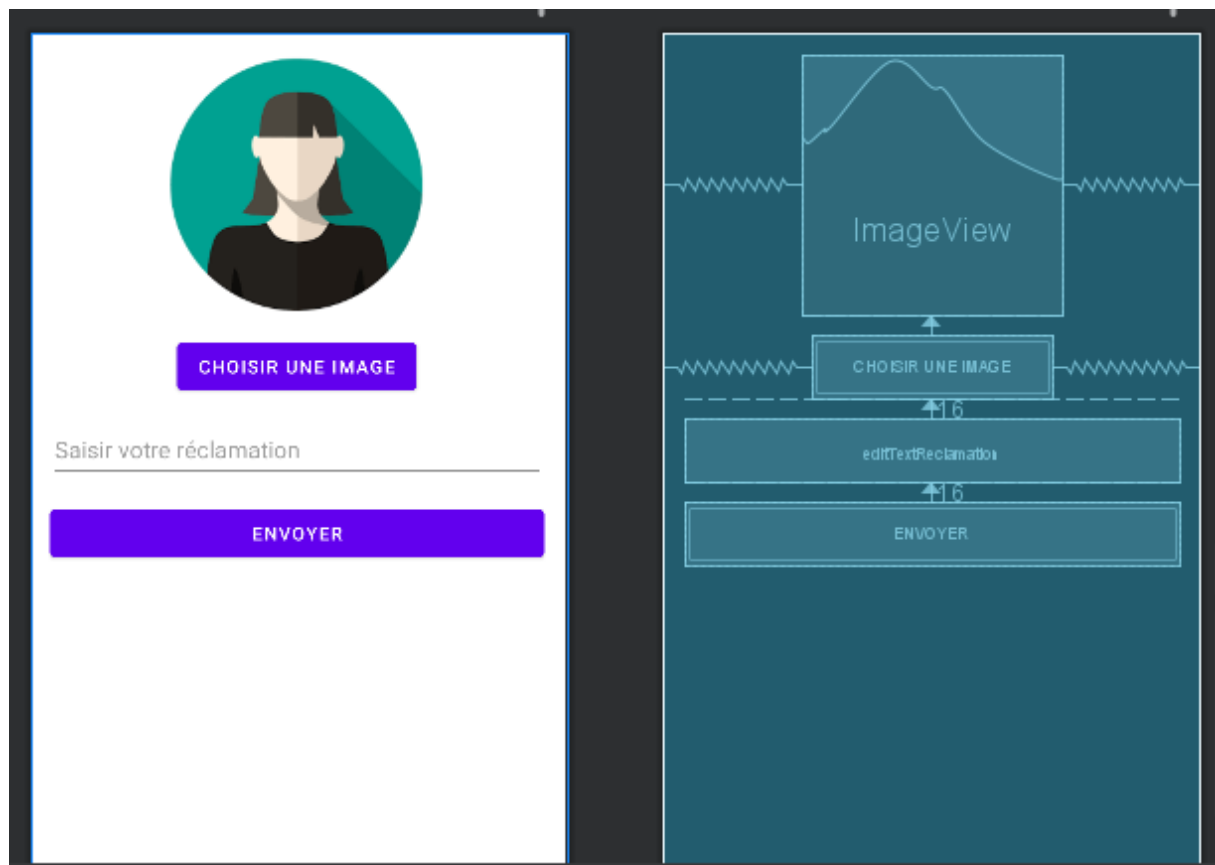

Atelier 3 : Création de l'activité de Réclamation

Projet : Gestion de syndicat de copropriété

Objectif : Accès à la galerie/Caméra
 Comprendre l'Intent
 Stocker les données dans Firestore



Etape 1 : Dans le dossier Drawable, ajouter l'image par défaut de votre interface

Etape 2 : Créer un nouveau Layout `new_claim.xml` et insérer le code suivant

Soit le code XML associé au Layout ci-dessus :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".reclamation">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:src="@drawable/avatar"
        android:layout_marginBottom="16dp" />

    <Button
        android:id="@+id/btnChooseImage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/imageView"
        android:layout_centerHorizontal="true"
        android:text="Choisir une image" />

    <EditText
        android:id="@+id/editTextReclamation"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/btnChooseImage"
        android:layout_marginTop="16dp"
        android:hint="Saisir votre réclamation"
        android:maxLines="5"
        android:minHeight="48dp" />

    <Button
        android:id="@+id/btnSend"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextReclamation"
        android:layout_marginTop="16dp"
        android:text="Envoyer" />

</RelativeLayout>
```

Etape 3 : Créer un nouveau Layout display_image.xml et insérer le code suivant

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

```
<ImageView  
    android:layout_centerInParent="true"  
    android:id="@+id/imageview"  
    android:layout_width="200dp"  
    android:layout_height="200dp" />  
</RelativeLayout>
```

>

Etape 4 : Assurez-vous d'ajouter les permissions nécessaires dans votre fichier AndroidManifest.xml pour accéder à la caméra et à la galerie.

```
<uses-permissions android:name="android.permission.CAMERA"/>  
<uses-permissions android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
<uses-permissions  
    android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Assurez-vous d'ajouter ces lignes justes avant la balise <application> dans votre fichier AndroidManifest.xml. Ces permissions permettent à votre application d'accéder à la Caméra et à la galerie pour prendre des photos ou choisir des images existantes.

Etape 5 : Assurez-vous d'avoir configuré Firebase Firestore dans votre projet.

Déclarez la dépendance de la bibliothèque Cloud Firestore pour Android dans le fichier Gradle de votre module (au niveau de l'application) (généralement app/build.gradle.kts ou app/build.gradle).

```
implementation("com.google.firebase:firebase-firestore")
```

Etape 6 : Créer les méthodes pour sélectionner Image de la Caméra ou de la Galerie

```
//open camera  
void openCamera(){  
    Intent takePicture = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
    startActivityForResult(takePicture, 0);  
}  
  
//select from Gallery  
void selectImage(){  
    Intent pickPhoto = new Intent(Intent.ACTION_PICK,  
        android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);  
    startActivityForResult(pickPhoto, 1);  
}
```

Etape 7 : Receive selected Image in onActivityResult

```
//get result  
protected void onActivityResult(int requestCode, int resultCode, Intent  
    imageReturnedIntent) {  
    super.onActivityResult(requestCode, resultCode, imageReturnedIntent);  
    switch(requestCode) {  
        case 0:
```

```
if (resultCode == RESULT_OK) {

Uri selectedImage = imageReturnedIntent.getData();
uriString = selectedImage.toString(); //storing uri (content path)
imageView.setImageURI(selectedImage);
}

break;
case 1:
if (resultCode == RESULT_OK) {
Uri selectedImage = imageReturnedIntent.getData();
uriString = selectedImage.toString(); //storing uri (content path)
imageView.setImageURI(selectedImage);
}
break;
}
}
```

Etape 8: Save Uri as a string for passing to the next page display_image

```
Intent i = new Intent(MainActivity.this, Page2Activity.class);
i.putExtra("img", uriString); //passing path to next page
startActivity(i);
```

Etape 9 : Convert String back to Uri path for showing in ImageView

```
Intent i = getIntent();
uriStr = i.getStringExtra("img"); //receiving path as string

Uri uri = Uri.parse(uriStr); //convert string to path
imageView.setImageURI(uri);
```

Code source complet

//Reclamation.java

```
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    Button camera;
    Button selectGallery;
    Button gotoNext;
    ImageView imageView;
    String uriString = "";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //find view by ids section
        gotoNext = findViewById(R.id.gotoNext);
        camera = findViewById(R.id.opencamera);
```

```
selectGallery = findViewById(R.id.selectGallery);
imageView = findViewById(R.id.imageView);

camera.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        openCamera();
    }
});

selectGallery.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        selectImage();
    }
});

gotoNext.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(MainActivity.this, Page2Activity.class);
        i.putExtra("img", uriString); //passing path to next page
        startActivity(i);
    }
});

//open camera
void openCamera(){
    Intent takePicture = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(takePicture, 0);
}

//select from Gallery
void selectImage(){
    Intent pickPhoto = new Intent(Intent.ACTION_PICK,
    android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
    startActivityForResult(pickPhoto , 1);
}

//get result
protected void onActivityResult(int requestCode, int resultCode, Intent
imageReturnedIntent) {
    super.onActivityResult(requestCode, resultCode, imageReturnedIntent);
    switch(requestCode) {
        case 0:
            if(resultCode == RESULT_OK){

                Uri selectedImage = imageReturnedIntent.getData();
                uriString = selectedImage.toString(); //storing uri (content path)
                imageView.setImageURI(selectedImage);
            }

            break;
        case 1:
            if(resultCode == RESULT_OK){
                Uri selectedImage = imageReturnedIntent.getData();
                uriString = selectedImage.toString(); //storing uri (content path)
                imageView.setImageURI(selectedImage);
            }
    }
}
```

```
}  
break;  
}  
}  
}
```

//DisplayImage.java

```
import androidx.appcompat.app.AppCompatActivity;  
  
//import  
public class DisplayImage extends AppCompatActivity {  
    ImageView imageView;  
  
    String uriStr = "";  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.display_image);  
  
        //find view by ids  
        imageView = findViewById(R.id.imageview);  
  
        Intent i = getIntent();  
        uriStr = i.getStringExtra("img"); //receiving path as string  
  
        Uri uri = Uri.parse(uriStr); //convert string to path  
        imageView.setImageURI(uri);  
  
    }  
}
```

Annexes :

Un objet [Intent](#) contient des informations que le système Android utilise pour déterminer quel composant démarrer

Un intent **explicite** vous permet de lancer un composant d'application spécifique, tel qu'une activité ou un service particulier dans votre application.

Un intent **implicite** spécifie une action pouvant appeler n'importe quelle application de l'appareil capable de l'effectuer. L'utilisation d'un intent implicite est utile lorsque votre application ne peut pas effectuer l'action, mais que d'autres applications le peuvent probablement et que vous souhaitez que l'utilisateur choisisse l'application à utiliser.

DR. ANIBOU CHAIMAE
4IIR

DEV MOBILE