

University of Plymouth

**School of Engineering,
Computing, and Mathematics**

PRCO304

Final Stage Computing Project

2019/2020

**TRADEWORK.NET – traders in your
neighbourhood**

Joseph Wood

10546126

BSc (Hons) Computer Science

Acknowledgements

I would like to thank my family with special mention to my mother for supporting and encouraging me through this project as she has done through many previously. As well as this, Maddison for being there to help support me and tell me when to stop.

Abstract

TradeWork.NET is an application built to fill a gap in the market where DIY/Trade work has no easily accessible application such as in the styles of Airbnb or Uber, where their applications have transformed their respective industry through hugely increasing accessibility. It provides both a tradespersons version of the app, allowing them to view, quote and accept jobs, and a user's version for which they can see local traders and post new jobs. Enabling the app to act as a way to connect and serve consumers to traders without the need for a middleman, encompassing the traditional methods of joining them and providing a review score-based system to create a community-driven quality assurance. The overall aim is to lower the time to engage with a trader and get the best value by comparing quotes and provide more opportunities for skilled workers. This report will be a critical analysis of how TradeWork.NET was built and whether or not it has met its initial objectives.

Table of Contents

Acknowledgements	2
Abstract	2
Table of Contents	3
Word Count.....	5
Code Link	5
1. Introduction	6
2. Background, Objectives and Deliverables	7
2.1 Background	7
2.2 Objectives	9
2.3 Deliverables.....	9
3. Method of Approach.....	10
3.1 Risk Assessment	10
3.2 Requirements	11
3.3 Technologies	13
3.4 Best Practices.....	14
3.5 Testing Strategy.....	17
3.6 Proposed Diagrams	18
3.7 Assumptions	21
3.8 UI Mock-up Wireframes	21
3.9 Database Design	22
3.10 Design Decisions	24
4. Legal, Social and Professional Issues.....	31
4.1 Social Issues:	31
4.2 Legal Issues	34
4.3 Professional Issues.....	36
5. Project Management	39
5.1 Trello	39
5.2 Agile	40
6. Stages	41
6.1 Stage 1 – preparation & Planning	41
6.2 Stage 2 – Pipelines and Feedback	41
6.3 Stage 3 – UI Development.....	42

6.4 Stage 4 – Logic and Backend Integration	42
6.5 Stage 5 – User Testing, Finishing Touches, Report	43
6.6 Overview.....	43
7. Output Documentation	44
7.1 Output Diagrams.....	44
7.2 Database Collection Structure	47
7.3 Completed User Stories.....	48
7.4 User Testing Feedback.....	48
8. End-Project Report	52
8.1 Design Adherence	52
8.2 Objective Completion.....	53
8.3 Project Conclusion Development Recommendations.....	54
9. Post-mortem	55
9.1 Technologies	55
9.2 Project Management Approach.....	55
9.3 Personal Performance	55
9.4 User Testing	56
9.5 Disability access	56
9.6 Overall	56
10. Conclusions	57
11. References	58
12. Appendices	60
12.1 User Guide: Installation.....	60
12.2 User Guide: User	61
12.3 User Guide: Trade	74
12.4 Project Management.....	80
12.5 Project Setup Document	102
12.6 UI Mock-Ups and Wireframes	103
12.7 Auth Guard	109
12.8 Blank Questionnaire and Interview.....	110
12.9 User Testing Actual Data:	112
12.10 Subscriptions and Memory Management	114
12.11 Pipelines	115

Word Count

Words – 10947

Code Link

GitHub – <https://github.com/jaswood>

Final Year Frontend – <https://github.com/jaswood/Final-Year-Frontend>

Final Year API – <https://github.com/jaswood/Final-Year-API>

Application – <https://tradesman-application.firebaseioapp.com/login>

1. Introduction

The following report is going to take a critical approach to the evaluation of the deliverables, processes and final product of this project in such a way that it will become apparent if it has been able to satisfy the objective of successfully lowering the barrier to entry of getting DIY or Trade work accomplished. While also the aim of enabling traders to be able to access more opportunities, enabling greater competition between them and therefore, increasing the health of the wider industry. The report will also be looking into how user feedback and trade client feedback has informed the final product, attempting to meet those requirements. It will also recount the challenges encountered within the software development process and ask if both the management used and the techniques exercised were adequate or could be improved.

2. Background, Objectives and Deliverables

Before the beginning of the project, a look at how or whether the project should proceed. This section outlines this stage.

2.1 Background

Research conducted before beginning the project.

2.1.1 Why is This Project Required?

This project is required because of the need to fill the gap that has formed in the easily accessible home DIY or renovation market. Apps that have swallowed this sector in other markets have become household names, Uber, Airbnb, Lyft, etc. All these applications lower the cost of entry to start working in this their respective field, as the field of home improvement was worth £83Bn to the UK economy in 2019 (O'Neill, 2019). These findings not just suggesting largescale changes such as extensions but also fixing disasters, the average UK household spending £589 on average annually (Aviva, 2017). This market, while having existing solutions lacks the killer app that can encompass the full worker – user ecosystem such as the aforementioned revolutionary apps. The time to create a job and have a worker take it up being an essential aspect of engagement, it is this that TradeWork.NET hopes to address along with providing a full solution.

2.1.2 Communication with End-Users

During the project, communication with two different traders who would be using the application if it were eventually released occurred. Prior to beginning the project, they were spoken to about their concerns with an application such as the one that would be produced, stressing areas of importance.

- Quoting needs to be an estimate
- Need to be able to revise that estimate
- Need to be able to say no
- Payment through the app is not as important

2.1.3 Market Research

Research into the market and of similar apps was conducted to gain an understanding of the features that would be required, as well as this, what were the pitfalls of the other related subject matter solutions that already existed (Fig 2.1). While also looking at applications that mimic what the project is attempting in a different market, i.e. Uber. Using “Takeaways” from these solutions when going into defining the objectives and requirements of the project.

Uber and Airbnb

These apps leverage the power of user review scores to inform a way of managing ethical conduct and a high level of standard on the application. If a user or worker does not have a good review score, it does not incentivise other respective users or

workers to interact with them. The self-managing user base shown in Airbnb also serves to provide informed reviews for future users (Airbnb, 2020)

Uber mobile also entirely relies on a home screen of a local map that displays the drivers in your area, it is this feature that has informed the design choices made going forward with TradeWork.NET (Uber United Kingdom, 2020)

Site	Link	Application Type	Features	Takeaways
Check a Trade	https://www.checkatrade.com	Website	Search for Postcode Search for Trade Contact Numbers Review Score Work Showcase Trader Page Shortlist	No Login Contact Numbers
Rated People	https://www.ratedpeople.com	Website	Type of Job Type of Tradesperson Job Urgency Complexity Survey Budget App Create a job My Jobs My Jobs Tradespeople Details	Complexity Survey
My Job Quote	https://www.myjobquote.co.uk	Website	Postcode Search Categories Categories Ask a Trade Forum Familiar Job Creation	Categories *ATF is good but not relevant
My Builder	https://www.mybuilder.com	Website	Advice Centre	
Trust A Trader	https://www.trustatrader.com/	App	Alerts	Alerts
Fixington	https://fixington.com/	Website	Callback Requests Leave a Review Online Quotes visit time booker Briefing on how it works	Visit Time

Fig 2.1: Market Research of Similar Solutions

2.1.4 Project Setup

A project setup document was used to fully outline the overall objective and convey some of the key ideas that will give the app its unique selling point (USP). It also describes the first look at some potential risks, how the solution will be architected and the problems that the project aims to solve with the application. Document found at (Appendix 12.5)

2.2 Objectives

Due to this pre-analysis and project setup (appendix 12.5), the following objectives were defined at the start of the project.

1. Build a PWA that works for both mobile and web
2. Have it deployable via pipeline to demonstrate DevOps Engineering
3. Allow users and traders to review each other
4. Use an interactive map to interact with jobs and companies
5. Allow users to experience a full solution that encompasses the journey of having trade work undertaken.
 - a. Create a Job
 - b. Communicate with traders
 - c. Choose the best trader
 - d. Be able to complete the work solely through using the application

2.3 Deliverables

This section will outline the deliverables of the project; these consist of the overall deliverable and smaller internal to the project deliverables.

2.3.1 Internal Deliverables

Planning Deliverable – Before beginning the project, producing the plans and proposals.

Wireframe UI Deliverable – Creating wireframes to plan and understand the UI/UX before implementation with the hopes of also getting valuable feedback (appendix 12.6)

Environment Deliverable – Creating the environments and pipelines for developing the product before continuing with development.

2.3.2 Overall Deliverable

The delivery of the product (Appendix 12.2/12.3).

3. Method of Approach

This section shall outline measures taken in project start-up and the construction of the project framework. Some of these proved to require changing during the project execution and will be addressed later.

3.1 Risk Assessment

A risk assessment was conducted before application development began to allow for

Risk	Risk Reduction Technique
Scope Creep	Adhere to the pre-determined MVP religiously until completion of features before attempting anything else
Feature Creep	Similar to above but for a specific feature implementation
Project Overrun/insufficient Time Management	Make sure programming is picked up within the first week of march to ensure enough time having put down planning. Complete MVP first
Poor UI Design	Create Wireframes to plan in beginning and consult peers for feedback on the effectiveness of the UI
Poor Code Quality	Conform to best practice learned on placement and the proposed structure of the software. Unit tests and integration tests
Mismanagement <u>In General</u>	Be consistent with Trello board submissions and attend every supervisor meeting
Hope Creep	Be realistic in how far you have gone and how much time you have left to continue to work and the likelihood of its completion
Poor Product Quality	Ensure that when a feature is released you test it thoroughly and have peers/end users test and provide feedback on the feature/product

the recognition and avoidance of these pitfalls while the project continued (Fig 3.1).

Fig 3.1: Risk Assessment

3.2 Requirements

Outlining the requirements for the project has been completed with them as user stories with acceptance criteria, there is a general backlog, and then those required to meet minimum viable product (MVP) are marked as so.

3.2.1 All Requirements

A breakdown of all requirements and their acceptance criteria can be found in the appendix (appendix 12.4), below is an extract of some stories with their acceptance

Title	As a	I Want to be able to	So that I can
Login	User & Tradesman	Log in to my own account	Have tailored content
Home Map	User & Tradesman	View a map of my local Tradesman	View potential Traders/Competition
Create Account	User & Tradesman	Create an account	have a trade or user account on the site
Create Job	User	Post an image to request a job	Have an initial quote and create a job
Create Job	User	Create a job	Have a tradesman come and look at the job and provide an initial quote
Job History	User	View my previous jobs	Do accounts, review a history of the work done
Pay Traders	User	Pay a tradesman	Make sure the work is rewarded
Tip Traders	User	Tip a tradesman	Ensure that work to a high standard is encouraged

criteria removed as examples (Fig 3.2).

Fig 3.2: Requirements Example with Acceptance Criteria removed

3.2.2 Minimum Viable Product Requirements

MVP is a crucial element to any project and is also a critical factor in why projects can fail to delivery; and it is a considerable contributor to risk (Nowak, 2019). From the beginning of the project, which of the project's user stories were part of the MVP were outlined in an avoidance strategy. These can also be seen in the appendix (appendix 12.4) and below is an extract (Fig 3.3).

Primarily all the avenues for payment have been listed as not MVP, there are other stories but mainly these. Declared not MVP because the objectives of the app do not necessarily require payment operation through the app, just to create a more accessible way for traders to interact with customers and encompass these interactions. Payment could be made using many ways externally; it would be the primary form of monetisation of the app, but the objective is not to make it commercially viable.

Acceptance Criteria		
When	Then	MVP
The login details are fulfilled	Authenticate account and provide result	Y
The user is a tradesman or user	The map should populate with either jobs or tradesman respectively	Y
They have no account	Provide them the opportunity to create an account and log in	Y
creating the post allow for the photo to be uploaded	Allow the tradesman to view it and give a quote	Y
creating the post	Allow the tradesman to view it and give a quote	Y
my jobs tab is selected	Show the history of jobs page	Y
the job is completed on the application	Pay the tradesman	N
the job is completed on the application	Provide the opportunity to tip the tradesman	N

Fig 3.3: MVP Analysis

3.3 Technologies

A list of all the technologies used in the building of the TradeWork.NET Project.

Frontend -

- Angular CLI (Version 9) - <https://angular.io/>
- Angular Material and CDK - <https://material.angular.io/>
- Typescript - <https://www.typescriptlang.org/>
- Firebase Hosting - <https://firebase.google.com/docs/hosting>
- Firebase Authentication - <https://firebase.google.com/docs/auth>
- Firebase Storage (Blob Storage) -
<https://firebase.google.com/docs/storage/web/start>
- Cloud Firestore (database) - <https://firebase.google.com/docs/firestore>
- Angular PWA - <https://angular.io/guide/service-worker-getting-started>
- Angular Google Maps -
<https://github.com/angular/components/blob/master/src/google-maps/README.md>
- GitHub - <https://github.com/>
- Firebase Cloud Functions - <https://firebase.google.com/docs/functions>
- Postcode.IO - <https://postcodes.io>

Backend -

- Docker - <https://www.docker.com/>
- NodeJS - <https://nodejs.org/en/>
- Kubernetes - <https://kubernetes.io/>
- Express - <https://expressjs.com/>
- GitHub - <https://github.com/>

Deployment Pipeline -

- Circle CI - <https://circleci.com/>
- GitHub - <https://github.com/>
- Firebase Hosting - <https://firebase.google.com/docs/hosting>

3.4 Best Practices

An outline of the best practices that have been attempted to be adhered to throughout development.

3.4.1 Code Structure

A breakdown of how the code in the frontend environment is structured.

>functions>src> – Contains all the code to run the cloud functions
>src> environment – Contains all environment files
>src>assets> – Contain all the assets for the application.
>src>app> – Contains files to run frontend, route the application and load all modules
>src>app>models – Contain all the class models
>src>app>enums – Contains all the enums for the app
>src>app>testing – Contain all the mock files for tests
>src>app>pipes – Contains the pipe files for use in html filtering
>src>app>services – Contains shared services for HTTP and logic, not on components
>src>app>component – Contains all the files for web pages (CSS, test files, html, logic files)
>src>app>components>services – Contains all the component specific services that are not shared
.circleci – Contains CI pipeline code for frontend
> – contains installation files

3.4.2 Pull Requests

Pull requests would be performed on every feature branch; during this process, this would review every file change submitted to ensure that it was as expected and remove glaring logic/syntax/semantic/style errors.

Further to this, each pull request adheres to the same naming convention to allow for easy backtracking (Fig 3.4). Also preserving the commit history for easy backtracking, this means it is easy to tell apart pull requests and regular commits.

```
feat(security): added an auth guard preventing users accessing routes... ...
jaswood committed 7 days ago ✓

fix unit tests
jaswood committed 7 days ago

added an auth gaurd to the app
jaswood committed 7 days ago

fix(mobile): restyled several screens for smaller mobile again ...
jaswood committed 7 days ago ✓

restyled several screens for smaller mobile again
jaswood committed 7 days ago

feat(mobile): Added mobile css for all pages ...
jaswood committed 7 days ago ✓
```

Fig 3.4: Pull Request Naming

3.4.3 Git Techniques

The git structure used was to have one main master branch and then have feature branches off this. Then, on merging, preserve the commit history to enable easy backtracking and rollback, should the deployment encounter a problem (Fig 3.5). The only times this was not adhered to was when pipeline configuration changes needed to be made (such as late in the project with cloud functions), pipeline builds only trigger on master branch commits so creating a pull request for each of these is inefficient.

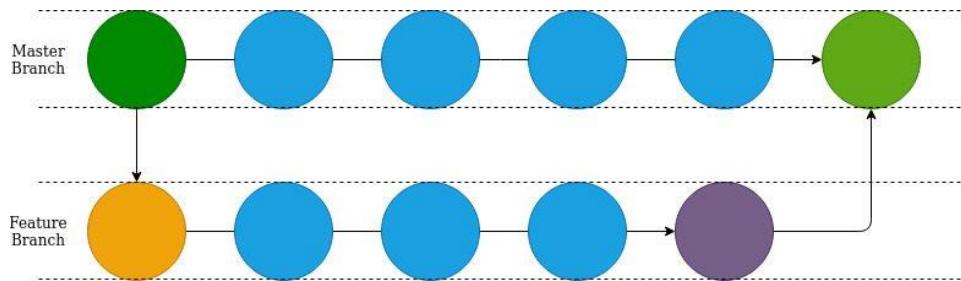


Fig 3.5: Git Strategy

3.4.4 Continuous Integration and Continuous Delivery

TradeWork.NET uses Circle CI (Circle Ci, 2020) to have unit tests, integration tests and builds checked before deploying to the project's firebase hosting platform. Builds are picked up after each commit to the master branch and then the workflow is executed in circle ci, there are pipelines for both the frontend and the backend; however, the backend was made redundant as the project progressed. Below is an example of the workflow for the frontend pipeline (Fig 3.6) and more details on how the pipelines were implemented can be found in the appendix (appendix 12.11).

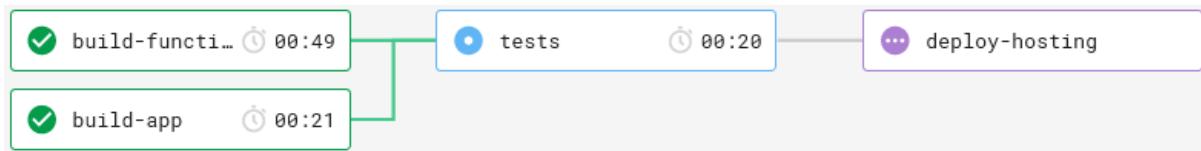


Fig 3.6: Frontend CI/CD pipeline

Componentisation and Enums:

The use of componentisation is extensive throughout the code base of TradeWork.NET, for example, the quoting component (appendix 12.2/3) is used in both the "Job Details" and the "Home Map". As well as this, there are a series of enums that have all been employed wherever possible. These used to reduce the amount of code duplication and potential errors. Both of these are allowing a more strict application of the SOLID design principles (What are the SOLID design principles?, 2019). In the GitHub repository, there are many further examples of this. The angular nine feature ng-template (Angular – Structural Derivatives, 2020) along with many other structural derivatives have also been used to allow for reuse in HTML too.

3.5 Testing Strategy

Testing strategies used while building and deploying the TradeWork.NET application.

3.5.1 Unit and Behavioural Tests

The Application has two kinds of test that run builds in the pipeline and by the developer before a pull request is submitted. There are (at time of writing) 81 unit tests and behavioural tests that are designed to cover the logic of the application and ensure that each component that the app uses can still load correctly should there be changes made to the app. These must pass before a pull request can be enacted (as a personal standard) and are forced in the pipeline, preventing the release of logically and behaviourally broken code.

3.5.2 Protractor Tests

Angular protractor tests (Protractor, 2020) also run in the pipeline, these ensuring that all of the HTML views load correctly before releasing.

3.5.2 Mandatory Integration Tests

After key features were completed (Home Map, Create Account, My Page, etc...) a checklist of the previous functionality was tested to ensure that no added functionality had inadvertently broken existing functionality and not been detected by other tests. The checklist would be updated with the most recent completion to date (Fig 3.7)

Feature	Functionality	Pass/Fail	Date
Home Map	Create Job	Y	04/05/20
	View Company	Y	04/05/20
	View Job	Y	04/05/20
	Open Chat	Y	04/05/20
	Create Quote	Y	04/05/20
	Accept Job	Y	04/05/20
My Jobs	Lists	Y	04/05/20
	View Job Details	Y	04/05/20
	Create Chat	Y	04/05/20
	View Company	Y	04/05/20
	Job Statuses	Y	04/05/20
	Create Quote	Y	04/05/20
Job History	Lists	Y	04/05/20
	Open Details	Y	04/05/20
Chats	List Chats	Y	04/05/20
	Open Chat	Y	04/05/20
	Chat with User	Y	04/05/20
	Chat with trader	Y	04/05/20
My Page	Edit Details	Y	04/05/20
	Edit Company Details	Y	04/05/20
	Upload Profile Image	Y	04/05/20
	View Gallery	Y	04/05/20

Fig 3.7: Mandatory testing document

3.6 Proposed Diagrams

Before beginning the project, some preliminary designs were made to have an understanding of the flow of the application and how the architecture would take shape.

3.6.1 Architecture Diagram

Proposed architecture –

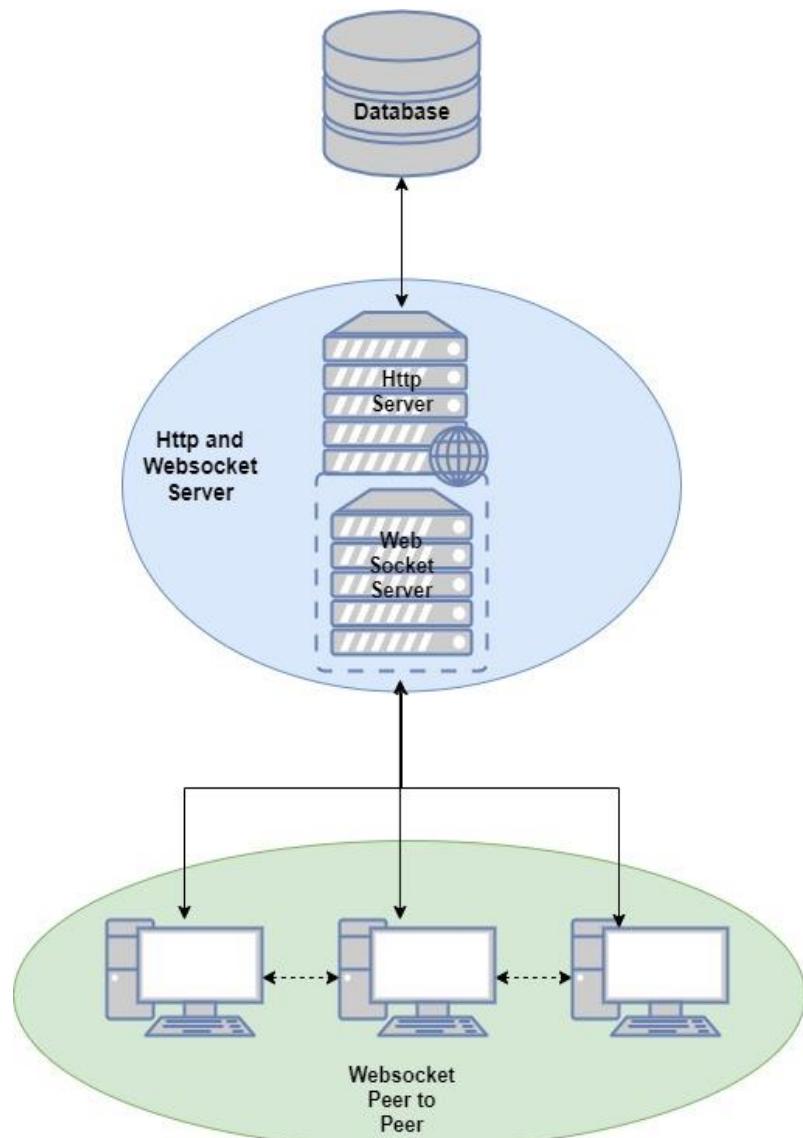


Fig 3.8: Proposed Architecture

3.6.2 Sequence UML Diagram

Proposed Sequence Structure –

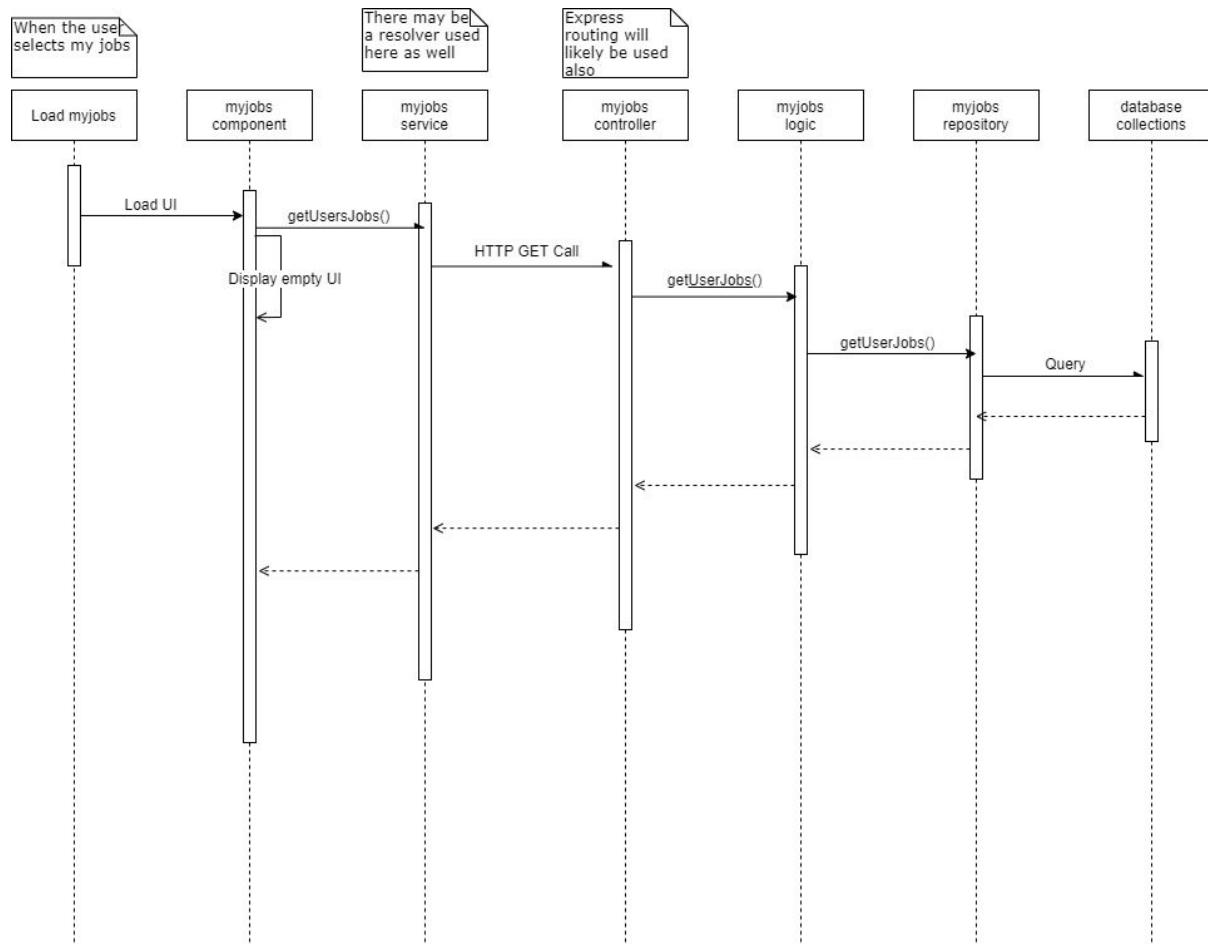


Fig 3.9: Proposed Sequence UML

3.6.3 Activity UML Diagram

Proposed application flow for a user –

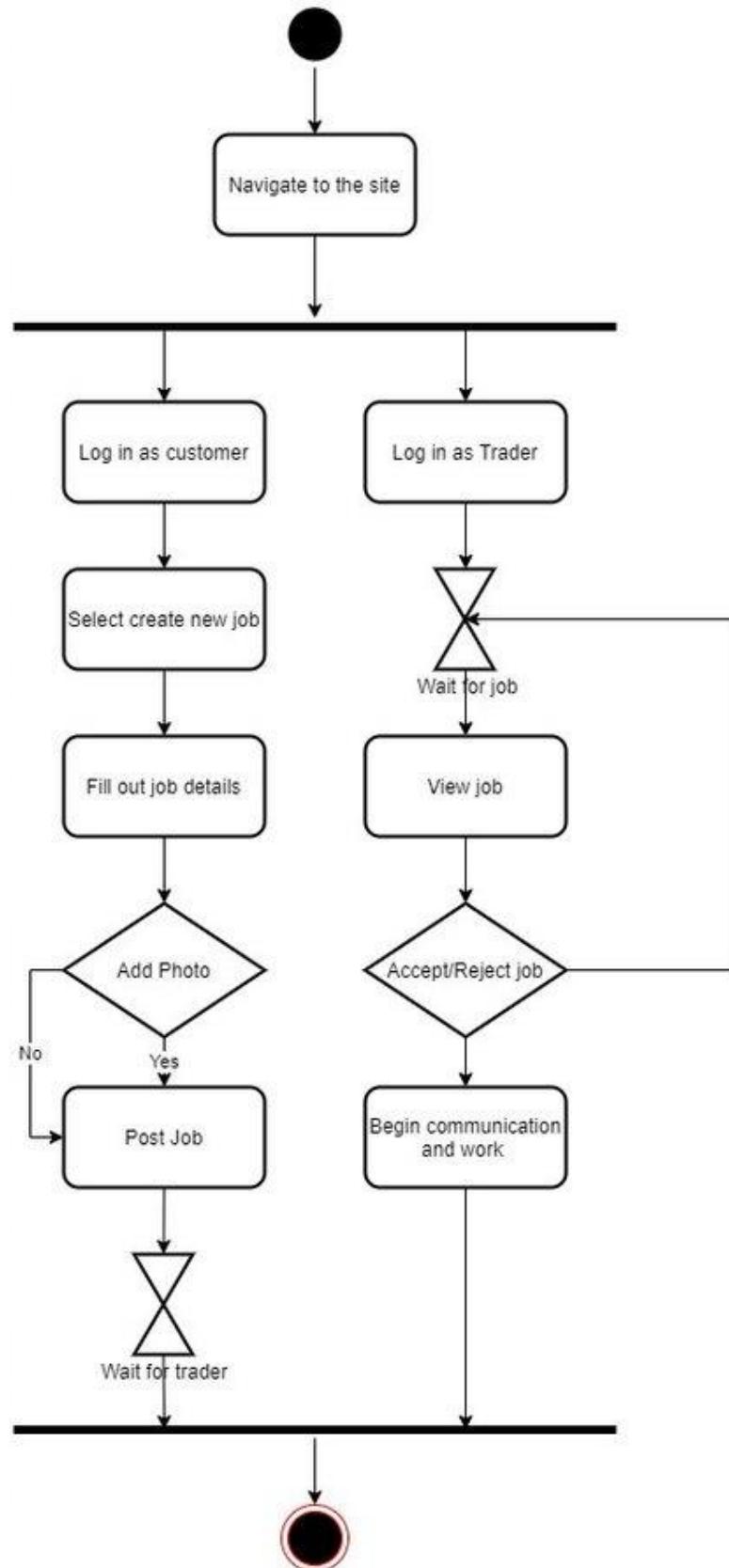


Fig 3.10: Proposed Activity Diagram

3.7 Assumptions

There are several assumptions made about users while they are using the application,

- A User is not also a trader – If this were the case, they would need two accounts
- A Trader is not also a user – If this were the case, they would need two accounts
- A Trader will only want to make one quote – This quote is revisable throughout the acceptance process
- A Traders Account will have shared information with their company on account creation.
- All types of user have a valid UK postcode.
- All users will communicate with each other to share very personal information regarding the job, for example, the exact location of the job issuers home.
- A job post will be concluded should it enter the “active” state.

3.8 UI Mock-up Wireframes

A design mock-up (wireframes) was completed to get an initial understanding of how the UI would look before the beginning of its construction; also used to gain early feedback on how the app would work and how the UI could be improved before committing to building any of it. A full catalogue of the wireframes is available in the appendix (appendix 12.6).

3.9 Database Design

A breakdown of why the project uses a NoSQL database instead of a traditional relational database model (RDBM) and the proposed entities before beginning development.

3.9.1 NoSQL vs Relational Database

Prior to the beginning of the project, both types of database were researched before deciding on which was best for the application, the critical reasons used to justify why a NoSQL database was chosen were as follows,

1. Scalability – The NoSQL database scales automatically with usage, whereas an RDBM requires the allocation of more resources (typically).
2. Requirements – If the scope of the project/requirements of the project is not clearly defined at the outset of the project, then RDBMs rigidity becomes challenging to work with. Compare that with the flexibility to change the structure of NoSQL Collections at any time.
3. Development Efficiency – As a single person dev team, the ability to make a change in the code and it immediately be applied to the data structure of the database is very valuable. An RDBM requires a database release alongside the code release or must be maintained in the database IDE if not using database as code practices.
4. Preparation – A NoSQL database requires you to initialise it, then it is up and running. It can begin being used; an RDBM requires setting up initial schemas, deployment options and hosting.
5. Data Consistency – To make schema changes in an RDBM, it is often required that you delete the contents of the table, adding more work to replenish the data and fix affected tables. NoSQL collections do not have this crux as they can be adapted to the new data model.

The above is not a discussion of why an RDBM is worse than a NoSQL database. It is a discussion as to why the use of RDBM did not suit the needs of the TradeWork.NET project.

3.9.2 Proposed NoSQL Collections

A proposed plan for the collections required in the NoSQL database for TradeWork.NET (Fig 3.11). These also then became the initial class designs for the frontend typescript project.

User:	trader:	Job:	Chat:	Completion State:
first name	first name	Quote	trader	job
last name	last name	location	user	state
dob	company name	description	chat history	
email	location	picture		
location	jobs	time frame		
jobs	email	trade		
review score	contact details	Completion state		
picture	review score	Issue Date		
chats	description	Conclusion Date		
	picture/s			
	trade type			
	chats			
generic user:	Company/trade:			
first name	company name			
last name	contact details			
dob	description			
email	pictures			
jobs	trade type			
review score				
location				
picture				
chats				
Trade Association?				

Fig 3.11: Initial Collection Designs

The user and trader collections (Fig 3.11) are a proposal of having two collections for users, beneath them the generic user and a company/trade collection is the alternative approach.

3.10 Design Decisions

Several software development decisions and issues occurred during the development of the project, and this section will outline some of these critical decisions, namely the decision not to have a node server and rely on Backend-as-a-Service (BaaS), but also other choices.

3.10.1 Initial Software Design Choices

A brief outline of some design decisions and why.

1. Why a Progressive Web App (PWA)? – a PWA (Progressive Web Apps, 2020) was chosen instead of a Cross-Platform-Mobile-App (CPMA) such as Ionic (Ionic, 2020) or Flutter (Flutter, 2020) due to a cross-platform-mobile-app having platform independence. This means a CPMA will not be able to access features of new updates or indeed all of a platform's available features, due to their abstraction until they are updated, if at all. Framework tools directly inhibit what you can and cannot do and with them compiling down to native language resulting in slower code due to lack of optimisation. Whereas the PWA means updates percolate straight away, a PWA can be used like an app and the product is not restricted to a Framework, particularly important when using the google maps API so extensively. Also using native was avoided because then it would only be deployable/usable in one ecosystem.
2. Why Angular 9? – It has built-in PWA (Progressive Web Apps, 2020) support and subsequently also already has native service workers for offline usage to make future use of with unstable connections better. That could be a problem in areas where users of the application would work. Fully supports googles lighthouse testing standard and Angular 9 (Angular CLI, 2020) is from the google stack and so pairs with the rest of the project well. It is very mature with a wealth of documentation. Finally, I am comfortable with it.
3. Why NodeJS as an API? – It maintains the same language stack, JavaScript, or Typescript. It is easily integrated with firebase and Node (Node.js, 2020) comes with fewer performance overheads than solutions such as dotnet core (.NET, 2020). The final product does not use a Node server, and this will be discussed later in the report.
4. Why Firebase? – Firebase being used for the project instead of an alternative such as Azure or AWS was quite an easy decisions. Both AWS and Azure are incredibly heavyweight with a high degree of complexity from the outset, firebase (Firebase, 2020) is incredibly easy to use and integrate with CI and CD, further to this it is in the same tech stack as Angular 9 (Angular CLI, 2020). It also has the ability to be a single solution to the projects backend instead of having a hosted in firebase app and a database in mongo (MongoDB, 2020).

3.10.2 Backend-As-A-Services or NodeJS API

Mid project the decision was made to make the switch from a traditional frontend, backend, database application in the form of a Node API backend, to a frontend, BaaS stack; this in the form of firebase BaaS; this meant a large proportion of planning, Stage 1, and Stage 2 work was no longer valid. Changing helped the overall product remain lightweight, responsive, and extensible and, will continue to do so as it scales. Raising the question, “Why did the project deviate from the original plan in the first place?” and also promotes a wider question of “Is there any need for a backend server in the current software development climate?”.

Why did the project switch from a Node API to Firebase BaaS –

While developing the original plans and environment preparations in Stages 1, 2 and beginnings of 3 planned and depicted the use of a Node API. However, The project made use of the Google Stack; Firebase Auth, Cloud Firestore NoSQL and Cloud Storage making there many situations where adhering to best practices, and the most efficient way, would mean to forgo the API; the login authentication for example. As development went on, the ability to listen to documents from the NoSQL database in a WebSocket manner would greatly improve user experience.

Implementing that functionality going through a middleman node server would have required extensive use of WebSocket's on every endpoint while still listening to the endpoints on the NoSQL provides. It would have required huge amounts of effort that was available on startup with BaaS.

With the understanding that the difficulty and code requirements would increase dramatically if the project were to maintain the path of having an API made the decision to step away from an API clear. Further, coupled with personal research of Cloud Functions in Firestore showing that logic could be abstracted off of a clients machine, led to the retirement of the Node backend in favour of BaaS. Allowing for the projects entire backend and frontend to be the same tech stack, which, while also has its problems, does mean everything works together intuitively.

Switching to frontend and BaaS meant that a pipeline and hosting environment could also be retired that allowed for cost-saving and further complexity reduction; an example of this is in the initially proposed sequence diagrams paired with the final sequence diagram (Fig 3.8 and x respectively).

Node Server vs Backend-as-a-Service –

For this section of the report, there will be a comparison between the BaaS implementation used with Firebase/store and traditional example of a node server using express, internal logic and must call to a database. This example node server being used because that is the environment that was initially architected for the solution.

Feature	BaaS – Firebase Application	Node Server Application
Deployment	Deployed with the frontend application and in its pipeline.	Deployed as its own application requiring its own build and release pipeline
Logic	Can offload logic that should not be on the client to cloud functions, can also listen to the database and trigger on CRUD interactions.	Can perform logic on the server and then post to the database. Third-party libraries could be used to listen to databases
Efficiency	From the front end to the database and back	From the frontend HTTP request to the node server, from the node server to the database and back
Complexity (In this project's implementation)	One language, one code base, one set of documentation, one deployment pipeline, one hosting solution	One language, Two codebases, Two sets of documentation, Two deployment pipelines, Two hosting solutions
Cost	One application + database + cloud functions	Two applications + database
Time	More effective as one codebase	Less effective as developing a feature requires both front and backends at the time
Reuse	Cloud functions can be reused, any frontend logic/interactions with it must be duplicated.	The entire app can be consumed infinitely (resource permitting)
Ethics	Tied into a company and its platform	Free to host implementation wherever

Fig 3.12: Comparison of BaaS to a Node Server

The comparison (Fig 3.12) suggests there are a lot of pros for a BaaS solution; however, it is an important point to stress that this is based on the implementation of a single frontend app. If it were considering applications that had many different frontend implementations and had many bespoke uses, then the ability for every frontend to reuse the same backend would be indisputable. That said, for a mobile app or relatively low complexity project, the flexibility, versatility, and further reduction in complexity offered from using a BaaS solution cannot be overlooked.

3.10.3 Fluid Display Layouts

Due to the use of a PWA and the knowledge that a lot of users do not necessarily have a traditional desktop or laptop computer, which has a large screen (Fig 3.13). TradeWork.NET is built to take advantage of smaller displays by restructuring the user interface to accommodate the space.

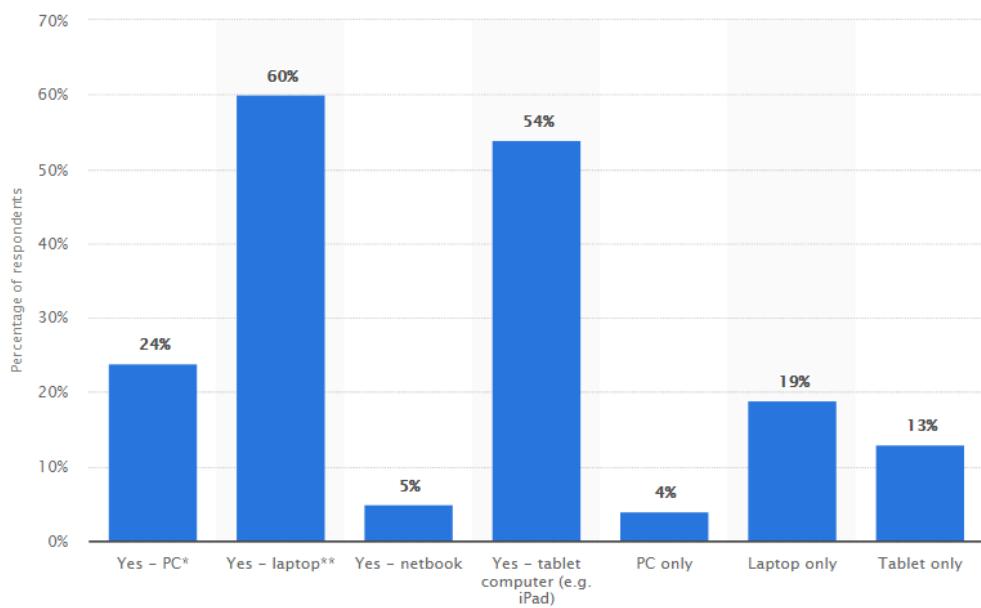


Fig 3.13: Statistic for the share of households with a computing device in the UK in 2019 with 2,675 respondents at 16 years or older (Statista Research Department, 2020)

A full breakdown of the different layouts of the user interface can be seen in the appendix (appendix 12.2/3). Changes to UI are the most significant aspects of the restructure, however in some area certain information is hidden in the HTML DOM when viewing on a smaller screen, where screen real estate is more valuable. Below is a list of pages where there are adaptive displays along with a brief description of the change for smaller screens (Fig 3.14).

Page	Mobile Display	Desktop/Tablet Display
Navigation Pane	Changes to horizontal navbar and removes hover pop-out	Navbar is vertical and has hover pop-out
Home Map	Post new job component converted to column and button moved to the top right of the screen	Post new job component wider and pin moved to bottom right
My Jobs Lists	Less Information in job headers	Full information in job headers
Job Details	Lays out all the detail in a vertical column. Hides some information from status at the top	Lays out all the detail in a horizontal row, Shows full information in status at the top
Job History	Same as Job details	Same as job details
My Page	Data is displayed in vertical column	Data is displayed in a horizontal row
Chats List	Less critical information in the chat header hidden	Full detail displayed in the chat header

Fig 3.14: Table of mobile and desktop displays relative changes

An example of these changes is below (Fig 3.15).

Job: Fix Our Electric Cookers FAST Location: PL4 6JH My Review Score: 4★
 Chat Accept Delete

Job Info
Job Image
Select Quote
Wild Wild Builders - £500
Inspect Company

Job Details > Fix Our Electric Cookers FAST

The job has quotes for review

Job: Fix Our Electric Cookers FAST
Location: PL4 6JH
My Review Score: 4★ Choose a quote to enable buttons

Chat
 Accept
 Delete

Job Info

Issue date:
Apr 29, 2020

Time frame:
1 Days

Budget:
£300

Trade type:
Electrician

Description:

Fix our electric cookers x 4, It is key to our business so must be done quickly

Fig 3.15: Example of mobile and desktop/tablet views

4. Legal, Social and Professional Issues

During the planning and development of the project, several issues of the above-described nature arose, this section will cover those issues and how they were resolved or further steps that should be taken.

4.1 Social Issues:

Going over the main social problems that occurred during the development of the project.

4.1.1 Job Completion Statuses

A challenge that occurred was determining who had the overriding authority when referring to the status of a job, should it belong to the trader or should it belong to the user, there were several decisions on how the job status interactions would be interacted with. What follows is a list of the status and their interaction logic.

1. Available – A job is posted up by a user and has had no interaction with it from a trader. In this state, it may be viewed and will remain available.
2. Quoted – A job has been provided with a quote, this will then update the status of the job posting from “Available” to “Quoted”. It will also now appear in the traders “Quoted Jobs” tab of “My Jobs” (Fig 4.1).

The image shows two screenshots of a mobile application interface. The top screenshot is a map view of a residential area with a yellow lightbulb icon indicating a job location. Overlaid on the map are details for a job: **Budget: 300**, **Review Score: 4**, and **Timeframe: 1 Days**. Below the map are three buttons: **Chat**, **Quote** (with a £ symbol), and **Accept** (with a checkmark). The bottom screenshot shows the 'My Jobs' page. The header has a blue bar with the title 'My Jobs'. Below the header is a navigation menu with icons for Home, Map, Profile, and Logout. The main content area has tabs for 'Active Jobs' and 'Quoted Jobs', with 'Quoted Jobs' currently selected. A search bar allows searching by job title. Below the search bar is a list of quoted jobs, with the first item being: **Fix Our Electric Cookers FAST**. The job details include: **Location: PL4 6JH**, **Completion State: Quoted**, and **Issue Date: Apr 29, 2020**.

Fig (4.1): Quoting from “Home Map” page

3. Trade Accepted – If a trader has accepted the job without a quote, then a quote will be made for the budget outlined by the job issuer, if a trader has provided a quote then the trader will accept the job for the amount they have quoted. The status will then update to “Trader Accepted”. A trader can do no more with the job status; the authority lies with the user.
4. Accepted – A trader will only successfully get a job if the user also agrees on them for the job; at this point, a user and trader have both accepted each other. The status will then update to “Active” and the job removed from the Home Map, while also removing all other traders from the job regardless of their level of job status.
5. Closed – A job can only be closed by a user, closing does not have to be agreed upon by the trader, once a job posting is closed it is removed from both the users and the traders “My Jobs” page and moved to the job history page. The option to review each users opposite is then presented.

4.1.2 Deleting a Job

Deleting a job is a social issue because of the implications it holds, a user cannot always delete a job posting due to the fact a trader may be using it to keep track of the situation. Furthermore, it could also be exploited to remove the record of the job from the trader; it is because of this that once a job has begun (become “Active”), it can no longer be deleted. Up until that point, a user can remove any job posting (Fig 4.2). Once a job has been deleted, it removes all information of itself from the database, and all traders will lose all associated data. This deletion is done to protect the personal information of the user.



Fig 4.2: A user’s quick actions from the “my jobs” page

4.1.3 Rejecting a Job

A trader can reject a job up until the job has been accepted by both trader and user (Fig 4.3). Once a trader refuses a job, it removes the quotes and that traders “Accepted”, should they have accepted the position prematurely. A trader is not able to reject a job after this as it could be exploited much like deleting the job.



Fig 4.3: A trader’s quick actions from the “my jobs” page

4.1.4 Cancelling an Active Job

At the moment, it is assumed that communication would happen between the trader and the user, and the job would be concluded. However, this is an assumption, and the app currently does not account for if this exterior or internal communication doesn't occur. A Post MVP feature would put forward the option to have both a user and trader agree a job should return to the "available" state or be deleted safely.

4.1.5 Review Scores Management

Utilising all the users to provide each other with review scores as a form of self-quality assurance on the platform creates its own issues, a review can only be conducted once the job has been concluded. Further to this, a review score cannot currently be changed, presenting the issue of human error, or "heat of the moment" decisions affecting a trader/user's rating. There is no solution to this currently implemented; however, the current review score is an average of all the reviews made about a user/trader, serving to eliminate the anomalies created by this problem. The option to edit a review could be made into a post MVP feature.

4.2 Legal Issues

4.2.1 GDPR and Data Protection

Both GDPR (General Data Protection Regulation (GDPR) – Official Legal Text, 2018) and the Data Protection Act (Data Protection Act 2018, 2018) are legal, social and ethical issues that need to be thought about adhered to when designing an application. Despite GDPR largely superseding the Data Protection Act, due to Brexit, Data Protection has also been considered in this report.

GDPR – The product tries to conform to GDPR rules (General Data Protection Regulation (GDPR) – Official Legal Text, 2018) by only taking the data that the user has provided and nothing more (such as scraping metadata). As well as this the app allows you to see all of the personal information held for a user on the “My Page” (Appendix 12.2/3 – “My Page”); currently, there is no way to delete accounts or data held by the application. Deletion would need to be implemented for the applications to become in full accordance with the law and proceed commercially. However, this was not within the scope of the project, as this is for educational purposes.

Data Protection Act – The Data Protection Act (Data Protection Act 2018, 2018) has been adhered to concerning the prevention of personal data being exposed to other users on the application and malicious agents. Several design decisions were made to account for this despite them not being in accordance with how the client for the project or user would expect the app to function. Examples of this would be, the prevention of a user requesting any other users personal account information, a trader being exposed only through the identity of their company which, therefore, abstracts the data away from being viewed. Chats on the app are a combination of abstractions since you can have many conversations with users/traders before the work is accepted and began, which would then allow you to build a personal relationship with a trader. Therefore, Chats use the job title for the users' name and the company name for the traders' name, protecting this information. Among other measures, the final to mention is that the “Home Map” (Appendix 12.2/3 – Home Map) that displays a job/company, does not show the exact location of a job, only a general area. Once the job has been accepted, the two can communicate for accurate information.

Going forward if the project were to continue much greater control of a user' personal data would need to be built into the system to satisfy GDPR; Deleting accounts, deleting jobs, deleting chats and exporting held data to view.

4.2.2 Protection of Personal Information on TradeWork.NET

Protection of data in TradeWork.NET is something that has been a key theme throughout the software design. Several measures are in place to prevent unauthorised access to data.

Hosted Data – The app data is hosted in using Firebase Hosting (Firebase, 2020) and their Cloud Firestore storage solution. This approach uses NoSQL with automatic 256-bit encryption before the data is written to the disk and can only be read when decrypted by an authorised user. It is also transferred over https, and the master keys of the encryption are also encrypted and rotated in a pool of master keys (Google Cloud, 2020).

Authorisation Data – Authentication is handled by Firebase (Firebase, 2020) and their Firebase Auth solution, in which the data is hashed and salted (Fig 4.4)

```
hash_config {  
    algorithm: SCRYPT,  
    base64_signer_key:  
    [REDACTED]  
    base64_salt_separator: [REDACTED]  
    rounds: 8,  
    mem_cost: 14,  
}
```

Fig 4.4: Auth security config properties

4.3 Professional Issues

Professional issues such as maintaining the ethics and conducts of the Association of Computing Machinery (ACM, 2020) and the British Computer Society (BCS, 2020).

The work on TradeWork.NET and the project are following BCS and ACM ethics. The project is open-source, available on GitHub, and the application can be used by anyone should they wish to use it. Furthermore, best practices have been attempted to be adhered to throughout and, as the project has continued, a more extensive array of them come into use as more best practices have become apparent through development. Following the BCS and ACM ethics of upholding and adhering to standards set across the industry as a whole.

The application also requires an age of over eighteen to register for using it. Due to the nature of the app, not that the content is mature, just the fact that a minor should not be able to be contracted to do work or potentially be contacted unannounced by unknown people/companies. There is an argument for sixteen plus in the UK due to the legal working age being sixteen (Child employment, 2020), but a company should employ these workers.

4.3.1 API Usage

The application makes use of several APIs. Firebase and cloud functions are subject to usage limitations to the number of calls; this would have implications such as preventing data being collected from the database. However, billing can be enabled, and the plan will automatically roll over into a paid plan and start working again immediately in a pay as you go manner.

Postcode.IO is used for the translating of postcodes into longitude and latitude to then paint onto google maps, and this would also have usage limits; however, the app does not hit those with demonstrative use cases. If this project were to be deployed commercially, then the solution that would be implemented would be to clone to the repo and self-host the service to avoid its limitations.

Finally, Google Maps API is subject to restrictions as well, and there is no self-hosting ability, the only solution to this is pay for usage, for educational purposes, the project does not require this. Google provides an initial amount of free credits that the project uses.

4.3.2 Security

TradeWork.NET includes multiple layers of security, not just those used to address GDPR and Data Protection issues.

Auth Guard – The frontend requires a user to log in to access the app, as well as this, in each route, there is an authorisation check that requires the user to remain signed in to access any of the other pages of the site (Appendix 12.7).

Password Management – Password fields are blocked out, and there is no way for any developer ever to be able to access this information in the database.

Database Rules – There are rules defined on the database that prevents any unauthorised access to perform CRUD actions on the database (Fig 4.5). While this is not an incredibly secure method because it could be spoofed with some work, it is still putting another barrier up. Further rules would be beneficial.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth.uid != null;
    }
  }
}
```

Fig 4.5: Database access rules

4.3.3 Memory Usage

A genuine issue with the development of TradeWork.NET was the occurrence of memory leaks. Due to the retrieval of data from the backend using lots of subscriptions, unless these subscriptions are handled effectively, they will continue to exist and use memory until the tab or browser is closed. A Subscriptions listens for new data from a source (WebSockets in this implementation); this is especially important when using a PWA where a user might not close the app for an extended period. During development, before effectively handling, memory usage was seen upwards of up to 8GB (in an extreme case), this is an unreasonable amount of memory, and much of the usage was the product of the memory leaks caused by these subscriptions.

The issue has been addressed by closing the subscriptions when components no longer require them to be loaded, such as a user leaving the “My Jobs” page and going to the Chats page. Several methods were used to do this, and they can be found in the appendix (appendix 12.10). Current fixed memory usage of the app can

be seen below (Fig 4.6).

Fig 4.6: Typical memory usage of TradeWork.NET



5. Project Management

There were several project management methodologies employed during the project, namely these were the use of Trello (Trello, 2020) for general process management and Agile methods. Working in regular sprints to create tangible deliverables, aiding in this the project was conducted with features being described in the form of user stories as opposed to just a list of features or requirements and an initial sprint plan being laid out at the beginning. Collectively these measures allowed for the effective management of the TradeWork.NET Project.

5.1 Trello

Trello was used continuously throughout the project to maintain up to date lists of what work was still left and to track existing work that was being undertaken. It also served to display blocked stories allowing for the figuring out what was of the highest priority (Fig 5.1). The breakdown of the categories allowing me to discern the differences between the progress of a story quickly, groups of particular interest are the blocked category, which was useful for monitoring things outside of my control such as ethical approval applications. As well as the redundant group, allowing for precise removal of certain technical backlog pieces that were no longer relevant after making the switch from a node backend.

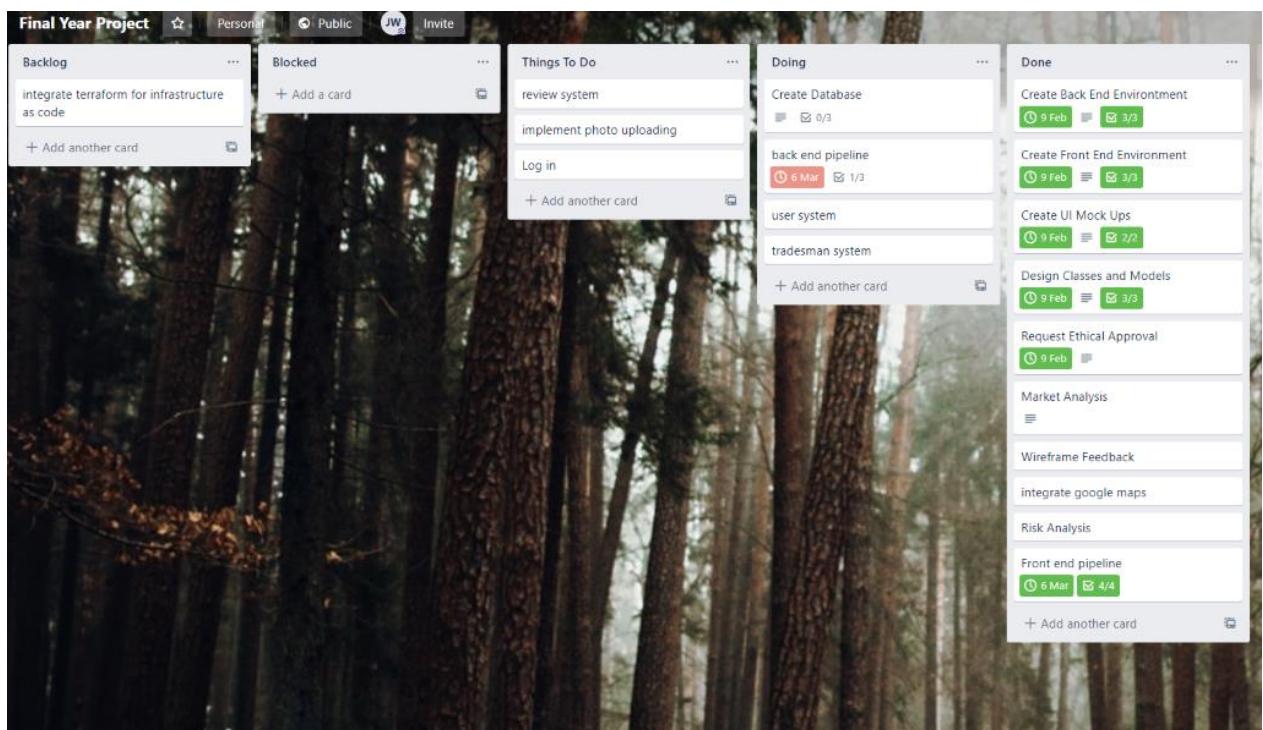


Fig 5.1: Example Trello Board

5.2 Agile

Agile methodologies were used to create manageable goals as the project scaled in complexity with continuous Trello submissions being used to keep track of sprints and the user stories defined (Appendix 12.4 – Trello Board and Sprint Reviews) making the Trello board into a pseudo sprint board alongside generally keeping track of the project. Reviews of how the project was proceeding would be held routinely; these can be seen in the appendix as the sprint reviews (appendix 12.4 – Sprint Reviews).

5.2.1 Sprint Plan

At the start of the project, a general sprint plan was conducted to outline how the project would proceed on a sprint by sprint basis (Appendix 12.4 – Sprint Plan). Performed at the start with an initial framework and then filled in as the project progressed and the needs of each sprint were becoming outlined. The plan also includes a work layover section which shows if work was remaining at the end of the sprint. Each sprint not representing a Trello submission, some are representing two or one.

5.2.2 Sprint Reviews

At the end of each sprint, a sprint review was held, a look at what went wrong, what went well, what could be changed and how I felt about the project (Appendix 12.4 – Trello Board and Sprint Reviews). They also served as to think about how much work could be taken into the next sprint, if there was any layover work remaining.

5.2.3 Release Early and Often

One of the key mantras of agile and scrum (Engel, 2018), through initial sprints, there were tangible documentation deliverables that could be used to measure this. Since sprint three (Appendix 12.4 – Sprint 3) there were concrete deliverables every time a software story was completed as part of a sprint, as soon as the code was merged with master, due to the CD aspect of the project, it was immediately released. Furthermore, quality standards were adhered to due to the testing techniques utilised.

6. Stages

A breakdown of the key steps within the project, this is not necessarily each sprint or Trello submission but a general analysis of specific periods during the project lifecycle designed to provide an understanding of how the project was conducted

6.1 Stage 1 – preparation & Planning

Trello's included – 1, 2, 3

Sprints Included – 1, 2

General Content – Initial product documentation, taking the project from an idea to having a detailed setup analysis. Determining the technology stack that should be used and collecting data about existing solutions. This sprint has no tangible functionality deliverable due to it being a preliminary setup sprint; however, it did deliver documentation and prepare the environments for development to begin, enabling the project to continue smoothly.

Deliverables – Market Analysis, Ethical Approval documentation, Proposed UML Diagrams, Trello Setup, User Stories, Sprint Plan, Creation of Environments, UI Wireframes

Sprint Layovers – Market Analysis was completed despite its layover from sprint one. This was a low impact piece of work, so it did not impact sprint two significantly. The frontend environment layover continued into sprint three due to not adding PWA support, but the environment was built, again a small amount of labour required to complete. The database was still requiring analysis.

6.2 Stage 2 – Pipelines and Feedback

Trello's Included – 4, 5, 6

Sprints Included – 3

General Content – Feedback from the UI development provided as well as a risk assessment being conducted, still prior to development beginning. Frontend and Backend pipelines were constructed, giving tangible deliverables in this sprint as well as the deliverable of more documentation. Complications arose in the development of the backend pipeline due to the hosting software choice (firebase) not having a basic node server hosting option, getting the pipeline to build was also not trivial, consuming much of the sprint and dampening morale considerably.

Deliverables – Risk Analysis, Frontend Pipeline, Feedback on UI

Sprint Layovers – The initial UIs began in this sprint, but that was due to needing to have a break from the backend pipeline. The backend pipeline spilt over due to its complexity, eventually ending up as a dockerised build hosting on the Google Cloud Platform in a Kubernetes cluster. This dramatically over-engineered solution took a lot of time to get to work.

6.3 Stage 3 – UI Development

Trello's Included – 6, 7

Sprints Included – 4

General Content – Finishing the UI designs for the interfaces; this was a simple task; more nuanced CSS became challenging. Integration of google maps into the application to make sure that it would work properly and be set up ready for when the feature development occurred, the maps were also tailored to my implementation, removing places and google earth are two examples of this.

Sprint Layovers – Authentication and create an account using firebase Auth was to be fully set up this sprint, login proved to be a more manageable task but working with the API to gain a better understanding of authentication was essential.

Deliverables – The User Interface, an interactive “Home Map” page with fully integrated google maps, login and Create Account pages (Majority of Auth)

Sprint Layovers – The login and create an account while simple to implement thanks to the API, still created some issues with UX that then, therefore, required fixing and continuing into the next sprint such as getting the data the application needed from third party auth providers.

6.4 Stage 4 – Logic and Backend Integration

Trello's Included – 7, 8

Sprints Included – 5, 6, 7

General Content – During these sprints, the majority of the logic for features was constructed, sprint five was primarily about NoSQL database integration and file uploading, working with the BaaS model. Building a platform for sprints six and seven to be able to develop quickly without having to be concerned with environments.

Deliverables – File Uploading, My Jobs feature, Chat Feature, Job History Feature, My Page Feature, Complete User and Trade Functionality, Home Map backend functionality.

Sprint Layovers – The review system was partially implemented this stage, in that all of the features that had been implemented had support for it. Still, the functionality did not exist yet, and they were built in anticipation of the next sprint. Trying to reflect the release early and often nature of agile, the working features did not wait for a large release or for the review feature to become available as they were ready to go already.

6.5 Stage 5 – User Testing, Finishing Touches, Report

Trello's Included – 9

Sprints Included – 8

General Content – This is sprint is now adding post MVP functionality to the system. The review system was integrated fully and so all of the features that used it started working, Cloud functions were leveraged for this and were a big part of the learnings of this stage, general polish, and a transition of focus from the product to the report.

Deliverables – Review System, Cloud functions, the report

Sprint Layovers – No Layovers from this stage/sprint, it is the final stage. While there is still a great deal of post MVP work to do, the project objectives and deadline have been met and so conclude this process.

6.6 Overview

The stages of development were neatly laid out with a clear trajectory and pace that the project maintained. Despite this, there were frequent amounts of left-over work that rolled into next sprints, and this is a reflection on my inability to see certain limitations about the amount of work taken into each sprint. In the future, a burndown chart may prove useful in predicting an overcoming these shortcomings more efficiently as well as tracking velocity throughout.

7. Output Documentation

This section will display the changes to UML, Collection structure and Architecture that occurred during development. As well as the user stories that were completed.

7.1 Output Diagrams

7.1.1 Architecture

The final architecture diagram (Fig 7.1) highlights the differences between the planned (Fig 3.7) and the final project architecture but also highlight the reduction in complexity and illustrate some of the points made in the BaaS versus Node backend discussion from early in the report (see NoSQL vs Relational Database section).

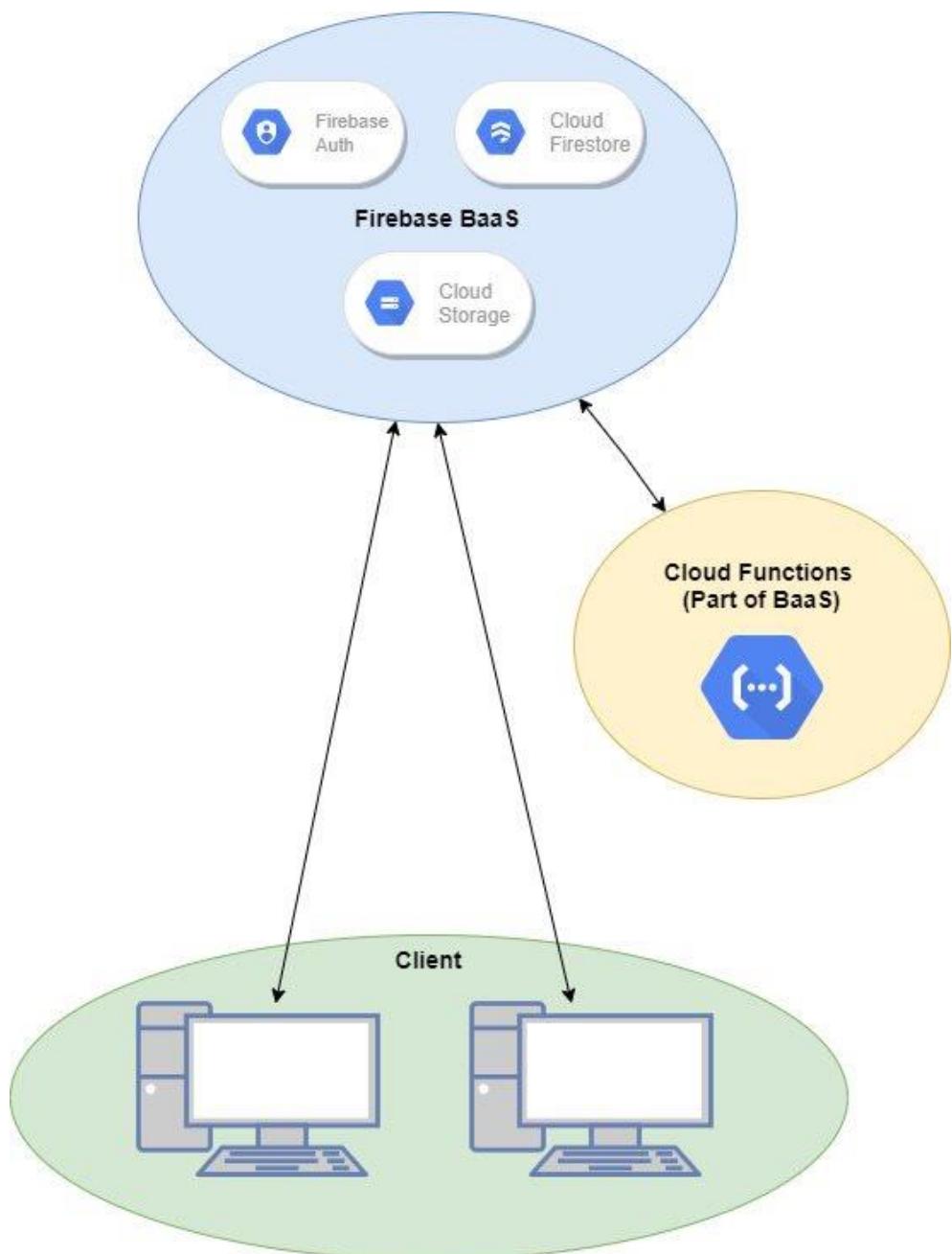


Fig 7.1 Final Architecture Diagram

7.1.2 Sequence UML

The final sequence UML (Fig 7.2) highlights the changes in architecture that occurred during the project. It shows the removal of several steps, making the logic quicker than in the proposed sequence (Fig 3.8).

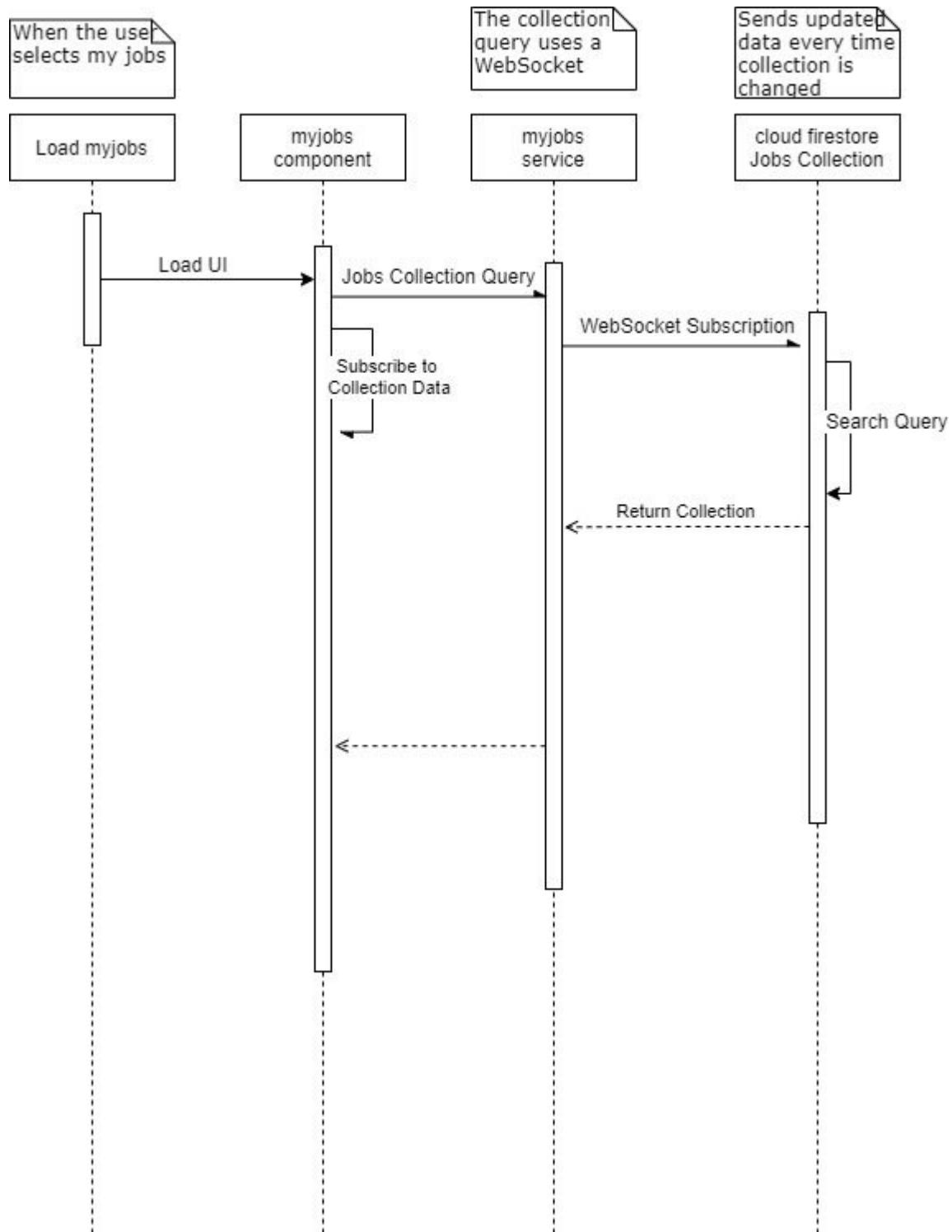


Fig 7.2: Final UML Sequence Diagram

7.1.3 Activity UML

The final activity UML (Fig 7.3) shows a similar layout to the prepared diagram (Fig 3.9). The difference between the two is that the final diagram shows an extended activity based on the more substantial focus for communication and user interaction in the final app.

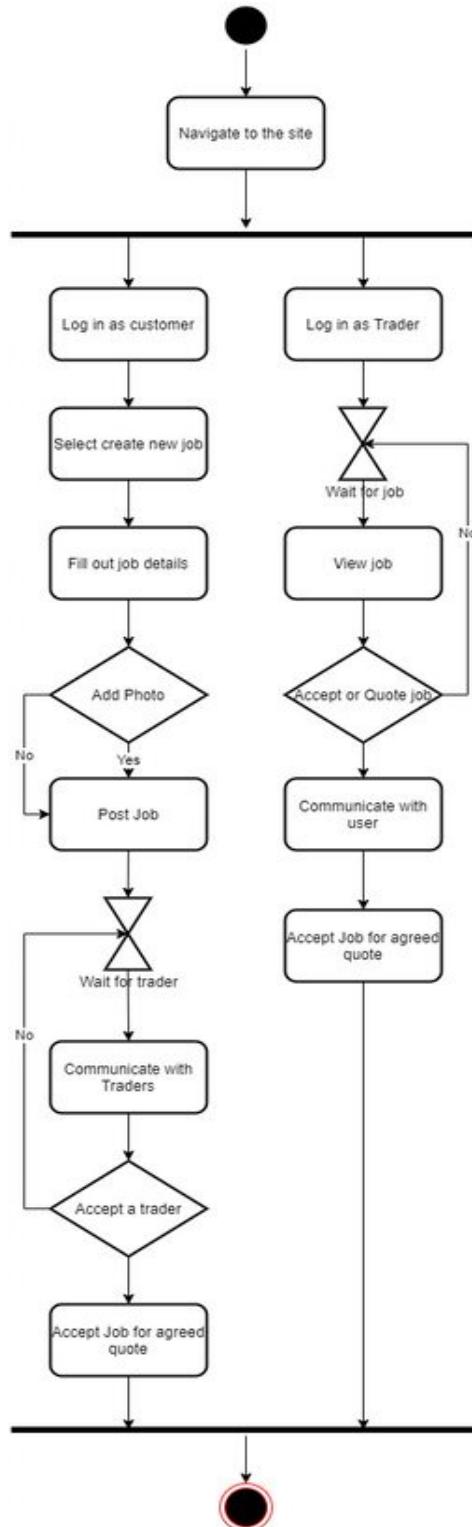


Fig 7.3: Final Activity UML

7.2 Database Collection Structure

The final collection structure (Fig 7.4) shows some major differences between the initial predictions (Fig 3.10). To note the amount of data duplication, due to the limitations of NoSQL collections it is necessary and okay to have duplicated data in these data structures to keep them performant and to account for their lack of true SQL querying.

Chats:	Companies	Jobs	Quotes
companyName	companyName	budget	amount
companyPicture	email	completionState	companyName
id	latlng	description	id
jobId	phoneNumber	id	jobId
jobPicture	photos	issueDate	traderUserId
jobTitle	postcode	issueUserId	
lastContact	tradeType	IngLat	
messages	userId	picture	
		postcode	
		quotes	
		reviewScore	
		timeframe	
		title	
		trade	
		workCandidates	
Reviews	Users		
comment	accountType		
score	displayName		
userId	dob		
	email		
	firstName		
	lastName		
	IngLat		
	phoneNumber		
	photo		
	postcode		
	reviewScore		
	userId		

Fig 7.4: Final collection structure diagram

7.3 Completed User Stories

These can be viewed in the user stories section of the appendix (appendix 12.4 – Requirements). All MVP stories were completed, as well as some post MVP stories.

7.4 User Testing Feedback

As previously mentioned in the testing section, the testing methodology was a key part of development during this project; also including user testing. An example of the blank questionnaire and interviews used can be found in the appendix (appendix 12.8), with the full testing data also available (appendix 12.9).

Summary of the results from user testing

Registration:

Was considered relatively easy to use, some difficulty arose around the labels used to describe inputs (Fig 7.5).

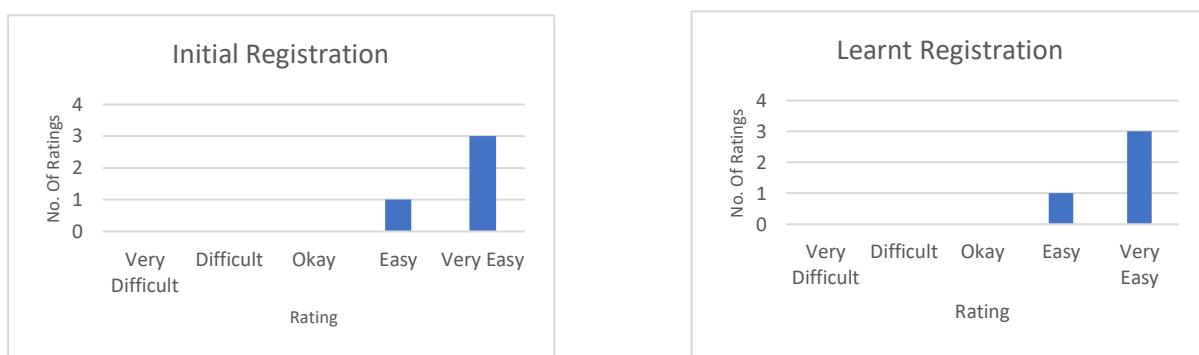


Fig 7.5: Registration test data

Login:

Login was generally received well, but one tester encountered an error when logging in but did not receive any error feedback (Fig 7.6).

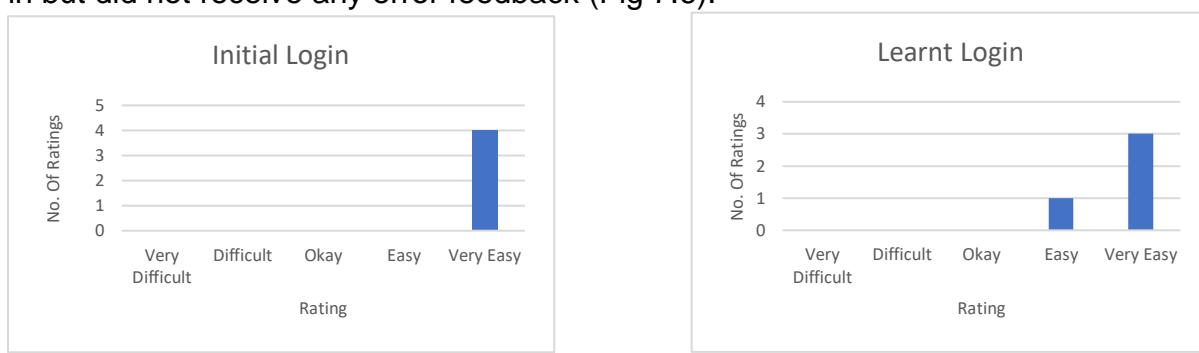


Fig 7.6: Login Test Data

Map Use:

Map use was considered very easy to use, even from an initial state (Fig 7.7).

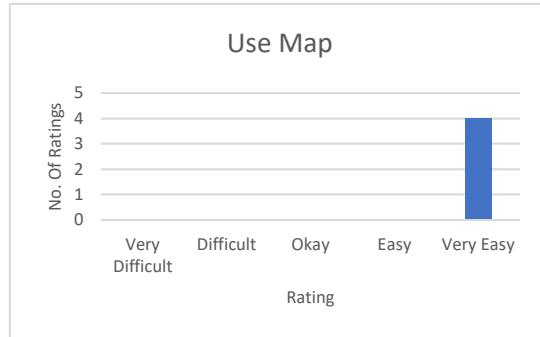


Fig 7.7: Use a Map Test Data

Job Creation:

Job creation was considered very easy to use, even from an initial state (Fig 7.8).



Fig 8: Create a Job Test Data

Finding Jobs:

Find jobs had an initial learning curve for some testers, but it was very intuitive to use after learning (Fig 7.9).

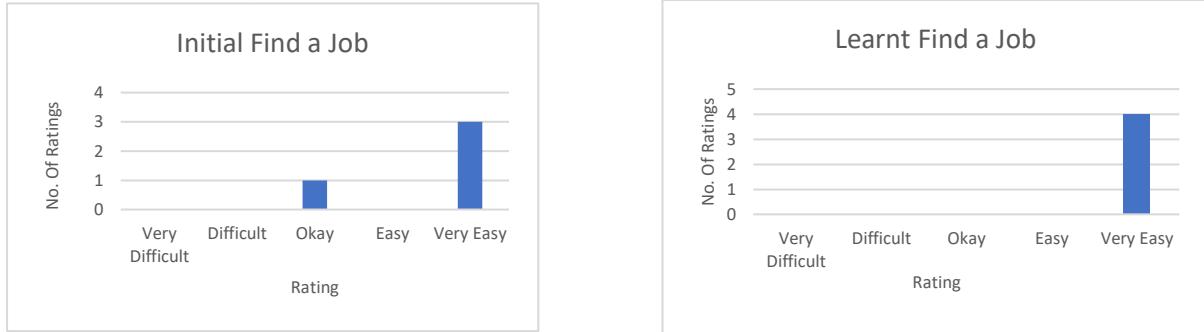


Fig 7.9: Find a Job Test Data

Opening Chats:

Opening chats provided an interesting result, the majority of tests found it initially tricky but once learnt, found it very easy (Fig 7.10).

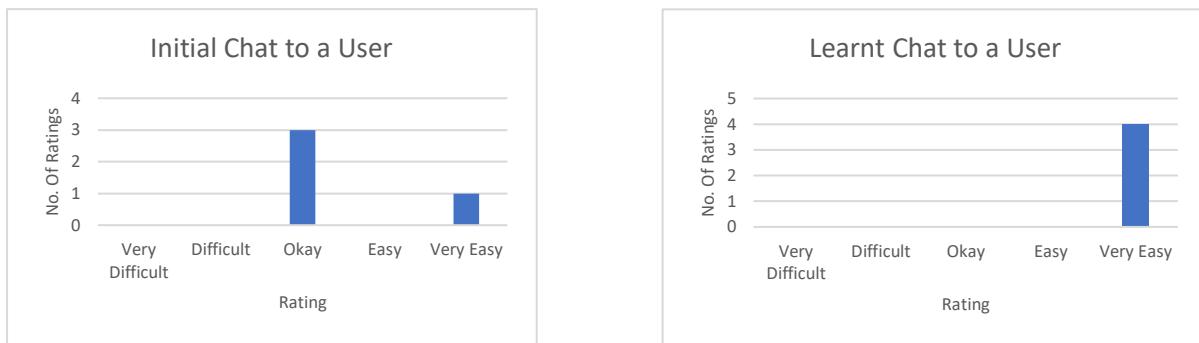


Fig 7.10: Chat to a User Test Data

Interview Question Feedback

Q1 Learnability: How easy (instinctive) was it to work out how to use the software?

Generally, the software was viewed as very easy and intuitive to use by every test applicant.

Q2 Usability: Once you had worked out how to use the software, how easy was it to use?

The understanding was that the software became even easier to use once familiar with its nuances.

Q3 Usefulness: How beneficial do you think this software would be to the business?

Points frequently mentioned in answers were time reductions in communication, making it easier to get companies names out there and be proactive.

Q4 Effectiveness: How do you feel the quoting system would impact your view on the pricing of a project?

The consensus was that it allows people to get the best value for their money and consider many traders.

Q5 Could you see yourself using the application in the future, both as a tradesperson or a consumer?

The applicants all responded positively to using it in the future, with one expressing the applications ability to grow in the future.

Extra Feedback and Feature Ideas From Testing

Having completed there were several feature changes/suggestions made.

- Error feedback on login
- A more clear indication of how to get into chats as a user
- Add labels before you click on the “What kind of User Are You?” in create account
- Are you sure you want to logout modal
- Filter jobs by trade
- Have notifications for job interactions
- The map could take advantage of your entered postcode.

Changes Made In Response

Several of suggested modifications have already been implemented into the system in response to the feedback provided in testing. These are currently just the small additions as MVP remains the primary goal. The other suggestions will be added to the post MVP backlog.

- Error feedback has now been added to the login page, triggering if the attempted login is incorrect.
- The “Job Details” page now has a piece of red text instead of black, outlining how to use the chat functionality.
- The “Home Map” now reads the users inputted postcode and centres the view at that location. Options to dictate how the map operates could be taken further in the future

8. End-Project Report

There were a lot of changes the project underwent regarding design. However, the business objectives remained consistent throughout, with only some of the initial project infrastructure resilience outlined in the project setup document (appendix 12.5) changing. This section will critically describe how these changed and if objectives were met with the final deliverables.

8.1 Design Adherence

8.1.1 Architecture

As illustrated in the output documentation, when compared with the proposed documentation, the architecture has entirely changed. The reasoning behind this as the use of the original architecture (Fig 3.7) serves only to increase the complexity of the project, which while demonstrates more skills, does not bring any significant gains to the project.

Reason – The decision to change the entire backend architecture from NodeJS and Socket.IO was made because of the project using firebase as the Database, Auth and Blob Storage. These systems were designed to work with a frontend, communicating directly to it, having to go through an API would have created significant problems with their functionalities.

Effects – The effect of this is a substantial reduction in complexity, a streamlined deployment and, due to the nature of a Cloud Firestore, Firebase Database, an easily interactable database with realtime updates to client applications.

8.1.2 User Interface and Flow

User interface and flow designs have generally been adhered to, this leading back to the original market analysis and comparisons to similar applications (Uber, Airbnb). The only change is presented in the extension of the UX on the post-project activity UML (Fig 7.3) to display the full interaction. The final applications flow lending itself to a larger amount of communication and comparison between potential offers.

8.2 Objective Completion

1. Build a PWA that works for both mobile and web
 2. Have it deployable via pipeline to demonstrate DevOps Engineering
 3. Allow users and traders to review each other
 4. Use an interactive map to interact with jobs and companies
 5. Allow users to experience a full solution that encompasses the journey of having trade work undertaken.
 - a. Create a Job
 - b. Communicate with traders
 - c. Choose the best trader
 - d. Be able to complete the work solely through using the application
- 1) As displayed in the installation guide (appendix 12.1) and the user guides for both mobile and web (appendix 12.2/3), the application has been built and deployed successfully as a PWA.
- 2) The project is fully deployable to hosting via pipeline and was developed from the beginning this way, demonstrating full CI/CD use. However, with the later inclusion of post MVP user stories, cloud functions were integrated into the project; this had the adverse effect of breaking the Circle CI implementation, and the cloud functions cannot currently be released via pipeline.

Further to this, in the initial product setup document (appendix 12.5), before the research and in the initial submission, there was the suggested use of Infrastructure as code, this became irrelevant due to the use of cloud-hosted solutions, arguably a better solution.

- 3) While not an MVP feature, due to the ability to review not being crucial to the project's delivery, it has been the first post MVP feature that has been implemented. This provides the ability for both customers and traders to review each other and create the review-based economy we see in other applications (Uber, Lyft, Airbnb).

There is currently no ability to view the reviewers' comment in any regard, the score being the only method to convey the review information, the feature falling short in this regard. The functionality to view these is the next most desirable feature.

- 4) The core of the application is the use of the interactive map that allows a user to either view companies or jobs. The jobs are also offering all the primary functionality from their interactive nodes. However, there is currently an issue with overlapping pins being inaccessible until the covering pin has cleared. This objective was vital to the accessibility of the application and even despite the temporary overlap issue, serves to provide a great user experience.

- 5) While the application does encompass the full journey of undergoing trade work and getting trade work as a worker, the objective dictates that it must encompass the entire journey, this includes payment and has not been implemented in the final deliverable. The reasoning for this is that getting the applications core functionality was more important to the project than enabling payment through the application. From discussion with traders outlined in the background communication, the impression gained is that a lot of transactions are made via cash, so this feature was not set as a requirement for MVP.
- a) A user may create a job and have it viewed.
 - b) A user may chat with all traders that express an interest in their job
 - c) A user may use the company info and the review scores to choose a trader of their preference.
 - d) Both traders and user have the tools to complete the work contract through application due to the inclusion of communication features such as chat.
- However, due to the above breakdown of payment, this objective is not truly met.

8.3 Project Conclusion Development Recommendations

If the project were to continue development after the conclusion of this project, there are several recommended areas where further development would make significant improvements in UX.

- Enable Stripe payment through the application, the ability to pay traders and then also the opportunity to monetise the app by adding a fee.
- Expand the review system to incorporate the written reviews
- Include further company functionality on the “Home Map” such as viewing their full image galleries.
- The ability to fully control the data the application holds on a user. Delete accounts/specific data such as company images.
- After speaking with end-users further, the use of a site visit option could also be a beneficial post MVP feature to implement

9. Post-mortem

A critique of the approach taken to various aspects of the project

9.1 Technologies

The technologies used in the development of the project should have been better reviewed before the project began. There was a considerable amount of wasted effort due to the removal of the backend midway through the project, due to the poor understanding of how firebase could be implemented, it serving to be a whole solution instead of just a database service.

Going forward with future projects, prior to the beginning of their development, there should be several solutions suggested and then the complexity of those solutions evaluated to find the best. Had this approach been used at the start of this project, subsequent wasting of time could have been avoided.

The choice of tech stack does not need to be reconsidered, as the full stack has extensive documentation and are all primary platforms for development in the future. The only shortcomings being that both the frontend and Backend are not open source so are subject to the whims of private owners; this could be rectified in the future by switching to open source alternatives such as vue.js and bootstrap for the frontend and create an API for the backend.

Given these points, the current implementation of the backend is a versatile, adaptive and an appropriate solution.

9.2 Project Management Approach

The project management used during the project which, while effective in maintaining a consistent pace of project development, did lack certain amounts of depth. The Trello cards could have been used in a more detailed manner to exactly track workpieces at any given time; fortunately, the use of user stories (appendix 12.4 – Requirements) helped to mitigate this. As well as this, the lack of estimations of stories providing the ability to get an understanding for velocity and generate burndown charts did hinder sprints, often taking in more than was attainable. The overall project was managed well, these changes would have created better personal management, allowing for informed opinions on the projects velocity, and a better understanding of what had been completed and still to do in the Trello.

9.3 Personal Performance

During the project, I believe I have challenged myself in both a sense of ability and understanding, having to learn new concepts and platforms. Points that I am proud of implementing is the Authentication and the overall user experience, having worked hard on both. I am also really pleased to have overcome challenges such as falling off the horse with the API. I do feel that during development, there were also areas where I could have pushed myself further, such as spending more time trying to figure out the pipeline with the review systems. However, that is post MVP and so is something I can address at a later date.

9.4 User Testing

More extensive user testing could have been performed, due to various complications from external factors, this fell short. The project would have benefited greatly as the feedback that was received was very useful.

9.5 Disability access

The project features certain pieces of disability access, such as aria labels; however, it does not feature any colour-blind features or speech output to assist users. While these may have been difficult to implement, they should have been factored into planning and at the very least added as post MVP functionality.

9.6 Overall

The project suffered its most challenging setbacks, not in functionality, but in the technologies, these have been largely addressed and allowed the project to be completed on time. The project delivers to the vast majority of its objectives and goes beyond in some cases. Still, it does not accommodate those with disabilities adequately, or enable transactions to be completed, and these should be a key focus going forward.

10. Conclusions

The project does generally meet the objectives defined at the beginning of the project and the report above. It is a full solution that could drive competition and accessibility into the trade market should it ever be made commercial. Its pitfalls were largely in the technologies and were recognised early enough to be avoided from doing irreparable damage to the delivery of the project in a timely manner. Going forward, it has clear post MVP features that while would bring value, are not necessary for the product to serve its purpose. Delivery of the project was done in a truly agile fashion, releasing on a feature by feature basis as soon as it was complete and passed testing.

On a personal note, I am very proud of the project I have delivered; it has been an idea in my head since the beginning of my placement year and seeing it realised is very gratifying. I am also pleased with the breadth of knowledge I have gained in a wealth of different technologies, the project serving to boost my confidence as a developer and in my skills greatly. Though the setbacks incurred were extensive, they have also provided a large amount of experience in such a way that I will not make the same mistakes with the architecture again.

11. References

- Legislation.gov.uk. 2018. Data Protection Act 2018. Available at: <<http://www.legislation.gov.uk/ukpga/2018/12/contents/enacted>>
- General Data Protection Regulation (GDPR). 2018. General Data Protection Regulation (GDPR) – Official Legal Text. Available at: <<https://gdpr-info.eu/>>
- Firebase. 2020. Firebase. Available at: <<https://firebase.google.com/>>
- Google Cloud. 2020. Server-Side Encryption | Firestore | Google Cloud. Available at: <<https://cloud.google.com/firestore/docs/server-side-encryption>> [Accessed 3 May 2020].
- Ionic Framework. 2020. Ionic - Cross-Platform Mobile App Development. Available at: <<https://ionicframework.com/>> [Accessed 3 May 2020].
- Flutter.dev. 2020. Flutter - Beautiful Native Apps In Record Time. Available at: <<https://flutter.dev/>> [Accessed 3 May 2020].
- web.dev. 2020. Progressive Web Apps. Available at: <<https://web.dev/progressive-web-apps/>> [Accessed 3 May 2020].
- Node.js. 2020. About | Node.Js. Available at: <<https://nodejs.org/en/about/>> [Accessed 3 May 2020].
- Microsoft. 2020. .NET | Free. Cross-Platform. Open Source. Available at: <<https://dotnet.microsoft.com/>> [Accessed 3 May 2020].
- Cli.angular.io. 2020. Angular CLI. Available at: <<https://cli.angular.io/>> [Accessed 3 May 2020].
- MongoDB. 2020. The Most Popular Database For Modern Apps. Available at: <<https://www.mongodb.com/>> [Accessed 3 May 2020].
- Uber. 2020. Earn Money By Driving Or Get A Ride Now | Uber United Kingdom. [online] Available at:
- Airbnb. 2020. Holiday Lets, Homes, Experiences & Places - Airbnb. [online] Available at: <https://www.airbnb.co.uk>
- Nowak, P., 2019. Why You Should Start All Your Projects With A Minimum Viable Product (MVP) - Nopio. Nopio. Available at: <<https://www.nopio.com/blog/start-projects-mvp-development/>>
- CircleCI. 2020. Continuous Integration And Delivery. [online] Available at: <<https://circleci.com/>> [Accessed 7 May 2020].
- Bcs.org. 2020. [online] Available at: <<https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/>> [Accessed 7 May 2020].

Acm.org. 2020. The Code Affirms An Obligation Of Computing Professionals To Use Their Skills For The Benefit Of Society.. [online] Available at: <<https://www.acm.org/code-of-ethics>> [Accessed 7 May 2020].

Trello.com. 2020. *Trello*. [online] Available at: <<https://trello.com/>> [Accessed 8 May 2020].

O'Neill, T., 2019. *It's Serious Business Playing House: Brits Spend £83Bn On Upgrading Their Home Every Year | Zopa Blog*. [online] Zopa Blog. Available at: <https://blog.zopa.com/2019/04/17/its-serious-business-playing-house-brits-spend-83bn-on-upgrading-their-home-every-year/>

Statista Research Department, 2020. *Share Of Households With A Computing Device In The UK In 2019, By Device Type*. [image] Available at: <<https://www.statista.com/statistics/387371/share-of-households-with-a-computing-device-uk/>> [Accessed 13 May 2020].

GOV.UK. 2020. *Child Employment*. [online] Available at: <<https://www.gov.uk/child-employment>> [Accessed 25 May 2020].

Protractor.angular.io. 2020. *Protractor*. [online] Available at: <<https://protractor.angular.io>> [Accessed 25 May 2020].

Medium. 2019. *What Are The SOLID Design Principles?*. [online] Available at: <<https://medium.com/bgl-tech/what-are-the-solid-design-principles-c61feff33685>> [Accessed 25 May 2020].

Angular.io. 2020. *Angular*. [online] Available at: <<https://angular.io/guide/structural-directives>> [Accessed 25 May 2020].

Engel, C., 2018. Early And Often—How To Release Software. [online] Medium. Available at: <<https://medium.com/@CodyEngel/early-and-often-how-to-release-software-bc852f0698bc>> [Accessed 25 May 2020].

12. Appendices

12.1 User Guide: Installation

For general use of the application go to the web address –

<https://tradesman-application.firebaseio.com/login>

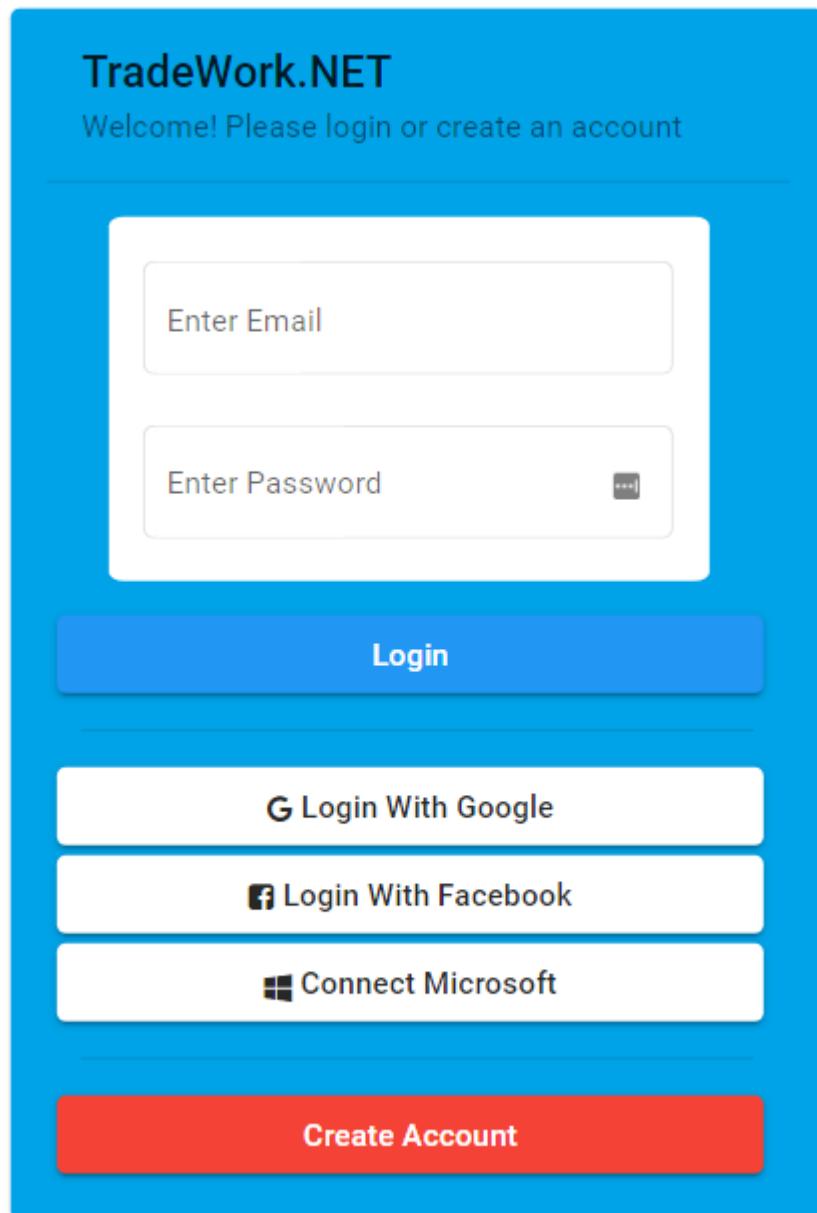
The following guide will outline how to get the dev environment setup and serving on your computer, and it assumes you already have node and git installed on your computer.

1. Open your command-line interface of choice
2. Install npm with the command – `npm install npm@latest -g`
3. Use npm to install the Angular CLI with command – `npm install -g @angular/cli`
4. Clone the GitHub repository
5. Scope your command line to the repository directory and run the command –
`npm install`
6. In the same directory run the command – `ng serve`

12.2 User Guide: User

The following guide is a user guide for how a consumer and not a trader would use the application; it will also show the mobile interfaces for each of the pages for clarity.

Login



Users may log in to the application or create an account, and they may also log in with Google, Facebook, or Microsoft. It is the first page a user is greeted by, and it does have error reporting for incorrect login attempts.

Create Account

Create Account [cancel](#)

Enter Details

First Name *	Last Name *
Joe	Blogs
Password *	Repeat Password *
.....
Minimum Length of 6	
Enter email *	
joeblogs	Nickname *
Must be a valid email	
Phone number *	Date of Birth *
07740301298	5/8/2002
postcode *	Profile Picture <input type="file"/>
PL2 222	

What Kind of User Are You?

I am a...
User

*Passwords do not match *Invalid postcode

Submit

A user may create an account, create an account has validation on all inputs and checks if the postcode is a valid postcode before allowing a user to create an account. They may also upload a profile picture.

If a user chooses to log in with an alternative method instead of creating an account, they will see an alternative create account screen that will only ask for the information the application needs and not account creation info.

First Sign-In Details [cancel](#)

Please provide some extra information

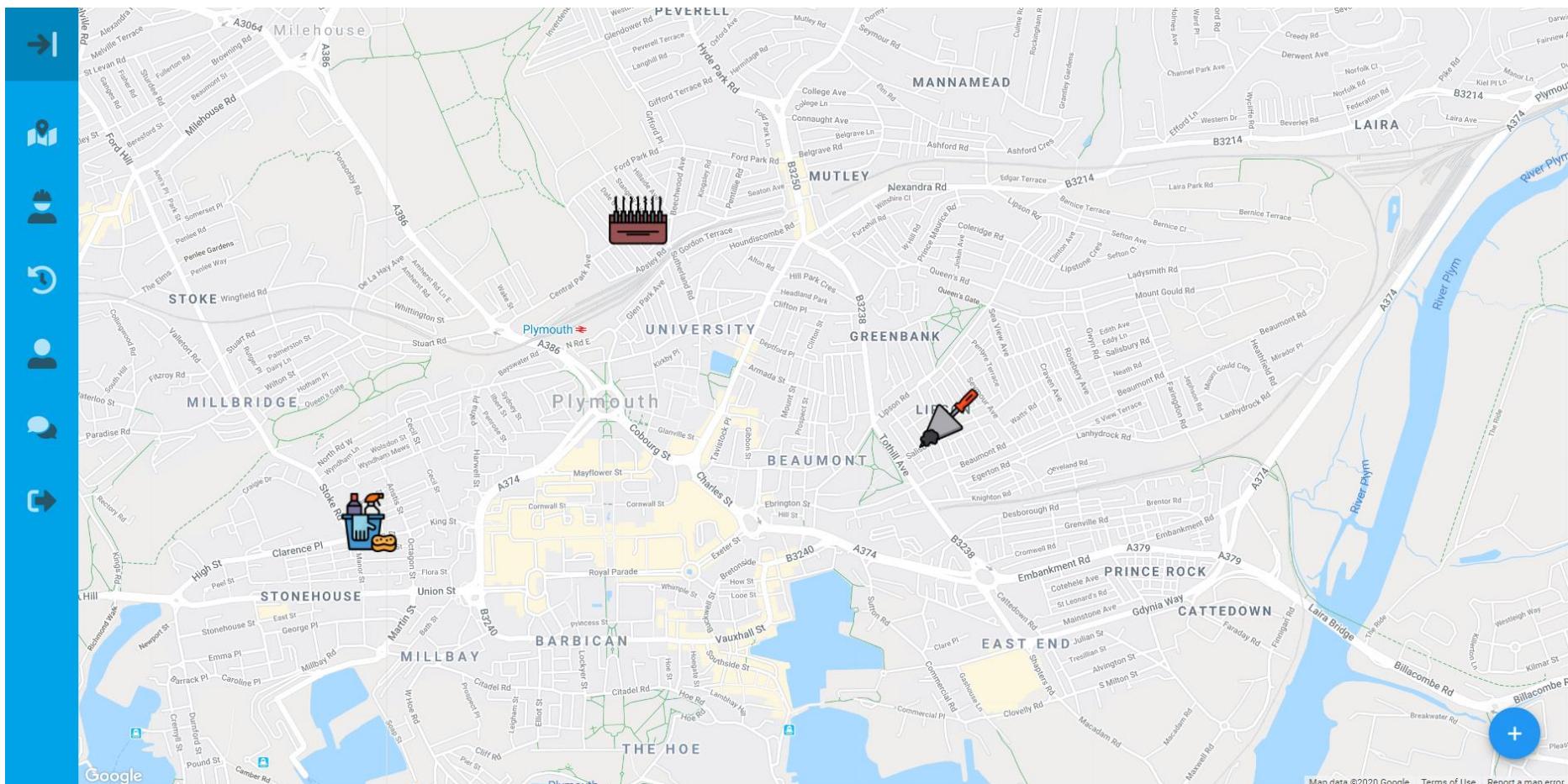
First Name *	Last Name *
Phone number *	Date of Birth *
postcode *	

What Kind of User Are You?

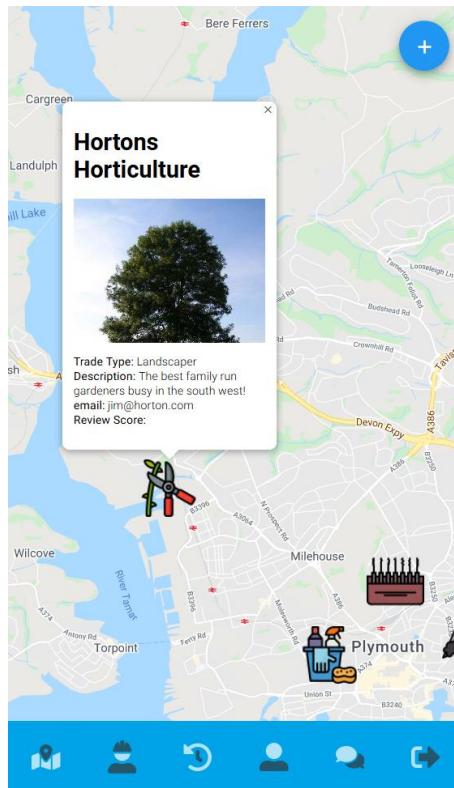
I am a...
Unselected

Submit

Home Map: Web and Mobile



The home map displays the navigation pane which transitions out from the side for readable headings. It also supplies the user with the main feature of the application, the home map itself. The map displays trade companies registered to the app in your local area. It also provides the option to create a new job via the “plus” button in the bottom right corner. The companies are also interactable.



The mobile display has the navigation pane moved to the bottom of the screen, and the add job moved to the top right.

Notice that the garden sheers have been clicked. Demonstrating the active click of the company pin, showing the company info and there first uploaded photo.

Post new Job:

Clicking post a new job triggers the new job component (web left, mobile right). All fields with custom validation the same as with the create account page.

My Jobs

The screenshot shows the 'My Jobs' page with a blue header bar. On the left, there is a vertical sidebar with icons for Home, Jobs, People, and Help. The main content area has a title 'My Jobs' and tabs for 'Active Jobs', 'Quoted Jobs', and 'Available Jobs'. A search bar with placeholder text 'Search by job title' and a letter 'B' is present. Below the search bar, three job entries are listed:

- Fix Broken Boiler**
Location: PL4 7DU Completion State: Available Issue Date: Apr 29, 2020
- Basement Reinforcement**
Location: PL2 1HP Completion State: Available Issue Date: May 21, 2020
- Build Extension**
Location: PL1 3BW Completion State: Available Issue Date: Apr 29, 2020

Each job entry includes a small thumbnail image and a right-pointing arrow icon.

Navigating to the “My Jobs” Page allows a user to see all their jobs and at the different stages of their jobs, quoted, available and active. You may also search by the title of the job.

The mobile display for “My Jobs” shows less data on each of the list items than the web ap

My Jobs

Active Jobs Quoted Jobs Available Jobs

Search by job title

	Fix Broken Boiler Location: PL4 7DU State: Available	>
	Basement Reinforcement Location: PL2 1HP State: Available	>
	Build Extension Location: PL1 3BW State: Available	>

≡ ⌂ ⌂ ⌂ ⌂ ⌂

Job Details

My Jobs > Job Details > Fix Our Electric Cookers FAST

Job Status: Quoted -> The job has quotes for review

Google

Job: Fix Our Electric Cookers FAST Location: PL4 6JH My Review Score: 4★

Chat Accept

Job Info

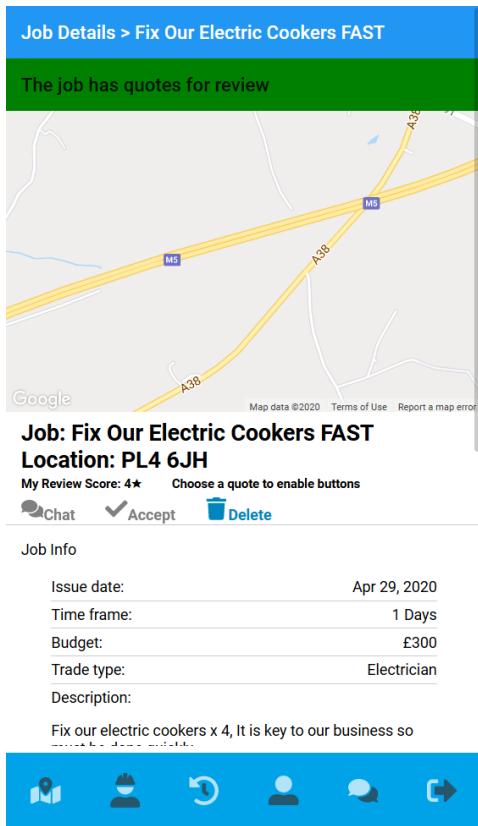
Issue date:	Apr 29, 2020
Time frame:	1 Days
Budget:	£300
Trade type:	Electrician
Description:	Fix our electric cookers x 4, It is key to our business so must be done quickly

Job Image

Select Quote

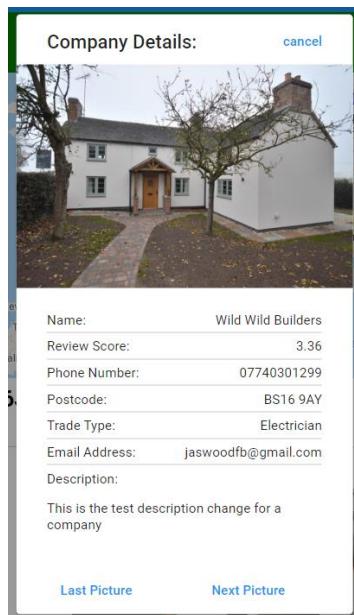
Wild Wild Builders - £500 [Inspect Company](#)

Clicking any job as a user takes you to that jobs “Job Details” page, providing you with the options to control your work. This example is using a quoted job; a user can select the quoting company in the right-hand info section and inspect the company to determine if they would like to accept them. There is a green status bar at the top telling the user the current state of the job and a google maps page showing the users location concerning the location of the company offering the quote. A user has the interaction options below the job title on the left. All other than “Delete” start as greyed out until a company has been selected to complete these actions with.



The mobile display converts the application to scrollable; it also shortens the texts displayed, such as the job status in the green at the top. Notice the greyed-out options even, not having selected a company yet.

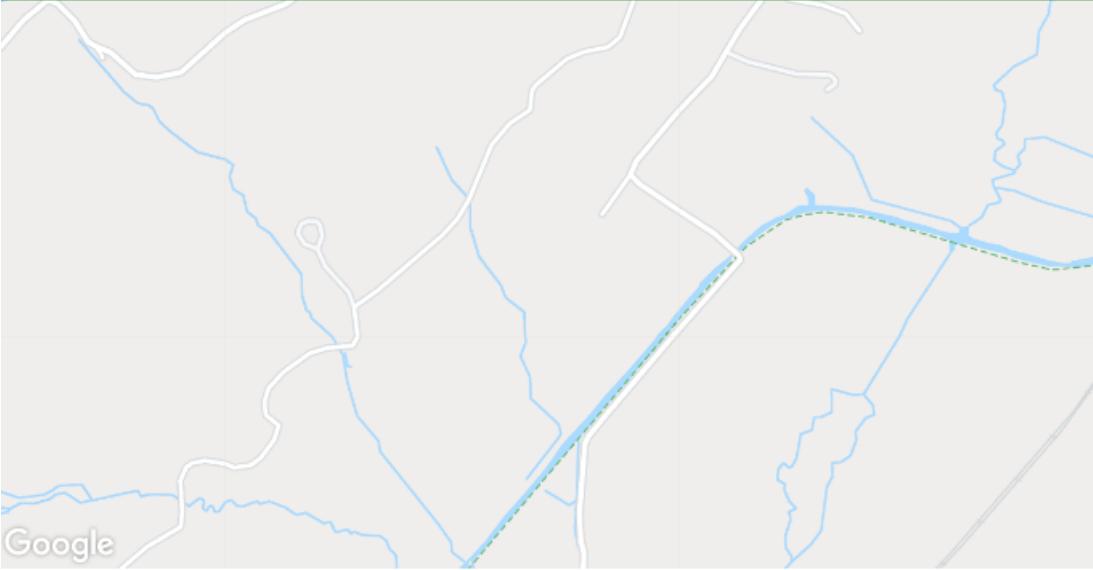
Inspect Company:



Upon choosing a company, you may select them and review their information, and this provides the above view on both mobile and web. It displays general company information and allows the user to browse their gallery of photos.

Review Job:

Job Status: Active -> The job is currently in progress



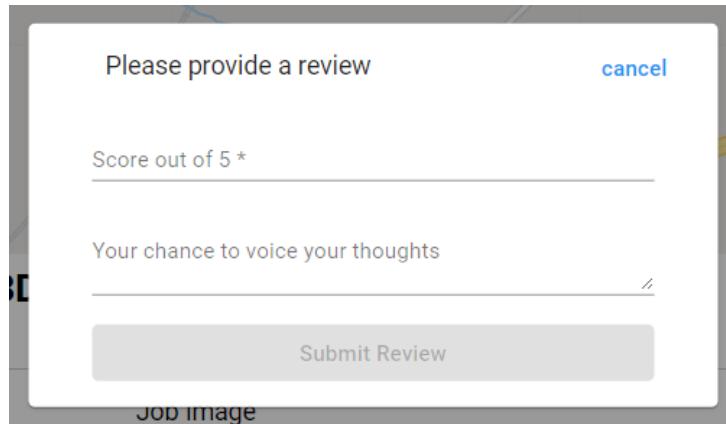
Job: 50 Changing Room Locks Location: PL2 3DF

 Chat  Complete Job

Job Info

Issue date: Apr 29, 2020

Once a job is underway, the options available to the user change; only providing the chat and complete options. Both on “Job History” and “Job Details” if the job is completed by pressing “Complete Job” the review not filled in then the user will be prompted for a review.



The review component also has verification on the inputs to make sure they are of the correct format. This component will appear whenever the job is viewed and remains unreviewed.

Job History:

Job history works in the same way as “My Jobs” being able to search through your historical jobs. Then on click, load that job posts “Job Details” however all of the options other than “chat” are removed or disabled, allowing the user to look back and talk to the trader still but not allowing them to do anything else with it. Mobile displays are below for demonstration, web views not shown due to size.

Job History

Search by job title
o

	Replacement Boiler Location: PL1 1DS State: Closed	>
	hot tub wiring Location: BS16 1ej State: Closed	>

Job Details > Garden Turfing

The job was completed on:



Google Map data ©2020 GeoBasis-DE/BKG (©2009), Google | Terms of Use | Report a map error

Job: Garden Turfing
Location: PL2 2PQ
My Review Score: 4★


Job Info

Issue date:	Apr 25, 2020
Time frame:	3 Days
Budget:	£400
Trade type:	Landscaper
Description:	Garden re-turfed 20 meters square

My Page:

The "My Page" screen displays a user's personal information and allows them to edit it. It includes a photo, account details, and a toggle switch for editing.

Left Screen (Toggle Off):

- Photo:** Two people standing in front of a building.
- Account Information:**
 - First Name: Joe
 - Last Name: Wood
 - Enter Email: jastur96@gmail.com
 - Nickname: Joe
 - Phone Number: 07740301299
 - Postcode: BS16 9AY
 - Review Score: 4
- Upload New Profile Picture:** An arrow pointing up.

Right Screen (Toggle On):

- Photo:** Same photo as the left screen.
- Account Information:**
 - First Name *: Joe
 - Last Name *: Wood
 - Enter Email *: jastur96@gmail.com
 - Nickname *: Joe
 - Phone Number *: 07740301299
 - Postcode *: BS16 9AY
 - Review Score: 4
- Upload New Profile Picture:** An arrow pointing up.

The "My Page" shows a user their personal information and allows them to change it; it only allows changing if the toggle is switched to "on". Each of the fields has validation, and the toggle will not allow itself to be turned back if there are validation mistakes, shown below. Users may also upload new profile pictures which are displayed as shown in the screenshots above. On mobile, the display looks the same. The review score may not be edited under any circumstances.

Account Information

First Name *
Joe

Last Name *
Wood

Enter Email *
jastur96@gmail.com

Nickname *
Joe

Phone Number *
07740301299

Postcode *
BS16 9AY

Must be of length 11
Review Score
4

Upload New Profile Picture

Chats:

The screenshot shows a mobile application interface titled "Chats". On the left is a vertical navigation bar with icons for location, profile, clock, and message. The main area has a blue header "Chats". Below it is a search bar labeled "Search". A list of six conversations is displayed, each with a small thumbnail image, the company name, the job title, and the last message timestamp.

Conversation	Last Message
Best Plastering Co. Job: Banister Spindles Replacing	10:58 29/4/2020
Polishing Polly Job: Fix Our Electric Cookers FAST	13:27 12/5/2020
Chris Wood Ltd. Job: hot tub wiring	20:05 17/5/2020
Wild Wild Builders Job: Key Man Please	12:47 16/5/2020
Best Plastering Co. Job: Garden Turfing	13:58 25/4/2020
Wood & Sons Job: Replace Ariel	15:30 29/4/2020

Shows a list of the current conversations that a user currently has open, they are clickable from various areas in the application, “Job Details”, “Job History” and “Trader Home Map”.

Once open the user will be able to interact with the company in a familiar chat environment. The chat page looks very similar on mobile.

The screenshot shows a mobile application interface titled "Chat with Wild Wild Builders". On the left is a vertical navigation bar with icons for location, profile, clock, and message. The main area shows a conversation between the user and "Wild Wild Builders". The user sent a message with a photo and a caption. The company responded with two messages: one asking for a quote and another suggesting £75. At the bottom, there is a text input field with placeholder text and a send button.

You're having a laugh with £50!

12:47 PM

Well I think that quote is quite reasonable.

11:03 PM

What do you think it should be?

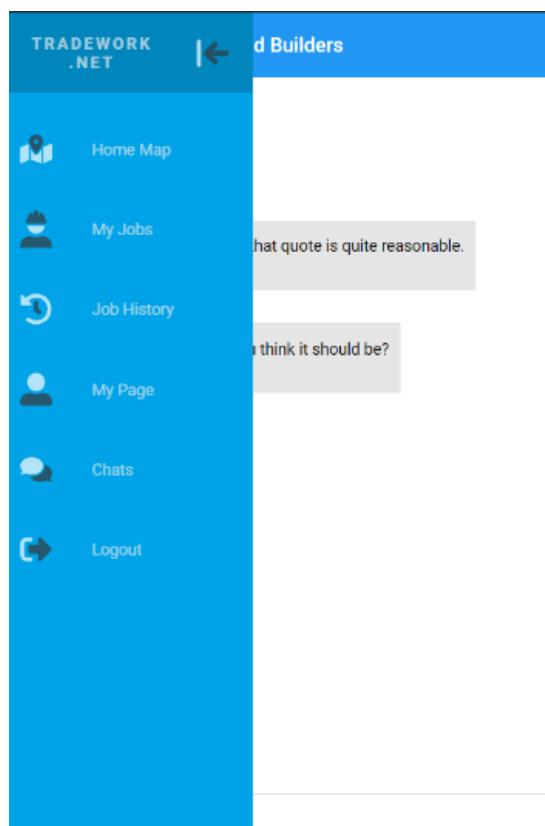
11:03 PM

£75?

11:05 PM

New Message... message ➤

Navigation and Logout:



The navigation pane pops out off the side to display the above, and logout is accessible from here.

12.3 User Guide: Trade

The following guide will highlight how a trader would use the application; it will only include the trade specific changes from the user guide and not be a repeat of the complete user guide shown in the previous section

Create Account:

Create Account [cancel](#)

Enter Details

First Name * Last Name *

Password * Repeat Password

Enter email * Nickname *

Phone number * Date of Birth *

postcode * Profile Picture

What Kind of User Are You?

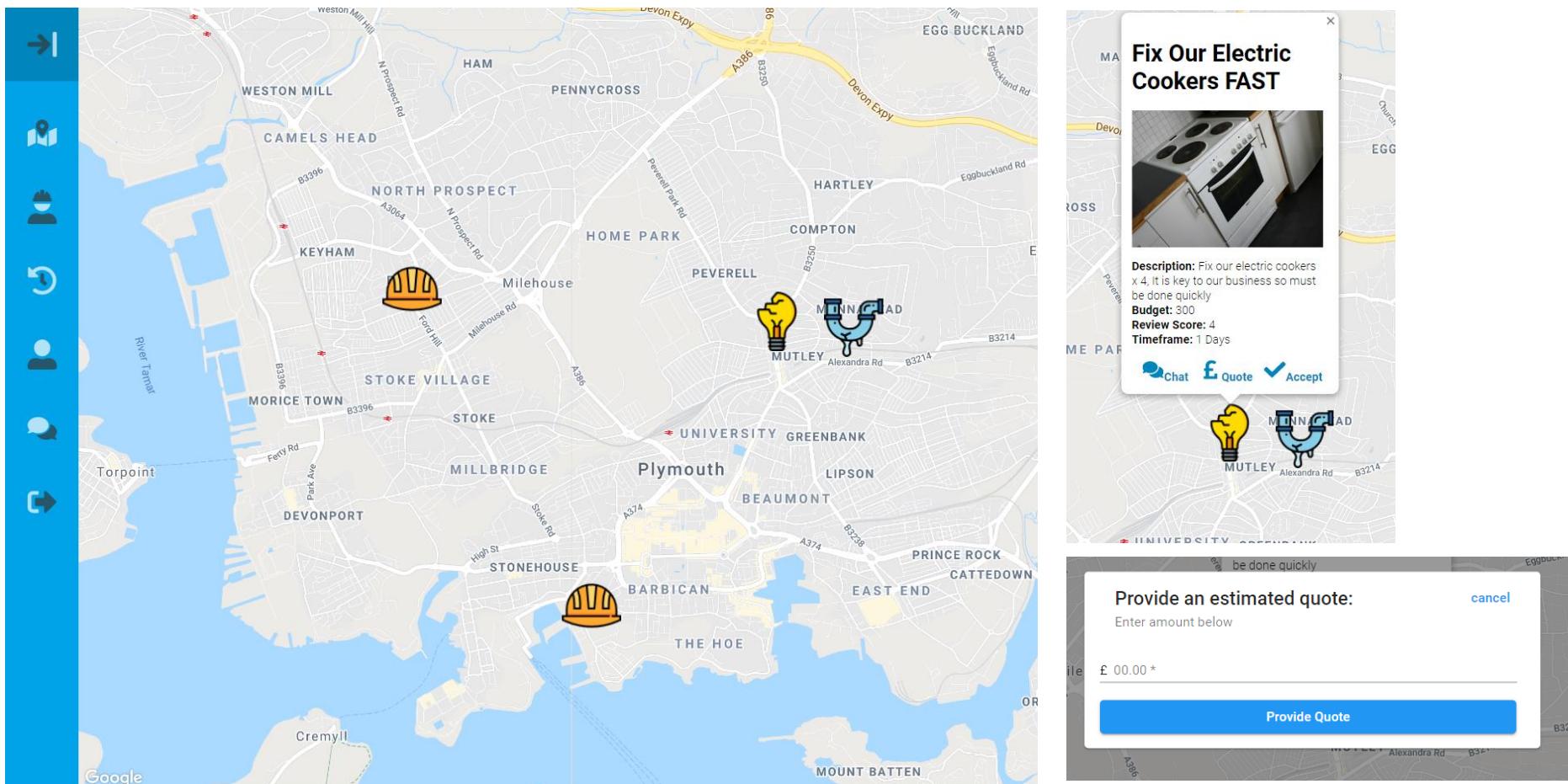
I am a...
Trader

Trade Required * companyName *

Submit

Create account with “What kind of user are you” set to trader provides the options for selecting the type of trade and the company name. These options are dynamically added and removed from the validation with trader user type selection and deselection.

Home Map:



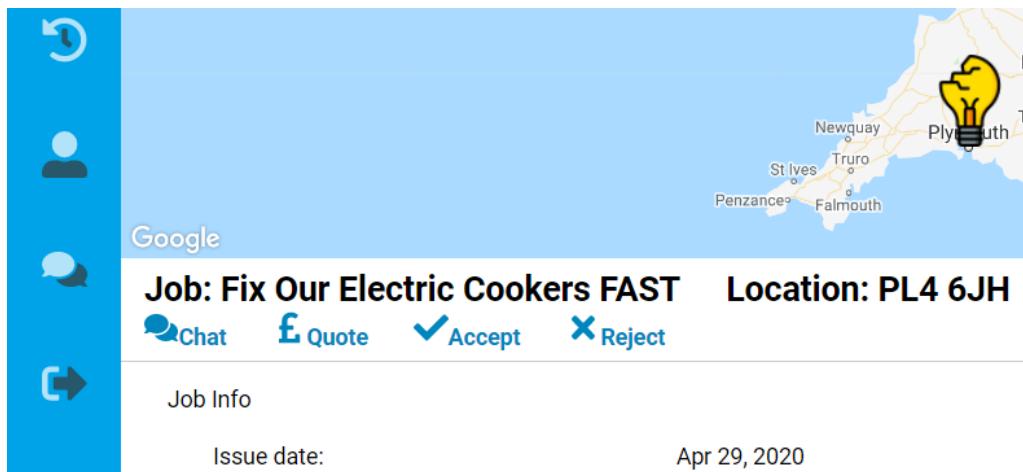
There is no option to create a job, the trader does see pins on the map, but these pins are jobs available in the area. These are clickable and show a job info pane. The info pane on the right shows the quick options allowable from the map. Accepting the job will mark the job as “Trader Accepted” straight away for the advertised budget, creating a quote the budget amount. “Chat” will straight away open a chat with that job issuer and finally “Quote” will open up the quote window to allow the trader to make a quote on the job, quote window is also shown on the right. The rest of the info pane providing valuable information on the job.

My Jobs:

The screenshot shows a mobile application interface titled "My Jobs". On the left is a vertical sidebar with icons for Home, My Profile, My Jobs, My Bookings, and My Quotes. The "My Jobs" icon is highlighted. The main content area has a blue header bar with the title "My Jobs". Below the header are tabs for "Active Jobs" and "Quoted Jobs", with "Quoted Jobs" being the active tab. A search bar labeled "Search by job title" is present. Two job entries are listed: "Key Man Please" (Location: CF45 4LB, Completion State: Quoted, Issue Date: May 16, 2020) and "Fix Our Electric Cookers FAST" (Location: PL4 6JH, Completion State: Quoted, Issue Date: Apr 29, 2020). Each entry includes a small thumbnail image.

"My Jobs" is used in the same way as the user except they do not see the available jobs tab due to traders not having the ability to post jobs, only interact with them.

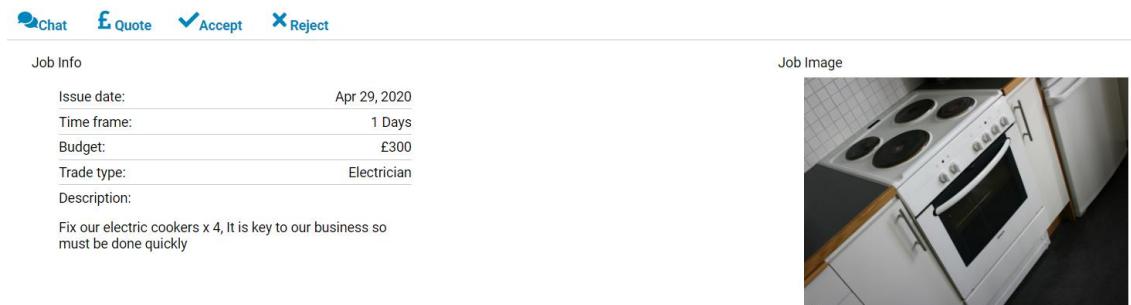
Job Details:



A screenshot of a job details page. On the left is a vertical blue sidebar with icons for refresh, user profile, message, and forward. The main area has a light blue header with a map of the southwest coast of England, showing locations like Newquay, Truro, St Ives, Penzance, and Falmouth. A yellow lightbulb icon is overlaid on the map. Below the header, the job title "Job: Fix Our Electric Cookers FAST" and location "Location: PL4 6JH" are displayed. Underneath are four buttons: "Chat" (blue), "Quote" (blue), "Accept" (green with checkmark), and "Reject" (red with cross). A horizontal line separates this from a "Job Info" section. In "Job Info", the "Issue date:" is listed as "Apr 29, 2020".

Once having clicked a job, the trader will be displayed the job details page with different options than a user, the ability to revise the quote made, accept the job, or, reject the job entirely.

There is also no quotes section on the page, only the job image and the job details.



A screenshot of a job details page. At the top are four buttons: "Chat" (blue), "Quote" (blue), "Accept" (green with checkmark), and "Reject" (red with cross). Below is a "Job Info" section with the following data:

Issue date:	Apr 29, 2020
Time frame:	1 Days
Budget:	£300
Trade type:	Electrician
Description:	Fix our electric cookers x 4, It is key to our business so must be done quickly

To the right is a "Job Image" showing a white electric cooker with four burners and an oven door below it, set into a kitchen counter.

Job History:

“Job History” is again the same as the “Job Details” page but just locked down, it also only shows the job image and the job info for a trade user.

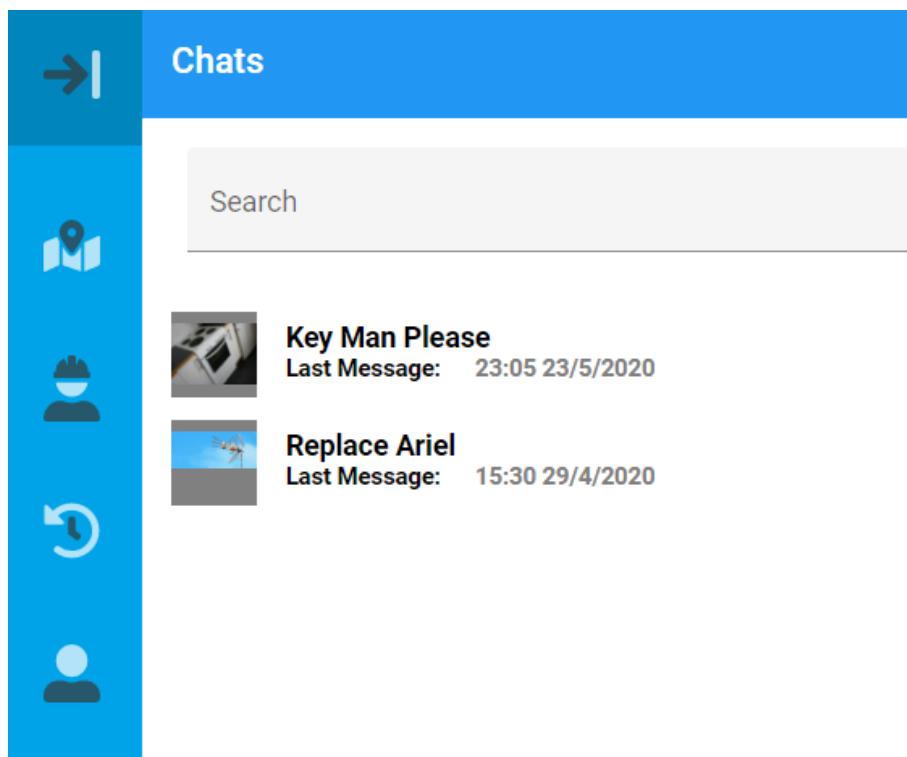
My Page:

The screenshot shows the 'My Page' section of a software application. On the left is a vertical navigation bar with icons for Home, Map, Profile, Activity, and Help. The main area has a blue header bar with the text 'My Page'. Below this is a large image of two people standing in front of a brick building. To the right of the image are two sections: 'Account Information' and 'Company Information'. Both sections have toggle switches at the top. Under 'Account Information', there are fields for First Name (Joe), Last Name (Andrews), Enter Email (jaswoodfb@gmail.com), Nickname (Joe), Phone Number (07740301299), and Postcode (BS16 9AY). Below these fields is a button to 'Upload New Profile Picture'. Under 'Company Information', there are fields for Enter Email (jaswoodfb@gmail.com), Trade Required (Electrician), Company Name (Wild Wild Builders), Review Score (3.36), and a 'Describe the Company' text area containing the placeholder 'This is the test description change for a company'. Below this is a 'Company Image' button and a link to 'View Gallery of 3'.

Now also displays the company information which operates with the same toggle logic as the user account information does. Also providing the option to upload company images and view them in a gallery that overlays the page.



Chat Page:



The chat page just shows less information for a trader. They do not need other information.

12.4 Project Management Requirements

Title	User			Acceptance Criteria			MVP
	As a	I Want to be able to	So that I can	Given a user	When	Then	
Login	User & Tradesman	Log in to my own account	Have tailored content	goes to the site	The login details are fulfilled	Authenticate account and provide result	Y
Home Map	User & Tradesman	View a map of my local Tradesman	View potential Traders/Competition	is on the home page	The user is a tradesman or user	The map should populate with either jobs or tradesman respectively	Y
Create Account	User & Tradesman	Create an account	have a trade or user account on the site	goes to the site	They have no account	Provide them the opportunity to create an account and log in	Y
Create Job	User	Post an image to request a job	Have an initial quote and create a job	wants to create a job	creating the post allow for the photo to be uploaded	Allow the tradesman to view it and give a quote	Y
Create Job	User	Create a job	Have a tradesman come and look at the job and provide an initial quote	wants to create a job	creating the post	Allow the tradesman to view it and give a quote	Y
Job History	User	View my previous jobs	Do accounts, review a history of the work done	needs to find previous jobs	my jobs tab is selected	Show the history of jobs page	Y
Pay Traders	User	Pay a tradesman	Make sure the work is rewarded	has had a job completed	the job is completed on the application	Pay the tradesman	N
Tip Traders	User	Tip a tradesman	Ensure that work to a high standard is encouraged	has had a job completed	the job is completed on the application	Provide the opportunity to tip the tradesman	N
Trader Communication	User	Receive in app communications with the tradesman	Communicate about the work in a more detailed manner	Has posted a job	Garnering interest with tradespeople	Allow for communication between the two clients	Y
Company Page	User	View the Traders Company Page	Look into the contact details of a worker their company details/portfolio	is talking with a tradesperson	the user investigates the trader	provide the user with the traders personal page with their details	Y
Review Scores	User	See the review score of a tradesman	Decide as to whether or not to accept a tradesperson	is talking with a tradesperson	the user investigates the trader	provide the tradesperson score	Y
Accept Jobs User	User	Reject or Accept and offer	Pick who I would like to do my work	has a tradesperson accept the job	Negotiations have concluded	provide the user with the ability to accept or reject the work	Y
Company Details	Tradesman	Put my details up on to a company page	Advertise my expertise	is setting up their account	a trader joins provide a page to display the details	display the details when the trader is selected	N
Review Scores	Tradesman	See the review score of customers	To decide if I want to accept a job	is talking with a customer	the trader investigates the customer	display the customers score	Y
Users	Tradesman	View the available jobs on the map	browse the jobs available to me	is on the home page	The user is a tradesman or user	The map should populate with either jobs or tradesman respectively	Y
Home Map	Tradesman	Accept a job	earn money and gain more customers	has posted a job	Negotiations have concluded	Allow the tradesman to accept the job	Y
Accept Jobs Trader	Tradesman	Review the work performed or customer worked for	inform others of there capabilities	has concluded a job	the payment has concluded	offer the opportunity to give feedback	N
Review System	User & Tradesman	View my job history	For accounting and portfolio purposes	needs to see the jobs they've done	on the home page	provide a history of jobs page	Y
Job History	Tradesman	Receive an email invoice	Have invoice proof for accounting	has concluded a job	payment has been processed	email both the trader and the user an invoice	N
Pay Traders	User & Tradesman	View account details	see what data is tied to my account	wants to view their account details	they click their profile	display the account details	Y
View My Account	User & Tradesman	Delete my account	Leave the site if I wish	wants to delete their account	goes on to the account details	provide them a button to delete their account with a warning	N
Delete Account	User & Tradesman	View local favourites/certified	Be sure of a tradesmans quality	wants to hire a tradesperson	they are confirming a job	show a certified symbol	N
Show Favourites	User	See the average costs of a tradesmans work	So you can be sure of quote accuracy	wants to hire a tradesperson	they are inspecting the traders details page	show the average cost of a job	N
Trader Metadata	User						

A full break down of the user stories used as requirements for the project, further to this it also includes whether these requirements were MVP or not.

Sprint Plan

Sprint Number	Start Date	Conclusion Date	Duration/Days	Task	Details	Trello's	Lay over work
1	24/01/20	07/02/20	14	Initial Setup	Trello Setup Initial documentation Backlog setup Request Ethical Approval Market Analysis	1,2	Market Analysis
2	07/02/20	21/02/20	14	Planning and Preparation	Pre-Dev Designs Create Backend Environment Create Frontend Environment Create Database Create UI Mock-Ups	3,4	Frontend Environment PWA Integration Create Database
3	21/02/20	06/03/20	14	Create Pipelines start UI	UI Feedback Risk Analysis Frontend Pipeline Backend Pipeline Initial UI for Trader and User	5,6	Backend Pipeline Initial UI's
4	06/03/20	20/03/20	14	Key Integrations and UI	Finish UI Integrate Google Maps Authentication and Create Account	6,7	Create Account + Auth Routing
5	20/03/20	27/03/20	7	Backend Integration with Frontend	File Uploading Build out NoSQL integrations (db)	7	
6	27/03/20	10/04/20	14	Feature Logic	My jobs backend logic and functionality Home Map backend logic and functionality Chat functionality	8	Job History
7	10/04/20	24/04/20	14	Finish Systems	Job history functionality My Page Complete User and Trader Functionalities	8	Review System
8	24/04/20	08/05/20	14	Less important features and Report	Review System (Post MVP) Cloud Functions Polish Report	9	

A sprint plan for the proposed sprints that occurred, it was updated with each sprint and used to keep track of what completed and current work. Note the shorter sprint to allow for a quick burst on the backend and tidy up the two weekly sprints to work more nicely with the Trello submission dates.

Trello Boards and Sprint Reviews

This section will show each Sprint with its associated Trello board/s followed by its sprint review; some Trello boards are repeated due to their association to multiple sprints. The spec was suggesting 2-week sprints but Trello boards at changing intervals.

Sprint 1

Trello 1

Final Year Project

Personal Public Invite

Backlog

- Log in
- Front end pipeline
- back end pipeline
- integrate google maps
- integrate terraform for infrastructure as code
- implement photo uploading
- review system
- user system
- tradesman system

+ Add another card

Things To Do

- Create Front End Environment (9 Feb) 0/3
- Create Back End Environment (9 Feb) 0/3
- Create Database (0/3)
- Create UI Mock Ups (9 Feb) 0/2
- Request Ethical Approval (9 Feb) 0/3
- Design Classes and Models (9 Feb) 0/3

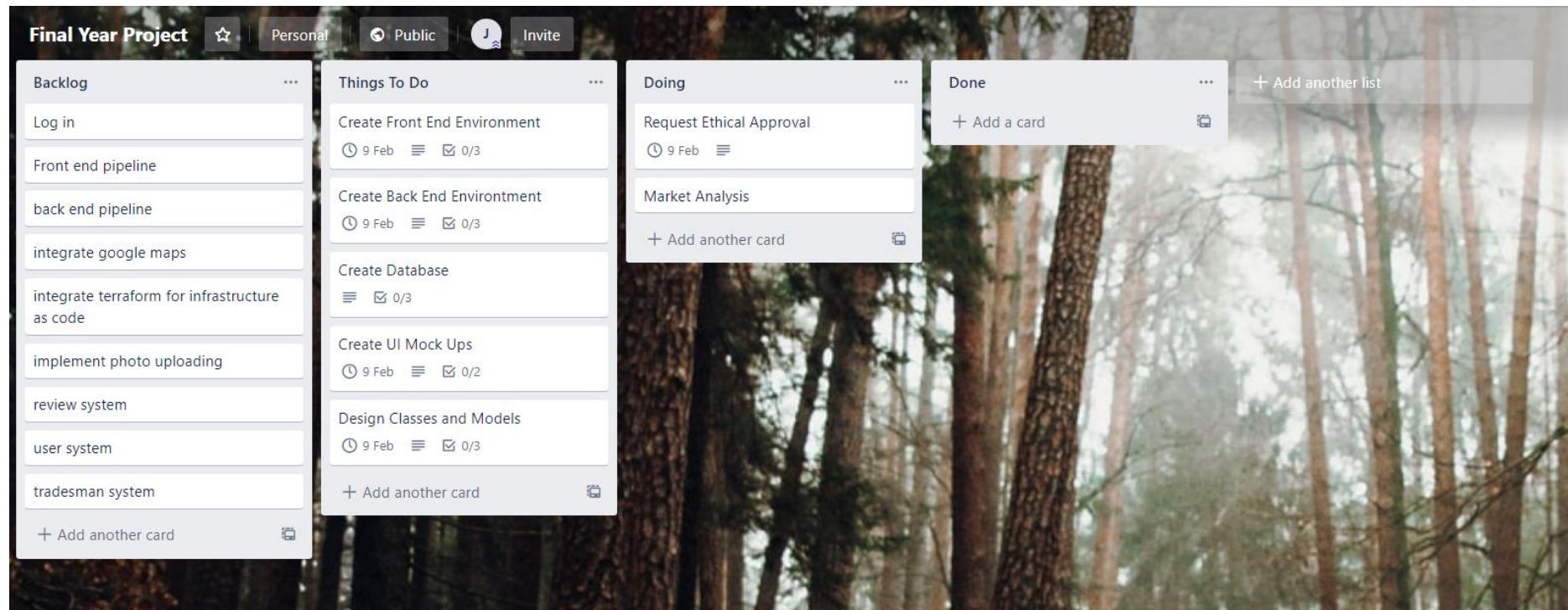
+ Add another card

Doing

Done

+ Add another list

Trello 2

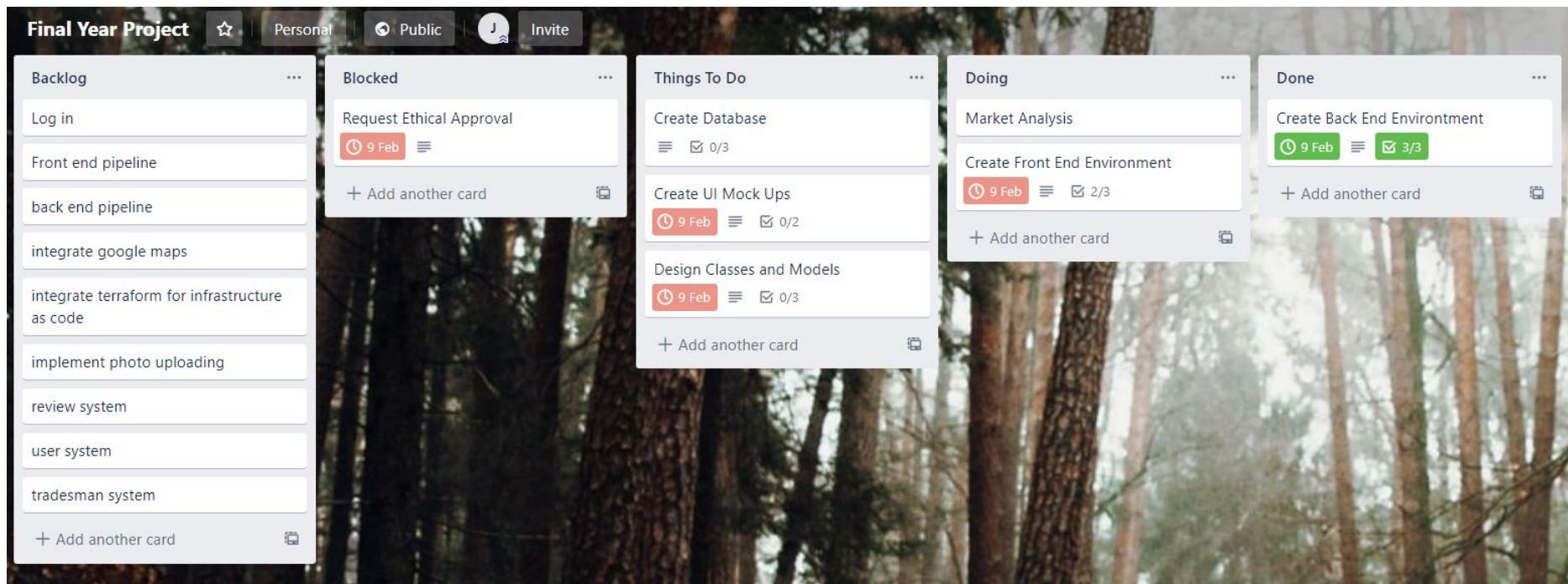


Sprint Review

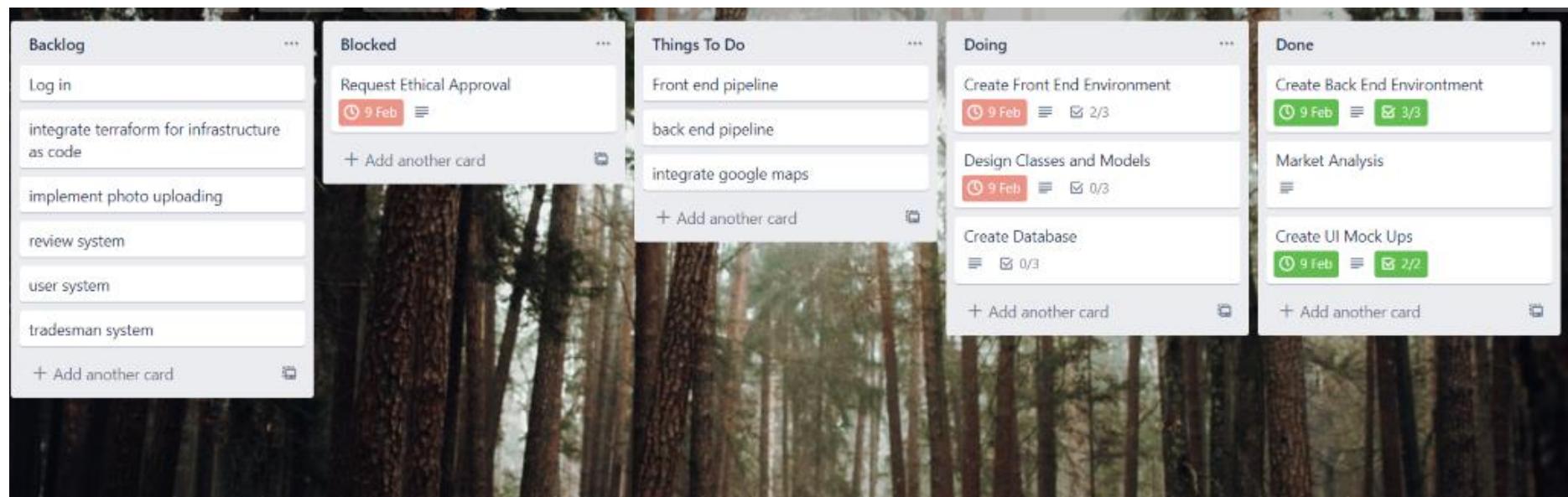
Sprint 1: 24/01/20 - 07/02/20	
What went well	What did not go well
What can I change	How do I feel
<p>Initial meeting with David,</p> <p>Trello setup was good with intial backlog being setup,</p> <p>Ethical approval has been started</p> <p>The effective approach taken so far has a good pace</p> <p>Getting wider information about products available and talking to trader</p>	<p>The amount of work required for ethical approval.</p> <p>Market analysis is still being undergone</p>
<p>Get the market analysis done as soon as possible.</p> <p>Submit ethical approval as soon as possible.</p>	<p>The project has kicked off well with a clear vision and I am positive about where it is heading</p>

Sprint 2

Trello 3



Trello 4

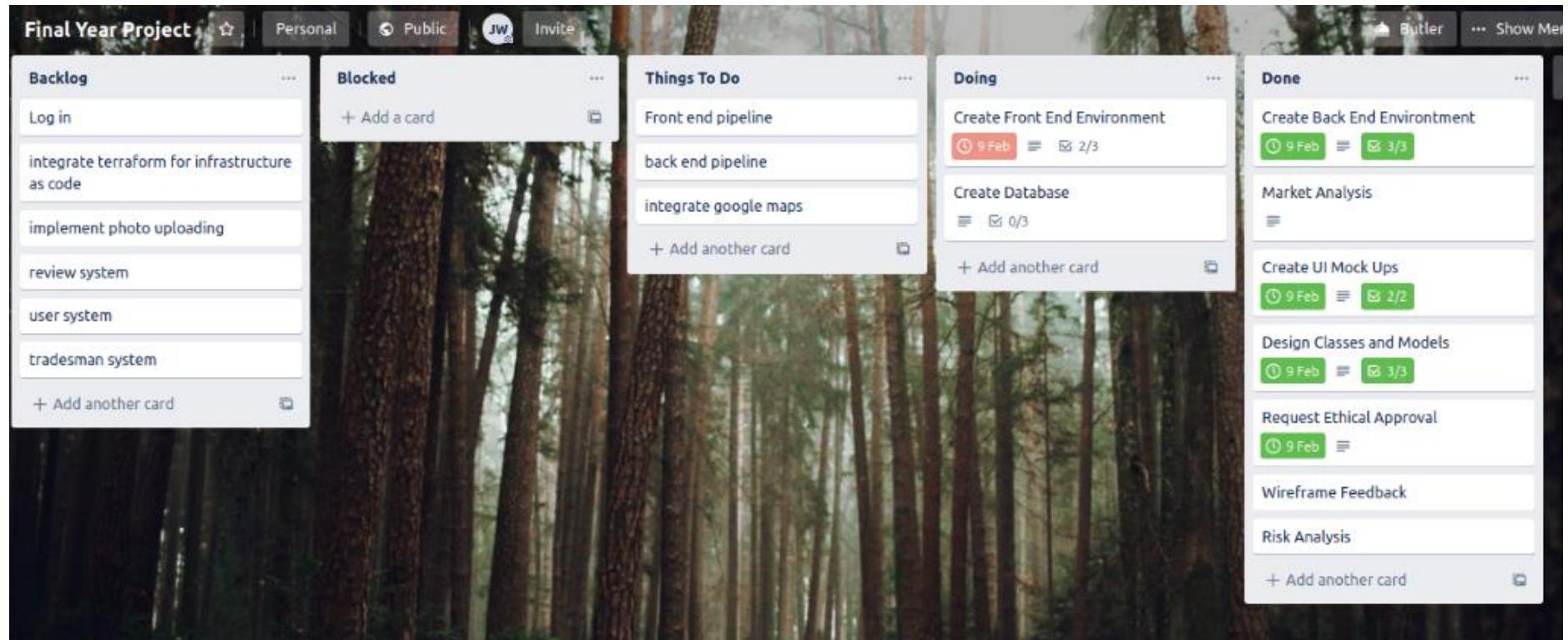


Sprint Review:

Sprint 2: 07/01/20 - 21/02/20	
What went well	What did not go well
What can I change	How do I feel
<p>Lay over from previous sprint was completed.</p> <p>Really pleased with the UI designs.</p> <p>Frontend and Backend environments are ready and hosted in github.</p> <p>Some good designs for how things will be architected</p>	<p>Ethical approval blocked by waiting for approval.</p> <p>Haven't made the frontend environment into a PWA yet, want to do some research</p> <p>Haven't gone in to firebase yet to create database or auth</p>
<p>Chase people for ethical approval</p> <p>Really focus on taking a look into PWA.</p> <p>Spend less time forcing things to be perfect (backend) allowing time to be directed to tasks that are now layed over</p>	<p>I feel good about the project, happy about designs and environments. I am also concerned that I haven't started programming yet but know that being prepared is a better decision.</p>

Sprint 3

Trello 5



Trello 6

The image shows a Trello board titled "Final Year Project". The board is organized into five columns: Backlog, Blocked, Things To Do, Doing, and Done. The background of the board is a photograph of a dense forest.

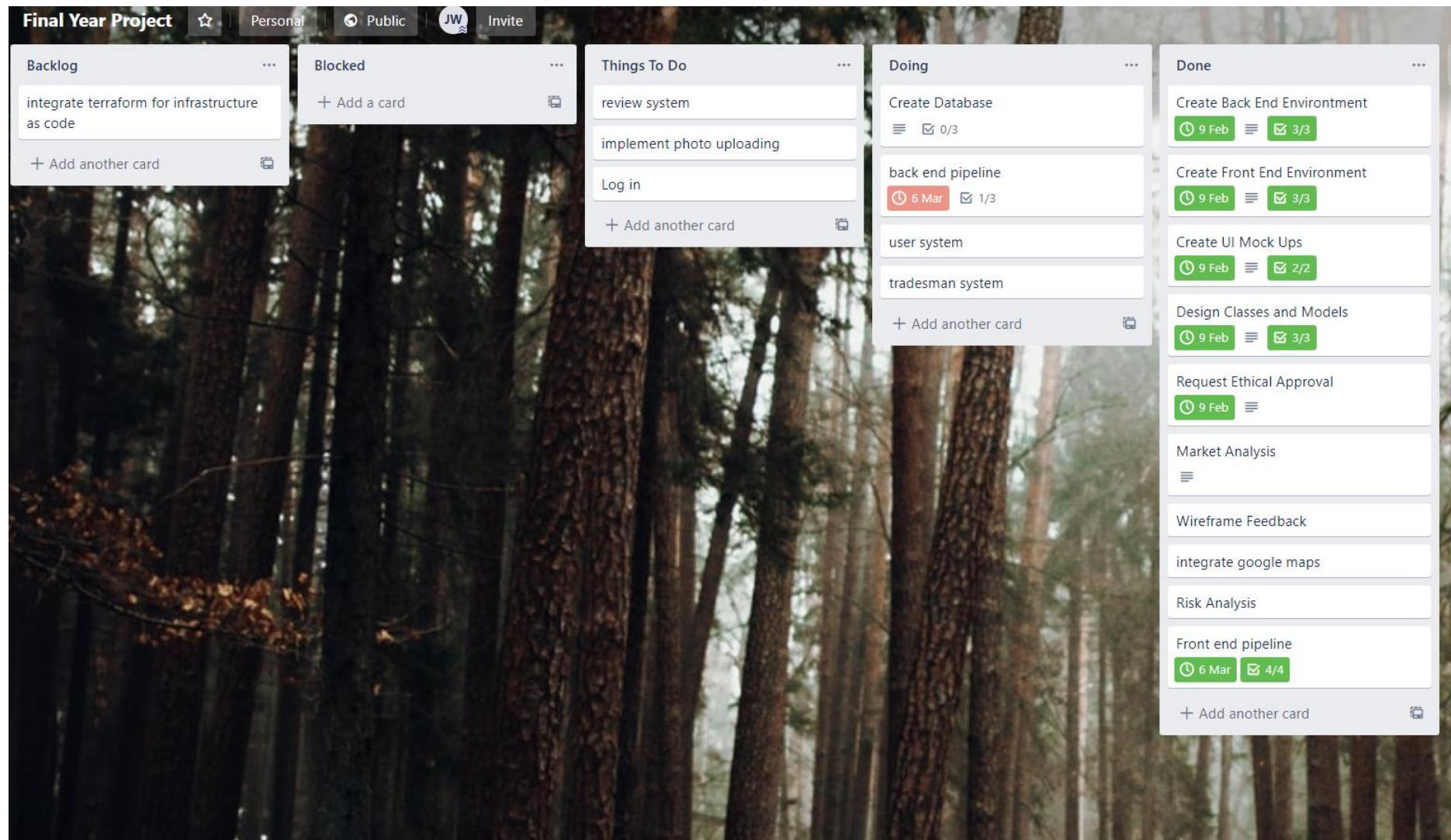
- Backlog:**
 - integrate terraform for infrastructure as code
- Blocked:**
 - + Add a card
- Things To Do:**
 - review system
 - implement photo uploading
 - Log in
- Doing:**
 - Create Database
 - 0/3
 - back end pipeline
 - 6 Mar 1/3
 - user system
 - tradesman system
- Done:**
 - Create Back End Environtment
 - 9 Feb 3/3
 - Create Front End Environment
 - 9 Feb 3/3
 - Create UI Mock Ups
 - 9 Feb 2/2
 - Design Classes and Models
 - 9 Feb 3/3
 - Request Ethical Approval
 - 9 Feb
 - Market Analysis
 -
 - Wireframe Feedback
 - integrate google maps
 - Risk Analysis
 - Front end pipeline
 - 6 Mar 4/4

Sprint Review

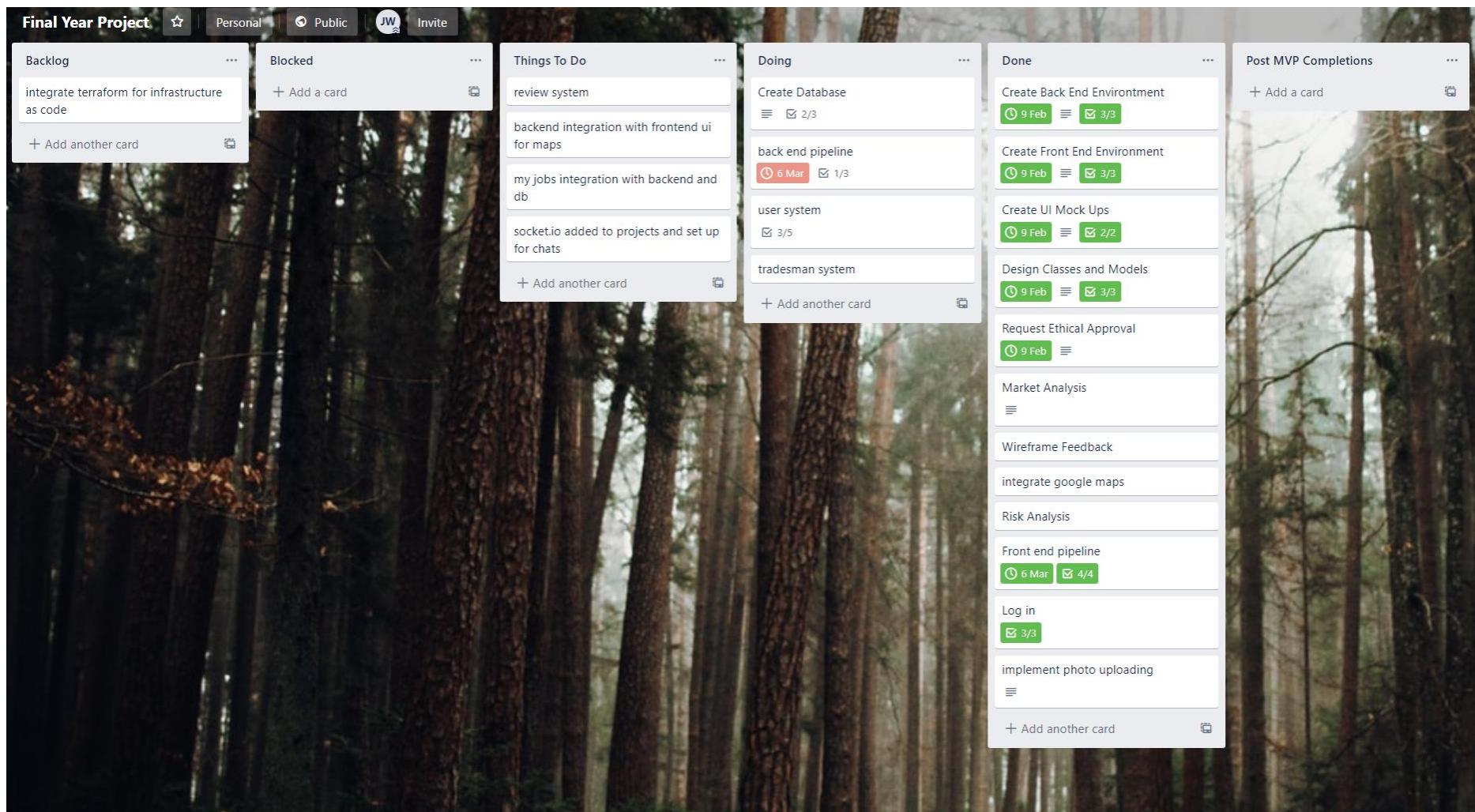
Sprint 3: 21/01/20 - 06/03/20	
What went well	What did not go well
What can I change	How do I feel
<p>Development has started! Just initial UI but it feels good to get something tangible finally.</p> <p>Frontend pipeline works beautifully</p> <p>UI Feedback was good, few changes recommended</p>	<p>Risk analysis was done a bit late in the project at this point, still useful to be aware of the risks before coding though.</p> <p>Backend pipeline was an absolute nightmare days and days wasted</p>
<p>Change the way the backend pipeline is integrated, take stock of different CI/hosting deployment options.</p> <p>Try to vary my time up a bit better, beating myself up about the backend pipeline was not good and really effected moral.</p>	<p>Run down about the backend pipeline, that seriously dampened my spirits. Positive about using the technology with Circle CI that went well with the frontend and keen to get on to full fledge feature development now but it feels good to get the UI started.</p>

Sprint 4

Trello 6



Trello 7



Sprint Review

Sprint 4: 06/03/20- 20/03/20	
What went well	What did not go well
What can I change	How do I feel
<p>Google is now working with the app, went through quite a few technologies but have finally settled on one that works nicely.</p> <p>General UI is built.</p> <p>How simple its been to integrate auth</p>	<p>There was some layover work because auth took a longer time than I thought but I am generally happy with things going on track.</p> <p>I am still not pleased with the backend pipeline.</p>
<p>Think manage time more effectively to not get spill over into other sprints to prevent layovers occurring</p>	<p>I am still not okay with the backend pipeline, but am happy its done.</p> <p>Further to this I am really pleased auth is almost there and things are still progressing at a good rate</p>

Sprint 5

Trello 7

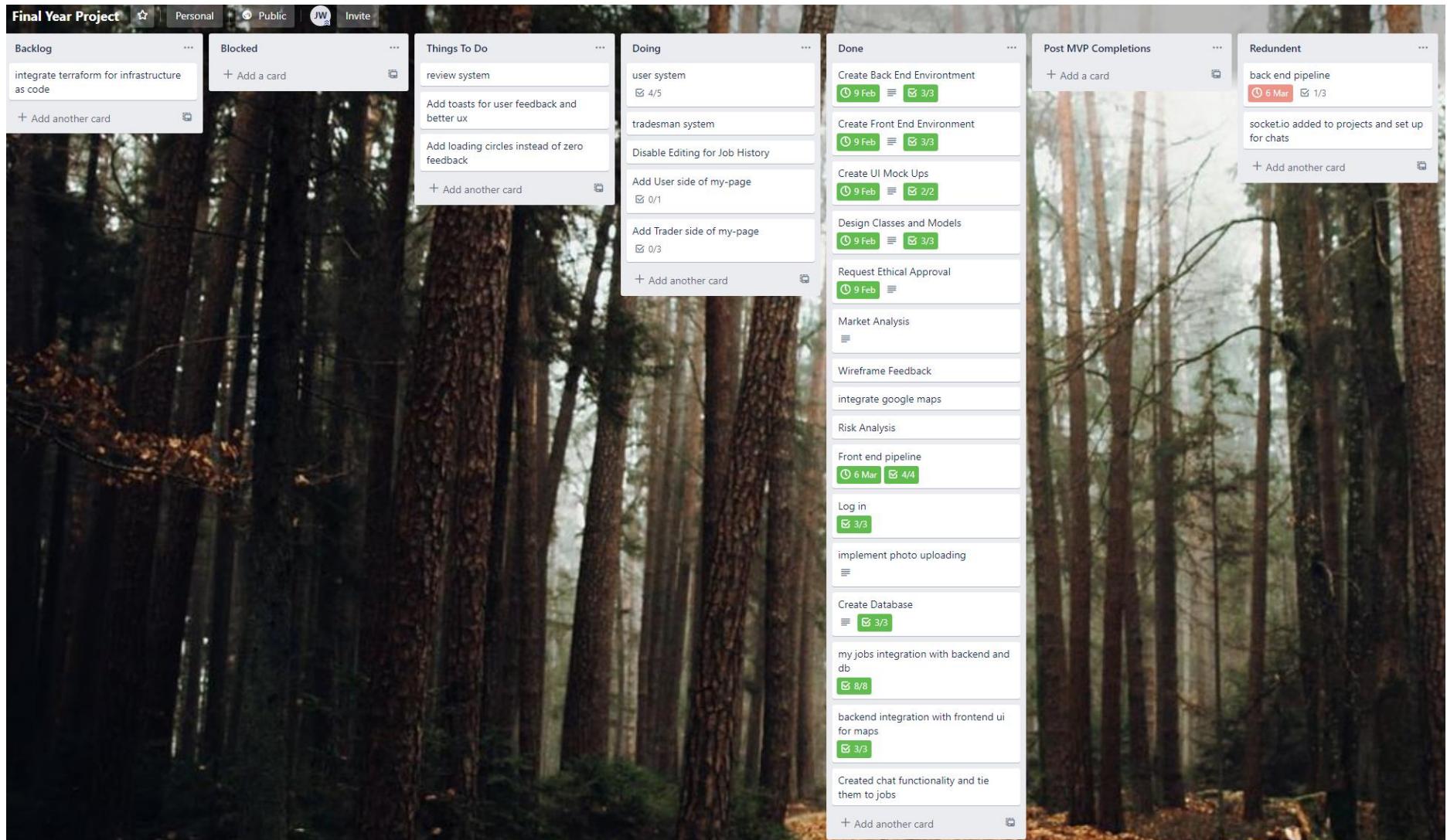
The Trello board is titled "Final Year Project". It features six columns:

- Backlog**: Contains one card: "integrate terraform for infrastructure as code".
- Blocked**: Contains one card: "+ Add another card".
- Things To Do**: Contains four cards:
 - review system
 - backend integration with frontend ui for maps
 - my jobs integration with backend and db
 - socket.io added to projects and set up for chats
- Doing**: Contains four cards:
 - Create Database (due 9 Feb, 2/3 completed)
 - back end pipeline (due 6 Mar, 1/3 completed)
 - user system (3/5 completed)
 - tradesman system (due 9 Feb, 0/1 completed)
- Done**: Contains ten cards:
 - Create Back End Environment (due 9 Feb, 3/3 completed)
 - Create Front End Environment (due 9 Feb, 3/3 completed)
 - Create UI Mock Ups (due 9 Feb, 2/2 completed)
 - Design Classes and Models (due 9 Feb, 3/3 completed)
 - Request Ethical Approval (due 9 Feb, 0/1 completed)
 - Market Analysis (0/1 completed)
 - Wireframe Feedback (0/1 completed)
 - integrate google maps (0/1 completed)
 - Risk Analysis (0/1 completed)
 - Front end pipeline (due 6 Mar, 4/4 completed)
- Post MVP Completions**: Contains one card: "+ Add another card".

Sprint 5: 20/03/20 - 27/03/20	
What went well	What did not go well
Change to a short burst sprint to get the database built out, full decision to abandon the api has been made which has allowed for a lot fo seamless communication which works really well. No Layover!	Certain ways of communicating with the backend aren't as nice as I would have liked
What can I change	How do I feel
Potentially explore different methods of getting data from the backend	I am really pleased with the decisions made this sprint, dropping the backend is definitely right decision as I felt like I was only doing it for the complexity. It has also decreased stress which is also good.

Sprint 6

Trello 8

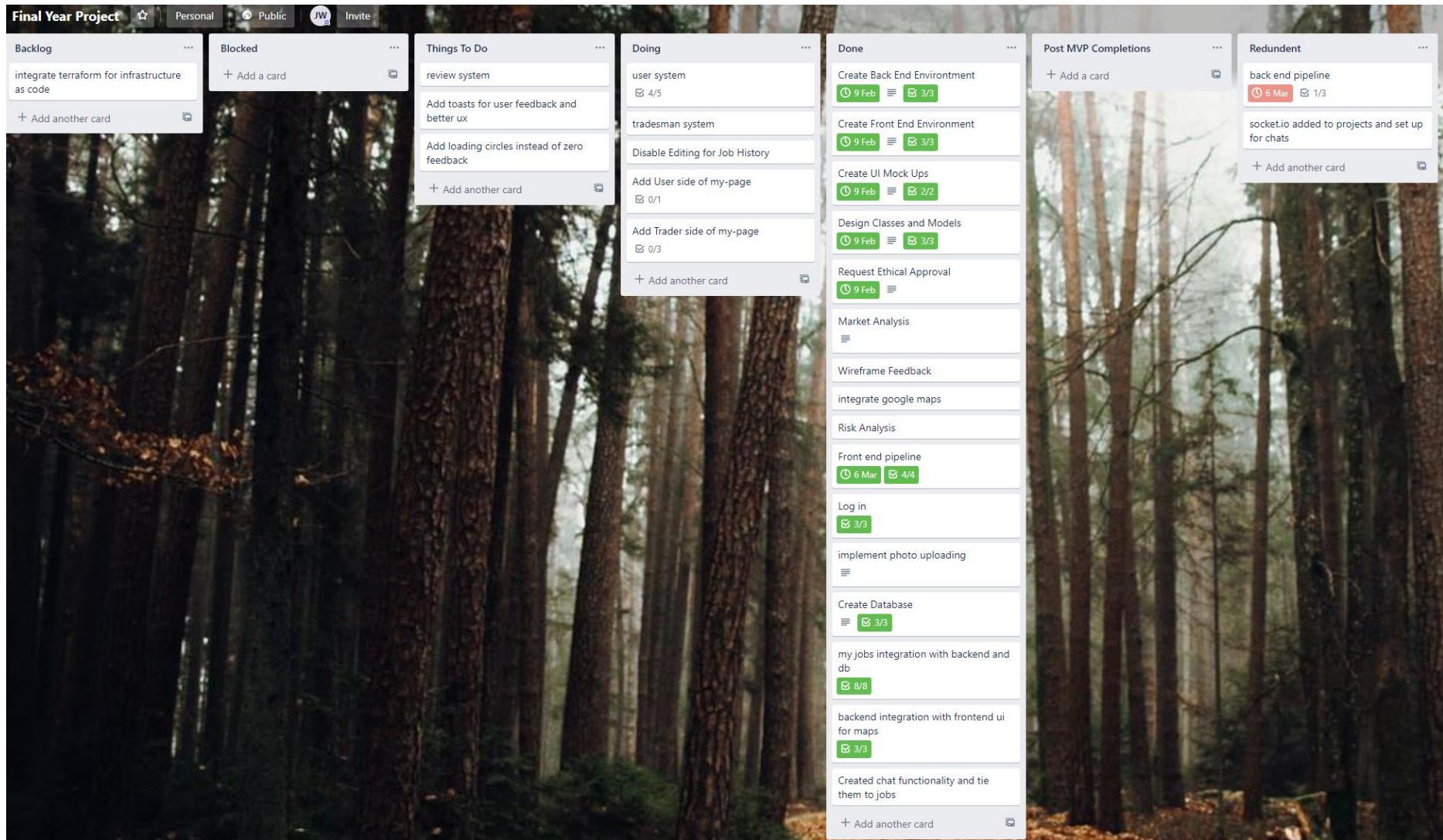


Sprint Review

Sprint 6: 27/03/20 - 10/04/20	
What went well	What did not go well
What can I change	How do I feel
<p>Hooking up the applications features to the backend to enable the logic to work is going well.</p> <p>The home map feels great to use.</p> <p>The big task of instant chat has been done and works well</p>	<p>Job history has sort of been left by the way side due to it being a sort of just dummer form of an active job.</p> <p>Had a lot of trouble getting the map to look the way I want due to the technology being used not having to great a documentation.</p>
<p>Take a look at the support of the technology you use in the app, not just the technologies you use for building the app.</p>	<p>Good, had some stints of not feeling like doing much this week due to problems, eventually diving in to the github repo for the maps component led to the answers, which in the end was nice because I had to engineer my own answers.</p>

Sprint 7

Trello 8

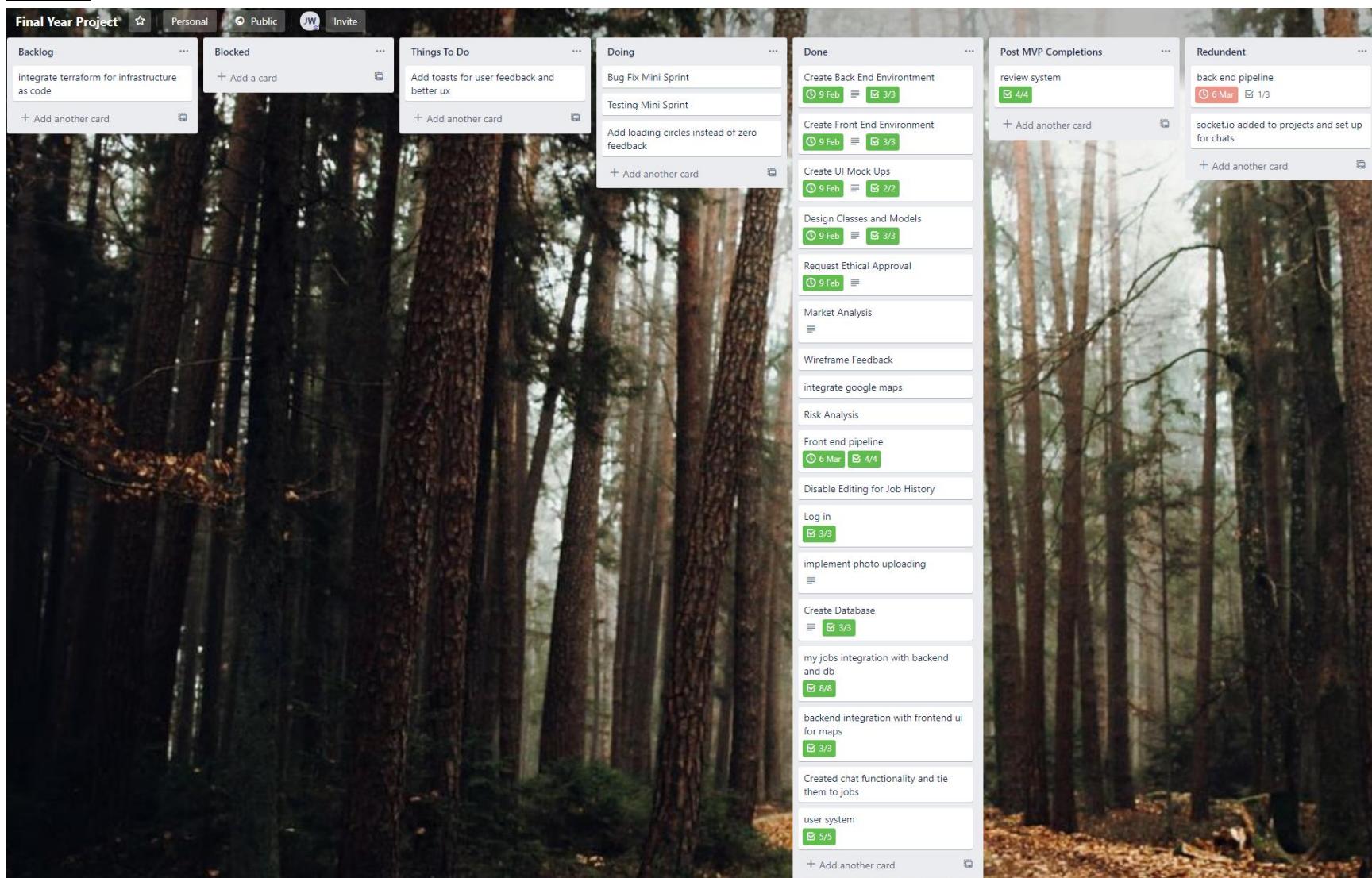


Sprint Review

Sprint 7: 10/04/20 - 24/04/20	
What went well	What did not go well
What can I change	How do I feel
All MVP functionality is getting there! Just polish being required to be done after some user testing. Really pleased with how personal details have been implemented in to "my page", It feels like the culmination of some of the stuff I've learned about html/css.	Had to rethink how users and traders logic work together, this was a bit of a mind melt but is done now. Some of the styling took longer than I would have wanted as well.
Plan the theorised key logic a bit better before beginning to develop that feature.	Really pleased, the app is demoable and I have a final product that I can be proud of just need to get the review system done to address one of my main objectives.

Sprint 8

Trello 9



Sprint Review

Sprint 8: 24/04/20 - 08/05/20	
What went well	What did not go well
What can I change	How do I feel
<p>Post MVP features have been added.</p> <p>I have learnt about using cloud functions to perform frontend logic.</p> <p>More polish added to the application got some good user feedback.</p>	<p>Cloud functions broke my pipeline and I spent far too long trying to fix them.</p>
<p>Really dive into learning the ins and out of the Circle CI tech and figure out how deploying could be done</p>	<p>Happy that I am coming to the end but also annoyed that something broke this close to the end. Just the report left so just going to cruise with it as I have other commitments like exams to work toward also.</p>

12.5 Project Setup Document

I am going to create an application that takes the accessibility of drive share apps such as uber and Ola and applies it to the trade/building sector with the aim of interrupting the market in the same way. It provides local Tradesman a platform to offer their skills and be available on a map of the users surrounding area, the user will then have a job that needs a tradesman completing, request a tradesman with a brief description of the job and an image of the site. From there the tradesman will see the review score of the customer and the required job to decide whether to take the job, also providing an “on a glance” quote. Once accepted the customer will be able to reject or confirm the tradesman based on their review score. The key idea being that it is easy to sign up as a customer or tradesman and start requesting work jobs from the comfort of your own home while also only needing a review score on the app to enable you to pursue a career in being a tradesman respectively, all being done in minutes. The application will have a desired down time of never and be fully resilient from any sort of technological failure and instantly re-deployable.

The important features of this project is that it will be fully resilient to going down with the API and PWA hosted in cloud services such as Azure or AWS, they will also have a full CI/CD pipeline allowing for resilience from software failures and for code written on a dev environment to be released to the live environment in minutes, unit tests in the pipelines for both client and server ensuring the resiliency. As well as this the infrastructure used to host the sites will be written using software such as Terraform so I can deploy the infrastructure as code, pipeline-able, rollback-able and in the case of any emergencies the environments can re-built from the code with client and server deployed in minutes instead of multiple days such as with large corporations. Hopefully demonstrating the capabilities of a DevOps Engineer to manage the whole system. As well as this I will be building a Progressive Web App (PWA) which will allow me to create a website and mobile app from once web development code base.

What I am solving with this problem is large gap in the market where there is currently no mobile/website partnership that brings the accessibility of Airbnb and Uber to the world of DIY which I think there is a dire need for. Along with the steadfast nature of a robust DevOps pipeline to keep this as an always up always available system so that even emergency callouts could be preformed on the system. This is needed because It has the potential to transform the DIY/Tradesmen industry.

Initial Risk Assessment:

Failure to Meet Sprint Deadlines – Not meeting sprint deadlines and taking too much work into the next sprint is a frequent problem in industry projects. To prevent this, I will be very honest with my estimations and meet for retrospectives with my project supervisor.

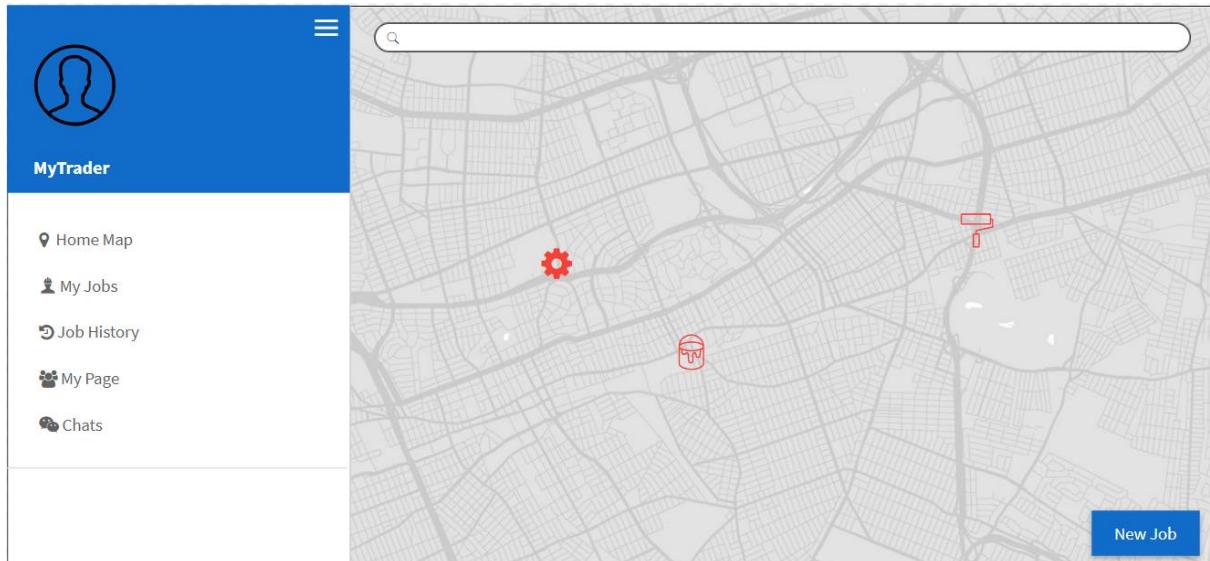
Difficulty learning the technologies – Prior to beginning the project I will spin up some trial environments to get my head around the technologies and see if there are holes in my knowledge.

Scope Change– To prevent myself being too close to the code and continuing with requirements that may be non-functional I will step back at the end of each sprint to re-evaluate my prioritisation.

Gold Plating – Strict adherence to the backlog and requirements set with an mvp nature in mind.

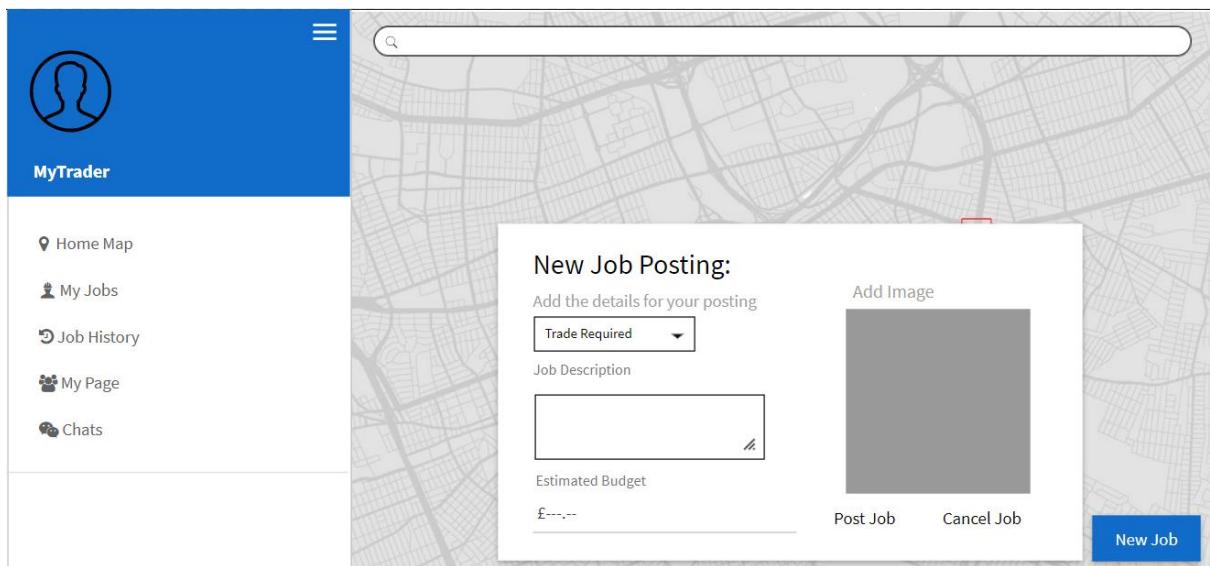
Over Promising – When initially setting requirements create a challenging task and provide stretch goals but do not over promise on delivery from the beginning.

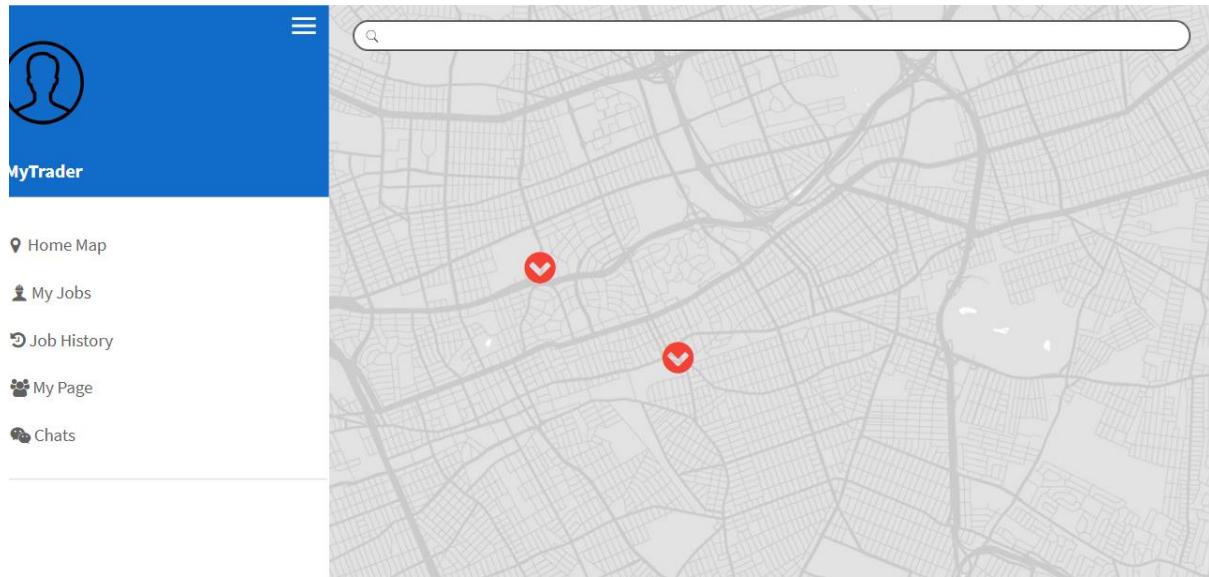
12.6 UI Mock-Ups and Wireframes



Home Map - User: This wireframe translated well into the final product, the new job button was transformed into a plus to fit nicer with the google design pattern Angular Material provides and the search bar was removed as it was not MVP.

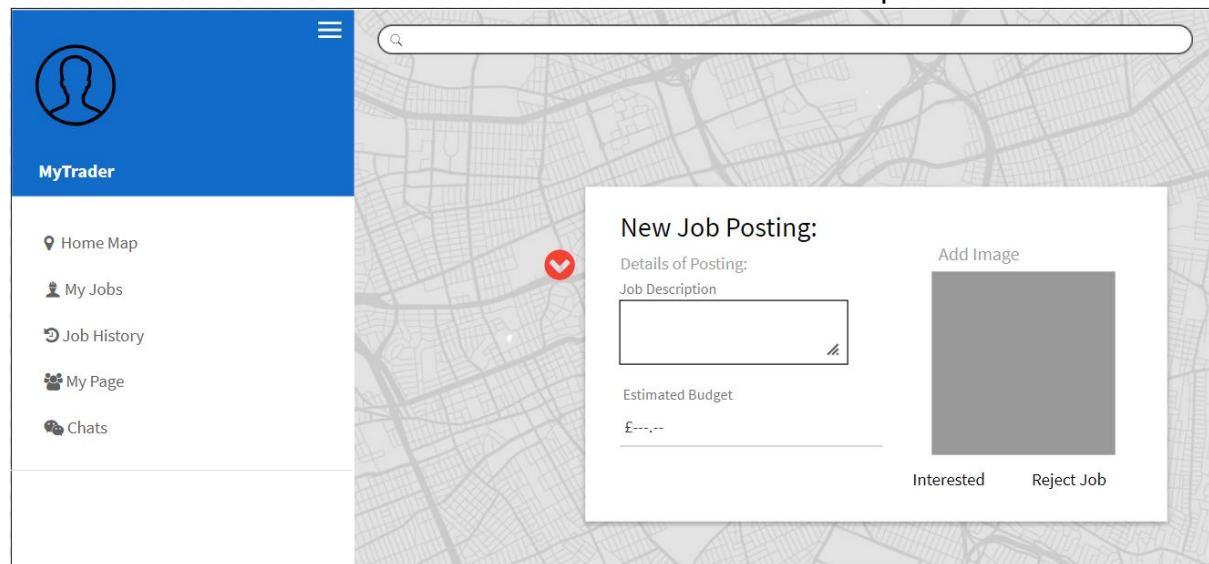
"New Job" posting has a lot more information available to it in the actual feature; this was a particular point from communicating with real traders.





Home Map – Trader: looks similar but has now got the same icons for a job that a user sees for traders. Otherwise, it would have been tough to tell the jobs apart

Clicking a job still presents some information; it is a quick snapshot, though not the full details as shown below. It also now offers full interaction options as well.

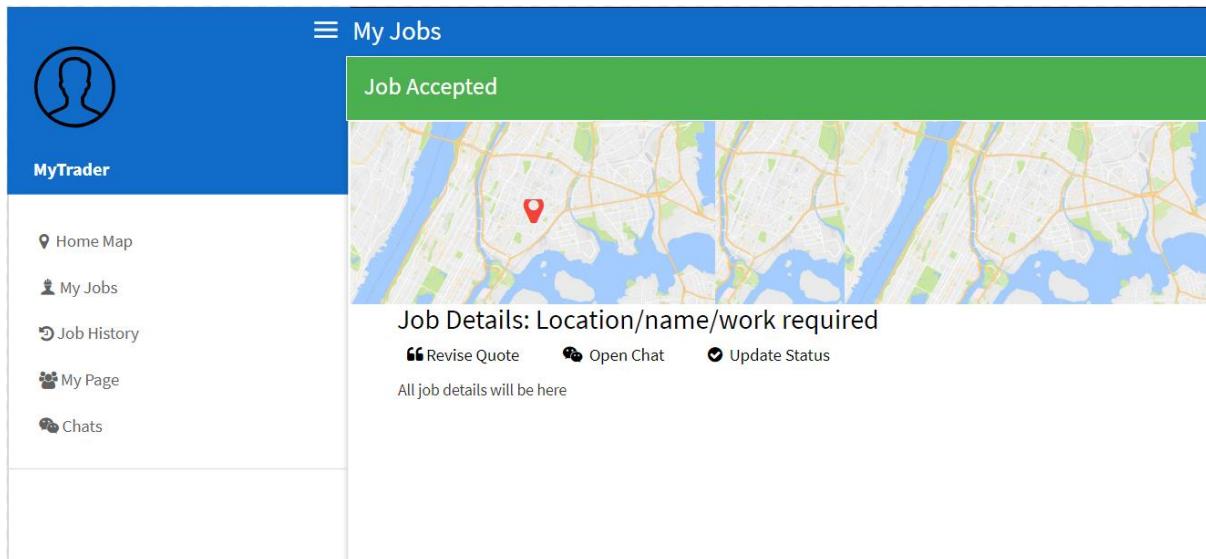


The screenshot shows the 'My Jobs' section of the application. At the top, there are tabs for 'Current Jobs' and 'Pending Jobs'. Below this is a search bar labeled 'Search for job'. Three job entries are listed, each consisting of a gray circular placeholder icon, the text 'Job Title', and a detailed description: 'Address, person info, rating score, start, finish, cost estimate'. To the right of each entry is a small blue information icon.

My Jobs – has transitioned pretty similarly.

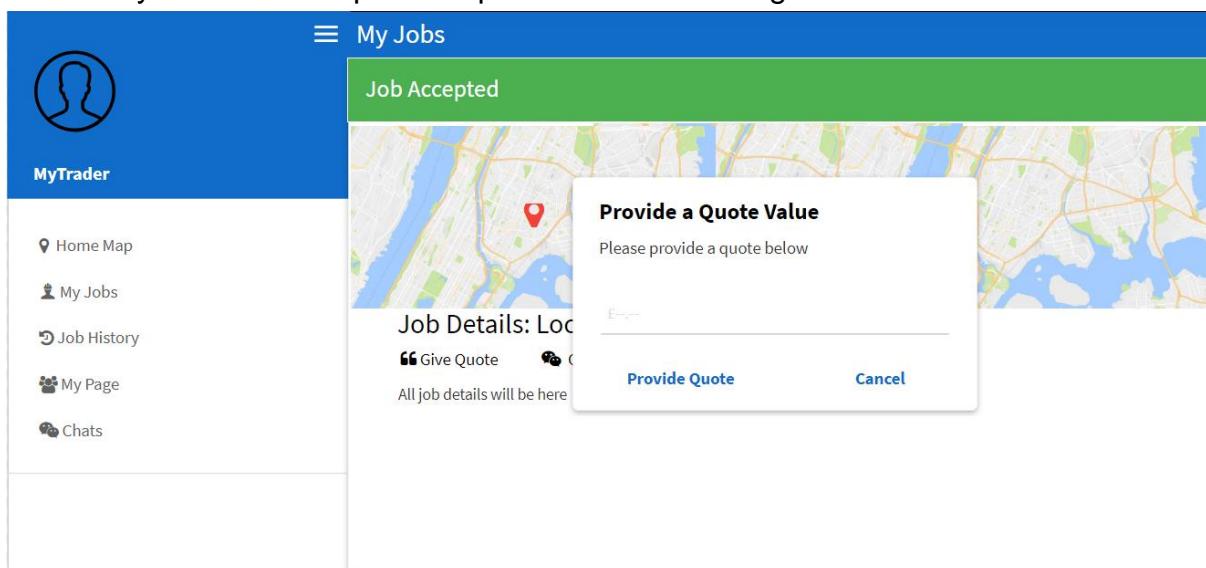
Job Details – A substantial amount of change has occurred on this page; it has gain options such as the delete/reject job functionality. It has also now got job images and company info available to see.

The screenshot shows the 'Job Details' page. At the top, there is a header with the title 'My Jobs' and a note about 'Job Status'. Below this is a map of a city area with a red location pin. The main content area is titled 'Job Details: Location/name/work required' and includes a 'Review Score: n/5'. Below the title are three buttons: 'Give Quote', 'Open Chat', and 'Update Status'. A note states 'All job details will be here'. At the bottom, there are two large buttons: a green 'Accept Job' button and a red 'Reject Job' button.



The flow of job details has transitioned nicely to the final product, status updating as indicated.

The ability to revise and provide quotes also translating well.

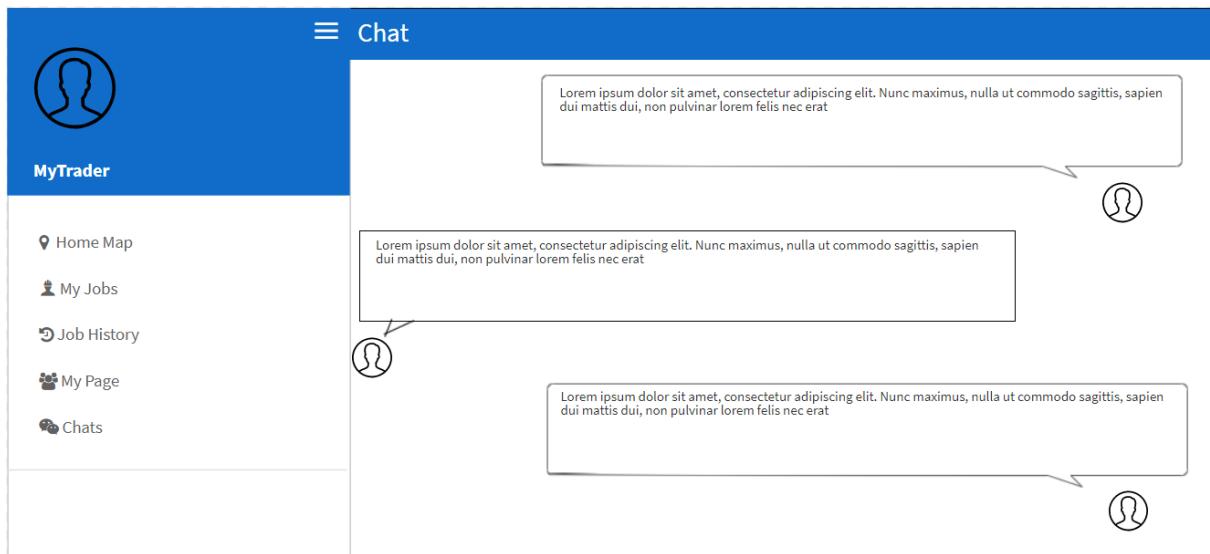


The wireframe shows the 'My Jobs' section of the application. It features a blue header bar with the 'MyTrader' logo. Below the header is a sidebar with links: 'Home Map', 'My Jobs', 'Job History', 'My Page', and 'Chats'. The main content area has a green header bar with the text 'Quote Provided: <Quote Amount>'. Below this is a map of a city area with a red location pin. A white box contains the text 'Job Details: Location/name/work required' and three buttons: 'Revise Quote', 'Open Chat', and 'Update Status'. At the bottom of this box, it says 'All job details will be here'.

The amount quoted appearing in the job status continuing to the main application as well.

The chats page is has remained the same despite one change; in the wireframe, we see actual names. However, in the actual product, the names have been abstracted out to for GDPR/Data Protection.

The wireframe shows the 'Chat' section of the application. It features a blue header bar with the 'MyTrader' logo. Below the header is a sidebar with links: 'Home Map', 'My Jobs', 'Job History', 'My Page', and 'Chats'. The main content area has a search bar labeled 'Search for job'. Below the search bar is a list of three contacts, each represented by a gray circular profile picture and a name. To the right of each contact is a right-pointing arrow icon. The contacts listed are 'Mark Wilis', 'User Details, trader or user etc', 'Joe Bloggs', 'User Details, trader or user etc', and 'Jane Doe', 'User Details, trader or user etc'.



The actual chat functionality remains mostly the same.

12.7 Auth Guard

The auth guard is essential in keeping a secure application; it will display access denied in the console if the user is not allowed to get to that URL, forcing them back to the login page.

```
canActivate(
  next: ActivatedRouteSnapshot,
  state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
  return this.auth.user$.pipe(
    take(1),
    map(user => !!user), //map to boolean
    tap(signedIn => {      //logic based on response
      if(!signedIn) {
        console.log('access Denied')
        this.router.navigate(['/login'])
      }
    })
  );
}
```

As well as this, in the routing file, it is applied to every route other than log in. In the future, some routes such as the map may be allowed to enter as a guest.

```
const routes: Routes = [
  { path: '', redirectTo: 'login', pathMatch: 'full' },
  { path: 'login', component: UserProfileComponent},
  { path: 'create-account', component: CreateAccountComponent },
  { path: 'home', component: NavBarComponent,
    children: [
      { path: '', redirectTo: 'map', pathMatch: 'full'},
      { path: 'map', component: MapComponent, outlet: 'navLinks', canActivate: [AuthGuard] },
      { path: 'my-jobs', component: MyJobsComponent, outlet: 'navLinks', canActivate: [AuthGuard]},
      { path: 'my-jobs/:id', component: JobDetailsComponent, outlet:'navLinks', canActivate: [AuthGuard]},
      { path: 'job-history', component: JobHistoryComponent, outlet: 'navLinks', canActivate: [AuthGuard]},
      { path: 'my-page', component: MyPageComponent, outlet: 'navLinks', canActivate: [AuthGuard]},
      { path: 'chats', component: ChatsComponent, outlet: 'navLinks', canActivate: [AuthGuard]},
      { path: 'chats/:id', component: ChatComponent, outlet: 'navLinks', canActivate: [AuthGuard]}
    ]
  }
];
```

12.8 Blank Questionnaire and Interview

Examples of the blank questionnaire and the interview,

Interview:

|Software Evaluation: Interview Plan

Q1 Learnability: How easy (instinctive) was it to work out how to use the software?

Q2 Usability: Once you had worked out how to use the software, how easy was it to use?

Q3 Usefulness: How beneficial do you think this software would be to the business?

Q4 Effectiveness: How do you feel the quoting system would impact your view on the pricing of a project?

Q5 Could you see yourself using the application in the future, both as a tradesperson or a consumer?

Q6 Any other suggestions, comments, or questions?

Questionnaire:

Software Evaluation: Questionnaire

Your participation in this survey will be greatly valued.
Completion and submission of this questionnaire will be taken as informed consent.
However, you are **not** required to participate. You can stop at any time.
This questionnaire does **not** assess you in any way.
Once submitted, it will not be possible to identify (and therefore withdraw) you.

Please be honest, and as specific as possible.

How easy was each facility to learn:

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>				
login	<input type="checkbox"/>				
use map	<input type="checkbox"/>				
create a job	<input type="checkbox"/>				
find that created job	<input type="checkbox"/>				
talk to a trader/user	<input type="checkbox"/>				
logout	<input type="checkbox"/>				

How easy was each facility to use (once you'd worked out how it worked):

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>				
login	<input type="checkbox"/>				
search	<input type="checkbox"/>				
find the job	<input type="checkbox"/>				
talk to a trader/user	<input type="checkbox"/>				
logout	<input type="checkbox"/>				

1. a) Overall, do you feel using the software would help you do the job and make you more accessible to customers?

Yes No

- b) Why?

.....
.....
.....
.....

2. Any other comments or suggestions

.....
.....
.....
.....
.....

12.9 User Testing Actual Data:

Tester 1:

Software Evaluation: Questionnaire

Your participation in this survey will be greatly valued.
Completion and submission of this questionnaire will be taken as informed consent.
However, you are **not** required to participate. You can stop at any time.
This questionnaire does **not** assess you in any way.
Once submitted, it will not be possible to identify (and therefore withdraw) you.

Please be honest, and as specific as possible.

How easy was each facility to learn:

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
login	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
use map	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
create a job	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
find that created job	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
logout	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

How easy was each facility to use (once you'd worked out how it worked):

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
login	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
search	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
find a job	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
logout	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

1. a) Overall, do you feel using the software would help you do the job and make you more accessible to customers?

Yes No

b) Why?
Easy to use, quick, no social interaction

2. Any other comments or suggestions
Better user feedback for selection of jobs and logging out

Software Evaluation: Interview Plan

Q1 Learnability: How easy (instinctive) was it to work out how to use the software?
Very easy

Q2 Usability: Once you had worked out how to use the software, how easy was it to use?
Even easier

Q3 Usefulness: How beneficial do you think this software would be to the business?

Will prove time efficient in finding work, but the introduction of a search bar would benefit the speed of job searching

Q4 Effectiveness: How do you feel the quoting system would impact your view on the pricing of a project?

It is a good implementation, but a guideline should be implemented for each job so users can gain an understanding of the cost of a certain project.

Q5 Could you see yourself using the application in the future, both as a tradesperson or a consumer?

Yes

Q6 Any other suggestions, comments, or questions?

The map centre could use the postcode in sign-up to centre there instead of asking for the location.

Tester 2:

Software Evaluation: Questionnaire

Your participation in this survey will be greatly valued.
Completion and submission of this questionnaire will be taken as informed consent.
However, you are **not** required to participate. You can stop at any time.
This questionnaire does **not** assess you in any way.
Once submitted, it will not be possible to identify (and therefore withdraw) you.

Please be honest, and as specific as possible.

How easy was each facility to learn:

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
login	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
use map	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
create a job	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
find that created job	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
logout	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

How easy was each facility to use (once you'd worked out how it worked):

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
login	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
search	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
find the job	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
logout	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

1. a) Overall, do you feel using the software would help you do the job and make you more accessible to customers?

Yes No

b) Why?

Due to current events, this would be very useful to evaluate a job without needing to enter a house. This also helps to broadcast my availability as a tradesman to locals

2. Any other comments or suggestions

Date is in American format.

Jobs would be easier found if the tradesman could filter to find jobs related to their craft

Software Evaluation: Interview Plan

Q1 Learnability: How easy (instinctive) was it to work out how to use the software?
It was very intuitive however initially it was confusing where the customer's job listing was located.

Q2 Usability: Once you had worked out how to use the software, how easy was it to use?
It was very easy to use once the job listings could be found

Q3 Usefulness: How beneficial do you think this software would be to the business?
Very beneficial for small companies to get their name out there

Q4 Effectiveness: How do you feel the quoting system would impact your view on the pricing of a project?
For a customer, it is useful to get continuous quotes from a tradesman as the evaluation develops and helps the customer get multiple quotes from other tradesman to create a bidding war.

Q5 Could you see yourself using the application in the future, both as a tradesperson or a consumer?
Yes, this website has room to grow to offer other contract work (such as software development) so this can be good for moonlighting.
As a consumer, I would use this website, especially if I am new to an area and don't know who to call

Q6 Any other suggestions, comments, or questions?
Notifications on when a contract has received a quote would be useful

Tester 3:

Software Evaluation: Questionnaire

Your participation in this survey will be greatly valued.
 Completion and submission of this questionnaire will be taken as informed consent.
 However, you are **not** required to participate. You can stop at any time.
 This questionnaire does **not** assess you in any way.
 Once submitted, it will not be possible to identify (and therefore withdraw) you.

Please be honest, and as specific as possible.

How easy was each facility to learn:

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
login	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
use map	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
create a job	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
find that created job	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
logout	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

How easy was each facility to use (once you'd worked out how it worked):

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
login	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
search	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
find the job	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
logout	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

1. a) Overall, do you feel using the software would help you do the job and make you more accessible to customers?

Yes No

- b) Why?
 direct contact with customers, can do it from your phone without having to call anyone

2. Any other comments or suggestions

Add trader or customer to your images, on creating account page (what kind of user are you?)

Software Evaluation: Interview Plan

Q1 Learnability: How easy (instinctive) was it to work out how to use the software?

very|

Q2 Usability: Once you had worked out how to use the software, how easy was it to use?

Very easy to use, straightforward

Q3 Usefulness: How beneficial do you think this software would be to the business?

Very useful, as you can be proactive and not have to wait for work to come to you

Q4 Effectiveness: How do you feel the quoting system would impact your view on the pricing of a project?

Makes it easier for negotiations, and if you want a job doing it makes it easier to get a quote – not needing to call anyone to come over

Q5 Could you see yourself using the application in the future, both as a tradesperson or a consumer?

I would use definitely it as a customer, not a tradesperson but can imagine it would be used often

Q6 Any other suggestions, comments, or questions?

n/a

Tester 4:

Software Evaluation: Questionnaire

Your participation in this survey will be greatly valued.
 Completion and submission of this questionnaire will be taken as informed consent.
 However, you are **not** required to participate. You can stop at any time.
 This questionnaire does **not** assess you in any way.
 Once submitted, it will not be possible to identify (and therefore withdraw) you.

Please be honest, and as specific as possible.

How easy was each facility to learn:

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
login	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
use map	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
create a job	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
find that created job	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
logout	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

How easy was each facility to use (once you'd worked out how it worked):

	very difficult	difficult	ok	easy	very easy
registration	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
login	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
search	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
find the job	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
talk to a trader/user	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
logout	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

1. a) Overall, do you feel using the software would help you do the job and make you more accessible to customers?

Yes No

- b) Why?
 The app helps with organisation as it is contained all in one app which is useful.

2. Any other comments or suggestions

It was quite difficult to know where to start a chat as a user.

Software Evaluation: Interview Plan

Q1 Learnability: How easy (instinctive) was it to work out how to use the software?

Very easy, it did not take long to work out how to use it.

Q2 Usability: Once you had worked out how to use the software, how easy was it to use?

Very easy.

Q3 Usefulness: How beneficial do you think this software would be to the business?

The app would help with organisation and to reduce time spent going to and fro with customers/traders.

Q4 Effectiveness: How do you feel the quoting system would impact your view on the pricing of a project?

It would make the process easier.

Q5 Could you see yourself using the application in the future, both as a tradesperson or a consumer?

Yes

Q6 Any other suggestions, comments, or questions?

Other than locating the chat as a user, it was very easy to use and could be a useful tool for both users and traders in the future.

12.10 Subscriptions and Memory Management

Three approaches were made to help manage the memory usage of subscriptions, these were;

1. Adding all subscription that did not have a definite termination point to an array of subscriptions on the component. Then when the component is destroyed, go through this array and unsubscribe from all.

```
this.subscriptions.push(this.companiesService.getCompanyByUid(user.uid).valueChanges())
  .subscribe((company) => this.company = company);

ngOnDestroy(): void {
  this.userSub.unsubscribe();
  this.subscriptions.forEach((subscription) => subscription.unsubscribe());
}
```

2. Unsubscribe from the user data observable when the component is no longer needed.

```
ngOnDestroy(): void {
  this.userSub.unsubscribe();
}

You, 2 months ago • uploading
```

3. If a subscription is only required for a finite amount of logic, terminate the subscription after it has completed.

```
//check postcode validity and convert to LatLong for google maps      You, a few seconds ago • Uncommitted change
var convertPostcodeSub = this.postcodeService.convertPostcodeToLatLong(this.form.get('postcode').value).subscribe(
  (data) => {
    let lat = ((data as any).result.latitude);
    let lng = ((data as any).result.longitude);
    this.newJob.lngLat = { lat, lng }
    this.newJob.completionState = CompletionState.avialable;
    this.showSpinner = true;
    this.jobsService.uploadNewJob(this.newJob).toPromise().then(() => this.dismissComponent());
    convertPostcodeSub.unsubscribe();
  },
}
```

12.11 Pipelines

A breakdown of how both the frontend and backend pipeline work.

Frontend Pipeline:

Below is a copy of the frontend YAML for the pipeline along with a brief list of the events that occur, in chronological order.

1. The workflow that dictates how the pipeline will run, the workflow below only allows the pipeline to run if it is on a master branch, and the pre-requisite steps have been met.

```
workflows:
  version: 2
  build-test-deploy:
    jobs:
      - build-app:
          filters:
            branches:
              only: master
            context: final-year
      - build-functions:
          filters:
            branches:
              only: master
            context: final-year
      - tests:
          filters:
            branches:
              only: master
            requires:
              - build-app
              - build-functions
            context: final-year
      - deploy-hosting:
          filters:
            branches:
              only: master
            requires:
              - tests
            context: final-year
```

2. Steps 1 and 2 occur simultaneously in the pipeline, installing the necessary packages for the functions and primary application. They also save their builds for quicker compilation later.

```

build-app:
  You, 3 months ago * yaml fix
  working_directory: ~/Final-Year-Frontend
  docker:
    #create docker image
    - image: circleci/node:10-browsers
  steps:
    - checkout
    - restore_cache:
        key: Final-Year-Frontend-{{ .Branch }}-{{checksum "package-lock.json"}}
    - run: npm install
    - run:
        name: "check Node.js version"
        command: node -v
    #save the node modules for use in both
    - save_cache:
        key: Final-Year-Frontend-{{ .Branch }}-{{checksum "package-lock.json"}}
        paths:
          - "node_modules"

build-functions:
  working_directory: ~/Final-Year-Frontend/functions
  docker:
    #create docker image
    - image: circleci/node:10-browsers
  steps:
    - checkout
    - restore_cache:
        key: Final-Year-Frontend-Functions-{{ .Branch }}-{{checksum "package-lock.json"}}
    - run: npm install
    - save_cache:
        key: Final-Year-Frontend-Functions-{{ .Branch }}-{{checksum "package-lock.json"}}
        paths:
          - "node_modules"

```

3. Step three runs the tests, both the protractor and the unit tests. It restores the previously saved cache to save time.

```

tests:
  working_directory: ~/Final-Year-Frontend
  docker:
    #create docker image
    - image: circleci/node:10-browsers

  steps:
    - checkout

    - restore_cache:
        key: Final-Year-Frontend-{{ .Branch }}-{{checksum "package-lock.json"}}

    - run: npm run test -- --watch=false --progress=false --browsers=ChromeHeadlessCI
    - store_test_results:
        path: test-results

    - run: npm run e2e -- --protractor-config=./e2e/protractor-ci.conf.js

```

- Step four builds and deploys the primary application to the firebase hosting solution. At this point, the cloud functions should also deploy, but they have been creating problems for the pipeline currently.

```
deploy-hosting:
  working_directory: ~/Final-Year-Frontend
  docker:
    #create docker image
    - image: circleci/node:10-browsers

  steps:
    - checkout
    - restore_cache:
        keys:
          - Final-Year-Frontend-{{ .Branch }}-{{checksum "package-lock.json"}}
          - Final-Year-Frontend-Functions-{{ .Branch }}-{{checksum "package-lock.json"}}

    - run:
        name: 'Build Project'
        command: 'npm run build'
    - run:
        name: 'Deploy to Firebase Hosting'
        command: './node_modules/.bin/firebase deploy --only hosting --token=$FIREBASE_TOKEN'
```

Backend Pipeline:

Below is a copy of the frontend YAML for the pipeline along with a brief list of the events that occur, in chronological order.

- The first step is the definition of the workflow, enforcing requirements on the backend.

```
workflows:
  build_update_deploy:
    jobs:
      - build
      - Build-Push-Image-Docker:
          requires:
            - build
      - deploy:
          requires:
            - Build-Push-Image-Docker
```

- Starts by performing an npm install to get the necessary dependencies

```
build:
  description: Install npm
  # machine option runs your jobs in a dedicated, ephemeral VM that has the following specifications:
  machine: true
  steps:
    - checkout
    # Install node
    - node/install
    # Install npm
    - node/install-npm
    # Download and cache dependencies
    - node/with-cache:
      steps:
        - run:
          name: Install application dependencies
          command: npm install
        - run:
          name: Run tests
          command: npm test
      # Save cache
      cache-key: package.json
      # Ignore non-checksum cache hits
      use-strict-cache: true| You, 3 months ago • updated yaml for circle to includde kuber and
```

- It then builds the docker container image, while allowing the version number to be changed on the build.

```
Build-Push-Image-Docker:
  description: Build and push image to Google Container Registry
  machine: true
  steps:
    - checkout
    - gcr/gcr-auth
    - gcr/build-image:
      image: tradesman-cd-gke
      tag: "v2" #Change version number e.g to 'v3' when updating application
    - gcr/push-image:
      image: tradesman-cd-gke
      tag: "v2" #Change version number e.g to 'v3' when updating application
```

- Then the deploy to a Kubernetes cluster in the google cloud platform hosting environment.

```
deploy:
  description: Deploy application to Google Kubernetes Engine
  machine: true
  steps:
    # Install `gcloud` and `kubectl` if not already installed.
    - gcp-gke/install
    # Initialize the `gcloud` CLI.
    - gcp-gke/init
    # Update a deployment Docker image.
    - gcp-gke/rollout-image:
      deployment: tradesman-api-cluster
      container: joe-wood-backend
      image: gcr.io/tradesman-finalyear-api/tradesman-cd-gke:v2 # change version when updating
```