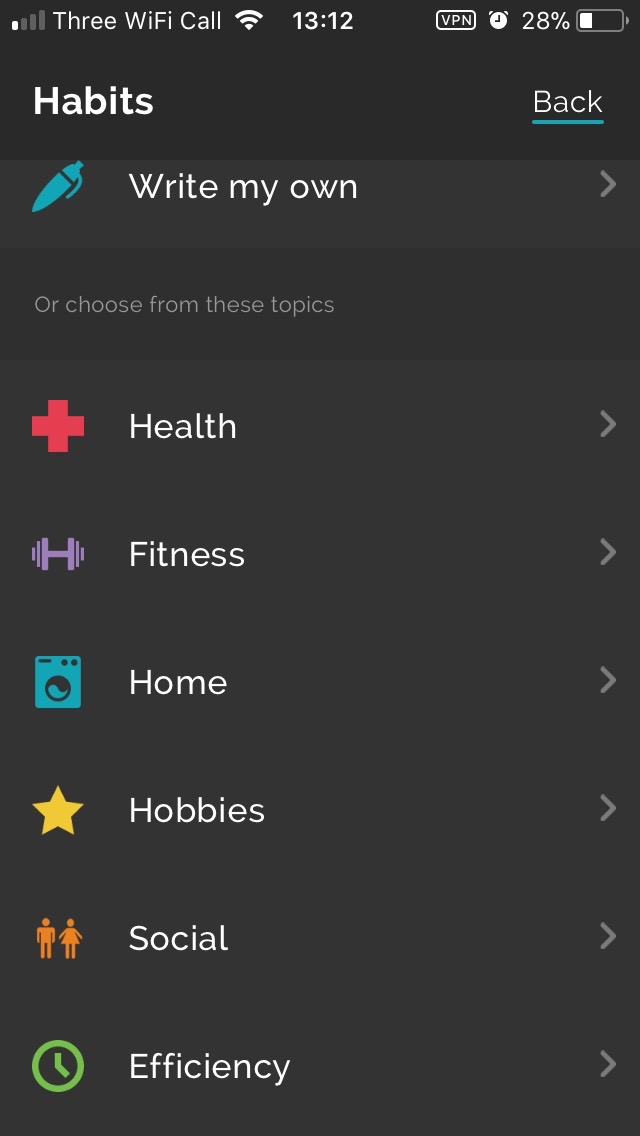
**Implementation**

All attachment videos will be named equivalent to the test

**Stage 1 – Basic Layout**

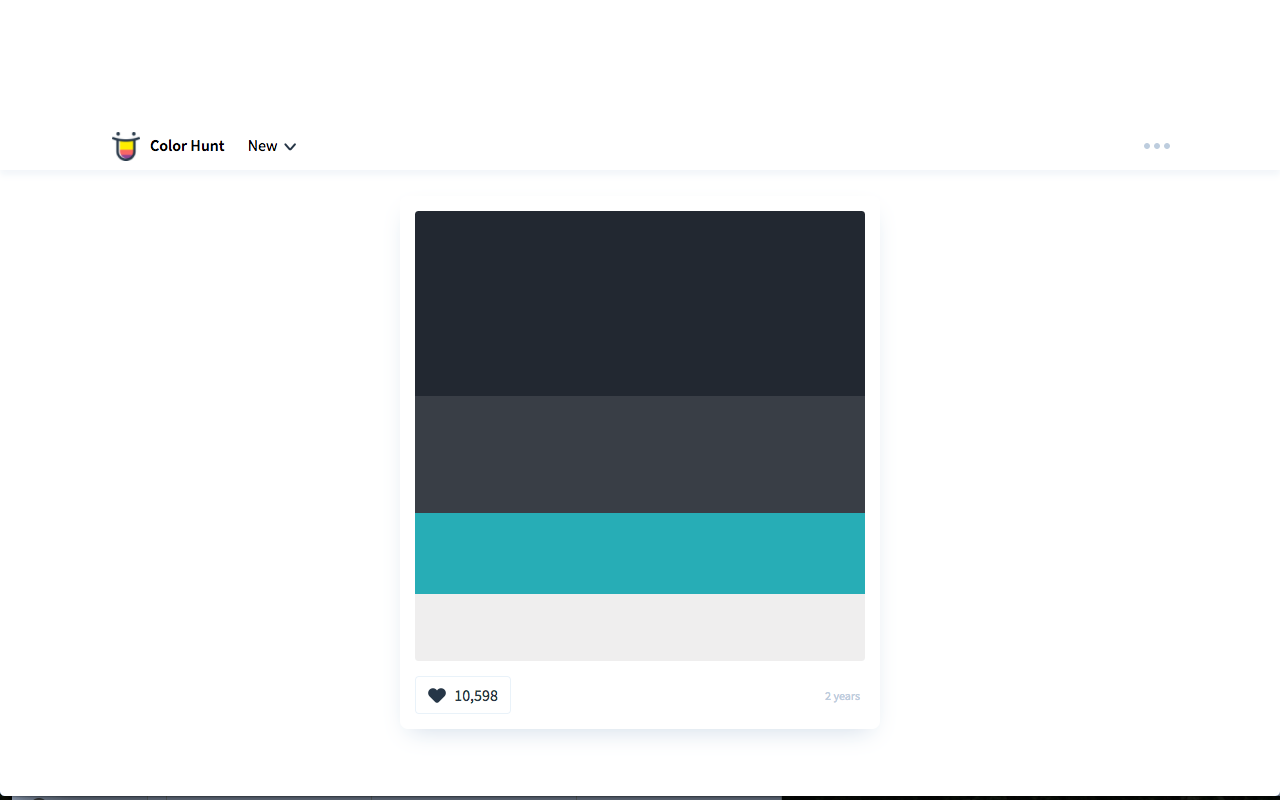
Within this first stage when developing my app, I had done a large recreation of my UI. This is due to when I had gone for user feedback within my focus climbing group the general output was that the view was not very nice. Many agreed that the strong red vs white contrast wasn’t very visually appealing. Yet, they had no bad comments about where everything is positioned and how it is laid out. The only issue was the colour. As well as this when receiving my feedback and reviewing my design and analysis I had realised I hadn’t looked up and researched any good layouts. As, my main issue, with all the other similar projects that I had researched, was that the layout wasn’t very appealing. Therefore, from this I did a large recreation of the colour scheme.

Within my focus group, one person had recommended the colours for the app:

****

**Productive – Habit Tracker**

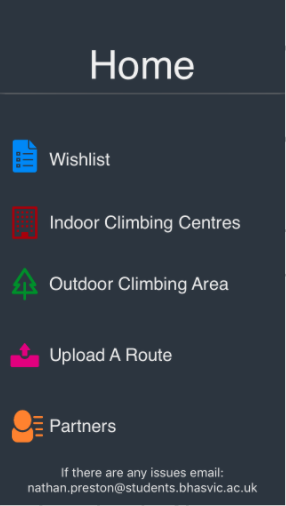
Once seeing this I saw that the colour scheme was very nice, minimal and sleek. As well as this they use a lot of dark colours, then will have multiple bright colours to create a strong contrast on the background.

****

Therefore, from here I realised that using dark colours look much nicer and much more slick. So I went onto the website: ColorHunt. Here I found this colour scheme. From this I decided that the first colour will be for the background. Second for lines/separations and details. Third, for buttons. And last for text.

The following screenshots shows my final UI.

**Home**

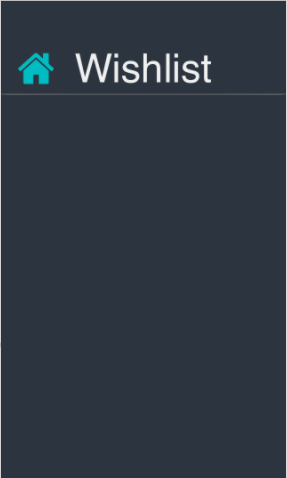
****Here you can see that I have used much calmer colours for the text and background, and used more colours in total. Firstly, I have used calmer colours as it is much more straining on the eyes. Secondly, using a light grey for the text, dark grey for the separator and a navy background presents a stronger differentiation between it all which is still very aesthetically pleasing.

Next I have used a new colour for each part as it shows a very clear differentiation for each section of my app. Doing this can make a mental link with the colour to the section. As well as this I have made an icon for each section, as I felt it was much more aesthetically pleasing and once again created an obvious differentiation between each section.

However, I have still used the same structure as I had planned.

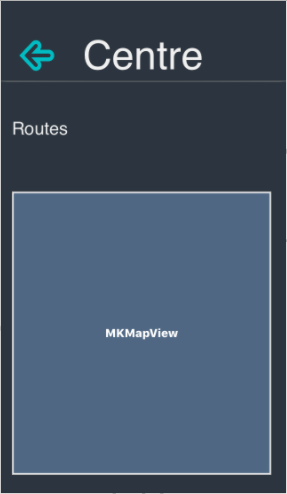
**Wishlist**

Here for the all the back buttons and the home buttons I have used a light blue to make a very clear option to go back. Yet it isn’t an eye saw. Therefore, it makes the new layout much easier to navigate and much nicer to look at.

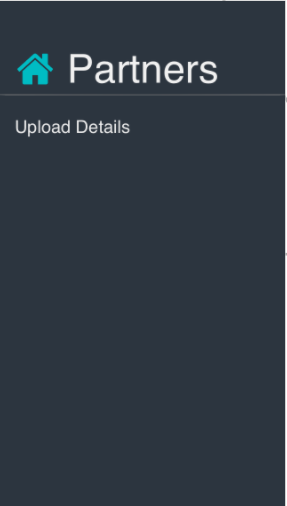
Similarly, to my old layout I have made a new icon for going back a page if the page before isn’t the home page, I have used a basic backwards arrow. I used this instead of my original idea as its much nicer too look at, plus as my new UI is very heavy upon icons I thought it would be satisfying with for the user if the pattern was followed.

**Indoor Outdoor**



**Climbing Centres Routes**

**Partners**



**Upload Details Upload a Route**



**Test 1 – Video 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 | Runs without error |
| 1b | Checking button for Indoor Gym works without error | -Button Click  -Valid | Indoor Gym Section opens | Runs without error |
| 1c | Checking button for Outdoor Area works without error | -Button Click  -Valid | Outdoor Area Section opens | Runs without error |
| 1d | Checking button for Wishlist works without error | -Button Click  -Valid | Wishlist Section opens | Runs without error |
| 1e | Checking button for Upload A Route works without error | -Button Click  -Valid | Upload A Route Section opens | Runs without error |
| 1f | Checking button for Partners works without error | -Button Click  -Valid | Partners Section opens | Runs without error |
| 1g | Checking button for Upload Details works without error | -Button Click  -Valid | Upload Details  Section opens | Runs without error |
| 1h | Checking button for home works without error | -Button Click  -Valid | Home opens | Runs without error |
| 1i | Scrolling up and down the pages | -Scroll  -Valid | Page scrolls | Runs without error |

**User Feedback**

I asked a focus group what they liked and disliked about my UI. The general feedback was that it was a very nice layout, but the general dislike was that the area to press an icon was too small. So to fix this I will make the area to be clicked larger, via creating a transparent space that you’ll be able to click.

**Stage Two – Interactive Map**

When implementing my map, I had encountered two very large errors, however they were all due to similar reasons. The main cause was the way I was going to have Stage 3 and Stage 2 work together.

As when I began this stage I knew that I would have to start a little bit of stage 3, as I need to go through a button to a climbing centre to get to the interactive map. Therefore, at the beginning of developing this stage I realised it would be much more efficient with time and storage to create a table and add all the centres there, rather than creating a view controller for each climbing centre.

This was as, this method meant that I only had to create one view controller, then depending on what place was chosen the necessary information would be filled in from variables and arrays. As well as this, within stage three I want to implement a search bar, and doing that in a table would be far easier and work much smoother, giving a smaller area for error.

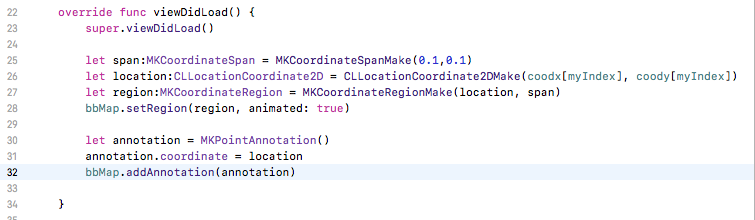
**Test 1**

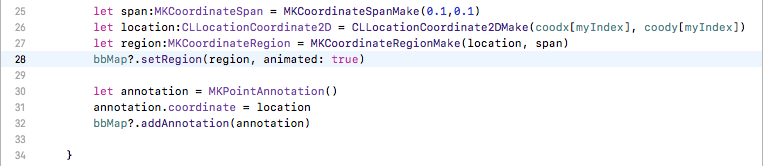
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 2a | Scrolling Around | -Map Scroll  -Valid | Scrolls without error | Crashed,  FAIL |
| 2b | Zooming In/Out | -Map Zoom  -Valid | Scrolls without error | Crashed,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

**Error 1**

../../../../Desktop/Screen%20Shot%202017-10-01%20at%2016.01.

This was my first large error that I had encountered. This error caused my whole program to crash, this was because where I had set the values for my marker, and where the map would be looking, I had force unwrapped my map. This caused the app to crash as it didn’t have a default value, so when it failed to get my data it crashed. To fix this I had put my code from:



To:

Doing this gave the map these values as the default values and were no longer being force unwrapped.

**Test 2**

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

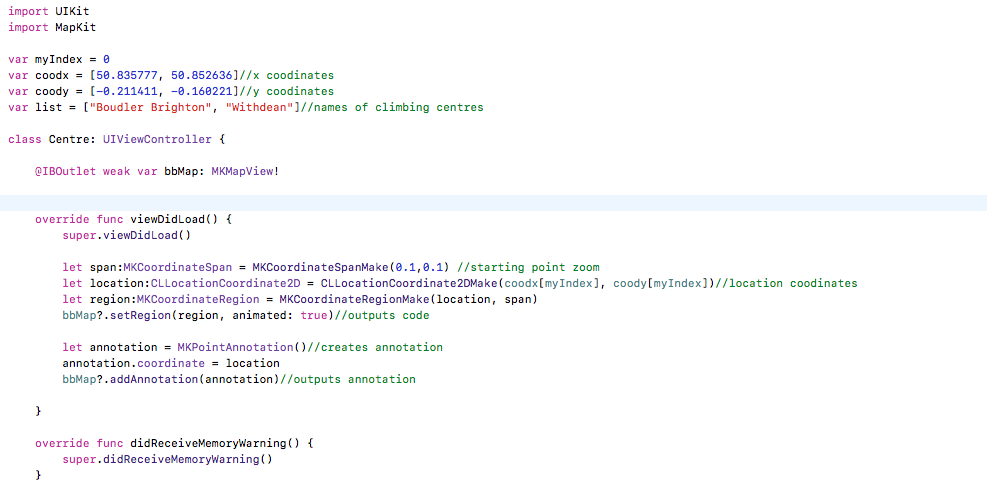
**Error 2**

This error was also connected to the first error. It was because I had put all the necessary code within the view controller for the page with the map, this includes the code for the table to operate. This was causing my variable which was the click input to always be 0, which meant that it only ever presented the map details for Boulder Brighton. Once I had realised this I put the following code into a separate file, and made the view controller with the table linked to this code. As well as fixing this error doing this meant I no longer needed the ‘?’ in the **Error 1**.

(Please ignore error messages, as it was a recreation after the fix).



**Final Code** (table file)

**Final Code** (map file)

**Test 3 – Video 2**

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

Before starting Stage 3, decided that took increase usability and ease of use that I would include the user’s location within the map, just to give a fast and simple idea of how far the desired climbing centre is from the user. For this to happen I had to add some parts to my app, such as creating a request to be able to have the user’s location. Therefore, the following is my final code for this section and everything else which was added.

(Code added: 24-26)

**Test 4 – Video 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2e | Map presents user location | -user location  -Valid | Placement is correct | Placement is correct |

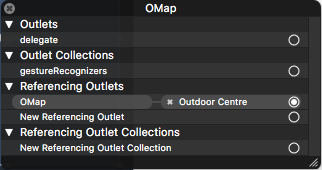
**Stage 3 - Inputting Climbing Areas/Gyms Nearby**

Before I added any code I had realised that I would have to create another two swift files. They are identical versions as what I had done in stage two, yet was for the outdoor centres. All the recreated code was exactly the same, except for I had new arrays for everything that related to an outdoor route.

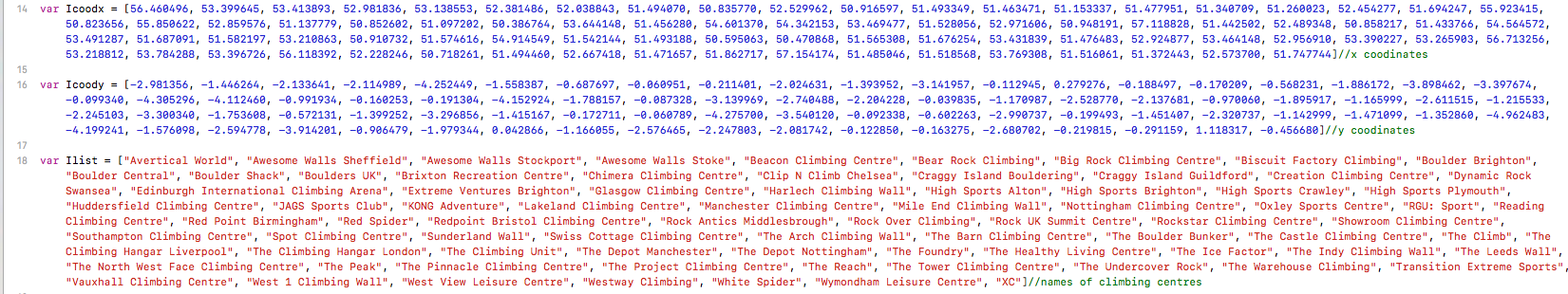
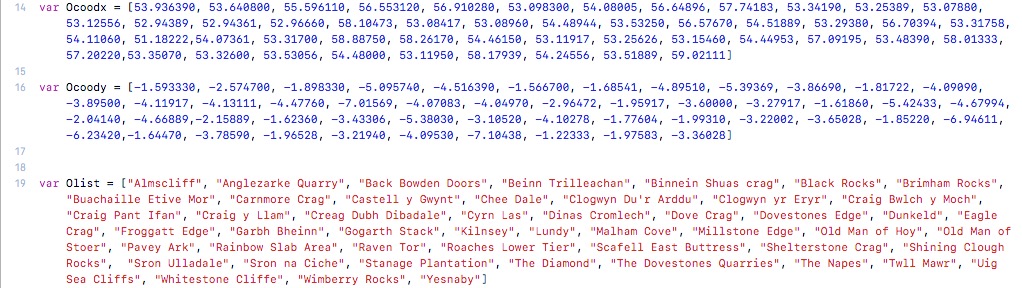
**Test 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 3a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place | Crash - FAIL |
| 3b | Searching for an invalid place | -‘John’  -Valid | No climbing centres shown | Crash - FAIL |

**Error 1**

When re-creating all of the code for my outdoor area I encountered a very large error that caused my code to crash. It was labelled as: ‘terminating with uncaught exception of type NSException’. It was due to the map which was for my outdoor centres had 3 outlets that I had deleted within the code, however did not remove from the map its self, therefore it followed to a non-existent reference. Once realising this I deleted the invalid outlets allowing the code to run as wanted.

Under ‘Referencing Outlets’ 3 other outlets were made that had no proper reference, as well as ‘OMap’, which is fully functional.

As within my Design I mention all the climbing centres that I will add, to achieve this I firstly put all the climbing centres in alphabetical order. Then I would use Google Maps to find each x/y co-ordinates of each location, then added that in order of the climbing centres. I did this as then the index of each location would be identical, making outputting this data to the map very easy.

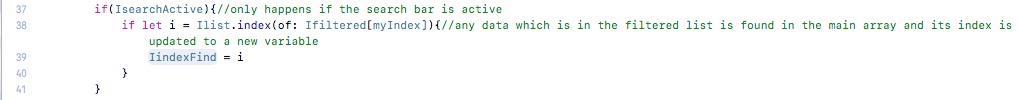
**Test 2 – Video 4**

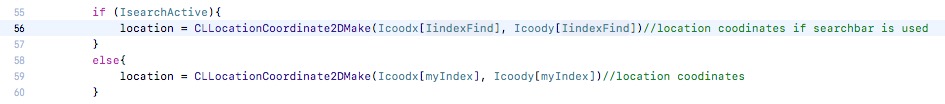
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 3a | Searching for a specific place | -‘Boulder Brighton’  -Valid | Presents specific place | Presents specific place |
| 3b | Searching for an invalid place | -‘John’  -Valid | No climbing centres shown | Instead of no climbing centres shown when nothing valid is entered, all are shown. I felt this was a better outcome. |

I added the code to the table file for my search bar, while using <https://shrikar.com/swift-ios-tutorial-uisearchbar-and-uisearchbardelegate/> as a basis to follow to implement my code.

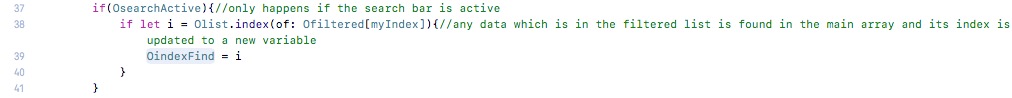
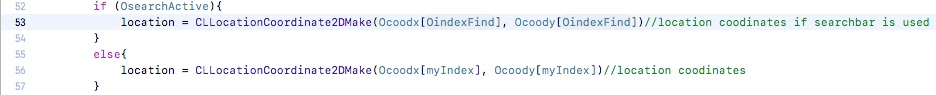
**Error 2**

This code did pass my tests for my Stage 3 tests, however would’ve caused my Stage 2 tests as I had encountered an error. This error was logical, so meant that my code ran fine. However, the error was that when something is searched for the table changes. So if you refer to the final code within Stage 2, I use a variable ‘myIndex’ which changes depending on which cell you chose. This code worked perfect when nothing was searched, however when something is searched the top result will cause ‘myIndex’ to have a value of 0. Yet Ilist[0] would only ever refer to ‘Avertical wall’. Therefore, I added the following to my code for a climbing centre to mean that the correct index was used if the search bar was used.

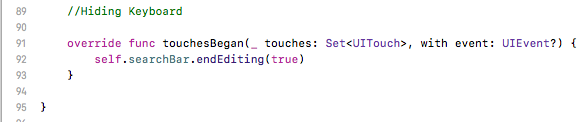
**Indoor** (IndoorCentre File)

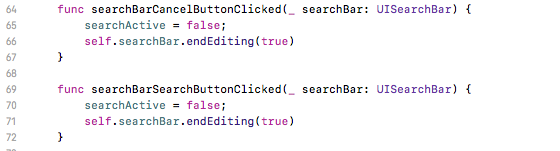


**Outdoor** (OutdoorCentre File)



Afterwards I felt that once you had inputted something into the search bar, that I was necessary to be able to remove the keyboard, if one felt it was no longer necessary. The following code was added to the table file.

(lines 91-93, 66, 71)



Stage%203/Table%20colour.pngAs well as this added a small piece of code into my table file to make the table a light grey instead of white, to follow the colour scheme of my app.

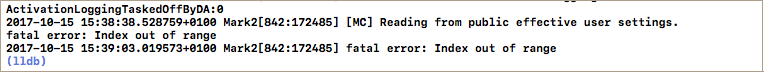
**Stage 4 – Partners Section**

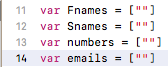
To create functional tests and make the integration of the data base smoother, I made multiple arrays that acts as the database for each collection of data. Then I created a simple table view, like I have with the previous stages. From there I created a collection of text fields that link into my code as inputs.

**Test 1**

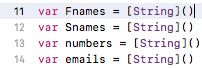
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 4a | Inputting no data to see if the validation has worked | -Invalid | Rejection of data. | Crash when trying to upload  FAIL |
| 4b | Inputting data into First Name, but not into Email or Phone Number. | -‘Nathan’  -Invalid | Rejection of data. | Crash when trying to upload  FAIL |
| 4c | Inputting data into Email, but not into First Name or Phone Number. | -‘np@aol.com’  -Invalid | Rejection of data. | Crash when trying to upload  FAIL |
| 4d | Inputting data into Phone Number, but not into First Name or Email. | -‘12345678901’  -Invalid | Rejection of data. | Crash when trying to upload  FAIL |
| 4e | Inputting data into First Name and Email, but not into Phone Number. | -‘Nathan’, ‘np@aol.com’  -Valid | Acceptance of data. | Crash when trying to upload  FAIL |
| 5f | Inputting data into First Name and Phone Number, but not into Email. | -‘Nathan’, ‘12345678901’  -Valid | Acceptance of data. | Crash when trying to upload  FAIL |
| 4g | Inputting data into First Name and Email and Phone Number. | -‘Nathan’, ‘n@aol.com’, ‘1234567890’  -Valid | Acceptance of data. | Crash when trying to upload  FAIL |
| 4h | Inputting data that has already been used. | -‘Nathan’, ‘np@aol.com’, ‘12345678901’  -Invalid | Rejection of data. | Crash when trying to upload  FAIL |
| 4i | Inputting data into description | -‘Nathan’, ‘np@aol.com’, ‘12345678901’  ‘I like climbing’  -Vali | Rejection of data. | Crash when trying to upload  FAIL |

**Error 1**

When trying to upload any details into my code I would crash and give the following error:

The cause to my error was due to my bad declaration of my arrays. I declared them like:

These declarations are incorrect as it only has one piece of data, so when attempting to append to the code it would overflow and crash. Therefore, I made a simple fix and made it into the following:



**Test 2 – Video 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 4a | Inputting no data to see if the validation has worked | -Invalid | Rejection of data. | Rejection of data. |
| 4b | Inputting data into First Name, but not into Email or Phone Number. | -‘Nathan’  -Invalid | Rejection of data. | Rejection of data. |
| 4c | Inputting data into Email, but not into First Name or Phone Number. | -‘test1@aol.com’  -Invalid | Rejection of data. | Rejection of data. |
| 4d | Inputting data into Phone Number, but not into First Name or Email. | -‘12345678901’  -Invalid | Rejection of data. | Rejection of data. |
| 4e | Inputting data into First Name and Email, but not into Phone Number. | -‘Nathan’, ‘test2@aol.com’  -Valid | Acceptance of data. | Acceptance of data. |
| 5f | Inputting data into First Name and Phone Number, but not into Email. | -‘Nathan’, ‘12345678901’  -Valid | Acceptance of data. | Acceptance of data. |
| 4g | Inputting data into First Name and Email and Phone Number. | -‘Nathan’, ‘test3@aol.com, ‘987654321’  -Valid | Acceptance of data. | Acceptance of data. |
| 4h | Inputting data that has already been used. | -‘Nathan’, ‘test3@aol.com, ‘987654321’  -Invalid | Rejection of data. | Rejection of data. |
| 4i | Inputting unique data from each section | -‘John’, ‘John@aol.com’, ‘1111111111’  ‘I like climbing’  -valid | Rejection of data. | Inputted correctly |

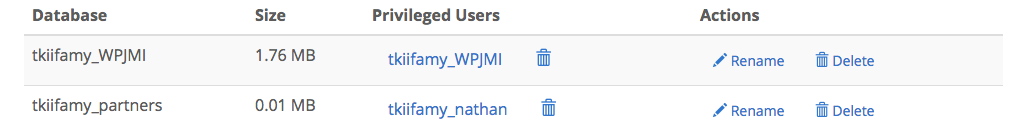
**Unplanned Code**

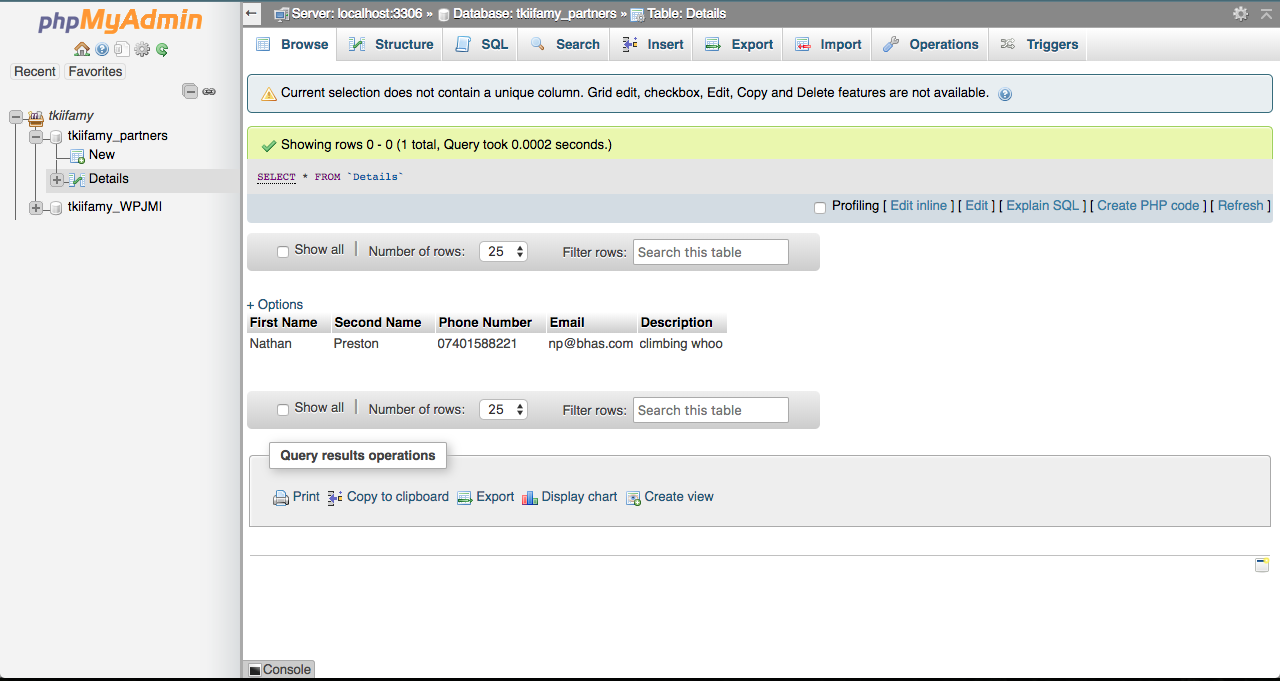
As seen in the video I had added some unplanned cod. First was to allow the description box to move into the centre of the screen, whilst it was being edited. I felt this was necessary as it’s nice to have all of the inputs on one screen but when the keyboard appears it would cover up all of the description box, not allowing the user to see what they’re typing. So I pushed it up whilst it is being changed then put it back to the original position once the changes have been finished. The second piece of code was to make a placeholder text. As within Xcode Text Fields have a premade function to allow you to add a placeholder, but for a description I needed multiple lines. So I needed to use a Text View. The code I made was just to change the colour of the font, and make the placeholder text disappear once touched. Plus, if it was untouched to make sure the placeholder text isn’t inputted along with the details of the user.

When creating the table, I had to create a new file to make each cell within the table custom, allowing me to change it the way I want. Very little code went into this file, just references to each piece of text.

**Stage 5**

Here I needed to have a server setup as the database for the Partners section must be available for multiple people to connect to. So I followed a tutorial (codewithchris.com/iphone-app-connect-to-mysql-database/) and had Bluehost suggested to use as a host as it is very useable.

So I began by creating a databse within the section MySQL databases. I created one database named: tkiifamy\_partners, and connected that to a user named: tkiifamy\_nathan. I allowed all privileges for the user and linked them.

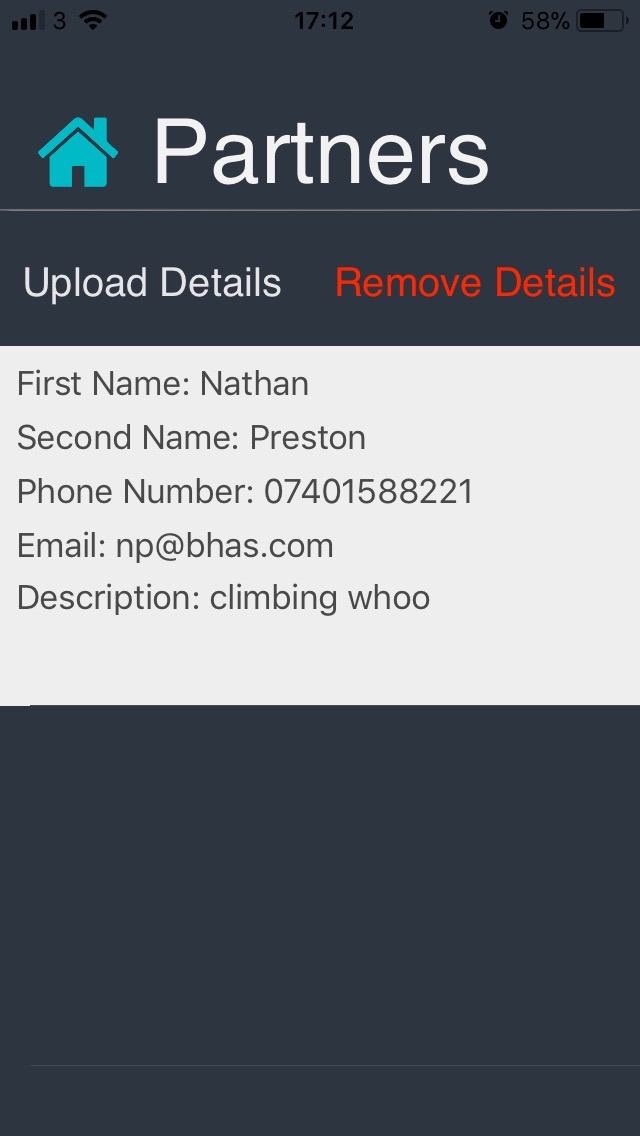
From here I used phpMyAdmin to create a table Details within this added my test row of data.

Once all the server part was created I needed to create a php file that would read my table, then output the contents of the table. From that I created two ‘Models’ to handle and store the data passed from the php file. It all gets passed between arrays and repeats error checks, outputting what the error would be.

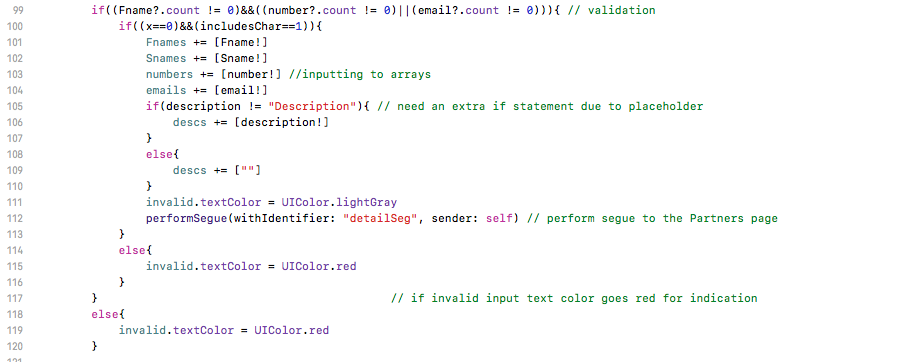
**Test 1**

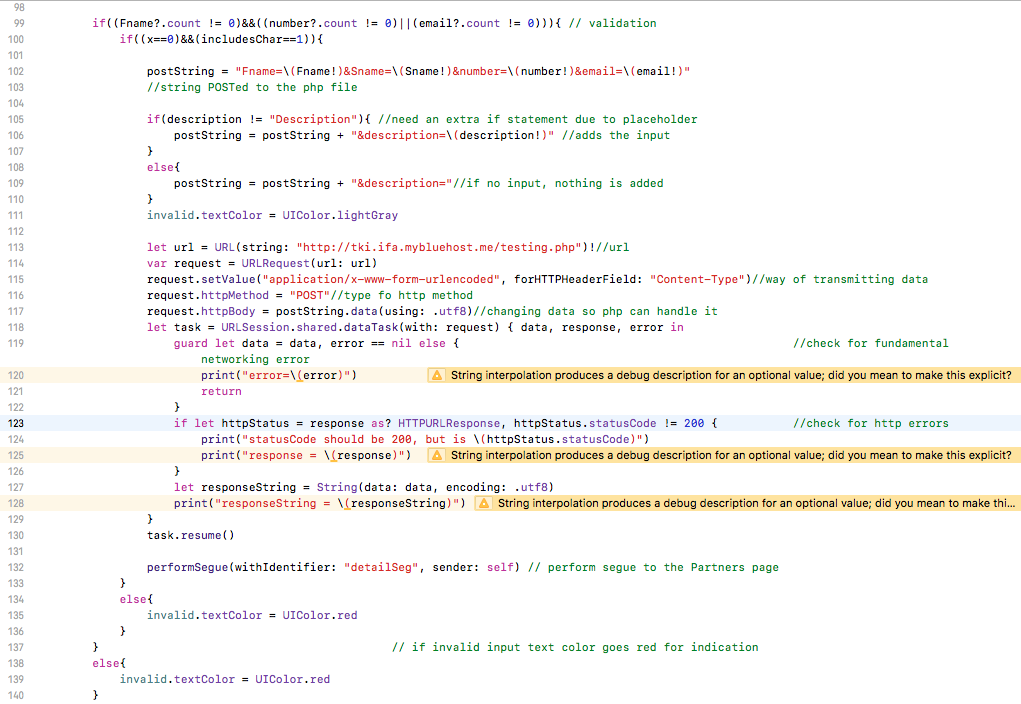
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No.** | **Description** | **Test Data** | **Expected Result** | **Actual Result** |
| 5a | Setup sections for the partner’s area. So 5 valid inputs. | - ‘Nathan’, ‘Preston’, ‘07401588221, ‘np@bhas.com, ‘climbing whoo’  -Valid | All data inputted without error | All data inputted without error |
| 5b | Outputting data | - ‘Nathan’, ‘Preston’, ‘07401588221, ‘np@bhas.com, ‘climbing whoo’  -Valid | Inputted data will be outputted | Inputted data will be outputted |

From the phpMyAdmin screenshot, you can see that I had added the test data there. Then from the screenshot from my app it shows that how the data had correctly been passed through and added to my app.



To make this work I had to remove the input to the arrays and change it for an input to a string, which would then be passed to the php file, and then inserted into the database with via SQL.





**Stage 6**

**Errors**

For this section I ran into multiple errors, all of which were due to my misunderstanding of how to POST to a database. I tried to follow multiple tutorials, however most caused lots of errors as they were outdated and as I tried to use multiple tutorials together many syntax errors were caused. The biggest error was due to forgetting a semi colon within my php file and therefore causing my program not to work.

However, once further researching how to push with swift and connecting my swift to my php file I used this tutorial as a basis: <http://www.ios-blog.co.uk/tutorials/swift/swift-how-to-send-a-post-request-to-a-php-script/>.

**Test 1 – Video 6**

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

**Error 1**

Here a logical error occurred, as I used all my tested validation from stage 4, all of which is fully functional as nothing has been changed, except from the input. However, I hadn’t fully changed the validation, with being able to re-input unique data. It needs to be adapted for the use of the database rather than various local arrays.