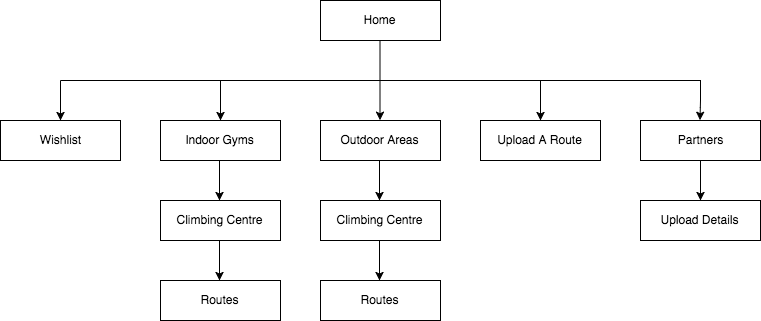
**Design**

There are multiple features within my app. Therefore, I am going to make it as obvious and simple as possible, allowing an ease of use for every user. I shall achieve this by making every feature have its own page. Separating the features will mean that each page will be different from one another, therefore allowing the user to be able to quickly differentiate between pages. As well as this, every page/feature will be clearly labelled, leaving no area for any confusion.

**Site Map**

This chart will present all my pages and what they link to.

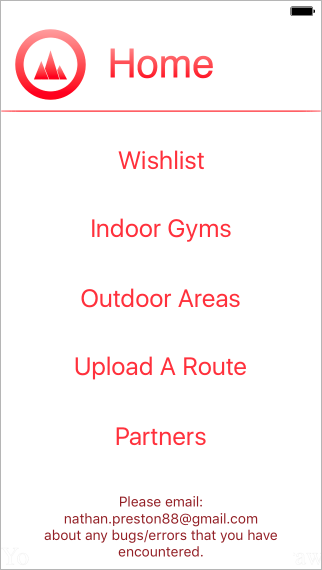


Here you can see that all my features have been separated into different pages, this means that nothing will be cluttered on my pages, allowing everything to be very obvious and easy to see. As well as that, being able to access anything from the home screen I feel is a necessity, as it means you can navigate multiple areas incredibly quickly without difficulty. The only examples where I have layered my pages is within my main features to avoid having too much happening on one page, such as within the routes within the climbing centre.

All these features I have mentioned, I believe, will make all of the UI much more user friendly and easy to use.

**Home Screen**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |

****

First of all, above the line are two very important features. Each page will contain a large title right at the top, to present to the user what page they are on. This helps them quickly recognise which page they’re on in case they forgot/clicked incorrectly. As well as that it just helps a user easily able to navigate quickly through my app. Secondly in the top left is my logo for the app. This logo represents the home page, so when the user sees this logo it is a visual way of seeing that they’re at the homepage.

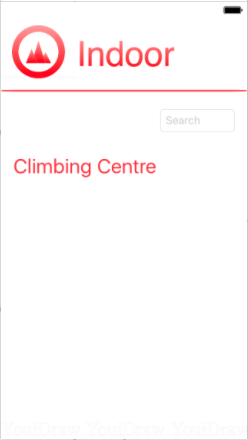
Below the line presents the 5 pages you can access via the home page. I decided to give a range of choices from the home page rather than having to go through multiple pages to get to the desired destination. Doing this I felt is a nicer and smoother method of navigating my app.

Also I have added some static text presenting my email, to act as an ‘Contact Me’. So any users will be able to report anything that has gone wrong or general information, such as what they would like to see within the app. I originally planned for this to be on its own page, however I felt that it was just an unnecessary use of a page when I can easily have it inputted to the homepage.

I’m using the font ‘Helvetica’, I chose it as it is a professional looking font, which also is very clear and basic. Further emphasising the point of simplicity within my app, nothing should be complex/confusing. Next, I used a fairly large font so all the page is being used up and all the options are very easy to see straight away. Also, I used the colour red as it creates a very strong contrast against the white background, which is again allowing the options to be easily read. Placing the text in the centre of the screen again means more space is being used up. Plus, having everything central is very aesthetically pleasing and makes my app look simple and minimalistic.

I have no images/silhouettes within my app, which is quite different to most climbing projects. Yet I feel it is easier for the eye to read/see if there is very little going on the screen. Following this idea, I chose to have a plain white background. Using white means anything I place will be obvious and wont blend in.

**Indoor Gym/Outdoor Areas**

****

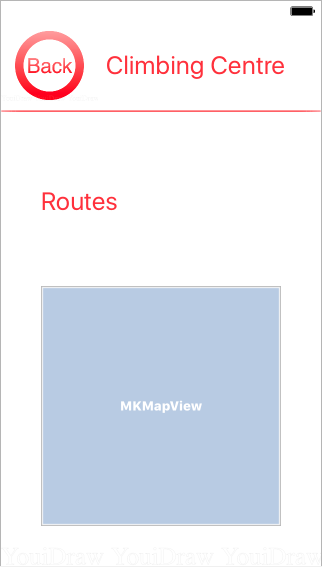
My Indoor Gym and Outdoor Area pages are identical apart from the title. I separated them as some people only go inside/outside. Therefore, it’s faster and more efficient to split them.

Here I have a logo similar to the one on the homepage, however the three mountains have been replaced with a home button. This button presents to the user that this user will take them back to the home. It is visually appealing and very simple compared to any other symbols. As well as that it is a generally used image for a home screen, making it more simple and easy to use for the user.

Within this page there will be a list of all the climbing centres that have been added. As there will be quite a large number of climbing centres, the user will be able to scroll down through them all. Having a scroll means that all the centres don’t have to be in a very small font and crammed into the small screen space. Yet, if the user doesn’t want to have to scroll until they find their climbing centre they can alternatively search with the bar in the top right. This means that finding the desired climbing centre can be very fast, but then at the same time the user can manually scroll and possibly find new climbing centres near them.

I have used the same font as the homepage, and I am going to keep it as a constant throughout the app, to make it look simple. As changing the font can be very off-putting even though it is very small, it can have a large impact on the aesthetic of the app.

**Climbing centre**

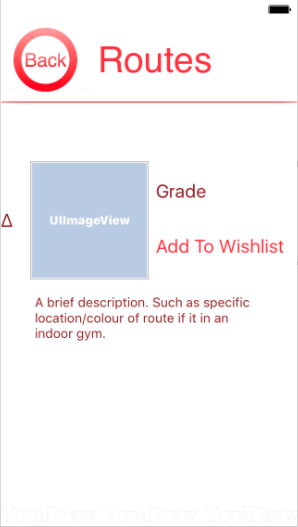


Firstly, within this page the logo has changed to the word back, rather than the house symbol. This is as the symbol only presents going back to the home page, as the page before this is either Indoor Gyms or Outdoor Areas the button will only take you back, not to the homepage.

From here the user is able to go to the routes page. Differing from home page I haven’t centred the text, this is as there may be extra information added about the climbing centre. So as there may be a little paragraph, the button would look wildly out of place if it were to be centred.

Also an interactive map will be added into this page once the location has been added. It will show where the climbing centre is and allow the user to be able to navigate where it is in comparison to other areas/where they are. It allows the user to smoothly navigate and makes it much nicer to be able to find the climbing centre, compared to a static map of the area.

**Routes**

Once fully setup this page can look very different. As users will be able to upload to this page, therefore some pages will have lots routes uploaded, then some may have only one. However, the pages will follow the exact same layout as what is shown: An image can be uploaded; a grade must be uploaded; the option to add this route to the Wishlist; an optional description. I have purposefully made the font of the description smaller than the rest. I did this as it isn’t as much as a priority as the other two entities. Plus, it allows more text within a smaller amount of space. The text box also has scroll bar within its self, so if a lengthy description was added it wouldn’t clutter the page.

In this page there is a change in colour of the font, it presents different functionalities. As, the dark red is static text: the grade, but the light red is a button: ‘Add to Wishlist’. The size of the writing is fairly small as the page should be very space friendly, as if there were lots of routes uploaded the scroll would take a long time to go across all the routes.

A scroll will be implemented, to allow the user to scroll through the numbers of routes. Yet the scroll won’t always be used, as said before, not all the climbing centres will have lots of routes.

**Wishlist**

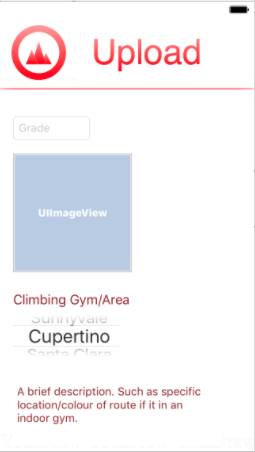
The Wishlist is very similar to the Routes area, as it will only contain a list of Routes. However, in this section the routes are only routes that someone has chosen to add. It also differs from the Routes section as it contains three pieces of static text: the grade, location and description. I felt having all the information possible was necessary, as having it all within the Wishlist allows fast access to the information.

As well as the static text I have added the ‘Remove’ button. I added this as a user may accidently add the route to their Wishlist and then want to remove it because they don’t want it there. Plus, once a user has completed the climb they may no longer see the need of having it upon their Wishlist.

I have also implemented a version of a bullet point here and within the Routes area. Doing this I feel will give the list of routes an order instead of just having a random cluster of information. It overall just aids the eye to identify that a new entity of the list has been added.

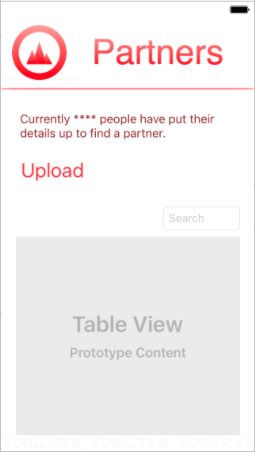
With similarity to the Routes page, there will be a scroll bar implemented to allow the user to add multiple routes and have no limitations.

**Upload A Route**

Within the ‘Upload A Route’ page there will be 4 areas possible for an input. Firstly, there is the grade, which is showing how hard the climb is. Secondly there is an upload for a photo of the climb its self to present where the climb is. Next, there will be a selection for the climbing area. I’ve made this a selection, rather than a typed input to the system, as it is an important field and I want no invalidity. Also having it as a selection can be easier for the user as then they don’t have to worry about spelling it correctly, as they can just scroll to their desired destination. Lastly, there is an area for a description identical to the text box in my Routes and Wishlist pages. As previously said, it has a scroll within its self therefore there is no need for a scroll bar on this page as everything fits perfectly.

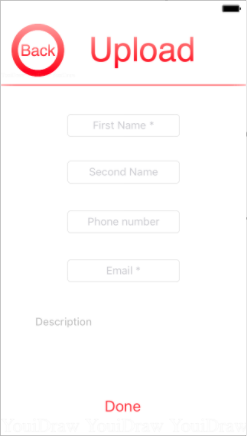
The layout may differ slightly for when it is all setup to follow the rules for the database. However, it will follow the general structure of what has been presented.

**Partners**

This page, once completed, will be one of the most complex. Firstly, it begins with a small static piece of text, which will contain the number people who have uploaded their details. Below, this is a link to the Upload Details page. I felt it was necessary to access the Upload within this page, as you will only ever upload your details once. Whereas, a user may upload numerous routes, hence why that is accessible from the home page. This is6 also why the Delete Details is on here as well, as it wouldn’t follow the continuity if the delete were on the homepage. Next, is there area that will present all the users that have uploaded their details to my app. As there could possibly be numerous people uploading their details I will add to features to make it more user friendly. Firstly: a search bar – having a search bar means that a user can find someone else very quickly without having to go through many other users before finding the specific user. Secondly: A scroll bar – having a scroll bar means that all the details won’t be cramped within this page and the font won’t have to be minimized to an unreadable size.

I have made the Upload button have the same font size as all the other buttons within this app. Doing will keep the continuity of my app, as well as that it means a user will quickly get used to what an interactive button looks like, rather than the user being confused on how to navigate. Then I have made the search bar and static text, the same size to make the writing look professional and it makes the whole page much more aesthetically pleasing.

**Upload Details**



On the last page of my app there are 5 inputs, 2 of which are have put a star on. These 2 stars present that these fields must have an input. I felt that these two inputs are vital for any communication, then the rest is down to the user if they wish to include that information. Therefore, I shall add some validation to check that both of these fields have been entered before any of it is uploaded to the database. The first four inputs are exactly the same, then the last input is a large text box – identical to the one used in: routes, Wishlist and the Home Screen. So, as previously said, the text box has a scroll bar built in so lots can be inputted and none of it will escape the view of the user.

../../../../../../../../Desktop/Screen%20Shot%202017-08-10%2**Stage 1 – Basic Layout**

Here I will be separating each part of my app into a new page, and putting buttons to allow the user to go through each page/feature. There will be almost no inputted code here, as with Xcode most simple visual features can be dragged and dropped. Doing this means that each of my program can be broken down and each feature. Achieving each feature being a separate page will allow the debugging and my testing be much smoother and easier. As well as that it means that if one feature doesn’t work, it shouldn’t have a knock on effect to my other pages/features. Allowing me to be able to focus at one part at a time.

For feedback at this stage I will go to a local climbing centre and present this prototype to multiple climbers and note down their feedback. From this I can see if the layout is nice, it has an ease of use and if anything needs to be added or removed.

Added to the project:

* A number of basic pages
* Buttons to navigate between them all
* Scroll bars

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 1a | Checking whether the app will open without it crashing/without error | -Icon Click  -Valid | App Opens without error |  |
| 1b | Checking button for Indoor Gym works without error | -Button Click  -Valid | Indoor Gym Section opens |  |
| 1c | Checking button for Outdoor Area works without error | -Button Click  -Valid | Outdoor Area Section opens |  |
| 1d | Checking button for Wishlist works without error | -Button Click  -Valid | Wishlist Section opens |  |
| 1e | Checking button for Upload A Route works without error | -Button Click  -Valid | Upload A Route Section opens |  |
| 1f | Checking button for Partners works without error | -Button Click  -Valid | Partners Section opens |  |
| 1g | Checking button for Upload Details works without error | -Button Click  -Valid | Upload Details  Section opens |  |
| 1h | Checking button for home works without error | -Button Click  -Valid | Home opens |  |
| 1i | Scrolling up and down the pages | -Scroll  -Valid | Page scrolls |  |

**Stage 2 – Inputting Every Climbing Area/Gym Nearby**

I’m manually adding all the climbing areas and gyms myself and won’t allow any to be added/removed. Adding this myself will allow less area for invalid inputs. As well as this I will add a search bar to allow the user to search for a specific place.

Added to my project:

* Climbing areas
* Search bar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 2a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 2b | Searching for an invalid place | -‘John’  -Valid | Nothing is shown. |  |

**Stage 3 – Interactive Map**

Once the user has chosen a specific place they want to look at, an interactive map will appear. Here the user can scroll and zoom, whilst having a maker upon the climbing gym/area.

When at my local climbing centre, I will present this interactive map and see if my possible clients find it useful. As well as this, I can get any feedback on any changes that could be made, such as the size and usability of the map.

Added to my project:

* Interactive map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 3a | Scrolling Around | -Map Scroll  -Valid | Scrolls without error |  |
| 3b | Zooming In/Out | -Map Zoom  -Valid | Scrolls without error |  |

**Stage 4 – Setting Up the Database**

Here I will set up the database which will have three tables; one for routes, one for the Partners section and one for the Wishlist. Once it has been created it will need to be linked to the app, so anyone using the app will be able to access this database, through the specific section of the app.

Added to my project:

* Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 4a | Setup sections for the Routes area. So 3 valid inputs. | - ‘Red’, ‘X’, ‘6a’, ‘Boulder Brighton’  -Valid | All data inputted without error |  |
| 4b | Outputting data | - ‘Red’, ‘X’, ‘6a’, ‘Boulder Brighton’  -Valid | Inputted data will be outputted |  |
| 4c | Setup image upload | -simple image  -Valid | Possible error, but image should be inputted to database |  |
| 4d | Outputting data with image | -‘Red’, Image, ‘6a’, ‘Boulder Brighton’  -Valid | Output all data that was inputted |  |

Database Examples

Here I will show the 3 tables I will have within database:

**Table One: Route Upload - Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Validation** |
| Primary Key | Integer | A unique number for each record | Must be unique |
| Picture | URL/Reference | A reference to the picture |  |
| Grade | String | The grade of the climbing route |  |
| Description | String | A brief piece of text describing the climb |  |

**Table One: Route Upload – Example**

|  |  |
| --- | --- |
| **Primary Key** | 1 |
| **Picture** | N/A |
| **Grade** | ‘6a+’ |
| **Description** | ‘Very bad holds’ |

**Table Two: Partners Details - Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Validation** |
| Primary Key | Integer | A unique number for each record | Must be unique |
| Fname | String | First name | Must be a string |
| Sname | String | Second name | Must be a string |
| phoneNum | String | Phone number | Must be only numbers |
| Email | String | Email | Must include an ‘@’ |
| desc | String | Brief description of user | No validation |

**Table Two: Partners Details - Example**

|  |  |
| --- | --- |
| **Primary Key** | 1 |
| **Fname** | ‘Nathan’ |
| **Sname** | ‘Preston’ |
| **phoneNum** | ‘07401588221’ |
| **Email** | ‘nathanpreston@aol.com’ |
| **desc** | ‘Im 17 and enjoy bouldering’ |

**Table Three: Wishlist - Data**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Description** | **Validation** |
| Primary Key | Integer | A unique number for each record | Must be unique |
| Picture | URL/Reference | A reference to the picture |  |
| Grade | String | The grade of the climbing route |  |
| Description | String | A brief piece of text describing the climb |  |

**Table Three: Wishlist - Example**

|  |  |
| --- | --- |
| **Primary Key** | 1 |
| **Picture** | N/A |
| **Grade** | ‘6a+’ |
| **Description** | ‘Very bad holds’ |

**Section 5 – Validation**

Within this section I will implement the only validation that there is within my project, and that lies in the Upload Your Details section. Here my app will be making sure that the user has entered their first name and added either their email address or their phone number. I am only validating these parts as I believe they’re the necessary pieces of information to contact someone.

Here will ask my clients whether the required information is enough or if it should be more or less. Therefore, I can see what the majority like and change my project to adapt to the need of the users.

Added to my project:

* Validating whether there's an input

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 5a | Inputting no data to see if the validation has worked | -Invalid | Rejection of data. |  |
| 5b | Inputting data into First Name, but not into Email or Phone Number. | -‘Nathan’  -Invalid | Rejection of data. |  |
| 5c | Inputting data into Email, but not into First Name or Phone Number. | -‘np@aol.com’  -Invalid | Rejection of data. |  |
| 5d | Inputting data into Phone Number, but not into First Name or Email. | -‘12345678901’  -Invalid | Rejection of data. |  |
| 5e | Inputting data into First Name and Email, but not into Phone Number. | -‘Nathan’, ‘np@aol.com’  -Valid | Acceptance of data. |  |
| 5f | Inputting data into First Name and Phone Number, but not into Email. | -‘Nathan’, ‘12345678901’  -Valid | Acceptance of data. |  |
| 5g | Inputting data into First Name and Email and Phone Number. | -‘Nathan’, ‘np@aol.com’, ‘12345678901’  -Valid | Acceptance of data. |  |

Data Dictionary

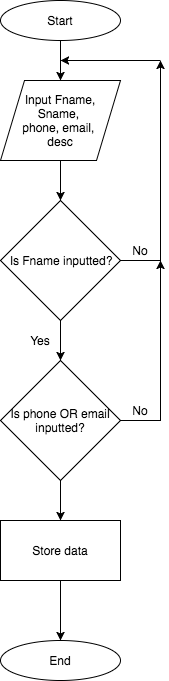
Here are all the key variables within this stage and will be re-used for every user as if the data is all valid the variables will be inputted to an array, therefore leaving the variables okay to be re-used.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** | **Validation** | **Example Data** |
| Fname | String | First name | Must be a string | ‘Nathan’ |
| Sname | String | Second name | Must be a string | ‘Preston’ |
| phoneNum | String | Phone number | Must be only numbers | ‘01825722323’ |
| Email | String | Email | Must include an ‘@’ | ‘nathanpreston@aol.com’ |
| desc | String | Brief description of user | No validation | ‘I climb walls’ |

Flowchart

Here is a simple flowchart of how my validation will work. It shows how if any of the data is incorrect the user will need to input all the data again. This is as it will be all on one page, so clicking the button to input the data will be made to input all the data entered.

Also as previously said the main validation, if the data is correct, is to see if a first name has been entered and a phone number or an email. If these rules are met, then the variables will be added to an array/database for the user.



Pseudo Code:

loop = 0 //Simple variable for a while loop to end

users[100][5]=0 //array for all users

WHILE loop == 0:

Fname = ‘’, Sname = ‘’, phoneNum = ‘’, email = ‘’, desc = ‘’ //initialising the variables

INPUT Fname, Sname, phoneNum, Email, desc //inputting data

IF len(Fname) > 0:

IF len(phoneNum)>0 OR (len(Email)>0 AND check(Email)) ==1: //both if statements checking if something has been inputted and if the email contains an ‘@’ symbol

users[0][0] = Fname

users[0][1] = Sname

users[0][2] = phoneNum

users[0][3] = Email

users[0][4] = desc //adding data to the array

loop = 1 //ending loop

END IF

END IF

END WHILE

FUNCTION check:

x = 1, yes = 0

WHILE x <= len(Email): //this function is what was called earlier and loops through

IF Email[x] == ‘@’: the string and searches for an ‘@’. If found 1 is outputted,

yes = 1 otherwise it’s 0

x = len(Email)+1

END IF

END WHILE

END FUNCTION

**Section 6 – Linking The Database to the App**

Here, once the data base has been setup, it will be linked to the app. So once all finished climbing routes will be able to be uploaded and then viewed by others. Therefore, what I shall be adding is the database to the app, allowing the partners and routes sections to have saved inputs. Therefore, the pages for uploading routes and details must also be setup. Next, there can’t be any invalid data that can be stopped as in the routes; the image is trusted to be relevant; there are multiple grading systems so the user can input anything; the place that the route is will be chosen through a specific drop down list.

Here I will try to get feedback on whether my clients like the way they’d be able to upload details/routes. As this is a very important section for my app I will go to Withdean Climbing centre and Boulder Brighton to get a range of opinions.

Added to my project:

* Linked Database
* Add an Upload a Route Section
* Add an Upload Details Section

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 6a | Input data to the database through the app | - ‘Red’, Image, ‘6a’, ‘Boulder Brighton’  -Valid | All data inputted without error |  |
| 6b | View Inputted data within the routes section | - ‘Red’, Image, ‘6a’, ‘Boulder Brighton’  -Valid | All data outputted without error |  |
| 6c | Input Details | - ‘Nathan’, ‘01234567890’, ‘n@aol.com’  -Valid | All data inputted without error |  |
| 6d | View Inputted data within the partners section | - ‘Nathan’, ‘01234567890’, ‘n@aol.com’  -Valid | All data outputted without error |  |
| 6e | Search for a specific partner | - ‘Nathan’,  -Valid | Inputted data is found |  |
| 6f | Output a counter of the number of people who have added their details | - add 3 sets of ‘details’  -Valid | Counter is at 3 |  |
| 6g | Make sure data that data is added to: ‘name’ with ‘phone number’ or ‘email address’ | - no data input  -Invalid | Make user input data |  |