|  |  |
| --- | --- |
| **Ex No: 6** | CREATE A SMS APPLICATION |
| **Date:** |  |

**AIM:**

To create a SMS Application.

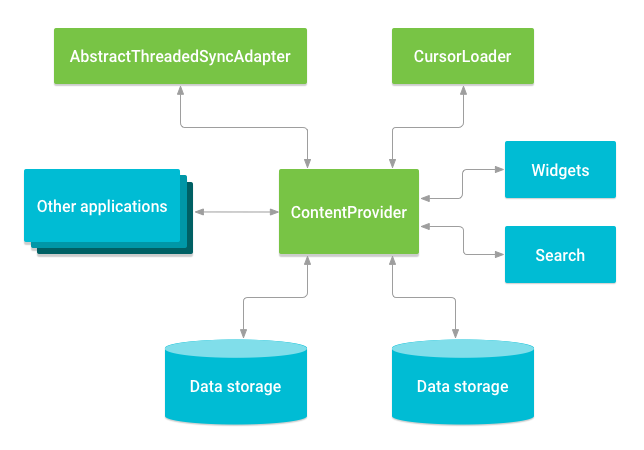
**DESCRIPTION:**

A Content provider manages access to a central repository of data. A provider is part of an Android application, which often provides its own UI for working with the data. However, content providers are primarily used by other applications, which access the provider using a provider client object. Together, providers and provider clients offer a consistent, standard interface to data that also handles interprocess communication and secure data access.

Cursor is a Interface whice returns collection of your query data. moveToFirst() is used to point the cursor position from where you want to get data from your cursor. There are methods moveToLast(), moveToNext(), moveToPrevious(), moveToPosition(position) by which you can iterate through your cursor by desired way.

In Android, you can use SmsManager API or devices Built-in SMS application to send SMS's. The permissions for the same are set:

<uses-permission android:name="android.permission.SEND\_SMS" />



**PROGRAM:**

XML Code-

**activity\_main.xml**

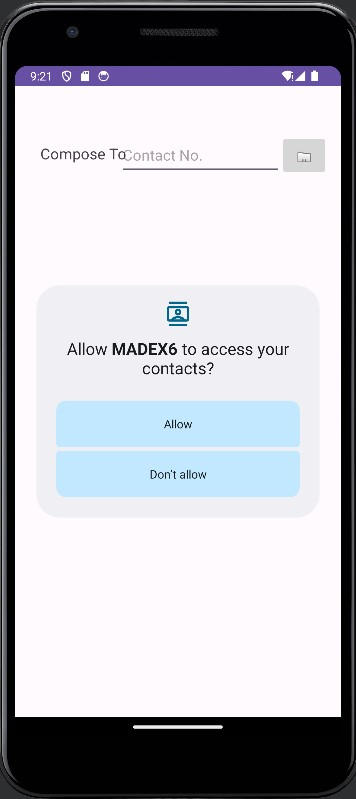
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/txtVw1"  
 android:layout\_width="134dp"  
 android:layout\_height="48dp"  
 android:gravity="center"  
 android:text="@string/compose\_to"  
 android:textAppearance="?android:attr/textAppearanceMedium"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.059"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.081" />  
  
 <EditText  
 android:id="@+id/Cname"  
 android:layout\_width="195dp"  
 android:layout\_height="50dp"  
 android:layout\_marginEnd="72dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/contact\_no"  
 android:inputType="text"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.082" />  
  
 <ImageButton  
 android:id="@+id/contactbook"  
 android:layout\_width="58dp"  
 android:layout\_height="51dp"  
 android:contentDescription="@string/contact\_book"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.954"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.082"  
 app:srcCompat="@android:drawable/sym\_contact\_card" />  
  
 <EditText  
 android:id="@+id/sms"  
 android:layout\_width="290dp"  
 android:layout\_height="48dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:hint="@string/message"  
 android:inputType="text"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.231"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.538" />  
  
 <ImageButton  
 android:id="@+id/send"  
 android:layout\_width="57dp"  
 android:layout\_height="49dp"  
 android:contentDescription="@string/send"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.898"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.538"  
 app:srcCompat="@android:drawable/ic\_menu\_send" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Java Code-

**MainActivity.java**

package com.example.madex6;  
  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.provider.ContactsContract;  
import android.telephony.SmsManager;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
 EditText msg\_send, cn;  
 TextView tv;  
 ImageButton open, send\_msg;  
 private static final int *CONTACT\_PERMISSION\_CODE* = 1;  
 private static final int *CONTACT\_PICK\_CODE* = 2;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 cn = findViewById(R.id.*Cname*);  
 msg\_send = findViewById(R.id.*sms*);  
 send\_msg = findViewById(R.id.*send*);  
 tv = findViewById(R.id.*txtVw1*);  
 open = findViewById(R.id.*contactbook*);  
 open.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 if(checkContactPermission()){  
 pickContactIntent();  
 }  
 else{  
 requestContactPermission();  
 }  
 }  
 });  
 send\_msg.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View view) {  
 if (ContextCompat.*checkSelfPermission*(MainActivity.this, android.Manifest.permission.*SEND\_SMS*)  
 == PackageManager.*PERMISSION\_GRANTED*){  
 sendMessage();  
 }  
 else{  
 ActivityCompat.*requestPermissions*(MainActivity.this, new String[]{android.Manifest.permission.*SEND\_SMS*}, 100);  
 }  
 }  
 });  
 }  
 private void sendMessage(){  
 String phoneno = cn.getText().toString().trim();  
 String message = msg\_send.getText().toString().trim();  
 if(!phoneno.equals("") && !message.equals("")){  
 SmsManager smsManager = SmsManager.*getDefault*();  
 smsManager.sendTextMessage(phoneno, null, message, null, null);  
 Toast.*makeText*(this, "SMS SENT SUCCESSFULLY", Toast.*LENGTH\_LONG*).show();  
 }  
 else{  
 Toast.*makeText*(this, "Type Some Message", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 private boolean checkContactPermission(){  
 boolean result = ContextCompat.*checkSelfPermission*(  
 this,  
 android.Manifest.permission.*READ\_CONTACTS*) == (PackageManager.*PERMISSION\_GRANTED* );  
 return result;  
 }  
  
 private void requestContactPermission(){  
 String[] permissions = {Manifest.permission.*READ\_CONTACTS*};  
 ActivityCompat.*requestPermissions*(this, permissions, *CONTACT\_PERMISSION\_CODE*);  
 }  
  
 private void pickContactIntent(){  
 Intent intent = new Intent(Intent.*ACTION\_PICK*, ContactsContract.CommonDataKinds.Phone.*CONTENT\_URI*);  
 startActivityForResult(intent, *CONTACT\_PICK\_CODE*);  
 }  
  
 public void onRequestPermissionResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults){  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if(requestCode == *CONTACT\_PERMISSION\_CODE*){  
 if(grantResults.length>0 && grantResults[0]==PackageManager.*PERMISSION\_GRANTED*){  
 pickContactIntent();  
 }  
 else{  
 Toast.*makeText*(this, "Permission Denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 if(requestCode == 100 && grantResults.length>0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*){  
 sendMessage();  
 }  
 else{  
 Toast.*makeText*(this, "Permission Denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data){  
 super.onActivityResult(requestCode, resultCode, data);  
 if(resultCode == RESULT\_OK){  
 switch (requestCode){  
 case CONTACT\_PICK\_CODE:  
 contactPicked(data);  
 break;  
 }  
 }  
 else{  
 Toast.makeText(this, "Failed to Pick Contact", Toast.LENGTH\_SHORT).show();  
 }  
 }  
  
 private void contactPicked(Intent data){  
 Cursor cursor = null;  
  
 try{  
 String phoneNo = null;  
 Uri uri = data.getData();  
 cursor = getContentResolver().query(uri, null, null, null, null);  
 cursor.moveToFirst();  
 int phoneIndex = cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER);  
 phoneNo = cursor.getString(phoneIndex);  
 cn.setText(phoneNo);  
 }  
 catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
}

**OUTPUT:**

**RESULT:**

Thus, the experiment for creating a SMS Application has been coded, compiled, and executed successfully.