Base code

This is the base code that was created to start the application. There were many changes made to the code to create the finished product. This code was separated into the Kotlin language and the XML language.

Main page

Kotlin + Java

package com.example.smsfilteringapplication  
  
import android.Manifest  
import android.app.Activity  
import android.content.Context  
import android.content.DialogInterface  
import android.content.Intent  
import android.content.IntentFilter  
import android.content.pm.PackageManager  
import android.net.Uri  
import android.os.Build  
import android.os.Bundle  
import android.provider.Telephony  
import android.util.Log  
import android.widget.Button  
import android.widget.Toast  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.app.ActivityCompat  
import androidx.core.content.ContextCompat  
import com.example.smsfilteringapplication.screens.Blacklist  
import com.example.smsfilteringapplication.screens.Evalmailbox  
import com.example.smsfilteringapplication.screens.KeywordManager  
import com.example.smsfilteringapplication.screens.Messagereporting  
import com.example.smsfilteringapplication.screens.Whitelist  
import com.example.smsfilteringapplication.services.SmsReceiver  
  
  
class MainActivity : AppCompatActivity() {  
 lateinit var receiver: SmsReceiver  
 override fun onCreate(savedInstanceState: Bundle?) {  
 //define and register receiver  
 receiver = SmsReceiver()  
 registerReceiver(receiver, IntentFilter("android.provider.Telephony.SMS\_RECEIVED"))  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 //ask to make default sms application. popup doesnt appear for some reason if someone has a way to fix that would be cool.  
 val setSmsAppIntent = Intent(Telephony.Sms.Intents.*ACTION\_CHANGE\_DEFAULT*)  
 setSmsAppIntent.putExtra(Telephony.Sms.Intents.*EXTRA\_PACKAGE\_NAME*, *packageName*)  
 startActivity(setSmsAppIntent)  
  
 //checks if permissions already exist if they do not it asks for them  
 if(ActivityCompat.checkSelfPermission(this,android.Manifest.permission.*WRITE\_CONTACTS*) != PackageManager.*PERMISSION\_GRANTED*){  
 ActivityCompat.requestPermissions(this,*arrayOf*(android.Manifest.permission.*RECEIVE\_MMS*, android.Manifest.permission.*WRITE\_CONTACTS*,android.Manifest.permission.*RECEIVE\_WAP\_PUSH*,android.Manifest.permission.*READ\_SMS*,android.Manifest.permission.*RECEIVE\_SMS*,android.Manifest.permission.*SEND\_SMS*,android.Manifest.permission.*SEND\_SMS*),111)  
 }  
 else  
 Toast.makeText(  
 *applicationContext*,  
 "permissions granted",  
 Toast.*LENGTH\_LONG* ).show()  
  
 //button navigation to other pages.  
 val evalbtn = findViewById<Button>(R.id.*eval\_mailbox\_button*)  
 evalbtn.setOnClickListener**{** val intent = Intent(this, Evalmailbox::class.*java*)  
 startActivity(intent)  
 **}** val BLbutton = findViewById<Button>(R.id.*BLbutton*)  
 BLbutton.setOnClickListener **{** val intent = Intent(this, Blacklist::class.*java*)  
 startActivity(intent)  
 **}** val MSbutton = findViewById<Button>(R.id.*MSbutton*)  
 MSbutton.setOnClickListener **{** val intent = Intent(this, Messagereporting::class.*java*)  
 startActivity(intent)  
 **}** val WLbutton = findViewById<Button>(R.id.*WLbutton*)  
 WLbutton.setOnClickListener **{** val intent = Intent(this, Whitelist::class.*java* )  
 startActivity(intent)  
 **}** val KWbutton = findViewById<Button>(R.id.*KWbutton*)  
 KWbutton.setOnClickListener **{** val intent = Intent(this, KeywordManager::class.*java*)  
 startActivity(intent)  
 **}** }  
 // runs after permissions request goes through to check if permissions were granted or not.  
 override fun onRequestPermissionsResult(  
 requestCode: Int,  
 permissions: Array<out String>,  
 grantResults: IntArray  
 ) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
 if(requestCode == 111 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*)  
  
 Toast.makeText(  
 *applicationContext*,  
 "permissions granted",  
 Toast.*LENGTH\_LONG* ).show()  
 }  
  
  
  
 //more testing functions  
 fun deleteSMS(context: Context, message: String, number: String) {  
 try {  
 Log.d("sms blocker", "Deleting SMS from inbox")  
 val uriSms = Uri.parse("content://sms/inbox")  
 val c = context.*contentResolver*.query(  
 uriSms, *arrayOf*(  
 "\_id", "thread\_id", "address",  
 "person", "date", "body"  
 ), null, null, null  
 )  
 if (c != null && c.moveToFirst()) {  
 do {  
 val id = c.getLong(0)  
 val threadId = c.getLong(1)  
 val address = c.getString(2)  
 val body = c.getString(5)  
 if (message == body && address == number) {  
 Log.d("sms blocker","Deleting SMS with id: $threadId")  
 context.*contentResolver*.delete(  
 Uri.parse("content://sms/$id"), null, null  
 )  
 }  
 } while (c.moveToNext())  
 }  
 } catch (e: Exception) {  
 Log.d("sms blocker","Could not delete SMS from inbox: " + e.message)  
 }  
 }  
 fun checkAndRequestDefaultSmsApp(activity: Activity) {  
 if (Telephony.Sms.getDefaultSmsPackage(activity) != activity.*packageName*) {  
 // Not default SMS app  
 val intent = Intent(Telephony.Sms.Intents.*ACTION\_CHANGE\_DEFAULT*)  
 intent.putExtra(Telephony.Sms.Intents.*EXTRA\_PACKAGE\_NAME*, activity.*packageName*)  
 activity.startActivityForResult(intent, 111)  
 } else {  
 // Already default SMS app, proceed to check runtime permission  
 checkAndRequestRuntimePermission(activity)  
 }  
 }  
 fun checkAndRequestRuntimePermission(activity: Activity) {  
 if (ContextCompat.checkSelfPermission(activity, android.Manifest.permission.*READ\_SMS*) !=  
 PackageManager.*PERMISSION\_GRANTED*) {  
  
 ActivityCompat.requestPermissions(  
 activity,  
 *arrayOf*(android.Manifest.permission.*READ\_SMS*),  
 111  
 )  
 } else {  
 // Permission already granted, proceed with reading SMS or blocked numbers  
 }  
 }  
  
 // Handle the permission result  
 // override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {  
 // super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
 // if (requestCode == 111) {  
 // if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {  
 // Permission granted, proceed with the action  
 // } else {  
 // Permission denied, handle the error  
 // }  
 // }  
 // }  
 }

XML code

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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"  
 android:background="#7BCCE4"  
 android:backgroundTint="#95D9ED"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="50dp"  
 android:background="#95D9ED"  
 android:fontFamily="serif"  
 android:padding="25dp"  
 android:selectAllOnFocus="false"  
 android:text="INBOX GUARDIAN"  
 android:textAllCaps="true"  
 android:textColor="#120D0D"  
 android:textIsSelectable="true"  
 android:textSize="34sp" />  
  
 <Button  
 android:id="@+id/BLbutton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="140dp"  
 android:layout\_marginTop="241dp"  
 android:layout\_marginEnd="140dp"  
 android:background="#5061C1"  
 android:text="Black List"  
 android:textAllCaps="true"  
 android:textColor="#EFE8E8" />  
  
 <Button  
 android:id="@+id/WLbutton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/BLbutton"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="140dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="140dp"  
 android:background="#5061C1"  
 android:text="White List"  
 android:textAllCaps="true"  
 android:textColor="#EFE8E8" />  
  
 <Button  
 android:id="@+id/MSbutton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/WLbutton"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="140dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="140dp"  
 android:background="#5061C1"  
 android:text="Messages"  
 android:textAllCaps="true"  
 android:textColor="#EFE8E8" />  
  
 <Button  
 android:id="@+id/KWbutton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/MSbutton"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="140dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="140dp"  
 android:background="#5061C1"  
 android:text="Keyword"  
 android:textAllCaps="true"  
 android:textColor="#EFE8E8" />  
  
 <Button  
 android:id="@+id/eval\_mailbox\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignTop="@+id/BLbutton"  
 android:layout\_alignBottom="@+id/KWbutton"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="120dp"  
 android:layout\_marginTop="-78dp"  
 android:layout\_marginEnd="120dp"  
 android:layout\_marginBottom="235dp"  
 android:background="#5061C1"  
 android:text="Evaluation Mailbox"  
 android:textAlignment="center"  
 android:textAllCaps="true"  
 android:textColor="#EFE8E8" />  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="216dp"  
 android:layout\_height="194dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginStart="120dp"  
 android:layout\_marginEnd="120dp"  
 android:layout\_marginBottom="35dp"  
 android:background="#3EB8EF"  
 android:backgroundTint="#95D9ED"  
 android:visibility="visible"  
 app:srcCompat="@drawable/inboxguardian2"  
 tools:visibility="visible" />  
  
</RelativeLayout>

Evaluation Mailbox

Kotlin +Java

package com.example.smsfilteringapplication.screens

import android.content.ContentValues  
import android.content.Intent  
import android.content.pm.PackageManager  
import android.net.Uri  
import android.os.Bundle  
import android.provider.BlockedNumberContract  
import android.widget.AdapterView  
import android.widget.Button  
import android.widget.ListView  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.content.ContextCompat  
import com.example.smsfilteringapplication.MainActivity  
import com.example.smsfilteringapplication.R  
import com.example.smsfilteringapplication.services.smsviewadapter  
import android.content.Context  
  
class Evalmailbox : AppCompatActivity() {  
 val sms\_id\_list = *arrayListOf*<String>()  
 private val fromlist = *arrayListOf*<String>()  
 val bodylist = *arrayListOf*<String>()  
 private val smsbodyList = *arrayListOf*<String>()  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*eval\_mailbox*)  
  
 val listView = findViewById<ListView>(R.id.*eval\_listview*)  
 //function to query database sms messages into the from list and body list must be called here @cameron  
 //these are filler values  
 smsbodyList.add("this is a test message")  
 fromlist.add("4096655441")  
  
 listView.*adapter*= smsviewadapter(this,fromlist,smsbodyList)  
  
 val mainmenubutton = findViewById<Button>(R.id.*eval\_mainmenubtn*) // navigation button to main menu  
 mainmenubutton.setOnClickListener **{** val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 **}** val refreshButton = findViewById<Button>(R.id.*eval\_additembtn*)  
 refreshButton.setOnClickListener **{** smsbodyList.clear()  
 fromlist.clear()  
 //need call to function to query messages from database and refresh fromlist and bodylist  
 listView.*adapter*= smsviewadapter(this,fromlist,smsbodyList)  
 **}** listView.*onItemClickListener* = AdapterView.OnItemClickListener **{** parent, view, position, id **->** // Set the message and title for the dialog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Manage Message")  
 builder.setMessage("Do you want to flag the message or approve the message? click any where outside of the window to cancel.")  
  
 // Add a Confirm button and its logic  
 builder.setPositiveButton("Approve") **{** dialog, which **->** // need function to remove sms from database here  
 // need function to write sms into file here  
 val values = ContentValues().*apply* **{** put("address", fromlist.get(position))  
 put("body", smsbodyList.get(position))  
 put("read", false) // 0 for unread, 1 for read  
 put("date", System.currentTimeMillis())  
 // For API level 19 and above, use "type" -> Telephony.Sms.MESSAGE\_TYPE\_INBOX  
 put("type", "1") // 1 for received SMS  
 **}** try {  
 this.*contentResolver*.insert(Uri.parse("content://sms/inbox"), values)  
 } catch (e: Exception) {  
 e.printStackTrace()  
 }  
 //smsbodyList.clear()  
 //fromlist.clear()  
 //sms\_id\_list.clear()  
  
 //call function to query messages from database into bodylist and from list  
  
 //listView.adapter= smsviewadapter(this,fromlist,smsbodyList)  
 **}** // Add a Cancel button and its logic  
 builder.setNegativeButton("Flag") **{** dialog, which **->** // need function to delete message from database here placeholder below  
 fromlist.removeAt(position)  
 smsbodyList.removeAt(position)  
 sms\_id\_list.removeAt(position)  
 //function to add number to blacklist  
 addNumberToBlockedList(fromlist.get(position))  
 //smsbodyList.clear()  
 //fromlist.clear()  
 //sms\_id\_list.clear()  
 //requery sms  
 listView.*adapter*= smsviewadapter(this,fromlist,smsbodyList)  
 **}** // set the dialog to close when the user touches outside of it  
 builder.setCancelable(true)  
  
 // Create and show the AlertDialog  
 val dialog = builder.create()  
 dialog.show()  
  
  
  
 **}** }  
 fun addNumberToBlockedList(number: String) {  
 if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.*WRITE\_CONTACTS*) == PackageManager.*PERMISSION\_GRANTED*) {  
 val values = ContentValues()  
 values.put(BlockedNumberContract.BlockedNumbers.*COLUMN\_ORIGINAL\_NUMBER*, number)  
 *contentResolver*.insert(BlockedNumberContract.BlockedNumbers.*CONTENT\_URI*, values)  
 } else {  
 // Handle the lack of permission here.  
 }  
 }

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"  
 android:background="#7BCCE4"  
 android:backgroundTint="#95D9ED">  
  
 <ListView  
 android:id="@+id/eval\_listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignBottom="@id/eval\_listview"  
 android:layout\_marginTop="60dp"  
 android:layout\_marginBottom="0dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginStart="5dp"></ListView>

<Button  
 android:id="@+id/eval\_mainmenubtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="5dp"  
 android:background="#5061C1"  
 android:text="main menu"  
 android:textColor="#EFE8E8"  
 app:layout\_constraintBottom\_toTopOf="@+id/eval\_listview"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/eval\_additembtn"  
 android:layout\_width="145dp"  
 android:layout\_height="48dp"  
  
 android:layout\_alignParentEnd="true"  
  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="5dp"  
 android:background="#5061C1"  
 android:text="refresh messages"  
 android:textColor="#EFE8E8"  
 app:layout\_constraintBottom\_toTopOf="@+id/eval\_listview"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toEndOf="@+id/eval\_mainmenubtn"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Blacklist

Kotlin +Java

package com.example.smsfilteringapplication.screens  
import android.annotation.SuppressLint  
import android.content.ContentResolver  
import android.content.ContentValues  
import android.content.Context  
import android.content.Intent  
import android.content.pm.PackageManager  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.os.PersistableBundle  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ArrayAdapter  
import android.widget.BaseAdapter  
import android.widget.Button  
import android.widget.EditText  
import android.widget.ListView  
import android.widget.TextView  
import android.widget.Toast  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.widget.SearchView  
import com.example.smsfilteringapplication.MainActivity  
import com.example.smsfilteringapplication.R  
import com.example.smsfilteringapplication.services.blacklistAdapter  
import com.google.android.material.floatingactionbutton.FloatingActionButton  
import android.provider.BlockedNumberContract  
import android.database.Cursor  
import android.widget.AdapterView  
import androidx.core.content.ContextCompat  
  
  
  
public class Blacklist : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*blacklist*)  
 val listView = findViewById<ListView>(R.id.*blacklist\_listview*)  
 listView.*adapter*= blacklistAdapter(this,getBlockedNumbers(this)) //custom list adapter telling list what to render.  
  
 val mainmenubutton = findViewById<Button>(R.id.*mainmenubtn*) // navigation button to main menu  
 mainmenubutton.setOnClickListener **{** val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 **}** val addItemButton = findViewById<Button>(R.id.*additembtn*)  
 addItemButton.setOnClickListener **{** //add item dialogue  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Add Item")  
  
 val inflater = LayoutInflater.from(this)  
 val dialogLayout = inflater.inflate(R.layout.*dialogue\_add\_item*, null)  
  
 val editText: EditText = dialogLayout.findViewById(R.id.*editTextItem*)  
 builder.setView(dialogLayout)  
  
 builder.setPositiveButton("Add") **{** \_, \_ **->** val newItem = editText.*text*.toString().*trim*()  
 if (newItem.*isNotEmpty*() && newItem.*matches*(Regex("^[0-9]+$"))) {  
 //if conditions are met the item is added to the back end blacklist and the list view is updated  
 addNumberToBlockedList(newItem)  
 listView.*adapter*= blacklistAdapter(this,getBlockedNumbers(this))  
 } else {  
 Toast.makeText(this, "Item cannot be empty and must be numbers only ", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** builder.setNegativeButton("Cancel") **{** dialog, \_ **->** dialog.cancel()  
 **}** builder.show()  
 //  
 listView.*adapter*= blacklistAdapter(this,getBlockedNumbers(this))  
 **}** listView.*onItemClickListener* = AdapterView.OnItemClickListener **{** parent, view, position, id **->** // Reading the text content of the clicked TextView  
 val textContent = findViewById<TextView>(R.id.*item\_phone\_number*).*text*.toString()  
  
 // Set the message and title for the dialog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Remove Number")  
 builder.setMessage("Do you want to complete this action?")  
  
 // Add a Confirm button and its logic  
 builder.setPositiveButton("Confirm") **{** dialog, which **->** // Perform actions after confirmation here  
  
 removeBlockedNumber(this,textContent)  
 listView.*adapter*= blacklistAdapter(this,getBlockedNumbers(this))  
 **}** // Add a Cancel button and its logic  
 builder.setNegativeButton("Cancel") **{** dialog, which **->  
  
 }** // set the dialog to not close when the user touches outside of it  
 builder.setCancelable(false)  
  
 // Create and show the AlertDialog  
 val dialog = builder.create()  
 dialog.show()  
  
  
  
 **}** }  
 fun addNumberToBlockedList(number: String) {  
 if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.*WRITE\_CONTACTS*) == PackageManager.*PERMISSION\_GRANTED*) {  
 val values = ContentValues()  
 values.put(BlockedNumberContract.BlockedNumbers.*COLUMN\_ORIGINAL\_NUMBER*, number)  
 *contentResolver*.insert(BlockedNumberContract.BlockedNumbers.*CONTENT\_URI*, values)  
 } else {  
 // Handle the lack of permission here.  
 }  
 }  
  
 fun removeBlockedNumber(context: Context, phoneNumber: String): Boolean {  
 val contentResolver: ContentResolver = context.*contentResolver* // Build the selection clause to find the blocked number  
 val selection = "${BlockedNumberContract.BlockedNumbers.*COLUMN\_ORIGINAL\_NUMBER*} = ?"  
 val selectionArgs = *arrayOf*(phoneNumber)  
  
 // Attempt to delete the number from the blocked list  
 val rowsDeleted = contentResolver.delete(  
 BlockedNumberContract.BlockedNumbers.*CONTENT\_URI*,  
 selection,  
 selectionArgs  
 )  
  
 // If rowsDeleted is more than 0, the operation was successful  
 return rowsDeleted > 0  
 }  
 fun getBlockedNumbers(context: Context): ArrayList<String> {  
 val blockedNumbersList = ArrayList<String>()  
  
 // Requires permission: Manifest.permission.READ\_BLOCKED\_NUMBERS  
 val cursor: Cursor? = context.*contentResolver*.query(  
 BlockedNumberContract.BlockedNumbers.*CONTENT\_URI*,  
 *arrayOf*(BlockedNumberContract.BlockedNumbers.*COLUMN\_ID*, BlockedNumberContract.BlockedNumbers.*COLUMN\_ORIGINAL\_NUMBER*),  
 null,  
 null,  
 null  
 )  
  
 cursor?.*use* **{** val numberIndex = **it**.getColumnIndex(BlockedNumberContract.BlockedNumbers.*COLUMN\_ORIGINAL\_NUMBER*)  
 while (**it**.moveToNext()) {  
 val number = **it**.getString(numberIndex)  
 blockedNumbersList.add(number)  
 }  
 **}** return blockedNumbersList  
 }  
  
}

XML

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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=".screens.Blacklist"  
 android:background="#7BCCE4">  
  
 <ListView  
 android:id="@+id/blacklist\_listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignBottom="@id/blacklist\_listview"  
 android:layout\_marginBottom="0dp"  
 android:layout\_marginTop="60dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginStart="5dp"></ListView>  
  
 <Button  
 android:id="@+id/mainmenubtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="5dp"  
 android:background="#5061C1"  
 android:text="main menu"  
 android:textColor="#EFE8E8" />  
  
 <Button  
 android:id="@+id/additembtn"  
 android:layout\_width="98dp"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_alignParentEnd="true"  
  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginRight="-152dp"  
 android:background="#5061C1"  
 android:text="add number"  
 android:textColor="#EFE8E8" />  
  
</RelativeLayout>

Whitelist

Kotlin +Java

package com.example.smsfilteringapplication.screens

import android.content.Intent  
import android.os.Bundle  
import android.view.LayoutInflater  
import android.widget.AdapterView  
import android.widget.Button  
import android.widget.EditText  
import android.widget.ListView  
import android.widget.Toast  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.app.AppCompatActivity  
import androidx.lifecycle.*lifecycleScope*import com.example.smsfilteringapplication.MainActivity  
import com.example.smsfilteringapplication.R  
import com.example.smsfilteringapplication.dataclasses.StringItem  
import com.example.smsfilteringapplication.services.blacklistAdapter  
import com.example.smsfilteringapplication.MyApp  
import com.example.smsfilteringapplication.dataclasses.DatabaseDriver  
import com.example.smsfilteringapplication.dataclasses.addNumber  
import com.example.smsfilteringapplication.dataclasses.realmQueryToArrayList  
import com.example.smsfilteringapplication.dataclasses.removeNumber  
import io.realm.kotlin.UpdatePolicy  
import io.realm.kotlin.ext.query  
import kotlinx.coroutines.launch  
  
public class Whitelist : AppCompatActivity() {  
 private var arrayListOfNumbers = *arrayListOf*<String>()  
 private val type = "Whitelist"  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*whitelist*)  
 arrayListOfNumbers = *realmQueryToArrayList*(type)  
  
  
 val listView = findViewById<ListView>(R.id.*whitelist\_listview*)  
 listView.*adapter*= blacklistAdapter(this, arrayListOfNumbers)  
  
 val mainmenubutton = findViewById<Button>(R.id.*whitlist\_mainmenubtn*) // navigation button to main menu  
 mainmenubutton.setOnClickListener **{** val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 **}** val addItemButton = findViewById<Button>(R.id.*whitlist\_additembtn*)  
 addItemButton.setOnClickListener **{** //add item dialogue  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Add Number")  
  
 val inflater = LayoutInflater.from(this)  
 val dialogLayout = inflater.inflate(R.layout.*dialogue\_add\_item*, null)  
  
 val editText: EditText = dialogLayout.findViewById(R.id.*editTextItem*)  
 builder.setView(dialogLayout)  
  
 builder.setPositiveButton("Add") **{** \_, \_ **->** val newItem = editText.*text*.toString().*trim*()  
 if (newItem.*isNotEmpty*()&& newItem.*matches*(Regex("^[0-9]+$"))) {  
 //if conditions are met the item is added to the back end blacklist and the list view is updated  
 *lifecycleScope*.*launch* **{** addNumber(newItem, type)  
 arrayListOfNumbers = *realmQueryToArrayList*(type)  
 listView.*adapter* = blacklistAdapter(this@Whitelist, arrayListOfNumbers)  
 **}** } else {  
 Toast.makeText(this, "Item cannot be empty, item must be numbers only", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** builder.setNegativeButton("Cancel") **{** dialog, \_ **->** dialog.cancel()  
 **}** builder.show()  
 **}** listView.*onItemClickListener* = AdapterView.OnItemClickListener **{** parent, view, position, id **->** // Reading the text content of the clicked TextView  
 //val textContent = findViewById<TextView>(R.id.item\_phone\_number).text.toString()  
  
 // Set the message and title for the dialog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Remove Number")  
 builder.setMessage("Do you want to complete this action?")  
  
 // Add a Confirm button and its logic  
 builder.setPositiveButton("Confirm") **{** dialog, which **->** // Perform actions after confirmation here  
 val numToRemove = arrayListOfNumbers[position]  
 *lifecycleScope*.*launch* **{** removeNumber(numToRemove, type)  
 arrayListOfNumbers = *realmQueryToArrayList*(type)  
 listView.*adapter* = blacklistAdapter(this@Whitelist, arrayListOfNumbers)  
 **}** //arrayListOfNumbers.removeAt(position)  
 //listView.adapter= blacklistAdapter(this,arrayListOfNumbers)  
 **}** // Add a Cancel button and its logic  
 builder.setNegativeButton("Cancel") **{** dialog, which **->  
  
 }** // set the dialog to not close when the user touches outside of it  
 builder.setCancelable(false)  
  
 // Create and show the AlertDialog  
 val dialog = builder.create()  
 dialog.show()  
  
  
  
 **}** }

XML

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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=".screens.Whitelist"  
 android:background="#7BCCE4">  
  
 <ListView  
 android:id="@+id/whitelist\_listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignBottom="@id/whitelist\_listview"  
 android:layout\_marginBottom="0dp"  
 android:layout\_marginTop="60dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginStart="5dp"></ListView>  
  
 <Button  
 android:id="@+id/whitlist\_mainmenubtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"

android:layout\_marginStart="5dp"  
 android:layout\_marginTop="5dp"  
 android:background="#5061C1"  
 android:text="main menu"  
 android:textColor="#EFE8E8" />

<Button  
 android:id="@+id/whitlist\_additembtn"  
 android:layout\_width="107dp"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_alignParentEnd="true"  
  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginRight="-152dp"  
 android:background="#5061C1"  
 android:text="add number"  
 android:textAlignment="center"  
 android:textColor="#EFE8E8" />

</RelativeLayout>

Messages

Kotlin + Java

package com.example.smsfilteringapplication.screens  
  
import android.content.ContentResolver  
import android.content.ContentValues  
import android.content.Context  
import android.content.Intent  
import android.content.pm.PackageManager  
import android.database.Cursor  
import android.net.Uri  
import android.os.Bundle  
import android.provider.BlockedNumberContract  
import android.view.LayoutInflater  
import android.widget.AdapterView  
import android.widget.Button  
import android.widget.EditText  
import android.widget.ListView  
import android.widget.TextView  
import android.widget.Toast  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.content.ContextCompat  
import com.example.smsfilteringapplication.MainActivity  
import com.example.smsfilteringapplication.R  
import com.example.smsfilteringapplication.services.blacklistAdapter  
import com.example.smsfilteringapplication.services.smsviewadapter

public class Messagereporting : AppCompatActivity() {  
 val sms\_id\_list = *arrayListOf*<String>()  
 private val fromlist = *arrayListOf*<String>()  
 val bodylist = *arrayListOf*<String>()  
 private val smsbodyList = *arrayListOf*<String>()  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*messagereporting*)  
  
 val listView = findViewById<ListView>(R.id.*message\_reporting\_listview*)  
 readSms()  
 listView.*adapter*= smsviewadapter(this,fromlist,smsbodyList)  
  
 val mainmenubutton = findViewById<Button>(R.id.*message\_reporting\_mainmenubtn*) // navigation button to main menu  
 mainmenubutton.setOnClickListener **{** val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 **}** val addItemButton = findViewById<Button>(R.id.*message\_reporting\_additembtn*)  
 addItemButton.setOnClickListener **{** smsbodyList.clear()  
 fromlist.clear()  
 readSms()  
 listView.*adapter*= smsviewadapter(this,fromlist,smsbodyList)  
 **}** listView.*onItemClickListener* = AdapterView.OnItemClickListener **{** parent, view, position, id **->** // Set the message and title for the dialog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Flag Message")  
 builder.setMessage("This will delete the message and add the senders number to the blacklist")  
  
 // Add a Confirm button and its logic  
 builder.setPositiveButton("Confirm") **{** dialog, which **->** // Perform actions after confirmation here  
 //insert logic to add the item to the sms flagged database  
 // the app will then delete the sms from the phone as it has been flagged as spam  
 addNumberToBlockedList(fromlist.get(position))  
 deleteSmsById(this, sms\_id\_list.get(position).*toLong*())  
 smsbodyList.clear()  
 fromlist.clear()  
 sms\_id\_list.clear()  
 readSms()  
 listView.*adapter*= smsviewadapter(this,fromlist,smsbodyList)  
 **}** // Add a Cancel button and its logic  
 builder.setNegativeButton("Cancel") **{** dialog, which **->  
  
 }** // set the dialog to not close when the user touches outside of it  
 builder.setCancelable(false)  
  
 // Create and show the AlertDialog  
 val dialog = builder.create()  
 dialog.show()  
  
  
  
 **}** }  
  
 fun deleteSmsById(context: Context, smsId: Long) {  
 val contentResolver: ContentResolver = context.*contentResolver* val uri = Uri.parse("content://sms/")  
  
 try {  
 // Attempt to delete the SMS  
 val rowsDeleted = contentResolver.delete(Uri.withAppendedPath(uri, "$smsId"), null, null)  
  
 if (rowsDeleted > 0) {  
 *println*("SMS with ID $smsId deleted successfully.")  
 } else {  
 *println*("No SMS found with ID $smsId.")  
 }  
 } catch (e: Exception) {  
 *println*("Error deleting SMS: ${e.message}")  
 }  
 }  
  
 private fun readSms() {  
 val cursor: Cursor? = *contentResolver*.query(Uri.parse("content://sms/inbox"), null, null, null, null)  
 if (cursor?.moveToFirst() == true) { // must check the result to prevent exception  
 do {  
 val msgData = cursor.getString(cursor.getColumnIndexOrThrow("body"))  
 val senderData = cursor.getString(cursor.getColumnIndexOrThrow("address"))  
 val smsidlocal = cursor.getString(cursor.getColumnIndexOrThrow("\_id"))  
 smsbodyList.add(msgData)  
 fromlist.add(senderData)  
 sms\_id\_list.add(smsidlocal)  
  
 } while (cursor.moveToNext())  
 } else {  
 // empty inbox  
 }  
 cursor?.close()  
 }  
 fun addNumberToBlockedList(number: String) {  
 if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.*WRITE\_CONTACTS*) == PackageManager.*PERMISSION\_GRANTED*) {  
 val values = ContentValues()  
 values.put(BlockedNumberContract.BlockedNumbers.*COLUMN\_ORIGINAL\_NUMBER*, number)  
 *contentResolver*.insert(BlockedNumberContract.BlockedNumbers.*CONTENT\_URI*, values)  
 } else {  
 // Handle the lack of permission here.  
 }  
 }  
}

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"  
 android:background="#7BCCE4">

<TextView  
 android:id="@+id/from\_label"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="24dp"  
 android:text="From : "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/from\_data"  
 android:layout\_width="166dp"  
 android:layout\_height="27dp"  
 android:layout\_marginStart="92dp"  
 android:layout\_marginTop="24dp"  
 android:text="test"  
 android:textColor="#000000"  
 android:textColorHighlight="#000000"  
 android:textColorHint="#000000"  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toEndOf="@+id/from\_label"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/body\_label"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="40dp"  
 android:text="Body : "  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/from\_label" />

<TextView  
 android:id="@+id/body\_data"  
 android:layout\_width="192dp"  
 android:layout\_height="129dp"  
 android:layout\_marginStart="92dp"  
 android:layout\_marginTop="40dp"  
 android:text="body"  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toEndOf="@+id/body\_label"  
 app:layout\_constraintTop\_toBottomOf="@+id/from\_data" />  
</androidx.constraintlayout.widget.ConstraintLayout>

Keywords/Phrases

Kotlin +Java

package com.example.smsfilteringapplication.screens  
  
import android.content.Intent  
import android.os.Bundle  
import android.view.LayoutInflater  
import android.widget.AdapterView  
import android.widget.Button  
import android.widget.EditText  
import android.widget.ListView  
import android.widget.TextView  
import android.widget.Toast  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.app.AppCompatActivity  
import androidx.lifecycle.*lifecycleScope*import com.example.smsfilteringapplication.MainActivity  
import com.example.smsfilteringapplication.R  
import com.example.smsfilteringapplication.services.blacklistAdapter  
import com.example.smsfilteringapplication.dataclasses.DatabaseDriver  
import com.example.smsfilteringapplication.dataclasses.addNumber  
import com.example.smsfilteringapplication.dataclasses.realmQueryToArrayList  
import com.example.smsfilteringapplication.dataclasses.removeNumber  
import kotlinx.coroutines.launch

class KeywordManager : AppCompatActivity() {  
 private var keyWordList = *arrayListOf*<String>()  
 private val type = "KeyWord"  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*keywordmanager*)  
 keyWordList = *realmQueryToArrayList*(type)  
 val listView = findViewById<ListView>(R.id.*keyword\_listview*)  
 listView.*adapter*= blacklistAdapter(this,keyWordList)  
  
 val mainmenubutton = findViewById<Button>(R.id.*keyword\_mainmenubtn*) // navigation button to main menu  
 mainmenubutton.setOnClickListener **{** val intent = Intent(this, MainActivity::class.*java*)  
 startActivity(intent)  
 **}** val addItemButton = findViewById<Button>(R.id.*keyword\_additembtn*)  
 addItemButton.setOnClickListener **{** //add item dialogue  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Add Keyword")  
  
 val inflater = LayoutInflater.from(this)  
 val dialogLayout = inflater.inflate(R.layout.*dialogue\_add\_item*, null)  
  
 val editText: EditText = dialogLayout.findViewById(R.id.*editTextItem*)  
 builder.setView(dialogLayout)  
  
 builder.setPositiveButton("Add") **{** \_, \_ **->** val newItem = editText.*text*.toString().*trim*()  
 if (newItem.*isNotEmpty*()) {  
 //if conditions are met the item is added to the back end blacklist and the list view is updated  
 *lifecycleScope*.*launch* **{** addNumber(newItem, type)  
 keyWordList = *realmQueryToArrayList*(type)  
 listView.*adapter* = blacklistAdapter(this@KeywordManager, keyWordList)  
 **}** } else {  
 Toast.makeText(this, "Item cannot be empty", Toast.*LENGTH\_SHORT*).show()  
 }  
 **}** builder.setNegativeButton("Cancel") **{** dialog, \_ **->** dialog.cancel()  
 **}** builder.show()  
 **}** listView.*onItemClickListener* = AdapterView.OnItemClickListener **{** parent, view, position, id **->** // Reading the text content of the clicked TextView  
 val textContent = findViewById<TextView>(R.id.*item\_phone\_number*).*text*.toString()  
  
 // Set the message and title for the dialog  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Remove Keyword")  
 builder.setMessage("Do you want to complete this action?")  
  
 // Add a Confirm button and its logic  
 builder.setPositiveButton("Confirm") **{** dialog, which **->** // Perform actions after confirmation here  
 val numToRemove = keyWordList[position]  
 *lifecycleScope*.*launch* **{** removeNumber(numToRemove, type)  
 keyWordList = *realmQueryToArrayList*(type)  
 listView.*adapter* = blacklistAdapter(this@KeywordManager, keyWordList)  
 **}  
 }** // Add a Cancel button and its logic  
 builder.setNegativeButton("Cancel") **{** dialog, which **->  
  
 }** // set the dialog to not close when the user touches outside of it  
 builder.setCancelable(false)  
  
 // Create and show the AlertDialog  
 val dialog = builder.create()  
 dialog.show()  
  
  
  
 **}** }  
  
  
 }

XML

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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=".screens.KeywordManager"  
 android:background="#7BCCE4">  
  
 <ListView  
 android:id="@+id/keyword\_listview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignBottom="@id/keyword\_listview"  
 android:layout\_marginBottom="0dp"  
 android:layout\_marginTop="60dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginStart="5dp"></ListView>  
  
 <Button  
 android:id="@+id/keyword\_mainmenubtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="5dp"  
 android:background="#5061C1"  
 android:text="main menu"  
 android:textColor="#EFE8E8" />

<Button  
 android:id="@+id/keyword\_additembtn"  
 android:layout\_width="108dp"  
 android:layout\_height="wrap\_content"  
  
 android:layout\_alignParentEnd="true"  
  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginRight="-152dp"  
 android:background="#5061C1"  
 android:text="Add Keyword"  
 android:textColor="#EFE8E8" />  
  
</RelativeLayout>