Com1032 – Mobile Computing

Coursework 2 Report

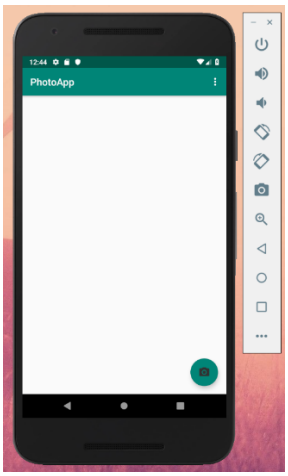
Contents

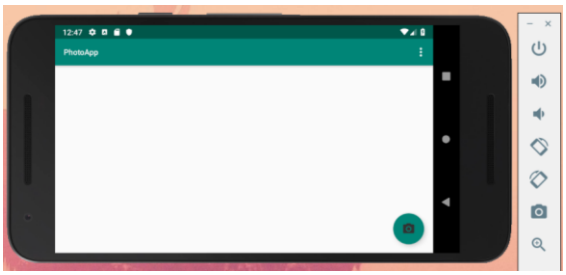
1. Purpose

PhotoApp is an Android application that lets the user take photos and then displays them all on the screen. The app works and any Android device with API 15 or higher (almost 100% of android device in the world), it doesn't require any connection to Wi-Fi or data usage. It makes use of the camera sensor application that is already implemented on the device. The photos taken are not stored within the shared external storage of the device but within the internal storage of the app; Therefore, if the app is uninstalled, the photos will be removed with it. The user can choose to use the app in portrait or landscape orientation. When a photo is taken it is added to the main screen where all photos are displayed, these photos can be selected and removed by the user if he/she wishes so.

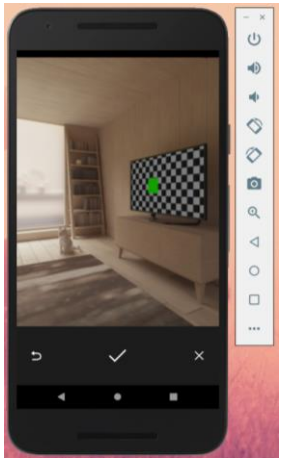
2. User guidance and navigation

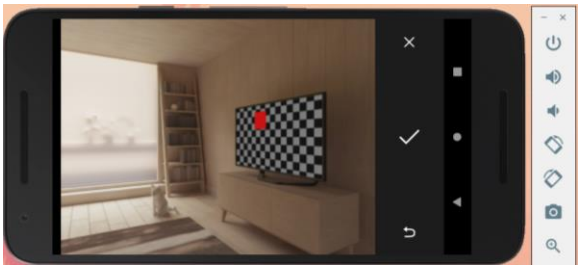
When the user first opens the application, he will see a blank screen with a 'FAB' button on the bottom right. This screen/activity is used to display all the photos that will be taken by the user.



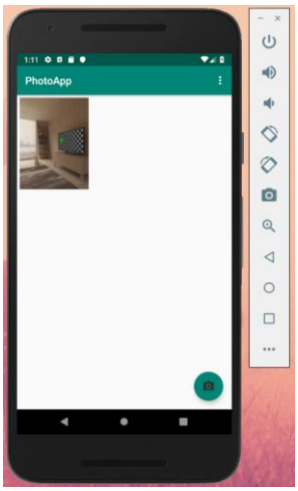


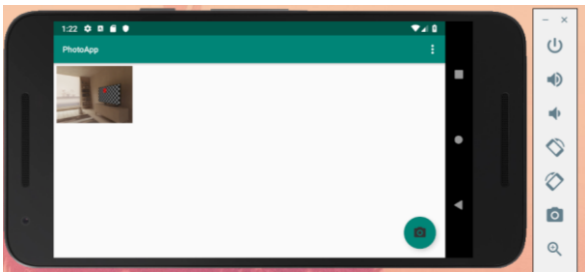
By clicking of the ‘FAB’ button with a camera icon, the camera application of the device is started. The app will wait until the user takes a photo and confirms the selection or leaves without taking a photo. (The left/bottom button will let the user take another picture, the middle button will confirm the selection and add the photos to the app, the right/top button will cancel the selection and leave the camera app)



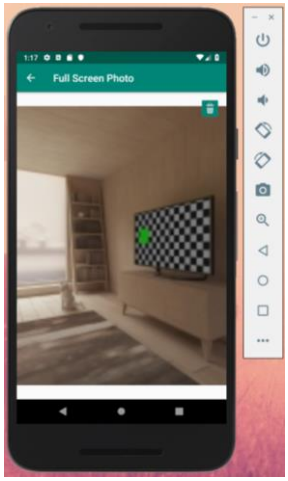


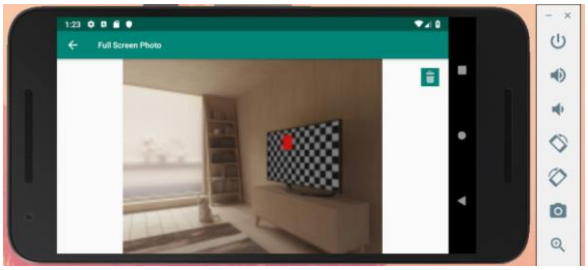
Once a photo is accepted, it is displayed on the main activity of the app. The user can now choose to take another photo, click on the phototo select it, or leave the app.



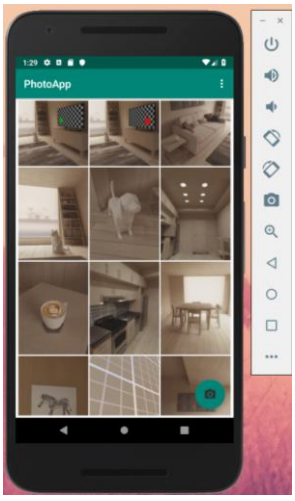


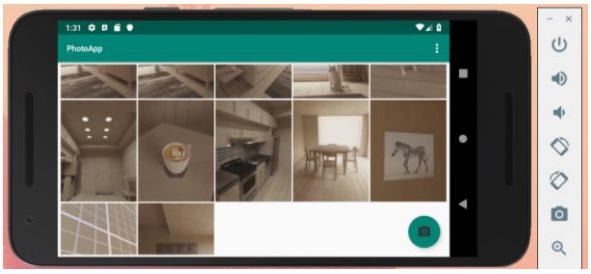
By clicking on a photo, a new activity is started, and the photo is display on the whole screen. The user can now press the back button (top left) and return to the state where he was before, or he can choose to delete the photo and remove it from the app.





After having added many photos to the app, notice how the main activity contains a recycler view that enables the user to scroll through all the photos.





3. Coursework objectives

3.1 Camera Sensor

In order to use the camera sensor on the device, we need to access the camera application already installed. To do this, I used an implicit Intent from the main activity, by passing as a parameter the “ACTION_IMAGE_CAPTURE” constant that belongs to the MediaStore class.

By calling the method “startActivityOnResult” and passing the previously created intent along with a request code, the camera app is launched. Using startActivityOnResult rather than startActivity ensures that when the user confirms his photo, the PhotoApp is brought into the foreground again and the data taken from the camera app is passed to this app.

The result code is needed as, in the “onActivityResult” method, different intents may be passed, not only intents from the camera app. Therefore, to avoid any error, we identify the intent from the camera app with request code 0. We will see in the next sections what other intent are passed in the method using another request code.

3.2 Threads

The app uses an implementation of threads through the AsyncTask class. This is used when a photo needs to be saved or removed from the internal storage. We do not want to slow down the main thread with a task can be done in the background.

Therefore, whenever a photo needs to be added or removed, a method is called which creates a new instance of the AsyncTask class. In the “doInBackground” method, the internal storage is accessed, and the file is added or removed. We don’t need to use any of the other methods available from the class as we don’t need to pass anything back to the main activity.

3.3 Second Activity

A second activity is started whenever the user clicks on one of the photos displayed in the main activity. This because a completely new interface with different functionalities needs to be used. This activity is a child of the main activity as it is concerned with one of the objects displayed in the main activity.

The activity is started using startActivityForResult with an intent containing extras that refer to the list of bitmaps and the position of the selected photo in that list. It also has a parameter request code 1, to ensure that the intent passed back is not confused with the camera app.

In the second activity the photo is displayed using an imageView. If the user presses the delete button, an intent is passed back with the position of the photo in the list, the main activity then uses a thread to delete the photo. If the user leaves the second activity using the “back button”, no intent is created and therefore the list remains unchanged.