Android Manifest

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:dist="http://schemas.android.com/apk/distribution"  
 package="capstone.com.birdclassify">  
  
 <dist:module dist:instant="true" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/birdlogo"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/birdlogo"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme">  
 <activity  
 android:name=".List12"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List11"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List10"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List9"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List8"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List7"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List6"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List5"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List4"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List3"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List2"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".List1"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".ListActivity"  
 android:exported="false"  
 android:screenOrientation="portrait"/>  
 <activity  
 android:name=".MainActivity"  
 android:exported="false"  
 android:screenOrientation="portrait">  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
 <activity  
 android:name=".SplashScreen"  
 android:exported="true"  
 android:screenOrientation="portrait">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

SplashScreen

package capstone.com.birdclassify  
  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.os.Handler  
import android.os.Looper  
import android.view.View  
import android.view.Window  
  
class SplashScreen : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.FEATURE\_NO\_TITLE)  
 window.decorView.systemUiVisibility =  
 View.SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION or View.SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE or View.SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION or View.SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN or View.SYSTEM\_UI\_FLAG\_FULLSCREEN or View.SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY  
 setContentView(R.layout.activity\_splash\_screen)  
  
 val handler = Handler(Looper.getMainLooper())  
 handler.postDelayed(**{** val intent = Intent (this, MainActivity::class.java)  
 startActivity(intent)  
 finish()  
 **}**,2000)  
 }  
}

MainActivity

package capstone.com.birdclassify  
  
import android.os.Bundle  
import android.widget.Toast  
import android.app.Activity  
import android.app.AlertDialog  
import android.content.Intent  
import android.graphics.Bitmap  
import android.graphics.BitmapFactory  
import android.graphics.Matrix  
import android.os.Build  
import android.provider.MediaStore  
import android.transition.Slide  
import android.transition.TransitionManager  
import android.view.Gravity  
import android.view.View  
import android.view.Window  
import android.widget.Button  
import android.widget.ImageView  
import android.widget.LinearLayout  
import android.widget.RelativeLayout  
import android.widget.TextView  
import androidx.annotation.RequiresApi  
import androidx.appcompat.app.AppCompatActivity  
import java.io.IOException  
import kotlin.system.exitProcess  
  
class MainActivity : AppCompatActivity() {  
 private lateinit var mClassifier: Classifier  
 private lateinit var mBitmap: Bitmap  
  
 private val mCameraRequestCode = 0  
 private val mGalleryRequestCode = 2  
  
 private val mInputSize = 224  
 private val mModelPath = "model.tflite"  
 private val mLabelPath = "labels.txt"  
 private val mSamplePath = "logonew.png"  
  
 lateinit var builder: AlertDialog.Builder  
 lateinit var mPhoto:ImageView  
 lateinit var mCameraButton:Button  
 lateinit var mGalleryButton:Button  
 lateinit var mDetectButton:Button  
 lateinit var mResult:TextView  
 lateinit var mResultTextView:TextView  
 lateinit var mainLay:RelativeLayout  
 lateinit var mResultName: TextView  
 lateinit var defBtn:TextView  
 lateinit var homeBtn:ImageView  
 lateinit var outBtn:ImageView  
  
 @RequiresApi(Build.VERSION\_CODES.O)  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.FEATURE\_NO\_TITLE)  
 window.decorView.systemUiVisibility =  
 View.SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION or View.SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE or View.SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION or View.SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN or View.SYSTEM\_UI\_FLAG\_FULLSCREEN or View.SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY  
 setContentView(R.layout.activity\_main)  
 mClassifier = Classifier(assets, mModelPath, mLabelPath, mInputSize)  
  
 mPhoto=findViewById(R.id.mPhoto);  
  
 resources.assets.open(mSamplePath).use **{** mBitmap = BitmapFactory.decodeStream(it)  
 mBitmap = Bitmap.createScaledBitmap(mBitmap, mInputSize, mInputSize, true)  
 mPhoto.setImageBitmap(mBitmap)  
 **}** mCameraButton=findViewById(R.id.mCameraButton)  
 mGalleryButton=findViewById(R.id.mGalleryButton)  
 mDetectButton=findViewById(R.id.mDetectButton)  
 mResult=findViewById(R.id.mResult)  
 mResultTextView=findViewById(R.id.mResultTextView)  
 homeBtn=findViewById(R.id.homeBtn)  
 defBtn=findViewById(R.id.defBtn)  
 outBtn=findViewById(R.id.outBtn)  
 mainLay=findViewById(R.id.mainLay)  
 mResultName=findViewById(R.id.mResultName)  
  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 exitProcess(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** Toast.makeText(this,"You're viewing the page.",Toast.LENGTH\_SHORT).show()  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.java))  
 **}**)  
  
 mCameraButton.setOnClickListener **{** val callCameraIntent = Intent(MediaStore.ACTION\_IMAGE\_CAPTURE)  
 startActivityForResult(callCameraIntent, mCameraRequestCode)  
 mResult.text = ""  
 mResultTextView.text = ""  
 mResultName.text=""  
 **}** mGalleryButton.setOnClickListener **{** val callGalleryIntent = Intent(Intent.ACTION\_PICK)  
 callGalleryIntent.type = "image/\*"  
 startActivityForResult(callGalleryIntent, mGalleryRequestCode)  
 **}** mDetectButton.setOnClickListener **{** val results = mClassifier.recognizeImage(mBitmap).firstOrNull()  
 mResult.text = results?.title  
 mResultTextView.text = "Confidence:" + results?.confidence  
  
 if (mResultTextView.text == "Confidence:null") {  
 mResult.text = "Undetected"  
 }  
  
 if (mResult.text == "Apo Myna Bird"){  
 mResultName.setText("Basilornis mirandus")  
 }else if (mResult.text == "Barred Rail Bird"){  
 mResultName.setText("Gallirallus torquatus")  
 }else if (mResult.text == "Blue Headed Fantail"){  
 mResultName.setText("Rhipidura cyaniceps")  
 }else if (mResult.text == "Blue Rock Thrush Bird"){  
 mResultName.setText("Monticola solitarius")  
 }else if (mResult.text == "Budgie"){  
 mResultName.setText("Melopsittacus undulatus")  
 }else if (mResult.text == "Blue Naped Parrot"){  
 mResultName.setText("Tanygnathus lucionensis")  
 }else if (mResult.text == "Bukidnon Woodcock Bird"){  
 mResultName.setText("Scolopax bukidnonensis")  
 }else if (mResult.text == "Junglefowl Chicken"){  
 mResultName.setText("Gallus gallus")  
 }else if (mResult.text == "Coleto Bird"){  
 mResultName.setText("Sarcops calvus")  
 }else if (mResult.text == "Coppersmith Bird"){  
 mResultName.setText("Megalaima haemacephala")  
 }else if (mResult.text == "Dove"){  
 mResultName.setText(" Columbidae")  
 }else if (mResult.text == "Black Naped Oriole"){  
 mResultName.setText("Oriolus chinensis")  
 }  
  
 **}** }  
  
 override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
 super.onActivityResult(requestCode, resultCode, data)  
 if (requestCode == mCameraRequestCode) {  
  
 if (resultCode == Activity.RESULT\_OK && data != null) {  
 mBitmap = data.extras!!.get("data") as Bitmap  
 mBitmap = scaleImage(mBitmap)  
 val toast = Toast.makeText(  
 this,  
 ("Image crop to: w= ${mBitmap.width} h= ${mBitmap.height}"),  
 Toast.LENGTH\_LONG  
 )  
 toast.setGravity(Gravity.BOTTOM, 0, 20)  
 toast.show()  
 mPhoto.setImageBitmap(mBitmap)  
 mResultTextView.text = "Your photo image set now."  
 } else {  
 Toast.makeText(this, "Camera cancel..", Toast.LENGTH\_LONG).show()  
 }  
 } else if (requestCode == mGalleryRequestCode) {  
 if (data != null) {  
 val uri = data.data  
  
 try {  
 mBitmap = MediaStore.Images.Media.getBitmap(this.contentResolver, uri)  
 } catch (e: IOException) {  
 e.printStackTrace()  
 }  
  
 println("Success!!!")  
 mBitmap = scaleImage(mBitmap)  
 mPhoto.setImageBitmap(mBitmap)  
  
 }  
 } else {  
 Toast.makeText(this, "Unrecognized request code", Toast.LENGTH\_LONG).show()  
  
 }  
 }  
  
  
 fun scaleImage(bitmap: Bitmap?): Bitmap {  
 val orignalWidth = bitmap!!.width  
 val originalHeight = bitmap.height  
 val scaleWidth = mInputSize.toFloat() / orignalWidth  
 val scaleHeight = mInputSize.toFloat() / originalHeight  
 val matrix = Matrix()  
 matrix.postScale(scaleWidth, scaleHeight)  
 return Bitmap.createBitmap(bitmap, 0, 0, orignalWidth, originalHeight, matrix, true)  
 }  
  
 override fun onBackPressed() {  
 val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 exitProcess(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 }  
  
}

ListActivity

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import android.widget.Toast  
import androidx.cardview.widget.CardView  
import kotlin.system.exitProcess  
  
class ListActivity : AppCompatActivity() {  
  
 lateinit var outBtn:ImageView  
 lateinit var homeBtn:ImageView  
 lateinit var defBtn:TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var c1:CardView  
 lateinit var c2:CardView  
 lateinit var c3:CardView  
 lateinit var c4:CardView  
 lateinit var c5:CardView  
 lateinit var c6:CardView  
 lateinit var c7:CardView  
 lateinit var c8:CardView  
 lateinit var c9:CardView  
 lateinit var c10:CardView  
 lateinit var c11:CardView  
 lateinit var c12:CardView  
  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 c1=findViewById(R.id.*c1*)  
 c2=findViewById(R.id.*c2*)  
 c3=findViewById(R.id.*c3*)  
 c4=findViewById(R.id.*c4*)  
 c5=findViewById(R.id.*c5*)  
 c6=findViewById(R.id.*c6*)  
 c7=findViewById(R.id.*c7*)  
 c8=findViewById(R.id.*c8*)  
 c9=findViewById(R.id.*c9*)  
 c10=findViewById(R.id.*c10*)  
 c11=findViewById(R.id.*c11*)  
 c12=findViewById(R.id.*c12*)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** Toast.makeText(this,"You're viewing the page.",Toast.*LENGTH\_SHORT*).show()  
 **}**)  
 c1.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List1::class.*java*))  
 **}**)  
 c2.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List2::class.*java*))  
 **}**)  
 c3.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List3::class.*java*))  
 **}**)  
 c4.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List4::class.*java*))  
 **}**)  
 c5.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List5::class.*java*))  
 **}**)  
 c6.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List6::class.*java*))  
 **}**)  
 c7.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List7::class.*java*))  
 **}**)  
 c8.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List8::class.*java*))  
 **}**)  
 c9.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List10::class.*java*))  
 **}**)  
 c10.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List11::class.*java*))  
 **}**)  
 c11.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List12::class.*java*))  
 **}**)  
 c12.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,List9::class.*java*))  
 **}**)  
  
 }  
}

Classifier

package capstone.com.birdclassify  
  
import android.content.res.AssetManager  
import android.graphics.Bitmap  
import android.util.Log  
import org.tensorflow.lite.Interpreter  
import java.io.FileInputStream  
import java.nio.ByteBuffer  
import java.nio.ByteOrder  
import java.nio.MappedByteBuffer  
import java.nio.channels.FileChannel  
import java.util.\*  
  
class Classifier(assetManager: AssetManager, modelPath: String, labelPath: String, inputSize: Int) {  
 private var INTERPRETER: Interpreter  
 private var LABEL\_LIST: List<String>  
 private val INPUT\_SIZE: Int = inputSize  
 private val PIXEL\_SIZE: Int = 3  
 private val IMAGE\_MEAN = 0  
 private val IMAGE\_STD = 255.0f  
 private val MAX\_RESULTS = 3  
 private val THRESHOLD = 0.4f  
  
 data class Recognition(  
 var id: String = "",  
 var title: String = "",  
 var confidence: Float = 0F  
 ) {  
 override fun toString(): String {  
 return "Title = $title, Confidence = $confidence)"  
 }  
 }  
  
 init {  
 INTERPRETER = Interpreter(loadModelFile(assetManager, modelPath))  
 LABEL\_LIST = loadLabelList(assetManager, labelPath)  
 }  
  
 private fun loadModelFile(assetManager: AssetManager, modelPath: String): MappedByteBuffer {  
 val fileDescriptor = assetManager.openFd(modelPath)  
 val inputStream = FileInputStream(fileDescriptor.*fileDescriptor*)  
 val fileChannel = inputStream.*channel* val startOffset = fileDescriptor.*startOffset* val declaredLength = fileDescriptor.*declaredLength* return fileChannel.map(FileChannel.MapMode.*READ\_ONLY*, startOffset, declaredLength)  
 }  
  
 private fun loadLabelList(assetManager: AssetManager, labelPath: String): List<String> {  
 return assetManager.open(labelPath).*bufferedReader*().*useLines* **{ it**.*toList*() **}** }  
  
 fun recognizeImage(bitmap: Bitmap): List<Classifier.Recognition> {  
 val scaledBitmap = Bitmap.createScaledBitmap(bitmap, INPUT\_SIZE, INPUT\_SIZE, false)  
 val byteBuffer = convertBitmapToByteBuffer(scaledBitmap)  
 val result = Array(1) **{** FloatArray(LABEL\_LIST.size) **}** INTERPRETER.run(byteBuffer, result)  
 return getSortedResult(result)  
 }  
  
  
  
 private fun convertBitmapToByteBuffer(bitmap: Bitmap): ByteBuffer {  
 val byteBuffer = ByteBuffer.allocateDirect(4 \* INPUT\_SIZE \* INPUT\_SIZE \* PIXEL\_SIZE)  
 byteBuffer.order(ByteOrder.nativeOrder())  
 val intValues = IntArray(INPUT\_SIZE \* INPUT\_SIZE)  
  
 bitmap.getPixels(intValues, 0, bitmap.*width*, 0, 0, bitmap.*width*, bitmap.*height*)  
 var pixel = 0  
 for (i in 0 *until* INPUT\_SIZE) {  
 for (j in 0 *until* INPUT\_SIZE) {  
 val `val` = intValues[pixel++]  
  
 byteBuffer.putFloat((((`val`.shr(16) and 0xFF) - IMAGE\_MEAN) / IMAGE\_STD))  
 byteBuffer.putFloat((((`val`.shr(8) and 0xFF) - IMAGE\_MEAN) / IMAGE\_STD))  
 byteBuffer.putFloat((((`val` and 0xFF) - IMAGE\_MEAN) / IMAGE\_STD))  
 }  
 }  
 return byteBuffer  
 }  
  
  
 private fun getSortedResult(labelProbArray: Array<FloatArray>): List<Classifier.Recognition> {  
 Log.d("Classifier", "List Size:(%d, %d, %d)".*format*(labelProbArray.size,labelProbArray[0].size,LABEL\_LIST.size))  
  
 val pq = PriorityQueue(  
 MAX\_RESULTS,  
 *Comparator*<Classifier.Recognition> **{** (\_, \_, confidence1), (\_, \_, confidence2)  
 **->** java.lang.Float.compare(confidence1, confidence2) \* -1  
 **}**)  
  
 for (i in LABEL\_LIST.*indices*) {  
 val confidence = labelProbArray[0][i]  
 if (confidence >= THRESHOLD) {  
 pq.add(Classifier.Recognition("" + i,  
 if (LABEL\_LIST.size > i) LABEL\_LIST[i] else "Unknown", confidence)  
 )  
 }  
 }  
 Log.d("Classifier", "pqsize:(%d)".*format*(pq.size))  
  
 val recognitions = ArrayList<Classifier.Recognition>()  
 val recognitionsSize = Math.min(pq.size, MAX\_RESULTS)  
 for (i in 0 *until* recognitionsSize) {  
 recognitions.add(pq.poll())  
 }  
 return recognitions  
 }  
  
}

List1

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import android.widget.Toast  
import kotlin.system.exitProcess  
  
class List1 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list1*)  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List2

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List2 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list2*)  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List3

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List3 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list3*)  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List4

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List4 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list4*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List5

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List5 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list5*)  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List6

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List6 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list6*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List7

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List7 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list7*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List8

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List8 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list8*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List9

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List9 : AppCompatActivity() {  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list9*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List10

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List10 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list10*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List11

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List11 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list11*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}

List12

package capstone.com.birdclassify  
  
import android.app.AlertDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.view.Window  
import android.widget.ImageView  
import android.widget.TextView  
import kotlin.system.exitProcess  
  
class List12 : AppCompatActivity() {  
  
 lateinit var outBtn: ImageView  
 lateinit var homeBtn: ImageView  
 lateinit var defBtn: TextView  
 lateinit var builder: AlertDialog.Builder  
 lateinit var bckBtn:ImageView  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 requestWindowFeature(Window.*FEATURE\_NO\_TITLE*)  
 *window*.*decorView*.*systemUiVisibility* =  
 View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION* or View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_FULLSCREEN* or View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY* setContentView(R.layout.*activity\_list12*)  
  
 outBtn=findViewById(R.id.*outBtn*)  
 homeBtn=findViewById(R.id.*homeBtn*)  
 defBtn=findViewById(R.id.*defBtn*)  
 bckBtn=findViewById(R.id.*bckBtn*)  
  
 bckBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
  
 outBtn.setOnClickListener(View.OnClickListener **{** val builder = AlertDialog.Builder(this)  
  
 builder.setTitle("Confirm Exit").setMessage("Do you really want to Exit?(Y/N)")  
 .setPositiveButton(  
 "Yes"  
 ) **{** dialogInterface, i **->** finishAffinity()  
 *exitProcess*(0)  
 **}**.setNegativeButton(  
 "No"  
 ) **{** dialogInterface, i **->** dialogInterface.dismiss() **}**.setCancelable(false)  
  
 .show()  
 **}**)  
  
 homeBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,MainActivity::class.*java*))  
 **}**)  
  
 defBtn.setOnClickListener(View.OnClickListener **{** startActivity(Intent(this,ListActivity::class.*java*))  
 **}**)  
 }  
}