GitHub Username: jasperjohns

WaterMarker

Description

At live events (sporting or otherwise) professional photographers like to post some images of the events virtually “live”. While that is relatively easy if the photographer is using a camera which supports NFC - it is harder for the photographer to add some sort of branding or copy right notification image to the post. The application WaterMarker will enable the photographer to just do that one their phone before posting the image.

Intended User

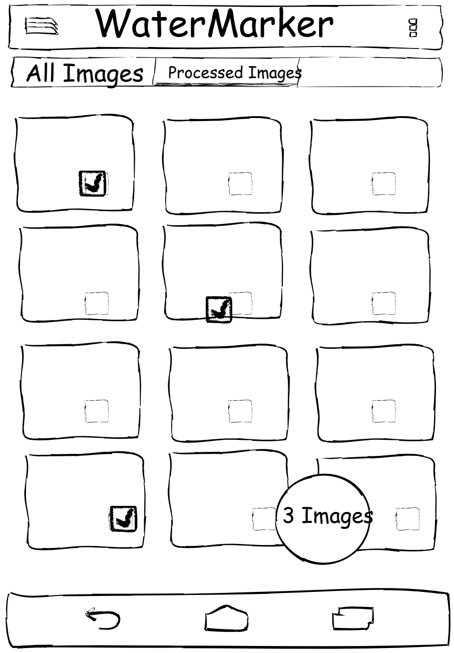
The applications is intended towards photographers who would like to protect their digital property before posting pictures on Social Media

Features

It will allow the user to browse the pictures on the phone
Select the images that need the water-mark applied
Save the “water-marked” images to a “Application Specific” directory
Allow the “water-marked” images to be “Shared”.
Allow the user to upload and change the “water-mark image”
Notify the user that the process of converting the images is complete and the images are ready to be posted.

User Interface Mocks

Screen 1

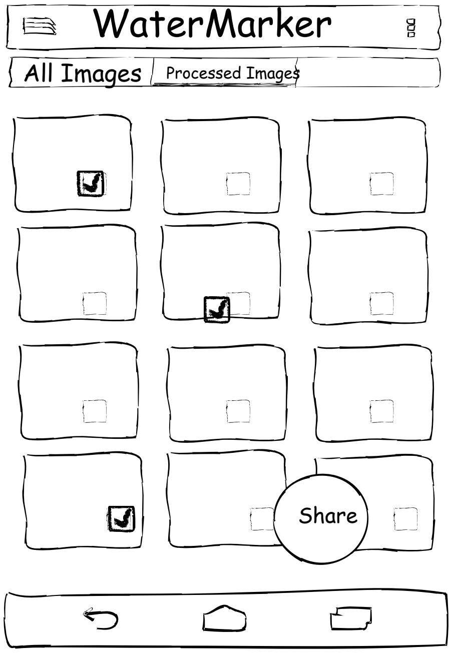


View all the pictures on the phone, excluding the processed pictures. On clicking the specific image in the list – a pop-up screen will display a slightly larger version of the image so the user can take a better look at the image.

To select the image to be processed the user would check the image that was super-imposed on the image.

When the user wanted to process the image they would click the FAB Button to initiate the process.

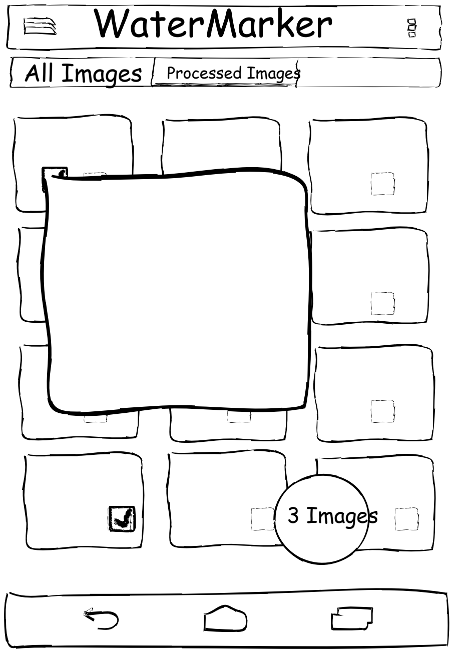
Screen 2



The second tab will only list the images that have been processed

The user will be able to Share the images in this folder via Social Media

Screen 2 Pop-Up Window



Key Considerations

Corner Cases:
Closing of the pop-up images will be done by just touching the pop-up image

In order to allow comparison between images – multiple pop-up will be displayed at the same time.
Via settings or the equivalent the user can choose what image they want to use as the “water-mark” image.

Libraries Used:
The Android-Universal-Image-Loader (https://github.com/nostra13/Android-Universal-Image-Loader)
The Page-Sliding-Tab-Strip (https://github.com/astuetz/PagerSlidingTabStrip) will be used to provide the sliding tabs for the application.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Create a shell application which will support Material Design (via AppCompat) and

Configure the libraries needed in the project.

Add Code to initiate the libraries (if needed) and write stub functions to make sure that the libraries are working as planned.

Task 2: Implement UI for Each Activity and Fragment

Build UI for MainActivity. This Activity will be house the Page-Sliding-Tab Strip and a ViewPager into which different fragments that correspond to the different tabs will be loaded.

Build the layout for the fragment based around the grid view which will display all the images in the Media Store – except for the ones that have already been processed.

Build the grid item layout for each of the images. A check box needs to be super-imposed on each image in the grid to allow for selection and de-selection.

Build a layout that will be used to display the image in a “pop-up” window

Build the layout for the fragment that will list all the images that have been processed -
Build the layout that will allow for selected image to be Shared via Social Media

Build the layout that will allow for a specific image to be used as the “water-mark” image via “Settings”

Task 2: Class Development

Create an Image Adapter to tie the selected images from the Media Store to be displayed by a grid view

Build the “View Images” Fragment that will load the images into the view

Build the “Processed Images” Fragment that will load the “processed” images into the view

Write an Async task to do the processing of the images

Send a notification when the process is complete

Write code to initiate sharing of content via Social Media

Task 3: Pop-Up UI & Settings

Tie the pop-display to an on-click of the image in the view

Tie the on-click of the pop-up with the closing of the pop-up
Add the ability to select the “water-mark” image via “Settings” and use that as part of the process.