**Android Studio**

Download from <http://developer.android.com/sdk/index.html>

That will install both Android Studio and the most up-to-date SDK.

-Note: you may need to install older versions to be compatible with the current code

Download the app source code from my website and unzip the folder.

Once Android Studio opens, you can import the project, and select the build.gradle from the unzipped project folder.

Java source files can be found here: FindingFido\app\src\main\java\com\linseykate\findingfido

XML Layout files can be found here: FindingFidoCopy\app\src\main\res\layout

**Getting started with Firebase**

Go to firebase.com.

Sign up for an account or log in (you will need a Gmail account).

Create a new app.

It will give you a URL. Use that URL to update all Firebase references in the code.

-Looks like this: Firebase ref = new Firebase("https://findingfido.firebaseio.com/");

Documentation for the functions can be found here: <https://www.firebase.com/docs/android/api/>

In the menu on the left, there is a Data tab, which will show you all the records currently stored. You can manually add, delete, and modify from here if necessary.

Also in the menu on the left is a Login & Auth tab. Scrolling down to the bottom of the page, you will be able to see all email addresses that currently hold accounts. You can manage them from there. Resetting passwords would require some additional code in the app.

**Running the app for the first time**

Plug the phone into your computer.

You will need to enable USB debugging on the phone itself (usually tapping Developer Options 7 times will do this)

On the toolbar of Android Studio, there should be a green play button (a triangle).

When you click that, a window should pop up, and the phone you connected should show up in the first section.

When the phone is highlighted, you can continue and the app will start running on the phone.

**How to use the app**

First, make sure you have the Wi-Fi on.

The app will open to a login screen. If this is the first time using the app, you will need to create an account. Fill out your email and a password, and tap the Sign In button.

The app will then open to the main page. In the upper right corner, there is a bell and three dots. The bell will open to the Notifications screen, and the three dots will show a settings menu. The Settings option will open the notifications preferences, and the Logout option will return you to the login screen.

There will also be two tabs: Photo and Browse.

The Photo tab is the default. You can tap on the Upload button to open the gallery and select a picture, or tap the Take a Photo button to launch the camera.

Either option will eventually get you to a screen with the picture and a blue-outlined box. This is where you can drag and resize the box to select the area of the photo to crop.

After cropping the photo, a larger-scaled version of the cropped photo will appear. Tapping, holding, and dragging an area on the screen will draw a white box over the picture. Tapping the Next button will move to the next prompt, and tapping the Undo button will clear the photo area. The arrow buttons in the bottom right corner can be used to shift the drawn box one pixel at a time. After three prompts, the Next button will turn into a Finish button, indicating it is the last box to be drawn.

After tapping the Finish button, you will return to the main screen, this time with some additional options shown. The photo will appear, and below it, a slide bar. You can drag the circle from left to right to approximate the size of the dog. Below that is a group of radio buttons. Use them to select the color that is most prominent on the dog’s face. Then, tap Submit.

**\*\*Note:** the app may crash here. It is likely something to do with a null pointer not getting reset somewhere. I think I have it narrowed down to an issue with opening the camera, then going back to decide you’d rather upload a picture, or using the Logout option. I am fairly confident the crashing could be fixed be resetting some data members on these events. If the app does crash, simply closing out of the app and reopening it has resolved issues in the past. However, the photo still uploads to Firebase, so you can go into the Data tab mentioned above and delete that record if you wish.

Upon tapping submit, a new screen will appear. The picture you just uploaded should show up at the top. If any matches were found, they will appear in a grid view on the bottom half of the screen. Tapping a picture will delete both that picture and the one you just uploaded from Firebase, since they were a match and no longer need to be found ☺

You can then tap the back button to return to the main screen.

The other tab on the main screen is the Browse tab. This shows a grid view of all the pictures currently saved in Firebase. If you just uploaded a picture and it did not have a match, you will need to tap the Refresh button to repopulate the grid view. However, if you did find a match and tap Refresh, it will not update correctly. You will need to close out of the app altogether and reenter to see the updated view.

Also, upon finding a match, the Notifications bell will now display some data when you tap the bell. However, this data will disappear as soon as you close the app; it will not persist through different sessions.

**Closing Notes**

If you keep the phone plugged in while you run the app, a log will appear at the bottom of the Android Studio window. This can be helpful in doing some debugging. The app currently has a few log statements that print some messages in that log, though it can be hard to sift through. If the app crashes while the phone is still plugged into the computer, it will also print a fairly detailed error message in red and suspend the log. It will usually give you a detailed error message (that is typically very Google-able) and sometimes even a line number and the specific file that is causing the error.

Some screenshots and video of the app working can be found on the Blog page of my website.