

# Application d'Envoi de SMS

## 1. Résumé du Projet

L'application "MessageApp" permet aux utilisateurs d'envoyer des messages SMS à des numéros de téléphone spécifiés. Elle comprend une interface utilisateur simple où les utilisateurs peuvent entrer le numéro de téléphone et le contenu du message, puis envoyer le SMS en appuyant sur un bouton.

## 2. Étapes de Construction du Projet

### Étape 1 : Création du Projet

- **Ouvrir Android Studio** et créer un nouveau projet.
- Sélectionner une "Empty Activity" (activité vide) et nommer le projet "MessageApp".
- Configurer le package et le minimum SDK (API niveau 21 recommandé pour la compatibilité).

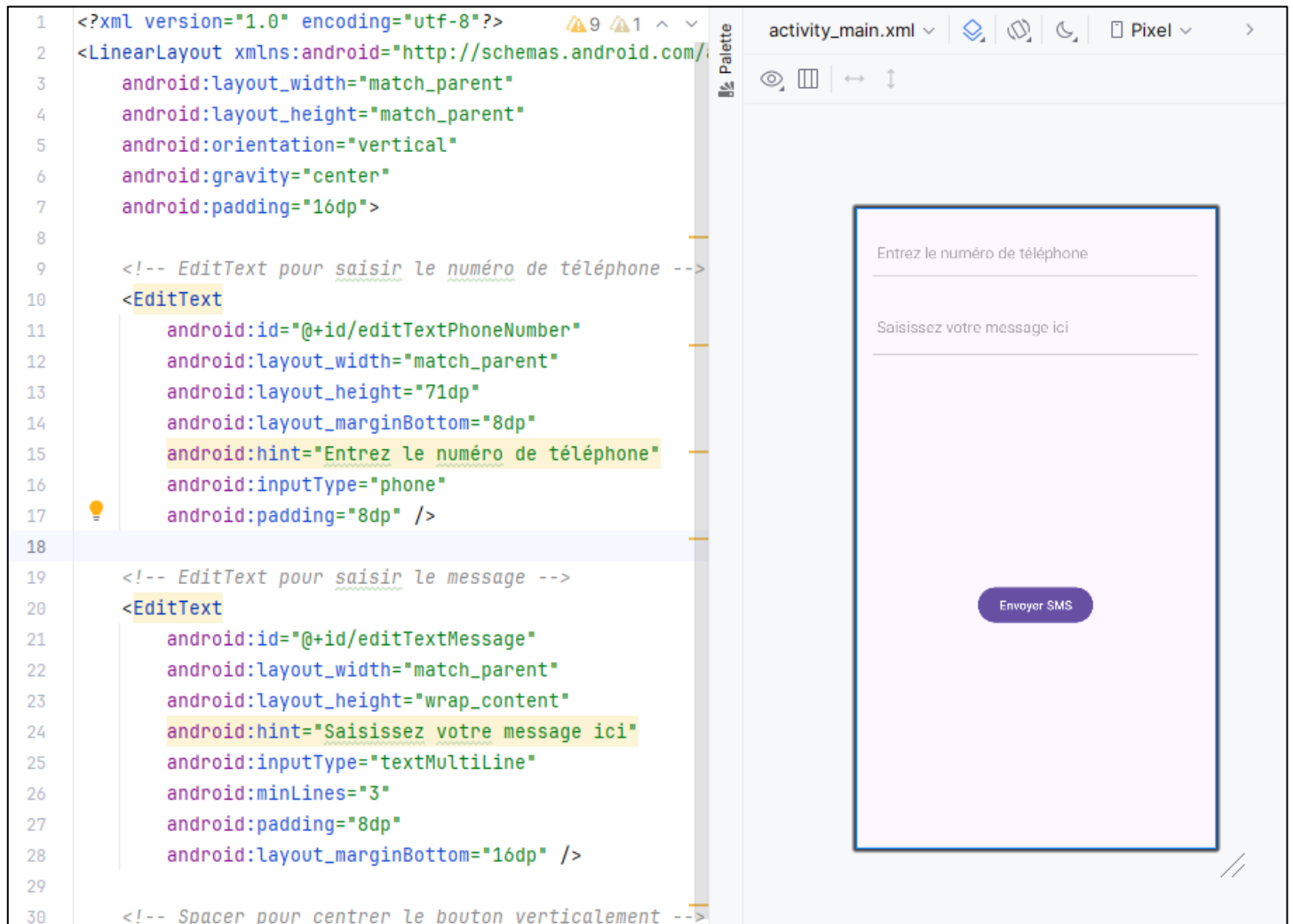
### Étape 2 : Configuration des Permissions

- **Ajouter les permissions** nécessaires pour envoyer des SMS dans le fichier AndroidManifest.xml. Cela inclut :
  - SEND\_SMS pour envoyer des messages.
  - READ\_SMS pour lire les messages (si nécessaire pour des fonctionnalités futures).
  - READ\_BASIC\_PHONE\_STATE pour les fonctionnalités liées à l'état du téléphone.

```
5
6      <uses-feature
7          android:name="android.hardware.telephony"
8          android:required="false" />
9
10     <uses-permission android:name="android.permission.READ_BASIC_PHONE_STATE"/>
11     <uses-permission android:name="android.permission.READ_SMS"/>
12     <uses-permission android:name="android.permission.SEND_SMS"/>
13
```

### Étape 3 : Conception de l'Interface Utilisateur

- **Créer le fichier de mise en page activity\_main.xml** pour définir l'interface utilisateur.
- Utiliser un LinearLayout vertical pour organiser les éléments.
- Ajouter deux EditText :
  - Un pour le numéro de téléphone (editTextPhoneNumber).
  - Un pour le message (editTextMessage).
- Ajouter un Button pour envoyer le message (buttonSendSms).



#### Étape 4 : Implémentation de la Logique d'Envoi de SMS

- **Créer la classe MainActivity** et initialiser les éléments de l'interface utilisateur dans la méthode `onCreate`.
- **Configurer le bouton d'envoi** pour qu'il déclenche l'envoi d'un SMS lorsque l'utilisateur clique dessus.
- Utiliser `SmsManager` pour envoyer le message en utilisant le numéro de téléphone et le texte saisi.

```

4  public class MainActivity extends AppCompatActivity {
5
6      private EditText phone ; 2 usages
7      private EditText message ; 2 usages
8      private Button envoi ; 2 usages
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         initActivity();}
15
16     private void initActivity(){ 1 usage
17         phone=(EditText) findViewById(R.id.editTextPhoneNumber);
18         message=(EditText) findViewById(R.id.editTextMessage);
19         envoi=(Button) findViewById(R.id.buttonSendSms);
20         createOnClickEnvoieButton();
21     }
22     private void createOnClickEnvoieButton(){ 1 usage
23         envoi.setOnClickListener(new View.OnClickListener() {
24             @Override
25             public void onClick(View v) {
26                 SmsManager.getDefault().sendTextMessage(phone.getText().toString(),
27                     scAddress: null,message.getText().toString(),
28                     sentIntent: null, deliveryIntent: null);
29             }
30         });
31     }
32 }

```

## Utilisation de l'Émulateur pour Tester l'Envoi de SMS

### 1. Utiliser l'Emulateur de Téléphone :

- L'émulateur d'Android peut simuler certaines fonctionnalités de SMS. Pour cela, vous devrez envoyer des messages à l'aide de la console de l'émulateur.

### 2. Envoyer des SMS via la Console ADB :

- Ouvrez la ligne de commande ou le terminal.
- Utilisez les commandes ADB pour simuler l'envoi de SMS. Voici comment procéder :
  - Ouvrez une fenêtre de terminal et tapez la commande suivante

```
PS C:\Users\Ihcannah_Remaht\AndroidStudioProjects\MessageApp> adb emu sms send 29291365 "ceci est un msms test (par Hnanachi Thamer"
OK
PS C:\Users\Ihcannah_Remaht\AndroidStudioProjects\MessageApp> 
```

