I am assuming that most of you were able to follow the following tutorial (suggested in the Sprint I backlog) that allowed your app to take a picture using camera Intent and save it in a folder on the external storage through the file provider besides displaying it in an ImageView.

https://developer.android.com/training/camera/photobasics#java

To achieve completion of Sprint I, we will do three key enhancements to this app:

- First, we will add the ability to scroll through all the photos that have been taken and stored in the external storage using left and right buttons and also the date the photo was taken.
- Second, we will add the ability to edit caption.
- Lastly, we will add another activity named SearchActivity that would help us find photos taken during the specified time window and/or caption that matches the specified keyword.

Lets make the first enhancement, but before we do that, lets look at where we are at this point and what the manifest file, layout file and the MainActivity code look like.

The manifest file describes the current composition of the app. It declares that the app is composed of a MainActivity class and a FileProvider. The Intent filter specified for the MainActivity makes it the launcher activity. In other words, we are telling the system that when the app is invoked, the MainActivity should be the one launched first even though the app may have additional activities.

The manifest file also declares that the application uses the camera feature and would request user to grant permissions to read and write the external storage for storage management of photos.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="com.example.myapplication">
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" android:maxSdkVersion="18" />
  <uses-feature android:name="android.hardware.camera" android:required="true" />
  <application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <provider
      android:name="androidx.core.content.FileProvider"
      android:authorities="com.example.myapplication.fileprovider"
      android:exported="false"
      android:grantUriPermissions="true">
      <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths" />
    </provider>
  </application>
</manifest>
```

The provider in the above manifest file identifies the file_paths file located in the xml resource folder. This xml folder should be created by right clicking on the resource folder of the Android Studio project and adding a new directory, naming it xml and then adding an xml file named file_paths in it. The content of this xml file should be as follows.

```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
<external-path name="my_images"
path="Android/data/com.example.myapplication/files/Pictures" />
</paths>
```

The activity_main.xml file inside the layout folder of the resource folder specifies a simple GUI that has one ImageView (to display the taken photo) and a button labeled SNAP to take photo using the onboard camera app.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/constraintLayoutLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="356dp"
        android:layout_height="265dp"
        app:srcCompat="@drawable/ic_launcher_foreground"
        tools:layout_editor_absoluteX="23dp"
        tools:layout_editor_absoluteY="39dp" />
    <Button
        android:id="@+id/snap"
        android:layout_width="wrap_content"
        android:layout_height="60dp"
        android:text="snap"
        android:onClick="takePhoto"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.786"
        tools:layout_editor_absoluteX="0dp" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

The MainActivity, listed below, handles the SNAP button click in its takePhoto() method by creating an empty file in the **Android/data/com.example.myapplication/files/Pictures** folder by calling createImageFile() method and passes the path as a part of an (implicit) intent requesting the system to use the onboard camera app to take the picture and save it in this newly created empty file. After the photo is successfully taken and saved, the return back to the MainActivity from the Camera App is handled in its onActivityResult() method where the file at the specified path is displayed through the ImageView.

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.FileProvider;
import android.content.Intent;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.ImageView;

import java.io.File;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class MainActivity extends AppCompatActivity {
   static final int REQUEST_IMAGE_CAPTURE = 1;
   String mCurrentPhotoPath;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
   public void takePhoto(View v) {
      Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
      if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
         File photoFile = null;
         try {
            photoFile = createImageFile();
         } catch (IOException ex) {
            // Error occurred while creating the File
         }
         // Continue only if the File was successfully created
         if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this, "com.example.myapplication.fileprovider", photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
         }
      }
   }
   private File createImageFile() throws IOException {
      // Create an image file name
      String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
      String imageFileName = "JPEG_" + timeStamp + "_";
      File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
      File image = File.createTempFile(imageFileName, ".jpg",storageDir);
      mCurrentPhotoPath = image.getAbsolutePath();
      return image;
   }
   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
         ImageView mImageView = (ImageView) findViewById(R.id.imageView);
         mImageView.setImageBitmap(BitmapFactory.decodeFile(mCurrentPhotoPath));
      }
```

```
    }
}
```

You can view all the pictures on the emulators file system by going to

view -> tools window -> device file viewer

and then expanding sdcard ->android->data->com.example.myapplication->files->pictures

It should be obvious that com.example.myapplication is the package name of this reference app which may differ if you chose a different name for your app.

Add the ability to scroll through all the photos that have been taken and stored in the external storage using left and right buttons and also display the date the photo was taken.

The manifest file will not change.

In the layout file add two additional buttons, a TextView and an EditView. The revised layout file should look like as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/constraintLayoutLayout"
    android:layout_width="match_parent"  android:layout_height="match_parent"
    tools:context=".MainActivity">


    <ImageView
        android:id="@+id/ivGallery"
        android:layout_width="356dp" android:layout_height="265dp"
        android:layout_marginStart="23dp" android:layout_marginTop="39dp"
        android:layout_marginEnd="32dp" android:layout_marginBottom="39dp"
        app:layout_constraintBottom_toTopOf="@+id/location"
        app:layout_constraintEnd_toEndOf="parent" app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" app:srcCompat="@drawable/ic_launcher_foreground" />

    <Button
        android:id="@+id/snap"
        android:layout_width="85dp"  android:layout_height="60dp"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="3dp" android:layout_marginEnd="194dp" android:layout_marginBottom="178dp"
        android:onClick="takePhoto"  android:text="snap"
        app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toStartOf="@+id/btnNext"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.786" />

    <Button
        android:id="@+id/btnNext"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
```

```
    android:layout_alignBaseline="@+id/snap" android:layout_marginStart="194dp"
    android:text="next" android:onClick="scrollPhotos"
    app:layout_constraintBottom_toTopOf="@+id/search" app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/snap" app:layout_constraintTop_toBottomOf="@+id/timestamp" />

  <Button
    android:id="@+id/btnPrev"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_alignParentEnd="true" android:layout_alignParentBottom="true"
    android:layout_marginEnd="82dp" android:layout_marginBottom="15dp"
    android:text="prev" android:onClick="scrollPhotos"
    tools:layout_editor_absoluteX="0dp" tools:layout_editor_absoluteY="622dp" />


  <TextView
    android:id="@+id/tvTimestamp"
    android:layout_width="340dp" android:layout_height="42dp"
    android:layout_marginStart="16dp" android:layout_marginTop="345dp"
    android:layout_marginEnd="55dp"
    android:text=""
    app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.255" />

  <EditText
    android:id="@+id/etCaption"
    android:layout_width="372dp" android:layout_height="wrap_content"
    android:layout_alignParentStart="true" android:layout_alignParentBottom="true"
    android:layout_marginBottom="272dp" android:ems="10"
    android:inputType="textPersonName"  android:text=""
    app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintTop_toTopOf="parent"
    tools:layout_editor_absoluteX="0dp" />

</RelativeLayout>
```

In the MainActivity, the following instance variables are added:

```
  private ArrayList<String> photos = null;
  private int index = 0;
```

and getPhotos(), scrollPhotos() and displayPhoto() methods are added. Also note the call to getPhotos() towards the end of onActivityResult() method and calls to getPhotos() as well as displayPhoto() in the onCreate() method.

Note that the name given to each photo file (in the createImageFile() method) not only has the timestamp, as before, but now also has a placeholder for the caption that we will utilize for the next advancement. It may therefore be better to remove all the photos from the folder and start fresh to avoid discrepancy with the previous version of the app.

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity; import androidx.core.content.FileProvider;
import android.content.Intent; import android.graphics.BitmapFactory;
import android.net.Uri; import android.os.Bundle; import android.os.Environment;
import android.provider.MediaStore; import android.view.View; import android.widget.EditText;
```

```
import android.widget.ImageView; import android.widget.TextView; import java.io.File;
import java.io.IOException; import java.text.SimpleDateFormat; import java.util.ArrayList;
import java.util.Date;

public class MainActivity extends AppCompatActivity {
   static final int REQUEST_IMAGE_CAPTURE = 1;
   String mCurrentPhotoPath;
   private ArrayList<String> photos = null;
   private int index = 0;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      photos = findPhotos();
      if (photos.size() == 0) {
         displayPhoto(null);
      } else {
         displayPhoto(photos.get(index));
      }
   }
   public void takePhoto(View v) {
      Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
      if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
         File photoFile = null;
         try {
            photoFile = createImageFile();
         } catch (IOException ex) {
            // Error occurred while creating the File
         }
         // Continue only if the File was successfully created
         if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this, "com.example.myapplication.fileprovider", photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
         }
      }
   }
   private ArrayList<String> findPhotos() {
      File file = new File(Environment.getExternalStorageDirectory()
            .getAbsolutePath(), "/Android/data/com.example.myapplication/files/Pictures");
      ArrayList<String> photos = new ArrayList<String>();
      File[] fList = file.listFiles();
      if (fList != null) {
         for (File f : fList) {
            photos.add(f.getPath());
         }
      }
      return photos;
   }
   public void scrollPhotos(View v) {
      switch (v.getId()) {
         case R.id.btnPrev:
            if (index > 0) {
               index--;
            }
            break;
         case R.id.btnNext:
```

```
            if (index < (photos.size() - 1)) {
               index++;
            }
            break;
         default:
            break;
      }
      displayPhoto(photos.get(index));
   }
   private void displayPhoto(String path) {
      ImageView iv = (ImageView) findViewById(R.id.ivGallery);
      TextView tv = (TextView) findViewById(R.id.tvTimestamp);
      EditText et = (EditText) findViewById(R.id.etCaption);
      if (path == null || path =="") {
         iv.setImageResource(R.mipmap.ic_launcher);
         et.setText("");
         tv.setText("");
      } else {
         iv.setImageBitmap(BitmapFactory.decodeFile(path));
         String[] attr = path.split("_");
         et.setText(attr[1]);
         tv.setText(attr[2]);
      }
   }
   private File createImageFile() throws IOException {
      // Create an image file name
      String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
      String imageFileName = "_caption_" + timeStamp + "_";
      File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
      File image = File.createTempFile(imageFileName, ".jpg",storageDir);
      mCurrentPhotoPath = image.getAbsolutePath();
      return image;
   }
   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent data) {
      if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
         ImageView mImageView = (ImageView) findViewById(R.id.ivGallery);
         mImageView.setImageBitmap(BitmapFactory.decodeFile(mCurrentPhotoPath));
         photos = findPhotos();
      }
   }
}
```

Add the ability to add/edit and display caption for each image.

An EditText was already added to the layout file to support this functionality.

Add the following method in the MainActivity:

```
private void updatePhoto(String path, String caption) {
   String[] attr = path.split("_");
   if (attr.length >= 3) {
      File to = new File(attr[0] + "_" + caption + "_" + attr[2] + "_" + attr[3]);
      File from = new File(path);
      from.renameTo(to);
```

```
    }
}
```

add the following line of code as the first line in the scrollPhotos() method of the MainActivity:

```
updatePhoto(photos.get(index), ((EditText) findViewById(R.id.etCaption)).getText().toString());
```

Add another activity named SearchActivity that would help find photos taken during the specified time window and/or caption that matches the specified keyword.

Add a button to the activity_main.xml layout file:

```
<Button
   android:id="@+id/btnSearch"
   android:layout_width="wrap_content" android:layout_height="wrap_content"
   android:layout_alignStart="@+id/ivGallery" android:layout_alignParentBottom="true"
   android:layout_marginBottom="18dp" android:text="search"
   tools:layout_editor_absoluteX="291dp" tools:layout_editor_absoluteY="635dp" />
```

Revise findPhotos() method as follows:

```
private ArrayList<String> findPhotos(Date startTimestamp, Date endTimestamp, String keywords) {
   File file = new File(Environment.getExternalStorageDirectory()
         .getAbsolutePath(), "/Android/data/com.example.myapplication/files/Pictures");
   ArrayList<String> photos = new ArrayList<String>();
   File[] fList = file.listFiles();
   if (fList != null) {
      for (File f : fList) {
         if (((startTimestamp == null && endTimestamp == null) || (f.lastModified() >= startTimestamp.getTime()
               && f.lastModified() <= endTimestamp.getTime())
         ) && (keywords == "" || f.getPath().contains(keywords)))
            photos.add(f.getPath());
      }
   }
   return photos;
}
```

Revise onActivityResult() as follows:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
   super.onActivityResult(requestCode, resultCode, data);
   if (requestCode == SEARCH_ACTIVITY_REQUEST_CODE) {
      if (resultCode == RESULT_OK) {
         DateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
         Date startTimestamp , endTimestamp;
         try {
```

```
            String from = (String) data.getStringExtra("STARTTIMESTAMP");
            String to = (String) data.getStringExtra("ENDTIMESTAMP");
            startTimestamp = format.parse(from);
            endTimestamp = format.parse(to);

        } catch (Exception ex) {
            startTimestamp = null;
            endTimestamp = null;
        }
        String keywords = (String) data.getStringExtra("KEYWORDS");
        index = 0;
        photos = findPhotos(startTimestamp, endTimestamp, keywords);

        if (photos.size() == 0) {
            displayPhoto(null);
        } else {
            displayPhoto(photos.get(index));
        }
      }
    }
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
      ImageView mImageView = (ImageView) findViewById(R.id.ivGallery);
      mImageView.setImageBitmap(BitmapFactory.decodeFile(mCurrentPhotoPath));
      photos = findPhotos(new Date(Long.MIN_VALUE), new Date(), "");
    }
}
```

Replace call to findPhotos() in the onCreate() method with the following:

```
photos = findPhotos(new Date(Long.MIN_VALUE), new Date(), "");
```

The activity_search.xml and SearchActivity are listed below:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SearchActivity">
    <TextView
        android:id="@+id/tvFromDateTime"  android:text="From: "
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/etFromDateTime" android:layout_alignStart="@+id/btnCancel"
        android:textSize="24sp" />
    <EditText
        android:id="@+id/etFromDateTime" android:inputType="date"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_alignParentEnd="true" android:layout_alignParentTop="true"
        android:layout_marginTop="38dp" android:ems="10" android:textSize="24sp" />
    <TextView
        android:id="@+id/tvToDateTime"  android:text="To: "
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_alignParentTop="true" android:layout_alignStart="@+id/tvFromDateTime"
        android:layout_marginTop="98dp" android:textSize="24sp" />
    <EditText
        android:id="@+id/etToDateTime" android:inputType="date"
```

```
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/tvToDateTime" android:layout_alignParentEnd="true"
    android:layout_marginEnd="31dp" android:ems="10"
    android:textSize="24sp" />
  <TextView
    android:id="@+id/tvKeywords" android:text="Keywords: "
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_alignStart="@+id/tvFromDateTime" android:layout_centerVertical="true"
    android:textSize="24sp" />

  <EditText
    android:id="@+id/etKeywords" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:layout_alignStart="@+id/go"
    android:layout_centerVertical="true" android:ems="10"
    android:textSize="24sp" />
  <Button
    android:id="@+id/btnCancel" android:text="Cancel" android:onClick="cancel"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_alignParentBottom="true" android:layout_marginRight="55dp"
    android:layout_marginEnd="55dp" />
  <Button
    android:id="@+id/go"  android:onClick="go" android:text="Go"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:layout_alignParentEnd="true" android:layout_alignTop="@+id/btnCancel"
    android:layout_marginEnd="128dp" />
</RelativeLayout>

package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent; import android.os.Bundle;
import android.view.View; import android.widget.EditText;
import java.text.DateFormat; import java.text.SimpleDateFormat;
import java.util.Calendar; import java.util.Date;
import java.util.Locale;
public class SearchActivity extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_search);
    try {
      Calendar calendar = Calendar.getInstance();
      DateFormat format = new SimpleDateFormat("yyyy-MM-dd");
      Date now = calendar.getTime();
      String todayStr = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault()).format(now);
      Date today = format.parse((String) todayStr);
      calendar.add(Calendar.DAY_OF_YEAR, 1);
      String tomorrowStr = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault()).format( calendar.getTime());
      Date tomorrow = format.parse((String) tomorrowStr);
      ((EditText) findViewById(R.id.etFromDateTime)).setText(new SimpleDateFormat(
          "yyyy-MM-dd HH:mm:ss", Locale.getDefault()).format(today));
      ((EditText) findViewById(R.id.etToDateTime)).setText(new SimpleDateFormat(
          "yyyy-MM-dd HH:mm:ss", Locale.getDefault()).format(tomorrow));

    } catch (Exception ex) { }
  }
  public void cancel(final View v) {
    finish();
  }
```

```
    public void go(final View v) {
        Intent i = new Intent();
        EditText from = (EditText) findViewById(R.id.etFromDateTime);
        EditText to = (EditText) findViewById(R.id.etToDateTime);
        EditText keywords = (EditText) findViewById(R.id.etKeywords);
        i.putExtra("STARTTIMESTAMP", from.getText() != null ? from.getText().toString() : "");
        i.putExtra("ENDTIMESTAMP", to.getText() != null ? to.getText().toString() : "");
        i.putExtra("KEYWORDS", keywords.getText() != null ? keywords.getText().toString() : "");
        setResult(RESULT_OK, i);
        finish();
    }
}
```

For UI Tests specify dates and keyword and check if the photos searched matched the expected results.

```
onView(withId(R.id.btnSearch)).perform(click());
      onView(withId(R.id.etFromDateTime)).perform(typeText(""), closeSoftKeyboard());
      onView(withId(R.id.etToDateTime)).perform(typeText(""), closeSoftKeyboard());
      onView(withId(R.id.etKeywords)).perform(typeText("caption"), closeSoftKeyboard());
      onView(withId(R.id.go)).perform(click());
      onView(withId(R.id.etCaption)).check(matches(withText("caption")));
      onView(withId(R.id.btnRight)).perform(click());
      onView(withId(R.id.btnLeft)).perform(click());
```