David Ihle

Thomas McDonald

Diego Valdez

# Introduction

Our Project Goal: For our final project, rather than implement a more complicated version of Yahtzee, we wanted to make a prototype of a social media app Thomas thought of. This project offered the perfect opportunity to do that, since it required a GUI on top of basic functionality. We also wanted to use something besides Swing to create the GUI, because a social media app with default Java panes would look a little dull. As the project progressed, we quickly realized we wouldn’t be able to implement everything we wanted in our app, but we wanted to create a minimum viable product that we can continue to improve upon in the upcoming years. So we settled on the Bunchin’ Social Media Simulator application!

# Design

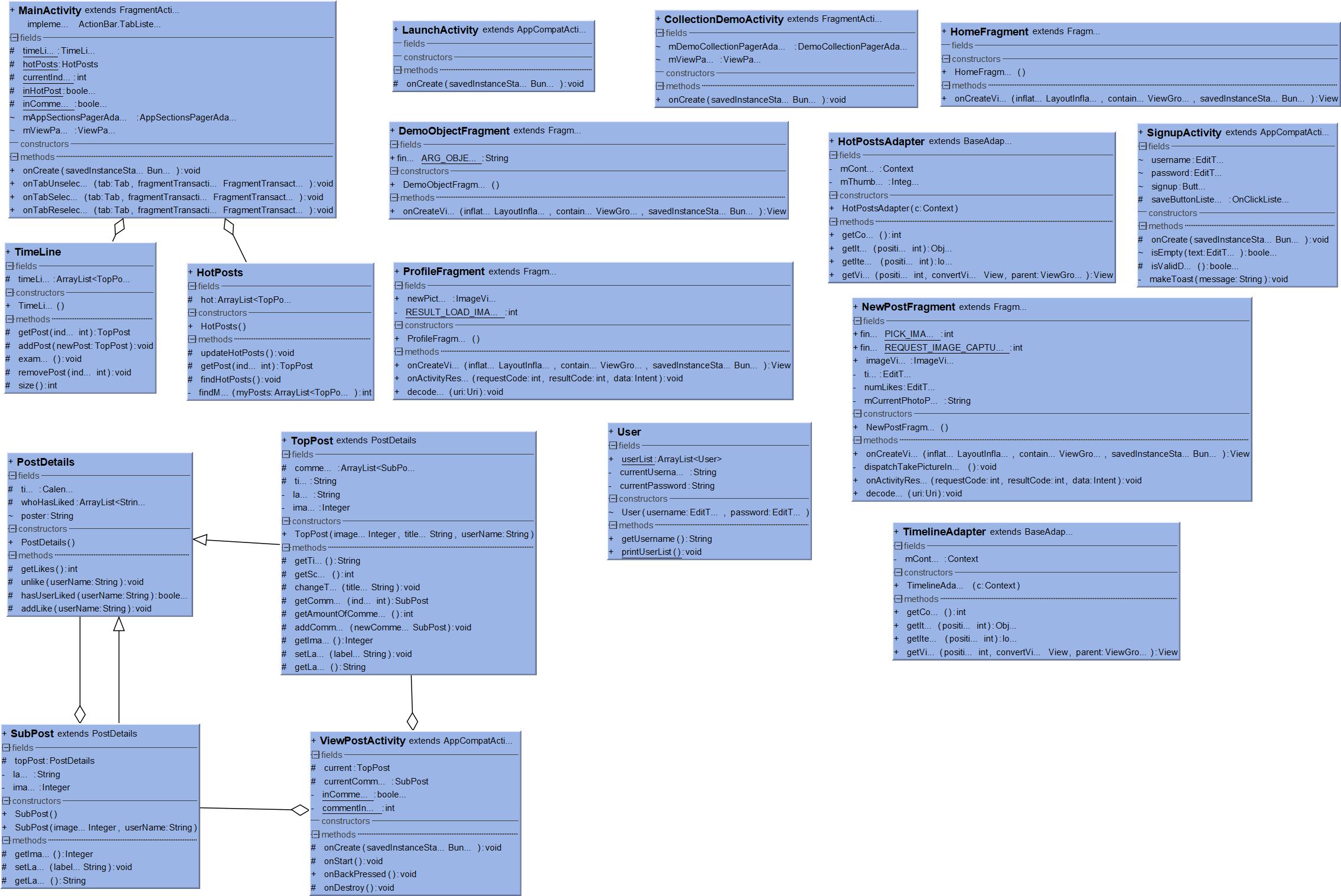
All three of us met to create the initial test plan, UML, game rules, etc. during the very beginning of the project.

Here is a list of the key features we wanted to include, along with brief explanation on how we implemented them and who was responsible for the implementation

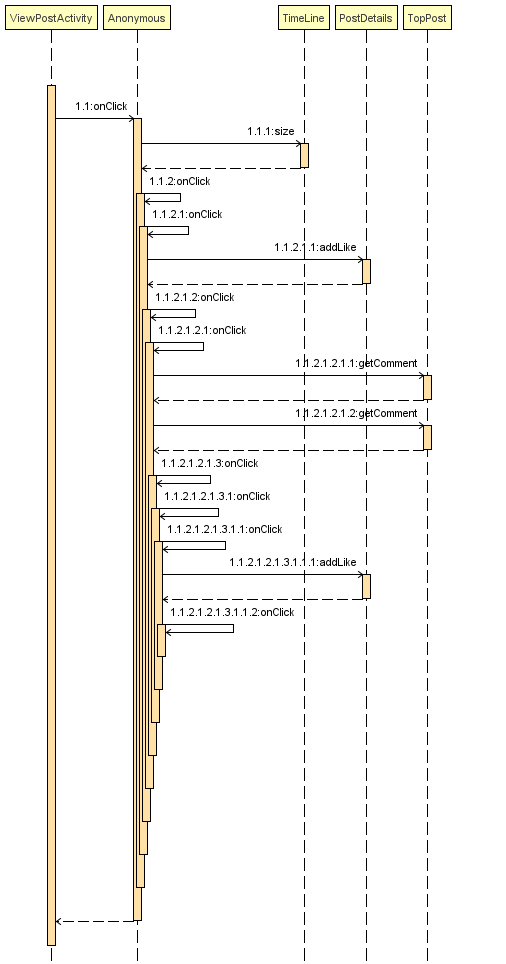
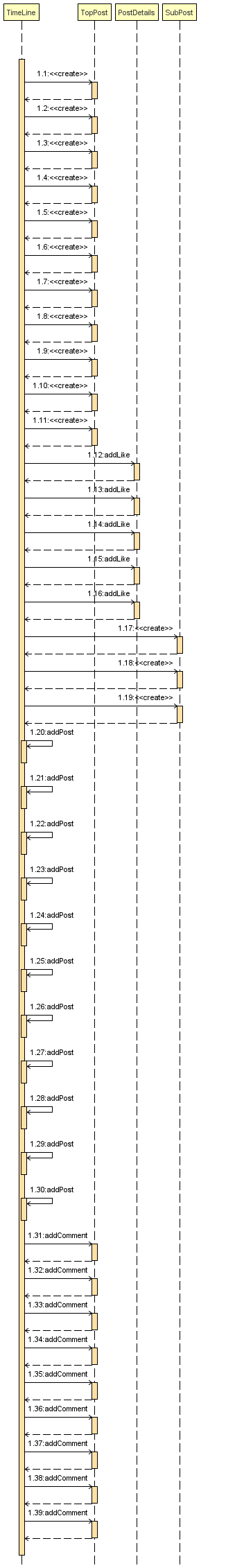
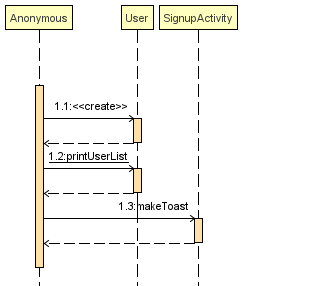
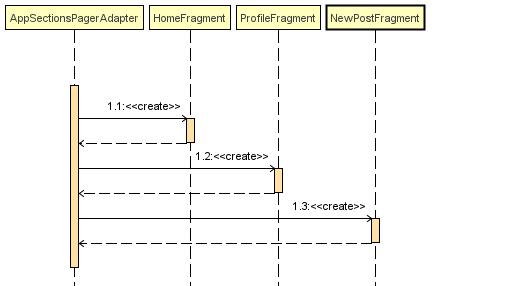
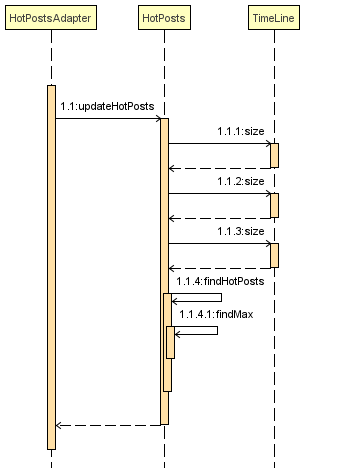
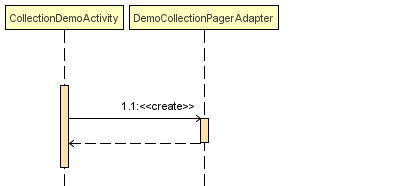
* User Creation & login - David
  + When the app first starts up, will prompt the user to sign up. On signup, will check to make sure the textboxes are not empty, and pass the information to the User constructor. The User Object will be created and added to a static ArrayList of Users. This will serve as the master list of all users after we decided to move away from the online Firebase implementation.
* Logo Design - David
  + With help from a friend, created the Concord app image using Adobe Illustrator
* Timeline - Thomas
  + Created the object Timeline which is incharge of storing all post viewable to the user. Posts can be easily added, accessed, or removed through the added methods. Also created classes PostDetails, TopPost, and SubPost which are all used throughout the program to store relevant data for any given post.
* Hot Posts- Thomas
  + Created the object which is used as the master copy of the hottests posts. The hot post are found by traversing through all post on timeline to find the top 8 with the highest scores and storing them inside an ArrayList. Scores are based off of time posted, amount of likes, and comments. Hot posts can be easily accessed and updated.
* Gridviews for timeline and hot posts - Diego
  + Created a gridview layout with XML in order to properly display the timeline and the hot posts for the user. The gridviews were then populated using two separate adapters that were responsible for retrieving the required information for each gridview and displaying them in the correct order
* Profile Page - David
  + Created an empty imageButton centered at the top of the screen to hold the profile picture. When the user clicks to change the profile picture, I asked for users access to camera roll, and let them select a new picture. Whenever the user finishes selecting a picture, coded onActivityResult, which will take the image selected, convert it to a bitmap image, resize it, and then display the image in the reserved location. That image will stay there even if you scroll away from the profile page. It also allows the user to create more users to add to the simulation
* Layout for New Post - Diego
  + The layout asked the user to input a label or description for the post that they wanted to upload, and then also asked them for the amount of likes that they wanted the post to have due to our app being a simulator for social media. The user would then have the option to either take a new picture using the camera, or pic an already existing photo from their photo gallery. The page then shows a preview of the picture that was picked, and gives the user an option to submit their new post
* Layout for View Posts - Thomas
  + Created layout for viewPost with XML in order to properly display an activated post. Setup buttons to be implemented for traversing posts, liking posts and navigation to sub posts and homescreen.
* Traverse Posts and comments - Thomas
  + Replaced relevant information such as title and username. Implemented listeners for the buttons laid out in XML to properly navigate through the hot posts, timeline and sub posts. Also created a system where the current activity would create another with the desired next post, before destroying itself.
* Like posts and comments - Thomas
  + Implemented into the object PostDetails, it stores an arrayList of users who have liked a post and checks to make sure a user does not like a post twice. Also give option to unlike a post or comment.
* Swipe View navigation - Diego
  + In order for our users to be able to navigate seamlessly through our application, we used a combination of a MainActivity and fragments. The MainActivity is what would be launched when the users first entered the application, and this hosted three different fragments that would display the other views from the user using case testing to determine the appropriate page to display to the user

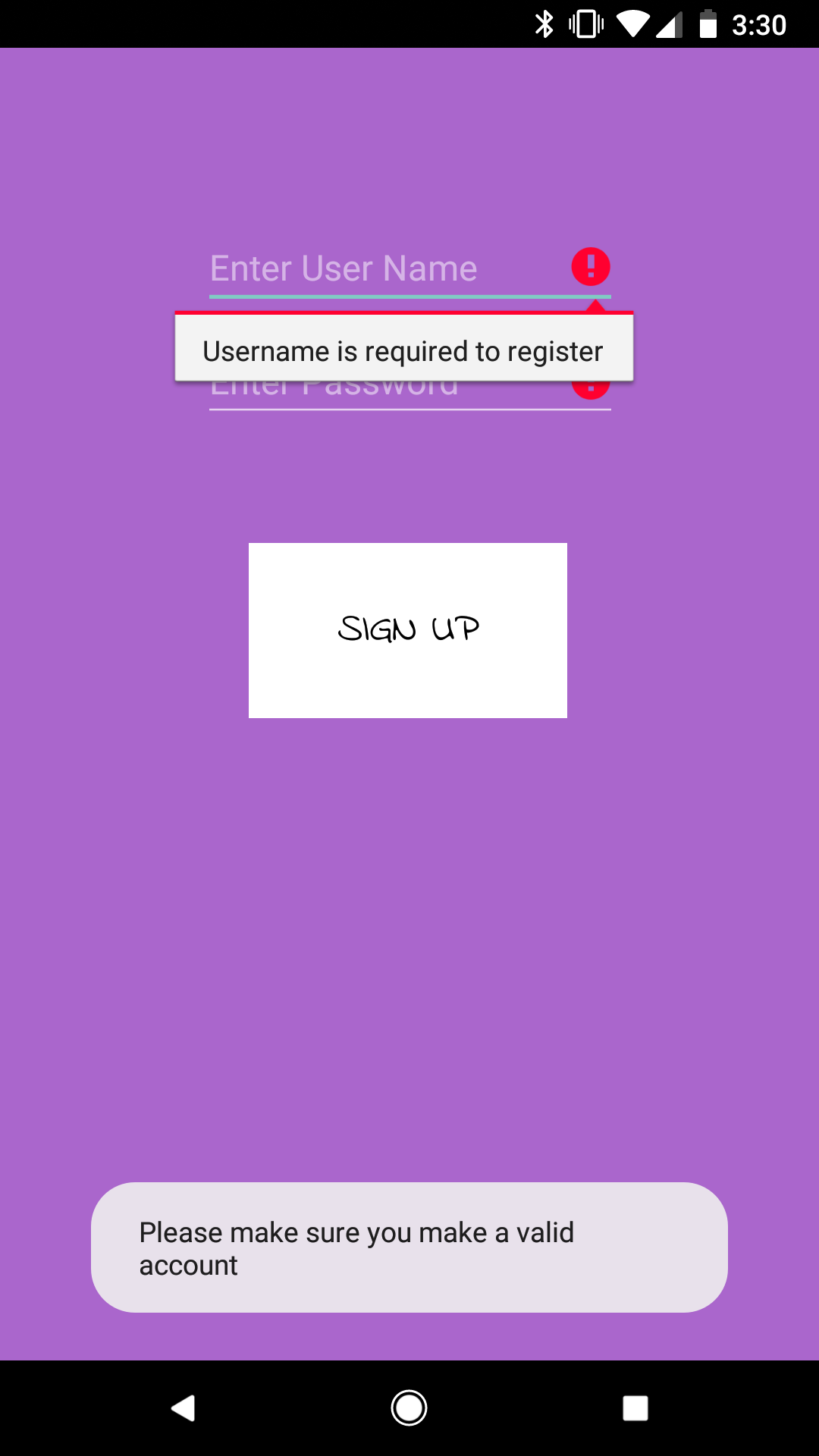
UML Diagrams

Class Diagram



Sequence Diagrams



Test Cases

**Launch Activity:**

1. User is prompted to Login or Signup
2. If Signup selected, send to sign up screen(\*Sign Up Activity)

**Sign Up Activity:**

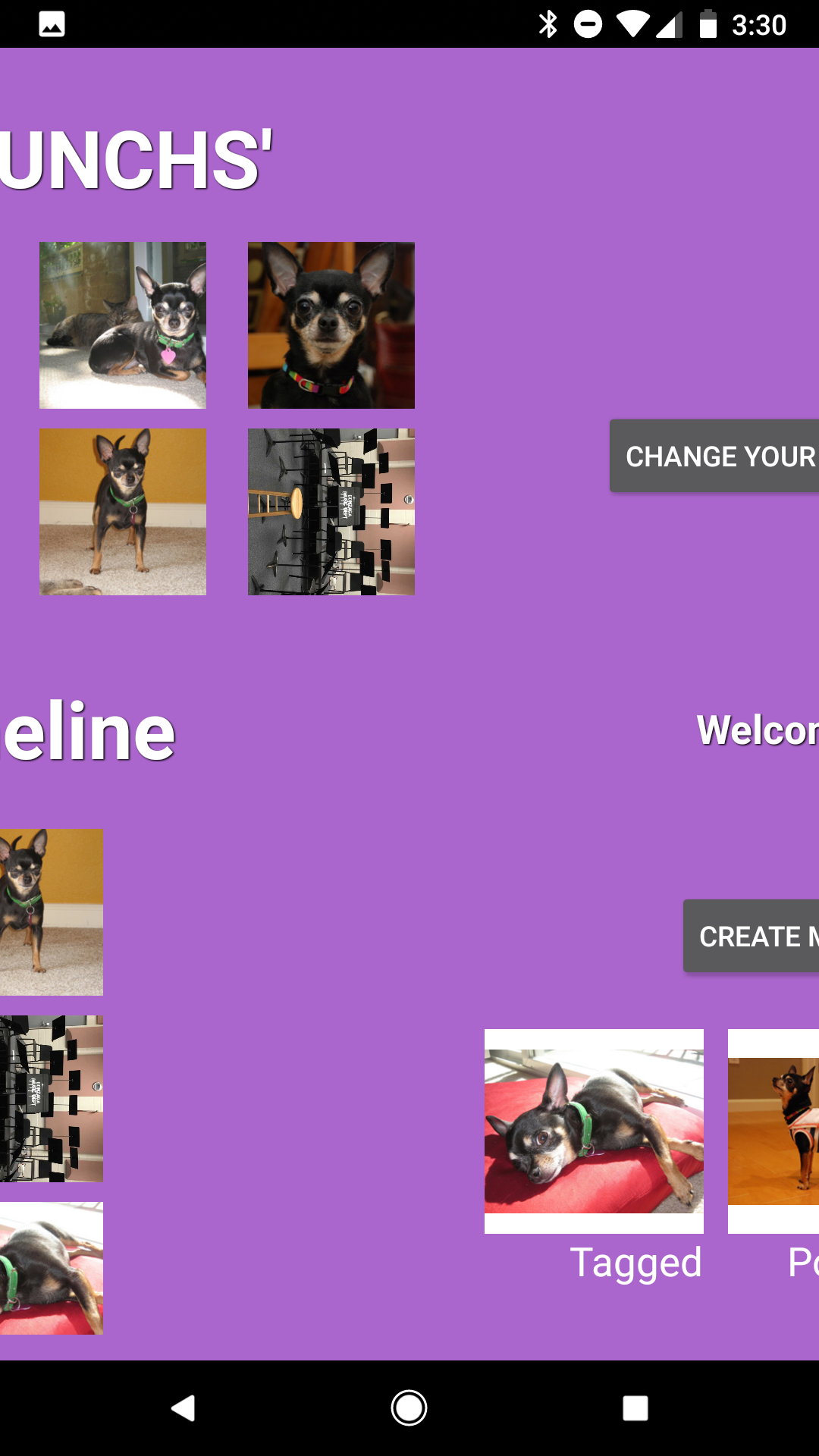
1. Allow users to enter a username and password.
2. If user taps signup,
   1. if username and password have not been filled out, repeat step 1
   2. If they have been filled out, send to \*Main Activity

**Main Activity:**

Contains 3 fragments Home, New Post, and Profile.

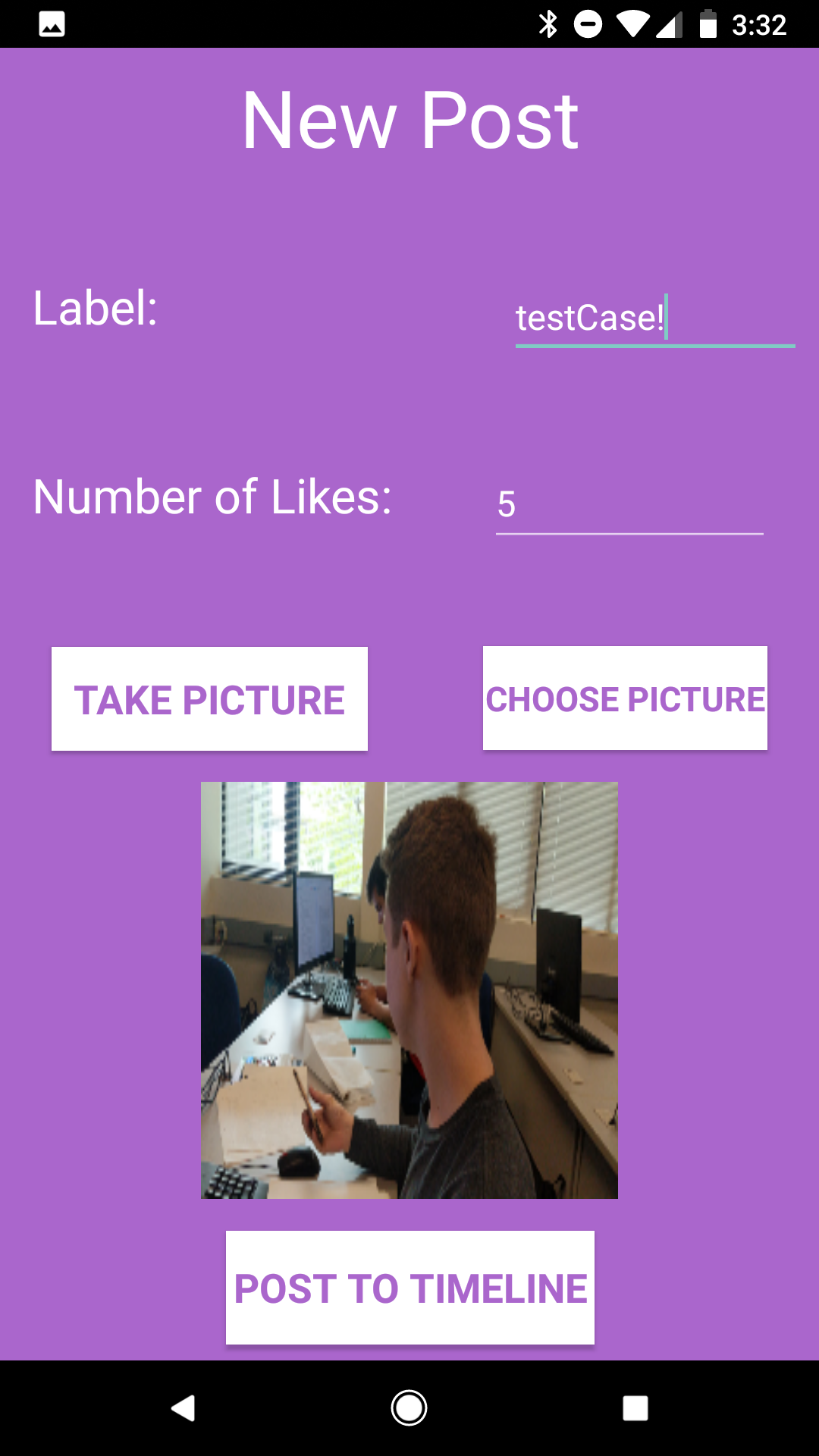
**Home Fragment:**

1. Display home screen which includes hot posts and a chronological timeline

If user swipes left, display Profile screen (\*Profile Fragment)

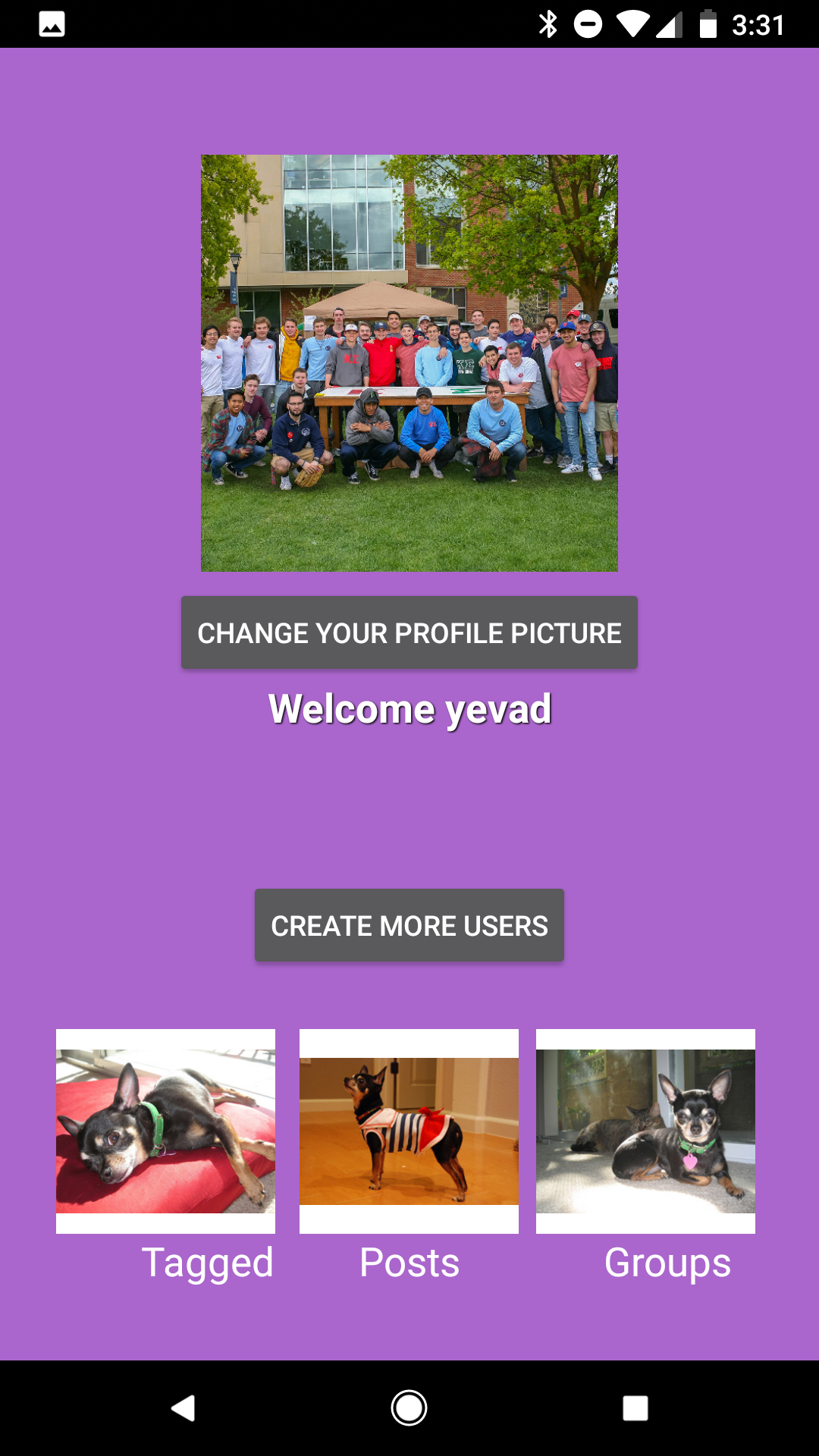
1. Hot Posts:
   1. If user taps on one of the 8 Hot Posts send to View Post screen(\*View Post Activity)
   2. Verify that the next post from View Post is also the next post in the hot posts or timeline.
2. Timeline:
   1. If user taps on one of the posts on timeline send to View Post screen(\*View Post Activity)
   2. Verify that the next post from View Post is also the next post in the timeline.
3. If user swipes up the screen will drag down revealing older posts on the Timeline

**New Post Fragment:**

1. Display new post screen which includes several buttons corresponding to the possible types of posts.
2. User can choose a label, amount of likes, and a photo for their post.
3. If user swipes right, display Profile screen(\*Profile Fragment)

**Profile Fragment:**

1. Display profile page which includes user specific name, profile picture, posts create new Users.
2. If user swipes right, display Home screen (\*Home Fragment)
3. If user swipes left, display New Post screen (\*New Post Fragment)
4. If user taps “change your profile picture” they can select a picture from their photo gallery to set as their new picture.
5. If users taps “Create More Users” they are brought to Sign up screen (SignUp Activity)



**View Post Activity:**

1. Display corresponding picture in background with title overlay in top left corner, username in the top right, and “like” icon in bottom right.
2. If user taps right side of screen, change current post send to next post(\*View Post) if one exists, else do nothing
3. If user taps left side of screen, send to previous post (\*View Post Activity) if one exists, else do nothing.
4. If user taps bottom of the screen display Sub-Post(\*View Post Activity)

1. If user taps back button,
   1. If user is in a top post display home(\*Home Fragment)
   2. If user is in a sub post display top post (\*View Post Activity)
2. If user taps “like” icon, add like to current post.

# Conclusion

The final project we presented in class was a lot narrower in scope than what we initially thought. Our first big branch from the initial plan was when we decided to simply create a local ArrayList of users rather than setting up an authentication system online. As a team however, we were able to take our individual assignments, implement them, and finally reach something presentable. This project as a whole gave us great experience in planning a computer science program, but it also gave the three of us invaluable experience in getting to know each other and working together because we will continue to work on this app in the future. We started texting much more frequently, updating each other on our progress and communicating what needed to be done by a specific deadline. We weren’t able to check off all our initial goals, but we learned to adapt and to create the best project we could.

**Issues:**

* Addressed
  + Didn’t know how to do much of anything in Android, so we used tutorials and YouTube videos to learn the programming language
  + Didn’t know how to access a user’s data from a database, so switched to implementing via ArrayList
  + Initially we did not know that we had to use fragments to implement the swipe view and we didn’t realize this until we had already built a lot of it with activities, so we essentially had to tear down much of what we had already built and replace everything with fragments, and not everything was able to be successfully ported over
  + Had a very difficult time in displaying the username on the profile page. Originally tried to set the EditText XML source to be the username field in the signup, but that didn’t work. When the user clicks, sign up, it will verify that the fields are filled out adequately, and will convert the input to a String value which can then be stored in the Username field. Then display this converted text on the page
  + Had a hard time navigating between activities, had to override onDestory and onCreate methods
* Unaddressed
  + The issue preventing us from being able to add the post to the timeline is that we couldn’t figure out how to transform bitmap images into Integer images. The bitmap image is created when you either take a picture or select one from gallery. This is what shows up on your <profile screen> or the <new post> screen. In the future we would make sure to use the same data types
  + Didn’t get to implement a friends list

**Future:**

Even as we wrote this final report, we were discussing how we can improve upon our app, and make it more professional. The first step is to restart the authentication system and implement Firebase. This will allow users to sign in via Gmail, Facebook, or email/username, and store their profile information in a database. This will make it easier to scale and to visualize the amount of users and app traffic. It will cut down on time that it takes to resize the list as well. We also want to polish off the design, make it look like a real app with better icons and real images. We plan on joining the Mobile App Development club as well next year, so we will have experienced people giving us ideas and support while making this project.