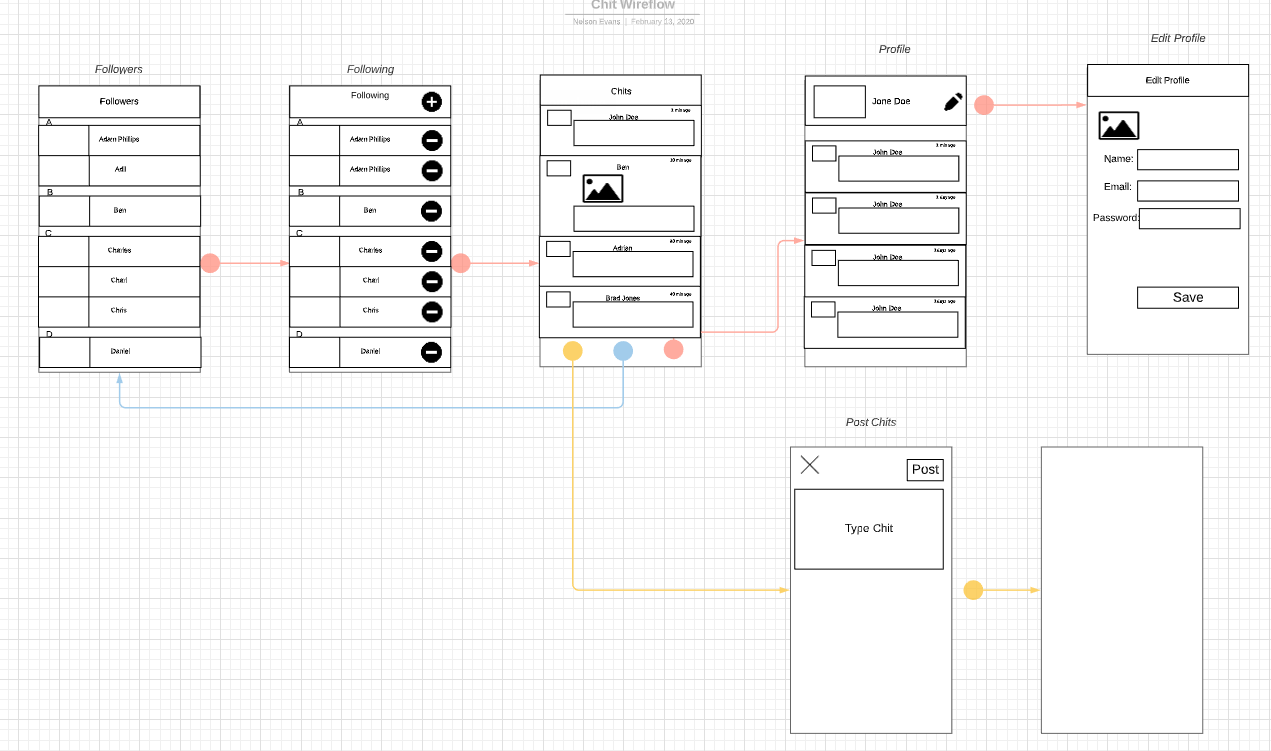
**Mobile Applications Development Report**

## How the application’s structure and flow was designed from the API specification

Wireframe 1



The first wireframe design that was created was initially sketched on paper and the overall page designs were created using Lucid Chart. The first design was very basic, the pages had no colour scheme, the idea was to use the basic design to get a better understanding about what kind of information would be transferred as you navigated through the pages. In the first design, there was no login page, the first page shown to the user was the chits page which consisted of a flat list to display all the chits retrieved from the server. Each chit contained an image for the profile picture of the user, their name, timestamp and the chit content itself and an image if posted by the user.

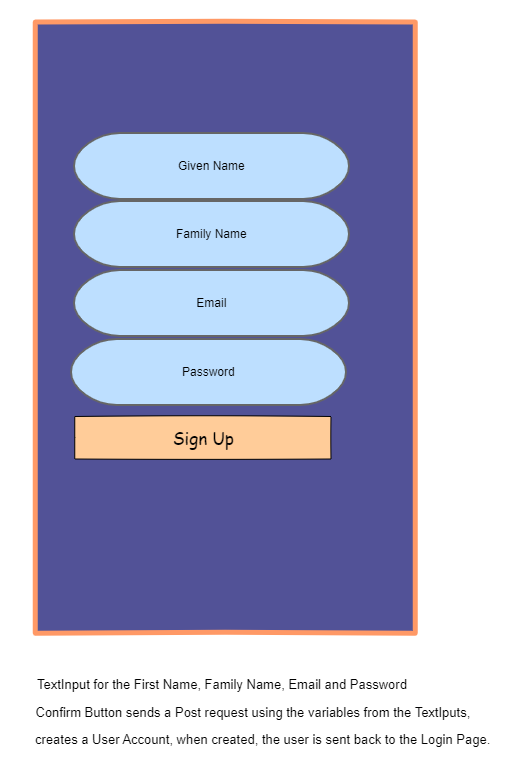
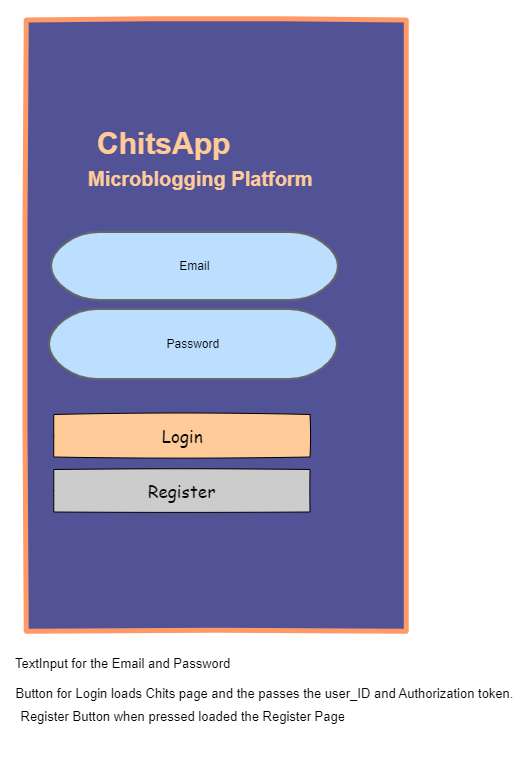
By starting with the Chits page, it gives the user an opportunity to experience the apps functionalities before thinking about joining the platform by creating an account, it doesn’t rush the user into any commitments. It was previously stated that there should be a login page, but users that do not have accounts should be able view the application but with some restrictions, with this design, these restrictions weren’t taken into consideration.

The type of navigation used for this design was a stack navigation firstly to store all the JavaScript pages, but also a Bottom Tab Navigation that displays the most important pages for the application which were the Chits page, the Following page and the Post Chits page. The primary method of navigating between the pages in the stack is by swiping left to open the Following and Followers pages and to swipe right load the Profile and Edit Profile pages. This would allow smooth transitions between the pages and when used in conjunction with the tab navigation, it allows the user to return to the main pages quickly. This feature would make it easier to return directly to the main Chits page instead of swiping multiple times until you reached the Chits Page.

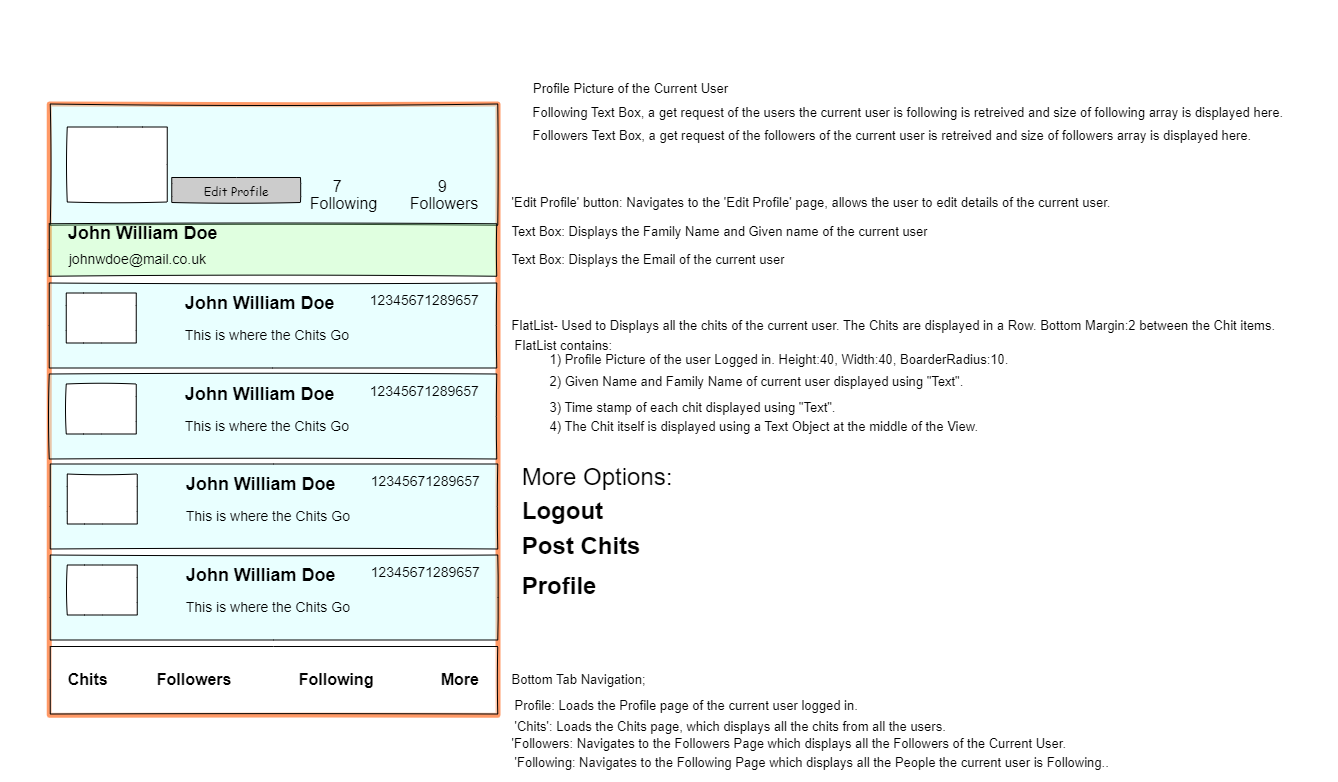
The Followers page consisted of a Flat List to display all the followers that the current user had, the users were to be displayed in alphabetical order with a letter label being used to separate the different users. The content in the Flat list are a profile picture of the user and a Text element to display the name of the users. The Following Page followed a similar structure to the Followers page, with the people the user currently follows being displayed in a Flat List, organised in alphabetical order with letter labels separating the different users. The only difference between the two pages were the addition of an unfollow button to the flat list in the Following page, this when pressed would unfollow the selected user.

The Profile page of the user currently signed in consisted of an header which displayed the profile picture of the user on the far left, the user’s name in the centre using a Text element and an image icon on the far right that when pressed opened the edit Profile page allowing the user to edit their profile details. All the chits that the user had posted were to be displayed on a flat list in a similar manner to the Chit’s page, the profile picture on the far left, the user’s name in the centre using a Text element and the timestamp of when the chit was posted in the far right using a Text element. The chit content is displayed in the middle of the flat list either as a Text element or an image.

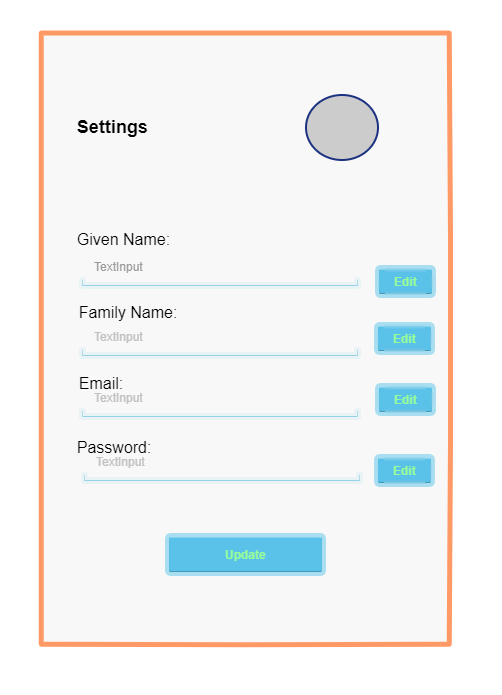
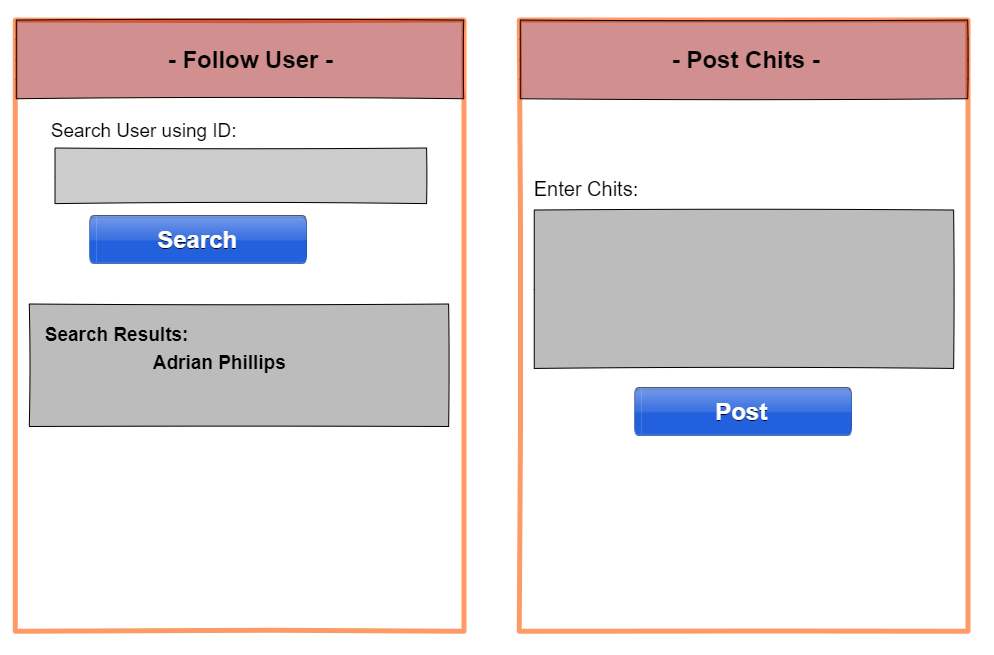
Wireframe 2



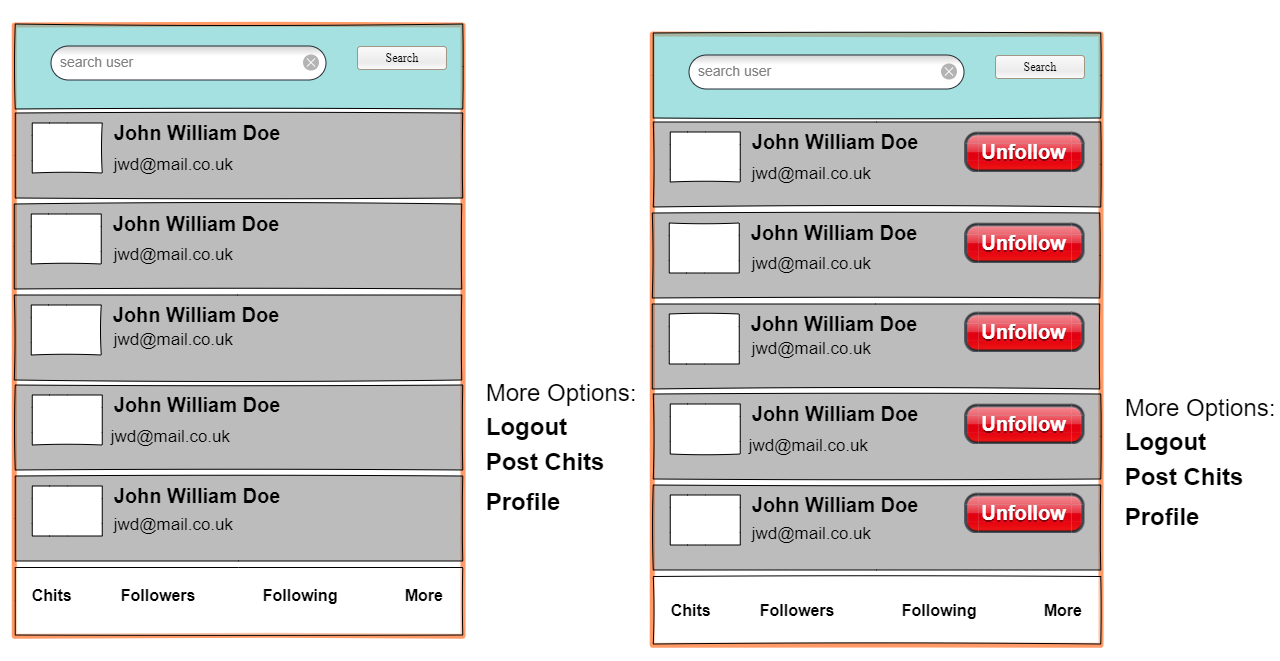
Like before most of the pages were stored in the stack navigation, in total there will be 10 pages, which are too many for them to be shown in the Tab Navigator, so instead the most crucial pages were displayed, and the others were hidden in the Stack Navigator. The first page to be loaded this time was the ‘Login’ Page, which consisted of two light blue Text Input buttons with curved boarder radius’, an orange Login and a grey Register Button. The background colour chosen was purple, with orange text, the colours chosen contrasted well with each other. When the Register button is pressed, the Register page is loaded, the Login and Register page have the same colour scheme and layout of the Text Input and buttons. It consists of 4 light blue Text Input and a Sign Up button which uses the input entered from the Text Inputs to generate an account for the user.

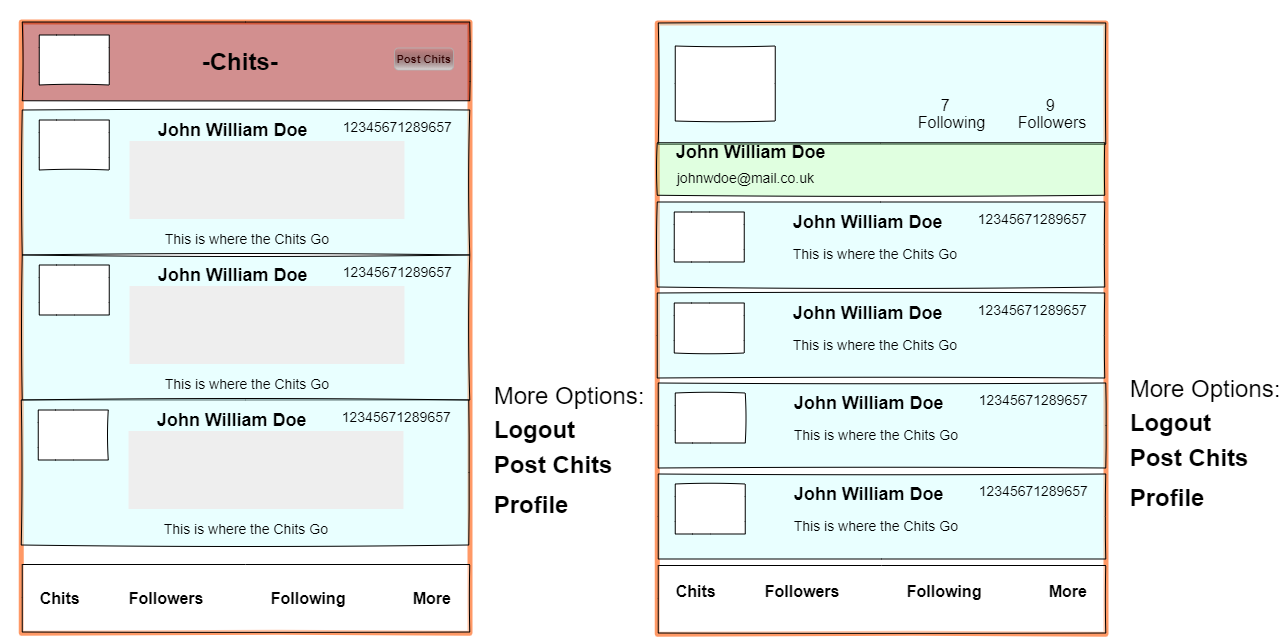


The wireframe design had some major changes, however the stack and Tab Navigation remained but the structure and flow of the pages changed. When the main app is loaded, the method of swiping left and right to load the different pages was discarded. The most important pages will be displayed on the Tab Navigator, these pages were the Chits page, Followers, Following and ‘More’s page. This page held other pages that were important but their inclusion to the Tab navigator would have cramped the space and made it unattractive. Instead they were hidden, when the ‘More’ option is clicked, the option to view the user’s profile is shown alongside the post chits and Logout. The Profile page is the first page to be loaded when login is successful, the page consists of light blue background and light green subsection which contains two Text elements to display the full name and email of the user. This subsection was a new introduction to this design which separated the header into 2 sections allowing more space to display the number of followers and people the user is following. Apart from that, everything remains the same for this page.



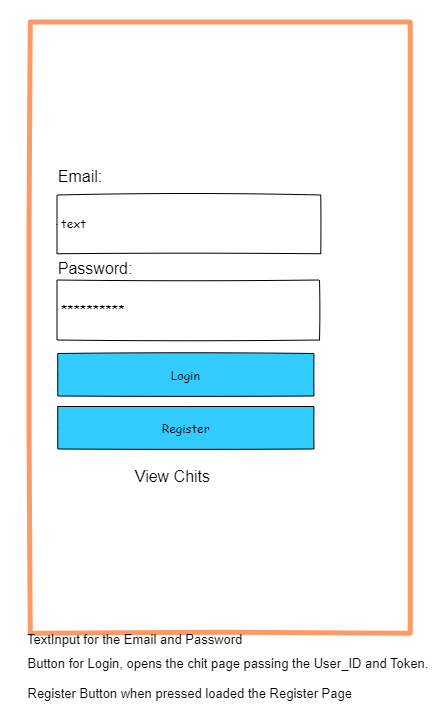
Changes were made to the overall colour scheme of the Edit Profile page, with the details of the user being displayed on the Text Boxes which were transparent, the Buttons to Edit the individual details are light blue and the font colour is light green. The background colour and the general design of this page was the best from this design. It gave the user the option to edit the details individually or all at once with the Update Button. The profile picture can be changed by pressing on the profile picture which opens the camera and allows a picture to be taken as the new profile picture. The new pages added are the Follow user’s page, which contains a Text Input and a Search button, this page is used to search users to follow, the result of the search is displayed on a Flat List. When the user presses the result Text elements containing the users name, the selected user is followed.



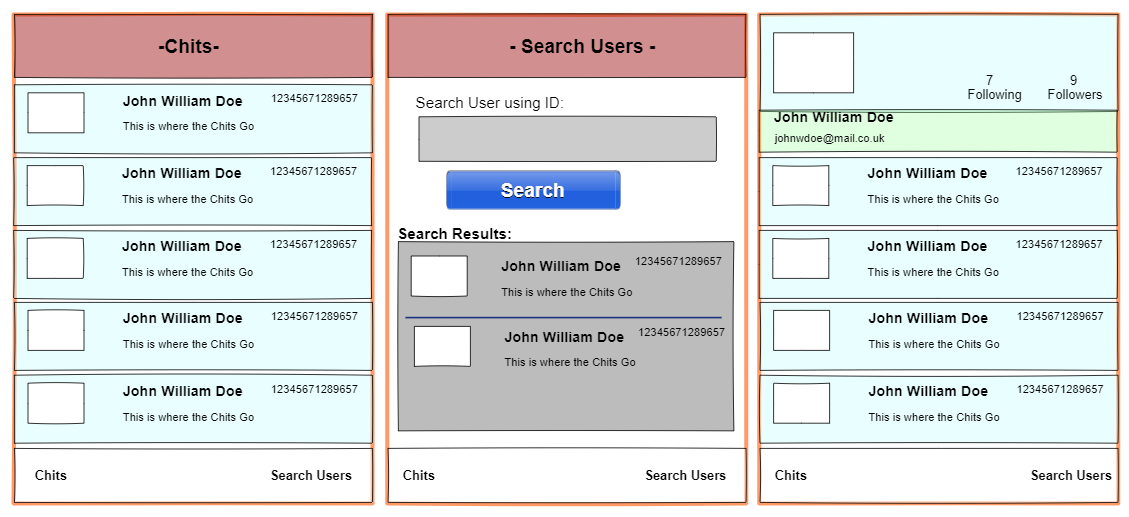


In general, the Chits page remained the same as the first design, only the colour scheme changes, with a button to post a chit being added in the header instead of the Tab Navigation. The new additions to the Followers and Following pages are the search bar which allows the user to filter through the user’s in the Flat List and display the results from the search bar. Compared to the first wireframe, the users are not organised in alphabetical order. A new page is added called ‘Selected Profile’, which whenever the profile pictures of the users in the Followers, Following and Chits are pressed, the profile pages of the selected users are loaded.

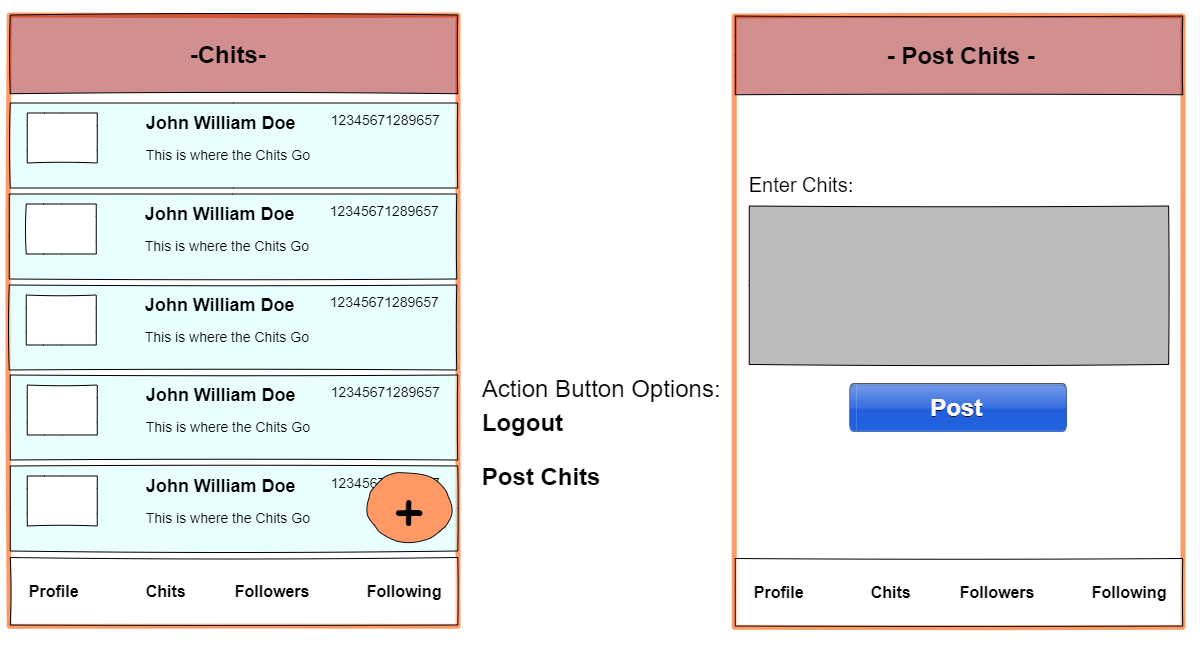
Wireframe 3



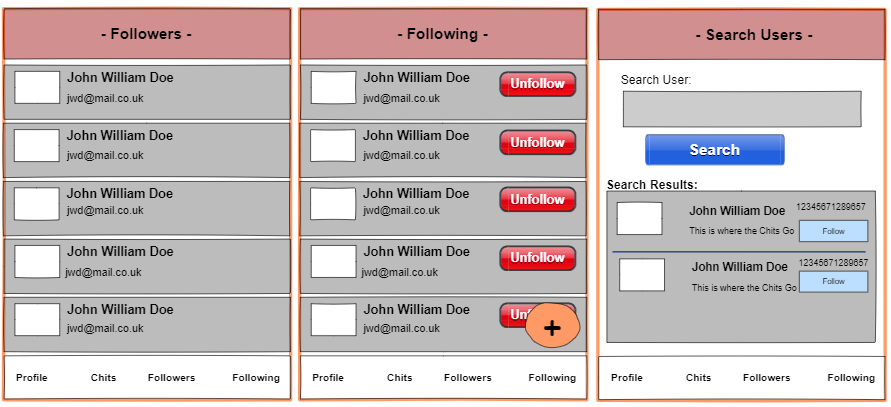
I decided to use wireframe 3 as the final design for the application instead of wireframe 2 because it had a simpler design in general which was much easier to implement on react native. The overall structure of wireframe 3 remained the same as the previous design but it was much easier on the eye but effective in accomplishing its purpose. The Login and Register pages remained the similar with the same elements being used and for the same purpose, but a new addition that is crucial for the project was added. This was the option for users that don’t have an account to experience some of the apps features before deciding to create an account. This was the addition of the ‘View Chit’s Text element, a new stack Navigation is created containing the normal Chits page available to all users, the Search User’s and Selected User’s pages. The Chits and Search User’s pages being added to a Bottom Tab Navigation to allow the user to switch between the two pages and the Selected User’s page being hidden in the Stack Navigator. When the ‘View Chits’ text is pressed, the ‘Chits’ page below is loaded.

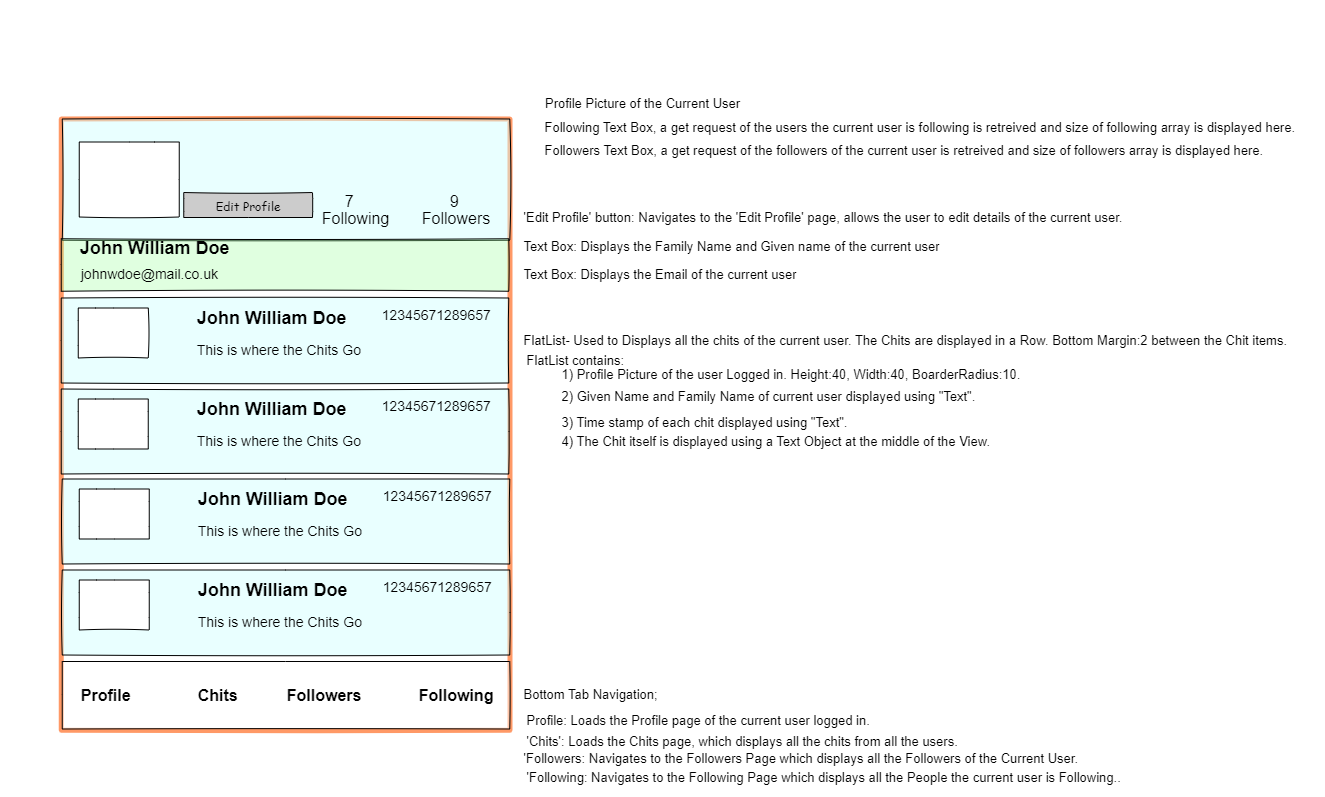


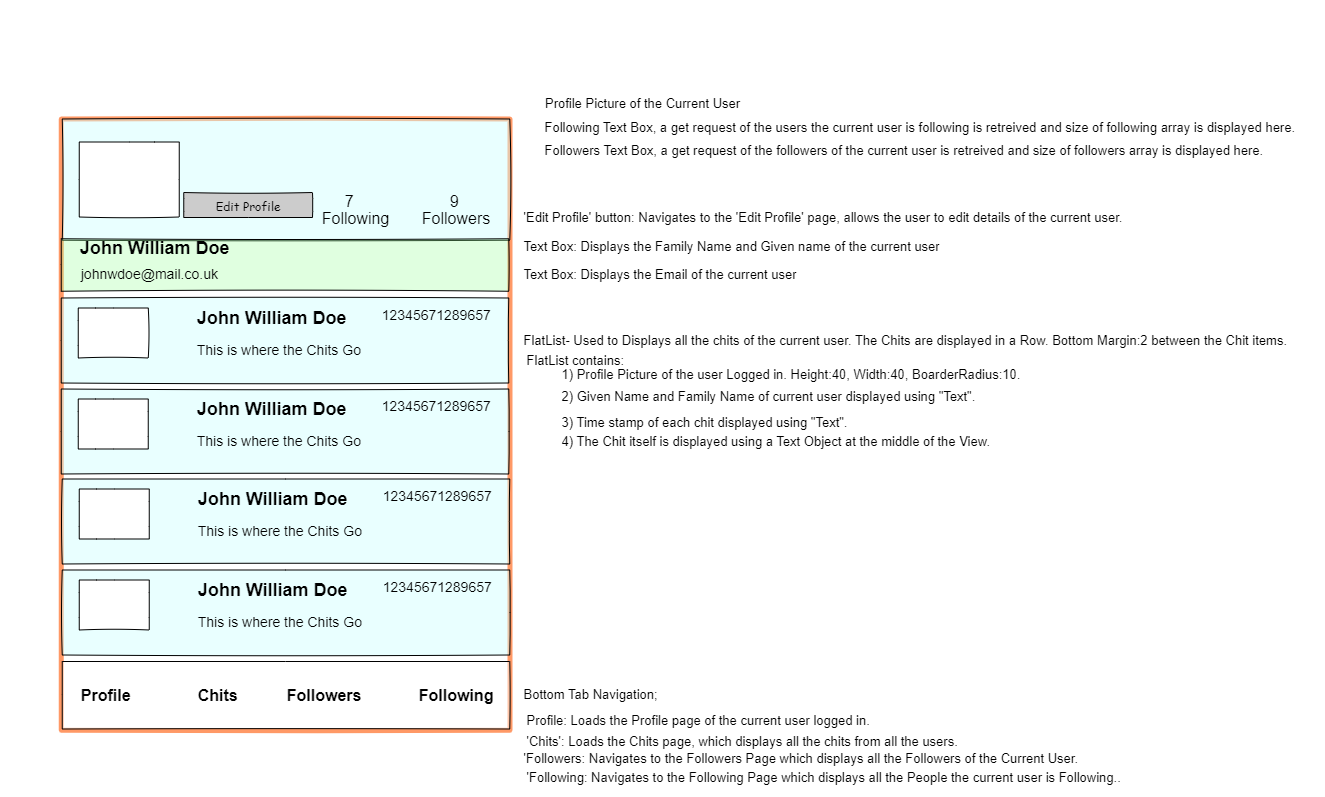
The screenshots above contain the pages available to the user’s that havent logged in, the features are almost the same as the user’s that have logged in apart from the fact that the Action Button containing the options to Post Chits and Logout is hidden in the Chits page and the button to follow a user in the Search User’s page is hidden. In general, this design gives the user an option to experience the app and hides aspects of the app available for members away from them. More features were added but the design still keeps it’s simpilicity. Like the previous design, whenever a user clicks on a profile picture of any of the users on the Flat Lists all the pages except the profile pages, the profile page of the user selected is displayed. Instead of duplicating the Chit’s, Search User’s and Selected Profile pages, the same pages are used for the users that have logged In and those that haven’t, by hidding features exclusive to logged in users, this prevents the duplication of pages and code which was premonate in the previous design.

Main App

The first page to be loaded is the Chits Page, the chits are displayed on a Flat List, with the user’s profile picture being clickable to load their profile page. Like the previous designs the Chits Page had little changes, the Header was reverted to only display the page title. The only changes made to this design were changing the pages shown in the Tab Navigator, the profile page was added back with the More option being removed. To replace it, an Action Button was added that would contain the options that enabled the user to logout and post Chit’s, and this gave it a cleaner look.



Profile Page



The overall layout of the Profile page remained the same as the previous design, except for the pages on the Tab Navigator, when the Edit Button is clicked, the page to edit the user details is loaded. Although the design of the Edit Profile page changed, the way it works remains the same, when the picture is pressed, an Image Picker is loaded which enables the user to change the profile picture.

## 

After the initial design of the final wireframe, new features have been added into the app that weren’t present in the design. In the Action Bar on the Chit’s page, a new option is added that enables the user to view all the chit drafts that the user has made, it enables the user to delete, edit and post the chits. The ability to store chit drafts and post them at a later date just gave it a more similarity to Email and social media platforms. In the Post Chits page, a new button was added that allowed the user to save the chit drafts instead of posting them, as well as having the ability to post image chits using the Image Picker with the picture taken being displayed to the user.

## The information contained within the README file

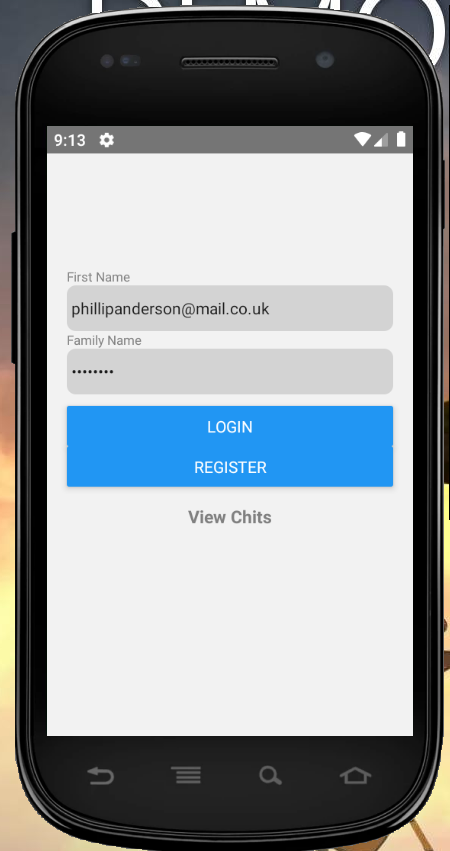
# Chittr-App

Chittr is a Microblogging platform that enable users to sign up for an account and publish 'Chits' (Short textual based) posts that are no longer than 141 characters. After creating an account, the user can add a profile picture for their account which can be updated anytime using the photos on the device or taking a photo using the phones camera. Users can follow their friends and colleagues to be updated on their activities and know what they are 'Chitting' about. A new feature was added that allows pictures to be published as chits, instead of just Text, you can share your experiences with your followers. This platform also allows users to get a taste of the experience before creating an account. Without an account, the user can view all the chits posted, search through and view the profiles of all the users on the platform.

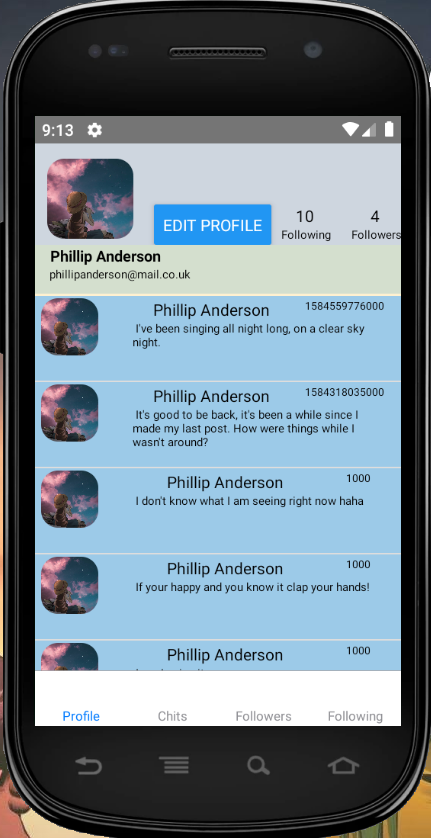
## Visuals

There are screenshots of the application below, that show how it looks, including how the application works.

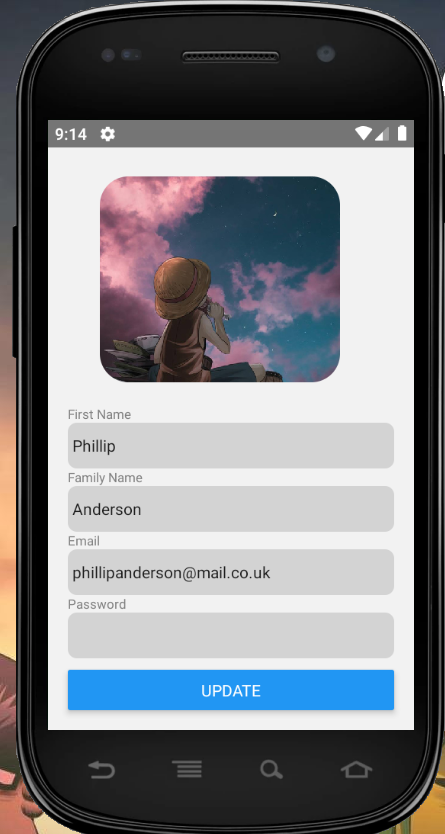
Login Screen

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Login_Screen.PNG)

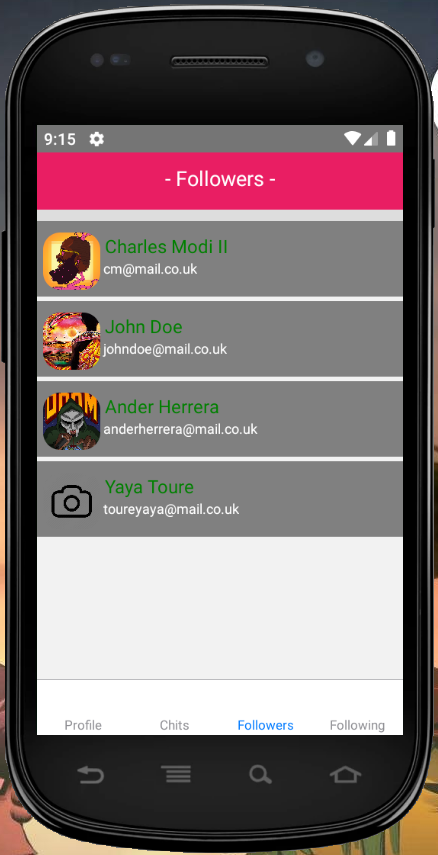
User Profile Screen

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/User_Profile.PNG)

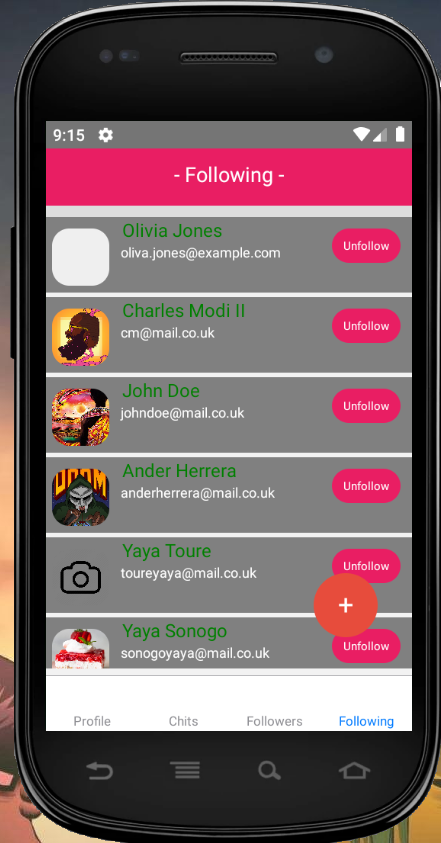
Edit Profile Screen

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Edit_Profile.PNG)

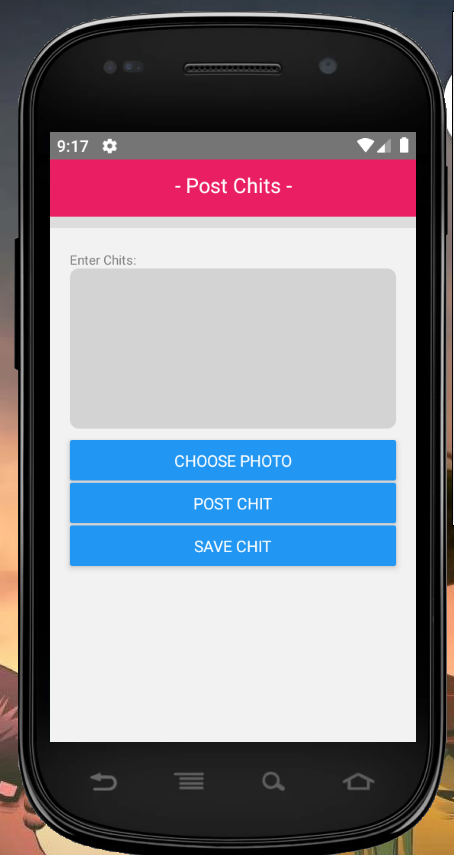
Followers Screen

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Followers.PNG)

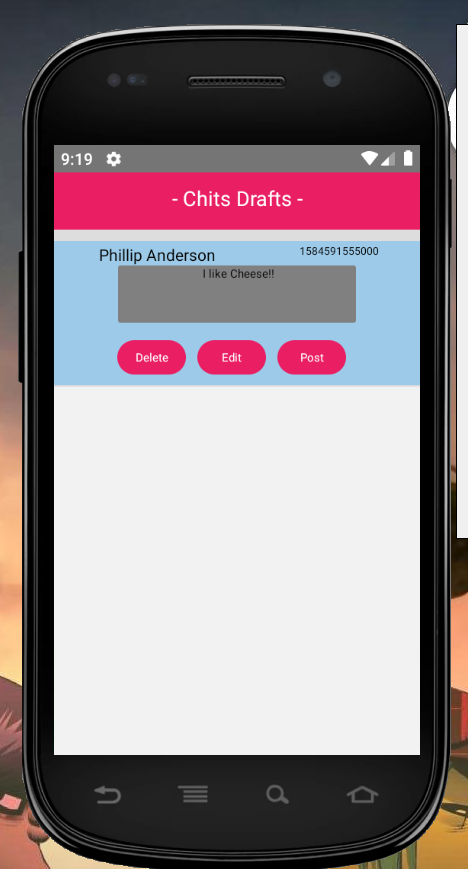
Following Screen

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Following.PNG)

Post Chits Screen

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Post_Chit.PNG)

View Saved Chit Drafts Screen

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Chit_Drafts.PNG)

## Installation

Before showing how to run the application, you will need to install the React Native platform on your machine. Click [Here](https://reactnative.dev/docs/getting-started) to redirect you to the instructions of how to install React Native CLI on your machine. On the link provided above, follow the instructions for the React Native CLI QuickStart.

Once the app has been installed, open the Command Prompt and navigate to the 'myApp' folder, then type 'npm install', this will install the node\_modules for the app. After installing the node\_modules, before running the Chittr app on your device, check to see that all the packages used in the app are in the node\_modules page and if they are skip the upcoming stage. If the packages are missing, navigate using the command prompt to the 'myApp' folder and use the package manager [npm](https://www.npmjs.com/) to install packages below that are used by the application:

npm install react-native-geolocation-service

npm install react-native-image-picker

npm i react-navigation-stack

npm i react-navigation-tabs

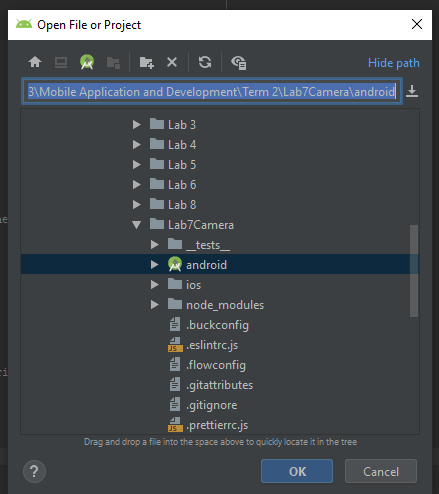
npm i react-navigation

npm i react-native-vector-icons

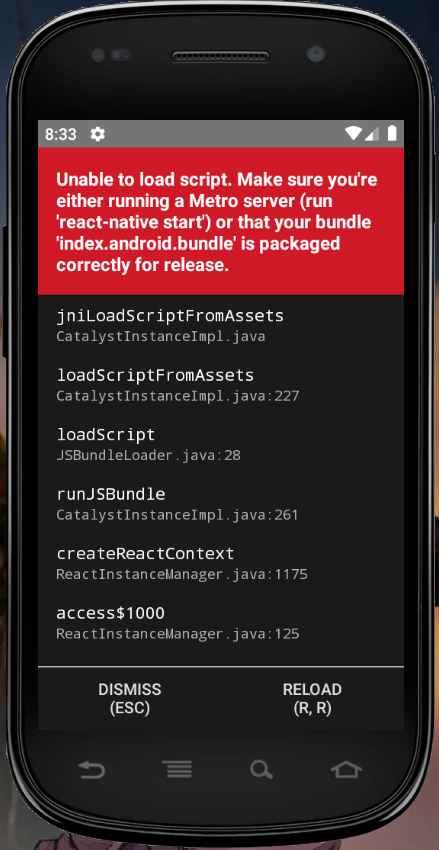
npm i react-native-action-button

## Steps to Run the Application

I will now guide you through how to run the application below.

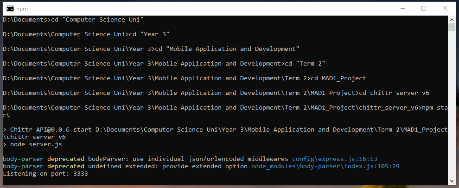
[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Open_Android_Folder.PNG)

After installing the packages above, to run the application first run Android Studio, open the 'android' folder inside the 'myApp' folder and run the program on an emulator.

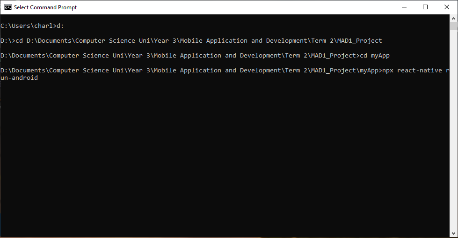
[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Run_Android_Project.PNG)

The App has finished loading on the emulator.

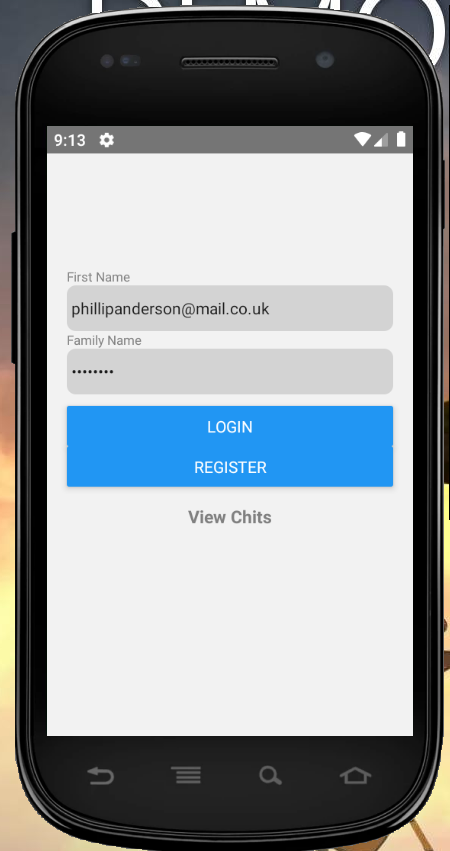
Start the Server

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Start_Server.PNG)

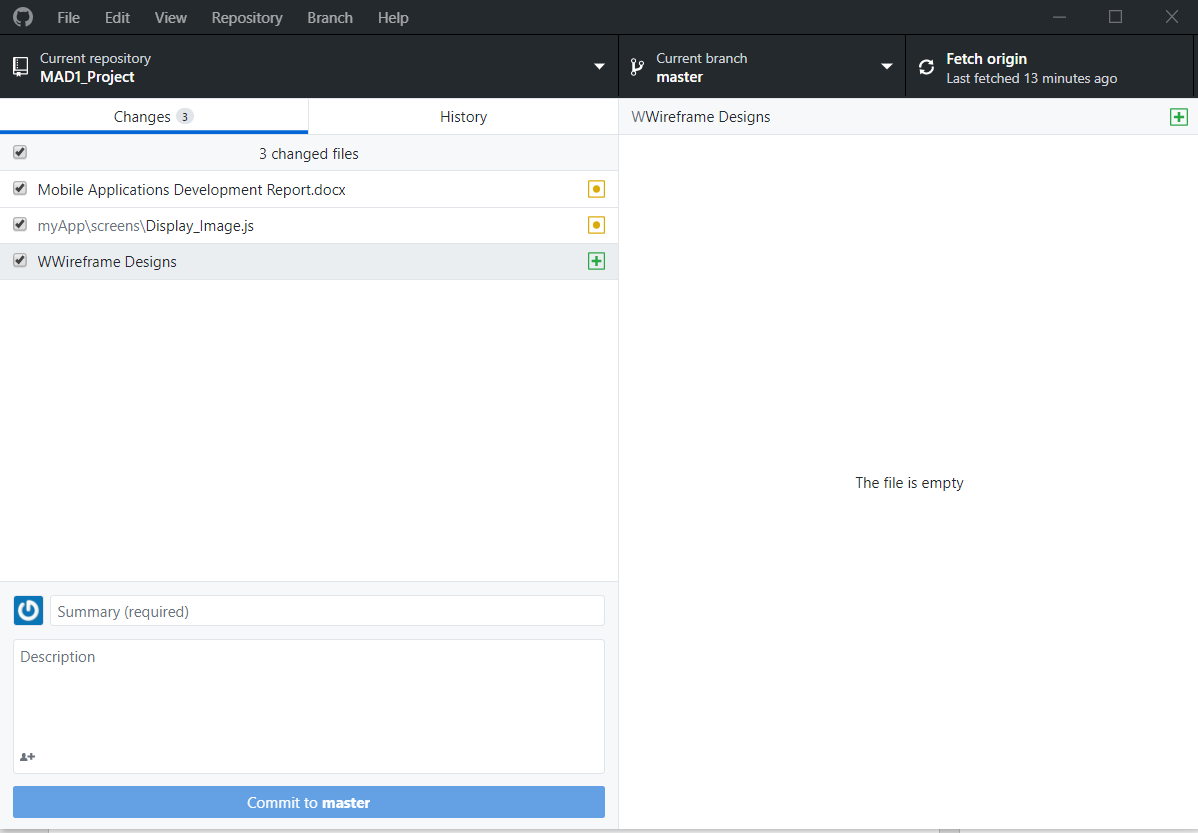
Once then emulator has finished loading, open the command prompt and navigate to the 'chittr\_server\_v6' folder and type npm start on the cmd.

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Run_App.PNG)

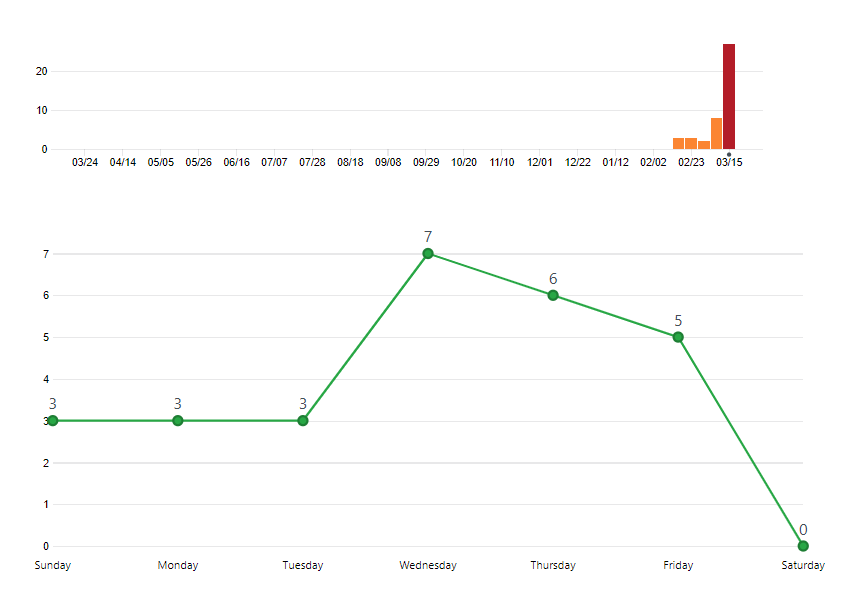
After the server is on, open another Command Prompt, navigate to the 'myApp' project directory and type npx react-native run-android. This will start the application and the Login Page will be displayed on the screen.

[](https://github.com/YellowSubmarine1/MAD1_Project/blob/master/Images/Login_Screen.PNG)

## How version control software has been used within the project.



GitHub was used throughout the project to keep track of changes to documents and files so that the current version is known, after each commit, the features that had been implemented into the product are highlighted before a commit is made, this allowed me to keep track of the changes made throughout the development of the project. It serves as a backup software, when issues occur with the current version that are difficult to identify or solve, by making note of the changes made after each commit, it allowed me to roll back to the previous working version. When problems occurred, I copied the changes that I had made on the files and rolled back to the previous version and added the code back to old version, this method allowed me to make progress relatively quickly and catchup to the point where the bug occurred.

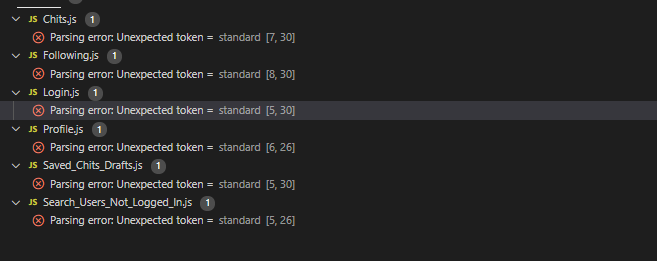


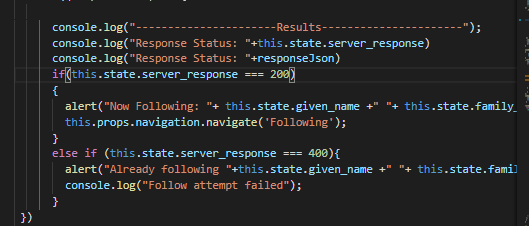
GitHub Desktop was used conjunction with the Trello Boards using Kanban methodology, I took tasks from the To-Do list of the Trello Board and worked on them until completion. After a few tasks were finished, they were moved onto the Complete List and a commit was made and the features or changes made were highlighted for that version. Alongside the Complete List on Trello and the GitHub commits, it gave a better understanding of the changes made, allowing me to track and control the changes made to the software.

As shown from the screenshot of the commit history, at the start of the project, the progress made was very slow, with only 8 commits being made from the first 3 weeks after the coursework was published, this was mostly because of the deadline for our personal projects was on 21st February. All the time that would have been spent on this coursework were spent on completing the personal project before the deadline. After the project was submitting on the 21st February, I slowly made progress on creating the three wireframe designs for the app layout and submitting a basic android studio application that was able to display the chits, followers and following list of a specific user. From the 8st March, I began to pick up the pace, invested most of my time on this project and completing more tasks from the Trello Board. The Screenshot above shows up to 8 or more commits were made per week till the deadline day.

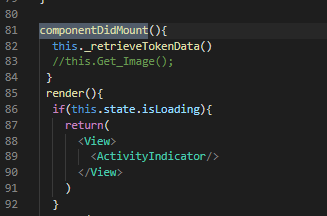
## Details of the code style guide used and how it was used

The code style that was used near the end of the app development was Standard JS, it was installed as an extension for the project workspace and as development progressed. After being run through all the JavaScript files in the current directory, it initially highlighted parsing errors such as tokens being used incorrectly.



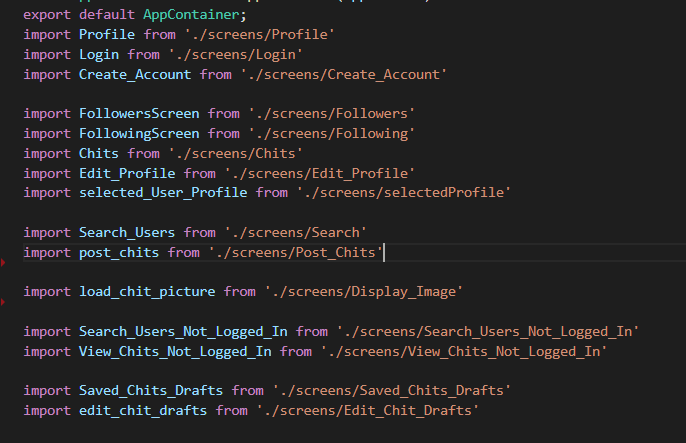


When using condition statements checking if the values of two variables are the same, instead of using ‘==’, ‘===’ was used to see if they were the had the same data type and the same value instead just the latter.

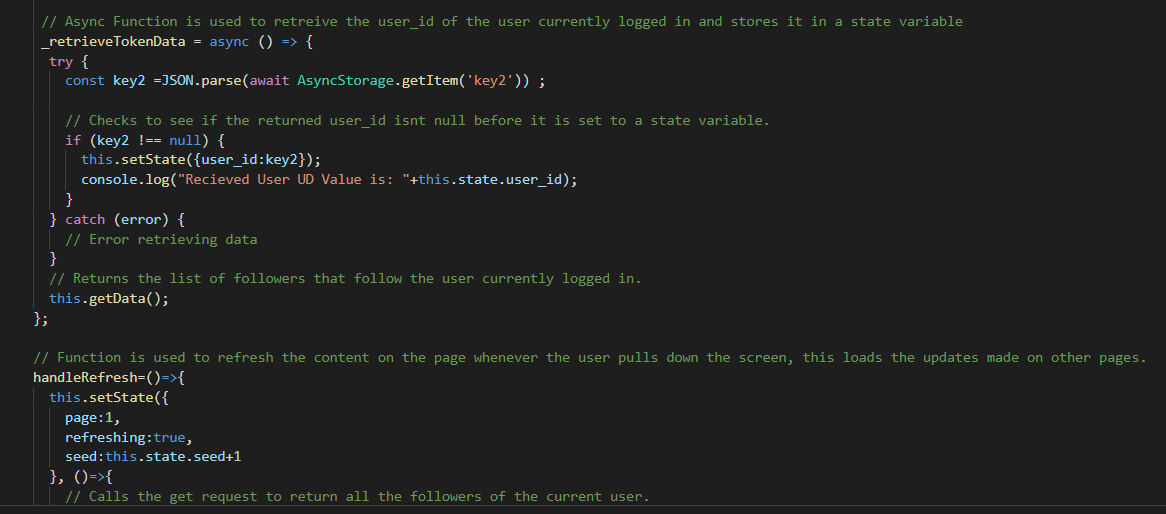


Some of the much easier coding conventions used was using tabs to indent the lines of code when required instead of used the spacebar and always ending a statement with a semicolon. I also made sure that setState() wasn’t invoked on the componentDidMount().

I mostly followed the Google JavaScript Style Guide throughout the app development to make sure the coding style was consistent, making it easy for cases where other developers get involved in the project.



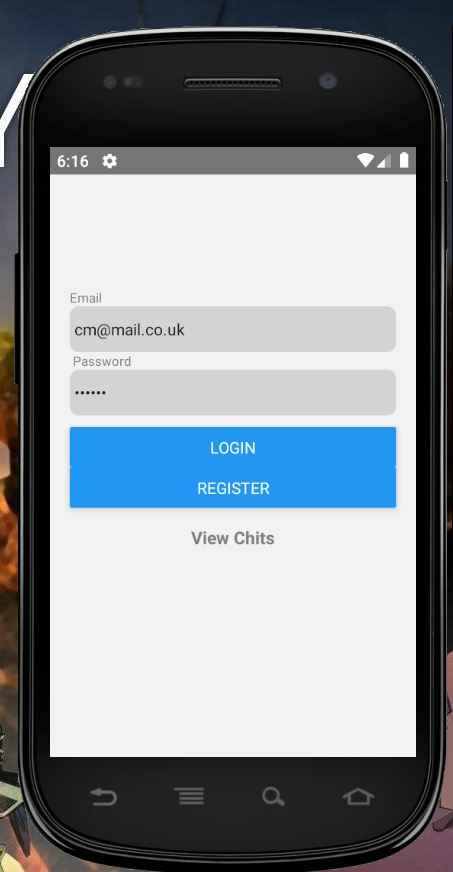
File names all started with capital letters and may include underscores (\_) or dashes (-), but no additional punctuation was used to separate the file name text and Import statements were not line wrapped.



Comments were placed on a separate line and not at the end of the line of code it is explaining, comment text began with an uppercase letter. Braces were used for all control structures even if the body only contained a single line, the control structure used the most was the if statement. Local variables were declared using either const or let, with the var keyword not being used at all.

## What testing has been carried out

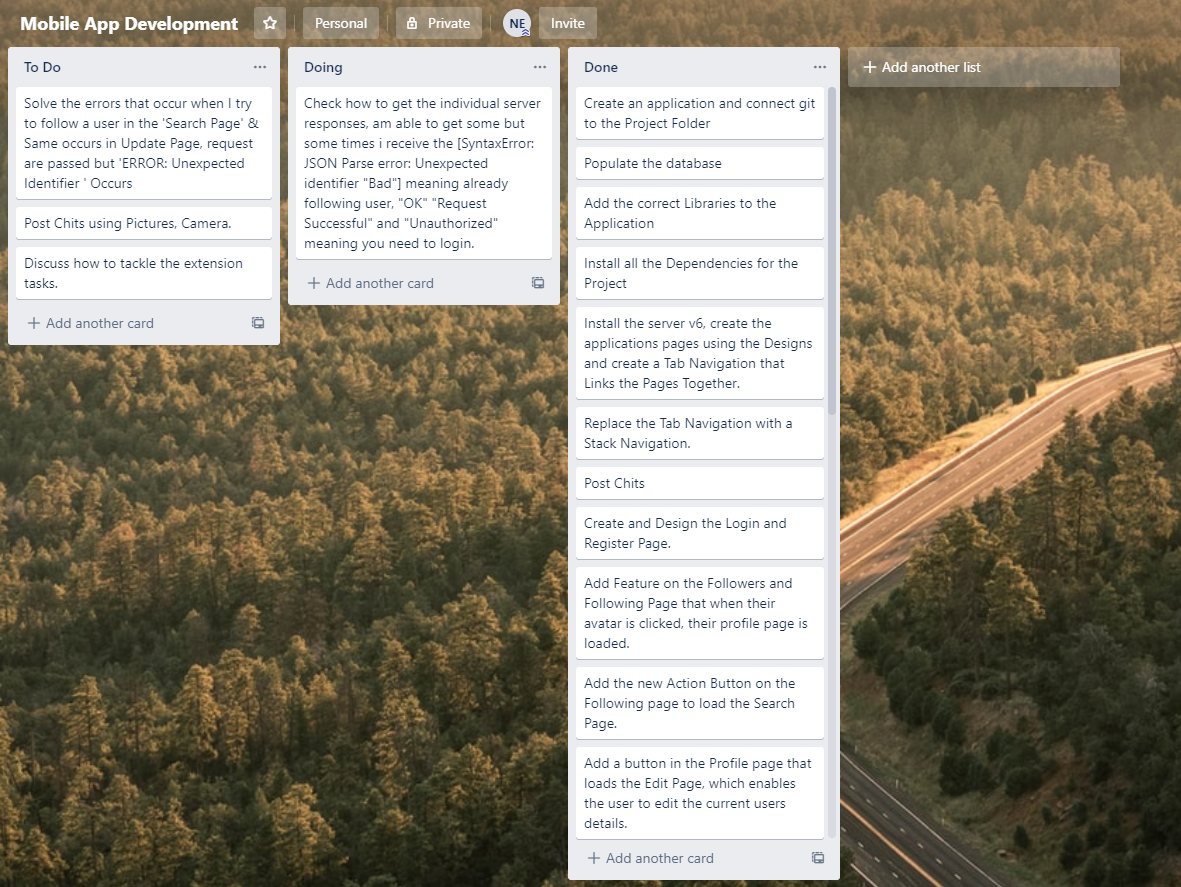
The first type of testing done on the app were input test, it was carried out on the Login page to see whether the user could gain access to the main app using a correct combination of email and password. If the email or password are incorrect, then access isn’t given, however if they are correct, then the Chits page is loaded.



Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Author** | **Function** | **Parameters** | **Expected** | **Actual result** |
| 1 | Charles Modi | logIn() | No parameters or invalid password/ Email | 400 – No access given, please input correct Username and password. | 400 – No access given |
| 2 | Charles | Login() | Correct Username and Password. | 201- OK: Profile Page is loaded. | 201- Access Given to User and Chits page is loaded.  Token and User ID returned. |
| 3 | Charles | attemptLogin  () | No parameters | 400 – Account not created. | 400- ‘Account not created, please try again’. |
| 4 | Charles | attemptLogin  () | All account details entered correctly | Response 200 - Account created. | Response 200: Account created, Please Login. |
| 5 | Charles | getData  () | User\_ID | 200- User Info found and returned. | 200- User Info found and returned but its for the wrong user. |
| 6 | Charles | getData  () | User\_ID | 200- User Info found and returned. | 200- Correct user found and details are returned. |
| 7 | Charles | LogOut() | No Token or Incorrect Token | 400- Unauthorized, please logIn. | 400 - Unauthorized, user isn’t logged out. |
| 8 | Charles | LogOut() | Correct Token | 200 – OK: User is logged out. | 200 – OK: User is logged out and login page is loaded. |
| 9 | Charles | updateProfile() | User\_ID and Token | 200 – OK: User Profile Updated. | 200 – OK: User profile is updated. |
| 10 | Charles | updateProfile() | User\_ID and Token | 200 – OK: User Profile Updated. | 200 – OK: Profile updated but the user ID belonged to the wrong user, so different profile is updated. |
| 11 | Charles | updateProfile() | User\_ID | 401- Unauthorized, profile isn’t updated. | 401- User Profile isn’t updated. |
| 12 | Charles | getProfilePicture() | Image url with User\_ID | 200: Profile Picture found and displayed. | 200: Profile Picture found and Displayed. |
| 13 | Charles | handleChoosePhoto  () | Token and Image | 200: Profile Picture added. | 200: Profile Picture added. |
| 13 | Charles | handleChoosePhoto  () | Image and no Token | 401: Profile Picture not added. | 401: Unauthorised request, need to be logged in. |
| 14 | Charles | postChit() | No Parameters | 200 – Chit Posted. | 400: Location permissions denied, and chit isn’t posted. |
| 15 | Charles | postChit() | No Parameters | 200 – Chit Posted. | 200: Location permissions accepted, and chit posted. |
| 16 | Charles | getChits() | No Parameters | 200 – Chit retrieved | 200- Chit Retrieved. |
| 17 | Charles | postImageChit  () | User\_ID, Token and Image URL | 200 – Photo Chit Posted. | 200 – Photo Chit Posted. |
| 18 | Charles | UnFollow  () | User\_ID, Token | 200 – Unfollows the user. | 200 – Unfollows the user and deleted user is removed from the Flat list. |
| 19 | Charles | followUser  () | User\_ID and Token | 200 – Now Following ‘Users Name’. | 200 – Successful, now following the user. |

## How the project was managed/what tools were used for organisation?



I used Trello to manage the project using the Kanban Methodology, I chose this because it was simple to understand and easy to set up on Trello. The Trello boards allowed me to make tabs of all the task that needed to be completed, which reduces time being wasted, overall improving the production rate.

I initially broke down the project requirements into individual tasks to complete, they started off as small and simple task to complete such, creating the first wireframe design and setting the environment up before the coding could begin. This was set up so that once I had completed these simple tasks, I would build up momentum and get more tasks completed in a short period of time. As time progressed, the tasks remaining on the To-Do and Doing list became more challenging and required more time and research to complete. By organising the tasks at the start based on difficulty, it allowed me time to complete the easier tasks quickly and fully focus on the more difficult remaining tasks.

The layout was simple but effective, by having a list of all the tasks that need to be complete and a list of the tasks that I am in progress of completing, this allowed me to attempt the tasks one at a time and flexibly interchange between the tasks am currently doing. All the other tasks were left aside until either a task is complete, or a problem occurs that prevents progress being made. In most cases, some tasks took too long to complete, which halted progress, this methodology is very flexible giving me the freedom to move onto another task and making a note of the progress made on the previous task. As time progressed, while completing some tasks, I sometimes found out that the current tasks had additional tasks that needed to be completed, this methodology allowed additional tasks to be added onto the To-Do list without causing any interruptions. The Trello Board allowed me to keep track of what tasks I was currently doing by putting them on the Doing List, it made it easier to continue from where I had previously left off and no time was wasted on thinking of tasks to complete. Once a task is completed, it is moved onto the Completed List, the To-Do list gives you freedom to pick the tasks to incorporate into the product, which kept a continuous flow of work and made sure that progress was made, and no time was wasted.

Kanban not only consist of manual guidelines or cards but also visualizations of the process outputs which makes the review of work easier. This could also highlight other potential problematic areas where additional attention is needed. The system is a very responsive and does not promote any lags or delays. As the tasks are continuously shifted between the columns of the Kanban cards, it automatically highlights the areas where any limiting factors are raised that could hold up the overall output which can be responded to as soon as possible by shifting and switching the resources from the other tasks.