**[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]**

Practical: 9

AIM: What is Difference between ListView and RecyclerView? Create SMS Android Application that can receive SMS by BroadcastReceiver and Interface, Send SMS and read & display all SMS of Inbox by using RecyclerView.



Submitted By: Kanzariya Dhavanik

Enrollment number: 21012022022



**Department of Computer Engineering/Information Technolog**

|  |  |
| --- | --- |
| **RecyclerView** | **ListView** |
| * The RecyclerView’s adaptor forces us to use the ViewHolder pattern. | * The ListView doesn’t give that kind of protection by default. |
| * Supported Vertical Scrolling and Horizontal Scrolling | * Supported Only Vertical Scrolling |
| * Use of less memory. | * More memory is used for a long list. Sometimes devices get hanged. |
| * Animations using ItemAnimator are easy and smooth. Animations like list appearance and disappearance, adding or removing particular views, and so on. | * It’s complex to use Animation and hard to handle it. |
| * Dividers between items are not shown by default. | * Dividers between items are shown by default. |
| * Use ItemDecorations to add margins and draw on or under an item View. | * ItemDecorations require customization |
| * RecyclerView is a more flexible control for handling a "list of data" | * ListView is a less flexible control for handling a "list of data" |

drawable folder:

**shape.xml**

**<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
 <item>  
 <shape android:shape="oval">  
 <size android:width="120dp" android:height="120dp" />  
 </shape>  
 </item>  
</selector>**

layout folder:

**sms\_item\_view.xml**

**<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="horizontal"  
 android:layout\_marginHorizontal="10dp"  
 android:padding="20dp">  
  
 <ImageView  
 android:id="@+id/icon"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:padding="5dp"  
 android:background="@drawable/shape"  
 android:src="@drawable/ic\_baseline\_person\_24"  
 />  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/textViewPhoneNo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Medium Text"  
 android:textSize="14dp"  
 android:textStyle="bold"  
 android:textAppearance="?android:attr/textAppearanceMedium"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginTop="5dp"  
 android:padding="2dp"  
 android:textColor="#4d4d4d" />  
 <TextView  
 android:id="@+id/textViewMessage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:layout\_marginLeft="10dp"  
 android:textSize="12dp"/>  
 </LinearLayout>  
</LinearLayout>**

**activity\_main.xml**

**<?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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
 <com.google.android.material.appbar.AppBarLayout  
 android:id="@+id/app\_bar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toTopOf="parent">  
 <androidx.appcompat.widget.Toolbar  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:title="Message Application"  
 app:titleTextColor="@color/white"/>  
 </com.google.android.material.appbar.AppBarLayout>  
 <com.google.android.material.card.MaterialCardView  
 android:id="@+id/card"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@id/l1"  
 app:layout\_constraintTop\_toBottomOf="@id/app\_bar"  
 android:layout\_margin="10dp"  
 >  
 <ListView  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="@color/cardview\_shadow\_end\_color"  
 android:layout\_margin="10dp"  
 />  
 </com.google.android.material.card.MaterialCardView>  
 <LinearLayout  
 android:id="@+id/l1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:weightSum="100"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 android:orientation="horizontal"  
 android:layout\_margin="20dp">  
  
 <LinearLayout  
 android:layout\_weight="90"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/phoneNoInput"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Mobile No"  
 app:endIconMode="custom"  
 app:endIconDrawable="@drawable/ic\_baseline\_phone\_android\_24"  
 style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/phoneNo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:inputType="phone"  
 />  
  
 </com.google.android.material.textfield.TextInputLayout>  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/msgInput"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Message"  
 app:endIconMode="custom"  
 app:endIconDrawable="@drawable/ic\_baseline\_message\_24"  
  
 style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/msg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:inputType="textLongMessage"  
 />  
  
 </com.google.android.material.textfield.TextInputLayout>  
 </LinearLayout>  
  
 <com.google.android.material.button.MaterialButton  
 android:id="@+id/sendButton"  
 app:fabSize="normal"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginHorizontal="10dp"  
 android:layout\_marginTop="5dp"  
 app:icon="@drawable/ic\_baseline\_send\_24"  
 app:iconSize="20dp"  
 android:elevation="5dp"  
 app:cornerRadius="15dp"  
 />  
  
 </LinearLayout>  
  
</androidx.constraintlayout.widget.ConstraintLayout>**

coding files:

**MainActivity.kt**

**package com.example.madpractical9\_21012022022  
  
import android.Manifest  
import android.content.IntentFilter  
import android.content.pm.PackageManager  
import android.net.Uri  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.provider.Telephony  
import android.telephony.SmsManager  
import android.widget.ListView  
import androidx.appcompat.app.AlertDialog  
import androidx.core.app.ActivityCompat  
import androidx.core.content.ContextCompat  
import com.example.madpractical9\_21012022022.databinding.ActivityMainBinding  
  
class MainActivity : AppCompatActivity() {  
 private lateinit var binding : ActivityMainBinding  
 private val SMS\_PERMISSION\_CODE = 100  
 private lateinit var lv: ListView  
 private lateinit var al: ArrayList<SMSView>  
 private lateinit var smsReceiver:SMSBroadcastReceiver  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 binding = ActivityMainBinding.inflate(layoutInflater)  
 setContentView(binding.root)  
  
 lv = binding.listView  
 al = ArrayList()  
  
 if(checkRequestPermission()){  
 loadSMSInbox()  
 }  
  
 smsReceiver = SMSBroadcastReceiver()  
 registerReceiver(smsReceiver, IntentFilter(**

**Telephony.Sms.Intents.SMS\_RECEIVED\_ACTION))  
 smsReceiver.setListner(ListenerImplement())  
 binding.sendButton.setOnClickListener{  
 val phone = binding.phoneNo.text.toString()  
 val msg = binding.msg.text.toString()  
 sendSms(phone,msg)  
  
 val builder : AlertDialog.Builder = AlertDialog.Builder(this@MainActivity)  
 builder.setTitle("Sent SMS")**

**builder.setMessage("SMS is sent.\nPhone No : $phone \n\n Message : $msg")  
 builder.setCancelable(true)  
 builder.setPositiveButton("OK",null);  
 builder.show()  
 }  
  
 }  
  
 private fun sendSms(sPhoneNo: String?,sMsg: String?){  
 if(!checkRequestPermission()){  
 return  
 }  
 val smsmanager = SmsManager.getDefault()  
 if(smsmanager != null){  
 smsmanager.sendTextMessage(sPhoneNo,null,sMsg,null,null)  
 }  
 }  
  
 inner class ListenerImplement:SMSBroadcastReceiver.Listner{  
 override fun onTextReceived(sPhoneNo: String?, sMsg: String?) {  
 val builder : AlertDialog.Builder = AlertDialog.Builder(this@MainActivity)  
 builder.setTitle("New SMS Received")  
 builder.setMessage("From : $sPhoneNo\n\n Message : $sMsg")  
 builder.setCancelable(true)  
 builder.setPositiveButton("OK",null);  
 builder.show()  
 loadSMSInbox()  
 }  
 }  
  
 private val isSMSReadPermission: Boolean  
 get() = ContextCompat.checkSelfPermission(this, Manifest.permission.READ\_SMS) ==**

**PackageManager.PERMISSION\_GRANTED  
 private val isSMSWritePermission: Boolean  
 get() = ContextCompat.checkSelfPermission(this, Manifest.permission.SEND\_SMS) ==**

**PackageManager.PERMISSION\_GRANTED  
  
 private fun requestSMSPermission() {  
 if (ActivityCompat.shouldShowRequestPermissionRationale**

**(this, Manifest.permission.READ\_SMS)) {  
 // You may display a non-blocking explanation here,**

**read more in the documentation:  
 // https://developer.android.com/training/permissions/requesting.html  
 }  
  
 ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.READ\_SMS,  
 Manifest.permission.SEND\_SMS,  
 Manifest.permission.RECEIVE\_SMS),  
 SMS\_PERMISSION\_CODE)  
 }  
 private fun checkRequestPermission(): Boolean {  
 return if (!isSMSReadPermission || !isSMSWritePermission) {  
 requestSMSPermission()  
 false  
 } else true  
 }  
  
 private fun loadSMSInbox() {  
 if (!checkRequestPermission()) return  
 val uriSMS = Uri.parse("content://sms/inbox")  
 val c = contentResolver.query(uriSMS, null, null, null, null)  
 al.clear()  
 while (c!!.moveToNext()) {  
 al.add(SMSView(c.getString(2),c.getString(12)))  
 }  
 lv.adapter = SMSViewAdapter(this,al)  
  
 }  
  
 override fun onDestroy() {  
 unregisterReceiver(smsReceiver)  
 super.onDestroy()  
 }  
}**

**SMSBroadcastReceiver.kt**

**package com.example.madpractical9\_21012022022  
  
import android.content.BroadcastReceiver  
import android.content.Context  
import android.content.Intent  
import android.os.Build  
import android.provider.Telephony  
  
class SMSBroadcastReceiver : BroadcastReceiver() {  
  
 interface Listner {  
 fun onTextReceived(sPhoneNo: String?, sMsg: String?)  
 }  
 private var listner: Listner? = null  
 fun setListner(lis: Listner) {  
 listner = lis  
 }  
 override fun onReceive(context: Context, intent: Intent) {  
 if (intent.action == Telephony.Sms.Intents.SMS\_RECEIVED\_ACTION) {  
 var sPhoneNo = ""  
 var sSMSBody = ""  
 if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M) {  
 for (smsMessage in Telephony.Sms.Intents.getMessagesFromIntent(intent)) {  
 sPhoneNo = smsMessage.displayOriginatingAddress  
 sSMSBody += smsMessage.messageBody  
 }  
 if (listner != null) {  
 listner?.onTextReceived(sPhoneNo, sSMSBody)  
 }  
 }  
 }  
 }  
}**

**SMSView.kt**

**package com.example.madpractical9\_21012022022  
  
class SMSView(val phoneNo:String,val msg:String) {  
}**

**SMSViewAdapter.kt**

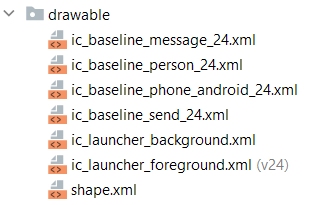
**package com.example.madpractical9\_21012022022  
  
import android.content.Context  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ArrayAdapter  
import com.example.madpractical9\_21012022022.databinding.SmsItemViewBinding  
  
class SMSViewAdapter(context: Context, private val array:ArrayList<SMSView>):  
 ArrayAdapter<SMSView>(context,array.size,array) {  
 override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {  
 //val currentItemView:View = LayoutInflater.from(context)**

**.inflate(R.layout.sms\_item\_view, parent, false)  
 // get the position of the view from the ArrayAdapter  
 val currentSms: SMSView? = getItem(position)  
 val binding = SmsItemViewBinding.inflate(LayoutInflater.from(context))  
 binding.textViewPhoneNo.text = currentSms!!.phoneNo  
 binding.textViewMessage.text = currentSms.msg  
 return binding.root  
 //return super.getView(position, convertView, parent)  
 }  
}**

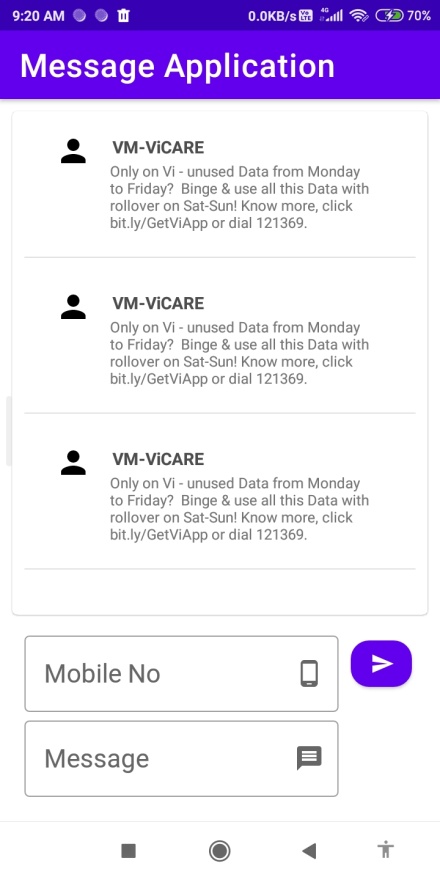
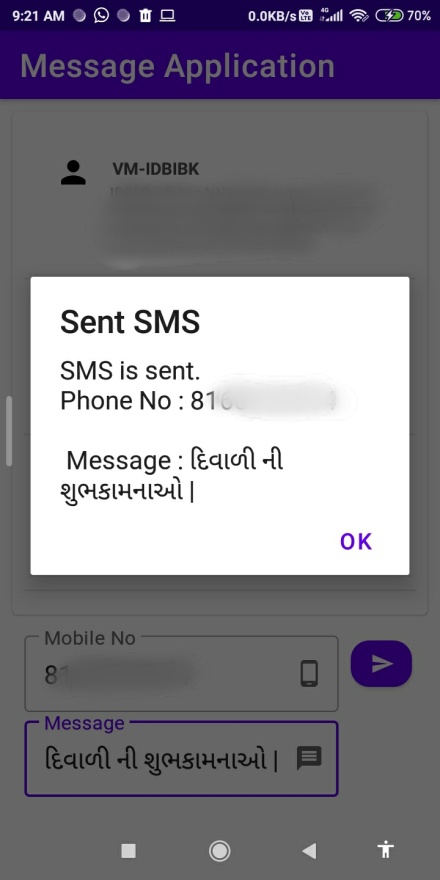
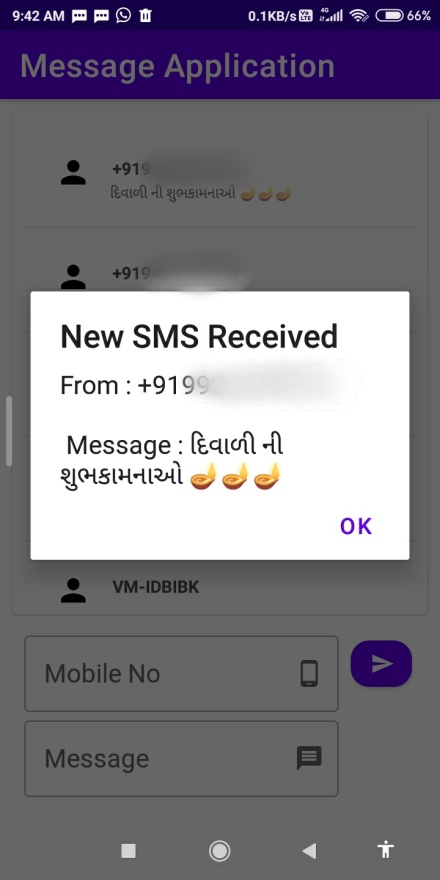
**AndroidManifest.xml**

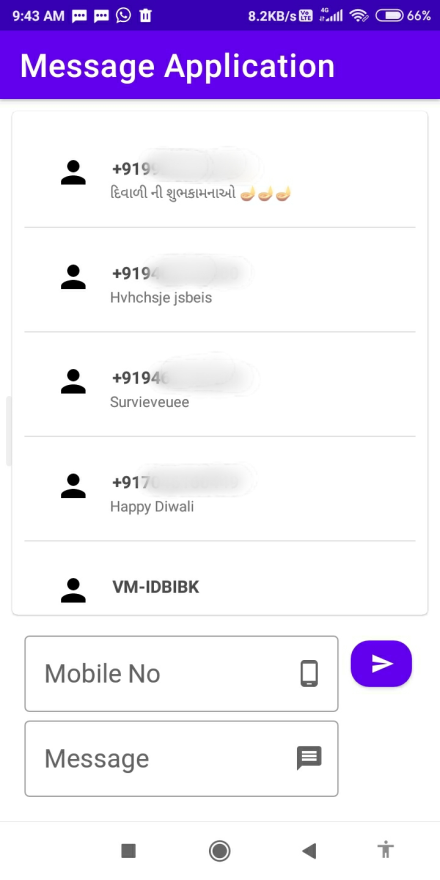
**<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.example.madpractical9\_21012022022" >  
  
 <uses-permission android:name="android.permission.RECEIVE\_SMS" />  
 <uses-permission android:name="android.permission.READ\_SMS" />  
 <uses-permission android:name="android.permission.SEND\_SMS" />  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="Practical 9"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MADPractical9\_21012022022"  
 tools:targetApi="31" >  
 <receiver  
 android:name=".SMSBroadcastReceiver"  
 android:enabled="true"  
 android:exported="true" >  
 </receiver>  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true" >  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>**

Resources:



**Output:**

**  **

****