

STAY AT  
**MINE**

# **Major Project Final Report**

BSc Interactive Multimedia Design

2015/16

Ryan Warke (B00606999)

**Mentor:** George Moore   **PSG:** 2

# Acknowledgements

---

I would like to take this opportunity to thank everyone who helped me through the development of the Stay At Mine web application. Firstly, I would like to thank my mentor, Dr George Moore, whose guidance throughout the process helped turn the idea into a reality.

I'd also like to thank all the members of the Interactive Multimedia Design teaching staff, whose lectures, guidance and support over my time on the course have helped me develop the skills needed in order to undertake this project.

Furthermore, I would like to thank all the members of the iCON Creative team, specifically Gareth Lynas, who allowed me to focus on the development of Stay At Mine, whilst also still working within the company. My time on placement with iCON undoubtedly contributed to me being able to undertake some of the more difficult coding aspects of the application.

I would also like to acknowledge the support of my fellow students, and good friends, Matthew Skinner and Nathan Nelson, whose ideas, assistance and advice have been invaluable in helping me to overcome some of the more complex problems surrounding the project.

Finally, I would like to thank my girlfriend Gail for helping me get through the past year, and being extremely understanding as the pressure mounted. Her assistance and comments throughout the design and testing of the project were crucial in getting Stay At Mine to the stage at which it currently stands.

# Contents

---

<b>1.0 Introduction</b>	<b>7</b>
<b>1.1 Background</b>	<b>7</b>
<b>1.2 Aims &amp; Objectives</b>	<b>8</b>
1.2.1 Aims	8
1.2.2 Objectives	8
1.2.3 Scope	9
<b>1.3 Overview of Work Undertaken</b>	<b>9</b>
<b>1.4 Overview of Report</b>	<b>9</b>
<b>2.0 Concept Definition &amp; Testing</b>	<b>10</b>
<b>2.1 Idea Generation</b>	<b>10</b>
<b>2.2 Contextual Research</b>	<b>10</b>
<b>2.3 Requirements Specification</b>	<b>13</b>
<b>2.4 Paper Prototyping</b>	<b>14</b>
2.4.1 Sitemap	14
2.4.2 6-Up / 1-Up	15
<b>2.5 Feasibility Testing</b>	<b>29</b>
2.5.1 Risks	29
<b>2.6 Methodology Selection</b>	<b>30</b>
<b>3.0 Design</b>	<b>30</b>
<b>3.1 User Experience Evolution</b>	<b>30</b>
3.1.1 Detailed Wireframes	30
3.1.2 Branding	31
3.1.3 User Flow	32

---

3.1.4 Mockups	33
<b>3.2 System Design</b>	<b>36</b>
3.2.1 Refined Sitemap	36
3.2.2 Client Server Model	36
<b>3.3 Logic Design</b>	<b>37</b>
<b>3.4 Data Design</b>	<b>38</b>
3.4.1 Database Design	38
<b>4.0 Implementation</b>	<b>38</b>
<b>4.1 Technology/Tool Review</b>	<b>38</b>
4.1.1 Server-Side Technologies	39
4.1.2 Client-Side Technologies	41
4.1.3 Frameworks	41
4.1.4 Databases	43
4.1.5 Other	44
<b>4.2 Technology/Tool Selection</b>	<b>46</b>
4.2.1 Server-Side Technologies	46
4.2.2 Client-Side Technologies	46
4.2.3 Frameworks	48
4.2.4 Database	49
4.2.5 Database	49
<b>4.3 Technology/Tool Use</b>	<b>50</b>
4.3.1 Client-Side Technologies	50
4.3.2 Server-Side Technologies	50
<b>4.4 Notable Challenges</b>	<b>51</b>
<b>5.0 Testing</b>	<b>54</b>

---

<b>5.1 Test Approach Selection</b>	<b>54</b>
5.1.1 White Box Testing	54
5.1.2 Black Box Testing	54
5.1.3 Cross Browser Testing	55
<b>5.2 Test Process</b>	<b>55</b>
<b>5.3 Test Results</b>	<b>55</b>
<b>5.4 User Survey Responses</b>	<b>56</b>
<b>6.0 Evaluation</b>	<b>57</b>
<b>6.1 Evaluate Test/Survey Results</b>	<b>57</b>
<b>6.2 Evaluate Project Outcomes</b>	<b>58</b>
<b>6.3 Evaluate the Methodology</b>	<b>58</b>
<b>7.0 Conclusion</b>	<b>59</b>
<b>7.1 Summarise Report</b>	<b>59</b>
<b>7.2 Reflect on What Happened</b>	<b>59</b>
<b>7.3 Reflect on Your Role</b>	<b>59</b>
<b>7.4 Suggest Future Work</b>	<b>60</b>
<b>8.0 References</b>	<b>61</b>
<b>9.0 Appendices</b>	<b>66</b>
<b>Appendix A</b>	<b>66</b>
<b>Appendix B</b>	<b>67</b>
<b>Appendix C</b>	<b>68</b>
<b>Appendix D</b>	<b>69</b>
<b>Appendix E</b>	<b>70</b>
<b>Appendix F</b>	<b>71</b>
<b>Appendix G</b>	<b>81</b>

---

<b>Appendix H</b>	<b>86</b>
<b>Appendix I</b>	<b>90</b>
<b>Appendix J</b>	<b>96</b>
<b>Appendix K</b>	<b>99</b>
<b>Appendix L</b>	<b>100</b>
<b>Appendix M</b>	<b>104</b>

## 1.1 Background

Stay At Mine is a fully responsive Web Application that aims at getting homeless people somewhere to stay for the night. The idea is similar in ways to ‘Airbnb’, in that it will rely on users to post rooms in their homes that they would be willing to offer to a homeless person to stay in for the night.

Any homeless person wanting to make use of the application will need to have a detailed profile created for them, which can be verified by a charitable organisation. This will be done to ensure the safety and security of any willing homeowners who wish to take in a homeless person. The information provided will also be used to create a filter system, that will allow homeowners to specify exactly what type of homeless person they are willing to accommodate.

Once a homeowner posts a vacancy, which they are willing to offer to a homeless person, and have provided very minor details about their accommodation and living arrangements (to ensure the safety and security of the homeless person), the homeless person will be able to apply for the vacancy. In order to apply for the vacancy, a charitable organisation, friend, or indeed the homeless person themselves (if they are fortunate enough to have access to the internet), will be able to submit a short application expressing their interest in staying the night. The homeowner will then review the application, and either accept or deny it based on their personal preference. Only when an application has been accepted will details of the address and directions on how to get there be revealed.

After the homeless person has spent the night, both parties will have the opportunity to review each other within the application. These comments will be reviewed before being made public, and any derogatory comments will need to be investigated.

The idea behind Stay At Mine came about after watching a Sky News bulletin that stated some pretty frightening statistics on homeless people. Sky News (2015) reported “there is no national figure for the number of homeless people in the UK ... because many do not even show up as a statistic.” This is a scary thought, to think that just because you are homeless, you’re not even worth being a statistic. Sky News (2015) continued to report that homeless people are “13 times more likely to be a victim of violence than you”, and that, on average, homeless people “die at just 47 years old”.

---

Upon further investigation into homeless people in the UK, I came across Homeless Link (2014), which contained facts and figures on homeless people in England alone. Homeless Link reports that, in 2014, 2,744 people were sleeping rough in England, an increase of 55% from 2010, meaning that the problem is only going to get significantly worse. [See Appendix A]

I think that the prospect of this application is very appropriate at the minute given the current migrant crisis currently taking place across Europe. Recent statistics show that, as of July 2015, the number of Syrian asylum applications for the UK totalled some 7,030, which promises only to rise drastically over the coming months. [National Post, 2015] [See Appendix B]

Therefore, I can foresee a great use of the Web Application, not only for current homeless people within the UK, but also for migrants entering the UK who may not have anywhere to stay upon entering the country.

## **1.2 Aims & Objectives**

### **1.2.1 Aims**

The intended outcome for the Stay At Mine Web Application is to provide a platform for homeless people to use as a stepping-stone, to help them to rehabilitate their life, and get back onto their feet. The application will allow them to focus on getting a job, or finding permanent accommodation, without having to worry about where they will be sleeping each night.

### **1.2.2 Objectives**

In order to meet this aim, the following objectives will need to be met:

- Research the best platform to build the application on. Will it be built from scratch using PHP, or will it incorporate a PHP framework such as Laravel or CodeIgniter?
- Ensure that the User Interface is easy and friendly to use, and only shows information that is relevant and useful to the users.
- Guarantee that the application is fully functional on all devices, and across all browsers.
- Ensure that the database is designed correctly from the outset.
- Design each stage of the application fully before coding any of it.
- Ensure that the application has been fully tested by a variety of different users, with a variety of different skill-sets.

### **1.2.3 Scope**

The scope of my work, for developing the Sheltered/Shelter Me Web Application will be:

- To develop a fully responsive Web Application that runs, displays and performs seamlessly on all devices and browsers.
- The application will allow users to sign up as either a homeowner or homeless user.
- Homeowners will have the ability to post a vacancy within their household.
- Homeless people will have the ability to apply for a vacancy within a household.
- A review process will take place at the end of the interaction, which will allow both users to assess each other.
- The design of the application will be of utmost importance, ensuring that the usability and user experience is well accounted for.

My work will not include the following:

- Help and support on how to use the application.
- Training for any potential charitable organisation that would be interested in using the application on behalf of a homeless person.

### **1.3 Overview of Work Undertaken**

In order to fully develop the Stay At Mine application, a number of research, development and testing phases had to be carried out. The project brought together all aspects of the Interactive Multimedia Design course, and has allowed me to implement knowledge gained from all years of the degree programme. The project also allowed for the learning of new technologies, such as Laravel, and the need to undertake all roles within a web development team.

### **1.4 Overview of Report**

The following report will follow the structure of the waterfall methodology, beginning with concept definition and testing. In this section, I will overview how I came up with the idea, my research behind it, and my requirements for the development. This will be followed by the design of the Stay At Mine application, which will incorporate, UX evolution, system design, and data design. I will then provide an overview of the implementation, followed by the testing phase, and finish off with an evaluation of the process and conclusion.

## 2.0 Concept Definition & Testing

---

### 2.1 Idea Generation

As previously stated, the idea behind the Stay At Mine application originated from a Sky News report highlighting the somewhat unknown statistics of homelessness in what is supposed to be a developed country. After being shocked by the news report, I began investigating further into homelessness within the UK, and was astounded to discover just how drastic it really is.

Having been left wanting to do something to help, I started to think what I could do to make a difference. As technology is such a major influence in our daily lives, and it is primarily what I am involved in on a daily basis, I began to think about how I could make use of technology to help benefit those who are in need of help. After watching AirBnBs successful rise over the past few years, I had the idea of using a similar system to provide homeless people with shelter with willing volunteer homeowners.

Further evidence for the need for such an application came in the midst of the migration crisis across Europe in the Summer of 2015. The influx of refugees, with nowhere to go once they reach the UK, means that the probability of homelessness across the UK increasing in the coming years, is high. Amongst the furore surrounding the refugees crisis, quite a number of influential figures, such as Bob Geldof, Jeremy Corbyn and Nicola Sturgeon have spoken out and offered the incoming refugees a place in their homes. [The Guardian, 2015]

The emergence of such significant support for welcoming refugees into your own home, combined with the frightening statistics surrounding homelessness within the United Kingdom, sparked the idea of an 'AirBnB' style application that would focus on connecting homeless people and refugees with homeowners willing to provide them shelter, a hot meal and cleaning facilities - basic human rights.

### 2.2 Contextual Research

What generated the idea behind the Shelter/Shelter Me Web Application, was a Sky News report addressing just how bad the issue was. However, before gathering any facts or figures on Homeless people, we must first take a look at who is defined as being 'Homeless'.

According to Crisis, "the national charity for single homeless people", homeless people are defined as:

- People sleeping rough;
- Single homeless people living in hostels, shelters and temporary supported accommodation;
- Statutorily homeless households;
- Aspects of ‘hidden homelessness’ amenable to statistical analysis using large-scale surveys, namely ‘concealed’;
- ‘Sharing’ and ‘overcrowded’ households.

[Crisis, 2015]

Therefore, according to the above definition, you are considered as being homeless regardless of whether or not you are sleeping rough on the street. This consequently allows me to narrow down my target audience specifically to those people who are sleeping rough, as someone who has accommodation in a hostel, shelter, temporary supported accommodation or a crowded household, would not actively be seeking a temporary reprieve from the harsh conditions of sleeping rough, nor should they take the shelter that would be offered to someone who is much more in need.

With the Belfast Telegraph reporting that the amount of people sleeping rough, according to the Department for Communities and Local Government (DCLG), has increased by 14% in the year 2013-2014, and 55% from 2010, the issue surrounding sleeping rough doesn’t seem to be getting any better. [Belfast Telegraph, 2015]

With the current migrant crisis presently taking place across Europe, the number of people within the UK sleeping rough proves only to increase in the coming months and years. This therefore provides a massive gap where the Shelter/Shelter Me Web Application could prove to be extremely useful. As said by The Telegraph, “some 473,887 people reached Europe by sea” in the first two weeks of September. Whilst not all of these 473,887 people will be travelling to the UK, and not all of them will be sleeping rough, there will still be a staggering amount of homeless people on the streets of the UK in the coming years. Also, the Application, if done well, would not have to be limited to just the UK, as the rest of Europe seems to be facing a much more difficult situation in terms of migrants, with Germany receiving over 80,000 asylum applications between April and June 2015, and Hungary receiving over 30,000 asylum applications during the same time period. [The Telegraph, 2015] [See Appendix C]

The UK doesn’t really compare to the figures presented by other European states, with the

British Government receiving a mere 4,305 asylum applications in July 2015, compared to some 37,525 applications made to the German Government, and 31,285 applications made to the Hungarian Government. [Electronic Immigration Network, 2015]

Whilst the facts and figures are certainly there to support the need for the development of such an application, it would still need to be determined whether or not the audience exists that would be willing to take in homeless people into their homes. In relation to a recent YouGov poll, some 11% of British people surveyed, voted that they would take a refugee into their home for up to six months. Whilst 11% seems like a drastically low figure, if this were a true reflection of the 64.6 million people living in the UK (Mid 2014 Estimate), then some 7.1 million people, in the UK, would be willing to take in a refugee, vastly surpassing the current number of asylum applications and homeless people combined. This therefore proves that the facts and figures are also there to support the fact that an audience also exists that would be willing to take in homeless people into their homes. [Office for National Statistics, 2015]

In terms of an existing solution, there isn't really anything out there that targets people sleeping rough, and helping them to find temporary shelter. However, perhaps the closest comparative solution would be Airbnb. Airbnb is a web application that allows users to post accommodation for other people to rent short term, i.e. if someone where travelling to Barcelona, rather than staying in a hotel in a tourist district, they could stay in a traditional Spanish apartment. Airbnb, or Air Bed & Breakfast as it was originally named, started off in 2007 as the consequence of two roommates who were unable to pay the rent of their San Francisco apartment. Therefore, the two roommates, Brian Chesky and Joe Gebbia, placed an airbed in the living room of their apartment and offered it out in return for cheap short-term rent in a city where the hotel market was extremely inundated with guests, and therefore allowed them to be able to pay their rent. In coordination with the SXSW (South by South West) festival, Chesky and Gebbia were able to secure their first three guests, each paying \$80 each per night. [Funders and Founders, 2015]

[See Appendix D]

Whilst the Airbnb model is comparative to the proposed Shelter/Shelter Me Web Application, one fundamental difference between the two is that Airbnb is a business whose sole purpose is to create an alternative layer of accommodation for well-off professionals on their travels, and therefore make money. The idea behind Shelter/Shelter Me is a purely not for profit idea, that would provide accommodation for those in need. This is in stark contrast to the Airbnb business model, whose 72,000-square-foot offices encapsulates a section of SoMa that has no fewer than five homeless encampments. [The Guardian, 2015]

## **“We believe in a world where all seven billion of us can belong anywhere” – Airbnb Motto**

However, in retaliation of the growing homeless population in their neighbourhood, Jamie Shin and Vito Catalani, decided to try to raise awareness of the 60,000 homeless people in New York City, through Airbnb. Their “Homeless Airbnb” project used the network as a platform to post listings of ‘homeless accommodation’ throughout New York City. These listings included places such as “one bedroom in Union Square” and “hot bedroom with semi private terrace”, which were essentially a park bench and a sheltered doorway respectively. These listings were offered for the low price of between \$10 and \$14, with all of the proceeds going to a New York City homeless shelter. [Triple Pundit, 2015] [Ad Week, 2015] [See Appendix E]

In Northern Ireland, Simon, a homeless charity, released a native iOS and Android application that allows homeless people to locate nearby hostels and food banks. The application uses the users GPS location to find the users nearest amenities, and also allows users to alert the charity to people who may be at risk. [BBC News, 2014] Whilst the application differs from the Shelter/Shelter Me idea, in that its sole purpose is for homeless people merely to locate potential shelter and food, and doesn’t go in any way to securing that accommodation or food for them, it does highlight the fact that homeless people do have access to the same technology as everyone else.

## **2.3 Requirements Specification**

The prospect of defining the requirements for such large-scale and important project was quite daunting. However, the introduction of the Volere Requirements Specification Template proved critical in putting together a list of requirements for the Stay At Mine application. The Volere Requirements Specification Template is a widely used and very popular guide for helping developers define the requirements of their projects, produced James & Suzanne Robertson. [Volere, 2015]

Perhaps the most important aspect of the Volere Template, at least in terms of the Stay At Mine application, are the Volere Snow Cards. These Snow Cards allow you to quickly and efficiently define your requirements, including all of the necessary data, but in an ordered functional manner. It was for this reason that I opted for using the Volere Snow Card method as a form of

obtaining the requirements for the Stay At Mine application.

The Snow Cards contain a number of headings such as, requirement number, requirement type and customer dissatisfaction, although not all of these were relevant for this specific project.

The headings that I opted to go for can be seen below:

- **Requirement #** - The number of the requirement.
- **Requirement Type** - Functional or non-functional requirement.
- **Description** - A brief overview of what is required.
- **Rationale** - A brief reason as to why it's required.
- **Dependencies** - Whether or not the requirement is dependent upon any other requirements.
- **Fit Criterion** - How the requirement is measured to test that the solution fixes the problem.

Due to the high volume of Volere Snow Cards needed, instead of using the card format, I have decided to outline the requirements in a table which follows the headings of the Volere Snow Cards. The requirements table can be seen in **Appendix F**.

## 2.4 Paper Prototyping

Having come up with an idea and figured out what would be required to make that idea into reality, the next stage is to start to visualise what the Stay At Mine application should actually look like. Aside from the actual development of the application, and ensuring that it works, this stage is probably one of the most crucial for the whole application, get this wrong and the probability of anyone using the end product is very slim. Therefore the best place to start is by sketching out a few ideas on paper, before refining them and ultimately designing up a final concept.

### 2.4.1 Sitemap

Before I could start sketching up ideas on how I wanted pages to look, I first had to have some idea of what pages were going to exist. To do this I did a quick sketch of a site map, that can be seen in figure 1 below.

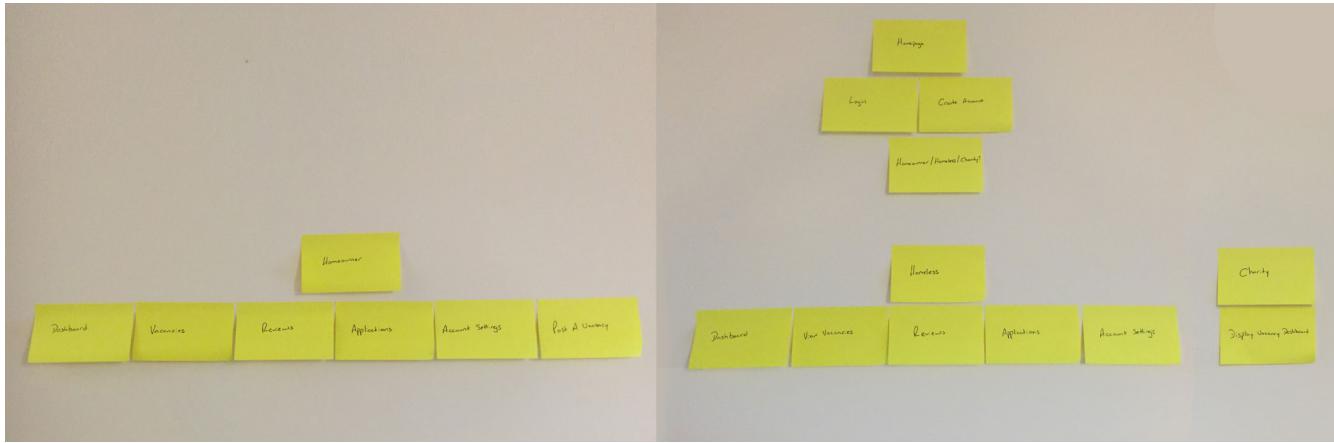


Figure 1: Original Sitemap

## 2.4.2 6-Up / 1-Up

6-Ups is a technique used to produce 6 rapid sketches of the same page. I opted for the 6-Ups method as I had previously found it very useful in brainstorming different layouts for the same information. Having to come up with six solutions for the same problem is a brilliant way of forcing you to think outside the box and can often foster some great ideas that you wouldn't have come up with if you just focused on one sketch.

### Homepage

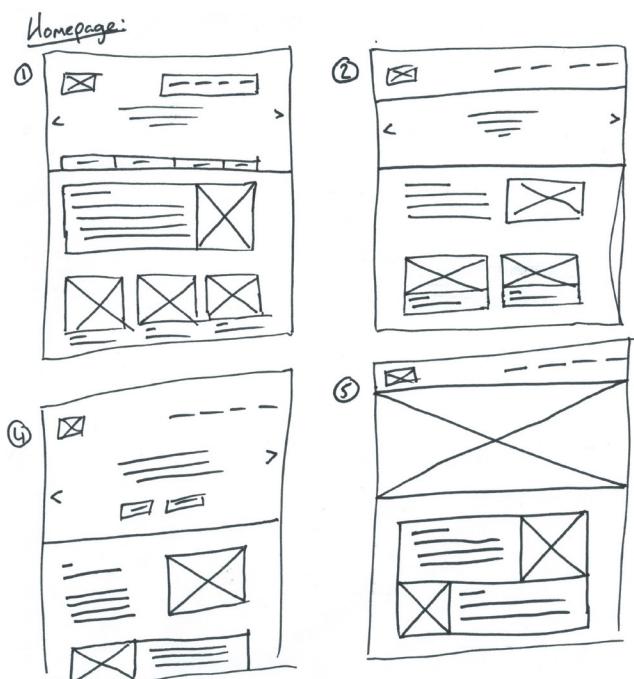


Figure 2: Homepage 6-Ups

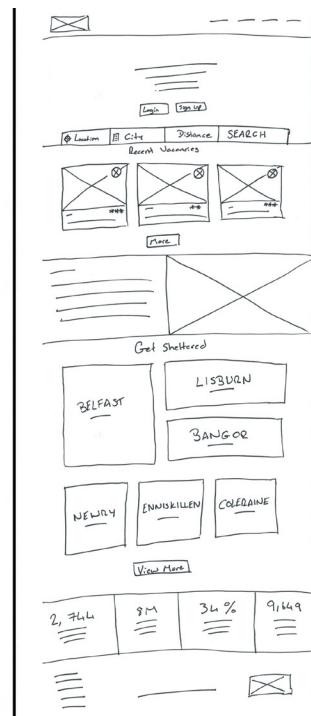


Figure 3: Homepage 1-Up

**Figure 2** above shows the 6-Ups that were created in the process of visualising the Homepage/Landing page to the Stay At Mine website. When moving onto the 1-Up for the Stay At Mine homepage, there wasn't necessarily one 6-Up that was chosen to focus on. Instead

aspects from each sketch were used to produce the final 1-Up that can be seen in **figure 3** above.

**Homepage 1:** The webpage begins with a full-width slideshow presenting images of homelessness, along with a powerful message that will encourage the user to sign up to use the service. The logo will appear in the traditional space in the top-left corner, with the menu appearing in the top-right corner. Overlapping the slideshow will be a search area that will allow the user to search quickly by city/current location and the distance they are willing to travel. As you scroll down the page, You will be provided with an overview of how the website works, along with three featured recent vacancies.

**Homepage 2:** This webpage is very similar to **Homepage 1** however, the menu is contained within a box, the search section has been removed, the number of featured recent vacancies has been reduced to two, and the information about how the application works is no longer container within a box.

**Homepage 3:** The slideshow section of **Homepage 3** is largely the same as **Homepages 1** and **2**, however, the information on how the application works has been brought up to overlap the slideshow slightly, and center within the page. A call to action has been set underneath the how to use information, enticing the user to sign up. The row of recent vacancies has been increased to four.

**Homepage 4:** The slideshow section of **Homepage 4** remains rarely altered from the previous. The major difference with **version 4** is that the recent vacancies row has been reduced to one recent vacancy, providing more information.

**Homepage 5:** This webpage takes a more minimal approach in its layout. The slideshow section features no text, and the information on how to use the application has been removed. Two recent vacancies are on display with an alternating layout.

**Homepage 6:** The final 6-Up is a combination of **Homepage 1** and **Homepage 2**. Coming across from **version 1** is the slideshow with the search area, and coming from version 2 is the recent vacancies section layout. However, in this version the number of recent vacancies that are on display has been increased to four.

**Homepage 1-Up:** For the 1-Up of the homepage, it was decided that various aspects of each 6-Up would be taken forward to produce a more well-rounded layout. Therefore, at the top of the 1-Up, you can see the header has remained constant, with the logo appearing at the left-hand side and the menu appearing at the right-hand side. The menu then overlaps a full-width slideshow that will contain an engaging message that will try to encourage users to sign up and use the application. Overlapping the bottom of the slideshow will be the search section that featured in a couple of the previous 6-Ups.

Directly below the main slideshow, three recent vacancies will be displayed in a carousel style element that will rotate every few seconds. The recent vacancies are then followed by a section which informs the user on how to use the application, and how it will benefit them. Following the instructional message, a grid style layout will appear that will allow the user to search for a vacancy within trending locations throughout Northern Ireland, with the image growing gradually smaller depending upon the popularity of that location.

Below the trending locations grid, there will be a display of four statistics about homelessness within the UK, in the hope that this encourages users to sign up. At the bottom of the page will be the same footer that will be displayed on every ‘static’ page within the site. The footer will consist of quick link menu items, the Stay At Mine logo, copyright information and links to the terms and conditions of the application.

## Searching for a Vacancy



Figure 4: Searching for a Vacancy 6-Ups

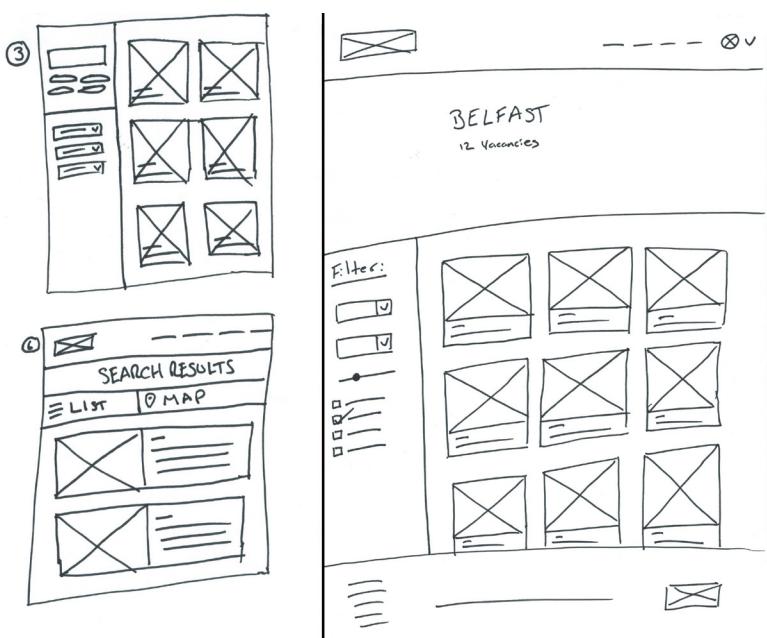


Figure 5: Searching for a Vacancy 1-Up

**Figure 4** above shows the 6-Ups that were created in the process of visualising the Search page. It was decided that 6-Up sketch number **2** would be predominantly used in the more

detailed 1-Up version for the searching for a vacancy page. The 1-Up can be seen in **figure 5** above.

**Search 1:** The first 6-Up for the search page keeps with it the traditional header from the homepage sketches. Below the header is a filter system that will allow the user to filter their search results to further reflect their needs. The layout of the search results is then in a two column layout, displaying the relevant vacancies in the same format as on the recent vacancies section of the homepage, therefore keeping a form of consistency within the site.

**Search 2:** The second 6-Up varies the traditional header of the page slightly by including the user's avatar at the top-right of the screen, allowing the user quick access to their account and logout facilities. Below the header will be a slightly blurred, full-width, image of the location that the user has searched for, with an overlapping heading displaying the user's search terms and a subheading of how many vacancies have been found. Following the full-width image is a search bar, that will be prefilled with the user's search terms, allowing the user to perform another search. The layout of the search results alternates each time with the thumbnail image of the vacancy either appearing on the left or right, with the description and vacancy data appearing on the alternate side.

**Search 3:** The third 6-Up removes the header section of the page, and instead opts for a more full-screen dashboard style layout. There is a sidebar that displays on the left-hand side of the screen that will allow the user to perform further search requests, as well as being able to filter their current search results. The search results are then displayed in a two column layout in the right-hand side of the screen, following the same format as the recent vacancies on the homepage.

**Search 4:** 6-Up number 4 follows the same layout as 6-Up number 3, however this time, there is a map above the search results. The search results are then displayed with the thumbnail image at the left-hand side, followed by the information of the vacancy on the right.

**Search 5:** The fifth 6-Up is a variation on 6-Up number 4, however with a more boxed layout. Instead of all of the sections being full-width, each section is constrained within a container, with some margin and padding in between.

**Search 6:** The final 6-Up reintroduces the traditional header section. The user is then given the option to display the search results either in a Google Map, or as a list. If the user chooses to display the results on a Google Map, each search result will be placed onto a Map, and if the

user would like more information on the vacancy, they can click on the map pin to be directed to that vacancies page. If the user chooses to display the results as a list, the results will be displayed in the same way as with 6-Up number 4.

**Searching for a Vacancy 1-Up:** Going forward with the development of the layout for the search results, I choose to predominantly develop further 6-Up number 2, whilst also incorporating various features from other 6-Ups. The header for the search page follows the traditional layout for the rest of the site, whilst also incorporating the user's avatar with links to their account settings and logout facilities. Below the header, a blurred image representing the user's searched for location is present in the background, with overlay text referencing the user search term, followed by a sub-heading of how many search results (vacancies) have been found. The search results are then displayed in the same format as the recent vacancies section of the homepage, so as to keep a form of consistency within the application. The search results section of the page also contains a sidebar on the left-hand side that displays a filter system that will allow the user to refine their search results, as well as performing another search function. At the bottom of the page will be the standard footer for the website.

## Viewing a Vacancy

**Figure 6** below shows the 6-Ups that were created in the process of visualising the page to view an individual vacancy. After sketching the 6-Ups for viewing an individual vacancy, it was decided the sketch number **1** would be progressed further. The 1-Up for the page to view an individual vacancy can be seen in **figure 7** below:

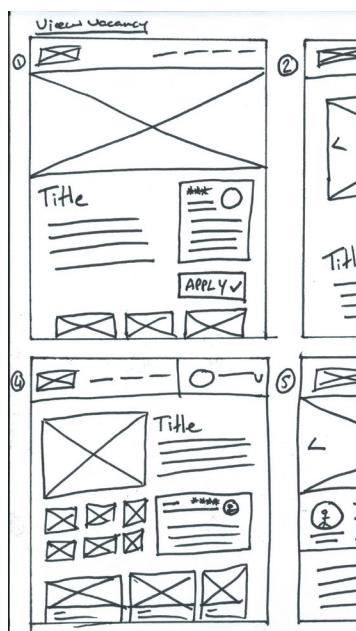


Figure 6: Viewing a Vacancy 6-Ups

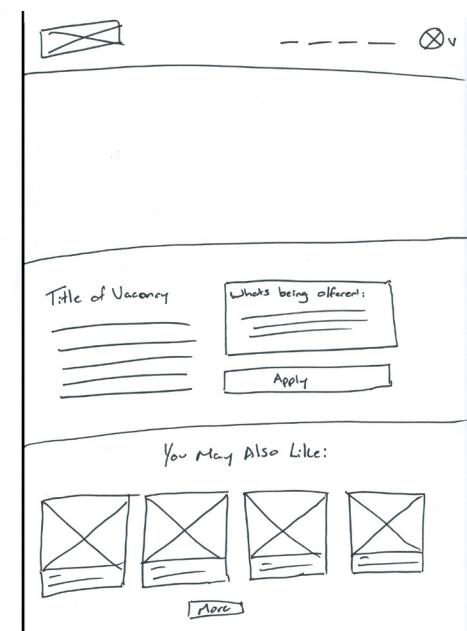


Figure 7: Viewing a Vacancy 1-Up

**Viewing a Vacancy 1:** The first 6-Up for viewing an individual vacancy incorporates the traditional header that is used throughout the website. A full-width slideshow will then follow the heading and will display images from the current vacancy. The main content area for the vacancy will be displayed in a two-third one-third grid with the left-hand side display the title and description of the vacancy and the right-hand side display a brief biography and rating of the vacancy homeowner. Below the brief biography of the homeowner will then be a call to action button where the user will be able to apply for the vacancy. Below the vacancy content will be a three column grid displaying three related vacancies that the user may also be interested in.

**Viewing a Vacancy 2:** The second 6-Up displays the vacancy sideshow in a boxed container, with preview thumbnails below allowing the user to view a specific image. The content for the vacancy is displayed largely in the same way as 6-Up number 1, with the tile and description being displayed on the left-hand side, and an overview of what is being offered with the vacancy, appearing on the right-hand side.

**Viewing a Vacancy 3:** The third 6-Up removes the full-width slideshow and replaces it with a Google Map depicting where the current vacancy is located. Below the map is a 50-50 grid layout, with half of the layout being taken up by a slideshow, homeowner avatar and rating, and the second half being taken up by the vacancy title, description and call to action button for the user to apply for the vacancy. The vacancy is then followed by three related vacancies, in a three column layout.

**Viewing a Vacancy 4:** 6-Up number 4 removes the full-width section at the top of the page and instead bumps all of the other information up. The vacancy information is formatted within a 50-50 grid, with half being dedicated to an image gallery, consisting of images from the current vacancy. The other half of the layout is taken up with the vacancy title and description as well as a section containing a brief biography of the homeowner and their rating. The vacancy information is followed by a three column grid containing related vacancies.

**Viewing a Vacancy 5:** 6-Up number 5 reintroduces the full-width slideshow and follows it with another section split up into a two-thirds one-third grid. Within the two-thirds section of the grid is a brief biography of the homeowner, along with their avatar and rating. In the one-third section is an overview of what is being offered with the current vacancy, along with a call to action button for the user to apply for the vacancy. Below this section is another two-thirds one-third section. Within the two-thirds section is the vacancy title and description, and within the one-third section is a Google Map depicting the location of the current vacancy.

**Viewing a Vacancy 6:** The final 6-Up for the viewing a vacancy page removes the need to have an individual page for each vacancy. Instead the vacancy information is presented in a pop-up modal, meaning that the user never has to actually leave their current page. Within the modal a gallery is present with thumbnail images, and beside the gallery is the title and description of the vacancy. Below the description of the vacancy is a call to action button for the user to apply for the vacancy. Following the vacancy information is a full-width section that displays the homeowners avatar, rating and brief biography.

**Viewing a Vacancy 1-Up:** As previously stated, sketch number **1** was chosen to be progressed further in the evolution of the viewing a vacancy layout. The layout remained largely the same, however, the homeowners information section has been replaced in favour of displaying what the vacancy has to offer, and the three column related vacancies area has been increased to allow for four related vacancies.

## Login

**Figure 8** below shows the 6-Ups that were created in the process of visualising the Login page. Whilst there were features from each 6-Up that could be of benefit, it was decided that 6-Up number **5** would be progressed further. The Login page 1-Up can be seen in **figure 9** below:

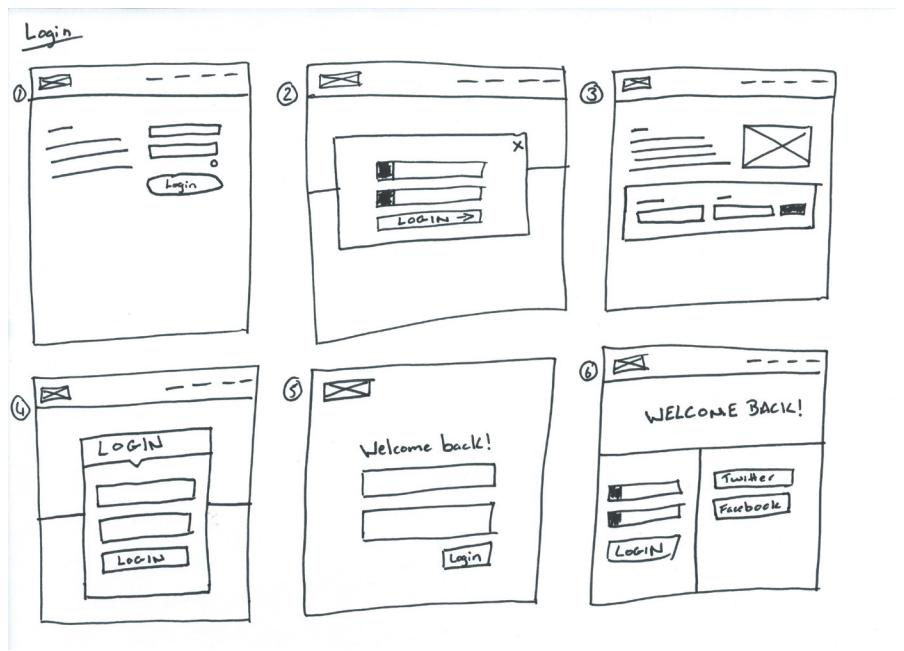


Figure 8: Login 6-Ups

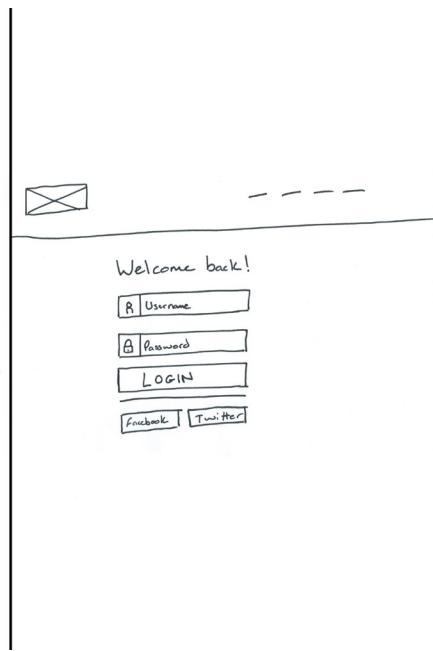


Figure 9: Login 1-Up

**Login 1:** The first login 6-Up is very minimal in its layout. The page will contain a background image that represents the functionality of the application and will be overlaid with the login information and form. The content of the login page will be displayed in a two column 50-50

layout, with information related to the benefits of the application, occupying one half, and the login form occupying the other.

**Login 2:** The second login 6-Up removes the need to have a separate login page, and instead displays the login form in a pop-up modal. Within this modal will be a basic login form, consisting of a username field, password field and login button.

**Login 3:** 6-Up number 3 tries to display more information to the user than its counterparts. The page begins by displaying a two column layout that will display a 50-50 column structure. Half of the 50-50 column will consist of a paragraph of text explaining to the user why using the application is of benefit. In the second half of the column will be an image that depicts the benefits of the application.

**Login 4:** The fourth 6-Up will consist of a background image that takes up 50% of the pages height. The login form will then be centred within the page and overlap the top background image. The actual login form itself will be fairly simple, only display an input field for the username and password, along with a full-width login button.

**Login 5:** 6-Up number 5 again goes down the minimal route. The whole page consists of the login form centred both vertically and horizontally within the page, consisting of a username and password input field, as well as a right-aligned login button. The background to the page could be either an image representing the use of the application, or an illustration depicting the problem that the application sets out to address.

**Login 6:** The final 6-Up brings with it the introduction of social media integration within the application. The top of the login page will be a thin full-width image that will display on top of it a welcome back message. The login form below the main image will be split up into a two column 50-50 grid. One half of the grid will display the simple login form, consisting of a username and password input field, and a login button. The second column will display two social media login buttons that will allow the user to login to the application using their preferred social media account.

**Login 1-Up:** As previously stated, it was decided that 6-Up number 5 would be investigated further as a layout option for the login page of the Stay At Mine application. I think the simplicity of the login page is key as, when a user goes to login to a system or application they are already aware of what the application does and why they are using it. Therefore I found that there was no need to display any information that would ultimately be disregarded by the user.

The login pages consists of a login form centred both horizontally and vertically within the page. Above the login form will be an inviting welcome back message. The login form will then consist of the two required username and password input fields, followed by a full-width login button. Below the traditional login form will be two social media buttons allowing the user to login to the application using one of their preferred social media accounts.

## Create Account

**Figure 10** below shows the 6-Ups that were created in the process of visualising the Create Account page. It was decided that sketch number **3** was going to be taken further in terms of creating a 1-Up and developing a layout for the create account page. The 1-Up sketch can be seen in **figure 11** below:

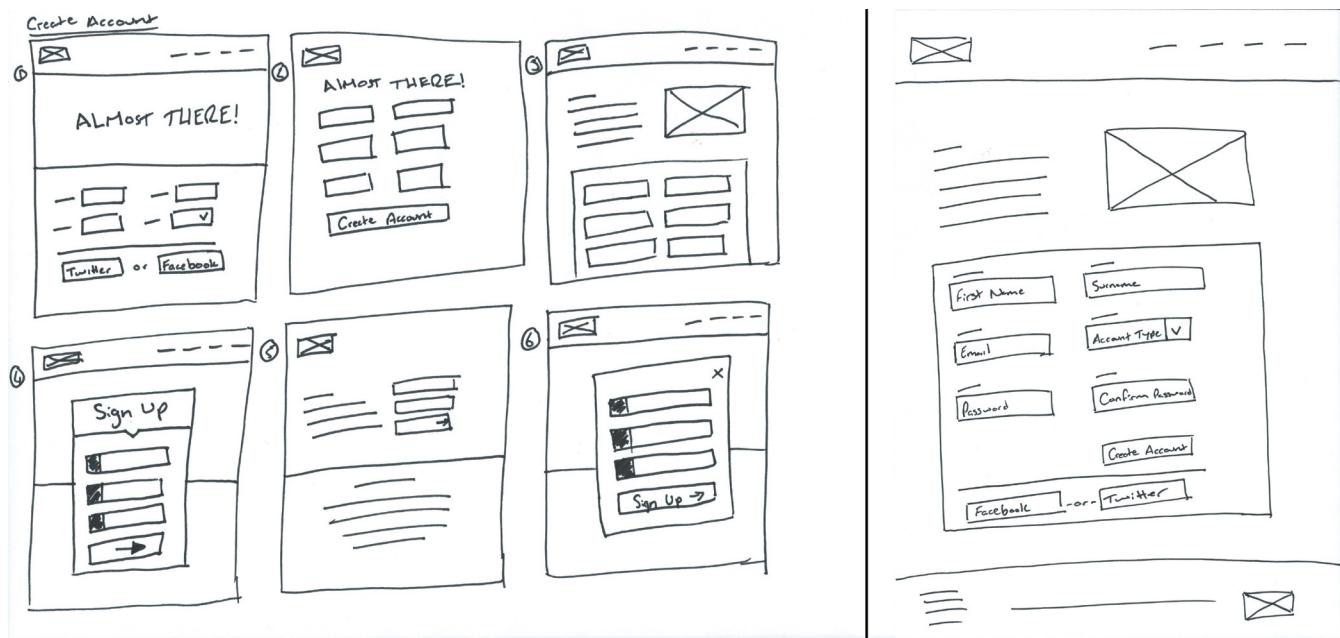


Figure 10: Create Account 6-Ups

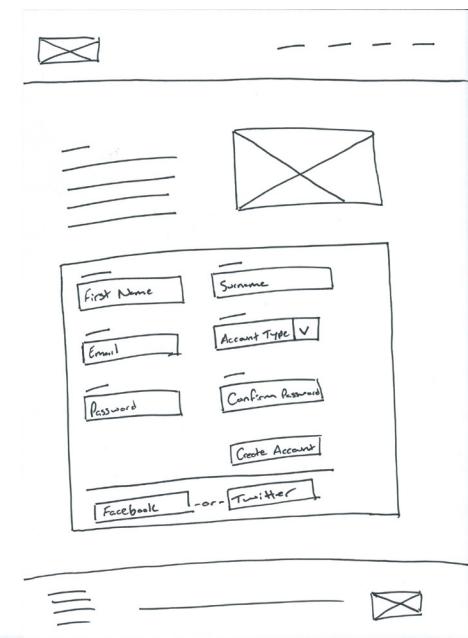


Figure 11: Create Account 1-Up

**Create Account 1:** The first 6-Up tries to contain some consistencies with the login page of the application. The page begins with a slim full-width header image, with the text “Almost There!” to encourage the user that they are almost signed up to use the application. Below the header image is a form that will obtain the basic details needed to create an application. The user will also have the option of creating an account using one of their chosen social media accounts.

**Create Account 2:** 6-Up number 2 attempts and producing a very minimal create account page. The page simply consists of a create account form that is centred both vertically and horizontally. The form incorporates the basic fields required in order to create an account, and has full-width create account button directly after the form. The background to the create account form has the potential to be either an image or illustration addressing the purpose for

signing up to use the application.

**Create Account 3:** The third 6-Up aims at providing the user will a little more additional information as to why they should use the service. Directly below the main menu is a two column grid that will display an informative message on the left hand side and an image depicting the need for the application on the right hand side. Below this will be the create account form, that will be confined inside a box.

**Create Account 4:** 6-Up number 4 follows the same format as one of the login form sketches. The page is made up of a 50% background image that will be overlapped slightly with the create account form.

**Create Account 5:** 6-Up number 5 is a slight variation on sketch number **3**, in that it aims at providing the user with informative information. However, the informative information has been flipped and now appears below the create account form.

**Create Account 6:** The final 6-Up aims at removing the need to have an individual page for the create account form. The form instead is populated inside a pop-up modal window, meaning that the user never actually needs to leave their current page in order to create an account.

**Create Account 1-Up:** It was decided that sketch number 3 would be the most suitable 6-Up to progress further. It was thought that providing some more information to the user on why they should use the service might be crucial in getting some users who may not be sure on signing up to commit to creating an account. therefore the create account page starts off with a two column layout consisting of a paragraph of text describing the problem that the application is aiming to address, followed by an image depicting the problem. This will then be followed by a form consisting of all of the relevant fields required in order to create an account. The create account form itself will be encompassed inside a box, separating it from anything else on the page, and hopefully highlighting it to the user.

## Dashboards

**Figure 12** below shows the 6-Ups that were created in the process of visualising the Create Dashboards page for the Homeowners, Homeless and Charity users of the Stay At Mine application. It was decided that sketch number **1** was going to be taken further in terms of creating a 1-Up and developing a layout for the user dashboards. Whilst the layout for each

user may vary slightly depending on their requirements, the basic layout will remain constant. The 1-Up sketch can be seen in **figure 13** below:

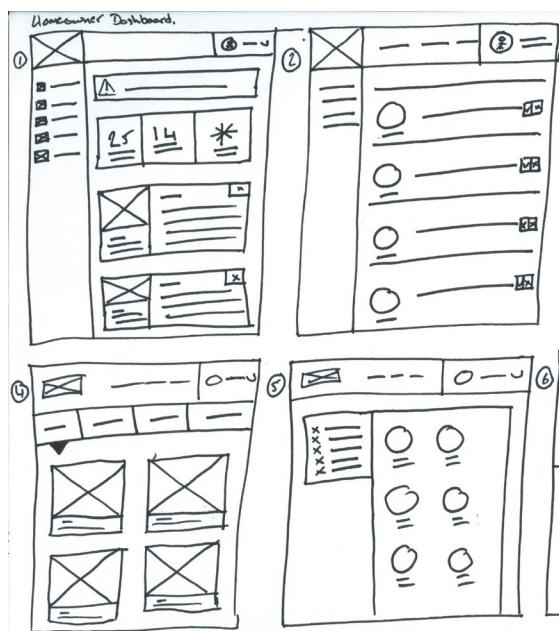


Figure 12: Dashboards 6-Ups

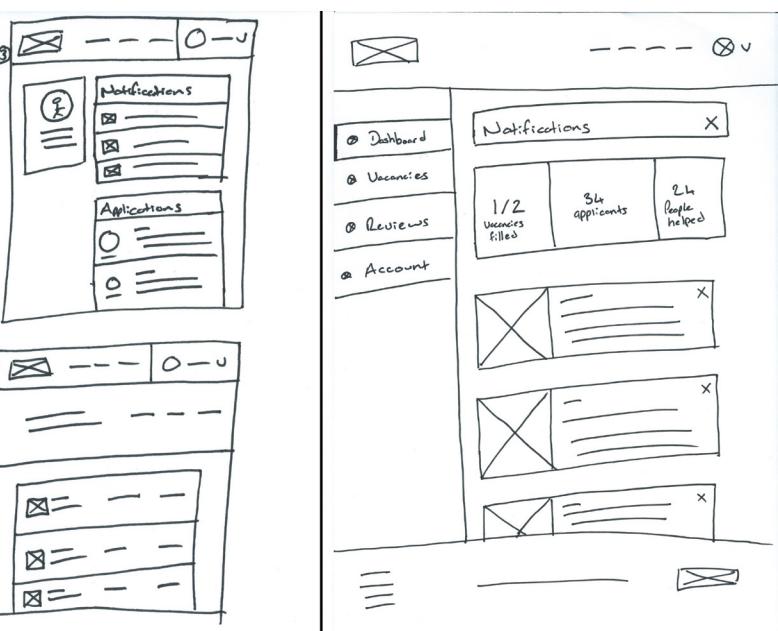


Figure 13: Dashboards 1-Up

**Dashboards 1:** The first 1-Up for the users dashboards eliminates the traditional aspects of the Stay At Mine website. This was done as it will help alert the user that they are now infact logged into the application, and are looking at information that is only relevant to them. The layout of the first dashboard consists of a header that will contain the Stay At Mine logo, in the top left, and the users avatar and dropdown links to logout, at the top right. The main content for the dashborad will be split into two sections. One section will be a thin left hand side sidebar that will contain the relevant links for that user. The second section will then contain the information related to the currently selected menu item. For representation purposes, each 6-Up will be displaying the “Dashboard” page of the dashboards, i.e. the users welcome screen. The top of this section will be any notifications that the user may have, which will be able to be dismissed if they are not needed. This will be followed by an area that will contain a visualisation of data related to the user. This data will be, for example, how many vacancies the user has filled and how many homeless people they've helped etc. Below this section will be, for homeowners, a display of any applications that they have recieived for any of their active vacancies. This area will allow the user to view more information on the applicant, as well as quickly accepting or declining the application. Other pages within the dashboard will follow the same style as this page.

**Dashboards 2:** 6-Up number two follows much the same layout as sketch number 1, however, it takes a more minimal approach. Instead of displaying a lot of information to the user, only important information is now displayed. For example, as displayed, if a homeowner is logged in

they will only be able to see their current applications for their active vacancies. As with sketch 1, the user will be able to either view more information of the applicant or accept or decline the application.

**Dashboards 3:** The third 6-Up removes the left hand side sidebar and instead replaces it with a section providing the user with their current profile information. On the right hand side of the screen, the user is presented with a few different sections, representing each section of the dashboard.

**Dashboards 4:** 6-Up number 4 completely removes the dashboard style theme of the dashboard and instead opts for a view more akin to the rest of the website. The dashboard reintroduces the traditional header that is used throughout the website, and below the header is a full-width section that allows the user to navigate throughout the different sections of the dashboard. As shown in the example sketch, the applications from the homeless users will be displayed in a format similar to the recent vacancies section of the homepage.

**Dashboards 5:** The fifth 6-Up tries to adapt the layout of sketch number 3, however in place of the users profile data is the dashboard navigation. The homeless user applications are then displayed in a two column grid, with the users avatars take precedent, followed by their name and rating.

**Dashboards 6:** The final 6-Up removes the need for the left hand side sidebar and instead provides a full-width section that contains the dashboard navigation. The current dashboard page will appear bigger than the other links and will be underlined. Below the dashboard navigation will be the homeless user applications. These applications will be displayed full-width within a contained box, giving the user an overview of the homeless users name, rating and brief biography.

**Dashboards 1-Up:** It was decided that the 6-Up number 1 would be the best option to expand further, and evolve the layout of the dashboards pages. The page consists of a full-width header, that will contain the site navigation, as well as the users avatar, and a dropdown menu for the user to log out. Below the main header will be a two column layout, one slim left-hand side sidebar, and the main content area. The dashboard sidebar will contain the dashboard navigation links. The navigation links will be wrapped in a container with a top and bottom border, as well as ample margin and padding. This will allow the links to appear as significant options within the page, and not become lost within the content. Each navigation option will

contain an icon that will be relevant to each section. The main content area will then consist of a collapsable notifications section that will display relevant notifications to the user. Below this will be a visualisation of the users statistics, which will be dependent upon each user group. Following that will be, for homeowners, the current homeless user applications, where the user will be able to view more information on the user, or accept or decline their application.

## Charity Display Dashboard

**Figure 14** below shows the 6-Ups that were created in the process of visualising the Charity Dashboard for a Charity using the Stay At Mine application to display within their organisation. It was decided that sketch number **1** was going to be taken further in terms of creating a 1-Up and developing a layout for the Charity dashboard. The 1-Up sketch can be seen in **figure 15** below:

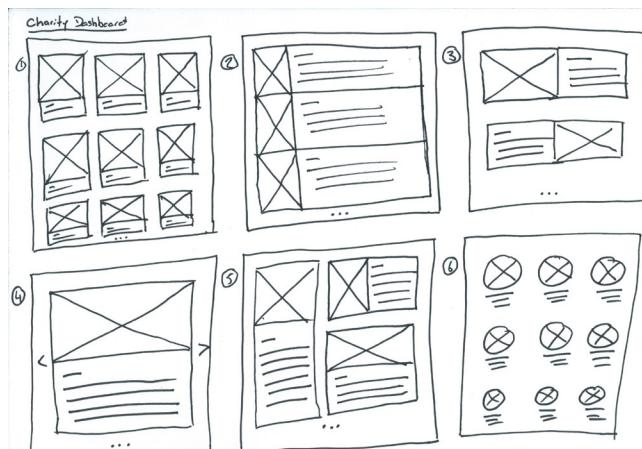


Figure 14: Charity Display Dashboard 6-Ups

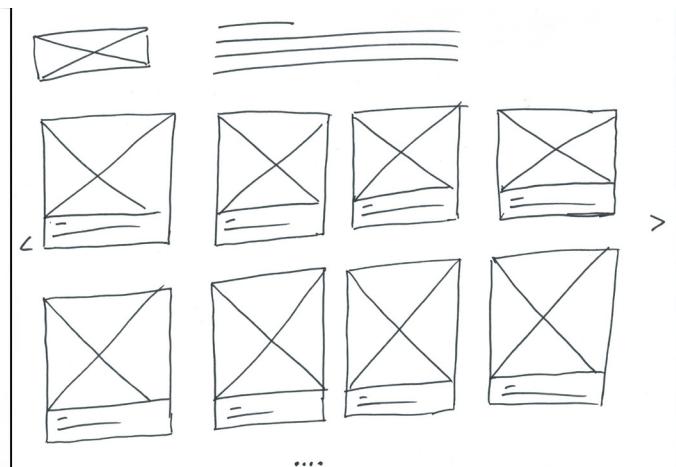


Figure 15: Charity Display Dashboard 1-Up

**Charity Display Dashboard 1:** The first Charity Display Dashboard aims at reusing the same recent vacancies style and layout from the homepage of the website. The vacancies are displayed in a three column layout, with a thumbnail image of the vacancy, followed by the vacancy title and brief description of the vacancy.

**Charity Display Dashboard 2:** 6-Up number 2 tries to alter the layout of the vacancies. Instead of bringing across the same style and layout from the recent vacancies section of the homepage, the vacancies are now displayed full-width within a container. Each vacancy consists of a one-third thumbnail image of the vacancy followed by a two-thirds title and description of the vacancy.

**Charity Display Dashboard 3:** The third 6-Up provides two vacancies and alternates them in their layout. The vacancies are displayed in two column layout with 50% being allocated to the

thumbnail of the current vacancy, and the other 50% being allocated to the title and description of the vacancy.

**Charity Display Dashboard 4:** The fourth 6-Up limits the number of vacancies displayed to just one. This therefore means that there is a lot more screen real estate to display information for the current vacancy, providing the viewer with a lot more information. As there is only one vacancy being shown, a carousel has been introduced to rotate the vacancies every few minutes so that it isn't just the one vacancy all show all of the time.

**Charity Display Dashboard 5:** 6-Up number 5 follows the grid style of the trending locations section of the homepage. In this layout, the vacancies are laid out in a random style grid, with their layout being determined by their location on the page. For example the vacancies on the left will have its thumbnail at the top followed by the vacancies information below, whereas the top right vacancy will display the thumbnail to the left of the vacancy, and the vacancy information to the left. The layout of each vacancy will always either have the thumbnail image on the left, right or top of the vacancy, with the vacancy information always following the thumbnail.

**Charity Display Dashboard 6:** The final 6-Up is an alternate version of the first sketch. However, with this version, the vacancies are not contained within a box. The thumbnail image of the vacancy is displayed within circular container, and is followed by the title and subheading of the vacancy, which is centred below the image.

**Charity Display Dashboard 1-Up:** As previously stated, it was decided that 6-Up number 1 would be the most viable option to go forward with for the 1-Up of the Charity Display Dashboard. It became clear whilst considering the 1-Up that the viewer would need some information on what to do if they wanted to apply for any of the vacancies that can see displayed. Therefore, at the top of the page, beside the Stay At Mine logo, will be a brief paragraph of text, that can be edited by each individual charity, that will inform the viewer of their nexts steps. Below this information will then be the current vacancies, however this has now been increased into four columns, and will rotate on a carousel every few seconds. The layout of the vacancies follows the style and layout of the vacancies on the recent vacancies section of the homepage, bringing through a layer of consistency throughout the application.

## 2.5 Feasibility Testing

In order to make sure the development of the Stay At Mine application was feasible, a functional prototype was required to be developed. The functional prototype focused on the main weak point within the application, with the hope that once the said weak point has been overcome, the development of the rest of the application should be totally feasible.

After evaluating the risks of the project, it was decided that the major weak point of development lay with my inexperience with the chosen framework Laravel. Therefore it was decided, that for my functional prototype, I would develop a very basic application using the Laravel framework, that is able to do some of the basic features required for the Stay At Mine application. These features would include:

- Registering a user
- Ability for a user to login
- Posting data to a database from a form
- Retrieving data from a database
- Displaying retrieved data to the user
- Uploading images to the filesystem
- Inserting image path to the database

### 2.5.1 Risks

Probably the biggest risk in developing the Stay At Mine application would be the decision to use the Laravel framework as the back bone. Having never used the Laravel framework before, there is going to be a massive learning curve in order to get all of the features and functionality required for a sophisticated application.

Asides from the major challenge of having to learn Laravel within a very short period of time, a lot of other smaller challenges also exist for the development of the Stay At Mine application, although these may prove to be less detrimental to the final outcome.

A full table of notable challenges, and an overview of how I plan to overcome each challenge, can be found in **Appendix G**.

## 2.6 Methodology Selection

Throughout the Interactive Multimedia Design course, we are always encouraged to use a variety of different methodologies. However, over the past few years of studying on the course, I have always swayed towards one specific methodology, the majority of the time without even thinking about it. I have always found it easiest to work in a very linear way, and have therefore found the **Waterfall** methodology best fitting towards my own workflow.

The waterfall methodology is a traditional methodology that originates from a manufacturing background, and is very linear in its format. [Seque Technologies, 2015] Each stage within the waterfall represents a distinct stage of the project's development and each stage usually needs to be completed before the next can begin.

In theory, this way of working in a developer environment is great, but in reality you are not granted the luxury of completing one stage of a project before starting the next, especially under the time restrictions of a Final Year IMD Major Project. Therefore, I have decided that the best methodology to undertake for the production of the Stay At Mine application would be the Modified Waterfall methodology. As its name would suggest, this methodology is a variation of the traditional Waterfall methodology. However, whilst the Waterfall methodology requires you to have completed each stage before you can move onto the next, the Modified Waterfall allows you the freedom to revisit different stages of your project to alter aspects.

Allowing for the ability to revisit aspects of the project is going to be critical in helping to develop a well rounded application as I am sure there will be many aspects that will be altered throughout the development phase, with requirements alternating and some becoming redundant.

---

## 3.0 Design

### 3.1 User Experience Evolution

#### 3.1.1 Detailed Wireframes

After previously completing some 6-Up / 1-Up sketches for the layout and user experience of the Stay At Mine application, the next step was to revisit those sketches, and provide more detailed wireframes that would help evolve the user experience. Having originally started out

---

working with pen and paper, for the purposes of the detailed wireframes, it was time to take to Photoshop in order to create a wireframe for each page where the main purpose is the usability of the application, rather than the generation of layout ideas.

Since there were so many wireframes, with the quality needing to be high for legibility, the detailed wireframes can be seen in **Appendix H**.

### 3.1.2 Branding

The branding behind the Stay At Mine application is of grave importance. Many iterations of the Stay At Mine logo have come and gone, along with many different naming conventions. So as to appeal to all sets of users (homeowners, homeless people, and charities), the name of the application had to be welcoming, yet suggestive at what the applications purpose was. The original name of the application was ‘Shelter Me’, however after suggesting this name to several people, it was suggested that this was a little too corporate. The name then went through a few variations, at one point being called ‘Get Sheltered’, then just simply ‘Sheltered’, before finally settling on ‘Stay At Mine’. Some logo variations can be seen in **figure 16** below:



Figure 16: Branding and Naming Iterations

The name ‘**Stay At Mine**’ was chosen as it proved to be a much more friendly naming convention that describes the exact purpose of what the application does, but with a less corporate feel. **Figure 17** below shows the evolution of version 1 of the ‘**Stay At Mine**’ logo.

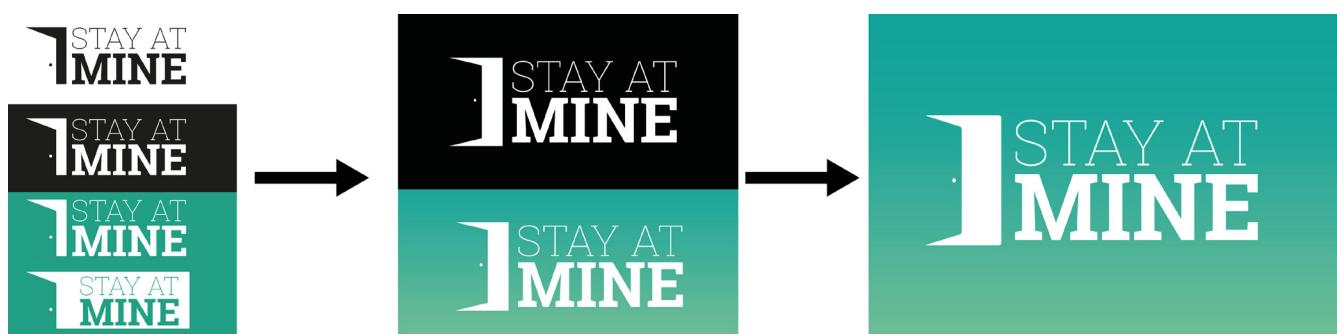


Figure 17: Evolution of version 1 of the Stay At Mine Logo

Whilst I was happy with version 1 of the logo, it was brought to my attention that a few people were unsure of what the logo actually was, with people commenting that the door looked like an 'I'. Therefore, I decided to refine the logo slightly, in order to diminish the confusion. The refined Stay At Mine logo can be seen in **figure 18** below:



Figure 18: Refinement of Stay At Mine logo

### 3.1.3 User Flow

As the Stay At Mine application consists of three user groups (Homeowners, Homeless and Charities) a number of different user flows had to be considered. The user flows for each user group is detailed below:

#### Homeless User Flow

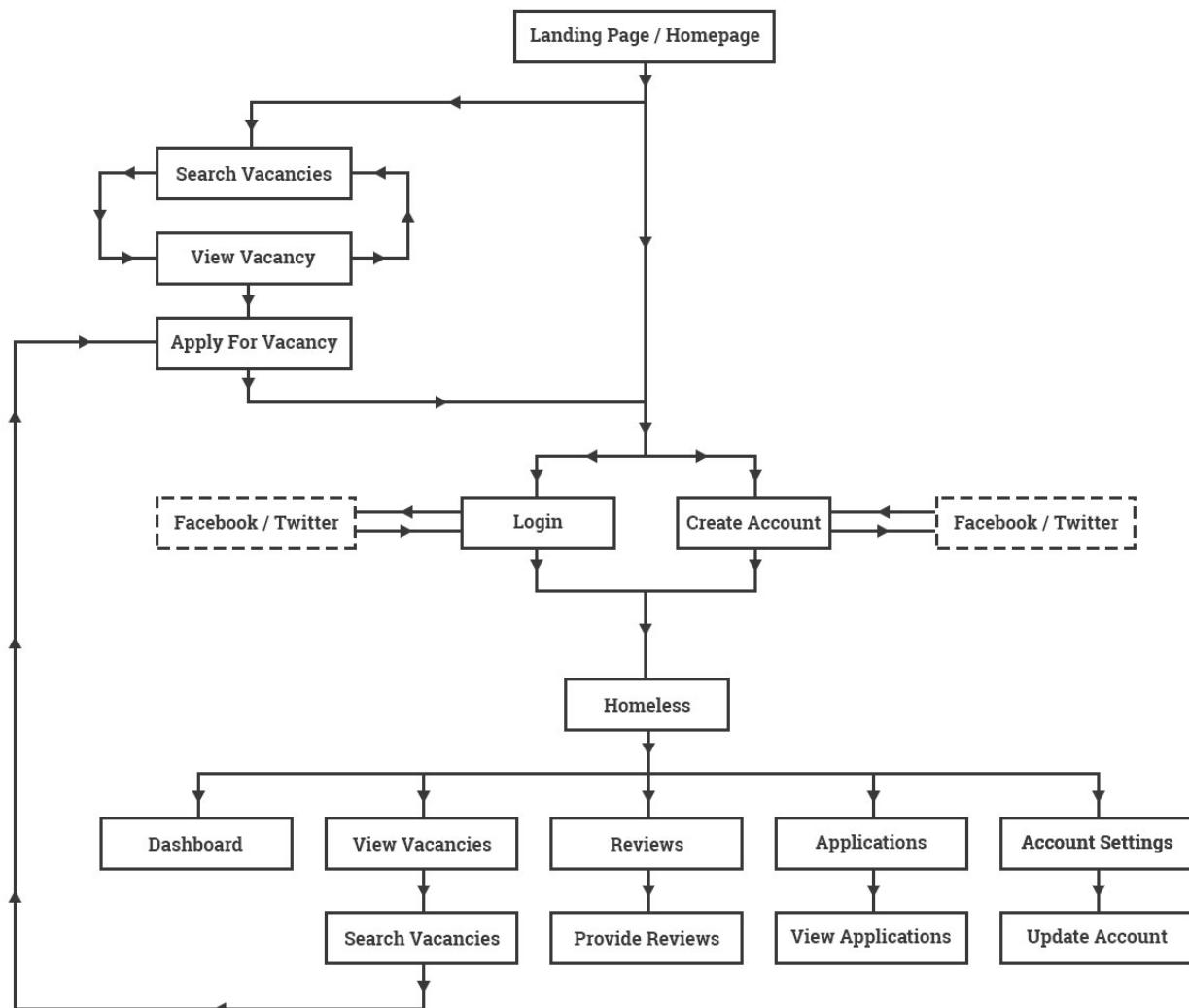


Figure 19: Homeless User Flow Diagram

## Homeowner User Flow

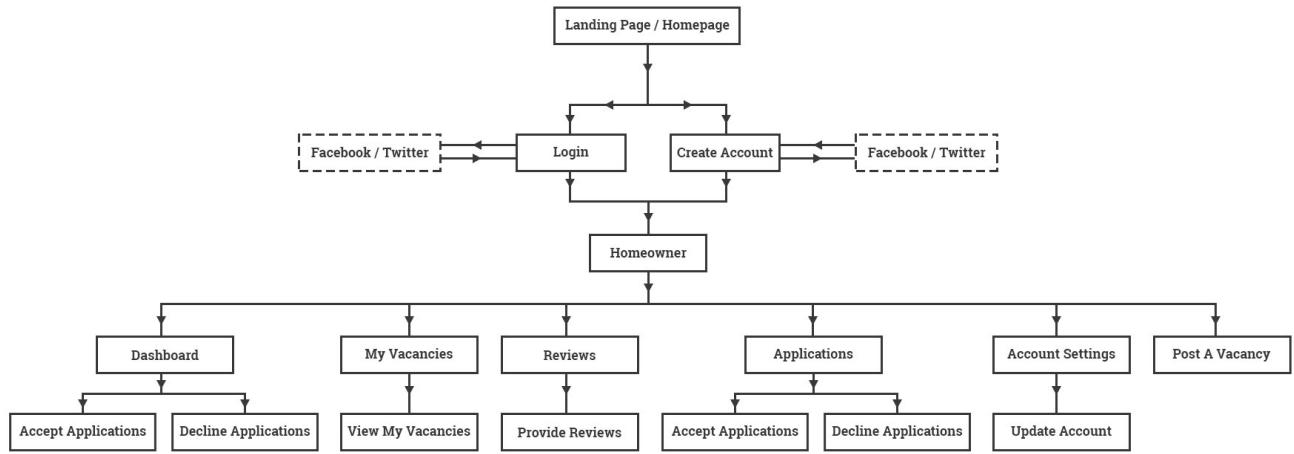


Figure 20: Homeowner User Flow Diagram

## Charity User Flow

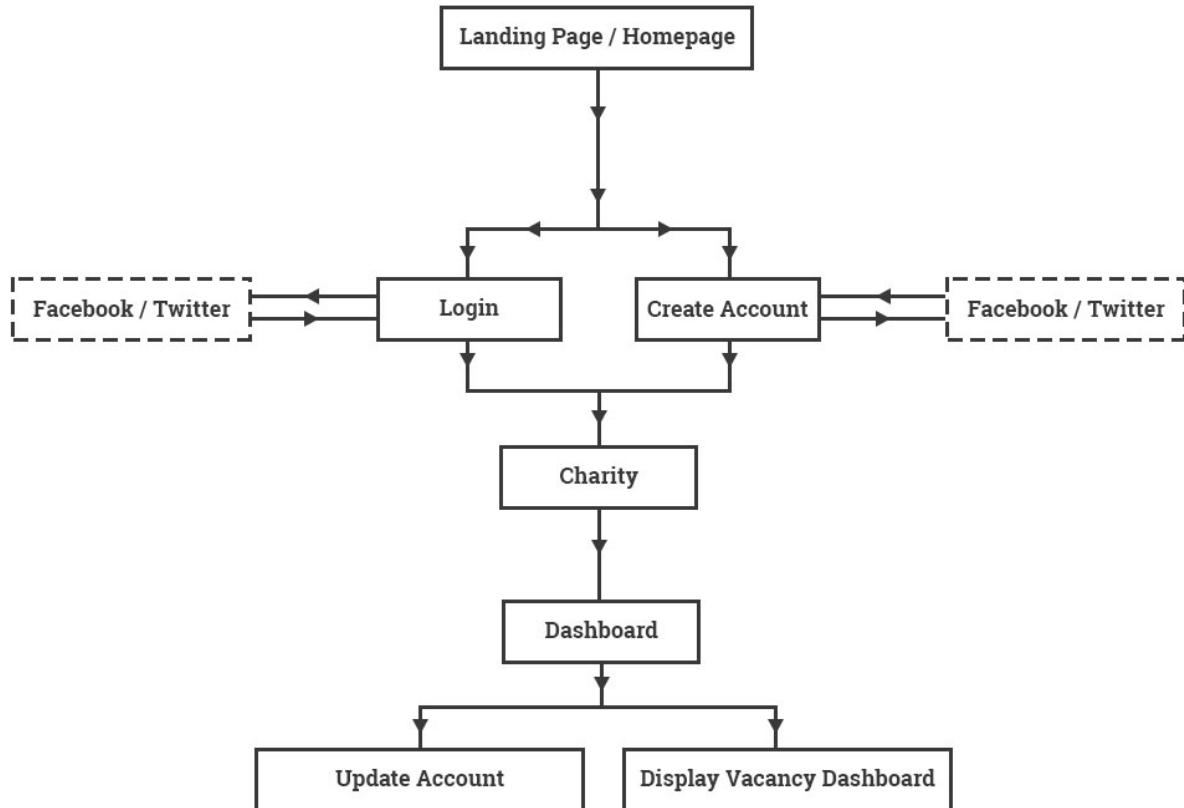
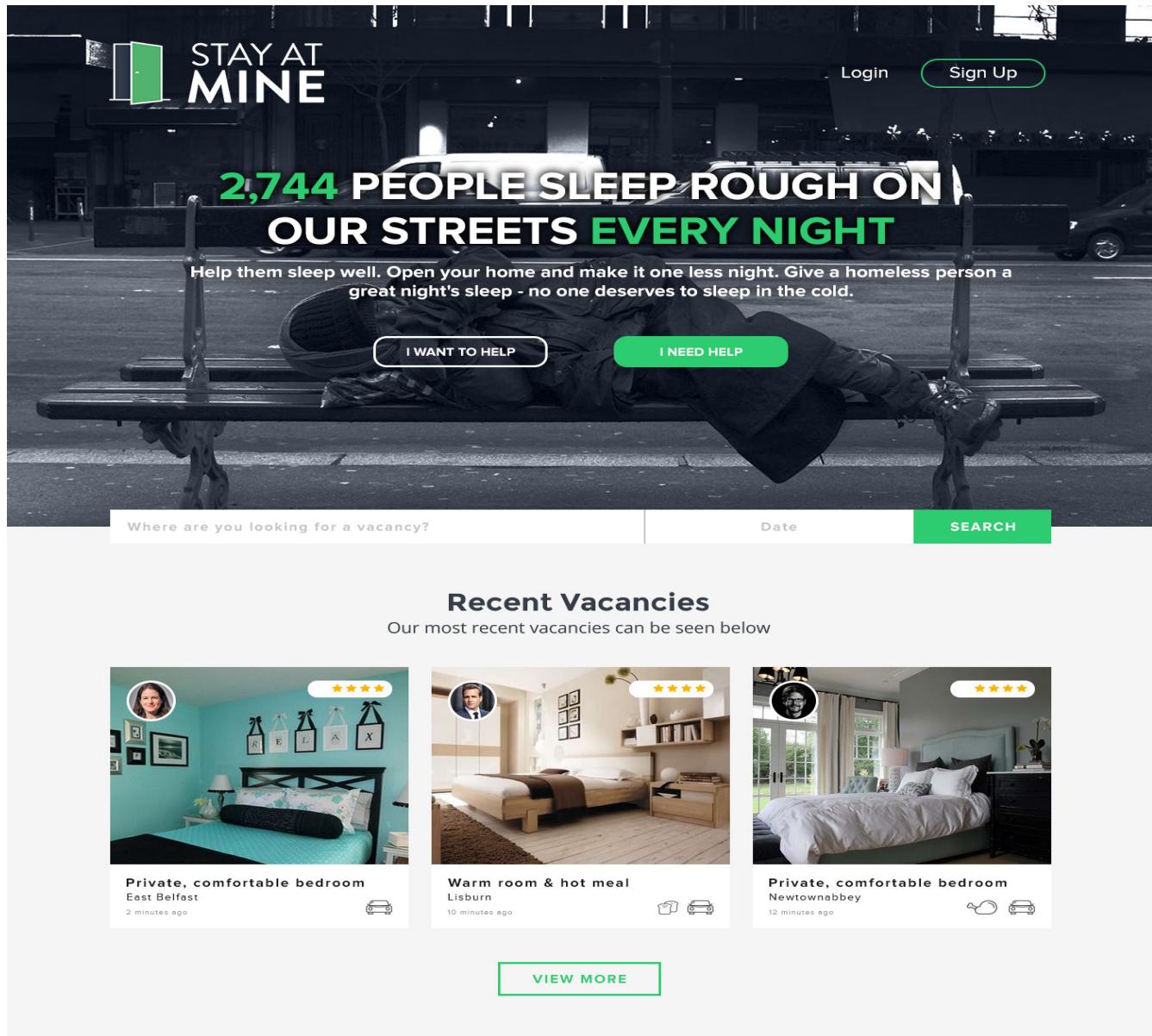


Figure 21: Charity User Flow Diagram

### 3.1.4 Mockups

Once the wireframes were done (as mentioned in 3.1.1), it was time to begin mocking up the wireframes into a more real world example of how the Stay At Mine application would aspire to

look. The process of creating mockups was done in Photoshop, and built upon the wireframes created previously, with only some minor modifications. In figures 22 and 23 below, you can see a sample of the mockups, however due to the volume, and size, of the mockups the full lineup can be seen in **Appendix I**.



## How It Works

Nobody deserves to be sleeping rough, yet thousands do, let's change that!

[HOMEOWNER](#) [HOMELESS](#) [CHARITY](#)

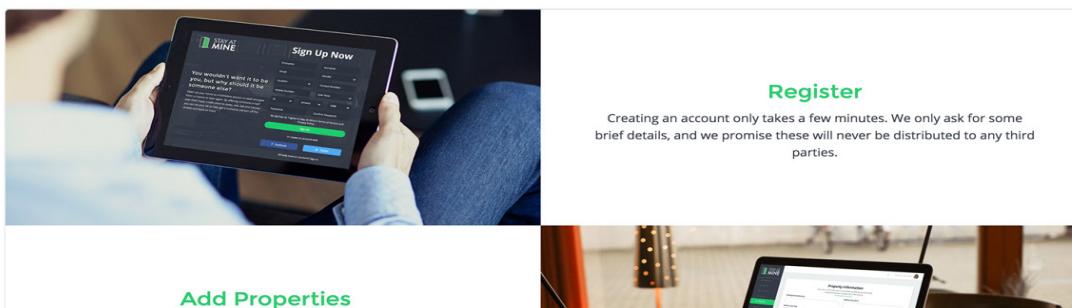


Figure 22: Partial view of Photoshop mockup of Stay At Mine Homepage

The dashboard features a dark header with the logo 'STAY AT MINE' and a green sidebar menu. The main area has a 'Dashboard' title and three cards: one showing '3/4 vacancies filled' with a green circle icon, another showing '34 applications' with a line graph icon, and a third showing 'You've helped 4 people get off the streets!' with a person icon. Below is a table of applications with columns for Applicant, Vacancy, Date, Applied, and Action.

APPLICANT	VACANCY	DATE	APPLIED	ACTION
John Doe East Belfast ★★★	Private, comfortable room, 31 Peachfield Road	Monday, 23rd March	2 days ago	<button>View Profile</button>
Joe Bloggs East Belfast ★★★	Private, comfortable room, 31 Peachfield Road	Monday, 23rd March	3 days ago	<button>View Profile</button>

Figure 23: Photoshop mockup of Stay At Mine Dashboard

After completing the Stay At Mine mockups, it was decided to create a style guide to assist with the development of the application, and increase consistency throughout the site. The style guide can be seen in **figure 24** below:

The style guide includes sections for Typography, Colours, Buttons, and User Avatar.

## TYPOGRAPHY

**EXAMPLE TITLE**  
Font: Montserrat Bold

**Example Header**  
Font: Montserrat Bold

**Example Sub-heading**  
Font: Montserrat Bold

This is an example paragraph of body text. *Ipsum dolor sit amet, sed purto ferri ubique ut, vim cu veritus posidonium. Idque voluptua assueverit cu cum, nec blandit moderatius ex. Ex quot constitutam cum. Volumus deterruisset ex mea. Eu quo facer intellegat, eum ea amet labitur. Te has melius moderatius ullamcorper, sea id doctus concludaturque.*

Font: Open Sans Regular

## COLOURS

#2ECC71	#373E49	#C4C4C4
#ECF0FO	FFFFFF	#222222

## BUTTONS

Version 1

<b>BUTTON</b>	<b>BUTTON</b>
---------------	---------------

Version 2

<b>BUTTON</b>	<b>BUTTON</b>
---------------	---------------

## USER AVATAR

Figure 24: Stay At Mine Style Guide

## 3.2 System Design

Once the User Experience phase had been completed, the next stage, in the development of the Stay At Mine Web Application, is to design how the system is going to function. This will include how data flows through the system and how the user will interact with the system at different stages.

### 3.2.1 Refined Site Map

Having previously put together a rough site map for the Stay At Mine application, before delving into the design of the system, the site map needed to be revisited to determine if any changes needed to be made, or if the site map could be refined in any way. The refined site map can be seen in **figure 25** below:

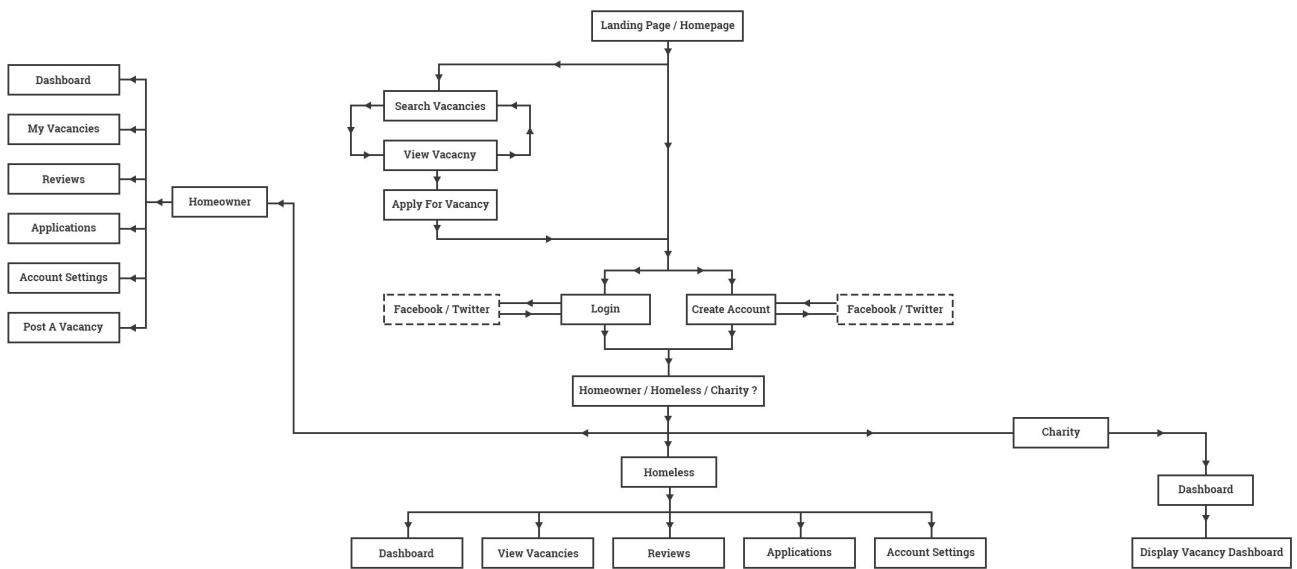


Figure 25: Refined Site Map

### 3.2.2 Client-Server Model

The client-server model, which can be seen in **figure 26** below, outlines the various technologies that come together to make up the Stay At Mine application. According to techopedia, a client server model is a “distributed communications framework of network processes among service requestors, clients and service providers”. [Techopedia, 2015] Therefore, the below diagram displays the process in which information is transferred between the server (where the application is hosted), and the client (where the user is viewing the

application i.e. iMac/iPhone using Safari or PC/Samsung Galaxy using Chrome), and the technologies that make up those communications i.e. HTML, CSS, JS and Laravel etc. The client and server are then connected through a network, in this case the internet.

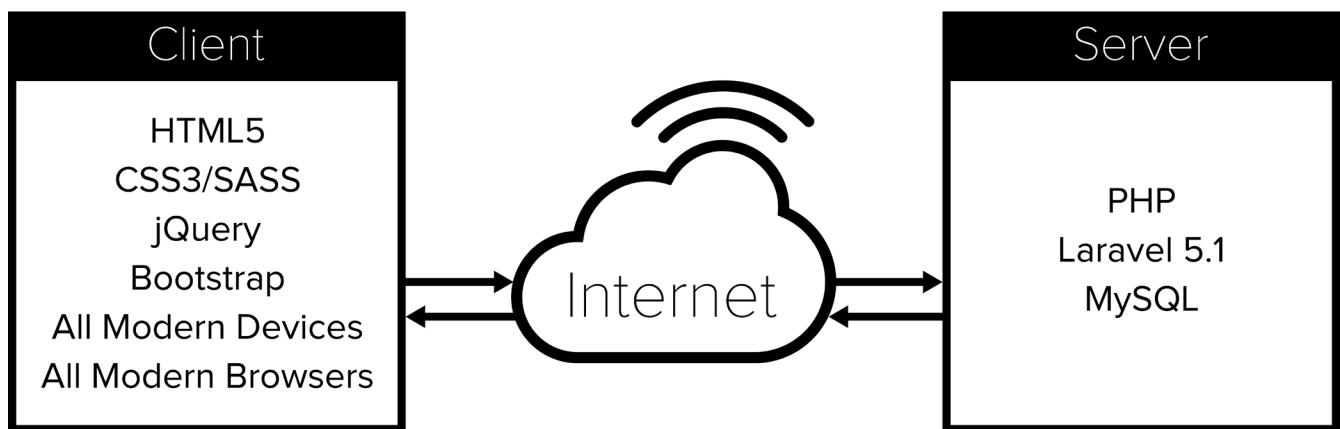


Figure 26: Client Server Model Diagram

### 3.3 Logic Design

As the Stay At Mine application will be built with the Laravel framework, it will incorporate the Model View Controller. A diagram of Laravels Model View Controller can be seen below:

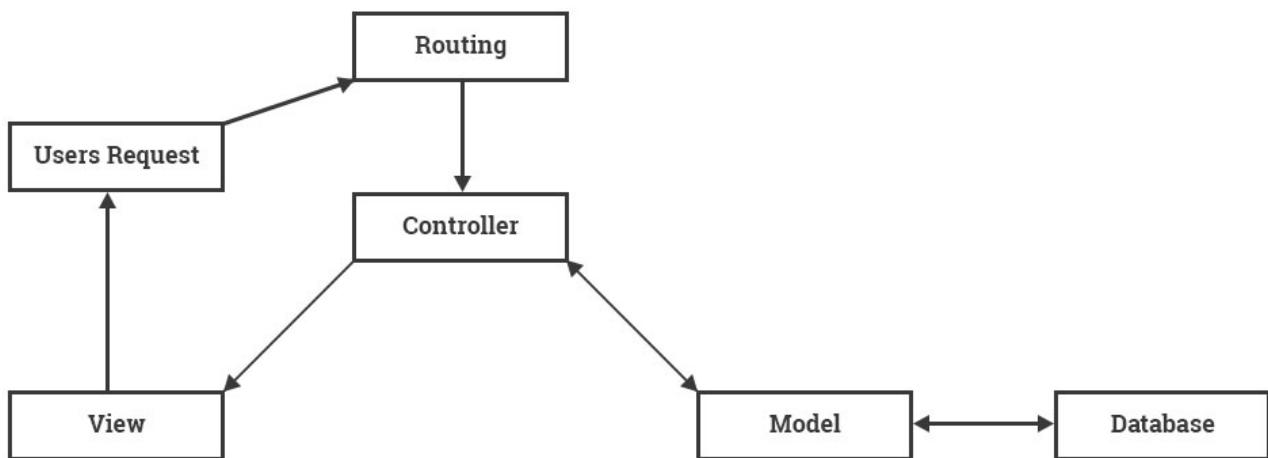


Figure 27: Laravel Model View Control Diagram

# 3.4 Data Design

## 3.4.1 Database Design

The database design for the Stay At Mine application can be seen in the diagram below:

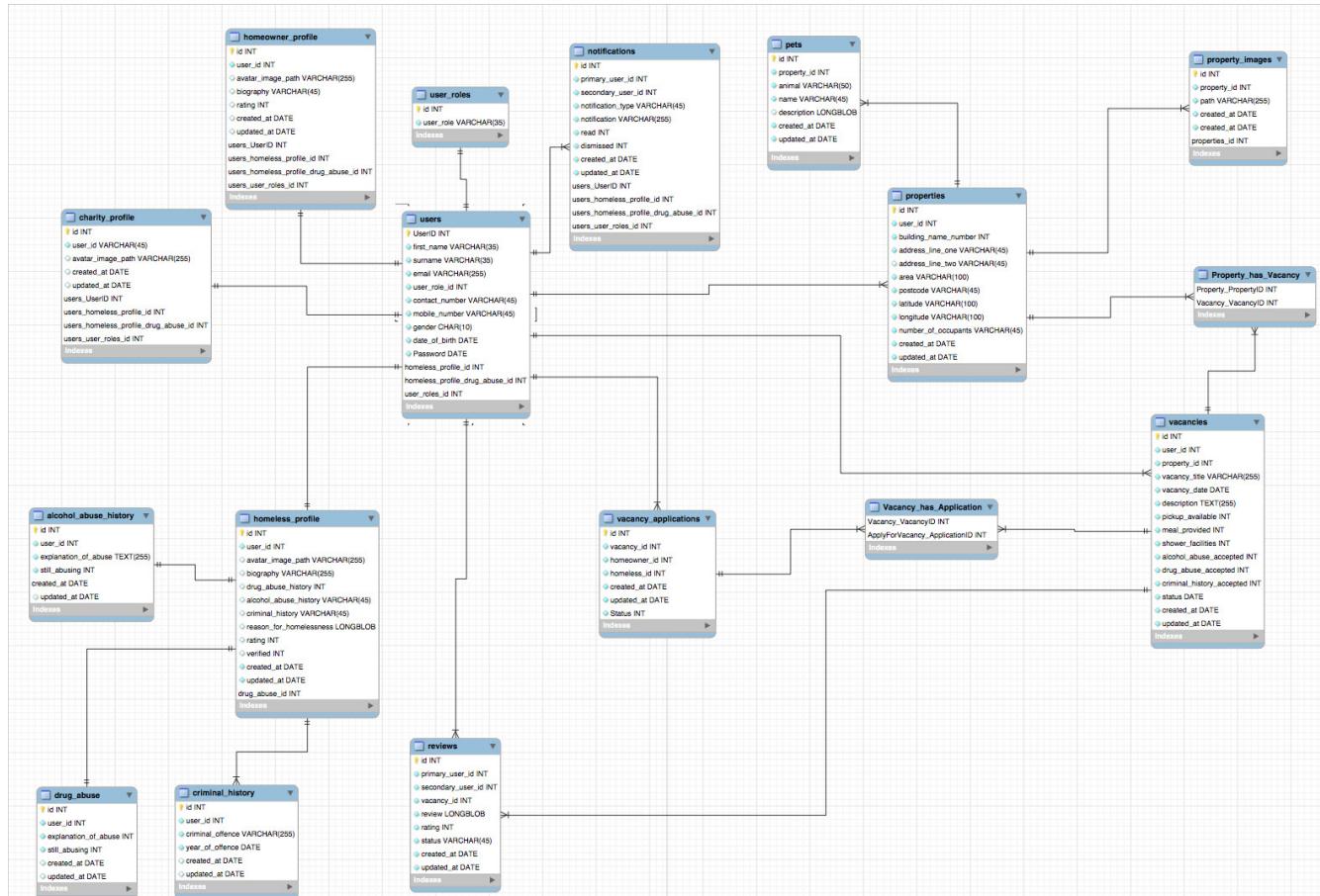


Figure 28: Relational Database Design Diagram

# 4.0 Implementation

## 4.1 Technology/Tool Review

Having designed how the Stay At Mine application should look, and how the data should flow through the application, it's then time to look at the best way to develop it. There are many different ways to develop each aspect of an application, and each of these ways have their own pros and cons. For me to come up with a definitive and informed solution for the Stay At Mine application, I therefore first had to delve into each area, before I could come to a conclusion on how I would go about building the Stay At Mine application.

## **4.1.1 Server-Side Technologies**

### **PHP**

According to W3Techs, PHP is one of the most widely used server side scripting languages available, with over 81% of all known websites using it as their server scripting language of choice. Having been around since 1995, PHP has gained massive support from developers, and given that it is open source, thousands of modules already exist to give solutions to a whole host of complex development problems. [Udemy, 2015] PHP is capable of interacting with a whole host of different databases, making it an excellent choice when developing a data intensive dynamic website. [wiseGEEK, 2015]

However, a major downside of PHP is its error handling. Due to PHP's development history, and support for dated functions, it can take a lot longer to figure out why a particular piece of code is not functioning. [wiseGEEK, 2015] Another disadvantage of using PHP over its competitors, is that it tends to perform at a slower pace. This is due to the fact that PHP is an interpreted language, meaning that, whilst the number of machines the code can run on are significantly higher, the code has to be reduced machine instructions at run-time, and therefore causes the code to perform at a slower pace. [Vanguard Software Corporation, 2015]

### **Java**

Java is a general purpose programming language first released by Sun Microsystems in 1995. Originating from an unsuccessful programming language called Oak, which was developed for use in handheld devices and set-top boxes, Java was developed to take advantage of the ever-growing World Wide Web. [Webopedia, 2015] A key benefit of developing with Java comes down to its developers core value “Write once, run anywhere”. This means that once an application has been developed using Java, it will be able to be used anywhere. [Java in a Nutshell, 2005]

However, whilst the core value behind Java is that it can be operated on any system, this is strictly true. In order for a Java application to be able to run on a system, that system must first have Java installed, otherwise the application will not be able to run. Another downside of using Java to develop a dynamic web application, is that in doing so you are ultimately saying goodbye to all of the worlds iPhone users as potentially users of your application. Steve Jobs

is quoted as saying “Java’s not worth building in. Nobody uses Java anymore. It’s this big heavyweight ball and chain.”, thus eliminating the use of Java in all iPhones past and present.  
[iPhoneFAQ, 2015]

## Ruby

Ruby is an object-orientated, general purpose programming language originating from Japan in the early 1990’s. According to the Ruby website, it “focuses on simplicity and productivity”, with “an elegant syntax that is natural to read and easy to write”. A major advantage of using Ruby as a programming language is that you can put together a basic web application very quickly, in a format that is easier to understand than many of its competitors.

However, whilst you may be able to build an application quickly, scaling that application as it becomes popular can be very difficult, and is thus why many popular websites that were first built on Ruby, such as Twitter, have now moved on to bigger and better things.

## Python

Python is an interpreted, object-orientated programming language that first appeared in 1991. The key benefit of using Python is how closely it resembles the English language, making it highly readable. Given the length of time it has been around, and the fact that it is open source, thousands of modules already exist, in the “CheeseShop”, for developers to download and use in their own systems.

A notable issue with Python for web development is that it is generally extremely slow in comparison to its rivals, and in a time when we expect a website to load in a matter of milliseconds, regardless of the content, choosing a slow server-side scripting language could be detrimental to the whole application. Another downside to Python, is its lack of documentation. Whilst the language claims to be highly readable, this is only an advantage if the material is there to allow developers to learn the language. With there being nearly double the amount of documentation out there for just PHP alone, never mind Perl and Java, it would be difficult to adopt a language where support seems limited.

## **4.1.2 Client-Side Technologies**

### **Google Maps**

Google Maps is probably the most widely used mapping API available, and can be found on the majority of websites. Launching in February of 2005, after being acquired by Google, the application was quickly developed into an API, allowing developers to place their own Google Maps onto a website. [API Evangelist, 2011] Whilst the API can look quite daunting at first glance, once you become familiar with it, it is extremely easy to use and utilise. For the purposes of the Stay At Mine application, only fairly standard functions would need to be developed, such as placing a marker pin on a map, and these can all be accomplished very easily with the Google Maps API.

### **Mapbox**

Mapbox is, according to its own website, the next generation of map design, and from their website, the service does look very impressive indeed. Mapbox is built on the basis of OpenStreetMap data and is said to have customisation at its core. [Juggernaut, 2015] With it, Mapbox offers a very easy to use interface and map customisation tool, TileMill, allowing for the extremely easy customisation of maps.

## **4.1.3 Frameworks**

### **Bootstrap**

Bootstrap is one of the most widely used frameworks for developing responsive websites, and was originally developed by a few Twitter employees who sought to bring some form of consistency within their internal tools. The Bootstrap framework allows for rapid development of web applications, that have a consistent look and feel on all devices. The framework can be customised to meet any developer's needs, including the pieces of the puzzle that you need, and leaving out the pieces that you don't. Bootstrap takes care of all of the boring aspects of developing a website, such as styling forms, alerts and pagination. [HTMLCenter, 2015]

However, Bootstrap isn't without its flaws. The framework has been criticised for being "bloated" and not following the best practices of web development. The end result is "crammed full of classes", the "HTML is no longer semantic" and "the presentation is no longer

separate from the content". [ZingDesign, 2015] Perhaps the most harrowing downside of using the Bootstrap framework is the emergence of a whole host of virtually identical websites. Whilst you may be able to develop a website at lightening speed, in reality it may end up looking like every other Bootstrap powered website.

## **Foundation**

Foundation is a responsive front-end framework for developing mobile-first web sites. Built primarily in HTML and CSS, Foundation makes use of modern web technologies whilst also seamlessly working in dated browsers. Like with Bootstrap, Foundation can be completely customised to each individual developers needs, allowing you to cut out the code that you don't need.

There is very little comparison of Bootstrap and Foundation, and both frameworks seem to be very effective at what they do, which is largely the exact same thing. It would seem that the only advantage of using one framework over the other would be personally preference.

## **Laravel**

Laravel is a free, open source, PHP framework first created by Taylor Otwell in 2011, and has quickly grown to become one of the most popular PHP frameworks available. Laravel follows the Model-View-Controller (MVC) architectural pattern "which enforces a separation between 'business logic' from the input and presentation logic associated with a graphical user interface (GUI)". [Laravel Book, 2015] The Laravel framework can be easily extended using Composer, and is highly customisable based on the developers personal preference. The blade templating engine that Laravel uses, provides a great foundation for fast and extendable PHP production. Another massive bonus for using the Laravel framework would be down to the Laracasts series, created by Jeffrey Way, that does an absolutely brilliant job at teaching various aspects of the Laravel framework, making it accessible to almost anyone.

However, as with all aspects of web development, no framework can come without its burdens. As Laravel is still a relatively new framework, the external documentation is quite limited, and thus finding an answer to a coding issue could prove challenging.

## **Codelgniter**

Codelgniter is a very powerful PHP framework that is “built for developers who need a simple and elegant toolkit to create full-featured web applications.” [Codelgniter, 2015] Codelgniter is said to be extremely easy to set up and use, making it highly accessible for newcomers. Codelgniter has a massive community and therefore a great support network for when things go wrong, as they ultimately will.

However, whilst Codelgniter seemed to have a massive fan base at one point, it would seem that this community is slowly slipping away as its users move on toward other more progressive frameworks such as Laravel. It would seem that Codelgniter is a great basecamp for rapid PHP development, however if you’re wanting to build a web application that has scalability in the modern era, a framework such as Laravel is more fitting.

### **4.1.4 Databases**

#### **MySQL**

MySQL is an open source Relational Database Management System (RDMS) that uses Structured Query Language (SQL). [Site Ground, 2015] It is one of the most widely used SQL languages on the web due to its ease of use, reliability, and scalability. A MySQL database can be used for an application that contains 1MB of data, as well as an application that contains TeraBytes of data.

However, Digital Ocean states that due to the way “certain functionality gets handled with MySQL renders it a little-less reliable compared to some other RDBMSs”. [Digital Ocean, 2015] Another flaw with MySQL is that its development is not community driven, and is therefore lagging behind. Oracle, the company now behind MySQL, doesn’t accept developer patches, and has no way for developers to communicate with them in order to support development.

#### **SQLite**

SQLite is a Relational Database Management System (RDMS) that is embedded into the users programme, rather than on a server. The entire database consists of just a single file that is stored on a disk, and therefore makes it highly portable.

However, as the database is embedded in the end application, it doesn't allow for multiple users, and therefore isn't applicable to a multi-user application.

## **PostgreSQL**

PostgreSQL is an Object-Relational Database Management System (ORDMS) that focuses on being standards compliment and extensible. [Digital Ocean, 2015] PostgreSQL brings with it a vast community of skilled developers, with an extensive range of documentation that could prove to be critical when developing a large scale application.

However, perhaps the biggest disadvantage to using PostgreSQL is the lack of hosting companies that support it, in comparison to its rivals. This would therefore limit the ability to choose a trustworthy and reliable hosting partner, which can be crucial for the stability of an application.

### **4.1.5 Other**

#### **SASS (Syntactically Awesome StyleSheets)**

SASS or Syntactically Awesome StyleSheets, to give it its proper name, is, according to its website "CSS with superpowers". [SASS LANG, 2015] SASS is basically a language that is used to produce Cascading StyleSheets for a website. Writing a SASS file allows you to nest all of your elements, include mixins, use variables, and ultimately combine them all together to produce one final distribution CSS file. A key benefit of using SASS over its rivals would be down to the fact the language still ultimately looks like CSS, just laid out in a slightly different way.

However, using the SASS framework does require a little bit of extra set-up, which can seem like a bit of a pain when with CSS you can start styling a website in seconds.

#### **Less**

Less is another CSS pre-processor that extends the CSS language. [Less CSS, 2015] Inspired by the SASS language, less came around a few years later in 2009. Just like with SASS, the Less framework still looks a lot like CSS, making it extremely easy to understand and pick up.

However, whilst there is a lot of community support and documentation for Less, it lacks behind SASS when it comes to debugging, meaning that whenever you run into an issue, you're left to try and work it out line by line. [Web Designer Mag, 2015]

## **GULP**

GULP is a JavaScript task runner that, according to Brandon Clapp, lets you automate tasks such as:

- Bundling and minifying libraries and stylesheets.
- Refreshing your browser when you save a file.
- Quickly running unit tests
- Running code analysis
- Less/Sass to CSS compilation
- Copying modified files to an output directory

[Brandon Clapp, 2015]

By automating the above tasks, you can save a lot of time doing menial, yet beneficial, everyday developer tasks. No developer would enjoy minify CSS or JS by hand, and that's where GULP comes in, allowing you to perform all of those menial tasks on the fly.

It would appear that the only disadvantage to using GULP would be its initial set up time, and learning curve. However, once you create a gulp file that undertakes all of the menial tasks that you need it to, a lot of time will be saved throughout the process of developing your application, and ultimately can save you a lot of time in the long run.

## **GRUNT**

GRUNT is another JavaScript task runner that allows developers to do pretty much the exact same thing that GULP does, just in a slightly different way. However, GRUNT has been criticised for its harder to understand syntaxes and laborious configuration, in comparison to GULP. [100PercentJS, 2015]

## **4.2 Technology/Tool Selection**

Once I had looked into various different approaches for building a web application, I then had to make a decision on what technologies I was going to use to develop the Stay At Mine application.

### **4.2.1 Server-Side Technologies**

#### **PHP**

After investigating a number of different solutions for a server-side scripting language, I decided that PHP would be the best option for the Stay At Mine application. PHP is used on over 81% of all known dynamic websites, and therefore suggests to me that it must be doing something right. Its ability to interact with a number of different database solutions, as well as its high popularity amongst developers, allows for the development of an application that has the scope of scaling without too much complications.

Potentially the most crucial aspect of choosing PHP over its rivals, comes down to my experience with the language. Having been using the language since being taught it in the second year of University, I have come to thoroughly enjoy using the language daily. My familiarity with the language, combined with the time frame of the development of the application, takes away the learning curve that would arise from choosing another, unfamiliar, language.

### **4.2.2 Client-Side Technologies**

#### **HTML**

HTML or **H**yper **T**ext **M**arkup **L**anguage, to give it its full name, is an authoring language that is used to write webpages for the distribution on the world wide web. [Webopedia, 2015] Whilst other languages exist for writing a webpage, it was decided that HTML5 would be used in the development of the Stay At Mine application. The decision for this didn't need to be investigated at all, HTML5 is the most up-to-date of the HTML languages and it brings with it a whole host of new functionality.

## **CSS**

CSS or **Cascading Style Sheets**, is to a webpage styling what HTML is to a webpage markup. According to the specification, CSS is the language that is used for describing the presentation of a webpage, i.e. fonts, colours and layout. [W3, 2015] CSS will be incorporated into the development of the Stay At Mine application to aid in transferring the Photoshop mockups into a functioning, designed, application.

## **JS/jQuery**

JavaScript is a lightweight, crossplatform, object-orientated scripting language that aims at adding a form of interactivity to a webpage. [Mozilla Foundation, 2015] First appearing in 1995, you would be extremely hard pushed to find a website that doesn't employ some form of the language. In order to make the functionality of my application more enjoyable for the user, and to assist was some of its functionality, JavaScript will be incorporated to allow for access and manipulation of the Document Object Model.

However, it is unlikely that very much vanilla JavaScript will be written within the application. Instead, a JavaScript library, jQuery, will be included that will simplify the whole experience of writing JavaScript. According to W3Schools, “jQuery is a lightweight, ‘write less, do more’, JavaScript library”, where the sole purpose is to make writing JavaScript easier. [W3Schools, 2015]

## **Google Maps API**

When it came to deciding what mapping api I was going to integrate into the Stay At Mine application, the choice wasn't overly difficult. Whilst Mapbox looks like a brilliant platform for customising maps, there isn't really too much map customisation needed for the application, as I don't think it will add any value to the end users experience. I feel that for this reason, the learning curve of learning a new platform, to essentially reach the same outcome, was unjustified. I have been using the Google Maps API for a number of years now, and feel that I am more than comfortable with the API to be able to incorporate all of the features needed for the Stay At Mine application.

## **4.2.3 Frameworks**

### **Laravel**

Once I had decided that PHP was the technology that I wanted to base the Stay At Mine application on, I then had to make a decision for what, if any, PHP framework would best suit the application. To be honest, this wasn't a very difficult decision. It would seem that at one point in time, CodeIgniter would have been the out and out winner, however, I think that, with the emergence of Laravel, CodeIgniter has very much lagged behind.

The Laravel framework seems like a brilliant base for pretty much any new application. The level of detail in the framework alone, with each doc block line having three characters less on every single line, for example, shows the amount of consideration that has gone into the framework. Laravel brings with it the Model-View-Controller method that seems to be the backbone of the majority of emerging applications, which highlights the fact that this must be the way forward within the developer community.

Aside from the actual framework itself, the Laracasts series by Jeffrey Way was probably the clinching point in terms of me choosing the Laravel framework. The online series is constantly updated and covers everything from a beginner setting up Laravel Homestead, to an intermediate developer submitting data to a MySQL/SQLite database, to an advanced developer processing payments using Stripe. Therefore, whilst Laravel is a relatively new framework, with limited documentation across the web, I think that there is plenty of relevant information in terms of building the Stay At Mine application.

### **Bootstrap**

After looking into both the Bootstrap and Foundation frameworks, as well as some dated frameworks such as Gumby, I decided that the best approach for the Stay At Mine application would be Bootstrap. In looking into Bootstrap and Foundation, it seemed that there is a lot of love out there for both of them, with Bootstrap perhaps being marginally in the lead. Therefore, it seemed that the choice between the two came down to largely personal preference. Having been using Bootstrap for a few years now, both at University and on my years Placement, I figured that it would be the most logical choice as I am already pretty familiar with how it works. Whilst a major drawback to the Bootstrap framework is that a vast majority of emerging Bootstrap websites look quite similar, I feel that with the right amount of customisation, there is

no need to worry about the Stay At Mine application looking too similar to another product.

## 4.2.4 Databases

### MySQL

Once I had taken a look at various databases, it became apparent that there only really was one option for the Stay At Mine application. Whilst SQLite seems like a great tool in terms of creating an application quickly, it doesn't allow for having multiple users, and offers little to no scalability. The major drawback to using PostgreSQL is the lack of hosting companies that support it, which I wouldn't want to affect the end result of the Stay At Mine application by hosting it on an unreliable hosting service. Therefore, since MySQL has been the backbone to every web application I have developed, it seems like the only real contender. Whilst it does have some drawbacks, such as no developer driven updates, I truly feel that the benefits of using MySQL far out weight the drawbacks.

## 4.2.5 Other

### SASS (Syntactically Awesome StyleSheets)

The decision between using Less and SASS for Stay At Mine was a difficult one. I hadn't really used CSS pre-processors too much, and therefore didn't really have any preconceived ideas of what is the best one. Having done my research into both Less and SASS and tinkered about with both of them, the final decision came down to the pre-processor that I felt most comfortable with, after all I would have to use it continuously in the upcoming months. I decided that due to the way SASS handles variables and the way it looks slightly more similar to normal CSS, that it would be the most beneficial pre-processor to use for the Stay At Mine application.

### GULP

Whilst GULP and GRUNT promise to do pretty much the same thing, it wasn't an overly difficult task picking between the two. Having used GULP briefly as part of the COM601 module, I found it extremely easy to understand and get to grips with, in comparison to GRUNT which look less beginner friendly. It would seem that even prominent members within the GRUNT community are moving away from the JavaScript task runner in favour of GULP, which

highlights the fact that GULP may be the better option. Ultimately the end result will be the same no matter what task runner you choose, however, the journey to get there with GULP seems a lot more pleasant than GRUNT.

## **4.3 Technology/Tool Use**

Once the technologies had been decided upon, the next stage was therefore to begin implementing these technologies, and start developing the Stay At Mine application.

### **4.3.1 Client-Side Technologies**

The client-side technologies used for Stay At Mine were extremely familiar, and form the basis of the Interactive Multimedia Design degree. Therefore the implementation of these technologies was fairly straight forward. That being said, through implementing technologies such as HTML, jQuery and the Google Maps API, a better understanding of each technology is now held.

### **4.3.2 Server-Side Technologies**

Having chosen PHP as my server scripting language, and Laravel as the PHP framework, it was the implementation of the server-side technologies that required the most effort and learning. Being fairly comfortable with PHP as a development language, it was only logical that it would be my development language of choice. However, the Laravel framework was completely new to me, and required an extensive amount of research and learning before I could delve into developing the Stay At Mine application with it.

Laracasts, an online tutorial service that claims to be “Netflix for developers”, was absolutely crucial in assisting my learning of the Laravel framework. The Laravel 5 fundamentals series helped me set up a Laravel application on my local machine, with the use of Homestead, ensuring that the correct file structure and good practices were followed from the outset of development.

The Model View Controller aspect of Laravel allowed for the simplification of data entry and database relationships, ensuring that relationships were correctly formed from the beginning. Following the Model View Controller approach proved to be a much easier way for implementing and understanding database schema.

```

    /**
 * Show all vacancies.
 */
public function index()
{
    $userID = \Auth::user()->id;
    $vacancies = Vacancy::latest('created_at')->where('user_id', '=', $userID)->where('vacancy_date', '>=', Carbon::today())->
        paginate(5, ['*'], 'vacancies');
    $pastVacancies = Vacancy::latest('created_at')->where('user_id', '=', $userID)->where('vacancy_date', '<=', Carbon::yesterday())->
        paginate(5, ['*'], 'pastVacancies');
    $applications = Application::whereHas('vacancy', function ($query) {
        $query->where('vacancy_date', '>=', Carbon::today());
    })->where(['homeowner_id' => $userID, 'status' => 'Confirmed'])->paginate(5, ['*'], 'applications');
    return view('dashboard.vacancies.index', compact('vacancies', 'pastVacancies', 'applications'));
}

/**
 * Post a new vacancy.
 */
public static function post(array $attributes)
{
    return new static($attributes);
}

/**
 * A vacancy can only be for one property.
 */
public function property()
{
    return $this->hasOne('App\Property', 'id', 'property_id');
}

/**
 * A vacancy can only have one owner.
 */
public function user()
{
    return $this->hasOne('App\User', 'id', 'user_id');
}

```

Figure 29: Vacancy Controller and Vacancy Model

## 4.4 Notable Challenges

### Tailored Vacancies

With homeless people coming from a whole variety of different backgrounds, and each having to cope with their own issues, it was important that vacancies were only displayed to homeless people who would be suitable to apply for it. This therefore meant that if a Homeowner posted a vacancy but specifically stated that they wouldn't be open to hosting a homeless person who had a history of alcohol abuse, the homeless person wouldn't see their post. This feature is important as it aims to decrease the homeless users disappointment of not getting accepted for a vacancy that they wouldn't have stood a chance for in the first place.

```

$vacancies = Vacancy::where('vacancy_date', '>=', Carbon::today());

if (Auth::check()) {

    $user = Auth::user();

    if ($user->user_role_id == 2) {

        $alcoholAbuse = $user->homeless_profile->alcohol_abuse_history;
        $drugAbuse = $user->homeless_profile->drug_abuse_history;
        $criminalHistory = $user->homeless_profile->criminal_history;

        if ($alcoholAbuse == 1 && $drugAbuse == 1 && $criminalHistory == 1) {
            $vacancies = $vacancies->where(['alcohol_abuse_accepted' => 1, 'drug_abuse_accepted' => 1, 'criminal_history_accepted' => 1]);
        } else if ($alcoholAbuse == 1 && $drugAbuse == 0 && $criminalHistory == 0) {
            $vacancies = $vacancies->where('alcohol_abuse_accepted', 1);
        } else if ($alcoholAbuse == 1 && $drugAbuse == 1 && $criminalHistory == 0) {
            $vacancies = $vacancies->where(['alcohol_abuse_accepted' => 1, 'drug_abuse_accepted' => 1]);
        } else if ($alcoholAbuse == 1 && $drugAbuse == 0 && $criminalHistory == 1) {
            $vacancies = $vacancies->where(['alcohol_abuse_accepted' => 1, 'criminal_history_accepted' => 1]);
        } else if ($alcoholAbuse == 0 && $drugAbuse == 1 && $criminalHistory == 0) {
            $vacancies = $vacancies->where('drug_abuse_accepted', 1);
        } else if ($alcoholAbuse == 0 && $drugAbuse == 1 && $criminalHistory == 1) {
            $vacancies = $vacancies->where(['drug_abuse_accepted' => 1, 'criminal_history_accepted' => 1]);
        } else if ($alcoholAbuse == 0 && $drugAbuse == 0 && $criminalHistory == 1) {
            $vacancies = $vacancies->where('criminal_history_accepted', 1);
        }
    }
}

```

Figure 30: Tailored Vacancies Code

As you can see from the code above, this feature was implemented by first obtaining the relevant information on the homeless user. The information obtained checks for a history of alcohol abuse, drug abuse, and criminal history. This information is then compared against the vacancies in the system to determine what vacancies suit the user, these will then be the only vacancies that the user sees.

## Vacancy Cycle

With the basis of the application being the ability to apply for, and eventually take up accommodation, the process of applying for a vacancy needed to be correct, and fully functional. The vacancy cycle takes into account the homeless and homeowner users requirements and aims at making the cycle as optimal as it can be for both parties.

```
/*
 * Apply for a Vacancy.
 *
 * @return \Response
 */
public function apply($id)
{
    $vacancy = Vacancy::findOrFail($id);
    $applicant = Auth::user();

    Application::create([
        'vacancy_id' => $vacancy->id,
        'homeowner_id' => $vacancy->user_id,
        'homeless_id' => Auth::user()->id,
    ]);

    Notification::create([
        'primary_user_id' => $vacancy->user_id,
        'secondary_user_id' => $applicant->id,
        'notification_type' => 'Application Made',
        'notification' => $applicant->first_name . ' ' . $applicant->surname . ' has applied for your vacancy at ' . $vacancy->property->building_name_number . ' ' . $vacancy->property->address_line_one . '',
    ]);

    flash()->success('Success', 'You have successfully applied!');

    return redirect()->back();
}
```

Figure 31: Code to Apply for a Vacancy

The initiation of the Vacancy Cycle comes in the form of a Homeowner posting a vacancy. Once this is done, Homeless users are able to view the vacancy, if they fit the Homeowners criteria, and if they wish to do so, apply for the vacancy. On applying for the Vacancy, the Homeowner is notified, and has the ability to review the applicant, viewing their profile, biography and previous reviews. Based on this information, the Homeowner can then make a decision as to whether or not they would like to accommodate that applicant.

If the Homeowner decides to approve the applicant, the applicant will be notified, and then must confirm if they will be taking up the vacancy. If the applicant decides to take up the offer, and confirms they will be, all other applicants for that vacancy will then be automatically declined. This is an important process in the Vacancy Cycle. As Homeless users are able to apply for up to three vacancies for one day, they may receive multiple acceptance offers. Therefore, not declining all other applicants until the Homeless user has confirmed, allows the Homeowner to offer the vacancy to other applicants if their offer is declined.

```
/**
 * Approve an application.
 *
 * @return \Response
 */
public function approveApplication($id)
{
    $homeowner = \Auth::user();
    $application = Application::findOrFail($id);
    $application->status = 'Accepted';
    $application->save();
    Notification::create([
        'primary_user_id' => $application->homeless_id,
        'secondary_user_id' => $homeowner->id,
        'notification_type' => 'Application Accepted',
        'notification' => $homeowner->first_name . ' ' . $homeowner->surname . ' has accepted your application. Confirm you will be attending.',
    ]);
    flash()->success('Success', 'You have successfully accepted an application!');
    return redirect()->back();
}
```

Figure 32: Code to Approve an Application

## Searching for a Vacancy

Perhaps the most difficult challenge to overcome was the ability for a user to search for vacancies. Whilst the concept is fairly rudimentary in terms of web development, when taking into account a users individual requirements, and only searching for specific locations, the process became very complex. One approach that was considered for filtering on the client side was Vue.js, however given the time restraints of the development, it was decided that undertaking the learning of another unknown framework would only be detrimental to the successful development of Stay At Mine. Therefore it was decided that a process of using GET parameters in the URL would be sufficient in terms of allowing the user to search and filter data. Whilst this process is not as fluid as having filtering and searching on the client-side, it provided a means of accomplishing the same outcome, with a much shorter time frame.

```

// Search criteria
$searchArea = Input::get('area');
$searchDate = Input::get('date');

$prepareDate = str_replace("/", "-", $searchDate);
$time = strtotime($prepareDate);
$comparableDate = date('Y-m-d', $time);

if ($searchArea !== null) {

    if (strcasecmp($searchArea, 'belfast') == 0) {
        $vacancies->whereHas('property', function ($areaQuery) {
            $areaQuery->whereIn('area',
                ['Belfast City Centre', 'North Belfast', 'South Belfast', 'East Belfast', 'West Belfast']);
        });
    } else {
        $vacancies->whereHas('property', function ($areaQuery) use ($searchArea) {
            $areaQuery->where('area', $searchArea);
        });
    }
}

```

Figure 33: Searching for a Vacancy Code

## 5.0 Testing

---

### 5.1 Test Approach Selection

Once the functional development of the Stay At Mine web application was completed, the next phase of the development cycle was to test various aspects of the system, and ensure that they work and perform as expected. In order to complete the testing phase, I will be undertaking a variety of different testing approaches for different aspects of the system.

#### 5.1.1 White Box Testing

White box testing is a process of testing an applications internal infrastructure. The test process is primarily focused on flaws in security, the flow of data input and output, and improving the design and usability of an application. [Guru 99, 2015] White box testing is just one part of the testing process that is combined with black box testing, to provide an extensive overview of the system as a whole. In laymans terms, white box testing tests how the application functions in the backend.

## 5.1.2 Black Box Testing

Black box testing is essentially the opposite of white box testing. Black box testing tests an application from a users point of view, and ensures that everything performs as the user would expect it to. With a black box test, the users sees no code, nor are they expected to understand the code behind the application. [Software Testing Help, 2015]

## 5.1.3 Cross Browser Testing

With white and black box testing taking care of the functionality of the application, the final aspect of the testing cycle is to ensure that the application displays consistently across all major browsers and devices. This is a crucial part of the testing cycle, as with todays technology, there are countless different versions of browsers in operation, and countless numbers of different devices being used to access the internet. Therefore, ensuring that all users get a consistent experience is of the upmost importance.

## 5.2 Test Process

The test cases used to test the system, for white box and black box testing, will be taken directly from the requirements that were created prior to the beginning of the development process. **[See Appendix F]** An example test case can be seen below, and the full table can be seen in **Appendix J**.

Test ID	Description	Outcome	Response (if failed)
1	The application will allow the homeowner to create an account for the application.	Passed	

In terms of testing the application for cross browser compatibility, the application will be loaded in various browsers and checked to ensure that it is consistent. It is not feasible to test the application on all of the different browsers and versions that are in operation, therefore the application will only be tested on the most widely used browsers. The tests and results of the cross compatibility browser tests can be seen in **Appendix K**.

## 5.3 Test Results

After completing the testing phase of the development cycle, it was time to step back and reflect on the results of the tests undertaken. With 20 out of the 22 functional tests passing (a pass rate of 91%), it is clear that the testing phase of the development cycle was a success. The two failing tests were the result of the limited time available for the development of the application and have been put forward for the future development of the application.

The cross browser testing of the application can also be crowned as a success, with only minor CSS styling changes needing amended for the iOS version of Safari.

## 5.4 User Survey

The final aspect of testing the Stay At Mine application was to test the non-functional requirements that were set out in **Appendix F**. As these requirements cannot be tested by carrying out a function, it was necessary to create a survey on Survey Monkey, and get people of various age ranges, and computer literacy to undertake the survey. Survey Monkey is an online tool that allows for surveys to be created and undertaken. The platform allows for simple surveys to be created, or indeed more complex surveys for more in-depth research. [Survey Monkey, 2016]

The questions asked in the survey are outlined below:

1. What is your age range?
2. What is your computer literacy?
3. On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how attractive would you say the Stay At Mine application is?
4. On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how trustworthy would you say the Stay At Mine application is?
5. On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how confident are you that you would be able to use the core functionality of the Stay At Mine website without any prior training?
6. How easy would you say it is to understand what the purpose of the Stay At Mine website is?

### 6.1 Evaluate Survey Results

The results of the user surveys were very positive and encouraging, allowing for some extremely useful feedback of the Stay At Mine application, from more neutral points of view. In total, ten people were able to complete the survey, with a variety of different age ranges, and computer literacy.

Two major concerns, in developing Stay At Mine, were that the application would be inviting and trustworthy, encouraging people to use the application. Therefore, two of the questions on the survey read **“On a scale of 1-10, how attractive would you say the Stay At Mine Application is?”** and **“On a scale of 1-10, how trustworthy would you say the Stay At Mine application is?”** respectively. These questions were aimed at testing the concerns of the specific non-functional requirements stated above. As demonstrated in **figures 34 and 35** below, the outcome of these questions were extremely positive, with attractiveness receiving an average rating of 9, and trustworthiness receiving an average rating of 9.3.

On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how attractive would you say the Stay At Mine application is?

Answered: 10 Skipped: 0

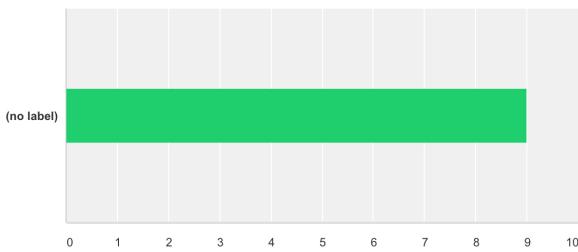


Figure 34: Survey response on attractiveness

On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how trustworthy would you say the Stay At Mine application is?

Answered: 10 Skipped: 0

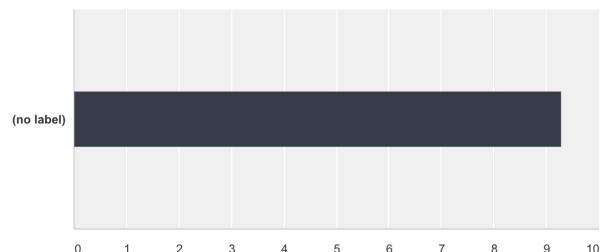


Figure 35: Survey response on trustworthiness

The survey responses stated that 60% of people surveyed thought that the information on the Stay At Mine website was extremely easy to understand, and 40% stated that it was quite easy to understand. Therefore it is clear that the message of what the application is, is being conveyed on the website.

The ability for people to use the application without any prior training was also a non-functional requirement. With this question receiving an average of 8.1, it is fair to assume that the majority of people think they would be able to use the application, however perhaps some refinement could help bring that number up. The completed survey results can be seen in **Appendix L**.

## **6.2 Evaluate Project Outcomes**

The final outcome of the Stay At Mine project is a thoroughly considered and developed web application that aims at providing those less fortunate a helping hand. The development of the application comes at a time when concerns are extremely high in Northern Ireland about how we care for our homeless population, with several deaths in recent months. The application has reached the majority of requirements that were set out at the start of the process, resulting in a tight knit application that acts as the middle man between homeowners and the homeless community.

It is clear from the user surveys that the branding, user experience, and general design of the Stay At Mine application have been well received, and tick the boxes of the non-functional requirements. This was an important aspect of the project. With the general vulnerability of the homeless community, it was essential that the Stay At Mine application provided them with a trustworthy portal.

On the backend of the application, homeowners are able to post vacancies and homeless users are able to apply for those vacancies. Homeowners are able to tailor their vacancies to suit their own needs, and homeless users are able to view tailored results of vacancies.

The use of the Laravel framework, whilst at the time was daunting, has brought with it a much more in depth knowledge of PHP as a development language, as well allowing for the continued learning of a framework that is widely used within the industry.

In terms of improving the Stay At Mine application, the features that were not implemented due to time restrictions, could be developed. The terminology used throughout the application could be refined by an expert in order to be more in touch with the homeless community. Efforts were made to try to make the terminology as inclusive and correct as possible, however some phrases such as “Alcohol Abuse Victim” could be refined further, so as to not cause any offence to users.

## **6.3 Evaluate the Methodology**

The modified waterfall methodology has been an extremely effective methodology for the development of the Stay At Mine application. I have always found that I work best in a linear path, and the waterfall methodology allows for this type of work flow. Having the flexibility

of the modified waterfall, to revisit and modify different aspects of the system, combined with the linear workings of the waterfall methodology, has been fundamental in ensuring the development journey was a smooth one. The modified waterfall fits neatly into the development process of a final year student, especially when unexpected issues/tasks arise.

## 7.0 Conclusion

---

### 7.1 Summary

The development of the Stay At Mine web application came with it many stimulating and, sometimes, frustrating challenges that called for the implementation of knowledge created from the whole duration of my time on the Interactive Multimedia Design course. The development also brought with it the need to enrich my learning with areas that aren't typically covered as part of my degree, such as the Laravel framework. Having dedicated a good percentage of my time to developing a concise and well thought-out plan, as well as considering and refining the design and user interaction of the application, the journey through the development cycle was largely a smooth one. Allocating considerable amounts of time to learning Laravel, culminated in a great admiration for the framework and the thirst to continue developing with it at the core. The process of learning Laravel brought with it a much deeper understanding of PHP itself, and will have been fundamental to my web development career.

### 7.2 Reflect on What Happened

Taking into account the complexity of the application, and with my degree dependant upon the successful development of it, the journey through the entire process, was all-in-all a fairly smooth one. Having devised a plan for the development of the application, and undertaking extensive research, potential issues were able to have been detected at an early stage, paving the way for fewer surprise delays.

### 7.3 Reflect on Your Role

Undertaking a project the size of Stay At Mine as an individual student was an extremely difficult task. For the project, I had to take on the roles of a developer, designer, project manager, and tester. As a web developer, I find that I am more comfortable developing a website rather than designing or planning it. However, as is the basis of an Interactive Multimedia Design Major Project, it was solely up to me to complete all aspects of the Stay At

Mine project, which meant stepping out of my comfort zone.

Being forced to undertake tasks that I wouldn't ordinarily volunteer for, allowed me to gain an insight into the various other roles within the web development industry, rather than just development. In doing so, I was able to enjoy the planning and design of the Stay At Mine application, and increase my skills in those areas.

I believe that I was able to perform the different roles required for the project to a high standard and the project ultimately ended up with the anticipated outcome.

## **7.4 Suggest Future Work**

In order to exploit the concept behind the Stay At Mine application to its full potential, a number of future developments would need to be considered to refine the application. The review process of the application should be the first point of refinement. With the limited timescale of the development of the application, the review process wasn't able to be developed as in-depth as what would be needed for a real-world implementation of the application. Before the application could be widely used by the public, the review system would need to be locked down in order to prevent malicious and derogatory comments being made about any parties involved in the process.

The notifications aspect of the application could also be refined further. At present, notifications are displayed in chronological order, however, the implementation of a 'sticky' notification system would allow for more important notifications, such as the acceptance of an application, to appear at the top of the list.

Whilst the key directive behind the Stay At Mine application has been to provide Homeless people with a short-term roof over their heads, the application could be repurposed to suit a whole variety of different purposes, for example student housing.

## 8.0 References

---

The homelessness monitor: England. 2015. [ONLINE] Available at: [http://www.crisis.org.uk/data/files/publications/HomelessnessMonitorEngland2015\\_ExecSummary\\_FINAL.pdf](http://www.crisis.org.uk/data/files/publications/HomelessnessMonitorEngland2015_ExecSummary_FINAL.pdf). [Accessed 16 October 2015].

14% rise in numbers sleeping rough - BelfastTelegraph.co.uk. 2015. 14% rise in numbers sleeping rough - BelfastTelegraph.co.uk. [ONLINE] Available at: <http://www.belfasttelegraph.co.uk/news/uk/14-rise-in-numbers-sleeping-rough-31024398.html>. [Accessed 16 October 2015].

Europe's migration crisis: how many people are on the move? - Telegraph. 2015. Europe's migration crisis: how many people are on the move? - Telegraph. [ONLINE] Available at: <http://www.telegraph.co.uk/news/uknews/immigration/11875036/Europe's-migration-crisis-how-many-people-are-on-the-move.html>. [Accessed 16 October 2015].

UK asylum applications hit six-year high in July | www.ein.org.uk. 2015. UK asylum applications hit six-year high in July | www.ein.org.uk. [ONLINE] Available at: <https://www.ein.org.uk/news/uk-asylum-applications-hit-six-year-high-july>. [Accessed 16 October 2015].

Population - ONS . 2015. Population - ONS . [ONLINE] Available at: <http://ons.gov.uk/ons/taxonomy/index.html?nscl=Population>. [Accessed 16 October 2015].

How AirBnb Started Or How 3 Guys Went from... | Funders and Founders Notes . 2015. How AirBnb Started Or How 3 Guys Went from... | Funders and Founders Notes . [ONLINE] Available at: <http://notes.fundersandfounders.com/post/82297315548/how-airbnb-started>. [Accessed 22 October 2015].

Homeless on the steps of Airbnb's headquarters: the faces of San Francisco's failure story | Technology | The Guardian. 2015. Homeless on the steps of Airbnb's headquarters: the faces of San Francisco's failure story | Technology | The Guardian. [ONLINE] Available at: <http://www.theguardian.com/technology/2015/mar/05/homeless-steps-airbnb-san-francisco>. [Accessed 22 October 2015].

Airbnb Campaign Brings Attention to NYC's 60,000 Homeless. 2015. Airbnb Campaign Brings Attention to NYC's 60,000 Homeless. [ONLINE] Available at: <http://www.triplepundit>.

The homelessness monitor: England. 2015. [ONLINE] Available at: [http://www.crisis.org.uk/data/files/publications/HomelessnessMonitorEngland2015\\_ExecSummary\\_FINAL.pdf](http://www.crisis.org.uk/data/files/publications/HomelessnessMonitorEngland2015_ExecSummary_FINAL.pdf). [Accessed 16 October 2015].

14% rise in numbers sleeping rough - BelfastTelegraph.co.uk. 2015. 14% rise in numbers sleeping rough - BelfastTelegraph.co.uk. [ONLINE] Available at: <http://www.belfasttelegraph.co.uk/news/uk/14-rise-in-numbers-sleeping-rough-31024398.html>. [Accessed 16 October 2015].

Europe's migration crisis: how many people are on the move? - Telegraph. 2015. Europe's migration crisis: how many people are on the move? - Telegraph. [ONLINE] Available at: <http://www.telegraph.co.uk/news/uknews/immigration/11875036/Europe's-migration-crisis-how-many-people-are-on-the-move.html>. [Accessed 16 October 2015].

UK asylum applications hit six-year high in July | www.ein.org.uk. 2015. UK asylum applications hit six-year high in July | www.ein.org.uk. [ONLINE] Available at: <https://www.ein.org.uk/news/uk-asylum-applications-hit-six-year-high-july>. [Accessed 16 October 2015].

Population - ONS . 2015. Population - ONS . [ONLINE] Available at: <http://ons.gov.uk/ons/taxonomy/index.html?nscl=Population>. [Accessed 16 October 2015].

How AirBnb Started Or How 3 Guys Went from... | Funders and Founders Notes . 2015. How AirBnb Started Or How 3 Guys Went from... | Funders and Founders Notes . [ONLINE] Available at: <http://notes.fundersandfounders.com/post/82297315548/how-airbnb-started>. [Accessed 22 October 2015].

Homeless on the steps of Airbnb's headquarters: the faces of San Francisco's failure story | Technology | The Guardian. 2015. Homeless on the steps of Airbnb's headquarters: the faces of San Francisco's failure story | Technology | The Guardian. [ONLINE] Available at: <http://www.theguardian.com/technology/2015/mar/05/homeless-steps-airbnb-san-francisco>. [Accessed 22 October 2015].

Airbnb Campaign Brings Attention to NYC's 60,000 Homeless. 2015. Airbnb Campaign Brings Attention to NYC's 60,000 Homeless. [ONLINE] Available at: <http://www.triplepundit.com/2015/03/airbnb-campaign-brings-attention-60000-homeless-nyc-streets/>. [Accessed 22 October 2015].

NYC Creative Duo Aims to Help Manhattan Homeless with AirBnb Listings | AgencySpy. 2015. NYC Creative Duo Aims to Help Manhattan Homeless with AirBnb Listings | AgencySpy. [ONLINE] Available at: <http://www.adweek.com/agency spy/nyc-creative-duo-aims-to-help-manhattan-homeless-with-airbnb-listings/81796>. [Accessed 22 October 2015].

Simon Community app to help homeless find shelter and food - BBC News. 2015. Simon Community app to help homeless find shelter and food - BBC News. [ONLINE] Available at: <http://www.bbc.co.uk/news/uk-northern-ireland-30219079>. [Accessed 22 October 2015].

Usage Statistics and Market Share of Server-side Programming Languages for Websites, December 2015. 2015. Usage Statistics and Market Share of Server-side Programming Languages for Websites, December 2015. [ONLINE] Available at: [http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all). [Accessed 30 December 2015].

What is the Client-Server Model? - Definition from Techopedia . 2016. What is the Client-Server Model? - Definition from Techopedia . [ONLINE] Available at: <https://www.techopedia.com/definition/18321/client-server-model>. [Accessed 03 December 2015].

What Are the Pros and Cons of PHP? (with picture). 2015. What Are the Pros and Cons of PHP? (with picture). [ONLINE] Available at: <http://www.wisegeek.com/what-are-the-pros-and-cons-of-php.htm>. [Accessed 30 December 2015].

Top 10 Programming Languages to Learn in 2014. 2015. Top 10 Programming Languages to Learn in 2014. [ONLINE] Available at: <https://blog.udemy.com/best-programming-language/>. [Accessed 30 December 2015].

Compiled vs. Interpreted Languages. 2015. Compiled vs. Interpreted Languages. [ONLINE] Available at: <http://www.vanguardsw.com/dphelp4/dph00296.htm>. [Accessed 30 December 2015].

David Flanagan, 2005. Java In A Nutshell, 5th Edition. 5 Edition. O'Reilly Media.

4 reasons to stick with Java -- and 4 reasons to dump it | InfoWorld. 2015. 4 reasons to stick with Java -- and 4 reasons to dump it | InfoWorld. [ONLINE] Available at: <http://www.infoworld.com/article/2687995/java/4-reasons-to-stick-with-java.html>. [Accessed 30 December 2015].

What is Java? A Webopedia Definition. 2015. What is Java? A Webopedia Definition. [ONLINE] Available at: <http://www.webopedia.com/TERM/J/Java.html>. [Accessed 30 December 2015].

Does the iPhone support Java? | The iPhone FAQ. 2015. Does the iPhone support Java? | The iPhone FAQ. [ONLINE] Available at: <http://www.iponfaq.org/archives/9731>. [Accessed 30 December 2015].

5 reasons NOT to use Twitter Bootstrap | Zing Design. 2015. 5 reasons NOT to use Twitter Bootstrap | Zing Design. [ONLINE] Available at: <http://www.zingdesign.com/5-reasons-not-to-use-twitter-bootstrap/>. [Accessed 30 December 2015].

The Bootstrap Framework ... Should You Use It or Not?. 2015. The Bootstrap Framework ... Should You Use It or Not?. [ONLINE] Available at: <http://www.htmlcenter.com/blog/the-bootstrap-framework-controversy-should-you-use-it-or-not/>. [Accessed 30 December 2015].

Introduction to Laravel Framework - Laravel Book. 2015. Introduction to Laravel Framework - Laravel Book. [ONLINE] Available at: <http://laravelbook.com/laravel-introduction/>. [Accessed 30 December 2015].

Laravel - The PHP Framework For Web Artisans. 2015. Laravel - The PHP Framework For Web Artisans. [ONLINE] Available at: <https://laravel.com/>. [Accessed 30 December 2015].

The Best Laravel and PHP Screencasts. 2015. The Best Laravel and PHP Screencasts. [ONLINE] Available at: <https://laracasts.com/>. [Accessed 31 December 2015].

CodeIgniter Web Framework. 2015. CodeIgniter Web Framework. [ONLINE] Available at: <https://www.codeigniter.com/>. [Accessed 31 December 2015].

SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems | DigitalOcean. 2015. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems | DigitalOcean. [ONLINE] Available at: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>. [Accessed 31 December 2015].

Sass: Syntactically Awesome Style Sheets. 2015. Sass: Syntactically Awesome Style Sheets. [ONLINE] Available at: <http://sass-lang.com/>. [Accessed 31 December 2015].

Sass v Less v Stylus: The pros and cons | Tutorials | Web Designer. 2015. [ONLINE] Available at: <http://www.webdesignermag.co.uk/sass-v-less-v-stylus-the-pros-and-cons/>. [Accessed 31 December 2015].

History of APIs - Google Maps API . 2016. History of APIs - Google Maps API . [ONLINE] Available at: <http://apievangelist.com/2011/01/30/history-of-apis--google-maps-api/>. [Accessed 07 January 2016].

5 things to know before choosing digital maps API: Google Maps Vs Mapbox explained.. 2016. [ONLINE] Available at: <http://nextjuggernaut.com/blog/google-vs-mapbox/>. [Accessed 07 January 2016].

Mapbox | Design and publish beautiful maps. 2016. Mapbox | Design and publish beautiful maps. [ONLINE] Available at: <https://www.mapbox.com/>. [Accessed 07 January 2016].

What is HyperText Markup Language - HTML? Webopedia. 2016. [ONLINE] Available at: <http://www.webopedia.com/TERM/H/HTML.html>. [Accessed 07 January 2016].

Getting started | Less.js . 2015. [ONLINE] Available at: <http://lesscss.org/>. [Accessed 31 December 2015].

What is gulp.js and why use it? | Brandon Clapp. 2015. [ONLINE] Available at: <http://brandonclapp.com/what-is-gulp-js-and-why-use-it/>. [Accessed 31 December 2015].

And just like that Grunt and RequireJS are out, it's all about Gulp and Browserify now - { 100PercentJS }. 2015. [ONLINE] Available at: <http://www.100percentjs.com/just-like-grunt-gulp-browserify-now/>. [Accessed 31 December 2015].

SurveyMonkey: Free online survey software & questionnaire tool . 2016. [ONLINE] Available at: <https://www.surveymonkey.com/>. [Accessed 03 April 2016].

Guru99com: White box testing. 2016. [ONLINE] Available at: <http://www.guru99.com/white-box-testing.html> [Accessed 15 April 2016].

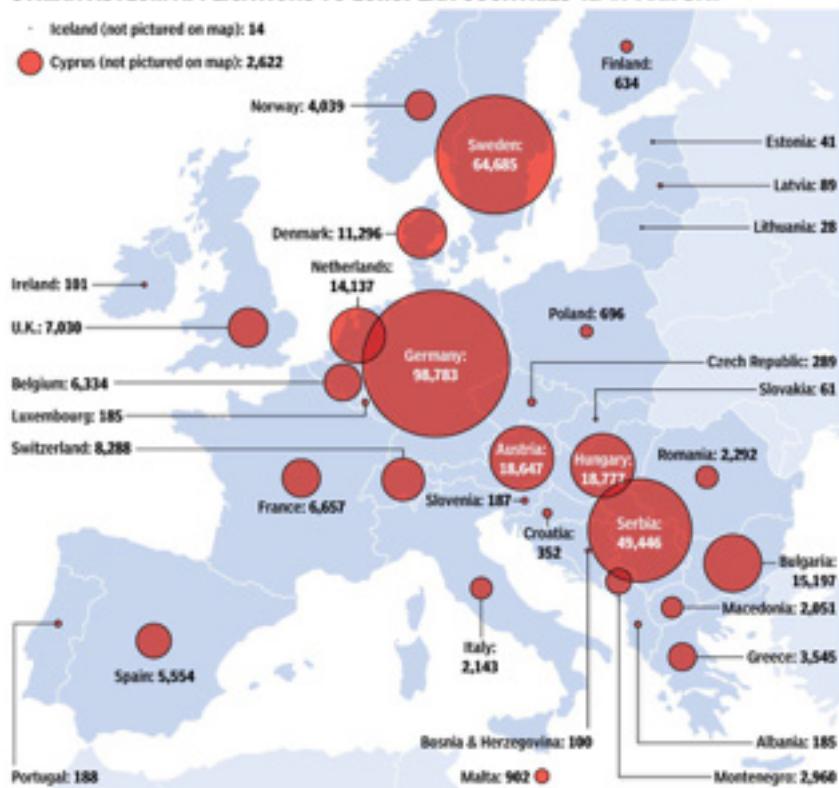
Software Testing Help. 2016. [ONLINE] Available at: <http://www.softwaretestinghelp.com/white-box-testing-techniques-with-example/>. [Accessed 15 April 2016].

### Appendix A

Year(s)	2010	2011	2012	2013	2014
People sleeping rough	1,768	2,181	2,309	2,414	2,744
% change from the previous year (in table)		23%	6%	5%	14%

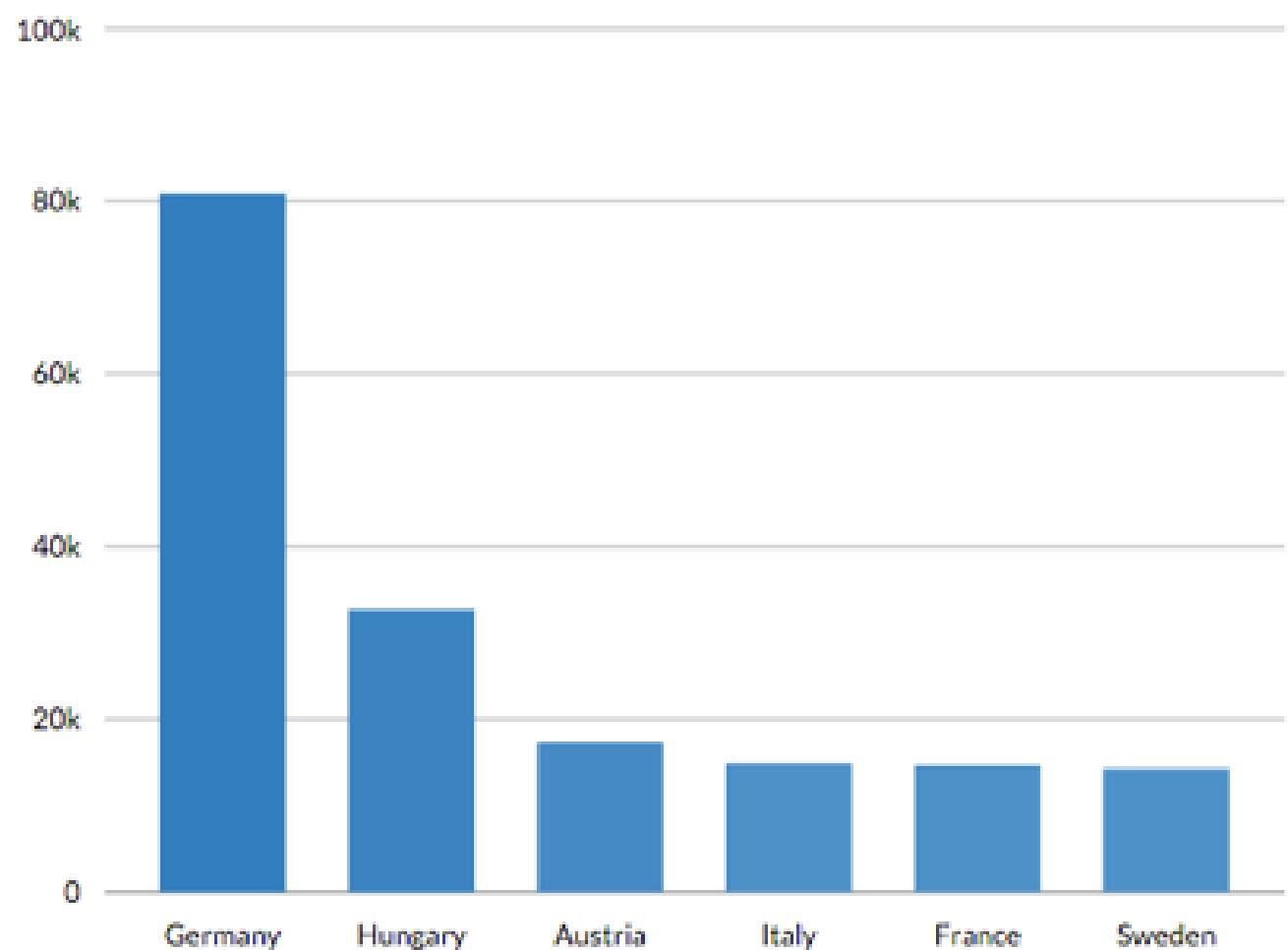
## Appendix B

### SYRIAN ASYLUM APPLICATIONS TO EUROPEAN COUNTRIES AS OF JULY 2015

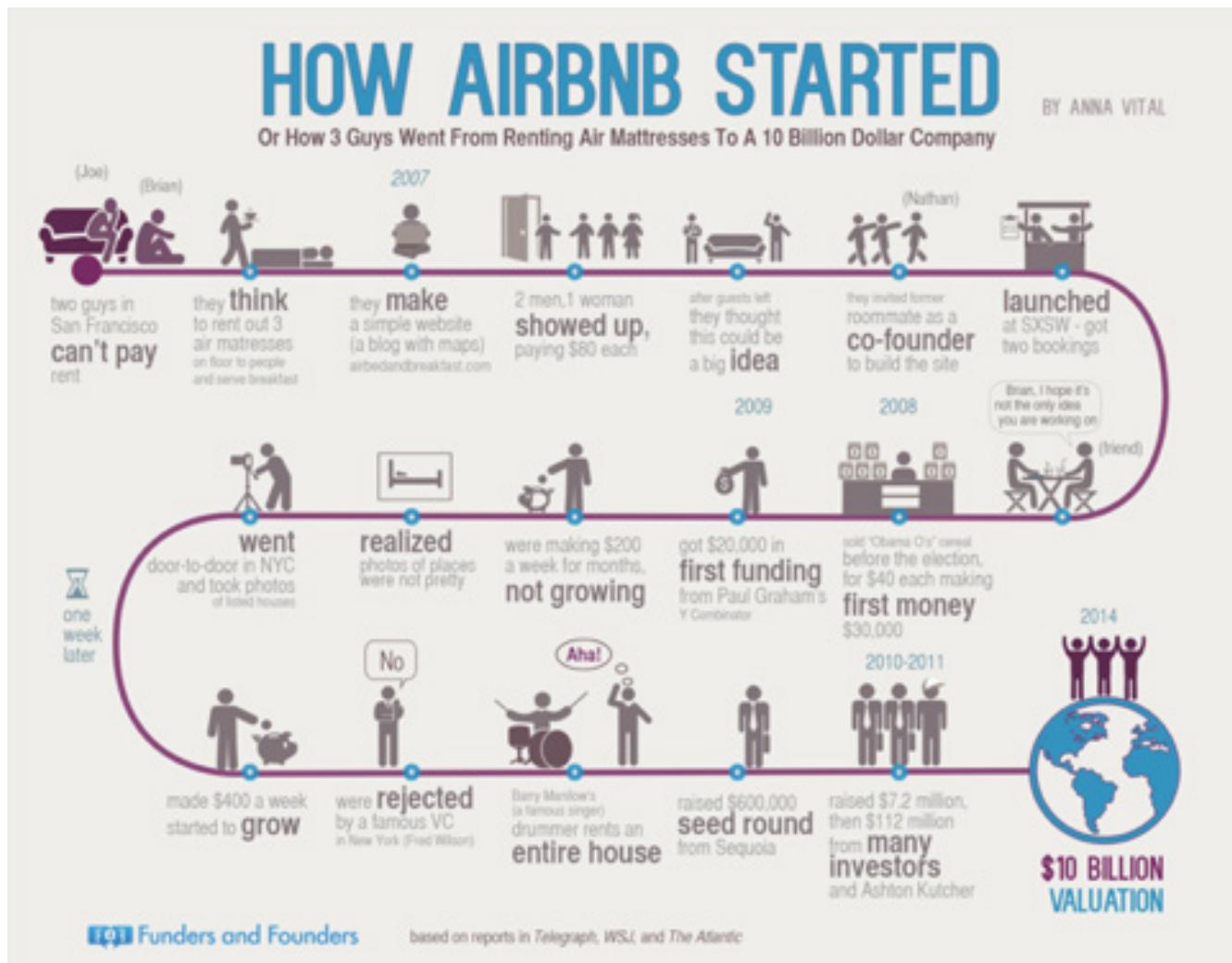


## **Appendix C**

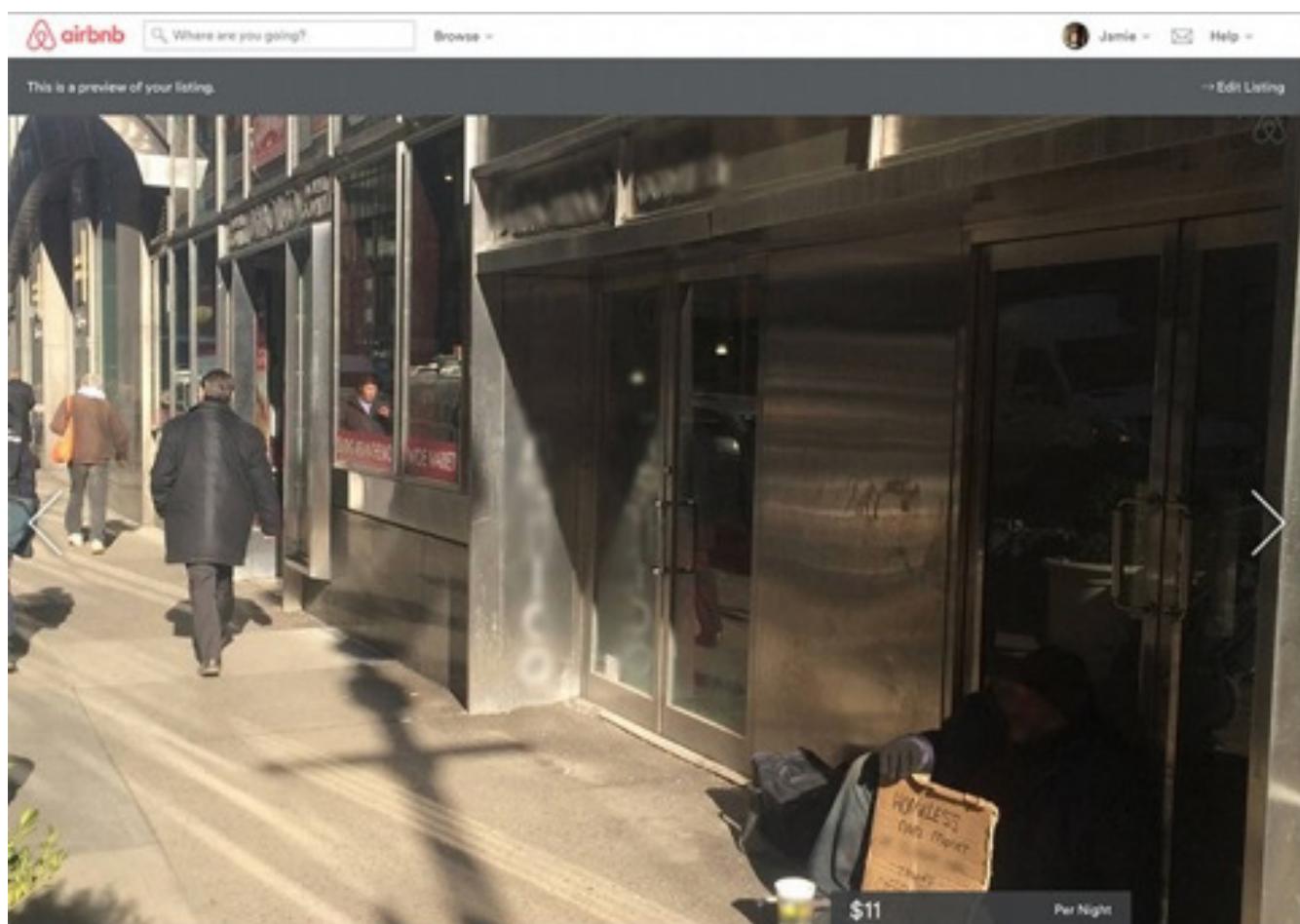
### **Countries with the greatest number of first time asylum applications, April- June 2015**



## Appendix D



## Appendix E



This is a preview of your listing.

airbnb Where are you going? Browse →

Jamie Help Edit Listing

Hot Bedroom w/ Semi Private Terrace  
New York, NY, United States

Check In  mm/dd/yyyy

Check Out  mm/dd/yyyy

Guests  1

\$11 Per Night

Request to Book

The screenshot shows a listing on the Airbnb website. The main image displays a street scene in New York City at night. A man in a dark coat walks away from the camera towards a building. In the foreground, a homeless person sits on the sidewalk, holding a cardboard sign that reads "HOMELESS This Mannequin". The listing title is "Hot Bedroom w/ Semi Private Terrace" located in New York, NY, United States. It includes a profile picture of the host named Jamie, a "Shared room" option, "1 Guest", and "1 Bed". The price is listed as \$11 per night, and there is a prominent red "Request to Book" button.

## Appendix F

<b>Requirement #:</b>	1
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The application will allow the homeowner to create an account for the application.
<b>Rationale:</b>	A homeowner will need to have an account to be able to post room vacancies within the application.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	When an account is created, the users name, age, gender, address, postcode, username and password will be added to the database.

<b>Requirement #:</b>	2
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The application will allow a homeless person to create an account for the application.
<b>Rationale:</b>	A homeless person will need to have an account to be able to apply for vacancies within the application.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	When an account is created, the users name, age, gender, location, username and password will be added to the database.

<b>Requirement #:</b>	3
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The application will allow a charity/organisation to create an account for the application.
<b>Rationale:</b>	A charity/organisation will need to have an account to be able to apply for vacancies on behalf of a homeless person.
<b>Dependencies:</b>	Requires the charity/organisation to be 'connected' with a homeless person within the application.
<b>Fit Criterion:</b>	When an account is created, the users name, age, gender, location, username and password will be added to the database.

<b>Requirement #:</b>	4
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The create account form will be validated for errors.
<b>Rationale:</b>	When filling in a form, human error can be present, and given the nature of the application, it is crucial that the correct details are recorded for all of the users.

<b>Dependencies:</b>	
<b>Fit Criterion:</b>	If there are any errors with the users details that they have input, the form will not be submitted, and the errors will be presented to the user on screen.

<b>Requirement #:</b>	5
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The application will allow the user to login as either a homeowner, homeless, or chairty user.
<b>Rationale:</b>	Allows the user to access their account and apply for/post vacancies.
<b>Dependencies:</b>	Requires the user to have previously signed up to use the application.
<b>Fit Criterion:</b>	The user will be able to login to the application, and will be taken to the correct section of the application according to the type of account they have.

<b>Requirement #:</b>	6
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The application will allow the user to reset their password if they forget it.
<b>Rationale:</b>	Will having so many passwords for different accounts, users will frequently forget their password.
<b>Dependencies:</b>	Requires the user to actually have an account for the application.
<b>Fit Criterion:</b>	The user will be sent a link that they will need to click to be redirected to a password reset page.

<b>Requirement #:</b>	7
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The application will allow the user to delete their account if they do not wish to use the application anymore.
<b>Rationale:</b>	There will be no need to keep accounts active for users who no longer wish to use the application.
<b>Dependencies:</b>	Requires the user to have an account.
<b>Fit Criterion:</b>	The users details will be disabled, and deleted after a certain time period. The user will no longer be able to login to the application.

<b>Requirement #:</b>	8
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The application will allow the user to update their account details.

<b>Rationale:</b>	On a regular basis, peoples personal details may change, be that a change of address or phone number, and would need to be kept accurate on the system.
<b>Dependencies:</b>	Requires the user to have an account.
<b>Fit Criterion:</b>	Any old details will be replaced by the updated details.

<b>Requirement #:</b>	9
<b>Requirement Type:</b>	Functional
<b>Description:</b>	Homeowner users must be able to post a new vacancy.
<b>Rationale:</b>	The whole basis of the application relies on homeowners publishing vacancies within their house. A homeless person would therefore need to see these vacancies in order to apply for them.
<b>Dependencies:</b>	Requires the homeowner to have an account.
<b>Fit Criterion:</b>	The new vacancy will no appear to homeless people seeking a vacancy.

<b>Requirement #:</b>	10
<b>Requirement Type:</b>	Functional
<b>Description:</b>	A homeowner must have the ability to upload photographs of the vacant room.
<b>Rationale:</b>	Whilst the homeless user will not be looking at the photographs to ensure the 5* hotel that they are booking meets their requirements, it would still be of some comfort for the homeless person to have a preview of the room that they could be staying in.
<b>Dependencies:</b>	Requires the homeowner to have an account.
<b>Fit Criterion:</b>	The photographs will be posted with their vacancy, and will be viewable by the homeless user.

<b>Requirement #:</b>	11
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The homeowner must be able to add their address as a location marker on a Google Map.
<b>Rationale:</b>	When the homeowner and homeless user agree upon a vacancy, it would be beneficial to the homeless user to be able to view a map of how to get to the homeowners address.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	The homeowners' address will be visible on a Google Map.

<b>Requirement #:</b>	12
<b>Requirement Type:</b>	Functional

<b>Description:</b>	The homeowner must be able to seamlessly add their address to a Google Map using just their postcode.
<b>Rationale:</b>	To post a location on a Google Map, you need the longitude and latitude coordinates. Most, if not all, users of the application will not know how to obtain their longitude and latitude coordinates, and therefore putting in their postcode will be the easiest solution.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	The users location will appear on a Google Map.

<b>Requirement #:</b>	13
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The homeowner must be able to adjust the Google Map pin, if it is placed inaccurately on the map.
<b>Rationale:</b>	Through experience using Google Maps, the location marker doesn't always get the users address 100% correct every time.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	The users location will appear accurately on the Google Map.

<b>Requirement #:</b>	14
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The homeowner must be able to specify what type of homeless person they would be happy to have staying with them.
<b>Rationale:</b>	Households with young children may not want to accommodate a homeless person who has a criminal past or drug/alcohol addiction.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Only homeless people who meet the homeowners criteria will be able to apply for the vacancy within their household.

<b>Requirement #:</b>	15
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The homeowner must be able to review all applications made for their vacancy.
<b>Rationale:</b>	Rather than the vacancies being first come first serve, as the homeless person will be entering the homeowners' private space, the homeowner should have the ability to first vet the homeless person and ensure that they would be a good fit for the household.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Only the accepted homeless person will be able to stay with the homeowner.

<b>Requirement #:</b>	16
<b>Requirement Type:</b>	Functional
<b>Description:</b>	Homeless users must only see the homeowners' exact location once they have both agreed on going ahead with the process.
<b>Rationale:</b>	To stop homeless people just seeing a vacancy within the application and turning up on the doorstep of the homeowner uninvited.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Homeowners will not experience any unwanted visitors.

<b>Requirement #:</b>	17
<b>Requirement Type:</b>	Functional
<b>Description:</b>	Homeless users must be able to apply for current vacancies.
<b>Rationale:</b>	The basis of the application lies on the ability for homeless people to be able to apply for temporary accommodation.
<b>Dependencies:</b>	Requires the homeless person to have an account to access the application.
<b>Fit Criterion:</b>	The homeless person will obtain temporary accommodation.

<b>Requirement #:</b>	18
<b>Requirement Type:</b>	Functional
<b>Description:</b>	The homeless user must only see, and be able to apply for, vacancies for which they fit the profile.
<b>Rationale:</b>	Rather than the homeowner getting applications that they have no intention of accepting, the users should only be able to interact if they are a match for each other.
<b>Dependencies:</b>	Requires homeowners to have specified a homeless person profile.
<b>Fit Criterion:</b>	No unwanted homeless people will be wasting their time applying for vacancies that they won't get.

<b>Requirement #:</b>	19
<b>Requirement Type:</b>	Functional
<b>Description:</b>	A homeless user must have the ability to have a verified profile.
<b>Rationale:</b>	It would be very easy for the homeless user to lie about any criminal history or drug/alcohol addiction. Therefore if the information that they upload to the site can be verified by a reputable charity, they will obtain a 'verified' symbol that will let homeowners know that their information is trustworthy.
<b>Dependencies:</b>	Requires cooperation from a reputable charity.
<b>Fit Criterion:</b>	Homeowners will have to spend less time fishing through applications if they can see that the information the homeless user is providing is accurate.

<b>Requirement #:</b>	20
<b>Requirement Type:</b>	Functional
<b>Description:</b>	A dashboard must be able to be displayed that will show current nearby vacancies. This could be displayed in the window of charities such as Helm Housing and the Simon Community.
<b>Rationale:</b>	Not all homeless people will have access to the internet. Therefore they will be able to view the vacancies through the window, and go inside to get assistance in applying for the vacancy.
<b>Dependencies:</b>	Requires charities to want to use the application, and display the dashboard in their windows.
<b>Fit Criterion:</b>	Homeless people who have no access to the internet will still be able to benefit from the application.

<b>Requirement #:</b>	21
<b>Requirement Type:</b>	Functional
<b>Description:</b>	Both users must be able to review each other at the end of the stay.
<b>Rationale:</b>	Given the fact that not all information posted on the website will be 100% accurate, a review process at the end of the stay will be essential in helping future homeless people and homeowners' decide upon which vacancies/homeless person to choose.
<b>Dependencies:</b>	Requires the homeowner and homeless person to have taken part in the process.
<b>Fit Criterion:</b>	Users will have a better understanding of who will be a good tenant and who will be a good host.

<b>Requirement #:</b>	22
<b>Requirement Type:</b>	Functional
<b>Description:</b>	Only positive comments will be able to be automatically published.
<b>Rationale:</b>	If the homeowner and homeless person experience some difficulties, either party could relieve their anger by writing a negative review on the website. This would obviously be a major issue as either party could then sue for slander. Therefore an system would need to be in place that will monitor comments.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	No negative comments, that are false, will be displayed on the website.

<b>Requirement #:</b>	23
-----------------------	----

<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application must have an attractive user interface.
<b>Rationale:</b>	A good user interface will provide the user with a good experience when they are using the application.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	A test case of end users will agree that the user interface of the application is attractive, and appealing to use.

<b>Requirement #:</b>	24
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application must be easy to use for people of all age ranges.
<b>Rationale:</b>	Due to the fact that being homeless comes with no bounds of age restriction, neither should the application that aims at helping out homeless people.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Users of all ages will test the application and agree that it is easy to use.

<b>Requirement #:</b>	25
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	Application must work seamlessly across all devices.
<b>Rationale:</b>	Given the amount of smartphones, tablets and phablets we have access to, ensuring cross-compatibility of the application, with no differences per device, is crucial in order to obtain a consistent user experience.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	The application will work seamlessly on a wide variety of tested devices, and will be built with the mind-set of being fully responsive.

<b>Requirement #:</b>	26
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	All information must be backed up regularly.
<b>Rationale:</b>	The process behind the application is based on the ability to obtain, store and display user data. Therefore ensuring that this data is backed up regularly is highly important.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	A check to ensure that the information is being backed up will be carried out.

<b>Requirement #:</b>	27
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application must be fast enough that the user does not grow impatient.
<b>Rationale:</b>	If a user spends too much time waiting on something to update, they will generally not bother using the application.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	A test of the application will be carried out to ensure that it works at a fast speed for a range of different internet connection speeds.

<b>Requirement #:</b>	28
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application must look trustworthy to the users.
<b>Rationale:</b>	As the users are sharing private information such as medical history, criminal history, home address etc, ensuring that the website looks like it can be trusted is key to making sure people actually use the application.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	A test will be carried out to decide if users would trust the application based on its design.

<b>Requirement #:</b>	29
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application should guide the user through the process to ensure that no errors are made.
<b>Rationale:</b>	Not everyone who will be using the application will be computer literate. Therefore, in order to not discriminate against those with less knowledge, the process should be easy and guided.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Users with little computer knowledge will be asked to use the application, and questioned on how easy they found it to use.

<b>Requirement #:</b>	30
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application should be able to be used by people with no prior training.
<b>Rationale:</b>	As the application would be set up as a charitable type organisation, there would be no profit margin that would allow for the training of users to use the application.

<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Users would be monitored using the application to ensure that it can be done without any training.

<b>Requirement #:</b>	31
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application should be available for use 24 hours a day, seven days a week.
<b>Rationale:</b>	Homeless people don't take breaks for holidays, or weekends, and therefore the application should be available to them whenever they need it.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	

<b>Requirement #:</b>	32
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application should work on all of the most popular browsers, including versions that are still widely used.
<b>Rationale:</b>	Not every user will be using the most advanced and up-to-date web browser. Therefore, in order to create a consistent user experience, it must be fully functional on all of the most popular browsers.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	The application will be tested on all of the most popular browsers, including previous versions.

<b>Requirement #:</b>	33
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	Only homeowners who receive applications from a homeless person will be able to view their personal data.
<b>Rationale:</b>	Ensuring that peoples information is used appropriately and not displayed to just any random person is essential in ensuring that the users trust the application and want to continue using it.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	A test will be carried out to try to access the information of any user on the system, without having prior communication to them.

<b>Requirement #:</b>	34
<b>Requirement Type:</b>	Non-functional

<b>Description:</b>	The application should prevent intentional abuse, such as MySQL injection.
<b>Rationale:</b>	Preventing the application from being hacked is essential in securing the trust of the users.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	MySQL injection, and other such security vulnerabilities, will try to be replicated.

<b>Requirement #:</b>	35
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	Users will be informed of information practices before collecting their data.
<b>Rationale:</b>	In order for the user to trust that the application will be using their personal information effectively, they should be informed of exactly how the application intends to use the information.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Without agreeing to the terms of the information practices, the user will not be able to sign up and use the application.

<b>Requirement #:</b>	36
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	Users will be informed of any changes to the information practices.
<b>Rationale:</b>	Keeping the user updated as to how the application uses their personal data is important in keeping the users trust, and also prevents any lawsuits that may come from the unknowing use of information.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	Anytime a change is made to the information practices, the user will be notified and will have to accept the terms before they can continue to use the application.

<b>Requirement #:</b>	37
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	The application will not be discriminatory to any users.
<b>Rationale:</b>	Due to the wide range of users that the application will target, it should not deter any potential users by being in anyway discriminatory.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	

<b>Requirement #:</b>	38
<b>Requirement Type:</b>	Non-functional
<b>Description:</b>	Personal data shall be stored and processed in compliance with the data protection act.
<b>Rationale:</b>	The storing and processing of the users data in compliance with the data protection act is essential in prevent any lawsuits.
<b>Dependencies:</b>	
<b>Fit Criterion:</b>	The storing and processing of data will be checked to ensure that it meets the requirements of the data protection act.

## Appendix G

Risk	Level	Elimination	Backup
Using the Laravel framework as the base for the application.	High	The use of the Laravel framework for development of the application is the biggest risk, as failure to learn this new technology in the specified time could be extremely detrimental to the application. A prototype will be developed that will allow for basic use of the Laravel framework, such as posting data to a database, and then displaying that data to the user. A great tutorial series also exists online called Laracasts that will prove crucial to the development of the application.	If the Laravel framework proves too difficult to overcome in the specified timeframe, the application will therefore have to be developed in vanilla PHP. The functionality that is required has previously been replicated numerous times, with plenty of online tutorials available. However, all of the benefits that come with using the Laravel framework, such as predefined authentication, will have to be hard coded, which will therefore encroach on the development time.

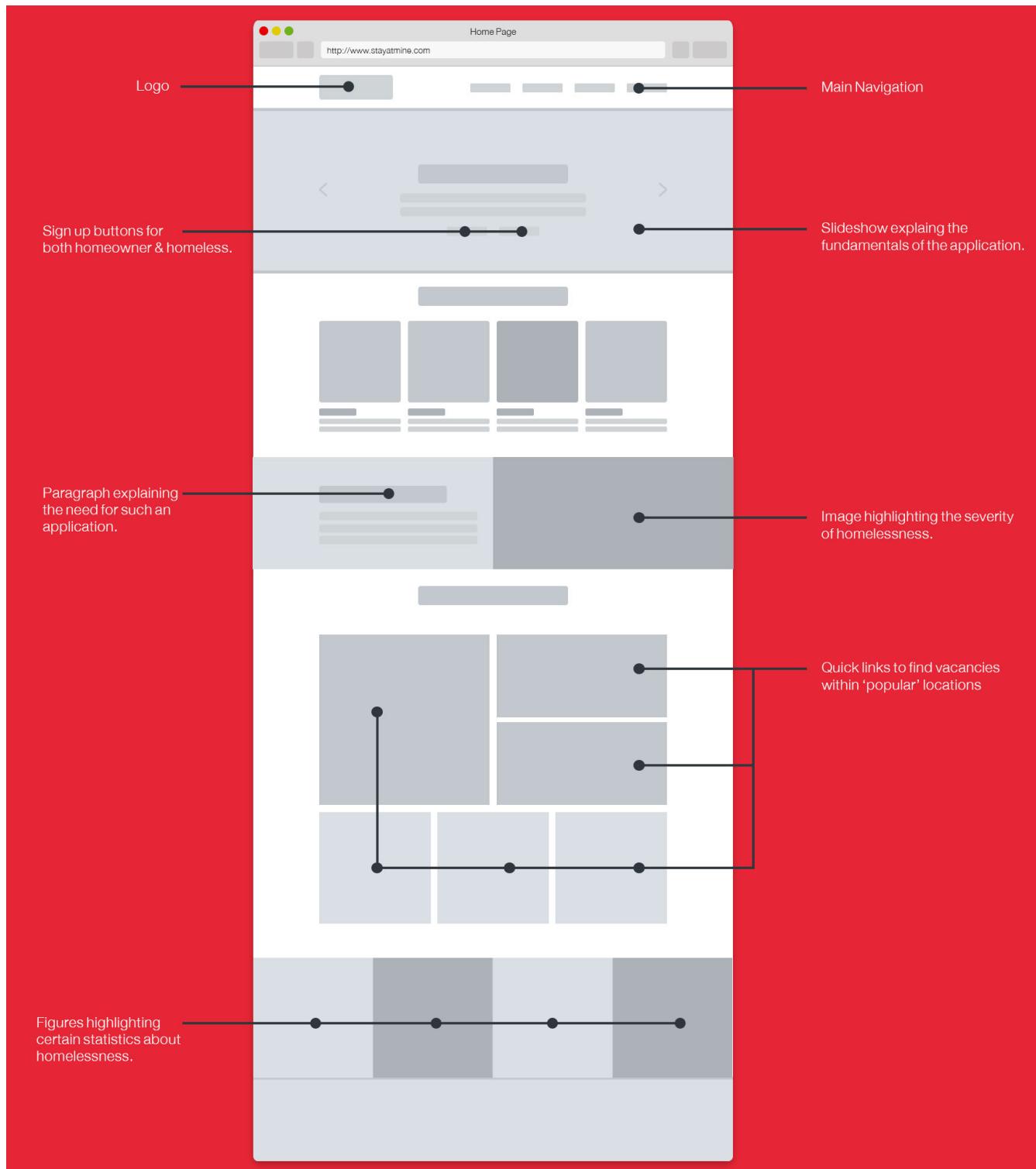
Using JavaScript to dynamically refresh the Charity Dashboard.	High	The ability to dynamically refresh the charity dashboard is crucial in making it a viable aspect of the application. No charitable organisation is going to want to manually refresh a webpage every few minutes to display up-to-date information. A number of tutorials exist online to prove that the functionality exists, and will be used as an aide to developing the functionality.	If the risk cannot be solved within the specified timeframe the decision will have to be made to remove the charity dashboard from the application, as no alternative can be suggested.
Having an automated review moderation process.	High	Allowing users to review each other after each interaction is crucial in ensuring that future users can enjoy a pleasant experience when using the application. To eliminate this risk, each review, when posted, will be scanned for a whole host of key 'derogatory' words. If the review contains any derogatory words it will then have to be reviewed by a human to determine if it can be posted. A number of online tutorials exists that prove that the functionality exists, these tutorials will be used to try to eliminate this risk from the application.	If the risk cannot be solved, the review process will need to be implemented without it, with the view to developing this feature in the future. The risk could be eradicated by removing the feature from the application, however the review process will be a crucial part of ensuring users have a pleasant experience.

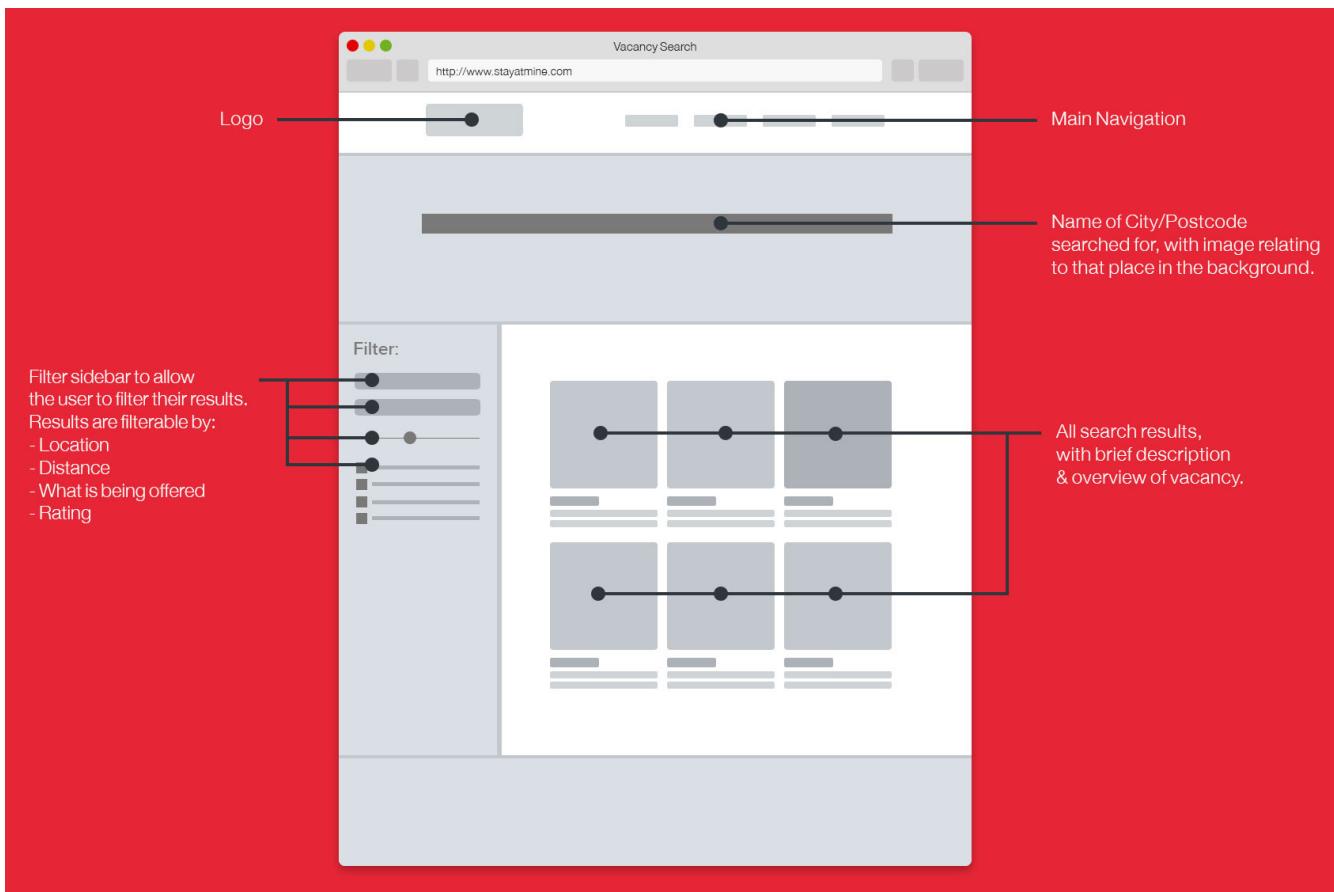
Using Laravel Socialite to allow Facebook/Twitter Login	Medium	<p>Allowing a user to login using one of their social accounts can increase the users experience as no one enjoys filling out menial information about themselves. The Laravel documentation is extremely detailed and should be sufficient in developing this functionality. However, a whole host of online tutorials also exist and can be used as an aide. In terms of the Facebook and Twitter side of things, their SDKs are also extremely detailed, with many users online posting detailed descriptions on how to create ‘apps’ for each platform.</p>	<p>If I am unable to integrate the Laravel Socialite option for allowing Facebook and Twitter login to the application, a number of other solutions also exist for allowing social logins with Laravel. One such option would be to use Adam Wathan’s ‘Eloquent OAuth’.</p>
Integrating Google Maps API to plot location by Postcode.	Medium	<p>In order to provide an accurate visualisation of a homeowners address, it will be plotted onto a Google Map. However, each user isn’t going to want to have to find out their longitude and latitude coordinates in order to do this. Therefore, having the longitude and latitude coordinates generated based on the users postcode would be the best option. This solution has already been implemented before, although the results weren’t always accurate. The Google Maps API is extremely detailed, and a massive community exists, should I run into any problems.</p>	<p>If I am unable to generate the users longitude and latitude from their postcode, within the provided timeframe, a less accurate location, such as the users city/town/street will have to be plotted onto the Google Map.</p> <p>If I am able to get the postcode to generate the longitude and latitude, but unable to increase the accuracy of the coordinates, I will have to include the ability for the user to manually adjust their location pin on the Google Map.</p>

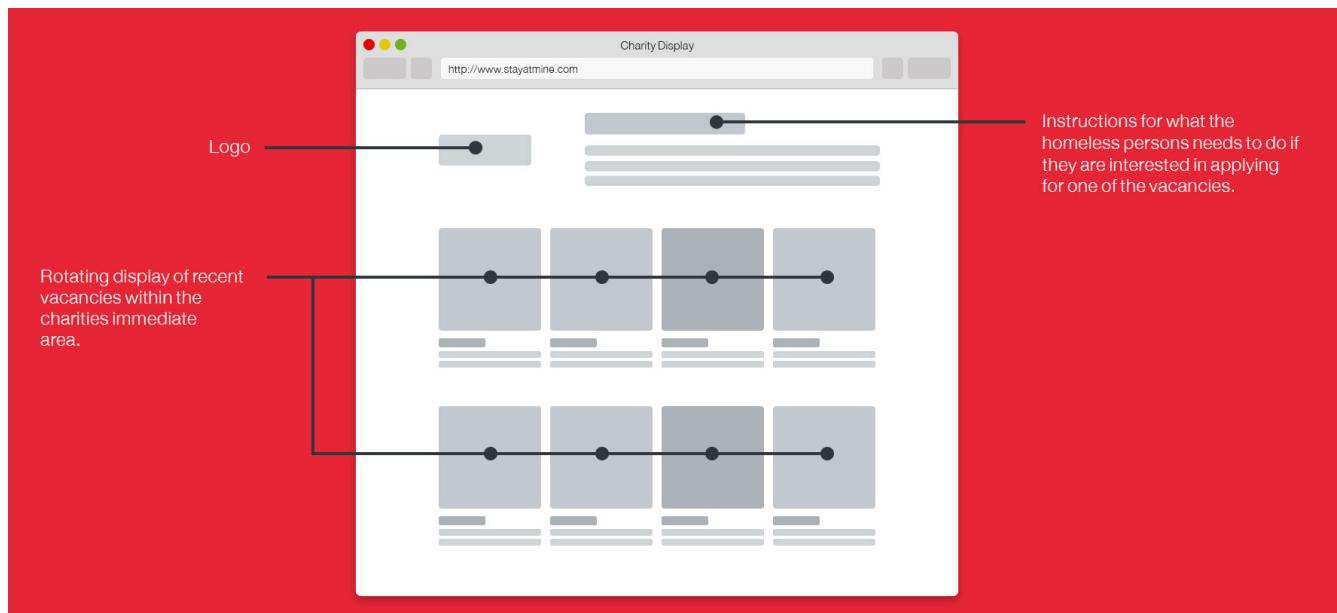
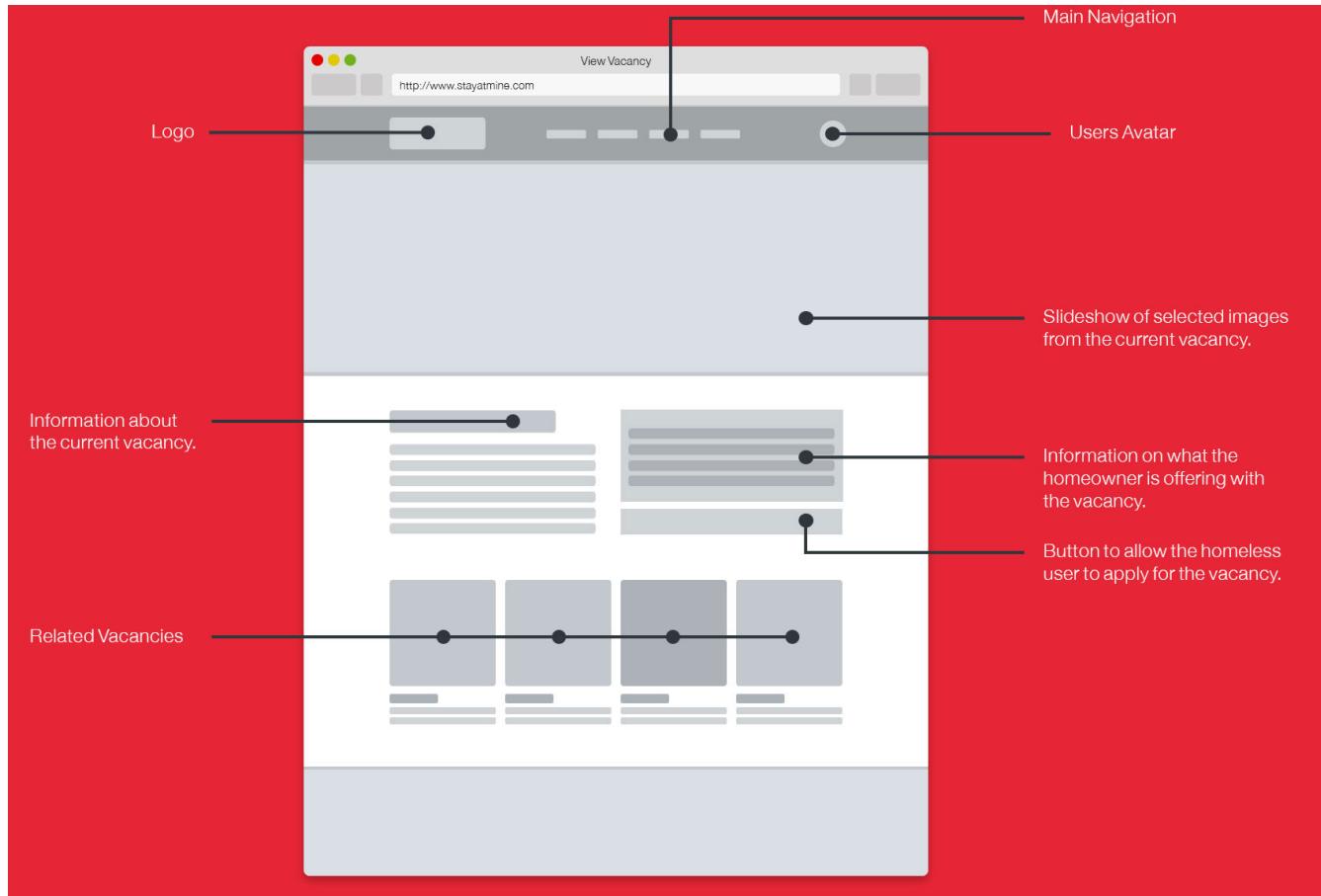
Integrating Google Maps API to provide directions between two specified locations.	Medium	<p>Once two users interact and arrange for the tenancy of a homeless person, there will need to be somehow for them to get to each location. The easiest way to do this would be to provide Google Maps directions between the two concerned parties, allowing for them to choose between modes of transport and alternative routes. A colleague from placement was able to implement this feature into an application, and will be able to point me in the right direction for developing the functionality.</p>	<p>If I am unable to generate directions using the Google Maps API, the another solution, such as 'MapBox' may have to be utilised. Failure to implement this feature in the timeframe will result in it having to be a future feature.</p>
Creating a graphical display of the Homeowners statistics using data from the database.	Low	<p>The homeowner dashboard will display a set of statistics for each homeowner, based on how many vacancies they've listed, how many homeless people they've helped etc. This information will be calculated from the database and displayed graphically using SVG. There are a few online tutorials that promise this as a solution and will be used as an aide to developing this functionality.</p>	<p>If I am unable to dynamically create visualisations with SVG, a less visual option will have to be adopted. This will just be numbers and an image representing what that number is associated with.</p>

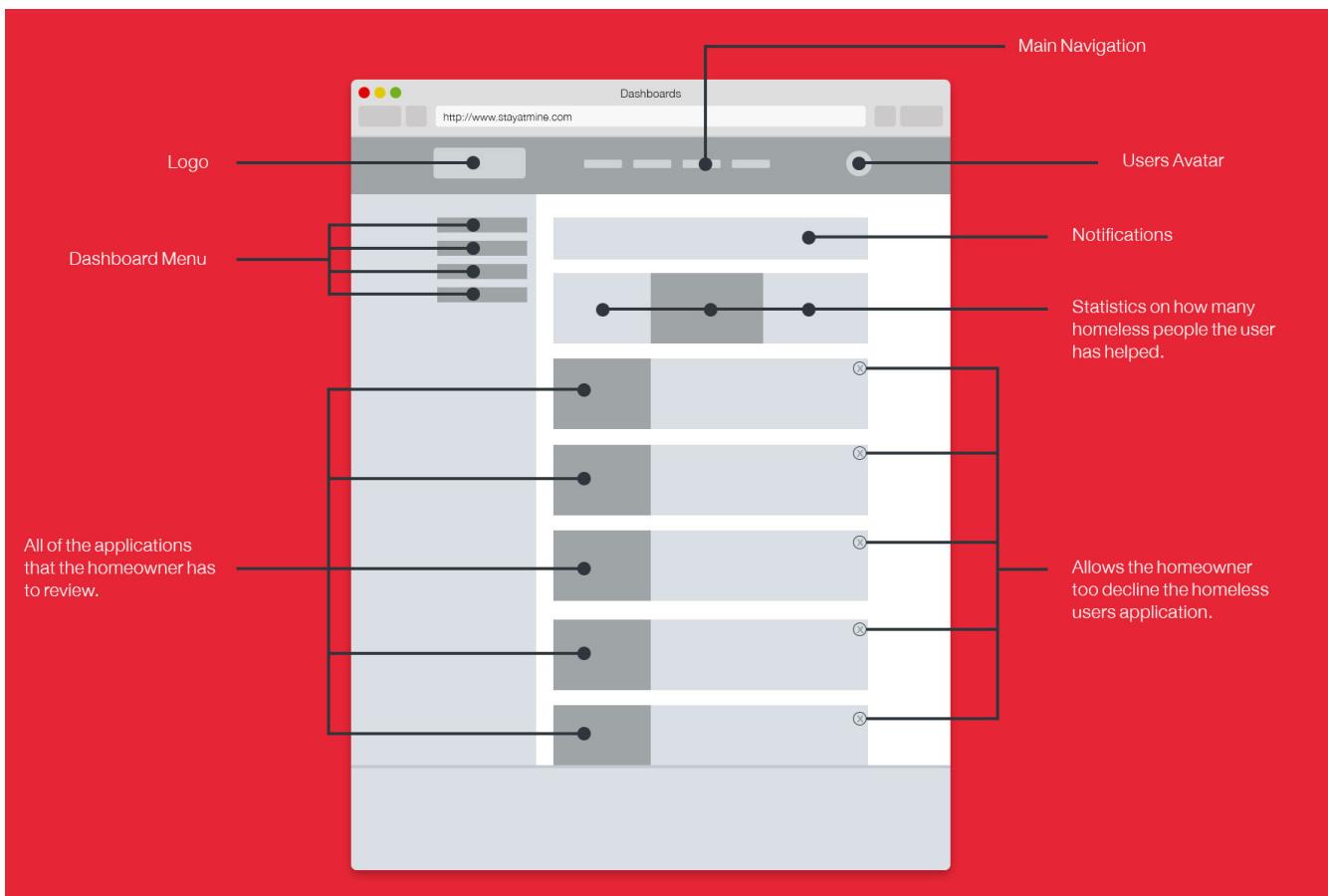
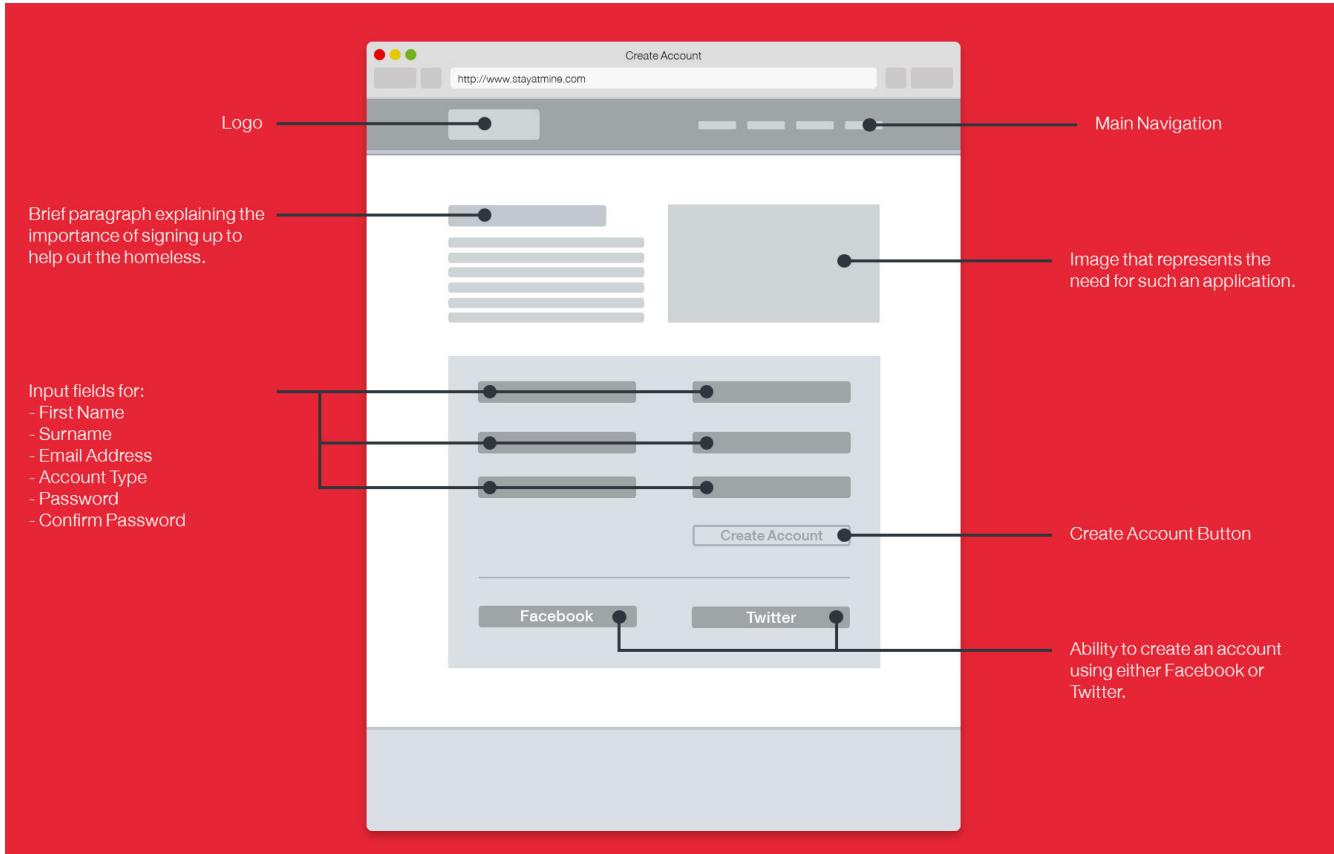
Facebook/Twitter Share	Low	<p>Social media is probably one of the best ways, in todays technological age, to spread the word of new technologies and innovations. Therefore allowing users to share that they are helping out the homeless may cause others to come onboard and use the application. Integrating Facebook and Twitter sharing is something that I have done in the past, and from memory is fairly straight forward. As well as the Facebook and Twitter SDKs, a number of online tutorials also exist that will assist in the development of this functionality.</p>	<p>If I am unable to implement this feature within the specified timeframe, it will have to be left out from the final application. I cannot say any other way to integrate social sharing than with the platforms own SDK, and there would be unable to develop another option. However, this ability is not detrimental to the actual functionality of the application.</p>
------------------------	-----	--	---

## Appendix H









# Appendix I

## Homepage Mockup

The homepage features a large banner with the text "STAY AT MINE" and "2,744 PEOPLE SLEEP ROUGH ON OUR STREETS EVERY NIGHT". Below the banner is a call-to-action message: "Help them sleep well. Open your home and make it one less night. Give a homeless person a great night's sleep - no one deserves to sleep in the cold." Two buttons are present: "I WANT TO HELP" (white background) and "I NEED HELP" (green background). A search bar at the bottom includes fields for "Where are you looking for a vacancy?", "Date", and a green "SEARCH" button.

**Recent Vacancies**  
Our most recent vacancies can be seen below

**Private, comfortable bedroom**  
East Belfast  
2 minutes ago

**Warm room & hot meal**  
Lisburn  
10 minutes ago

**Private, comfortable bedroom**  
Newtownabbey  
12 minutes ago

**VIEW MORE**

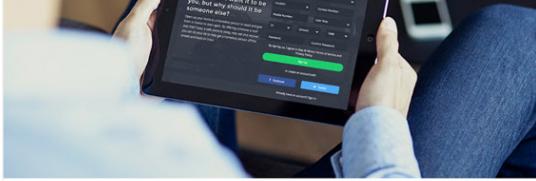
## How It Works

Nobody deserves to be sleeping rough, yet thousands do, let's change that!

**HOMEOWNER**   **HOMELESS**   **CHARITY**

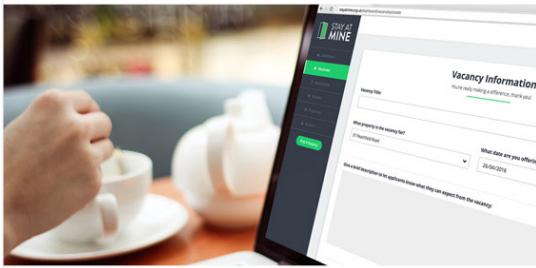


Danister



## Add Properties

Once you've created an account, just add your address to your profile. You can upload some images of the room in your property that you'll be providing. If you own more than one property, you can add multiple properties without needing to create a second account.



## Receive Applications

Once you've posted a vacancy, applicants who match your criteria will be able to apply to take up your vacancy. It is then up to you to decide upon which applicant you'd like to stay with you. It is only once you've accepted an application that applicants will be able to view your address and contact details.



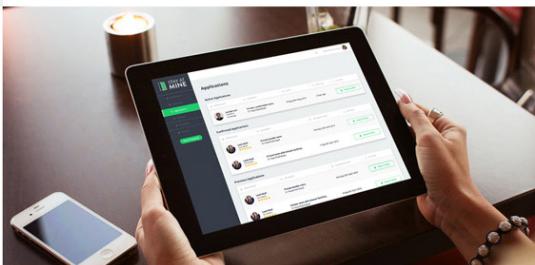
## Register

Creating an account only takes a few minutes. We only ask for some brief details, and we promise these will never be distributed to any third parties.



## Post a Vacancy

After linking a property with your account, you'll then be able to post a vacancy. In posting a vacancy, you'll be able to specify the profile of applicants you'll be willing to accept, as well as what you are willing to provide for them such as a meal, lift and shower facilities.

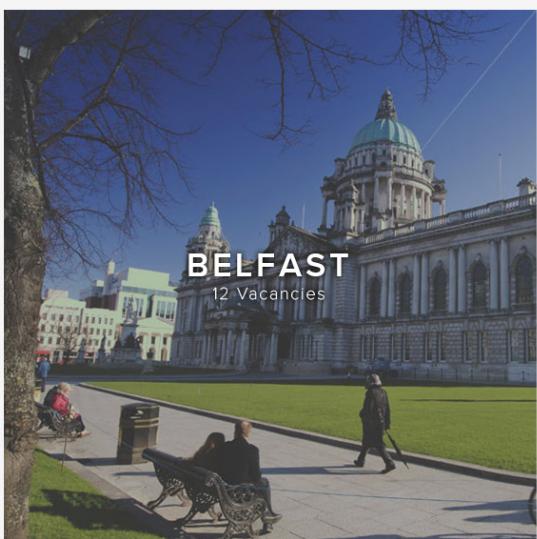


## Help someone

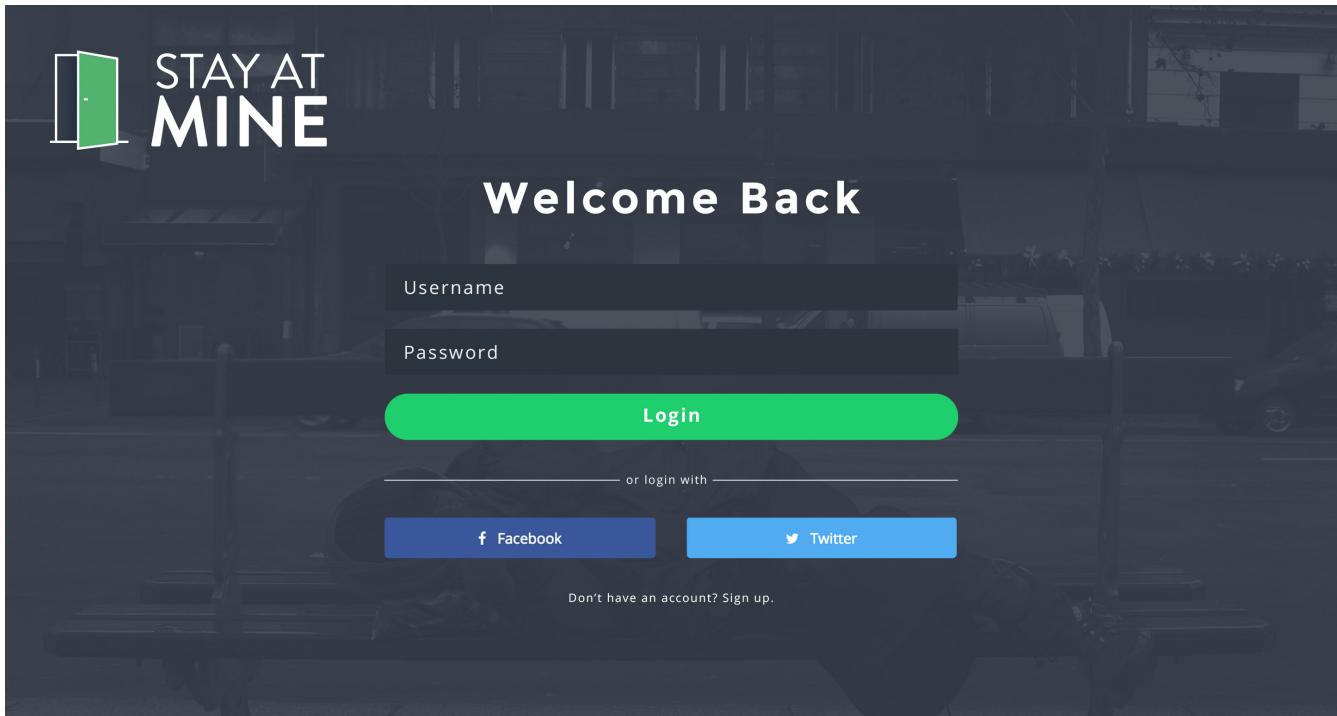
On approving an application, you will be able to get in contact with the applicant and arrange the finer details of the vacancy. After you host a homeless person, you will both be asked to provide a review of each other in order to inform future users.

## Get Sheltered

Find shelter in your local area.



## Login Mockup



The login page features a dark background with a blurred image of a person sitting on a bench. At the top left is the logo 'STAY AT MINE' with a green door icon. The main heading 'Welcome Back' is centered above two input fields: 'Username' and 'Password'. A large green 'Login' button is below them. Below the buttons is a horizontal line with the text 'or login with' and two social media links: 'Facebook' and 'Twitter'. At the bottom is a link to 'Sign up'.

STAY AT  
MINE

Welcome Back

Username

Password

Login

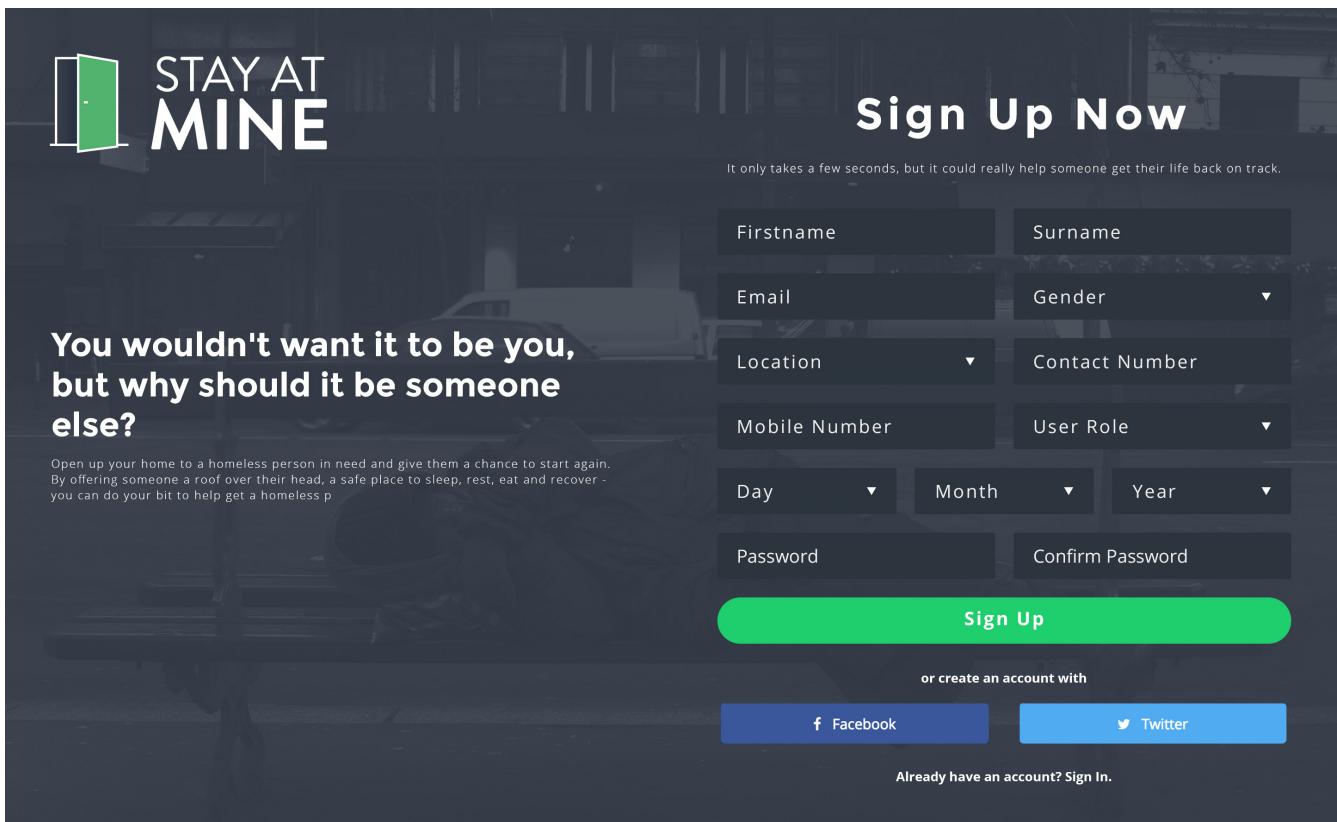
or login with

f Facebook

Twitter

Don't have an account? [Sign up](#).

## Create Account Mockup



The create account page has a dark background with a blurred image of a person sitting on a bench. On the left, there's a message: 'You wouldn't want it to be you, but why should it be someone else?' and a small note about helping homeless people. On the right, the heading 'Sign Up Now' is displayed above a form with fields for Firstname, Surname, Email, Gender, Location, Contact Number, Mobile Number, User Role, and Date of Birth (Day, Month, Year). Below the form is a 'Sign Up' button and a link to 'or create an account with' social media links for Facebook and Twitter. At the bottom is a link to 'Sign In'.

STAY AT  
MINE

Sign Up Now

It only takes a few seconds, but it could really help someone get their life back on track.

You wouldn't want it to be you,  
but why should it be someone  
else?

Open up your home to a homeless person in need and give them a chance to start again.  
By offering someone a roof over their head, a safe place to sleep, rest, eat and recover -  
you can do your bit to help get a homeless p

Firstname

Surname

Email

Gender

Location

Contact Number

Mobile Number

User Role

Day

Month

Year

Password

Confirm Password

Sign Up

or create an account with

f Facebook

Twitter

Already have an account? [Sign In](#).

## Dashboard Mockup

The dashboard features a dark sidebar on the left with the 'STAY AT MINE' logo at the top. Below it is a green navigation bar with the 'Dashboard' option selected. To the right of the sidebar is a main content area titled 'Dashboard'. It includes three cards: one showing '3/4 vacancies filled' with a green circle icon, another showing '34 applications' with a wavy line icon, and a third card on the right titled 'You've helped' with a green '4' and the text 'people get off the streets!'. The main content area also contains a section titled 'Applications' with a table listing two entries. At the bottom of the page is a copyright notice: '© Stay At Mine 2016'.

## Charity Display Mockup

The charity display mockup features a dark background with the 'STAY AT MINE' logo on the left and the word 'Lisburn' on the right. Below the logo is a grid of eight room listings arranged in two rows of four. Each listing includes a small profile picture of a person, a five-star rating, and a brief description. The descriptions are: 'Private, comfortable bedroom Lisburn 2 minutes ago', 'Warm room & hot meal Lisburn 10 minutes ago', 'Private, comfortable bedroom Lisburn 12 minutes ago', 'Private, comfortable bedroom Lisburn 2 minutes ago' (repeated), 'Private, comfortable bedroom Lisburn 2 minutes ago', 'Warm room & hot meal Lisburn 10 minutes ago', 'Private, comfortable bedroom Lisburn 12 minutes ago', and 'Private, comfortable bedroom Lisburn 2 minutes ago' (repeated). Each listing also includes small icons for a bed, a car, and a key.

# Search Vacancies Mockup

STAY AT  
MINE

Login Sign Up

# Search Results

9 Vacancies Available

Filter Vacancies

PROVIDED

- Shower Facilities
- Pick Up
- Meal Provided

ACCEPTED

- Alcohol Abuse Victims
- Drug Abuse Victims
- Criminal History

PETS

- No Pets
- Dogs
- Cats

Update Vacancies

Order By:

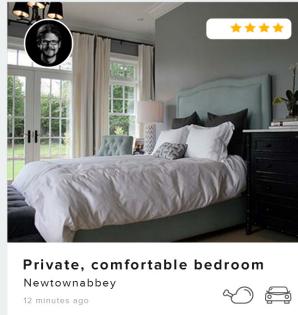
Showing vacancies 1-9 of 9



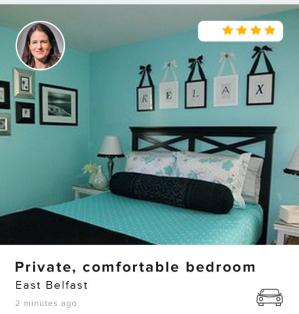
Private, comfortable bedroom  
East Belfast  
2 minutes ago



Warm room & hot meal  
Lisburn  
10 minutes ago



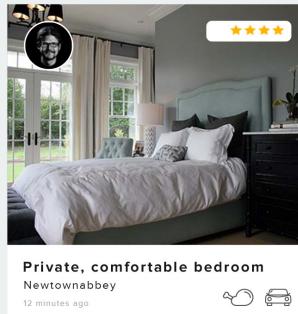
Private, comfortable bedroom  
Newtownabbey  
12 minutes ago



Private, comfortable bedroom  
East Belfast  
2 minutes ago



Warm room & hot meal  
Lisburn  
10 minutes ago



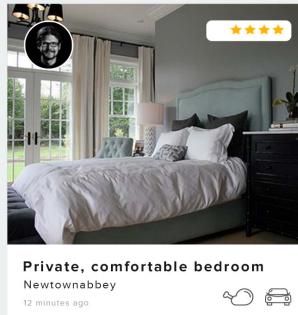
Private, comfortable bedroom  
Newtownabbey  
12 minutes ago



Private, comfortable bedroom  
East Belfast  
2 minutes ago



Warm room & hot meal  
Lisburn  
10 minutes ago



Private, comfortable bedroom  
Newtownabbey  
12 minutes ago

© Stay At Mine 2016. All Rights Reserved.

## View a Vacancy Mockup



f t m

Apply for Vacancy

### Vacancy Title

Posted: 1 week ago | 0 Application(s)

Lorem ipsum dolor sit amet, tantas tacimates perseveris id has, virtute definiebas vituperatoribus mea ut, cum mundi utamur legimus et. Usu ut commodo corpora, purto ipsum posidonium sea no.

Malorum feugiat mea et, sed in summo erant complectitur. Ei vel quem imperdiet dissentias. Id duo dignissim posidonium dissentias, vis eu facer recusabo maiestatis. Cu vel verterem electram posidonium, velit eirmod conceptam ei pro. Omittam patrioque scriptorem his id.



Ryan Warke

★★★★★ (1 ratings)

Lorem ipsum dolor sit amet, tantas tacimates perseveris id has, virtute definiebas vituperatoribus mea ut, cum mundi utamur legimus et. Usu ut commodo corpora, purto ipsum posidonium sea no.

Lorem ipsum dolor sit amet, tantas tacimates perseveris id has, virtute definiebas vituperatoribus mea ut, cum mundi utamur legimus et. Usu ut commodo corpora, purto ipsum posidonium sea no.

#### What's provided

Car - Pick Up

Food - Meal

Shower Facilities

#### Reviews

1 Review ★★★★★



★★★★★

Fantastic host! If I was invited back, I would definitely return. Would 100% recommend to anyone!

Leah Wall

April 2016

## Appendix J

<b>Test ID</b>	<b>Description</b>	<b>Outcome</b>	<b>Response (if failed)</b>
<b>1</b>	The application will allow the homeowner to create an account for the application.	<b>Passed</b>	
<b>2</b>	The application will allow a homeless person to create an account for the application.	<b>Passed</b>	
<b>3</b>	The application will allow a charity/organisation to create an account for the application.	<b>Passed</b>	
<b>4</b>	The create account form will be validated for errors.	<b>Passed</b>	
<b>5</b>	The application will allow the user to login as either a homeowner, homeless, or charity user.	<b>Passed</b>	
<b>6</b>	The application will allow the user to reset their password if they forget it.	<b>Passed</b>	
<b>7</b>	The application will allow the user to delete their account if they do not wish to use the application anymore.	<b>Passed</b>	
<b>8</b>	The application will allow the user to update their account details.	<b>Passed</b>	
<b>9</b>	Homeowner users must be able to post a new vacancy.	<b>Passed</b>	

<b>Test ID</b>	<b>Description</b>	<b>Outcome</b>	<b>Response (if failed)</b>
<b>10</b>	A homeowner must have the ability to upload photographs of the vacant room.	<b>Passed</b>	
<b>11</b>	The homeowner must be able to add their address as a location marker on a Google Map.	<b>Passed</b>	
<b>12</b>	The homeowner must be able to seamlessly add their address to a Google Map using just their postcode.	<b>Passed</b>	
<b>13</b>	The homeowner must be able to adjust the Google Map pin, if it is placed inaccurately on the map.	<b>Passed</b>	
<b>14</b>	The homeowner must be able to specify what type of homeless person they would be happy to have staying with them.	<b>Passed</b>	
<b>15</b>	The homeowner must be able to review all applications made for their vacancy.	<b>Passed</b>	
<b>16</b>	Homeless users must only see the homeowners' exact location once they have both agreed on going ahead with the process.	<b>Passed</b>	
<b>17</b>	Homeless users must be able to apply for current vacancies.	<b>Passed</b>	
<b>18</b>	The homeless user must only see, and be able to apply for, vacancies for which they fit the profile.	<b>Passed</b>	

<b>Test ID</b>	<b>Description</b>	<b>Outcome</b>	<b>Response (if failed)</b>
<b>19</b>	A homeless user must have the ability to have a verified profile.	<b>Failed</b>	Future development.
<b>20</b>	A dashboard must be able to be displayed that will show current nearby vacancies. This could be displayed in the window of charities such as Helm Housing and the Simon Community.	<b>Passed</b>	
<b>21</b>	Both users must be able to review each other at the end of the stay.	<b>Passed</b>	
<b>22</b>	Only positive comments will be able to be automatically published.	<b>Failed</b>	Future development.

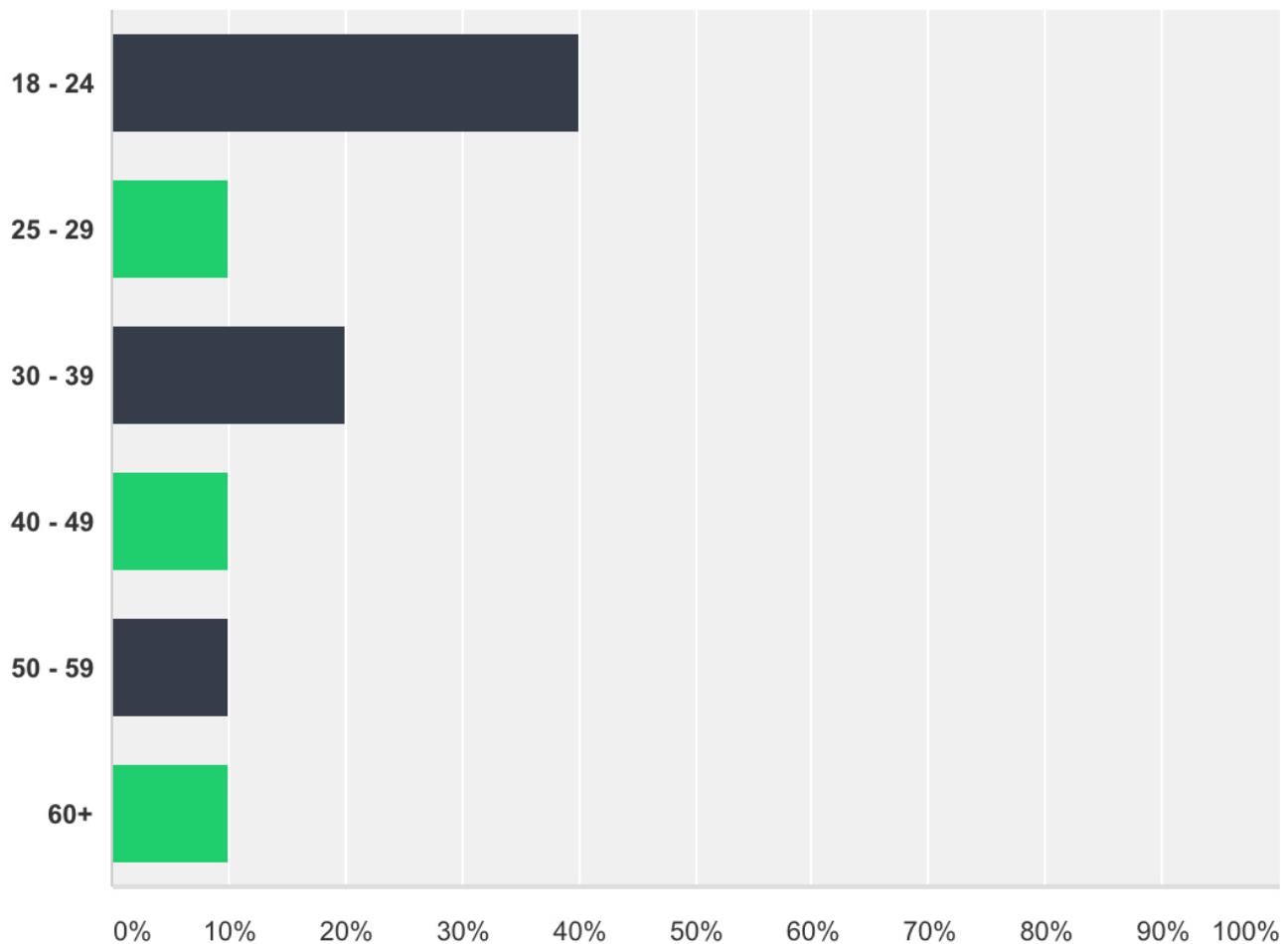
## Appendix K

Browser	Outcome	Response (if failed)
Chrome	Passed	
Safari	Passed	
Firefox	Passed	
Opera	Passed	
Internet Explorer 11+	Passed	
Safari Mobile (iOS)	Passed	Minor CSS changes. Have been amended.

## Appendix L

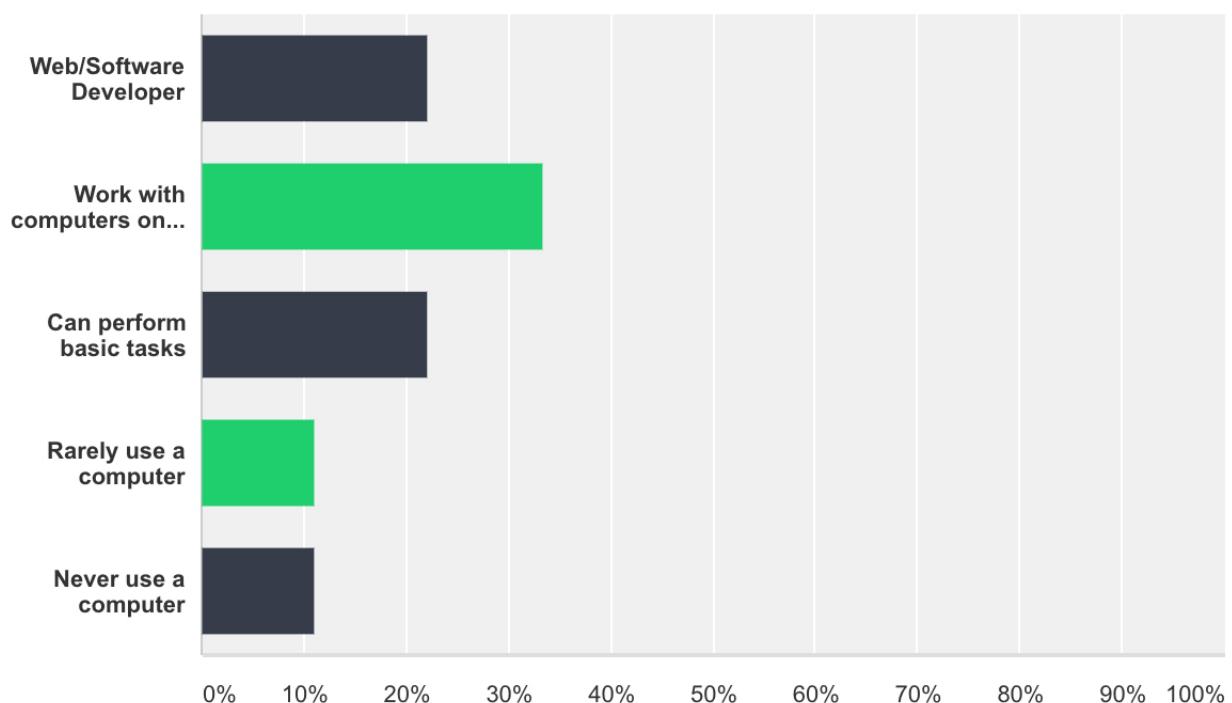
### What is your age range?

Answered: 10 Skipped: 0



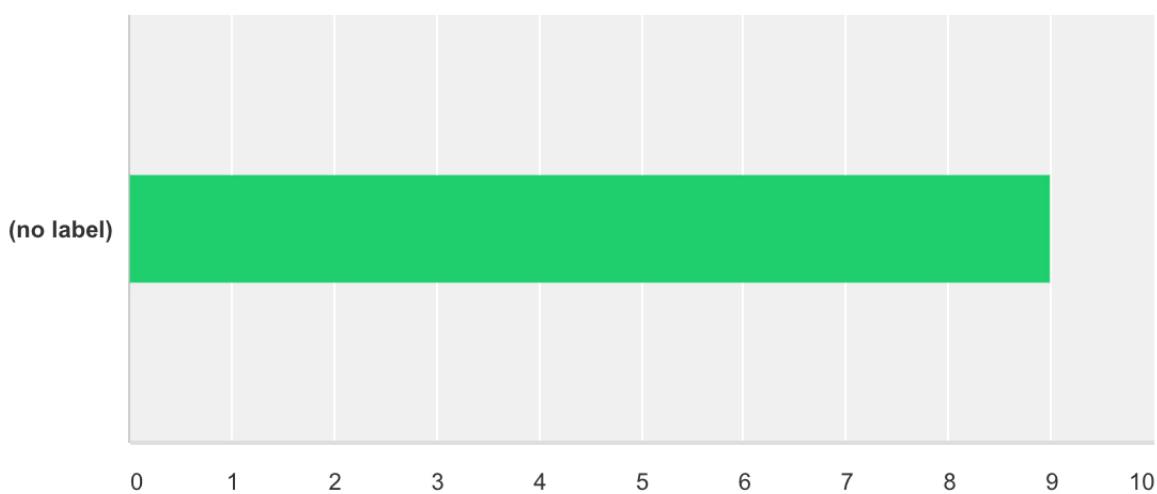
## What is your computer literacy?

Answered: 9 Skipped: 1



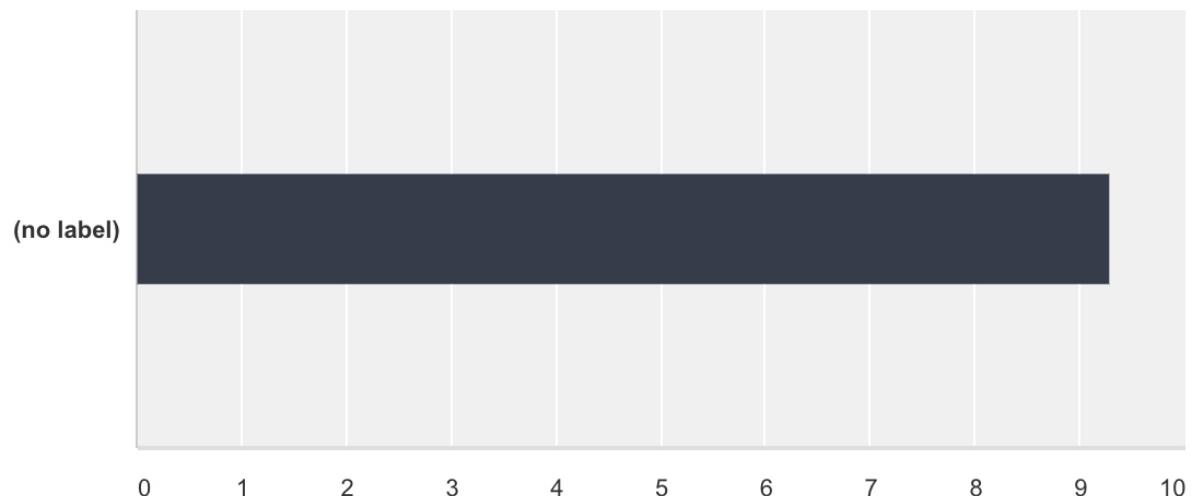
**On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how attractive would you say the Stay At Mine application is?**

Answered: 10 Skipped: 0



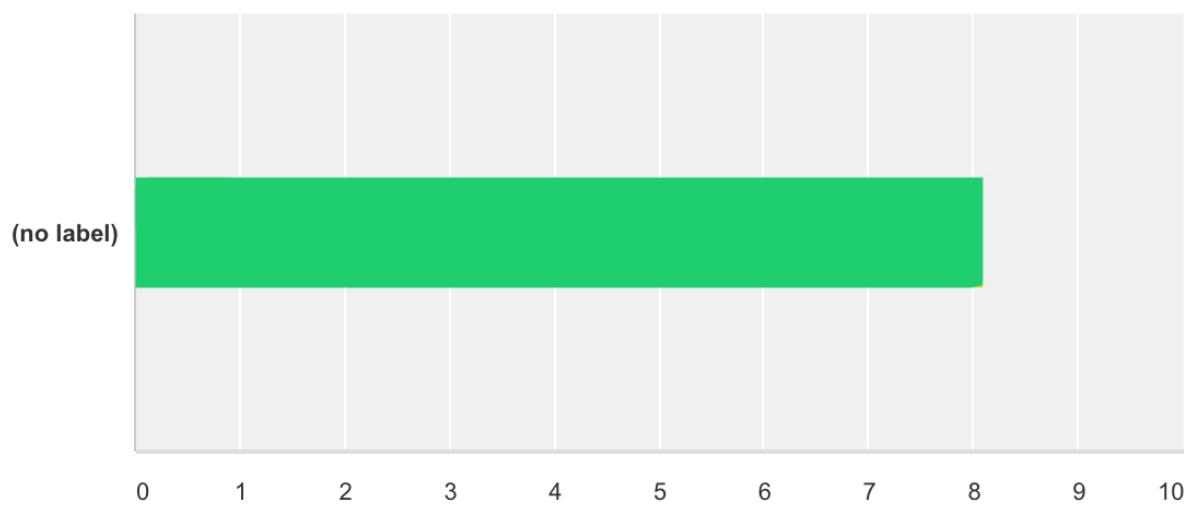
**On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how trustworthy would you say the Stay At Mine application is?**

Answered: 10 Skipped: 0



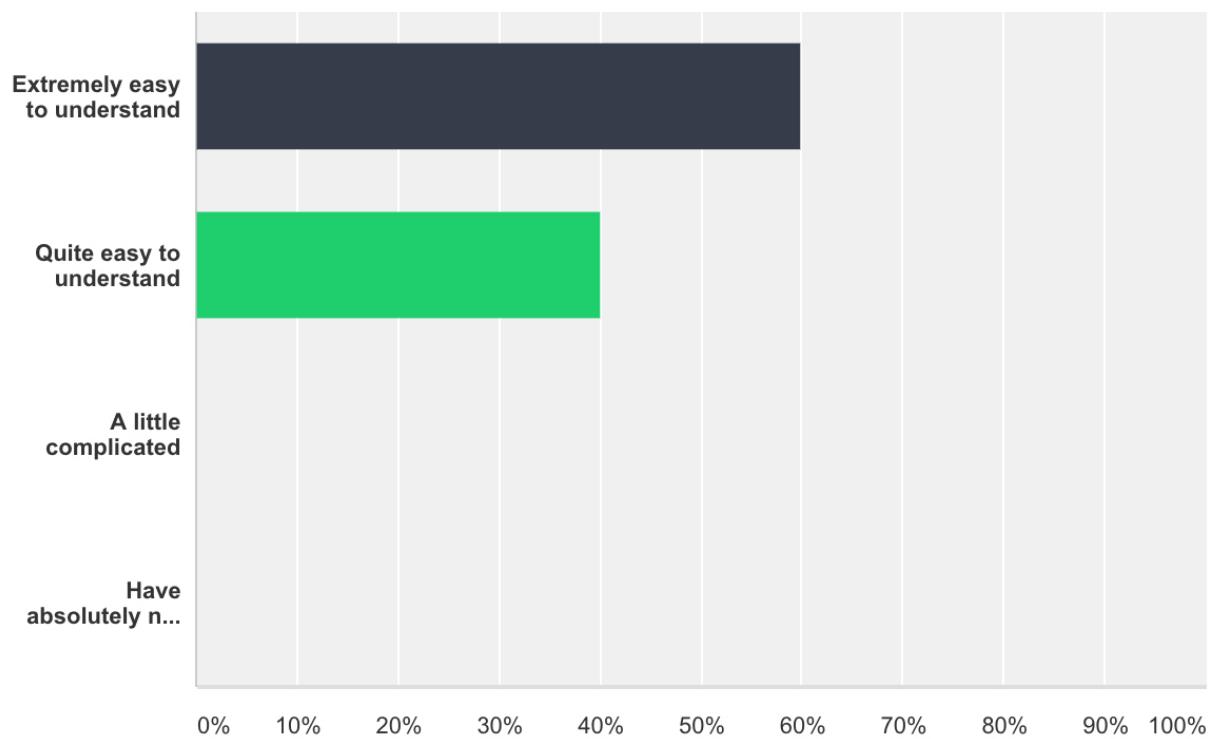
**On a scale of 1-10 (with 1 being the lowest and 10 being the highest), how confident are you that you would be able to use the core functionality of the Stay At Mine website without any prior training?**

Answered: 10 Skipped: 0



## How easy would you say it is to understand what the purpose of the Stay At Mine website is?

Answered: 10 Skipped: 0



## **Appendix M**

### **Research Integrity Certificate**

4/27/2016

Blackboard Learn

*This is to certify that*

# Ryan Warke

*has successfully completed*

**Research Integrity (taught courses)**

*on*

**25 March 2016**