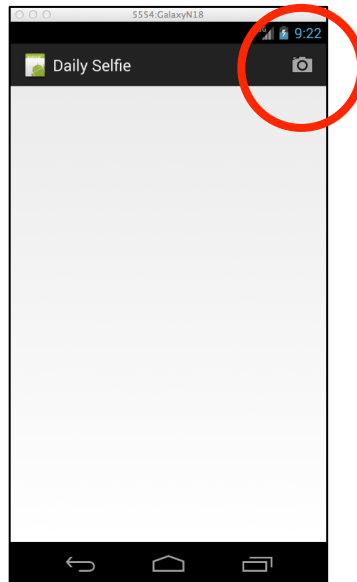


## Lab: Daily Selfie

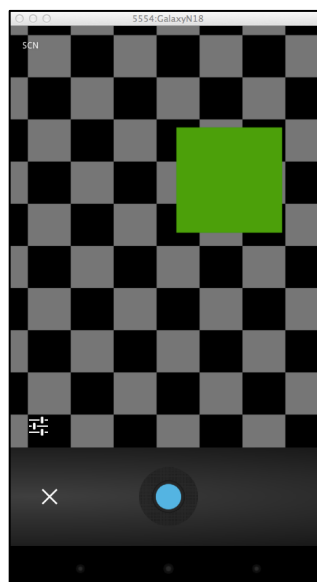
### Objectives:

Create an application from scratch that periodically reminds the user to take a selfie - a picture of one's self taken from one's device. Over time the user will capture many selfies and thus will be able to see him or herself change over some period of time.

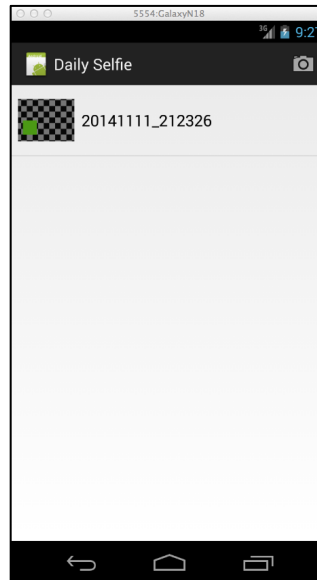
For example, here's a screenshot of an app that I created.



**Requirement #1:** If the user clicks on the camera icon on the ActionBar, the app will open up a picture-taking app already installed on the device. Here's a screenshot of my app after I clicked on the camera icon, and the Camera app has opened up. In this case, my AVD has a single emulated front-facing camera.

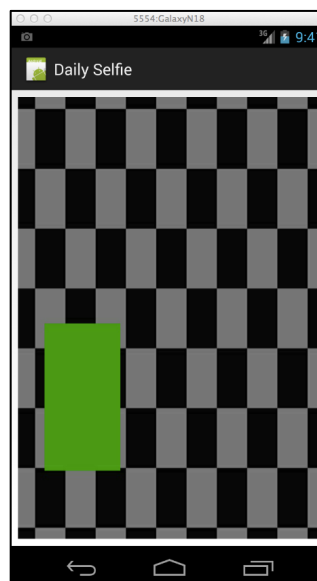


**Requirement #2:** If the user now snaps a picture and accepts it, the picture is returned to the DailySelfie app and then displayed in some way to the user along with other selfies the user may have already taken.



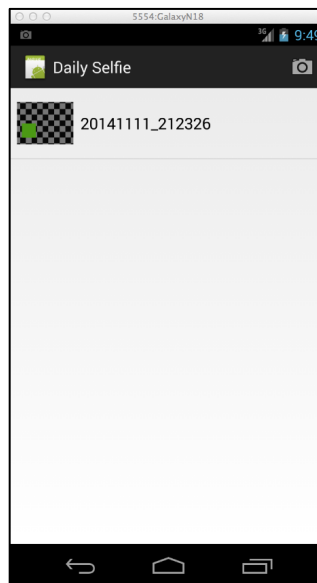
In this case I used a ListView to show all the selfies the user has taken, but you can use some other display approach. For instance, it might be useful to group selfies by the date they were taken. However, you do this, the user should be able eventually see a small view for each selfie.

**Requirement #3:** If the user clicks on the small view, then a large view will open up, showing the selfie in a larger format. Here's my app after, I clicked on the one small view shown in the screenshot above.

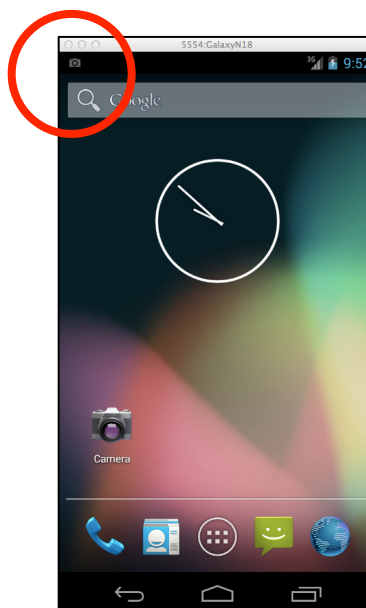


Hitting the back button in this case brings the user back to the ListView. The behavior of your app may be different depending on how you organize it, but there must be a way to go from a small view to a large view and then back to the original small view.

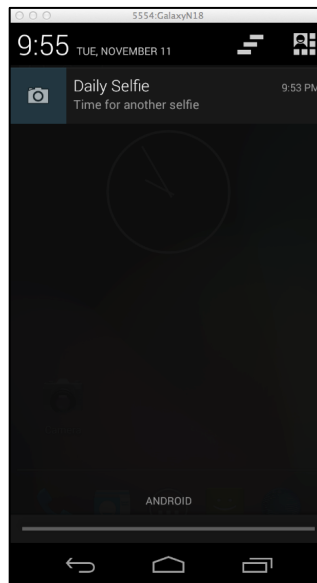
Whenever the user takes a selfie, the image data must be stored in some permanent way. **Requirement #4:** In particular, if the user exits the app and then reopens it, they should have access to all the selfies saved on their device. You may optionally include some way for the user to delete some or all of their selfies.



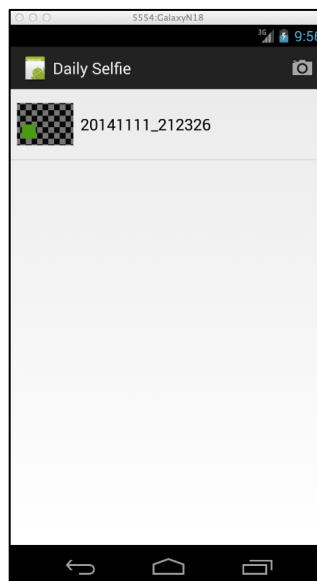
**Requirement #5:** Because the user wants to take selfies periodically over a long period of time, the app should create and set an Alarm that fires roughly once every two minutes. In a real app, this would most likely be set to a longer period, such as once per day. We will fire the alarms roughly every two minutes to make assessment easier. When one of these alarms fires, a notification area notification should be placed in the notification area, as shown below.



Pulling down on the notification drawer should expose a notification view.



Clicking on this notification view should bring the user back to the application.



**Tips:** I **strongly recommend** that you read the following information on how to take pictures. <http://developer.android.com/training/camera/photobasics.html>. I also **strongly suggest** you use parts of the sample code to linked on this page to keep the project effort manageable. Pay particular attention to the `setPic()` method, which shows you how to read a Bitmap from a file in a memory-efficient way.

Use your creativity to design your app and user interface. Design your own visual layout and navigation approach. Just make sure that your approach meets all the requirements.

Take a look at the `dailyselfie.mp4` screencast for a run-through of the application's features.

### Submission

To submit your work you will need to export your DailySelfie project as a compressed zip file. Then you will submit this zip file to the Coursera system. Grading for this assignment will be done using Peer Assessment. That is, students will review each other's code and give feedback.