Project 2: CrimeIntent Pro Edition (120 points)

Tian Guo, 2017/11/04

1 Project Overview

The aim of this project is to get you familiar with Android apps that use the RecyclerView, Android Camera, Images and Storage, Content Providers and third-party Libs. A few important things to know before starting this project.

- If you have any questions regarding this project, please post it in the forum with the part you have questions with (using line numbers to the left).
- Inside the same zip file, there should be two code zip files and a project description pdf. CriminalIntent.zip is the starter code for the main project while, facetracker.zip[1] is the starter code for face detection.
- This project is due by 10:00pm on **Nov. 12** and accounts for 10% of course scores. Please do start early on this project!
- Each student must complete this project individually. However, I do encourage you to share ideas, advice and resources with each other. Please contact me if you have questions about what constitutes appropriate collaboration.

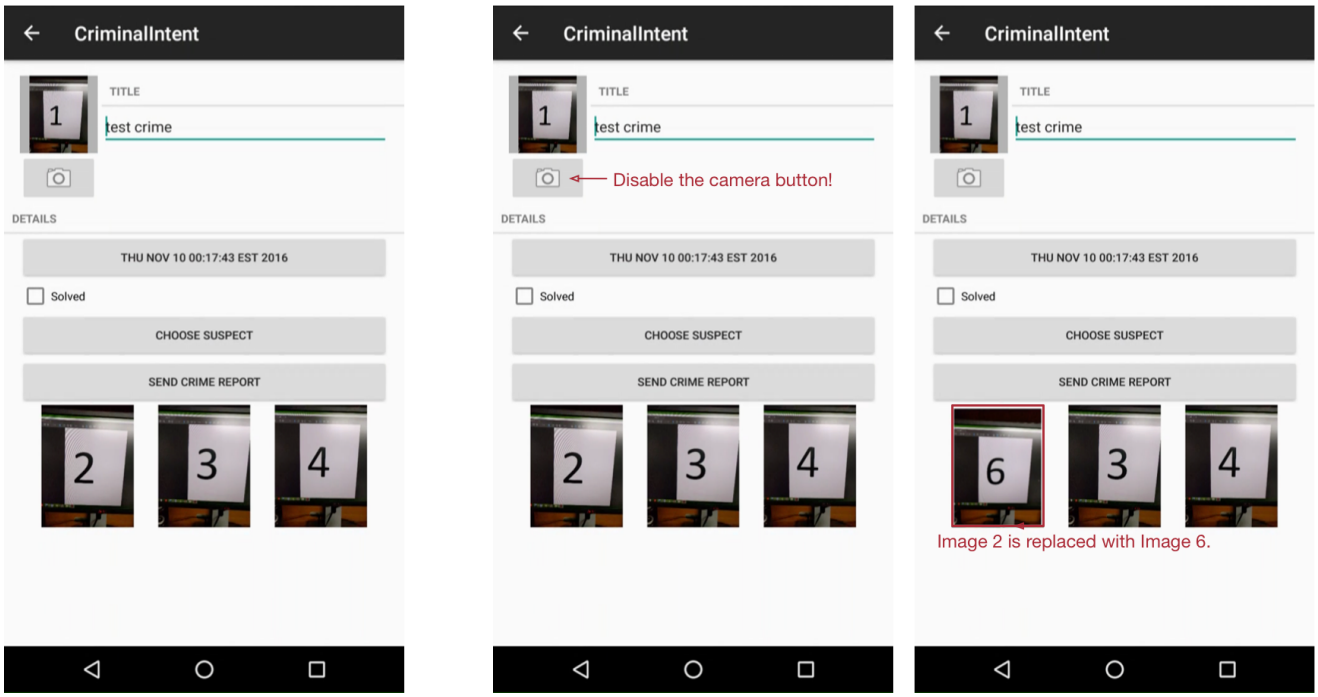
2 Project Submission

The require submission of project two will include your implemented Android project code, a writeup document about how you implement the feature, an Android APK, and a captured screen video file.

- **Your Android project code.** Before submitting, please take the time to make sure you have a clean build by following the same guide from Project 1 about clean your project build . Make sure you have documented your implementation adequately with in-line comments.
- **Writeup document.** Include a writeup document that describes the high-level steps you take in implementing each requirement. It is crucial to explain what you have done, especially in scenarios that you didn't fully implement the features. This helps teaching staffs to decide how many partial points to assign.
- **APK.** Build and generate an APK for your app. Check out this post if you don't know how yet.
- **Screen Capture.** Record session of running this app with all the implemented features on an Android device. If you are using a physical device, you can simply download a free screen capture app, such as `AZ Screen Recorder`. If you are using an AVD, you would have to first install Google Play on the emulator and then download the app; alternatively, you can follow this guide to setup an AVD that comes with pre-installed play store .Also check out this link for alternative ways to perform screen capture. Note please keep your video file as small as possible, for example you can reduce the video quality to 480p; alternatively, you're welcome to upload the video capture to public hosting sites such as YouTube and provide us the URL link.
- **Zip file.** Lastly, Create a zip file, named `cs4518Project2.zip`, from a directory that contains your modified Android project code, the writeup document, your APK, and captured screen video file. Submit `cs4518Project1.zip` through Instruct Assist[2].

[1] Face detection concepts are explained in https://developers.google.com/vision/face-detection-concepts and You can also take a look at other sample codes at https://github.com/googlesamples/android-vision

[2] `https://ia.wpi.edu/cs4518/files.php`



(a) Feature Rich CriminalIntent. (b) The only four images. (c) The favorite four images.

Figure 1: Four images in total.

3 Project Requirements

3.1 One photo per crime isn't enough, How about four? (30 points)

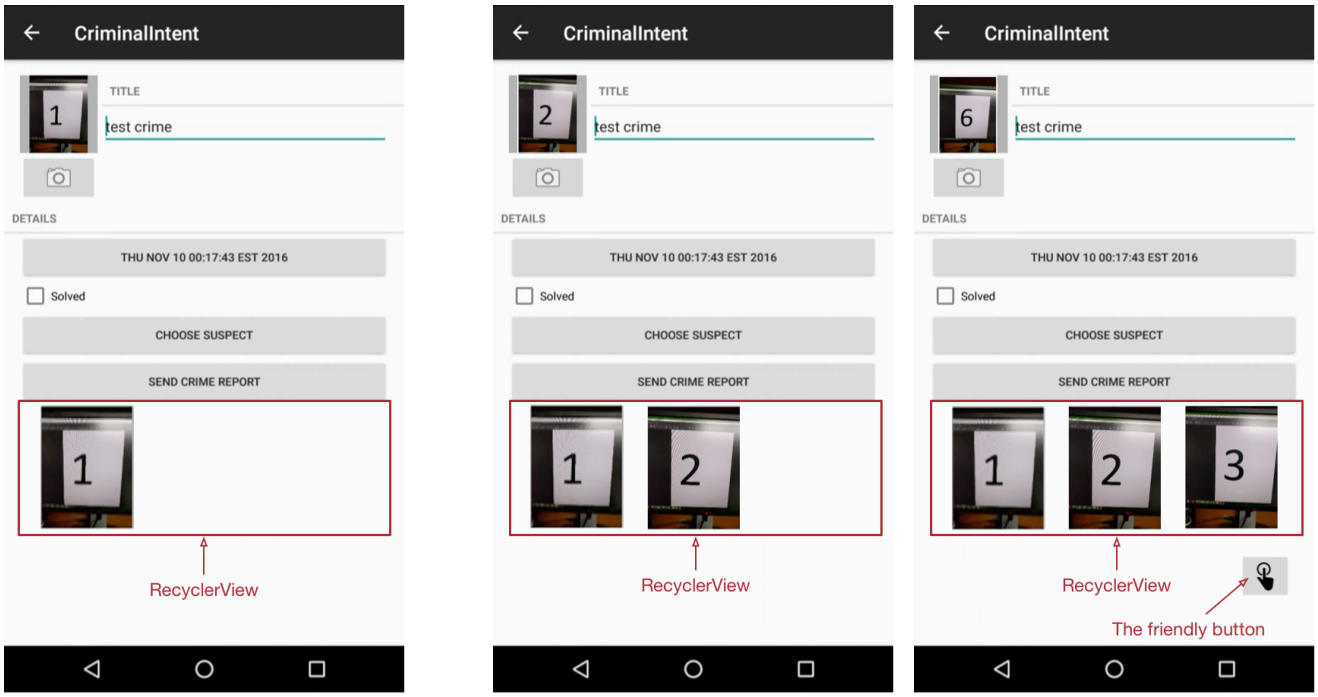
Currently, the CriminalIntent app only allow user to store one image with each crime—taking a second image replaces the first image with the second. How could that be sufficient to record a vivid pet crime? We want at least three more photos to remind us how our cats are stealing food, milk or even water from the counter!

- (10 points) Therefore, in this section, you will modify the code, so that it can store three more images below the "SEND CRIME REPORT" button as shown in Figure 1(a). The numbers in each image indicate the order they were taken.
- (10 points) Next, once users have taken all four images, you will then disable the camera button. (Really, if you allow users to take more, they will fill their phones with all the cat pictures!)
- (10 points) But what if users change their minds about which four images they want to store with the current crime? For this, you will make all four images "clickable". When user clicks any one of the four images, the camera app will be launched again and allow user to replace For example, if users want to replace the image 2 with image 5, they will simply press the image with number 2, and image 5 will be shown in where image 2 was. And image 2 will be disassociated and delete from this crime.

3.2 As many photos as user wants (50 points)

Well, restricting the number of photos we can store in a single crime is convenient design and implementation for developers. But users aren't so fond of the lost freedom. Therefore, in this section, we want to empower the users and provide them the flexibility to take as many photos as they want for each crime.

- (10 points) "There ain't no such thing as a free lunch". We will only allow paid user to take as many photos as they want. After all, we have to redesign our database, store all the images and may cost us money if one day we decide to backup all the crimes to the cloud! To mimic the paid user, you will implement a setting UI using SharedPreferences that allows users to switch to paid user status.



(a) (b) (c)

Figure 2: RecyclerView Illustration.

- (20 points) For such paid user, you will implement a RecyclerView with LinearLayoutManager below the "SEND CRIME REPORT" button. Every time users press the camera button, an image will be displayed both in the imageView above the camera button as well as in the RecyclerView, see Figure 3. Note that in Figure 2(c), image 6 will be displayed once users scroll to the left.

- (20 points) Scrolling through ten's crime photos might be fine, but what if users go so overboard that they took hundreds of pictures? To handle this scenario, you will implement a "user-friendly" button, as shown in Figure 2(c), that will become visible right under the RecyclerView after more than 6 images were taken for each crime. When user press the button, they will be able to see all the images taken so far for the current crime. You will be manage these images in a GridLayoutManager, again using RecyclerView.

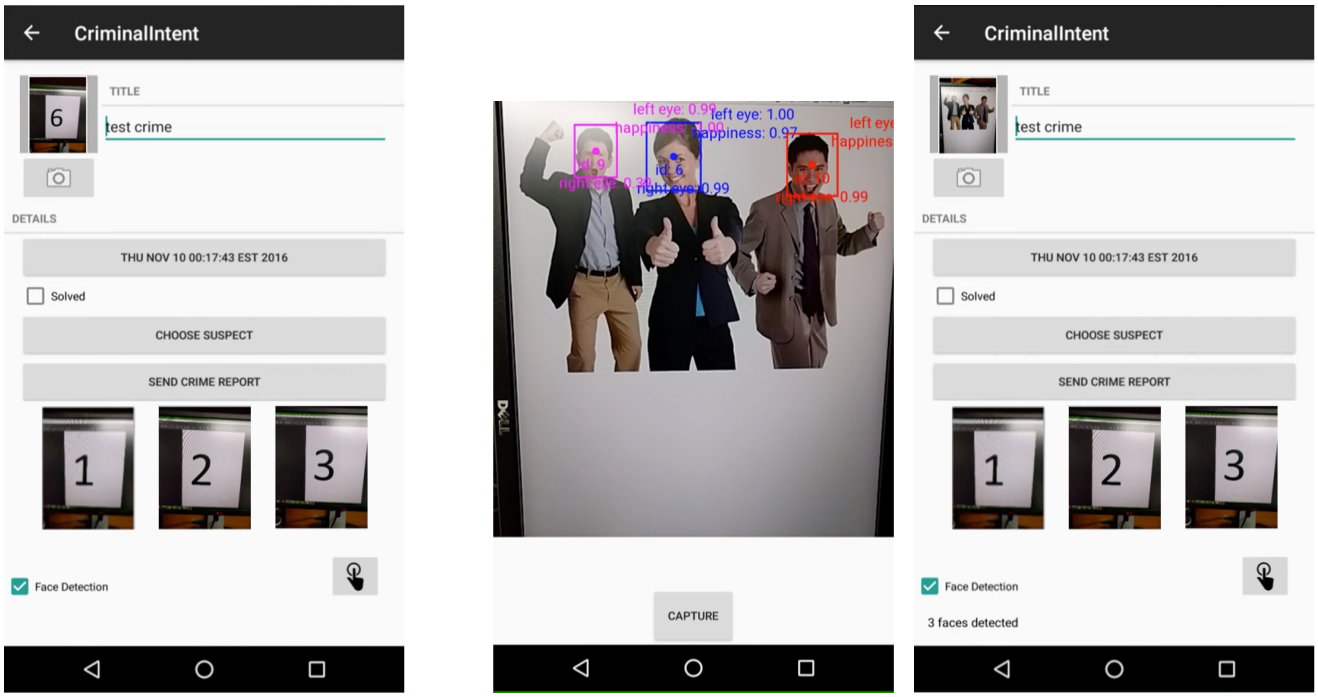
3.3 Where are all the crime photos? (20 points)

What good of a crime app it is if users can't browse all the photos in their favorite Android photo gallery app? To make users happy, you will modify the behavior of camera button so that images will show up in the Android's Gallery. Note that, we still want to display all the image previews in our own CrimeIntent app as shown in Figure 2. It is just now all the photos are stored in different places.

3.4 Who are involved in our investigation? (20 points)

- (10 points) If users want our help in counting the face, we should help them. For this, you will implement a checkbox that allow users to specify whether they want to put face detection into work or not. If the "Face Detection" is enabled, when user press the camera button, our app will display the camera preview using the sample code facetracker.zip. See Figure 3(a) and Figure 3(b).

- (10 points) After users capture the images, these images will be saved in the same way as when "Face Detection" is not enabled. In the detailed crime screen, a summary of the total number of faces detected across all images will be displayed below the checkbox. See Figure 3(c). That is, if we take another photo with two person, the summary will be updated to "5 faces detected".



(a) (b) (c)

Figure 3: Face Detection Illustration.

3.5 Swipe to delete an crime image (10 points)

What if users take an undesirable photos and hit the okay button by accident? Currently, CrimeIntent does not support delete such images. To provide this feature, you will make appropriate changes so that users can delete unwanted photos from both the Linear and Grid RecyclerViews. There are a few ways to support item deletion in RecyclerView: (1) refer to BNRG Page 335 Challenge "Adding Swipe to Dismiss" to allow users to delete an crime item from the crime list; (2) https://medium.com/@fanfatal/android-swipe-menu-with-recyclerview-8f28a235ff28; (3) http://blog.teamtreehouse.com/contextual-action-bars-removing-items-recyclerview. As long as users can delete the images, you have the freedom to choose however you want to implement it.