**Social Media App – README**

This is an App with many features and functionalities. Hence this detailed README File.

The app makes an honest attempt to be “Androidy” – including the colour schemes and UI interfaces.

1. Application obtains some data from a network source – Gets a user selected image from the internet. User has to type in the URL in the format shown.

2. Application must store data in a local sqlite database – Stores the image in an sqlite database. There is a Database Explorer button that allows user to see what is in the database, uses a library to do this: <https://github.com/sanathp/DatabaseManager_For_Android>. This is a library which I will be looking to incorporate in all my Android database apps.

3. Application must facilitate the sharing of data from your application to another - Provides buttons to share a picture or to share text. The image sharing function allows the user to take a picture from the camera and share it, or to retrieve an image from the database and share it. The text sharing routine allows the user to type in and share text.

4. Application must allow other applications to share data with it - handles accepting and displaying images from other apps.

5. Use appropriate Android framework components to display your data in some sort of view – Displays Images and Histogram in relevant views.

6. Provide the ability for your user to search for specific data contained in application – Implements SearchView and shows the image retrieved from the database.

7. Application should provide some kind of useful processing of the data- Application provides a histogram of the image retrieved from the database. It relies on OpenCV Manager – gives the user a prompt to install OpenCV Manager. It yes – all good; if no – tells the user that it cannot show the histogram. This is the recommended way to implement OpenCV Manager, there is a ‘static initialisation’ method which does not need the installation step, but that is deprecated so using ‘async initialisation’ in my solution.

Also tried to implement exif data display, however, discovered that the downloading of the image from the net strips the exif data as implemented, would need to implement metadata-extractor library to get the image exif data while downloading. Could have done this but decided to go with the (more attractive) histogram. Both solutions were very interesting to work through.

There is extensive error checking implemented – to ensure that required data is entered correctly and with prompts and error messages provided where necessary. This is important for the stability of the solution.