HouseProud:
An Online Booking Web Application for Tasks Around the Home

A dissertation submitted in partial fulfilment of

the requirement for the degree of

MASTER OF SCIENCE in Software Development

In

The Queen’s University of Belfast

By

Justin Kelly 40214842

Date of Submission: 03 October 2018

School of Electronics, Electrical Engineering & Computer Science

CSC7057 – INDIVIDUAL SOFTWARE DEVELOPMENT PROJECT

A signed and completed cover sheet must accompany the submission of the Individual Software Development dissertation submitted for assessment.

Work submitted without a cover sheet will **NOT** be marked.

Declaration of Academic Integrity

Before signing the declaration below please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook.
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.
6. Software and files are submitted on a memory stick.
7. Journal has been submitted.

I declare that I have read both the University and the School of Electronics, Electrical Engineering and Computer Science guidelines on plagiarism - http://www.qub.ac.uk/schools/eeecs/Education/StudentStudyInformation/Plagiarism/ - and that the attached submission is my own original work. No part of it has been submitted for any other assignment and I have acknowledged in my notes and bibliography all written and electronic sources used.

I am aware of the disciplinary consequences of failing to abide and follow the School and Queen's University Regulations on Plagiarism.

Name: (BLOCK CAPITALS) **JUSTIN KELLY**

Student Number: **40214842**

Student's signature ____________________ *Date of submission* 03/10/2018

Acknowledgements

I would like to thank my project supervisor, Angela Allen for all her support and guidance throughout the duration of the project. My family and friends for their continuing support. Finally, I would like to thank my wife Belinda who during the duration of this project give birth to our second child and made it possible for me to commit the time and effort for this project and the MSc Software Development course in general without her this would not have been possible.

Abstract

Name: Justin Kelly

Student Number: 40214842

The HouseProud system is a web application that connects users with contractors to outsource tasks needing done around the home. From gardening to window-cleaning users can outsource tasks by completing the ‘Post a Job’ form which consists of a title, description, due date and a fair budget which allows registered contractor to view job and make an offer which will be accepted or rejected by the user. Key skills employed in delivering the project include development platforms PHP, JavaScript and general project management within an agile development model.

Table of Contents

APPENDICES

1.0 Introduction

A proposal was put forward to a project team to commission an application called HouseProud which will be an online booking service for handy person services. For example, these jobs include gardening, window cleaning, home cleaning, furniture assembly and general jobs around the house that the homeowner may not have the skills or the time to do. The idea of this application is to be able to make quick and easy bookings with trusted contractors.

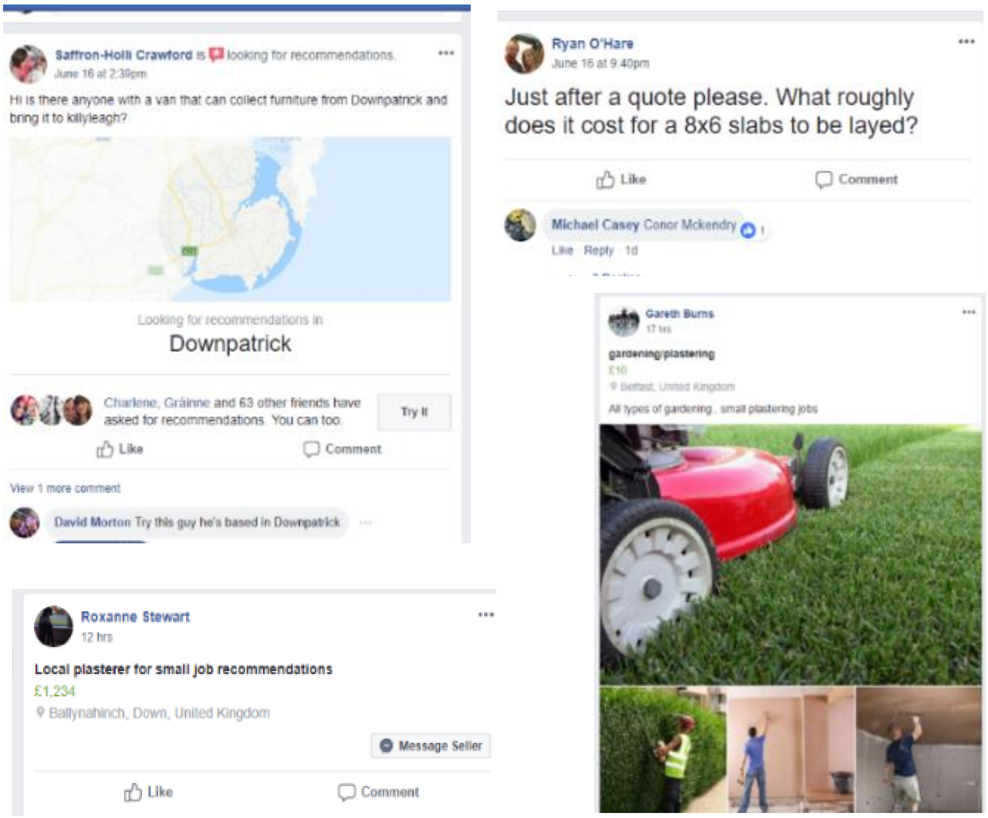
Perhaps a household needs a window cleaner; the guttering needs to be cleaned out; or the garden fence has broken in the back yard. These are just a few examples of the everyday tasks that potentially need done around the house. Do you know someone who you could rely on to do these jobs for you? Maybe, or maybe not. The fact is many people do not and are wary when it comes to hiring someone to do a job for them and worry about how much it will cost or whether the contractor will do a good job.

1.1 Problem Specification

The proposed solution to these problems is to create an application, so people can simply go to one site for all their everyday tasks and hire trusted contractors to complete the job for a fixed price.

Currently many people go through different channels such as Facebook or Gumtree to try and find someone to do the required task, where an example of such a listing (from a Facebook post) is presented in Figure 1.1.

Figure 1.1: Examples of Job Requests in a Local Facebook Page



Saffron-Holli Crawford is looking for recommendations.
June 16 at 2:39pm

Hi is there anyone with a van that can collect furniture from Downpatrick and bring it to killyleagh?

Looking for recommendations in
Downpatrick

Charlene, Gráinne and 63 other friends have asked for recommendations. You can too.

View 1 more comment

David Morton Try this guy he's based in Downpatrick

Ryan O'Hare
June 16 at 9:40pm

Just after a quote please. What roughly does it cost for a 8x6 slabs to be layed?

Michael Casey Conor Mckendry
Like · Reply · 1d

Gareth Burns
17 hrs

gardening/plastering
£10
Belfast, United Kingdom
All types of gardening.. small plastering jobs

Roxanne Stewart
12 hrs

Local plasterer for small job recommendations
£1,234
Ballynahinch, Down, United Kingdom

However, based on experience and anecdotal information, problems regularly occur when procuring services in this manner. For example, being unable to view the profile of the person being recommended; no idea of the price until contractor arrives on site (users may find it difficult to reject price when that person is already at their home); and no idea if the person is qualified to do the job.

1.2 Project Aim

The overall aim of this project is outlined as follows:

> *"To develop an application which allows users to post jobs requiring done around the home and decide which offer to accept from trusted contractors".*

1.3 Project Objectives

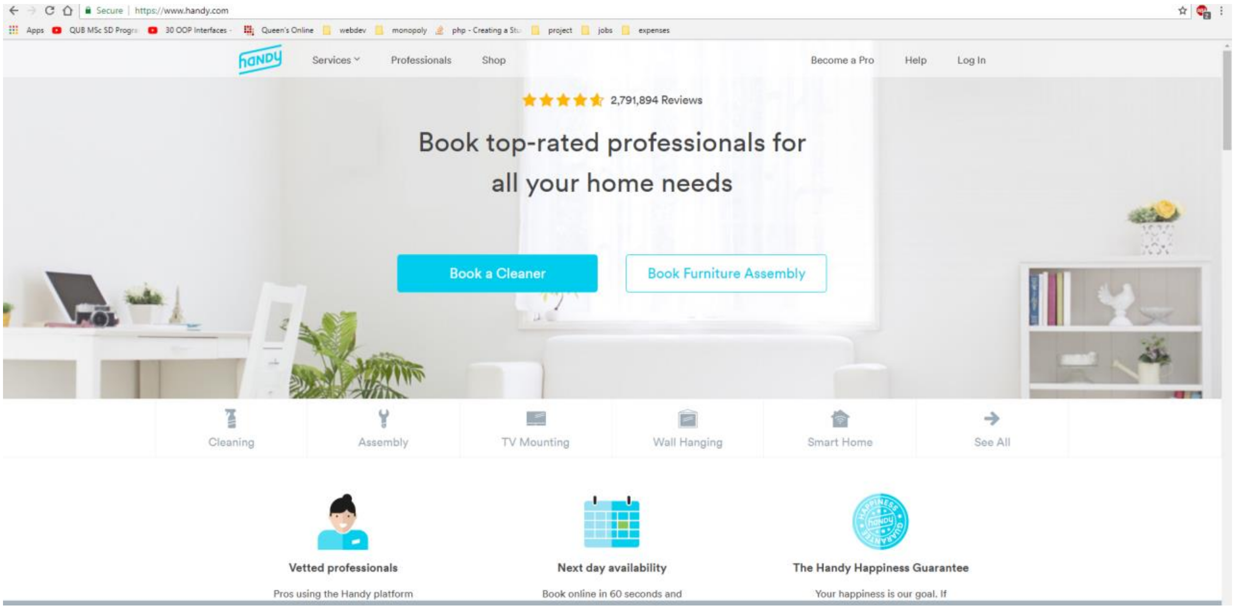
The objective of this web application is to help provide a bridge between skilled people and customers who are unable or unwilling to do to the odd jobs that are part and parcel of maintaining a home. This platform will be mutually beneficial to each party as tradespeople will be able to earn extra money without the hassle and expense of advertising, while the customer will have a professional complete tasks saving them the trouble of doing it themselves. From a contractor's point of view, especially those who are small and local based, trying to enhance their reputation is always difficult and many try different ways to advertise their business. These include business cards, newspapers, and websites such as Gumtree and Facebook. However, these techniques are quite often unsuccessful. The most common way in which many contractors achieve more demand is through 'word of mouth' from clients they have already completed a job for.

One of the key features on the web application is a rating/ review system where previous clients will be encouraged to leave feedback of their experience with the tradesperson. This is a great way for potential clients to view contractor's previous jobs and judge for themselves if they are the right person for the job.

The rating system will also include a profile picture and biographical description of the contractor (bio), that will give the user that extra bit of reassurance which the likes of websites like Gumtree cannot provide. The platform will be simple and basic so that tradespeople with even a low level of experience with such technology will be able to use. The option to communicate easily between tradespeople and potential clients will be provided, where they can discuss the finer details of each job before committing to an offer.

An extensive internet search (using Google) did not identify any applications currently available in Northern Ireland that provide this kind of service. However, there are similar applications available in other countries, such as handy.com (Figure 1.2 which covers USA, Canada and part of the UK) and localheroes.com (which cover part of the UK). It was reported in 2016 that handy.com generated one million dollars per week. In 2017 it was reported that the estimated value for this company was five hundred million dollars (Kosuff 2015). Obviously with countries like America, Canada and major cities like London the demand for these services will be on a larger scale and generate more income but there is still a demand in this country for such services as every homeowner, tenant, landlord or even business owner often need tasks done which they themselves are unable to carryout for various reasons.

Figure 1.2: Screenshot of Homepage for handy.com



2,791,894 Reviews

Book top-rated professionals for all your home needs

Vetted professionals

Pros using the Handy platform

Next day availability

Book online in 60 seconds and

The Handy Happiness Guarantee

Your happiness is our goal. If

The proposed HouseProud application will help solve issues users have with finding the right person/company by creating a one stop application where all the relevant bodies can be found depending on the type of job and proximity to the job in question. With a simplified estimated pricing system, it's a no obligation way of getting a price for a specific job. If the user doesn't want the work to be done, no problem. This will avoid face to face rejection, which some may find difficult whilst someone is in their home.

Another objective of this project is to create a user-friendly application and aim to cut out the endless searches for people to hire for everyday jobs around their house. The proposed application intends to create an environment where users post a job they are needing completing and be able to go through a list of contractors and pick who they believe will be best fit for the job at hand.

With the benefit of today's technological advancements, it is much simpler to create an application to make such bookings online. This can be seen by the success of booking.com, uber and Airbnb which allow users to complete the required process quickly and easily.

As previously mentioned, there is no simple online booking application for these types of services in this country, this type of software will provide users with assurances and confidence on being able to find the right professional to do the job. By providing an easy to use user interface (UI) and by allowing users to get as much information as possible without being overbearing, even individuals with limited knowledge of technology will be able to book an appointment with relative ease.

An identified problem in this sector is pricing of jobs, as many people are often worried about what something will cost. However, an upfront pricing strategy in place will offer peace of mind to users. Another common problem when using contractors is missed appointments due to contractors not turning up or consumers giving wrong information. With the HouseProud application, all relevant information and feedback on contractor and user performance will be recorded and displayed providing transparency throughout.

From the prospective of the professional, contractors will be able to pick the jobs which suit them best. A contractor can easily earn extra money from these one-off jobs and build up their reputation with every job completed successfully.

With this application there is going to be more flexibility in terms of booking an appointment, previously you could only contact and try organising jobs within business hours however through the applications bookings will be able to be made at any time of the day.

1.4 Target Users

Primarily this application will be for homeowners, tenants, landlords and business owners. For those who have little time or lack the ability for whatever reason to complete maintenance/cleaning etc around the house, be it a very small job to a larger scale tasks, this application will be ideal for them. It will cut out the need for people to trawl the internet, make calls, carry out endless searches, etc looking for the best person suitable.

The application will ensure that it is only genuine and trusted contractors who can complete jobs and tasks that is required from the users. Users will be able to use this application via their smartphone, tablet or laptop being able to complete a simple process for a job request in a matter of minutes.

When contractors register with the site they must first go through a vetting process that includes the contractor providing evidence of a background check (i.e. a recent Access NI certificate), identification verification, etc. They will set the jobs in which they can carry out e.g. removals, gardening, window cleaning etc and the locations in which they can cover.

Users will have to register with the site before being able to request jobs. Once registered they will be able to describe the job which they require carried out, the location, and suggest a fair budget for the job in addition to a reasonable deadline in which they are looking the job completed by.

Once a job is posted to the site an alert will be sent out to the contractors who have indicated (when they registered) that they are able to carry out the job, they are invited to provide a no obligation quote. This allows the user to look at profiles and review the contractor to decide who they want to do the job.

Once the job is assigned the user and contractor can directly message one another through the web application to sort out details. Once the job is complete payment is processed and the user is free to leave a review for the contractor so other users can view.

1.5 Project Plan

A project programme, displayed as a Gannt chart, is presented in Figure 1.3, outlining the plan which the team intend to follow throughout the project delivery cycle.

The key to a successful project is in the planning. Creating a well thought out and realistic project plan helps establish project goals, deliverables and schedule. It is the hope that with setting out clear and achievable deadlines and plans for a specific task the end goal will be successfully completed within the given time frame.

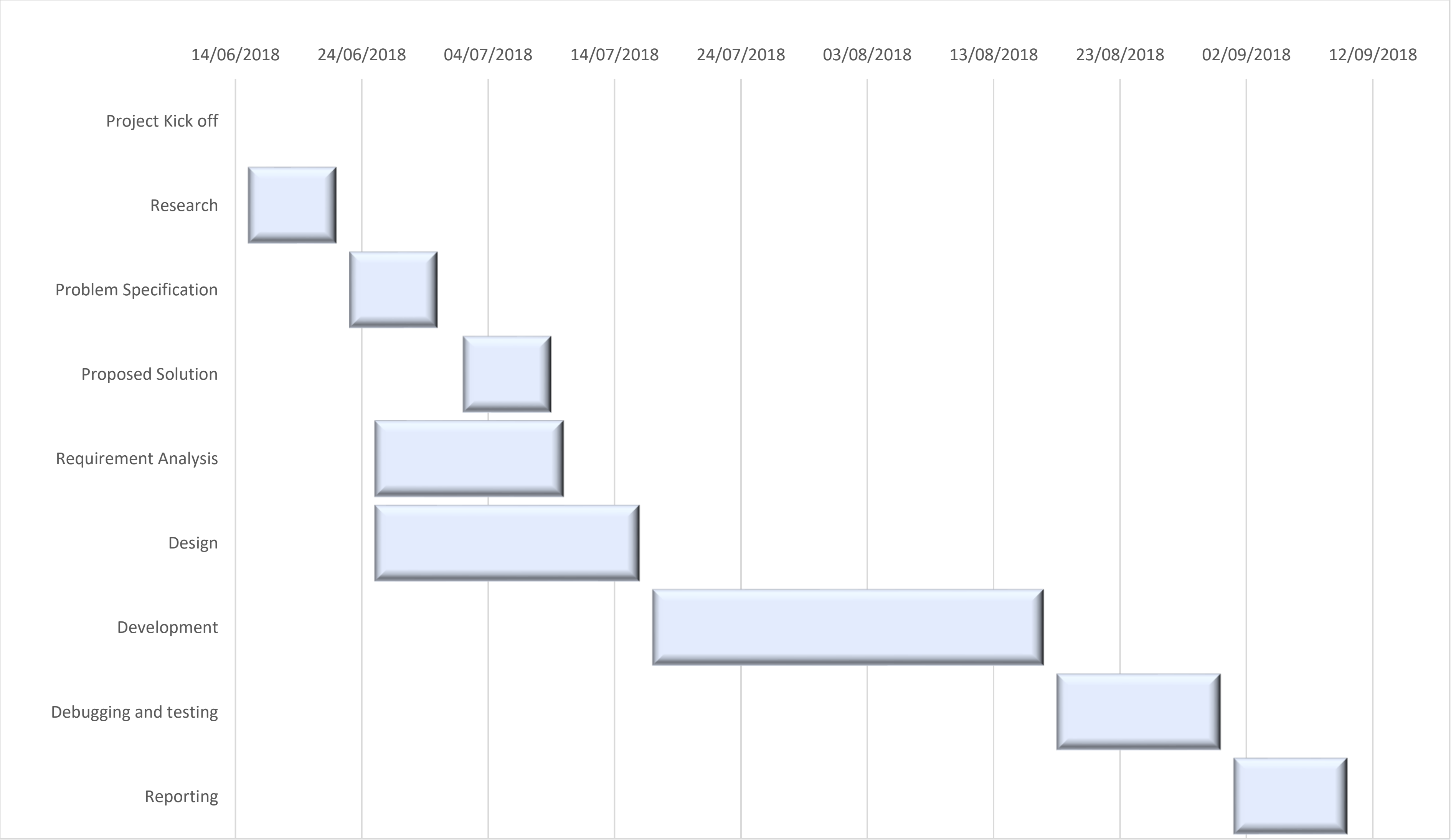
The team plan to use this project plan as a guide throughout the duration of the project. Project plans can often change but hopefully with the benefits that come with a clear plan from the outset, a project's success will be more likely. The chart below demonstrates how tasks have been divided and assigned deadlines to each respective stage.

With any good plan, potential risks must be considered from the beginning. At this early stage, I have identified the following potential problem areas:

- During the development stage, problems may arise when trying to develop functions which may be outside of the developer's area of experience. This may make a task last longer than expected.
- Another problem area would be at the debugging and testing stage as the allocation of resources may not be sufficient to carry extensive testing.

As the project progresses, the programme will be reviewed throughout its duration to assess how closely each stage set out in the plan is being followed. This will be essential in monitoring progress and to ensure the project will be completed by the predicted deadline.

Figure 1.3: Gannt Chart Outlining Project Plan



2.0 Proposed Solution and Justification of Development Model

This section discusses the proposed solution outlined in Chapter 1, giving explanations and insight into certain decisions.

2.1 Application Proposal

The proposal at this stage is to create an application where users and contractors can register with the site. Once registered, users would be able to post the relevant details of a specific job which they need completing. This would include a job title, job description, job category, address, budget and so on. This information then becomes available to the contractors who work in that area and who have the required expertise to complete the task. The contractor can decide to place an offer on the job and then it is up to the user to accept or reject the proposal. As previously stated, with no such application currently up and running for this area, it would provide a simple solution to a problem which a lot of households in Northern Ireland currently have when struggling to find the right person/people to carry out household tasks.

One of the main motivations behind this proposal was from potential users who expressed how difficult it was for people to find the right person to do a specific job. This application will offer the user the benefit of connecting quickly with suitable tradesmen/women, with the added reassurance of seeing their customer reviews before committing. The developer is looking to create an application where users can sign up and within two minutes be able to post the job in which they are looking someone to do. It is essential that the process for posting a job is not a long winded or complicated one as users would become disinterested rendering the application unsuccessful.

2.2 Proposed Design

This section will discuss the factors to consider when developing an outline design, these include the following:

- Core Users;
- Growth in Technology;
- Development Platform.

2.2.1 Core Users

The core user group within the application will predominantly be home owners and contractors whose skillsets will vary when it comes to using technology. A simple and easy to use application is essential. This will be taken under consideration in the design stage.

2.2.2 Growth in Technology

The market for smartphones, tablets and laptops continues to grow year after year. A survey carried out in 2016 shows 89% of adults in the UK now use the Internet weekly (ons.gov.uk) this is an increase of 38% from 2006. The market continues to grow with the internet available nearly everywhere you go. The internet provides huge benefits to businesses for communication as it is the easiest way for customers and clients to connect. People nowadays are much more likely to use the internet for tasks like insuring their car or booking a holiday rather than by phone call.

Long gone are the days of searching through the 'Yellow Pages' looking for someone to carry out a job. With such significant and continued growth in technology, it is essential that businesses use this to their advantage. As Bill Gates once said, "If your business is not on the internet, then your business will be out of business".

The idea of this application is that it will do the advertising for the contractors. This could be beneficial to the smaller contractor as they will often be unsure of which channels to use when it comes to promoting their own business and they can often be quite hesitant before spending money on advertisement that may or may not be successful.

2.2.3 Development Platform

Before commencing development, the developer had to decide on the development platform, essentially which type of application should be provided, i.e. a web application or a mobile device app for a smartphone or tablet. Before making this decision, the developer considered the pros and cons of both, with the resultant outputs presented in Tables 2.1 and 2.2.

Table 2.1: Advantages and Disadvantages of Using a Mobile Application

Mobile App Advantages	Mobile App Disadvantages
Functionality – a mobile app can access and utilise the functions of external smartphone features such as access to calendar, use of camera, and trigger notifications.	**Cost** – a mobile app is created specifically for the operating system. For example, if you develop an app for IOS it will not work on Android and vice versa. This leads to separate apps having to be developed for the different operating systems and leading to more costs.
User experience – mobile apps are designed specifically for a certain device and takes into consideration features such as screen size and capabilities.	**Maintenance and upgrade** – apps required maintenance and upgrade on a regular basis. To keep users interested, the app will be required to be constantly upgraded otherwise users will look for an alternative.

Table 2.2: Advantages and Disadvantages of Using a Web Application

Web App Advantages	Web App Disadvantages
Compatibility – this allows the target audience to use the app regardless of the device in which that they are using. This allows the web application to provide a broader audience	**Platform limitations –** web applications are unable to make use of capabilities of the device and cannot be engaged on the web app, such as use of the camera, notifications service etc.
Cost-effective - by providing a responsive web application that can fulfil its requirements it can prove to be more cost effective than a mobile app as this allows it to be used on several platforms.	**User experience** – with mobile apps being designed for a specific device, it is much easier to provide a better user experience.

2.3 Proposed Solution

Considering the issues discussed in the previous Section, the most effective proposed solution identified was to create a web application. The main reason it was selected was the ability for a web application to obtain a broader audience reach and being accessible across multiple platforms.

With the target audience being so broad e.g. homeowners, tenants, handymen, and business owners, it was felt the web application has the potential to reach a wider audience giving it the best chance of succeeding.

It is essential to create a quick and responsive design that can perform on numerous platforms. This is a vital component for a web app to be successful as users would quickly lose interest and search elsewhere for an alternative.

2.4 Development Strategy

This will discuss the development strategy the developer is looking to implement. The development strategy will concentrate on the following identified core areas:

- System Front End;
- System Database;

2.4.1 System Front End

This will comprise the rendered user interface. The proposal is to develop a simple, intuitive web app which will be pleasing to look at, easy to use and will be able to operate across many platforms and device types (mobile, laptop, tablet, etc.).

2.4.2 System Database

With the content of the web application changing everyday (i.e. jobs, users, etc) it has been identified at an early stage that a robust database is essential to administer the application. Such a database would be used to store vital information such as user's login details, address, jobs, etc. A traditional relational database is going to be useful to store the information needed and collect the relevant data from the database.

The developer has decided to use a mySQL database as it is believed it will be a suitable platform for the development of the web application. When combined with PHP it can deliver highly exclusive solutions. Some of the main reasons why it was have chosen to use mySQL is:

- Performance – mySQL is known for its exceptional performance and scalability and it is why so many web-based businesses continue to use mySQL. Before creating the application, performance is a vital component and providing optimum speed will only help enhance this application;
- Reliability – with application being made available 24/7, downtime can be detrimental to customer loyalty, and potentially prove very costly;
- Security – with frequent money transfers security is a component that requires a great deal of consideration. With mySQL being renowned for its data security and support this will greatly benefit the application;
- Design – one of the main reasons mySQL is considered the number one choice for web applications is the design, as it is tailored and optimized specifically for web applications. It continues to evolve along with the web to remain the number one choice;
- Cross-platform availability – with the plan to distribute the application across numerous platforms mySQL contributes to this flexibility and choice.

2.5 Development Model

One of the first decisions the developer made was identifying the optimum software development methodology. It is very important for the team to use a methodology that will be cost effective as well as being highly appropriate to the nature of the project that is being developed in order to increase the chances of project success. With a tight deadline of twelve weeks it is pivotal that the correct development methodology is selected.

After carefully considering both an agile scrum development methodology and a waterfall development methodology, the agile scrum development approach was identified to be a better fit. This will ensure that the development of the application is delivered in the most organised and relevant manner.

One of the main issues we had with a waterfall approach was that the user would have been dissatisfied with the delivered software product, as the deliverables in this method are based upon documented requirements. This often means that the user will not see what has been delivered until its almost finished. By that time, changes can be difficult to implement. Another issue with the waterfall method is the ‘big bang’ approach where testing and evaluation is not carried out to the end, so it is hard to know how successful it will really be until the end and if there are major issues it might be too late to fixed. Also, the next step cannot be carried out until the preceding step is completed and if there are major issues with that preceding step significant delays could occur.

With agile scrum development being based on the idea of incremental and iterative development, this means that each phase of the development is revisited over and over. This will iteratively improve the quality of the project by regular feedback from the user. One of the main benefits behind agile development is the ability to divide the development cycles up into smaller sections. This allows each section to be passed through a cycle and be completed in stages. By breaking down the project into manageable sections, this will allow the team to focus on high-quality development, testing and overall collaboration. Also, by producing frequent segments and performing testing and reviews during each iteration, quality should be then improved by the finding and fixing of defects quickly and at an early stage.

By applying this development model, the developer is confident that this will allow the project to be delivered in the most efficient manner possible. This approach usually requires a team to work together to carry out the process. However, even though this project will be carried out individually the principles of the agile approach will be used to make sure that the project is completed successfully, and all the deadlines are met.

The benefit of having work reviewed and evaluated by potential users following the completion of defined development sprints will be beneficial to the overall outcome of the project. The overarching approach to this project will ensure development activities are user-centric, focusing development of features around the perceived needs of potential users. This allows the developer the opportunity to gain valuable feedback early in the project and the opportunity to make changes if needed. This will be hugely beneficial to the overall development by having the opportunity to constantly refine and reprioritise the overall product backlog.

It is anticipated that development activities will take around 6 weeks, with the initial phase of developing core functions taking around 4 weeks. Sprints will be introduced after the initial phase. At the end of each sprint the developer should have a potentially shippable product. However, from the feedback and evaluation process at the end of each sprint is will be decided whether further work is required, or product is ready.

3.0 Requirements Analysis and Specification

In this chapter we will be setting out requirements which will essential for the success of the project and the process involved.

3.1 Requirements Elicitation Process

It is widely known that many software development projects fail due to incomplete, unclear, or incorrect requirements, therefore the requirements elicitation process is essential to the success of the project. This process helps to identify, elicit, analyse and document core project requirements which will underpin and help drive the app development process.

According to BABOK (2018) the ten elicitation techniques which are commonly used are:

- Brainstorming;
- Document Analysis;
- Focus Groups;
- Interface Analysis;
- Interviews;
- Observation;
- Prototyping;
- Feedback Comments Analysis;
- Requirement Workshops, and;
- Survey/Questionnaire.

Obviously certain requirements elicitation techniques could not be applied to this project due to certain circumstances, i.e. the project team comprising one-person, and with no clients to interview or complete surveys. Consequently, the main techniques which were applied are Brainstorming, Feedback Comments Analysis, and Observation.

This requirements elicitation process basically began in the project planning stage when the proposal for the web application was developed, i.e. the production of a simple online booking application for basic jobs around the home. This initial idea helped identify the main aim and objectives for this project (Section 1.2 and 1.3).

At the commencement of the project a brainstorming session was carried out to develop the aim and objectives of the project. After going through the aim and objectives of the project the developer was able to conduct another brainstorming session to outline a range of ‘User Stories’ and outline non-functional requirements. These were developed by taking into consideration potential users of the web application and functions a typical user would want to see on the application. These requirements were again reconsidered throughout the agile development process at the evaluation stage. The user-stories and non-functional requirements will be discussed further in the next sections.

3.2 User Stories

User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically follow a simple template:

> As a < type of user >, I want < some goal > so that < some reason >. (Mountain Goat, 2018).

User stories were mainly used in this project to define the main requirements of the system, derived using the viewpoint of a potential user of the application to consider what is required and why. The above template was used as a guide when writing the user stories.

Acceptance criteria is included in each user story to specify conditions under which a user story is fulfilled. The inclusion of acceptance criteria is a vital component not only for the eliciting the vision of the project but for the development process as well. Clearly written criteria help provide a solution to the functionality that are intended to be implemented.

The user stories developed considered the limited time available to deliver the project, so requirements were prioritised to account for such. This helped to ensure core requirements were addressed first so the application could function, then subsequent requirements that were then added would add to the overall user experience.

The MoSCoW system was applied to prioritise requirements, categorised by the following (Dai Clegg, 1994):

- Must (M): Defines a requirement that has to be satisfied for the final solution to be acceptable;
- Should (S): This is a high-priority requirement that should be included if possible, within the delivery time frame. Workarounds may be available for such requirements and they are not usually considered as time-critical or must-haves;
- Could (C): This is a desirable or nice-to-have requirement (time and resources permitting) but the solution will still be accepted if the functionality is not included e.g. The HR system “could” send out notifications on pending leave dates;
- Won’t (W): This represents a requirement that stakeholders want to have but have agreed will not be implemented in the current version of the system. That is, they have decided it will be postponed till the next round of developments

A summary of the user stories is presented in Table 3.1 (overleaf) and categorised in terms of priority in relation to the MoSCoW method. User stories categorised as ‘Won’t Have’ have been excluded.

Table 3.1: Summary of User Stories Developed

User Story	Priority
#01-Registering	Must
#02-Logging in	Must
#03-Posting a Job	Must
#04-Apply to become a contractor	Must
#05-When a user becomes a contractor	Must
#06-Browsing jobs as a contractor	Must
#07Contractor making an offer on a job	Must
#08-Contractor viewing jobs that they have made an offer on	Must
#09-Contractor changing job offer	Must
#10-Leaving a review	Must
#11-Administrator makes decision on contractor	Must
#12-Change Personal Information	Must
#13-Change password	Must
#14-Edit job and change details	Must
#15-Removing a job that has not been assigned	Must
#16-View offers from contractors	Must
#17-Assigning job	Must
#18-Viewing completed jobs	Must
#19-View contractor’s profile page	Must
#20-View all posted jobs in a table	Should
#21-Log in and view user’s own profile	Should
#22-Contractor asking a question about a job	Should
#23-Contractor leaving a review for user	Should
#24-Contractor viewing user’s profile	Should
#25-Report a contractor	Should
#26-Administrator can issue warning/remove contractor	Should
#27-User adding a payment method	Could
#28-Contractor being able to change their details	Could
#29-Administrator can view all users	Could
#30-Administrator can view all contractors	Could
#31-Adding pictures for jobs	Could

The user stories are provided in full in Appendix A. An example of a user story can be seen below:

#07	Browsing jobs as a contractor
As a registered contractor I want to be able to browse jobs, so I can make offers for jobs	
Acceptance Criteria When the registered contactor is logged in they can go to the 'browse jobs' section and should be able to view jobs and make an offer on jobs that meet their criteria i.e. county that they cover, and particular job category matches that of their preferences. A registered contractor should not be able to make an offer or access a job that does not meet their criteria	

3.3 Non-Functional Requirements

Non-functional requirements specify criteria that can be used to judge the operation of a system, rather than specific behaviours. These non-functional requirements were identified during the elicitation process. The non-functional requirements for this project are outlined in Table 3.2 below:

Table 3.2 Non-functional Requirements

Category	Ref	Requirement
Security	SE 01	To access the application valid login credentials are required
Target Devices	TD 01	The system should operate/render consistently on all mobile and desktop devices
Reliability	RE 01	The system should be user friendly and support the a number of users at any one time
Usability	US01	The interface, font and colour scheme used in the application shall be consistent throughout
Performance	PE 01	Response times – There shall be no apparent delay on opening any of the application's pages
	PE 02	Processing times – There shall be no apparent delay when carrying out a function (e.g. posting a job or making an offer)

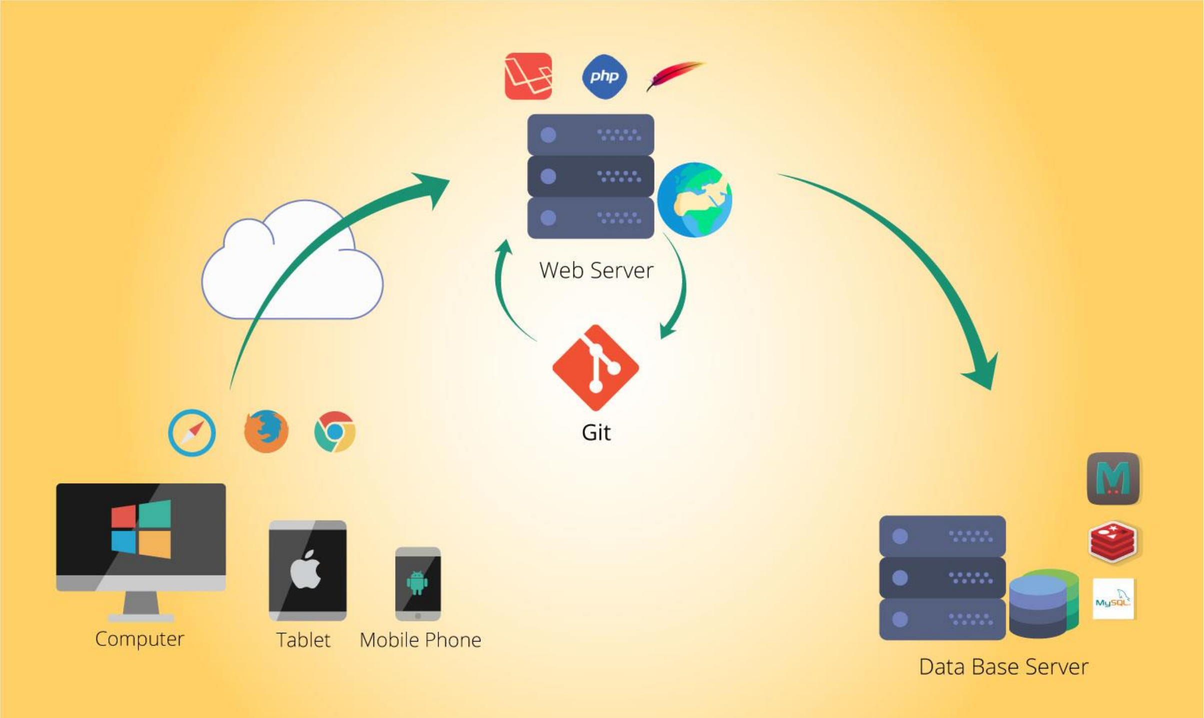
4.0 Design

This section will investigate the planned design of the web application. Designs of how the final application might look, along with how the application will change when being rendered on different devices and decisions on the design for the user interface. The database design will also be a focus of discussion and consideration in this chapter.

4.1 System Architecture

The design for the web application is based on the application proposal in Section 2.1 and the core functions identified from the requirements elicitation process. Considering the core system functionality required to be provided, which includes the application needing the ability to store information (i.e. account details, job details, etc.) an example of the architecture which the system will implement is presented in Figure 4.1. This shows how users will interact with the application via HTTP requests, where responses will be rendered in a browser. The web server helps manage the flow of the application and liaises with the database server to process requests from users and their responses. The database server is where the data is stored (persistent storage) and retrieved.

Figure 4.1: Diagram of System Architecture



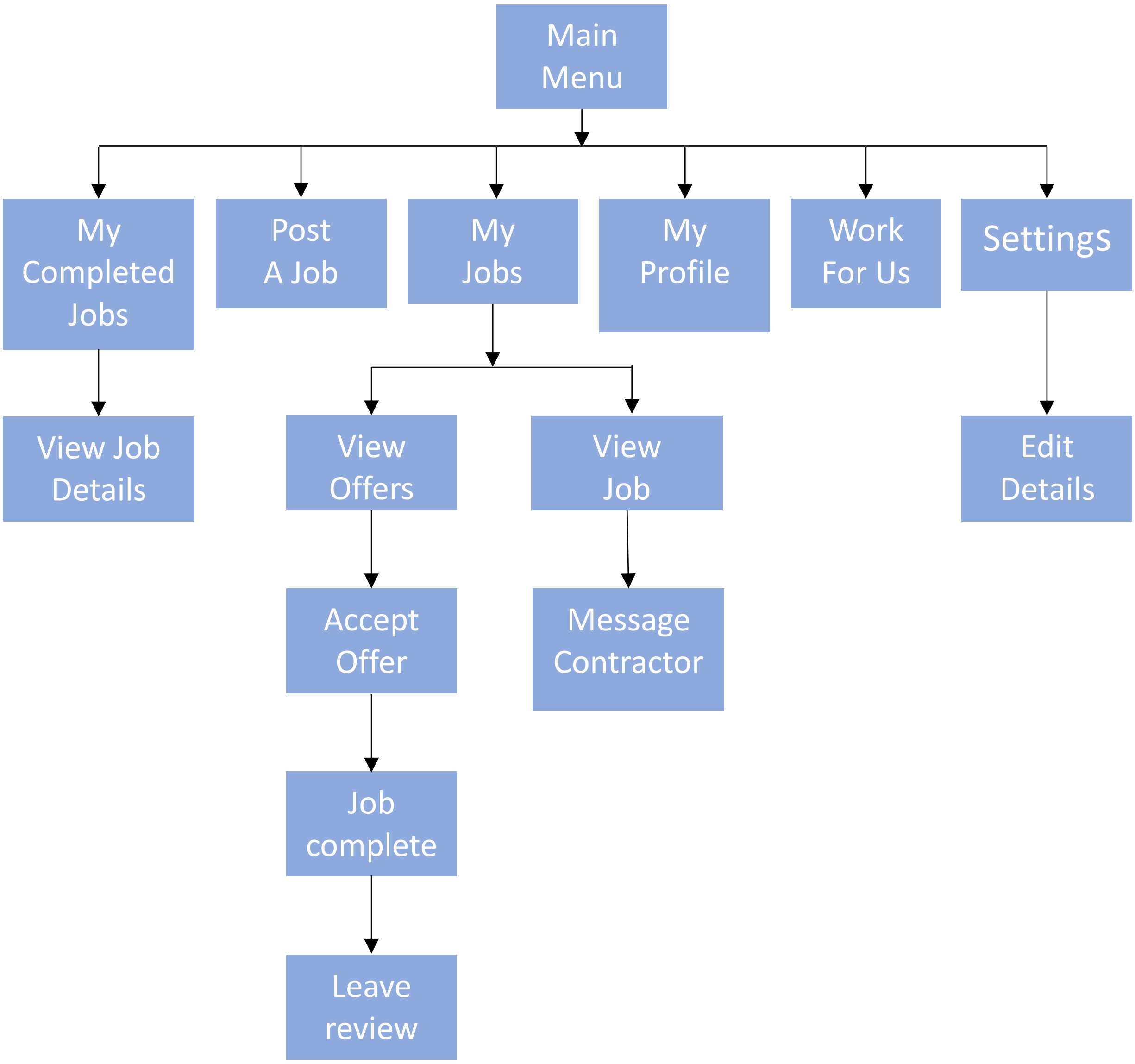
4.2 System Overview

During the requirement stage it was realised that the application will require different access levels, specifically for a general user, contractor and administrator. To help with the application development, a process flow diagram for each access level was prepared that identifies the structure, activities and functions that are likely to be required. Which will be presented in the following sub-sections:

4.2.1 General User

This section will discuss the structure, activities and functions that are likely to be required of a general user.

Figure 4.2 Flow diagram for the activities and functions for a general user



The above diagram (Figure 4.2) identifies the activities and functions that allow the core requirements for a general user be met along with an enhanced user experience. Specifically, the following:

- Ability to post a job
- Ability to view my jobs
- View ongoing jobs and completed jobs
- View offers for jobs and the ability to accept/reject these offers
- View my profile
- View settings and the ability to edit these settings i.e. password, profile picture etc.

4.2.2 Contractor

This section will discuss the structure, activities and functions that are likely to be required of a registered contractor.

Figure 4.3 Flow diagram for the activities and functions for a contractor

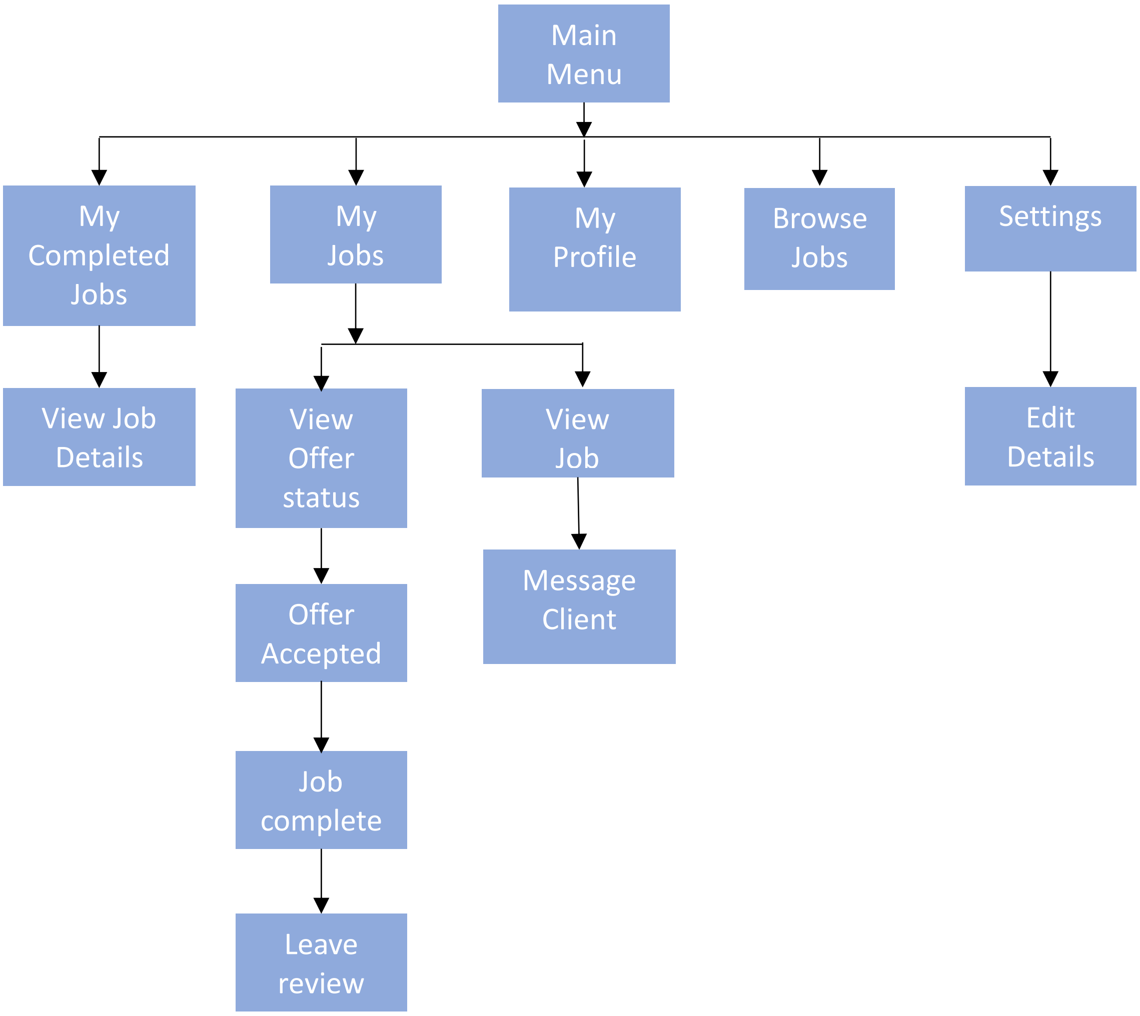


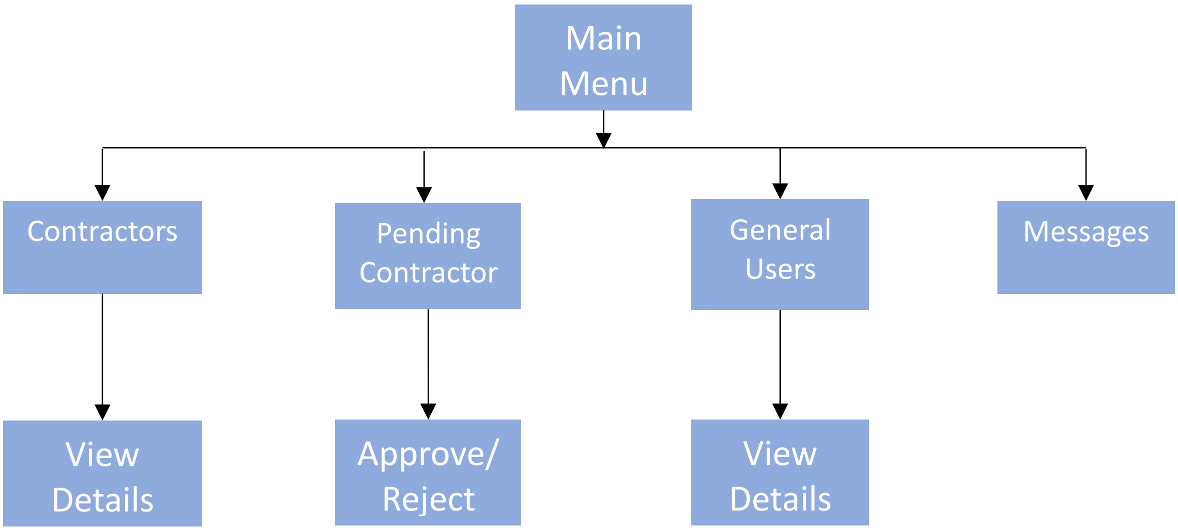
Figure 4.3 shows how a contractor would be able to operate within the application. The functions in which they should be to control in the application are:

- The ability to browse potential jobs
- The ability to place an offer on a job that matched their criteria i.e. area that they cover and area of expertise.
- The ability to view their own profile
- The ability to view their personal details and edit them
- View ongoing jobs and message the client
- View completed jobs
- Leave a review for a client once the job has been complete

4.2.3 Administrator

This section will discuss the structure, activities and functions that are likely to be required for the administrator of the web application.

Figure 4.4 Flow diagram for the activities and functions for an administrator



The above diagram (Figure 4.4) considers how an administrator for the application will be able to carry out such functions:

- View general users of the application
- View contractors of the application
- Be able to approve or reject pending contractors by ensuring they meet or don't meet the criteria i.e. police check, id verification etc.
- View messages from users i.e. reports, etc.

These diagrams will provide a framework on how the application development activities will take place and support the interface design process.

4.3 Development Platform

As previously mentioned in Section 2.3 the developer has chosen to development a web application rather than a mobile application. When coming to this decision the developer considered the timescale for the delivery of my project and the users which are being target. Using a web application should help reach a broader audience.

Along with the web application the developer will be incorporating PHP, MySQL, and JavaScript into the system, these will be discussed further in the following sections:

4.3.1 PHP

The use of PHP within the application allows for easy manipulation of a MySQL database which is being used to store the information from the application. Along with integration with JavaScript which gives the ability to create real-time updates more easily provides the application with a foundation to build on. Other reasons why PHP (Kalam 2017) will be used within the application include the following:

- **Efficiency** – PHP can enhance the performance of an application as it is accessible when writing code as well as reliable too when dealing with a number of different web pages;

- **Cross Platform** – using PHP allows the application to operate across any number of operating systems. PHP works exceptionally across all and can also be easily interfaced with MySQL;

- **Control** – PHP prevents the need for long scripts and has the capability of preforming actions in a few lines of code and allows for maximum control over the application. PHP allows for easy editing which essential for my web application;

- **Security –** it is commonly known as one of the most secure ways of developing a web application as it has a security layer that protects against viruses and threats;

- **Tested –** PHP has been used for many years by millions of people and its capabilities have been tested and proven.

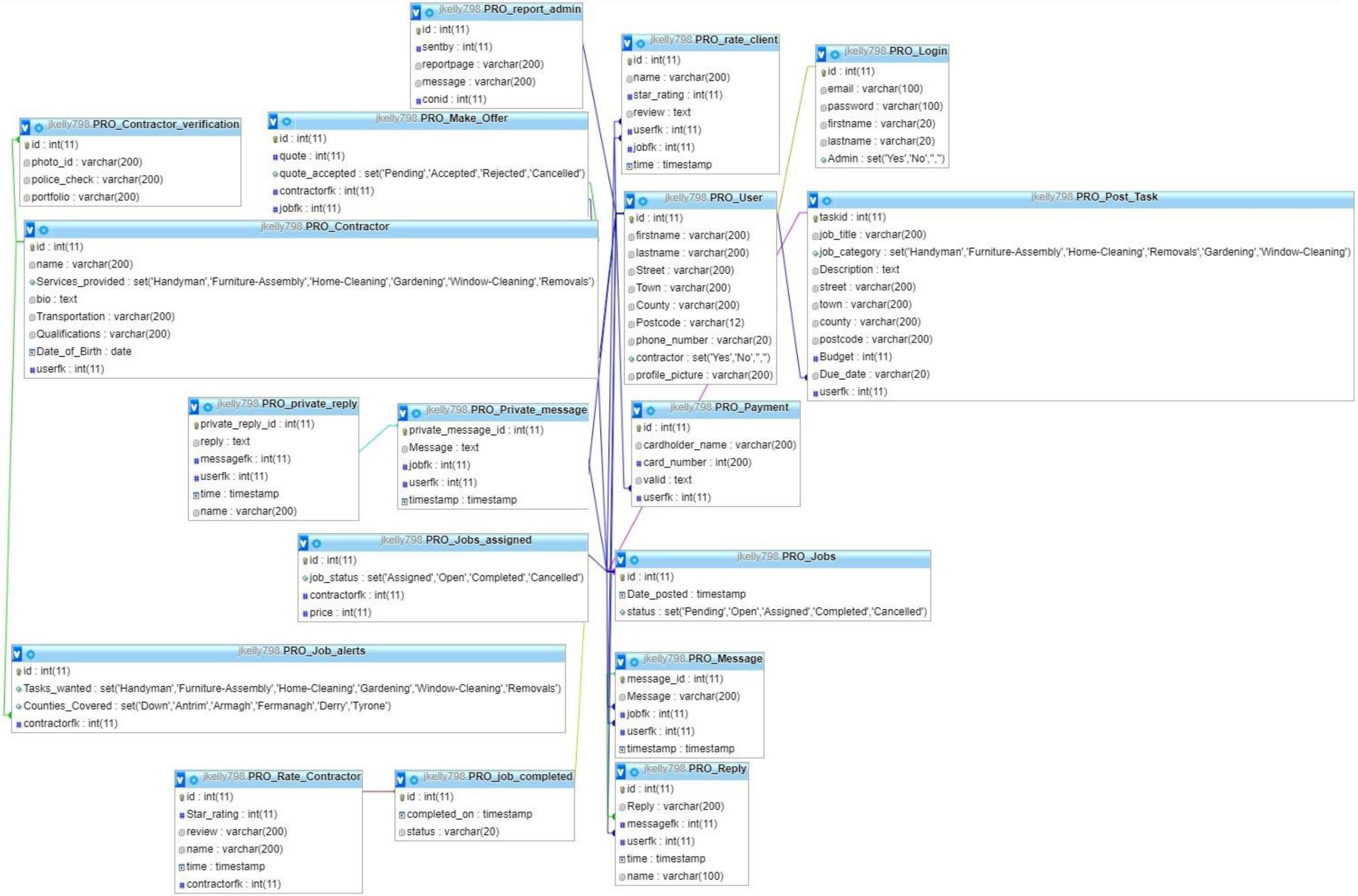
4.3.2 MySQL Database

As mentioned in Section 2.5, a MySQL database will be implemented to help dynamically serve and populate the web application. When considering the requirements for the system in context of the relevant data which is required to be saved to benefit the functionality of the application, it is anticipated the system will be required to store the following information:

- User’s email and password;
- User’s personal details i.e. address, phone number, profile picture, and payment method;
- Job details, i.e. title, description, budget, address, etc.;
- Contractor’s details, services provided, profile, etc.;
- Contractor’s offer for a job that they would like to bid on;
- Rating details for both a contractor and a client.

With this information being stored it allows the application to communicate via PHP with the database using the structured query language and display the details it is querying on the application itself. The proposed database structure is conceptualised in an Entity Relationship (ER) diagram, which is presented in Figure 4.5:

Figure 4.5 Proposed ER Diagram for the HouseProud Web Application



jkelly798 PRO_report_admin
- id : int(11)
- sentby : int(11)
- reportpage : varchar(200)
- message : varchar(200)
- conid : int(11)

jkelly798 PRO_rate_client
- id : int(11)
- name : varchar(200)
- star_rating : int(11)
- review : text
- userfk : int(11)
- jobfk : int(11)
- time : timestamp

jkelly798 PRO_Login
- id : int(11)
- email : varchar(100)
- password : varchar(100)
- firstname : varchar(20)
- lastname : varchar(20)
- Admin : set('Yes','No','','')

jkelly798 PRO_Contractor_verification
- id : int(11)
- photo_id : varchar(200)
- police_check : varchar(200)
- portfolio : varchar(200)

jkelly798 PRO_Make_Offer
- id : int(11)
- quote : int(11)
- quote_accepted : set('Pending','Accepted','Rejected','Cancelled')
- contractorfk : int(11)
- jobfk : int(11)

jkelly798 PRO_User
- id : int(11)
- firstname : varchar(200)
- lastname : varchar(200)
- Street : varchar(200)
- Town : varchar(200)
- County : varchar(200)
- Postcode : varchar(12)
- phone_number : varchar(20)
- contractor : set('Yes','No','','')
- profile_picture : varchar(200)

jkelly798 PRO_Post_Task
- taskid : int(11)
- job_title : varchar(200)
- job_category : set('Handyman','Furniture-Assembly','Home-Cleaning','Removals','Gardening','Window-Cleaning')
- Description : text
- street : varchar(200)
- town : varchar(200)
- county : varchar(200)
- postcode : varchar(200)
- Budget : int(11)
- Due_date : varchar(20)
- userfk : int(11)

jkelly798 PRO_Contractor
- id : int(11)
- name : varchar(200)
- Services_provided : set('Handyman','Furniture-Assembly','Home-Cleaning','Gardening','Window-Cleaning','Removals')
- bio : text
- Transportation : varchar(200)
- Qualifications : varchar(200)
- Date_of_Birth : date
- userfk : int(11)

jkelly798 PRO_private_reply
- private_reply_id : int(11)
- reply : text
- messagefk : int(11)
- userfk : int(11)
- time : timestamp
- name : varchar(200)

jkelly798 PRO_Private_message
- private_message_id : int(11)
- Message : text
- jobfk : int(11)
- userfk : int(11)
- timestamp : timestamp

jkelly798 PRO_Payment
- id : int(11)
- cardholder_name : varchar(200)
- card_number : int(200)
- valid : text
- userfk : int(11)

jkelly798 PRO_Jobs_assigned
- id : int(11)
- job_status : set('Assigned','Open','Completed','Cancelled')
- contractorfk : int(11)
- price : int(11)

jkelly798 PRO_Jobs
- id : int(11)
- Date_posted : timestamp
- status : set('Pending','Open','Assigned','Completed','Cancelled')

jkelly798 PRO_Job_alerts
- id : int(11)
- Tasks_wanted : set('Handyman','Furniture-Assembly','Home-Cleaning','Gardening','Window-Cleaning','Removals')
- Counties_Covered : set('Down','Antrim','Armagh','Fermanagh','Derry','Tyrone')
- contractorfk : int(11)

jkelly798 PRO_Message
- message_id : int(11)
- Message : varchar(200)
- jobfk : int(11)
- userfk : int(11)
- timestamp : timestamp

jkelly798 PRO_Rate_Contractor
- id : int(11)
- Star_rating : int(11)
- review : varchar(200)
- name : varchar(200)
- time : timestamp
- contractorfk : int(11)

jkelly798 PRO_job_completed
- id : int(11)
- completed_on : timestamp
- status : varchar(20)

jkelly798 PRO_Reply
- id : int(11)
- Reply : varchar(200)
- messagefk : int(11)
- userfk : int(11)
- time : timestamp
- name : varchar(100)

Normalisation

Database normalisation is process used to organise a database into tables and columns. The idea is that a table should be about a specific topic and that only those columns which support that topic are included.

At this stage it is important to try and normalise the database as much as possible as this process helps efficiently organise the data in the database. Normalisation helps eliminate redundant data and dividing larger tables into smaller tables, defining relationships between the tables so data can be isolated making it easier to insert, update and delete.

Throughout the development of the database design the process of normalisation was carried out by applying the defined principles of such, specifically the three forms of normalisation comprise and is defined as the following (Rollins, 2009):

- First Normal Form (1NF): No repeating elements or groups of elements;
- Second Normal Form (2NF): No partial dependencies on a concatenated key;
- Third Normal Form (3NF): No dependencies on non-key attributes.

These methods are particularly important at the design stage as it makes sure that the database does not have any repeating elements, all data is stored in one place and all data items are stored together, allowing data to be stored in separate tables. However, through the inclusion of foreign keys in related tables, data can be retrieved together through queries i.e. PRO_Job, PRO_job_assigned from Figure 4.5, etc. which means within the database dependencies are now logical, as it now can uniquely identify the row of another table which it relates to. This restricts invalid data from being inserted into the database.

4.3.3 JavaScript

Using JavaScript allows development of dynamic content within the application. Within the application itself number of events should trigger the execution of JavaScript, which include:

- **Input Validation –** JavaScript will be used to validate input from users. For example, an email address entered by a user. JavaScript will dynamically query the database to check if the email address is available. Also, JavaScript will be used to check that passwords match and meet the criteria.
- **User Interaction** – when a user adds something to the application i.e. posts a job, accepted an offer, or has left a message, JavaScript will be used to display a proper message to the user and let them know an action has taken place. This benefits the application by making it usable and productive for the users.

The developer believes by creating a web application with this platform he is giving himself the opportunity to create an application to best of his ability with the constrained timescale for the project.

4.4 User Interface

User interface is something that can be quite often neglected in the design of a web application quite often resulting in unsatisfied users and failed projects. So, to ensure that the user interface is not ignored and that a clear and easy use user interface is implemented, I decided to take into consideration certain principles when designing the user interface for the application:

- **User-Centred Design:** As the target user (i.e. homeowners, tenants, handymen, business owners) is so broad it is important to design an interface that will be simple and easy use so even the less computer-literate will be able to navigate through the application without any problems;

- **Clarity:** Everything in the application from text to functions needs to be clear and simple. Even colours and graphics should be used in a simple manner and in a way that makes sense and is not confusing to the user;

- **Consistency :** The application should have the same or similar functions and looks throughout. This is the same for the colour scheme as it should be consistent throughout. Users should also have an idea what to expect when using the application i.e. sign up, sign in, post a job or view jobs, etc;

- **Keep it Simple:** According to Sarah Perez (2016) nearly one in four people abandon an application after only one use. The interface which I plan to implement is to try and make the users feel comfortable by keeping operations simple. The interface should almost feel invisible to the user. E.g. use of icons where possible in place of text to identify functions;

- **Reducing Burden on the User:** By anticipating what a user wants to do and reduce the burden on them i.e. form design having the opportunity to have some fields pre-chosen or filled out e.g. if a user wants a window cleaner the web application should give the user the opportunity to click on a picture of a window cleaner and this will lead them to a form which has the job category pre-filled;

- **Responsive:** The user interface must have the ability to handle different screen sizes i.e. laptop, smartphone, tablet, etc. and the ability to be responsive on each of these devices. With the application targeting a wide audience it is essential for its success to be functional on these different devices.

4.4.1 Colour

Taking the above principles into consideration, it was determined that the colours and fonts in the user interface will be kept consistent throughout the application. In terms of the colour scheme it was believed that to provide a design that is relevantly simple but professional, three colours will be used, specifically white as a background, and incorporating the colours black and blue. The colour scheme suggested is anticipated to stand out and provide a layout which should simple to use but have a functional and professional feel.

4.4.2 Font

One of the main elements to consider when deciding on a font type and size is that it is there to be read and not for decoration, so it is important that all font can be clearly read by the user. With the application intended to be functional across many platforms and a range of devices (including mobile devices) it is vital that the font scales down when on a smaller screen and is responsive throughout. It was decided to use Helvetica font type consistently throughout as it is clear, simple and easy to understand. Font sizes will vary throughout when needed i.e. headings will be displayed larger than normal text for instance.

4.4.3 Logo

Even though the production of a logo was not defined as a core requirement, providing a logo for the system will help to add to the professional feel of the user interface. A logo can also give the application and, by extension, the overall service an identity, so it can be easily recognised by a user and enhance their experience.

Any logo and icons associated with the HouseProud system (and brand) should try and inform the user of the key functionality of the system. It was determined that a design approach to keep any graphics clean as simple as possible would be adopted, to convey the brand name (HouseProud) and get the message across that the application is for home owners who are looking work done around the house. The finalised logo will be discussed further in Chapter 5.

4.4.4 Navigation and Layout

To ensure that the application is easy to use even for the less computer literate any key action points, i.e. submissions and text input areas, will be clearly labelled in a clear and concise manner so there is no confusion.

The flow path of the application between various pages is a crucial aspect of the design process that influences the non-functional requirement such as usability, performance, and reliability. With this to consider wireframe diagrams of the application were produced which will provide the basis for the development of the layout. The following diagrams show activities from a general user perspective (Figure 4.6) and a registered contractor’s perspective (Figure 4.7).

Figure 4.6 Wireframe diagram showing activities for a general user

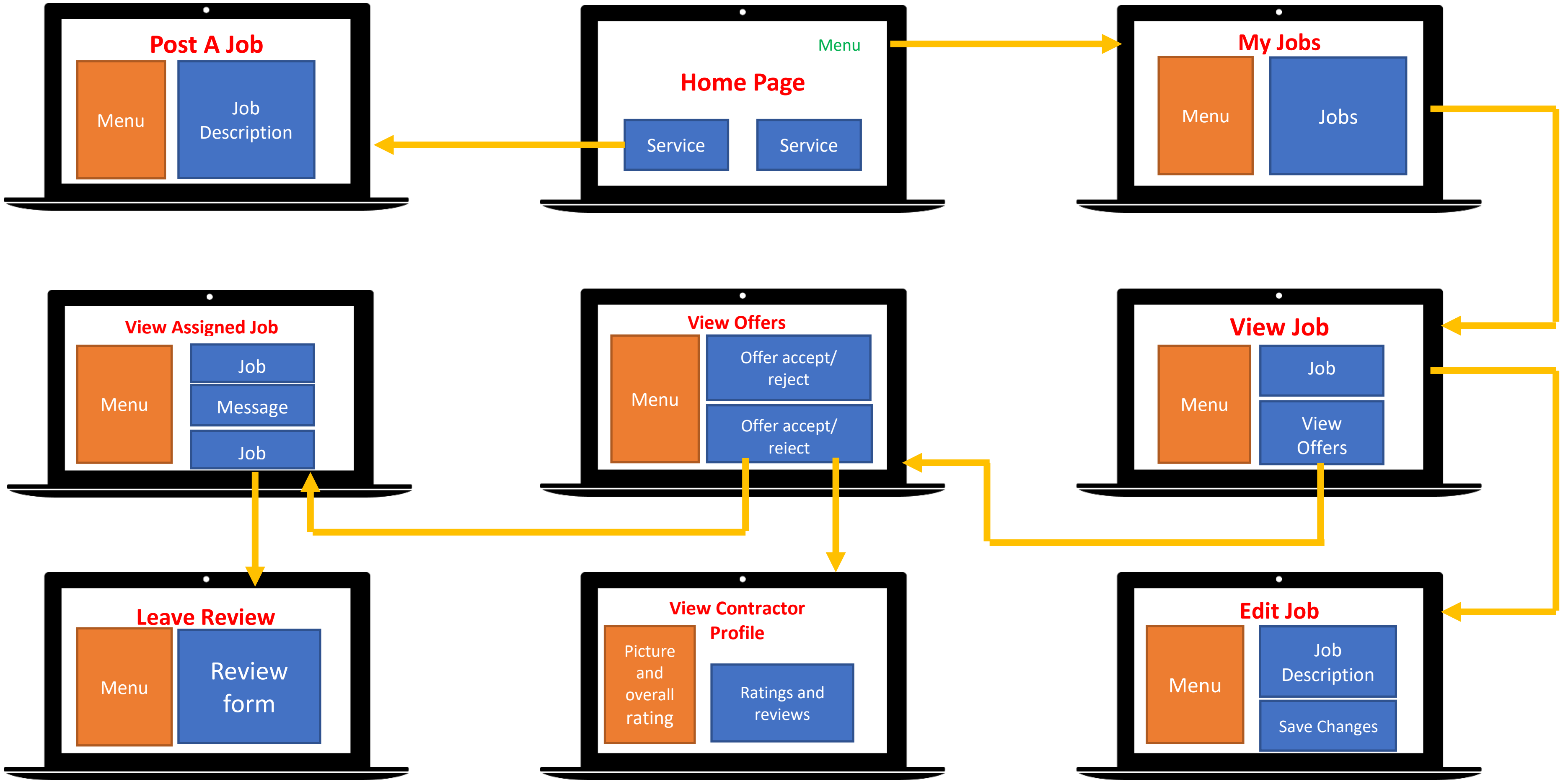
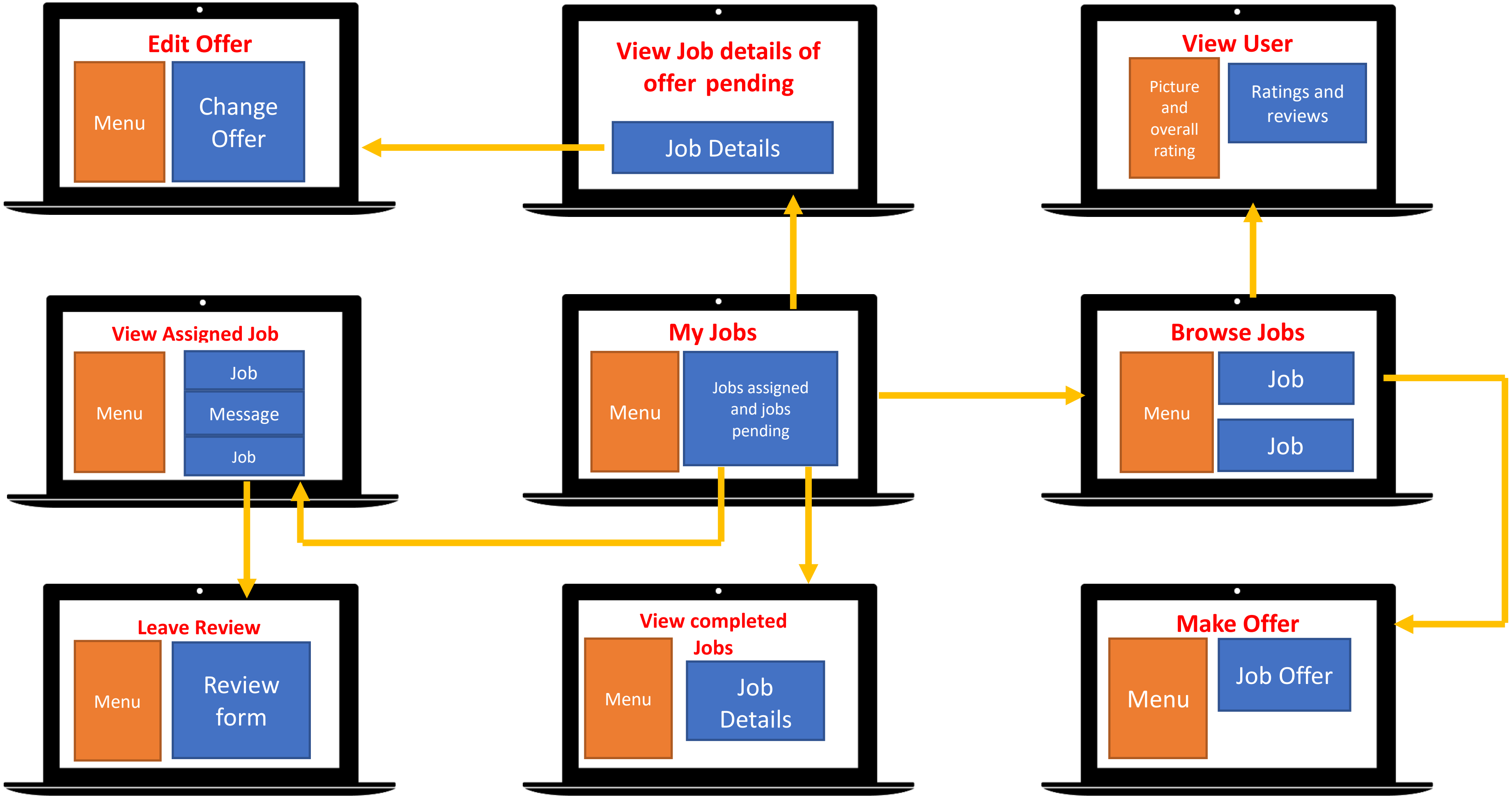


Figure 4.7 Wireframe diagram showing activities for a registered contractor



4.4.5 Portability

One of the main non-functional requirements of the system is having the ability to render/operate consistently on all mobile and desktop devices (Table 3.2 Requirement TD01). Ensuring that when the application is rendered on the phone it is operationality, clean, easy to use and free from clutter. Figures 4.8 and 4.9 demonstrate the plan of action to meet these requirements on the 'Home' page and 'My Jobs' page. These designs provide a basis for the development and will be discussed in detail in Chapter 5.

. Figure 4.8 The proposal for the design of the home page on a laptop and a smartphone

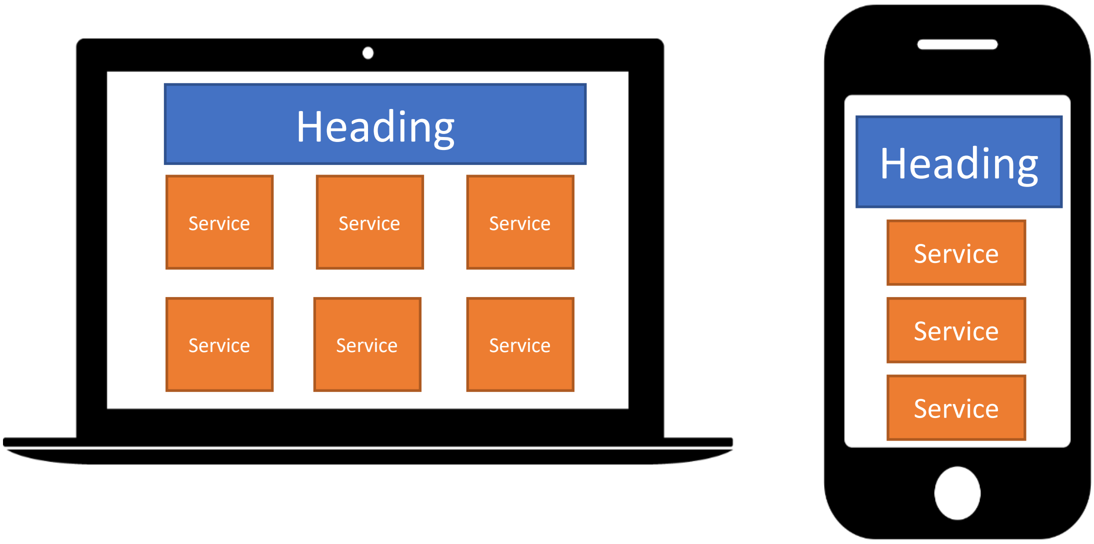
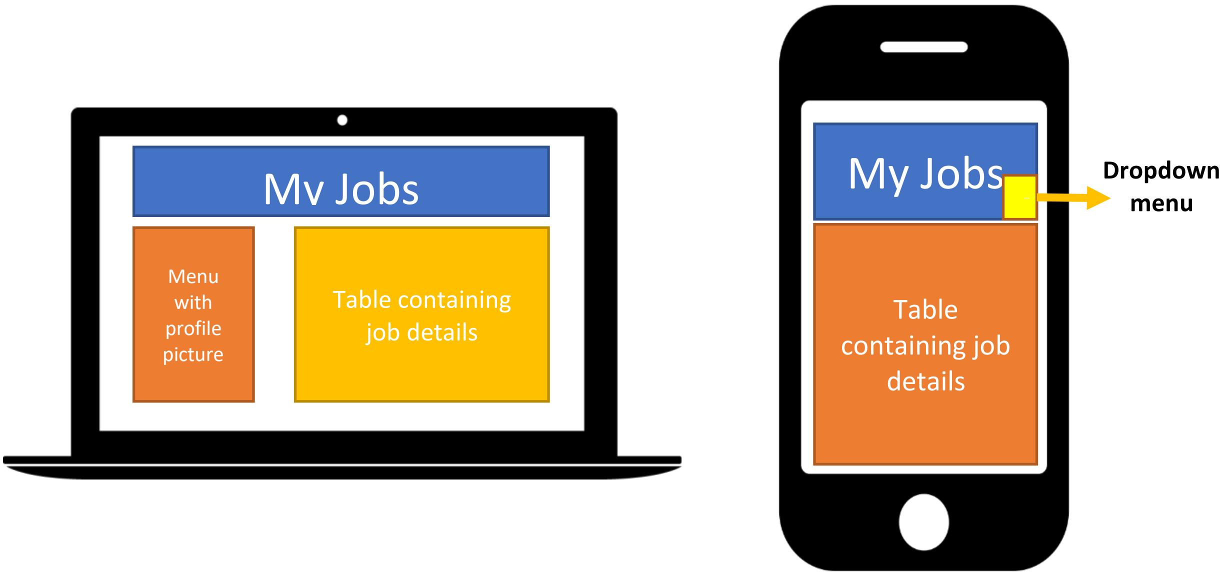


Figure 4.9 The proposal for the design of how the 'My Jobs' page will be rendered on a laptop and smartphone device


Mv Jobs
Menu with profile picture
Table containing job details
My Jobs
Dropdown menu
Table containing job details

5.0 Implementation

This section outlines the methodology used in the development of the application. We will look at the implemented interface and discuss in detail some of the main functions of the application. The testing strategy undertaken throughout the development is also discussed in this section.

5.1 User Interface

The decision was taken before the development stage to use and identify a CSS Framework for the application. However, the principles mentioned in Section 4.5 were taken into consideration when identifying the most appropriate interface template to use. A template was subsequently resourced from themeforest.com. My decision for sourcing a CSS Framework was for the following reasons:

- **Save Time:** With the project having tight time constraints it is believed that by using a template will save time on design and display.
- **Usability:** By using this template allows the project to have an application which pays more attention to user experience and allows development of a polished and as professional as possible giving the time constraints.
- **Customisation:** The flexible design allowed for easy customisation and made possible for the application to be developed in a way that is suited best to meet the requirements.
- **Responsiveness:** The template was fully responsive for many elements and reduced the development needed to provide an application that was fully responsive across many platforms.

Even though a CSS Framework was used I still wanted to incorporate the design decisions that were discussed in Section 4.5, i.e. colour, font, consistency, and reducing burden on user. Screenshots from the layout of the home page and another section are shown overleaf in Figures 5.1 and 5.2.

Figure 5.1 Home page of HouseProud web application

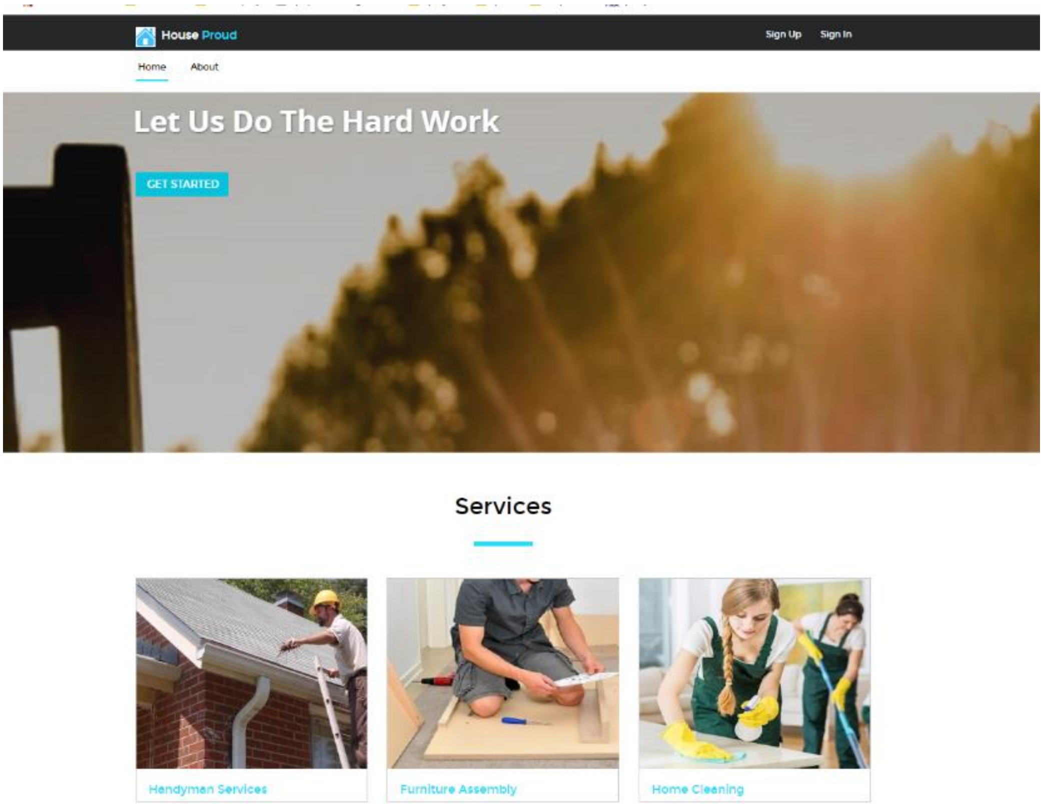
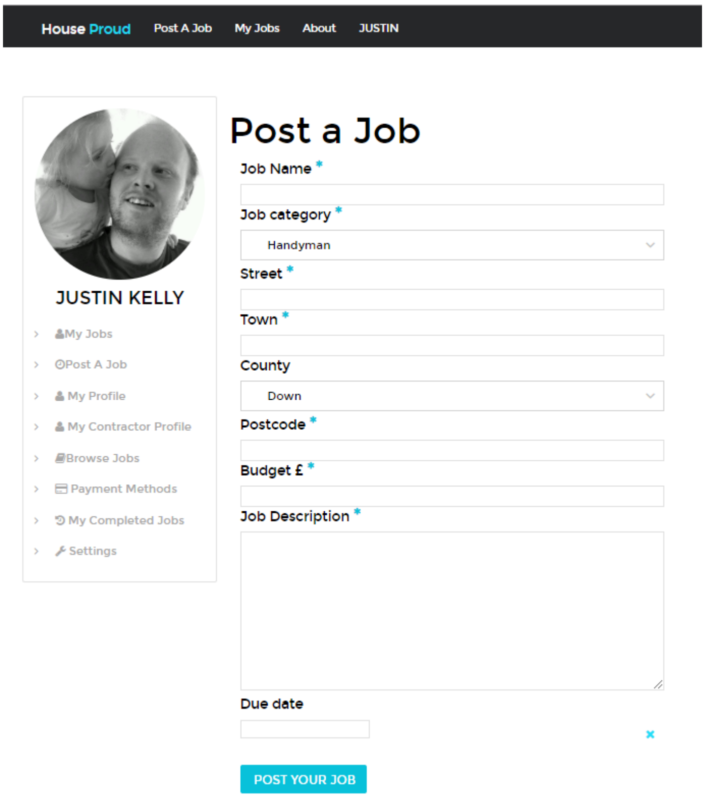


Figure 5.2 The post a job page for the House Proud web application



Post a Job

Job Name *

Job category *
Handyman

Street *

Town *

County
Down

Postcode *

Budget £ *

Job Description *

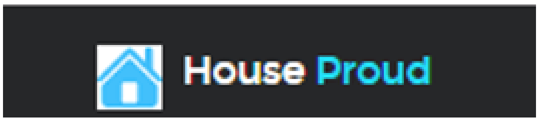
Due date

From Figure 5.1 you can see how it has been developed in a way to try and keep it as simple as possible by introducing icons instead of text to identify functions. For example, if a user that was signed in clicked on the handyman service picture they would be directed to the 'Post a Job' page (Figure 5.2) with the job category pre-filled with the service required.

5.1.1 Logo

Figure 5.3 shows the logo implemented for the application. The implemented logo is consistent with the design principles outlined in Section 4, specifically the colour scheme and the logo itself along with the brand name, that should give the application an identity which is easily recognised.

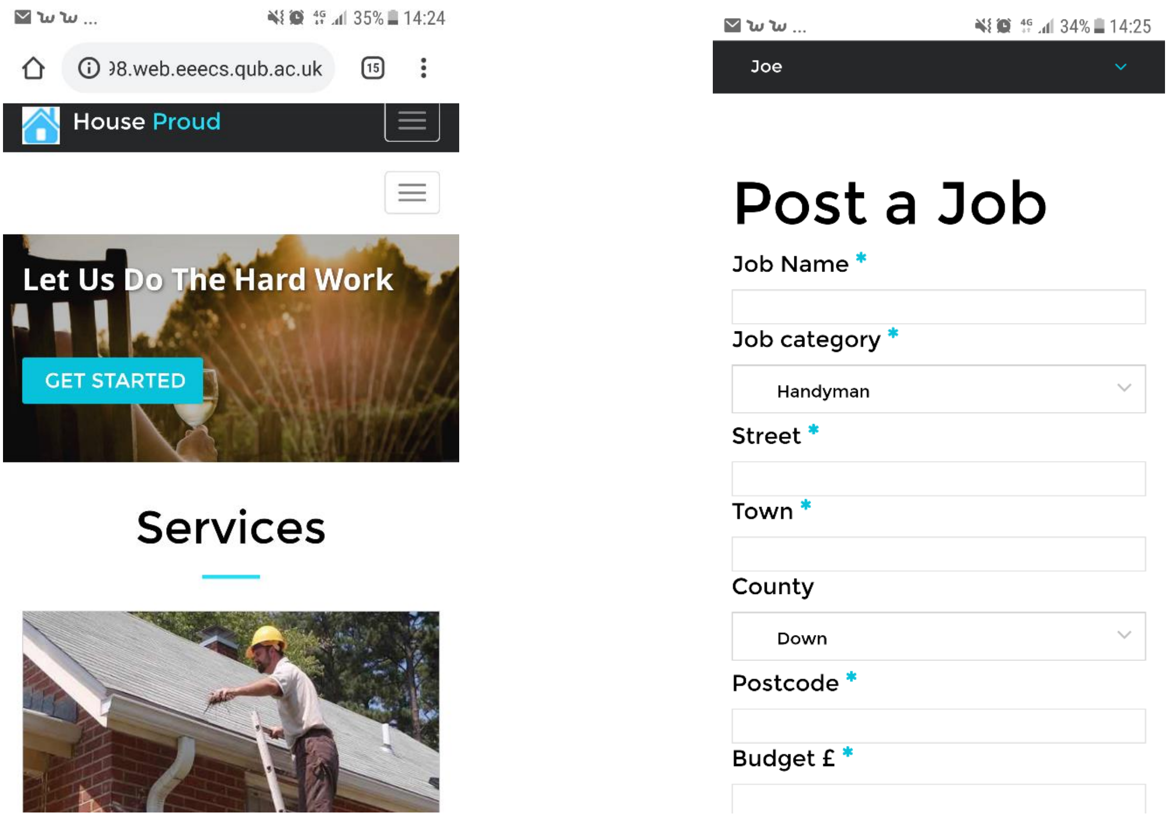
Figure 5.3 Logo for House Proud



5.1.2 User Interface on Smaller Screen

One of the main features of the application is the ability to render/operate consistently on all mobile and desktop devices as per requirement TD 01 (Table 3.2). The design of the CSS framework chosen fortunately allows a lot of the design aspects to be adjusted to suit the bespoke needs of the HouseProud web app, as demonstrated in Figure 5.4 (below).

Figure 5.4 House Proud home page and post a job page rendered on a smartphone



Post a Job

Job Name *

Job category *

Handyman

Street *

Town *

County

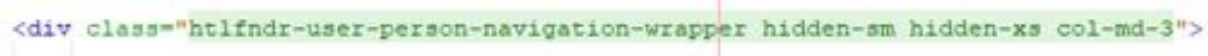
Down

Postcode *

Budget £ *

However, several adjustments were made throughout to ensure each page ran smoothly on a smaller device. This included amendments to the side menu (that can be seen in figure 5.2) which, when rendered on the smaller screen, made it difficult to navigate through the application. This was corrected by adding the code presented in figure 5.5 to the div of the side menu. By adding 'hidden-sm and 'hidden-xs' to the div (element containing code to display side menu) means that when the web application is rendered on a smaller screen menu does not display, reducing the clutter and making it easy to navigate through using a smaller device i.e. smartphone, tablet.

Figure 5.5 code added to hide side menu when on a smaller screen



```
<div class="htlfndr-user-person-navigation-wrapper hidden-sm hidden-xs col-md-3">
```

5.2 System Development

With the framework in place and ready to use the development could begin by looking closely at the requirements and design aspects. Development started by looking closely at the 'Must have' requirements (as categorised under the MoSCoW system) and the functions essential to allow the application to be usable.

This section discusses in detail of some of the most significant aspects of the system implementation, with a focus on the following key areas:

- User Sign up and password;
- Browse jobs and make offer;
- Message and reply.

5.2.1 User Sign-up and Password Check

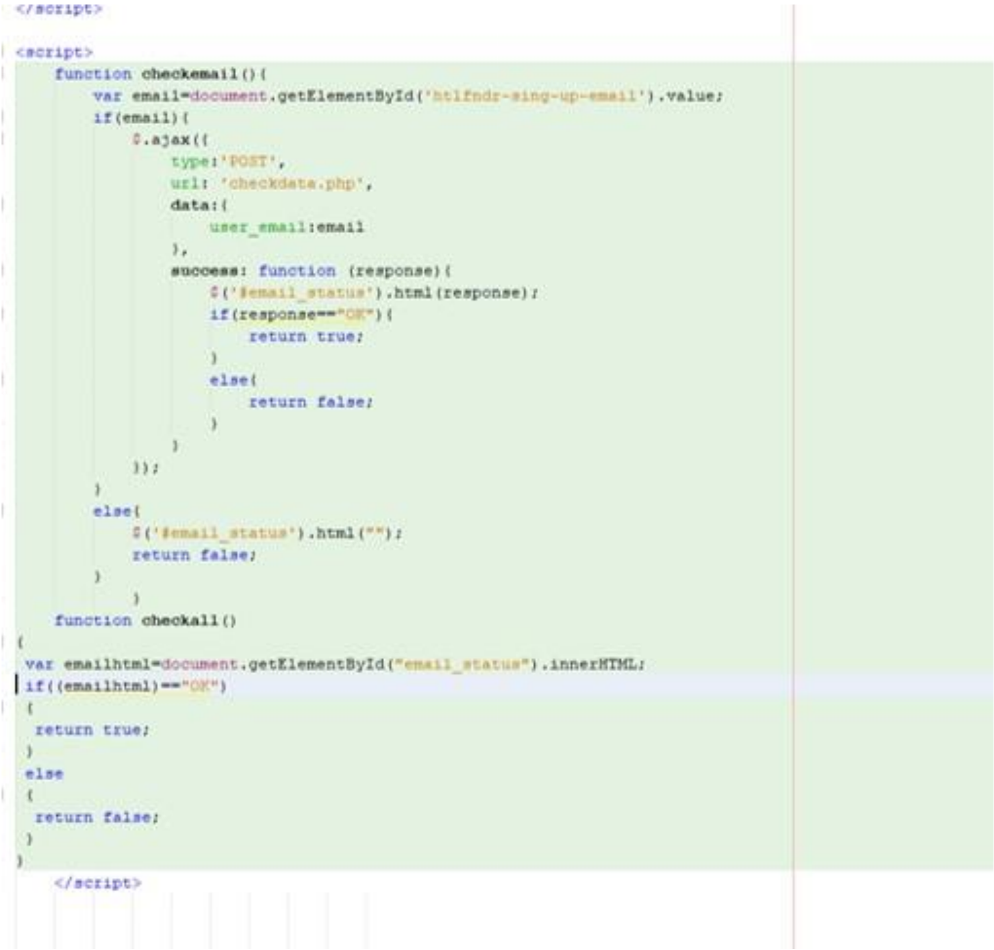
This is what the developer believed to be a core element to the success of the application. Previously in a different project the developer was able to development a sign-up registering form which checked the database if the username already existed. However, if the email address was already taken it redirected the user to a different page, displaying a message stating that the email address has been taken. This meant the user would have to go back and fill out the form again. The same process occurred if the password didn't meet the correct format. However, it was decided that this method would not be suitable for the needs of the web application.

User Sign-up

Before development for this took place, it was decided that what was needed was something which would be intuitive to user and wouldn't allow the user to lose interest straight away. The idea was to develop a function incorporating jQuery, PHP and MySQL so that once a user entered their email address the function would display a message without the page refreshing to indicate whether the username was available.

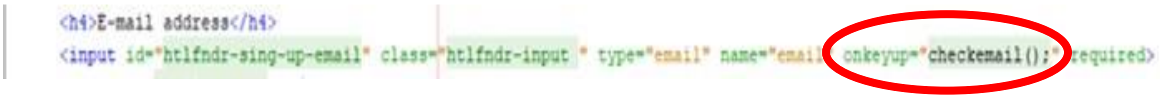
A term which is commonly used within software development is 'Do not reinvent the wheel'. Which is a principle that was applied to the development process for this project considering the constrained time scales. After some research an Ajax Login Form created by Mudit Jain (2018) was identified which was able to be customised and incorporated into the HouseProud system development. The following diagrams (Figures 5.6, 5.7, 5.8 & 5.9) show how the function was used in the system to check if the email exists already.

Figure 5.6 Function used to check if username exists



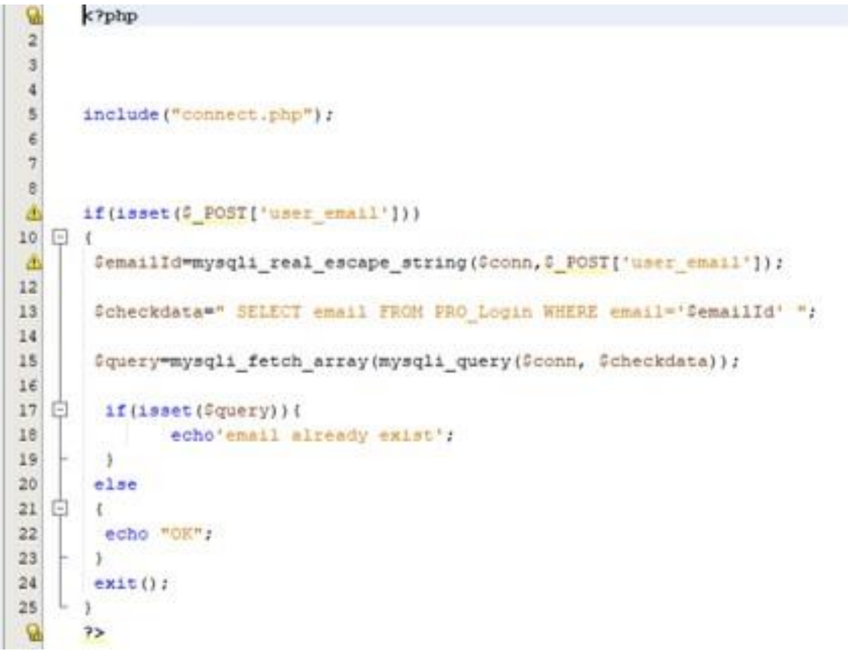
```
</script>

<script>
    function checkemail(){
        var email=document.getElementById('htlfndr-sing-up-email').value;
        if(email){
            $.ajax({
                type:'POST',
                url: 'checkdata.php',
                data:{
                    user_email:email
                },
                success: function (response){
                    $('#email_status').html(response);
                    if(response=="OK"){
                        return true;
                    }
                    else{
                        return false;
                    }
                }
            });
        }
        else{
            $('#email_status').html("");
            return false;
        }
            }
    function checkall()
{
 var emailhtml=document.getElementById("email_status").innerHTML;
 if((emailhtml)=="OK")
 {
  return true;
 }
 else
 {
  return false;
 }
}
    </script>
```

Figure 5.7 How function is called within the form



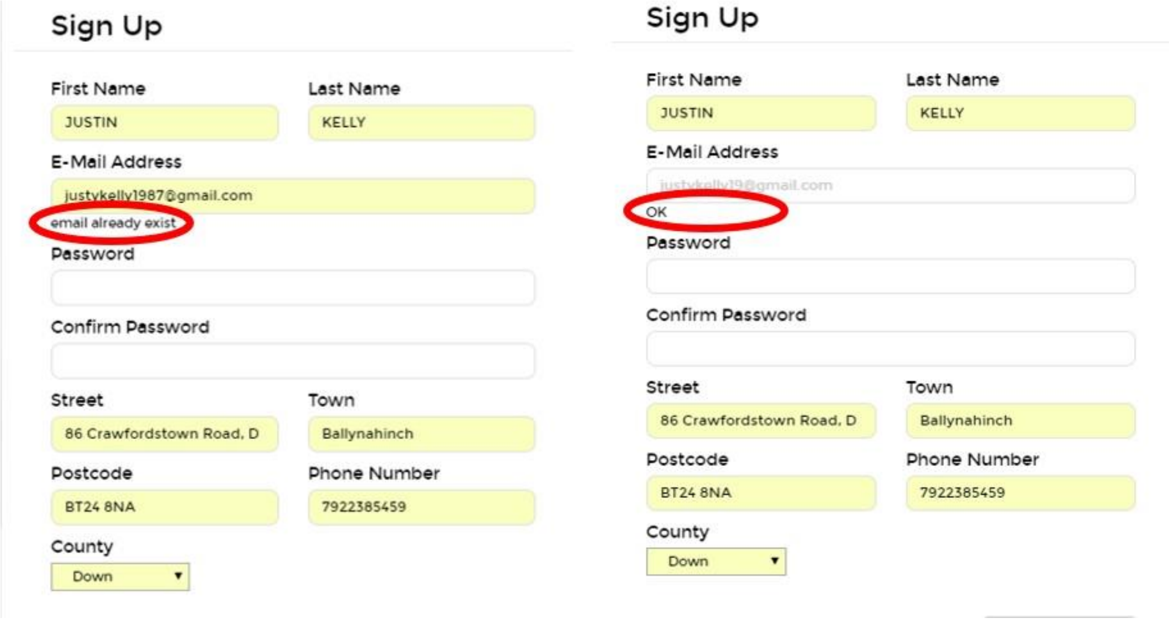
```
<h4>E-mail address</h4>
<input id="htlfndr-sing-up-email" class="htlfndr-input " type="email" name="email" onkeyup="checkemail();" required>
```

Figure 5.8 The PHP used to check the database



```
<?php



include("connect.php");



if(isset($_POST['user_email']))
{
 $emailId=mysqli_real_escape_string($conn,$_POST['user_email']);

 $checkdata=" SELECT email FROM PRO_Login WHERE email='$emailId' ";

 $query=mysqli_fetch_array(mysqli_query($conn, $checkdata));

  if(isset($query)){
        echo'email already exist';
  }
 else
 {
  echo "OK";
 }
 exit();
}
?>
```

Figure 5.9 How the function is displayed to the user



Sign Up

First Name: JUSTIN
Last Name: KELLY
E-Mail Address: justykelly1987@gmail.com
email already exist
Password:
Confirm Password:
Street: 86 Crawfordstown Road, D
Town: Ballynahinch
Postcode: BT24 8NA
Phone Number: 7922385459
County: Down ▾

Sign Up

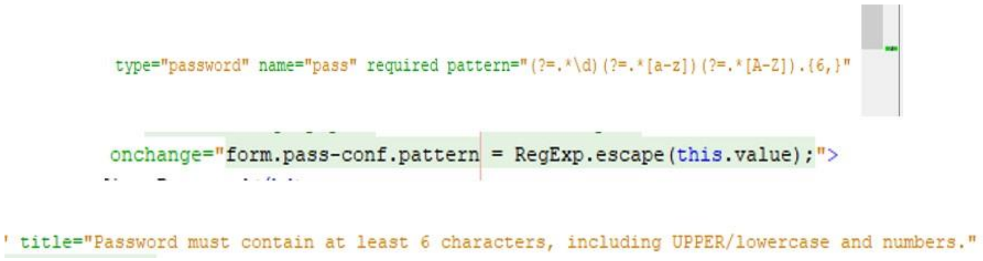
First Name: JUSTIN
Last Name: KELLY
E-Mail Address: justykelly19@gmail.com
OK
Password:
Confirm Password:
Street: 86 Crawfordstown Road, D
Town: Ballynahinch
Postcode: BT24 8NA
Phone Number: 7922385459
County: Down ▾

The above diagrams show the procedure which is taken each time a user tries to sign up to the web application showing how the email address is checked as soon as it is entered. This method provides clarity to the user and allows the user to change the entry without being re-directed and having to start the form again.

Password Check

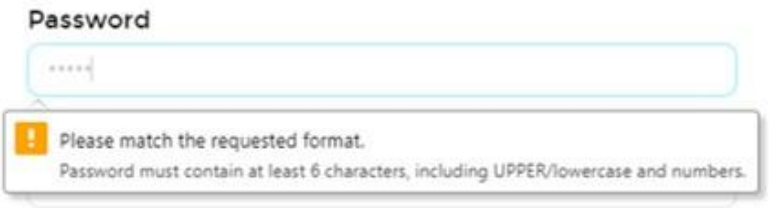
With the password there were certain elements which needed to be taken into consideration before development. As per the requirements the password had to contain at least six characters, at least one lowercase character and at least one uppercase character. This proved not that difficult to implement as by putting the code outlined in figure 5.10 into the form I was able to display the front end presented in figure 5.11 to the user. This again provided great usability to the user as a message was displayed when the wrong format was used for the password without being re-directed.

Figure 5.10 Details in form to setup password format



```
type="password" name="pass" required pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{6,}"

onchange="form.pass-conf.pattern = RegExp.escape(this.value);">

' title="Password must contain at least 6 characters, including UPPER/lowercase and numbers."
```

Figure 5.11 Message displaying to the user if password is of the wrong format

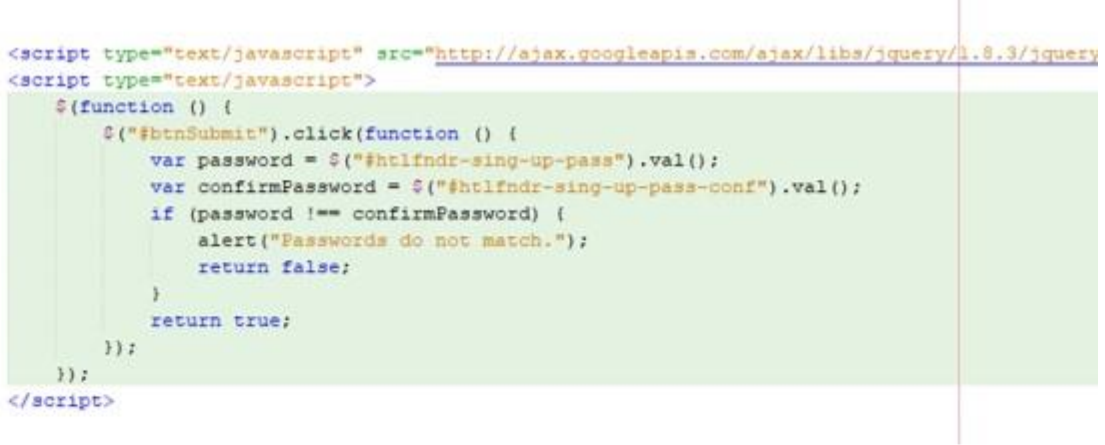


Please match the requested format.

Password must contain at least 6 characters, including UPPER/lowercase and numbers.

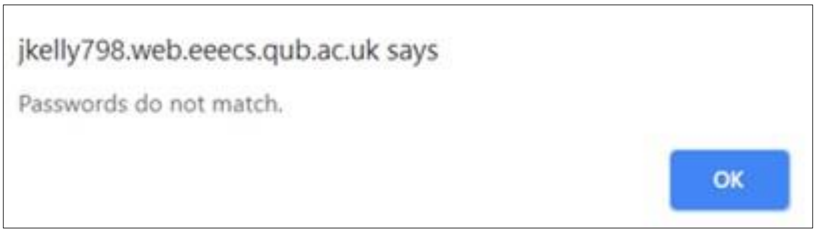
Another element that had to be considered, which is generally very common, was to ensure that when passwords are entered a check is carried out whether they match, with a message displayed to confirm. This was carried out by customising a google API so that it could be used within the HouseProud system. By adding the function outlined in figure 5.12 into the code, the system was able to check if both passwords matched and if they did not, the message presented in figure 5.13 was displayed to the user.

Figure 5.12 function to get passwords match



```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery
<script type="text/javascript">
    $(function () {
        $("#btnSubmit").click(function () {
            var password = $("#htlfndr-sing-up-pass").val();
            var confirmPassword = $("#htlfndr-sing-up-pass-conf").val();
            if (password !== confirmPassword) {
                alert("Passwords do not match.");
                return false;
            }
            return true;
        });
    });
</script>
```

Figure 5.13 Message displayed to user when passwords did not match



jkelly798.web.eeecs.qub.ac.uk says

Passwords do not match.

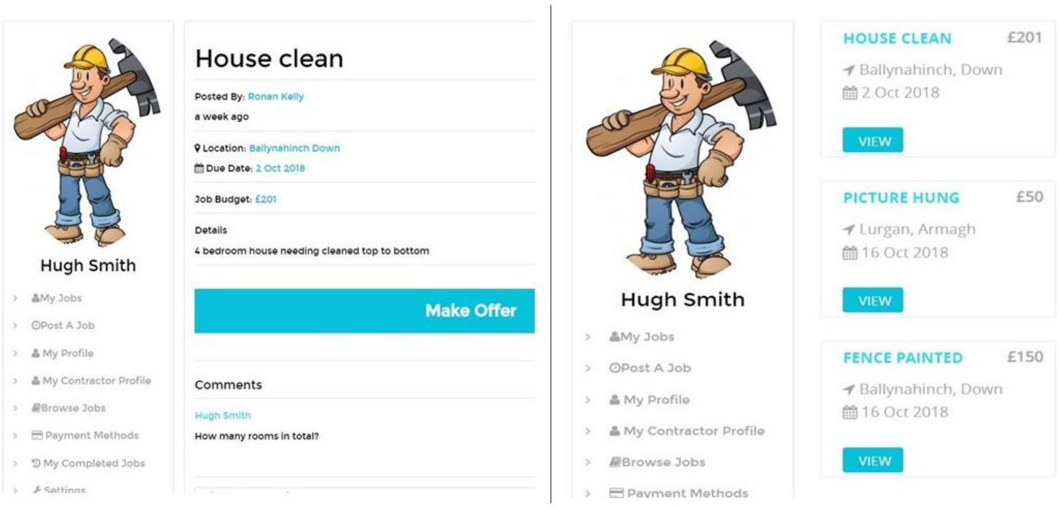
These sign up and password functions that were used within the system may seem basic practice to most applications. However, considering that this is the first function new users will use on the application it is of great importance that this works fluently and without any problems, as issues at this stage could leave negative impression on the application and potentially damage its success.

5.2.2 Browse Jobs and Make Offer

This function relates to user story '#07-Browsing jobs as a Contractor' and user story '#08-Contractor Making an Offer on a Job'. With the acceptance criteria in both these requirements stating that this can only be done by a contractor with whose criteria has met that of the job i.e. area covered matches the area of the job and the job category matches that of the contractor e.g. window cleaning job and contractor is a window cleaner.

The idea behind this 'criteria' was that when a contractor was registering with HouseProud, they would select from a list the services they can provide (defined as job categories they can cover). For example, whether they can provide handyman and gardening services, and the geographical area (Counties) they cover. When a user posts a request for a job to be carried out that contains a job category and County that matches the details of this contractor, the contractor would be able to view the job details in the 'Browse Jobs' section, as illustrated in Figure 5.14.

Figure 5.14 Example of the browse jobs section and view job page



House clean

Posted By: Ronan Kelly

a week ago

Location: Ballynahinch Down

Due Date: 2 Oct 2018

Job Budget: £201

Details

4 bedroom house needing cleaned top to bottom

Comments

Hugh Smith

How many rooms in total?

HOUSE CLEAN £201

Ballynahinch, Down

2 Oct 2018

PICTURE HUNG £50

Lurgan, Armagh

16 Oct 2018

FENCE PAINTED £150

Ballynahinch, Down

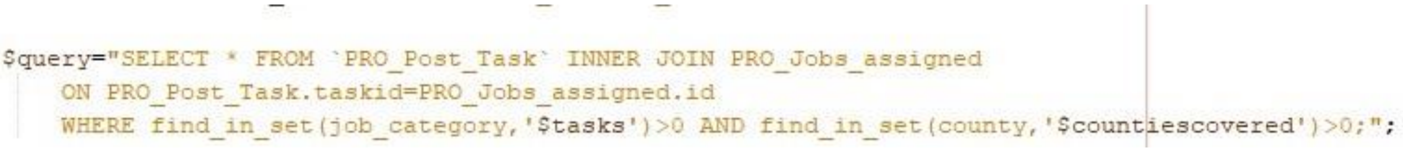
16 Oct 2018

When this area was developed it was thought everything was done successfully until testing was carried out (as discussed further in Chapter 6), which found that any contractor was able to access the view job via the URL. This ultimately meant that a contractor was able to make an offer on that job which did not meet their criteria.

This obviously was a concern as it was failing two of the 'Must Have' requirements (User Story #07 & #08 Appendix A). As the job category and Counties covered within the MySQL database where stored as a set, this meant more than one value could be stored, e.g. Counties covered by a single contractor could include Down and Armagh. The difficultly was matching these up to the county in which the job was located and restricting access to the contractors whose criteria was different.

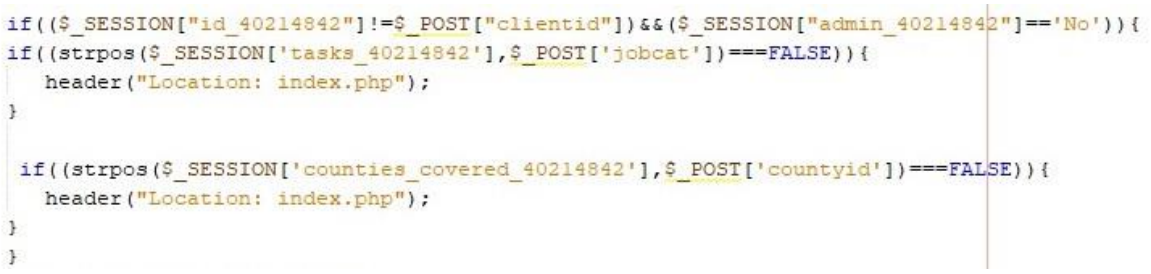
The issue was researched (using online resources), where a solution was finally identified to meet the specified requirements. The code outlined in figure 5.15 includes 'find in set' within the query, this finds the position of a string within a comma-separated list of strings e.g. if a job is in County Down and a Contractor area covered is Armagh, Down, Derry. This would return a result greater than 0 and display the job. An if statement with 'strpos' in it was also used -which is like the 'find in set' function, this function finds the first occurrence of a string inside another string (e.g. Contractor saved job categories are Handyman, Window-cleaner so if a job included either one of these categories it would display on the contractor's page). This successfully only showed the jobs which they could view and make offers on. Both methods required session variables to be set for the Counties covered and job category for the contractor which allowed the data to manipulated to carry out this request.

Figure 5.15 Query used in browse jobs section



```
$query="SELECT * FROM `PRO_Post_Task` INNER JOIN PRO_Jobs_assigned
    ON PRO_Post_Task.taskid=PRO_Jobs_assigned.id
    WHERE find_in_set(job_category,'$tasks')>0 AND find_in_set(county,'$countiescovered')>0;";
```

Figure 5.16 Method used to deny access to unauthorised users



```
if(($_SESSION["id_40214842"]!=$_POST["clientid"])&&($_SESSION["admin_40214842"]=='No')){
if((strpos($_SESSION['tasks_40214842'],$_POST['jobcat'])===FALSE)){
   header("Location: index.php");
}

 if((strpos($_SESSION['counties_covered_40214842'],$_POST['countyid'])===FALSE)){
   header("Location: index.php");
}
}
```

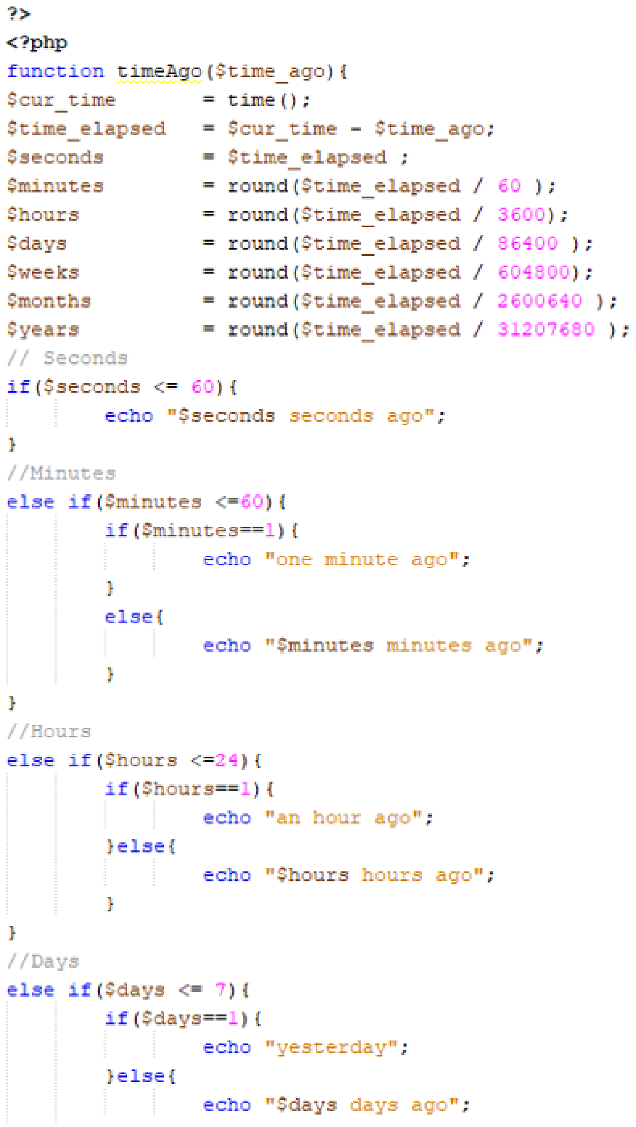
5.2.3 Message and Reply

Although not a 'must have' requirement but a 'Could Have' requirement (user story #23 – Contractor asking a question about a job) if developed successfully it was believed that this would be a nice feature to have where a contractor can ask a question and a user would be able to post a response. This would allow the contractor to obtain more information on a job before placing an offer.

Message

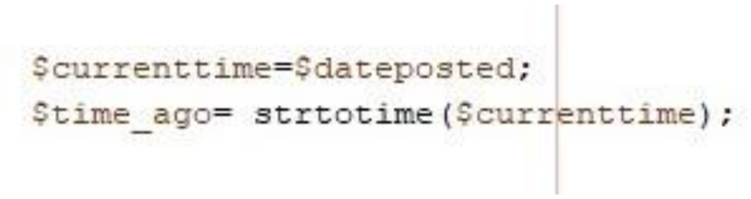
The idea was to have this section underneath the job details where a contractor could ask a job about that specific job (e.g. to determine how many windows require being cleaned) and the user could then reply with an indication when the message was sent (e.g. 10 minutes ago). This was achieved by adding the function ‘timeAgo’ (function used to determine how long ago something was submitted) to the PHP code, as illustrated in figure 5.17. When a message was submitted, it was timestamped automatically into the database, which was subsequently able to be set as a variable through a query. The function outlined in Figure 5.18 was then used to calculate, using the timestamp query and the current time, how long ago a message was sent and display how long ago a message was submitted, as illustrated in figure 5.19. By providing this feature it benefits the application with clearer usability and performance.

Figure 5.17 TimeAgo Function



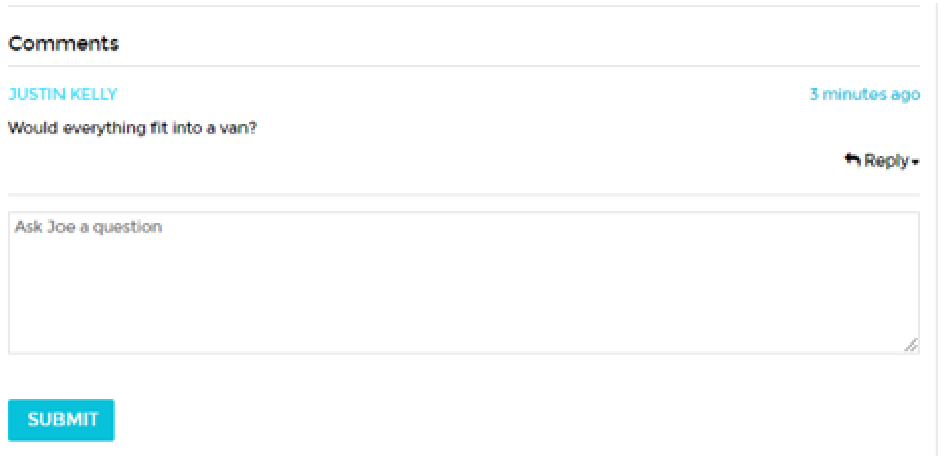
```
?>
<?php
function timeAgo($time_ago){
$cur_time       = time();
$time_elapsed   = $cur_time - $time_ago;
$seconds        = $time_elapsed ;
$minutes        = round($time_elapsed / 60 );
$hours          = round($time_elapsed / 3600);
$days           = round($time_elapsed / 86400 );
$weeks          = round($time_elapsed / 604800);
$months         = round($time_elapsed / 2600640 );
$years          = round($time_elapsed / 31207680 );
// Seconds
if($seconds <= 60){
        echo "$seconds seconds ago";
}
//Minutes
else if($minutes <=60){
        if($minutes==1){
                echo "one minute ago";
        }
        else{
                echo "$minutes minutes ago";
        }
}
//Hours
else if($hours <=24){
        if($hours==1){
                echo "an hour ago";
        }else{
                echo "$hours hours ago";
        }
}
//Days
else if($days <= 7){
        if($days==1){
                echo "yesterday";
        }else{
                echo "$days days ago";
```

Figure 5.18 How function timeAgo is used



```
$currenttime=$dateposted;
$time_ago= strtotime($currenttime);
```

Figure 5.19 How message is displayed



Comments

JUSTIN KELLY 3 minutes ago

Would everything fit into a van?

Reply

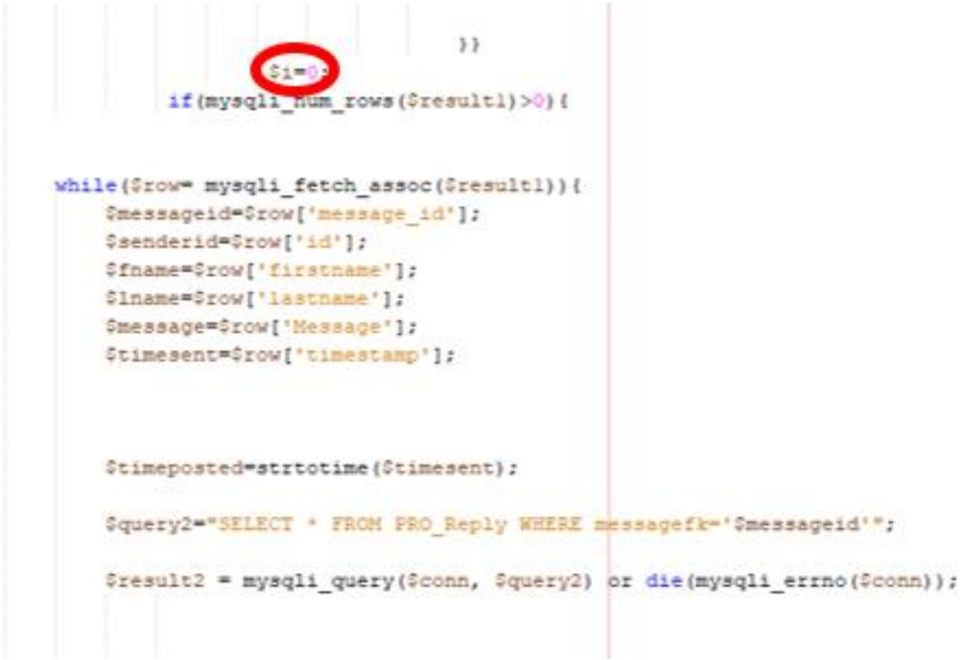
The reply function is very similar to the message function in terms of displaying the message and the time it was sent. The idea of function was that once a message is submitted a message is displayed to confirm the message has been sent and the page reloads so that the message submitted is displayed straight away.

This is developed within the system by calling a specific function once the form has been submitted. The function then has the capabilities to insert the message into the database, display a message to the user and reload the page so the message is then displayed.

However, when the function had to deal with more than one reply per message it did not execute the demand and problems occurred.

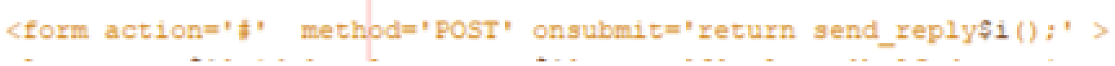
This was corrected by setting a variable to equal zero outside of the 'while loop', subjecting the form to the functions of the loop and getting it to call the function by putting in the form tags 'onsubmit= return send_reply(with the name of the variable here, seen in figure 5.21), this means when the reply was submitted it was able to complete the action as each reply function was now unique . The function would then follow in the loop with it being called the same name along with the variable (figure 5.22). At the end of the loop the variable would then increment, allowing each function to be unique and provide the capabilities within the application for there to be as many replies as possible within one message.

Figure 5.20 Variable set to zero outside of while loop



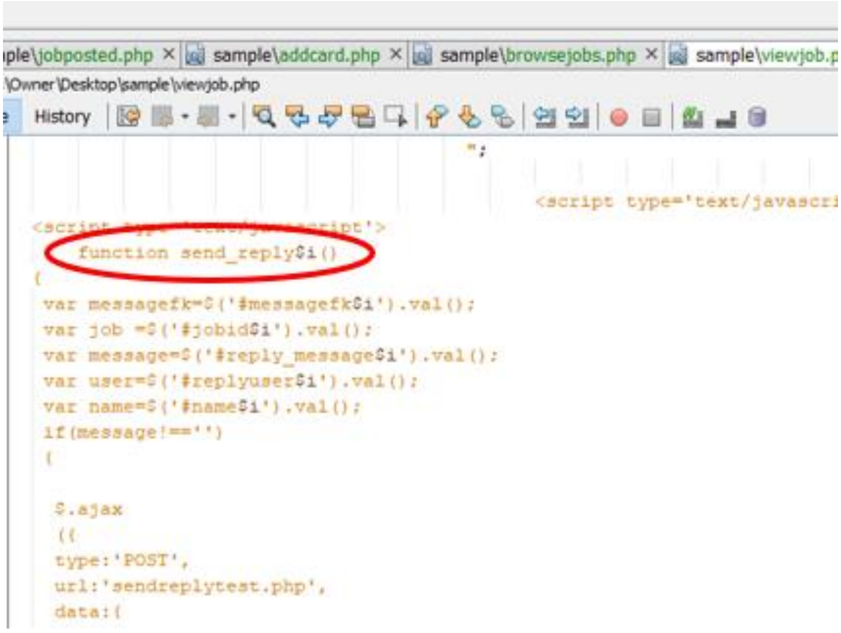
```
                                    }}
                $i=0;
            if(mysqli_num_rows($result1)>0){

    while($row= mysqli_fetch_assoc($result1)){
        $messageid=$row['message_id'];
        $senderid=$row['id'];
        $fname=$row['firstname'];
        $lname=$row['lastname'];
        $message=$row['Message'];
        $timesent=$row['timestamp'];



        $timeposted=strtotime($timesent);

        $query2="SELECT * FROM PRO_Reply WHERE messagefk='$messageid'";

        $result2 = mysqli_query($conn, $query2) or die(mysqli_errno($conn));
```

Figure 5.21 calling function within the form tags



```
<form action='#'  method='POST' onsubmit='return send_reply$i();' >
```

Figure 5.22 Send reply function in the while loop



```
                                          ";

                                                    <script type='text/javascri
<script type='text/javascript'>
  function send_reply$i()
{
 var messagefk=$('#messagefk$i').val();
 var job =$('#jobid$i').val();
 var message=$('#reply_message$i').val();
 var user=$('#replyuser$i').val();
 var name=$('#name$i').val();
 if(message!=='')
 {

  $.ajax
  ({
  type:'POST',
  url:'sendreplytest.php',
  data:{
```

5.2.4 System Development Overview

During the development stage it was discussed with the testing team after the initial development stage that a registered contractor should be able have the capabilities to ‘post a job’ to other contractors to reduce the need for contractors to have two different accounts. For example, a window cleaner may require a removal team. It was therefore agreed that a contractor of HouseProud would be able to do these functions within the contractor account. This would then mean once a contractor was registered to the application they had the capabilities to carry out functions which general users were able to i.e. ‘post a job’. This reduces the need for users to have more than one account.

5.3 Testing Strategy

Applying an effective testing strategy is key to a project’s success by not only ensuring that the project requirements have been achieved but also to ensure that the project is not delivered with unacceptable defects. With this project exhaustive testing was not possible due to time constraints and a decision had to made on which test levels to implement that provide a successful strategy with the time available. The following test levels were implemented:

5.3.1 Formal System Testing

Formal system testing essentially evaluates the system’s compliance with the specified requirements. Carried out at each development cycle, this involved designing test procedures using the acceptance criteria within the user stories and workflow required to execute the test. The test procedure is then used to define the test conditions and test cases.

With limited development time available for the project it was decided that core requirements (‘Must Have’) would only be tested at this stage. It was deemed prudent to concentrate the testing on these core functions as it was felt the application would not be functionable without these requirements being successfully met.

User Stories #01 - #19 (as summarised in Table 3.1) were deemed the most appropriate to consider at this level of testing as they contained functions that would require to be fully operational to cover the basic functionality of the application.

These user stories covered 19 of the 31 of the user stories, where it was hoped to test the remaining requirements in the other test levels. The formal testing process was carried out in both sprints and the testing results are presented in Appendix B with the results being evaluated in Chapter 6.

5.3.2 Usability Testing

This involves a potential user who was not involved in the development activities. This method provides valuable feedback from a user’s perspective as it is of one which is likely to be neutral considering they have no ties to the project.

The usability testing focused on the non-functional requirements of the application (as outlined in Table 3.2) and selected core functions defined in User Stories. This involved the developer and potential user during both sprints as the developer was able to witness first hand:

- How easy it is to use the application;
- Is the application convenient to the end user;
- How easy it is for users to complete basic functions;
- How fast the main function can be completed;
- How many errors do the user make, their severity, and whether users can recover from the errors;
- How much the user liked using the application (subjectively);
- Usability testing – asking potential users to complete a particular task, to see where they encounter problems and experienced confusion.

During the testing the developer was able to provide guidance and ask the user to complete the requested functions.

A report for this testing was completed with results which also allowed the user to leave comments to provide valuable feedback to which the application could benefit from, and which the developer was able to take into consideration at the next stage of the development. A copy of the report is presented in Appendix C, with the results evaluated in the Chapter 6.

5.3.3 Exploratory Testing

With time severely limited the developer believed that this method would provide flexibility so that the developer had the ability to quickly react to changes and adapt the project accordingly in to order to deliver the best possible project.

The exploratory testing would be carried out without script however will follow the following structure:

- Explore – by reviewing the project objectives outlined in Section 1.3, project requirements (as defined in the user stories presented in Appendix A) and non-functional requirements (outlined in Table 3.2) and understanding what each function should be doing, to provide a fuller understanding of the application;
- Test design – creating strategies to carrying out while testing to evaluate the application;
- Execution – explore the application to see how it works, e.g. browse as a user, contractor and administrator;
- Results – when testing was completed the tester was able record any problems and present these to the developer.

This method was again carried out after the initial development and a further sprint, with the resultant test reports presented in Appendix D**.** The results of which will be discussed in Chapter 6.

6.0 Evaluation and Conclusion

This section will appraise the project and whether the project aims and objectives where achieved by considering the results of the testing activities in context of the stated requirements to evaluate the overall success of the project. Suggestions for further work will also be discussed.

6.1 Developer Platform

The developer implemented the choses which were made in chapter 4. Using PHP, MySQL and JavaScript accordingly throughout to fulfil the needs and requirements of the project. The developer previously worked briefly with these before and was able to go into greater detail and use these to a greater effective than with previous projects. However, this was not completed without difficulty as many functions implemented within the system where new to the developer and some took more time than expected.

6.2 Project Appraisal

This section will discuss the project's performance in compliance with the aim and objectives discussing in detail the test results and how well the project was managed.

6.2.1 Project Aims and Objectives

During the early stages of the project aim and objectives were developed (Sections 1.2 and 1.3), along with the project requirements (as defined in user stories presented in Appendix A) and complied with throughout. From the beginning the developer made the decision to focus attention on the 'must have', 'should have' and non-functional requirements as these would help achieve the main aims and objectives of the project. Successfully, these requirements were 100% complied with and helped deliver a fully operational system. The 'Could have' requirements where considered those that would not have a negative impact on the system if they were not achieved and would be addressed if time was left. Out of the 5 'could have' user stories 2 where not fully complied with (i.e. user stories #28 and #32), however their omission from the system did not have any negative impact and therefore did not have an impact on the overall aims and objectives.

6.2.2 Testing Results

All the testing for the system was carried out after the initial development stage and one further sprint. The results will now be evaluated in the following sections.

Formal Testing

A summary of the results is shown in Table 6.1 (overleaf) with a copy of the test conditions and test cases (for both stages) presented in Appendix B**.**

Table 6.1 Summary of test cases carried out during formal testing

Development Stage	Test cases executed	Total Passed	Total Failed
Initial	34	30	4
Sprint 1	34	34	0

The four test cases that failed after the initial development were deemed major defects and required immediate attention before release. The results presented in Table 6.1 show that at the next development sprint these test cases were fully tested again. The user stories were only deemed fully complied with when no test cases returned failed.

The developer found the formal testing method very effective as it was able to test a large part of the functionality of the application by following the test conditions and cases ensuring that user stories were fully complied with.

Usability Testing

Table 6.2 shows a summary of results found during the usability testing, once again a copy of report is in Appendix C**.**

Table 6.2 Summary of Results Found During Usability Testing

Development Stage	Total Non-Functional requirements tested	Total Failed	Total tasks carried out	Total tasks failed
Initial	6	3	16	3
Sprint 1	6	0	16	0

The test results after the initial development stage show that 6 of the initial 22 tests carried out failed at this stage. The developer found this method very beneficial to the overall development of the application as it allowed first hand observation of a potential user who had no ties to the project itself. The feedback proved very valuable to the developer as he was able to take this information on board amend the issues and develop further.

Some of the main results to come from this testing method after the initial development stage was the number of non-functional requirements failing which basically shows how the application is qualitatively performing. This was a major issue at the time and clearly showed the application was not ready for release at this stage.

One of the suggestions after the initial development stage from the tester was to allow the contractor to be able to do the functions a general user can do, i.e. if a contractor is looking a personal job done which is out of their level of expertise (for example where a contractor who is a window-cleaner is looking is garden cut). This was considered a relevant and applicable suggestion, and the development team was able to amend the application to suit as this would prevent the contractor having to have two separate accounts.

Exploratory Testing

A copy of the report for the exploratory testing can be found in Appendix D**.** This method proved very valuable to the application's development as it identified a lot of minor defects and some major errors which could have been very damaging to the application. These included:

- Logging out – session did not fully destroy when user logged out and when the next person logged in on the same device they were able to view the previous user's details. The code used to fix this can be seen in Appendix E**.**
- Using the back button on several pages caused the error 'confirm form resubmission' and once the back button was pressed again the user was redirected to the index page. The code which resolved this issue is in Appendix E**.**

The developer believes the testing methods used to test the application were successful considering the tight time constraints for the project. However, if given time more testing methods would have been implemented.

The results that were found at each stage allowed the developer to react quickly and adapt them accordingly so that the project teams goal could be achieved in delivering a functionality web application.

6.3 Management of Project

This refers to the project plan set out in Section 1.5 and Gannt chart presented in figure 1.3**.** Due to foreseen personal circumstances which were discussed in detail with the project supervisor the deadline for the project was extended three weeks. The plan was adjusted to suit, and key milestones were all met accordingly. Regular meetings were held with the project's supervisor to discuss issues and monitor progress. A copy of the meeting minutes is presented in Appendix F**.**

6.4 Further Work

Despite the requirements being successfully met within the application the developer suggested the following further work which could be carried out that would benefit the system:

- More control for the administrator
- GPS location on jobs
- Notifications
- Payment
- Contractor no-show
- Additional Services
- Adding pictures

6.4.1 More Control for the Administrator

Due to time constraints the developer was unable to go into detail with the administrator side of things. Despite being able to remove or issue a warning to contractor when reported. The admin functions are still limited to what they can do. The developer has suggested the following tasks which an administrator should be able to carry out:

- View/remove user's jobs
- Remove/issue warning to a general user

6.4.2 GPS Location on Jobs

Due to the developer's lack of expertise in this area he was unable to implement an Application Programming Interface (API) relating to a GPS. The developer believed that this would have been a feature that would increase user experience as it could have been used when a contractor was browsing jobs to place an offer. Contractors could view in a map how far roughly the job is from their location giving the contractor an estimate on travel time when considering placing an offer.

6.4.3 Notifications

Another feature which the developer believed would be hugely beneficially to the application is a feature for notifications to both the user and contractor in situations such as:

- When a user has posted a job, a notification could be sent to all potential contractors that meet the criteria for that job will a link to the job and giving them the possible to place an offer quickly and easily;

- A user could then receive a notification, informing them that an offer has been placed on their job with a link to view the offer and decide whether to accept or not;

- Once a user has assigned a job to a contractor could then receive a notification informing them of this decision.

6.4.4 Payment

With the application only being a prototype, the developer did not go into much detail in terms of payment. If application was brought to market this is a feature that would have to be implemented if brought forward, including storing of card details in the system database, and sending and receiving money.

6.4.5 Contractor No-Show

Something the developer would have like to introduce was a feature for the user to report when a contractor has failed to show for a job in which they have committed to doing. This could lead to a fine for the contractor or a negative review.

6.4.6 Additional Services

The developer has only included several services that can be carried out by a registered contractor. However, if the prototype was put forward for release there could be endless possibilities for services that could be added, including anything from dog grooming, to fixing a phone, to baking a cake. The developer believes this addition would increase the demand for application and the overall success if implemented correctly.

6.4.7 Adding Pictures

Being able to add pictures to job would be a feature the developer would have liked to see implemented. This would provide clarity to the contractor as they would be able to obtain pictures of what the job entailed before placing an offer on the job.

6.5 Conclusions

In considering the overall project and the aim and objectives which were originally set out in Section 1.2 and Section 1.3, the project delivered by the developer can be considered a somewhat success as the stated project aim and objectives were met, as demonstrated by compliance with the core requirements of the project. Ultimately, a functionable web application has been developed which should be easy to use for users of all ages.

The developer felt that there was a wide range of functionality provided using the PHP and JavaScript development platforms being used thoroughly throughout to provide users with the ability to carry out required functions such as signing up, posting a job, accepting offers, assigning jobs, etc without encountering problems.

However, even though the project has met requirements the developer believes it is still not ready for commercial release as the function ‘Payment’ (section 6.4.4) would need to be included in detail ensuring users could send money and contractors could receive money for jobs completed within the application.

References

BABOK 05/08/18- last update - ELICITATION TECHNIQUES USED BY BUSINESS ANALYSTS – available (online): https://www.bridging-the-gap.com/elicitation-techniques-business-analysts/ [05/08/2018]

BILL GATES - THE GRID BLOG -– last update – 05/07/18 available (online): https://blog.thegrid.io/if-your-business-is-not-on-the-internet-then-your-business/ [05/07/18]

DAI CLEGG(1994) - MOSCOW: REQUIREMENTS PRIORIZATION TECHNIQUE –available (online): https://businessanalystlearnings.com/ba-techniques/2013/3/5/moscow-technique-requirements-prioritization [06/07/2018]

DOWNPATRICK BUY AND SELL, 05/07/2018 – last update available (online) at https://www.facebook.com/search/top/?q=downpatrick%20buy%20%26%20sell Accessed (June 2018)

DR A.P.J. ABDUL KALAM – May 2017 – Last update – WHAT ARE THE ADVANTAGES OF USING PHP FOR WEB DEVELOPMENT? – available (online): https://www.quora.com/What-are-the-advantages-of-using-PHP-to-make-a-web-app [15/09/2018]

ENVATO MARKET – last update 05/07/18- DEMO WEBSITE – available (online):http://preview.themeforest.net/item/spagere-online-booking-html-website-template/full_screen_preview/20698527?_ga=2.155564091.1230451988.1530606911-1211440529.1530006648 [05/07/2018]

HANDY.COM – 04/06/2018 – last update available (online) at www.handy.com [04/06/2018]

INTERNET ACCESS – HOUSEHOLDS AND INDIVIDUALS UK 2018 last update – September 2018 available online: https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2018 [15/06/2018]

MAYA KOSOFF BUSINESS INSIDER, 2018 – last update. Handy valuation (online) available: http://uk.businessinsider.com/handy-raises-50-million-at-a-500-million-valuation-2015-11?r=US&IR=T [09/06/2018]

MOUNTAIN GOAT 2018 – last update - USER STORIES – available (online): https://www.mountaingoatsoftware.com/agile/user-stories [06/07/2018]

MUDASSAR KHAN - July 2015 - JQUERY FAQS — PASSWORD AND CONFIRM PASSWORD VALIDATION USING JQUERY AVAILABLE (online)::http://www.jqueryfaqs.com/Articles/Password-and-Confirm-Password-validation-using-jQuery.aspx [04/07/2018]

MUDIT JAIN - April 2018 -Last update - AJAX LOGIN FORM USING JQUERY, PHP AND MYSQL – available (online): http://talkerscode.com/webtricks/ajax-login-form-using-jquery-PHP-and-mysql.PHP [06/07/2018]

MUDIT JAIN – April 2018 – last update - CHECK USERNAME AND EMAIL AVAILABILITY FROM DATABASE USING AJAX – available (online): http://talkerscode.com/webtricks/check%20username%20and%20email%20availability%20from%20database%20using%20ajax.PHP [08/07/2018]

ROLLINS, A., 11 August 2009 – last update. DATABASE NORMALIZATION: FIRST, SECOND AND THIRD NORMAL FORMS [HOMEPAGE OF ANDREW ROLLINS], Available (online): http://www.andrewrollins.com/2009/08/11/database-normalization-first-second-and-third-normal-forms/ [02/09/2018].

SARAH PEREZ 2016 – last update - NEARLY 1 IN 4 PEOPLE ABANDON MOBILE APPS AFTER ONLY ONE USE –available (online): https://techcrunch.com/2016/05/31/nearly-1-in-4-people-abandon-mobile-apps-after-only-one-use/ [05/09/2018]

SHYAM SUNDAR VERMA 2016- last update-TECHNOLOGIES TO CRAFT A SCALABLE WEB SOLUTION –available (online): https://readybytes.in/blog/technologies-to-craft-a-scalable-web-solution [04/08/2018]

STACK OVERFLOW – July 2018 – last update - MYSQL LIKE MULTIPLE VALUES – available (online): https://stackoverflow.com/questions/4172195/mysql-like-multiple-values [07/07/2018]

Appendix A – User Stories

Must Have

#01	Registering
As an unregistered user I can register so that I can log in	
Acceptance Criteria • Must enter a first and last name • Must enter email address • Must enter a password and confirm it • Must enter an address Once all the correct details are entered the unregistered user can become a registered on the site.	

#02	Registering
As a user, I am required to enter a strong password when creating my account	
Acceptance Criteria • Must have at least 8 characters • Must contain at least 1 digit • Must contain at least 1 uppercase letter • Must contain at least 1 symbol The HouseProud web application should only allow these credentials for a valid password.	

#03	Logging in
As a registered user I can log in to the website to view my account	
Acceptance Criteria • Must enter valid username • Must enter valid password If correct login credentials are used the web application will log you in to the home page and your first name should appear on the nav bar.	

#04	Posting a job
As a registered user I want to be able to post a job so that my job can be reviewed by a contractor	
Acceptance Criteria When the user clicks on ‘Post a job’ and enters the details for the job (i.e. job name, job category, job address, job budget, job description and job due date). Once submitted the web application should display a message notifying the user that the job has been posted and now available to view in the ‘my jobs section’. Also, once the reviewer has submitted the job is now available for the appropriate contractors to view in the ‘browse jobs’ section.	

#05	Apply to become a contractor
As a registered user I want to be able to apply to be a contractor, so I can make offers for jobs and complete tasks	
Acceptance Criteria When the user clicks on ‘Work for us’ they follow the on screen instructions and fills out the required fields (i.e. services provided, counties covered, date of birth, transportation, qualifications, bio, upload a copy of their portfolio, a police check and photography identification) once form is submitted a message will indicating that the form has been submitted. Once submitted the form will be sent to the administrator of the site and the contractor will await their decision. Once a decision is made a notification email will be sent to the contractor	

#06	When a user becomes a contractor
As a registered contractor I want to be able to log in, so I view the contractor’s section	
Acceptance Criteria When a registered user first becomes a registered contractor, they should be able to log in with their existing credentials and view the contractor’s section (i.e. browse jobs, contractor profile)	

<table>
<tr><td>#07</td><td>Browsing jobs as a contractor</td></tr>
<tr><td colspan="2">As a registered contractor I want to be able to browse jobs, so I can make offers for jobs</td></tr>
<tr><td colspan="2"><b>Acceptance Criteria</b>
When the registered contactor is logged in they can go to the ‘browse jobs’ section and should be able to view jobs and make an offer on jobs that meet their criteria i.e. county that they cover and particular job category matches that of their preferences.
A registered contractor should not be able to make an offer or access a job that does not meet their criteria</td></tr>
</table>

<table>
<tr><td>#08</td><td>Contractor making an offer on a job</td></tr>
<tr><td colspan="2">As a registered contractor I want to be able to make an offer on a job so that it can be considered for assignment</td></tr>
<tr><td colspan="2"><b>Acceptance Criteria</b>
When a registered contractor is ‘browsing jobs’ they should be able to make an offer for a job that meets their criteria (i.e. area covered and correct job category). Once an offer has been made there should be a message displayed which states that an offer has been made.</td></tr>
<tr><td>#09</td><td>Contractor viewing jobs that they have made an offer on</td></tr>
<tr><td colspan="2">As a registered contractor I want to be able to view the jobs I have made an offer on</td></tr>
<tr><td colspan="2"><b>Acceptance Criteria</b>
When a registered contractor once an offer has been made job should be moved to the ‘my jobs’ section where contractor can check the status of offer in which they have made.</td></tr>
</table>

#10	Contractor changing job offer
As a registered contractor I want to be able to change an offer that I have already made on a job so that it the offer will change	
Acceptance Criteria When a registered contractor is has made an offer on a job they should then be able to change the offer that they have made. However, this can only be done if the original offer is still pending.	

#11	Leaving a review
As a registered user, I want to be able to leave a review for a contractor so that potential clients can view on the contractor’s profile	
Acceptance Criteria When a job is completed the registered user who posted the job should then have the option to leave a review for the contractor that completed the job. Once the review is complete it should then be able to view on the contractor’s profile page.	

#12	Administrator makes decision on contractor
As administrator of the site I want to be able to view pending contractors so I can make a decision on whether to accept or reject a contractor’s submission	
Acceptance Criteria Once a registered user has completed a form to be a contractor, it should be sent to the ‘pending contractors’ section of the administrator’s profile where only the administrator of the site can view. The administrator should be able to view the pending contractor’s details and make a decision to accept or reject proposal. If proposal is accepted an email should be sent to the user notifying that they are now a contractor and user is added to the list of registered contractors. If proposal is rejected an email should be sent to the user notifying of the decision that has been made. Once a decision has been made by the administrator the form application from that particular user should be removed.	

#13	Change Personal Information
As a registered user I want to be able to change my personal information	
Acceptance Criteria When a registered user is logged in they should be able to go to the settings section and change information that has been saved to the system (i.e. address, phone number). Once the changes have made and submitted they will replace the details that had originally been saved.	

#14	Change Password
As a registered user I want to be able to change my password	
Acceptance Criteria When a registered user is logged in they should be able to go to the ‘settings’ section and in this section they should be able to change their password by entering their existing password and entering their new password. Their new password must meet the criteria which is: • Must have at least 8 characters • Must contain at least 1 digit • Must contain at least 1 uppercase letter • Must contain at least 1 symbol	

#15	Edit job and change details
As a registered user I want to be able to view my posted jobs, so I can amend details	
Acceptance Criteria When a registered user is logged in they should be able to go to ‘my jobs’ section and view/edit a job in which they have posted. They should only be able to edit a job if no offers have been made on this job.	

#16	Removing a job that has not been assigned
As a registered user I want to be able to remove a job	
Acceptance Criteria If a registered user has posted a job and the job has yet to be assigned, the user should be able to remove job and it will be then removed from their jobs section. Once a job has been assigned the user should eb able to remove this job and have committed to it.	

#17	View offers made from contractors
As a registered user I want to be able to view offers on my jobs, so I can decide to either accept or reject	
Acceptance Criteria Once a contractor has made an offer on a user's job, the user is then able to view the offer which has made and should then be able to accept or reject the job. Once a decision has been made the status of the contractor's offer should change on the contractor's own profile.	

#18	Assigning job
As a registered user I want to be able to accept an offer so that a job is assigned to the chosen contractor	
Acceptance Criteria Once a registered user has accepted an offer from a contractor the status of the job should then changed to 'assigned' and once assigned only the contractor and the user can access the details for the job.	

#19	Viewing completed jobs
As a registered contractor/user I want to be able to view my completed jobs so that can view the job details	
Acceptance Criteria Once a job has been completed the job should then be moved to the ‘my jobs completed’ sections of the user that posted the jobs and contractor that completed the jobs, so they can view details of the job (i.e. price, details, etc)	

#20	View Contractor’s profile page
As a registered user, I want to able to view a contractor’s profile page, so I can view their profile	
Acceptance Criteria When a contractor makes an offer on a user’s job they should be able to click on their name and view their profile, on the profile page their will be a profile picture, description, a list of tasks which they carry out and the area which they cover. On this profile page there should also be an overall average star rating of the jobs which they have carried out and a list of individual ratings and reviews of the previous jobs which the contractor has completed.	

Should Have

#21	View all posted jobs in a table
As a registered user I want to be able to view my posted jobs	
Acceptance Criteria Once a user has posted a job, it will be able to view in a ‘my jobs’ sections, from there the user will be able to view the job details i.e. offers, edit details, etc.	

#22	Log in and view user’s own profile
As a registered user I want to be able log in, so that I can view my profile	
Acceptance Criteria Once a registered user has logged in they should be able to view their own profile (i.e. ratings and reviews)	

#23	Contractor asking a question about a job
As a registered contractor I want to be able to ask a question about a job so that I can find out more about the job	
Acceptance Criteria A contractor should be able to ask a question about a job before they have made an offer or when they have made an offer. This should then allow the user who posted the job to reply to the contractor, so they can find out more information on the job.	

#24	Contractor leaving a review for user
As a registered contractor I want to be able to leave a review so that other contractors can view user’s reviews	
Acceptance Criteria When a registered contractor has completed a job they should have an option to leave a review for the user. This should be scored on how well the user has described the job and how professional they were when working with them. Once a review is completed the contractor should be then brought to a thank you page, and the review will then be available for other contractors to see on the user’s profile page.	

<table>
<tr><td>#25</td><td>Contractor viewing user’s profile</td></tr>
<tr><td colspan="2">As a registered contractor I want to be able to view a user’s page so that I can decide whether to make an offer on their job</td></tr>
<tr><td colspan="2"><b>Acceptance Criteria</b>
When a registered contractor is ‘browsing jobs’, they should be able to view the profile of the user who posted the job. On this page there should be a profile picture and ratings and reviews from previous contractors who have done work for this user.</td></tr>
</table>

<table>
<tr><td>#26</td><td>Report a contractor</td></tr>
<tr><td colspan="2">As a user I want to be able to report a contractor when something goes wrong</td></tr>
<tr><td colspan="2"><b>Acceptance Criteria</b>
When a user believes it is necessary to report a contractor to the administrator for a certain reason i.e. not turning up, doing a bad job, etc. they should be able to fill out a form that reports the contractor to the administrator of the web application.</td></tr>
</table>

<table>
<tr><td>#27</td><td>Administrator can issue warning/remove contractor</td></tr>
<tr><td colspan="2">As administrator for the web application I want to be able to issue a warning to a contractor</td></tr>
<tr><td colspan="2"><b>Acceptance Criteria</b>
When a user has reported a contractor, the report should be sent to the administrator so that they can view the report and make a decision whether to ignore, issue a warning or remove a contractor.</td></tr>
</table>

Could Have

#28	User adding a payment method
As a user I want to add a payment method so that I can pay for a job	
Acceptance Criteria A user should be able to add a payment method ie credit card so that it is saved to their profile and used to pay for jobs.	

#29	Contractor being able to change their details
As a registered contractor I want to change my details so it is saved to my profile	
Acceptance Criteria A registered contractor should be able to change their details i.e. bio and transportation. Once changed it will be saved to their profile	

#30	Administrator can view all users
As administrator for the web application I want to be able to view all users of the web application	
Acceptance Criteria The administrator should be able to log in to their account and be able to view users you use the web application.	

#31	Administrator can view all contractors
As administrator for the web application I want to be able to view all registered contractors of the web application	
Acceptance Criteria The administrator should be able to log in to their account and be able to view the registered contractors of the web application.	

#32	Adding pictures for jobs
As a user I want to be able to add a picture of a job so potential contractors can see	
Acceptance Criteria The user should be able to add a picture for the job in the messaging section of the job	

Won’t Have This Time

#33	Jobs marked on a map
As a registered contractor I want to locate jobs on a map so I can decide if I want to place an offer	

#34	Money to be transferred once a job is complete
As a user I want to transfer money to the contractor once a job is complete so I can pay the contractor the money they are owed	

#35	Notification when an offer has been made
As a user I want to receive a notification once a contractor has made an offer so I can decide whether or not to accept it.	

#36	Notification when an offer has been accepted or rejected
As a registered contractor I want receive a notification on an offer that I have made so I can see the decision of the offer.	

#37	Administrator be able to remove a user
As administrator I want to be able to remove a user so they can no longer use the web application	

#38	Administrator be able to remove a job
As administrator I want to be able to remove a job so it is no longer available on the application	

Appendix B - Test Conditions and Procedures

Test Condition ID	Description	Source	Priority
Sec_TCon_1	To show that unregistered users cannot navigate to pages that there are not authorised too.	User Story 01	High
Sec_TCon_2	To show that each user's email address must be unique	User Story 01	High
Sec_TCon_3	To Show that when registering each field must contain data before continuing	User Story 01	High
Sec_TCon_4	To show that when registering a password that the system will only accept a password containing at least 6 characters, 1 number, 1 uppercase character and 1 lowercase character.	User Story 02	High
Sec_TCon_5	To show when a registered user signs in that they are directed to the home page and that their first name appears on the navigation bar	User Story 3	Medium
Sec_TCon_6	To show that only registered contractors can access contractor's section i.e. browse jobs, make offers	User Story 6	High
Sec_TCon_7	To show that only the administrator for the application can view pending contractors and accept/reject their proposal	User Story 12	High
Sec_TCon_8	To show that when a registered user changes a password that the system will only accept a password containing at least 6 characters, 1 number, 1 uppercase character and 1 lowercase character.	User Story 14	High
Us_TCon_9	To show once logged in as a general user, they will be able to navigate to	User Story 4	High

	‘Post a Job’ and successfully complete the form		
Us_TCon_10	To show once a user has submitted a form for post a job that it is available to view in my jobs section	User Story 4	Medium
Us_TCon_11	To show that a registered contractor can only view the jobs which meets their criteria i.e. area covered and job category	User Story 7	High
Us_TCon_12	To show that a registered contractor can make an offer on available jobs	User Story 8	High
Us_TCon_13	To show once a registered contractor has placed an offer on a job that the job is available in the contractor’s my job section	User Story 9	Medium
Us_TCon_14	To show that a registered contractor can change the offer that they have made on a job	User Story 10	High
Us_TCon_15	To show that once a job is complete that a user can leave a review for the contractor who completed the job	User story 11	Low
Us_TCon_16	To show that as a registered user of the application that they can change personal information	User Story 13	High
Us_TCon_17	To show that a user can edit job details of a job that they have posted	User Story 15	High
Us_TCon_18	To show that a user can remove a job that they have posted	User Story 16	Medium
Us_TCon_19	To show that users can view offers from contractors for a job that they have posted	User Story 17	High
Us_TCon_20	To show that a user can accept an offer from a contractor for a job that they have posted	User Story 18	High

Us_TCon_21	To show that a user can reject an offer from a contractor for a job that they have posted	User Story 18	High
Us_TCon_22	To show that a general user can view completed jobs	User Story 19	Low
Us_TCon_23	To show that a registered contractor can view completed jobs in which they have done	User Story 19	Low
Us_TCon_24	To show that a general user can view a contractor’s profile page	User Story 20	Low

Development Stage – Initial Date – 13/08/2018 Tester – Justin Kelly

Test Case ID	Objective	Preconditions	Expected Result	Test Conditions	Status	Severity
Sec_TCase_1	Check that unregistered users are unable to access pages that require log in credentials	On the home page	Redirected to home page	Sec_TCon_1	Passed	
Sec_TCase_2	Check that an email address can not be used more than once	Sign up	Message displaying that email address already existed	Sec_TCon_2	Passed	
Sec_TCase_3	Check that when users sign up each field must contain data before proceeding	Sign up	Unable to proceed until each field is complete	Sec_TCon_3	Passed	
Sec_TCase_4	Check that users are unable to register with a password shorter than six characters, not containing at least one number, one lowercase or one uppercase	Sign up	Password will not be accepted	Sec_TCon_4	Passed	
Sec_TCase_5	Check that users are unable to register with a password of six characters but does not contain at least one number, one lowercase or one uppercase	Sign up	Password will not be accepted	Sec_TCon_4	Passed	

Sec_TCase_6	Check that users can register with a password containing at least six characters along with at least one number, lowercase and uppercase character	Sign up	Password will be accepted	Sec_TCon_4	Passed	
Sec_TCase_7	Check that when a registered user signs in that they are directed to the home page with their first name on the navigation bar	Sign in	User will be directed to the home page with their first name displayed on the navigation bar	Sec_TCon_5	Passed	
Sec_TCase_8	Check that a general user cannot access a contractor's section i.e. make offers, browse jobs	Signed in as a general user	User will be redirected to the home page	Sec_TCon_6	Passed	
Sec_TCase_9	Check that a registered contractor can access the contractor's section i.e. make offers, browse jobs	Signed in as a contractor	Will be able to access pages	Sec_TCon_6	Passed	
Sec_TCase_10	Check that a general user cannot access admin page to view pending contractors	Signed in as general user	Will be redirected to the home page	Sec_TCon_7	Passed	

Sec_TCase_11	Check that a contractor cannot access admin pages to view pending contractors	Signed in as a contractor	Will be redirected to the home page	Sec_TCon_7	Passed	
Sec_TCase_12	Check that the administrator can access the page to view pending contractors	Signed in as admin	Will be able access page	Sec_TCon_7	Passed	
Sec_TCase_13	Check that the administrator can accept/reject a pending contractor’s proposal	Signed in as admin	Will be able to accept/reject a pending contractor’s proposal	Sec_TCon_7	Passed	
Sec_TCase_14	Check that a user cannot change their password to something that does not meet the credentials i.e. at least six characters, one number, one lowercase and one uppercase character	Signed in as a user	Will not be able to change password	Sec_TCon_8	Passed	
Sec_TCase_15	Check that a user will be able to change their password to a something that meet the credentials i.e. at least six characters, one number, one lowercase and one uppercase character	Signed in as user	Will be able to change password	Sec_TCon_8	Passed	

Us_TCase_16	Check that when a user is signed in they can access the ‘Post a job’ page	Signed in as user	Will be able to access page	Us_TCon_9	Passed	
Us_TCase_17	Check that a user can complete the ‘Post a Job’ with correct details i.e. address, budget(number only)	Sign in as user on the ‘post a job’ page	Will be able to post a job	Us_TCon_9	Passed	
Us_TCase_18	Check that once a user has submitted a for the ‘post a job’ that the jobs now appears on their my jobs section	Sign in as user	Job will now appear in my jobs section	Us_TCon_10	Passed	
Us_TCase_19	Check that contractor can only view jobs that meet their criteria	Signed in as contractor	Contractor will only be able to view jobs that meet their criteria	Us_TCon_11	Failed	Major
Us_TCase_20	Check that a contractor can only make offers on jobs that meet their criteria	Signed in as contractor	Contractor will only be able to make an offer on a job that meets their criteria	Us_TCon_12	Failed	Major
Us_TCase_21	Check once a contractor has placed an offer on a job that it can	Signed in as contractor	Job will be able to view in the my jobs section	US_TCon_13	Passed	

	be viewed in the contractor’s my jobs section					
Us_TCase_22	Check that contractor can change their offer that they have made on a job	Signed in as contractor, status of job must be pending	Contractor will be able to change offer	US_TCon_14	Passed	
Us_TCase_23	Check that once a job is complete that a user can leave a review	Signed in as a user and user has selected the job complete option	There will be an option to leave a review	Us_TCon_15	Passed	
Us_TCase_24	Check that a user of the application can change their personal information	Signed in as a user	That save changes that the new information will be changed	Us_TCon_16	Passed	
Us_TCase_25	Check that a user can change the details of a job that have posted when no offer has been made	Signed in as user	That the user will be able to edit the job that they have posted	Us_TCon_17	Passed	
Us_TCase_26	Check that a user cannot change the details of a job that they have posted when there is an offer pending in that job	Signed in as user	User will be unable to make changes to the job	Us_TCon_17	Failed	Major

Us_TCase_27	Check that a user can remove a job that they have posted when there is no offers pending	Signed in as user	User will be able to remove job	Us_TCon_18	Passed	
Us_Tcase_28	Check that a user cannot remove a job that an offer has been accepted	Signed in as user	User will be unable to remove job	Us_TCon_18	Failed	Major
Us_TCase_29	Check that a user can view offers from contractors that have made an offer	Signed in as user	User will be able to view offers from contractor	Us_TCon_19	Passed	
Us_TCase_30	Check that a user can accept a job from a contractor for a that they have posted	Signed in as user	User will be able to accept an offer for their job and that contractor will be assigned	Us_TCon_20	Passed	
Us_TCase_31	Check that a user can reject an offer from a contractor who has made an offer for their job	Signed in as user	User will be able to reject an offer from a contractor and that offer will be removed from the job	Us_TCon_21	Passed	
Us_TCase_32	Check that a general user can view completed jobs	Signed in as user and have completed jobs	User will be able to access a page	Us_TCon_22	Passed	

			where their completed jobs are			
Us_TCase_33	Check that a contractor can access completed jobs page where the jobs that they have worked on are there	Signed in as contractor and have completed	Contractor will be able to access a where there completed jobs are	Us_TCon_23	Passed	
Us_TCase_34	Check that a user can view a contractor's profile page	Signed in as user	User will be able to view contractor's profile page	Us_TCon_24	Passed	

Development Stage – Sprint 1   Date – 30/08/2018   Tester – Justin Kelly

Test Case ID	Objective	Preconditions	Expected Result	Test Conditions	Status	Severity
Sec_TCase_1	Check that unregistered users are unable to access pages that require log in credentials	On the home page	Redirected to home page	Sec_TCon_1	Passed	
Sec_TCase_2	Check that an email address cannot be used more than once	Sign up	Message displaying that email address already existed	Sec_TCon_2	Passed	

Sec_TCase_3	Check that when users sign up each field must contain data before proceeding	Sign up	Unable to proceed until each field is complete	Sec_TCon_3	Passed	
Sec_TCase_4	Check that users are unable to register with a password shorter than six characters, not containing at least one number, one lowercase or one uppercase	Sign up	Password will not be accepted	Sec_TCon_4	Passed	
Sec_TCase_5	Check that users are unable to register with a password of six characters but does not contain at least one number, one lowercase or one uppercase	Sign up	Password will not be accepted	Sec_TCon_4	Passed	
Sec_TCase_6	Check that users can register with a password containing at least six characters along with at least one number, lowercase and uppercase character	Sign up	Password will be accepted	Sec_TCon_4	Passed	
Sec_TCase_7	Check that when a registered user signs in that they are directed to the home page with their first name on the navigation bar	Sign in	User will be directed to the home page with their first name displayed on the navigation bar	Sec_TCon_5	Passed	

Sec_TCase_8	Check that a general user cannot access a contractor's section i.e. make offers, browse jobs	Signed in as a general user	User will be redirected to the home page	Sec_TCon_6	Passed	
Sec_TCase_9	Check that a registered contractor can access the contractor's section i.e. make offers, browse jobs	Signed in as a contractor	Will be able to access pages	Sec_TCon_6	Passed	
Sec_TCase_10	Check that a general user cannot access admin page to view pending contractors	Signed in as general user	Will be redirected to the home page	Sec_TCon_7	Passed	
Sec_TCase_11	Check that a contractor cannot access admin pages to view pending contractors	Signed in as a contractor	Will be redirected to the home page	Sec_TCon_7	Passed	
Sec_TCase_12	Check that the administrator can access the page to view pending contractors	Signed in as admin	Will be able access page	Sec_TCon_7	Passed	
Sec_TCase_13	Check that the administrator can accept/reject a pending contractor's proposal	Signed in as admin	Will be able to accept/reject a pending contractor's proposal	Sec_TCon_7	Passed	
Sec_TCase_14	Check that a user cannot change their password to something that	Signed in as a user	Will not be able to change password	Sec_TCon_8	Passed	

	does not meet the credentials i.e. at least six characters, one number, one lowercase and one uppercase character					
Sec_TCase_15	Check that a user will be able to change their password to a something that meet the credentials i.e. at least six characters, one number, one lowercase and one uppercase character	Signed in as user	Will be able to change password	Sec_TCon_8	Passed	
Us_TCase_16	Check that when a user is signed in they can access the ‘Post a job’ page	Signed in as user	Will be able to access page	Us_TCon_9	Passed	
Us_TCase_17	Check that a user can complete the ‘Post a Job’ with correct details i.e. address, budget(number only)	Sign in as user on the ‘post a job’ page	Will be able to post a job	Us_TCon_9	Passed	
Us_TCase_18	Check that once a user has submitted a for the ‘post a job’ that the jobs now appears on their my jobs section	Sign in as user	Job will now appear in my jobs section	Us_TCon_10	Passed	

Us_TCase_19	Check that contractor can only view jobs that meet their criteria	Signed in as contractor	Contractor will only be able to view jobs that meet their criteria	Us_TCon_11	Passed	
Us_TCase_20	Check that a contractor can only make offers on jobs that meet their criteria	Signed in as contractor	Contractor will only be able to make an offer on a job that meets their criteria	Us_TCon_12	Passed	
Us_TCase_21	Check once a contractor has placed an offer on a job that it can be viewed in the contractor's my jobs section	Signed in as contractor	Job will be able to view in the my jobs section	US_TCon_13	Passed	
Us_TCase_22	Check that contractor can change their offer that they have made on a job	Signed in as contractor, status of job must be pending	Contractor will be able to change offer	US_TCon_14	Passed	
Us_TCase_23	Check that once a job is complete that a user can leave a review	Signed in as a user and user has selected the job complete option	There will be an option to leave a review	Us_TCon_15	Passed	
Us_TCase_24	Check that a user of the application can change their personal information	Signed in as a user	That save changes that the new information will be changed	Us_TCon_16	Passed	

Us_TCase_25	Check that a user can change the details of a job that have posted when no offer has been made	Signed in as user	That the user will be able to edit the job that they have posted	Us_TCon_17	Passed	
Us_TCase_26	Check that a user cannot change the details of a job that they have posted when there is an offer pending in that job	Signed in as user	User will be unable to make changes to the job	Us_TCon_17	Passed	
Us_TCase_27	Check that a user can remove a job that they have posted when there is no offers pending	Signed in as user	User will be able to remove job	Us_TCon_18	Passed	
Us_Tcase_28	Check that a user cannot remove a job that an offer has been accepted	Signed in as user	User will be unable to remove job	Us_TCon_18	Passed	
Us_TCase_29	Check that a user can view offers from contractors that have made an offer	Signed in as user	User will be able to view offers from contractor	Us_TCon_19	Passed	
Us_TCase_30	Check that a user can accept a job from a contractor for a that they have posted	Signed in as user	User will be able to accept an offer for their job and that contractor will be assigned	Us_TCon_20	Passed	

Us_TCase_31	Check that a user can reject an offer from a contractor who has made an offer for their job	Signed in as user	User will be able to reject an offer from a contractor and that offer will be removed from the job	Us_TCon_21	Passed	
Us_TCase_32	Check that a general user can view completed jobs	Signed in as user and have completed jobs	User will be able to access a page where their completed jobs are	Us_TCon_22	Passed	
Us_TCase_33	Check that a contractor can access completed jobs page where the jobs that they have worked on are there	Signed in as contractor and have completed	Contractor will be able to access a where there completed jobs are	Us_TCon_23	Passed	
Us_TCase_34	Check that a user can view a contractor's profile page	Signed in as user	User will be able to view contractor's profile page	Us_TCon_24	Passed	

Appendix C – Usability Testing Results

Usability testing form

Development Stage – Initial Tester – Ryan Kelly Date – 14/08/18Observer – Justin Kelly (Developer)

Feature name/item to test	Task Completed (Yes or No)	Comments/Improvement ideas
Sign up to application	**Yes**	Was able to sign up with no issues, very straightforward
Post a job	**Yes**	Task completed, however can be confusion what to enter into the budget column. Whole numbers or decimals?
Edit Job (Change price)	**Yes**	Straightforward
Register to be contractor (Click on 'work for us')	**Yes**	n/a
Change profile picture	**No**	Was able to add a document as a picture
Make offer on a job (as a contractor)	**Yes**	Once offer was made offer status changed to pending
Edit job offer (as a contractor)	**Yes**	Straightforward
Remove job (as a general user)	**No**	Cannot remove job
Edit settings (change address)	**Yes**	Straightforward
Reject job offer from contractor	**Yes**	Could this be done by a message displaying rather than directing to another page?
Accept job offer from contractor	**Yes**	Same as previous

Message user as contractor about a job	**Yes**	Works great
Reply to contractor	**No**	If there was more than one reply feature would not execute
View Contractor’s profile	**Yes**	One clicked on contractor’s name was directed to contractor’s profile page as expected
Click on complete job option are you able to leave a review?	**Yes**	Once job complete button is selected then directed to leave review page for that contractor
Report contractor	**Yes**	However, is there an option to select if contractor doesn’t show by the due date?

Requirement	**Passed? (Y/N)**	**Comments**
SE 01 To access the application valid login credentials are required	**Y**	Redirected to the home page whenever it is tried to access a page requiring log in credentials
TD 01 The system should render/operate consistently on all mobile and desktop devices	**N**	The functions which are complete run fine on a laptop however on a mobile device some pages require attention
RE 01 The system should be user friendly	**N**	When filling out the ‘post a job’ form unsure of what data is required to be entered
US 01 The interface, font and colour scheme used in the application shall be consistent throughout	**N**	Not all pages are finished and will require further testing at next stage
PE 01 Response times – There shall be no apparent delay on opening any of the application’s pages	**Y**	With the pages available there seems to be no apparent delay when opening different application pages
PE 02 Processing times – There shall be no apparent delay when carrying out a function (e.g. posting a job or making an offer)	**Y**	Once a form is complete, the form is processed quickly
Comments/ further work needed		
Further work suggestion: If a contractor was looking to use the application for their own personal use e.g. looking a removal job done as they cannot do this using their contractor profile. Would help the performance and usability of the site if the contractor was able to use their profile for personal use too rather than having two accounts. Should there be a feature for a contractor to leave a review for a user once they complete a job for them. So potential contractors can view their profile to see how well they explain jobs and how fair their budget is etc		

Features that require immediate attention

- Post a job form
- When rendered on a smartphone difficult to navigate through the application

Development Stage – Sprint 1 Tester – Ryan Kelly Date – 05/09/18Observer – Justin Kelly (Developer)

Feature name/item to test	**Task Completed (Yes or No)**	**Comments/Improvement ideas**
Sign up to application	**Yes**	n/a
Post a job	**Yes**	Less confusion now
Edit Job (Change price)	**Yes**	n/a
Register to be contractor (Click on 'work for us')	**Yes**	n/a
Change profile picture	**Yes**	Error seems to be corrected now
Make offer on a job (as a contractor)	**Yes**	n/a
Edit job offer (as a contractor)	**Yes**	n/a
Remove job (as a general user)	**Yes**	Able to remove job when no offers have been made
Edit settings (change address)	**Yes**	n/a
Reject job offer from contractor	**Yes**	Message displaying once feature has been executed
Accept job offer from contractor	**Yes**	Same as previous
Message user as contractor about a job	**Yes**	N/a
Reply to contractor	**Yes**	Reply feature seems to be resolved
View Contractor's profile	**Yes**	N/a

Click on complete job option are you able to leave a review?	**Yes**	n/a
Report contractor	**Yes**	However, is there an option to select if contractor doesn’t show by the due date? (still the same)

Requirement	**Passed? (Y/N)**	**Comments**
SE 01 To access the application valid login credentials are required	**Y**	Redirected to the home page whenever it is tried to access a page requiring log in credentials
TD 01 The system should operate/render consistently on all mobile and desktop devices	**Y**	Application is running consistently on both mobile device and laptop
RE 01 The system should be user friendly	**Y**	System is now much more user friendly
US 01 The interface, font and colour scheme used in the application shall be consistent throughout	**Y**	Application consistent throughout
PE 01 Response times – There shall be no apparent delay on opening any of the application's pages	**Y**	With the pages available there seems to be no apparent delay when opening different application pages
PE 02 Processing times – There shall be no apparent delay when carrying out a function (e.g. posting a job or making an offer)	**Y**	Once a form is complete, the form is processed quickly
Comments		
From the last test huge improvements have been made and system seems to be meeting requirements. Previous defects have been corrected to a satisfactory standard. Suggested work at previous stage has been taken into consideration and implemented successfully.		

Appendix D – Exploratory Testing Report

Exploratory Testing Report Overall Summary	
Development Stage	Initial Development stage
Date	14/08/18
Testing Time	2 hours
Summary of Bugs/Defects Identified	• When going back on several pages display the error ‘confirm form resubmission’ and eventually are directed to the index page. • Query is broken when registering to be contractor • When logging out, session does not fully destroy • Reply function not fully working • Able to add a document as a profile picture • Settings page is showing menu at the top of the page on smaller screens
Overall Summary	Several minor and major errors have been discovered in this session and will resolved before the next session takes place

Exploratory testing Report Overall Summary	
Development Stage	Sprint 1
Date	05/09/18
Testing Time	2 hours
Overall Summary	Errors have been resolved and there seems to no defects affecting the application’s performance and functionality

Appendix E – Significant Sections of Code

Code to destroy session

```
<?php


session_start();
setcookie(session_name(), '', 100);
session_cache_limiter ('private_no_expire, must-revalidate');
session_unset();
session_destroy();
$_SESSION = array();
header("Location: index.php");


  exit;


  ?>
```

Code used so user can use the back button

```
header('Cache-Control: no cache'); //no cache
session_cache_limiter('private_no_expire');
```

Query used to retrieve average rating of contractor

```
$query1="SELECT AVG(Star_rating), COUNT(Star_rating) FROM PRO_Rate_Contractor
                                        WHERE contractorfk=$contractorid";
    $result2 = mysqli_query($conn, $query1) or die(mysqli_error($conn));
    if(mysqli_num_rows($result2)>0){
    $rows= mysqli_fetch_assoc($result2);
    $star=$rows['AVG(Star_rating)'];
    $reviews=$rows['COUNT(Star_rating)'];
    }

    $query2="SELECT * FROM PRO_Rate_Contractor WHERE contractorfk='$contractorid'";
     $result3 = mysqli_query($conn, $query2) or die(mysqli_errno($conn));
```

Function to send message

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
    function send_message()
{
 var message=$("#message").val();
 var job =$("#jobid").val();

 var user=$('#user').val();

 if(message!=="")
 {

  $.ajax
  ({
  type:'POST',
  url:'sendmessage.php',
  data:{



   message:message,
   job:job,

   user:user

  },
  success:function(response) {
  if(response==="OK")
  {
    location.reload();
    alert('Message sent, now able to view in the comments section');
```

Function to make offer

```
 <script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
    function make_offer()
{
 var offer=$("#offer").val();
 var contractor=$("#contractor").val();
  var job= $("#job").val();

 if(offer!==""){

  $.ajax
  ({
  type:'POST',
  url:'makeoffer.php',
  data:{


   offer:offer,
   contractor:contractor,
   job:job
  },
  success:function(response) {
  if(response==="OK")
  {
        location.reload();
    alert("Offer has been made job can now be viewed in my jobs");
  }
  else
  {
```

PHP used to make offer

```
<?php
session_start();
if((!isset($_SESSION["id_40214842"]))||(empty($_POST['offer'])))
{
   header("Location: index.php");
}
    if((!isset($_SESSION["contractorid_40214842"]))||(empty($_POST['offer'])))
{
      header("Location: index.php");
   }

include("connect.php");
$id=$_SESSION['id_40214842'];
$offer=mysqli_real_escape_string($conn,$_POST['offer']);
 $job=mysqli_real_escape_string($conn,$_POST['job']);
 $contractor=mysqli_real_escape_string($conn,$_POST['contractor']);

 $sql="SELECT * FROM PRO_Make_Offer WHERE contractorfk = '$contractor' AND jobfk='$job'";
  $check = mysqli_query($conn, $sql);

if(mysqli_num_rows($check)==0){

     $select_data="INSERT INTO PRO_Make_Offer (quote,quote_accepted,contractorfk,jobfk) VALUES
    ('$offer','Pending','$contractor','$job')";



 $query= mysqli_query($conn, $select_data)or die(mysqli_error($conn));
```

Appendix F Project Supervisor Meeting Minutes

CSC7057 Minutes of Meeting

Student Name: Justin Kelly

Student Number: 40214842

Date of Meeting: 08/06/18

Time of Meeting: 2pm

Other Attendees at Meeting: Angela Allen

Key Points Arising from Meeting:

- Project Kick off
- Proposal
- Problem Specification
- What is already on the market

Next Action as a result of Meeting:

- Database Design
- Problem specification complete
- Research

CSC7057 Minutes of Meeting

Student Name: Justin Kelly

Student Number: 40214842

Date of Meeting: 28/06/18

Time of Meeting: 11.30pm

Other Attendees at Meeting: Angela Allen

Key Points Arising from Meeting:

- Er diagram
- Requirements
- Problem Specification
- Design of web application

Next Action as a result of Meeting:

- Chapter 1 to be almost ready
- Have an idea of the design
- Complete Database

CSC7057 Minutes of Meeting

Student Name: Justin Kelly

Student Number: 40214842

Date of Meeting: 16/07/18

Time of Meeting: 2.00pm

Other Attendees at Meeting: Angela Allen

Key Points Arising from Meeting:

- Chapter 1 completion
- Functionality of web app
- Design of web application

Next Action as a result of Meeting:

- Chapter 2 to be started
- Work on functionality

CSC7057 Minutes of Meeting

Student Name: Justin Kelly

Student Number: 40214842

Date of Meeting: 26/07/18

Time of Meeting: 11.00am

Other Attendees at Meeting: Angela Allen

Key Points Arising from Meeting:

- What has been done since previous meeting
- Functionality of web application
- What functions need to be completed

Next Action as a result of Meeting:

- Function to be almost ready
- Continue to work on report

CSC7057 Minutes of Meeting

Student Name: Justin Kelly

Student Number: 40214842

Date of Meeting: 14/08/18

Time of Meeting: 11.00am

Other Attendees at Meeting: Angela Allen

Key Points Arising from Meeting:

- What has been done since previous meeting
- What functions need to completed
- Finish application within the next few days

Next Action as a result of Meeting:

- Finish Chapters 3 and 4
- Keep Angela updated on situation with my wife (due to give birth in the next week)

CSC7057 Minutes of Meeting

Student Name: Justin Kelly

Student Number: 40214842

Date of Meeting: 03/09/18

Time of Meeting: 11.00am

Other Attendees at Meeting: Angela Allen

Key Points Arising from Meeting:

- What has been done since previous meeting
- Extension required due to birth of son
- Requirement methods
- Testing methods

Next Action as a result of Meeting:

- Continue to work on report
- Keep Angela updated on progress
- Able to communicate via email

Appendix G – URL and Access Details

URL - http://jkelly798.web.eeecs.qub.ac.uk/houseproud/

Username	Password	Access Level
jkelly798	Password123	Administrator
jkelly798@qub.co.uk	Password123	General User
justykelly1987@gmail.com	Password123	Contractor