



HERIOT-WATT UNIVERSITY

ROAD2WORK

GROUP PROJECT

SOFTWARE ENGINEERING

Stage 3 - Final Product

Members:

Rory MALCOLM
Daniel WAGHORN
Vitalija ŽILINSKAITĖ
Stuart MARPLES
Mariana FANTINI
Stoyan NIKOLOV
Roman MASON

Supervisor:
Dr. Fiona MCNEILL

April 1, 2016

Contents

1	Introduction	1
2	Marketing Analysis and Strategy	2
2.1	Executive Summary	2
2.2	Situation Analysis	2
2.2.1	Overall Objectives	2
2.3	SWOT Analysis	3
2.4	Porter's Five Forces	6
2.5	Implementation of Marketing	7
2.5.1	Marketing Mix	7
2.6	Marketing Analysis	8
2.6.1	PESTLE	8
2.6.2	Marketing Analysis	9
2.6.3	Target Market	9
2.6.4	Stakeholders	9
3	Final Application Development	11
3.0.1	Courses data	30
3.0.2	Job Data	31
4	Final Usability Evaluation	38
4.1	Testing Throughout Development	38
5	Project Evaluation	49
5.1	Organisation	49
5.2	Implementation	51
5.3	Product	51
5.3.1	Functional Requirements	51
5.3.2	Non-Functional Requirements	53
6	Guides	56
6.1	User guide	56
6.2	Operations and Maintenance Guide	65
6.2.1	Admin Guide	65
	Appendices	70
A	Usability Questionnaire	71
B	Project Commit Log	74
C	Project Diary	78

1. Introduction

In the final stage of the group project we completed the remainder of the development. This report outlines the company marketing strategy for the final product as well as reviews the final stage of the development. It covers how well the group worked as a team and how the processes we set up worked in practice as well as any remarks the group as a whole has now that the development is complete.

In addition to the above a more detailed Usability Study conducted on the end product is included which aims to highlight areas in which we could improve the interface going forward. An overview of the application design and implementation is also included in order to give an understanding of the various top level components that the system comprises of and how these are brought together and orchestrated to create the final product.

2. Marketing Analysis and Strategy

2.1 Executive Summary

Road2work was set up at Heriot-Watt University in order to provide students with a way of finding their preferred career path and their preferred destination. To do this we created an online career path finder in order for students to achieve this with no hassle and in the quickest time possible.

Road2Work have set out a number of marketing goals that will be achieved for the site:

1. Build a recognisable brand
2. Increase awareness of Road2work within Universities, Colleges and other educational institutions
3. Reach a certain number of members on the site

To make these goals achievable Road2work have devised a marketing strategy which considers both their end users and their competitors. We at Road2Work put the end user (*Members*) at the forefront of what we do. To make it easy for our users to find out about our site we will advertise it within the universities and provide webpage links on the university websites and thus making it possible for us to establish members quickly and easily . To widen the range of our audience we will use the search engines and Facebook pages which in turn can help us build a recognisable brand.

2.2 Situation Analysis

2.2.1 Overall Objectives

Road2work strive to provide a stress free way of finding the correct career path for members at the click of a button. We also pride ourselves on becoming a company which is associated with putting the customer first. This is why we have designed objectives which we aim to reach, not only to do with the customer's satisfaction but also maximising efficiency and ensuring the well-being of employees.

Overall goals:

- Customer Satisfaction

Road2work want to provide a high quality and fast service to the users of road2.work. In order to do this we ensure that the GUI is easy to understand and responsive which ensures it displays well on a wide range of devices.

- Maintaining the site

Road2work aim to constantly update and maintain the site. By maintaining the site and keeping it up to date with all the universities, jobs and courses it will allow ease of use for the users. In turn this will increase the number of members using the site as results will be current and accurate.

2.3 SWOT Analysis

Road2work have used a SWOT analysis to outline our strengths and weaknesses along with the opportunities and threats we may face while executing the project.



Figure 2.1: SWOT Analysis

• Strengths

We feel that the strengths we carry within the group such as having an array of different skills and a quick way of communication will help us and the Road2work project to flourish and become successful. Not only does the project require our communication between ourselves on a daily basis but it also requires us to combine our skills to create such a well-designed and easy to use site which is available on multiple platforms.

• Weaknesses

Although the project consists a group of people who together have a wide range of skills the group still does not have a huge amount of experience which could be seen as a weakness. Even though there is not huge amount of experience when working together as a group we feel that it can be built upon throughout the project cycle. We also feel that even though we don't have a huge amount of experience within the field we can grow and become a market leader.

• Opportunities

With Road2work being an emerging start-up company the opportunities for us are endless. We have the opportunity to grow into a recognised brand throughout the UK and in time become recognised globally throughout the sector. We also have the opportunity to grow as a team and employ more people to bring fresh ideas thus giving us more chance of attracting new members to the site.

• Threats

The threats in which can arise within the company are sites that are already on the market or other sites that are coming into the market at the same time. Not only this but there is a threat that workers may leave the company taking information and knowledge gained from the company and use it to help other sites.



Figure 2.2: SWOT - Strengths and Opportunities

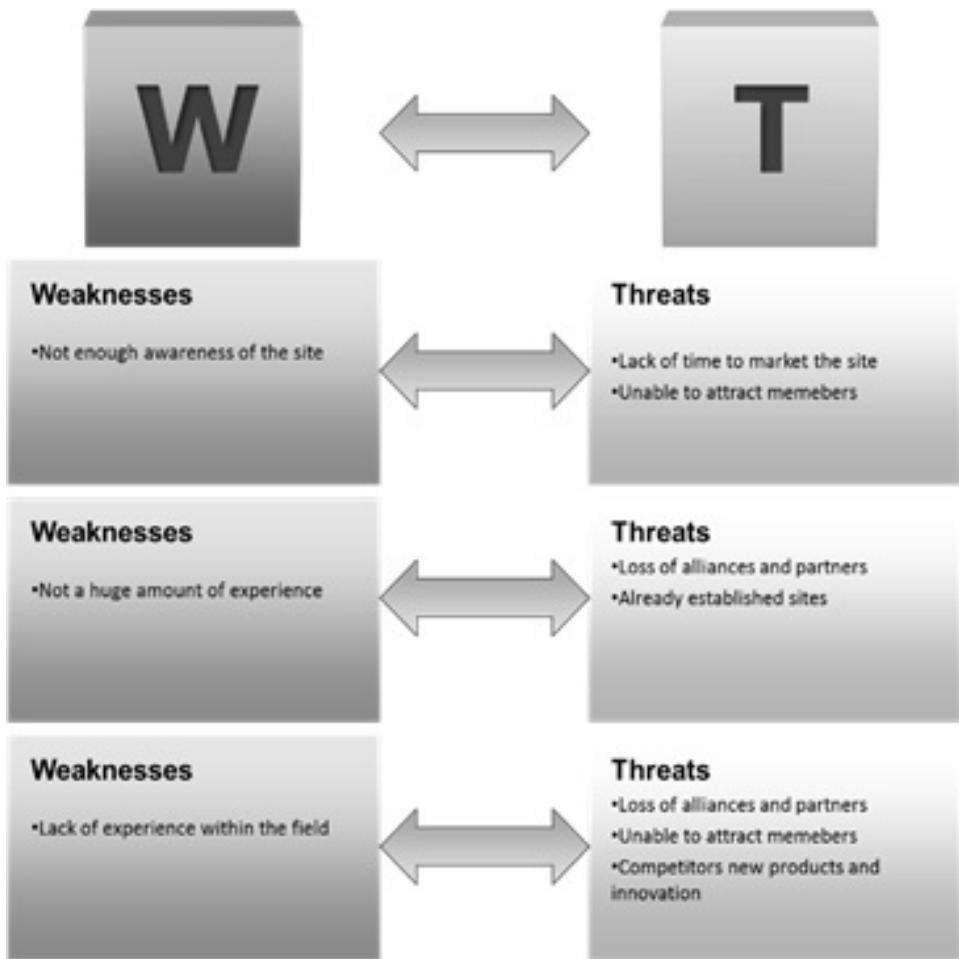
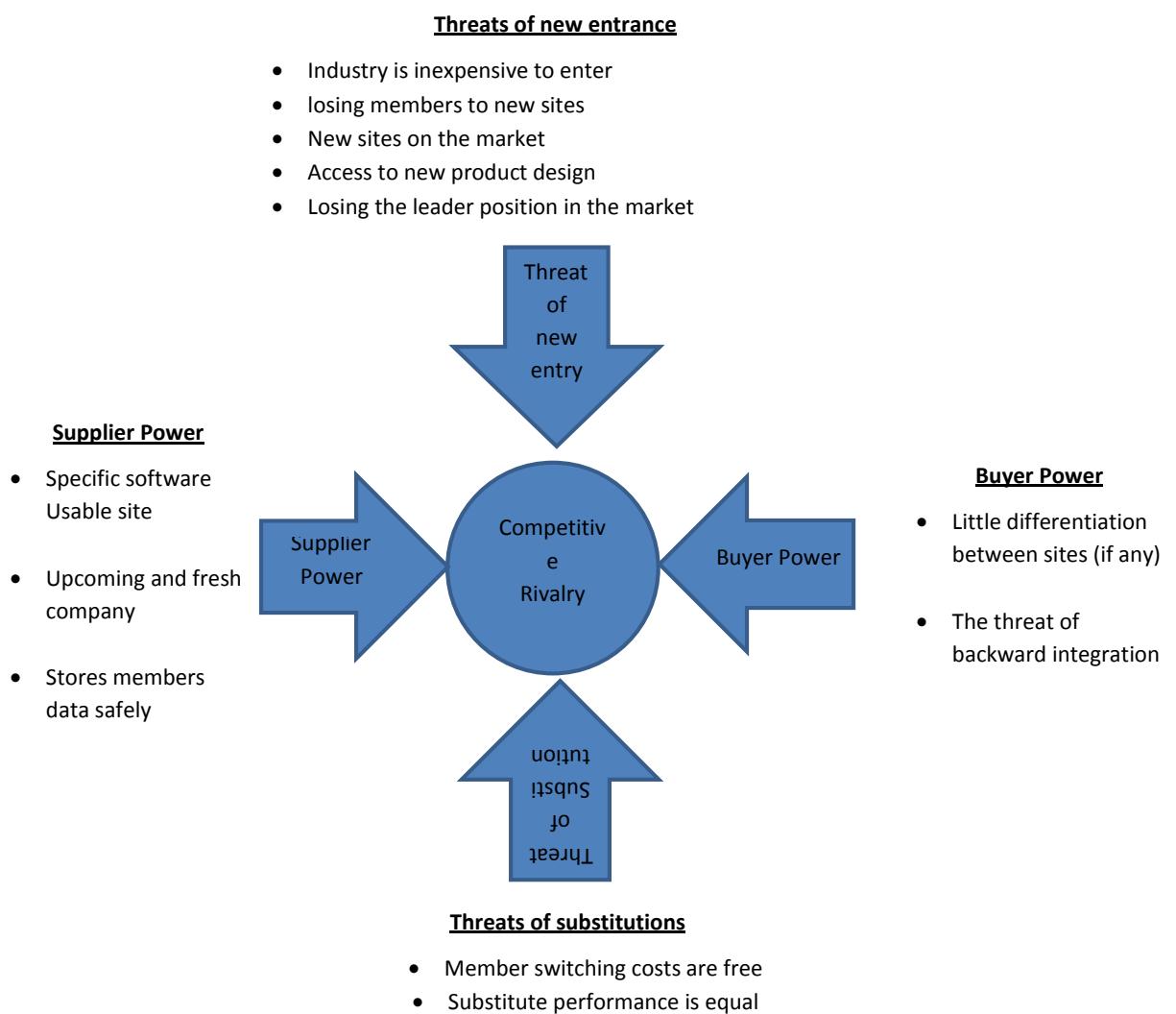


Figure 2.3: SWOT - Weeknesses and Threats

2.4 Porter's Five Forces

Within the marketing plan Porter's five forces model is a key element because it analyses the level of competition within the industry and gives a clear outline of what to expect. Not only does it analyse the market it will also develop a clear business strategy from which Road2work can follow.



2.5 Implementation of Marketing

The next phase of our marketing strategy is to define the way in which Road2work will achieve the goals in which to implement the promotion of the site.

In terms of the marketing plan and the structure it should be constantly reviewed and evaluated as the months go on to ensure that the targets are being met. In reviewing the plan and current situation in this way it is possible to implement corrective action if needed and make tactical decisions to adapt to a constantly evolving market. The way Road2work will implement the marketing strategy is by:

- **Promoting on social media**

By promoting on social media the company can reach a wide range of potential new members at once with quick and easy access to the site. Social media offers a reasonably inexpensive marketing platform whilst having the potential for a very large exposure to potential customers.

- **Continuous search engine optimisation**

Using search engines such as Google and Bing can also bring Road2work a lot more publicity by placing ourselves at the top whenever searched. Continual analysis of search analytics can help identify strategic search queries and niches which can be exploited to increase the amount of customers gained through organic search traffic.

- **Involved with customers on social media and blogs**

Engaging with customers on social media and blogs will help Road2work to establish a connection with their members which will increase the customer satisfaction of the business and gain insight into their needs and wants.

- **Industry Conferences**

Road2work plan to demonstrate at industry conferences in order to advertise the various reseller opportunities available as an additional part of our business model. This can create a new revenue stream and provide an enticing offer to universities, recruitment agencies and more by being able to use a whitelabelled and standalone version of our product for their own internal operations.

- **University boards**

Advertising within the university will be the best and direct way of marketing the website to the students as they will always be present at the university and will see advertisements relevant to them.

- **University website advertising**

Advertising on the universities websites will be another way of marketing the site where students can easily gain access to it and where the site will be visible to the potential members.

- **Industry Publications**

In order to raise awareness of our whitelabelled solution available to resellers, schools, colleges and careers services Road2work plan to advertise in relevant industry publications and magazines in order to effectively reach target markets.

By regularly reviewing the ways we have chosen to market the site it will allow us to get an idea of which ones are successful and how to continue with future efforts.

2.5.1 Marketing Mix

Within the marketing plan it is Important that it contains the marketing mix which consists of the '4 Ps': Product, Place, Price, Promotion. The reason for using this analysis is because it helps determine if the demand for the product is high enough to release it onto the market and also determine the place of the product, where in our case it will be online. Not only does it help with these two elements, it also helps with knowing how much you should price the product for and also the promotion of it to the members.

- **Product**

When creating the product it is essential that the product has a use in the market and will bring a value to the user. The site Road2work has created gives school leavers and university students a clear way of planning their future and will also aid them in making decisions that can be sometimes hard to make, automatically making the product useful within the market. When researching competitors in the field we found that none of the sites had a function where by you could tailor course suggestions based on your Interests and grades, so when creating the site we ensured this was a key feature. By creating this functionality for the site this allows our users the gain a more realistic experience when using our site than others. Not only does this put us above our competitors in terms of features but it also gives the user a better experience as this makes the site easier to use and more personal. When looking at the overall design of our site it matches our target market's desires (school leavers between the ages of 17-18 and university students) such that it is an easy to use site.

- **Place** Another important feature within the marketing mix is the placement of the product. As our product is a website it will be solely based online. As a result of this making it accessible whenever to our members is essential. It will also be accessible as it is a fully responsive site on all platforms. With our site coming into the market it is essential that we enter the market knowing our competitors and target market as they will have already been in the market longer than us and as the internet is an ever-changing and a growing market which we feel we need to work to get established quickly. We will achieve this by working to place our site at the top of search engines such as Google and Bing as well as also be regularly active on social media sites such as Facebook and Twitter.

- **Price**

A vital part of any new product is getting the price strategy correct as soon as it hits the market and researching the competitors pricing. However with the research which was put in with our product other sites such as www.schools.com and www.planitplus.com offered the service for free. We had to take this into account when deciding on whether or not to charge for our services. Road2work will be a free service to our customers as we felt that this would appeal to a larger volume of customers. By choosing the same pricing strategy as our customers we have chosen the “Competitive pricing strategy” which keeps our product within the competitions price range as we did not want to price ourselves out of the market.

- **Promotion**

Promotion is one of the most important aspects of the marking mix, if done well this will bring a lot of attention to the product and make people aware of it before the release. There are numerous ways in which the promotion of Road2work can be executed. We will use the direct marketing approach by promoting the website on university, college and school websites we can obtain publicity to a lot of potential members in one place. Another way we can market the site is by the use of social media, we can use the sponsored ads bar on the side of Facebook and also use a Facebook page with a paid ad campaign.

For marketing to potential business customers such as university careers service we will also employ a direct marketing approach via targeted advertising in industry publications.

2.6 Marketing Analysis

The Marketing analysis section of the report concentrates on the different analysis functions that can be carried out when first marketing the product to new customers, creating a PESTEL analysis will help analyse the business environment as a whole. PESTLE is an acronym which stands for:

- **Political**
- **Economic**
- **Social**
- **Technological**
- **Legal**
- **Environment**

By analysing these six factors at the right time it will help Road2Work make better decisions at the right moments and also help us judge what may happen in the future by looking at the present. Below is the PESTLE analysis for Road2work.

2.6.1 PESTLE

1. Political

Road2work need to consider a lot of political issues that may affect them as a company with things such as government changes and trading policy amendments happening all the time as society is ever changing. We at Road2work will make sure we are constantly complying with the rules and regulations not only as a company but also as a product on the market if there are any changes. Taxes are set at a fixed rate by the government and hence are outwith of our control, which will affect us as a company and also our site (the product). Applicable taxes would include income tax and corporation tax. Furthermore because we would present our product at schools around Scotland we would make sure that it is presented within a reasonable manner where we would not push young adults in schools to sign up if they did not want to. Road2work would also take into consideration public opinion and if the public has any issues with our product we would work closely with them to solve any issues.

2. Economic

The economic factors of PESTLE are similar to those of the political views in ways where by not only do they affect the organisation but they also affect the product in turn. Although the economy is still volatile, this is not an issue for this product as it has been funded by Lockheed Martin where they already obtain government funding for this sort of project to help young or unemployed individuals whom can benefit from assistive tools such as our product.

Due to the economy not being at its strongest point and the unemployment rate is quite high this could raise potential number of users of the site which may have a positive impact on the economy as our site has the potential to give people a better direction for their future.

3. Social

Living in a society which has a high demand for people with degrees suggests that companies look for only the most skilled workers within the field, meaning that if a person in question wants to gain their preferred job they require a university degree or a qualification of the same standard. In addition to that Scotland has paid tuition fees which helps make this service more appealing regardless of social background.

4. Technological

Technology is ever changing within the 21st century so it is important that Road2work monitor and keep up-to-date with the changes that are happening within the technology sector. Our product will be at the forefront of the different platforms which are available making our site user friendly, feature rich and accessible. Other technological advances will consist of software updates to the different platforms which means we will have to monitor the software closely to keep our site usable and secure.

5. Legal

Different legal requirements need to be followed as we as a company don't want to under any circumstances come into any legal trouble or be liable for litigation. There are a variety of different legal requirements we need to follow, not just for the company but for the product also. As the information we are collecting is personal data, which also means it is confidential, we need to have data protection laws in place which would allow us to have this data and also the evaluation on a regular basis of the intellectual property law would need to be carried out. Relevant agencies such as the Information Commissioner's Office (ICO) will need to be notified of the nature of our data processing activities.

6. Environment

As our product is online not all the environmental issues will need to be considered in terms of our product however as a company this is not the case. Promoting our brand will mean we need to use posters, leaflets and business cards all of which use paper or card, even though this is recyclable we can reduce the amount of paper we use by promoting more on social media which would not harm the environment, making us a greener and more economical company.

We will also endeavour to host our software application within carbon neutral datacenters.

2.6.2 Marketing Analysis

When analysing the market it is mainly focused on the competitors already established within the market. Being a new company coming into the market it is essential that we analyse the market and outline potential threats which include similar sites and companies.

The potential customers we have outlined to be a threat towards our company are www.schools.com and PlanItPlus. Although www.schools.com does not have a timeline view of the desired goal we still feel it is a threat towards our company. *Planitplus* are the biggest threats to Road2work as they have the closest product to ours in the current market.

2.6.3 Target Market

The target market we are aiming to reach is people of all backgrounds that need guidance or are just stuck on a certain stage of their career. We aim to aid these people with suggestions about what they can do with the qualifications they have and how to get to the career they want.

2.6.4 Stakeholders

Our product will have to take into considerations a lot of different stakeholders when developing through to launch of the application as it is important to keep all stakeholders happy in order to launch a successful product. In this case especially customer and communities are of particular interest as they are a big aspect of the application's target market therefore we have to make sure they are happy with what we produce. We will identify four distinct groups of stakeholders which are customers, organisations & communities, employees, and financial groups as these four are considered as key groups once by Fill (2006). As it's important to identify all the stakeholders involved in the project

and not leave anyone out.

Customer

- Teachers and mentors
- School leavers
- University students
- Lockheed Martin
- Careers Services
- Universities & Colleges

Organisations & Communities

- UCAS
- SQA
- Unistats

Employees

- The workforce employed by Road2work
- Potential contractors for temporary work

Financial Groups

- Scottish Government

3. Final Application Development

Application development for the final stage continued from where we left off in terms of functionality from the Stage 2 Demo. At this point we had most of the tooling in place for completion such as IDEs, Virtual Machines and Version Control. In terms of features we had around half of the database migrations completed as well and a fully implemented authentication system which supported distinct user roles which would be essential for the project.

The branding had been established for the product when the marketing website (available at <http://meet.road2.work>) was created during the previous stage. Views and templates for the web application had also been created at this stage which incorporated this branding to keep a consistent company identity, which were already integrated with the PHP framework we were using which was *Laravel* [14]. Each group member had a copy of the local development site set up on their *Vagrant* [25] virtual machine ready to continue. The administrative back-end was also partially in place and had the user management facility completed.

System Overview

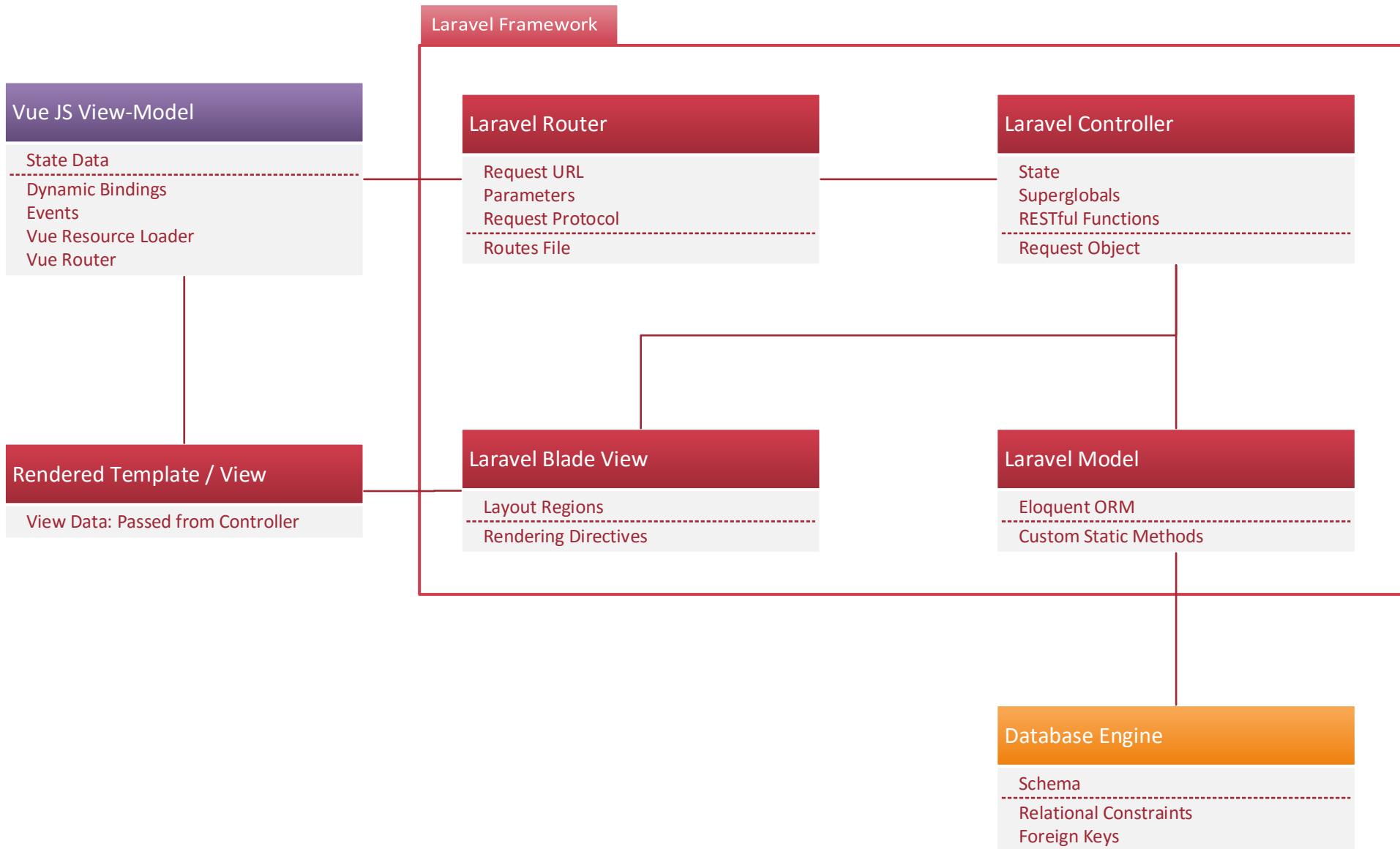
The system comprises of three main components: Laravel which acts as the back-end application framework and persistence layer, which is coupled with a MySQL database in our deployment setup. The third component is the use of Vue JS for our reactive front-end framework which offers seamless two-way integration with the back-end.

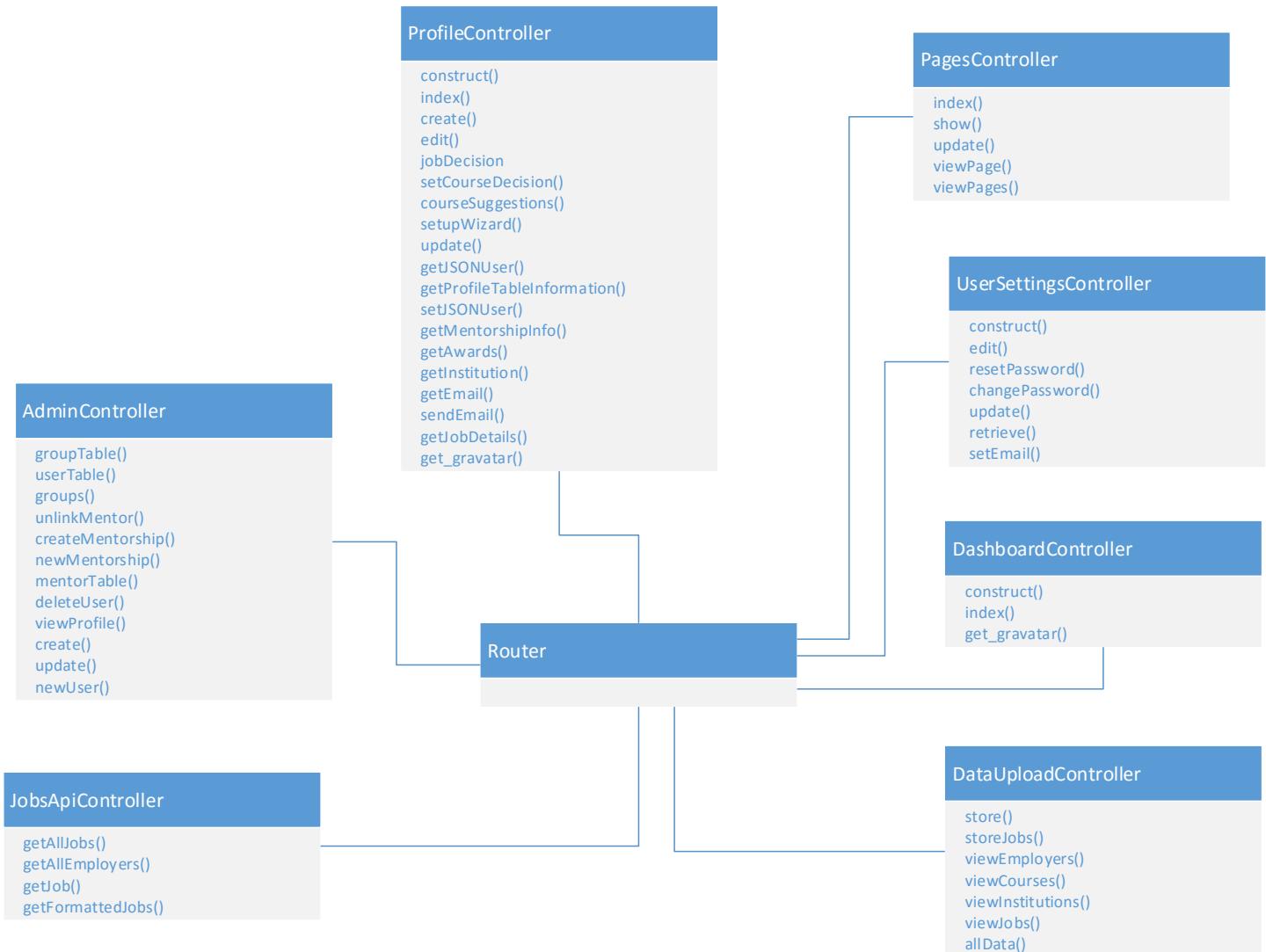
The diagram overleaf illustrates the associations between high level system components and their composition.

Vue JS components can make their own requests and can also receive data from the server without additional views being rendered. The view templates are typically only used as a basis for the initial output on which Vue JS binds to and runs upon.

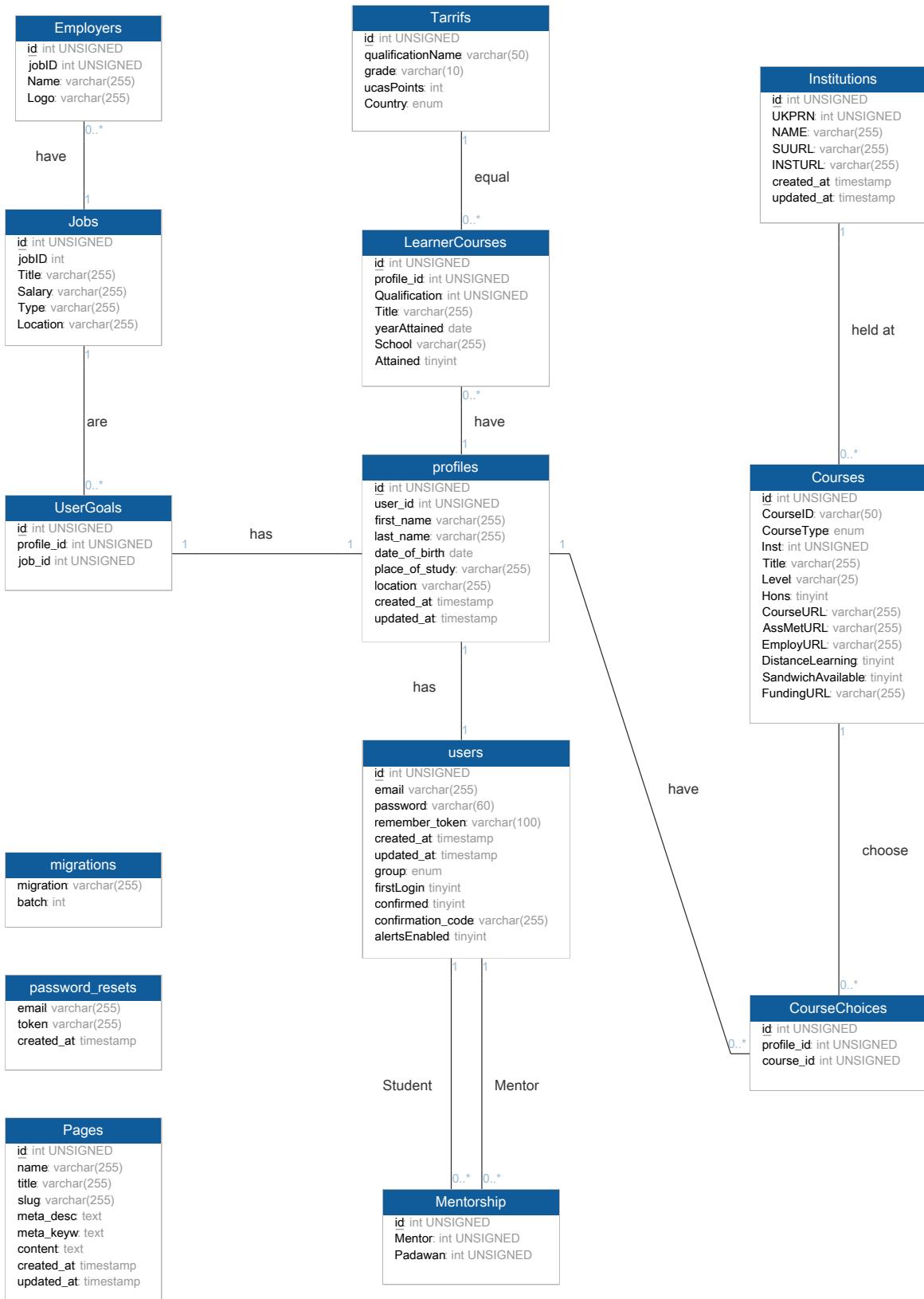
The database engine acts as the primary storage component for Laravel where data is concerned. The schema implements much of the business logic in its design, an effort which when combined with the Eloquent ORM makes the representation of entities within code simple and coherent. Laravel also makes use of local filesystem storage for temporary storage of files such as data upload files rather than reading them on-the-fly via the upload stream.

The second diagram outlines the controllers used within the application, each of which is key in organising the business logic and representing it in a sensible and maintainable structure.





Database Schema



Our database schema as illustrated by the ER diagram on the previous page is as minimal as we could make it whilst maintaining an expressive structure that effectively modelled the key relations in our application.

We made use of linking tables for relations such as course and job choices as well as mentor relations to ensure minimal data duplication and no transitive dependency. These linking tables also make it easy to manage relationships, for example to add a new mentorship relation only the two user IDs are required. The simpler data representation requires more effort in development since multiple database joins are required to obtain the necessary data to show additional information such as the names of users for display in the user interface.

Not all of the tables are crucial to the application logic, for instance there are standalone tables such as the `Pages` table which contains only data used for rendering the content managed static pages.

In order to create this schema a user simply has to run the database migrations via the `artisan migrate` command.

Final Interface Design

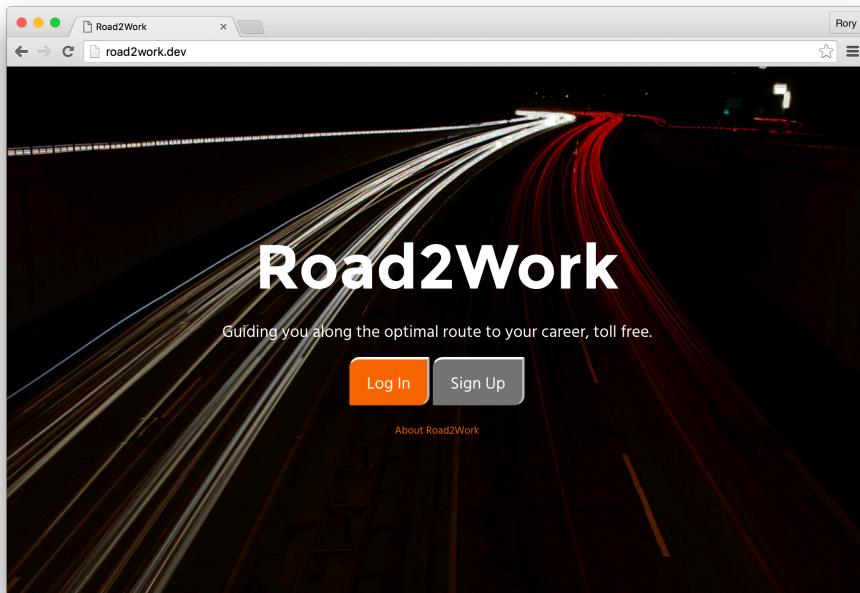


Figure 3.1: The homepage.



Figure 3.2: The homepage mockup.

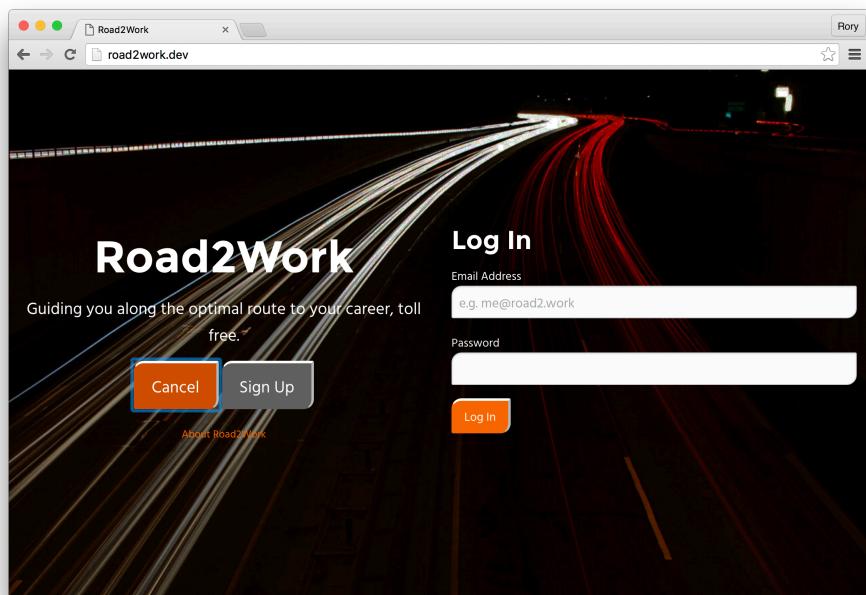


Figure 3.3: The login screen.



Figure 3.4: The login screen mockup.

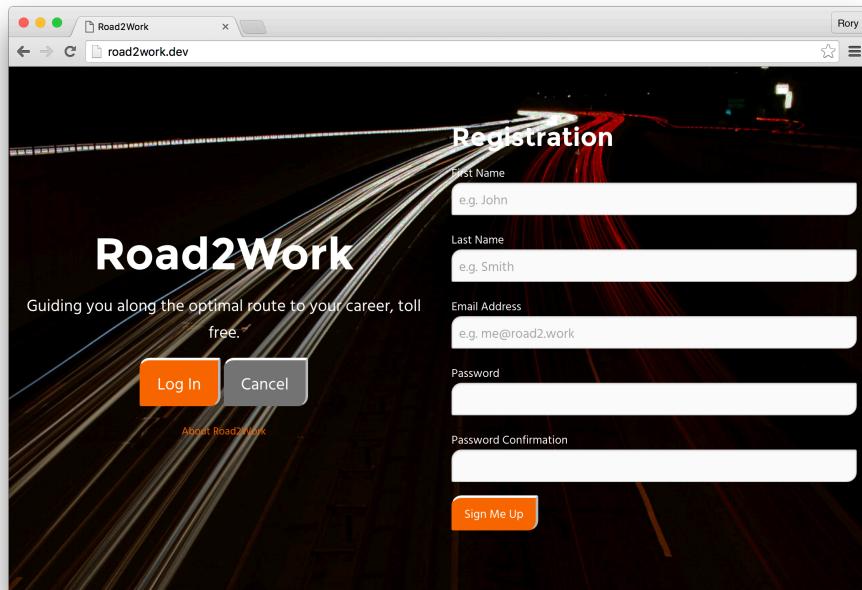


Figure 3.5: The new user screen.

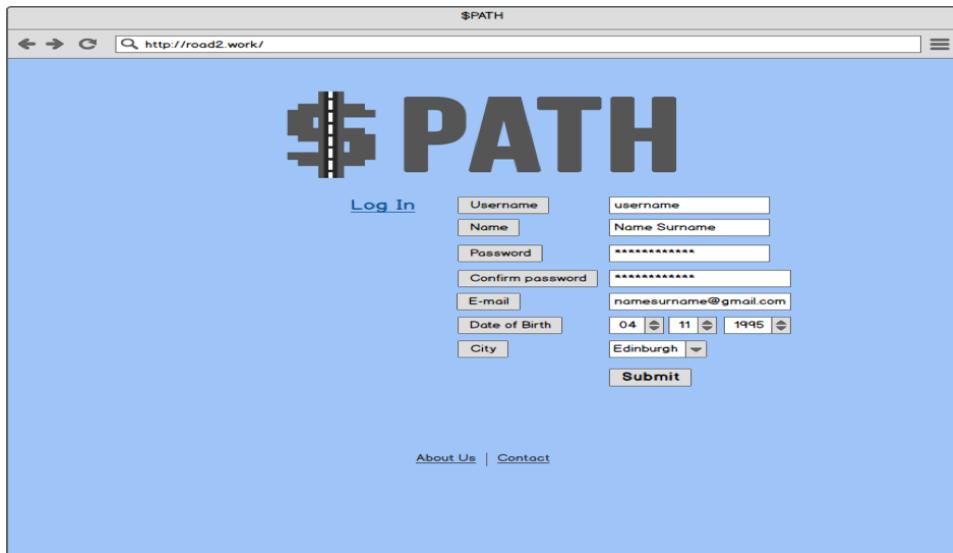


Figure 3.6: The new user screen mockup.

The homepage, login and register process for the most part remains unchanged. There have been some abstractions from the register form into the profile set up page as we felt this was more relevant there as opposed to on their initial sign up. The pages have also been edited to fit better visually with the branding of the rest of the site.

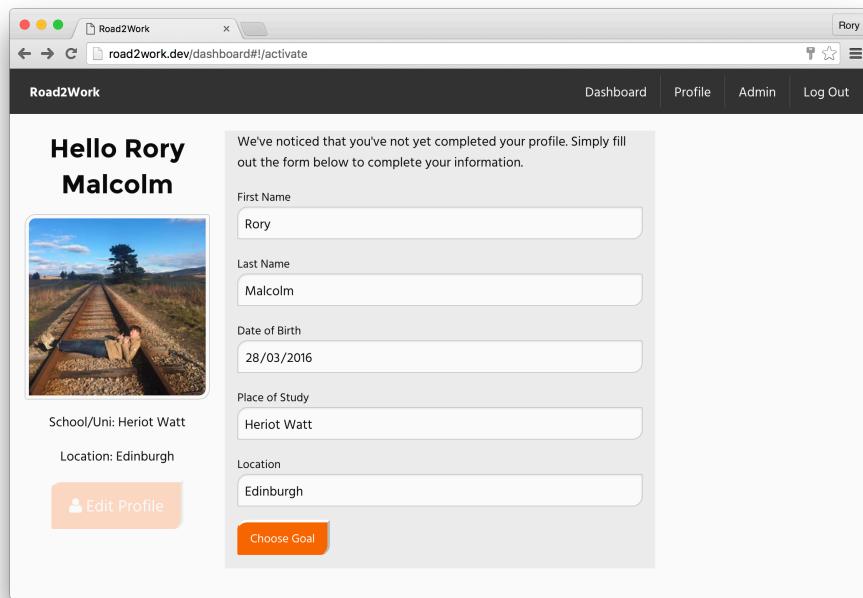


Figure 3.7: The beginning of the user set up stage.

This is the beginning of the user set up stage, they are asked to enter their information in order to display it on their profile, this adds a level of personalisation and familiarity with the system for the user.

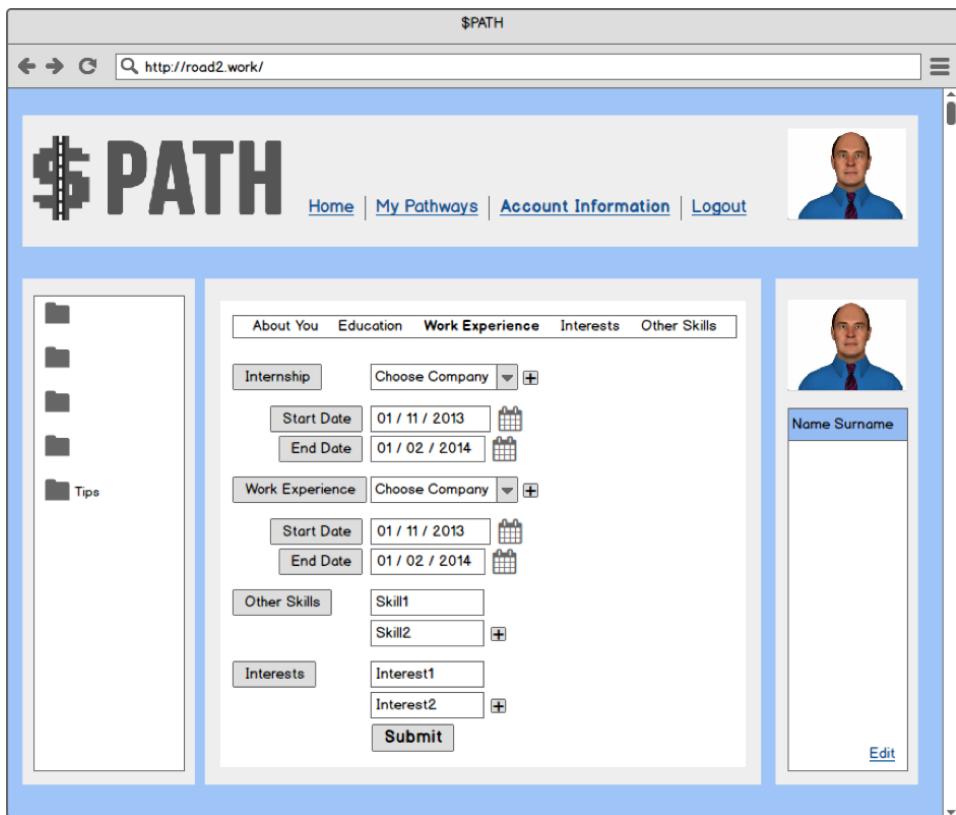


Figure 3.8: The user profile edit mockup.

This is the initial mockup we were working from, while it is similar to the end result a lot of the information fields have been split up over multiple pages as we found this was less overwhelming and provided a better user experience.

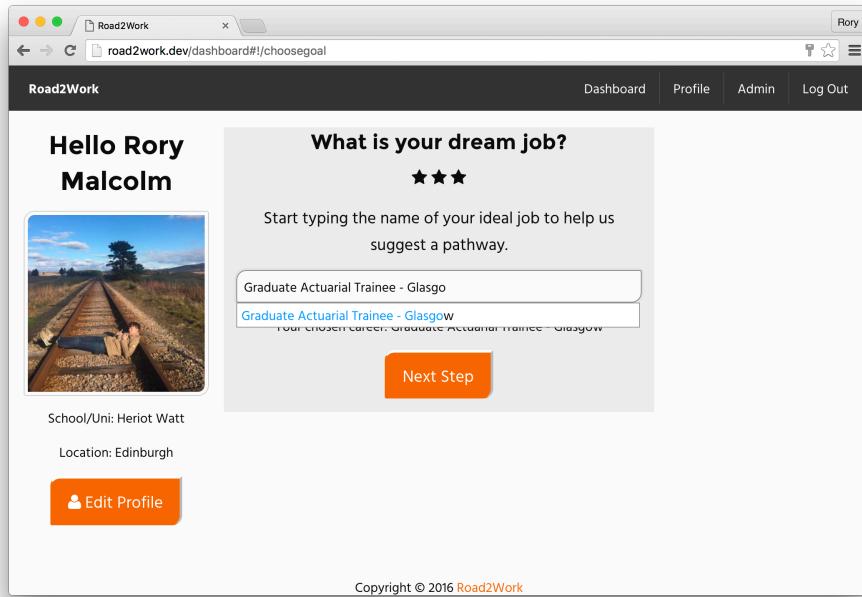


Figure 3.9: The user choosing their job.

This process was isolated onto a single page as we felt it heightened the importance of this step, it sets the direction in which the rest of the profile set up page will continue in. It has an autocomplete function to improve usability.

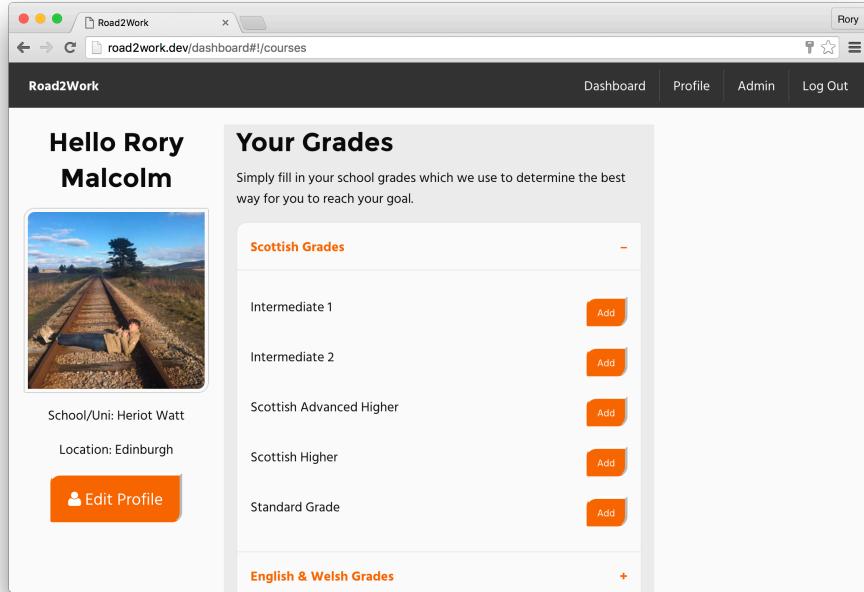


Figure 3.10: The user entering their grades.

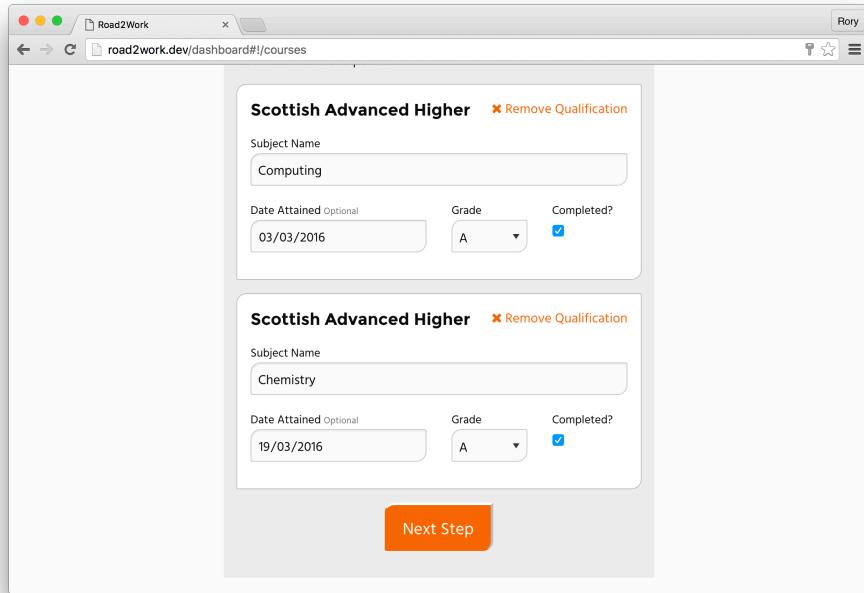


Figure 3.11: The user entering their grades.

The user enters the number of grades they have, then scrolls down and offers more details on grade attained and the date they received them, the number of UCAS points they have is then calculated and stored for performing calculations upon later on in the process.

The screenshot shows the Road2Work dashboard for user 'Rory'. On the left, there's a profile section with a placeholder photo of a person sitting on train tracks, labeled 'Hello Rory Malcolm'. Below it are fields for 'School/Uni:' and 'Location:', and buttons for 'Edit Profile' and 'Invite Mentor'. In the center, a 'Course Suggestions' section lists several courses with their titles and institutions, each accompanied by a '+' button. To the right, an 'Overview' section displays 'Current Aim' (No Goal Set), 'Next Steps' (No Steps Pending), and 'Steps so Far' (No Courses Completed).

Course Title	Institution
Actuarial Mathematics and Statistics Hons.	Kingston University
Actuarial Science and Diploma in Industrial Training Hons.	Heriot-Watt University
Actuarial Sciences Hons.	University of East Anglia
Actuarial Science Hons.	City University London
Mathematics with Actuarial Science Hons.	Queen Mary University Of London
Actuarial Science Hons.	Edge Hotel School
Actuarial Mathematics Hons.	The University Of Liverpool
Actuarial Science and	University of Manchester

Figure 3.12: The user choosing their course.

This is the result of the course search engine which is applied upon the name of the job they chose as their "Dream Job", the user then selects a course from the list provided for selection.

The screenshot shows the Road2Work dashboard for user 'Rory'. The central part of the screen displays detailed information about the 'Actuarial Science and Diploma in Industrial Training' course, including its title, duration (FullTime), college/university (Heriot-Watt University), course ID (201617_5G5B), and level (BSc Hons.). Below this, four buttons provide links to 'View Institution', 'Visit Course Webpage', 'View Funding Information', 'View Employment Info.', and 'View Assessment Methods'. At the bottom of this panel is a large green button labeled 'Choose Course'. To the right, the 'Steps so Far' section indicates 'No Courses Completed'. At the very bottom of the page, there are two additional course suggestions: 'Actuarial Sciences Hons.' at 'University of East Anglia' and 'Actuarial Science Hons.' at 'City University London', each with a '+' button.

Figure 3.13: The user choosing their course.

The user expands the course in order to provide more information about it and possibly make a selection.

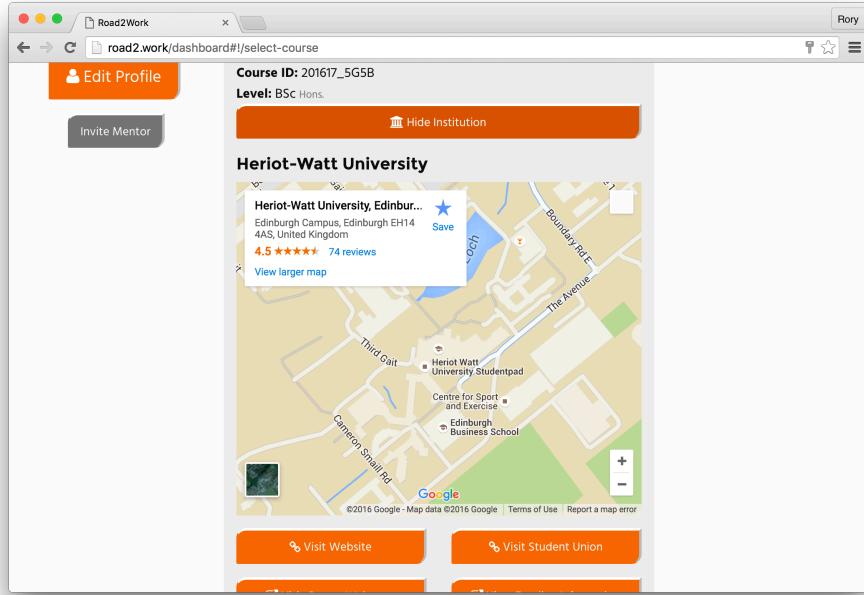


Figure 3.14: The user choosing their course.

The user can show more information about the institution the course will be completed at.

The screenshot shows the Road2Work dashboard with the URL road2.work/dashboard#/additional-courses. The user is viewing 'Additional Courses Required'. The page includes a profile section for 'Hello Rory Malcolm' with a profile picture of a person sitting on train tracks. Below this, there are fields for 'School/Uni:' and 'Location:', each with an 'Edit Profile' button. To the right, under 'Additional Courses Required', it says: 'You don't quite meet the requirements for the course you have selected with your current grades. Please add additional grades you plan to attain in order to reach entry requirements for your course.' It also states: 'These will appear in your pathway and you can update their status as you progress by ticking 'Completed' when editing grades.' A note below says: 'You require an additional 72 points.' A section titled 'Scottish Grades' lists four items: 'Intermediate 1', 'Intermediate 2', 'Scottish Advanced Higher', and 'Scottish Higher', each with an 'Add' button. On the right side, there are sections for 'Overview', 'Current Aim' (No Goal Set), 'Next Steps' (No Steps Pending), and 'Steps so Far' (No Courses Completed).

Figure 3.15: The user adding additional courses.

If the user does not have the grades to start the course yet, they are instructed to enter additional courses that they may want to complete in order to attain the correct number of UCAS points.

The screenshot shows the Road2Work dashboard for a user named Rory. On the left, there's a profile section with a photo of a person sitting on train tracks, the name "Hello Rory Malcolm", and details like "School/Uni: Heriot Watt" and "Location: Edinburgh". Below this are buttons for "Edit Profile" and "Invite Mentor". In the center, a large orange timeline bar represents the "Current Timeline" with four circular nodes. A green button labeled "★ Start new Pathway" is positioned below the timeline. To the right, the "Overview" section includes a "Current Aim" section for "Graduate Actuarial Trainee - Glasgow" with a "Required Grade A" note, and a "Next Steps" section listing "Scottish Higher Maths" and "Actuarial Science and Diploma in Industrial Training BSc". At the bottom, a "Steps so Far" section lists completed subjects: "Scottish Higher Computing", "Scottish Higher Chemistry", and "Scottish Higher Politics".

Figure 3.16: The user viewing their pathway.

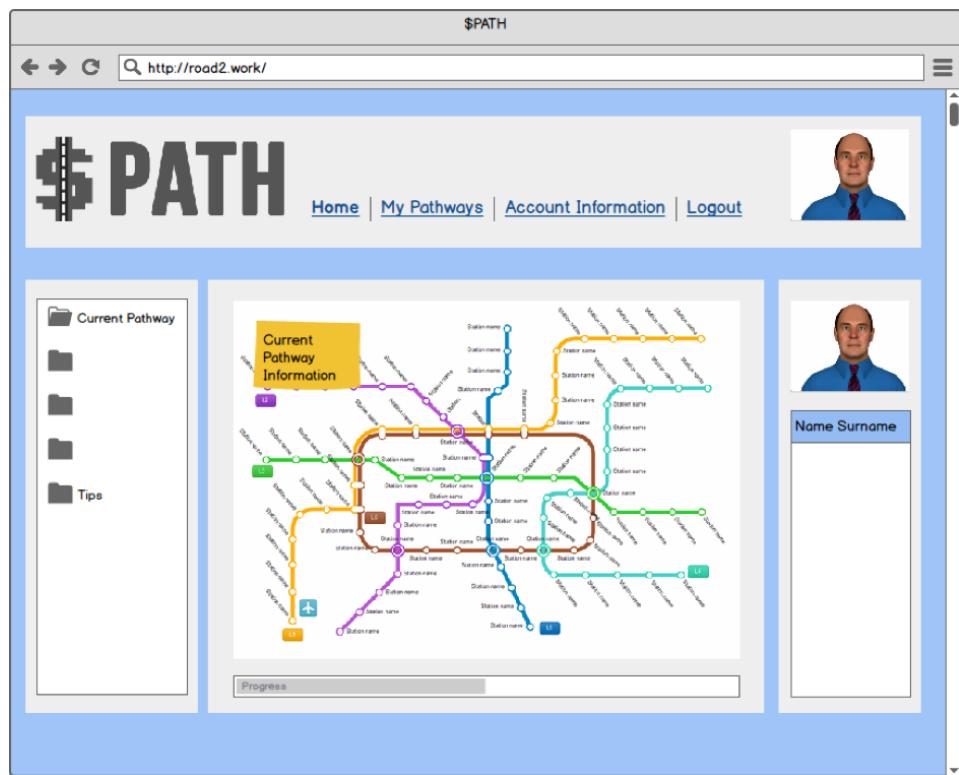


Figure 3.17: The mockup for user pathways

We originally envisioned a London Underground-esque graph with the stations being the steps taken to successfully be prepared to access the job, however we found that this was too complicated for the structure of pathways and was confusing to users in the initial usability tests. We then decided to move away from this to a more linear iteration, with the pathway showing where the user is now, and each dot on the line being a step to take towards their dream job, with the final dot on the line being that job. This is much more simple and readable, but still communicates the element of progression towards a goal that the original idea had.

The screenshot shows a user profile for 'Hello Rory Malcolm'. The profile picture is a photo of a person sitting on train tracks. The user's name is displayed prominently. Below the profile picture, there are buttons for 'Edit Profile' and 'Invite Mentor'. To the right, there is a section titled 'Your Goal Job' which includes the job title 'Graduate Actuarial Trainee - Glasgow', projected salary ('Competitive salary'), job type ('graduate scheme'), and location ('Glasgow'). A map of Glasgow and surrounding areas is shown, with a red dot indicating the location. Below the map, there is a section titled 'Potential Employers' with 'Mercer' listed and a logo for 'MERCER'. On the far right, there are sections for 'Overview', 'Current Aim', 'Next Steps', and 'Steps so Far', each listing specific subjects like 'Scottish Higher Maths'.

Figure 3.18: The user viewing job information

This is the user viewing additional information about the job they have chosen, it's location, the company it is with, the salary and the type of employment.

The screenshot shows the same user profile as Figure 3.18. In the center, there is a form titled 'Send a Mentor Request' with a text input field for 'Email Address' containing 'e.g. mentor@road2.work' and a 'Send' button. To the right, the 'Overview', 'Current Aim', 'Next Steps', and 'Steps so Far' sections are visible, identical to Figure 3.18. The left side of the screen shows the user's profile information: name 'Hello Rory Malcolm', profile picture (train tracks), school/university 'Heriot Watt', location 'Edinburgh', and buttons for 'Edit Profile' and 'Invite Mentor'.

Figure 3.19: The user linking their profile with their mentor's.

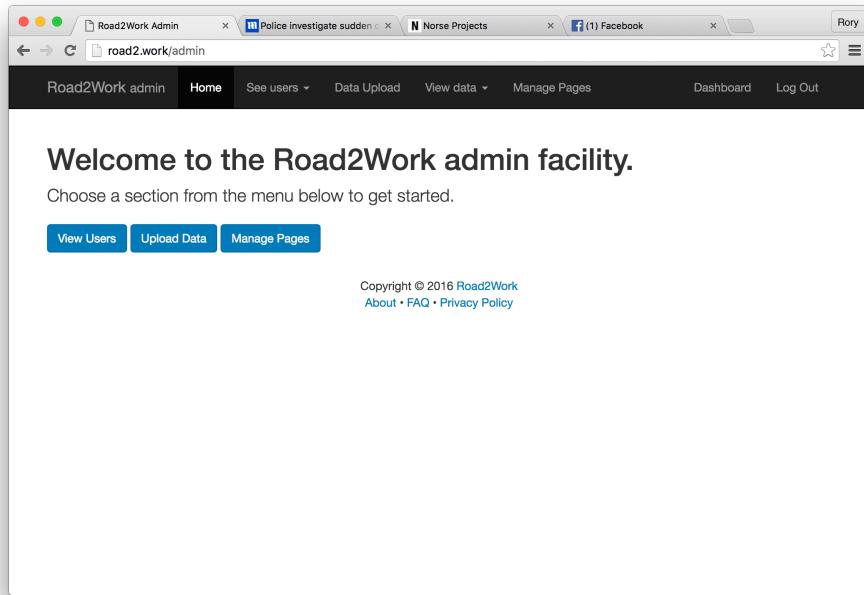


Figure 3.20: The homepage of the admin facility.

A screenshot of the 'See users' page within the Road2Work Admin interface. The title bar and header are identical to Figure 3.20. The main content area has a heading 'Groups'. Below it is a table with the following data:

Standard Mentor Admin All [New User](#)

Showing 20 users.

#	Name	Email	Actions
13	Daniel 222 Waghorn	daniel@daniel-waghorn.com	View User
33	Daniel Waghorn	hello@daniel-waghorn.com	View User
34	Stuart Marples	stuartmarples95@gmail.com	View User
35	Vitalija Zilinskaite	ziliinskaitevitalija@gmail.com	View User
36	Rachel Henderson	rchenderson7@gmail.com	View User
37	Viktor Falkengren	viktor7k@gmail.com	View User
38	Barbeesha Bashwell	binderiya.batbayar@gmail.com	View User

Figure 3.21: The admin viewing all the users.

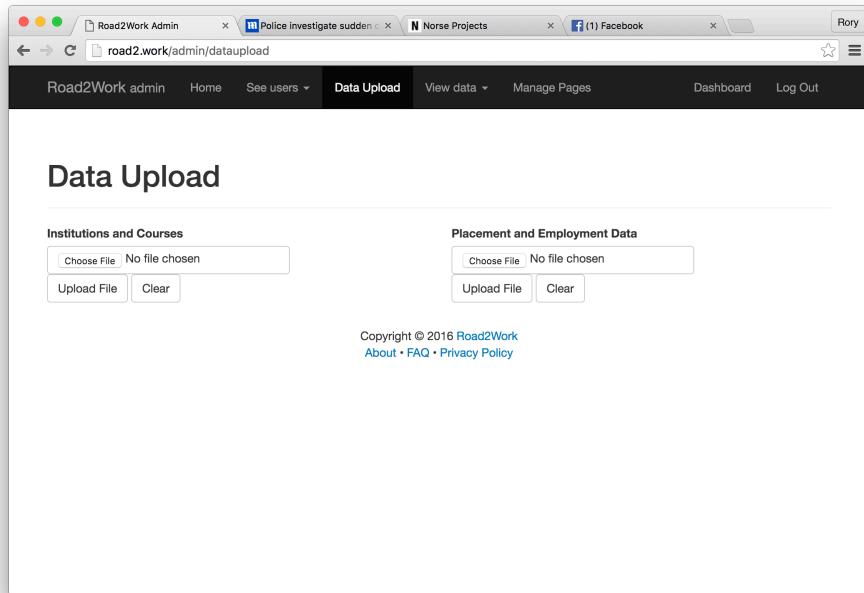


Figure 3.22: The admin in the data upload interface.

The screenshot shows a web browser window titled 'Road2Work Admin' with the URL 'road2.work/admin/view/institutions'. The header is identical to Figure 3.22. The main content is a table titled 'View data' showing a list of institutions. The columns are '#', 'Name', 'Student Union URL', and 'Institution URL'. The data is as follows:

#	Name	Student Union URL	Institution URL
1	ABDO College of Education	Student Union	Institution
2	University Of Aberdeen	Student Union	Institution
3	University Of Abertay Dundee	Student Union	Institution
4	Aberystwyth University	Student Union	Institution
5	Accrington And Rossendale College	Student Union	Institution
6	Activate Learning	Student Union	Institution
7	AECC Chiropractic College	Student Union	Institution
8	Alton College	Student Union	Institution
9	Amersham & Wycombe College	Student Union	Institution
10	Anglia Ruskin University	Student Union	Institution

Figure 3.23: The admin viewing all the institutions.

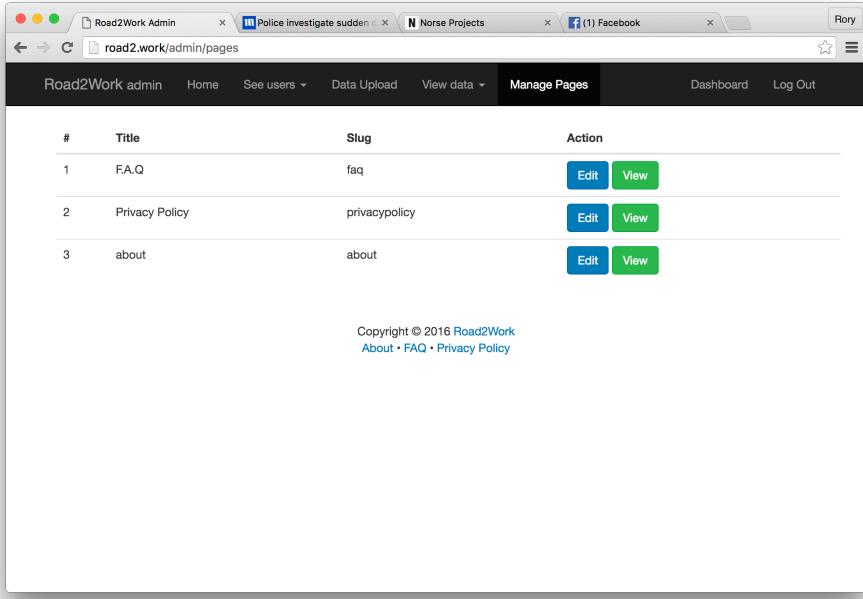


Figure 3.24: The admin in the pages CMS interface.

The admin interface has a uniform interface, allowing them to control all features and database tables to ensure that the product works correctly.

Implementation

Data Sourcing and Processing

We progressed onto looking sources of data and how we could generate the desired pathways from it and how this data could be modelled and consumed by our application. Initially we planned to use data similar in format to the example data provided on Vision in .csv format so we explored solutions that would allow us to import this data for processing by the application. Eventually we settled for *PHPExcel* which we wrapped within a service container for Laravel. Adding packages such as this was trivial since we included *Composer* [4] with our project, which is essentially a package manager for PHP and pulls open source packages from *Packagist* [18].

PHP dependencies are managed via a `composer.json` file which lists packages and the required versions. These are then installed to a `vendor` folder in the project root. Advantages of working this way include easy syndication of project dependencies since each team member can simply run `php composer install` to install everything that is required, as well as being able to easily update packages as new versions are released.

Moving forward we explored potential sources of data online which essentially turned up little that we found of use. Whilst searching we came across the Unistats API [24], which we found to provide a substantial amount of course and institution data for nearly all of the colleges and universities in the UK.

Using the *Postman* API Client [19] we queried and explored the API which we found could return JSON data which is naturally the most useful to us since within PHP there are good encoding and decoding functions which can convert JSON to regular PHP objects and vice versa.

Once satisfied with the endpoints we were interested in a PHP script was built to act as an API scraper which would accumulate data and format it into nested JSON which we could then consume with our web application for a data import. Once we were satisfied with the data the scraper could obtain we removed *PHPExcel* from our project dependencies.

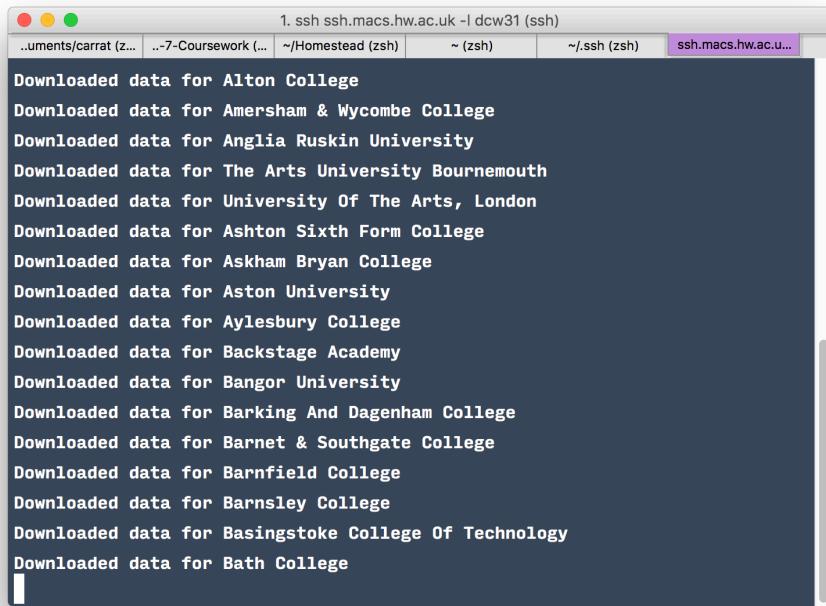
Figure 3.25 shows an example of the API Scraper running with the Unistats API.

The resulting JSON file was approximately 93MB in size, contained 429 institutions, with which there are 49,732 courses on offer including full-time, part-time and distance learning courses. These ranged in levels from Bachelors with and without Honours, Masters and college courses such as HNDs.

We encountered some difficulty in reading the resulting JSON file within PHP due to its sheer size. Normally for most files you could simply read the file stream into a variable, then call `json_decode()` on it and have an array of PHP objects returned which can be easily iterated over for processing. In our case however this would require a significant amount of memory, and despite increasing the script memory limit to over 2GB we encountered issues before we could even parse the file.

The solution we settled for involved using a streaming JSON parser which essentially reads the file progressively and partially without committing it in its entirety to memory. We found an open source project on *GitHub* [9] called *jsonstreamingparser* [21] which allowed us to read the import file in this manner.

This package was required via Composer and then installed into the project. We had to write a custom Listener in order to hook into the events such as `beginObject` and `endObject` at levels specific to our data file. Based on these events we could then accumulate an entire institution's courses, insert a database entry for the institution, then iterate over each course and in turn store it in the database.



A screenshot of a terminal window titled "1. ssh ssh.macs.hw.ac.uk -l dcw31 (ssh)". The window shows a list of college names being processed sequentially:

```

Downloaded data for Alton College
Downloaded data for Amersham & Wycombe College
Downloaded data for Anglia Ruskin University
Downloaded data for The Arts University Bournemouth
Downloaded data for University Of The Arts, London
Downloaded data for Ashton Sixth Form College
Downloaded data for Askham Bryan College
Downloaded data for Aston University
Downloaded data for Aylesbury College
Downloaded data for Backstage Academy
Downloaded data for Bangor University
Downloaded data for Barking And Dagenham College
Downloaded data for Barnet & Southgate College
Downloaded data for Barnfield College
Downloaded data for Barnsley College
Downloaded data for Basingstoke College Of Technology
Downloaded data for Bath College

```

Figure 3.25: API Scraper

Similarly for job and placement data we found another data source, in this case the *Prospects.ac.uk* website [20]. Whilst exploring the website we noticed that parts of it were an SPA or *Single Page Application* since they loaded new data asynchronously rather than via full page refreshes. We examined the network requests being made in the background by our browser's developer tools and figured out the main API endpoints for the AngularJS application running on the front-end.

Again we used Postman to examine the data returned and from this we created another API scraper script in order to retrieve job data from the website which created another JSON import file for the data upload. The resulting file was significantly smaller than the course data file and as such we were able to read the whole file into memory at once and then iterate over it and insert into the database.

This scraper also retrieved the logos of any employers which were linked to jobs and saved them into a folder which was then copied into the `public` folder of the website in order to embed them into employer descriptions displayed by the website.

Data uploads were made via the dedicated administrative back-end of the website which is only accessible to authenticated users whom are in the 'admin' user group. This page is illustrated in Figure 3.26 which shows the different sections for uploading each import file.

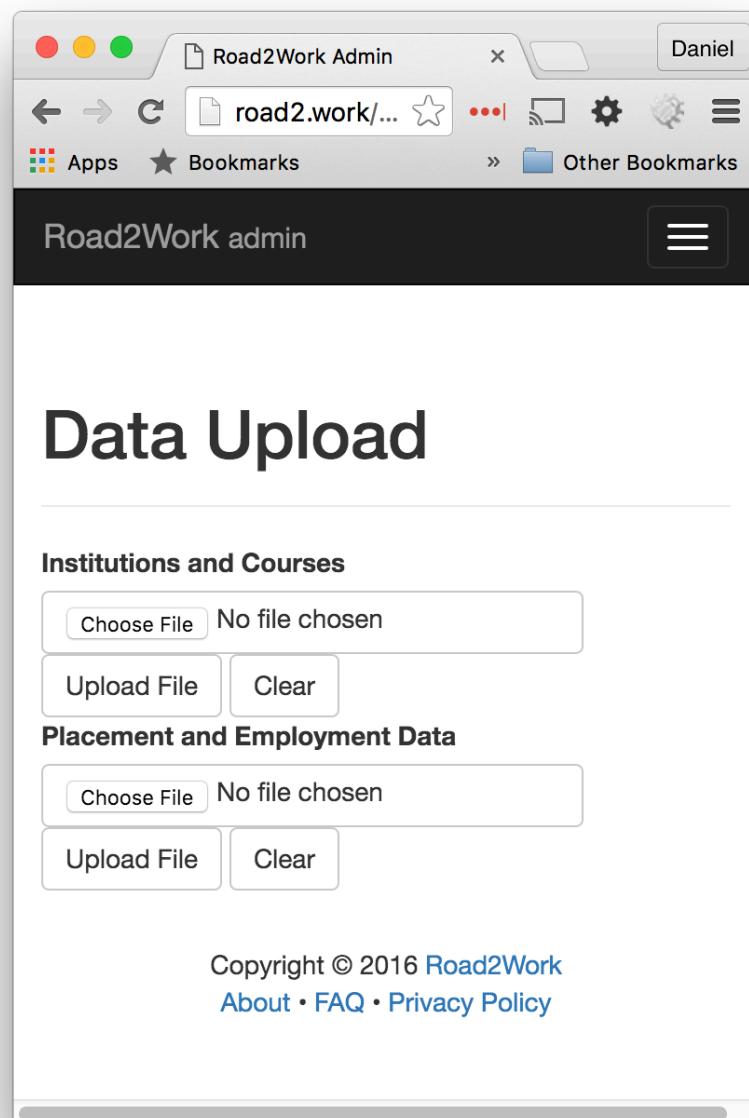


Figure 3.26: Data Upload Utility

Data Import Files

3.0.1 Courses data

As explained before, the course information used for the project was obtained with the use of an API called Unistats. This process returned a large amount of data that had to be filtered in a way that got only the important data to our databases.

All the fields of the data obtained are shown in the table below, along with the database table where this was used and the name of the field it received.

Parameter	Information
ApiUrl	Not used in the database.
Country	Not used in the database.
NSSQuestion24	Not used in the database.
NSSQuestion24Population	Not used in the database.
Name	Name of the university. Goes in the database under Institution's information as Institutions.NAME.
NumberOfCourses	Not used in the database.
PUBUKPRN	Not used in the database.
QAAReportUrl	Not used in the database.
SortableName	Not used in the database.
StudentUnionUrl	URL of university's Student Union. Goes in the database under Institution's information as Institutions.SUURL.
StudentUnionUrlWales	Not used in the database.
UKPRN	Number of the university in the UK Register of Learning Providers. Goes in the database under Institution's information as UKPRN.
ApiUrl	Not used in the database.
KisCourseId	Course's Key Information Set (KIS) ID. Goes in the database under Course's information as Courses.CourseID.
KisMode	States if course is PartTime or FullTime. Goes in the database under Course's information as Courses.CourseType.
Title	States the name of the course. Goes in the database under Course's information as Courses.Title.
TitleInWelsh	Not used in the database.
ApiUrl	Not used in the database.
ApplicationUKPRN	Not used in the database.
AssessmentMethodsUrl	URL of the university's assessment methods site. Goes in the database under Course's information as Courses.AssMetURL.
AssessmentMethodsUrlWales	Not used in the database.
AverageInScheduledLearningAndTeaching	Not used in the database.
AverageInWrittenExams	Not used in the database.
CoursePageUrl	The course's page URL in the university site. Goes in the database under Course's information as Courses.CourseURL. This information is also used in the Institutions database with the root address of the website only, as Institutions.INSTURL, allowing the user to access the university's website.
CoursePageUrlWales	Not used in the database.
DistanceLearning	States if course has an option for distance learning or not. Goes in the database under Course's information as Courses.DistanceLearning.
EmploymentDetailsUrl	Employment information for future students. Goes in the database under Course's information as Courses.EmployURL.
EmploymentDetailsUrlWales	Not used in the database.
FeeVariesByYear	Not used in the database.
FeeVariesWithInflation	Not used in the database.
FeeWaiverAvailable	Not used in the database.
FeesToBeConfirmed	Not used in the database.
FoundationYearAvailable	Not used in the database.

Honours	States if course has an option for honours degree or not. Goes in the database under Course's information as Courses.Hons.
IsPartTime	Not used in the database.
JACSCodes	Not used in the database.
KISType	Not used in the database.
KisAimCode	Not used in the database.
KisAimLabel	States the label of the course (if it's a Masters, BSc or others). Goes in the database under Course's information as Courses.Level.
KisMode	Not used in the database.
LDCSCodes	Not used in the database.
LearningAndTeachingMethodsUrl	Not used in the database.
LearningAndTeachingMethodsUrlWales	Not used in the database.
LocationChangesDuringCourse	Not used in the database.
MaximumFeeForEnglandDomicile	Not used in the database.
MaximumFeeForNIDomicile	Not used in the database.
MaximumFeeForScotlandDomicile	Not used in the database.
MaximumFeeForWalesDomicile	Not used in the database.
MeansTestedSupport	Not used in the database.
NonCreditAssessment	Not used in the database.
NonMeansTestedSupport	Not used in the database.
PercentAvailableInWelsh	Not used in the database.
RelatedKISCourseIds	Not used in the database.
SandwichAvailable	States if course has an year abroad option. Goes in the database under Course's information as Courses.SandwichAvailable.
StudyLevel	Not used in the database.
SupportDetailsUrl	URL for university's financial support information. Goes in the database under Course's information as Courses.FundingURL.
SupportDetailsUrlWales	Not used in the database.
TotalNumberOfStages	Not used in the database.
YearAbroadAvailable	Not used in the database.

3.0.2 Job Data

In a similar way to obtaining the course information, we obtained the job information used for this project from a data source (Prospects.ac.uk website) and filtered the important fields to our purposes.

All the fields of the data obtained are shown in the table below, along with the database table where this was used and the name of the field it received.

Parameter	Information
lastPage	Not used in the database.
totalNumberOfJobs	Not used in the database.
Id	Job ID from data obtained. Goes in the database under Job's information as Jobs.jobID.
Title	Name of job. Goes in the database under Job's information as Jobs.Title.
Logo url	Logo of the company. Goes in the database under Employer's information as Employers.Logo.
Logo width	Not used in the database.
Logo height	Not used in the database.
Employer name	Name of the company. Goes in the database under Employer's information as Employers.Name.
Employer logo	Not used on the Job's database.
Employer id	Not used on the database.
Salary tnr	Not used in the database.
Salary text	Salary for the offered job. Goes in the database under Job's information as Jobs.Salary.
Location tnr	Not used in the database.

Location text	Location of the offered job. Goes in the database under Job's information as Jobs.Location.
typeOfJob tnr	Not used in the database.
typeOfJob text	Type of offered job. Goes in the database under Job's information as Jobs.Type.
closingDate	Not used in the database.
displayTo	Not used in the database.
continuousRecruitment	Not used in the database.
saved	Not used in the database.
isFeatured	Not used in the database.
isNew	Not used in the database.
jobCategories	Not used in the database.
totalEmployerJobs	Not used in the database.
employerKeyword tnr	Not used in the database.
employerKeyword text	Not used in the database.
employerSlug	Not used in the database.
jobSlug	Not used in the database.

Content Management System

Around the same stage the Content Management System was started. This forms the basis for static pages such as the ‘Privacy Policy’, ‘About’ and ‘Terms of Service’. In order to achieve this an additional migration for a *Pages* table was added to create the desired schema.

A standard template for a full page was also created into which the various content and meta content for each page is rendered into. The ‘Pages’ controller is invoked whenever a URL does not match an explicitly defined route in the `routes.php` file, and if an entry for it is not found present in the database the application will return a 404 Not Found error.

The administrative interface was also extended to include a page editor where the user chooses from a list of pages defined within the system, choosing whether to ‘Edit’ or ‘View’ them. The editor features a WYSIWYG interface to facilitate easy editing of HTML content as shown in Figure 3.27.

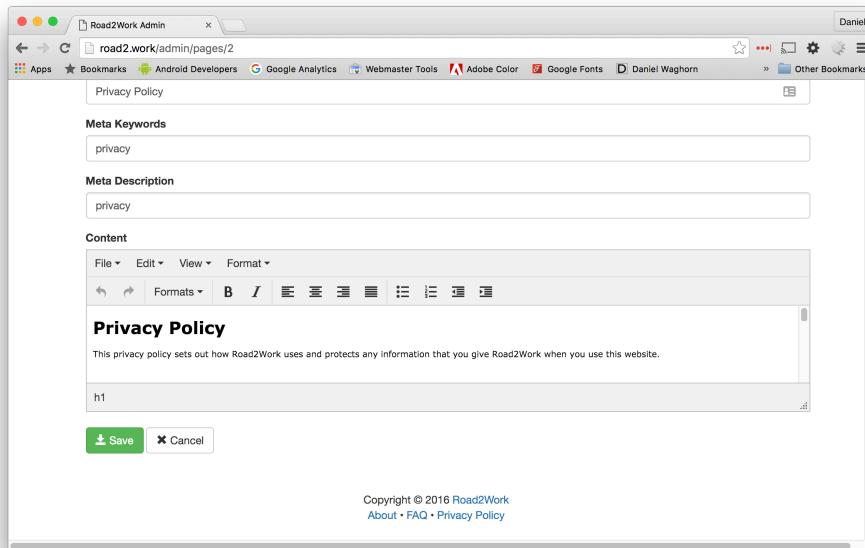


Figure 3.27: Admin Page Editor

Dynamic SPA Front-end

As we moved forward into Stage 3 we added *VueJS* [26] to our development stack to use as a front-end JavaScript framework which we used to create a modular user interface with reusable components. Vue is a relatively new project being only around a year old, however has amassed a very large community of developers and contributors. It is extremely lightweight yet well featured as a reactive framework and can be extended with various official and community

contributed addons.

We decided to use this for the front-end because of its modularity and when combined with *Vue Resource* [27] it makes the passing of data to and from the front-end trivial via AJAX calls to API endpoints. With this we can two-way bind elements to a JavaScript representation of data via the v-bind directive and update the DOM to respond to changes in state and data automatically without the requirement to use DOM manipulation libraries such as *jQuery* [10].

In order to create a Single Page App or SPA with these Vue components we made use of *Vue Router* [28] which renders components selectively based on the URL and parameters similar to libraries such as *director* [6].

Figure 3.28 illustrates the component based composition of the interface where it has been inspected via the *Vue DevTools* chrome extension. Note the section on the right showing the selected *CourseInstance* component's internal data structure. For repetition of components modelling entities e.g. a course grade we simply render multiple components by placing code in the markup as such:

```
<course-instance :course="c" v-for="c in courses"></course-instance>
```

Another feature which we made extensive use of was the event system to allow inter-component communication along the parent-child chain. An example of where this was used was for ‘Remove Qualification’ where an event would be dispatched upwards from the qualification *CourseInstance* component to be removed until it reached the parent component which held a reference to all grades. This grade is then removed from the parent’s internal data structure and the v-for automatically re-renders the document to reflect the change.

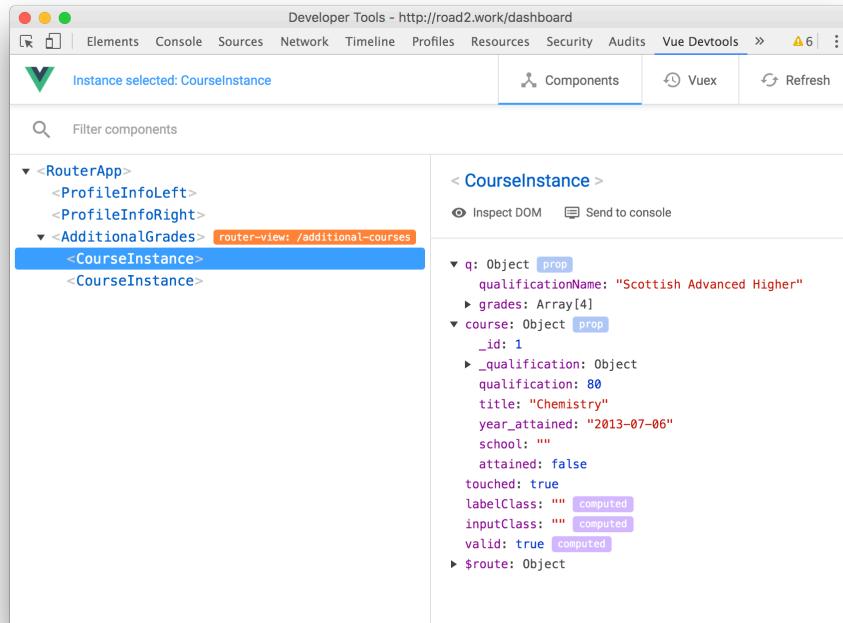


Figure 3.28: Vue Component Visualisation

Components were built as required for each section of the front-end and then combined to create complete views. This promotes modularity and component re-use which in turn increases cohesion and maintainability. Each complete view was then defined with parent components within *Vue Router*, and linking between views is trivially achieved via the v-link directive which corresponds to the defined routes within *Vue Router*.

Build Process and Laravel Elixir

We spent a good deal of time setting up a build process that was effortless and allowed individual team members to work on the application simultaneously without any conflicts between builds and dependencies. At the center of our build system lies *Laravel Elixir* [11] which is essentially a wrapper for the *Gulp* [8] build system which is based on *Node* [16].

We defined tasks within our `gulpfile.js` file which are run from the command line by invoking `gulp` within

the project directory, or `gulp watch` to automatically run tasks as files are updated. We also made use of `npm` [17] which is a package manager like Composer for Node. This makes use of a separate file for node dependencies which is synchronised between team members via Git, where a team member can simply run `npm install` or `npm update` to install required project dependencies.

Compilation of CSS

All of the project CSS stylesheets are derived from the compilation of *SCSS / Sass* stylesheets. *Sass* [22] is an extension of CSS which allows developers to make use of additional features such as variables, mixins, nesting of scope, control structures and more in order to make authoring maintainable and modular stylesheets easier and faster.

The main `app.css` file is built from a file named `app.scss` within the `resources/assets/sass` directory in the project root, and consists of imports of various partials to make the individual elements easier to find when editing.

Within our Gulpfile we added the line `mix.sass(app.scss, ...)` which tells Gulp to compile the file and place it within the `public/assets/css` directory where `public` is where our web root is mapped to. The compilation is done by the `node-sass` module which is listed within our dependencies file.

Compilation of JavaScript

The project JavaScript assets were compiled in a similar fashion to the CSS although the instructions in the Gulpfile were slightly different. The line `mix.browserify('app.js');` instructs Gulp to use *Browserify* [3] to preprocess the required JavaScript files in various ways.

This is done via *transforms* which are essentially stages in the build pipeline which alter the source file as it progresses. We used a number of transforms for the JavaScript in our project such as `es2015` and `es2016` which allowed us to use newer revisions of the JavaScript syntax in source which are then transformed into equivalents which are supported by browsers which may not have a JavaScript engine which supports the latest syntax.

Browserify also enables the use of `es6` modules with old browsers by transforming the `require('...')` directive into scoped JavaScript, allowing us to easily integrate node module versions of libraries such as `jQuery`, `Vue` and `d3.js` and have them output into a single bundled output file.

Vue Components and Vueify

Another transform we made extensive use of was *vueify* [29], which was integral to our design workflow since it allows us to encapsulate components into single files which combine the template, scoped styling (which is also preprocessed as *Sass*), and JavaScript logic in the form of an `es6` module as shown in Figure 3.29.

These modular components can then also make use of other components, even recursively via `import` or `require` statements.

With this setup individual components and their associated functionality were delegated across team members and by working out dependencies between components allowed us to prioritise their development and work in a more efficient way by determining the critical path.

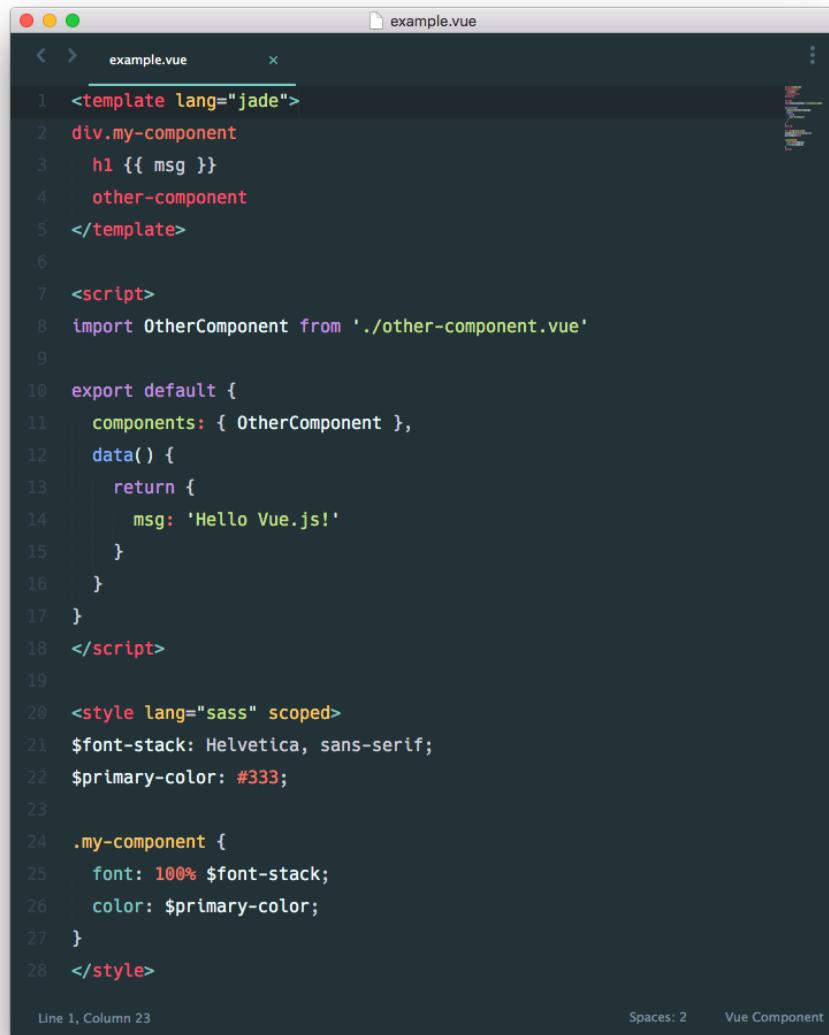
Once completed main view components were simply imported via `require('...')` in `app.js` and integrated with `Vue Router`.

Laravel Artisan CLI and Migrations

Laravel ships with its own official Command Line Interface named `artisan` [2], which lies at the center of many framework build and lifecycle operations. Use cases include situations such as the creation of a new controller class, where a command such as `php artisan make:controller MyController` can be run and `artisan` will create the new controller with correct namespacing and boilerplate code.

It is also used when making database specific modules such as Models and Migrations. Laravel makes use of a query builder and Object Relational Mapper or ORM known as *Eloquent*. A major advantage of using a technology such as this is it makes code database independent, so if the database engine was to be changed in the future e.g. from MySQL to DB2 or even a cloud based service then no code changes would have to be made other than the database configuration in the `.env` file.

Factories were also another key feature which we used to seed the database tables with viable test data. In one



The screenshot shows a code editor window titled "example.vue". The code is written in a combination of HTML, JavaScript, and CSS. The HTML part includes a template section with a Jade-like syntax, a script section importing another component, and a style section using SASS. The JavaScript part defines a component with a msg prop and a scoped style block.

```
<template lang="jade">
  div.my-component
    h1 {{ msg }}
    other-component
</template>

<script>
import OtherComponent from './other-component.vue'

export default {
  components: { OtherComponent },
  data() {
    return {
      msg: 'Hello Vue.js!'
    }
  }
}
</script>

<style lang="sass" scoped>
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

.my-component {
  font: 100% $font-stack;
  color: $primary-color;
}
</style>
```

Figure 3.29: Vue Component

```

source php artisan
Laravel Framework version 5.1.31 (LTS)

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version          Display this application version
  --ansi                Force ANSI output
  --no-ansi              Disable ANSI output
  -n, --no-interaction   Do not ask any interactive question
  --env[=ENV]             The environment the command should run under.
  -v|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output,
                          2 for more verbose output and 3 for debug

```

Figure 3.30: Artisan CLI

example we needed to test functionality of the User Admin facility, so instead of manually creating entries in the database we simply ran a script similar to below with an REPL interface.

```
factory(App\User::class, 50)->create();
```

This gave us 50 fake user entries with viable data since the factory class consults database schema and examines column names and field types whilst generating the data.

All table and schema definitions are made in code via a Blueprint within a migration. This is a significant advantage when working in teams of any size since it keeps schema contained within version control and is much more preferable than the curation of shared SQL files which can easily become hard to manage.

Migrations can be run via the `artisan migrate` command which connects to the database and creates the required database schema, where our table definitions are independent of the database engine used and hence are translated into the required SQL on-the-fly.

Artisan also lies at the centre of many more core functions such as putting the site into maintenance mode and taking it out with `artisan down` and `artisan up` respectively. Another useful feature is `artisan tinker` which starts an REPL session enabling us to interact with Models and other parts of the framework to debug and simulate operations without having to integrate a test method in a controller or similar.

Varant VM and Laravel Homestead

For local development all team members installed the *Vagrant* [25] Virtual Machine wrapper which can use VMWare or VirtualBox as VM provider. We preferred this over a system such as *Docker* [7] since Docker requires a large amount of configuration in comparison to Vagrant on Mac OS X and Windows. Another reason is that there is an official *Laravel Homestead* [13] box (essentially a pre-packaged installation on a VM) which could easily be added to Vagrant.

The Homestead box provides a development environment exactly the same as the environment we planned to deploy to in production. Once the box was added we simply edited the accompanying `Homestead.yaml` file to map local folders to the VM and set up sites for use by the `nginx` web server. We then run the `vagrant provision` command to update the path and port mappings and configurations within the VM.

To interact within the virtual machine we used `vagrant ssh` to start an SSH session under the root account allowing us to change configurations as required in addition to development commands making use of Composer and Artisan.

Xdebug

Xdebug [30] is an extension to the PHP module used within most web servers to execute PHP scripts. It allows us to set breakpoints and debug live PHP code via the interpreter which is an indispensable tool when it comes to developing complex aspects of the application such as the streaming JSON parser or the UCAS calculations.

We integrated it with our IDEs and set up the Vagrant VM instance as a remote interpreter. When a breakpoint was set and reached it allowed us to inspect the current state of the application including variables, traces and also provided us with an interactive console which was useful for evaluating expressions and testing potential fixes.

This significantly improved the efficiency of our debugging workflow since we could quickly ascertain the causes of problems by examining internal state rather than relying on more primitive methods such as print statements or vague log files to determine causes and potential solutions.

Laravel Forge and DigitalOcean

For our deployment of the final product we made use of the *Laravel Forge* [12] service, combined with *DigitalOcean*'s [5] Droplet VPS cloud service. Working with these two services allowed us to provision a fully scalable cloud VPS within a matter of minutes. Laravel Forge provisions and installs all of the necessary programs such as PHP 7, nginx and others automatically once the accounts are linked.

There was a small amount of configuration required via SSH in order to fully tailor the new server to our needs however this was minor such as setting MySQL memory limits accordingly and adding SSH keys.

One of the key features of this setup with was important was the ability to deploy the latest build straight from our GitHub repository. This way whenever a new commit or merge was put onto the master branch it would automatically deploy to the live site.

This could be further extended down the line to hook into continuous integration systems such as *TravisCI* [23] so that as soon as a new commit passes the defined unit and other tests correctly it is deployed to the server. Working in this way fits naturally with the scrum workflow so that once a deliverable component is completed it can be deployed through this process.

4. Final Usability Evaluation

4.1 Testing Throughout Development

To test our user interface design before we had started developing our solution we carried out usability testing, where users were presented with mock-ups of our site and told to attempt to navigate the website from page to page. This allowed us to ensure that our visual cues were correct that the format of our website was not too alien as to confuse users and that the final version of our product, if it matched the mock-ups, would work. During development whenever a new feature or page was added it was rigorously tested, the developer would work with the page or feature, testing it in every scenario they thought might be possible to ensure there were not bugs in the code we had not thought about and missed. Whenever we reached a new stage of the code, where a feature was implemented we considered complete we got external testers to run through it, as someone who is not involved in the development process brings something new to the table and may try to do something that the developers had not thought about and expose a flaw in our system. After we had finished the development of the website we carried out usability tests again with different people to those who had carried them out at the beginning and during the development process, they were told to navigate through our new user setup wizard to and oversee doing so as to ensure that our process guides a new user through the process effectively and coherently.

Testing Protocol

Aim: The aim of this study is to test that the system we have built is designed in such a way that a new user can navigate their way around the system and read the information we present them with easily. Our system aims to provide school leavers with a pathway on to their chosen career goal. It takes in what qualifications and experience they already have and tells them the steps they must complete to be fully qualified for their desired role.

Introduction: The website asks a user on the homepage to either log in or register a new account. If the user has not created an account before, they need to confirm their email address before logging in for the first time. The user is then shown the profile page, where they can add more details, alongside the ones already entered during the registration process. Up next, Road2Work suggests the user to start typing in the career goal, and choose one of the jobs dynamically appearing under the input box. It then asks the user to enter the qualifications and experience they have obtained up to this point. The website then calculates the pathway that they must take and presents it in a readable manner.

Description: This study will present you with a fully functional website and first of all, ask you to complete two tasks: new user registration, and obtaining the pathway to your desired career. You will then have the opportunity to answer questions about the website on a 1-5 scale, 1 meaning “Strongly Agree”, and 5 meaning “Strongly Disagree”. You will then be asked for general comments on the site’s layout. Your answers will be kept anonymous and there are no right or wrong answers and any comments you have will be extremely valued. You can ask to stop the study at any time.

Test Plan

1. Objectives Our main objectives for the usability test are to examine the capability of the website to provide the information to the user in an effective manner so that they can understand what our product is providing them with, and make use of it. It is essential that the website is easy to navigate so it can even be used by users who may not be computer literate. It is also important to ask the subjects if they find the website aesthetically pleasing as this makes using our website less of an effort, which should provide a better user experience.

2. Participants Ideally we would like to test on brand new school leavers, however this is very difficult to do due to issues with child protection and data capture of results they provide. We have access to a large portion of the student population who were recently in such a position themselves; these students have also not completed their “pathways” themselves, meaning there are still steps to take to pursue their desired career. This means that students are useful to test on and provide meaningful results, and will make up the main portion of our test subjects.

3. Task Scenarios Our aims in testing are to test that the users can understand the information that they are presented with and can navigate around the site using the linking and navigation structure we have built. The first task we are giving the user to complete is to register themselves as a new user. Registration process is considered to

be complete when the new user is verified by clicking the link in the automatically generated email sent to the email address provided. The analysis of this task's results will ensure that the primary interaction with our system runs smoothly and requires little effort.

The second task the user is asked to complete is obtaining the pathway steps to the career chosen by the user. It is crucial to make sure that the main functionality of our system (building the pathway for an individual user) is intuitive and provides no errors. This will also test a number of subordinate features, such as UCAS points calculation and entering grades already obtained.

4. Metrics First of all, the quantitative data we will collect is the test subject's age and the average number of hours spent using the computer per week. We will then observe the user's performance completing given tasks and collect quantitative objective data having measured the time taken to complete those tasks. Quantitative subjective data we will collect is the results of the Likert scale questions which range from 1-5 with 1 being that the user strongly agrees with the given statement and 5 being that they strongly disagree with the given statement.

5. Questions The qualitative data that we will collect is the user's response to the question about specific things they liked or disliked about the website, as well as their response to the opportunity to make any additional comments about the system. This should allow us to spot any errors we may have overlooked during the system development process, as well as potential improvements and/or additional features we have not thought of before.

Study Description

For our usability study we drew up a questionnaire through which we would guide subjects after asking them to register as a new user and navigate to the pathway for the desired career. They would be asked to grade the statements on a 1-5 Likert scale from "Strongly Agree" through to "Strongly Disagree". The question/statement topics overview the main functionality of the system as well as the user interface. Test subjects were asked to provide any additional comments they would like to make about the general layout of the website, and mention the features they particularly liked or disliked. Having the information of how easy or difficult it is to navigate through the website allows us to evaluate the overall structure of the system.

An example of this questionnaire can be found in the Appendix ??.

Questions Subjects Were Asked

The questions asked by the group members conducting the study were as follows:

1. How old are you?

2. What is your occupation?

3. In a week, how much time do you spend on a computer? These are general questions to discover more about the user's familiarity with computers. The answers to these questions allow us to determine if experienced users encounter different problems than less experienced users. In addition to that, these answers help us to determine whether our visual cues are strong enough to guide the user through all the features of the system.

4. The registration process was intuitive and easy to navigate through. Registration is most likely the very first interaction a new user will have with our system. Hence, it is important that the process runs smoothly and does not throw the user off. Since our registration process involves email verification which means navigating away from our website, it is especially important that the user does not encounter any difficulties.

5. It was clear what parts of the page were for new users and which parts are for existing users. We wanted to perform this test to ensure that our log in screen communicated its purpose correctly and that users knew which parts to use depending on if they were new or existing.

6. The dashboard screen was clearly laid out. This was to test whether our dashboard was easy to navigate and that the sections we had defined were capable of communicating their purpose to the user. It is important that we make sure the user understands the information displayed on the right hand side, i.e. the progress made is obvious, as well as the remaining path to complete is detailed enough, but does not confuse the user.

7. The process of entering current qualifications was intuitive. It is important to get the feedback on this part of our system, since the process of entering the grades can get relatively complicated, depending on the amount of courses the user has completed or is planning to complete.

8. The pathway display represented the data well. We used a linear graph display to illustrate full pathway on the main page of the dashboard. We feel it is a format which is successful in displaying time lines and pathways, and wanted to ensure that our users agreed.

9. It was easy to understand the pathway to the chosen career. The pathway visualisation is the main part of our system, and the display the user will see every time they log in to the system. Thus, it is crucial for it to be easily understandable. The progress made so far, as well as the remaining steps to take should be obvious.

10. It was easy to find any extras like job information from the layout. This tests our layout's ability to separate primary content, such as the pathway visualisation, from secondary content, such as information about the goal career the user has selected.

11. The profile section has the information laid out in a coherent manner. This was to test whether our profile section was easy to navigate and that the sections we had defined were capable of communicating their purpose to the user. It is also important that changing the user information is just as easy as filling it in for the first time.

12. (For users in Mentor group only) The information about users being mentored is laid out in a coherent manner Since system views for mentors are different than the ones for standard users, it is important to test whether the information laid out on the dashboard is displayed effectively and does not confuse the user.

13. I found the overall layout of the website was sensible. This was a question about the site's general layout and if the participant found all our design decisions made sense.

14. The interface on the website was well organised and easy to navigate. While the main functionality of the website is undoubtedly the most important aspect, aesthetically pleasing websites are more appealing from a customer's perspective and help drive usage.

15. What do you like/dislike about the website layout? This was a deliberately subjective request for comments on the website's layout to find out if there were any parts the participant found confusing or unnecessary.

16. Is there any additional comments you would like to make? This was the opportunity for the user to mention any additional details that would not suit as an answer to any of the questions provided, but they feel we should be aware of.

Usability Study Outline

During our discussions we decided that once the registration process has been completed, the main part of the usability study should focus on the elements in the dashboard. However, test subjects were also asked to edit the profile and/or change some details, so that they would have enough information and user experience to be able to evaluate the system structure well.

Task 1: New User Registration

The screen in Figure 4.1 shows the homepage once the "Sign Up" button has been clicked, thus register action has been invoked. The Sign Up page is used in order for users to register on Road2Work website if they have not used the system before. The system asks for basic details, such as first name, last name, a valid email address and a password that needs to be confirmed by typing it in twice.



Figure 4.1: Sign Up view

The screen in Figure 4.2 shows the homepage once the user has entered their details and clicked “Sign Me Up”.

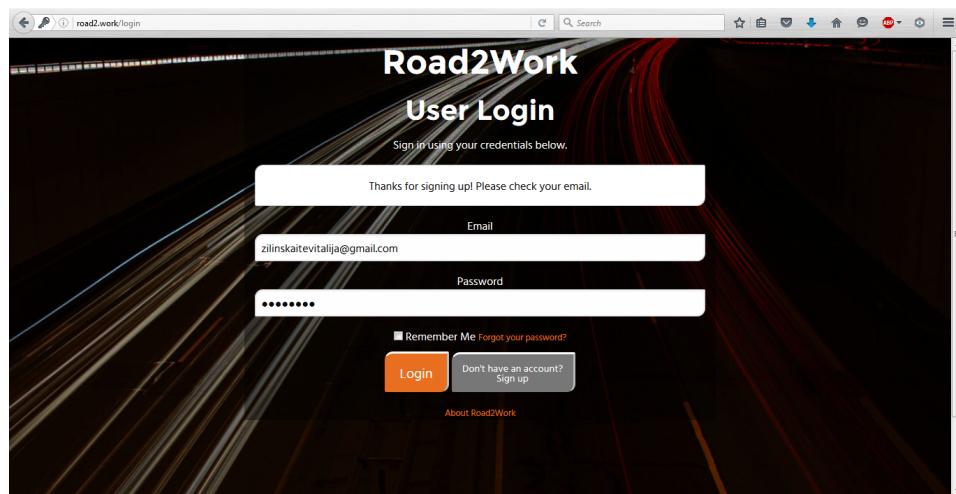


Figure 4.2: New User Activation view before the email address has been verified

Next up, if the user details are valid, the system suggests the user to check their email, as the verification link has been sent to that address. The user then has to open their email, and click the link that can be found in the email:

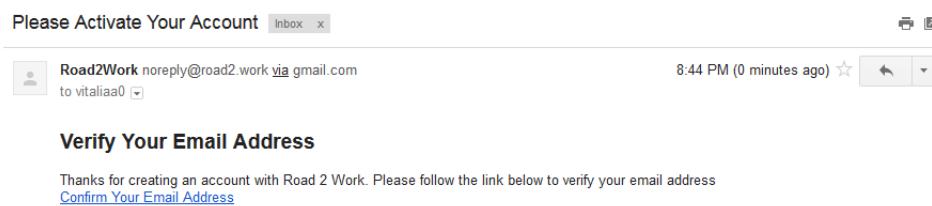


Figure 4.3: The email that contains verification link

Once the user has clicked the link in the email, homepage screen changes into the one we see in Figure 4.4. The system confirms a successful verification, which implies that Task 1 is complete.

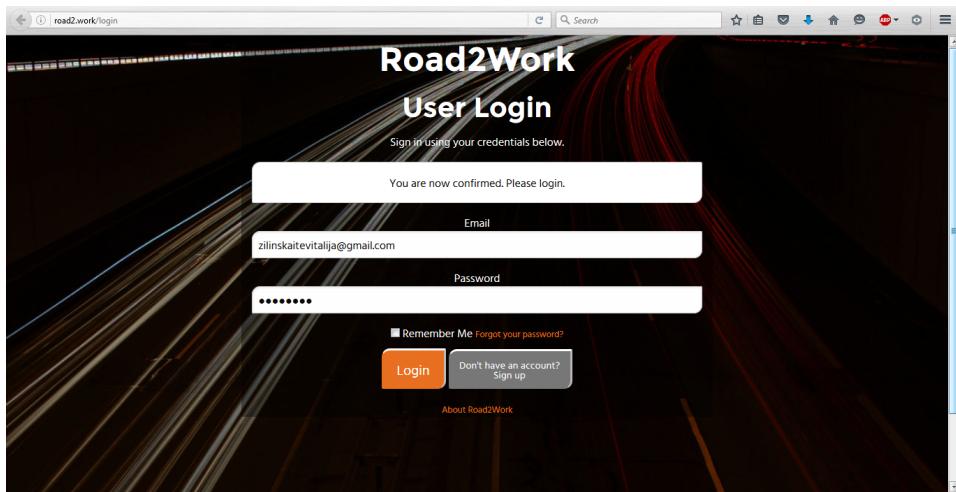


Figure 4.4: New User Activation view after the email address has been verified

Task 2: Obtaining a Pathway to the Chosen Career

The screen in Figure 4.5 shows the page for a registered user that has already filled out the remaining details for their profile. They have now decided to start a new pathway, but the goal career is yet to be chosen.

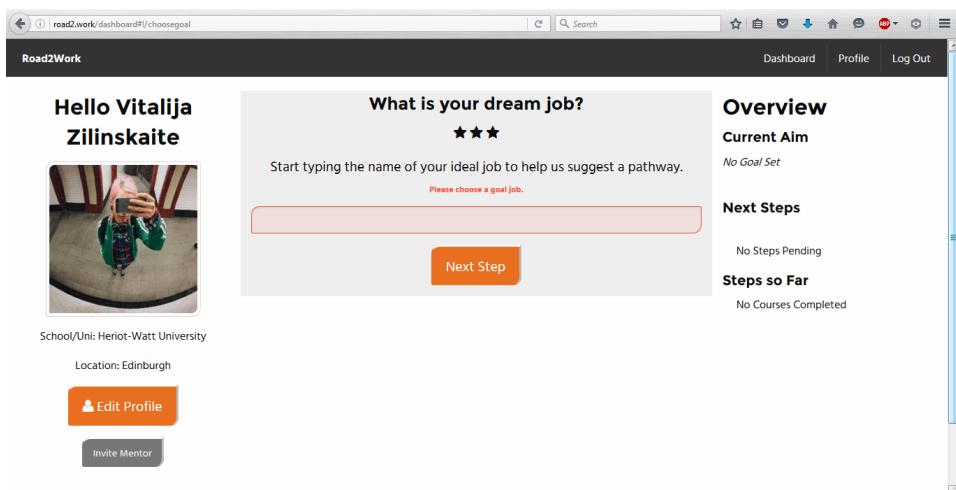


Figure 4.5: First step towards building the pathway

Once the user has decided on the desired career, it can be selected if it matches one of the 1143 options that our system contains in the database(Figure 4.6).

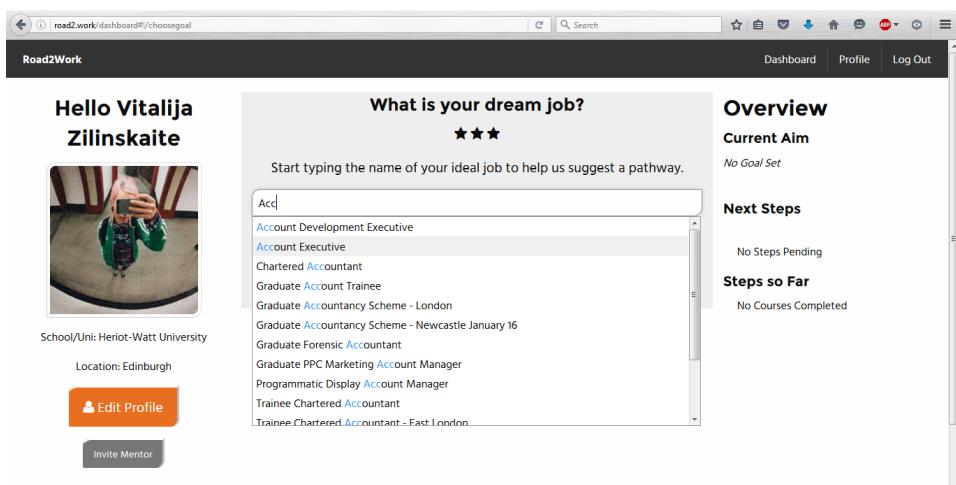


Figure 4.6: Road2Work helps the user choose one of 1143 available careers

To determine the best pathway, our system first asks the user to add any qualifications they have attained to date (Figure 4.7). The grades are all set up as A by default, but could be changed. If the subject name has been left empty, the website persists that the user enters a valid module name, and does not let them proceed beforehand (Figure 4.8). Once the courses attained have been entered in a valid format, our system dynamically calculates the UCAS points, and “Next Step” button can be clicked (Figure 4.9).

The screenshot shows the Road2Work dashboard. On the left, there's a profile section for 'Hello Vitalija Zilinskaite' with a photo, 'School/Uni: Heriot-Watt University', 'Location: Edinburgh', and buttons for 'Edit Profile' and 'Invite Mentor'. The main area is titled 'Your Grades' with a sub-instruction: 'Simply fill in your school grades which we use to determine the best way for you to reach your goal.' It lists 'Scottish Grades' including 'Intermediate 1', 'Intermediate 2', 'Scottish Advanced Higher', 'Scottish Higher', and 'Standard Grade', each with an 'Add' button. Below this is a section for 'English & Welsh Grades' with an '+'. To the right, there's an 'Overview' panel with 'Current Aim' (No Goal Set), 'Next Steps' (No Steps Pending), and 'Steps so Far' (No Courses Completed).

Figure 4.7: Available qualification types range from Scottish to Northern Ireland grades

This screenshot shows the 'Selected Grades' page. It displays two qualification entries under 'Scottish Advanced Higher': one for 'Mathematics' (Date Attained: 25/05/2013, Grade: A, Completed: checked) and another for 'e.g. Chemistry' (Date Attained: 25/05/2013, Grade: A, Completed: checked). A red error message 'Please enter a qualification title.' is visible next to the second entry. At the bottom is a large orange 'Next Step' button.

Figure 4.8: The user is not allowed to proceed if the input format is incorrect

This screenshot shows the 'Selected Grades' page again, but now with valid entries. It lists two 'Scottish Advanced Higher' qualifications: 'Mathematics' (Date Attained: 25/05/2013, Grade: A, Completed: checked) and 'Economics' (Date Attained: 25/05/2013, Grade: B, Completed: checked). The red error message from Figure 4.8 is no longer present. The 'Next Step' button at the bottom is now enabled and ready to be clicked.

Figure 4.9: The system calculates UCAS points as valid courses are being uploaded

The user is then presented with the relevant course options available(Figure 4.10). The button next to each course expands the panel and provides more information about each course and institution it is being taught at (Figure 4.11). The expanded section allows the user to visit relevant websites for the institution, as well as, for example, institution's direct location on the map (Figure 4.12).

The screenshot shows the Road2Work dashboard with a user profile for 'Hello Vitalija Zilinskaite'. On the left, there's a profile picture and basic information: 'School/Uni: Heriot-Watt University' and 'Location: Edinburgh'. Below this are buttons for 'Edit Profile' and 'Invite Mentor'. The main area is titled 'Course Suggestions' and lists several courses with their institutions and a '+' button to expand each entry. To the right, there's an 'Overview' section with 'Current Aim' (Account Executive), 'Next Steps' (No Steps Pending), and 'Steps so Far' (Scottish Advanced Higher Mathematics and Economics). A vertical sidebar on the far right has a scroll bar.

Course Title	Institution	Action
Accounting and Finance (Masters) Hons.	De Montfort University	+
Accounting with Economics Hons.	Edinburgh Napier University	+
Accounting Hons.	The University Of Essex	+
Accounting & Finance and Psychology Hons.	Liverpool Hope University	+
Accounting with Economics Hons.	The University Of Essex	+
Media Production & Moving Image and Accounting Hons.	The University of Northampton	+
Accounting and Finance Hons.	Stockton Riverside College	+
Accounting and Management Hons.	Kaplan Open Learning (Essex) Limited	+

Figure 4.10: Suggested course options, selected from 49732 available in the system's database

This screenshot shows a detailed view of a course suggestion for 'Accounting and Management' from 'Kaplan Open Learning (Essex) Limited'. It includes the college/university name, course ID, level, and a 'Choose Course' button. Below this, there are two more course suggestions: 'Accounting and Financial Management' and 'Accountancy'. Each course entry has a '+' button to expand for more details. The interface is similar to Figure 4.10 but focuses on one specific course.

Figure 4.11: The overview with more links is provided for each course

This screenshot shows the expanded view of the 'Accounting and Management' course from Figure 4.11. The 'View Institution' button has been clicked, revealing a detailed map of the 'University of Essex Colchester' campus. The map shows various buildings and landmarks, including 'Wivenhoe Park', 'North Towers', and 'University of Essex Colchester Campus'. It also includes a 'Directions' button and a 'Save' button. The map is integrated directly into the dashboard without requiring a full page reload.

Figure 4.12: Course location can be inspected without as little as reloading the page

If the previously entered qualifications accumulate enough UCAS points for the course the user decides to choose,

“Choose Course” directs the user to the individually calculated pathway (Figure 4.13), and sets it as a new default dashboard view. The system also updates the sidebar on the right hand side, updating the information needed. Hovering over each node of the pathway graph provides more information about the selected step (Figure 4.14). At this point Task 2 is considered complete.

Figure 4.13: The visualisation of newly calculated pathway completes Task 2

Figure 4.14: Each node of the timeline graph contains information about the corresponding step.

Results of Usability Study

N.B. Questions 1, 2 and 3 helped to collect quantitative objective data, since the information provided by user is general details about them, such as age, occupation etc.

The answers for questions 3-14 contain quantitative subjective data, since the questions asked user’s opinions about various parts of the system, and used Likert scales for their answers. These scales range 1 (“Strongly Agree”) through to 5 (“Strongly Disagree”).

Questions 15 and 16 asked the test subjects to write some additional comments about the layout and functionality of the system. This way we were able to obtain qualitative subjective data.

The usability test investigator measured the time taken to complete Task 1 and Task 2 for each test subject, collecting quantitative objective data.

Question	Candidate 1	Candidate 2	Candidate 3	Candidate 4	Candidate 5	Candidate 6
1	18	20	19	18	18	21
2	Student	Student	Student	Student	Student	Student
3	10-20	10-20	10-20	5-10	5-10	10-20
4	1	2	1	2	2	2
5	1	2	1	2	1	2
6	1	2	2	2	2	2
7	2	2	3	2	3	2
8	2	3	2	2	3	2
9	1	1	1	2	2	1
10	4	5	2	2	4	3
11	2	1	1	3	2	1
12	—	—	—	—	—	—
13	1	1	2	2	1	1
14	1	1	1	2	1	1
15	Dislike: once I chose the pathway, the “Start new pathway” option is too prominent.	Like: nice colours, simple layout, quick registration process. Dislike: pathway needs to be hovered over.	Like: nothing in particular, it was well laid out with easy colours, straight to the point.	—	Like: graph is nice.	Dislike: I didn’t see at first that it calculates UCAS points.
16	None	None	I thought that the timeline layout could’ve been more clear, like didn’t know I had to put my mouse over circles to find out about my steps. But overall I liked the idea of the website.	—	Would be easier if login fields were on the landing page, rather than having to click one of the buttons.	None

Figure 4.15: Usability test results

s	Candidate 1	Candidate 2	Candidate 3	Candidate 4	Candidate 5	Candidate 6
Task 1	55	104	145	89	49	75
Task 2	169	210	165	433	158	221

Figure 4.16: Usability test results showing the time taken for each candidate to complete the tasks in seconds

Test Study Analysis

We carried out our study on 6 subjects. *Nielsen Norman Group* [15] suggests that the best results come from testing 5 users, and supports their statement with a graph, provided in Figure 4.17. The curve clearly shows that it is best to test with at least 15 users to discover all the usability problems in the design. However, the real goal of usability engineering is to improve the design and not just to document its weaknesses. As a result, being able to recruit 11 representative customers, we split the study into 2 smaller tests (first one having been done in the Stage 1 of our project) instead of a single, elaborate study. We have also been suggested by the proxy customer to test on 6-7 subjects at a time, hence this study has been carried out not on 5, as suggested by *Nielsen Norman Group*, but on 6 subjects.

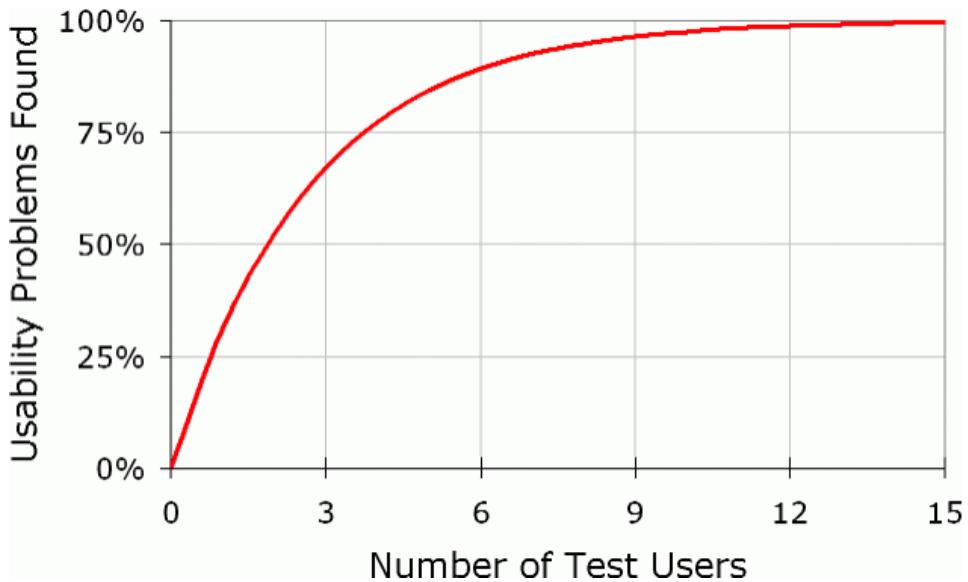


Figure 4.17: The curve illustrating the number of usability problems found in a usability test with n users (Tom Landauer and Jakob Nielsen, 2000)

From the results in Figure 4.15 we have identified one clear issue with the current design, along with the three smaller ones, which are all expanded in the “Problems Identified” section below. We have also been suggested that having login fields right away on the landing page rather than displaying them when the “Log In” button is clicked would have cleared some confusion. We will take this into consideration in the further development of Road2Work. The Likert scale results are mostly positive, with most subjects having agreed that the registration process is intuitive and the general interface of the website is easy to navigate through. One third of all subjects have explicitly mentioned the colours of the user interface as something they liked, along with the layout being “simple” or “well laid out”.

The results in Figure 4.16 helped us determine the quality of the registration process, and the complexity of obtaining a pathway to the chosen career. According to Appsee’s [1] Mobile User Behavior study (2015), the average time user spends on the registration screen is 36 seconds. The study has also found that the average number of text fields in apps with email registration is 3.8. These numbers are significantly smaller than the results we obtained in our usability study with the average registration time being 86 seconds, along with the 5 fields presented in the registration form. The problems identified using these results are addressed in the “Problems Identified” section below.

As well as having measured the new user registration time in our study, we collected the data on the time taken to obtain the career pathway as well. The average time the user takes to get from choosing their dream career to having an individual pathway presented is 226 seconds. We have used this result combined with the subjects’ answers to questions 7 (“The process of entering current qualifications was intuitive”), 8 (“The pathway display represented the data well”) and 9 (“It was easy to understand the pathway to the chosen career”) that can be found in Figure 4.15. The average answer for these three questions combined is 1.97, which on the provided Likert scale stands for “Agree”. This result shows a high customer satisfaction of the process to obtain the pathway, as well as the process being easy to understand. This means that the process takes just enough time, and we can use that time measure for the future development of our system.

Problems Identified

Registration The analysis of usability test results in Figure 4.16 has created concerns about the time it takes the user to register a new account. The average time it took our users to perform the task is 86 seconds, which is 50 seconds more than the average mobile user registration, which is 36 seconds [1]. Despite the fact that we also require more information than the average, this is mostly due email verification that we use for every new user. While this is a highly useful feature for desktop users, the mobile users of our website would possibly encounter difficulties trying to register. Despite Road2Work being fully responsive, we have yet to discover and implement a smooth new user verification process when the system is being used on the phone.

Pathway Representation There was one clear issue that our users encountered; despite most test subjects having agreed that the pathway display represented the data well (see: question 8 in Figure 4.15), over 30% of all users mentioned having had difficulties understanding the pathway visualisation provided. The users have all stated that the currently completed steps as well as the steps yet to be complete, were not obvious from the graph. This was because once the pathway has been presented, the user needs to hover over the graph node (*i.e.* a step) to discover more information about it. We see this as a primary feature that needs to be updated in the following version of our

system, as the visualisation is one of the most important parts of the website. However, this will not be a difficult thing to fix, since as little as a text box suggesting the user to hover over the circle for more information about each step, would make a great difference. Another solution to this problem would be different colours for completed/future steps, as well as the text displayed right away, without having to hover over the node.

Obtaining the Pathway There were two minor issues that our users encountered building and then having obtained the pathway. First of all, the presentation of UCAS points being calculated is not easily seen by the user who has not visited the website before, and thus is not aware of it being there. We take this feedback very seriously, although the fix will be quick; we will make sure the system notifies the user about their UCAS points being calculated, as well as make sure the website displays the points more prominently.

Another problematic detail one of the test subjects has mentioned is the “Start New Pathway” button. Since it is just as big and prominent both before and after the pathway calculation, we have been suggested that this creates confusion once the pathway has been obtained. The user can be easily confused and intuitively want to click the button again, even though their pathway is already presented. Part of this issue is related to the pathway representation issue discussed above, but it will also be beneficial to make the button smaller or change the text to “Change Your Pathway” in the future versions of our system.

Conclusion

To conclude the findings of our usability study, the candidates had a mostly positive reaction to Road2Work. Overall the website layout was well received with the users finding it easy to interact with and find their way around the site. The issues that some users brought up have been considered and we have put in place a plan to alleviate them which will be incorporated into our system design before we continue the development process.

In particular we need to ensure that the final visualisation of the pathway is clear and concise. Furthermore, we need to ensure that the system runs just as smoothly on mobile phones as it does on the desktop computers. We will review our decisions in the problematic areas and examine alternative solutions which will hopefully offer a better user experience.

5. Project Evaluation

5.1 Organisation

We worked well together as group, our progress through the project was helped by our adoption of scrum principles to try and encourage and incorporate collaborative work. We aimed to work across different parts of the system to increase the speed at which we could develop, to share and build upon our individual knowledge to create a team of well rounded and well informed developers. Roman used his skills he has gained from his Information Systems studies to produce a marketing document, he stayed in sync with us during our development, steering the direction of the product towards the people that he planned to market it to. The rest of us through our Computer Science and Systems study have picked up development skills, with some of us working externally as programmers and we managed to pick up on knowledge we have gained from these sources and apply it collectively to further the project.

The group has employed a variety of methods and tools in order to support communication and collaboration between members. Details of these are outlined below:

1. **Facebook** - We used a private Facebook group where all members can communicate and post announcements via the Group wall. This was useful since we could easily refer to posts at a later date. The Figure 5.1 below shows where this group would come in use: one of the group members could easily share the information with all of us about the software we had decided to use, along with the needed links and comments provided. A group chat was also set up on Facebook so that instant communication can be easily managed between the group. It was the most useful for quick communication to clarify details when working on collaborative tasks and documents, as well as organising the meet-ups.

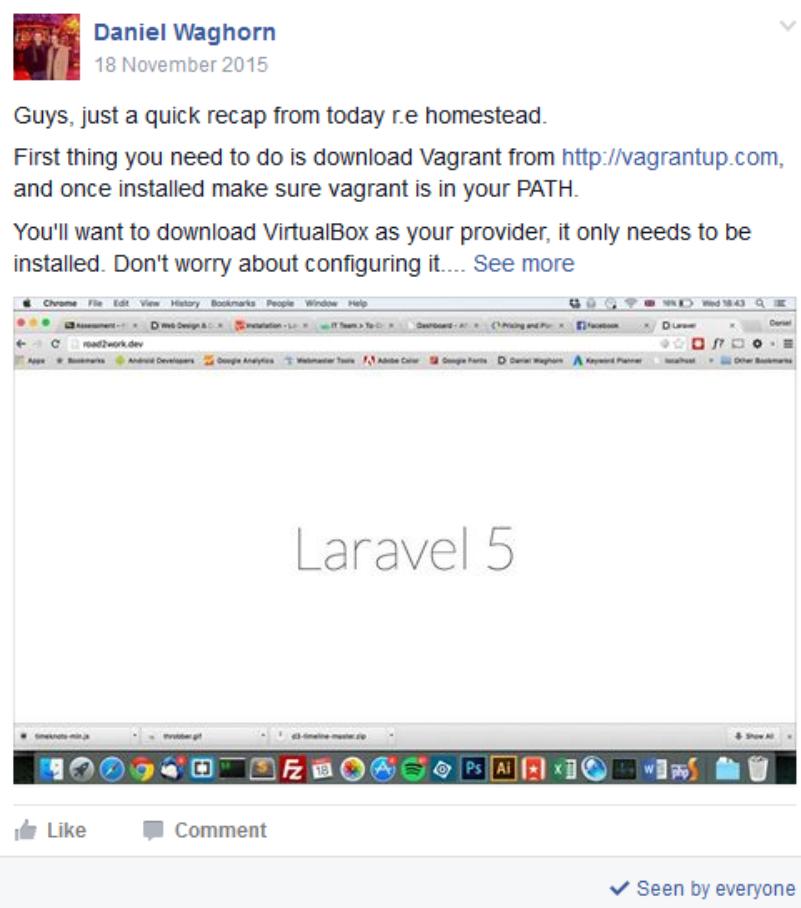


Figure 5.1: An example of our Facebook group being use

2. **Wunderlist** - At the heart of our task management was a to-do list so that all members could see the tasks that needed to be carried out. This helped keep the group focused and organised. Wunderlist also offers a number of highly useful collaborative features, such as the ability to assign tasks to other members, comment on individual items and add files, as well as break larger tasks down into sub-tasks. Wunderlist is not only a

desktop application that could be easily accessed on the computer, but it is also available as a mobile app as we can see in Figure 5.2. This means that all the group users were in-sync with the project development updates wherever they were, and would get a notification whenever a certain task would be finished or assigned to them.

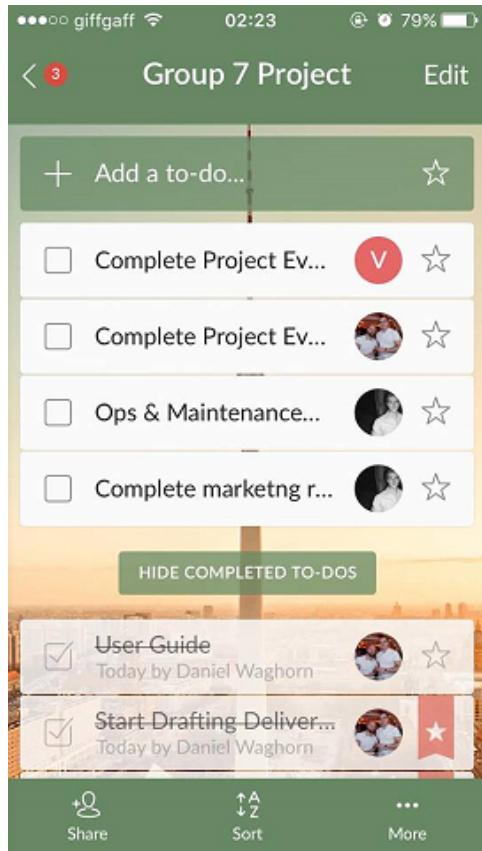


Figure 5.2: Wunderlist on mobile showing what tasks still need to be done for the project

3. GitHub - Version Control is essential whenever undertaking a large group project, and GitHub seemed like an obvious choice. Git is simple yet powerful and in most cases favourable over other systems such as SVN and CVS. The reason we used Git and GitHub in this case is firstly Git is simpler to use than Subversion which most group members had first exposure to in first year with mixed feelings. GitHub also offered a number of private repositories free to students which are ideal for holding our project files whilst offering a whole host of collaborative features such as automatic merging, branches, issue tracking and more.

Version Control is not only useful for source code development, but also enabled multiple members of group to work on important documentation files at the same time, later seamlessly merging the updated versions. We were also able to revert to previous file versions when required. The details of commits to our Version Control System can be found in the Appendix ??.

4. Wordpress - While GitHub is an excellent tool to keep track of the technical details of the project as it progresses, a private blog set up on Wordpress enabled us to make sure that the general group work progress was being made consistently. The entries in this diary were only made for each time we met up and worked together, whereas the actual work done, together or separately, is documented in Version Control System, so it was easy to make sure we were working together often enough. We were also able to make sure that the meetings with the Project Manager were initiated when needed. From the entry summary that can be found in the Appendix ??, we can see that the beginning of the project was more documentation-oriented, slowly moving towards the meetings being strictly technical, where we would focus solely on the system development. This worked incredibly well, as having thoroughly prepared project planning documents in the early stages helped us develop the system rapidly and always making sure we meet all the requirements. It can also be seen from the diary that as the project progressed, we needed less meetings with the Project Manager. This was due to all the requirements and general project development details having been cleared up in the early stages of the project, meaning we could focus more on the system development.

5.2 Implementation

In terms of implementation we believe that we have produced a robust, scalable and concise solution that meets the key requirements outlined in the project specification. The tactical use of software frameworks meant that we could focus primarily on the application and business logic rather than spending time on routine tasks such as user authentication.

Whilst implementing the application the team worked diligently with a great deal of collaboration, often in pairs so we had a degree of continuous code review.

The modular nature of the application ensures that future amendments and features can be implemented quickly whilst minimising potential side effects and keeping the possibility of introducing bugs to a minimum. We made an effort to use open source software wherever possible as well as an industry standard stack. This results in having a large pool of talent from which we can employ additional developers in the future if the business sees the requirement to do so.

A minor improvement to the implementation which could be implemented in the future could be the addition of a graph database to the back-end as this would allow us to produce a more powerful decision engine which could offer better suggestions to users and allow us to derive trends from data within the system which can be used to further improve the product through research and development.

5.3 Product

We feel our product has a lot to offer over similar products on the market. It's strength lie in it's extra features such as the search engine to find courses relevant to the users chosen job, this means that not only does our product help users to achieve their goals, but it can also make them aware of courses at institutions they may have not been aware of in the first place. We feel our product has a layer of polish, which adds professionalism and ease of use for the user, it is also written in a modular way with little coupling so it can be extended if new features become a demand from the market. There are a few issues that have arisen with the final build, the search engine could take another few months of solid development time alone to get to a level where it was incredibly advanced and similar to search engines at other industry leading job sites. However outside of this our final usability study reported that our subjects found the product very usable, with little confusion or ease of access issues, we feel this is down to the strength of the hand holding wizard we have implemented to guide our users through the process from beginning to a complete profile.

5.3.1 Functional Requirements

The table below provides and overview of the functional requirements achieved throughout the project. The Completion column provides brief information about the extent of the completion of each requirement:

- Fully Completed - Requirements that have full functionality with error checks.
- Completed - Requirements that have been completed and provide functionality, but not many error checks.
- Partial - Requirements that have been partially completed and provide some functionality, but not full due to dependencies on other requirements.
- Not Completed - Not completed requirements.
- Removed - Removed requirements

No.	Description	Priority	Completion
f1.	The system will present a landing page which offers various options and calls to action	Must Have	Fully Completed
f2.	The system will offer the user the option to log in	Must Have	Fully Completed
f3.	The system will have a forgotten password facility for existing users	Must Have	Fully Completed
f4.	The system could offer the user the option to use a social media account (Facebook, Google) to log in	Could Have	Removed
f5.	The system will offer the user to create a new account	Must Have	Fully Completed
f6.	The system could offer the user the option to use social media (Facebook, Google) to register a new account	Could Have	Removed
f7.	The system will ensure that the user has a valid email address before allowing them to sign in	Must Have	Fully Completed

f8.	The system will use email verification to ensure the user has an active account	Must Have	Fully Completed
f9.	The system will allow the user to choose their areas of interest	Must Have	Fully Completed
f10.	The system will have a set of mandatory information required to register a new user	Must Have	Fully Completed
f11.	The system will have two kinds of users: primary (student) and secondary (mentor, teacher, parents)	Must Have	Fully Completed
f12.	The system will offer secondary users an option to give recommendations to students	Must Have	Not Completed
f13.	The system will differentiate between the primary and secondary users permissions	Must Have	Fully completed
f14.	The system will lead a new user through a profile creation wizard	Must Have	Fully Completed
f15.	The system will require less personal information from secondary users	Must Have	Partial
f16.	The system will allow the user to edit information in their profile at a later date	Must Have	Fully Completed
f17.	The system could restrict the information the user can change	Could Have	Fully Completed
f18.	The system will check the user authentication status on all "protected" areas, and restrict data to those with access rights	Must Have	Fully Completed
f19.	The system will be able to send emails to the user	Must Have	Fully Completed
f20.	The system will alert the user if there are any new opportunities in their area of interest	Must Have	Not Completed
f21.	The system will alert the user if any opportunity of their interest is no longer available	Must Have	Not Completed
f22.	The system will allow the user to opt out from receiving alert emails	Must Have	Not Completed
f23.	The system will hold a profile for each primary user	Must Have	Fully Completed
f24.	The system will offer multiple suggestions on how to <i>improve</i> a learner's employability and reach a target goal	Must Have	Not Completed
f25.	The system will display a learner's current progression and journey to date	Must Have	Fully Completed
f26.	The system will display suggestions in a clear and concise manner whilst presenting key information effectively	Must Have	Needs Improvement
f27.	The system could offer a degree of simulation / interactivity in order to explore options presented at later stages	Could Have	Not Completed
f28.	The system could show the cost of achieving a particular pathway	Could Have	Not Completed
f29.	The system will allow authorized secondary users a detailed view of a user's pathway	Must Have	Partial
f30.	The system will notify the user by email about any changes in the current pathway	Must Have	Not Completed
f31.	The system will offer alternative pathways to users where grades or qualifications do not satisfy desired pathways	Must Have	Partial
f32.	The system will feature an administrative back-end for Admin users	Must Have	Fully Completed
f33.	The system will feature a CMS as part of the administrative back-end for editing static page content	Must Have	Fully Completed
f34.	The system will support linking secondary accounts to "mentor" primary accounts	Must Have	Completed
f35.	The system will allow Admin users to create, edit and delete user accounts from the back-end	Must Have	Fully Completed
f36.	The CMS feature of the system will require back-end admin users to be authenticated by a user ID and password	Must Have	Fully Completed
f37.	The CMS will have a backup feature	Must Have	Not Completed
f38.	The CMS should be easy to use and provide an intuitive graphical user interface	Should Have	Fully Completed

User

Most of the User requirements were achieved and below are detailed the ones with partial functionality and the not completed ones.

- f4. The system could offer the user the option to use a social media account (Facebook, Google) to log in

- **f6.** The system could offer the user the option to use social media (Facebook, Google) to register a new account

Both of these requirements were removed after a discussion about privacy and personal details in one of the lectures. It was decided that these features were no longer mandatory on the basis of privacy concerns through the use of real personal data for the project.

- **f12.** The system will offer secondary users an option to give recommendations to students
- **f20.** The system will alert the user if there are any new opportunities in their area of interest
- **f21.** The system will alert the user if any opportunity of their interest is no longer available
- **f22.** The system will allow the user to opt out from receiving alert emails

These four requirements are interdependent and thus the non completion of one would result in complications in the others. Requirement **f12** was not implemented because, the implementation of secondary users was partial.

Pathway

- **f24.** The system will offer multiple suggestions on how to *improve* a learner's employability and reach a target goal.

Although this was not technically achieved the system provided suggestions when a learner has still not completed enough courses to progress by calculating UCAS points needed for progression.

- **f27.** The system could offer a degree of simulation / interactivity in order to explore options presented at later stages.

This function was not implemented to the desired degree and thus is marked as uncompleted.

- **f28.** The system could show the cost of achieving a particular pathway.

The system provided a number of links to Institutions and Courses where costing information was displayed, but the system did not have a certain feature to show that in the Pathway.

- **f30.** The system will notify the user by email about any changes in the current pathway.

As the system did not have automated updating data sets it did not send emails about changes. But if in the future the data is updated automatically the email feature can be easily implemented with the use of the Mail Controller.

Administrative

Most of the Administrative requirements were achieved and below are detailed the ones not completed.

- **f37.** The CMS will have a backup feature.

5.3.2 Non-Functional Requirements

The table below provides an overview of the non-functional requirements achieved throughout the project. The Completion column provides brief information about the extent of the completion of each requirement:

- Fully Completed - Requirements that have full functionality with error checks.
- Completed - Requirements that have been completed and provide functionality, but not many error checks.
- Partial - Requirements that have been partially completed and provide some functionality, but not full due to dependencies on other requirements.
- Not Completed - Not completed requirements.

No.	Description	Priority	Completion
nf1.	The system will make use of responsive design principles to ensure it displays well on all devices	Must Have	Fully Completed
nf2.	The system will display correctly in Internet Explorer, Firefox, Chrome and Safari without visual defects caused by differing browser implementations	Must have	Fully Completed
nf3.	The system will incorporate good UX practices to ensure that user interaction requires minimal effort and is simple	Must Have	Fully Completed
nf4.	The system will be built on established web standards and make use of accessibility features to improve UX for impaired / disabled users	Must Have	Completed
nf5.	The system will aim to be aesthetically pleasing to users	Must Have	Completed
nf6.	The system will ask the user to save currently edited content before they navigate away from the page in order to save them from possibly losing work	Must Have	Completed
nf7.	The system will securely (encrypted) store personally identifiable information.	Must Have	Not Completed
nf8.	The system will hash user passwords	Must Have	Fully Completed
nf9.	The system will encrypt the users sessions	Must Have	Fully Completed
nf10.	The system will make use of CSRF protection	Must Have	Fully Completed
nf11.	The system will authenticate users before allowing access to restricted areas	Must Have	Fully Completed
nf12.	The system could have data loss prevention	Could Have	Completed
nf13.	The system could have a timeout period after a certain amount of login attempts	Could Have	Not Completed
nf14.	The system could have an audit log with date, time, user, action, prior value and new value	Could Have	Partial
nf15.	The audit log could have integrity protection	Could Have	Partial
nf16.	The system will allow multiple users to access the site at the same time	Must Have	Fully Completed
nf17.	The system will be available 24 hours a day, 365 days per year	Must Have	Fully Completed
nf18.	The system will not have more than 3 hours of scheduled downtime per month	Must Have	Completed
nf19.	The system will not have more than an average of 1 hour of unscheduled downtime per month	Must Have	Completed
nf20.	The system will support 500 concurrent users without performance degrading	Must Have	Completed
nf21.	The system will complete the log in process (providing that the log in details are correct) within 5 seconds	Must Have	Fully Completed
nf22.	The system could display a message if it is unable to log in a customer within 5 seconds	Could Have	Not Completed
nf23.	The system could try to log in a customer automatically unless it fails to do so at first attempt for whatever reason	Could Have	Fully Completed
nf24.	The system will take 2 seconds or less to send an email for account verification	Must Have	Fully Completed
nf25.	The system will store user records	Must Have	Completed
nf26.	The system could store users' qualifications and pathways in a Data Warehouse	Would Like To Have	Not Completed
nf27.	The system could deactivate a user's account after 730 days without logging in	Could Have	Not Completed

Security

- **nf7.** The system will securely (encrypted) store personally identifiable information.

This feature was not implemented.

- **nf13.** The system could have a timeout period after a certain amount of login attempts.

Although the system does not have a time out feature it will begin to reject user log in after certain amount of failed attempts as it is rate limited.

Preformance/Availability

- **nf22.** The system could display a message if it is unable to log in a customer within 5 seconds.

This feature was not implemented.

- **nf26.** The system could store users' qualifications and pathways in a Data Warehouse

- **nf27.** The system could deactivate a user's account after 730 days without logging in.

This feature could be easily implemented with the use of a CRON to check user log in timestamps.

6. Guides

This chapter provides a general overview of the system written in non-technical terminology. It contains a User Guide and an Operations and Maintenance Guide. The summary outlines the uses of the system in supporting the activities of the user and staff. Briefly describes and depicts graphically the tools, communications, and networks used by the system.

6.1 User guide

The User Guide contains all essential information for the user to make full use of the system. This manual includes a description of the system's functions and capabilities, contingencies and alternate modes of operation, and step-by-step procedures for system access and use.

The chart below describes the basic flow of the system.

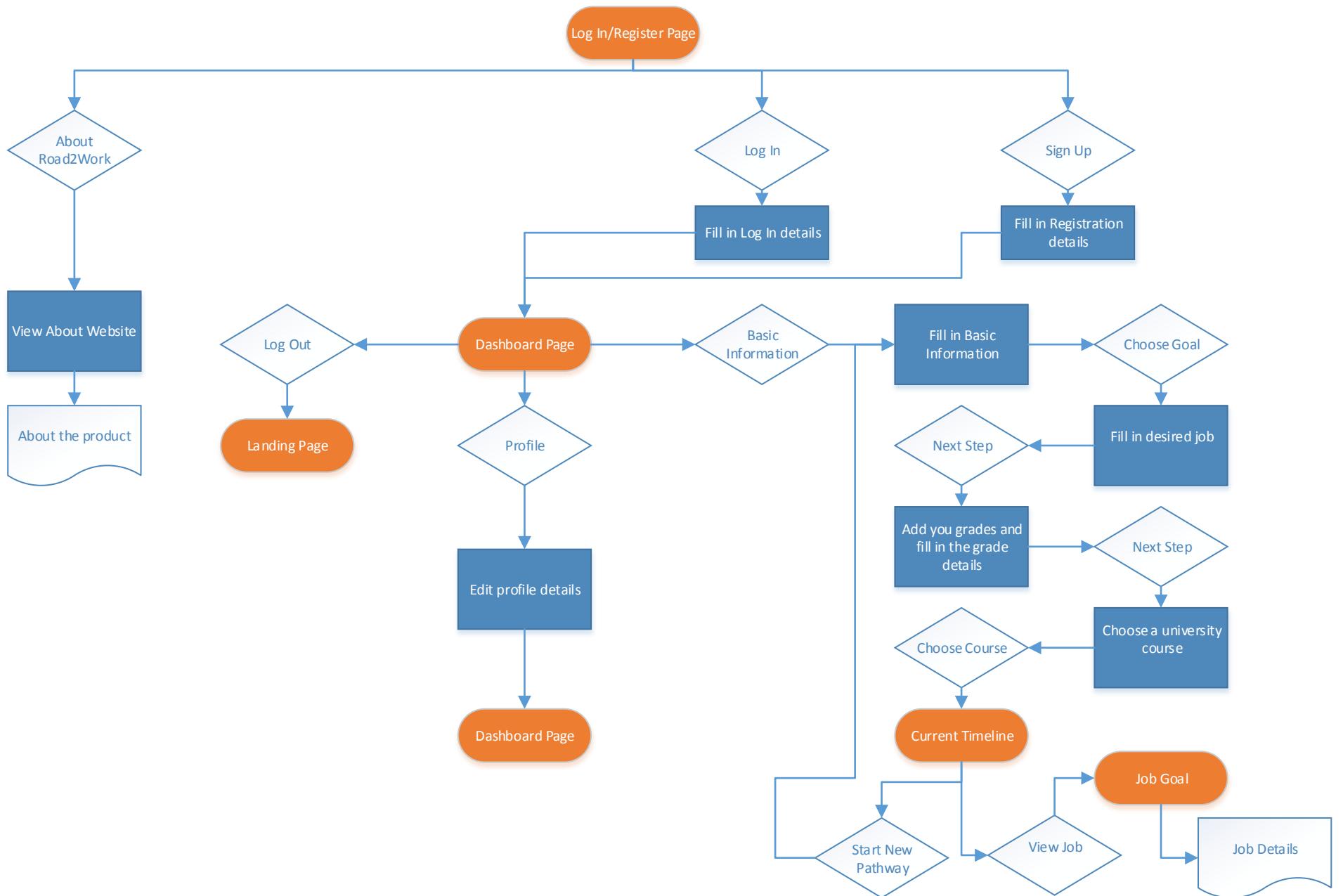




Figure 6.1: Landing Page

1. Landing page that provides the option to Log In and Sign Up.

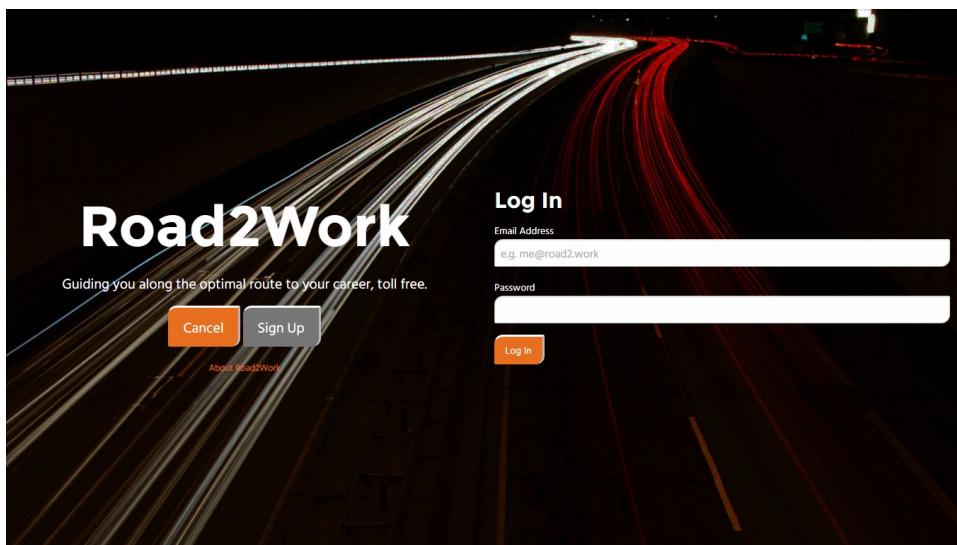


Figure 6.2: Landing Page - Log In

2. Once the Log In button is pressed the Log In form shows up providing the ability to log into the website via Log In details (Username, Password).

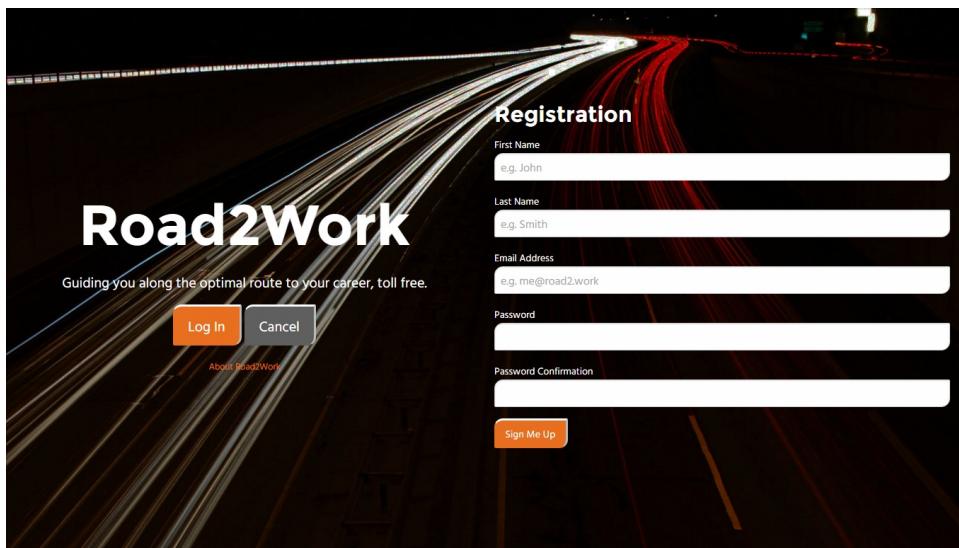


Figure 6.3: Landing Page - Sign Up

3. Pressing the Sign Up button brings up the Registration form, which requires First Name, Last Name, Email Address, Password and Password Confirmation.

Figure 6.4: Dashboard - Basic Information

4. After signing into the website you are welcomed by a Basic Information form where you can supply Your Date of Birth, Place of Study and Location. Clicking the Choose Goal button directs you to the Choose Goal Page.

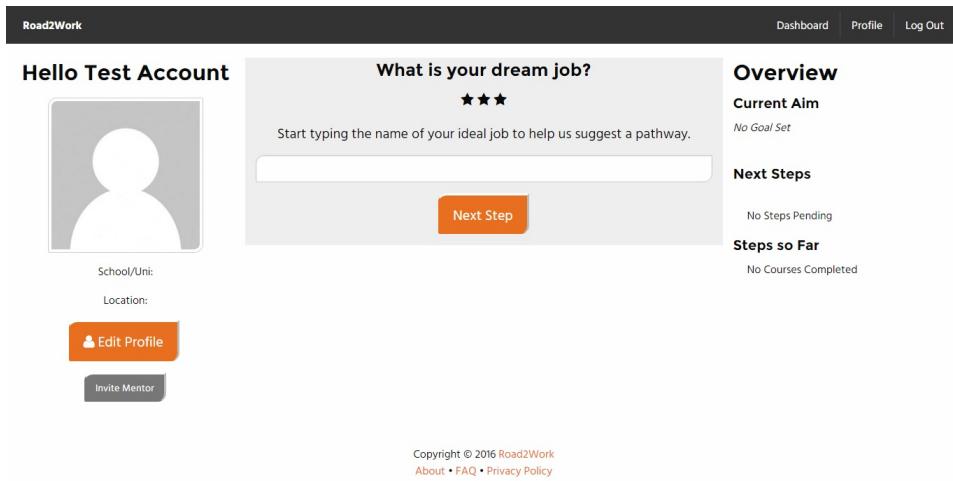


Figure 6.5: Dashboard - Dream Job

5. Here the user can input their desired job as shown in the next figure.

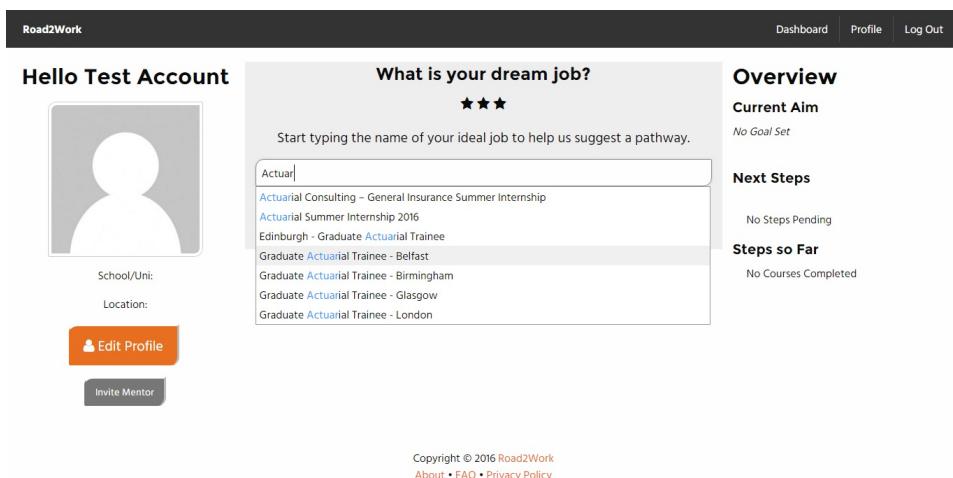


Figure 6.6: Dashboard - Dream Job Input

Hello Test Account

What is your dream job?
★★★
Start typing the name of your ideal job to help us suggest a pathway.

Graduate Actuarial Trainee - Belfast
Your chosen career: Graduate Actuarial Trainee - Belfast

School/Uni:
Location:
Edit Profile
Invite Mentor

Overview
Current Aim
No Goal Set

Next Steps
No Steps Pending

Steps so Far
No Courses Completed

Copyright © 2016 Road2Work
[About](#) • [FAQ](#) • [Privacy Policy](#)

Figure 6.7: Dashboard - Dream Job Selected

- Once you have selected a job you can progress with setting up your account by pressing Next Step.

Hello Test Account

Your Grades
Simply fill in your school grades which we use to determine the best way for you to reach your goal.

Scottish Grades
Intermediate 1 Add
Intermediate 2 Add
Scottish Advanced Higher Add
Scottish Higher Add
Standard Grade Add

English & Welsh Grades +

Northern Irish Grades +

Overview
Current Aim
No Goal Set

Next Steps
No Steps Pending

Steps so Far
No Courses Completed

Next Step

Figure 6.8: Dashboard - Your Grades

- After selecting your desired career the system will prompt for your grades. Grades can be chosen from the menu and added by using the Add button.

Northern Irish Grades

Selected Grades

You have 120 UCAS points.

Scottish Advanced Higher ✖ Remove Qualification

Subject Name: Computing

Date Attained (Optional): 17/05/2013

Grade: A

Completed?:

A level with additional AS(9 units) ✖ Remove Qualification

Subject Name: e.g. Chemistry

Please enter a qualification title.

Date Attained (Optional): dd/mm/yyyy

Grade: A*A

Completed?:

Next Step

Figure 6.9: Dashboard - Your Grades Selected

8. Selected grades require you to input a Subject Name. The field Date Attained is optional. The checkbox Completed? is to show whether that subject has completed or is still in progress. Qualifications can be easily removed by the X Remove Qualification link. Once the grades have been selected you can continue by clicking Next Step.

Road2Work

Hello Test Account

School/Uni:

Location:

Edit Profile

Course Suggestions

Here's our course suggestions for you. Click on the course title to get more information about the course.

Course Title	Institution	
Graduate of the RNCM	Royal Northern College Of Music	+
Actuarial Science and Mathematics Hons.	University of Manchester	+
Actuarial Science Hons.	The University Of Kent	+
Actuarial Science and Risk Management Hons.	Queen's University Of Belfast	+
Actuarial Science Hons.	University of York	+
Undergraduate Masters in Chiropractic (Extended)	Bpp University Limited	+
Actuarial Science Hons.	Newham Training And Education Centre	+
Professional Graduate Certificate in Education (Lifelong Learning Hons.)	Farnborough College Of Technology	+
Actuarial Mathematics and Statistics Hons.	Kingston University	+
Actuarial Mathematics & Statistics Hons.	Newham Training And Education Centre	+

Overview

Current Aim
No Goal Set

Next Steps
No Steps Pending

Steps so Far
No Courses Completed

Figure 6.10: Dashboard - Course Suggestions

9. Your application might lead you to a Course Suggestions page depending on the choice of job. If your selected job is Work Experience, Voluntary or an Apprenticeship this section will be skipped. View and selecting a course is as easy as pressing the + (Plus) button and then Choose Course.

The screenshot shows a list of course options for 'Actuarial Science and Diploma in Industrial Training'. The courses listed are:

- Actuarial Science Hons. at The University Of Essex
- Actuarial Science Hons. at The University Of Essex
- Actuarial Sciences Hons. at University of East Anglia
- Actuarial Science and Diploma in Industrial Training Hons. at Heriot-Watt University

Below the list, there is a section titled 'Actuarial Science and Diploma in Industrial Training' with the following details:

- FullTime
- College / University: Heriot-Watt University
- Course ID: 201617_5G5B
- Level: BSc Hons.

Below this section are four buttons:

- [View Institution](#)
- [View Course Webpage](#)
- [View Funding Information](#)
- [View Employment Info.](#)
- [View Assessment Methods](#)

A large green button in the center says 'Choose Course' with a thumbs-up icon.

Figure 6.11: Dashboard - View Course

- Viewing a course provides some basic information about that course and links to the course webpage, the assessment methods of that institution and funding information. Clicking View Institution provides information about that institution as detailed below.

The screenshot shows a map of Heriot-Watt University's Edinburgh Campus. The map includes labels for 'Heriot-Watt University, Edinburgh Campus, Edinburgh EH14 4AS, United Kingdom', '4.5 ★★★★☆ 74 reviews', 'View larger map', 'Directions', 'Save', 'Heriot-Watt University Student Union', 'The Avenue', 'Heriot Watt University Student Pad', 'Cameron Small Rd', 'Centre for Sport and Exercise', and 'Edinburgh Business School'. Below the map are several buttons:

- [Visit Website](#)
- [Visit Student Union](#)
- [View Course Webpage](#)
- [View Funding Information](#)
- [View Employment Info.](#)
- [View Assessment Methods](#)

A large green button in the center says 'Choose Course' with a thumbs-up icon.

Figure 6.12: Dashboard - View Institution

The screenshot shows the Road2Work dashboard. At the top, there are navigation links: Dashboard, Profile, and Log Out. On the left, a sidebar for 'Hello Test Account' includes fields for School/Uni (Heriot-Watt University) and Location (Edinburgh), with 'Edit Profile' and 'Invite Mentor' buttons. The main content area is titled 'Additional Courses Required'. It displays a message about not meeting course requirements and lists 'Scottish Grades' and 'English & Welsh Grades'. Below this is a table of courses with 'Add' buttons:

Course	Action
A level with additional AS(9 units)	Add
GCE & AVCE Double Award	Add
GCE A level and AVCE	Add
GCE AS & AS VCE	Add
GCE AS Double Award	Add

To the right, the 'Overview' section shows 'Current Aim' (No Goal Set), 'Next Steps' (No Steps Pending), and 'Steps so Far' (No Courses Completed).

Figure 6.13: Dashboard - Additional Courses Required

11. This page will be displayed only if you are required to select more grades because of the chosen University or College course.

The screenshot shows the Road2Work dashboard. At the top, there are navigation links: Dashboard, Profile, and Log Out. On the left, a sidebar for 'Hello Test Account' includes fields for School/Uni (Heriot-Watt University) and Location (Edinburgh), with 'Edit Profile' and 'Invite Mentor' buttons. The main content area is titled 'Current Timeline'. It shows a timeline with five stages represented by orange circles connected by a horizontal line. A callout box over the first stage says: 'Pending Course: A level with additional AS(9 units)', 'Current Aim: Chemistry', and 'Required Grade: A*A'. Below the timeline are buttons for 'Start new Pathway' and 'View Job'. The 'Overview' section on the right provides details about the current aim (Graduate Actuarial Trainee - Belfast), next steps (A level with additional AS(9 units) Chemistry, A level with additional AS(9 units) Test, and BSc), and completed steps (Scottish Advanced Higher Computing). Copyright information at the bottom includes: Copyright © 2016 Road2Work, About, FAQ, and Privacy Policy.

Figure 6.14: Dashboard - Current Timeline

12. After selecting your University/College course you are directed to the Current Timeline page. This page provides information about your pathway as well as the ability to start a new pathway by the Start new Pathway button. Each one of the circles represents a stage in your path. The Overview section on the right provides overview information about your path. Current Aim represents the job you have selected, Next Steps are the courses that you need to tackle in order to get closer to your goal and the Steps so Far are you completed steps. The Current Timeline provides a button for viewing your selected job as detailed below.

The screenshot shows the Road2Work dashboard with the following details:

- Hello Test Account**: User profile section with a placeholder profile picture, school/university (Heriot-Watt University), location (Edinburgh), and buttons for "Edit Profile" and "Invite Mentor".
- Your Goal Job**: Job listing for "Graduate Actuarial Trainee - Belfast" with projected salary (competitive), job type (graduate scheme), and location (Belfast).
- Map**: A map of Belfast, Northern Ireland, showing various neighborhoods and roads.
- Potential Employers**: A list including "Mercer" with a "View My Pathway" button.
- Overview**: Sections for "Current Aim" (Graduate Actuarial Trainee - Belfast) and "Next Steps" (A level with additional AS(9 units) Chemistry, Required Grade A/A; A level with additional AS(9 units) Test, Required Grade A/A). It also lists "Steps so Far" (Scottish Advanced Higher Computing).

Figure 6.15: Dashboard - View Job

13. This page provides basic job information and potential employers.

6.2 Operations and Maintenance Guide

6.2.1 Admin Guide

The Admin Guide contains all essential information for the admin to make full use of the system.

The screenshot shows the Road2Work admin welcome page with the following structure:

- Navigation Bar**: Includes links for "Road2Work admin", "Home", "See users", "Data Upload", "View data", "Manage Pages", "Dashboard", and "Log Out".
- Welcome Message**: "Welcome to the Road2Work admin facility. Choose a section from the menu below to get started."
- Buttons**: "View Users", "Upload Data", and "Manage Pages".
- Page Footer**: Copyright © 2016 Road2Work, About • FAQ • Privacy Policy.

Figure 6.16: Admin - Welcome Page

Upon arriving at the Admin page the user is greeted by a welcome page. From here they can choose to view users by the View Users button, upload data by the Upload Data button and manage pages by the Manage Pages button. The same features are provided from the menu bar on the top of the page as well as the ability to view data from the View Data dropdown menu.

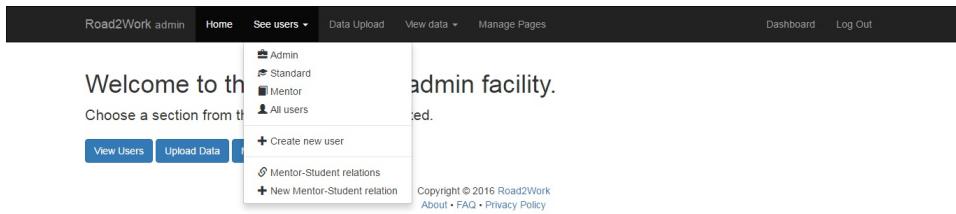


Figure 6.17: Admin - Users Dropdown Menu

The See users dropdown menu on the top of the page provides a lot of different features. Admin, Standard, Mentor and All Users upon clicked provide information about the users in the system. Create New User brings the admin to a new page where they can create new users. The Mentor-Student Relations and New Mentor-Student relation are used for viewing and creating student mentor relations respectively.

The screenshot shows a form titled 'Test Account' for editing a user. The fields are: Name (Test), Surname (Account), Email (empty), Group (Admin), and Password (empty). Below the form are buttons: 'Cancel', 'Save changes' (highlighted in blue), and 'Delete User'. At the bottom right, there are links for 'View Users', 'Upload Data', 'Dashboard', and 'Log Out'. A copyright notice at the bottom right reads 'Copyright © 2016 Road2Work'.

Figure 6.18: Admin - View User

This page provides the admin with the ability to edit user data and delete the user altogether.

Create a new user

Name	<input type="text"/>
Surname	<input type="text"/>
Email	<input type="text"/>
Group	<input type="text" value="Standard"/> <input type="text" value="Standard"/> <input style="background-color: #0070C0; color: white; font-weight: bold; border: none; width: 100%; height: 1.2em; margin-bottom: 0.5em;" type="text" value="Mentor"/> <input type="text" value="Admin"/>
Password	<input type="password"/>
	<input type="button" value="Cancel"/> <input type="button" value="Register user"/>

Copyright © 2016 Road2Work
[About](#) • [FAQ](#) • [Privacy Policy](#)

Figure 6.19: Admin - Create New User

As stated above in Figure 6.17 upon clicking Create New User the admin is taken here, where they can register a new user. The group dropdown menu provides different user groups.

Data Upload

Institutions and Courses	Placement and Employment Data
<input type="button" value="Choose file"/> No file chosen	<input type="button" value="Choose file"/> No file chosen
<input type="button" value="Upload File"/> <input type="button" value="Clear"/>	<input type="button" value="Upload File"/> <input type="button" value="Clear"/>

Copyright © 2016 Road2Work
[About](#) • [FAQ](#) • [Privacy Policy](#)

Figure 6.20: Admin - Data Upload

When the admin chooses the Upload Data option from the welcome page they are taken here, where they have a choice to upload Institutions and Courses or Placement and Employment Data. The admin is required to upload a JSON file and submit it with the Upload File button.

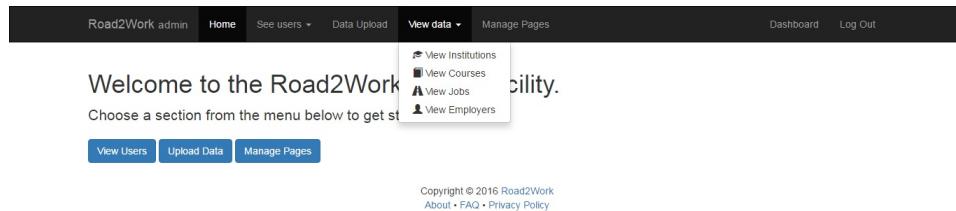


Figure 6.21: Admin - View Data Dropdown

Selecting the View Data dropdown menu allows the admin the options to View Institutions, View Courses, View Jobs, View Employers.

Road2Work admin	Home	See users	Data Upload	View data	Manage Pages	Dashboard	Log Out
#	Name			Student Union URL	Institution URL		
1	ABDO College of Education			Student Union	Institution		
2	University Of Aberdeen			Student Union	Institution		
3	University Of Abertay Dundee			Student Union	Institution		
4	Aberystwyth University			Student Union	Institution		
5	Accrington And Rossendale College			Student Union	Institution		
6	Activate Learning			Student Union	Institution		
7	AECC Chiropractic College			Student Union	Institution		
8	Alton College			Student Union	Institution		
9	Amersham & Wycombe College			Student Union	Institution		
10	Anglia Ruskin University			Student Union	Institution		
11	The Arts University Bournemouth			Student Union	Institution		
12	University Of The Arts, London			Student Union	Institution		
13	Ashton Sixth Form College			Student Union	Institution		

Figure 6.22: Admin - View Institution

This is the View Institutions page which provides options to visit the student union of the Institution and the Institution website.

Product Installation

This guide details the installation of the Road2work platform on a new web server. It assumes that your server meets the following requirements:

- PHP $\geq 5.5.9$
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension

- MySQL, SQL Server or PostgreSQL
- Webserver such as Apache, Nginx or IIS

- Node v3.0 or newer

- Global installation of PHP Composer

Once your server meets the aforementioned requirements follow the instructions below.

1. Copy the contents of the application CD into a folder within the scope of your web server application. Map a VirtualHosts or similar site entry's document root to the public folder.
2. Copy the .env.example file from the application root to a file named .env. Next open the newly copied file with a text editor and set the parameters required as follows:

- APP_ENV: Set to local if testing on a local server, otherwise set to production.
- APP_DEBUG: Set to true if running locally otherwise set to false.
- APP_KEY: Leave at default, this is set in a later step.
- DB_HOST: Set to the database host, in most cases localhost if the web server also acts as database server.
- DB_DATABASE: Corresponds to the name of the database to be used by the application.
- DB_USERNAME: Set this to the username of the database user for the application.
- DB_PASSWORD: Set this to the password of the database user for the application.
- CACHE_DRIVER, SESSION_DRIVER and QUEUE_DRIVER: Leave these parameters as default unless you have a special configuration that requires otherwise.
- MAIL_DRIVER: Set this to the mail driver for sending application email, in most cases SMTP.
- MAIL_HOST: Hostname of the mail service to be used.
- MAIL_PORT: Port to use for send mail protocol.
- MAIL_USERNAME: Username for outgoing mail server.
- MAIL_PASSWORD: Password for outgoing mail server.
- MAIL_ENCRYPTION: Encryption for transmissions to outgoing mail server.

Once complete save the file and close your editor.

3. cd into the root directory of the application. Next run npm install and wait for the process to complete.
4. Once the process is complete run gulp and wait for the process to complete. The CSS and JavaScript assets will now have been compiled.
5. Run php artisan key:generate. This sets the application key in the .env which is used to encrypt user sessions and other data.
6. Run composer install and wait for the process to finish.
7. Run php artisan migrate and wait for the process to finish. This creates the necessary database schema for the application.

Finally visit the application URL as defined in your hosts entry. The website should show the homepage, from where you may sign up for an account to begin using the website.

Manually Creating Admin Users

User groups can be changed within the User Admin section of the website. In order to gain access to the Admin section a user needs to be manually promoted.

In order to do this you'll need to be able to edit the database. Once authenticated run UPDATE Users SET group = 'admin' WHERE email = 'YOUR_EMAIL' replacing YOUR_EMAIL with the email address of the account to promote.

Appendices

A. Usability Questionnaire

Usability Questionnaire

Before

1. How old are you?
2. What is your occupation?
3. In a week, how much time do you spend on a computer?

Less than 1 hour 1-5 hours 5-10 hours 10-20 hours over 20 hours

Time-measured tasks

(Inv. use only) Ask them to register as a new user.

(Inv. use only) Ask them to navigate to the pathway.

Login

1. The registration process was intuitive and easy to navigate through

Strongly Agree

1

2

3

4

5

Strongly Disagree

2. It was clear what parts of the page were for new users and which parts are for existing users

Strongly Agree

1

2

3

4

5

Strongly Disagree

Dashboard

1. The dashboard screen was clearly laid out

Strongly Agree

1

2

3

4

5

Strongly Disagree

2. The process of entering current qualifications was intuitive

Strongly Agree

1

2

3

4

5

Strongly Disagree

What did you like/dislike about the website layout?

Is there any additional comments you would like to make?

B. Project Commit Log

November 18th, 2015

Work done by: Daniel Waghorn

Commits: 492fa8d37331ba155f6916e7024277c9c6641de6

We initially started to build the solution, we began with an installation of the Laravel framework on the Homestead virtual machines which were controlled via the Vagrant container software that we planned to use for rolling out the development of the project.

January 5th, 2016

Work done by: Daniel Waghorn

Commits: 920afa73606cca1d3d1e334a53b073db677fe09c

This is when the beginning of the website began to take form, we started with the homepage, the login and sign up interfaces.

January 19th, 2016

Work done by: Stoyan Nikolov, Rory Malcolm

Commits: 6303f6114f6dc3be80af7d8abd0431e25d073a1a,
c9a8a29a694838880dbbe49c57caf1b4e842bac0

This is when we began to look at using the PHPExcel library to deliver information from the provided Excel files as our job, course and employment details, this was phased out a later date after we realised that there were avenues of doing this that were better and faster methods of doing this.

We also began to add the profiles section of the website, where the users information would be displayed to them, allowing them to edit it if they needed to.

January 22nd, 2016

Work done by: Stuart Marples, Rory Malcolm, Daniel Waghorn

Commits: c9a8a29a694838880dbbe49c57caf1b4e842bac0,
cdb7903b4079c7e588dcb6bc7645ccafee303e8c, 8ea0e5dbde5f5388632fc2159bd25ee633558d73

We implemented an authentication process which we then rolled out to the pages that needed to ensure a user was logged in before they were accessed, we also created the view from which everything in the app would be viewed through.

January 23rd, 2016

Work done by: Stuart Marples, Rory Malcolm, Vitalija Zilinskaite, Marian Fantini, Daniel Waghorn

Commits: e162a6d92cab43d12546ffb66f98623f7e3e27db,
ba8dd8601a3606769a2e21306670705032e98600, 81d0ca02be04ea518d2cd3c970b9af6416b99fec

This was when we first added the ability for users to fit into a number of groups and for different groups of users to have different access privileges with regards to which areas of the site they could access.

The admin interface then began to take shape, from which an administrator could control the database via interfaces that we designed for them.

January 24th, 2016

Work done by: Daniel Waghorn, Roman Mason

Commits: 343680f549e0c9d0c1b7adcdad86958d2b516545

We finished the company website, with all the marketing information that we needed to sell the product.

January 25th, 2016

Work done by: Rory Malcolm, Daniel Waghorn, Marian Fantini, Vitalija Zilinskaite

Commits: 83d287436a520b7cf63020d11b73163ca219c11f,
2d07651fb04076d4b5662a6ad57f8b4e6f35275b

Added gravatars, more work on profile section and admin interfaces.

January 26th, 2016

Work done by: Stoyan Nikolov, Rory Malcolm, Vitalija Zilinskaite, Mariana Fantini, Rory Malcolm, Stuart Marples

Commits: 68d570a6cecbba3592d746cf7ed9200affdde5064,
a072974207f99d3916fa8aa2b6b9ea35c3cbd57b, 3f1b172627894d5d2a209762ee362156b51ffe68

More development of admin and profile section, authentication systems changes and updating of user interface look & feel.

February 11th, 2016

Work done by: Daniel Waghorn, Rory Malcolm, Mariana Fantini

Commits: a11cb15b4236459d4c961e7a47197bea2bfb8aca,
7219a80f68dad166200c1daf1f9d0c4c161d998f, b13ab79eda777158884b202bab4248c0f8851ce

We added a first time log in initial wizard for the user to set up their profile, elements of the admin interface to add new users to the system and set up tooling for delivering our javascript in a compiled format from lots of different sources and modules via Babel.

February 12th, 2016

Work done by: Rory Malcolm

Commits: 9857e1c680812f210adcdef312326759a6a78503

Started the migration to Vue.js for the initial profile setup.

February 17th, 2016

Word done by: Stoyan Nikolov

Commits: 8c714a40be18cdcdc9cfbee49ffb91dab3043aef

We began working on the the data upload process for the internal databases.

February 18th, 2016

Work done by: Marian Fantini, Daniel Waghorn, Stoyan Nikolov

Commits: 044176fb6a836575a03180a6771a16f0eb441095,
43ddb85f7289eb90a8ae660b7851b7c375cb6b0c, 43ddb85f7289eb90a8ae660b7851b7c375cb6b0c

More changes to admin interface, the data upload section and javascript delivery.

March 1st, 2016

Work done by: Rory Malcolm

Commits: 15dfcda25c3795b4980b6f1eb7a07ab54360b342

Made the API a RESTful version, with correct parameters for updating the user model.

March 5th, 2016

Work done by: Rory Malcolm

Commits: 57828ce86ebee97b8cfdb989d2fe66991ed4f4c

Finished the user setup component, receiving and sending data via the API, it is ready for it to be mapped in vue router.

March 8th, 2016

Work done by: Stoyan Nikolov

Commits: 82acae81929a3bff810992fb3fc75d30f7950891

We worked more on the data upload section of the website.

March 15th, 2016

Work done by: Rory Malcolm, Daniel Waghorn, Marian Fantini, Vitalija Zilinskaite, Stoyan Nikolov, Stuart Marples

Commits: 128614e6e7bd6a1b267638dfcf99a0b43df8765a

Beginnings of JSON Streaming parser, database changes, changes to the user interface.

March 16th, 2016

Work done by: Rory Malcolm, Daniel Waghorn, Marian Fantini, Vitalija Zilinskaite, Stoyan Nikolov, Stuart Marples

Commits: 375d5a625cdac72d5958f40383c7d8e936d65d2d

Finished JSON Streaming parser, got information loaded into the database in large amounts.

March 17th, 2016

Work done by: Rory Malcolm, Daniel Waghorn, Marian Fantini, Vitalija Zilinskaite, Stoyan Nikolov, Stuart Marples

Commits: fa6098ae5c54b370b059a0034d46c56c0c378f65

Added pages CMS, added mentorship interface.

March 20th, 2016

Work done by: Rory Malcolm, Daniel Waghorn, Marian Fantini, Vitalija Zilinskaite, Stoyan Nikolov, Stuart Marples

Commits: 0a4bccff065ce19dc5b9e6ba55324fd5e90e38e6

Updates to all areas of website, course selection api and interface complete for users, added service to get information about institutions and the courses they offer.

March 21st, 2016

Work done by: Rory Malcolm, Daniel Waghorn, Marian Fantini, Vitalija Zilinskaite, Stoyan Nikolov, Stuart Marples

Commits: bd22875c7d44c7020be6500062a97736d27f8b51

Added suggestion of courses to the user, added email verification and forgot password services.

March 22nd, 2016

Work done by: Rory Malcolm, Daniel Waghorn, Marian Fantini, Vitalija Zilinskaite, Stoyan Nikolov, Stuart Marples

Commits: dc4c29fa78f097d2dc0158e13c185981fa51919d

Added timeline visualization for the users pathway to their chosen job, taking in account of the qualifications they have and the qualifications they need to receive in order to get there, basically complete and ready for demo.

C. Project Diary

29/03/16 (Group Meeting)

Stage 3 demo.

28/03/16 (Group Meeting)

We all worked together to develop various technical system features, and made sure everyone is confident and ready for the presentation tomorrow.

27/03/16 (Group Meeting)

We all worked together to develop various technical system features.

26/03/2016 (Group Meeting)

We all worked together to develop various technical system features. We have also started to rehearse for the demo we will have on Tuesday the 29th, and the report is taking shape.

14/03/16 (Manager Meeting)

The manager meeting went well, and we are now going to focus on the main features of our system (creating pathways etc.).

This week we will also start on documentation so we would have enough time to show the drafts to Fiona and receive feedback.

12/03/16 and 13/03/16 (Group Meeting and Manager Meeting)

Our application is now using Vue.js framework. Therefore, most of the logic is now directly manipulating data, which is useful since editing/updating forms is going to be done quite often (user details etc.). The dashboard is now running as a dynamic single-page application, meaning that it's composed of relatively small, self-contained, and sometimes reusable components.

This will enable each of us to work independently for some time now, which is incredibly useful since now every group member can plan and work on their own pace and whenever they choose to when deadlines for other courses are approaching.

25/02/2016 (Group Meeting)

Some changes to the admin side of a new user have been made, and the work on data upload has been started.

18/02/2016 (Group Meeting)

The page for a first time user account setup has been added to the website and the administrative side.

11/02/2016 (Group Meeting)

We all met up to discuss the demo, and fixed some minor mistakes in the report.

27/01/16 (Manager Meeting)

Stage 2 demo

25/01/16 (Group Meeting)

26/01/16 (Group Meeting)

During these two meetings we all worked together on different parts of system's functionality to get it ready for the demo on the 27th (Wednesday).

- First of all, user registration/login forms have been fully completed, with possible 'Standard' or 'Mentor' roles once registered. Middleware was implemented to check user permissions for that.
- Basic dashboard view with user details, space for pathways, and pathway details has been created.
- Basic profile editing and settings editing has been implemented.
- Back-end administrative tool to manage user profiles has been implemented.

23/01/16 (Group Meeting)

Some of us have met to continue working on the project, Roman and Dan completed the promotional website for our product, which can be found at <http://meet.road2.work/>

14/01/16 (Manager Meeting)

On the first meeting with the manager after the winter break, we briefly discussed where we are at the moment, and what our next steps will be. Fiona's feedback was good so far, since we've already started the system development, and everyone knows their roles.

12/01/16 (Group Meeting)

On our first meeting after the winter break, we discussed some details of what has been done, and what still needs to be done. The website development has been started.

23/10/15 (Group Meeting)

We have now finished all that was needed for usability tests.

22/10/15 (Group Meeting)

We were all working on different bits of usability tests - website layout mock-ups, questionnaires, required forms. Moreover, we finished off the needed UML diagrams, and Risk Analysis document.

22/10/15 (Manager Meeting)

We made sure that everyone knows what their role is, as well as discussed the usability testing that will take place next week. We have also been reminded to send initial write-ups to Fiona before, e.g. using questionnaires for usability testing.

14/10/15 (Group Meeting)

We went over a few of the remaining bits of documentation that need finished off and assigned tasks to get this done which are on Wunderlist.

The main steps now are to complete a few UML diagrams for the system to supplement the functional and non-functional requirements that we have already done.

We also had a technical discussion based on skills and previous experience in order to decide which technologies and frameworks to use in order to complete the project. We have decided to go with Laravel.

08/10/15 (Manager Meeting)

On our second meeting with Fiona, we went over a set of initial requirements we all thought of. We also discussed the next steps and goals for the following week - mostly work towards getting the initial requirements document completed.

01/10/15 (Customer Meeting)

I got some things from the customer meeting too. Most of it Vitalija already put here, the ones that I didn't see in the other post are those:

1. If we consider the accessibility question, we have to pay attention so it doesn't interfere with the responsive design
2. It would be good to show plans for future versions of the system
3. For practical reasons, we could focus on some courses for the first version of the system
4. The primer user is the student; mentors and teachers can give recommendations
5. Costs: reference to average salary
6. Show the job location in the system (could offer a maps and pictures to the user)
7. The system could show the students the qualifications they should look for to follow a certain career

01/10/15 (Customer Meeting)

Main problems:

1. What future career support can be given to people with a certain set of skills?
2. Given that there are many possible opportunities, how to make the system easy to use?

Main points:

1. Establishing where the person is at the moment (age, position, qualifications etc.);
2. Showing pathways available;
3. Enabling the user to explore possible career prospects / opportunities.

Questions/Answers:

1. *Should the application include accessibility options for disabled or visually impaired users? (If so, which disabilities would you like covered (sight/hearing impediments/dyslexia)?)*
We don't necessarily have to implement it, but we should think about it. Responsive design and adaptation to different devices is more important.
2. *Does the design of the system (UI etc) need to follow the current Lockheed Martin (LM) theme if so do you have any style guides we can use?*
Reusing content is a good idea, but design solutions are mostly up to us.
3. *Employers are listed as secondary users. Is the intention to allow employers to view user's career paths as a kind of visualised CV? If this is the case will we be allowing employers to search for people with the right skills using this system? Should we think of it almost like a social network with people setting their profiles to private if they only want to share them with certain people, or setting them to public to allow potential employers and educators to search for them? Should employers be able to see the end users' grade?*
The system is primarily aimed at the person who wants to make a career, but it is also a good idea to implement views for potential employers. There could be a feature to let schools know that e.g. a set of people are interested in a certain field. Another thing to think about is enabling users to connect with people with similar interests or skills. Here we should also think about privacy settings.
4. *Can secondary users be segregated (i.e. parents and teachers) to track the Educational progress especially for younger users (high school), and advisors/employers/Universities etc. when they are looking to progress or gain employment?*
Progress tracking for parents, teachers etc. should only be a secondary purpose, if we decide to implement it. Could be mentioned as a possible update for later versions of the system.
5. *Would you like filter earnings, jobs, locations for future pathways as a setting?*
That would be good.
6. *How much is our system expected to support the user after they have chosen the future path they want?*
Our system would be a good tool to keep track of lifelong learning and future achievements as well.
7. *Is this aimed primarily at UK applicants or do we include international visitors? Should we take into account international students who may be residing in Scotland/UK?*
For practical reasons, the system is aimed at Scottish users only.
8. *If the user does not have e.g. good enough grades to receive any possible pathways, should the system offer any alternatives?*
Yes, visualisation for that is very important.
9. *Can the website application host ads (for revenue)?*
Nothing would stop us from doing that, but we have to be careful with ads (so that it would not appear that the pathways are biased).

10. *Should the users be able to log in using their social media accounts? What about sharing their achievements etc.?*
Yes, logging in using social media accounts is a basic requirement. However, sharing is not compulsory.
11. *Does there need to be any 'back-end' / admin functionality from the product to allow management or ops staff to manage various aspects of the software? If so then what sorts of permissions are they're allowed?*
There needs to be a system admin support (super admin). Other roles: user (student), mentor (teacher, parent). They should all have different views and permissions of the system.
12. *Should there be any verification that the input data is correct?*
No, we are going to assume that everything the user enters is correct (except for basic things, e.g. age range)
13. *Would the user be able to contact the relevant organisation that they are interested in regarding the course/job that they offer?*
The system should only point the user to the right direction (school's website), and suggest applying there.
14. *What are the key sets of information described? How many different subjects/industries should we cover? (should it be limited to STEM)*
It is a formal requirement that the system covers all major industries / subjects. However, implementing a huge variety of subjects for every major industry would be too difficult.
15. *The specification says that the website is to "...be able to be examined frequently and often over period of time." How does Lockheed Martin foresee the user interacting with the website over an extended length of time?*
The user should be notified if any changes related to their chosen pathway (e.g. the course is not available anymore) have been made.
16. *Can we use email users for attentions and alerts?*
Yes.
17. *Can we use emails for account verification/activation?*
Yes.
18. *In terms of legislation, and protection of data and users, what is the earliest age users can sign up to the application?*
There are no age requirements, but in reality we would have to check child safety guidelines. At this point we will assume that the website is safe for everyone to use, as long as basic encryption (passwords) is being used.
19. *Would you like a Content Management System included?*
CMS is good for reusing bits of software, but we also have to think about how to combine internal and external data efficiently.
20. *Should we cover legacy pieces of software? Such as internet explorer, previous versions of chrome, etc... What versions of web browser do we base it on?*
The system should support the latest versions of the most popular browsers. It should also display nicely on different devices (tablets, phones, etc.).

Most importantly, keep it simple, always think of user experience, and explain every decision made.

6. Bibliography

- [1] Appsee: Mobile analytics platform. [https://www.appsee.com/.](https://www.appsee.com/)
- [2] Artisan console. [https://laravel.com/docs/5.1/artisan.](https://laravel.com/docs/5.1/artisan)
- [3] Browserify. [http://browserify.org/.](http://browserify.org/)
- [4] Composer: Dependency manager for php. [https://getcomposer.org/.](https://getcomposer.org/)
- [5] Digitalocean: Simple cloud computing, built for developers. [https://www.digitalocean.com.](https://www.digitalocean.com)
- [6] Director: a tiny and isomorphic url router for javascript. [https://github.com/flatiron/director.](https://github.com/flatiron/director)
- [7] Docker: An open platform for distributed applications for developers and sysadmins. [https://www.docker.com.](https://www.docker.com)
- [8] Gulp.js: the streaming build system. [http://gulpjs.com/.](http://gulpjs.com/)
- [9] How people build software github. [https://github.com/.](https://github.com/)
- [10] jquery: write less, do more. [https://jquery.com.](https://jquery.com)
- [11] Laravel elixir. [https://laravel.com/docs/5.0/elixir.](https://laravel.com/docs/5.0/elixir)
- [12] Laravel forge: Painless php servers. [https://forge.laravel.com/.](https://forge.laravel.com/)
- [13] Laravel homestead. [https://laravel.com/docs/5.2/homestead.](https://laravel.com/docs/5.2/homestead)
- [14] Laravel: The php framework for web artisans. [https://laravel.com/docs/4.2/introduction.](https://laravel.com/docs/4.2/introduction)
- [15] Nielsen norman group: Evidence-based user experience research, training, and consulting. [https://www.nngroup.com.](https://www.nngroup.com)
- [16] Node.js. [https://nodejs.org/en/.](https://nodejs.org/en/)
- [17] npm: Node package manager. [https://www.npmjs.com/.](https://www.npmjs.com/)
- [18] Packagist: The php package repository. [https://packagist.org/.](https://packagist.org/)
- [19] Postman api client. [https://www.getpostman.com/.](https://www.getpostman.com/)
- [20] Prospects.ac.uk. [https://www.prospects.ac.uk/.](https://www.prospects.ac.uk/)
- [21] salsify/jsonstreamingparser : A json streaming parser implementation in php. [https://github.com/salsify/jsonstreamingparser.](https://github.com/salsify/jsonstreamingparser)
- [22] Sass: Syntactically awesome stylesheets. [http://sass-lang.com/.](http://sass-lang.com/)
- [23] Travis ci: Test and deploy with confidence. [https://travis-ci.org/.](https://travis-ci.org/)
- [24] Unistats: The official website for comparing uk higher education course data. [https://unistats.direct.gov.uk/.](https://unistats.direct.gov.uk/)
- [25] Vagrant. [https://www.vagrantup.com/.](https://www.vagrantup.com/)
- [26] Vue js: Reactive components for modern web interfaces. [http://vuejs.org.](http://vuejs.org)
- [27] Vue resource: Resource component for vue.js. [https://github.com/vuejs/vue-resource.](https://github.com/vuejs/vue-resource)
- [28] Vue router: The official router for vue.js 0.12+. [https://github.com/vuejs/vue-router.](https://github.com/vuejs/vue-router)
- [29] Vueify: single file vue components. [https://github.com/vuejs/vueify.](https://github.com/vuejs/vueify)
- [30] Xdebug - debugger and profiler tool for php. [https://xdebug.org/.](https://xdebug.org/)