**COMPX318 - Android Project Report**

Members:

Bianca Craill 1345170

Daniel Wilson 1345359

Ella Cai 1516922

Kelvin David 1345360

*Project Brief: Develop an Android App to allow people to coordinate small group meetings. Specifically, this app should allow members of the University Pokemon Go Group to coordinate raids.*

**Features/User Interface:**

*Interface 01 - Dashboard (Fig. 1)*

Our app will open with a dashboard displaying some key information and a feed of posts about upcoming raids. These raid posts will display the following information:

* + Star Rating (difficulty) of the Pokemon
  + Location/Gym
  + Starting/Hatching Time
  + Amount of total users following the raid (interested & confirmed)
  + Whether the raid is proposed or confirmed to go ahead

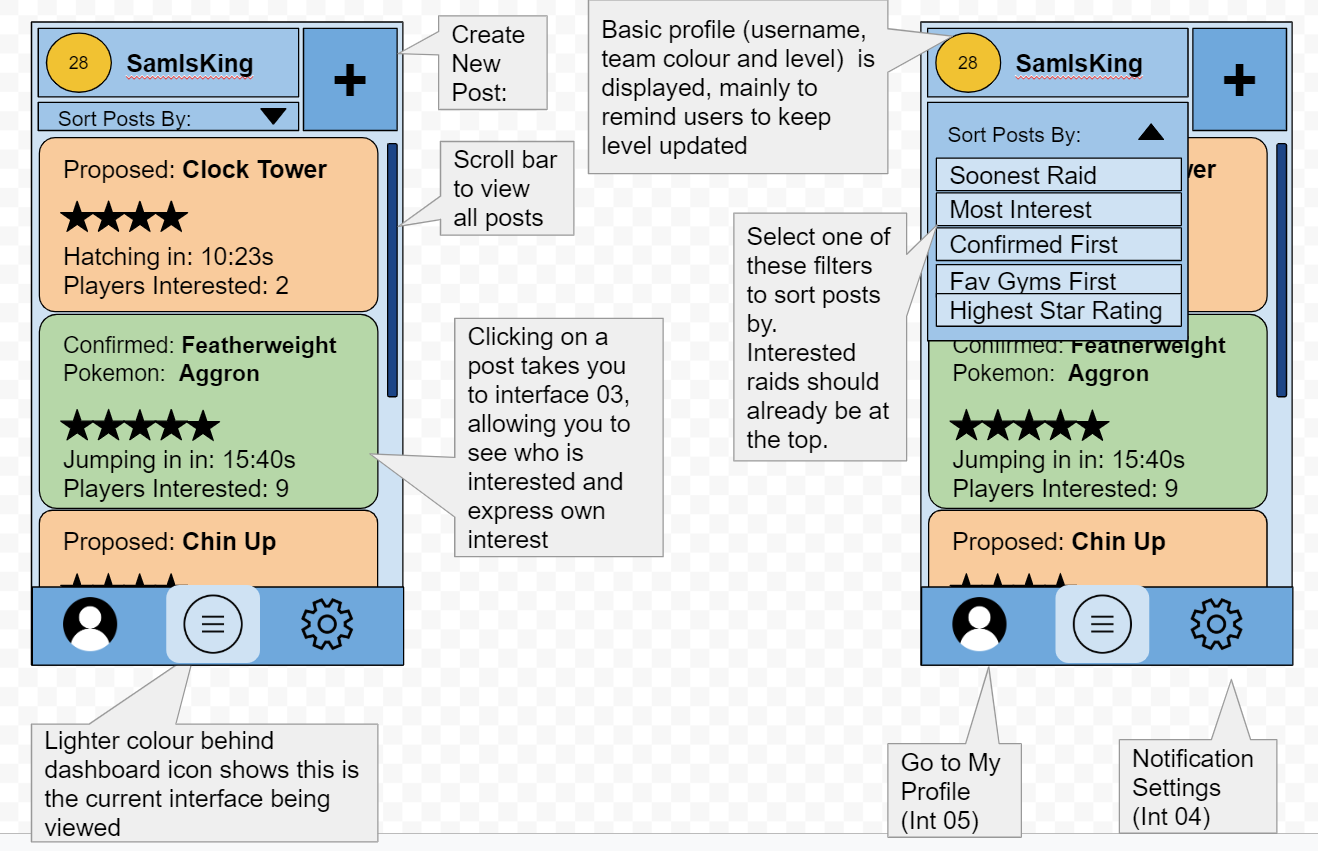
A filter functionality was planned to be implemented for the user to sort the posts by different criteria:

* + User’s own Interest expressed raids shown first by default
  + Favourite Gym(s)
  + Soonest Starting Time
  + Number of Players Expressing Interest
  + Star Rating of the Pokemon (raid difficulty)

However, due to time constraints this additional feature wasn’t prioritised for implementation.

At the bottom of the display are three buttons. Each takes the user to a different interface. The central one will take the user back to this dashboard from elsewhere in the app. The left button will take the user to a page where they can view their profile information (Interface 05). The right button takes the user to a selection of settings where they can alter preferences (Interface 04).

At the top right is a button with a plus symbol on it. This will take the user to a form needing information inputted to create a new post for other users to see (Interface 2).



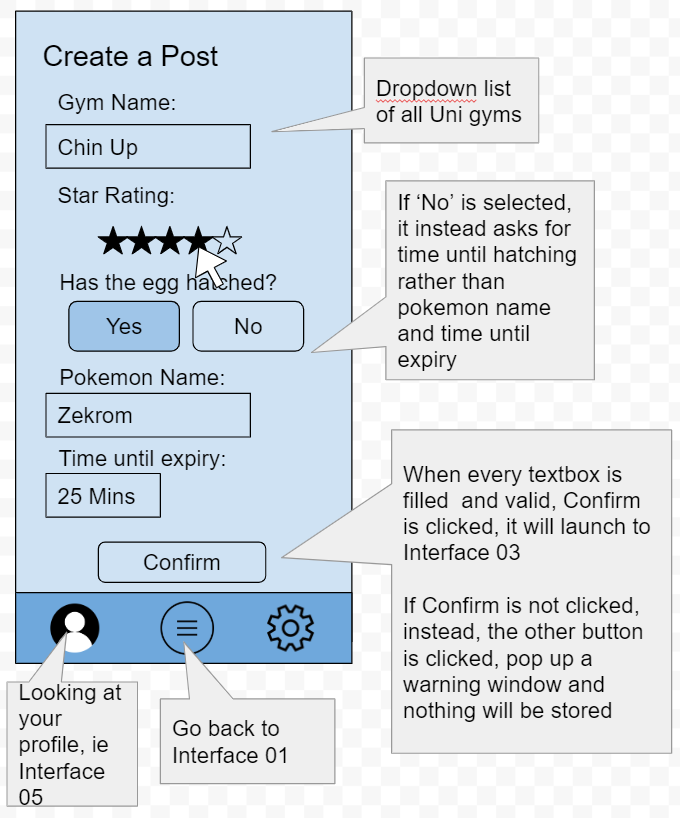
*Figure 1:* An approximate design of the appearance of the dashboard and its functionality. Green posts indicate that a raid has been confirmed to go ahead, while orange ones are just proposed raids.

*Figure 1.1:* Final dashboard layout. Proposed raids are shown in red and confirmed raids are shown in orange. From the posts you can see the location, star rating and time of hatching/jump in/raid finish.

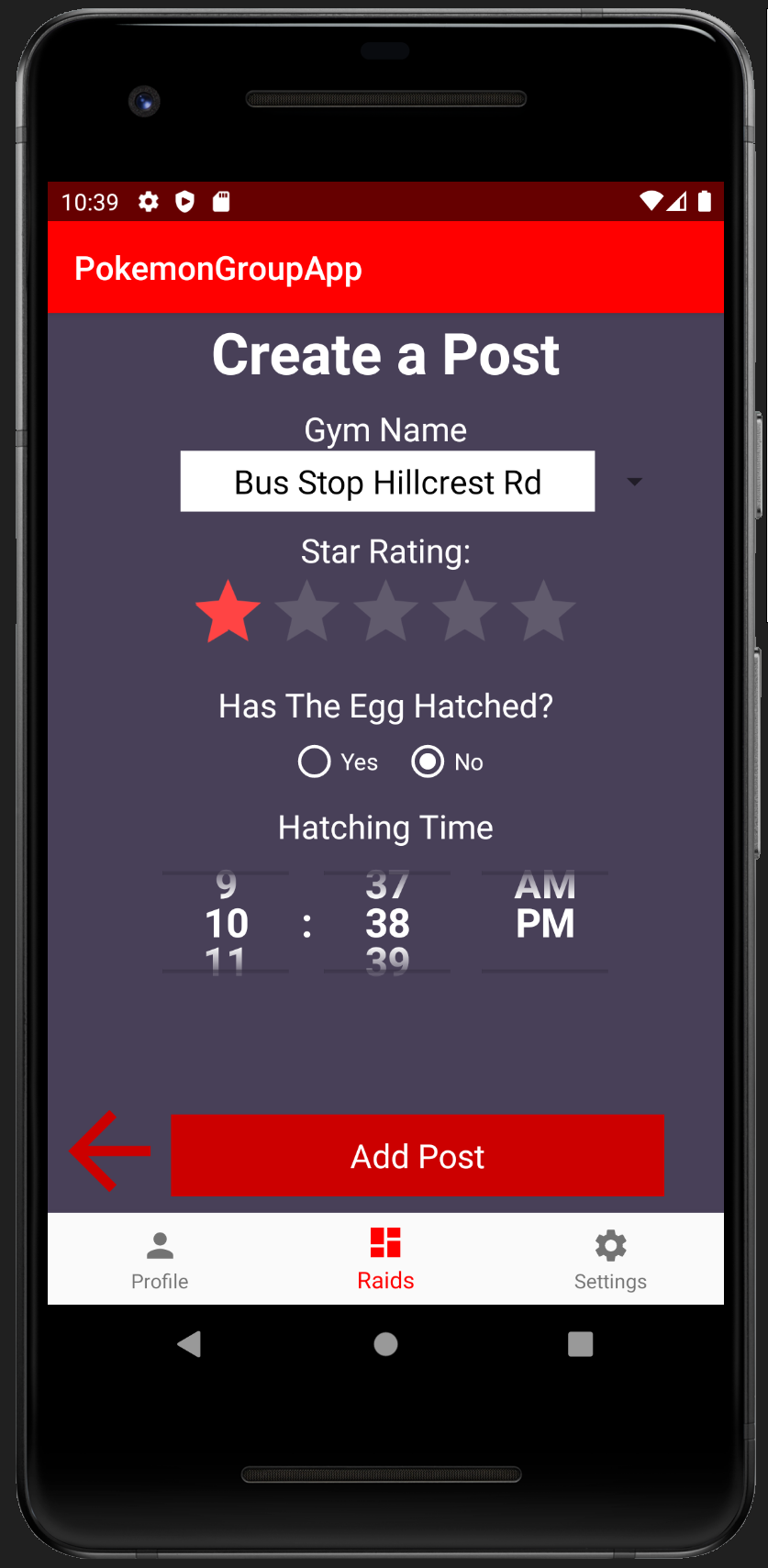
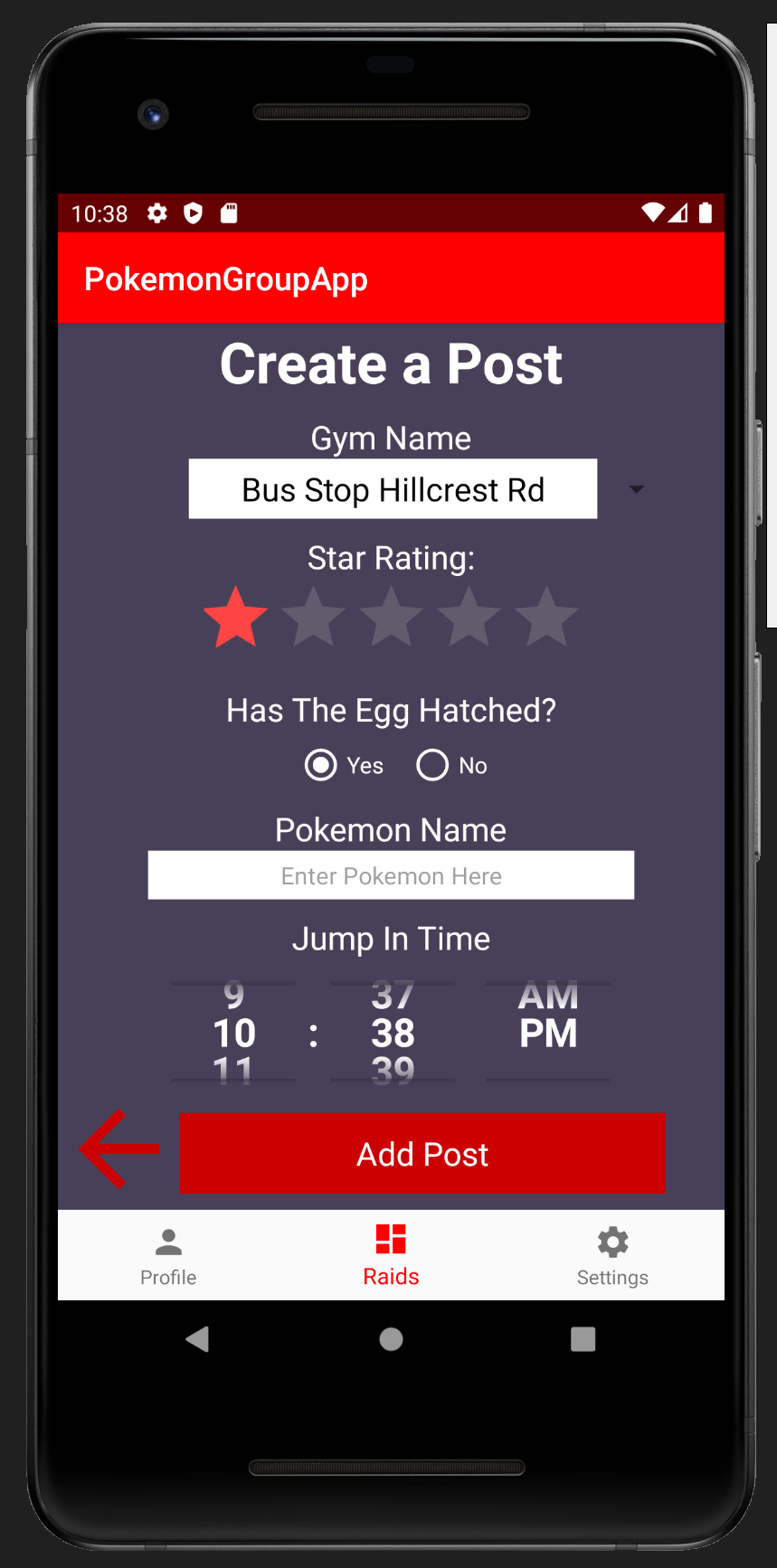
*Interface 02 - Create Post (Fig. 2)*

Upon clicking the ‘create post’ button on the dashboard, the user is taken to this second interface. There are multiple fields to be completed, each asking for some important information about the new proposed raid:

* + Location/Gym (Choose from a list, rather than type this, possible because app is specific to uni gyms)
  + Star Rating of Pokemon (1-5)
  + If it has hatched: Time the raid expires AND pokemon type
  + If it has NOT hatched: Time the raid will hatch



*Figure 2:* An approximate design of the interface for creating a new post. Different information is inputted in different formats. For example, The gym can be selected from a list of university gyms, the star rating can be clicked on and the pokemon name needs to be typed.

  
*Figure 2.1:* Final UI for creating a post. Different options are available depending on whether the user says the pokemon has hatched or not. If not, it asks for the hatching time. If it has, it asks for the pokemon name and preferred jump in time.

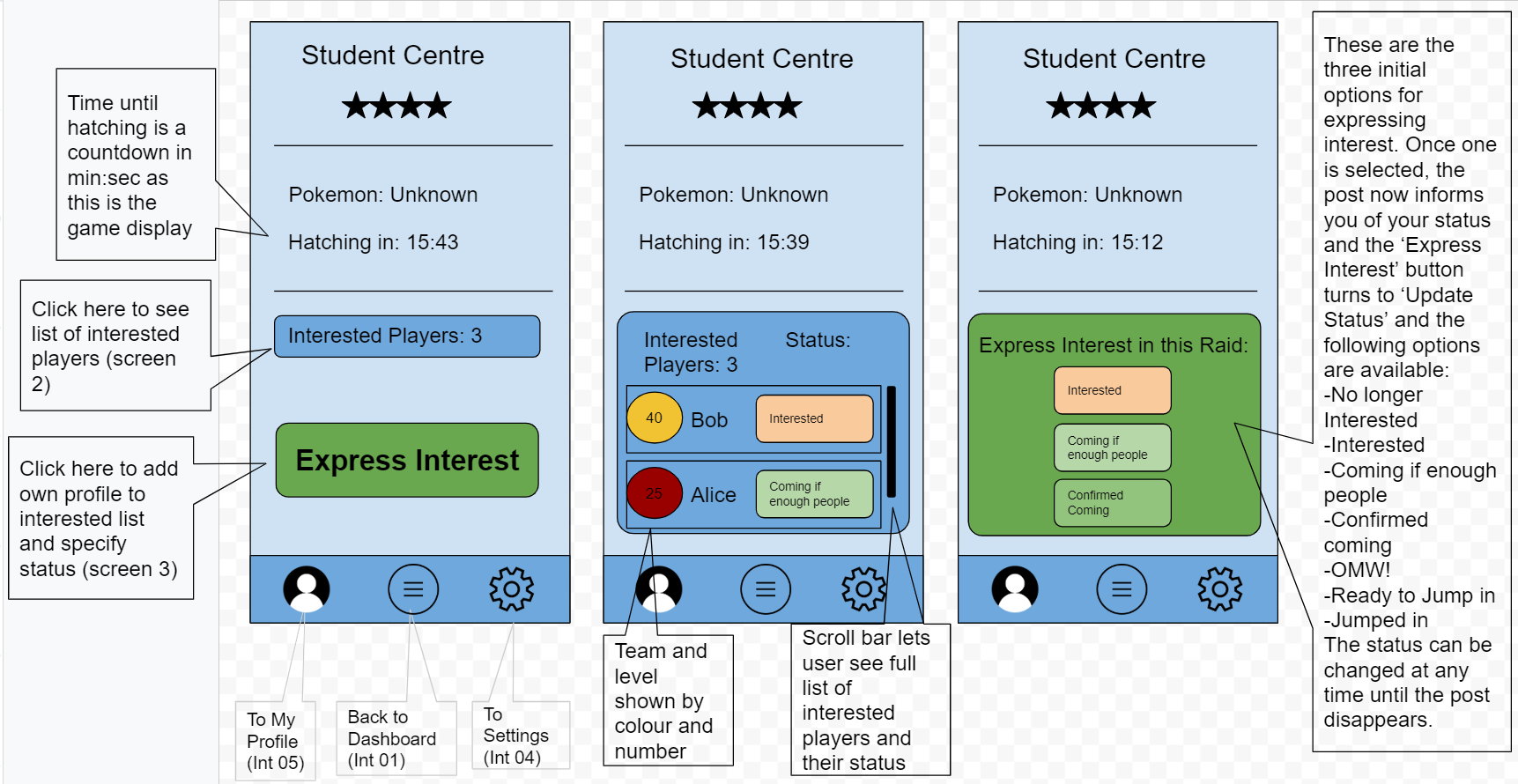
*Interface 03 - Detailed Posts (Fig. 3)*

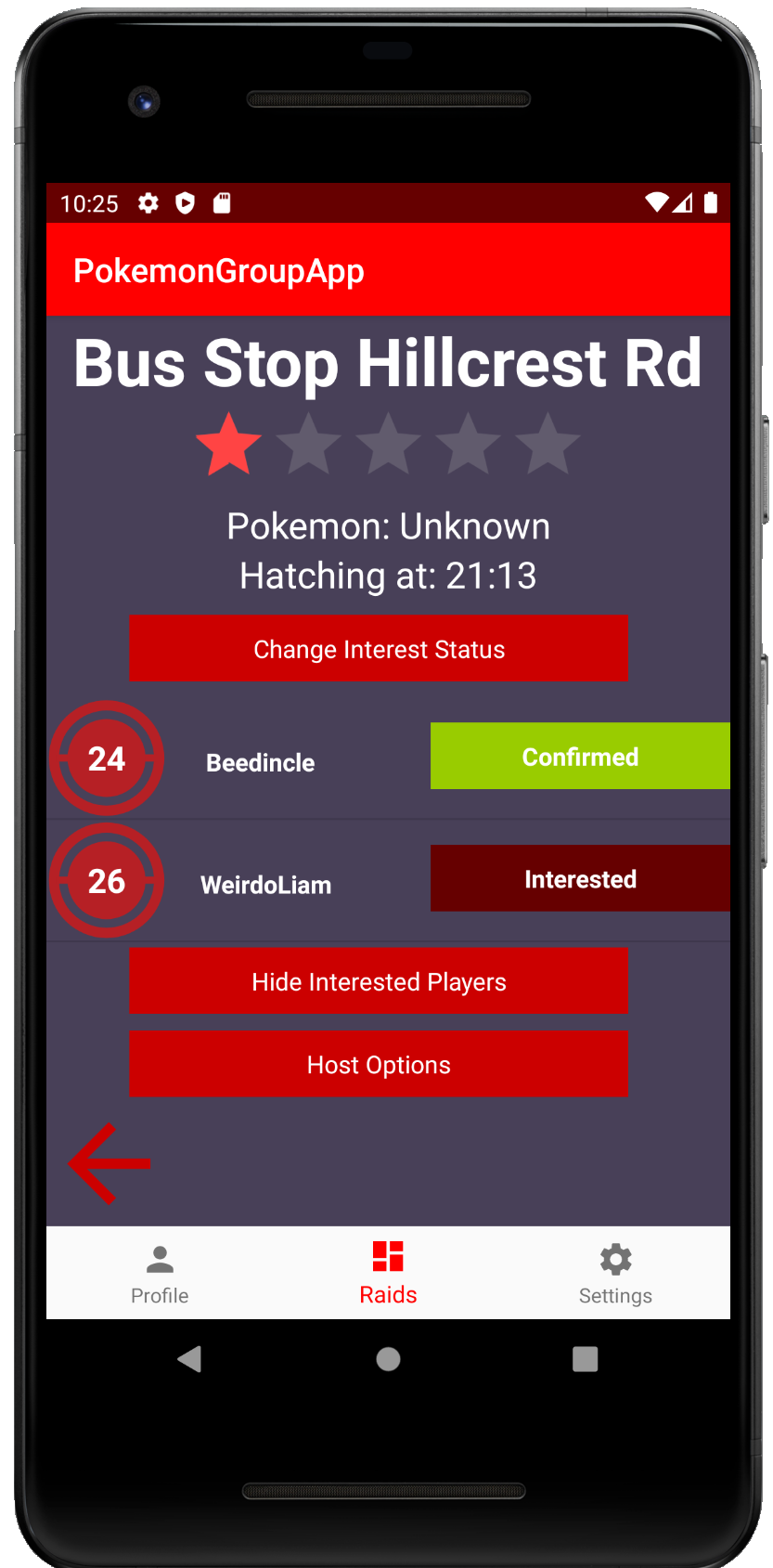
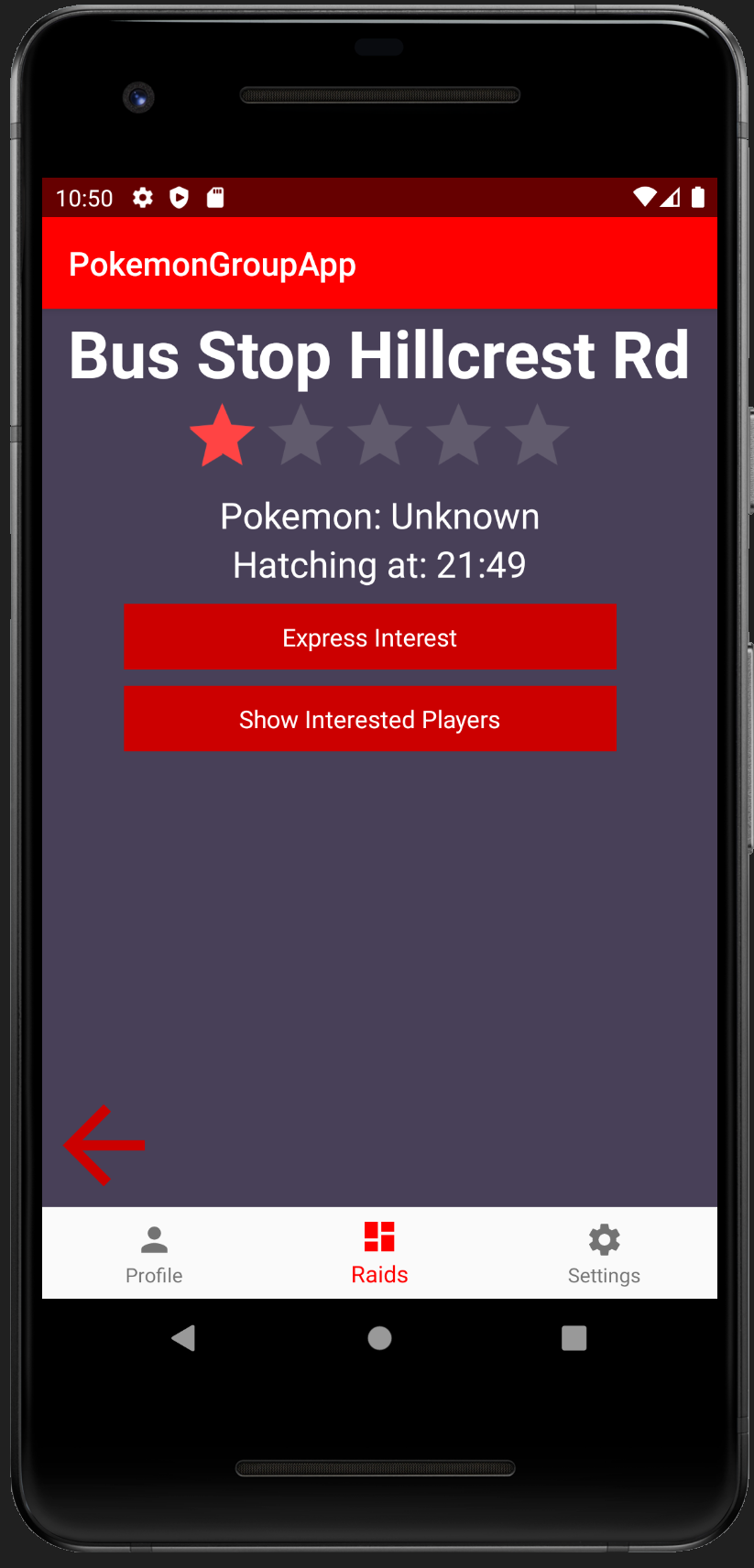
After clicking on any of the posts in the feed, the user gets brought here to see more information. It displays all the basic information shown in the post but also shows details about the other users interested in this raid.The players expressing interest can have one of the following statuses:

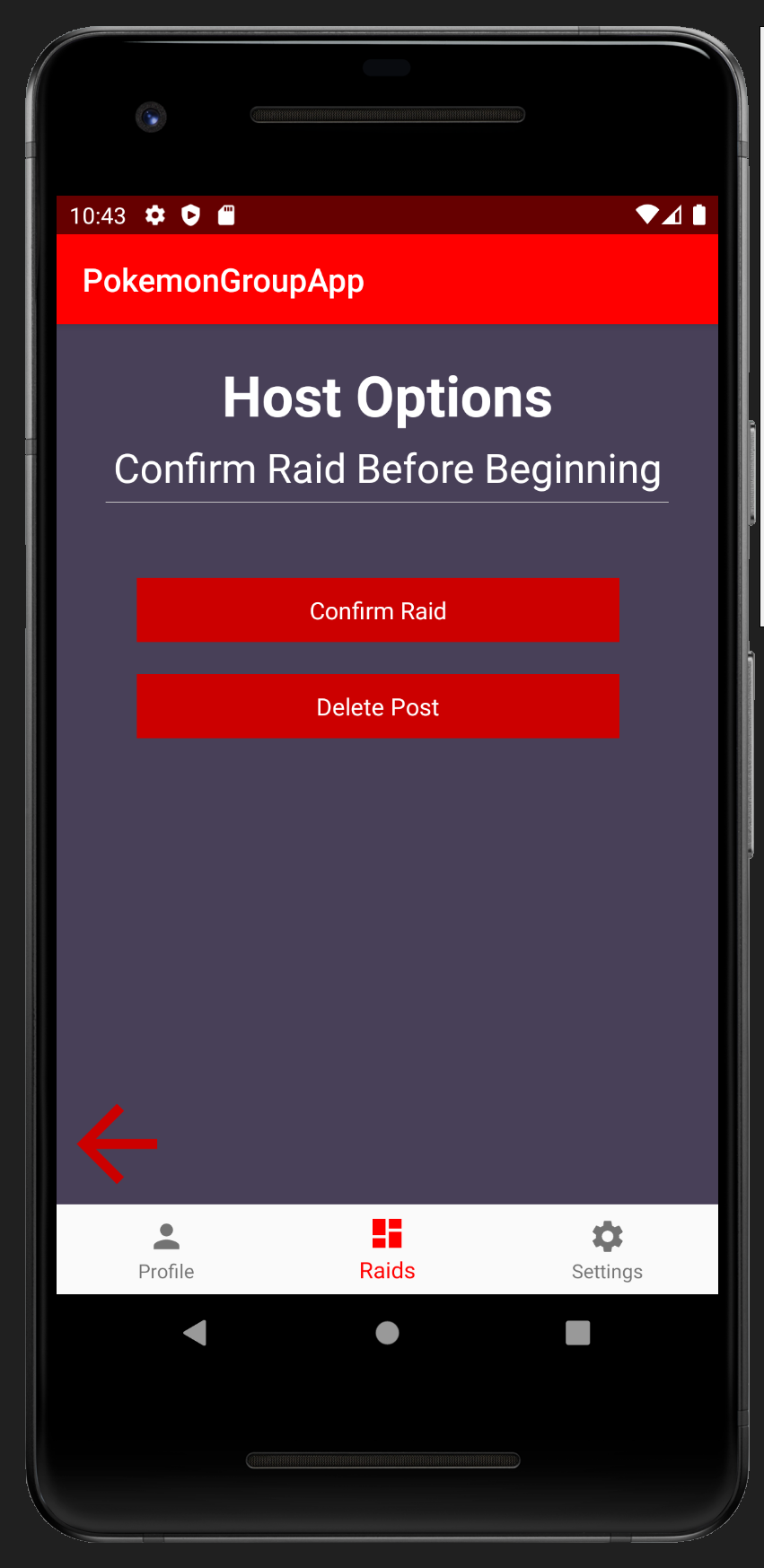
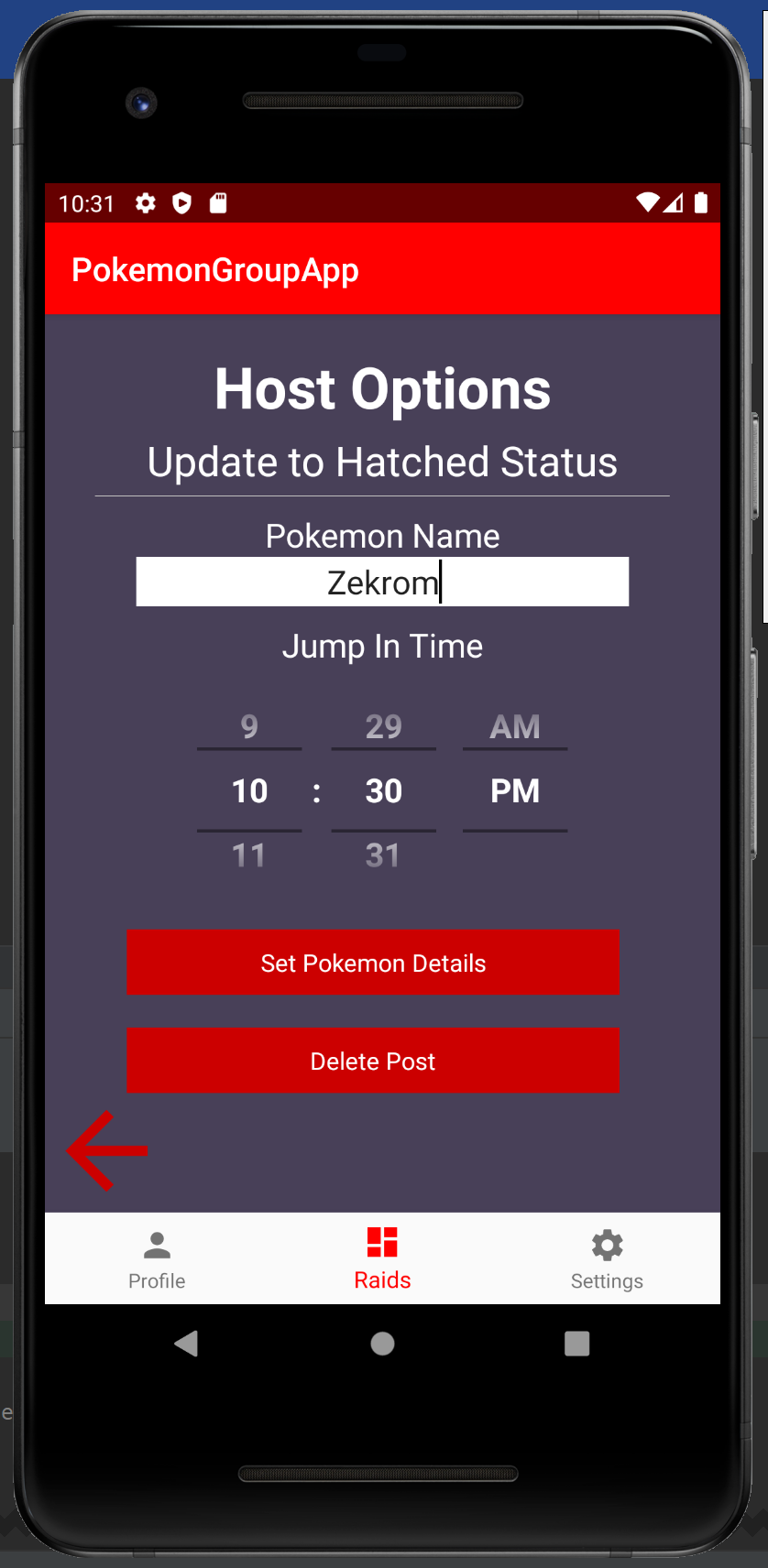
* + Interested
  + Coming if enough people
  + Confirmed Coming
  + OMW!
  + Ready to Jump In
  + Jumped in (this is to inform latecomers who can still see the post that they may have missed it.

The ‘Host’ of the raid (The user who created the post) is in charge of updating the others who are interested and has some further options. The Host can switch the raid from ‘proposed’ to ‘confirmed’ when they judge that there is enough interest to win the raid. When deciding this, the Host also chooses a ‘jump in’ time. Once confirmed, all interested users are notified and told of the ‘jump in’ time. They can then keep updating their statuses to ‘confirmed coming’, ‘OMW!’ and ‘ready to jump in’ as appropriate so the host knows where everyone is at.

Once all are ready to jump in, the host sends out another notice to jump in now.

  
*Figure 3:* An approximate design of the functionality of the detailed post view. Users can view the other interested players and express or update their own interest.

  
*Figure 3.1:* Final UI for viewing the details of a post. A regular user has the option to express interest or view other interested players (left screenshot). When viewing interested players it shows their username, level and current status (right screenshot). The host also has a third button available which brings up some additional host options (*Fig. 3.2)*.

  
*Figure 3.2:* Final UI for the Host Options interface. If the pokemon was initially unhatched the host can update the post with the pokemon name and jump in time once it has hatched (left screenshot). They can then confirm the raid if they decide there is enough interest to defeat it (right screenshot). The host may also delete a post at any time.

*Interface 04 - Settings*

The setting page can be accessed from the bottom right button on the dashboard and allows the user to modify their notification preferences:

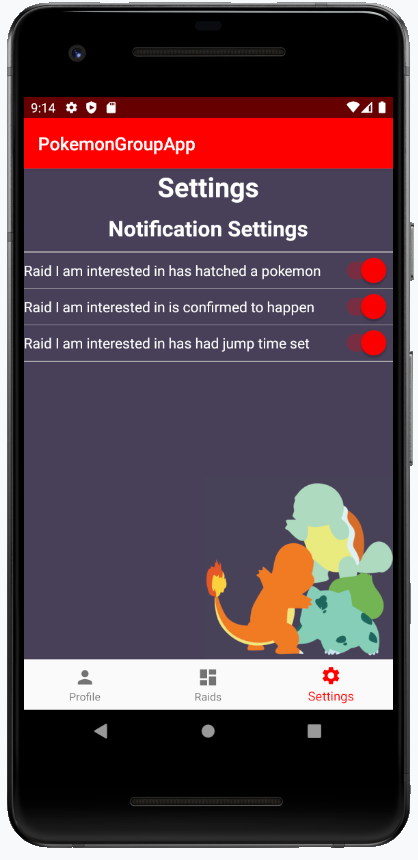
By default, the user is notified when:

* + - A raid I am interested in has hatched
    - A raid I’m interested in is confirmed
    - A host of a raid I’ve confirmed for is telling me to ‘jump in’

The user has the option in the settings tab to uncheck these notifications if they want to.

Some additional notifications that were suggested but not implemented in the interests of time include:

* + - There is a raid with at least [n] people interested
    - A raid I am hosting has hatched, notify me to edit post (would require being triggered by the current time matching the set hatching time)

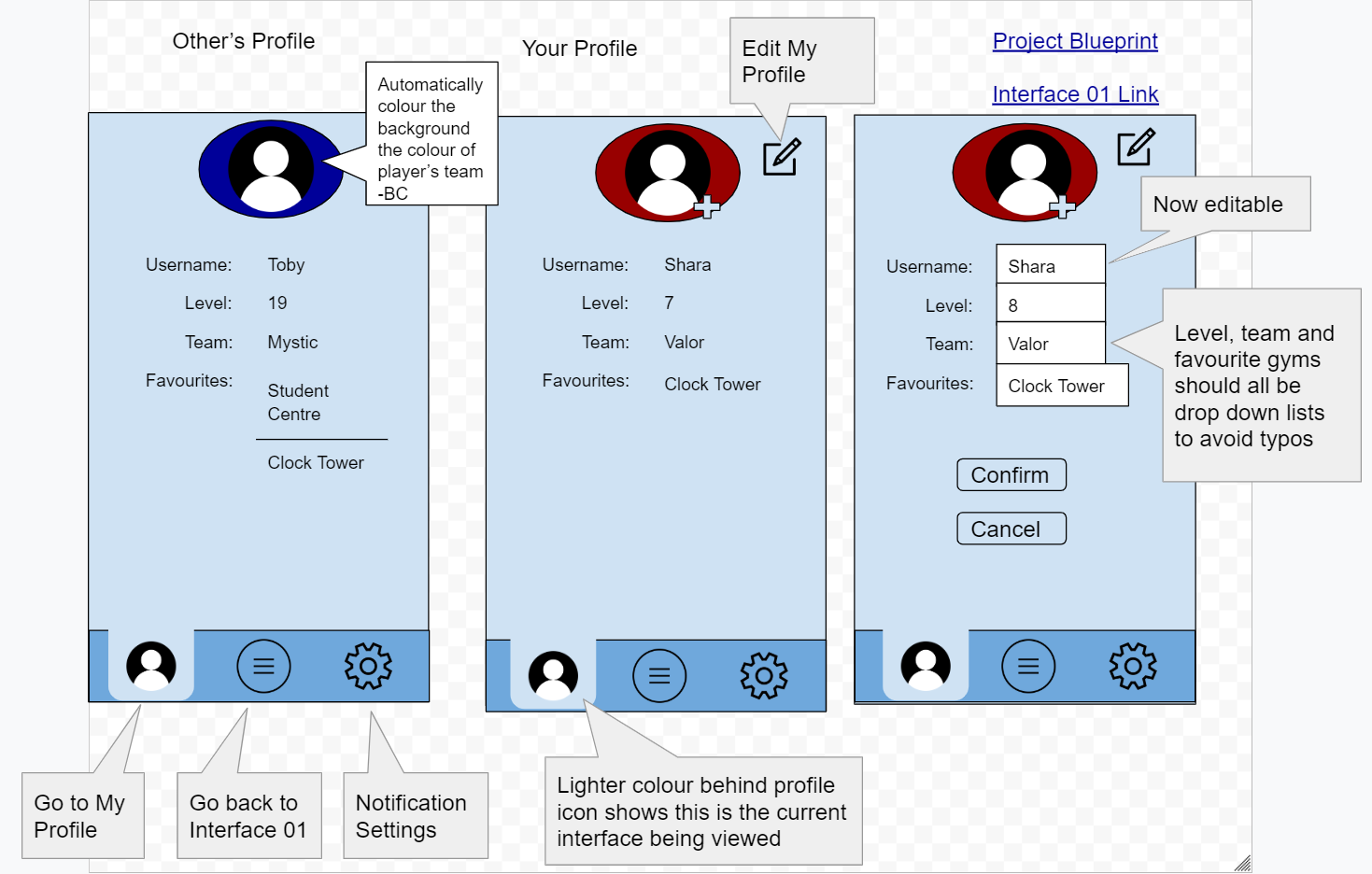


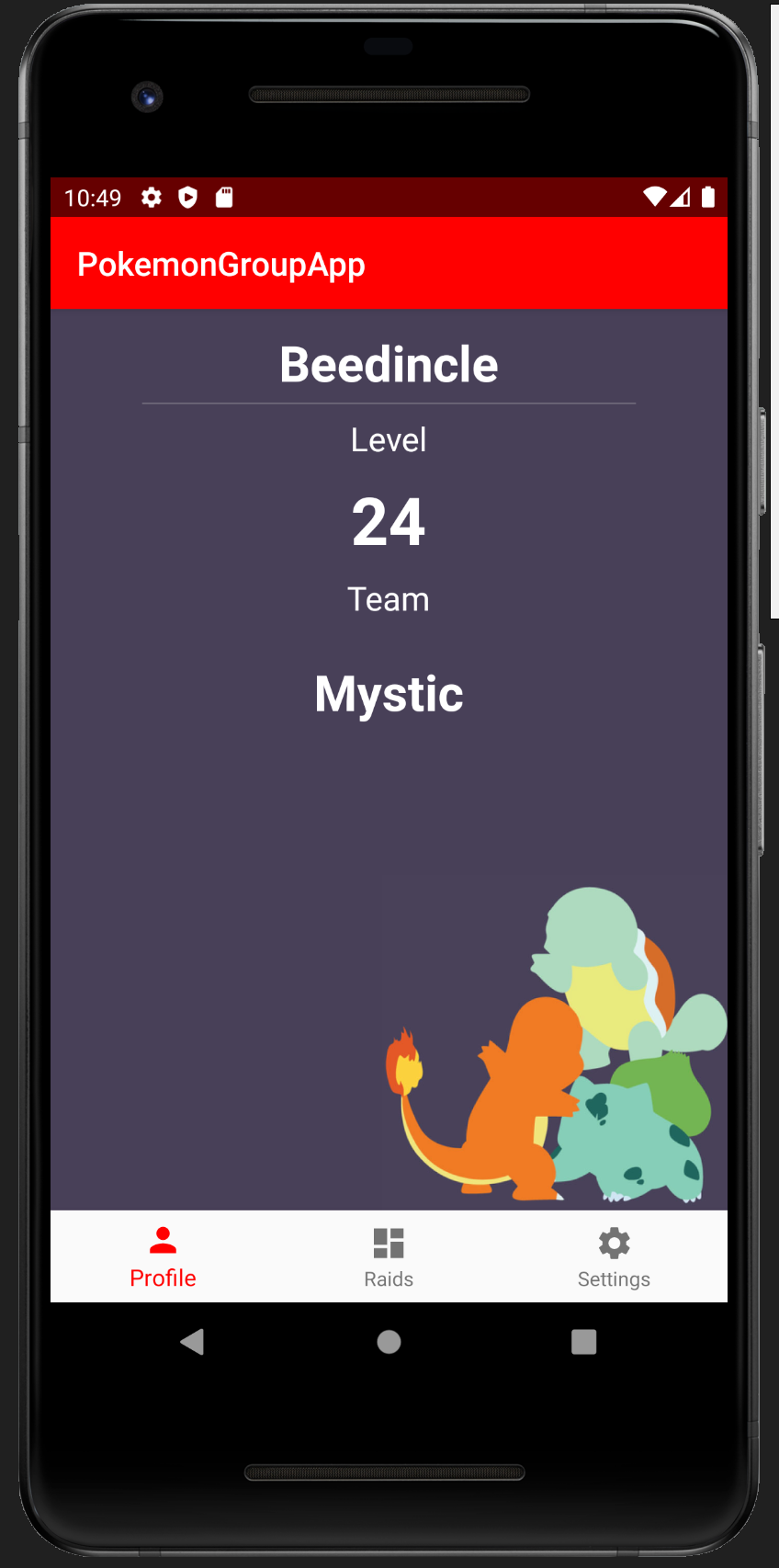
*Figure 4.1 (Left):* The final interface for the notifications settings.

*Interface 05 - My Profile (Fig. 4)*

This interface is accessible through the bottom left button on the dashboard and displays all the relevant information about the user. We had initially planned to allow the user to edit their details (*Fig.* 5) but decided that this feature was not essential as the user inputs this data each time they start up the app anyway.

* + Pokemon Go username (text box)
  + Team (dropbox: ‘Instinct’, ‘Mystic’ or ‘Valor’)
  + Level (1-40)
  + Favourite Gyms (can select any number from a list of Uni gyms)

  
*Figure 4:* An initial design of the appearance of the profile interface.

  
*Figure 5.1 (Left):* The final display for the profile interface.

*Figure 5.2 (Right):* The final welcome screen interface.

**Incorporation of User Interface Ideas:**

1. We have strived where possible to eliminate clutter by showing information through colours rather than with more words. A good example of this is in the post displays, where lots of information needs to be displayed in a small space. One piece of information can be shown easily through colour and that is the status of raid confirmation. The posts are displayed in red if they have been just proposed but switch to orange when the host confirms the raid as going ahead (*Fig.* 1). This makes it easy for a user to scroll through the feed and quickly see confirmed raids if they aren’t interested in waiting around to see which raids are going ahead.
2. We have designed our input interfaces for post creation and profile updates to take as much load off the user as possible. It is assumed that this app will be used just by players around Waikato University so there is a set list of ten gyms that they can select from when creating a post or selecting favourite gyms. This eliminates the possibility of typos in typing creating errors for the interpretation of messages on the receiving end. It also makes it easier for the user to just recognise the names of gyms rather than flick back to the Pokemon Go app to check for exact wording.  
   We have also implemented a 5-star graphic for the selection of raid difficulty, multiple choice options for inputting information such as team name and a digital clock scrollbar for setting the time that raids hatch.
3. Our main dashboard screen accounts for the accessibility of the thumb zone. In the centre, where it is most easily accessible, there is the list of raid posts that can be scrolled through or clicked on. The size of each post is big enough to be easily clickable on a touchscreen.  
   At the bottom the icon in the middle is the easiest to reach and takes you back to the main screen. On the left and right are the profile and settings links, respectively, and these are harder to reach as they are less frequently needed.   
   In the top right, reasonably out of reach is the ‘create post’ button. This helps prevent users from accidentally bringing up the create post screen.

**Communication Strategy:**

In our main activity we had 2 methods to provide the core ability to send and receive messages, ReadMessage() and SendMessage(). In different situations where devices are sending out data or making requests, the SendMessage() method would read information from the device and send a message which begins with a particular key with data following it. In the OnCreate section of the MainActivity a thread would be added which is always listening for messages being received through the ReadMessage() method, this would accept a message echoed out by the server and determine which callback function to run, as determined by the starting code for how it would need to run to split up the variables of the message.

The following planning was used to define the different types of messages:

<MESAGE\_TYPE>~<VARIABLE1>~<VARIABLE2>~....

All variables will be excluding the given separating character “~” (as enforced by the validation of input fields in the application to make sure these values are not used).

<MESAGE\_TYPE>:

Keywords will determine how the message will be interpreted and how many variables the message should consist of.

(MESAGE\_TYPE\_KEY: **RP**) **Raid Initial Post:** Allows for a new post to be created and posted to the dashboard

Format:

RP~<USERNAME>~<LEVEL>~<TEAM>~<TIME>~<LOCATION>~<RATING>~<POKEMON>~

<USERNAME>: Name of the user making the raid post

<LEVEL><TEAM>: Team and level of the user posting the raid, “I” for Instinct, “M” for Mystic or “V” for Valor. Level will be digits from 1 to 40.

<TIME>: Starting/Hatching Time in 24 hour format “17:00”

<LOCATION>: Location/Gym where raid is being held

<RATING>: Value between 1 and 5 stars to represent the star rating of the pokemon egg.

<POKEMON>: The type of pokemon which is expected to hatch, If not determined then “NONE” is sent

(MESAGE\_TYPE\_KEY: **RU**) **Raid Update Details:** Allows for a host to add details and make actions for events regarding a raid posting.

Format:

RU~<TIME>~<LOCATION>~<HOST>~<HATCHTIME>~<CERTAINTY>~<JUMPTIME>~<POKEMON>~<ADDREMOVE>~

<TIME><LOCATION><HOST>: Time, Location name and Name of Host User who posted to identify the raid that is being updated

<BATTLETIME>: Time to battle it if a egg has hatched, in 24 hour format “17:00”, if egg is not hatched “NONE” is sent

<HATCHTIME>: Time remaining until an egg will hatch, in 24 hour format “17:00”, if egg is already hatched “NONE” is sent

<CERTAINTY>: Whether a raid is proposed or confirmed, “P” for proposed, “C” for confirmed.

<JUMPTIME>: Time in which the leader wants a group to jump in, in 24 hour format “1700”, if <CERTAINTY> is not confirmed “NONE” is sent.

<POKEMON>: The type of pokemon which is expected to hatch, If not determined then “NONE” is sent

<ADDREMOVE>: Determines if the request is for adding or removing the raid. “A” for Add, “R” for Remove.

(MESAGE\_TYPE\_KEY: **RI**) **Raid User Interest:**

Is used to send a message to the server once a user expresses interest in one of the raids

Format:

RI~<<TIME>~<LOCATION>~<HOST>~<USER>~<LEVEL>~<TEAM>~<STATUS>~<ADDREMOVE>~

<TIME><LOCATION><HOST>: Time, Location Name and Name of Host User who posted to identify the raid that the request is going to be in effect for.

<USERNAME>: Name of the user making the raid interest message

<LEVEL><TEAM>: Team and level of the user registering their interest, “I” for Instinct, “M” for Mystic or “V” for Valor. Level will be digits from 1 to 40.

<STATUS>: The following integers are used to indicate the relevant statuses:

Interested : 0

Coming if enough people : 1

Confirmed Coming : 2

OMW! : 3

Ready to Jump In : 4

Jumped in : 5

No longer interested: 6

<ADDREMOVE>: Determines if the request is for adding or removing the status. “A” for Add, “R” for Remove.

(MESAGE\_TYPE\_KEY: **UM**) **Update Me:**Just a two-character message sent to the server on start-up to announce that they are a new user. All existing users who receive this message will then loop through their list of raids and send a ‘Send Old Post’ message for each one and loop through each raid’s list of interested users and send a ‘Raid Interested User’ message for each of those.

(MESAGE\_TYPE\_KEY: **OP**) **Send Old Post:**

Is used to send an existing post to a new user in response to a message asking for it. Contains all details from the create new post and update post messages in order to send through the post and its updates simultaneously.

**Storage:**  
Data is stored on each individual device. This means that when a new user joins it must access the pool of data from the other devices via the server. Upon startup, a new device requests to be updated by all other connected devices. They respond with specialised messages that transfer each raid’s creation details and recent updates. They also send messages listing the interested users for each raid. This is possible because each device will not add duplicate posts or interested users so the other existing devices receiving these messages will ignore them.

**Team member contributions:**

Kelvin:  
-UI Concepts and planning  
-Start screen interface  
-Dashboard interface  
-Create post, View detailed post and Host options interfaces

-Data structures for storing raid data and reading the

-Notification refinement and implementation

-Time picker implementation

**Percentage:** 25%

Ella:  
-UI concepts and planning  
-Profile interface  
-Notification Settings interface with toggle switch implementation

-Notification initial setup

-Editibility and features for future versions implemented

-Organisation of data used for the UI

**Percentage:** 25%

Daniel:  
-Establishing socket connections and messaging functions in synchronised threads.

-Planning for data being sent and protocol for sending with callbacks to interpret.  
-Implementing the messaging and update callback code responsible for creating initial posts, updating posts, confirming raids as well as code for triggering notifications in the messaging.

-Adjusting process of different stages of updating raid details through restricting what the UI displays.  
-Writing of the communication strategy section of the report (Summary, Raid Initial Post, Raid Update Details, Raid User Interest).

**Percentage:** 25%  
  
Bianca:  
-Initial app concept and ongoing consultations regarding the suitability of our implementations and making sure vocab used is ideal for the target audience (Pokemon Go Players).  
-UI concepts and planning.

-Code for ignoring duplicate posts / user updates.  
-Report formatting and writing of the incorporation of UI ideas section.  
-Implementation of the messaging and update code responsible for expressing interest, and updating new users of the posts which existing users have.

-Writing of the communication strategy section of the report (Send Raid Data, Send Old Post).

**Percentage:** 25%

**Links to Editable UI:**

Ella’s Interface 1: Dashboard

<https://docs.google.com/drawings/d/14sMxNEXidwrdRpLd7CvHd-YXc1quUzd9BsbX1SJwDR4/edit>

Bianca’s Interface 1: Dashboard

[COMPX318 - Project one - Alternate Interface 01](https://docs.google.com/drawings/d/1r7SXY21AJfsnFvCk7LkulBd3Ftmnb9_w-sVp9H-el7k/edit?usp=sharing)

Interface 2: Create Post

<https://docs.google.com/drawings/d/1iWladcMIrt7k1htZxYIK8aOeb3rjXOo4CN7buWWc718/edit>

Interface 3: Detailed posts

<https://docs.google.com/drawings/d/1cMlvTmIb6G524B236LIbWhahkz3mcC6GEwIOQ6lfeo0/edit>

Interface 5: My Profile

<https://docs.google.com/drawings/d/1dxIT57bSYZ2htQY7z4v_Mx15R3hMNLujzKGNRbWu5b0/edit>