**Mobile Application Development Assignment 2**

Implement the Chittr API as a React Native application

[**Wireframes:** 2](#_Toc35469625)

[Design 1: 2](#_Toc35469626)

[Design 2: 6](#_Toc35469627)

[**README File:** 13](#_Toc35469628)

[**Version Control Software:** 14](#_Toc35469629)

[**Code Style Guide:** 15](#_Toc35469630)

[**Testing:** 16](#_Toc35469631)

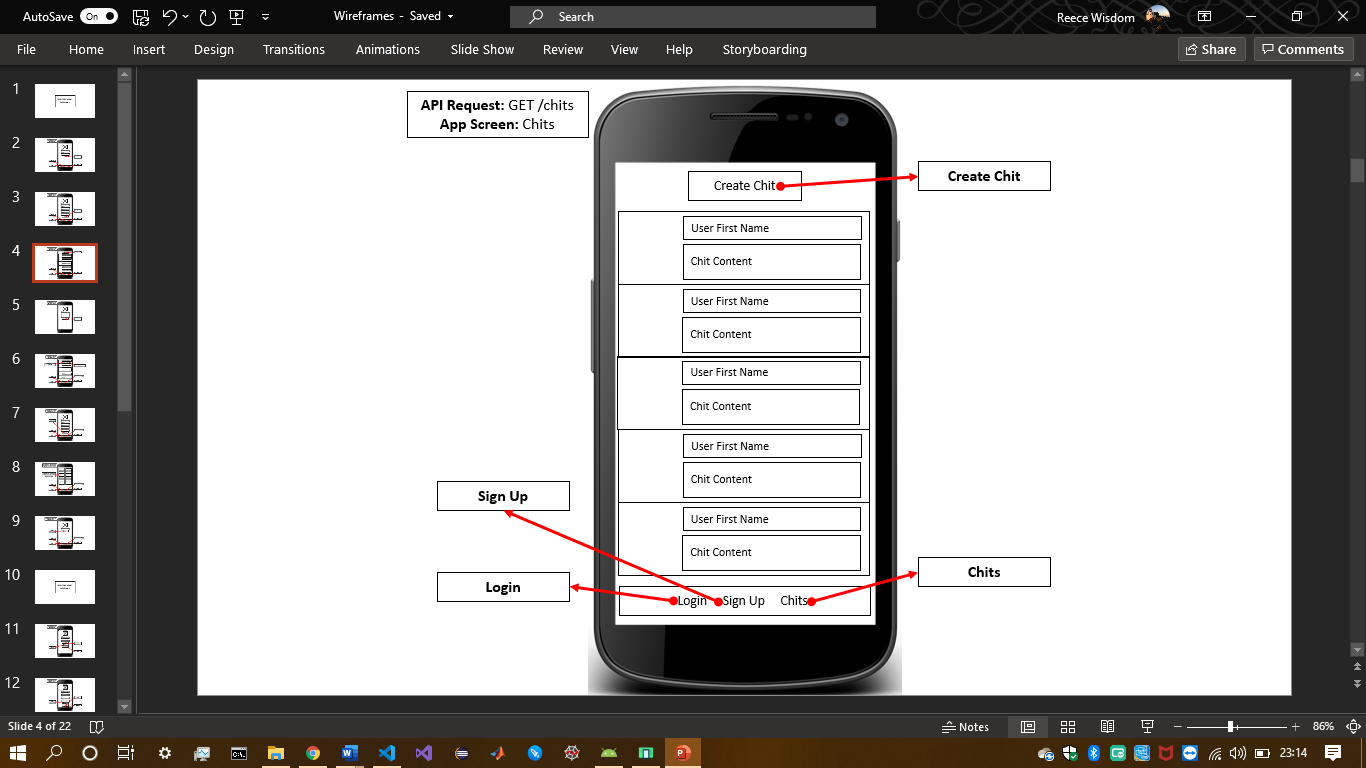
[**Project Management:** 17](#_Toc35469632)

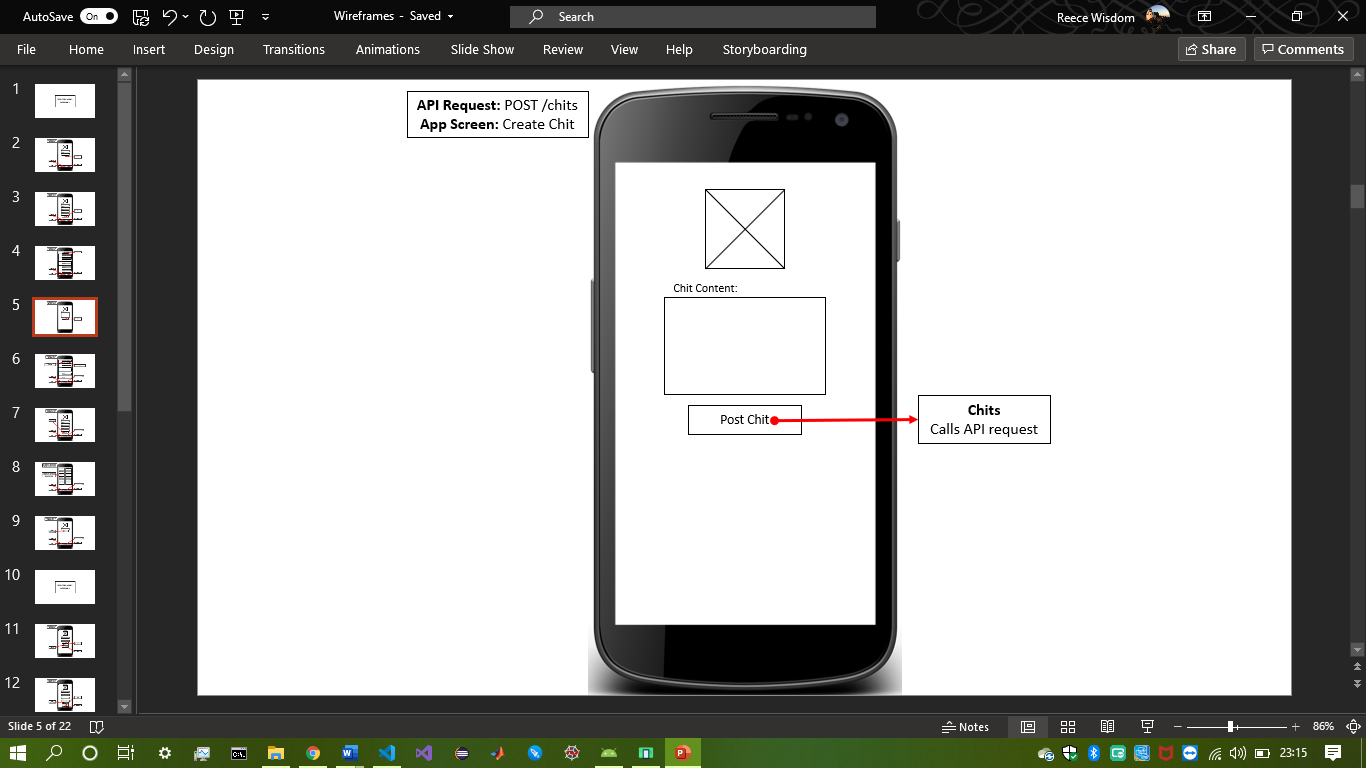
**Name:** Reece Wisdom

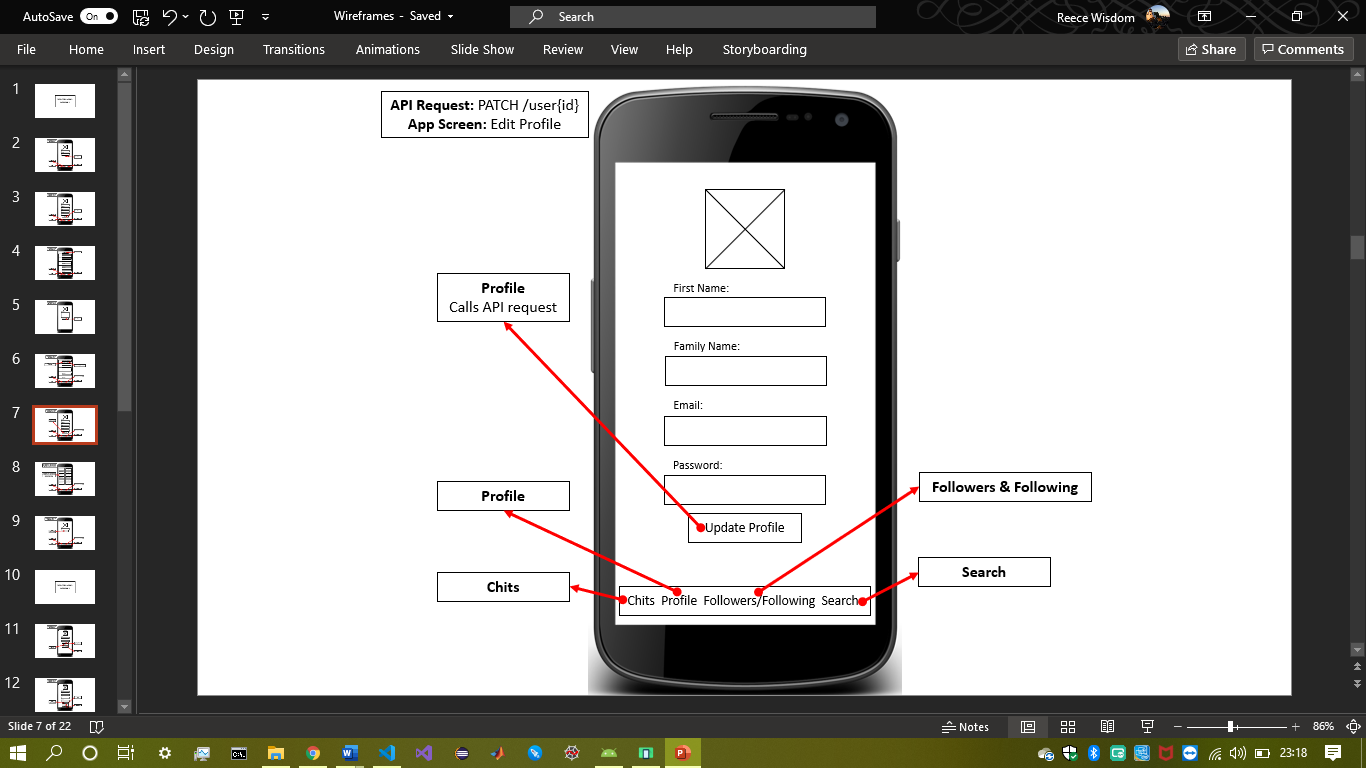
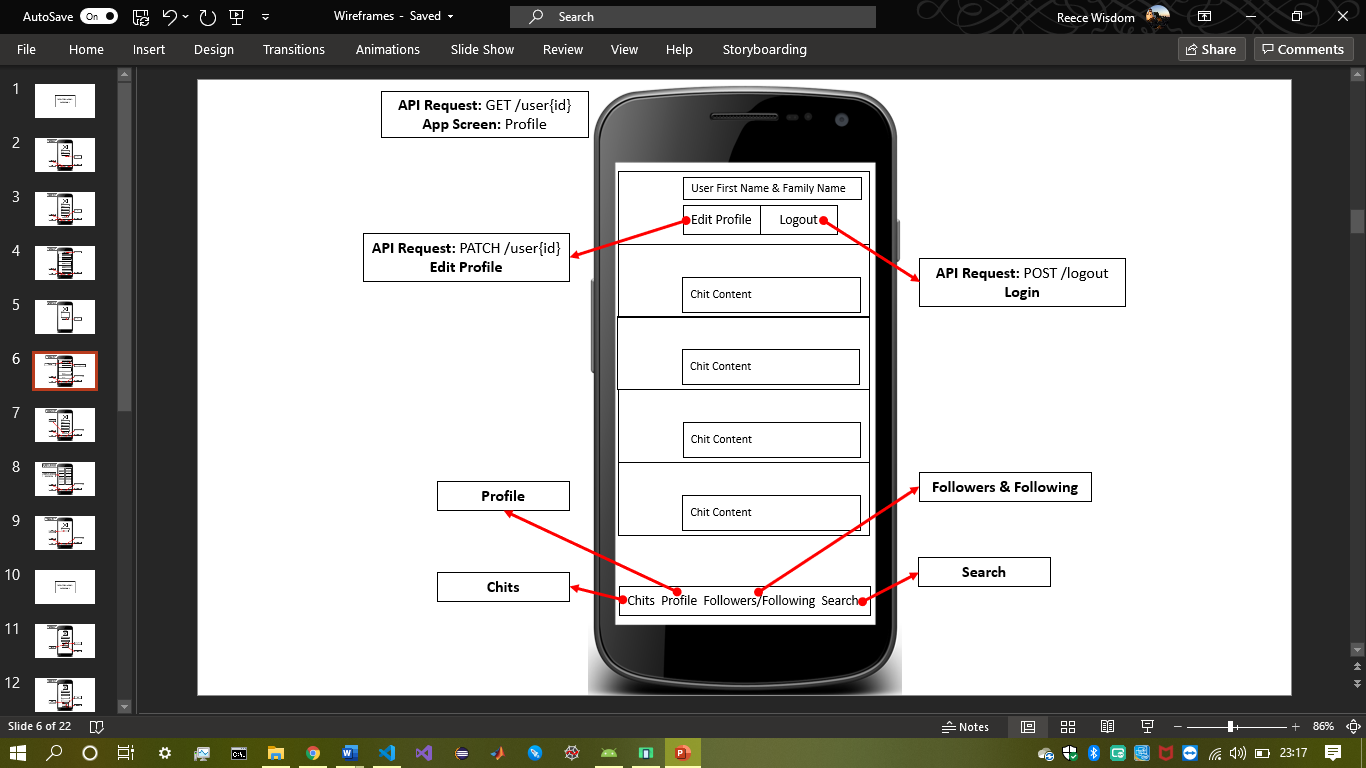
**Student ID:** 17033849

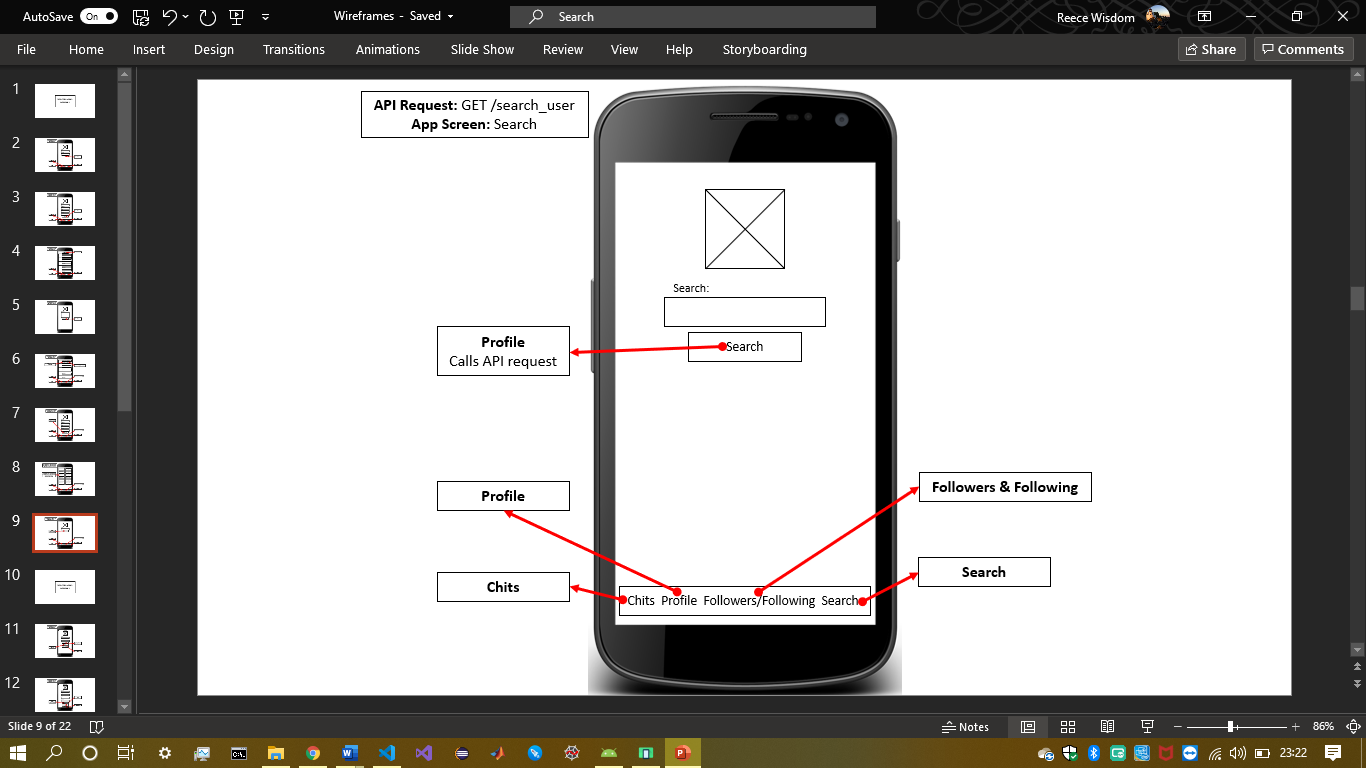
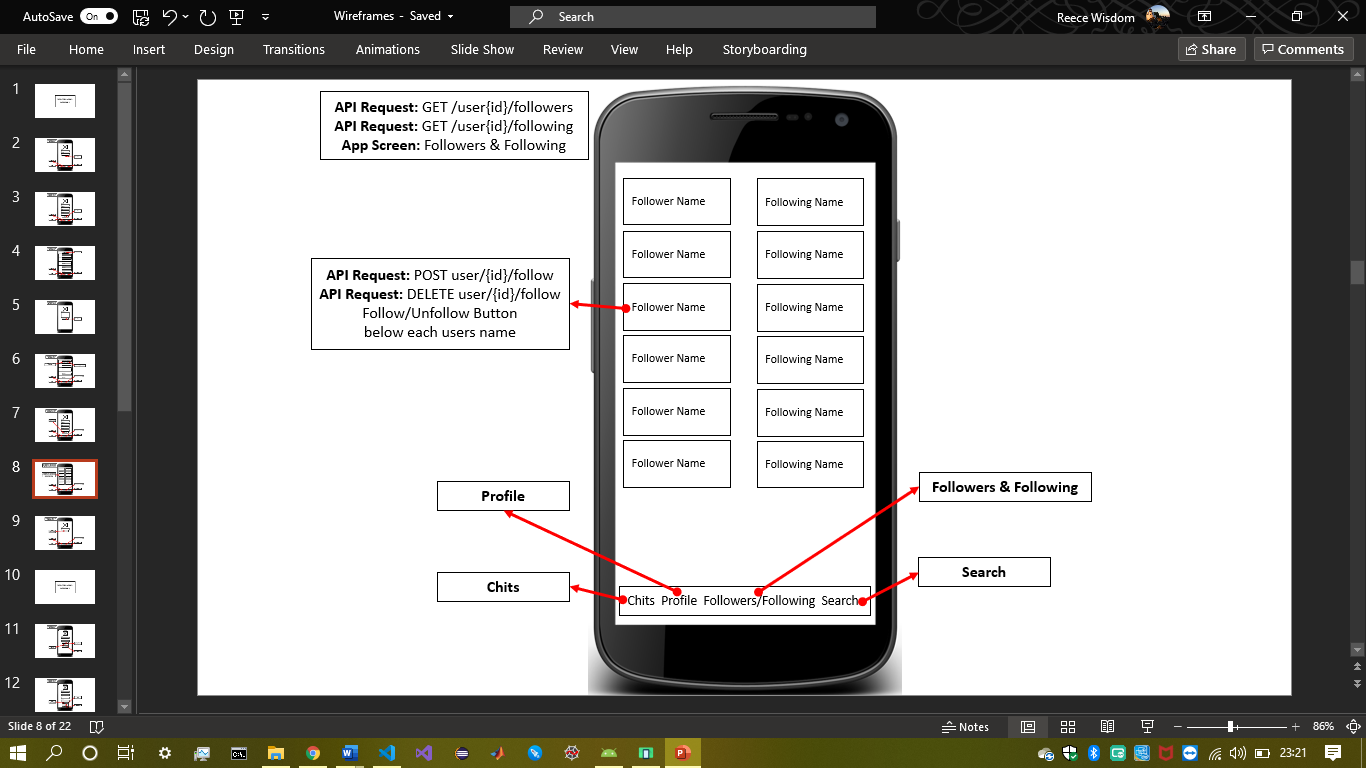
## **Wireframes:**

### Design 1:

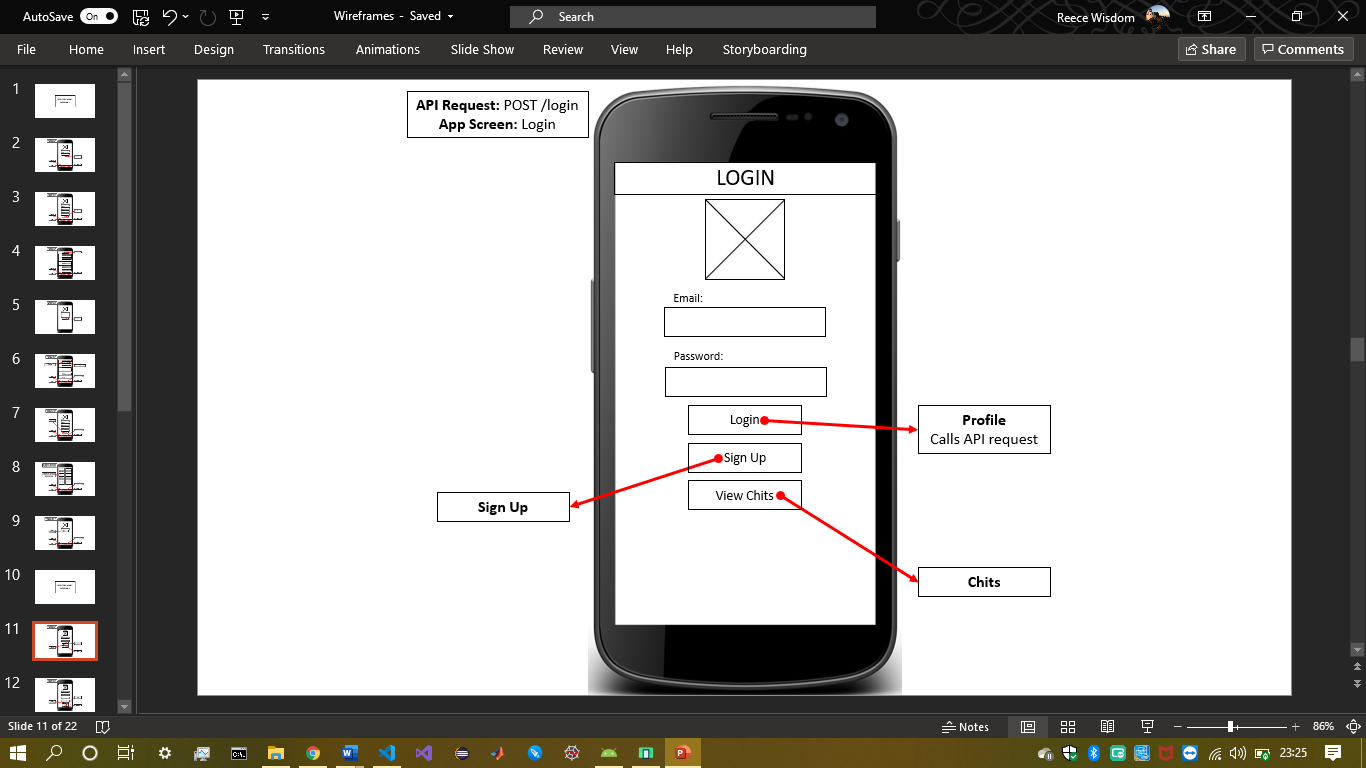
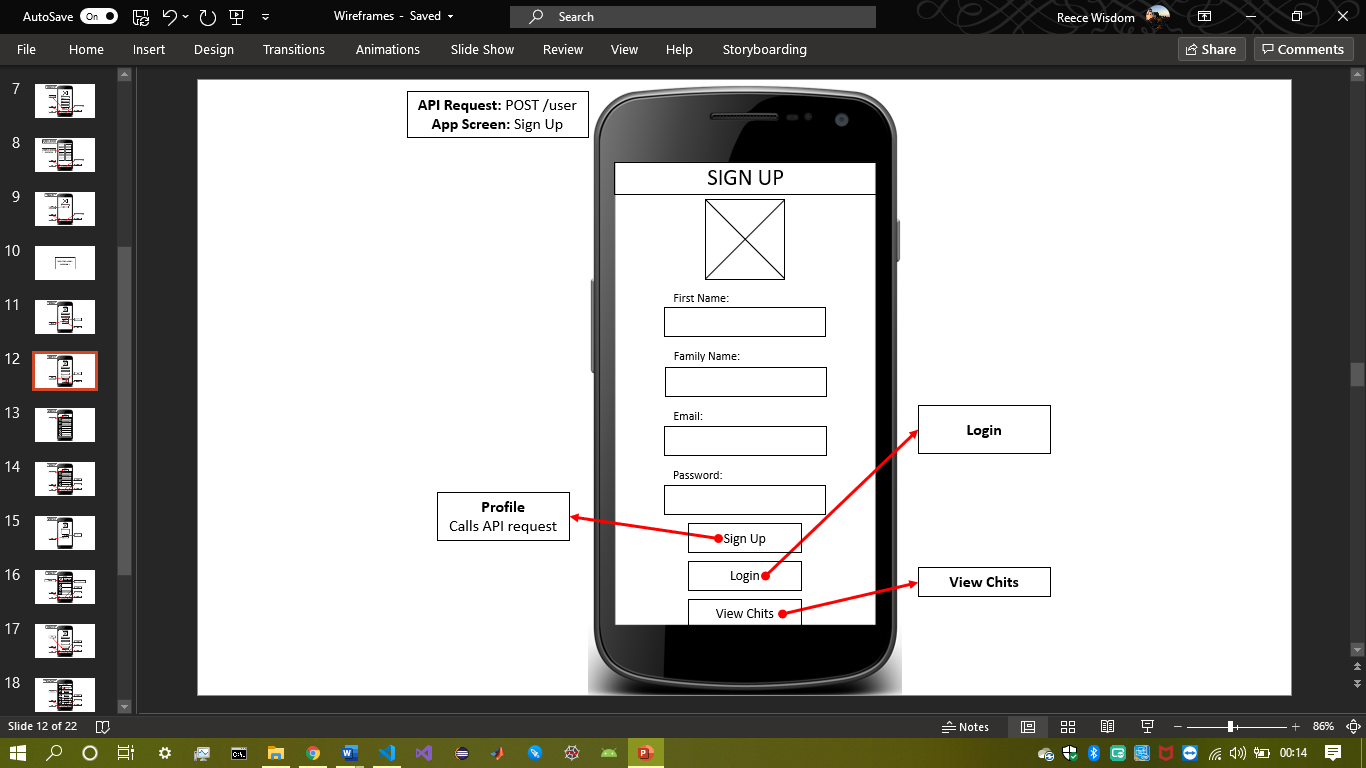


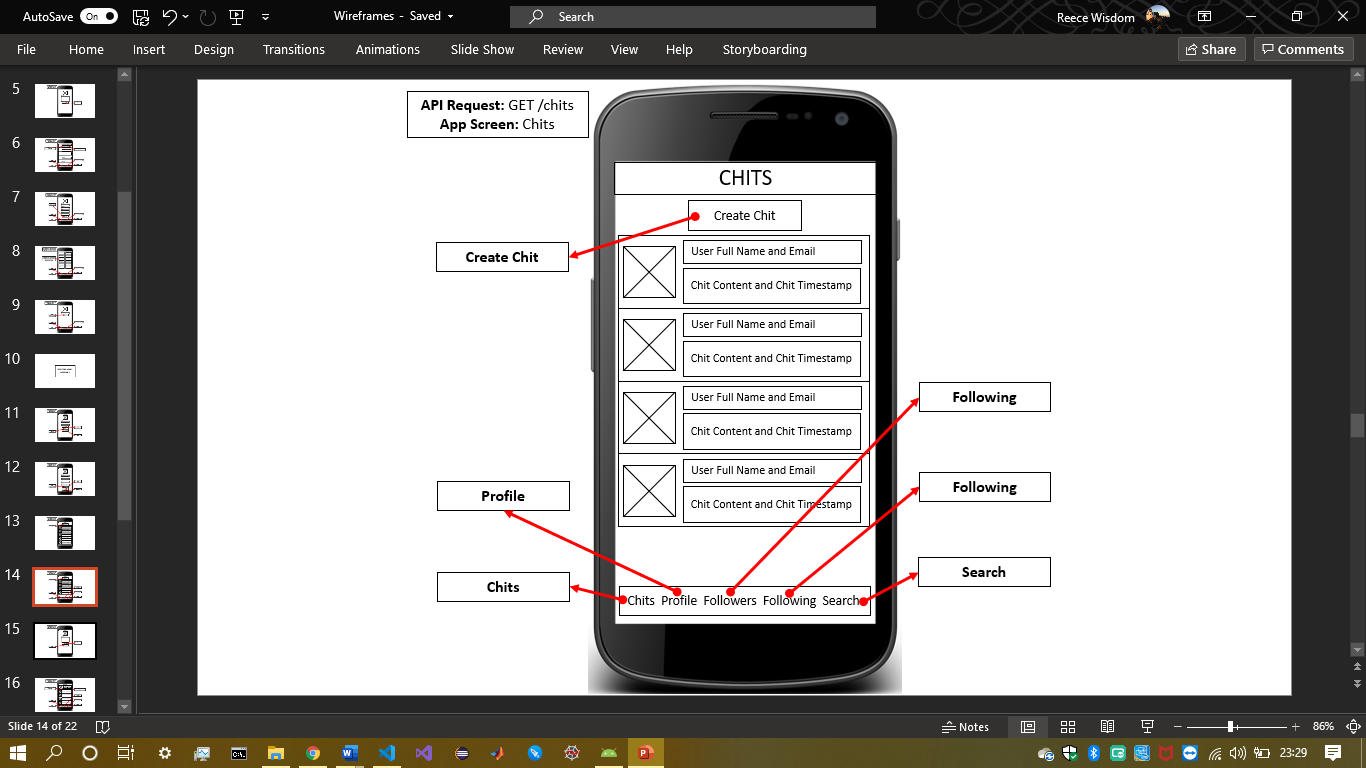
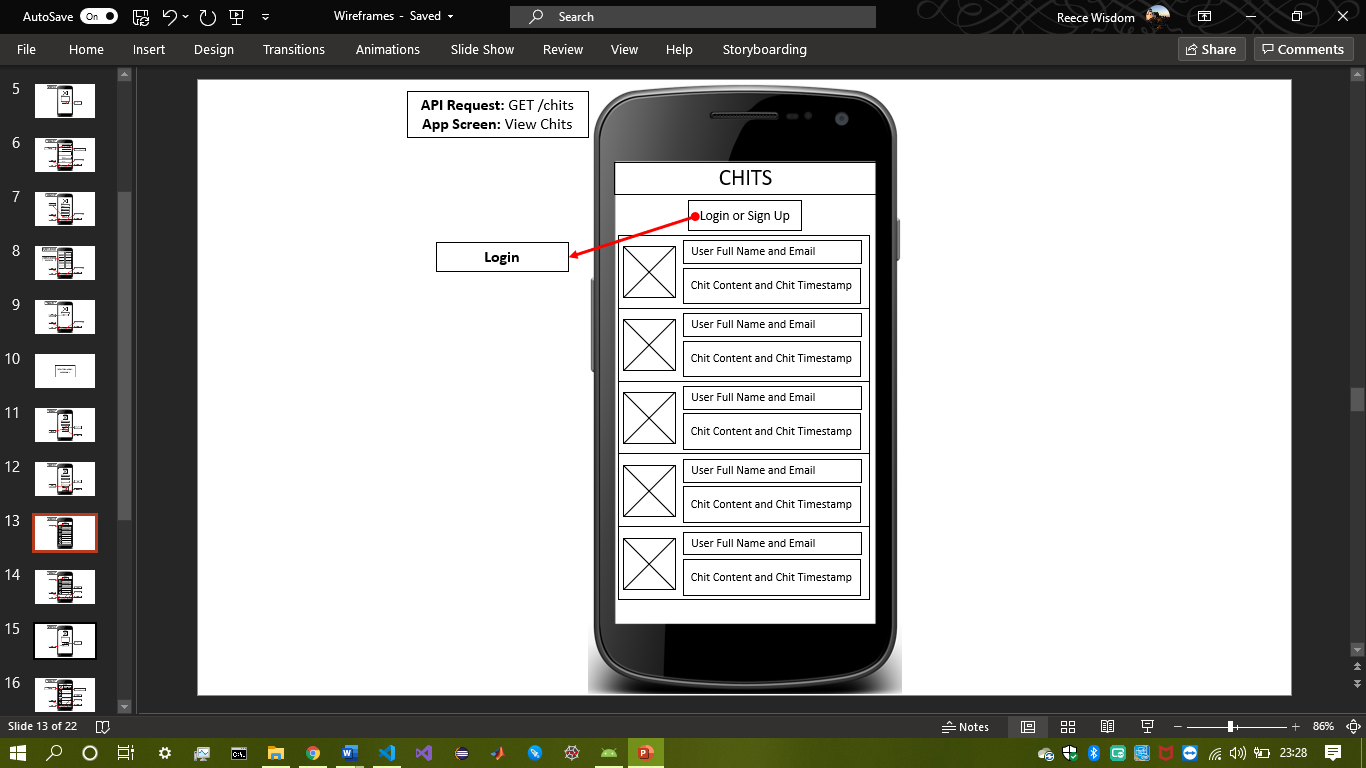


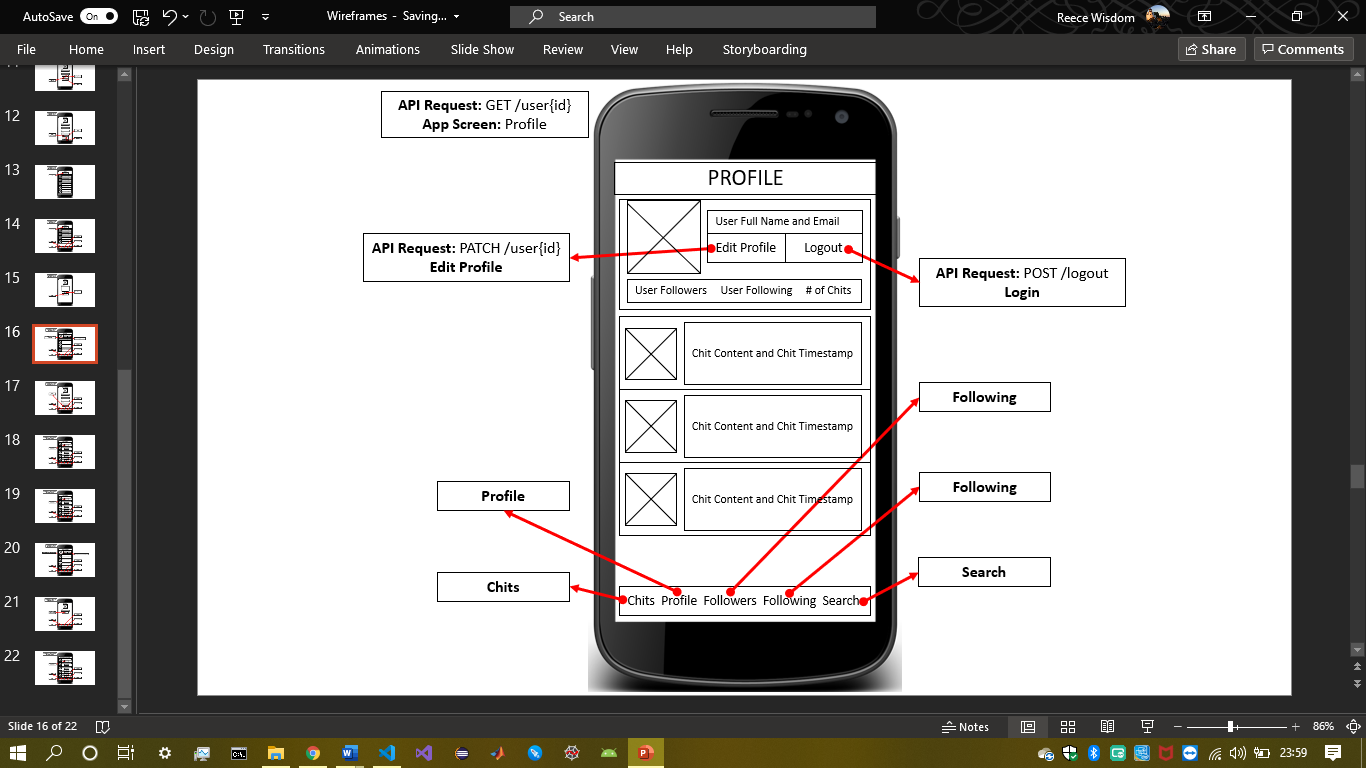
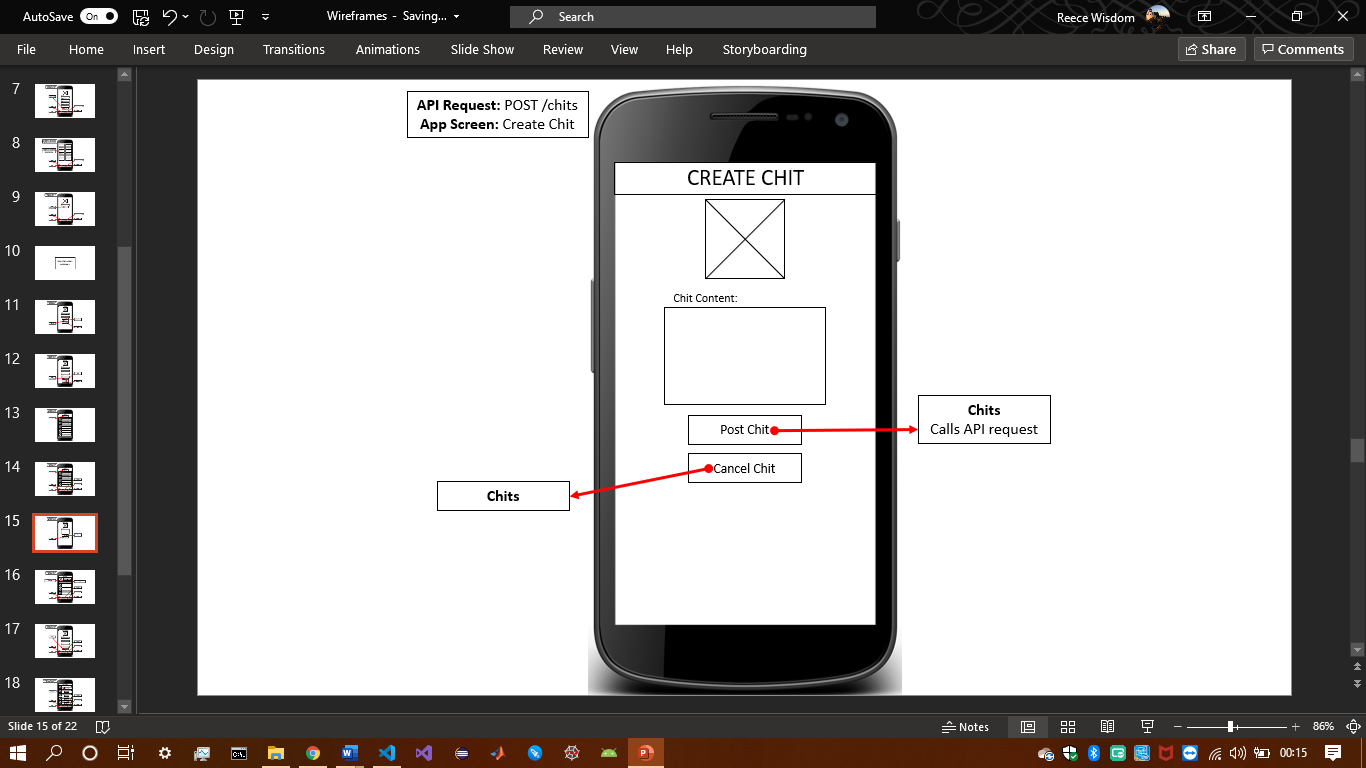


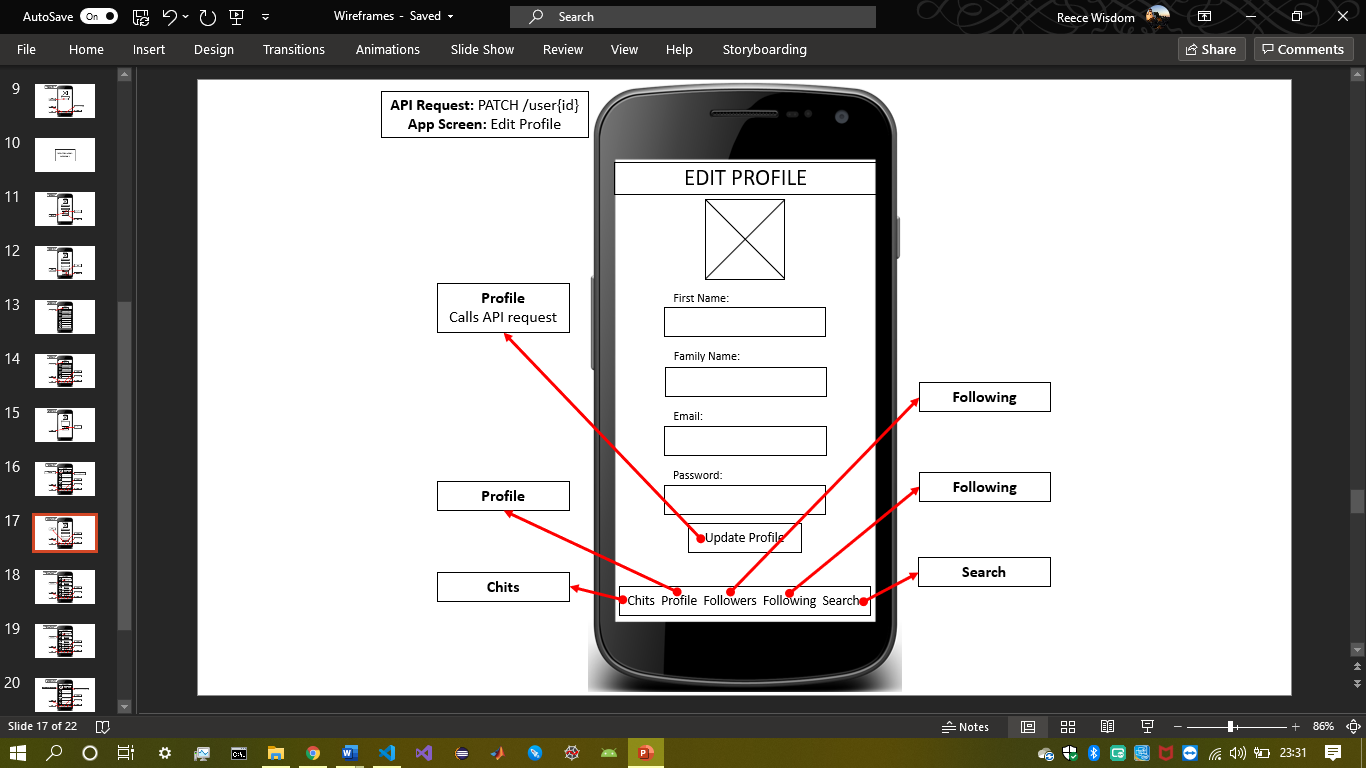
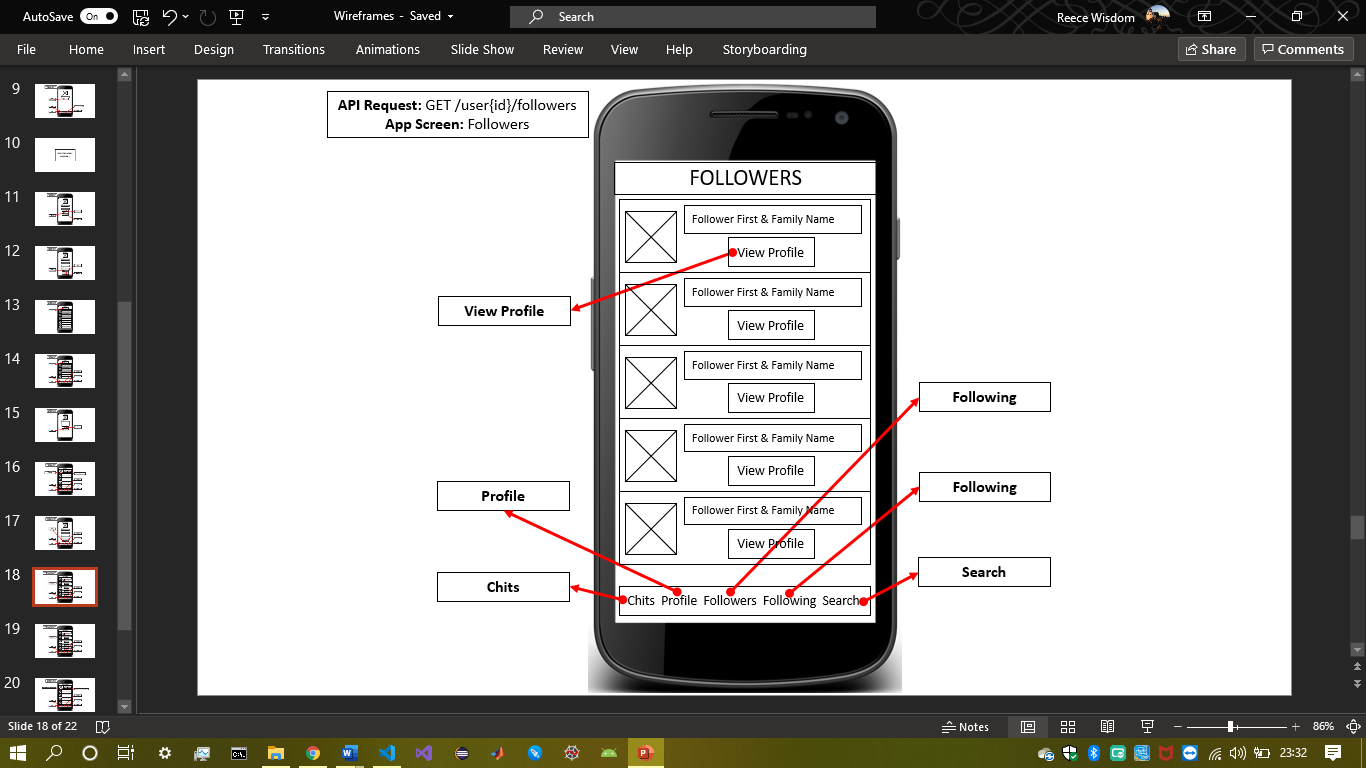


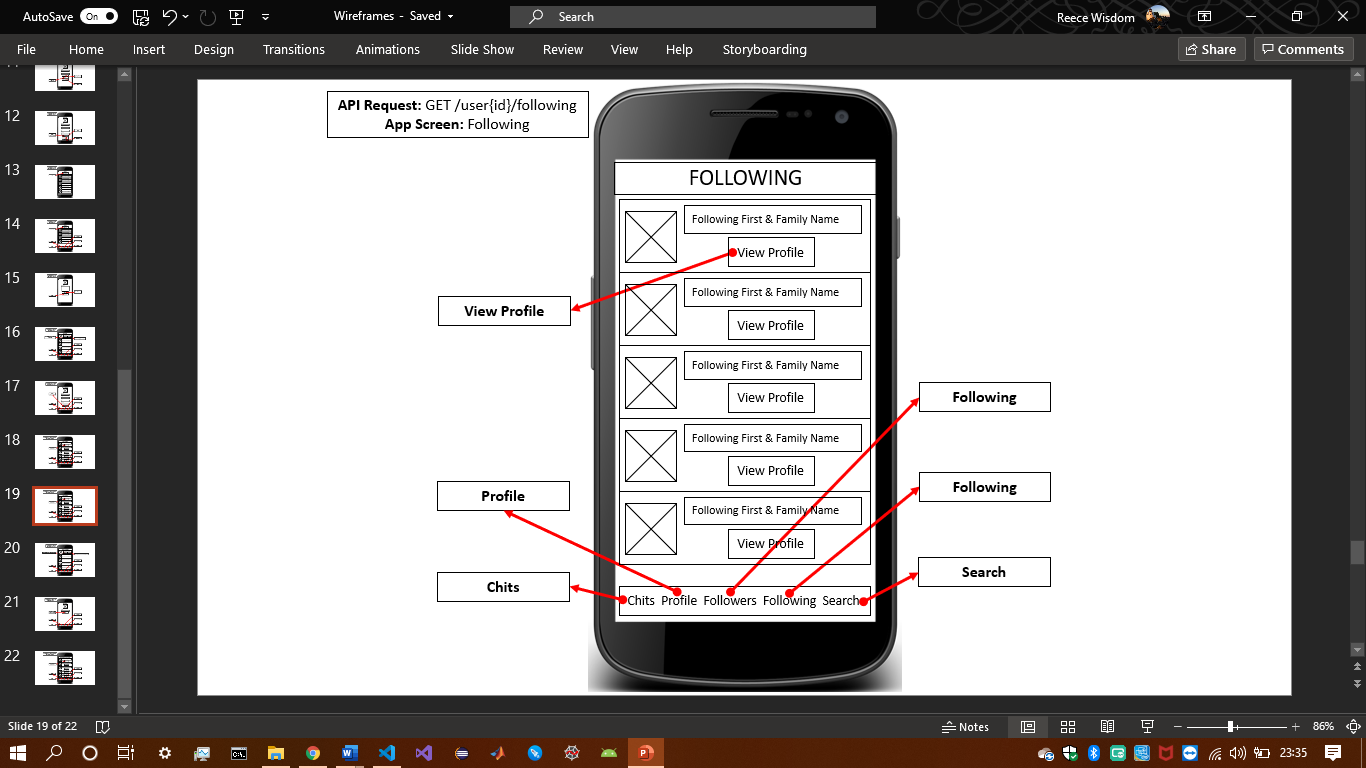
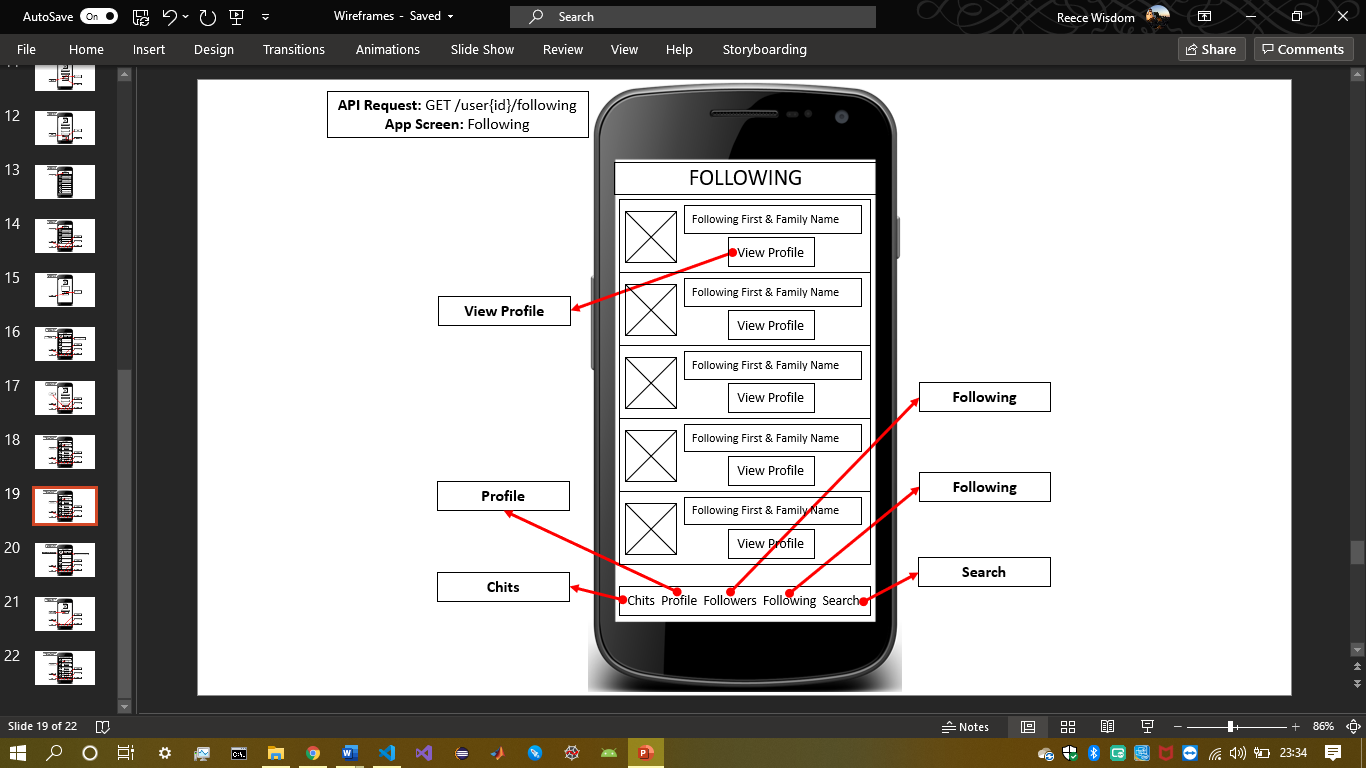
### Design 2:

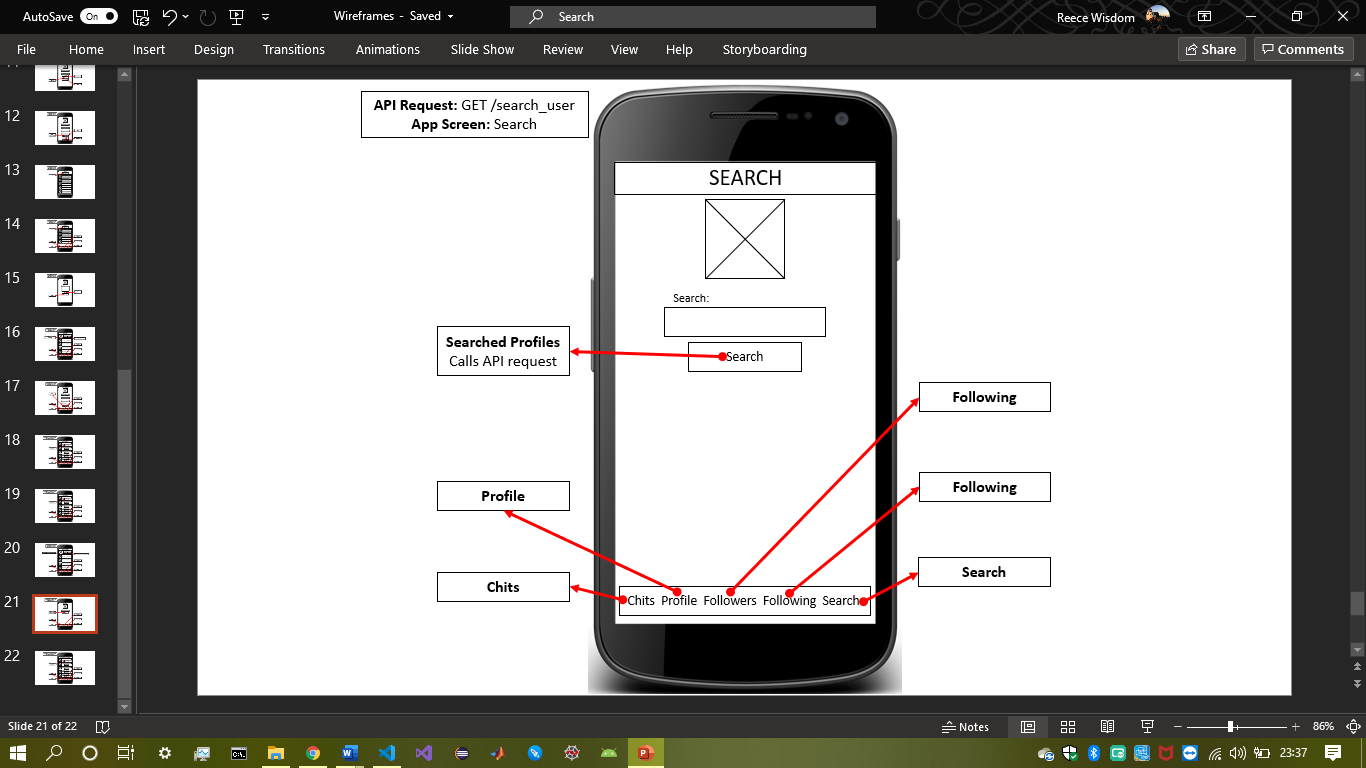
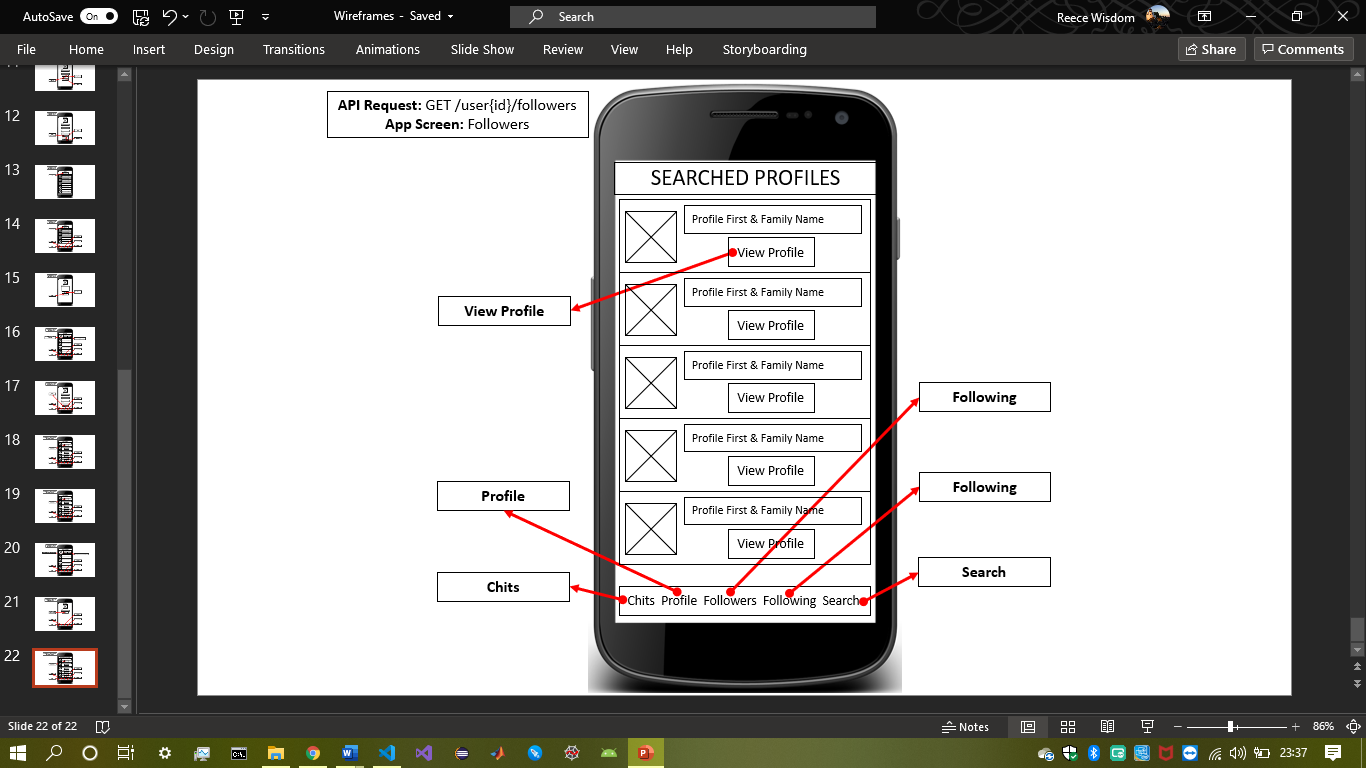
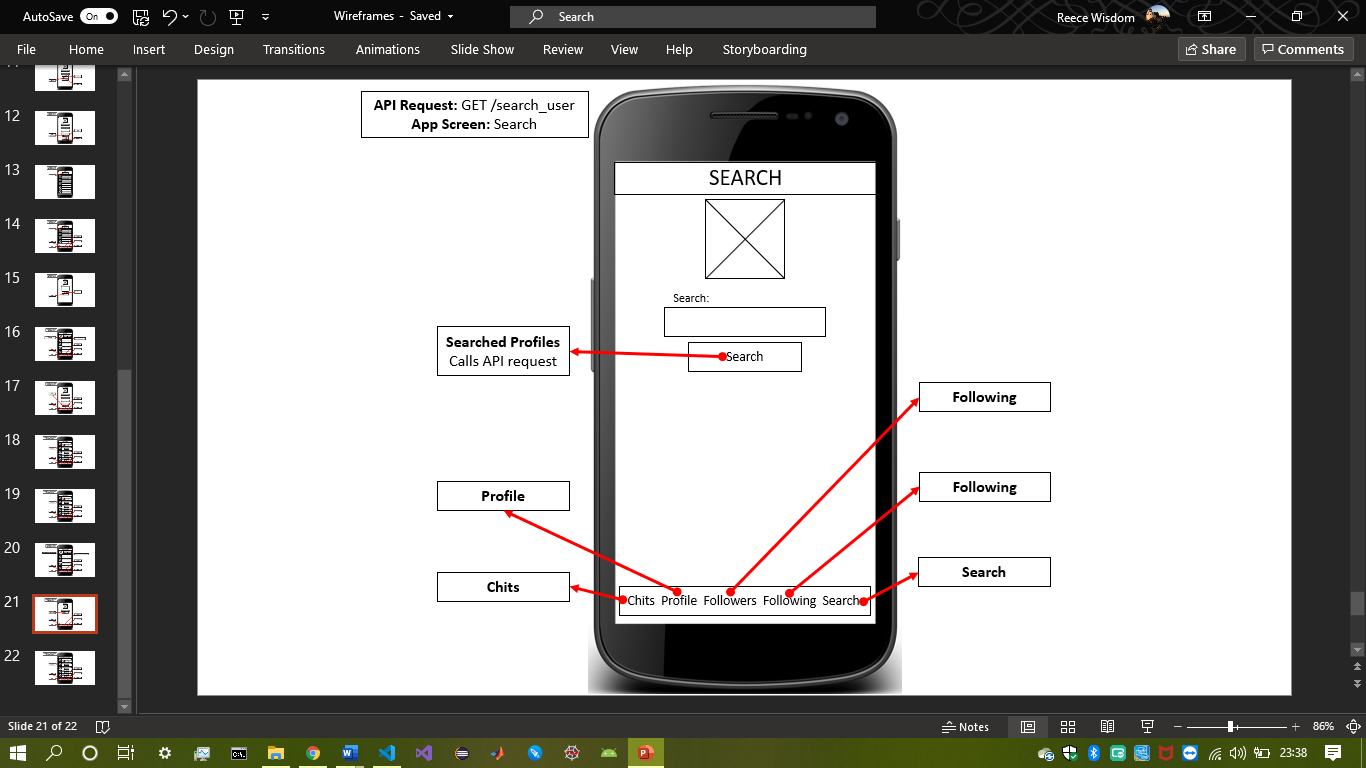










The wireframes were created to initially sketch out my Chit mobile application’s design, flow and layout. For my application I decided to choose the second iteration for many different reasons. Firstly, within the second wireframing attempt, headers were integrated for each unique screen. This is an advantage over the first iteration of wireframes as this implementation made the application more user-friendly as it provides more context to distinguish individual screens and it also better resemble similar real-world products/apps, e.g. ‘Instagram’ and ‘Twitter’.

Furthermore, another reason for selecting the second set over the first set is that the ‘View Chits’ screen was added. This is advantageous as this screen allows users to view Chits without logging into an account or signing up.

Additionally, within the first set of wireframes the ‘Search’ screen’s API call would direct the user to the profile of the user searched for. However, within the later set of wireframes, a ‘Searched Profiles’ screen was added as searching for a user has the possibility of returning more than one user. Therefore, this screen shows a list of all the users and their profile can be viewed from that list via a ‘View Profile’ button.

Other additional advantageous differences include:

* The addition of the current profile’s email address, current followers and following number and number of chits for a user on the ‘Profile’ and ‘View Profile’ screen to provide more detail of a user’s profile.
* Splitting the ‘Followers & Following’ screen into their own individual screens to make the data easier to read (e.g. not confusing the followers and following lists.)
* ‘Cancel Chit’ button in the ‘Create Chit’ screen as there was not way to navigate back to the other pages without posting a chit prior to its implementation.
* Adding a timestamp for each chit posted by a user so the time the chit was posted can be seen and adding a default user image for each chit posted to make the app more aesthetically pleasing by adding colour, imagery and reducing whitespace.

## **Version Control Software:**

The existing and recognised version control method for managing my project is was uploading to ‘GitHub’ via ‘GitHub Desktop’.

**A screenshot of a social media post

Description automatically generated**

## **Code Style Guide:**

The JavaScript style guide I implemented throughout the design of my React Native application was the W3Schools JavaScript style guide: <https://www.w3schools.com/js/js_conventions.asp>

For example, *camelCase* for variable names in hand with starting each variable name with a letter, e.g. *‘chitList’, ‘followerList’*, etc. However, variables which were used to store JSON responses from the API requests were named after their corresponding JSON name to avoid confusion, e.g. *‘given\_name’* was named after the *‘given\_name’* JSON response to an API request.

Other implementations from the style guide included, but are not limited to:

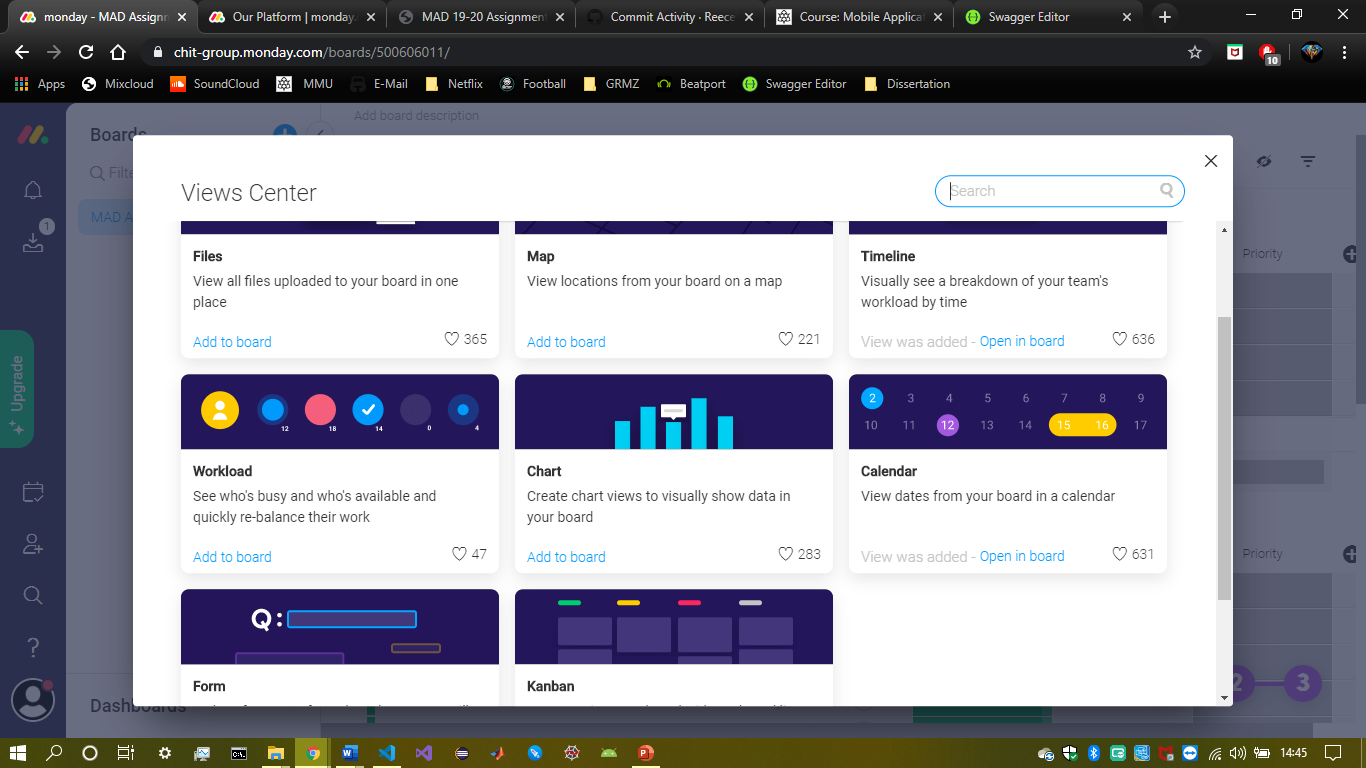
* Spacing around operators
* 2 space code indentation
* Complex compound statements have opening brackets at the end of the first line
* Closing brackets occur on a new line
* Complex statements are not ended with a semicolon
* Opening brackets are on the same line as the object name
* Quotes around string values
* Avoiding hyphens in naming

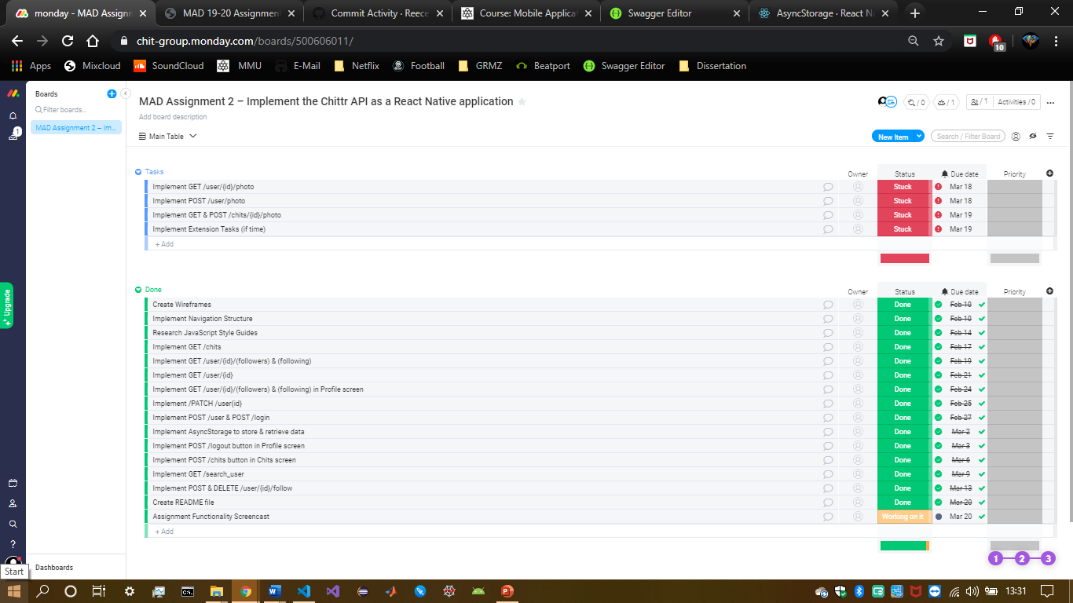
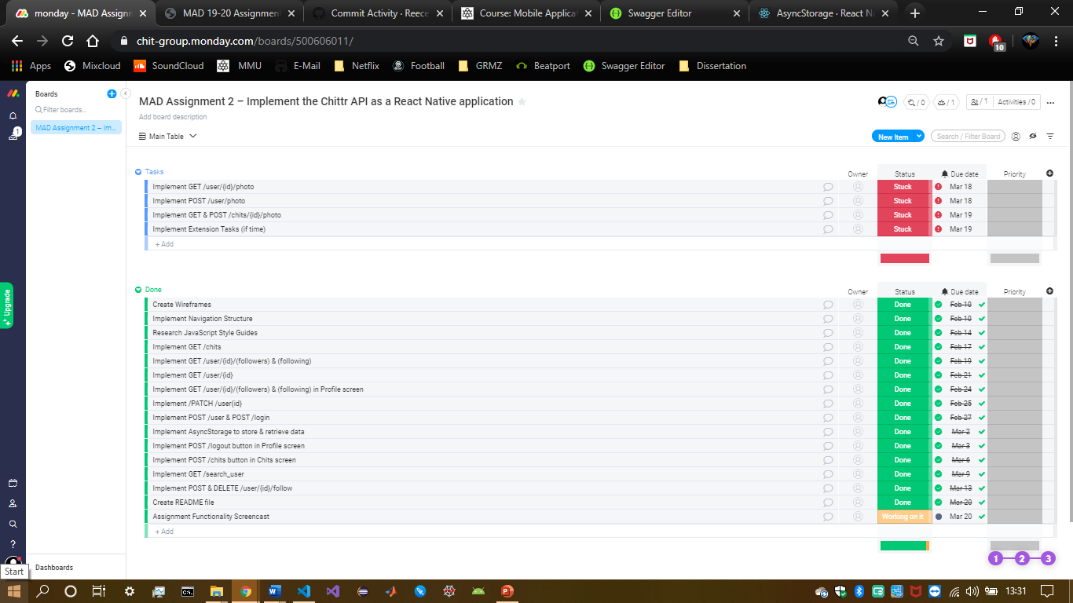
## **Testing:**

Don’t know how to test the React Native project using ‘Jest’.

## **Project Management:**

For the management of my project I used the work operating system *‘Monday.com’* to organise the required tasks to develop a functioning mobile application using React Native. This tool was chosen as it is a free to use software; therefore, easily accessible and unlike other project management tools, *Monday.com* allows users to display the table of tasks in different methodologies. For example, as shown below, tasks could be displayed as:

*Table* *– File – Map – Timeline – Workload – Chart – Calendar – Form – Kanban*



Furthermore, and additional reason for using this form of project management is because for each task integrated into the table, update messages and notes can be added. In my case, I used the ‘Info Boxes’ tab to add notes (‘Add a Note’ tab) for some of the tasks to remind me on the steps I should take to complete the task. For example, when implementing GET ‘/user{id}’, GET ‘/user/{id}/followers’, GET ‘/user/{id}/following’ and PATCH ‘/user/{id}, the {id} variable was hardcoded in to begin with until ‘AsyncStorage’ was successfully implementing to store the ‘user\_id’ variable retrieved from the ‘Login’ screen.

