**Main activity**

**package** com.examples.terminal  
  
**import** android.app.PendingIntent  
**import** android.content.Intent  
**import** android.net.Uri  
**import** androidx.appcompat.app.AppCompatActivity  
**import** android.os.Bundle  
**import** android.telephony.SmsManager  
**import** android.widget.Toast  
**import** androidx.recyclerview.widget.LinearLayoutManager  
**import** androidx.recyclerview.widget.RecyclerView  
**import** com.example.terminal.R  
  
**class** MainActivity : AppCompatActivity() {  
 **private lateinit var itemList**: RecyclerView  
 **val items** = *arrayOf*(  
 ItemModel(**"Amna"**, **"090"**),  
 ItemModel(**"Amina"**, **"032"**),  
 ItemModel(**"Bushra"**, **"042"**),  
 ItemModel(**"Rabia"**, **"030"**),  
 ItemModel(**"Zunaira"**, **"025"**)  
 )  
 **override fun** onCreate(savedInstanceState: Bundle?) {  
 **super**.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 **itemList** = findViewById(R.id.*items\_list*)  
 **val** adaptor = Adaptor(**items**)  
 **itemList**.*adapter* = adaptor  
 **itemList**.*layoutManager* = LinearLayoutManager(**this**)  
  
 adaptor.setOnItemClickListener(**object**:Adaptor.onItemClickListener{  
 **override fun** onItemClick(position: Int) {  
 sendSMS(**"1234"**, **"hi"**)  
 }  
 })  
 }  
  
 **private fun** sendSMS(phoneNumber: String, message: String) {  
 **val** sentPI: PendingIntent = PendingIntent.getBroadcast(**this**, 0, Intent(**"SMS\_SENT"**), 0)  
 SmsManager.getDefault().sendTextMessage(phoneNumber, **null**, message, sentPI, **null**)  
 }  
}

adapter

**package** com.examples.terminal  
  
**import** android.view.LayoutInflater  
**import** android.view.View  
**import** android.view.ViewGroup  
**import** android.widget.Button  
**import** android.widget.TextView  
**import** androidx.recyclerview.widget.RecyclerView  
**import** com.example.terminal.R  
  
**class** Adaptor(**val items**:Array<ItemModel>): RecyclerView.Adapter<Adaptor.ViewHolder>() {  
  
 **private lateinit var mlistener**:onItemClickListener  
 **interface** onItemClickListener  
 {  
 **fun** onItemClick(position: Int)  
  
 }  
 **fun** setOnItemClickListener(listener: onItemClickListener)  
 {  
 **mlistener**=listener  
 }  
  
 **override fun** onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
 **val** view = LayoutInflater.from(parent.*context*).inflate(R.layout.*product\_item*, parent, **false**)  
 **return** ViewHolder(view, **mlistener**)  
 }  
  
 **override fun** onBindViewHolder(holder: ViewHolder, position: Int) {  
 holder.**title**.*text* = **items**[position].**title** holder.**quantity**.*text* = **items**[position].**quantity**.toString()  
  
 }

item model

**package** com.examples.terminal  
  
**class** ItemModel(  
 **val title**:String,  
 **val quantity**:String  
)

build gradle

buildFeatures**{** viewBinding **true  
}**

receiver

**package** com.examples.terminal  
**import** android.content.BroadcastReceiver  
**import** android.content.Context  
**import** android.content.Intent  
**import** android.widget.Toast  
  
**class** MyReceiver: BroadcastReceiver() {  
 **override fun** onReceive(context: Context, intent: Intent) {  
 *// This method is called when the BroadcastReceiver is receiving an Intent broadcast.* Toast.makeText(context,**"MyReceiver"**,Toast.*LENGTH\_LONG*).show()  
 }  
}

service

**package** com.example.services  
  
**import** android.app.Service  
**import** android.content.Intent  
**import** android.nfc.Tag  
**import** android.os.IBinder  
**import** android.util.Log  
  
**class** MyService : Service() {  
  
 **val TAG** =**"MyService"  
  
 override fun** onBind(intent: Intent): IBinder?=**null  
  
 override fun** onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {  
 **val** datastring= intent?.getStringExtra(**"ExtraData"**)  
 datastring?.let**{  
 if** (datastring != **null**) {  
 Log.d(**TAG**,datastring)  
 }  
 **}** Thread**{  
 while** (**true**){}  
 **}**.start()  
 Log.d(**TAG**,**"Service has been Running"**)  
 **return** *START\_STICKY* }  
  
 **override fun** onDestroy() {  
 **super**.onDestroy()  
 Log.d(**TAG**,**"Service has been Killed"**)  
 }  
}  
  
}