

Compte Rendu TP5 :

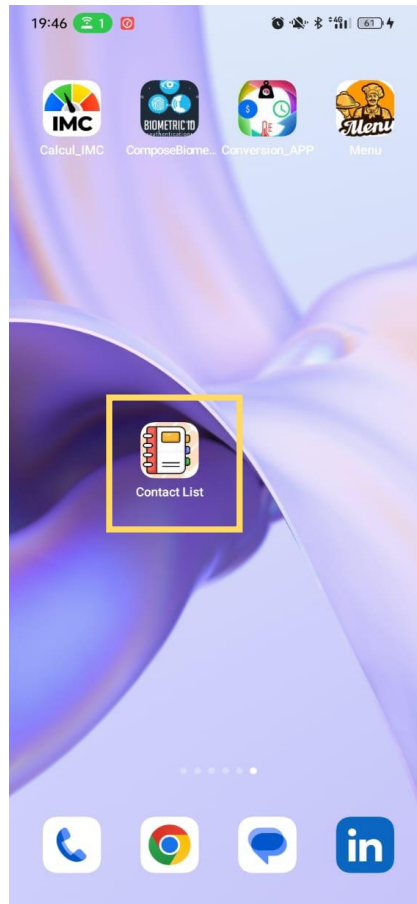
Créer l'Application Contact Intent

Réalisé par :
MALEK SGHAIER

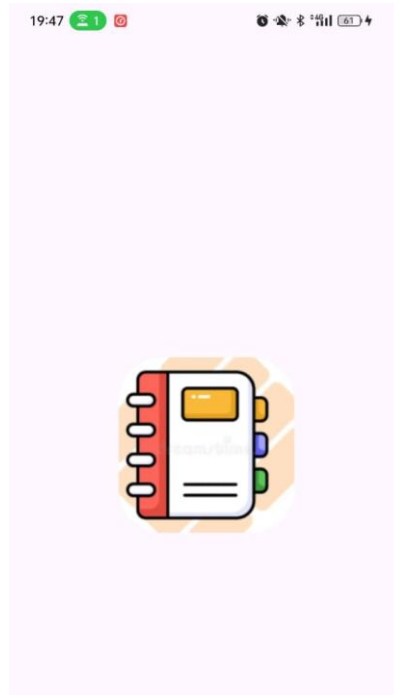
2DNI 1

1. Partie Design :

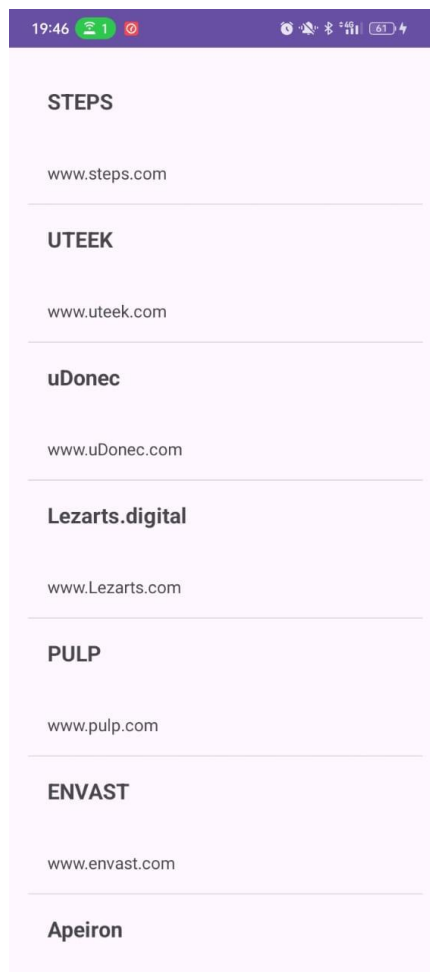
- Icône de l'application :



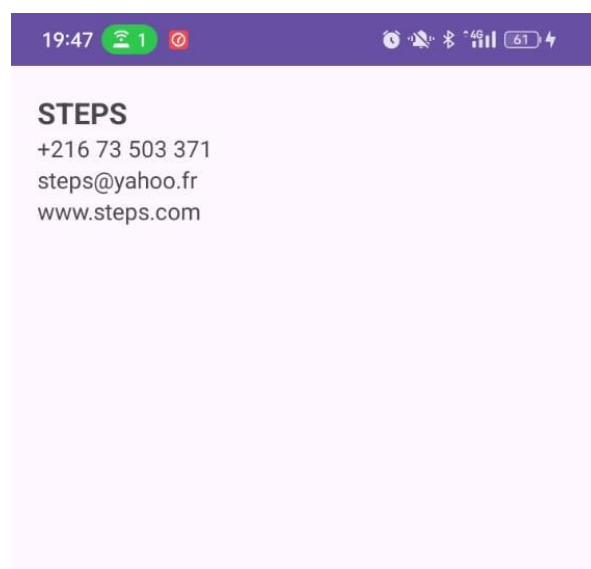
- Splash Screen :



- **Contact :**



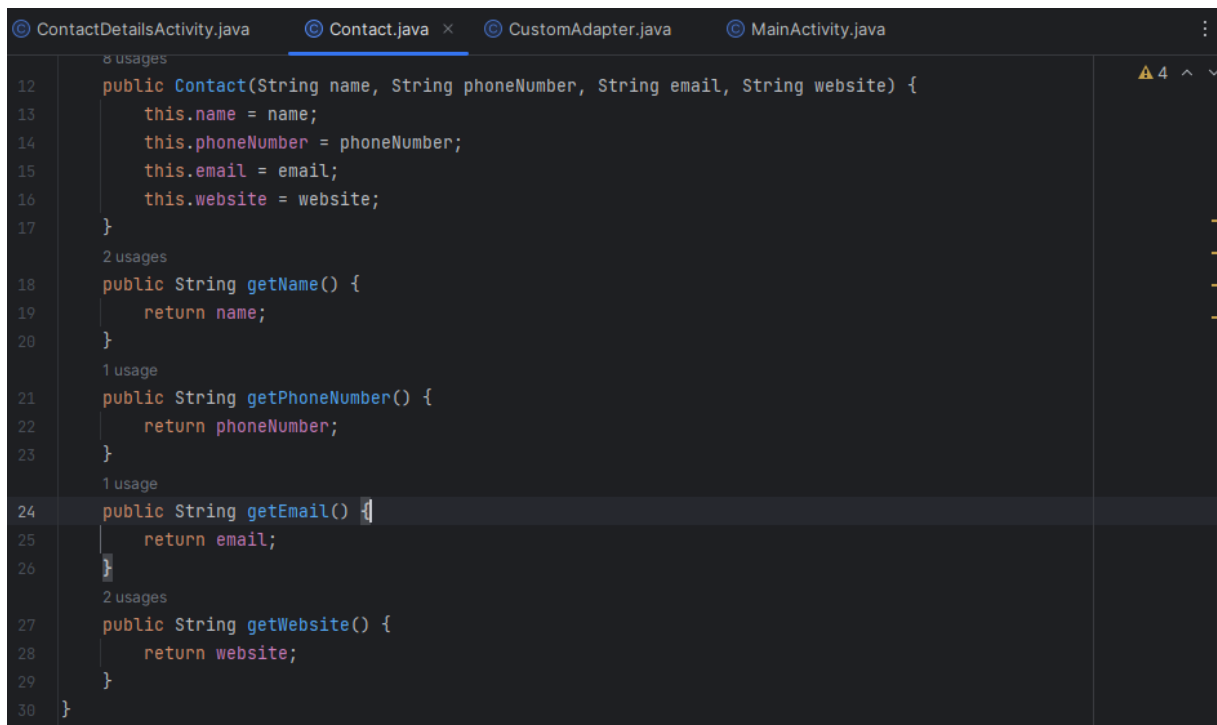
- **Informations du Contact :**



2. Partie Code :

- **Contact.java :**

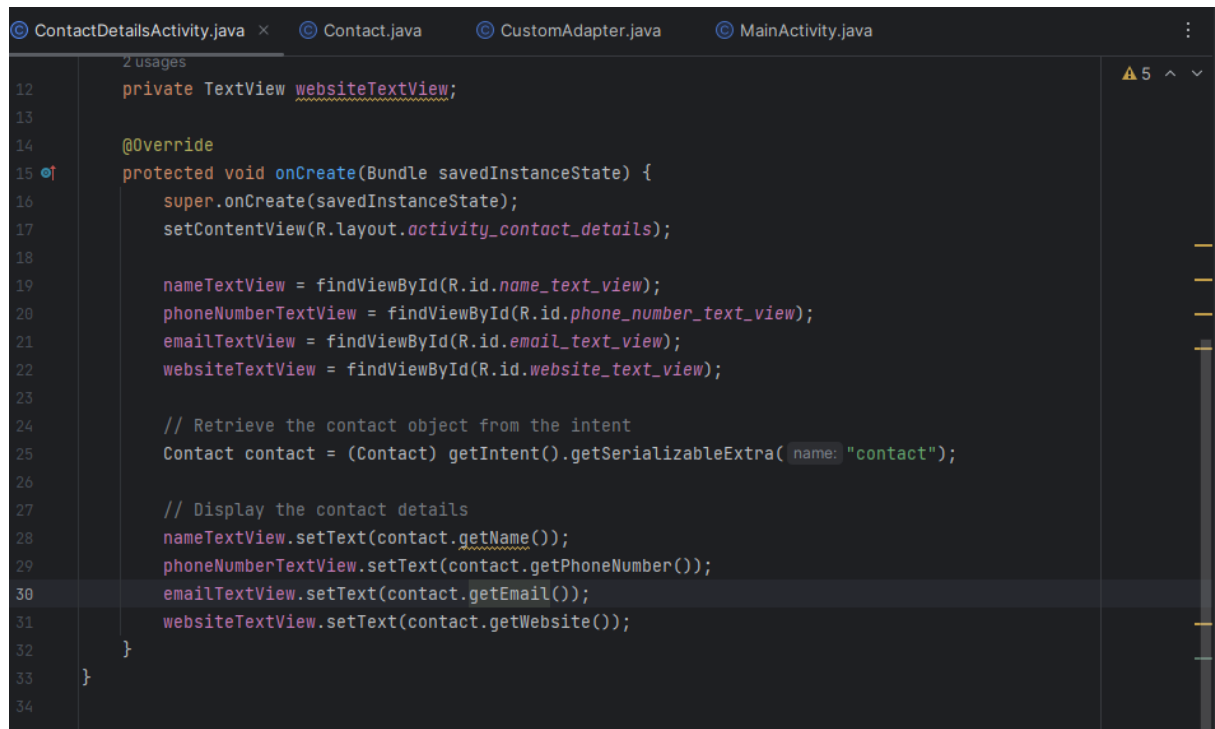
Dans ce code, nous avons créé une classe `Contact` qui représente les informations d'un contact. Cette classe contient des attributs tels que le nom, le numéro de téléphone, l'e-mail et le site web. En implémentant l'interface `Serializable`, nous permettons à un objet de cette classe d'être converti en une séquence d'octets, ce qui le rend apte à être stocké dans un fichier ou à être transmis entre différents composants d'une application Android. Les méthodes `getName()`, `getPhoneNumber()`, `getEmail()` et `getWebsite()` sont fournies pour accéder aux valeurs de ces attributs à partir d'autres parties du code, assurant ainsi une encapsulation des données.



```
12 public Contact(String name, String phoneNumber, String email, String website) {
13     this.name = name;
14     this.phoneNumber = phoneNumber;
15     this.email = email;
16     this.website = website;
17 }
18
19 2 usages
20 public String getName() {
21     return name;
22 }
23
24 1 usage
25 public String getPhoneNumber() {
26     return phoneNumber;
27 }
28
29 1 usage
30 public String getEmail() {
31     return email;
32 }
33
34 2 usages
35 public String getWebsite() {
36     return website;
37 }
38 }
```

- **ContactDetailsActivity.java :**

Son but est d'afficher les détails d'un contact. Dans la méthode **onCreate()**, les éléments de l'interface utilisateur (TextView) sont associés à leurs identifiants dans le fichier de présentation XML. Ensuite, l'objet **Contact** est récupéré à partir de l'intent qui a été transmis à cette activité. Les détails du contact sont ensuite extraits de cet objet et affichés dans les TextView correspondants. Ainsi, lorsque cette activité est lancée, elle affiche les détails du contact transmis via l'intent.



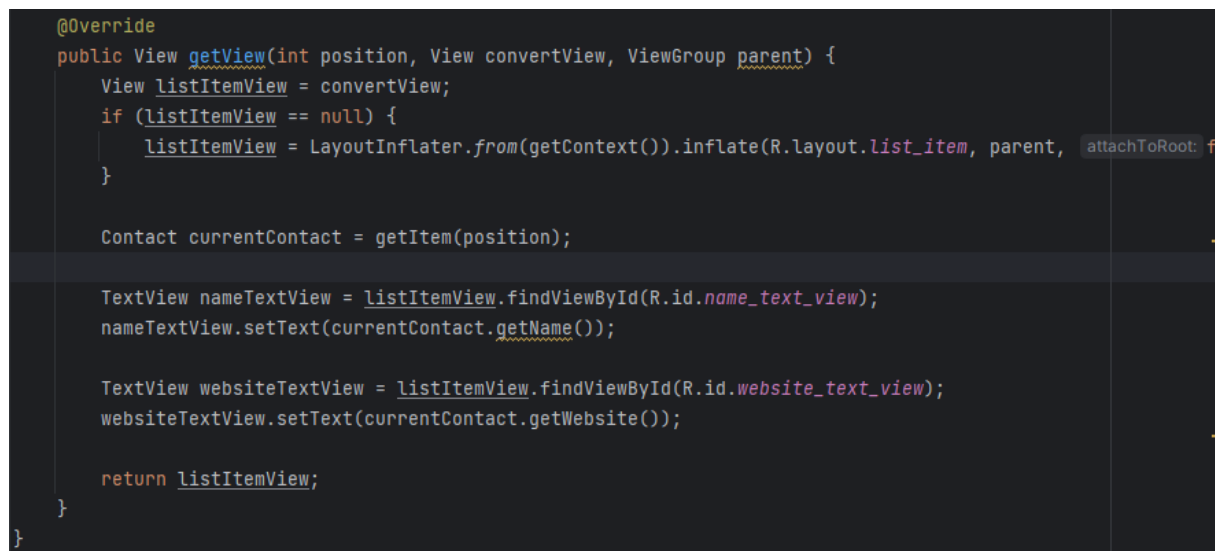
```

ContactDetailsActivity.java x Contact.java CustomAdapter.java MainActivity.java
2 usages
12 private TextView websiteTextView;
13
14 @Override
15 protected void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.activity_contact_details);
18
19     nameTextView = findViewById(R.id.name_text_view);
20     phoneNumberTextView = findViewById(R.id.phone_number_text_view);
21     emailTextView = findViewById(R.id.email_text_view);
22     websiteTextView = findViewById(R.id.website_text_view);
23
24     // Retrieve the contact object from the intent
25     Contact contact = (Contact) getIntent().getSerializableExtra( name: "contact");
26
27     // Display the contact details
28     nameTextView.setText(contact.getName());
29     phoneNumberTextView.setText(contact.getPhoneNumber());
30     emailTextView.setText(contact.getEmail());
31     websiteTextView.setText(contact.getWebsite());
32 }
33
34

```

- *CusterAdapter.java :*

Pour afficher une liste de contacts dans une application Android. L'objectif principal de cet adaptateur est d'afficher les noms et les sites web des contacts dans une ListView. L'adaptateur récupère les informations de chaque contact et les affiche dans les TextView appropriés de chaque élément de la liste, en utilisant la méthode getView().



```

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View listItemView = convertView;
    if (listItemView == null) {
        listItemView = LayoutInflater.from(getContext()).inflate(R.layout.list_item, parent, attachToRoot: f
    }

    Contact currentContact = getItem(position);

    TextView nameTextView = listItemView.findViewById(R.id.name_text_view);
    nameTextView.setText(currentContact.getName());

    TextView websiteTextView = listItemView.findViewById(R.id.website_text_view);
    websiteTextView.setText(currentContact.getWebsite());

    return listItemView;
}
}

```

- *mainActivity.java* :

Dans cette activity, on a écrit le code qui affiche une liste de contacts. Lorsque l'utilisateur clique sur un contact, il est redirigé vers une nouvelle activité qui affiche les détails complets de ce contact.

```
12 <> public class MainActivity extends AppCompatActivity {
13
14     3 usages
15     private ListView listView;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21
22         listView = findViewById(R.id.listView);
23
24         // Create a list of contact objects
25         List<Contact> contacts = new ArrayList<>();
26         contacts.add(new Contact( name: "STEPS",  phoneNumber: "+216 73 503 371",  email: "steps@yahoo.fr",  website: "www.steps.com"));
27         contacts.add(new Contact( name: "UTEK",  phoneNumber: "+216 55 295 578",  email: "uteek@yahoo.fr",  website: "www.uteek.com"));
28         contacts.add(new Contact( name: "UDONEC",  phoneNumber: "+216 75 225 487",  email: "uDonec@yahoo.fr",  website: "www.uDonec.com"));
29         contacts.add(new Contact( name: "Lezarts.digital",  phoneNumber: "+216 98 288 943",  email: "Lezarts@yahoo.fr",  website: "www.Lezarts.com"));
30         contacts.add(new Contact( name: "PULP",  phoneNumber: "+216 72 779 221",  email: "pulp@yahoo.fr",  website: "www.pulp.com"));
31         contacts.add(new Contact( name: "ENVAIST",  phoneNumber: "+216 71 000 018",  email: "envast@yahoo.fr",  website: "www.envast.com"));
32         contacts.add(new Contact( name: "Apeiron",  phoneNumber: "+216 71 799 018",  email: "Apeiron@yahoo.fr",  website: "www.Apeiron.com"));
33         contacts.add(new Contact( name: "MS Agency",  phoneNumber: "+216 71 998 098",  email: "msAgency@yahoo.fr",  website: "www.envast.com"));
34
35         // Create a custom adapter and set it to the list view
36         CustomAdapter adapter = new CustomAdapter( context: this, contacts);
37         listView.setAdapter(adapter);
38
39         // Set a click listener for the list items
40         listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
41             @Override
42             public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
43                 // Get the selected contact
44                 Contact selectedContact = contacts.get(position);
45
46                 // Start a new activity to display the contact details
47                 Intent intent = new Intent( packageContext: MainActivity.this, ContactDetailsActivity.class);
48                 intent.putExtra( name: "contact", selectedContact);
49                 startActivity(intent);
50             }
51         });
52     }
```