MAJOR PROJECT REPORT

Interactive Multimedia Design – 2015

Abstract

"The challenge was to create a web application that could create a random trip for users. This report will detail the various stages throughout the creation, development, testing and evaluation of Matooi."

Alan McCaig – B00587565

Peer Support Group 9 – Simon Fraser

Acknowledgements

"I would like to personally thank everyone involved with IMD over the past 4 years, especially Peter Nicholl, George Moore and my mentor Simon Fraser, for making this project possible. I would also like to thank the Web team at AVBrowne for helping to develop my skills through my placement year and introducing me to Laravel." – Alan McCaig

Contents

1 – Introduction

This report will explain in detail how the concept of creating a random trip generator was developed into a fully functional web application called Matooi.

1.1 – The Challenge

The initial challenge was to create a project that would combine and show all of the skills which have been developed throughout the 4 years of studying Interactive Multimedia Design (IMD). In order to successfully complete the challenge it was decided that a web application would be created. After generating several ideas it was determined that the web application would be based on the concept of creating random trips for users.

1.2 – Work Undertaken

In order to create Matooi allot of work had to be done, initially the idea for the creation of the project had to be imagined and then realised. The idea of the project was looked into to check that it was a feasible task and see if there could be any constraints which could affect the outcome of it.

The next step involved designing the project such as designing the flow of the data and logic and also developing a good user experience (UX). After this a functional prototype of the web app was created this gave an opportunity to see if the rest of the project would be manageable.

The system and UX designs where then refined based on the findings from the functional prototype. At this stage the project was developed and tested, this report is the final step which will contain the evaluation and conclusion to the project.

1.3 – Aims of Report

The aim of this report is to detail the steps throughout the process that was taken during the creation of Matooi, a random trip generating web app. As a waterfall methodology was applied to the creation of Matooi the report will take a similar format, starting with requirements analysis and continuing through the design, implementation, testing and evaluation of the project.

2 – Concept Definition and testing

This section will describe the initial concepts that were considered and how the chosen concept was then tested to ensure that it was feasible.

2.1 – Idea Generation

After brainstorming several potential ideas it was decided that 3 of them would be taken forward and researched further in order to see which one would be the most feasible.

2.1.1 – Concept 1 – Steam Achievement Tracker

The first idea was to develop an online steam achievement tracker. Steam is one of the world's largest online gaming platforms and it offers access about each users stats via its own API. The achievement tacker would allow users to view a data visualization about their profile and also allow them to track achievements for each game.

2.1.2 – Concept 2 – Streamer Profiles

The second idea was to create a site that would allow streamers from Twitch.tv to create profiles. These profiles would allow them to combine their content from sites such as Twitch, YouTube and other sites that they may post content to. This would allow viewers of these streams to be able to access all of their favourite streamers content on one site.

2.1.3 – Concept 3 – Random Trip Generator (Chosen)

The third idea originated from 'random road trips', these would be when a group of people would drive in a random direction and see what unique and interesting places they could find. There are various web and native mobile applications that allow users to plan a road trip however none of these offer a way to randomize the trip. During research it was even found that a way to randomize a trip was a requested feature for a popular road trip planner, RoadTrippers (Roadtrippers Community, 2015).

2.2 – Requirements Specification

At this point it was decided that concept 3 (random trip generator) would be the most credible concept to take forward and so the following requirements specification was created.

2.2.1 – Aims and Objectives

Aims

The aim of the project was to create a fully functional web application that feels like a native mobile application however will work responsively with higher resolution screens as well. The web app will be able to select random locations based on the user's preferences and combine these to create a trip. The web app would then display this to the user in the form of a map.

Objectives

The primary objectives that where created to help achieve the aim are as follows:

- Design and develop a user friendly user interface (UI) that feels like a native application when used on mobile devices.
- Develop user accounts for the site that will allow users to save and view previously saved trips.
- Develop a search feature that will choose random locations nearby a user based on their preferences.
- Develop a function to link the locations together and display this to the user in the form of a map.

The following secondary objectives where created to show how the app could be expanded after its initial release:

- Develop the application into a native application for Android and IOS platforms.
- Monetize the web application in various potential ways (such as subscription fees and on-site adverts).
- Expand available languages.

2.2.2 – Goals

Goals where then created in order to help achieve the aims and objectives of the web app, these goals were as follows:

- A working online web app that contains the following features:
    - A user account system – Users should be able to sign, edit their account settings and save trips.

    - A working random location selector – The web app should be able to find nearby locations based on the user's preferences.
    - The web app should be fully functional on both mobile devices and desktops.
- The web app will initially be usable within the UK therefore using English as the main language and using UK metrics (such as miles).

2.2.3 – Stakeholders

The stakeholders of the app were taken into account, however as the project is being made for University it was decided that the main stakeholders would be the Lecturers, the student and the users. There is a wide audience of which the app will cater to, from those who are wanting to travel across the country to those who are just wanting to go for a small drive.

2.2.4 – Tasks

A set of tasks where created in order to help define what the project would involve, after creating these tasks the constraints and potential risks were also considered to help determine if the project was feasible. The tasks involved were as follows:

- Design
    - Branding – A unique and iconic brand should be developed for the web app.
    - User Interface Design – A user interface (UI) should be designed for the app, this will involve creating wire frames, paper prototypes and various mock-ups.
    - Database Design – As the app will have a user account system and have the ability to save trips it will require a database, this database will require an ERD to determine the relationship values.
- Front End Development
    - UI Development – The UI will need to be coded, this could use a framework such as Foundation in order to help with the process.
- Backend Development
    - Database Creation – The database will need to be created in line with the previously designed rules.
    - Locations Search Function – The search function will need to be developed, this will require using a form to get the users preferences about the locations.

- Map Creation Function – The locations will need to be used to create a map with directions for the user.
- User Accounts – User accounts will need to be created along with features such as account creation and account editing.
- Save Trips Function – The trips will need to be saved into the database and have appropriate features available, such as being able to view previously saved trips and delete them.

- Testing
  - Functionality testing – All the features of the site will need to be tested, involving both backend and frontend testing.
  - Usability testing – Tests should be performed to see if there are any issues with the usability of the web app.

2.2.5 – Constraints

Various constraints where looked at as issues that could potential arise throughout the development of the web app. There were 3 types of constraints that where looked at these were solution constraints, schedule constraints and budget constraints.

Solution Constraints

There were many constraints that where directly related to the solution (the web app) that where discussed, these are as follows:

- Browser compatibility – as the web app was made for use on the internet it needed to take into consideration certain issues that browsers have. Taking into account desktop and mobile browsers, the web app was restrained to using appropriate technologies (such as HTML5, CSS3, etc) in order for the solution to work.
- Use of APIs – As the web app is about getting potential trip locations for users it was decided that an API would be used to retrieve this information, this however added a constraint to the web app as the information and access to the APIs information would be restricted based on the APIs restrictions (such as limited calls to the API or the data that the API stores).
- Making the web application feel like a native mobile app – as the web app was planned to be tailored to mobiles this meant that it would be constrained with various things

that would be required to make it usable on small screen devices. These constraints where things such as resolution, connectivity speed (such as keeping the app quick as people may use 3g to access it), etc.

Schedule Constraints

As the project was being created for University the schedule was limited to the deadlines set. Another issue that would affect the schedule would be the knowledge of the student as this could affect the timings of certain development stages and could ultimately affect the outcome of the project. To help minimize the chances of the schedule being affect a methodology was applied to the project (discussed in section 2.5) and from this a schedule was made.

Budget Constraints

As the project was for University there was no initial funding available for it, however certain avenues where researched, such as using crowd funding (such as Kickstarter) or even potential bursaries. It was decided however that the project would be feasible with minimal cost by using open source code where possible and using images and other media that was under the creative commons license.

2.2.6 – Risks

Risks where looked into to see what could potentially affect the development of the application and to see if anything could be done to avoid/minimize the damage if the risk had occurred. The following table (table 2A) lists all of the risks along with the chance of it happening and the level of the risk. Each risk also has a contingency stated of things that can be done to avoid or help minimize the damage.

Table 2A

	Risk	*Chance of occurrence*	*Level*	*Contingency*
Personal Skills	Lack of knowledge in using API's.	High	High	The APIs are an integral part for the application as this is where the majority of the app's data will come from, the only option for not using APIs is to manually create a

				database with the info that the APIs would of brought, however this will take up allot of time and also limit the app's global reach and accuracy immensely.
	Lack of knowledge of using Laravel.	High	Moderate	In a worst case scenario the app could be coded using basic PHP and HTML or another framework could be looked into, however this will increase development time.
	Lack of knowledge of using HTML Geolocation	Moderate	Low	HTML5 Geolocation will be required to get the users current location, HTML5 is not supported on all browsers however there are various libraries available that could be trailed. In a worst case scenario the application could get the users coordinates by asking them to select where they would like the road trip to start.
	Lack of knowledge using Foundation framework.	Low	Low	The use of foundation is to decrease front end development time (specifically the responsive design of the app) as a backup Bootstrap could be used or if necessary the responsive grid could be made from scratch, however this will increase development time.
Application Features	The randomized search feature is the main part of the application, without it	Moderate	High	In a worst case scenario the whole application will need to be restructured in order to either change or remove the randomized search.

	functioning the rest of the app is pointless.			
	The ability to save road trips that are generated.	Low	Moderate	The application does not need this function in order to work and so in a worst case scenario it could be removed from the app. However cookies could be a solution in order to allow the user to see what road trips they had created previously.
	Login and User accounts	Low	Moderate	Again this feature is not needed in order for the application to function however it does improve the overall usability of the app, in a worst case scenario the app could possibly use cookies as a way to show previous results to the users.
External Resources	APIs (Google Places and Maps APIs) may lose/change functionality overtime.	Low	High	Through the development any potential changes to the use of the APIs will need to be documented and acted upon, also the app should display appropriate messages if it fails to retrieve data from these sources (for example if they are down the app should display that it is having trouble receiving information and to try again later)
	API Call limits	Low	Moderate	Google APIs have a 100,000 request per day limit, this limit should be kept in mind when creating the app and try to use functions to limit the need for calling the API as much as possible.

	The use of BitBucket to store the applications source code, if Bitbucket where to go down.	Low	Low	Local copies of the source code should be regularly backed up as to eliminate the possibility of completely losing the code of the whole application.

2.2.7 - Functional and Non-Functional Requirements

At this point a set of functional and non-functional requirements where created. The requirements where created as snow cards, all of which can be found in Appendix A, and this section will give a broad overview of them.

As the objective of the web app was to create random trips using locations based off of the user's preferences it was decided that a user account system would be developed, this would also allow for users to save trips that were generated for viewing again later.

Below is a broad overview of the functional (1-13) and non-functional (14 and 15) requirements, the numbers in brackets refer to the full snow cards which can be found in Appendix A.

- Users Accounts (1-5)
    - Account creation - a user should be able to create an account. (1)
    - Account login – a user with a valid account should be able to securely login. (2)
    - Account logout – a user with a valid account that is logged in should be able to securely log out. (3)
    - Editing account settings – a user should be able to edit their account settings, such as their email, password or other details. (4)
    - An account is required to access the trip application – a user must be logged in on a valid account to have access to the trip generator. (5)
- Random trip generation feature (6-9)
    - Locations types – a user will be required to fill out a form specifying the types of locations they would like to visit and the amount of locations. (6)
    - Locations radius – a user will specify the radius that they would like to search for the locations. (7)

    - Locations order – the app will organize the locations in distance so that the app generates an optimized route for the trip. (8)
    - Connect locations on map – the app should generate a trip using the locations and display them on a map. (9)
- Randomized trips features (10-12)
    - Save generated trips – a user should be able to save a generated trip. (10)
    - Export trips – a user should be able to export a generated trip to Google Maps (for use of Google Maps navigation and offline usage). (11)
    - Delete saved trips – a user should be able to delete previously saved trips. (12)
- Aesthetics and Performance (13-15)
    - The app should work responsively – the web app should work responsively on small resolution screens and up. (13)
    - The app should be usable in the UK – the app will primarily be made for use in the UK, using English and the UK metric system (miles). (14)
    - The app should work on mobile connections (3g) – the app should be usable on slow connections and mobile networks. (15)

2.3 - Paper Prototyping

The next step in the process involved creating an initial paper prototype for how the application would work this helped to get an understanding of how the application could be create in order meet the requirements.

2.3.1 - App Structure

As it was decided that the web app would be a trip generator and contain user accounts, several different ways that the app could be structured where looked into, it was decided that the structure of the app could be as follows.

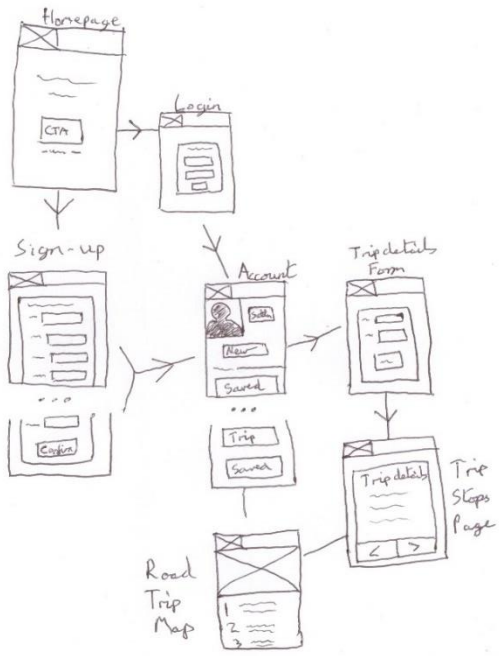


Figure 2A – The original prototype for the structure of the application

As you can see from figure 2A a user would first view the homepage, this was planned to contain information about the app and sign up/login buttons. From the homepage the user would be able to access the trip generator by logging in or creating an account, either way this would direct them to the account page.

The account page would contain all of the users saved trips along with a buttons linking the user to create a new trip and edit their account's details.

The new trip page would contain a form that requires the user to specify the amount of locations that they would like to visit and the radius in which to search for the locations. The app would then find suitable locations and list them to the user.

The final page would display a map and directions between each location. This page would also contain 2 other buttons, these would be used for saving the trip and exporting the trip to Google Maps.

2.3.2 - Visual Comps

6-ups

After deciding on a structure that would match the requirements of the app 6-ups where created for each page, these 6-ups consisted of 6 different possible layouts that each page could have. For example figure 2B shows the 6-up's that where created for the homepage, as you can see the main change with each version was how to place the menu. As the web app was aiming to feel like a native application on mobile all 6-ups where designed for mobile.

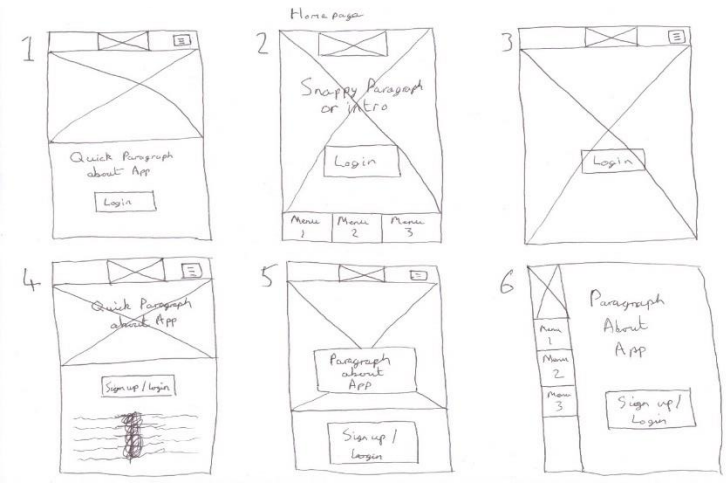


Figure 2B – 6-ups for the homepage of the app

After deciding on a consistent layout for all of the webpages a low fidelity mock-up was created for each, these are detailed below.

Homepage

Figure 2C shows the homepage along with 2 versions of the menu, logged in and logged out. The basic idea was to have the menu accessible by a hamburger icon for mobile devices, this would slide out the menu from the right side of the screen when pressed. The logged in version of the menu would give the user access to creating a new trip, account page, about page and sign out whilst the logged out version would only display the option to sign in or view the about page. A bar at the top of each page would host the logo for the web app and the hamburger menu icon.

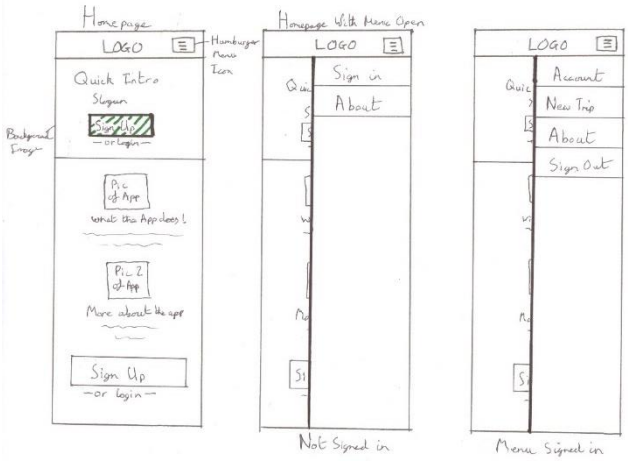


Figure 2C – The homepage and mobile menu (logged in and out)

The home page layout was created to have a short introduction section which would contain a background image. The introduction section would give a brief overview of the app and have a sign up call to action button (CTA). The rest of the page would give a more detailed look at the web app with a final sign up CTA placed at the bottom

Sign Up, Login and Account page

Figure 2D shows the sign up, login and account page, both the sign up and login pages where planned to be forms therefore to keep the forms consistent throughout it was decided that they would appear in a popup styled box, this would also allow for the ability to add these forms in as an Ajax form later on if possible. All the forms where styled with the labels directly above the input fields and a confirm button at the bottom of the forms.

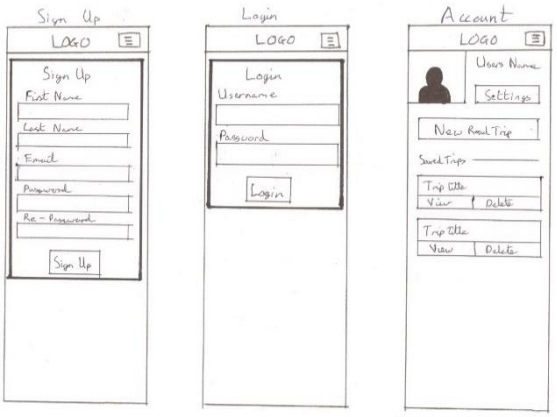


Figure 2D - The sign up, login and account pages

The accounts page was created with the idea that a user could upload a photograph of themselves and have it displayed to them. Beside this would be their name and a settings

button to access the account settings page. Beneath this section there would be a new trip button and finally the page would list the users saved trips giving them the ability to view or delete each trip.

Trip Details Form, Locations Page and Final Trip Page

Figure 2E shows the trip details form, locations page and final trip page. As with the other forms the trip details form was created with a similar layout. The locations page was planned with the idea of having information about each stop listed in a box, each box would contain an image, the name and a short description about the stop. The bottom of the stops page would have a 'next' button which would then create the trip and display it to the user on the final trip page.

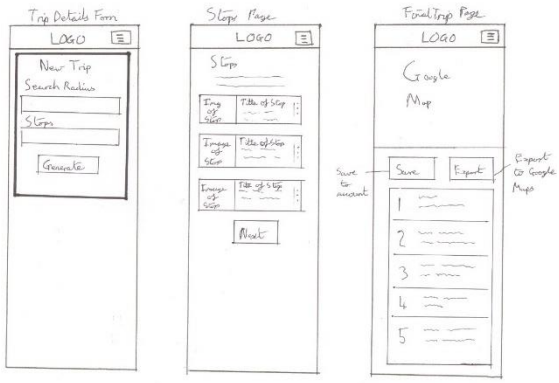


Figure 2E – The trip details, locations and final trip pages

The final trip page contained a Google Map of the trip that was created, beneath this would be a save (save the trip) and export (export the trip to Google Maps application/website) button, finally the final trip page would list the directions for the user.

Higher Resolution Visual Comps

As the forms where planned to be used as potential Ajax forms the idea was to keep them a similar layout when the application was used on higher resolution screens. However the homepage, account page and final trip page could have a different layout to make use of the extra space, the following images are of the low fidelity wireframes of these pages for higher resolution screens.

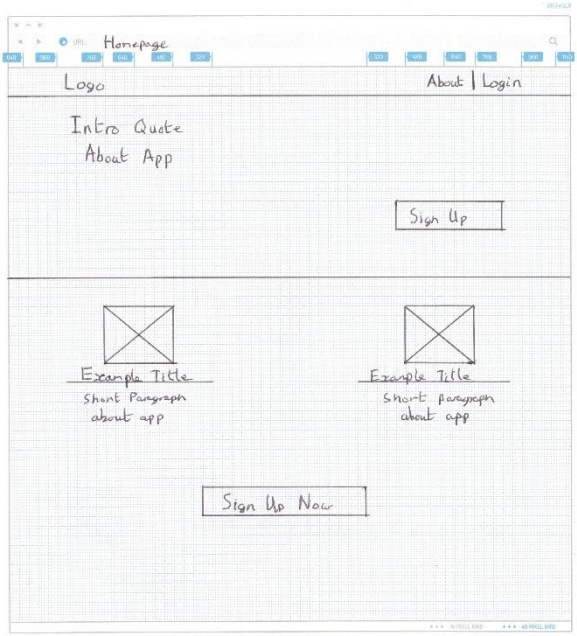


Figure 2F- The desktop version of the homepage

As figure 2F shows the higher resolution version of the homepage would expand out to use more of the space, this would make it smaller in height but give the users more to view over the width of the screen.

As with the mobile version this page would contain an intro section giving the user a quick view of what the app could do along with a sign up CTA. The rest of the page would fully explain what the web app can do and contain another sign up CTA at the bottom of the page.

Figure 2G shows the accounts page, this page again would utilize the extra resolution by filling out the width of the screen, it would also add a background image to the top where the users name and settings button are, this is to separate the page slightly and help the new trip button stand out more.

The saved trips would take up the rest of the page again with an option to view or delete each trip.

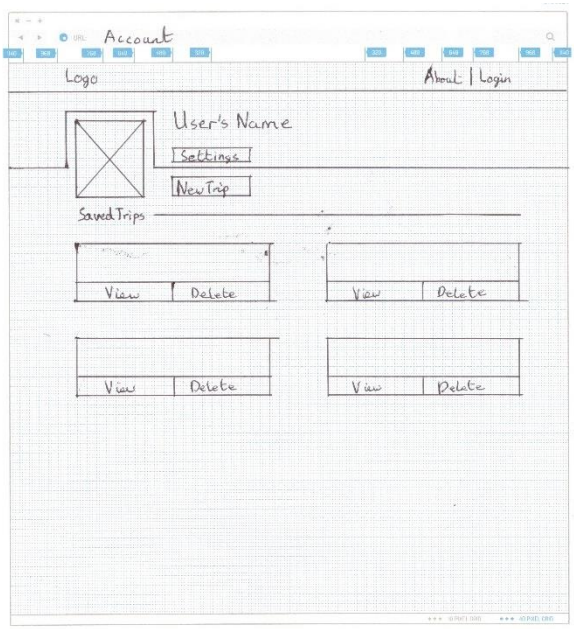


Figure 2G – The desktop version of the account page

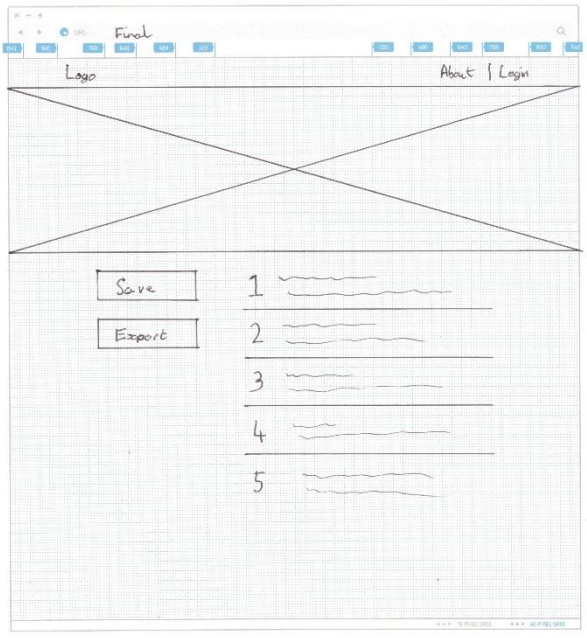


Figure 2H – The desktop version of the final trip page

Figure 2H shows the final trip page, the idea was to have the map of the final trip take up the full width at the top of the screen and then have the directions beneath with a side panel for the save and export buttons.

2.4 – Feasibility Testing

At this point a functional prototype was created, this prototype was aimed at seeing if the project was feasible and therefore the prototype selected a piece of the project that looked to be the most difficult. The part of the project that was selected for the functional prototype was the location search features, this involved getting the users location and finding the nearby locations through the API.

In the end the prototype was successfully created with the ability of getting the users location and then choosing random locations nearby them and finally displaying this to the user. As the prototype was successful this proved that the current aims and objectives would be achievable.

2.5 – Methodology Selection

At this point in the development of the project a methodology was selected and applied to the planning of the project, this was to help keep the project focused and on track and make it as successful as possible.

2.5.1 – Waterfall Methodology

Several types of methodologies that could be applied to the project were researched, such as Rapid Application Development (RAD) and Agile, however the methodology that was chosen was the modified waterfall.

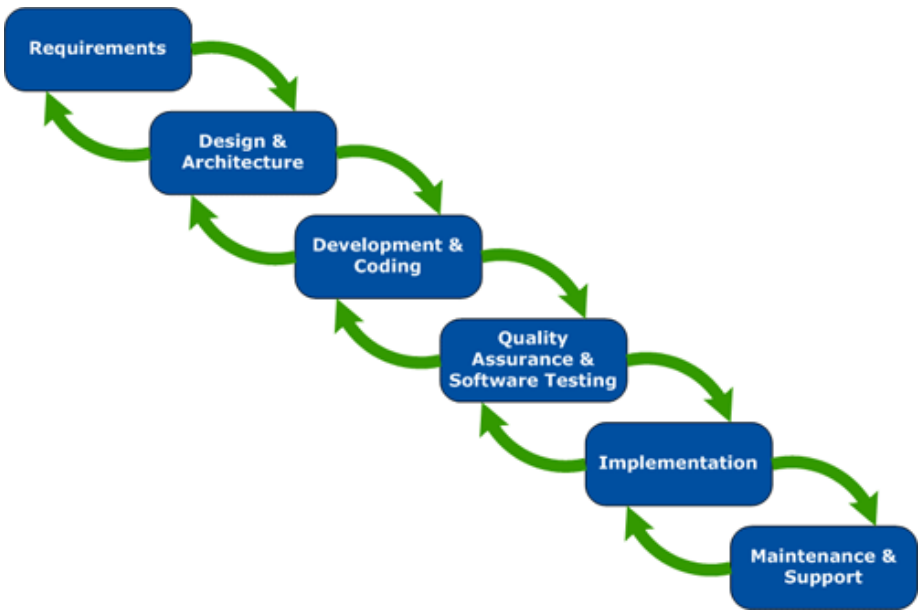


Figure 2I (Ledbrook, 2012) - A diagram showing the modified waterfall method

The modified waterfall methodology, figure 2I, was selected as it matches up with the tasks and deadlines that had been set for the project. This would allow the project to go through a logical process of design then development then implementation and evaluation whilst allowing it to revisit a previous phase if problems were found at a certain point.

2.5.2 – Gantt Chart

The following Gantt chart (figure 2J) was created based upon the projects deadlines and using the modified waterfall methodology. As you can see the chart has been split into 5 different stages (requirements, design, development, testing and evaluation) each with its own subsections. The time allocated at each stage overlaps with the previous and next stage as this allows time to go back a step and change/fix anything within the project at each step.

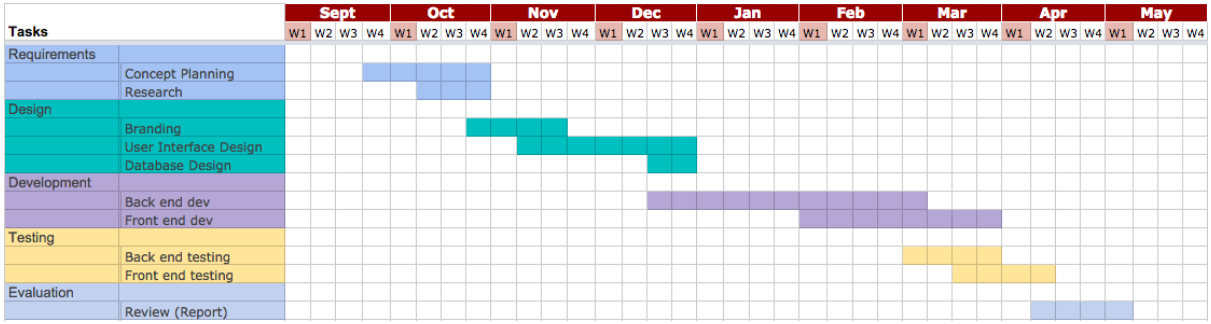


Figure 2J – The schedule created in the form of a Gantt chart based off of the modified waterfall methodology

3 – Design

The design section will give a detailed look at each step of the UX, system, logic and data design stages.

3.1 – UX Design Evolution

This section will give an in-depth look at the evolution of the UX design starting from the creation of the mind maps and wireframes through to the visual comps and final product. There were 3 main stages throughout the UX design evolution, these where the initial designs, refined designs and the final product, the following sections in this chapter will show the designs at each stage and discuss how they changed.

3.1.1 – Mind Map

At the start of the design stage a mind map was created to help get a good understanding of what the design and feel of the app should be aiming for.

After mind mapping the application it was decided that the app should focus on feeling fun and giving the user a sense of adventure and freedom.

Figure 3A – A mind map plotted at the start of the design process

3.1.2 – Mood Board

A mood board (figure 3B) was then created based off of the mind map, it contained images of places, objects and other sites that where related to the mind map. This gave a view of what sorts of colours, textures and imagery could be used to make the web app feel adventurous and fun.

Figure 3B – A mood board created to help get a feel of colours and textures the app could use

3.1.3 – Initial Designs

The initial designs were created after the paper prototypes had been made, these started with the creation of style tiles and visual comps of each page for the web app.

Style Tile – Version 1

The initial versions of the style tile were based on trying to get a good colour and typographic combination. The very first version (figure 3C) used the Trisa Pro and Merriweather typefaces along with a combination of white, red, blue, green and black. The colour combinations where taken from an image of a road stretching into the distance with a clear blue sky.

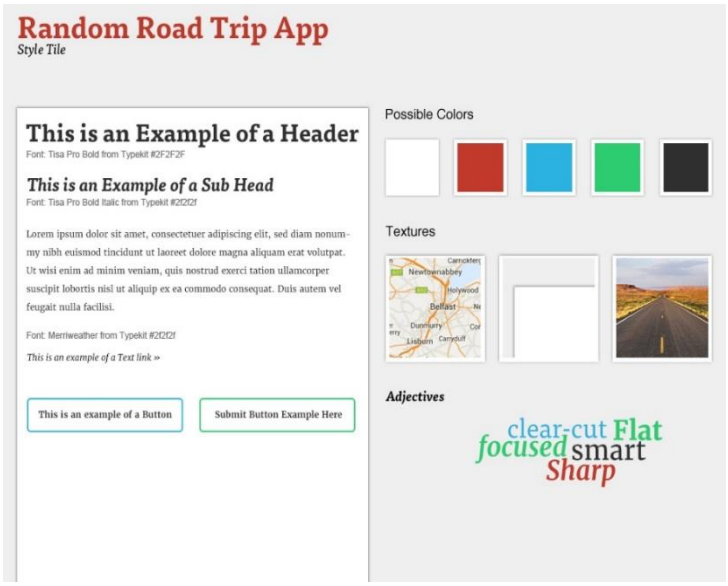


Figure 3C – The initial style tile

The tile also had a slightly grey background which then had the body section with the type and buttons on it lifted off the background by using a slight drop shadow.

Style Tile - Version 2

As the application was about creating a trip, and so would contain maps, the second style tile (figure 3D) focused on styling a map to match the site. For this version the typefaces remained the same, the colours where changed to slightly brighter shades and the drop shadow was increased.

This version was allot more vibrant and blue and adding the styled map also allowed for more consistency.

Figure 3D – The second iteration of the style tile

Visual Compositions Grid

As one of the main aims of the project was to make the web app feel like a native mobile application but also scale up responsively to higher resolution screens as well the initial visual compositions where made using a grid system. The grid system used was called ‘frameless’, it was chosen because it doesn’t have fixed widths however still works with columns meaning the web app could be designed for mobile and fluidly scaled up.

Using the frameless grid each page for the visual compositions where created 4 times, these were mobile wireframes, desktop wireframes, mobile low fidelity and desktop low fidelity. The frameless grid template was created using a 320px 4 column grid and a 1024px 12 column grid.

Homepage

As with the paper prototyping the homepage layout stayed the same, the idea being that the user would be greeted initially with an introduction to the web app along with a CTA to sign up to the site. The following content would then give more detail about the web app and the page would end with another sign up CTA.

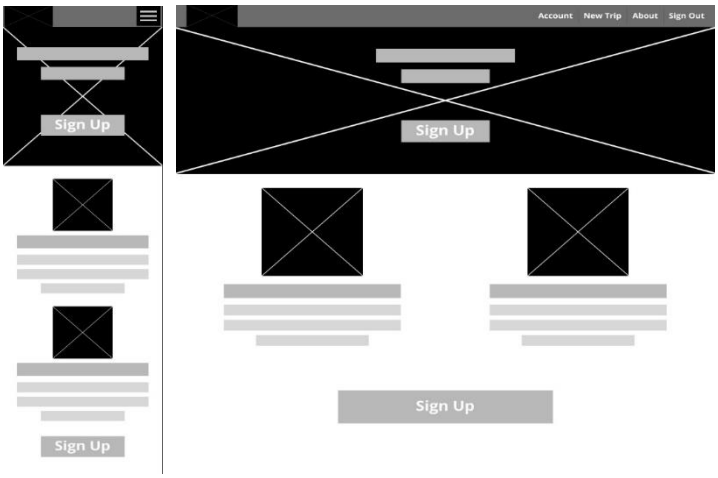


Figure 3E – Wireframe of the homepage

Applying the colours and typography from the style tile allowed the CTAs to stand out. A background image was add to the introduction section with a radial black transparent overlay to make the text and CTA stand out more from the image.

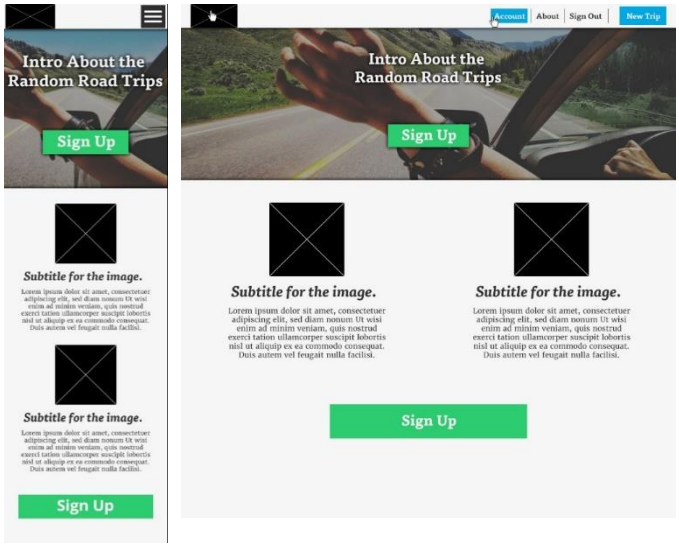


Figure 3F – Low fidelity mockup of the homepage

Navigation Bar

It was decided that for mobile screen sizes the navigation bar would be hidden off screen and accessible from a hamburger menu icon. It was also decided that the navigation bar would change based on whether or not the user was logged into an account.

Figure 3G – Wireframe and low fidelity mock-up of the menu both mobile and desktop, logged in and logged out.

Figure 3G shows the navigation menu slide out from the side on mobiles (pictured left) and the navigation bars with the links changing based on a user being logged in or out (pictured right). A CTA styled button was also added to the navigation menu, this would be used for the 'new trip' button.

Forms

The site needed to contain several forms (such as register, login, new trip details, etc.) therefore to help improve the UX the forms where created to be consistent throughout.

Figure 3H shows the mobile and desktop wireframes and mock-ups for the sign up form. Forms where kept very minimal and straight forward to ensure ease of use, the top of the form containing a title for the form followed by each input. Each input would show the label above it and at the bottom right of the form would be an 'advance' button, in this case 'Sign Up'. The advance button would use the green colour and be positioned to the right of the form container.

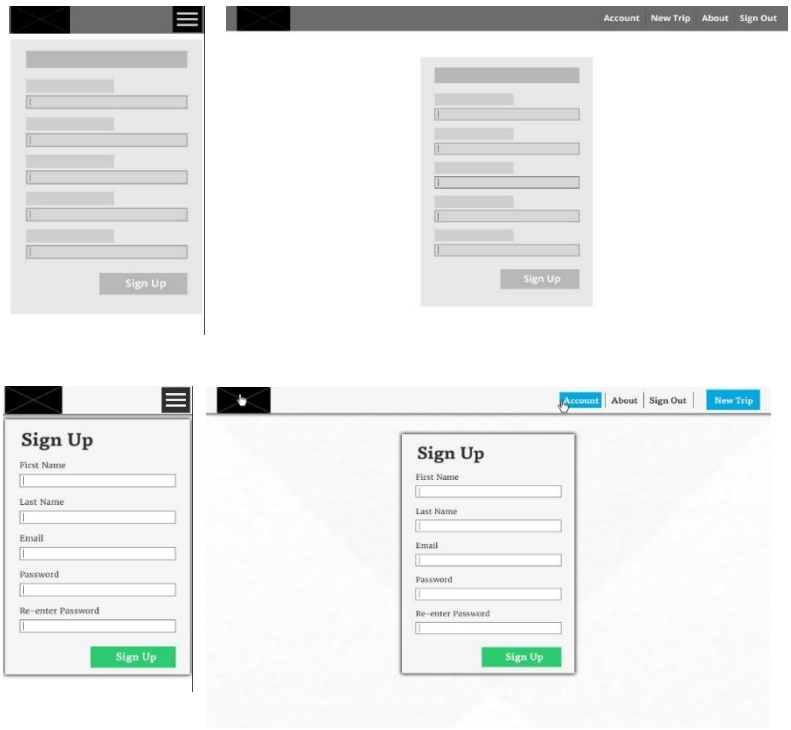


Figure 3H – Wireframe and low fidelity mock-up of the sign up form

Locations Page

The locations page would display, to the user, each of the locations that where randomly selected for their trip. Each location would show an image of the location, the name of the location and a short description about it. In the original design, figure 3I, you can see there were 3 small dots to the right of each location, these where planned to suggest to the user that they would be able to drag and drop them to arrange the order in which they would visit each location. However through development it was found that the locations could be automatically sorted to make the quickest route for the trip and therefore this feature was scrapped as reflected in later designs.

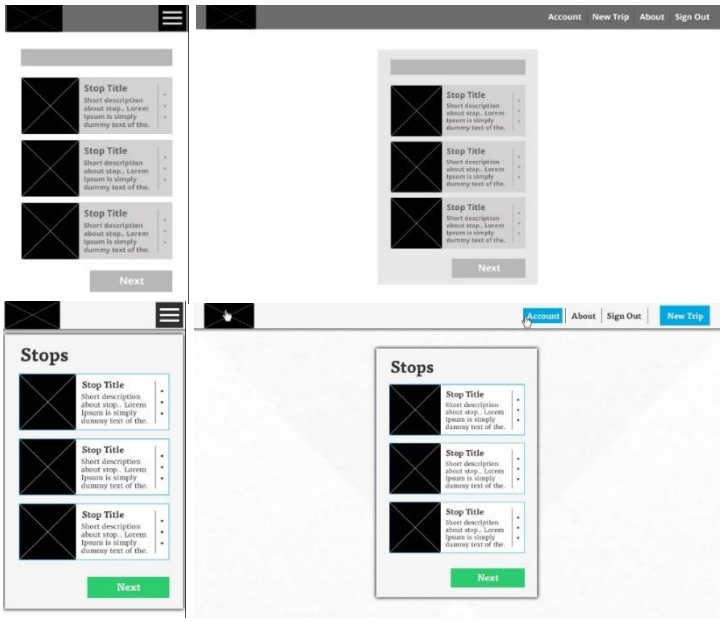


Figure 3I – Wireframe and low fidelity mock-up of the locations page

Account

The accounts page would be the main page for the user, as when they logged in it would be where they were redirected to. The profile page needed to contain links to the rest of the site, such as creating a new trip, access to editing their profile information and a list of all the saved trips.

Figure 3J shows the initial designs of the account page. At the top of the page on both mobile and desktop size the users name and profile picture will be shown. Beside the profile picture will be a button to link the user to the edit account settings page and beneath this will be a button to link the user to create a new trip. The final section of the page will display all of the users saved trips, each one containing a title, view and delete button.

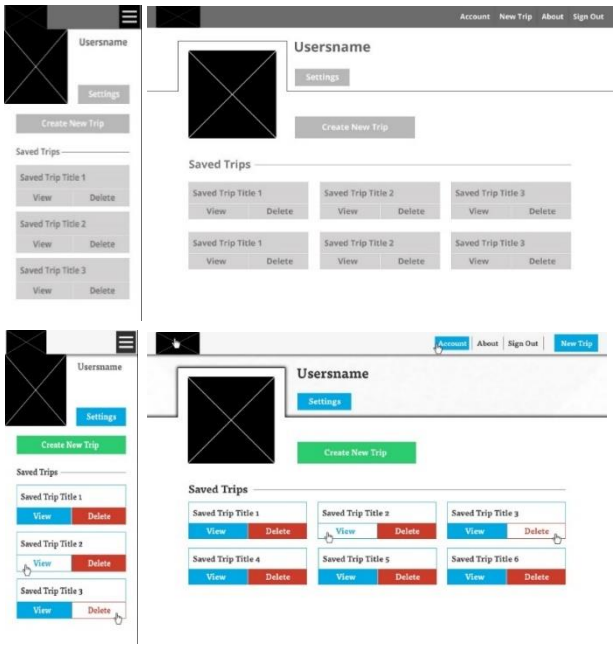


Figure 3J Wireframe and low fidelity mock-up of the account page

Final Trip Page

The final trip page would need to show the user a map with the selected locations plotted upon it and also display the directions to the user. As shown in figure 3K the map will take up the full width at the top of the page. Beneath this changes based on the screen resolution, small screens will show the save and export buttons with the directions positioned directly beneath that, whereas higher resolution screens will show the directions beneath with a side panel that contains the two buttons.

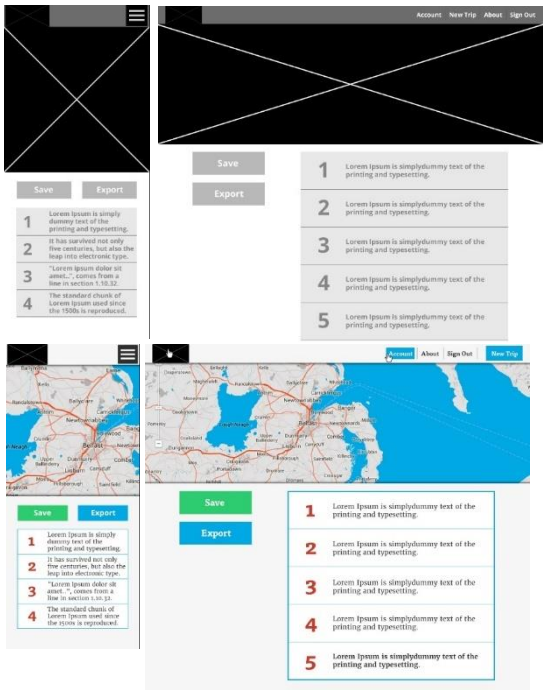


Figure 3K Wireframe and low fidelity mock-up of the final trips page

3.1.4 – Branding

At this stage in the design process a brand was developed for the web app, the branding started with brainstorming a name for the web app.

Naming

The name of the web app needed to be something catchy, short and easy to remember. Initially the name 'Trip It' had been used for the project however this was found to be already used by a travel help app therefore it was decided a new name should be made.

There were several names that were created such as 'Lets Go!', 'Where to go?' and 'Make a Trip'. However using the phrase 'Make A Trip Out Of It' and abbreviating it to 'Matooi' caught on, Matooi was unique, short and quirky and also contain the meaning of the app which you can use to make a trip.

Typographic treatment

Figure 3L – Initial typographic treatment of the brand

It was decided that the brand would consist of a typographic treatment of the name and a unique icon. To start off the typographic treatment several typefaces where looked into which would also keep with the aim of the branding. After looking through several possible typefaces 'Hallo Sans Black' was selected and the dot for the 'I' was replaced with a location icon.

Iconography

The original idea for the icon was to have a radar with blips on it that would spell out the letter ‘M’ or display a location icon or question mark. This idea didn’t work well however but lead the way to the final logo, the question mark and location pin where both taken and combined as you can see in figure 3M. The icon also replaced the location icon for the dot of the ‘I’.

Figure 3M – Iconography for the brand

Colour

Whilst designing the logo it was created in various shades of blue this was to try and incorporate the feeling of going on a trip when the sun was shining and the sky was blue. This two toned approach really stuck out, using a darker shade for the type and part of the icon and the lighter shade to highlight the question mark and so after slightly adjusting the shade of blues the final logo was created.

Final Brand

The final brand of Matooi is a unique, fresh and fun looking brand. The logo and type can be used on their own or used in combination and the icon works at all sizes. The brand also works in white, black and greyscale and its two tone colours can be easily changed if the brand ever evolved. Figure 3N shows the final brand along with black, white and various colour combinations and it also shows how it would look and feel in real world environments.

Figure 3N – Final branding in a range of colours

3.1.5 – Refined Designs

The refined designs incorporated the logo into the site and tried to flesh

everything out, such as images and actual text instead of placeholders. This section will show the refined designs for Matooi and describe any changes and why they were made.

Style Tile – Version 3

This version changed the white background to a light grey to help match the map. At this stage it was decided that the icons could be inpcorporated into the designs as well as this would allow for a better overall UX.

Figure 3O – Third iteration of the style tile

Homepage

The main aim for this iteration of the designs was to add more subtle details throughout. As shown in figure 3P the logo was added to the navigation bar and more imagery was added to the homepage. Drop shadows where added to the text and CTAs that where placed over sections that contain background images and the placeholder text had been replaced with appropriate text of which explained what Matooi can do and why you should sign up.

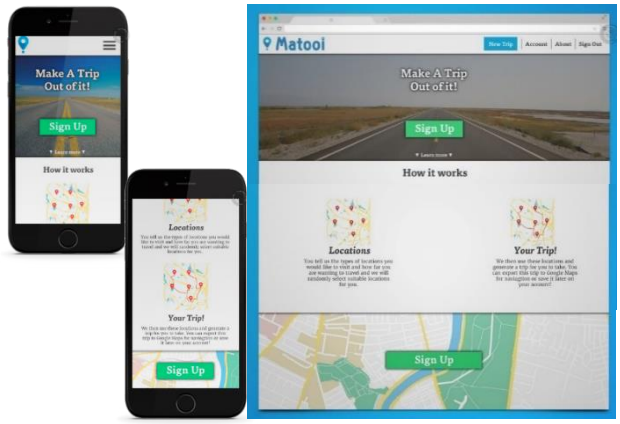


Figure 3P – high fidelity mock-up of the homepage

Forms

The forms stayed the mostly same for this iteration, using the same layout and colours, however the section that contained the form had its background colour changed to match the grey for the navigation bar. The box also had a drop shadow added to it to help it stick out from the background. Figure 3Q shows the login form along with how the register form would look with validation errors.

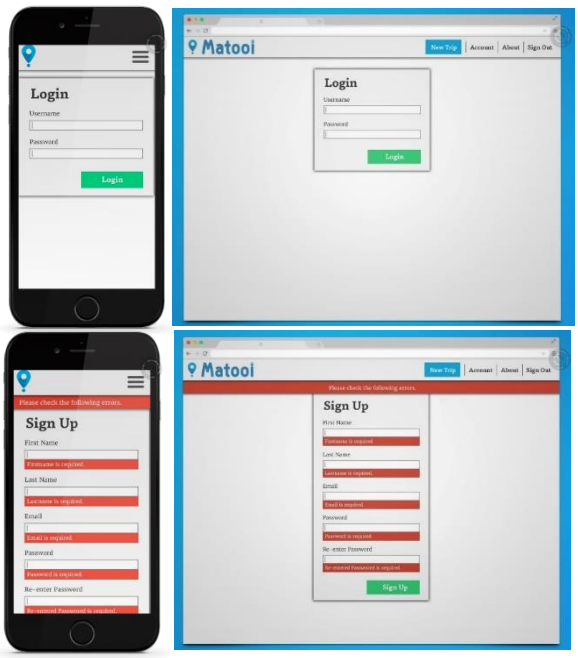


Figure 3Q - high fidelity mock-up of the login and form errors

Account Page

The user accounts page was tweaked slightly to use up more of the available space at higher resolutions. The settings button was changed to a small cog and positioned at the bottom right of the user profile picture and a background heading image was used to help fill up some whitespace in the desktop version. Each saved trip also had its type resized to help fit and differentiate the title from the buttons.

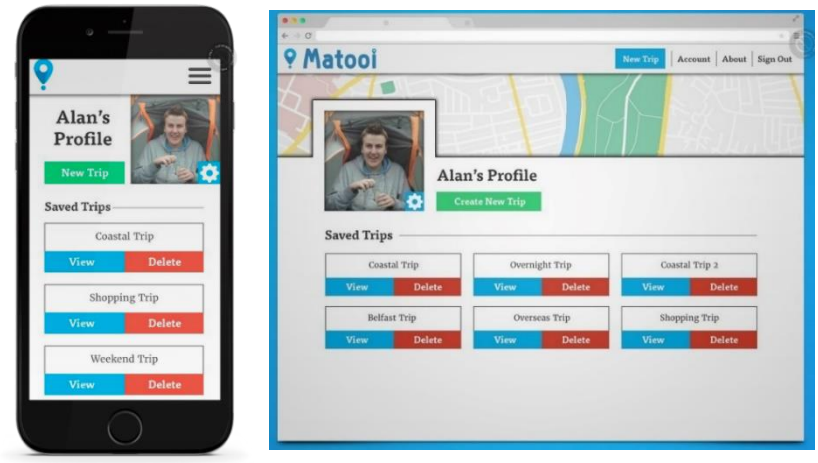


Figure 3R - high fidelity mock-up of the accounts page

Locations Page

Figure 3S shows the refined iteration for the locations page, this page was given a big overhaul, the main change being the details for each location. Each location would be displayed with the name, a picture and short description about the location, below this would be a 'display more' button. The display more button would extend the selected locations container and show the user more information about that location.

Figure 3S - high fidelity mock-up of the locations page

Directions Page

The direction page was adjusted to help with the UX by adding some icons to the directions also the map colours were changed to give the whole site a colourful look.

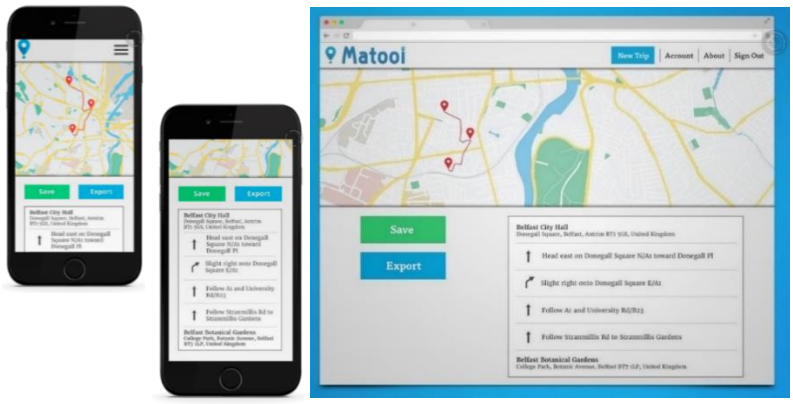


Figure 3T - high fidelity mock-up of the final trip page

3.1.6 – Final Designs

The feedback received for the refined design created 4 main issues that needed to be resolved, these were:

1. The typography throughout the site did not match the branding.
2. The colours where very technical looking and overall the site needed more vibrancy.
3. The drop shadow overuse should not be required on the type and CTAs over a background image if the background image was tailored to make them stand out.
4. The site needed more imagery, especially some feel good imagery for the homepage.

To address these issues major changes where applied to the whole site, these changes will be detailed in this section along with screen captures of the final product.

Homepage

To fix some of the previously mentioned issues several changes occurred to the whole site. The typefaces chosen where changed to match the smooth serif branding font, this changed the original Meriwether and Tisa Pro typefaces to Museo-sans and Nimbus-sans. Another noticeable change is the removal of all drop shadows, instead the site now has gone for a much flatter and clean look. The final big change was the colour palette, the background colour for sections throughout have been changed from grey to white and the site uses more shades of blue (to match the brand) with the darkest shade being used for the background of the site.

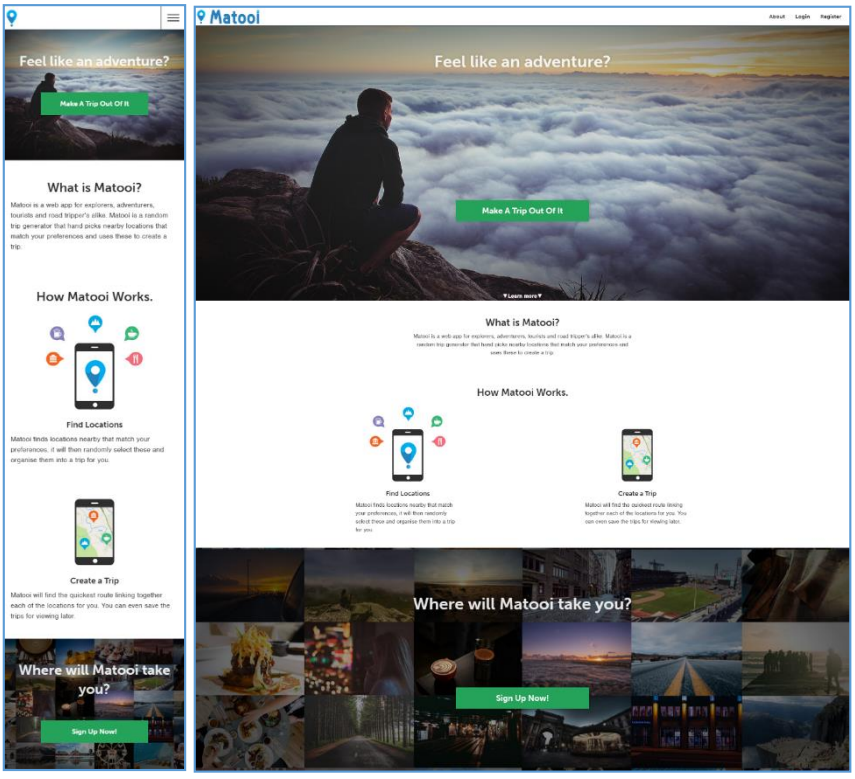


Figure 3U – Final design of the homepage

As you can see in figure 3U more imagery was added to the homepage along with an extra paragraph containing a brief description of what Matooi is.

Forms

The forms layout stayed the same for the final designs, however the colour changes where applied and the gaps at the sides and top of the form containers were removed to tighten up the mobile version.

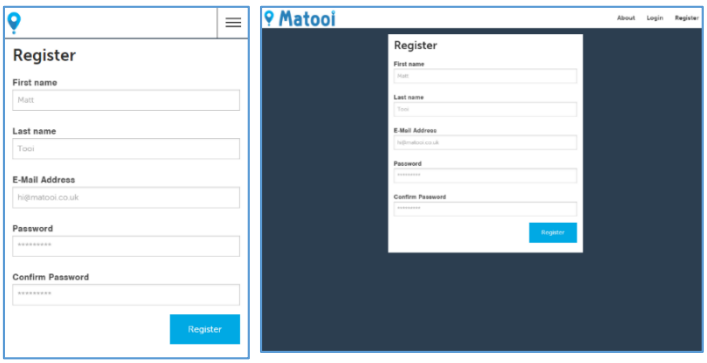


Figure 3V – Final design of the login form (consistent with all forms)

The new trip form also had a map added to it and an adjustable radius circle around the user's location, this would change when the user changed the radius to make. Also as at this stage in the project it was decided that a user would be able to specify which types of locations they would like therefore to help with usability icons where added to the type checkboxes.

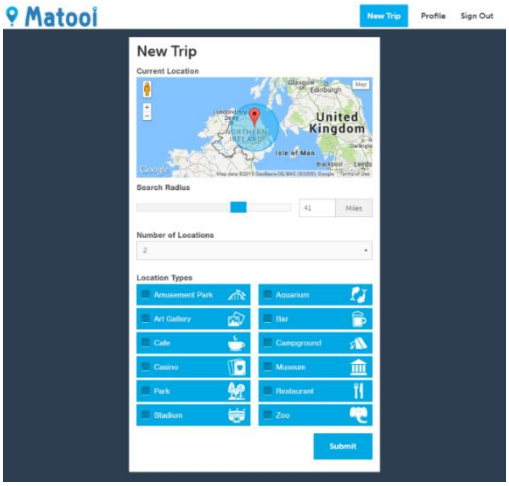


Figure 3W – Final design of the new trip page

Account

The accounts page was shuffled around to give the site a more consistent feeling, mainly using the white background placed upon the dark blue for the majority of the site. The desktop version placed the profile picture and buttons to the left side of the page whilst the mobile version kept the stacked version. It was also decided that when the screen size was really limited, such as older lower resolution smart phones, that the profile picture would be removed as it would help keep the account page focused on the functionality of the application opposed to it taking up vital limited space.

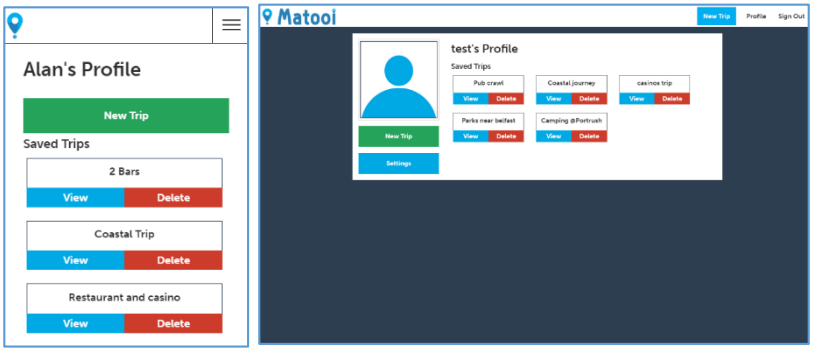


Figure 3X – Final design of the account page

Final Trip Page

The final trip page also received an overhaul for the final version, it was decided that a much better user experience was created by adding an infinitely scrolling panel to the left of the screen for the directions and letting the map fill the rest of the screen. The save and export buttons where placed above the directions panel.

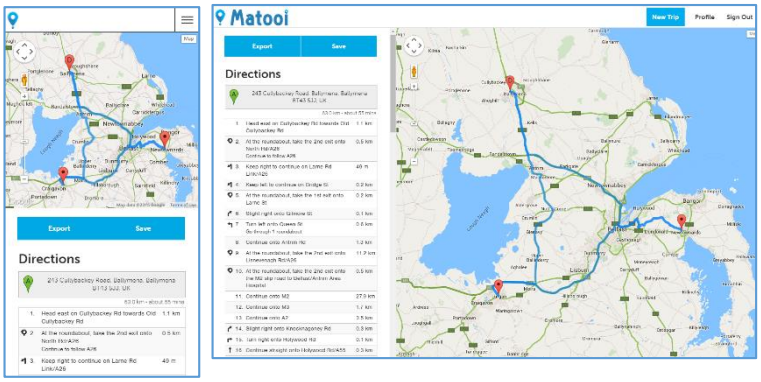


Figure 3Y – Final design of the final trip page

3.2 – System Design

The system design section will show, in detail, what tools and technologies where used in order to create Matooi.

3.2.1 – Technologies

At this stage of the process several technologies where researched to see which would be best to use in order to help the project successfully achieve its aims and objectives. This involved researching coding languages, scripting, APIs and external services to see which technologies would be useful for the project.

Languages

As the project was planned to be a web application there were several languages that could be used below is a list of the selected ones along with their reasoning:

- HTML5 - Hypertext mark-up language (HTML) version 5 has many features that will help with the development of the app. It will be used to create the basic mark-up of the web pages that will be contained within the app and also has some built in features (new in HTML5) that will be useful for the web app, for example geolocation. (W3schools.com, 2015)
- CSS3 - Cascading style sheet (CSS) version 3 will be used for styling the front end of the web app. CSS is used to style the elements on webpages such as each elements position, colours, size, etc. New with CSS3 is CSS transitions and animations which can be utilized to add app like animations to the web app. (W3schools.com, 2015)

- PHP - PHP is a server side language, meaning that the users will only see the results of the PHP and not the underlying code. PHP is mainly used to send, receive and display data from HTML elements into server storage. PHP will be required as the web app will have a user account system and also will need to be able to store trips, therefore this can be achieved by using PHP and MySQL to pass data to and from a database. (Php.net, 2015)

Scripting

Scripting languages where looked at to help make the web application feel more dynamic, responsive and to help with the overall user experience, below is the selected scripting languages along with their reasoning:

- JavaScript - JavaScript is an object oriented computer programming language that is primarily used on the web. It is used to make webpages dynamic and interactive. It will be useful for the web app as the web app will need to have data passed around it (such as the locations and final trips) and also interact with the APIs. JavaScript was also used in conjunction with Geolocation.js library in the functional prototype to help with the accuracy of the HTML5 geolocation, this proved useful and therefore solidified JavaScript as a useful technology for the development of Matooi. (jquery.org, 2015)
- JQuery - JQuery is a library of JavaScript, it makes HTML manipulation, event handling, animation and Ajax much simpler. JQuery will be useful for the app as it will be able to be used in conjunction with CSS3 to create interactive components and help achieve the native app feel. It also allows for adding Ajax into the web app could be useful in certain parts of the app to allow the user to load data into the page they are currently on without having to reload the whole page. (jquery.org, 2015)

Database Management

As Matooi needed a database for storing user and trip information a database management system needed to be used, for the case of Matooi it was decided that MySQL would be the best technology for this. MySQL is an open source database management system. It can be used in conjunction with PHP to manage passing data to and from a database. As previously

mentioned this will be useful for the web app as it will have user accounts and also need to store information about the trips. (Mysql.com, 2015)

Frameworks

To help speed up development and allow for more complex features and functionality several frameworks where looked into, both front end and backend, below is the list of frameworks that were selected for Matooi.

- Foundation - Foundation is a front end framework that comes with a responsive grid layout, this will be used as a base for the CSS throughout the site, mainly the responsive grid, and the rest of the site will then be customized to match the current design of the site. The benefits to using foundation is mainly to speed up development time with the frontend. Foundation was used in the functional prototype and worked flawlessly therefore foundation will be used as the CSS framework for the project. (ZURB, 2015)
- Laravel - Laravel is a PHP framework that has grown rapidly over the past couple of years, it is a MVC (Model Views and Controllers) framework and so will allow for separation of logic. Laravel has many new and innovative features that can be used to aid in the development of the web app, for example it uses a blade templating system that allows you to create chucks of repeatable HTML and then insert them into a base template. Laravel is also free and again does not require any adverts in order for you to use it. Laravel also uses a dependency manager (Composer) which allows for easy updating of the whole framework and its dependencies, there is also allot of open source and usable dependencies that can be called in if the project requires which will aid in the development of the project. (Otwell, 2015)

APIs

In order to get information on various potential locations APIs where looked at, specifically location and mapping APIs (such as Foursquare, Google Maps, etc.). Originally Foursquare was going to be used due to its vast information available for locations however due to the limitations of the API it was decided that the following APIs would be used instead.

Google Places is an API that can be used to find locations and details about the locations. The following Place requests are available:

- Place Searches
- Place Details
- Place Actions
- Place Photos
- Place Autocomplete
- Query Autocomplete

These features will allow for detailed information to be used within Matooi and will help make the app usable with the same reach as the Google Places API, which is global. The API is limited to 1000 requests per 24 hours, which could be quite limiting, however this can be extended to above 100,000 requests per 24 hours if needed. This API was successfully used in the functional prototype and was easy to retrieve the required data from therefore will be used in the web app to get information about nearby locations for the users. (Google Developers, 2015)

The second API to be used is Google Maps JavaScript API v3 this API allows Matooi to create maps with ease. The API also offers various features that can help with the user experience of Matooi, such as creating radius circles on the map, showing the users location and displaying the route for a user to take for their trip. The API allows for 1,000,000 calls per day and this can be extended if needed. (Google Developers, 2015)

External Services

During the development it was found that an external mailing service was required, this was to allow the functionality of sending emails to users (such as password resets). After looking at various mailing services Mandrill was chosen. Mandrill is a mailing service created by Mailchimp, it allows for fast delivery of email starting with no cost and scaling based on usage. (Mandrill, 2015)

3.2.2 – Logic Design

The next section will discuss the initial design of the system logic, this will show how the previously listed technologies will be implemented to form a system that will be able to achieve the requirements.

Client Server Model

As Matooi is a web application the client model server best suit the characteristics of the relationships between the technologies that will be used on the client, internet and server. Figure 3Z shows how each of the selected technologies will fit into the client server model.

Figure 3Z – Client Server Model and how each technology will fit in relation to Matooi

As you can see from Figure 3Z all of the frontend and visual elements that the client will see are shown in the client section. The internet section contains the APIs and external services that Matooi will use. The server section contains all of the backend technologies that the user will not be able to see but will be used to move and manipulate information around in order for the application to work.

3.2.3 – Data Design

Design Patterns

There are various repeatable elements, both frontend and backend, that were taken into account during this stage of the project, these patterns where pieces of repeatable sections for the site (mainly code) and so could be created in a reusable way which could help speed up the development and functionality of Matooi.

For the frontend there were various features on the app that would stay the same or be similar throughout, such as the menu bar. Laravel was chosen as it has allot of useful features, one of which helped out with repeatable frontend code, which Laravel refers to as 'blade templating. Laravel's blade templating engine allows for 'blades' (basically HTML) to be created, these blades can then be used as sections or templates that can be applied with ease throughout Matooi. Below is a list of various parts throughout the site that leveraged this system.

- Top menu bar
- Navigation
- Sections to display each of the users saved trips
- Sections to display the details for the randomized locations
- The sections to display the final map for the user

For the backend design patterns the gang of four (GOF) principles were applied, this meant that the backend patterns would be separated into creational, structural and behavioural patterns.

- Creational Patterns – Several objects needed to be created in order to handle certain functions of the app, these were as follows:
    - Account object – this will handle the login, logout and creation of user accounts.
    - Locations object – this will be used to select, choose and get details about the random locations.
    - Trip saving object – this will be used to save and delete trips.
- Structural Patterns – It was decided that the adapter pattern would be used as every object would require its own interface.

- Behavioural Patterns – The behavioural pattern used was the 'chain of responsibility' pattern as each object for Matooi is independent of itself and each object will need to interact with the next on in order for the flow of the application to work (ie. User logs in > generates new locations > views the trip > saves the trip).

Database Design

As Matooi needed a database in order to store the user account information and saved trips a database needed to be designed in order to make sure that the structure and relationships would work. To do this an entity relationship diagram (ERD) was created as shown in figure 3AA.

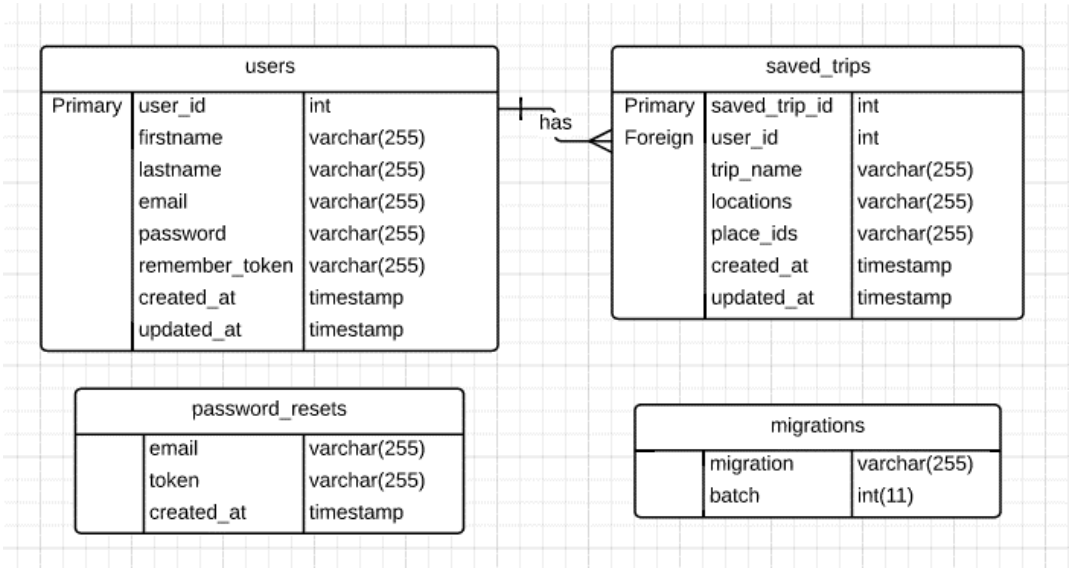


Figure 3AA – ERD showing the structure and design of the required database

As you can see Matooi required 4 tables, a user's table, saved trips table, password resets table and migrations table. The migrations table was created as the project uses Laravel, one of Laravels features allows you to create migrations for database tables and their contents so when you deploy or share your site with someone else they can easily migrate your migrations and that will create the required tables for the project.

The users table was created to store the information for the users, it contains a user id which is unique for each user and the primary key for the table. The first name and last name are used throughout the site to personalise it for each user, these are both stored as varchars. The email and password are used for the login in the site and the email is also used for sending users their password should they forget it. Emails stored must be unique and the passwords are stored in hash format to improve security. Finally the remember token, created at and

updated at are used for security and to allow the user to set a cookie for the site to remember them and log them on automatically. The user table has a 1 to many relationship with the saved trips table as one user is able to save many trips.

The saved trips table contains 2 id's, one of which is the saved trip id and is the primary key and the other which is the user id and is a foreign key. The saved trips table also contains a field for the name of the saved trip, this is input by the user when they are saving a trip. The next 2 fields are the locations field and the place id's field, both of these fields store information about the locations for the trip, the locations storing the longitude and latitude of the trip and the place id's storing the place id for each location taken from Google Places API.

The final table is the password reset table, this table is used for when a user requests to reset their password. The table stores the email of the password reset request and sends an email containing the token key, the created at field is then compared against the current time whenever a user clicks the link in their email. The user must use the link within an hour otherwise the token will not work and a new email must be sent.

Site Map

Based on the features and functionality that Matooi needed a sitemap was created to show the structure and connections that the pages would take. As shown in figure 3AB a guest user will be able to view the homepage, about, register and login pages whereas an authorized user will be able to view the account, new trip, locations, final trip and settings pages.

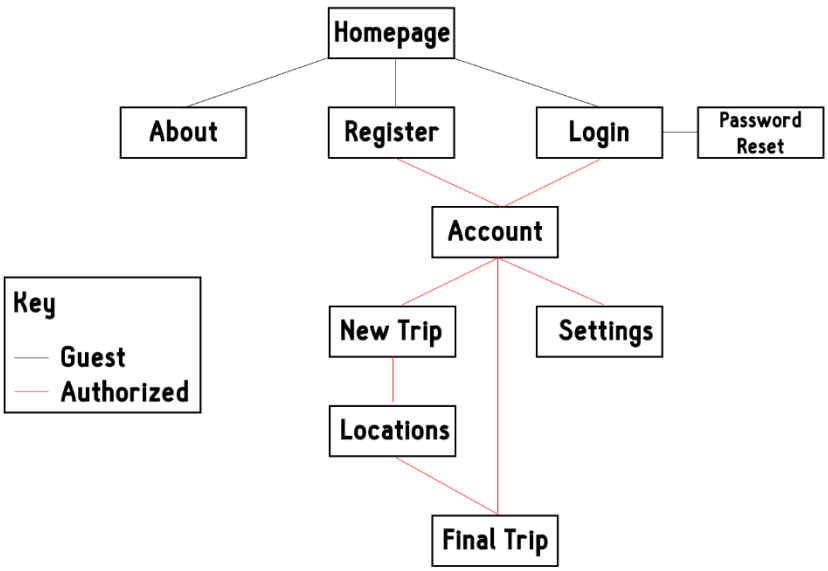


Figure 3AB – Final map of the site

The saved trips forms (such as save a trip and delete a saved trip) will use AJAX and therefore do not have a separate page for them. The account

page links through to the final trip page as a user will be able to select a saved trip and this will then take them through to the final trip page with the trips details.

4 – Implementation

The implementation section will give an in-depth look at the development process which took all of the designing and planning and put it into practice. This section will detail the major features of the Matooi along with what tools where used in the development process and any notable challenges that occurred throughout.

4.1 – Matooi Functionality

This section will give a detailed overview for each feature within Matooi, this will start with site wide features and then detail each pages functionality.

4.1.1 – Site Wide Functionality

Routing and Controllers

As Laravel is an MVC framework and uses a routing feature each of the major functions where created with specific routes and controllers being applied to them. For example when a user creates a new trip, via the '/trip/new' URI the route will direct the application to the 'TripsController'. Matooi contains 4 main controllers, these are:

- Welcome Controller – Access through this controller requires the user to be a guest and the routes pointing to it are for the homepage and about page.
- Profile Controller – Access to this controller requires the user to be authorized, the routes pointing to this controller are for the profile and profile editing functions.
- Trips Controller – Access to this controller requires the user to be authorized, the routes pointing to this controller are all of the trip URIs, such as creating a new trip or viewing a saved trip.
- Auth Controller – This controller is used to login, create accounts and reset passwords.

Setting up the controllers in this way allowed Matooi to use practical and smart URIs for its routing as all URIs to do with trips can be found at '/trip/' and anything to do with the users profile will be at '/profile/', etc.

Authorization

An authentication middleware was used in each of the controllers to specify whether a user needs to be logged in in order to access a certain URI. For Matooi the authenticate middleware is used to check if the user is logged in, if they are they are marked as 'authorized' and if they are not then they are marked as 'guest' this means that each controller has either the guest only or authorized only applied to them, these are as follows:

- Welcome Controller – Guest only, anyone marked as a guest trying to access other areas, i.e. the profile page, will be redirected to the homepage.
- Profile and Trip Controllers – Authorized only, anyone marked as authorized can access the rest of the app but cannot access the homepage and so any redirects for an authorized user will send them to the profile page.

Views and Blade Templates

Views were created for each page that the user can see, these views contain mostly HTML but also contain some Laravel specific content. To start off a base template was created for Matooi, this template contains the HTML for the navigation bar (both mobile and desktop) and uses Laravel's syntax to check if the user is logged in or out to display the different versions of the menu. Below the navigation bar there is more Laravel syntax to check if there has been a session message sent (in Matooi's case these can be either success or error messages) and if so it will display it.

The middle section contains a main section div with Laravel syntax to 'yield' the content, this means that for each of the other views this template will be applied and the content for the views will be placed into the main section. The final section contains the script tags along with another yield syntax for adding page specific scripting files, finally at the very bottom is the closing tags for the body and HTML.

Each of the pages that can be seen in Matooi are created using the base template along with the page specific view, these views are organized into a similar folder structure as the controllers, for example all the views relating to the trip functions and controller are in a folder called trip. The views are usually called at the last line in a function of a controller, for example when a user who is not logged in visits '/' (the base URL for Matooi) the controller

simply returns the view of 'pages/home' which in turn combines the HTML from the base template with the content of the home view and presents this to the user.

Requests

Another feature of Laravel is setting up a request class for each form. Each form within Matooi has a request class tied to it, this allows for setting up validation rules specific to each field in the form. The request class can then be applied to the controllers function that the post of the form routes to, this will make the form validate before it enters into the function and if it fails redirect the user back to the form to display the errors.

Location Helper Class

In order be able to access the Google Places API to obtain details about nearby and specific locations a helper class was made. The locations helper class main functions are to retrieve data from the places API, for example when a user selects that they would like a trip with 4 locations in a 40 mile radius the locations helper will first receive a list of up to 200 hundred locations within the radius. After it has been confirmed that there are enough locations available the specified amount will then be randomly chosen, for each location that is chosen the location helper will make another call to the Places API to get details about that location.

Responsive Development

Matooi was built responsively using the mobile and up method, this means that the menu and all elements will change position depending on screen size. The site contains a navigation bar with links to the right side when viewed on desktops and features an off canvas navigation menu that slides out from the right of the screen when the screen is smaller. Some elements, such as the users profile picture, disappear at very small resolutions to allow for a better UX by focusing on the important parts.

4.1.2 – Page Specific Functions

This section will give a list of functions and features that where created on each page.

Homepage

The homepage is made up mostly of static HTML, it contains an introduction to the web app, along with a CTA, and then moves on to explain what the application is about/ Finally at the bottom the page displays another call to action.

About

The about page is again made up of static HTML that details what Matooi is, why it was made and what tools and technologies were used to make it.

Register

The register page contains a form, when the form is submitted it is validated, if the validation passes then a new user is stored in the users table and the user is directed to the profile page.

Login

The login page contains a form, when the form is submitted it is validated, if the validation passes the user is logged in and directed to their profile page.

Password Reset

The password reset page contains a form, when the form is submitted it is validated, if the validation passes then a token is generated and stored in the password reset table along with the email entered. This token is usable for an hour and links the user to a form that allows them to reset their password.

Account

When the account page is loaded the controller gets the information of the currently logged in user and any trips that they have saved, using Laravel syntax each trip is displayed or a message that the user has no trips saved displays. Along with linking the user to the other features of Matooi the following functions happen.

When a user selects the option to delete a saved trip a form will display with a yes or no answer, if the user hits yes an AJAX function will pass the ID of the trip to delete it, if successful the page will refresh displaying a success session message.

Settings

The settings page contains 3 forms, one for the user to edit their account details, another to change their password and the third to change their profile picture. Each of these forms get validated when the user submits them and if they are successful then the users details get updated and they are redirected to the account page with a success session message.

New Trip

The new trip page has various functions that occur throughout it, these are as follows. Initially the user is asked to share their location, if they disallow it then an error message will show, otherwise Matooi will find the users location and display this on a Google Map.

There is a slider for the user to set the radius of the search, as the user slides this a set of JavaScript functions, one of these displays the value of the slider in an input field whilst another function passes the value into a function that draws a circle on the map around the user's location. This gives the app more of a visualization of how wide it will search for locations and help improve the UX.

When the form gets submitted it will need to pass validation, if it successfully passes validation then all of the users specifications for the types of locations they would like to see get passed into the location helper class. From here the nearest 200 locations are found and checked against the amount of locations the user has requested. If there are not enough locations the user is redirected back to the form with an error session message, if there are then the specified amount of locations are randomly picked from the list. This list only contains a place id for each location therefore another function in the location helper is run, this function gets each place id and passes it to Google Places API for more information on the location. An array is then created from these details picking out specific things, such as the name, longitude and latitude.

Locations

The locations page receives the selected locations array and displays all of this information to the user. In the background there is a hidden form that contains the place id, longitude and latitude values for each location, when the user clicks the next this data gets passed to the final trip page.

Final trip

The final trip page initially gets the users location again, this is in case the user created a trip at a location and then saved it and is now in a different starting location. The page gets the latlng value of each location and places this into a directions call from the Google Maps JavaScript V3 API, this displays the directions to the user. A function is then called to create

the export button link. The final part to the page allows the user to save the trip, this is done using a popup modal that contains a form.

The popup saved trip form sends the trip information, along with a name the user must enter, and saves it into the database. If the validation passes and the trip is saved successfully then a success flash message is displayed.

4.2 – Technology and Tools

This section will give an overview of the technology and tools used throughout the development stage of Matooi, each tool will be described in relation to why it was selected and how it was used.

4.2.1 – Git

Git is a version control system it works by creating a repository, this repository can contain several branches for development. The version control works by being able to add certain files and committing them, for example during Matooi's development after a feature was coded a commit would off been made titled with the feature.

Git was used throughout Matooi's development as it offered several benefits, it allowed for branches to be developed, this meant that the main branch (staging) could contain the working solution and other branches could be made to create other features or even experiment with certain features, when these were complete and working they could easily be merged into the staging branch. Another feature that Git allows is storing the repository using an external service, this is where BitBucket was used.

4.2.2 – BitBucket

Bitbucket is a hosting service for projects that use a revision control system. Bitbucket has various features such as issue tracking and pull requests and it is also free to use. Bitbucket was used for Matooi as it was a safe and secure way to back up the whole repository and codebase. Using issue tracking allowed the development to keep a list of functions and features that still needed work on.

4.2.3 – Terminal, Composer and Artisan

Terminal was used for Git, Composer and Artisan commands. Composer is a dependency manager for PHP it allows you to declare the dependencies that are required for the project

and will install and keep them up-to-date for you with 1 line of code. Artisan commands allow for quick and easy development in combination with Laravel, for example artisan commands such as 'artisan make:migration' allow you to easily create a migration to add a new column into your database.

Both Composer and Artisan commands where used during the development of Matooi, the main advantages being ease of use and helping to speed up development. Composer was used to initially setup a new Laravel project and also to install some dependencies. Artisan commands where used throughout development for various tasks, such as creating and running the migrations, creating controllers, creating models and using artisan tinker to test backend functionality via writing PHP in the terminal.

4.2.4 – Sublime Text

Sublime text is a text editor created specifically for coding, it has various shortcuts and quick keys and is easily customisable making it a useful tool in any developer's tool belt. The majority of Matooi's code was written with sublime text, its useful quick keys, easily customisable code colours and ability to download plugins (such as emmet) meant it was a vital tool for quick development.

4.2.5 – Dploy.io

Dploy.io is a deploy service from Beanstalk, it allows for quick and easy deployment of code from a repository to a test or live environment. As Matooi used Bitbucket as its hosted repository it used Dploy.io to allow for easy deployments to the live server once it was initially finished. The main reason for this was ease of use, it meant when changing or fixing code the code would be committed and pushed to Bitbucket, this commit would then get deployed to the live environment.

4.2.6 – Fortrabbit

Originally the live environment was planned to be the University server, however when the project had finished its initial development and was going to be deployed to the University server several problems occurred. The main issue was that Laravel expects to have its folder structure at the top level of the server infrastructure but the University server was a shared hosting platform.

It was decided an external hosting platform would be used, after looking into which services were available and suitable Fortrabbit was chosen. Fortrabbit is a PHP hosting service with flexible pricing based on usage. Fortrabbit is currently where Matooi's is hosted on its live environment.

4.3 – Notable Achievements and Challenges

The whole project was a challenge as it had used technologies that the student had little or no experience with using, this section will give a brief overview of any notable achievements and challenges that occurred during the project.

4.3.1 – Notable Achievements

When planning the project it was decided that the whole project should use technologies and create something that the student had never used/done before, as such below is a list of notable achievements that occurred throughout the project.

- Use of Laravel – The student was familiar with certain aspects of Laravel however had never created a project from start to finish using this framework. Specific achievements using the Laravel framework are as follows:
    - Creating a helper class – Creating the location helper class, or creating a usable class in general was never something that the student had done before.
    - Creating requests – The student had used validation within Laravel previously, however had never used the idea of creating a request class for each form.
- Use of APIs – The student had never used an API to pass information and retrieve data from, especially not two APIs that were used in conjunction with each other.
    - Google Place API – retrieving nearby locations and then creating functions to randomly select a specified amount of these locations and finally retrieve more information about each location.
    - Google Maps JavaScript V3 – Using Google Maps to display directions, the user's location and display a radius circle which adjusts based on the users input.
- Geolocation features – This was the first time that the student had used geolocation to get location information about the user.

4.3.2 – Notable Challenges

Throughout the development process several notable challenges occurred, this section will explain each challenge and how they were overcome.

Laravel Version Change

Just after the start of development had occurred Laravel 5 was released, with this release came substantial changes throughout, it was decided that the newer version would be used as it brought some features that would improve the overall development. This challenge was overcome by using some development time to also learn how to use and implement these new features.

Development Hardware Faults

The primary computer that was being used for Matooi's development had a hardware failure during the development stage, this lead to some downtime in the development whilst this problem was being fixed. Fortunately as the project was using Git and Bitbucket all of the projects source code was saved safely on this external service.

Production Server Problems

As previously mentioned when the project was finished and ready to be deployed to the production (University) server there were several issues with using Laravel within the shared hosting space. After several attempts at fixing this it was decided that an external service would be used to host Matooi instead of using up testing time trying to resolve this issue.

5 – Testing

This section will detail how testing was approached in regards to Matooi, it will then detail the testing process and give an overview of the test results.

5.1 – Test Approach Selection

When researching various testing approaches a pattern emerged for testing web applications, this pattern usually consisted of 3 categories of testing. These 3 categories are as follows. (Ibm.com, 2015)

5.1.1 – Usability Testing

Usability testing involved ensuring that the user interfaces complied with standards, for example ensuring error messages occur in red or other suitable warning colours. This broke

down into several tests that applied to Matooi such as testing that the links off the whole site work, ensuring the appropriate colours where used for errors and success messages and ensuring that the site's pages passed a validator.

5.1.2 – User Acceptance Testing

The main objective of user acceptance testing is to ensure the application is can run successfully and be able to do what it is meant to do. Several methods of testing can be applied to this stage such as browser compatibility and insuring form validation.

5.1.3 – Performance Testing

Performance testing is when we would put the application under stress tests and push it to its limits, this can be applied to Matooi as trying to reach the API limits and putting the server under load.

5.2 – Test Process

The testing process was made up of various tests in order to cover as many of the previously mentioned categories as possible. This section will detail what tests where applied to each category and how they tested Matooi.

5.2.1 – Usability Testing

For the usability testing several tests where ran with the aim of trying to ensure that Matooi had a good and easy to use interface, these are as follows.

Validating code

To test that the code on each page was valid the W3 Validator tool (Validator.w3.org, 2015) was used. This was done by submitting the URL for each page in the Matooi web app to the validator and then fixing any errors that occurred.

User Observation

The user observation test consisted of 3 users who were given access to Matooi, they were then asked to complete several tasks and whilst they were doing this they were asked explain what they were thinking. The user observations where done with 1 person using a mobile and 2 using the desktop version, each user started on the homepage of the site without an account and where asked to complete the following tasks:

1. Register an account on the site.

2. Create a new trip that contains 3 locations within a 45 mile radius, the locations must be restaurants.
3. Save the trip you created.
4. Edit your password for your account.
5. View the trip that you previously saved.

5.2.2 – User Acceptance Testing

This stage mainly tested the functionality of the whole site, this was done through a case test. The test was setup displaying what the application should be doing throughout and then checked to see if this was the case. The site was also tested in all the latest browser to check that it functioned correctly.

5.2.3 – Performance Testing

Matooi was put under a stress test to see how fast it would take before it reached its limit. This was done using an application called 'WAPT' (Loadtestingtool.com, 2015) that was able to create up to 20 people using the application at one time, this made it possible to see how much each trip creation used for the limits of the APIs and also see how stable the server was.

The test that was ran in WAPT started at the homepage, it then visited the register and about page and then logged in. Once logged in the program would create a new trip and finally go to the profile page and view a previously saved trip. This test simulated 20 people doing this at the same time.

5.3 – Test Results

This section will give an in-depth look at the results of each test, it will also explain for each test if anything was changed in Matooi as a result of them.

5.3.1 – Validating Code

The validating code test consisted of going through each page within Matooi using the W3 Validator, the tests mostly returned positive with a few exceptions. Some pages where missing alt tags for the images, these were easily fixed and revalidated. All pages where tested and fixed until they returned as valid as shown for the homepage in figure 5A.

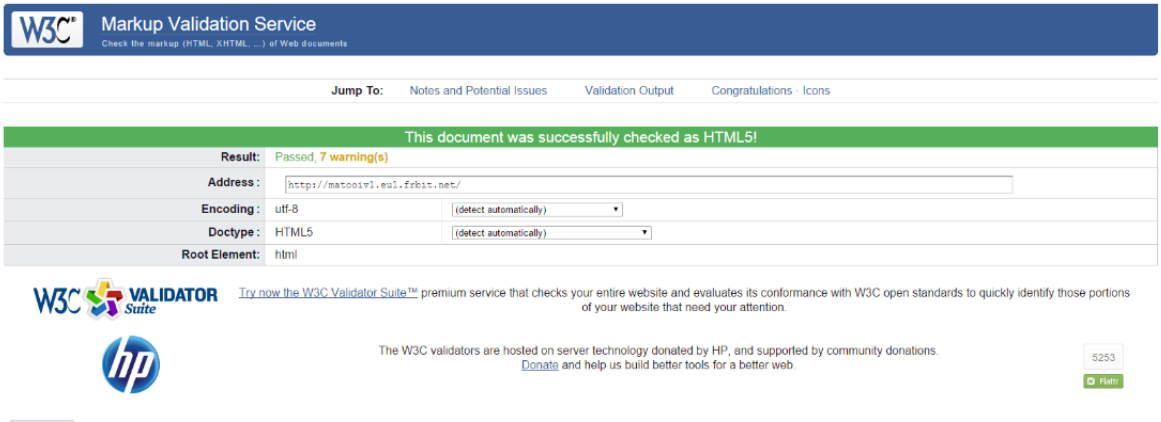


Figure 5A – The homepage being shown as valid using W3 Validator

Another site was found during testing called Nibbler (Nibbler, 2015), this site scanned through the pages of Matooi and then gave them a score in various area's based on certain criteria. The categories were as follows:

- Accessibility – This was based on headings, internal links, mobile compatibility, page titles, URL format and code quality.
- Experience – This was based on social networking site links, popularity, internal links, server behaviour, images, mobile compatibility, printability and URL format.
- Markerting – This was based on social networking site links, analytics, social interest, popularity, amount of content, internal links, domain age, page titles and meta tags.
- Technology – This was based on headings, internal links, server behaviour, images, domain age, mobile compatibility, meta tags, printability, URL format and code quality.

The results, shown in figure 5B, was highly rated, the main thing that brought the score of the site down was the lack of social networking links and the popularity of the site (which as it is only available is not fair to judge). Removing the marketing score from the total brings the overall score to '8' which is very respectable.

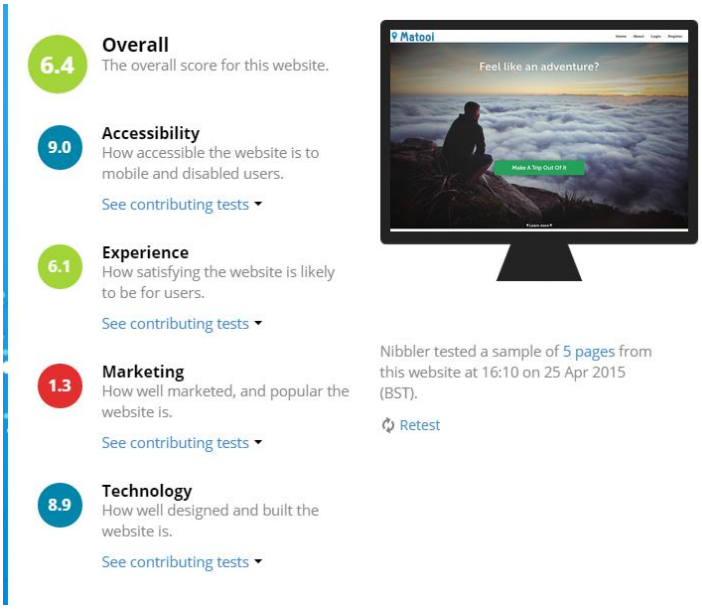


Figure 5B – The score of Matooi's overall experience using Nibbler

5.3.2 – User Observation

As mentioned this test involved observing 3 users whilst they were asked to complete certain tasks the table of results can be found in Appendix B. The results and comments were mostly positive along with the time it took for each user to complete the task. The main issue found during the observations was that some users tried to find access to the account settings page through the menu on the site. Because of the results it was decided that a link would be added to the main navigation menu on desktop and mobile version to make the edit account page more accessible.

5.3.3 – Case Testing

The case testing results were stored in a table, the full table can be seen in Appendix C, this section will give an overview of the results. The tests where segmented into each page and the navigation (both logged in and logged out). All of the tests that were ran in the case test completed successfully meaning that all aspects of Matooi were in complete and working order meaning that Matooi was fully functional and achieved its functional requirements.

5.3.4 – Browser Compatibility Testing

One of the objectives of Matooi was to create a web application that feels like a native application on mobile devices, to ensure that the site is usable and will give every user the same experience browser compatibility was done. The browsers that Matooi was checked on for this test involved all of the latest versions of the most popular browsers (both mobile and desktop). This test consisted of going through all of the latest browsers using an online tool

called Browserstack (Browserstack.com, 2015), the full list of browsers that were tested is as follows:

- Android
  - Galaxy S5
  - Nexus 6
  - Nexus 5
  - Nexus 9
  - Nexus 7
- Windows Phone
  - Lumia 930
- iOS
  - iPhone 6 Plus
  - iPhone 6
  - iPhone 5
  - iPad Air
  - iPad Mini
- Windows 8.1
  - IE 11
  - Firefox 36
  - Chrome 41
  - Opera 27
- OS X Yosemite
  - Safari 8
  - Firefox 36
  - Chrome 41
  - Opera 27

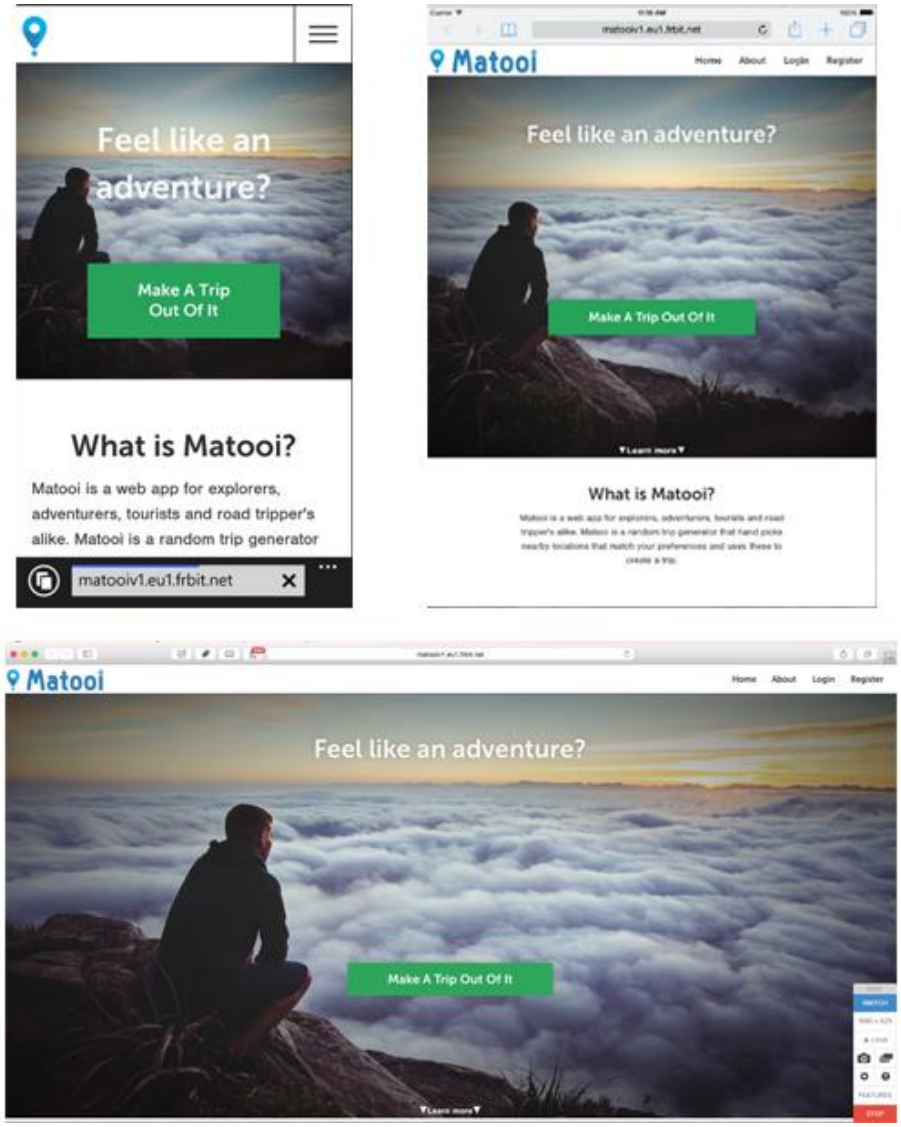


Figure 5C – Windows Phone, IPad Mini and Safari displaying Matooi from Browserstack

Figure 5C shows the homepage displayed on a Lumia 930, iPad Mini and OS X Safari 8, during testing the only issue found was the background images on the homepage not displaying correctly due to the use of the cover attribute, this was easily fixed and all devices were retested. Matooi is capable of running with consistent usability on all modern browsers, mobile and desktop.

5.3.5 – Performance Testing

The performance testing involved using WAPT 8.1 to try and stress out the application. The test was ran by recording the actions that needed to be tested on the site, once these were recorded the test was ran several times emulating 20 people using the site per test. Figure 5D shows the setup of one of the tests.

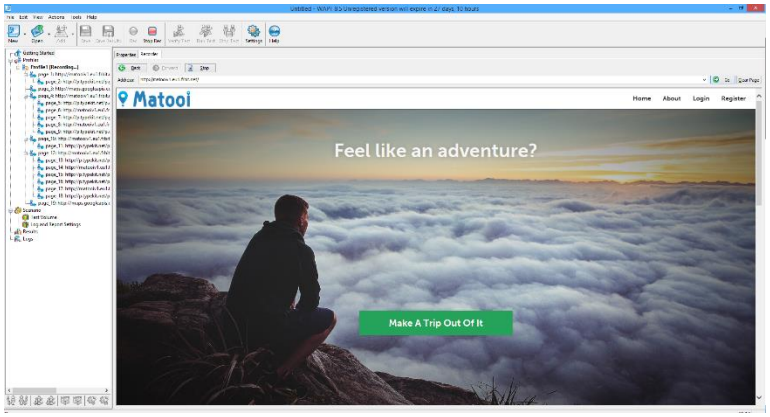


Figure 5D – WAPT setting up the tests for Matooi

The test was ran multiple times and included stress tests and usage tests to see if the server could reach its limits. After several tests no errors had occurred, the summary of one of the tests can be seen in figure 5E.

Summary

Profile	Successful sessions	Failed sessions	Successful pages	Failed pages	Successful hits	Failed hits	Total KBytes sent	Total KBytes received	Avg response time, sec (with page elements)
Profile1	255	0	4969	0	15186	0	9763	825330	0.08(0.57)

Number of active users

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00
Profile1	5	11	17	20	20	20	20	20	20	20
Total	5	11	17	20	20	20	20	20	20	20

Successful sessions (Failed sessions)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile1	2(0)	10(0)	22(0)	28(0)	33(0)	32(0)	29(0)	38(0)	28(0)	33(0)	255 (0)
Total	2(0)	10(0)	22(0)	28(0)	33(0)	32(0)	29(0)	38(0)	28(0)	33(0)	255 (0)

Successful pages (Failed pages)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile1	63(0)	249(0)	424(0)	609(0)	557(0)	649(0)	611(0)	607(0)	624(0)	576(0)	4969 (0)
Total	63(0)	249(0)	424(0)	609(0)	557(0)	649(0)	611(0)	607(0)	624(0)	576(0)	4969 (0)

Successful hits (Failed hits)

Profile	0:00:00-0:01:00	0:01:00-0:02:00	0:02:00-0:03:00	0:03:00-0:04:00	0:04:00-0:05:00	0:05:00-0:06:00	0:06:00-0:07:00	0:07:00-0:08:00	0:08:00-0:09:00	0:09:00-0:10:00	Total
Profile1	254(0)	822(0)	1335(0)	1819(0)	1774(0)	1843(0)	1875(0)	1888(0)	1749(0)	1827(0)	15186 (0)
Total	254(0)	822(0)	1335(0)	1819(0)	1774(0)	1843(0)	1875(0)	1888(0)	1749(0)	1827(0)	15186 (0)

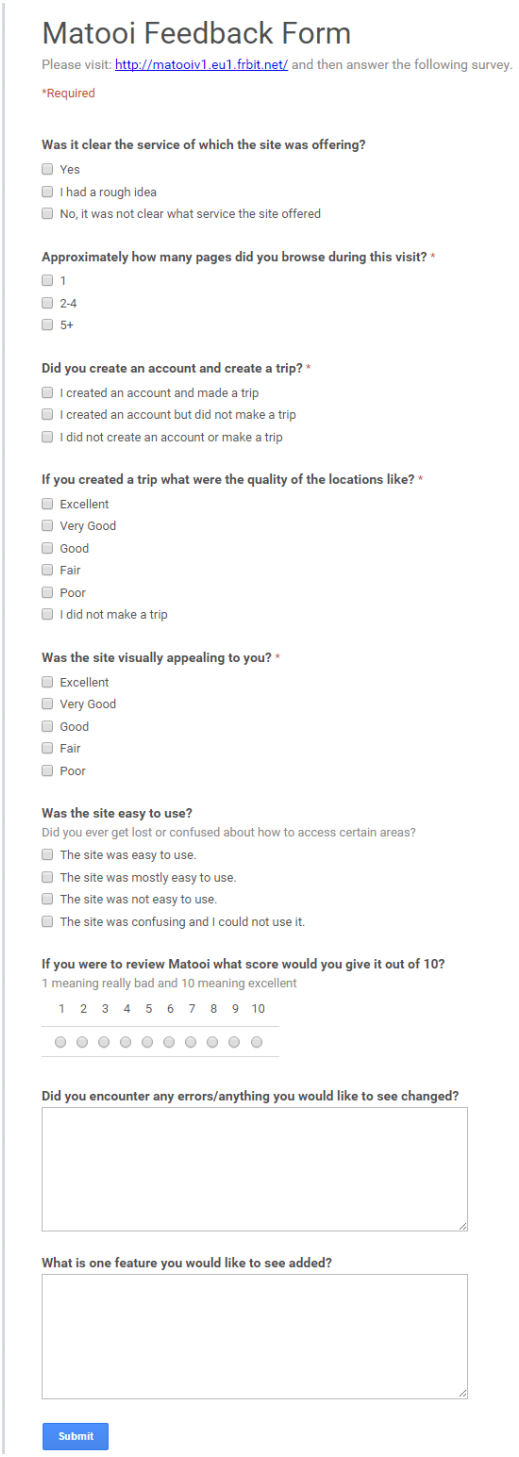
Figure 5E – Results of the tests from WAPT on Matooi

5.4 – User Survey

The final step in testing Matooi involved creating a user survey, this user survey was placed online for a week and linked on various forms related to trips asking for users to complete it. The questions where created using Google Docs questionnaire, as this allowed for easily customisable questions and also immediate access to any answers submitted, for the full table of results see Appendix D.

5.4.1 – Survey Questions

The aim of the survey was to find out 3 things about Matooi as such the questions were created to attempt to get an idea of a typical user's point of view. The 3 main topics that the survey tried to cover where to get feedback on whether the home and about pages effectively describe the application and enticed people to sign up. The second topic was to try and see if the application was easy to use and aesthetically pleasing. The third topic was to see if the users liked the locations that they were receiving in their random trips. The last 2 optional questions asked the user if they found any errors and if they could add one feature what would it be.


Matooi Feedback Form

Please visit: http://matooiv1.eu1.frbit.net/ and then answer the following survey.

*Required

Was it clear the service of which the site was offering?
- Yes
- I had a rough idea
- No, it was not clear what service the site offered

Approximately how many pages did you browse during this visit? *
- 1
- 2-4
- 5+

Did you create an account and create a trip? *
- I created an account and made a trip
- I created an account but did not make a trip
- I did not create an account or make a trip

If you created a trip what were the quality of the locations like? *
- Excellent
- Very Good
- Good
- Fair
- Poor
- I did not make a trip

Was the site visually appealing to you? *
- Excellent
- Very Good
- Good
- Fair
- Poor

Was the site easy to use?

Did you ever get lost or confused about how to access certain areas?
- The site was easy to use.
- The site was mostly easy to use.
- The site was not easy to use.
- The site was confusing and I could not use it.

If you were to review Matooi what score would you give it out of 10?

1 meaning really bad and 10 meaning excellent

1 2 3 4 5 6 7 8 9 10

Did you encounter any errors/anything you would like to see changed?

What is one feature you would like to see added?

Submit

Figure 5F – The user survey made using Google Docs

5.4.2 – Survey Response

Over the week that the survey was available it generated 11 responses, most of the feedback revived was positive, this section will go through each question and give an overview of what was learnt from the answers.

Approximately how many pages did you visit?

This question was asked to see how much the user had used the app and also to help see if they visited enough pages to of created a trip. The majority of users visited 5+ pages whilst 4 users visited between 2 and 4 pages and finally 1 users visited 1 page. This allowed us to see that the majority of people answering the survey had most likely signed up and potentially created a trip.

Did you create an account and a trip?

Again this question was asked to see what areas of Matooi the rest of each user's answers would apply to, the majority of users created an account and made a trip, whilst 1 user created an account but did not make a trip and finally 2 users neither created an account or a trip. From these responses we can see that the majority of users were able to experience most of the application and therefore their results can be applied to all sections.

How would you rate the visual appeal of the site?

The answer to this question was based on a 5 point scale rating from poor to excellent. As shown in figure 5G the majority of users rated the visual appeal of the site as very good this was a good rating overall however showed that there was room for improvements.

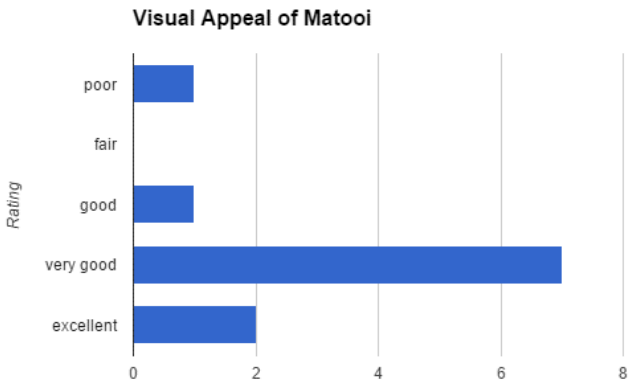


Figure 5G – Bar Chart showing the answers from the survey

Was the site easy to use?

This question had a 3 point scale rating which was 'the site was easy to use', 'the site was mostly easy to use' and 'the site was confusing and I could not use it'. The majority of users said that the site was easy to use, 4 users stated that the site was mostly easy to use and finally 1 person stated that the site was confusing. At this point it should be mentioned that one of the users responses for all of the questions always selected the most negative answer, this means that the user who completed the survey with these answers may not have used the site and/or filled out the survey quickly and negatively on purpose. Overall the answers were mainly positive meaning that the users found Matooi easy to use.

If you created a trip what did you think of the quality of the locations?

This question was created to see what the users thought of the quality of the locations that were chosen for them again it used a 5 point scale from poor to excellent and also had an option for user who did not create a trip. Overall this question received a mixed response as sometimes some of the locations that are selected in Matooi can be places that you may not want to go, this will be explained in more detail in the evaluation section.

User Rating of Matooi.

The final required question asked users to rate Matooi out of 10, this was to get an overview of how each user would rate Matooi. Overall the highest rating was 9 and the lowest was 1, the most selected rating was 7. For an initial release of Matooi the rating is not bad, however it also shows that there is still some room for improvement.

User input questions

The final 2 questions were not required and also were asking for written feedback on Matooi. Out of the 11 users only 3 answered both questions and 1 answered 1 of them.

The initial question asked if the user ran into any issues whilst using the web application, 1 person answered saying they did not find any errors whilst the other two answers were more suggestions than errors. One asked to show more pictures of the application on the homepage so first time users would get more of an idea of what the app is. The final user mentioned that sometime the application picked locations that they did not want to visit.

The final question asked if there were asked if they could add 1 feature to Matooi what would it be, several good points where raised, here are the features that where mentioned:

- Expand the search radius and amount of locations available for one trip.
- Two users suggested being allowed to change a location if they didn't like it, for example there could be a button to repack each location.
- A user asked for the ability to share their trip with their friends.

5.4.3 – Survey Summary

Overall the survey was invaluable as it gave insight into how users view Matooi's usability and features. The survey in summary showed that Matooi is currently in a fully functional and usable state, however there are some improvements that could be made to make the overall application feel and run smoother.

Feedback from users suggested several features that could be implemented in Matooi's future such as sharing their trips on social sites and being able make Matooi re-pick a location if they don't like it.

6 – Evaluation

This section will evaluate the testing, methodology, project outcomes and plan and explain if ultimately the project met its aims and objectives successfully.

6.1 – Testing Evaluation

The testing involved allot of in-depth tests each of which tested a unique aspect of the application. During the testing stage some errors or issues occurred however generally these

were easy to fix and therefore were fixed and retested. One of the most vital tests was the user survey this gave a real look at how typical users perceived Matooi. The feedback received from the user survey allowed us to see where the project would head next and how it should improve.

As mentioned the user survey showed mixed results for the locations that the application generated, this can be looked into further as when testing some of the locations pulled from the Google Places API can be a place that you wouldn't consider going for a trip. For example if you search for galleries some shops that sell art will appear in the locations as a gallery, this is not an issue with Matooi however and is more so the fault of the Google Places API. To fix this issue more filtering of locations could be looked into or potentially a new API could be implemented into Matooi if it contained more suitable locations.

Overall the tests used were extensive and allowed testing of the application from various aspects, specifically the usability, user acceptance and performance testing. They have given a good overview of what Matooi is in its current form and whilst it is a usable and successful project it also has room to improve and expand in the future.

6.2 – Project Outcomes

Initially the aim of the project was to create a web application that would test the students' skills and knowledge and challenge what they had learnt over the past 4 years. Ultimately the project was successful and it also challenged the students' skills by using technologies and that they had never used before.

The aim of the project was to create a web application that worked responsively on desktop and mobile. The application should be able to generate a random trip for the user to take and show this to them in the form of a map. Matooi in its current form is capable of this and much more, such as the ability to save trips and export them to Google Maps. As the aim of the project and each of the functional and non-functional requirements that where set have been achieved Matooi can be deemed a successful project.

6.3 – Methodology Evaluation

Early in the process it was decided that Matooi would use the modified waterfall method to help plan how the project would be created and developed. The modified waterfall methodology is a series of steps through development, starting at investigation and ending

with evaluation. This methodology was chosen as it matched each step of the deadlines and reports.

Overall the modified waterfall methodology enabled the creation of a focused plan which matched the deadline requirements at each stage. This methodology allowed Matooi to be planned, designed and developed in a step by step process but also allowed it to go back a step if required giving the project a bit more freedom if changes needed to be made. Ultimately the methodology applied helped the project become a success, however if the deadlines did not exist a scrum methodology maybe would have been better as it would allow the users to be involved more often which may of helped refine the project.

6.4 – The Plan Evaluation

A plan and schedule were defined after the methodology was chosen, this was to help keep the project focused and also help it become reality. The plan which incorporated the modified waterfall methodology to break the project into steps enabled allot more focus at each step throughout.

Initially the investigation and planning phases created a set of aims and objectives based off of the project idea of creating a random trip generator. At this stage it was thought that the requirements created were good and allowed the project to achieve the aim however after the more recent testing, specifically the user survey, it was decided that if the investigation stage could be done again it would be a good idea to incorporate a typical user more often. Users were involved during the investigation stage however the main set of functional and non-functional requirements were created as a logical set of requirements in order to achieve the aim. This has not affected the outcome of the app in a massively negative way however as Matooi has achieved the aims and also received good feedback during the user survey.

The tools and technology selection stage involved allot of research to see what tools could be used, especially in combination with each other in order to bring Matooi to life. Ultimately the in-depth research that was conducted allowed for a good selections of tools and technologies to be found which in turn lead to easier and faster development.

The design phase initially was only planned to have 2 main stages, the initial designs and the refined designs, however after receiving some negative feedback on the refined designs it was decided that the designs needed an overhaul before they were applied to the project.

This is one of the benefits of using the modified waterfall methodology as it allowed the project to take a step back to the design stage in order to refine the UX. Ultimately the projects schedule was not affected in any major way as of this. Overall the final refinement of Matooi allowed for an overall more visually appealing product with the initial designs and paper prototypes helping to achieve a good UX.

The implementation stage was quick and relatively straightforward due to the planning and design phases. No major issues occurred during the development of Matooi the only thing that may off affected the timing was that several pieces of technology required more learning in order to use appropriately, however this was accounted for in the original schedule and so did not cause any issues.

7 – Conclusion

This final section will summarize the report and reflect on what occurred throughout. It will then suggest future work that could be applied to Matooi.

7.1 – Report Summary

In summary this report has covered, in depth, the creation of Matooi a random trip generator. The report has outlined how the projects idea was originally created and researched and then developed. After the idea had been fleshed out and an aim and several objectives had been set the web app was the designed and developed. After the development had finished the project was then tested in various ways to ensure it was fully functional and usable from a typical user's perspective. Finally the project as a whole was evaluated and from this it was deemed that the project had met its aims and objectives and therefore was a success.

7.2 – Reflection of Work Undertaken

As the project was created in its entirety by one person, the student, they had to take up different roles throughout each stage. Initially during the idea generation allot of ideas were created, however the random trip generator was the only one that was really fleshed out. After choosing to take this idea further aims and objectives were created, it was at this stage that if the project had to be redone more input would be taken from a typical users point of view.

After the requirements and technology selection was completed the design and development of the web app occurred. It was during the design phase that the brand and brand name 'Matooi' was created. After the development had completed the app was put through several tests to test all aspects of the app.

The final section required Matooi to be evaluated and it was here that the project was deemed successful as it had achieved its main aim. Overall the project created a solid plan initially, mainly throughout applying the modified waterfall method to the schedule, this gave an insight as to how much time each section would have. Due to the fact that extra time was given to the stages, specifically the development, it allowed the development to stay focused whilst also allowing for experimentation and learning throughout. Ultimately the research and effort that went into the planning allowed Matooi to successfully achieve its aims and objectives.

7.3 – Future Work for Matooi

As Matooi is now a fully functional random trip generator the next step would be to improve and evolve it. Many features and functions could be added to it, some of which were suggested through user surveys. Below are potential features and improvements for the future of Matooi.

- Social sharing – allow users to share links to their randomized trips on social sites.
- Re-pick locations – allow the users to re-pick a location if they do not like it instead of having to resubmit a whole new trip request.
- Monetization – allow potential clients to pay for their location to be suggested during the list of locations or other onsite adverts.

7.4 – Conclusion

Overall from the initial investigation to writing this conclusion the student has been challenged and applied themselves in order to bring Matooi from idea to reality. The original idea was to create a web application that would allow users to generate a random trip, this idea was successfully achieved along with all of the functional and non-functional requirements. Matooi in its current form is a fully working application, as it stands it is a good base that could now be expanded upon by adding more features and adjusting its current functionality to make Matooi even better in the future.

8 - References

Browserstack.com, (2015). Sign into the Best Browser Testing Tool. [online] Available at: https://www.browserstack.com/ [Accessed 26 Apr. 2015].

Google Developers, (2015). Google Maps JavaScript API v3 Google Developers. [online] Available at: https://developers.google.com/maps/documentation/javascript/ [Accessed 16 Apr. 2015].

Google Developers, (2015). Google Places API Google Developers. [online] Available at: https://developers.google.com/places/ [Accessed 16 Apr. 2015].

Ibm.com, (2015). Summary of web application testing methodologies and tools. [online] Available at: http://www.ibm.com/developerworks/library/wa-webapptesting/ [Accessed 25 Apr. 2015].

jquery.org, j. (2015). jQuery. [online] Jquery.com. Available at: http://jquery.com/ [Accessed 14 Apr. 2015].

Ledbrook, L. (2012). Waterfall Project Management: An Overview. [online] Project Community. Available at: http://projectcommunityonline.com/waterfall-project-management-an-overview.html [Accessed 15 Apr. 2015].

Loadtestingtool.com, (2015). WAPT - Web Application Load, Stress and Performance Testing. [online] Available at: http://www.loadtestingtool.com/ [Accessed 25 Apr. 2015].

Mandrill, (2015). Transactional Email from MailChimp - Mandrill. [online] Available at: https://www.mandrill.com/ [Accessed 16 Apr. 2015].

Mysql.com, (2015). MySQL :: The world's most popular open source database. [online] Available at: https://www.mysql.com/ [Accessed 14 Apr. 2015].

Nibbler, (2015). Website report for matooiv1.eu1.frbit.net. [online] Available at: http://nibbler.silktide.com/en/reports/matooiv1.eu1.frbit.net [Accessed 25 Apr. 2015].

Otwell, T. (2015). Laravel - The PHP Framework For Web Artisans. [online] Laravel.com. Available at: http://laravel.com/ [Accessed 15 Apr. 2015].

Php.net, (2015). PHP: Hypertext Preprocessor. [online] Available at: http://php.net/ [Accessed 13 Apr. 2015].

Roadtrippers Community, (2015). Random Road Trip Generator | Roadtrippers Customer Community. [online] Available at: https://getsatisfaction.com/roadtrippers/topics/random_road_trip_generator [Accessed 10 Apr. 2015].

W3schools.com, (2015). CSS3 Introduction. [online] Available at: http://www.w3schools.com/css/css3_intro.asp [Accessed 13 Apr. 2015].

W3schools.com, (2015). HTML5 Introduction. [online] Available at: http://www.w3schools.com/html/html5_intro.asp [Accessed 13 Apr. 2015].

W3schools.com, (2015). JavaScript Tutorial. [online] Available at: http://www.w3schools.com/js/ [Accessed 14 Apr. 2015].

ZURB, i. (2015). Foundation | The Most Advanced Responsive Front-end Framework from ZURB. [online] Foundation.zurb.com. Available at: http://foundation.zurb.com/ [Accessed 15 Apr. 2015].

9 - Appendix

9.1 - Appendix A

The full list of functional and non-functional requirements are presented below in the form of snow cards.

Requirement Number: 1		**Requirement Type:** Functional
Description: A User will be able to create an account on the web app		
Rationale: A user will need to sign up to the web app to be able to use it, this is so we can allow users to store their trips.		
Fit Criterion: A user will be able to create an account through a sign up form.		
Version: 1	**Dependencies:** N/A	**Last Modified:** 14/11/14

Requirement Number: 2		**Requirement Type:** Functional
Description: A with an existing account will be able to securely sign in to the app.		
Rationale: A user that has an existing account will be able to sign into the web app in order to be able to use it.		
Fit Criterion: A user will be able to login to the app if they already have an account, once logged in this will bring them to the account screen through which they will be able to access the rest of the features of the web app.		
Version: 1	**Dependencies:** 1	**Last Modified:** 14/11/14

Requirement Number: 3		**Requirement Type:** Functional
Description: A user who is signed in will be able to securely log out.		
Rationale: A user that has an existing account will be able to sign out of the app.		
Fit Criterion: A user that is logged in will have the option to sign out of the app.		
Version: 1	**Dependencies:** 1,2	**Last Modified:** 14/11/14

Requirement Number: 4		**Requirement Type:** Functional
Description: A user will be able to edit their account settings.		

Rationale: A user will be able to edit some basic information for their account, such as their password, this is to allow them to have more security by changing their password if the one they use has been compromised.

Fit Criterion: A user will be able to edit their account settings through an account settings page.

Version: 1 **Dependencies:** 1,2 **Last Modified:** 23/03/15

Requirement Number: 5 **Requirement Type:** Functional

Description: A user will have access to generating a random trip.

Rationale: A user will be able to get access to the feature that allows them to generate a random trip, this means that you need an account to generate a trip as it needs to store certain information.

Fit Criterion: Only a registered user will be able to access the generate trip page.

Version: 1 **Dependencies:** 1,2 **Last Modified:** 14/11/14

Requirement Number: 6 **Requirement Type:** Functional

Description: A user will need to enter details about the type of trip they would like to have generated.

Rationale: A user will be able to affect the search parameters of the random trip generator, this will allow them to customize the area that they would like to search and the amount of locations they would like to stop at.

Fit Criterion: The user will be asked to enter the miles radius they would like to search, the amount of stops they would like to make and select the types of locations they would like to visit.

Version: 1 **Dependencies:** 1,2,5 **Last Modified:** 23/03/15

Requirement Number: 7 **Requirement Type:** Functional

Description: The app will need to generate a specified amount of locations within a set radius.

Rationale: The app will need to generate a list of locations so that it can connect the locations together on a map to display on the map.

Fit Criterion: The app will generate a list of locations that fit the search criteria and display this to the user.

Version: 1 **Dependencies:** 1,2,5,6 **Last Modified:** 14/11/14

Requirement Number: 8 **Requirement Type:** Functional

Description: The locations will be sorted in order of distance to optimize the route the trip will take.

Rationale: This will make the overall experience of the app better as it makes sense ordering the locations by closest.

Fit Criterion: The app will show the user the list of locations and then optimize the route when generating the trip map.

Version: 1 **Dependencies:** 1,2,5,6,7 **Last Modified:** 23/03/15

Requirement Number: 9 **Requirement Type:** Functional

Description: The app will need to connect the locations together on a map and display this to the user.

Rationale: This is so that the user will be able to see the output from the app, which will be a map of a trip for them to take.

Fit Criterion: The app will connect all of the locations together on a map and display this map with directions to the user.

Version: 1 **Dependencies:** 1,2,5,6,7,8 **Last Modified:** 14/11/14

Requirement Number: 10 **Requirement Type:** Functional

Description: The user will be able to save the trip for viewing later.

Rationale: This means that users will be able to build up a range of random trips and will be able to view them at a later date.

Fit Criterion: The app will allow the user to save the trip which will be accessible by their account at a later date.

Version: 1 **Dependencies:** 1,2 **Last Modified:** 14/11/14

Requirement Number: 11 **Requirement Type:** Functional

Description: The user will be able to export the final trip to Google Maps.

Rationale: This will allow them to use Google Maps verbal instructions and allow them to save the trip for offline usage.

Fit Criterion: The user will have the option to export the final trip to Google Maps.

Version: 1 **Dependencies:** 1,2 **Last Modified:** 14/11/14

Requirement Number: 12 **Requirement Type:** Functional

Description: The user will be able to delete previously saved trips.

Rationale: If a user has a previously saved trip they will be able to remove this from their account, this will allow them to keep the saved trips organized.

Fit Criterion: On the users previously saved trips page there will be an option on each trip to delete it.

Version: 1 **Dependencies:** 1,2 **Last Modified:** 14/11/14

Requirement Number: 13 **Requirement Type:** Functional

Description: The app will work on mobiles and tablets.

Rationale: As a user may want to use this app on the go the app will need to make to work responsively on mobile devices whilst also adjusting to larger screens.

Fit Criterion: The app will be made from a mobile and up approach allowing to achieve a native application feeling whilst also working on larger screens.

Version: 1 **Dependencies:** N/A **Last Modified:** 14/11/14

Requirement Number: 14		**Requirement Type:** Non-functional
Description: The app will initially be available in English and use the UK metric system.		
Rationale: This is so that the initial launch of the app will not require using up time on making the app multilingual but could be expanded upon after its initial release.		
Fit Criterion: The app will work with the UK metric system and be written in English.		
Version: 1	**Dependencies:** N/A	**Last Modified:** 14/11/14

Requirement Number: 15		**Requirement Type:** Non-functional
Description: The app will need to load quickly on mobiles that are potentially using 3g.		
Rationale: As users may use the map on the go they may only be able to connect to slow networks therefore the app should be made to load as quickly as possible.		
Fit Criterion: The app will use various tools to enable quick loading for APIs and any plugins that are used.		
Version: 1	**Dependencies:** N/A	**Last Modified:** 14/11/14

9.2 - Appendix B

The table of results below are the results from the user observation test that was undertaken with 3 users, 2 of which were using desktop version and 1 was using mobile.

User 1 – Desktop version		
Task	**Time to Complete (approx.)**	**Observations**
Register an account on the site.	45s	Found task straight forward. Said that site required little amount of details to use.
Create a new trip that contains 3 locations within a 45 mile radius, the locations must be restaurants.	1m 30s	Found task straight forward. Mentioned retrieving locations was a bit slow.

Save the trip you created.	30s	Found task straight forward.
Edit your password for your account.	1m 20s	Found it strange to have to go back to the profile page to access settings.
View the trip that you previously saved.	30s	Found task straight forward.

User 2 – Desktop version		
Task	**Time to Complete (approx.)**	**Observations**
Register an account on the site.	30s	Found task straight forward. Commented on the imagery used on the homepage.
Create a new trip that contains 3 locations within a 45 mile radius, the locations must be restaurants.	1m	Found task straight forward. Commented on the more info section for each location asking why it linked to another site.
Save the trip you created.	20s	Found task straight forward.
Edit your password for your account.	50s	Found task straight forward.
View the trip that you previously saved.	20s	Found task straight forward.

User 3 – Mobile version		
Task	**Time to Complete (approx.)**	**Observations**
Register an account on the site.	1m	Found task straight forward. Commented on the slide out menu saying it felt like something an app would have.

Create a new trip that contains 3 locations within a 45 mile radius, the locations must be restaurants.	1m 20s	Found task straight forward. Commented on the locations imagery being quite big.
Save the trip you created.	35s	Found task straight forward.
Edit your password for your account.	1m 50s	Had trouble finding the edit account page, eventually asked for help.
View the trip that you previously saved.	30s	Found task straight forward.

9.3 - Appendix C

The following table contains a full list of tests that were conducted on each page along with their expected result and actual result.

Homepage			
ID	*Test*	*Expected Result*	*Actual Result*
A1	Click the 'make a trip out of it button'	The button should link to the registration page	The button links to the registration page
A2	Click the 'sign up now' button	The button should link to the registration page	The button links to the registration page
Navigation (Not Logged In)			
ID	*Test*	*Expected Result*	*Actual Result*
B1	Click the 'home' button	The button should link to the homepage	The button links to the homepage
B2	Click the 'about' button	The button should link to the about page	The button links to the about page
B3	Click the 'login' button	The button should link to the login page	The links to the login page

B4	Click the register button	The button should link to the register page	The button links to the register page
Navigation (Logged In)			
ID	*Test*	*Expected Result*	*Actual Result*
C1	Click the 'New trip' button	The button should link to the new trip page	The button links to the new trip page
C2	Click the 'profile' button	The button should link to the profile page	The button links to the profile page
C3	Click the 'settings' button	The button should link to the edit profile page	The button links to the edit profile page
C4	Click the 'sign out' button	The button should sign the user out and redirect them to the homepage	The button signs the current user out and redirects them to the homepage
Register			
ID	*Test*	*Expected Result*	*Actual Result*
D1	A user inputs invalid data into the form	The user should be redirected back to the form with the errors displayed	The user is redirected back to the form with the errors displayed
D2	A user enters appropriate data	The user's account should be created and they should be logged into their account and redirected to the profile page	The user is created and they are redirected to their profile page
Login			

ID	*Test*	*Expected Result*	*Actual Result*
E1	A user inputs invalid data into the form	The user should be redirected back to the form with the errors displayed	The user is redirected back to the form with the errors displayed
E2	A user enters appropriate data	The user should be logged in and directed to the profile page	The user is logged in and directed to the profile page
Profile			
ID	*Test*	*Expected Result*	*Actual Result*
F1	The user clicks the 'new trip' button	The button should link to the new trip page	The button links to the new trip page
F2	The user clicks the settings button	The button should link to the settings page	The button links to the settings page
F3	The user clicks to view a trip	The button should link to the appropriate stored trip	The button links to the appropriate stored trip
F4	The user clicks the button to delete a trip	A popup should appear asking the user to confirm the delete	A popup appears asking the user to confirm if they want to delete the trip
F5	The user clicks 'yes' to delete the trip	The trip should be deleted and the page should refresh with a success message being displayed	The trip is deleted and the page refreshes

			with a success message
F6	The user clicks 'no' to delete the trip	The popup asking to confirm the delete should close	The popup asking to confirm the delete closes
New Trip			
ID	*Test*	*Expected Result*	*Actual Result*
G1	When the page loads the users location should be gathered	If the users location cannot be gathered then an error message should display, otherwise the users location should display on the map	If the users location is not reachable then an error message displays, otherwise the users location displays on the map
G2	The user slides the miles radius slider	The input box displaying miles should change along with the radius circle on the map	The miles change in the input box along with the radius circle on the map
G3	The user submits the form with invalid data	The user is redirected back to the form with error messages	The user is redirected back to the form with error messages
G4	The user submits the form correctly, but not enough locations are found	The user gets redirected back to the new trip page with an error message telling them to expand their search radius	The user is redirected back to the new trip form and an error message is displayed

G5	The user successfully submits the form	The locations page loads	The locations page loads
Locations			
ID	*Test*	*Expected Result*	*Actual Result*
H1	The locations page loads and displays each locations info	The locations page loads and displays each locations info, if the location does not have an available image then a map with its location will show instead	The locations page loads with each location listed, any locations without images have a map with the location of the location displayed
H2	The user clicks the 'next' button	The final map page loads	The final map page loads
Final Map			
ID	*Test*	*Expected Result*	*Actual Result*
I1	When the page loads the map should display	The map displays with the directions to each location on it and written instructions should appear in a panel to the left of the screen	The maps loads and displays the route between each location, text directions display in the left panel
I2	The user clicks the export button	The map with the same location loads into Google Maps external site	The map with the same locations is displayed on the external Google Maps site

I3	The user clicks the save button	A popup should appear asking the user to give the trip a name	A popup appears asking the user for a trip name
I4	The user does not enter a trip name but submits the form	Validation errors should appear almost instantly	Validation errors appear almost instantly
I5	The user successfully completes the save trip form and clicks save	The popup should close and a success message should be displayed	The popup closes and a success message is shown
Edit Profile Page			
ID	*Test*	*Expected Result*	*Actual Result*
J1	The user enters inappropriate data in the edit details form	The users should be redirected back to the form with errors	The user is redirected back to the form with errors
J2	The user enters appropriate data into the edit details form	The user should be redirected to the profile page with a success message	The user is redirected back to the profile page with a success message
J3	The user enters inappropriate data in the password change form	The users should be redirected back to the form with errors	The user is redirected back to the form with errors
J4	The user enters a wrong password into the current password field	The users should be redirected back to the form with errors	The user is redirected back to the form with errors

	The user enters appropriate data into the change password field	The user should be redirected to the profile page with a success message	The user is redirected back to the profile page with a success message
	The user submits an image that is not the right size or filetype in the change profile picture form	The users should be redirected back to the form with errors	The user is redirected back to the form with errors
	The user submits an image that is an appropriate filetype and size	The user profile picture should be resized, cropped and saved and they should be redirected back to the profile page with a success message	The users profile picture gets cropped, resized and saved and they are redirected back to the profile page with success message

9.4 - Appendix D

Timestamp	Approximately how many pages did you browse during this visit?	Did you create an account and create a trip?	How would you rate the visual appeal of the site?	Was the site easy to use?	If you created a trip what were the quality of the locations like?	If you were to review Matooi what score would you give it out of 10?	Did you encounter any errors/anything you would like to see changed?	Was it clear the service of which the site was offering?	What is one feature you would like to see added?
21/04/2015 21:33:23	2-4	I created an account and made a trip	Good	The site was mostly easy to use.	Good	7			
22/04/2015 05:30:13	5+	I created an account and made a trip	Very Good	The site was easy to use.	Very Good	8	No errors.	Yes	I would like if I could set a higher amount of locations and bigger radius to search
22/04/2015 15:32:51	5+	I created an account and made a trip	Very Good	The site was easy to use.	Fair	7		I had a rough idea	
22/04/2015 15:10:01	2-4	I created an account but did not make a trip	Very Good	The site was easy to use.	I did not make a trip	8		I had a rough idea	
24/04/2015 20:11:47	5+	I created an account and made a trip	Excellent	The site was easy to use.	Very Good	9		Yes	Possibly add the ability to repick 1 location if you dont like it.
24/04/2015 20:40:20	2-4	I did not create an account or make a trip	Very Good	The site was mostly easy to use.	I did not make a trip	6		I had a rough idea	
24/04/2015 21:03:32	5+	I created an account and made a trip	Excellent	The site was easy to use.	Very Good	8		Yes	
25/04/2015 12:30:52	1	I did not create an account or make a trip	Poor	The site was confusing	I did not make a trip	1		No, it was not clear what	

				and I could not use it.				service the site offered	
25/04/2015 13:57:02	2-4	I created an account and made a trip	Very Good	The site was mostly easy to use.	Good	7		Yes	
25/04/2015 14:02:16	5+	I created an account and made a trip	Very Good	The site was mostly easy to use.	Good	7	Show more pictures of the app on the homepage so I can get a better understanding of it before I make an account.	I had a rough idea	Allow me to share my trip that I created with my friends.
25/04/2015 14:27:10	5+	I created an account and made a trip	Very Good	The site was easy to use.	Poor	7	The locations that were chosen for me looked like places I would not like to visit.	Yes	Allow me to change each location where it revealed the locations to me. I mean add a way for me to rechoose a location if I dont like it.