

**MOBLILE APPLICATION DESIGN AND DEVELOPMENT**

**Assessor’s name:** Ho Nguyen Phu Bao

**Learner’s name:** Phung Tan Sang

**FPT ID:** GCS190084

**Greenwich ID:** 001247397

**Class:** TCS2207

**Subject code:** COMP1786

**Table Contents**

[1 Add function 3](#_Toc120028261)

[1.1 Basic information 3](#_Toc120028262)

[1.2 Screenshots 3](#_Toc120028263)

[1.3 Code 4](#_Toc120028264)

[2 Previous function 6](#_Toc120028265)

[2.1 Basic information 6](#_Toc120028266)

[2.2 Screen shots 7](#_Toc120028267)

[2.3 Code 8](#_Toc120028268)

[3 Next function 8](#_Toc120028269)

[3.1 Basic information 8](#_Toc120028270)

[3.2 Screen shots 9](#_Toc120028271)

[3.3 Code 9](#_Toc120028272)

[4 Take photo by camera 10](#_Toc120028273)

[4.1 Basic information 10](#_Toc120028274)

[4.2 Screenshots 11](#_Toc120028275)

[4.3 Code 13](#_Toc120028276)

**Table of Figures**

[Figure 1 - Add function 4](#_Toc120028296)

[Figure 2 - Previous function 8](#_Toc120028297)

[Figure 3 - Next function 10](#_Toc120028298)

[Figure 4 – Camera 12](#_Toc120028299)

[Figure 5 - Files option 13](#_Toc120028300)

[Figure 6 - Photos option 13](#_Toc120028301)

1. **Add function**

**1.1 Basic information**

|  |  |
| --- | --- |
| Student name | **PHUNG TAN SANG** |
| Who did you work with? Note that for  logbook exercises you are allowed to  work with one other person as long as  you give their name and login id and  both contribute to the work. | **Name: Phung Tan Sang**  **Login ID: ps5179c** |
| Which Exercise is this? Tick as  appropriate. | Create "Add" button that is used to add new images to the gallery from copying and adding links. |
| How well did you complete the  exercise? Tick as appropriate. | 🞏 I tried but couldn't complete it.  🗹 I did it but I feel I should have done better.  🞏 I did everything that was asked.  🞏 I did more than was asked for. |
| Briefly explain your answer to question | I have created an Add button that adds a new image link to the gallery. |

* 1. **Screenshots**

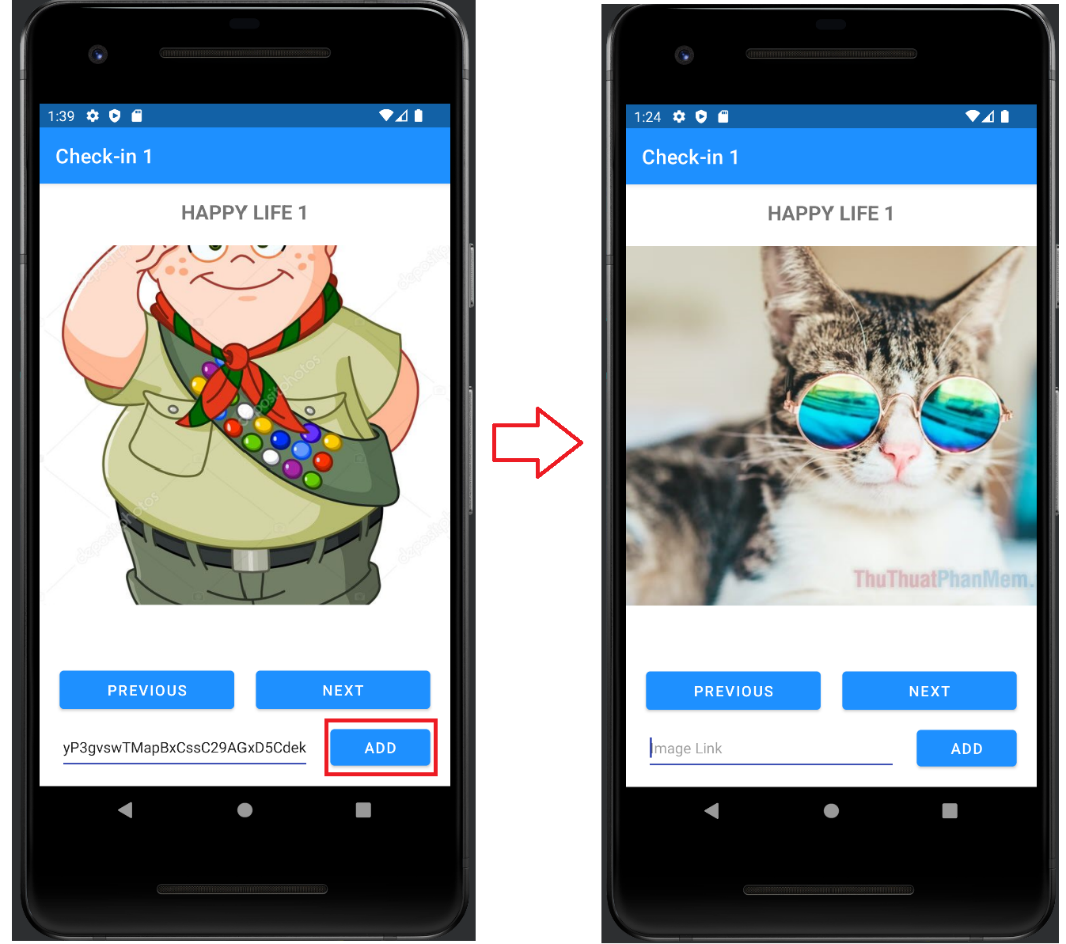


Figure - Add function

The user needs to enter the image link in the "Image link" field and click the "Add" button  to add a new image to the gallery. However, the words "Added successfully." appears after clicking "Add" to finish adding the image.

* 1. **Code**

<EditText  
 android:id="@+id/etImageLink"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:hint="Image Link"  
 android:inputType="textPersonName"  
 android:textSize="15sp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/btnAdd"  
 app:layout\_constraintEnd\_toStartOf="@+id/btnAdd"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="@+id/btnPrevious"  
 app:layout\_constraintTop\_toTopOf="@+id/btnAdd"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
<Button  
 android:id="@+id/btnAdd"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="15dp"  
 android:text="Add"  
 android:textSize="15sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="@+id/btnNext" />

Above is the code of activity\_main.xml to display the "Image link" field and the "Add" button is used to get the image link and add that image to the gallery.

protected ArrayList<String> getImageList() {  
 ArrayList<String> imageList = new ArrayList<>();  
 imageList.add("https://st.depositphotos.com/1001911/2358/v/950/depositphotos\_23582275-stock-illustration-boy-scout.jpg");  
 https://cdn3.vectorstock.com/i/1000x1000/28/87/cartoon-happy-little-boy-scout-vector-31942887.jpg  
 getImageListFromFile(imageList);  
  
 Toast.*makeText*(this, "Get list successfully.", Toast.*LENGTH\_SHORT*).show();  
  
 return imageList;  
}  
  
protected void loadImage() {  
 Picasso.*with*(this).load(\_imageList.get(\_currentIndex)).into(imageView);  
}

The device will get the image address through the link and return it to the device.

protected void addImage() {  
 String imageURL = etImageLink.getText().toString();  
  
 \_imageList.add(imageURL);  
 writeURLToFile(imageURL);  
  
 etImageLink.setText("");  
  
 Toast.*makeText*(this, "Added successfully.", Toast.*LENGTH\_SHORT*).show();  
}

The device after receiving the image, addImage will add the image to the list and display it on the app. Add successfully will display the words "Added successfully."

protected void writeURLToFile(String url) {  
 try {  
 OutputStreamWriter outputStreamWriter = new OutputStreamWriter(openFileOutput(*\_FILE\_NAME*, *MODE\_APPEND*));  
 outputStreamWriter.write(url);  
 outputStreamWriter.close();  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 Toast.*makeText*(this, "File not found.", Toast.*LENGTH\_SHORT*).show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
}  
  
private void getImageListFromFile(ArrayList<String> imageList) {  
 try {  
 InputStream inputStream = openFileInput(*\_FILE\_NAME*);  
  
 if (inputStream != null) {  
 InputStreamReader inputStreamReader = new InputStreamReader(inputStream);  
 BufferedReader bufferedReader = new BufferedReader(inputStreamReader);  
  
 String url = "";  
 while ((url = bufferedReader.readLine()) != null) {  
 imageList.add(url);  
 }  
  
 inputStream.close();  
 }  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 Toast.*makeText*(this, "File not found.", Toast.*LENGTH\_SHORT*).show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
}

This code has the function to get the URL and convert it to an image that is displayed on the screen after it has been added to the list.

1. **Previous function**

**2.1 Basic information**

|  |  |
| --- | --- |
| Student name | **PHUNG TAN SANG** |
| Who did you work with? Note that for  logbook exercises you are allowed to  work with one other person as long as  you give their name and login id and  both contribute to the work. | **Name: Phung Tan Sang**  **Login ID: ps5179c** |
| Which Exercise is this? Tick as  appropriate. | Create "Previous" button to view previous image. |
| How well did you complete the exercise? Tick as appropriate. | 🞏 I tried but couldn't complete it.  🗹 I did it but I feel I should have done better.  🞏 I did everything that was asked.  🞏 I did more than was asked for. |
| Briefly explain your answer to question | I created a Previous button to give users the flexibility to view previous photos right at the main interface without having to go to the gallery. |

**2.2 Screen shots**

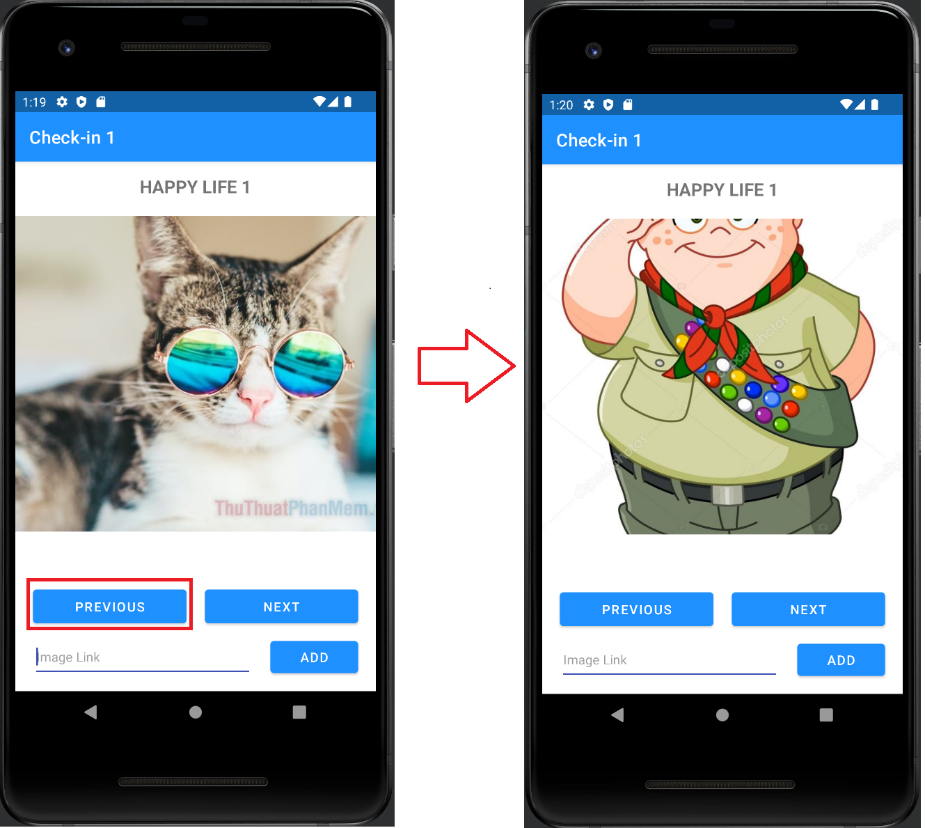


Figure - Previous function

Below the image, there are 2 buttons "Previous" and "Next". If the user wants to see the previous image, he needs to click on the "Previous" button . The image container will change to the previous image.

* 1. **Code**

<Button  
 android:id="@+id/btnPrevious"  
 android:layout\_width="0dp"  
 android:layout\_height="50dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="60dp"  
 android:layout\_marginEnd="10dp"  
 android:text="Previous"  
 android:textSize="15sp"  
 app:layout\_constraintEnd\_toStartOf="@+id/screenDividerHorizontal"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/imageView" />

The Previous button is formatted and displayed on the screen through the above code.

protected void previousImage() {  
 --\_currentIndex;  
 loadImage();  
}

As the image is changed above (Figure 2), it can be seen that the image has been changed after clicking the Previous button. This code helps the image displayed on the app at the moment to return to the previous image --\_currentIndex

# **3 Next function**

**3.1 Basic information**

|  |  |
| --- | --- |
| Student name | **PHUNG TAN SANG** |
| Who did you work with? Note that for  logbook exercises you are allowed to  work with one other person as long as  you give their name and login id and  both contribute to the work. | **Name: Phung Tan Sang**  **Login ID: ps5179c** |
| Which Exercise is this? Tick as  appropriate. | Create "Next" button to view the image later. |
| How well did you complete the  exercise? Tick as appropriate. | 🞏 I tried but couldn't complete it.  🗹 I did it but I feel I should have done better.  🞏 I did everything that was asked.  🞏 I did more than was asked for. |
| Briefly explain your answer to question | I created a Next button to give users the flexibility to view later photos right at the main interface without having to go to the gallery. |

* 1. **Screen shots**

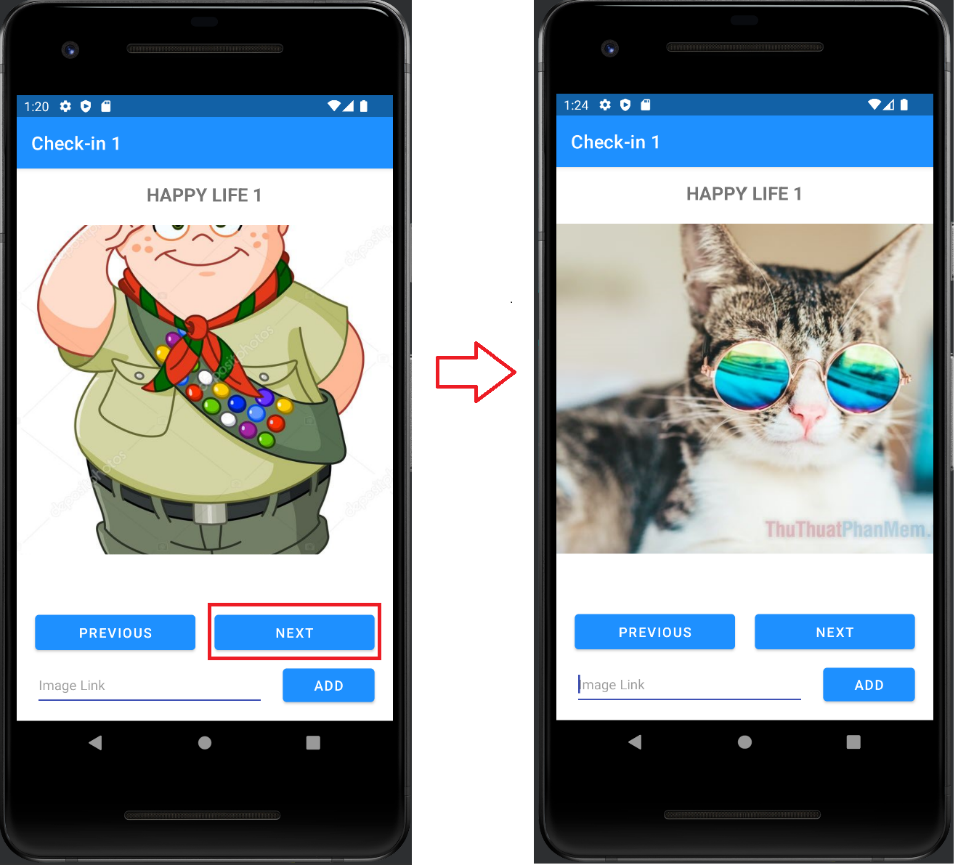


Figure - Next function

Below the image, there are 2 buttons "Previous" and "Next". If the user wants to see the image immediately after, they need to click on the "Next" button . The image container will change to an image afterwards.

* 1. **Code**

<Button  
 android:id="@+id/btnNext"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="20dp"  
 android:text="Next"  
 android:textSize="15sp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/btnPrevious"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/screenDividerHorizontal"  
 app:layout\_constraintTop\_toTopOf="@+id/btnPrevious" />

The Next button is formatted and displayed on the screen through the above code.

protected void nextImage() {  
 ++\_currentIndex;  
 loadImage();  
}

As the image is changed above (Figure 3), it can be seen that the image has been changed after clicking the Next button.

This code helps the image to display on the app at the time it will switch to the image then ++\_currentIndex

1. **Take photo by camera**

**4.1 Basic information**

|  |  |
| --- | --- |
| Student name | **PHUNG TAN SANG** |
| Who did you work with? Note that for  logbook exercises you are allowed to  work with one other person as long as  you give their name and login id and  both contribute to the work. | **Name: Phung Tan Sang**  **Login ID: ps5179c** |
| Which Exercise is this? Tick as  appropriate. | Create a "Camera" button that uses the mobile camera to take pictures. |
| How well did you complete the  exercise? Tick as appropriate. | 🞏 I tried but couldn't complete it.  🞏 I did it but I feel I should have done better.  🗹 I did everything that was asked.  🞏 I did more than was asked for. |
| Briefly explain your answer to question | The "Camera" button makes it possible for users to access the camera on their own mobile device to take the pictures they want to take and keep. |

## **4.2** **Screenshots**

The user wants to access the camera on the mobile device to take pictures to save the surrounding scenery at the current time. So, clicking "Take picture" button  will have 3 options Files, Photos, Camera:

- Files: helps users access libraries on Google Drive or Google Photos accounts.

- Photos: to access the photo library on the device.

- Camera: access the device's camera to take live photos.

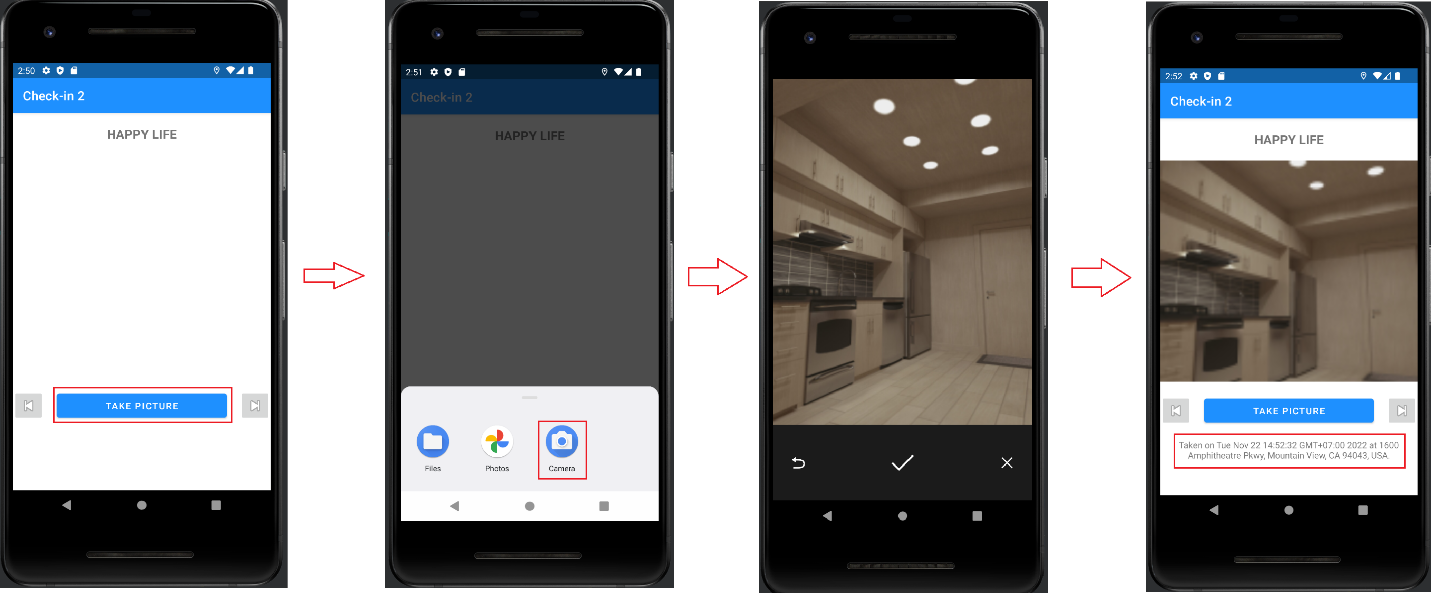


Figure – Camera

Click the "Camera" button to directly capture any image the user wants.

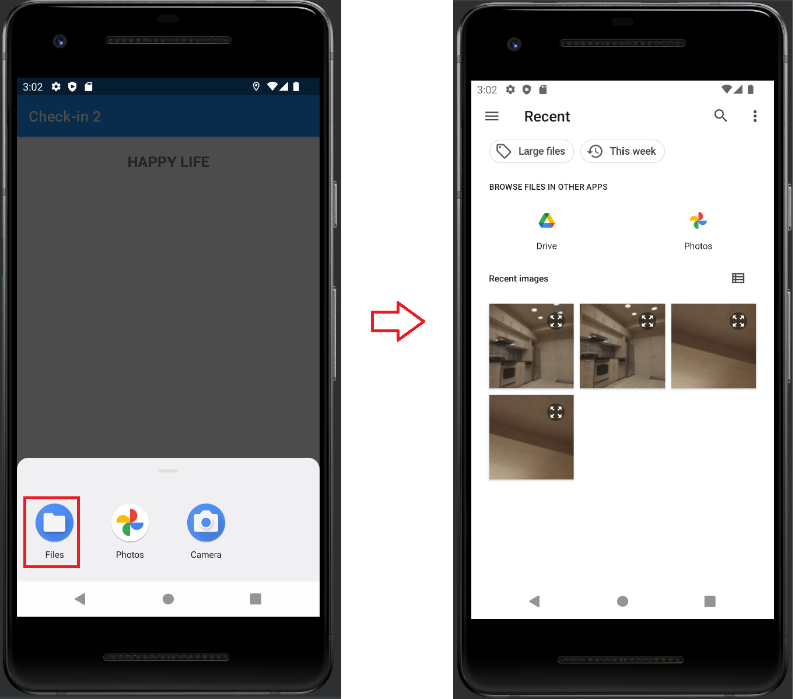


Figure - Files option

Files will contain images that have been saved on Google Drive and Google Photos accounts. Users can get images from the two platforms above as long as there is image data.

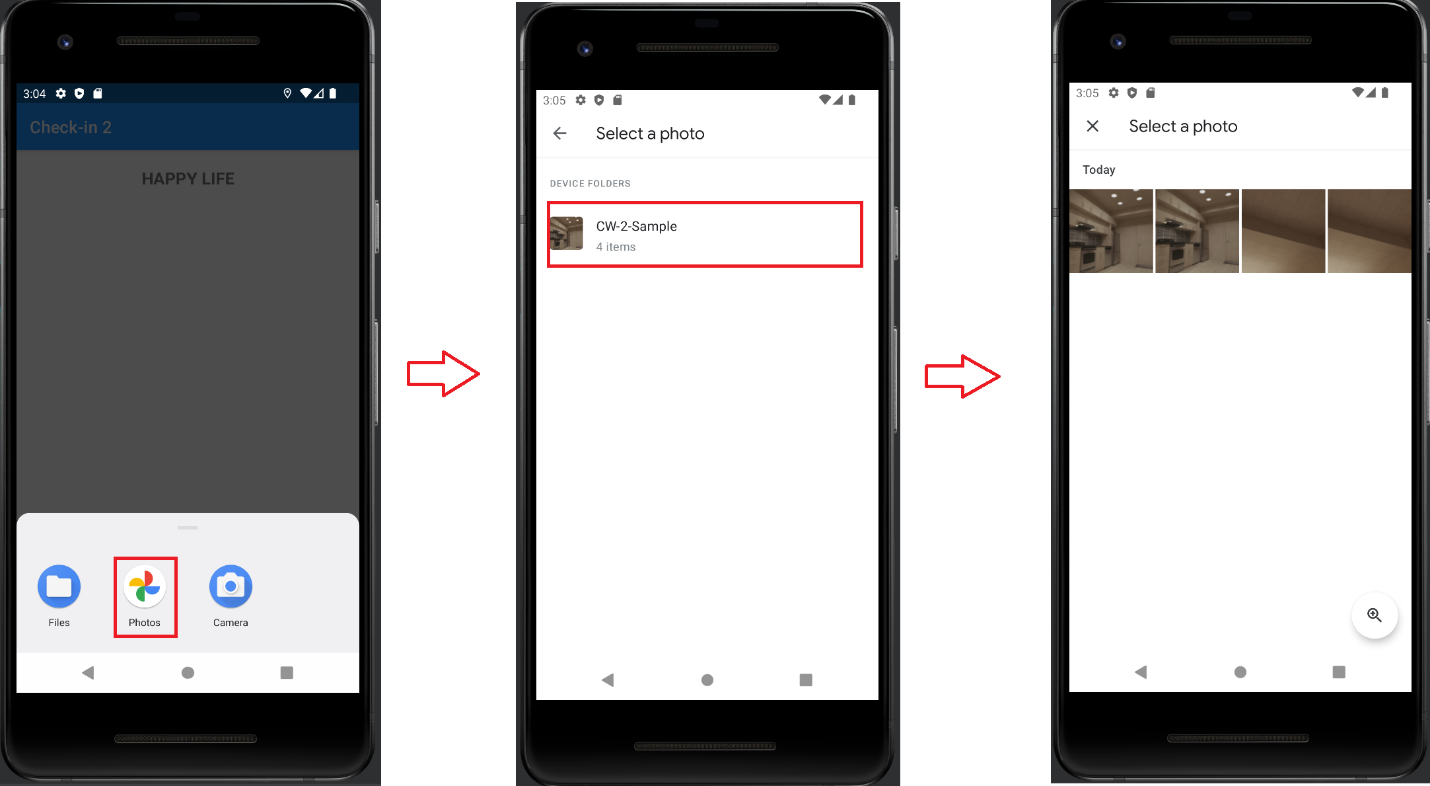


Figure - Photos option

Photos is your mobile device's photo library. They may or may not be saved to any platform. Users can also select images in this gallery. But note, the images in the gallery are only on the user's device and not in any cloud platform.

Photos uploaded or taken will include details of the date, time and place where it was taken.

## **4.3 Code**

<Button  
 android:id="@+id/btnTakePicture"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:text="Take Picture"  
 android:textSize="15sp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/btnPrevious"  
 app:layout\_constraintEnd\_toStartOf="@+id/btnNext"  
 app:layout\_constraintStart\_toEndOf="@+id/btnPrevious"  
 app:layout\_constraintTop\_toTopOf="@+id/btnPrevious" />

The "Take Camera" button is formatted to be displayed on the app through the above code.

protected boolean allPermissionsGranted\_CAMERA() {  
 for (String permission : *REQUIRED\_PERMISSIONS\_CAMERA*)  
 if (ContextCompat.*checkSelfPermission*(this, permission) != PackageManager.*PERMISSION\_GRANTED*)  
 return false;  
  
 return true;  
}  
  
private boolean allPermissionsGranted\_GPS() {  
 for (String permission : REQUIRED\_PERMISSIONS\_GPS)  
 if (ContextCompat.*checkSelfPermission*(this, permission) != PackageManager.*PERMISSION\_GRANTED*)  
 return false;  
  
 return true;  
}

Two pieces of code used to decentralize and authorize Camera and GPS.

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
 if (requestCode == *REQUEST\_CODE\_PERMISSIONS\_CAMERA*) {  
 if (!allPermissionsGranted\_CAMERA()) {  
 Toast.*makeText*(this, "Camera Permissions not granted by the user.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 }  
  
 if (requestCode == REQUEST\_CODE\_PERMISSIONS\_GPS) {  
 if (allPermissionsGranted\_GPS()) {  
 startGPS();  
 return;  
 }  
  
 Toast.*makeText*(this, "GPS Permissions not granted by the user.", Toast.*LENGTH\_SHORT*).show();  
 }  
}

Used to get authorization results. If the user does not allow Camera and GPS access, the app cannot use the camera function on the device.

protected void startGPS() {  
 // Check permissions.  
 if (!allPermissionsGranted\_GPS()) {  
 ActivityCompat.*requestPermissions*(this, REQUIRED\_PERMISSIONS\_GPS, REQUEST\_CODE\_PERMISSIONS\_GPS);  
 return;  
 }  
  
 locationListener = location -> {  
 currentLocation = location;  
 Toast.*makeText*(MainActivity.this, "Get current location successfully.", Toast.*LENGTH\_SHORT*).show();  
 };  
  
 locationManager = (LocationManager) getSystemService(Context.*LOCATION\_SERVICE*);  
 locationManager.requestLocationUpdates(LocationManager.*GPS\_PROVIDER*, LOCATION\_REFRESH\_TIME, LOCATION\_REFRESH\_DISTANCE, locationListener);  
}

The App will update the device's location when the user allows GPS access.

protected void loadImage() {  
 if (\_imageList == null || \_imageList.size() == 0) {  
 Toast.*makeText*(this, "No Image !!!", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 Uri imageUri = \_imageList.get(\_currentIndex);  
  
 imageView.setImageURI(imageUri);  
  
 try {  
 if (android.os.Build.VERSION.*SDK\_INT* >= android.os.Build.VERSION\_CODES.*Q*) {  
 InputStream inputStream = getContentResolver().openInputStream(imageUri);  
 ExifInterface exifInterface = new ExifInterface(inputStream);  
  
 String tag\_address = "Not Found";  
  
 String tag\_date = exifInterface.getAttribute(ExifInterface.*TAG\_DATETIME\_ORIGINAL*);  
 String tag\_latitude = exifInterface.getAttribute(ExifInterface.*TAG\_GPS\_DEST\_LATITUDE\_REF*);  
 String tag\_longitude = exifInterface.getAttribute(ExifInterface.*TAG\_GPS\_DEST\_LONGITUDE\_REF*);  
  
 if (tag\_longitude != null && !tag\_longitude.isEmpty() &&  
 tag\_latitude != null && !tag\_latitude.isEmpty()) {  
 double latitude = Double.*parseDouble*(tag\_latitude);  
 double longitude = Double.*parseDouble*(tag\_longitude);  
  
 Geocoder gcd = new Geocoder(getBaseContext(), Locale.*getDefault*());  
 List<Address> addresses = gcd.getFromLocation(latitude, longitude, 1);  
  
 if (addresses.size() > 0) {  
 tag\_address = addresses.get(0).getAddressLine(0);  
 }  
 }  
  
 String description = "Taken on " + tag\_date + " at " + tag\_address + ".";  
 tvDescription.setText(description);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
}

If no images are loaded yet, "No image" will be displayed.

If the address cannot be obtained, it will display "Not found"

protected void takePicture() {  
 // Ask for camera permissions.  
 if (!allPermissionsGranted\_CAMERA()) {  
 ActivityCompat.*requestPermissions*(this, *REQUIRED\_PERMISSIONS\_CAMERA*, *REQUEST\_CODE\_PERMISSIONS\_CAMERA*);  
 return;  
 }  
  
 Intent imageCaptureIntent = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*);  
  
 Intent fileIntent = new Intent(Intent.*ACTION\_GET\_CONTENT*);  
 fileIntent.setType("image/\*");  
  
 Intent galleryIntent = new Intent(Intent.*ACTION\_PICK*, MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);  
 galleryIntent.setType("image/\*");  
  
 Intent chooserIntent = Intent.*createChooser*(imageCaptureIntent, "Select Image");  
 chooserIntent.putExtra(Intent.*EXTRA\_INITIAL\_INTENTS*, new Intent[]{fileIntent, galleryIntent});  
  
 startActivityForResult(chooserIntent, *REQUEST\_CODE\_CAMERA*);  
}

When the user has granted permission to access the Camera. App can access camera to take pictures. There are 2 places used to store images as well as get images on the app: Files and Photos.