**Project 1 - MOFOTO APPLICATION**

**Name: \_\_Maiar Khattab\_\_**

**Email: \_\_\_**mkhattab@horizon.csueastbay.edu **\_\_\_**

**Section 0 Introduction and Proposal:**

Maiar Khattab

4/17/17

CS 4521

Android App Proposal

**MOFOTO Android Application**

**Concept Summary:**

The purpose of my application is for users to upload and share photos with other users. The user has to sign up and login to access their account and upload photos they have taken and want to share with friends. Each user will have their own profile where their photos are stored. The users will all have access to the main page/dashboard which allows them to view other users’ profiles by clicking on their username. The users’ photos will be saved on a server in order for all users to accesses other users’ photos.

**Audience**:

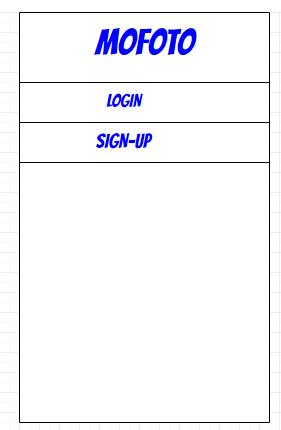
The demographics/audience my app targets is all ages starting from 12 years old. The app is meant to share photos with other users, people of all ages are intended to use it.

**Application Cost and Projected Success:**

I will make the app free because gaining users is an essential part of the app because without users the application will be boring. It relies on users sharing photos with one another.

**Monetization:**

The ways I plan on making money on this application is having advertisements embedded on the user’s dashboard. Another way to is to limit the number of posts a user can make unless they pay to upgrade their account and can post as many posts per day.

**Interface Mockups**:

In figure1 it is the simple LOG in screen, where a user can login

To the Application or if they do not have an account they will

Sign-up.

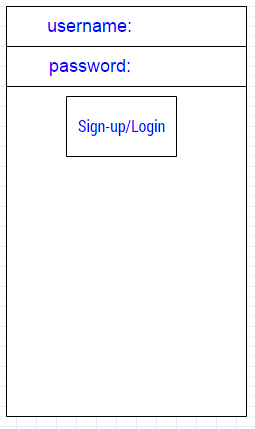


Figure2:

The user will enter their

Credentials to login to

Their accounts.

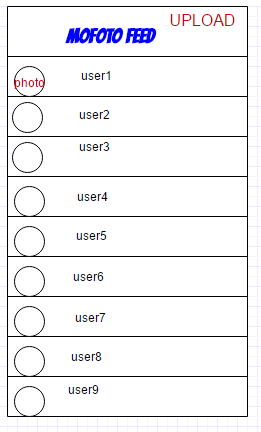


Figure 3:

This is the view in which

The users will be able to

See the photos shared

By other users by clicking

On the username to view

The photos. Also the upload

Button allows the users to

Upload their photos.

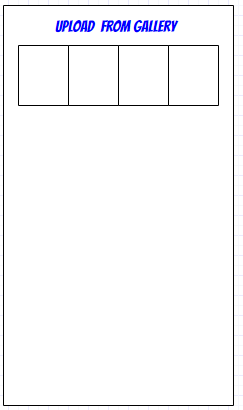


Figure 4:

This is the view when the user

Chooses to upload from their

Photo gallery, I will implement

Access to camera where the

User can upload a photo they

Took via the camera in the app

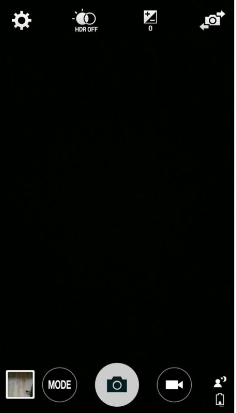


Figure 5:

After clicking the camera button.

User can open camera to use to

Take a photo, which can be used

To upload onto user profile

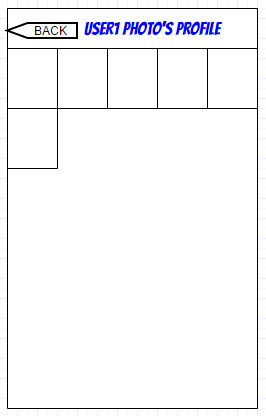


Figure 6:

This is the user’s profile

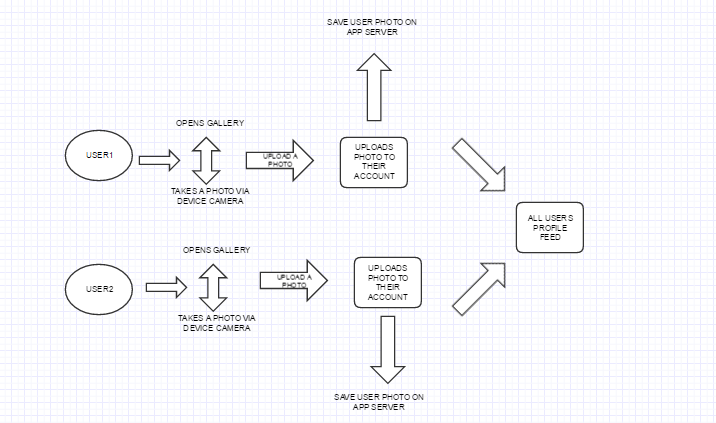
Where they and others can

View their photos, they can

Go back using the back button

To go to the main photo

Feed page.

USE CASE:

References:

<https://play.google.com/store/apps/details?id=com.instagram.android&hl=en>

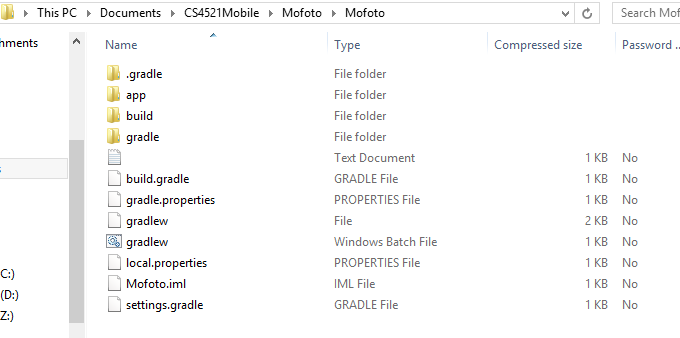
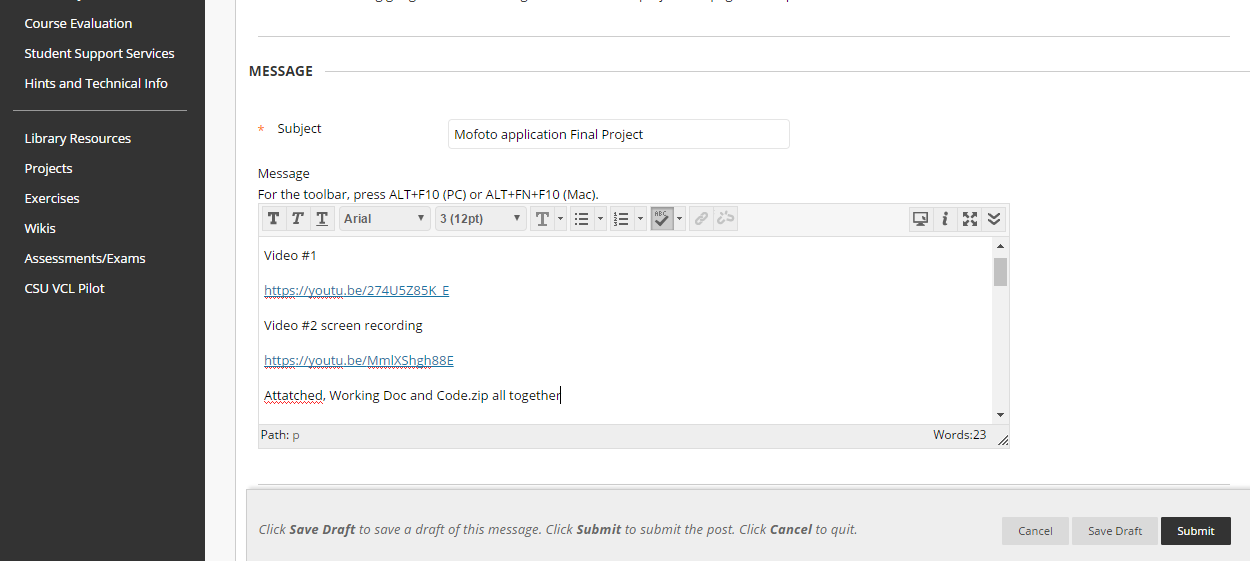
<https://developer.android.com>

Device Sensor Processing-

I will be implementing the use of the camera on the device. So that the user has the option of taking a photo via the camera and uploading that to their profile. I am not sure what algorithms I will be implementing for this application yet, but I will research and see what is most beneficial for my application.

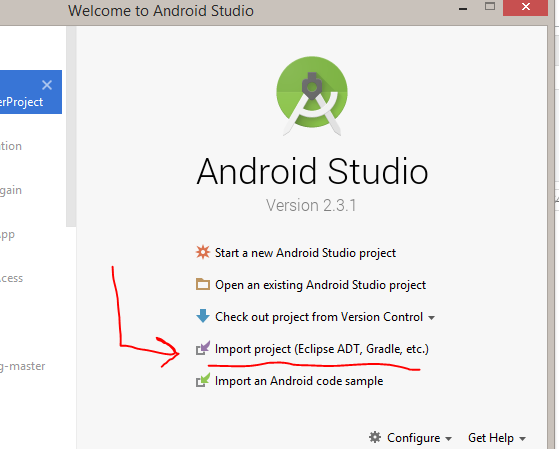
**Section 1 Execution Instructions:**

*Instructions for me to download and run your code. YOU NEED to show me screen shots of you doing this from your uploaded blackboard code.....this forces you to make sure that I can run your code. You MUST have the following screenshots AND give description on what to do: screenshot 1.1 = screen shot of your files uploaded to Project 1 turn in folder on blackboard*

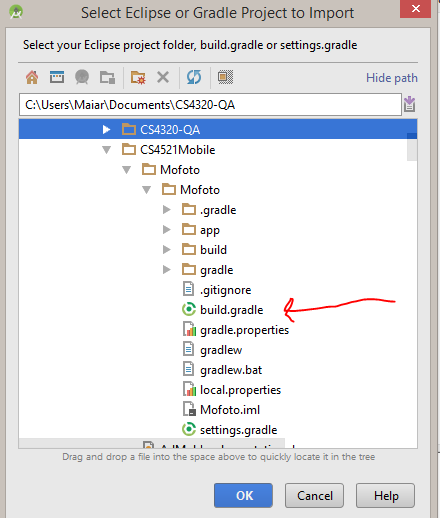


**screenshot 1.2 = directory view of "temp" directory you unzipped file to showing the unzipped files and directory structures**.

**Here is where I moved the downloaded zip file, and I can view all of the files in the project.**

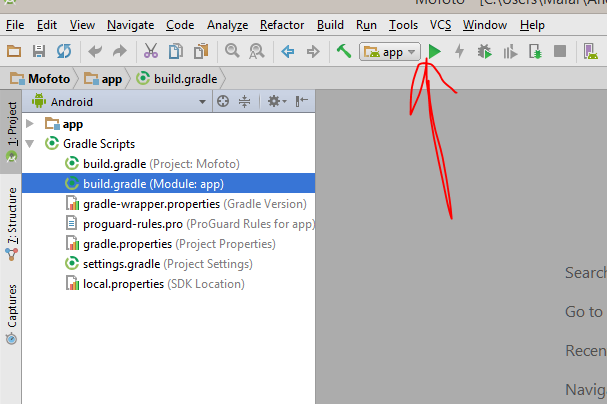


This is where you would click to view where the unzipped download is, and select it and open it in android studio.

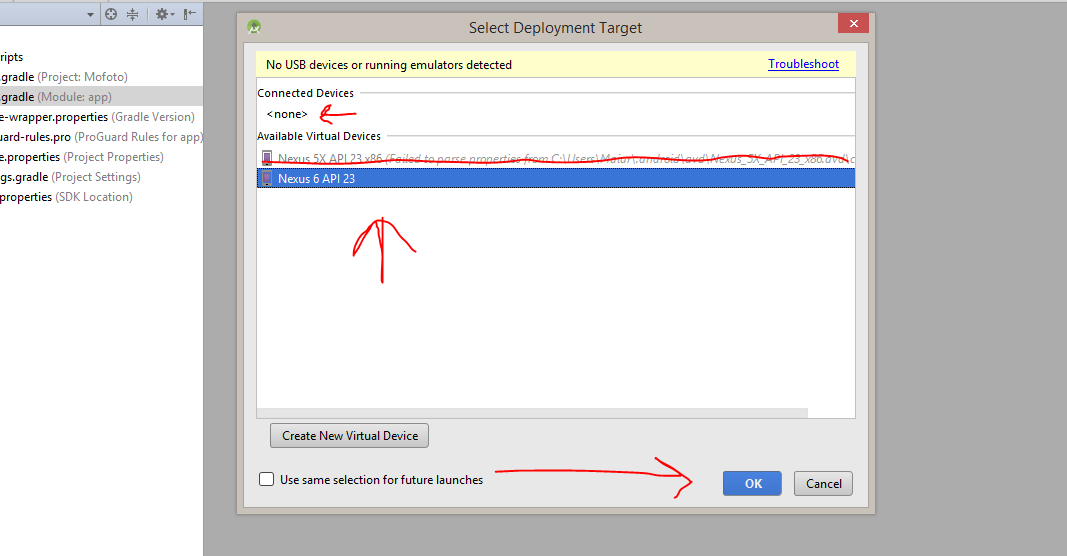


**screenshot 1.3 = Eclipse running where you have opened up project file in "temp" directory.**

**Be sure to unzip the downloaded application. Here you select buil.gradle to build the application.**



Click the run button, to run the application



**screenshot 1.4 = Android studio running the application - show screenshot of it running. If I must do something beyond simply hitting the "run" button, you need to give screenshots and step by step instructions.**

**This is the next step, select an emulator or your android device and then OK. The App will run successfully.**

**Section 2 Code Description**

*A describing how code is structured and the state of how it works. Give a description for each filename listed.*

*MainActivity.java –this is responsible for the user being able to login and sign up to the mofoto application. It authenticates the login and password.*

*ParseInit.java –this initializes the parse server I created with AWS, it enables the local datastore, and initializes the server and sets up the user object for my application. It takes in my application id, the client key, and the server.*

*UserListActivity.java –this gives functionality to the users feed which shows a list of registered users. Methods in this class are the startCamera, getPhoto from gallery, logging out functionality. It also asks permission to access the users’ photos and camera. It stores the images in the parse server Image Class.*

*UserProfileActivity –this is the users’ personal profile where their photos are stored and can be viewed. It creates a Linear Layout of Image Views unique to the number of photos uploaded by adding a new image view per upload. The Image Object for the Parse server is created here.*

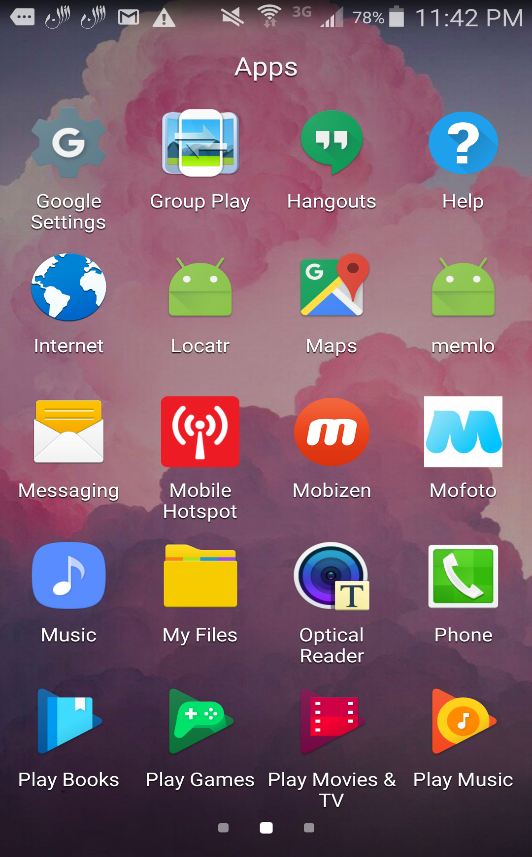
*Activity\_main.xml –this is the layout file for the main activity that sets up the login/signup activity in the application.*

*Activity\_user\_feed.xml –is a linear layout with a scroll view that allows the users to scroll throught their own profile and other users’ profiles and view the content.*

*Activity\_user\_list.xml – this shows the listView of all users in the application*

**Section 3 Testing**:

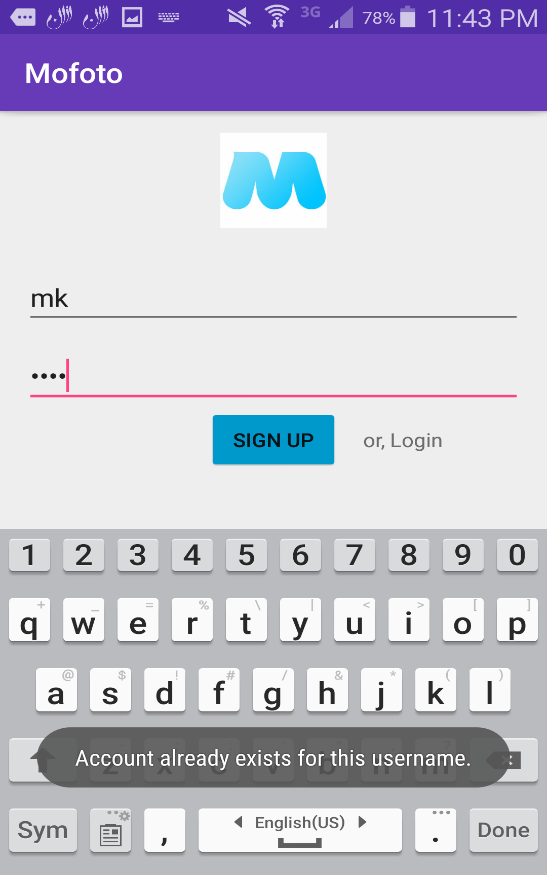
*here you give screen shots of you running the various stages of the program as detailed here:*

**section 3.1: starting application**

screenshot 3.1a= showing icon and resulting starting GUI.

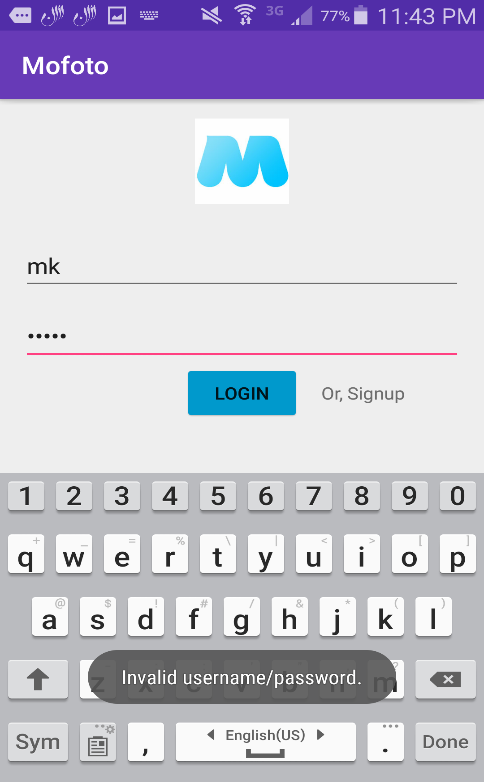
* The initial start screen with application icon

**section 3.2:**



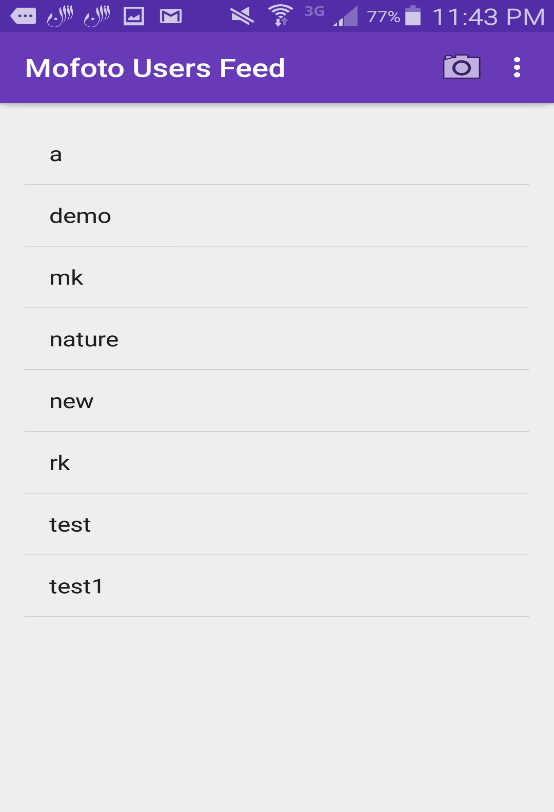
screenshot 3.2a = screen shot 1

This is the login/sign-up page, here the user mk already has an account so when they press sign up and error toast is given because the account already exists.



screenshot 3.2b = screen shot 2.

This is the screen where the user will log in if they don’t enter the correct password or username and error message will show, or else it will login the user.



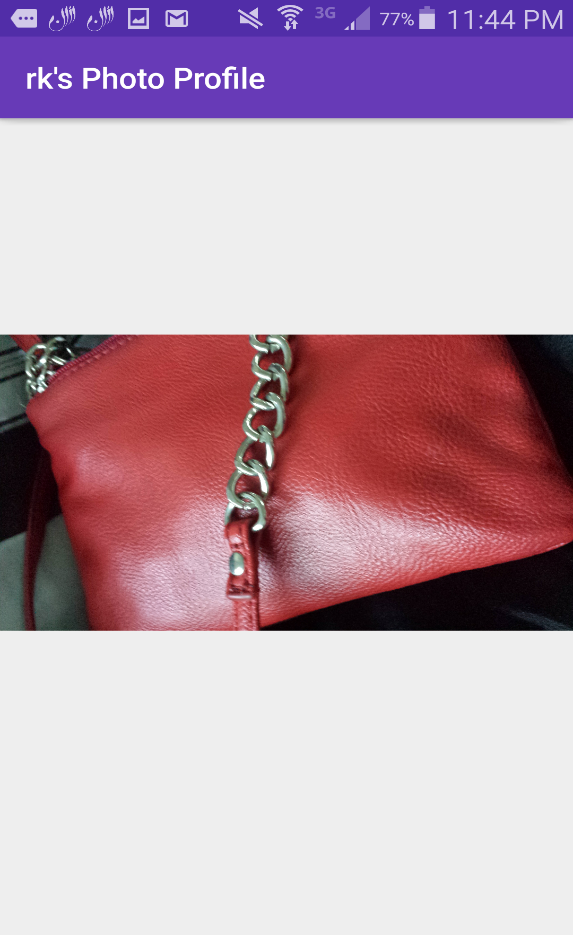
screenshot 3.2c

Once the user has successfully logged in they will be able to view the user feed, which includes all users who have an account with the application. They have the option on this screen to upload from the device gallery or access the Camera and take a picture to upload to their personal profile.

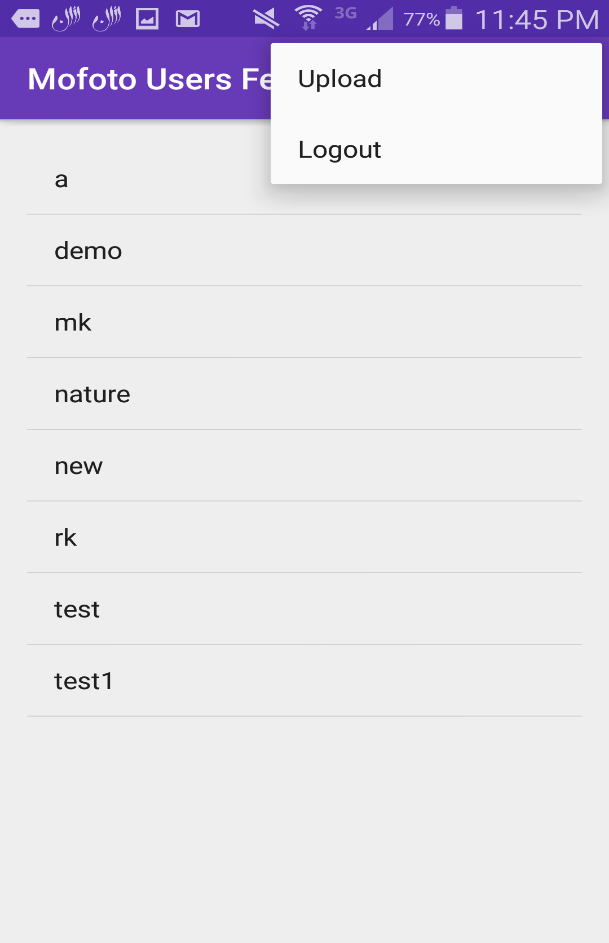


screenshot 3.2d

This is after the user mk clicked on the user demo’s profile page, here we can see that the name of the user in this case demo shows at the top bar and displays demo’s photo profile. This happens for each unique user’s profile. Also we can see that demo has one uploaded photo in their profile.

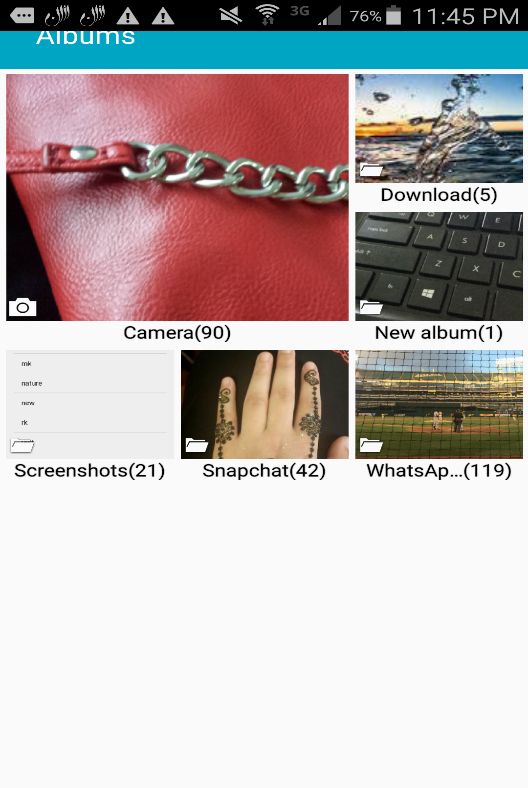
Screenshot 3.2e

This screen shows a user profile who used the camera to take a photo of my purse and upload directly to their Mofoto profile.



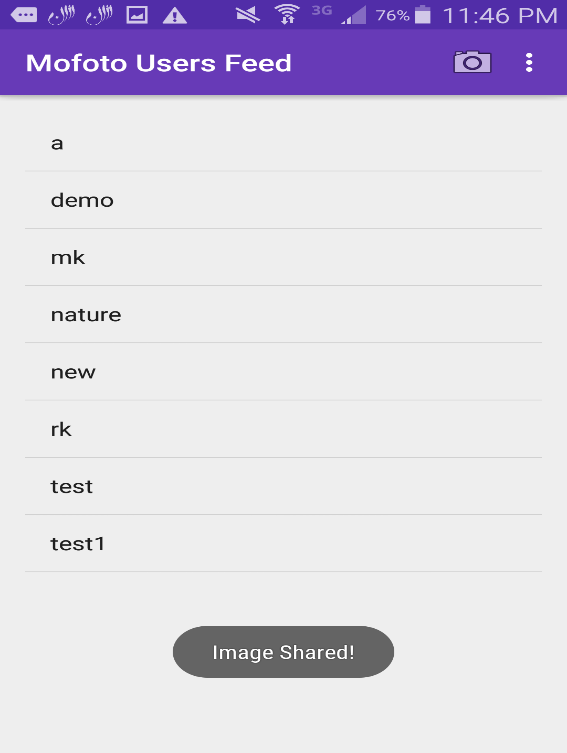
Screenshot 3.2f

In the users list page we can click down and view the menu which offers the logout functionality and the upload functionality, to the left is the camera button which access the device camera and allows the user to take a picture and upload it to their profile through the application.

Screenshot 3.2g

This is my devices photo gallery which the user will see once they click on the upload button in the menu drop down described above, Here they can browse through their photos and select which photo they would like to share on their profile.

Screenshot 3.2h



This message “Image Shared” appears after the user has successful uploaded a preexisting photo from the gallery or after they have taken a photo from the device’s camera. Of successful this toast will show and the users’ photo will be added to the database.

* The camera is used in the application for taking pictures directly and posting them to the user’s personal profile and saving the photo in png format to the database.

**Section 5 YouTube URL**

Video #1

<https://youtu.be/274U5Z85K_E>

Video #2 screen recording

<https://youtu.be/MmlXShgh88E>