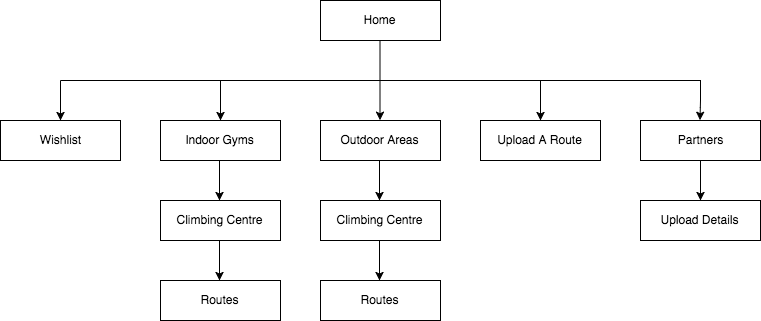
**Design**

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**

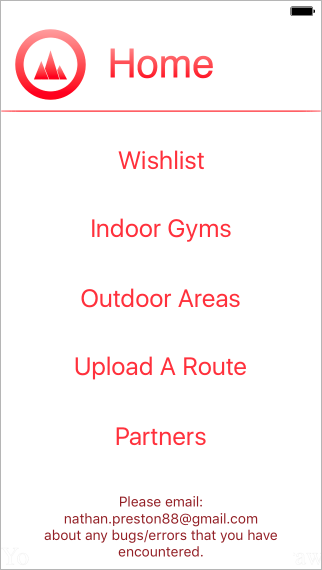
This chart will present all my pages and what they link to.



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**

****

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.

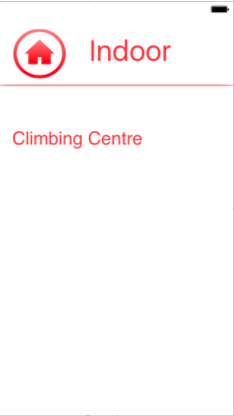
Below the line presents the 5 pages you can access via the home page. I decided to give a range of choices from the home page rather than having to go through multiple pages to get to the desired destination. Doing this I felt is a nicer and smoother method of navigating my app.

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**

****

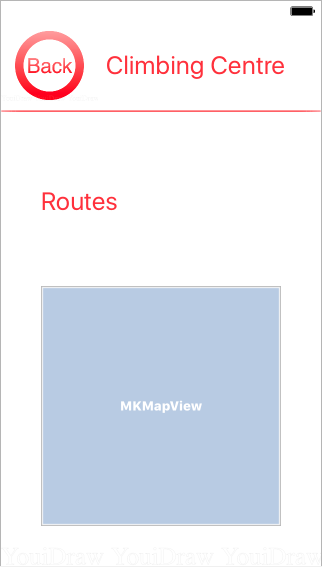
My Indoor Gym and Outdoor Area pages are identical apart from the title. I separated them as some people only go inside/outside. Therefore, it’s faster and more efficient to split them.

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**

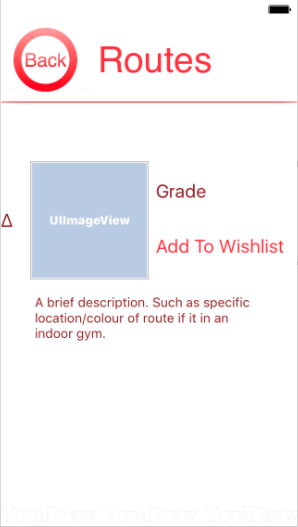


Firstly, within this page the logo has changed to the word back, rather than the house symbol. This is as the symbol only presents going back to the home page, as the page before this is either Indoor Gyms or Outdoor Areas the button will only take you back, not to the homepage.

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.

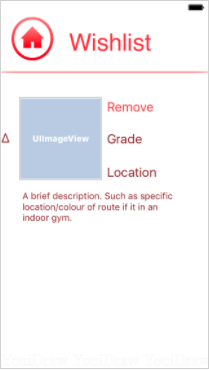
**Routes**

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**

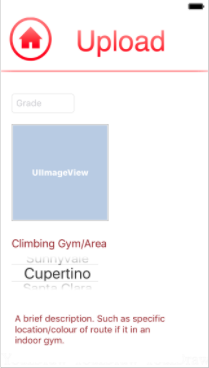
The Wishlist is very similar to the Routes area, as it will only contain a list of Routes. However, in this section the routes are only routes that someone has chosen to add. It also differs from the Routes section as it contains three pieces of static text: the grade, location and description. I felt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

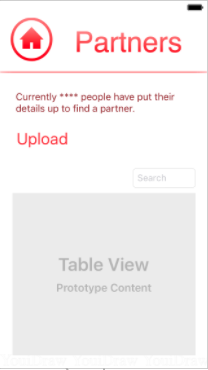
With similarity to the Routes page, there will be a scroll bar implemented to allow the user to add multiple routes and have no limitations.

**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.

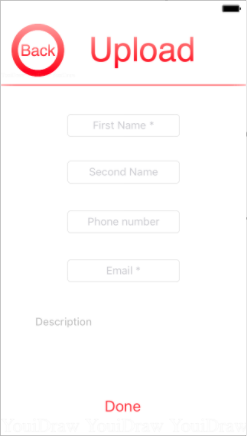
The layout may differ slightly for when it is all setup to follow the rules for the database. However, it will follow the general structure of what has been presented.

**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 is 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.

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 – 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.

Added to my project:

* Interactive map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 2a | Scrolling Around | -Map Scroll  -Valid | Scrolls without error |  |
| 2b | Zooming In/Out | -Map Zoom  -Valid | Scrolls without error |  |
| 2c | Map starts at the climbing centre | -Address  -Valid | Placement is correct |  |
| 2d | Map starts at a different climbing centre | -Address  -Valid | Placement is correct |  |

The code for this part is very simple and they’re won’t be any key variables, only one which stores the location of the climbing centre. All the code in this part will be adding the library to the code for map configuration; creating a reference to the map within the code; inputting the correct location.

**Stage 3 – Inputting Climbing Areas**

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** |
| 3a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place |  |
| 3b | Searching for an invalid place | -‘John’  -Valid | Nothing is shown. |  |

**Indoor**

Doing this I will not be inputting every climbing centre in Britain, instead I shall be inputting 80 climbing centres. I decided the best way to distribute the climbs would be by looking at the BMC’s release of every climbing centre in the UK. Then from this adding a proportional amount of climbs depending on the area. Such as 38% of the booklet is taken up by climbing centres within the South East. There for 38% of 80 is 30, so I shall be entering 30 climbs within for the South East. Then from this I will pick the seemingly most popular climbing centres within the areas of the Britain, whilst trying to achieve a spread of centres so everyone should be near to one which has been added. (BMC Booklet attached)

I felt this was the fairest way as it would take up a lot of time to have to input over 300 climbing centres into my app. I felt this way because adding more centres is very simple and one centre doesn’t take long to input. Therefore, I shall begin with 80 climbing centres and then proceed with my other stages of my app, to allow my app to be fully functioning. Then once complete I can continue to add more climbing centres, if there is enough time.

**Climbing Gyms/Centres:**

South East (38% - 30 Centres): Big Rock Climbing Centre, Biscuit Factory Climbing, Boulder Brighton, Boulder Shack, Brixton Recreation Centre, Chimera Climbing Centre, Clip N Climb Chelsea, Craggy Island Bouldering, Craggy Island Guildford, Extreme Ventures Brighton, High Sports Alton, High Sports Brighton, High Sports Crawley, JAGS Sports Club, Mile End Climbing Wall, Reading Climbing Centre, Red Spider, Southampton Climbing Centre, Swiss Cottage Climbing Centre, The Arch Climbing Wall, The Castle Climbing Centre, The Climb, The Climbing Hangar London, The Reach, Vauxhall Climbing Centre, West 1 Climbing Wall, Westway Climbing, White Spider, Wymondham Leisure Centre, XC

South West (11% - 9 Centres): High Sports Plymouth, Oxley Sports Centre, Redpoint Bristol Climbing Centre, Rockstar Climbing Centre, The Barn Climbing Centre, The Boulder Bunker, The Project Climbing Centre, The Undercover Rock, The Warehouse Climbing

Midlands (9% - 7 Centres): Awesome Walls Stoke, Bear Rock Climbing, Boulder Central, Creation Climbing Centre, Red Point Birmingham, The Pinnacle Climbing Centre, The Tower Climbing Centre

Peak District (9% - 7 Centres): Awesome Walls Sheffield, Nottingham Climbing Centre, Showroom Climbing Centre, The Climbing Unit, The Depot Nottingham Climbing Centre, The Foundry, The Healthy Living Centre

North West (9% - 7 Centres): Awesome Walls Stockport, Manchester Climbing Centre, Rock Over Climbing, The Climbing Hangar Liverpool, The Depot Manchester, The North West Face Climbing Centre, West View Leisure Centre

Yorkshire (3% - 2 Centres):Huddersfield Climbing Centre, The Leeds Wall

Lake District (3% - 2 Centres): KONG Adventure, Lakeland Climbing Centre

North East (3% - 2 Centres): Rock Antics Middlesbrough, Sunderland Wall

Wales (9% - 7 Centres): Beacon Climbing Centre, Boulders UK, Dynamic Rock Swansea, Harlech Climbing Wall, Rock UK Summit Centre, Spot Climbing Centre, The Indy Climbing Wall

Scotland (9% - 7 Centres): Avertical World, Edinburgh International Climbing Arena, Glasgow Climbing Centre, RGU: Sport, The Ice Factor, The Peak, Transition Extreme Sports

**Outdoor**

Outdoor climbing areas are fairly different to indoor gyms/centres as firstly, the routes generally will always be constant and will not be taken down, like an indoor route. As well as that, these climbing areas are spread very differently to indoor gyms. Such as indoor gyms are placed but outdoor areas are found, and generally lots of outdoor areas are in the Peak District, Wales and Scotland. So I will not be able to evenly distribute the routes over Britain. However, it is rare for only one route to be by its self, normally there are a cluster of climbs within an area. Therefore, to make my range as useful as possible I will look at the top 100 climbing routes from UKClimbing.com, then I will look where the climb is from and take that as an input. I shall repeat this for the top 50 climbs. I have chosen less here compared to the indoor centres as many more climbs can be within one area compared to an indoor gym.

**Outdoor Climbing Areas:**

Almscliff, Anglezarke Quarry, Back Bowden Doors, Beinn Trilleachan, Binnein Shuas crag, Black Rocks, Brimham Rocks, Buachaille Etive Mor, Carnmore Crag, Castell y Gwynt, Chee Dale, Clogwyn Du'r Arddu, Clogwyn yr Eryr, Craig Bwlch y Moch, Craig Pant Ifan, Craig y Llam", Creag Dubh Dibadale, Cyrn Las, Dinas Cromlech, Dove Crag, Dovestones Edge, Dunkeld, Eagle Crag, Froggatt Edge, Garbh Bheinn, Gogarth Stack, Kilnsey, Lundy, Malham Cove, Millstone Edge, Old Man of Hoy, Old Man of Stoer, Pavey Ark, Rainbow Slab Area, Raven Tor, Roaches Lower Tier, Scafell East Buttress, Shelterstone Crag, Shining Clough Rocks, Sron Ulladale, Sron na Ciche, Stanage Plantation, The Diamond, The Dovestones Quarries, The Napes, Twll Mawr, Uig Sea Cliffs, Whitestone Cliffe, Wimberry Rocks, Yesnaby

**Stage 4 – Partners Section**

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. Plus, I’ll make sure certain data can’t be re-inputted, as a phone number or email address is unique. Also I will create a table where all the details will be outputted to. Yet an arrays made at this stage may be just purely for testing purposes, as the details will eventually be uploaded to an array for many to access.

Added to my project:

* Creating a table
* Adding necessary features for an upload section
* Validating whether there's an input

x

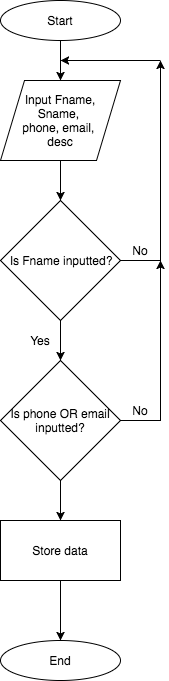
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 4a | Inputting no data to see if the validation has worked | -Invalid | Rejection of data. |  |
| 4b | Inputting data into First Name, but not into Email or Phone Number. | -‘Nathan’  -Invalid | Rejection of data. |  |
| 4c | Inputting data into Email, but not into First Name or Phone Number. | -‘test1@aol.com’  -Invalid | Rejection of data. |  |
| 4d | Inputting data into Phone Number, but not into First Name or Email. | -‘12345678901’  -Invalid | Rejection of data. |  |
| 4e | Inputting data into First Name and Email, but not into Phone Number. | -‘Nathan’, ‘tesr2@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. |  |
| 4g | Inputting data into First Name and Email and Phone Number. | -‘Nathan’, ‘test3@aol.com, ‘987654321’  -Valid | Acceptance of data. |  |
| 4h | Inputting data that has already been used. | -‘Nathan’, ‘test3@aol.com, ‘987654321’  -Invalid | Rejection of data. |  |
| 4i | Inputting unique data from each section | -‘John’, ‘John@aol.com’, ‘1111111111’  ‘I like climbing’  -valid | Rejection 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

i=0, y=0, looper = 0

WHILE(looper<len(users[0][2]))

IF phoneNum == users[looper][2]

y=1 //this loops and checks if the phone number is already used

END IF

END WHILE

Looper = 0

WHILE(looper<len(users[0][3]))

IF phoneNum == users[looper][3] //this loops and checks if the email is already used

i=1

END IF

END WHILE

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

IF (i==0) AND (y==0):

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 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

break

END IF

END WHILE

Return yes

END FUNCTION

**Stage 5 – Setting Up the Database for Partners Section**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 5a | Setup sections for the partner’s area. So 5 valid inputs. | - ‘Nathan’, ‘Pres’, ‘123’, ‘n@aol.cm’, ‘climbing’  -Valid | All data inputted without error |  |
| 5b | Outputting data | - ‘Nathan’, ‘Pres’, ‘123’, ‘n@aol.cm’, ‘climbing’  -Valid | Inputted data will be outputted |  |

Database Examples

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

**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 |

**Partners Details - Example**

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

**Stage 6 – Uploading Details**

Here Implement my stage 4 to be able to work with the database. As my stage 4 only worked for local databases so it needs to work with my database.

Added to my project:

* Database upload

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 6a | Input Valid Set of Data | -Valid | Data Is Uploaded |  |
| 6b | Input repeated data | -Invalid | Data isn’t uploaded |  |

**Stage 7 – Deleting Details**

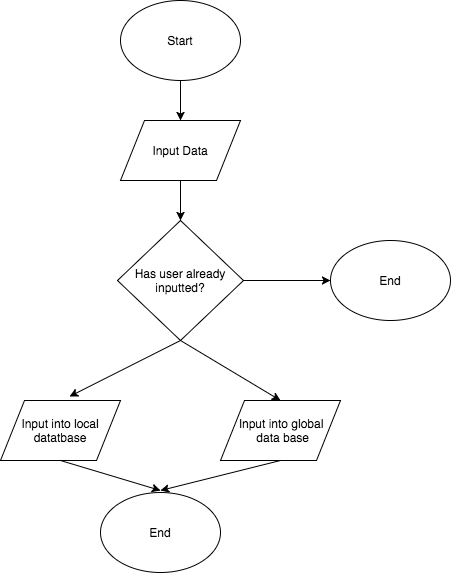
Here I will make it so one user can only ever upload one set of details and also allow the user to delete this. I haven’t added the validation about only allowing one person to set up one set of details for an ease of testing.

Added to my project:

* Validation
* Deleting Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 7a | Trying to input two sets of different data | ‘N’, ‘123’,  ‘J’, ‘456’  -Invalid | Rejection of second piece of data. |  |
| 7b | Deleting data | -valid | Data deleted |  |
| 7c | Re-inputting data after deletion | ‘N’, ‘123’,  -Valid | Data is inputted. |  |

To achieve this, I will have make the input save locally and globally. Therefore, when it validates the users input, it will check if they have anything previously saved. If so, they won’t be allowed to add an input. Also when they wish to delete their details I will search the global data base to find the data so it can be deleted.

****

**Stage 8 – Database for Routes**

Here, I will repeat stage 5 but for the routes within both outdoor and indoor climbing centres. I will create one database for inside, and one for outside. Yet I will have to make it to be able to support an image upload. So it will be a bit harder to do, as images in databases aren’t very simple.

Added to my project:

* Database for inside and outdoor routes
* Upload a Route Section

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 8a | Add data into database without picture | - ‘6c’, ‘Boulder Brighton’, ‘Big heel hook’  -‘6a’, ‘Boulder Brighton’, ‘Cave’  -‘6a’, ‘White Spider’, Fun…’  -Valid | All data inputted without error |  |
| 8b | Data outputted correctly | - ‘6c’, ‘Boulder Brighton’, ‘Big heel hook’  -‘6a’, ‘Boulder Brighton’, ‘Cave’  -‘6a’, ‘White Spider’, Fun…’  -Valid | All data outputted without error |  |
| 8e | Add data into database with picture | -‘6a’, ‘Boulder Brighton’, ‘big dino’, image  -Valid | All data inputted without error |  |
| 8d | Data outputted correctly | -‘6a’, ‘Boulder Brighton’, ‘big dino’, image | All data outputted without error |  |
| 8c | Input through app without picture | -‘6a’, ‘Boulder Brighton’, ‘big dino’, image  -Valid | Data is inputted without error |  |
| 8f | Data outputted correctly | -‘6a’, ‘Boulder Brighton’, ‘big dino’, image  -Valid | All data outputted without error |  |
| 8g | Input through app with picture, see if it is outputted | -‘6a’, ‘Boulder Brighton’, ‘big dino’, image  -Valid | Data is inputted and outputted correctly |  |

**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’ |

**Stage 9 – Adding/Removing to and from Wishlist**

Once the data base has been fully setup I will be creating the last part. This will include adding routes from climbing centres into the wishlist. The options will be within the ‘Routes’ section within each climbing centre page, it will have a button to allow the user to add that route to their wishlist. Once this has been pressed it will appear at the bottom of their list of climbs that they are interested in. then when the user has completed the climb or is just no longer interested in it, they have the option to remove it from their wishlist.

Added to my project:

* Useable wishlist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 9a | Adding a route to my wishlist | - ‘Red’, Image, ‘6a’, ‘Boulder Brighton’  -Valid | Route appears in wishlist. |  |
| 9b | Adding multiple routes to my wishlist | - ‘Red’, Image, ‘6a’, ‘Boulder Brighton’  -Valid | Routes appears in wishlist, in correct order. |  |
| 9c | Remove all routes from wishlist | - Remove function  -Valid | Route is removed. |  |

**Table Two: 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 |  |
| Location | String | The location of the climbing area |  |

**Table Two: Wishlist - Example**

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