below shows the platform architecture for this application. (figure 1.7)



Expression Engine design

To understand how to build an application with Expression Engine a great deal of research had to be completed. The book *Building Websites with ExpressionEngine 2* was purchased and examined thoroughly. This book introduced add-on developers and websites to visit. By reading this book it provided insight to how an EE sites works and explained the limitation and parameters that could be used when developing this application the framework.

Murphy, LM, 2010. Building Websites with ExpressionEngine 2. 1st ed. Birmingham: Packt Publishing.

Appendix 3.4 projects a diagram of how the data structure should appear after the correct channels and templates have been developed. Expression Engine allows for database development through a CMS control panel. This feature allows for SQL queries to be wrote and allows the developer to access the database direct tables directly.

When publishing with EE the developer will have a set of rules and instructions to set to enable the functionality of that particular table in the database. The accounts channel will be used to hold information about each business account in the database. When publishing the channel the developer will need to apply a short name for the channel. This is essential when developing Expression Engine loops, as the system will need to know which database table to pull content from.

Expression Engine plugins

In **appendix 3.5** a matrix has been created, this table shows the list of add-ons used to compile this application. The add-ons can be defined as three specific types, modules, extensions and plugins. The difference between the three types are, that plugins are enhancements that have small run-time. The plugin structure is developed to allow the developer to affect the output of Expression Engine in various ways.

Modules are weightier systems, anything that can be accomplished as a stand-alone function can be developed as a module. Modules store information into the database.

Extensions are hooks in the system, these hooks can use the add-ons to append and pass information to Expression Engine. Stand alone functionality can be added to the interface through extensions.

Expression Engine focuses and relies on add-ons for functional behaviour, this forces the system to operate as the developer desires. Below you can read about two plugins that will aid this system. A full list of add-on descriptions and usage can be found in **appendix 3.7**.

IF:ELSE Plugin

This plugin is used to help write conditional statements for templates in the system. This plugin has its own specific set of tags which can be found in the documentation online. This will allow the developer to manipulate how the content is looped and displayed to the user in a much simpler way. With IF:ELSE the developer will be able to write IF statements alongside the HTML5 that is integrated into the system. The example below shows, IF the Facebook custom field has been attributed for this entry show the information inside the statement. If there was no information in the Facebook custom field, the content inside the tags would be ignored. (*figure 1.8*)

```
{if acc_facebook}
<li class="facebook">
    <a href="{acc_facebook}" alt="Facebook">
        <i class="icon-facebook-circled"></i>
    </a>
</li>
{/if}
```

Freeform Module

The freeform module will be used to create forms in the application. Forms would be considered a stand-alone application as it has a specific duty to carry out. To integrate the contact form an ID must be assigned to it alongside other relative parameters. An area worth noting would be that the admin can manage all incoming forms from the control panel. Below you can see an example of a module loop as well as the control panel view. (*figure 1.9*)

Form	Submissions	Moderate	ID	Composer	Edit	Duplicate	Delete
Contact contact	Submissions 55	Moderate 2	1	N/A			
feedback feedback	Submissions 11	Moderate 11	2	N/A			

(*figure 1.10*)

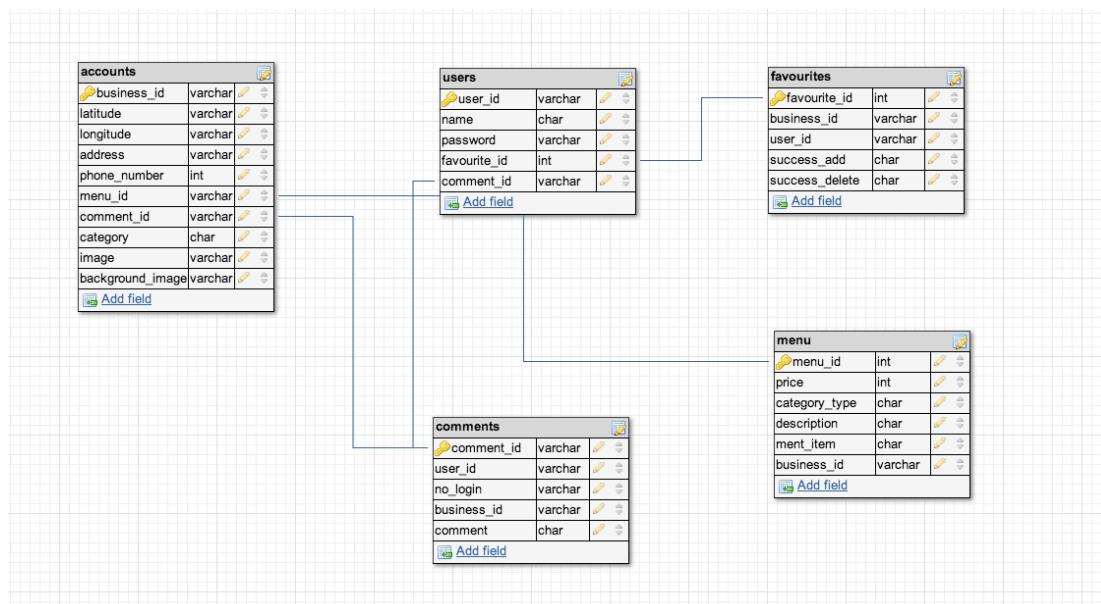
```
{exp:freeform:form
    form_name="feedback"
    form:id="contact_page"
    required="email"
    form:class="clearfix"
    inline_errors="yes"
}
{freeform:field:email attr:class="style" attr:placeholder="Email Address Here"}
{freeform:field:message attr:class="style" attr:placeholder="Enter Message Here"}
<input type="submit" class="inpbtn" value="Submit"></input>
{/exp:freeform:form}
```

Appendix 3.8 shows the groups of templates built to hold information in all areas of the application. The templates are split into relative groups, categorising the content each group can produce. For example, the scripted ajax calls with the URL parameter will target a template from that ajax group. The usercp group will consist of all templates related to the users information. The browse group will be templates with pages that aid the search functionality of each account. The pages template is for standard pages such as index and contact.

The channels that will be published include Accounts, Pages and Members. The accounts channel will hold all the information about each account. The pages channel will be set up for the standard inside pages. The members channel will hold information about all registered users for the application. As the system design has been put in place it was felt can see a case diagram could create a visual representation of the documentation provided for this section (see appendix 3.12).

Data design

The database design for this application will be done through Expression Engine. The logic design for this application applied the information that is required to develop this database. The channels, custom fields, templates, uploads and add-ons used in the project will publish tables in the database that hold information published from each respective group.

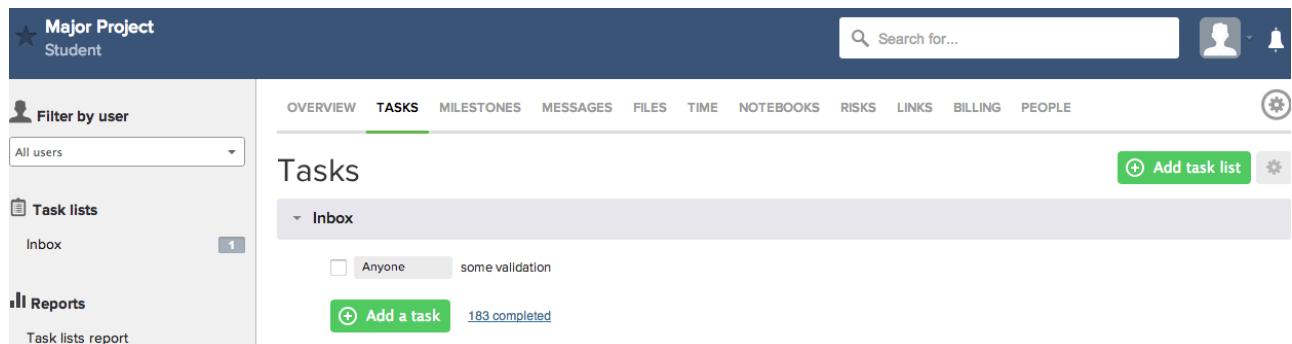


chapt. 3.

Implementation

To manage the implementation of the application, it was decided that the best way to keep on top of all tasks would be to sign up with an online project management tool. The chosen project management software was teamwork.com. This software was chose as the interface was easy to navigate around and the tools that come with the product were very useful. The tools included elements such as creating task lists, timing tasks, creating reports and generating gantt charts.

The original task for the development of this project was split into small projects to build the application in the correct order. The tasks started with the Expression Engine build, then moved onto HTML5 chops of the web pages and finally onto the integration. This process allowed for the development to be completed in cycles, completing each task before starting the next.

A screenshot of the Teamwork.com project management interface. The top navigation bar includes 'Major Project' (with a star icon), 'Student', a search bar, and user icons. Below the bar, the 'OVERVIEW' tab is active, followed by 'TASKS' (which is underlined in green), 'MILESTONES', 'MESSAGES', 'FILES', 'TIME', 'NOTEBOOKS', 'RISKS', 'LINKS', 'BILLING', and 'PEOPLE'. On the left sidebar, there are sections for 'Filter by user' (set to 'All users'), 'Task lists' (with 'Inbox' selected, showing 1 item), and 'Reports' (with 'Task lists report'). The main content area is titled 'Tasks' and shows an 'Inbox' section. It includes a checkbox for 'Anyone', a note 'some validation', and a green button labeled '+ Add a task' with the text '183 completed' next to it. There are also settings and add task list buttons in the top right of the main content area.

Appendix 3.9 shows a gannt chart generated during the implementation phase. This phase included important tasks as well as additional snag lists created during the testing phase of the project. There was 183 tasks completed in total.

Creating multiple environments

To connect to the database it has been decided that a multiple connection environment would be best suited for this application. A multiple environment connection will allow for creating connections to multiple databases. The use of this factor is to aid the online commits of the project for testing in a live

environment. When working of a local database the application will need to be put live and the database connection often needs to be changed each time to work in a production environment. To allow for the application to work locally and live without haven't to change connections a custom configuration bootstrap file for ExpressionEngine will be used.

It should be made clear at this stage how to navigate around the file structure of the application. To understand where to find each set of files refer to **appendix 3.13**.

Building Expression Engine Channels

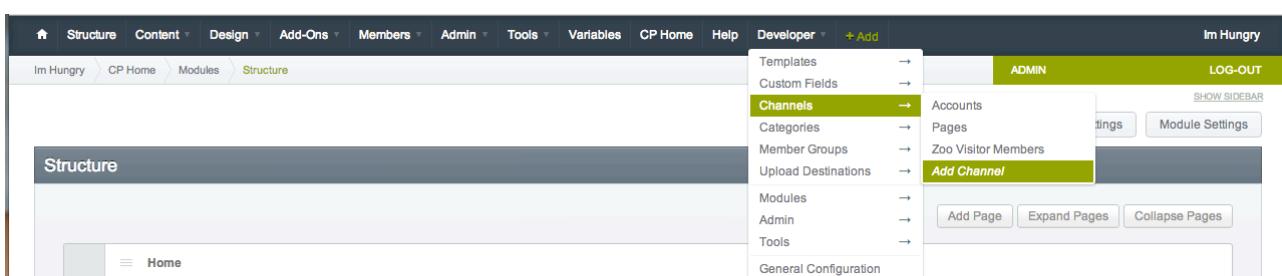
Before going into detail about a selection of relevant builds required to enable this system it should be noted that appendix 3.10 provides information about how to navigate around the admin side of the application. If the reader would like to access the control panel the login details are as follows:

Domain: <http://www.majorproject.jamcrowe.com/cpanel.php>

Username: peter

Password: majorproject

There were three custom channels developed for this application. The channels published included accounts, pages and members. The accounts channel will hold all information about each business account in the directory. Below you can see process for developing the accounts channel. (*figure 2.2*)



When publishing a channel, a name must be chosen and a list of rules and restrictions can be applied to that channel. These rules effectively give permissions for default modules to be linked to this channel. All settings will

remain as the default no, although comments need to be allowed in this channel as the user will be able to write reviews about each business. This will allow this channel to accept the comments module, linking that module to the accounts channel. (figure 2.3)

The screenshot shows a configuration form titled 'Comment Posting Preferences'. It contains five sections with radio button options:

- Allow comments in this channel?**: Options are Yes (selected) and No. A note below says: "Update all existing entries with this setting?"
- Require membership in order to post comments?**: Options are Yes (radio button) and No.
- Enable CAPTCHA for Comment Posting?**: Options are Yes (radio button) and No. A note below says: "A CAPTCHA is an image containing a security code that users have to submit. Please consult the user guide for more info."
- Require email address to post comments?**: Options are Yes (radio button) and No.
- Moderate Comments?**: Options are Yes (radio button) and No. A note below says: "If set to yes, comments will not be visible until a moderator approves them."

Building Expression Engine Custom fields

To build a custom field navigate to developer > custom fields > add new. A shortname needs to be added to this group. The short name is that name that will be used to identify the group. When this has been completed a new custom field group will be in the custom field section. The next step is to go to developer > custom fields > ..(the new field group) > add field. By adding the fields to this group an action is applied to each of the new fields added.

Below it displays all the custom fields added to accounts channel. When publishing custom fields the purpose of each particular field needs to be considered, each field can have an add-on assigned to it. When each account has been published, a unique entry_id is allocated to that entry. Below you can see a full list of fields added to the accounts channel. (figure 2.4)

The screenshot shows a table listing custom fields for the 'Im Hungry Accounts' group. The columns are ID, Field Label, Order, Type, and Delete. There are 15 rows of data:

ID	Field Label	Order	Type	Delete
27	Username	1	Text Input	Delete
13	Address	2	Text Input	Delete
10	Phone	5	Text Input	Delete
12	Email	6	Email Address	Delete
19	Website	7	Text Input	Delete
20	Twitter	8	Text Input	Delete
21	Facebook	9	Text Input	Delete
22	Google Plus	10	Text Input	Delete
23	Location	11	Google Maps for ExpressionEngine	Delete
24	Latitude	12	Text Input	Delete
25	Longitude	13	Text Input	Delete
78	Logo	17	File	Delete
72	acc_image	23	File	Delete
73	acc_menu	24	Matrix	Delete
74	Description	25	Textarea	Delete

You will notice that each field has an order and a type. The order of the custom fields is the order that the control panel will display each field in the control panel when the admin is publishing a new account. The field type is the functionality that each field can access information through. A text input for example is a single line text input, this field type is suitable for fields such as address name, lat and lon. When developing each of these custom fields the developer needs to create a short name for the field. This short name will be the tag used to call it through the Expression Engine loop.

The custom field for menu uses a matrix field, this matrix field creates a table in the accounts channel and allows the admin to enter the menu for each business. This field type clearly labels columns in the matrix, this informs the admin of what information goes into which column. Also for the menu number section of the menu it is important that only numbers can be added into that column. Below you can see the matrix configuration. (*figure 2.5*)

The location field uses Google maps for Expression Engine, this add-on allows the admin to publish the location for each account. When hooking up this add-on it is important to note that the longitude and latitude need to be defined within these field settings. Referring to *figure 2.4*, longitude and latitude have been defined as custom fields. The reason that this add-on requires these fields to be defined is that when the developer is pulling content from this add-on into application, they will be using the fields latitude and longitude. This add-

on will plot the longitude and latitude in to these fields when the admin locates an address for each business. (*figure 2.6*)



When the custom fields have all been published they need to be linked to the channel that will use them.

Building Expression Engine Templates

This project requires six template groups to be published. The template groups that will be published include pages, wrappers, browse, usercp and ajax. Each template group will be used to organise relative information.

The process of developing a template is to navigate to developer > templates > add template group. When arriving at this page a unique name must be given to the group. When this name has been submitted a new group will appear in the templates section above. The next step is to add templates to the templates groups developed. Developer > templates > ..(..your template) > add template. Each template will hold the relative HTML and EE content that is required for that group template (see appendix 3.8).

Building Expression Engine Upload destinations

The purpose of building Expression Engine upload destinations is to allow a directory for accounts images to be uploaded to. Two destinations have been allocated for this project, logos and main images. When building these directories a few rules need to be set to ensure that the application works correctly. Firstly an upload URL needs to be declared, this is so that the application knows where to upload the images to for this destination. For this application an images folder has been created with an upload directory. A file size needs to be declared, the largest file sized that will be uploaded will be 2mb, the file size has been stated in bytes.

Edit File Upload Preferences

Preference	Setting
* Descriptive name of upload directory	Main Images
* Server Path to Upload Directory	/usr/local/perm/vhosts/245568/webspace/httpdocs
* URL of Upload Directory	http://www.majorproject.jamcrowe.com/images/u
* Allowed File Types	Images only
Maximum File Size (in bytes)	2097152
Maximum Image Width (in pixels)	3000

The uploads directory is also important when creating custom fields that use the add-on file . When creating a custom field that has been given the fieldtype file, a upload directory must be stated. This allows for the upload related to that custom field to be uploaded to the correct directory.

Channel:entries loops

When each template group has been published it is important to understand that the content that each template holds can have static HTML or dynamic content published through the database. To enable dynamic content an Expression Engine loop needs to be written with applicable custom fields wrapped inside.

In the templates section it was discussed that various templates needed to be created, *figure 2.7* displays how templates will hold information by working together. This figure shows the wrapper template, which includes the html-header and html-footer, similarly to a php include. The wrapper template then uses the add-on stash to pull in content from another template group. (*figure 2.7*)

Wrapper- includes header and footer for the application and pulls in the template content
Using the add-on stash {exp:stash:get name='mainContent'}

```
{embed="wrappers/home"}  
The wrapper is embedded to the template  
{stash:mainContent}  
ALL HTML AND EE LOOPS  
{/stash:mainContent}  
*TEMPLATE
```

***WRAPPER**

The accounts channel is used for pulling in details about each business in the database. In order for an Expression Engine loop to pull in information from a channel, a channel:entries loop must be coded. A channel entries loop has parameters that indicate the restrictions as well as the output of that particular loop.

The loop below is targeting the accounts channel, there is limit of 1, this allows for the entry to only loop once and will show the content inside the loop once. There will be no pagination as this is a single entry for the account profile page. The url title pulls in the add-on freebie. Freebie targets the URL segments, this tag is declaring the title of the page to be the third URL segment, eg <http://www.testsite.com/first/second/third>. Any content inside the loop can be dynamically looped through the system, outputting information from the database (*figure 2.8*).

```
{exp:channel:entries channel="accounts" url_title="{freebie_3}" limit="1" dynamic="off" status="open|active"  
disable="pagination|categories|category_fields|member_data"}
```

Favourites

The favourites section has a few elements that are required to make it work successfully. The favourites are added to the database through the add-on favourites. This add-on allows users to favourite the businesses they like and add them relatively to their registered account. As with most Expression Engine

functionality an Add-on isn't as simple as installing it and it works. To ensure the favourite system works correctly it needs to be Expression Engine integrated and a script needs to be written to allow flowing functionality.

The first step is to set up an xml template, this template has been created in the usercp template. This template uses the favourite:save function, this function will allow a success message to be show if a favourite has been successfully added to the database.

A heart icon from the font pack fontello has been designed at the bottom of each business account. The class favourite-link has been added to that icon, when the link has been clicked and the member_id is equal to 5, this will allow users to add the business to their favourites.

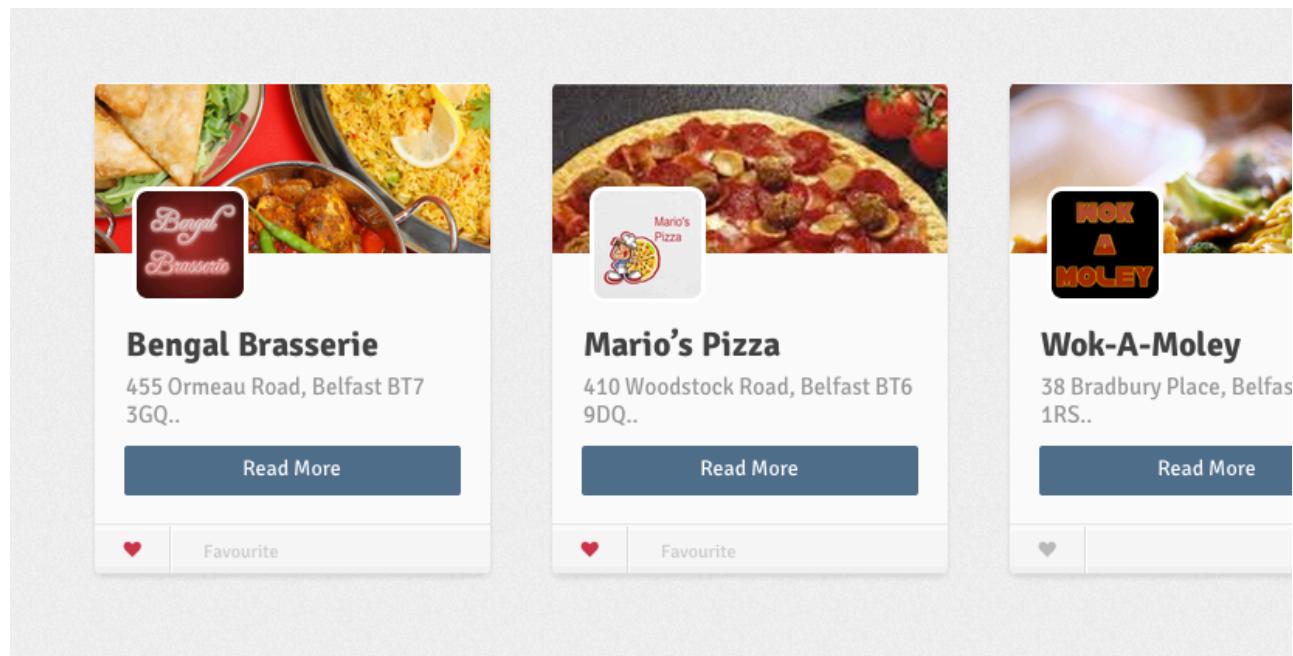
To save a favourite, the URL needs to link to the xml template that was previously published, a variable is then added after. The variable value is that of the data-url attribute, that attribute has been assigned the EE tag {url_title} (*figure 3.2*).

```
29 |         <section class="favourite_wrap">
30 |             <a href="javascript:void(0);"
31 |                 class="favourite-link addAsFav accountFav{url_title}" data-url="{url_title}"
32 |             ><div class="fav_wrap"><i class="icon-heart"></i></div><div class="fav_word"></div></a>
33 |         </section>
```

```
$(".favourite-link").click(function () {
  if (member_group_id == "5") {
    //if user is logged in do this
    eachAccount = $(this).attr("data-url");
    //value is dynamic through ee, data-url value is the url of each business
    $(".accountFav" + eachAccount).addClass("addAsFav") && $.ajax({
      type: "GET",
      url: "/usercp/my-favourites-add/" + eachAccount,
      dataType: "xml", //gets xml template and account url
      success: function (event) {
        $(".accountFav" + eachAccount).css("color", "#ccc").removeClass("addAsFav").html('<div class="fav_wrap"><i class="icon-heart" style="color: #ccc;"></i></div><div class="fav_word">Business favourited</div>')
        //adds colour to icon and favourite text to each account that has been favourited
      }
    })
    //success
  });
} else $.get("/ajax/register-login", function (e) {
```

If the favourite has been saved successfully the class addAsFav is removed and new elements are added to the favourite_wrap section. A red coloured heart is and the word favourite are part of the elements added to the HTML. This indicates to users that the business has been successfully added. For validation for this section if the member_group_id is not equal to 5 the script will fetch

the login/registration forms with the lightbox function. Below you can see an example of two accounts that are favourited and one not favourited (*figure 3.3*).



The favourites can be viewed on the my-favourites page for each individual user. When this page is accessed it declares at the top of the page how many favourited businesses that user has linked to their account.

You have 16 Favourites linked to your account

To apply this count to the users favourite page a favourite:entries loop must be coded. This loop declares that only open and active accounts can be displayed and that its targeting accounts channel(where all the businesses were published). A conditional statement has been set up two display two different messages depending on the amount of favourites linked to the user. The first conditional states that if the favourite count is equal to zero, display a message that tells the user they have no favourites. If there are favourites the amount will be shown.

```

12 {exp:favorites:entries channel="accounts" favorites_count="yes" status="open|active"}
13 {if {favorites_count} != '0'}
14     <h1 id="favourites_list"><span id="favourite_count">You have {favorites_count}</span> Favourite<span
15         id="favourite_plural">{if favorites_count != 1}s{/if} linked to your account</span></h1>
16 {/if}
17 {if {favorites_count} = '0'}
18     <h1 id="favourites_list">
19         <span id="favourite_count">{favorites_count}</span>
20         Favourite<span id="favourite_plural">{if favorites_count != 1}s{/if}</span>
21     </h1>
22     <div class="fav_report">
23         <h3>Sorry you haven't selected any favourites from the directory yet</h3>
24         <p>Why not add some now by seeing who's <a href="/browse/nearby-results">nearby</a></p>
25     </div>
26 {/if}
{/exp:favorites:entries}

```

Favourites sidebar

The top rated business needed to be in a list, it was decided that the loop would be coded inside an unordered list, wrapping around a list item. The parameters for the loop declare the accounts channel should be targeted, ordered by count, with a limit of 5.

The outcome of this loop is to show the top 5 rated businesses, each business will show the number they are rated, the title of the business and the amount of likes received for that business. An additional add-on has been used in this loop, that add-on is called EE-hacksaw. The EE hacksaw allows us to limit the amount of words in the title of each business and append characters after the limit has been broken. This add-on is useful for long named businesses, as their titles can break the design (see *appendix 3.5*).

Comment:entries loops

Comment entries is a module which would be commonly used for blog sections in Expression Engine sites. The comment entries loop works as form. All form fields that are coded inside the loop work as the fields of the comment entries form. The comment entries loop can be located on the reverse side of each business entry displayed. Each business can be reviewed, by filling in the form in the loop.

The review will be submitted on that businesses profile page in the reviews section. The form is able to dynamically populate individual entries by targeting the entry_id, entry entry published entry is assigned a unique entry_id. A class has to be applied to the form rather than a form ID, The reason for this is

because there will most likely be multiple entries on each page. This form will be submitted via ajax during later client side scripting, therefore ajax= “yes” will permit this.(figure 2.9)

```
{exp:comment:form channel="accounts" entry_id="{entry_id}" dynamic="yes"
form_class="review_submit" ajax="yes"}
    <label>Name</label>
    <input type="text" name="name" placeholder="name" class="review_name" style="border: 1px solid #ccc !important"/>

    <label>Review</label>
    <textarea class="mycomment" name="comment" cols="70" rows="2" style="border: 1px solid #ccc !important" class="review_comment"></textarea>
    <input type="submit" value="Submit Rating &quo;" name="submit" class="back_button1">
</div>
{/exp:comment:form}
```

To display reviews a comment:entries loop must be wrote into the profile page. This loop will display the name of the reviewer, the date of the review and the review left by the user. Each comment has been looped around the HTML5 article tags, this helps keep the loop semantically correct. The comments are linked to the accounts channel with a limit of 50 comments, sorted in ascending order (figure 2.10).

```
<div id="tabs-3">
    <div class="tab-info">
        <figure></figure>
        <h3>{title}</h3>
        <div class="address">{acc_address}</div>
    </div>
    <div class="blog_wrap">

        {exp:comment:entries channel="accounts" sort="asc" limit="50"}
        <article>
            <div class="name">
                <p>By {name} on {comment_date format="%Y %m %d"}</p>
            </div>
            <div class="comment">
                {comment}
            </div>
        </article>
    {/exp:comment:entries}
```

An issue that arose during the review section build was that after each review was submitted the page would reload by default. This was an issue as users would not know whether their review had been successfully posted. It was decided that the best way to tackle this issue would be to use the plugin form.js. Using an ajax form would be beneficial for this function as this will prevent the page from reloading after submission.

Appendix 3.11 shows the ajax script for each review form. The form has a hidden gif, this gif is used to make users aware that their review is being processed. When the form is submitted the gif is shown and the input values are cleared, returning the form true. When the form has successfully been submitted the gif is hidden and an alerted success message is displayed.

Registration and Login

The registration and login functionality of the application has a few components that make it work. The login uses various plugins that includes Expression Engine Zoo Visitors and Lightbox Me. Zoo Visitors allows users to add their login and registration through EE, this will insert the entry into the database. Lightbox Me will be used to place the form in a lightbox window.

The login and registration forms need to be inside an Expression Engine loop. The loop is relative to the add-on used for this feature. The parameters of the this loop allow group members 5. The parameter json=“yes” has been applied as the template for registration and login will be called through the ajax form datatype parameter in the client-side script.

The form is distinguished by the ID. Inside the loop includes, form fields, each of these form fields are identified by EE by the name attribute. There are two hidden fields linked to the password input element. Those hidden elements agree to accept terms for registration and apply the group_id “5” when users register. Group_id 5 is the group assigned to registered users (see **appendix 3.12**).

The login and registration forms are templates in the ajax group template. An important factor in how the ajax form will work will depend on the group_id of the user. Group_id (5) is assigned to each registered member. This script works similarly for the favourites section, registration and login. The login script will be used as an example and should be considered as relevant to how the other builds were developed (see **appendix 3.13**).

The first step of this script was to define the group_id as a global variable that can be understood as javascript. Currently the group_id is an Expression Engine formatted database value. To make this value jQuery usable a variable needs to be wrote with the value of an Expression Engine tag, {logged_in_group_id}.

The important issue here is that this variable needs to be declared inside script tags as well as inside the Expression Engine tags. The script needs to be placed in the header of the html-header template. Below you can see an example of the script written to declare this functionality. Also there is another screen shot that shows the inspect element function Google Chrome offers, this example shows that after login the group_id is declared as 5 (*figure 3.1*).

```
22
23     <script>
24
25         //Show group id of user
26         //5 = Registered user
27         member_group_id = {logged_in_group_id};
28         var site_url = "{site_url}";
29
30     </script>
```

```
//Show group id of user
//5 = Registered user
member_group_id = 5;
var site_url = "http://majorproject.dev";

</script>
```

Now that the group_id is declared is the next part of the script can be written. A class named login-link has been assigned to the login button in the navigation. When this class has been clicked an IF statement is immediately applied, the script reads that IF the group_id is not equal to 5 then run the script(5 being registered user). The script then uses \$.get request to fetch the template user-login from the ajax group. This login form can be found in this template.

The next step of the process integrates the plugin Lightbox Me. This plugin is targeted by accessing #user_favAccess, this div is where the plugin will be applied to when they \$.get request fetches the template. The jQuery element .html is used to get all elements from the ID. When the lightbox is

activated the background is set to white with an opacity of 0.9. These rules apply the white background described in the design section of the report.

When the lightbox is activated the class .userProgress is hidden, this element will be used to produce a success message if login is successful. The plugin form.js targets the form ID, the template will be pulled in by the \$.get request, using the datatype json. If the event is successful the member_group_id is changed to 5 and the user is now logged in. The additional functionality removes the form elements and changes the navigation as the user will require different functionality has a logged in member. The registration and login nav elements are removed and the logout and my favourites navigation links have been appended.

The final part of the script animates the hidden userProgress element to show a success message. The lightbox has been triggered to close after the success message has been show. An else statement has been coded to alert the user if something about their login was unsuccessful (see [appendix 3.4](#)).

Flip Animation

An intricate part of the functionality for this application is how users receive their information. As stated in the design specification and during the comment:entries section, each account in the directory will have their information displayed on a double sided div. The information will be displayed on the front and back of the entry. To create this functionality two sets of code had to written, CSS3 for applying the 3d transform and jQuery to apply the flip animation when an event occurs.

The flip animation was inspired by <http://davidwalsh.name/css-flip>. This tutorial shows how to write CSS that applies the flip animation. The article goes into further detail and allows developers to apply the flip based on hovers or clicks. In appendix 3.6 the CSS has been can be seen.

The process of how the CSS works is that the flipAnimation container sets the perspective of the container. The inner element that is targeted will flip 180 degrees when the event occurs. The front and back elements are positioned with absolute positioning, the back-face visibility of the elements is hidden, this prevents any elements being displayed during the flipAnimation. The z-index rule for the front of the div is low so that it is displayed as the top layer.

The main issue that came with making the function element applicable was that the tutorial <http://davidwalsh.name/css-flip> based the flip animation upon the flipped div being the same width front and back. The div in this application is twice as wide on the reverse side. The double width is there to allow for room for a review form as well as additional information about the account. The issue being that when the div was flipped it would need to know whether to flip left or right to ensure that the flipped div stayed inside the 1180px container.

The concept behind this functionality was to target the main wrapper around the account business cards, that wrapper is named accounts_frame. The accounts_frame then has a function appended to it. To target the flippable divs, a variable has been declared for which area to target for each flip. The variable is named elements, this represents the element that will be part of each animation. The method used to target the flippable area was `$('.accounts_frame >div >div')`; what this is saying is target all children elements of the div (the elements we want to flip). The elements variable is then chained with an each function, this each function sets a set of rules for each account business card. Three variables have been set, \$front, \$back and \$click.

\$front is used to target the first div for each account business entry(this is the front of the div). \$back is used to target the last div for each account business entry(this is the back of the div). \$click is set to make the button class the clickable area to initiate the click. Each button on the front of the card has the class button. As the variables are inside an each statement it is important to

declare that the button class used the also has this as part of the script, this is used to set the current object for the each loop.

There are two clickable areas that cause events within the flip animated divs, when the button class is clicked this will flip the div over to the back, event.preventDefault() has been used to prevent the browser from trying to open a link. The backside has a close button, when this is clicked the card flips back to the front using the same method.

For the div to find out which way to flip the card a separate function was created. To create this function new jQuery elements had to be explored and understood to ensure that the flip would be able to output the correct position. The jQuery element .offset() was used to create this function. Offset allows the statement to get the position or coordinates of all elements, or a set of matched elements that are relative to the targeted area. This rule will allow the function to find the position of the element. The statement reads, for each \$elements, if left position is greater than width add rightSide class, remove leftSide and vice versa.

Geo location

Geolocation is an important requirement for the development of this application. Perhaps the primary feature of this application as it allows users to search for businesses that are located near to them. In order to enable the function a few steps must be taken, a geolocation navigator script needs to be run, the values of the users coordinates need to be passed into the Expression Engine add-on Google maps for Expression Engine, then finally search results will be pulled in from the accounts channel listing the businesses in order of closeness to the user.

The first step of this process was to write the HTML and the Expression Engine loop. A div with the ID nearby_results_wrap is wrapped around the Expression Engine gmap:search tags. The gmap search tags works as a form with a set parameters that target the accounts channel. The important part of this script

leads to the attributes that have been written in the geocode_field parameter. This parameters targets the input element with the name location and the class coordinates. (*figure 3.2*)

```
77      <div id="nearby_search_wrap" class="hide2">
78          {exp:gmap:search
79              channel="accounts"
80              latitude_field="acc_latitude"
81              longitude_field="acc_longitude"
82              geocode_field="location"
83              distance_field="distance_max"
84              cache_post="false"
85              return="browse/nearby-results"}
86          <div>
87              <input type="hidden" name="status" value="open" />
88              <input type="hidden" name="location" class="coordinates" value="" />
89              <input type="hidden" name="distance_max" value="50" />
90              <input type="submit" value="submit" />
91          </div>
92      {/exp:gmap:search}
93  </div>
```

To pass the lat and lon values into the empty value in the location input element, the geolocation.js script needs to be implemented. The nearby results button has the ID nearbySubmit, when the ID is clicked an event occurs. This ID pulls in the geoPosition.init() and geoPosition.getCurrentPosition from the geolocation.js script, it passes the function nearbySearch and noLocation through that function. The nearbySearch function passes the lat and lon values into the coordinates class, this is added to the location input. These scripts together allow the Expression Engine plugin to determine the location of the user and present the related results on to the nearby-results page (*figure 3.4*).

```
483      function nearbySearch(event) {
484          $(".coordinates").val(event.coords.latitude + "," + event.coords.longitude);
485          $("#nearby_search_wrap form").submit()
486          //collects longitude & latitude values
487      }
488
489
490      $("#nearbySubmit").click(function () {
491          geoPosition.init() && geoPosition.getCurrentPosition(nearbySearch, noLocation)
492      });
493  
```

Way points, bells & whistles

It was felt that overall the user experience of this application would be very important and how the information was delivered to users could determine how they digest the information. Waypoints are a good way to trigger events with a purpose at a particular point of navigation. It was decided that as the home page is separated into sections, when each section reaches the top of the window a waypoint marker would trigger a new effect.

There are two main features in regards to the home page waypoint applied transitions. The first point is a function that adds opacity to the header as it

fades into the next section of the page. The function runs when the window scrolls from the top of the page and chains the css rule of opacity depending on the scroll position of the browser. The secondary section of the page is a clean grey/white background and the use of opacity helps blend the scroll into that section smoothly (*figure 3.5*).

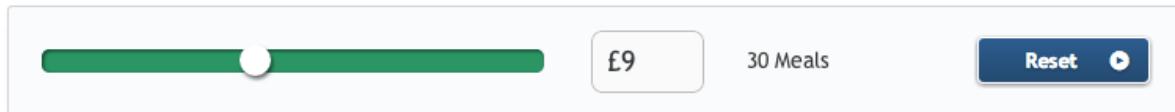
```
733     var windowHeight = window.innerHeight; //inner height of window
734     $(window).on('scroll', function() {
735         var scrollTop = $(this).scrollTop();
736         $('#search').css('opacity', (1 - scrollTop/windowHeight) );
737         //adds remove opacity of area depending on scroll position of the div
738     });
739 }
```

The second point to make about the waypoint functionality is that a css animate library has been added to the way points. Each of the animation rules apply where the selected elements will appear from and what transition they will use when they enter the screen. The script below is an example of how the way points for the our mission section on the home page are applied. When the div #latest reaches the top of the screen a setTimeout function adds a delay to each of the elements and adds the class determined through the animate.css library. On this occasion two elements will fade in down and one will fade in up (*figure 3.6*).

```
$('#latest').waypoint(function() {
    setTimeout(function(){ $('#latest h2').addClass('animated fadeInDown'),0);
    setTimeout(function(){ $('#latest .spantext').addClass('animated fadeInDown'),100);
    setTimeout(function(){ $('#accounts_animate .account_container').addClass('animated fadeInUp'),300);
}, { offset: '20%' });
```

Menu Filter

To filter the menu an input range element will slide left to right filtering the prices based on the value displayed beside the element. Each menu price has the class menu_price. The menu price content has an integer as the value. A variable named filter has been set, this holds the value of 0 and represents the value of the menu filter. When the input range changes position the filter variable value increases or decreases. The statement reads that if the parsed integer is greater than the menu price value fade out. The number of meals can be displayed through the count variable (see **appendix 3.10**).



Conditionals

A small feature on the profile page is the dynamical population of the menu categories. The categories are wrapped in conditional that states if category is available create an a href tag. The dynamic tags have the id {category}. This applies the category name to each entry. The plugin smooth scroll is used to create a smooth transition to the section of the page targeted (see [appendix 3.12](#)).

EE sitemap

A dynamic site map was created for this application. All published entries are displayed, this includes pages and accounts. The outcome of a dynamic site map is that anytime a new business has been published that business will appear in the site map. The loop for this functionality targets the accounts channel and orders the list by title. The site map is dynamic in various ways, you can see from the figure below that the link is populated based on the url_title and the name displayed is listed as the title pulled from the accounts channel.

```
33
34
35 <ul>
36   {exp:channel:entries channel="accounts" dynamic="off" status="open" paginate="bottom"
37     disable="categories|category_fields|member_data" orderby="title"}
38     <li><a href="/browse/profile/{url_title}" title="{title}">{title}</a></li>
      {/exp:channel:entries}
    </ul>
```

Chapt 4.

Testing

To test this product the black-box method of software testing was applied to indicate whether a verification and validation practice was successful throughout the development lifecycle. The requirements specification will be used to base test cases on the outcome of the requirements. Using this method of testing has helped evaluate the features on the application.

Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

Williams, L, 2006. Testing Overview and Black-Box Testing Techniques . Testing Overview and Black-Box Testing Techniques , [Online]. 1, 1. Available at: <http://agile.csc.ncsu.edu/SEMMaterials/BlackBox.pdf>[Accessed 15 April 2014].

The technique for verification will show whether or not the application behaves the way it is required to and the validation will show whether or not the product meets the requirement asked of it. This report will use acceptance testing and the functional/system testing method with in the black-box process.

Aside from black-box testing a further case study will be made using the web developers checklist. This checklist will help identify areas of testing that had not been previously considered.

Functional and system testing

Using this method of testing will examine the design of the application at a high-level and test requirements specifications stated in **appendix 1.3**. The

functional testing will be compared against the functionality declared in appendix 1.3 and the system testing will be initiated against the fully integrated system, determining the compliance of the system against the requirements. During the system testing method several classes of case study can be applied, stress testing, performance testing and usability testing.

Verification & functional tests

A full list of functional tests can be found in **appendix 3.8**. This phase of testing created case studies against areas such as form validation, flip animation, search functions, links and browser testing.

There were a few issues that arose during this testing phase of the application. Three case studies tested that coexisted surrounded the review section functionality.

This function allowed users to submit a form and alerted a message informing users that review had been submitted. Although the fault that occurred during the process was that if a user submitted a blank message an alert message would still be displayed informing the user that their review was submitted. This issue was tackled numerous times and a validation process was never determined for this feature. The outcome of this testing review was determined as a fault, the functionality rationale for fault was agreed as the form potentially had validation, although it didn't fill its full requirement.

Browser testing was an important feature in this application. The application was able to process full functionality in all high-level browsers, these browsers included Opera, Safari and Firefox. The issues arose using the Internet Explorer browser. A modernizer script was added to aid browsers that didn't support the CSS3 effects that are applied during the waypoint animations and the flip animation. The flip animation was an issue using IE10, the flip would not z-index the animation correctly initially. This issue was decreed as a fault. A fix

was issued that allowed entry to be viewed on both sides, although the transition wasn't smooth.

It can never be determined what browser or device that the application will be viewed on. In case a browser is outdated it has been decided to post a notification if the browser is older than IE8. The message will link the user to <http://browsehappy.com/>. This site informs users that they should update their browser in order to have enriched usability of their online experience. (*figure 4.1*)

```
<!--[if lt IE 7]><p class="chromeframe">Your browser is <em>ancient!</em> <a href="http://browsehappy.com/">Upgrade  
to a different browser</a> or <a href="http://www.google.com/chromeframe/?redirect=true">install Google Chrome Frame<  
/a> to experience this site.</p><![endif]-->
```

Performance testing

To test the performance of the application case studies were applied to test the page speed score and optimisation of images. Applying this methods can increase overall usability and determine what areas need to be reviewed in regards to improving the performance of the application.

It was decided that two separate page speed tests would be conducted and the results from both tests could determine an overall snag list for improving the page load speed for this application. The tests will be performed on the home page of the application.

Usability testing

When considering how to approach the usability testing of the application the web developers checklist was used to find out which element could be added to the application to improve usability. The usability factors included adding a 404 page, fav-icon, friendly-URL's and including search boxes.

A 404 page had not been previously considered and neither had a fav-icon. These two usability features are in effect in all websites and it was important to add them at this stage, a design has been created and implemented into the application to adhere to this usable feature (see **appendix 4.4**).

The application had already implemented user-friendly URL's. This conclusion was agreed after reading an article at <http://moz.com/blog/11-best-practices-for-urls>. This article declared that descriptive URL's are better than numbers. This has been completed by creating semantic URL's(new business URL's will be named the same as the business title). Expression Engine by default separate words by hyphenating them, this was also recommended. HTML validation has been completed at <http://validator.w3.org/>

Performance testing was added to improve the page speed of the application. See **appendix 4.3**.

Validation & user reflection

To validate the black-box testing from a user perspective it was decided that two groups of users would view the project with two plugins that allow for introductions to websites. The plugins used are called intro.js and trip.js, the implementation of these programs will assist users with a step by step guide of how to navigate around the project. To correlate with this phase of testing a survey will be completed by users after these phases to see if the aims and objectives of the program have been met.

Trip.js

Implementing trip.js was a simple task, to integrate this plugin an external stylesheet and js script were connected and then a script was run that targeted divs selected by the developer with a message that described the div. The rationale behind the method of testing was to see if users could agree with the implemented areas of the screen. During this testing process users were able to identify with highlighted areas. (*figure 4.4*)

```

28         var trip = new Trip([
29           {
30             sel : $('#logo'),
31             content : 'This is the logo'
32           },
33           {
34             sel : $('#worlds_birth'),
35             content : 'This is the navigation'
36           }
37         ], options); // details about options are listed below

```

Intro.js

Intro.js had a similar set up to the previous testing method. The integration included add external scripts and linking the information from data type attributes. Users found intro.js to be easier to understand than trip.js.

```

31   <script type="text/javascript">
32     if (RegExp('multipage', 'gi').test(window.location.search)) {
33       introJs().start();
34     }
35   </script>
36   <!-- Add IntroJs styles -->
37   <link href="introjs.css" rel="stylesheet">
38
45   </div>
      <div id="worlds_birth" data-step="2" data-intro="Navigation">

```

Test results

The tests results from the user testing showed that all users were aware of what the message the application was providing and the service that it offered. None of the users had any issues registering with the application. 90% of the users received an email after registration. To determine what the issue was with the one user that didn't receive an email the members area in the control panel was examined. The user had misspelled their email address. The email added was markmcdowell@hotmail.con, the email should have been .com at the end.

All users were able to post a reviews and were able to interact and identify the favourites functionality. One user was not able to use the voice recognition input for search. There was no explanation for this fault. Although an assumption that the user had no voice input on their device or else their accent could not be understood was determined. The test can be taken at the following address: <https://www.surveymonkey.com/s/WF72FG3>

Chapt 5.

Evaluation

As the project has now been designed, implemented and tested it allowed for reflection on the success and failures that were met during these processes. The initial reflection lead back to the requirement specifications to understand whether the priorities were truly correct. This practice enabled an evaluation of the requirements and an understanding of any failures. The results from the testing process will be discussed and the overall outcome of the project will be assessed.

The testing process of this application had users view the project with an assisted step-by-step guide and then answer a survey based on functionality and rationale of the application. The results that were anticipated from the testing phase was that the users would follow a precise pattern when navigating through the application. The intended pathway for first time users, was for them to land on the home page and make their way through each section of the home page. This pathway would inform users about any features that application has and therefore increase awareness of what the application can actually do.

After users have navigating to the bottom of the home page, they should make their way back to the featured business section and either attempt to favourite a business or else apply the flip animation and see further details. The next step would be to view the menu of that particular business and then filter and read reviews. It was always assumed that features such as reviewing a business or using the search nearby feature would be recognised more by returning users that were familiar with the product.

The results received from the user testing provided a completely different insight into how the application had initially been intended. The first notably factor was the users felt the main background image was irrelative to the project

ideology, although they did feel it was complimentary with the other elements. The pathway that users followed on entry to the application can be seen in appendix 4.4 a UML diagram has been created.

The verification testing opened up avenues that had not been considered beforehand. Areas such as 404 pages, fav-icons, page performance and functional testing had been overlooked as the project had been kept close at hand for so long. Spending consistent time on this project created a fixation that deterred an open perspective and lead to small details being overlooked.

The project outcomes generated the majority of the requirements that were stated. The project allows for users to login and register with the application, this creates interactivity and brand recognition. If a user is willing to register they are willing to return to the application and this makes adheres to many of the functional requirements.

The administrative role has been given less priority in this report than it actually deserves, as there were two sides to the usable features that were important for this project to be truly successful. For this project to be commercially viable and admin would need to control the system from the control panel. An admin can manage all emails and reviews that are input by users of the application. An admin also controls all the content and they should be able to clearly identify the simple solutions to managing this content.

The methodology chose for this application was the waterfall method. In retrospect this methodology was not the best process for this application. Developing an Expression Engine application requires persistent developing hours and as the technology is most likely going to be new to a developer the functionality requirements are continually considered during all aspects of the build.

Developing with an unfamiliar software can be daunting and can often lead to dismay and questioning judgement. This chosen methodology on reflection that would have suited this project would have been the agile method. This would have allowed cycles of development and would have prevented the intimidation that came with working with new software.

This particular project lead to over extending phases of the methodology due to the fact that Expression Engine understanding was always being considered. During the phases that were not related to implementation, Expression Engine add-ons and documentation was constantly being reviewed and understood. Developing with a new software with unfamiliar add-ons was quite daunting as the thought of the unknown was always kept in mind. The meaning behind this was that if a plugin was used and it was not fully understood then if a problem was to occur then there would be know one to find that solution.

There were positives that came from using a new technology, the ability to build a sandbox of tools and implement those tools for usability from an administrative and user role was a huge task. The knowledge gained from this build has been very useful and the tools that are required to apply functionality opened up new avenues in regards to future projects and the direction that they may take.

The plan changed when two stakeholders were introduced to the project and gave their professional opinions on how to build the product. The project was originally considered to be a basic Expression Engine application that would provide experimentation with a CMS, which was chosen due to its user-friendly interface. When professionals intervened with the project plan, they revealed a realisation of how lucrative this product would be, due to the system it was built with. The plan shifted from marketing to students to building a secure system with reliable add-ons allowing it to be sold as a commercial product.

Chapt 6.

Conclusion

This report covered concept that took the web application I'm Hungry from start to finish. The project was managed with the waterfall methodology and this method helped breaks the deliverables down into phases that were completed step by step. The research conducted was used to produce results that were considered true and correct in order to deliver the project as close to the required specification as possible. The implementation of this project included building an Expression Engine sandbox and integrating it into a client-side scripted application. This implementation inevitably produced a dynamic web app that could be managed by an administrator and provided interaction by users.

To project manage this application the software Teamwork was used to keep the deliverables on track. At an early stage of the project a gannt chart was created and this chart outlined the tasks that were originally conceived to produce a major project. As the project progressed the initial deliverables had not been rationally decreed. When it was realised that the tasks at hand were as vast as they were a new project management plan needed to be met. In order to keep on top of the development tasks, each task that was required was listed with a project timescale and date of completion. This tool helped manage the development and projected timescales to complete each task within. Overall it helped create closure in regards to the finish of the project.

The design stage of the project was perhaps the simplest or smoothest phase. During this phase there was a clear ideology that was aimed to be met by applying a particular style to the brand. It was felt that the design stage met all expectations and overall created a professional looking website that represented the brand through recognition and style.

When developing this project the initial requirements didn't change that much from start to finish. Although what did change was the method of developing the project. When the concept of building a project with Expression Engine was conceded it was felt that it would be an interesting method of building a project and that it wouldn't cause many issues. Although in reality building with a framework and using a CMS that requires a specific set of add-ons restricted the freedom for custom coding for this application.

A question that has been considered since the build was completed, was whether or not the project would be built the same way if it was to be done all over again. This question caused a predicament, as the project would not have been able to meet as many requirements as it did if it hadn't been integrated with Expression Engine. Although if the entire project was built bespoke it would have had a different set of aims and objectives as the admin management and business input would have been planned differently. Without the CMS the data entry would have taken much longer and there would have been no way to manage all user inputs from one panel.

As the project manager, designer and developer of this project I felt that the responsibility to deliver the project weighed a lot on my shoulders. In order to deliver this project I had to manage the outcome of each stage myself and ensure that each deliverable was met to a standard that I felt I could put my name beside. Reflecting on the whole project I now feel as though I was designing and implementing a product that represented I'm Hungry the brand and Sam Crowe the web developer.

The most difficult part about my role was understanding the rules and restrictions of Expression Engine. Integrating an application into EE takes time and consideration to understand the logic behind each add-on. When building an application this way it is not as simple as installing a plugin and it works immediately. There is precise documentation that needs to be followed to integrate an add-on. To get the best usage out of an add-on most likely that

add-on will require additional scripting to make it function the way that you want. Not all add-ons work together and they need to be integrated correctly to ensure they perform the task required.

I would recommend Expression Engine for any project that has a big budget. A wide range of add-ons have commercial licenses required to use them. The best thing about building an application with Expression Engine is that its much simpler to add functionality to a bespoke project. If I was to build this application again I wouldn't use EE. It would have taken half the time to use the google API and create a database as well as implementing custom code. A new method of development was used for this project and it did cause a lot of stress. The positive side to building with EE was that the project became commercially viable.

When the concept was created for this project there was no intention of furthering the development after the final deadline. Although during the project risk assessment surprise interest was shown from a stakeholder. A beta version of the project has been submitted to Bluegator Creative. This company deals with overseas clientele and they have showcased the product to a client in the United States. The company believed that this product could be advantageous as they have been looking to build a similar concept over the last 12 months.

The final statement for this project is that I believe the project was a success. There was a minimal financial cost to develop the project with some necessary Expression Engine add-ons. Although the functionality that was intended was delivered and a new method of development was learnt. Both of these factors provide a positive reflection on the experience that was had during the longevity of this project.

References:

- EllisLab. 2014. expressionengine. [ONLINE] Available at: <http://ellislab.com/expressionengine/>. [Accessed 18 April 14].
- Taylor Leopold, (2014), YI43jGrjSwi4bKJBOjkl_NZ14%20(265%20of%20589 [ONLINE]. Available at:[https://s3.amazonaws.com/oomf-com-files/YI43jGrjSwi4bKJBOjkl_NZ14%20\(265%20of%20589\).jpg](https://s3.amazonaws.com/oomf-com-files/YI43jGrjSwi4bKJBOjkl_NZ14%20(265%20of%20589).jpg)[Accessed 18 April 14].
- Bjorn Borresen. 2013. seolite. [ONLINE] Available at: <http://devot-ee.com/add-ons/seolite>. [Accessed 18 April 14].
- Buildwithstructure. 2013. Structure. [ONLINE] Available at: <http://buildwithstructure.com/documentation>. [Accessed 18 April 14].
- ee-zoo. 2013. zoo visitor. [ONLINE] Available at: <http://ee-zoo.com/add-ons/visitor>. [Accessed 18 April 14].
- Put your lights on. 2011. Developer. [ONLINE] Available at: <http://www.putyourlightson.net/developer>. [Accessed 18 April 14].
- Doug Avery. 2013. freebie. [ONLINE] Available at: <http://devot-ee.com/add-ons/freebie>. [Accessed 18 April 14].
- Pixel & Tonic. 2013. Matrix. [ONLINE] Available at: <http://devot-ee.com/add-ons/matrix>. [Accessed 18 April 14].
- boldminded. 2013. Snippet Sync. [ONLINE] Available at: <http://devot-ee.com/add-ons/snippet-sync>. [Accessed 18 April 14].
- Brett DeWoody. 2010. Hacksaw. [ONLINE] Available at: <http://devot-ee.com/add-ons/hacksaw>. [Accessed 18 April 14].
- Mark Croxton. 2010. Hacksaw. [ONLINE] Available at: <http://devot-ee.com/add-ons/ifelse>. [Accessed 18 April 14].
- Mark Croxton. 2013. Stash. [ONLINE] Available at: <http://devot-ee.com/add-ons/stash>. [Accessed 18 April 14].
- Objective HTML. 2013. Google Maps for Expression Engine. [ONLINE] Available at:<https://www.objectivehtml.com/google-maps/documentation/>. [Accessed 18 April 14].
- Solspace. 2013. Freeform. [ONLINE] Available at: <http://www.solspace.com/docs/freeform/>. [Accessed 18 April 14].
- Solspace. 2013. Favorites. [ONLINE] Available at: <http://www.solspace.com/docs/favorites/>. [Accessed 18 April 14].
- Responsable. 2012. ResponsableCSS. [ONLINE] Available at: <http://responsablecss.com/>. [Accessed 18 April 14].
- Less. 2012. Less. [ONLINE] Available at: <http://lesscss.org/>. [Accessed 18 April 14].

estebanav . 2013. geoposition.js. [ONLINE] Available at: <https://github.com/estebanav/javascript-mobile-desktop-geolocation>. [Accessed 18 April 14].

jQuery. 2013. jQuery. [ONLINE] Available at: <http://www.jquery.com>. [Accessed 18 April 14].

Malsup. 2012. form.js. [ONLINE] Available at: <http://malsup.com/jquery/form/>. [Accessed 18 April 14].

buckwilson. 2010. Lightbox me. [ONLINE] Available at: <http://buckwilson.me/lightboxme/>. [Accessed 18 April 14].

kswedberg. 2014. jQuery smooth scroll. [ONLINE] Available at: <https://github.com/kswedberg/jquery-smooth-scroll>. [Accessed 18 April 14].

Modernizr. 2013. Modernizr. [ONLINE] Available at: <http://modernizr.com/download/>. [Accessed 18 April 14].

imakewebthings . 2014. jQuery Waypoints. [ONLINE] Available at:<https://github.com/imakewebthings/jquery-waypoints>. [Accessed 18 April 14].

Webtogether-EE-Config. 2013. Abban Dunne. [ONLINE] Available at:<https://github.com/webtogether/Webtogether-EE-Config>. [Accessed 18 April 14].

teamwork. 2014. Sign Up. [ONLINE] Available at: <https://www.teamwork.com/>. [Accessed 18 April 14].

usabli.ca . 2013. intro.js. [ONLINE] Available at: <http://usablica.github.io/intro.js/>. [Accessed 18 April 14].

EragonJ. 2014. trip.js. [ONLINE] Available at: <http://eragonj.github.io/Trip.js/>. [Accessed 18 April 14].

David Walsh. 2012. Create a CSS Flipping Animation. [ONLINE] Available at:<http://davidwalsh.name/demo/css-flip.php>. [Accessed 18 April 14].

Survey Monkey. 2014. Survey Monkey. [ONLINE] Available at: <https://www.surveymonkey.com/>. [Accessed 18 April 14].

Pixeden. 2012. iPhone 5 Psd Vector Mockup. [ONLINE] Available at: <http://www.pixeden.com/psd-mock-up-templates/iphone-5-psd-vector-mockup>. [Accessed 18 April 14].

Derek Hogue - Amphibian Design. 2013. Deviant. [ONLINE] Available at: <http://devot-ee.com/add-ons/deviant>. [Accessed 18 April 14].

Causing Effect. 2014. Template Variables. [ONLINE] Available at: <http://devot-ee.com/add-ons/template-variables>. [Accessed 18 April 14].

Max Lazar. 2012. MX Google Maps. [ONLINE] Available at: <http://www.eec.ms/add-ons/mx-google-map>. [Accessed 18 April 14].

NSM Morphine. 2012. NSM Morphine Theme. [ONLINE] Available at: <http://ee-garage.com/nsm-morphine>. [Accessed 18 April 14].

Appendices

1.1 Idea generation



The mind map that can be seen in appendix 1.1 shows a list of relative words that were related to the word hungry. These words were generated by 6 different users that had no previous connection to the project.

1.2 Requirement types and Priority levels

The requirement types and priority levels have been set to help gain a rationale for the range of requirements for this application. The requirement types and priority levels can be seen below.

Requirement Type 1 : Functional

Requirement Type 2 : Look & Feel

Requirement Type 3 : Usability

Requirement Type 4 : Performance

Requirement Type 5 : Maintainability

Requirement Type 6 : Security

Requirement Type 7 : Constraints

Priority Level 1 : Lowest

Priority Level 2 : Medium-Low

Priority Level 3 : Medium

Priority Level 4 : Medium-High

Priority Level 5 : Highest

1.3 - Requirement specifications

There has been 19 requirements stated for this application. The requirements range from concepts such functional classes like allowing users to register to creating look & feel classes that will promote brand awareness. A full list of requirements can be seen below.

ID	Type	Description	Priority	Dependancies
1	Functional	To create a web application that holds information and locations of Belfast Take-aways.	5	All
2	Functional	Using an API to aid the search function to find restaurants nearest to your location.	5	#007
3	Functional	Implementing the input Range element, to filter menu prices	4	#018
4	Functional	Users can register and login to save favourite businesses	4	#019
5	Maintainability	Developing is made easier with php Framework CodeIgniter, this will power the chosen CMS Expression Engine, this will allow clients to manage their content easily.	5	
6	Look & Feel	The about section will be placed on the homepage, this will reduce page to page navigation.	3	#001, #008

ID	Type	Description	Priority	Dependancies
7	Functional	Search by name of the business to find a match or similarly spelt result, displaying information on a results page.	3	#019
8	Usability	Design the application to provide simple navigation throughout the site to allow users to browse without complications.	3	#001, #008, #013
9	Maintainability	An email address needs set up for support and an admin user, allowing management of the application.	4	none
10	Performance	Setting up multiple environments to allow local and production environments for the database.	2	#019
11	Look & Feel	Enlisting the Marketing Executive from the Creativity Hub to help steer the product towards the correct target audience	1	#012
12	Look & Feel	A logo or tagline that represents the brand relatively which will help make the application memorable.	3	
13	Performance	Ensuring the product is available for use 24 hours per day, 365 days per year. The product needs to have no broken design elements and all links working.	4	#015, #001
14	Constraints	User registers and receives a message thanking them for registration as well as a reminder of their username and password.	4	#019
15	Constraints	Users may view this application on various browsers and operating systems, the development team needs to consider usability across all platforms.	4	#001, #008
16	Usability	All information used to populate the database needs to lawful and not infringe on data protection of the businesses involved.	4	none

ID	Type	Description	Priority	Dependancies
17	Performance	To ensure pages load quickly and content file size is minimised, to produce a faster and better experience when viewing the application.	3	#019
18	Usability	Each Business selected to be in the database needs to have menu items associated with it, allowing for these items to be searched.	2	#003
19	Performance	To build a sandbox of tools that will enable EE to perform the heavy lifting of back-end coding	5	none

1.4 - Just Eat email

This email was received from Just Eat, the email reflects an attempt to gain inside information into how this business operates. The intention was to find out how business menu were entered into the database for this company.

Just-Eat UK Info info@just-eat.co.uk 6 months ago 

Good Evening Samuel,

Thanks for Emailing Info@

We can inform you that this is done through a separate department who handle this. If you take a screenshot picture of the menu that you have at the moment we will transfer it your online menu on our website.

If you are interested in becoming a member of JUST EAT give us a Call on 0844 243 7777 (Until 2AM daily)

Kind Regards

Marzuq W

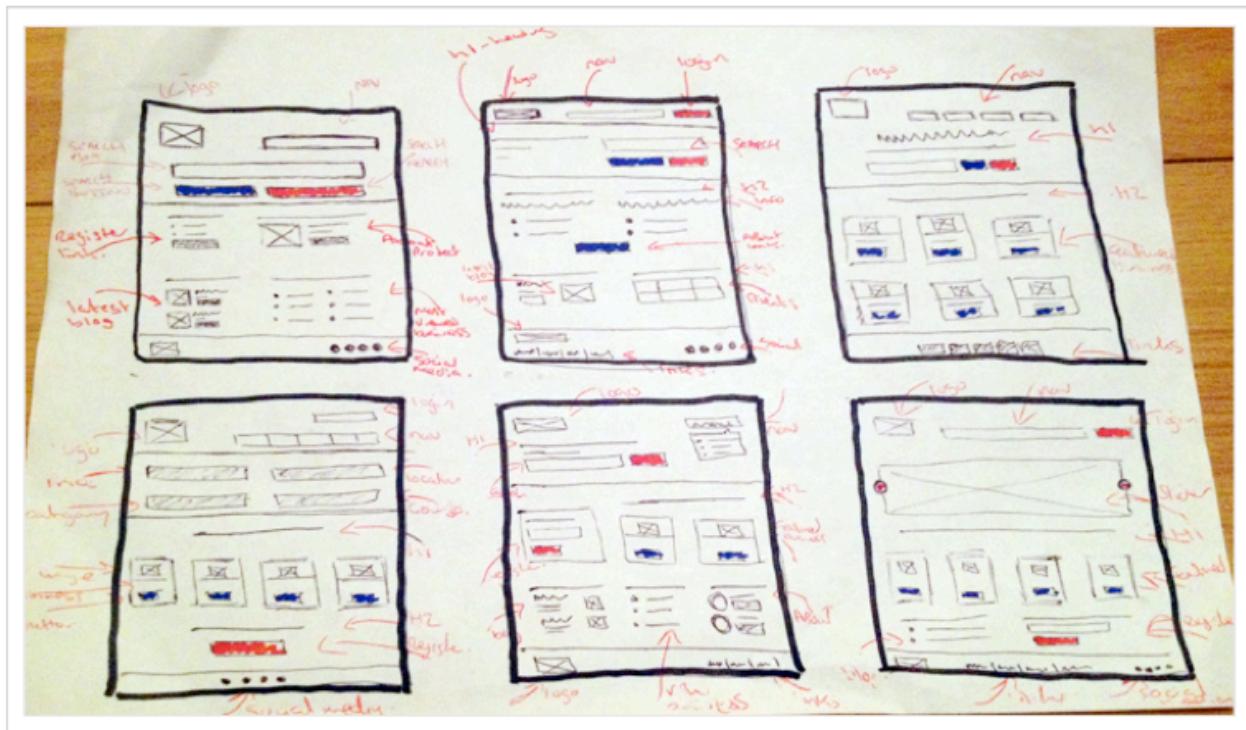
Customer Care Team & Anti-Cooking Activist

1.5 – 6-Up paper prototype

The 6-Up prototype was created to map out the functionality of the home page. The concept behind the 6-Up was to show how user will perceive this page when they land on it. The results from this prototype, mapped out the layout for the homepage and plotted potential interactive areas of the page that will enrich the user experience. The sketches allowed for reflection in regards to the

priority of content and helped make use of the requirements specifications. When considering the best solution for content and layout, articles on user experience were researched.

The realisation taken from this prototype was to enhance the importance of the search features on the home page. The search functionality is the core of the application and this feature applies the most memorable interactivity in the application.

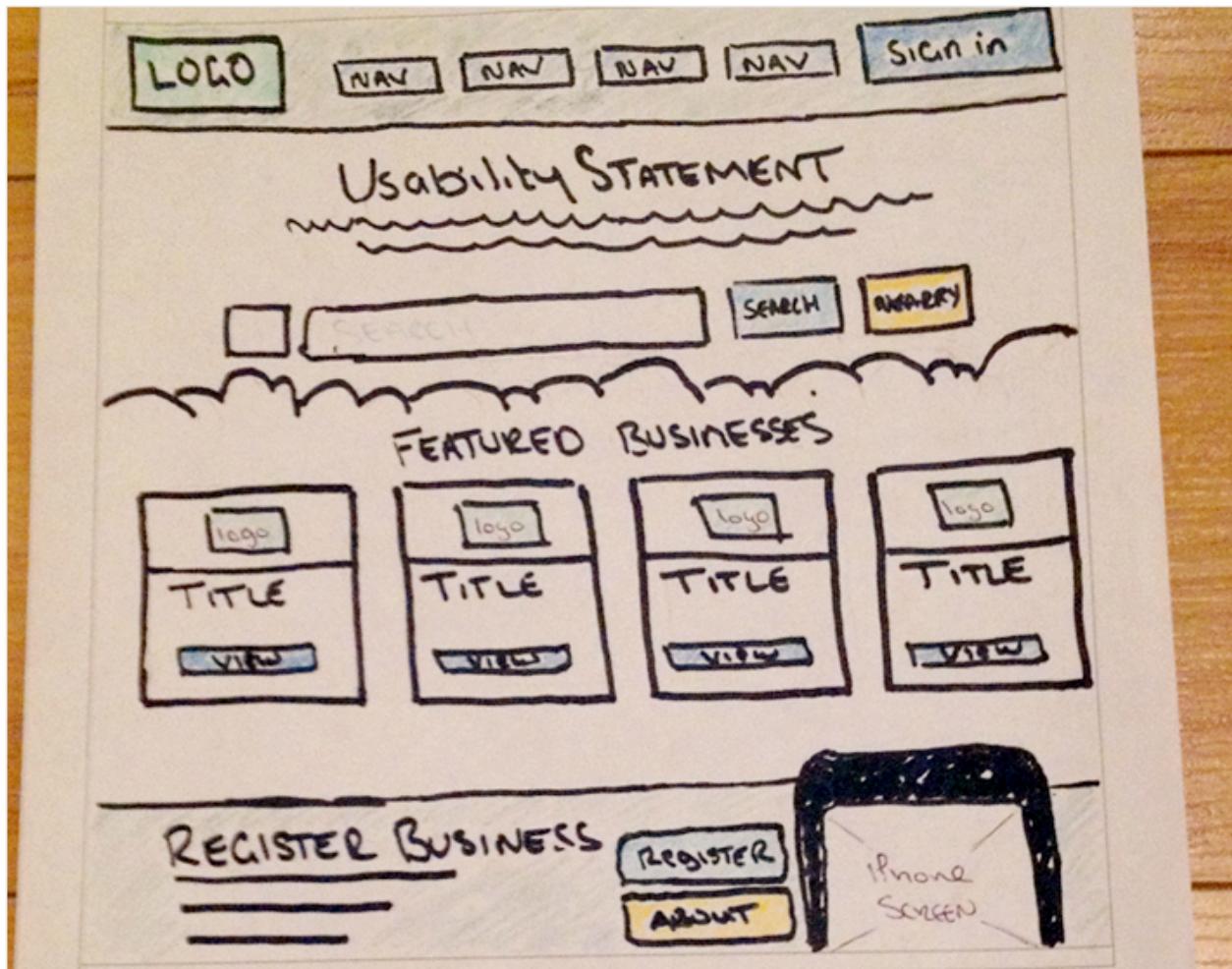


1.6 - 1-up paper prototype

The 1-up prototype process was used to piece together the best concepts of the previous sketches and help generate a more rounded and functional wireframe.

The final prototype wireframe includes a logo, navigation and login button in the header area. By grouping these items together it would allow for navigation and brand recognition throughout each user-experience. The search area will be introduced with a tagline indicating what to do when arriving on the site. Each search method can be clearly distinguished due to the difference of colour, highlighting the buttons.

The homepage will include four featured businesses. The businesses displayed will be the highest rated businesses in the directory. Users will be able to view information about businesses by viewing the front of each entry. If a user wants more detail about a business they can click view more, this event will flip the div 180degrees and further information will be presented on the back of the entry .

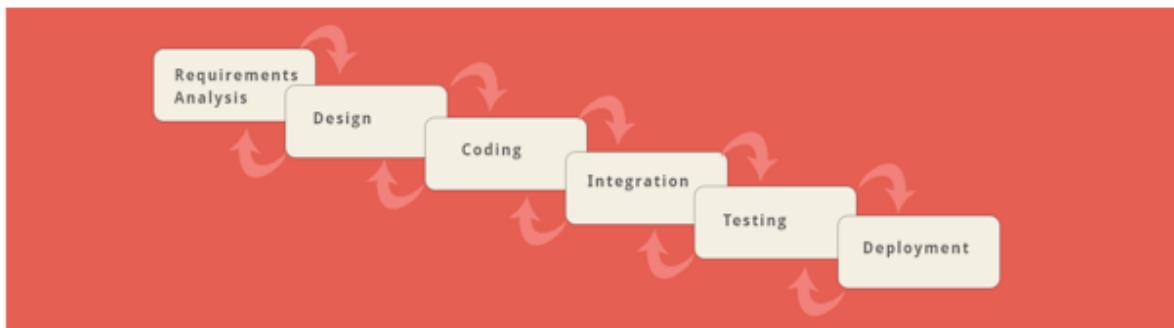


1.7 - Survey Monkey - Desirable features survey

It was decided that to gain a user perspective on what features would benefit this project that a survey would be created to generate results. The survey was entitled - 'Desirable Features - Restaurant Directory'. The survey can be taken from the following domain: <https://www.surveymonkey.com/s/W6WKJ6P> .

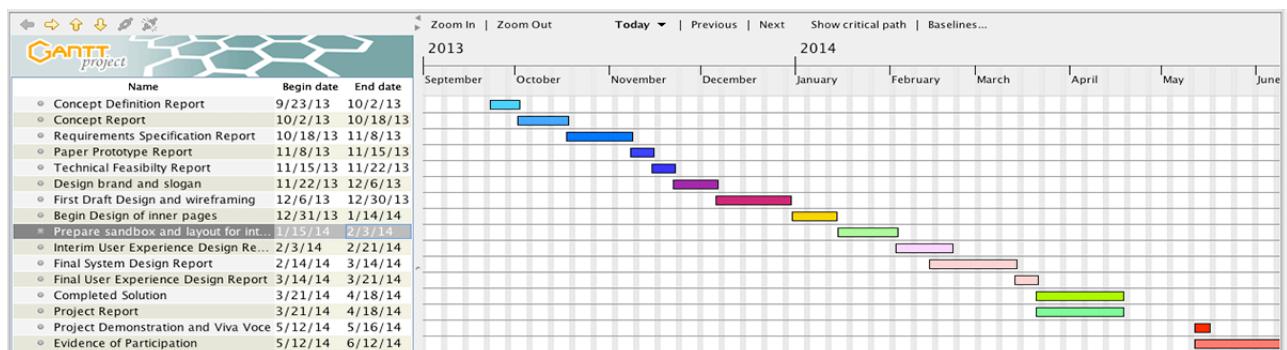
1.8 - waterfall methodology

The diagram below shows the rigid model that is represented when working with the waterfall methodology. The six phases are completed one after the other.

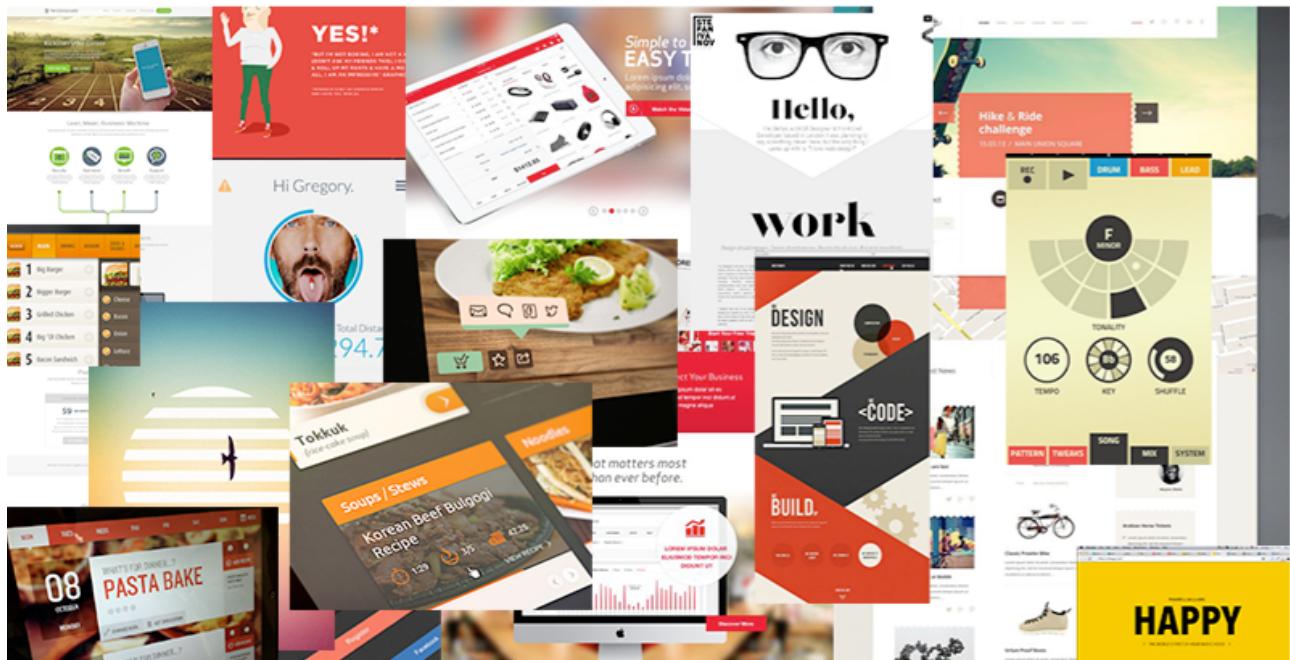
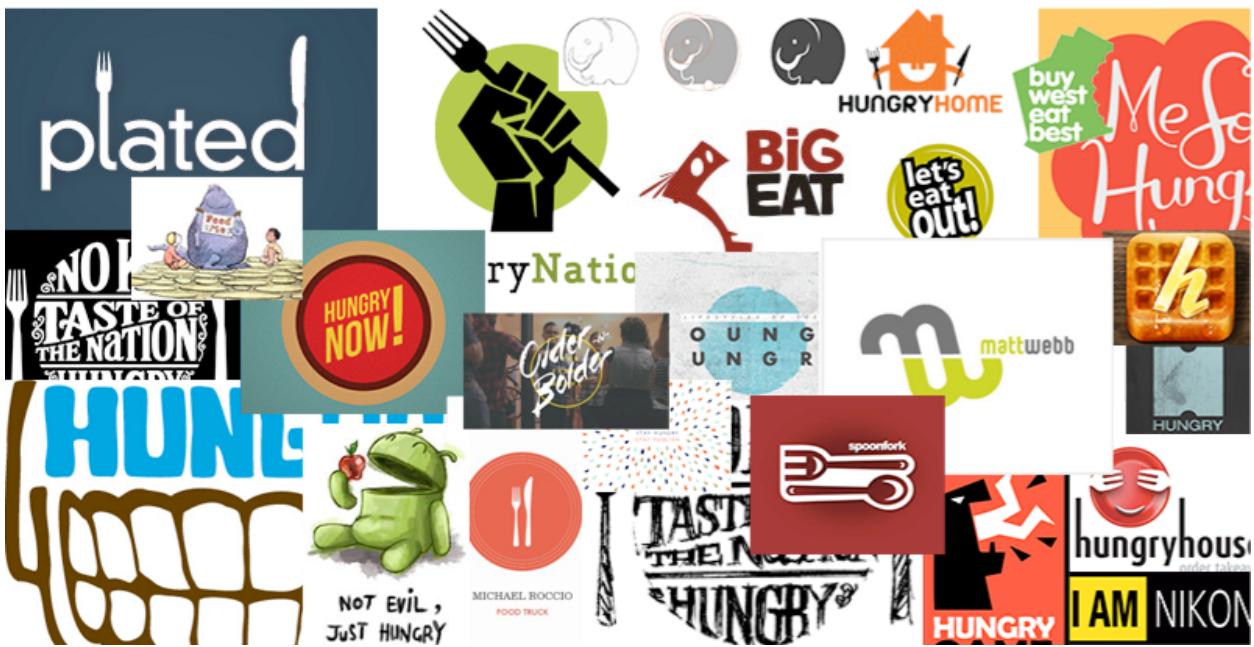


1.9 - Gantt chart volume 1

The gantt chart that was created for this early projection of the project based the milestones and phases against the university deliverables. This gannt chart is a high-level gannt chart, therefore individual detailed tasks have yet to be confirmed.



2.1 – Mood boards for branding and web



2.2 – Competitor research

This is the three competitor logos for Just-eat, Hungry House and Nifty Nosh. The colour palettes use a similar format, representing warm colours.



2.3 - Colour palette and font book

The product doesn't directly sell food online and it was felt that this was an opportunity to pull away from the related businesses. It was felt that going against the grain and using a cold colour scheme would make the project unique. To create a complimentary colour scheme the Adobe application Kuler was used. Each colour scheme generates 5 colours that work together and will allow begins the process for creating a swatch.



This font is a google font, it has variations of the font that would allow for it to be used as heading or paragraph tags. Below is font book sample to demonstrate it.



2.4 - Branding sketches

When creating these sketches the word association of a knife, fork and a plate were conceived to represent hunger. The sketches included show various ways

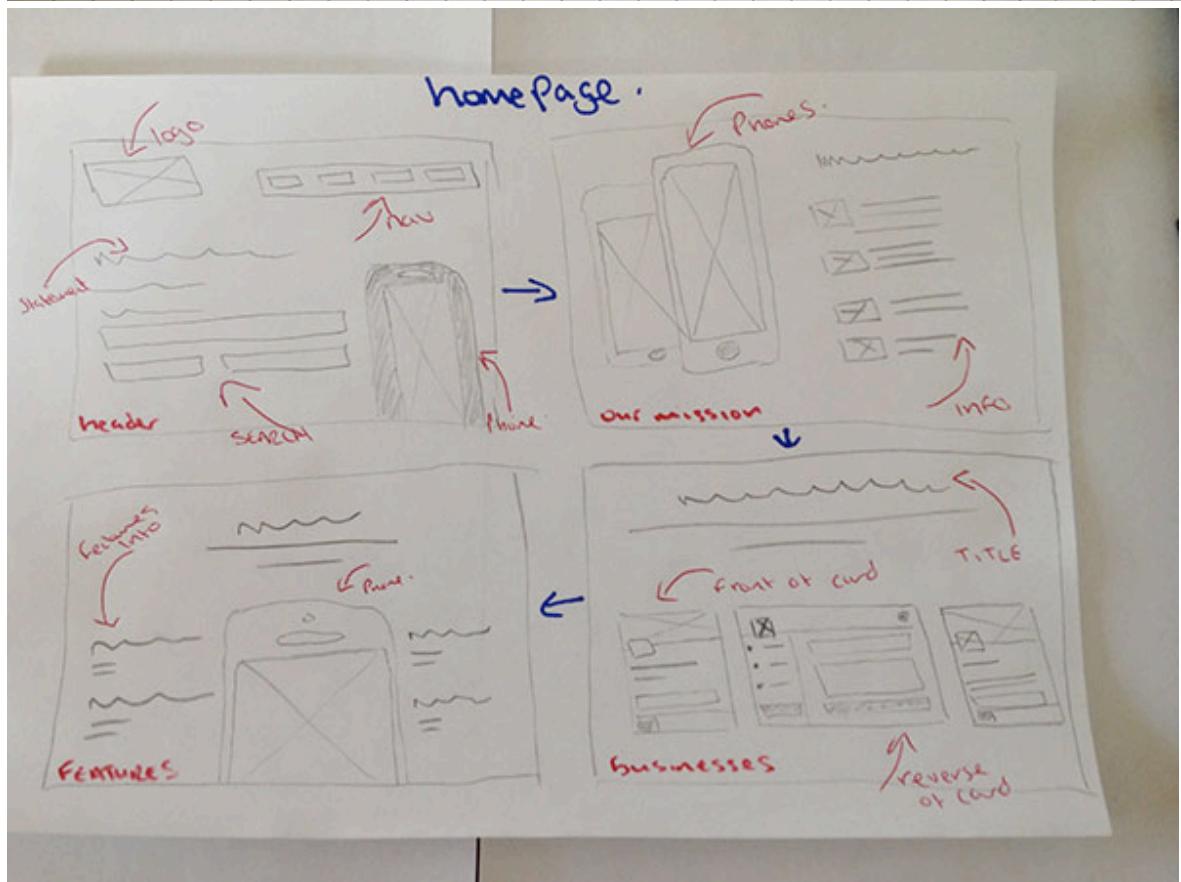
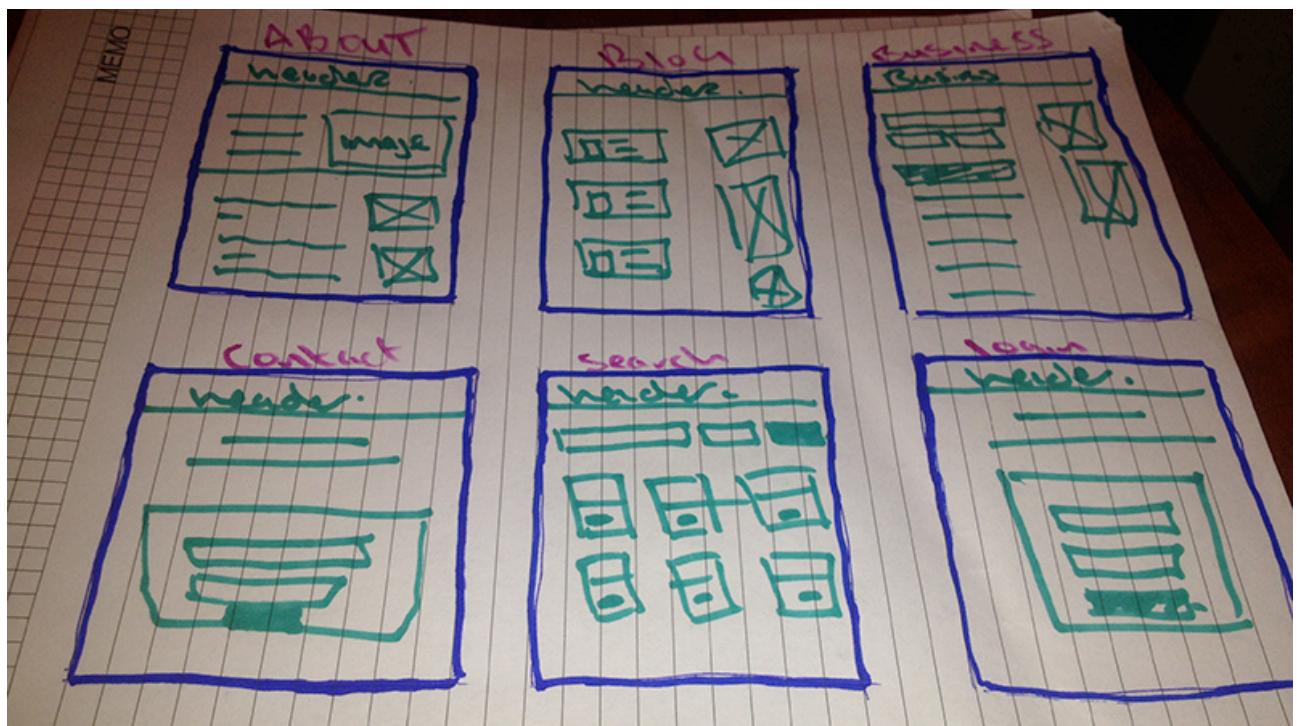
of implementing a fork into various sketched type faces. These designs were part of the original concept that was disregarded.

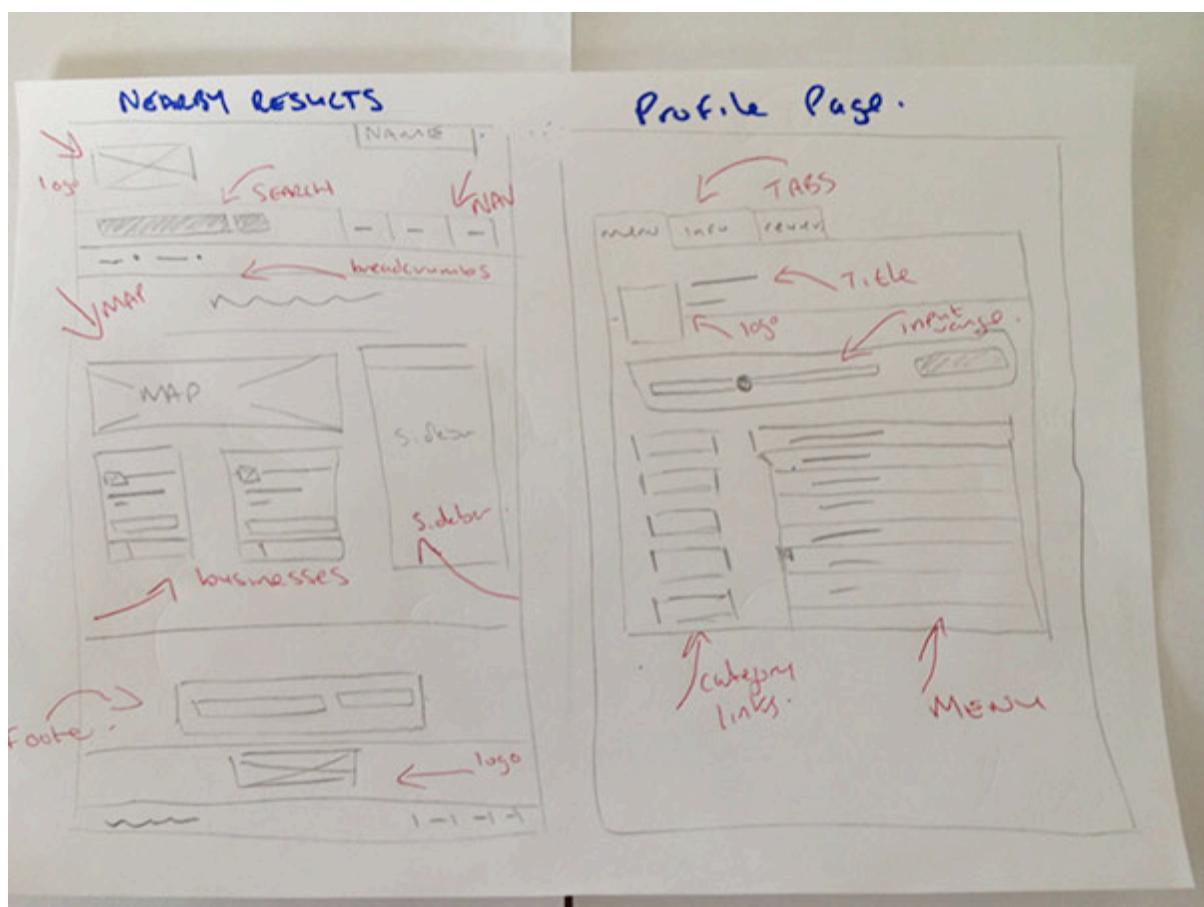
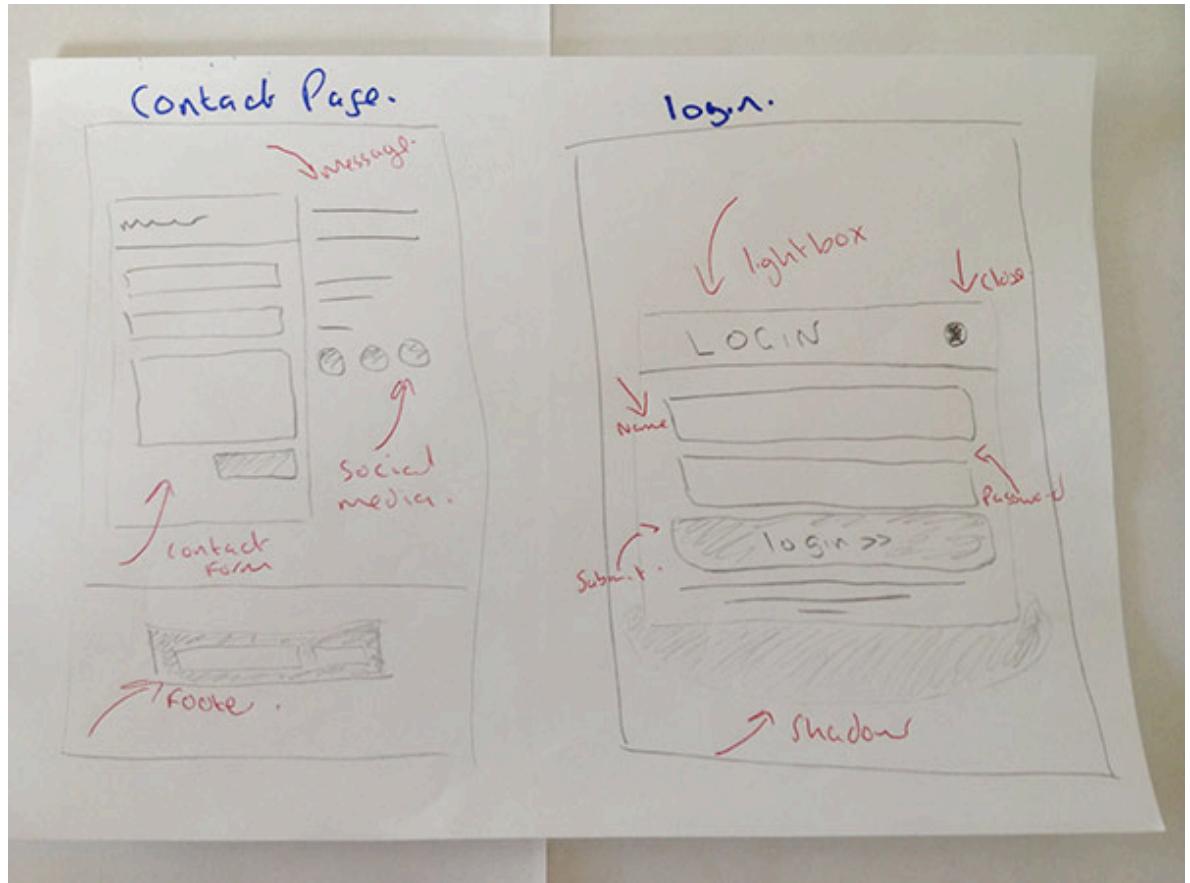


It was decided that location and hunger were the two main factors behind the ideology of the brand. To bring these concepts together experimentation with location icons were integrated into the design. It was decided that using a knife and fork inside the pin with the negative space to represent a plate would represent hunger.

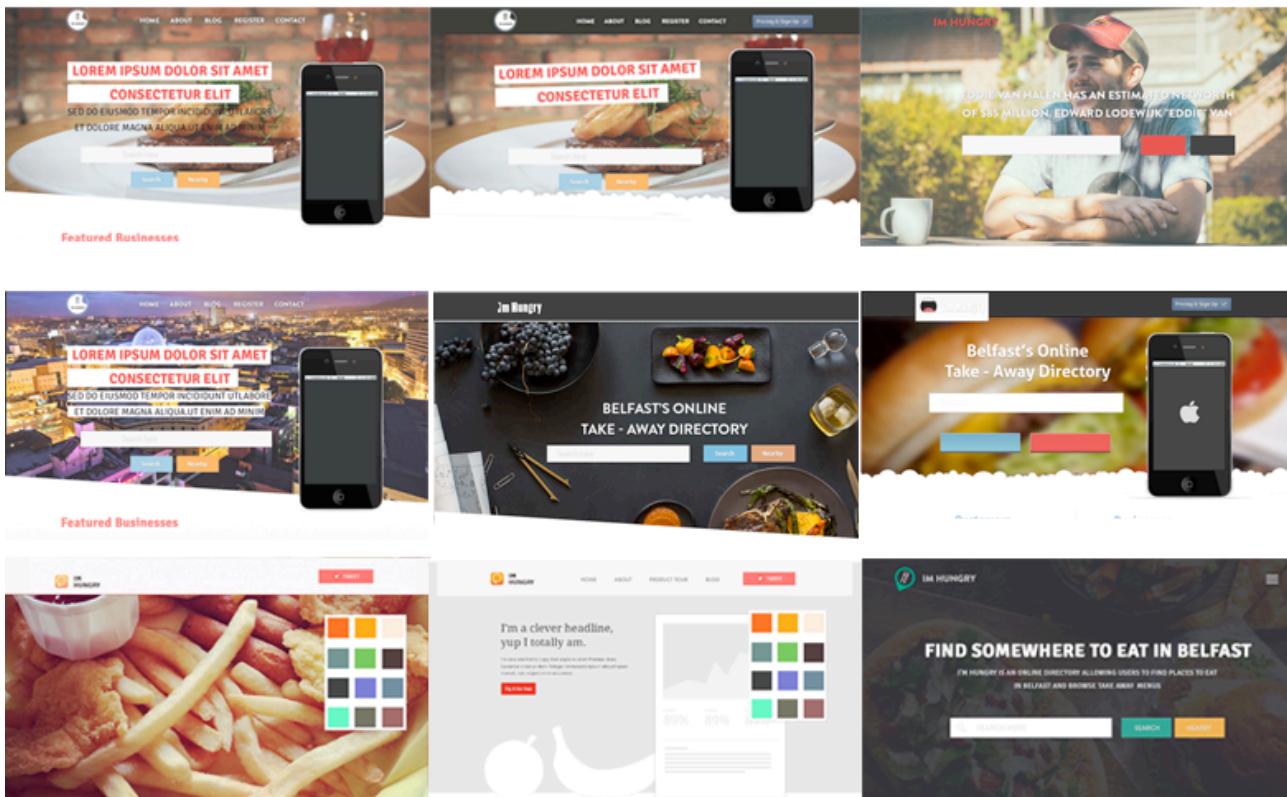


2.5 - Web page wireframes



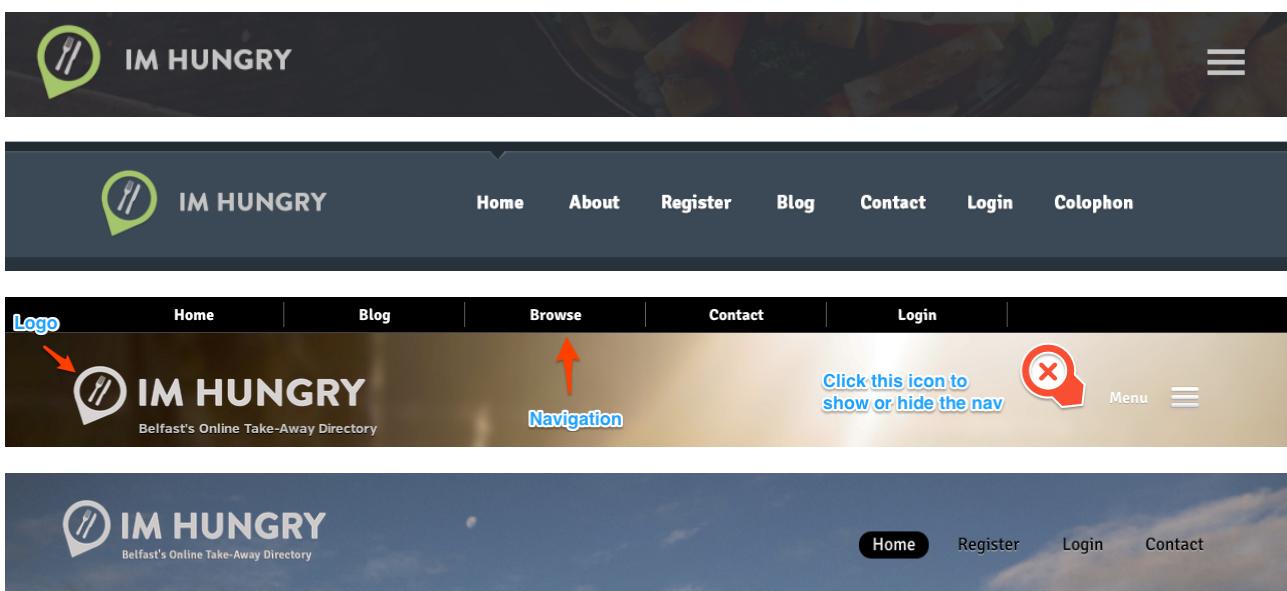


Web page creative experimentation

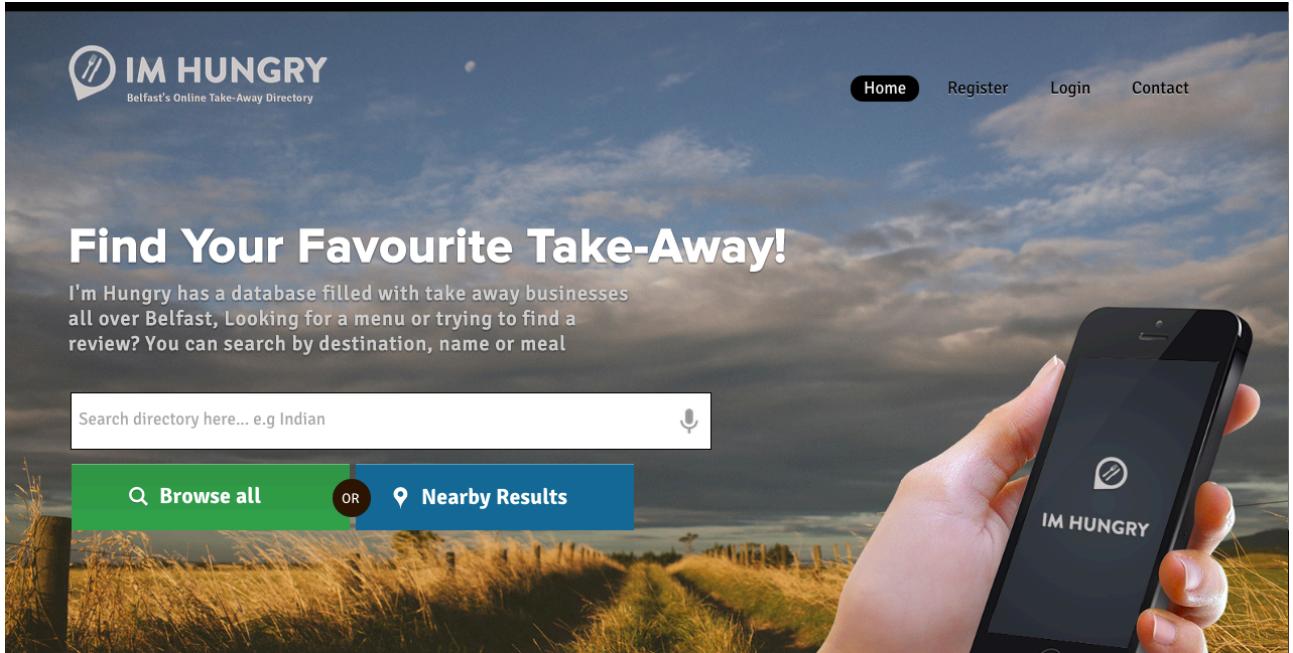


2.6 – navigation choices

The navigation went through a lot of design experimentation. The reason behind this constant changes was that usability of the application was becoming less apparent until a horizontal navigation was implemented.



2.7 - Home page header



2.8 - Our mission section

A screenshot of the IM HUNGRY mobile application. Two phones are shown side-by-side: a white iPhone on the left and a black iPhone on the right. Both phones display the same app interface. The screen shows a search bar with the placeholder 'Search Restaurants..', a green 'Search' button, and a blue 'Nearby' button. Above the search bar, there is a banner with the text 'FIND YOUR FAVOURITE TAKE-AWAY'. Below the search bar, there is a list of restaurant results, with the first one being 'Guiseppe's burger bar' located at '14 york road'. A 'Read More' button is visible next to the result. To the right of the phones, there is a section titled 'Our Mission' with a sub-section titled 'Give Customers' featuring a speech bubble icon and text about letting businesses know user preferences. Another sub-section titled 'Give businesses' features a thumbs-up icon and text about providing an online presence and early warning signs.

- Application features section

Features

I'm Hungry allows users to locate local take-aways, browse their menus and read and write reviews

Sign Up

Users can sign up to this application and favourite the businesses that they like the most

Feedback for business

A way of finding out what's going well and spark new ideas for innovation

Browse Menu's

Each business in the directory has a full menu available to browse, allowing users to filter through items

Find Food

Users can use the search nearby feature and find the closest restaurants to their location

2.9 – Global Footer

Business Sign Up

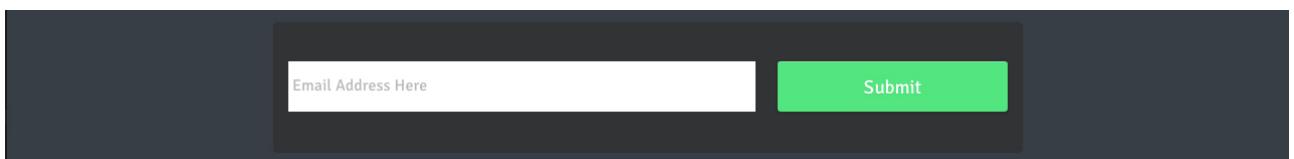
Interested in adding your business to the I'm Hungry directory? Add your email address here for an information pack

Submit

Belfast's Online Take-Away Directory

Copyright © 2014 I'm Hungry. All Rights Reserved.
Home
Contact
Colophon
Sitemap

The footer has dark shades of navy as the main colour, to highlight the functionality it was decided to use a bright green colour as the hover for the button. This bright colour almost gives of the effect of lighting up due to the juxtaposition of opposing colours.



2.10 - Form Design (login/registration)

The image shows two overlapping modal windows. The left window is titled 'Login' and contains fields for 'Email' and 'Password', with a 'Login' button at the bottom. The right window is titled 'Register' and contains fields for 'Name', 'Email', and 'Enter Password', with a 'Register' button at the bottom. Both windows have a message at the bottom: 'Signin now to find your favourite take-away businesses or why not search [Nearby](#) and see what businesses are near you?' and 'Register now to favourite your favourite take-away businesses or why not search [Nearby](#) and see what businesses are near you?'. The background of the application interface is visible through the windows.

2.11 - Profile page

The profile page for 'Zeera' restaurant. At the top, there are tabs for 'Menu', 'Information', and 'Reviews'. Below the tabs, the restaurant's logo and name ('Zeera') are displayed, along with its address ('280 Ormeau Road, Ormeau Belfast BT7 2GB'). A rating bar indicates a rating of £10. To the right, a sidebar titled 'Nearby Businesses' lists several restaurants: Ramzy's, Pizza Pro, Beatties, Test, Tacomex, Ming's Garden, Fon-A-Kebab, and Little Caesar. On the left, a 'Menu' section shows categories like Starters, Balti-Biryani-Dishes, Tandoori-Dishes, Chefs-Specials, Meat-Curries, and Seafood-Vegetable-Curries. The 'Starters' category is expanded, showing items like 'Zeera Special Mixed Platter' (4.40), 'Veggie Pakora' (2.55), 'Onion Bhaji' (2.55), 'Chicken Pakora' (2.95), and 'Seekh Kebab' (2.55). The 'Balti-Biryani-Dishes' category is also shown, with 'Chicken Balti' listed at 7.25.

Category	Item	Price
Starters	Zeera Special Mixed Platter	4.40
	Veggie Pakora	2.55
	Onion Bhaji	2.55
	Chicken Pakora	2.95
	Seekh Kebab	2.55
Balti-Biryani-Dishes		
	Chicken Balti	7.25

2.12 - Sidebar

Font icon pack Fontello has been used to create the heart icon for this design. Fontello has been used for all icons that can be seen in this application. The design shows the top 5 rated businesses in the application.

Top rated businesses		
1	Mario's..	♥ (7)
2	Wok-A-Moley	♥ (7)
3	Zeera	♥ (6)
4	Bengal Brasserie	♥ (5)
5	Shimla Indian..	♥ (3)

3.1 - Expression Engine Specification

There are a few key elements that need to be understood when developing an Expression Engine application. One thing that should be noted is that although EE uses third party add-ons to produce functionality, those add-ons cannot simply be installed and then produce functionality. Each add-on requires integration and understanding as they need to be implemented into templates and loops to generate dynamic code.

The tools that come in the box with this CMS are called Add-Ons. Each Add-On has its own specific set of tags that allow the Add-On to be linked to the dynamic content. The Add-Ons are used to loop, pull or integrate database information into the application. The add-on performs the same duty as an API, allowing the content from the database to be accessed and updated through a specific set of tags relative to each add-on.

It should be noted that an Expression Engine add-on would not be similar to a Wordpress widget. The difference being that widgets in Wordpress generally work without implementation, where as Expression Engine add-ons need to be installed, integrated into the code complying with the restrictions of the add-on

and most likely will require additional scripting to make them work the specifically the way the developer requires.

- **Channels** - The Channel module allows developers to display content that has been created using the publish section, this is found in the control panel. This module is the most important module in EE
- **Channel:entries** - The channel entries tag shows the content created by publishing a channel. This would be the main tag in EE. This tag allows developers to retrieve and display content.
- **Channel fields** - Any fields that have been assigned to a channel can be accessed by giving them a name without spaces in it.
- **Templates** - Templates contain the majority of the information or content you want to display in the front-end of the application. When EE calls a Template, the system renders and processes the template to parse tags, variables, snippets and embeds within the template that has been called.
- **Snippets** - This is reusable tags or template parts. Snippets are useful for making changes across the site with in one call. Similar to a php include.
- **Embeds** - Expression Engine templates can be embedded within each other, this is useful creating a global header/footer and embed this into another template that has any kind of content.
- **Conditionals** - This is a type of variables that allows you to conditionally show information depending on whether or not the condition is met. This would replicate an IF statement
- **File** - File is a module that allows any information that is associated with the file uploads to be displayed
- **Field type** - Field types are used for handling all varieties of information or data that can be stored in the channel modules.

3.2 - *Plugin scripts*

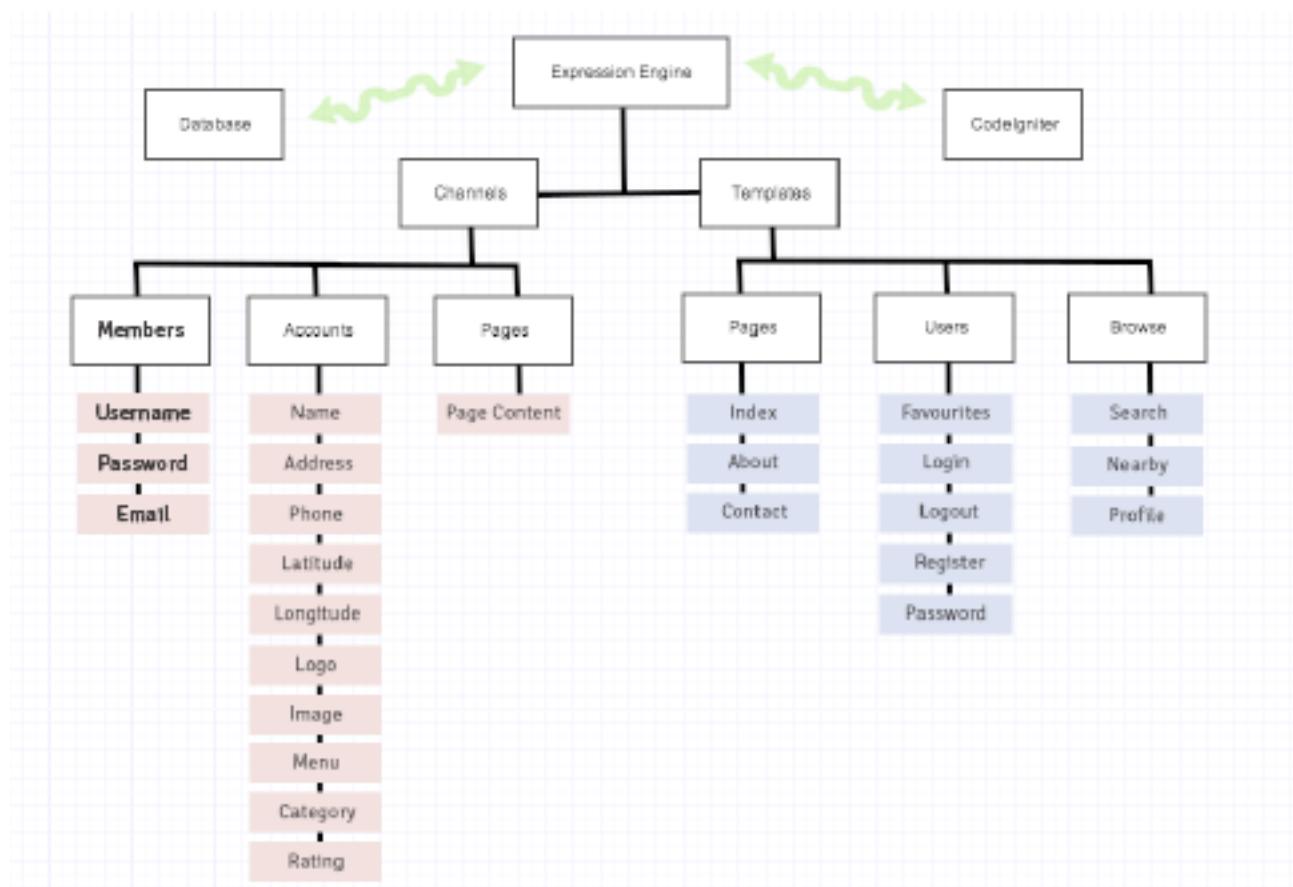
Plugin name	Usage	Resource
Geoposition.js	Uses google maps geolocation to find the position of the user during the nearby results feature	https://github.com/estebanav/javascript-mobile-desktop-geolocation

Plugin name	Usage	Resource
jQuery.min.js	jQuery library to aid the javascript functionality of the application	http://www.jquery.com
jquery-ui-1.10.4.custom.js	Custom UI, will implement tabs in the profile page	http://www.jquery.com
jquery.form.js	Will be used to aid AJAX forms with EE	http://malsup.com/jquery/form/
jquery.lightbox_me.js	Will be used to place forms in a lightbox window	http://buckwilson.me/lightboxme/
jquery.smooth-scroll.js	This will provide smooth scroll to navigated areas of the profile page	https://github.com/kswedberg/jquery-smooth-scroll
modernizr-2.5.3.min.js	Implemented for legacy browser support	http://modernizr.com/download/
waypoints.min.js	Used on homepage for executing a function when an element is scroll to.	https://github.com/imakewebthings/jquery-waypoints/
myscript.js	Combined with geoposition.js and jquery.lightbox_me.js. This also includes all custom scripts for the application	
wt-config.php	This will be used to connect to multiple environments	https://github.com/webtogether/Webtogether-EE-Config

3.3 - Logic design - via google docs

The logic design for this application has been set up to plan the build of the EE functionality. The EE functionality that needs to be determined is which sitemap, channels and templates need to be built to make the EE integration successful. By planning a logical design the add-ons that are required to make this product functional can be determined. This template has been created in Google Drive and made public at this address – <https://docs.google.com/spreadsheet/ccc?key=oAtKNnxXOhi76dC1HQWhmLUdmM3FzVG9hVoNIRjZodmc&usp=sharing>

3.4 - Low-level Database relationship diagram



3.5 - Table of contents for EE Add-ons

Add-on name	In the box	Third Party	Resource
Comment	yes	no	http://ellislab.com/ expressionengine
Favorites	no	yes	http:// www.solspace.com /docs/favorites/
File	yes	no	http://ellislab.com/ expressionengine
Freeform	no	yes	http:// www.solspace.com /account/login/software/license/freeform/free/
Google Maps for Expression Engine	No	yes	https:// www.objectivehtml .com/google-maps/ documentation/

Add-on name	In the box	Third Party	Resource
Search	yes	no	http://ellislab.com/expressionengine
seolite	no	yes	http://devotee.com/add-ons/seolite
structure	no	no	http://buildwithstructure.com/documentation
Zoo visitor	no	yes	http://ee-zoo.com/add-ons/visitor
Freebie	no	yes	http://devotee.com/add-ons/freebie
Matrix	no	yes	http://devotee.com/add-ons/matrix
Snippet sync	no	yes	http://devotee.com/add-ons/snippet-sync
EE hive hacksaw	no	yes	http://devotee.com/add-ons/hacksaw
IfElse	no	yes	http://devotee.com/add-ons/ifelse
Stash	no	yes	http://devotee.com/add-ons/stash
File	yes	no	http://ellislab.com/expressionengine
Developer	no	yes	http://www.putyourlightson.net/developer
Deviant	no	yes	http://devotee.com/add-ons/deviant
Template Variables	no	yes	http://devotee.com/add-ons/template-variables

Add-on name	In the box	Third Party	Resource
MX google Maps	no	yes	http://www.eec.ms/ add-ons/mx- google-map
NSM Morphine	no	yes	http://ee- garage.com/nsm- morphine.

3.7 - EE Add-on usage

Favorites - This add on allows members to save favourites that have been displayed through channel entries. An xml file will save the information and link it to a members profile.

Freeform - This module allows you to create simple forms for EE website. The forms can be managed through the CMS

Google Maps for Expression Engine - Enables the Google map API.

SeoLite - This ensures that the title-tag is placed on all pages on the application, uses an embed or snippet

Structure - This is used to create pages, navigation, manage content through an interface.

Zoo-Visitor - Allows for control over site members, aids with registration and login

Freebie - This allows developers to use the segments of a URL to pull information from EE

Matrix - Allows for content to be grouped together in columns, creating a link between content. Good for content that was be in table format

Snippet Sync - Allows the management of snippets in one place, grouping them together

EE Hivehacksaw - Strips HTML and allows for excerpts or a character count to be added to content

Stash - Allows for content to be stashed for reuse, good for managing template content

3.8 – Templates structure

```

▼ templates
  ▼ default_site
    ▼ ajax.group
      register-login.html
      user-login.html
      user-register.html
    ▼ browse.group
      index.html
      nearby-results.html
      profile.html
      search.html
    ▼ pages.group
      contact.html
      index.html
      sitemap.html
      standard.html
    ▼ usercp.group
      favourites.html
      logout.html
      my-favourites-add.xml
      my-favourites.html
    ▼ wrappers.group
      .home.html
      .inside-standard.html
      .inside.html

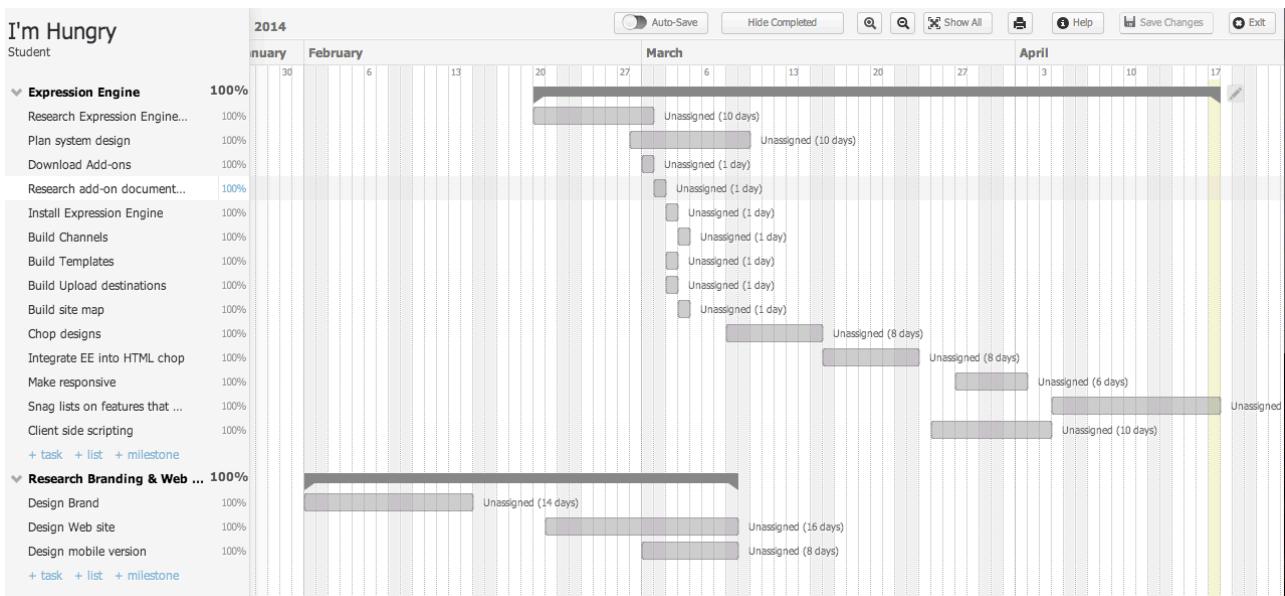
```

```

877
878
879
880  /*
881   * = Menu Filter
882   * =====
883
884
885  $(function(){
886    $(".discourse-menu").on("click", ".menu-item", function(e) {
887      e.preventDefault();
888      var filter = $(this).attr("data-filter");
889      // Loop through each item in the menu
890      $(".menu-item").each(function() {
891        if ($(this).attr("data-filter") == filter) {
892          $(this).addClass("selected");
893        } else {
894          $(this).removeClass("selected");
895        }
896      });
897    });
898  });
899 });
900 // Display the number of items
901 var num_items = $("#file-count").text();
902 $("#file-count").text(num_items);
903 });
904 });
905 //Reset
906 $('#reset').click(function() {
907   $(".menu-item").removeClass("selected");
908   $("#file-count").text("0");
909 });

```

3.9 –Gannt Chart volume 2



3.10 – User guide for EE

To gain access to the admin side of the application, login with the details provided on page 62 of the report. Upon login the following screen will be

displayed. This screen is the home page of the control panel.

The screenshot shows the Control Panel home page with a navigation bar at the top. The main area is divided into three sections: 'Create' (with options for Entry, Template, Template Group, and Channel), 'Modify or delete' (with options for Entry, Template Group, Most Recent Entry, and Pending Comments), and 'View' (with options for Site, Recent Comments, Recent Entries, and User Guide). A sidebar on the right includes links for Site, Recent Comments, Recent Entries, and User Guide.

To publish a new account navigate to content > publish > accounts. This will direct the user to the publish page for accounts. The fields are clearly labelled and it will be easy to publish a new account from here. After a new account has been published, the new account can be searched and found by a user.

To publish a new channel, template, custom field or upload directory navigate to the developer tab. The range of developer tools can be found here.

The screenshot shows the Control Panel developer tab with a dropdown menu open. The menu includes options such as Templates, Custom Fields, Channels, Categories, Member Groups, Upload Destinations, Modules, Admin, Tools, and General Configuration. Other tabs like Health Check, Structure, and Template Variables are visible at the bottom.

To create or view any global variables in the application navigate to the variables tab and a full list of global variables can be found here. The importance of global variables is to make all text content of the application that is not linked to a channel editable for the admin. This allows for full control of all content.

To view all comments that have been submitted to via the review inputs in the front end of the application navigate to Recent comments.

To monitor all emails that have been sent to the admin of the application navigate to forms. This will clearly have the different forms labelled, identifying which contact form they came from.

The screenshot shows the 'Forms' section of the CP Home. At the top, there are tabs for 'Forms', 'Fields', 'Field types', 'Notifications', 'Utilities', 'Preferences', and 'Help'. Below the tabs, there's a 'New Form' button. A table lists two forms: 'Contact' (ID 1, 55 submissions, Moderate 2) and 'feedback' (ID 2, 11 submissions, Moderate 1). Each row has edit, duplicate, and delete buttons.

Form	Submissions	Moderate	ID	Composer	Edit	Duplicate	Delete
Contact contact	Submissions 55	Moderate 2	1	N/A			
feedback feedback	Submissions 11	Moderate 1	2	N/A			

3.11 - Review ajax script

```

4      $(document).ready(function(){
5          $(function(){
6              $('.back_comment').review_input.gif').hide();
7              var options = {
8
9                  beforeSubmit: showRequest, // pre-submit callback
10                 success:     showResponse // post-submit callback
11             };
12             // bind form using 'ajaxForm'
13             $('.review_submit').ajaxForm(options);
14
15             // pre-submit callback
16             function showRequest(options) {
17                 $('.back_comment').review_input.gif').show();
18                 $('.review_name').val('');
19                 $('.mycomment').val('');
20                 return true;
21             }
22
23             function showResponse() {
24                 $('.back_comment').review_input.gif').hide();
25                 alert('Thanks, your review has been submitted!')
26             }
27         });
28     });
29 });

```

3.12 - Login/register EE code

```

{exp:zoo_visitor:registration_form id="favouriteRegisterForm" allowed_groups="5" json="yes" include_jquery="no"}

<div class="form_elements">

    <div class="form_line reg_info addPadding1">
        <div class="form_input">
            <input id="username" name="member_firstname" type="text" placeholder="Name"/>
        </div>
    </div>

    <div class="form_line reg_info addPadding2">
        <div class="form_input">
            <input id="username" name="username" type="text" placeholder="Email"/>
        </div>
    </div>

    <div class="form_line">
        <div class="form_input">
            <input type="hidden" value="5" name="group_id">
            <input id="accept_terms" name="accept_terms" type="hidden" value="1" />
            <input id="password" name="password" type="password" placeholder="Enter Password"/>
        </div>
    </div>

    <div class="form_line">
        <div class="form_submit">
            <input name="submit" type="submit" class="white" value="Register" />
        </div>
    </div>

</div>

```

3.13 - login /registration code

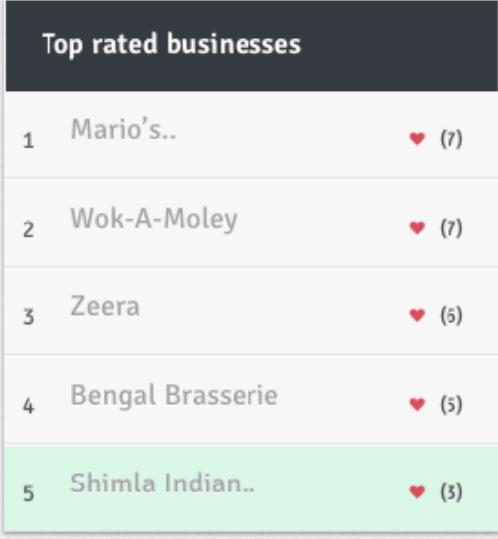
```

671     $(".login-link").click(function (e) {
672         if (member_group_id != "5") {
673             //opens link if the user isn't logged in (5 is the group member_id)
674             $.get("ajax/user-login", function (e) {
675                 //gets user login template ajax folder
676                 $("#user_favAccess").html(e);
677                 $("#user_favAccess").css('background', 'transparent');
678                 $("#user_favAccess").lightbox_me({
679                     centered: !0,
680                     overlayCSS: {
681                         background: "white",
682                         opacity: 0.9
683                     },
684                     onLoad: function () {}
685                 });
686             });
687
688             $(".user_login .userProgress").hide();
689             $("#userLoginForm").ajaxForm({
690                 dataType: "json",
691                 success: function (event) {
692                     if (event.success) {
693                         member_group_id = "5";
694                         //group_id is 5 if the user login is successful
695                         $('.background-shadow img').remove();
696                         $("#userLoginForm .form_elements").remove();
697                         $("#userLoginForm p").remove();
698                         $("#worlds_birth div nav ul li#signin, header#header_inner #main_container div nav.browsing ul li#signin").remove();
699                         $("#worlds_birth div nav ul li#registration, header#header_inner #main_container div nav.browsing ul li#registration").remove();
700                         $("#worlds_birth div nav ul, header#header_inner #main_container div nav.browsing ul").append('<li><a href="#>');
701                         $('.user_login').addClass('no_bg');
702                         //changes navigation if the user is logged in, my favourites added, login/register removed
703                         $(".userProgress").fadeIn(400, function () {
704                             $('.userProgress').animate({
705                                 opacity: 1
706                             }, 2200, function () {
707                                 $('#user_favAccess').fadeOut(200, function () {
708                                     $("#user_favAccess").trigger("close")
709                                 })
710                             })//success message
711                         })
712                     } else alert("Something was\\n't right, try again!")//alerts error
713                 }
714             });
715         }
716     });
717 });
718 });
719 });
720 });

```

3.5 – Top rated favourites

This script shows how the top favourites section is displayed on screen. The code beside it shows the loop that was coded to apply this functionality with a limit of 5 businesses to the sidebar.



Rank	Business Name	Favorites Count
1	Mario's..	7
2	Wok-A-Moley	7
3	Zeera	5
4	Bengal Brasserie	5
5	Shimla Indian..	3

```
<ul id="nav-sub" class="nav-list">
{exp:favorites:rank
    channel="accounts"
    orderby="count"
    sort="desc"
    limit="5"
    pagination="bottom"
}
<li class="clearfix" style="position: relative">
    <div class="rank" style="float: left; display: inline; padding-right: 10px; font-size: 13px">{absolute_count}</div>
    <div class="acc_title" style="float: left; display: inline">
        <a href="/browse/profile/{url_title}">
            {exp:eehive_hacksaw words="2" append=".."}{title}{exp:eehive_hacksaw}</a>
    </div>
    <div class="public_count" style="float: right; display: inline; text-align: right">
        <i class="icon-heart" style="z-index: 5; font-size: 11px; color: #d1364d"></i>
        <div class="index" style="z-index: 99999; font-size: 11px; float: left; font-family: signika-semibold; border-bottom: 1px solid black; padding-bottom: 2px; margin-right: 5px">{favorites_count_public}</div>
    </div>
</li>
{/exp:favorites:rank}
</ul>
```

3.6 – flip CSS

```

1  .flipAnimation{
2      position: relative;
3      -webkit-perspective: 600;
4      -moz-perspective: 600px;
5      perspective: 600px;
6      &.outside{
7          z-index: 102;
8      }
9      .on.outside{
10         z-index: 103;
11     }
12     .front,
13     .back{
14         position: absolute;
15         overflow: hidden;
16         height: 100%;
17         width: 100%;
18         -webkit-backface-visibility: hidden;
19         -moz-backface-visibility: hidden;
20         -webkit-transform-style: preserve-3d;
21         -moz-transform-style: preserve-3d;
22         -webkit-transition: ease-in-out 0.8s;
23         -moz-transition: ease-in-out 0.8s;
24         transition: ease-in-out 0.8s;
25     }
26     .front{
27         z-index: 100;
28     }
29     .back{
30         z-index: 99;
31         -webkit-transform: rotatey(180deg);
32         -moz-transform: rotatey(180deg);
33         transform: rotatey(180deg);
34     }
35 }

```

3.7 – Flip Animation jQuery

```

431     var $elements = $(".accounts_frame >div >div");
432     $elements.each(function(){
433         var $accountInfo = $(this);
434         var $front = $(":first-child");
435         var $back = $(":last-child");
436         var $click = $('.button', this);
437         $(this).addClass('flipAnimation');
438         $front.addClass('front');
439         $back.addClass('back').removeClass("hide");
440         $click.click(function(event){
441             event.preventDefault();
442             $elements.removeClass('on');
443             $accountInfo.addClass('on outside');
444             setTimeout(function(){
445                 $elements.removeClass('outside');
446                 $accountInfo.addClass('on outside');
447             },300);
448         });
449         $('a.close').click(function(e){
450             e.preventDefault();
451             $accountInfo.removeClass('on');
452             setTimeout(function(){
453                 $accountInfo.removeClass('outside');
454             },600);
455         });
456     });
457 }

```

```

458 var accountContainer = $(this);
459 function flipPosition(){
460     var accountContainerPosition = accountContainer.offset(); //retrieves current position of the element
461     var accountContainerWidth = accountContainerPosition.left + accountContainer.width()/2;// flips the card towards the centre after each animation
462     $elements.each(function(){
463         var accountInfo = $(this).offset();
464         if(accountInfo.left > accountContainerWidth){
465             $(this).removeClass('leftSide');
466             $(this).addClass('rightSide');
467         }else{
468             $(this).removeClass('rightSide');
469             $(this).addClass('leftSide');
470         }
471     });
472 }
473 flipPosition();

```

3.8 - Functional testing

Test ID	Description	Mistake	Fault	Failure	Error	Specification
1	Cursor pointer on all links, to indicate that area it is clickable					Yes
2	A hover is visible over all links					Yes
3	Spelling and grammar					Yes
4	Contact form validation					Yes
5	Sign up form validation					Yes
6	Main Search works					Yes
7	Inner search works			No		
8	Search nearby works					Yes
9	Flip animation works					Yes
10	Review can be submitted					Yes
11	Review validation		No			
12	User can register					Yes
13	User can login					Yes
14	Email sent after registration					Yes
15	User can filter menu					Yes
16	Sidebar pulls in 8 nearest restaurants					Yes
17	User can favourite businesses					Yes
18	Menu category links to heading					Yes
19	Map location shows					Yes

Test ID	Description	Mistake	Fault	Failure	Error	Specification
20	Reviews display					Yes
21	IE supported		No			
22	Firefox supported					Yes
32	Opera supported					Yes
33	Works on mobile devices					Yes
34	User name displays after login on inner page					Yes
35	Contact form submits form					Yes
36	Way points work					Yes
37	Lightbox works					Yes
38	responsive design					Yes
39	Reset button		No			

3.10 – Menu filter

```

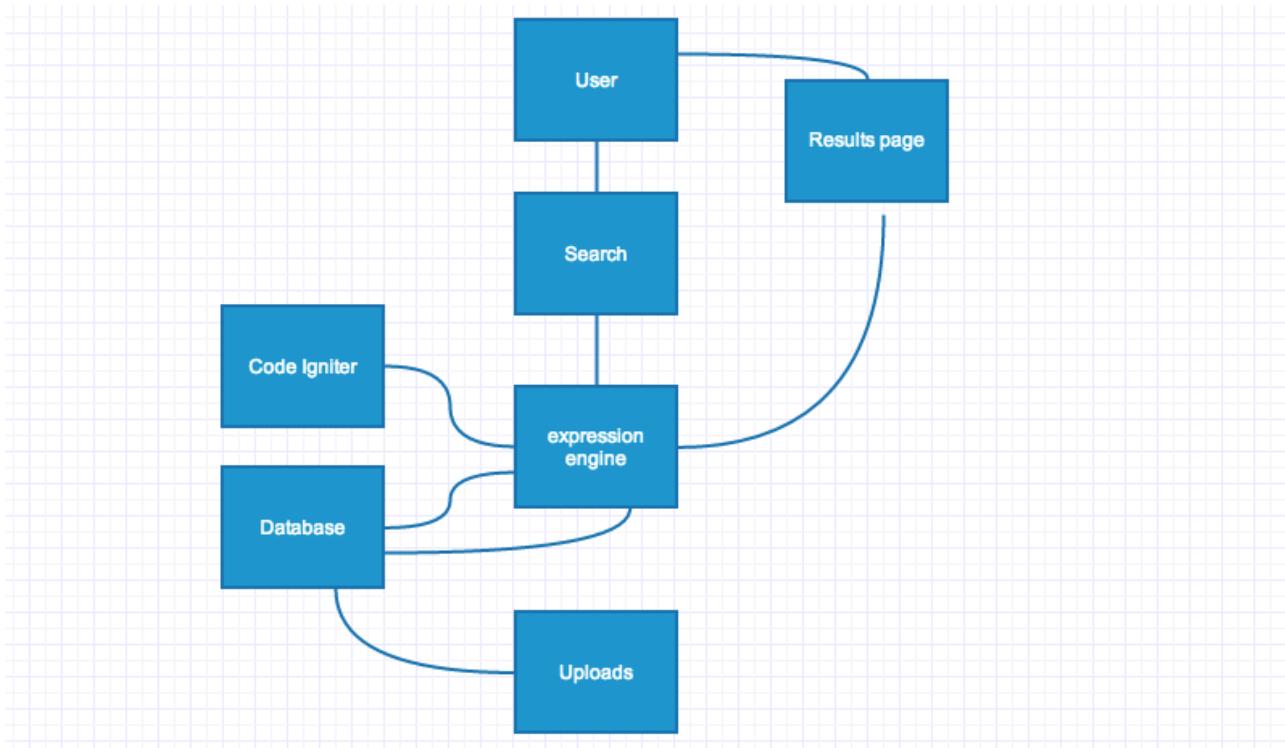
918 $(function(){
919     $(".menu_filter").click(function(){
920         // Retrieve the input field text and reset the count to zero
921         var filter = $(this).val(), count = 0;
922         // Loops through the menu
923         $(".menu_price").each(function(){
924             // fades out menu item if its not available
925             if (parseInt($(this).text()) > filter) {
926                 $(this).parent().fadeOut();
927                 // Show the menu items
928             }
929             else {
930                 $(this).parent().show();
931                 count++;
932             }
933         });
934         // Displays the amount of meals
935         var numberItems = count;
936         $("#filter-count").text(count +" Meals ");
937     });
938
939         //Resets the input
940         $('#reset').click(function() {
941             $(".menu_price").show();
942         });
943         //updates the value of the input range element
944         $('[type=range]').change(function(){
945             var newval= $(this).val();
946             $("#slidernumber").text(newval);
947         });
948 });

```

3.11 – Populating categories

```
49          {acc_menu}
50              {if category}
51                  <a href="#{category}">{category}</a>
52              {if:else}
53                  <a href="#" style="display: none">{category}</a>
54              {/if}
55          {/acc_menu}
56      </div>
57      <div id="menu_heading">
58          <ul class="commentlist">
59              {acc_menu}
60                  <li id="{category}">
61                      {if category != ""}
62                          <div class="menu_cat" style="position: relative !important">
63                              {category}
64                              <span class="left"></span>
65                          </div>
66                  {/if}
```

3.12 – Case diagram



3.13 – File structure

wt-config.php	Today 14:00	16 KB	PHP
themes	27 Mar 2014 21:20	--	Folder
templates	27 Mar 2014 21:12	--	Folder
default_site	Today 12:27	--	Folder
wrappers.group	Today 12:19	--	Folder
usercp.group	14 Apr 2014 00:44	--	Folder
pages.group	Today 12:17	--	Folder
browse.group	8 Apr 2014 19:42	--	Folder
ajax.group	2 Apr 2014 21:22	--	Folder
.DS_Store	Yesterday 13:06	6 KB	Document
.DS_Store	16 Apr 2014 14:31	6 KB	Document
system-penguin	27 Mar 2014 21:12	--	Folder
snippets	2 Apr 2014 19:57	--	Folder
robots.txt	27 Mar 2014 21:45	24 bytes	Plain...cument
index.php	13 Apr 2014 22:16	6 KB	PHP
images	Yesterday 13:06	--	Folder
uploads	Today 13:27	--	Folder
main-images	Today 13:32	--	Folder
logos	Today 13:31	--	Folder
index.html	27 Mar 2014 21:55	114 bytes	HTML...ument
.DS_Store	Today 13:33	15 KB	Document
_thumbs	27 Mar 2014 21:13	--	Folder
.DS_Store	Today 13:27	12 KB	Document
favicon.ico	6 Apr 2014 02:04	1 KB	Windo...image
cpanel.php	14 Apr 2014 01:06	6 KB	PHP
assets	27 Mar 2014 20:40	--	Folder
less	Yesterday 23:45	--	Folder
js	Yesterday 23:45	--	Folder
images	Today 15:32	--	Folder
fonts	30 Mar 2014 20:57	--	Folder
css	13 Apr 2014 23:03	--	Folder

4.1 – Google page speed results

To improve the Google page, all scripts need to be compressed and compiled into one page of javascript. This task will be completed on the day of the launch of the application.

www.majorproject.jamcrowe.com ANALYZE

Mobile Desktop

51 / 100 Suggestions Summary

! Should Fix:

Enable compression

Compressing resources with gzip or deflate can reduce the number of bytes sent over the network.

Enable compression for the following resources to reduce their transfer size by 476.9KiB (80% reduction).

- Compressing http://www.majorproject.jamcrowe.com/assets/css/style.css could save 148.2KiB (87% reduction).
- Compressing http://www.majorproject.jamcrowe.com/ could save 62.5KiB (91% reduction).
- Compressing http://www.majorproject.jamcrowe.com/assets/js/libs/jquery-1.7.1.min.js could save 59.3KiB (64% reduction).
- Compressing http://www.majorproject.jamcrowe.com/assets/css/animate.css could save 51.5KiB (92% reduction).
- Compressing http://www.majorproject.jamcrowe.com/...sets/js/libs/jquery-ui-1.10.4.custom.js could save 43KiB (73% reduction).
- Compressing http://www.majorproject.jamcrowe.com/assets/js/myscript.js could save 34.4KiB (81% reduction).
- Compressing http://www.majorproject.jamcrowe.com/assets/js/libs/jquery.form.js could save 30.8KiB (72% reduction).
- Compressing http://www.majorproject.jamcrowe.com/assets/css/jquery-ui-1.10.4.custom.css could save 17.8KiB (79% reduction).



4.2 – *imageOptim*

ImageOptim			
File	Size	Savings	
bengal-brasserie.gif	2,686	0%	
bombay-brasserie.gif	2,348	0%	
burger_king_crowns_by_micheldani-d...	100,319	10.4%	
fon-a-kebab.gif	2,224	0%	
four-star.jpg	14,505	2.6%	
fourstarpizzacavehil.jpg	14,505	2.6%	
images-1.jpg	11,282	5.0%	
thai.jpg	20,109	3.5%	
shadow.png	19,406	0%	
pattern.png	39,886	2.7%	
noise-2.png	2,740	0%	
noise-1.png	2,747	0%	
no-logo.png	1,162	56.9%	
no_main.jpg	53,308	4.9%	
logo.png	22,738	27.9%	
homescreen.png	325,602	10.3%	
grad_fav.png	135	0%	
gallery2.jpg	48,722	8.2%	
features3.png	59,416	1.5%	

+ Saved 555.8KB out of 8.6MB. 9.9% per file on average (up t... Again

4.3 – Further Testing

Pingdom

The first round of page speed tests came from Pingdom. The results from this performance test on the home page declared that the page was graded 84/100. A score of above 90 is considered a good grade to have a web page perform at. The issue that was causing the page to have a lower grade was that the resources in the header needed to be compressed. There are a lot of individual jQuery scripts in the assets folder, these scripts will be minified to reduce load time. (figure 3.7).



Google page speed

The second round of page speed tests were conducted with google page speed, according to Google a score of above 85 indicates the page is performing well. The results from this round declared the application was performing with a score of 51/100. The results show that the issue with the performance is the amount of scripts and stylesheets that are being processed on page load. This secondary test has confirmed that minimizing scripts will be essential to improve performance (see appendix 4.1).

A common issue that was also noticed in both rounds of page speed tests was that loading of images. To reduce image size, software program imageOptim has been downloaded and all images have be run through this application. ImageOptim reduces image sizes by compressing parameters and removing avoidable color profiles and any comments linked to the images (see appendix 4.2).

Browser Testing

To test this application in multiple browsers it was decided that the online application browserstack would be used. Browserstack allowed for multiple browsers to be tested including legacy browsers such as outdated versions of Internet Explorer. Issues did arise using IE, this was expected as the application using CSS3 for interactivity and animation. The browser testing caused a lot of issues with windows xp simulated tests, at first this raised concern. Although research declared it was no longer supported.

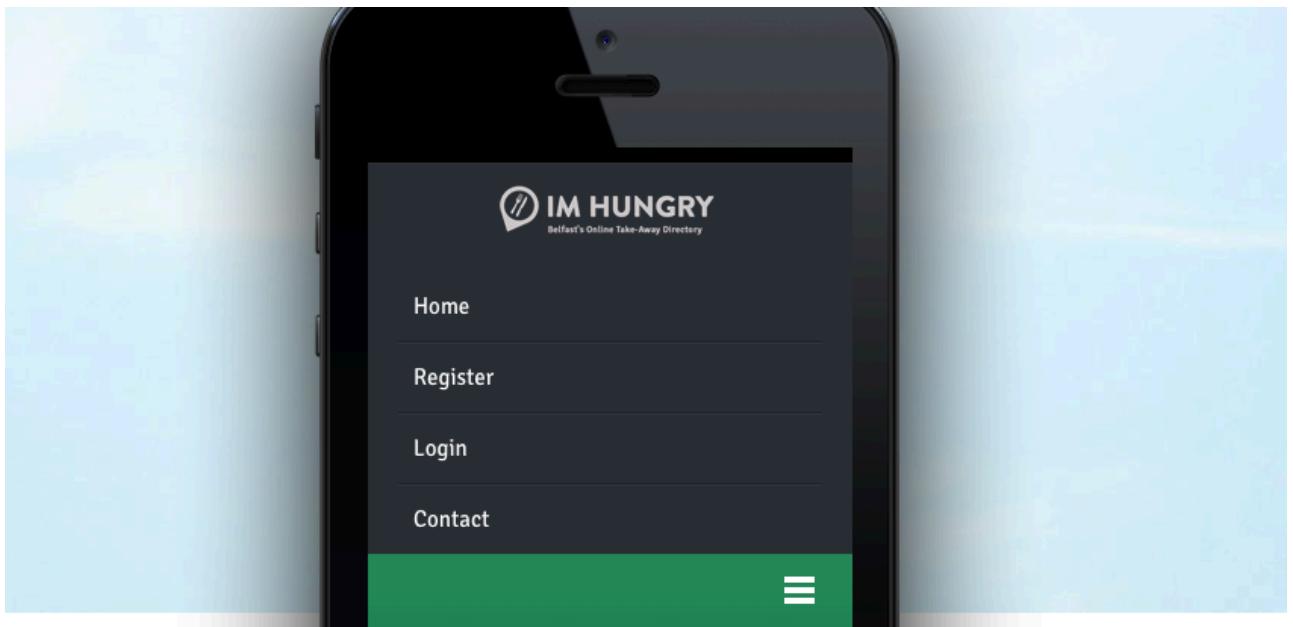
Windows XP support has ended - Microsoft Windows

windows.microsoft.com/en-GB/windows/end-support-help ▾

Get end of support information for Windows XP and find out what you need to know to upgrade to Windows 8.1 or find a new PC.

Mobile testing

There are many devices that this application could be viewed on and a method to test all or at least the majority of them needed to be concluded. It was decided that using mobiletest.me would be a good way to gain a perspective of how users will use the application across a wide range of devices. By testing with mobileme there were responsive bugs that had not been realised until this method of testing was included. One bug in particular was that the login and registration has not been hooked up correctly for mobile devices. The devices tested included iPad Air, iPhone 5, HTC One, Nokia Lumia and the Samsung Galaxy Y. Issues also arose with the Samsung mobile, this device has a small screen, smaller than the standard width of 320px. A new media query need to be wrote to target the actual screen size of this device. Research showed that this screen size was 240 x 320px.



4.4 - 404 page

