เริ่มต้น Coding สร้าง Mobile App อย่างมืออาชีพด้วย

Kotlin และ Android Studio

ฝึกฝนเพื่อก้าวสู่การเป็นนักพัฒนาแอปพลิเคชันบนแพลตฟอร์มที่มีผู้ใช้งานหลายพันล้านคนทั่วโลก

- ฝึกฝนการ Coding ด้วยภาษา Kotlin ซึ่งเป็น Official Android Development Language
- เรียนรู้การสร้าง App ด้วย Android Studio ตั้งแต่เริ่มต้นจนเผยแพร่ใน Play Store
- เหมาะสำหรับนักเรียน/นักศึกษา หรือผู้สนใจฝึกฝนการพัฒนา Mobile App

ไฟล์ตัวอย่างภายในเล่ม
https://serazu.com/
9786164872561

ศุภชัย สมพานิช

INFOPRESS

สารบัญ

บทที่ 1 เตรียมความพร้อมก่อนพัฒนา Android Apps 1

แนวทางการนำเสนอของหนังสือเล่มนี้ 1
รูปแบบของโปรแกรม Android Studio 1
การดาวน์โหลดและติดตั้งโปรแกรม Android Studio เวอร์ชัน Stable 3
การดาวน์โหลดและติดตั้งโปรแกรม Android Studio เวอร์ชัน Preview 7
การติดตั้ง Android SDK เพิ่มเติม 9
การสร้างโปรเจ็กต์ใน Android Studio แบบ Empty Activity 15
ทำความรู้จักกับสภาพแวดล้อมของ Android Studio 18
การสร้าง Android Emulator สำหรับทดสอบโปรเจ็กต์ 20
การทดสอบโปรเจ็กต์ด้วย Android Emulator 23
การทดสอบโปรเจ็กต์ Android Apps บนเครื่องจริง (ใช้กับ Android 8.0 ขึ้นไปเท่านั้น) 24
การเปิดและปิดโปรเจ็กต์ 28
การอัพเดตโปรแกรม Android Studio ให้เป็นเวอร์ชันใหม่ 29
การอัพเดตภาษา Kotlin ให้เป็นเวอร์ชันใหม่ล่าสุด 30
วิธีการ Export โปรเจ็กต์เป็นไฟล์ Zip 31
การดาวน์โหลดและติดตั้งส่วนขยายของภาษา Kotlin (kotlin-android-extensions) 32
การรัน Android Emulator เป็นแบบหน้าต่างใน Android Studio 33

บทที่ 2 พื้นฐานการพัฒนา Android Apps 37

ทำความรู้จักกับโครงสร้างโปรเจ็กต์แบบ Empty Activity 37
ทำความรู้จักกับระบบ dependencies ใน Android Apps 42
การอัพเดตเวอร์ชันใหม่ของรายการ dependencies 44
Android Apps แรกของผู้อ่าน 44
การสร้างปุ่มกดแบบ Floating Action Button 53
การกำหนดขนาด widget 57
แนวทางการอัพเดตโปรเจ็กต์ Android Apps ยุคเก่า 59

บทที่ 3 ทำความรู้จักกับระบบ Layout ของ Android Apps 65

พื้นฐานการกำหนดตำแหน่งบน ConstraintLayout 65

การกำหนดตำแหน่งกึ่งกลาง (แนวตั้งหรือแนวนอน) ของ ConstraintLayout 69

การยกเลิกการยึดติดของ ConstraintLayout 71

การยึดติดระหว่าง widget ด้วยกันเอง 72

การระบุตำแหน่งด้วย Guideline 74

การใช้งาน Layout แบบ LinearLayout 77

บทที่ 4 พื้นฐานการสร้างส่วนแสดงผลแบบรายการ list ด้วย RecyclerView 83

พื้นฐานการสร้างรายการ list โดยอาศัย RecyclerView 83

พื้นฐานการปรับแต่งรายการ (Custom Row) ใน RecyclerView 97

พื้นฐานการทำงานกับสถานะ (State) ของแต่ละแถว 107

บทที่ 5 การประยุกต์ใช้งาน RecyclerView 111

การจัด Layout แบบแนวนอนด้วย LinearLayout Manager แบบ HORIZONTAL 111

การแสดงข้อมูลแบบตาราง Grid ด้วย GridLayout Manager 116

การโหลดไฟล์รูปภาพจากเว็บไซต์ภายนอกด้วย Glide 117

บทที่ 6 ระบบเมนูและการแจ้งเตือน 133

การสร้างระบบเมนูพื้นฐาน 133

การสร้างเมนูแบบ Context 139

การสร้างเมนูแบบ Popup 144

หน้าต่างแจ้งเตือนแบบ AlertDialog 148

การสร้าง AlertDialog แบบมีรายการ list 151

บทที่ 7 ทำงานกับส่วนแสดงผลหลายหน้าจอด้วย Fragment 155

การสร้างโปรเจ็กต์แบบ Basic Activity 155

เขียนโค้ดกับโครงสร้างโปรเจ็กต์แบบ Basic Activity 157

การนำทางแบบ Bottom Navigation 164

เขียนโค้ดกับโครงสร้างโปรเจ็กต์แบบ Bottom Navigation Activity 166

บทที่ 8 การสร้างโปรเจ็กต์ที่มีเมนูแบบ Navigation Drawer 171
การสร้างโปรเจ็กต์ที่มีเมนูแบบ Navigation Drawer 172
เขียนโค้ดกับโครงสร้างโปรเจ็กต์ที่มีเมนูแบบ Navigation Drawer 175

บทที่ 9 การทำงานกับฟังก์ชันพื้นฐานของ Mobile Devices 187
สิทธิ์การขอใช้อินเทอร์เน็ต (android.permission.INTERNET) 187
ฟังก์ชันการโทรออก (android.permission.CALL_PHONE) 191
การใช้งานกล้อง (android.permission.CAMERA) 198

บทที่ 10 การสร้างไฟล์แชร์ข้อมูลแบบ SharedPreferences 207
การจัดการข้อมูลที่อยู่ใน SharedPreferences 207

บทที่ 11 การพัฒนาแอปร่วมกับแผนที่ Google Maps 217
พื้นฐานการใช้งานแผนที่ Google Maps 217
การใช้งานแผนที่ Google Maps กับโปรเจ็กต์ที่มีอยู่เดิม 225
ฟังก์ชันพื้นฐานของแผนที่ Google Maps 227
การกำหนดชนิดของแผนที่ Google Maps 235
การค้นหาที่อยู่และปักหมุดในแผนที่ 238
การปรับแต่ง Style ให้กับแผนที่ Google Maps 246
การปรับแต่งรายละเอียดในแผนที่ 251
การใช้งานแผนที่ Google Maps โดยอาศัยคลาส Intent 254

บทที่ 12 ทำงานกับฐานข้อมูล SQLite ด้วย Room 259
Android Architecture Components คืออะไร 259
การจัดการข้อมูล CRUD ในฐานข้อมูล SQLite ด้วย Room 260

บทที่ 13 การสร้างแอนิเมชันด้วย Motion Layout 293
ข้อกำหนดขั้นต้นของการใช้งาน Motion Layout 293
พื้นฐานการสร้างแอนิเมชันแบบเคลื่อนย้ายตำแหน่ง 294
การแก้ไขคุณสมบัติ (หรือแอตทริบิวต์) ในแอนิเมชัน 299
การสร้างแอนิเมชันแบบทำงานโดยอัตโนมัติ 303

บทที่ 14 การทำระบบ Login ด้วย Google Sign-in 309

การสร้างแอปขอใช้บริการ Google Sign-in 309

บทที่ 15 Jetpack Compose 325

พื้นฐานการสร้างโปรเจ็กต์โดยอาศัย Jetpack Compose 325

ระบบธีมของ Jetpack Compose 330

รายการสี (Color.kt) 332

รูปแบบฟอนต์ (Type.kt) 334

รูปร่าง Shape (Shape.kt) 335

การจัด Layout ตามแนวนอนด้วยคลาส Row 336

การปรับตำแหน่งตามแนวนอนและแนวตั้ง 338

การจัด Layout ตามแนวตั้งด้วยคลาส Column 342

การใช้งาน ConstraintLayout แบบ Compose 344

การจัดตำแหน่ง ConstraintLayout 348

เหตุการณ์ (Event) ใน Jetpack Compose 351

บทที่ 16 การสร้างไฟล์ apk หรือ aab สำหรับใช้งานจริง 355

ไฟล์ apk (หรือ aab) คืออะไร 355

วิธีการสร้างไฟล์ apk 356

การติดตั้งไฟล์ Signed APK ในเครื่องจริง 362

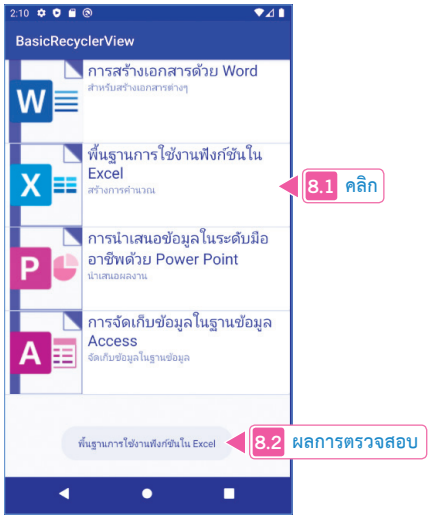
บทที่ 17 การอัพโหลดแอปขึ้น Google Play Store 367

การสมัครบัญชีนักพัฒนา Android Apps 367

ขั้นตอนการอัพโหลดไฟล์ Signed APK ไปสู่ Google Play Store 370

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        rv.layoutManager = LinearLayoutManager(this)

        val div = DividerItemDecoration(this, DividerItemDecoration.VERTICAL)
        rv.addItemDecoration(div)

        val rv_dataLists = mutableListOf<rv_data>()
        rv_dataLists.add(rv_data(R.drawable.word,"การสร้างเอกสารด้วย Word", "สำหรับสร้างเอกสารต่างๆ"))
        rv_dataLists.add(rv_data(R.drawable.excel, "พื้นฐานการใช้งานฟังก์ชันใน Excel", "สร้างการคำนวณ"))
        rv_dataLists.add(rv_data(R.drawable.powerpoint,"การนำเสนอข้อมูลในระดับมืออาชีพด้วย Power Point", "นำเสนอผลงาน"))
        rv_dataLists.add(rv_data(R.drawable.access, "การจัดเก็บข้อมูลในฐานข้อมูล Access", "จัดเก็บข้อมูลในฐานข้อมูล"))

        val adt = adt(rv_dataLists)
        rv.adapter = adt
    }
}
```

8. ท้ายที่สุด ให้ผู้อ่านทดสอบรันโปรเจ็กต์ พบว่า แต่ละแถวมีการแสดงรูปภาพและข้อความตามที่เราออกแบบไว้ในไฟล์ rv_row แล้ว ดังรูปที่ 4-20



รูปที่ 4-20 ผลการรันตัวอย่างที่ 4-2

การแสดงข้อมูลแบบตาราง Grid ด้วย GridLayout Manager

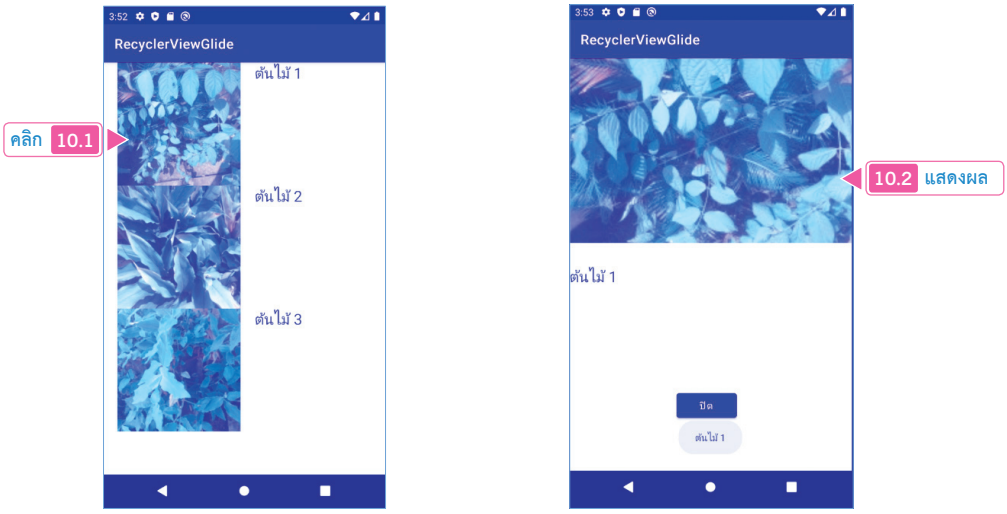
การแสดงรายการข้อมูลแบบตาราง Grid เป็นอีก 1 รูปแบบที่เราเห็นได้ในแอปต่างๆ โดยมีหลักการที่สำคัญคือ แต่ละช่องมีขนาดเท่ากันหมด

ตัวอย่างที่ 5-2 การแสดงข้อมูลแบบตาราง Grid ด้วย GridLayoutManager

ผู้เขียนนำตัวอย่างที่แล้วมาปรับปรุงใหม่ โดยการกำหนดให้ RecyclerView แสดงผลในรูปแบบตาราง Grid ด้วย GridLayoutManager ได้เลย โดยกำหนดให้แสดง 3 คอลัมน์ (GridLayoutManager(this, 3)) เพราะว่า Layout ของข้อมูลแต่ละช่องที่อยู่ในไฟล์ rv_row.xml มีขนาดตายตัวเท่ากันอยู่แล้ว

สคริปต์ Kotlin ที่ 5-2 การแสดงข้อมูลแบบตาราง Grid ด้วย GridLayoutManager (\app\java\com.example.basicrecyclerview\MainActivity.kt)

```
package com.example.basicrecyclerview

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.recyclerview.widget.DividerItemDecoration
import androidx.recyclerview.widget.GridLayoutManager
import androidx.recyclerview.widget.LinearLayoutManager
import com.example.basicrecyclerview.adapter.adt
import com.example.basicrecyclerview.model.rv_data
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        rv.layoutManager = GridLayoutManager(this, 3)

        val div = DividerItemDecoration(this, DividerItemDecoration.VERTICAL)
        rv.addItemDecoration(div)

        val rv_dataLists = mutableListOf<rv_data>()
        rv_dataLists.add(rv_data(R.drawable.word,"Word", "เอกสาร"))
        rv_dataLists.add(rv_data(R.drawable.excel,"Excel", "การคำนวณ"))
        rv_dataLists.add(rv_data(R.drawable.powerpoint,"Power Point", "นำเสนอผลงาน"))
        rv_dataLists.add(rv_data(R.drawable.access,"Access", "ข้อมูลในฐานข้อมูล"))
```



รูปที่ 5-12 ผลการรันตัวอย่างที่ 5-3

อธิบายการทำงานของโค้ด

1. ที่คลาส adt ทำหน้าที่เป็น Adapter ใส่ข้อมูลให้กับ RecyclerView ต้องการข้อมูล 3 ค่าคือ
   - **พารามิเตอร์ _context: Context** ทำหน้าที่แทนส่วนแสดงผลปัจจุบัน
   - **พารามิเตอร์ _arrName: ArrayList<String>** ทำหน้าที่เก็บรายชื่อหรือข้อความของรูปภาพ
   - **พารามิเตอร์ _arrUrl: ArrayList<String>** ทำหน้าที่เก็บพาธ URL ของไฟล์รูปภาพ

\app\java\com.example.recyclerviewglide\adapter\adt.kt

```
class adt(_context: Context, _arrName: ArrayList<String>, _arrUrl: ArrayList<String>) : RecyclerView.
Adapter<adt.ViewHolder>() {
```

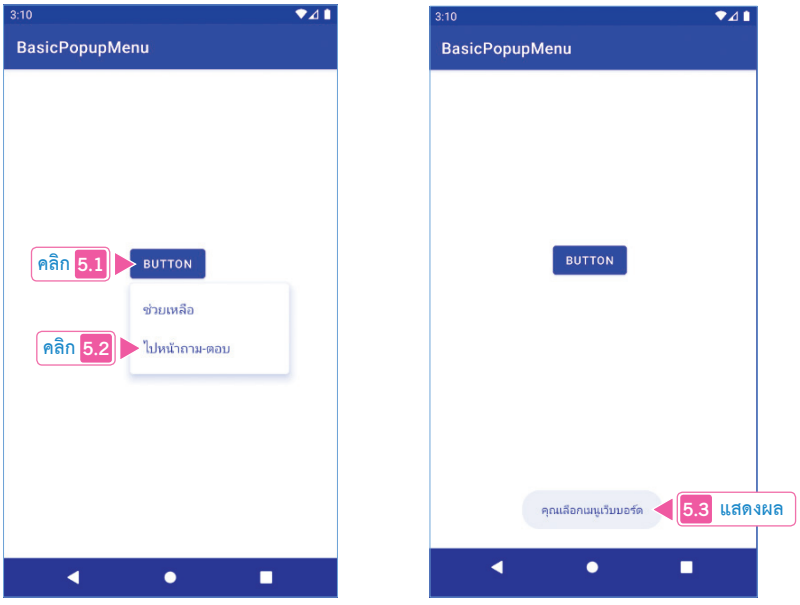
2. สร้างตัวแปรที่ชื่อว่า arrName, arrUrl และ Context รับทั้ง 3 ค่ามาเก็บไว้ก่อน

\app\java\com.example.recyclerviewglide\adapter\adt.kt

```
    private var arrName = ArrayList<String>()
    private var arrUrl = ArrayList<String>()
    private val Context: Context

    init {
        arrName = _arrName
        arrUrl = _arrUrl
        Context = _context
    }
```

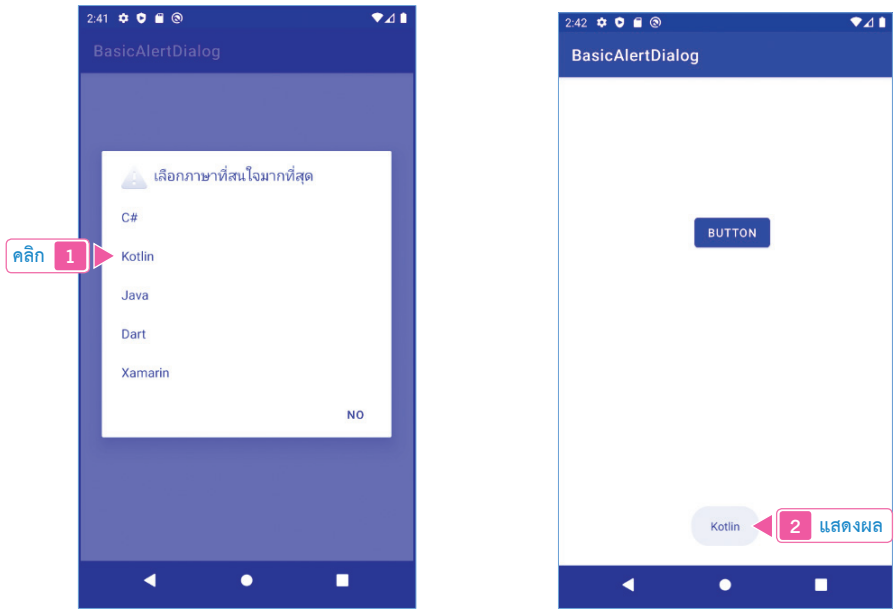
```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        cmdShow.setOnClickListener {
            val mnuPopup: PopupMenu = PopupMenu(this, cmdShow)
            mnuPopup.menuInflater.inflate(R.menu.menu_popup, mnuPopup.menu)
            mnuPopup.setOnMenuItemClickListener { item ->
                when(item.itemId) {
                    R.id.mnuHelp ->
                        Toast.makeText(this, "ผู้อ่านเลือกเมนูช่วยเหลือ", Toast.LENGTH_LONG).show()
                    R.id.mnuWebboard ->
                        Toast.makeText(this, "ผู้อ่านเลือกเมนูเว็บบอร์ด", Toast.LENGTH_LONG).show()
                }
                true
            }
            mnuPopup.show()
        }
    }
}
```

5. ท้ายที่สุด ให้ผู้อ่านทดสอบรันโปรเจ็กต์ จากนั้น คลิกปุ่มกด Button ก็จะพบกับเมนูแบบ Popup ปรากฏขึ้นมา ดังรูปที่ 6-10

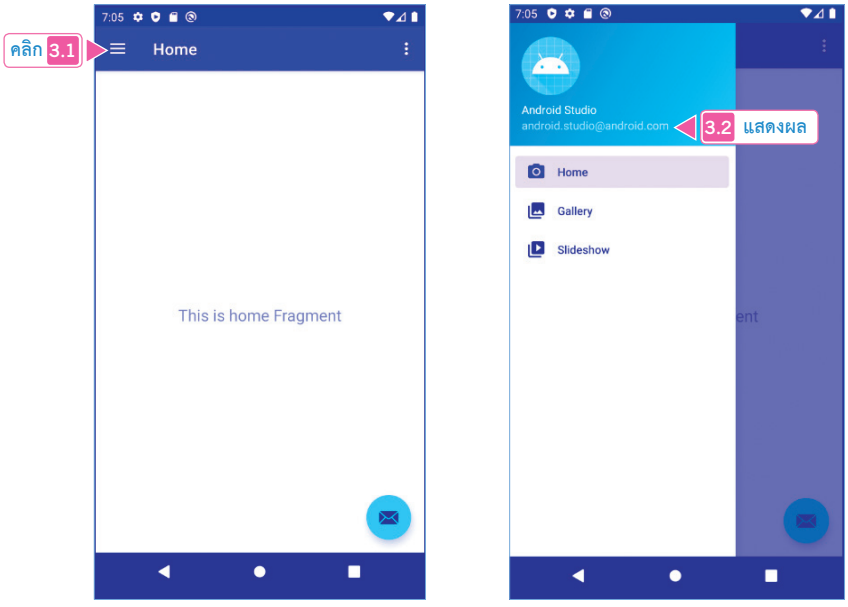


รูปที่ 6-10 แสดงเมนูแบบ Popup

```
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        cmdShow.setOnClickListener{
            val arr = arrayOf("C#", "Kotlin", "Java", "Dart", "Xamarin")
            AlertDialog.Builder(this).apply {
                setTitle("เลือกภาษาที่สนใจมากที่สุด")
                setIcon(android.R.drawable.ic_dialog_alert)
                setItems(arr) { _, selected ->
                    Toast.makeText(applicationContext, arr[selected], Toast.LENGTH_LONG).show()
                }

                setNegativeButton("No") { _, _ ->
                    Toast.makeText(applicationContext, "ผู้อ่านคลิกที่ปุ่ม No", Toast.LENGTH_LONG).
show()
                }
                show()
            }
        }
    }
}
```

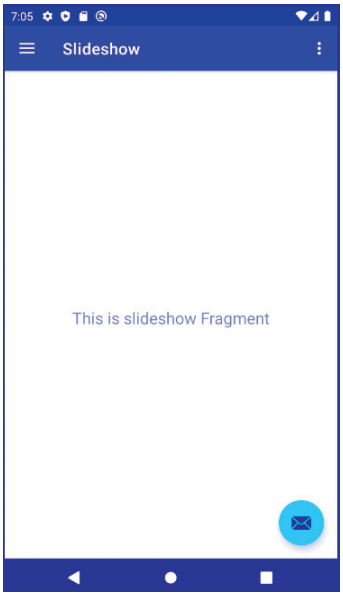


รูปที่ 6-12 แสดง AlertDialog แบบมีรายการ list

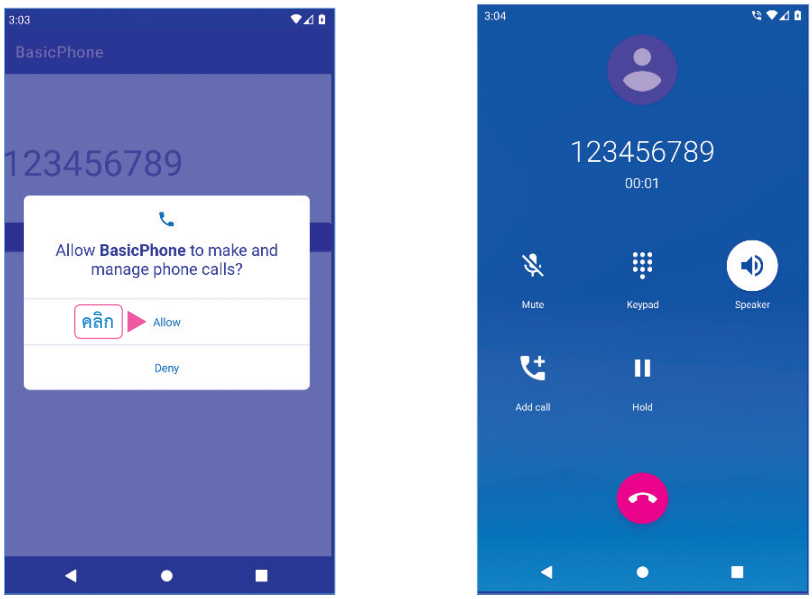


รูปที่ 8-5 แสดงแถบเมนูแบบ Navigation Drawer

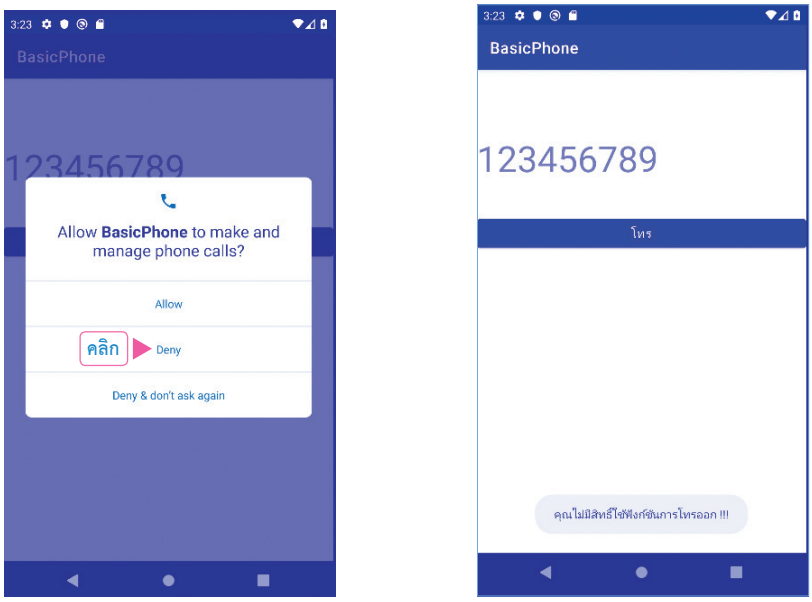
จากรูปที่ 8-5 เมื่อคลิกเลือกรายการเมนูในแถบ Navigation Drawer ก็จะสลับมาที่หน้าจอนั้นๆ เป็นส่วนแสดงผลที่เกิดมาจาก Fragment



รูปที่ 8-6 ส่วนแสดงผลแบบ Fragment



รูปที่ 9-6 กรณียินยอม (Allow) ให้โทรออกได้ตามปกติ



รูปที่ 9-7 กรณีไม่ยินยอม (Deny) ให้โทรออก

5. ท้ายที่สุด ให้ผู้อ่านลองไปดูสิทธิ์ขอโทรออกของแอป โดยการเลือกการ Settings > Apps > Phone permission พบว่า ผู้อ่านสามารถเปิด-ปิดฟีเจอร์นี้ได้ตลอดเวลา ดังรูปที่ 9-8

การใช้งานกล้อง (android.permission.CAMERA)

กล้อง (Camera) ถือเป็น Hardware พื้นฐานที่มีการใช้งานมากที่สุด ในกรณีที่เราต้องการพัฒนาแอปที่ต้องเรียกใช้กล้อง จึงเป็นอีก 1 เรื่องที่เราต้องเรียกใช้งานให้เป็น ซึ่งมีการทำงานอยู่ 2 โหมดคือ

1. โหมดการถ่ายรูปภาพ
2. โหมดการถ่ายคลิปวิดีโอ

ตัวอย่างที่ 9-3 การใช้งานกล้อง (android.permission.CAMERA) มีขั้นตอน ดังนี้

1. ที่ไฟล์ AndroidManifest.xml ขอใช้สิทธิ์ที่ชื่อว่า android.permission.CAMERA ก่อน

สคริปต์ XML ที่ 9-3 การใช้งานกล้อง (android.permission.CAMERA) (\app\manifests\AndroidMinifest.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.basiccamera">

    <uses-permission android:name="android.permission.CAMERA" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.BasicCamera">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

2. ที่ส่วนแสดงผลหลัก activity_main.xml กำหนดให้มี 2 ปุ่ม เพื่อเรียกใช้กล้อง 2 แบบคือ
   - **ถ่ายรูปภาพ** เป็นหน้าที่ของปุ่มกดที่ชื่อว่า cmdImage (android:id="@+id/cmdImage")
   - **ถ่ายวิดีโอ** เป็นหน้าที่ของปุ่มกดที่ชื่อว่า cmdVideo (android:id="@+id/cmdVideo")

รูปที่ 9-13 กรณีถ่ายคลิปวิดีโอ

รูปที่ 9-14 กรณีถ่ายคลิปวิดีโอเสร็จแล้ว

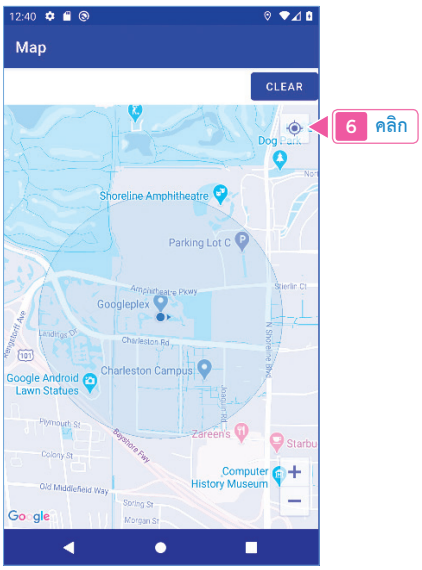
อธิบายการทำงานของโค้ด

1. สร้างตัวแปรที่ชื่อว่า PERMISSION_REQUEST_CODE ทำหน้าที่ตรวจสอบการทำงานในการขอสิทธิ์ใช้กล้อง โดยกำหนดค่าเริ่มต้นเท่ากับ 200 จากนั้น ในฟังก์ชัน onCreate() ตรวจสอบสิทธิ์ Manifest.permission.CAMERA ก่อนเลยว่า อยู่ในสถานะยินยอมหรือไม่

\app\java\com.example.basiccamera\MainActivity.kt

```
class MainActivity : AppCompatActivity() {
    private val PERMISSION_REQUEST_CODE = 200

    override fun onCreate(savedInstanceState: Bundle?) {
```

6. ท้ายที่สุด ในกรณีที่ผู้อ่านต้องการกลับไปยังตำแหน่งปัจจุบันของผู้อ่าน ให้คลิกปุ่ม  ดังรูปที่ 11-15



รูปที่ 11-15 กรณีกลับไปยังตำแหน่งปัจจุบันของผู้อ่าน

อธิบายการทำงานของโค้ด

1. ที่คลาส MapsActivity ให้ทำ
   - **ตัวแปรออบเจ็กต์ GoogleMap** ที่ชื่อว่า mMap (private lateinit var mMap: GoogleMap) ทำหน้าที่แทนบริการแผนที่ Google Maps
   - **ตัวแปร Lat** ทำหน้าที่เก็บพิกัดละติจูด (var Lat = 13.803742)
   - **ตัวแปร Lng** ทำหน้าที่เก็บพิกัดลองจิจูด (var Lng = 100.554603)
   - **ค่าคงที่ LOCATION_REQUEST_CODE = 200** (private val LOCATION_REQUEST_CODE = 200) ทำหน้าที่ตรวจสอบสถานะขอสิทธิ์เข้าถึงตำแหน่งปัจจุบัน

ผู้เขียนกำหนดพิกัดละติจูดกับลองจิจูดไว้ที่สวนจตุจักร ผู้อ่านสามารถเปลี่ยนพิกัดเริ่มต้นในแผนที่ได้ที่นี่

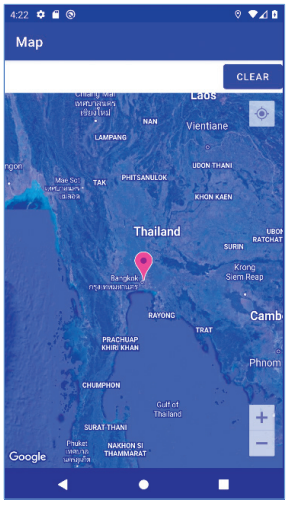
\app\Kotlin\com.example.usinggooglemaps\MapsActivity.kt

```
class MapsActivity : AppCompatActivity(), OnMapReadyCallback {
    private lateinit var mMap: GoogleMap
    var Lat = 13.803742
    var Lng = 100.554603
    private val LOCATION_REQUEST_CODE = 200
```

5. แบบ GoogleMap.MAP_TYPE_HYBRID ดังรูปที่ 11-20

\app\java\com.example.usinggooglemaps\MapsActivity.kt

```
mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID)
```



รูปที่ 11-20 แผนที่แบบ GoogleMap.MAP_TYPE_HYBRID

การค้นหาที่อยู่และปักหมุดในแผนที่

ความต้องการพื้นฐานอย่างหนึ่งของการใช้งานแผนที่ก็คือ เราต้องการค้นหาสถานที่ตามที่เราสนใจ และอยากปักหมุดสถานที่ดังกล่าวไว้ด้วยนั่นเอง

ตัวอย่างที่ 11-3 การค้นหาที่อยู่และปักหมุดในแผนที่

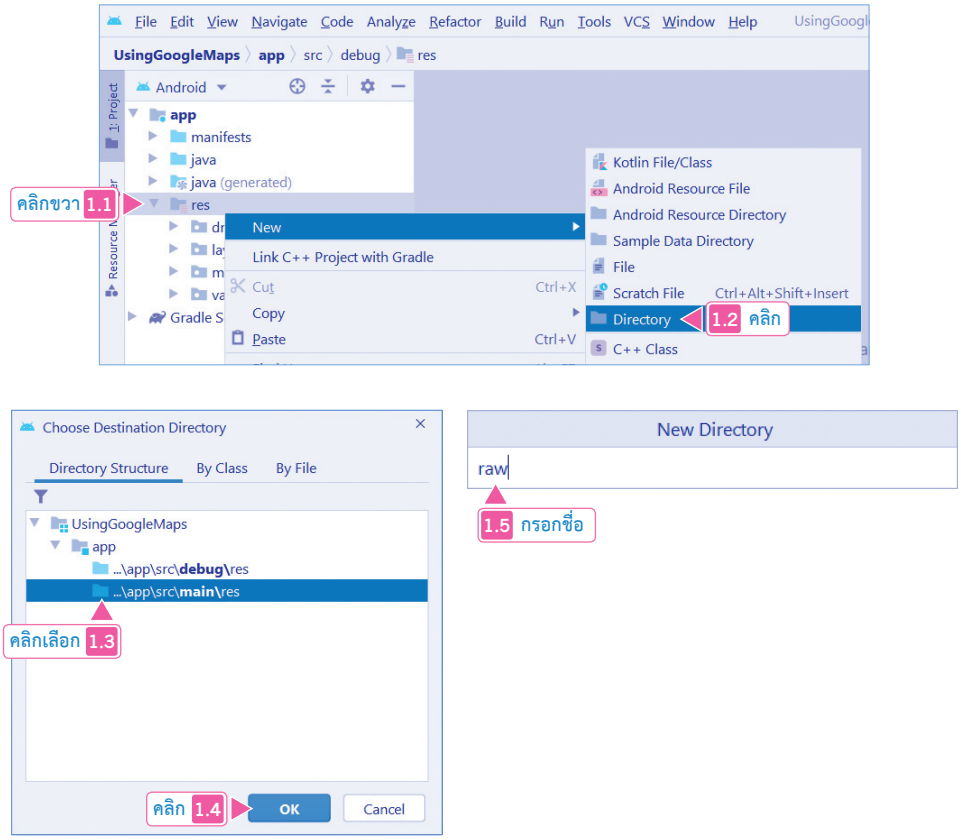
ผู้เขียนนำตัวอย่างที่แล้วมาทำต่อ โดยการเพิ่มช่องค้นหาสถานที่ด้วย โดยมีขั้นตอนดังนี้

1. ที่ส่วนแสดงผลหลัก activity_maps.xml เพิ่มช่องค้นหาสถานที่ตั้งชื่อว่า edtSearch (android:id="@+id/edtSearch") และมีปุ่มกด Button สำหรับสั่งให้ค้นหา ตั้งชื่อว่า cmdSearch (android:id="@+id/cmdSearch") ดังรูปที่ 11-21

การปรับแต่ง Style ให้กับแผนที่ Google Maps

รายละเอียดต่างๆ ที่อยู่ในแผนที่มีมากมาย ผู้อ่านสามารถปรับแต่ง Style แผนที่ให้แสดงผลตรงกับความต้องการของเราได้อีกด้วย โดยมีขั้นตอน ดังนี้

1. ในโปรเจ็กต์ที่มีการขอใช้บริการแผนที่ Google Maps ที่โฟลเดอร์ res ให้ผู้อ่านคลิกขวา เลือกคำสั่ง New > Directory เพื่อสร้างโฟลเดอร์ที่ชื่อว่า raw ขึ้นมาก่อน



รูปที่ 11-23 แสดงการสร้างโฟลเดอร์ raw

2. จากนั้น คลิกขวาที่โฟลเดอร์ raw เลือกคำสั่ง New > File สร้างไฟล์ว่างที่ชื่อว่า style_json.json ทำหน้าที่ปรับแต่ง Style ให้กับแผนที่ Google Maps ดังรูปที่ 11-24

บทที่ 14

การทำระบบ Login ด้วย Google Sign-in

ในการทำแอปประเภทที่มีการกำหนดให้ผู้ใช้งานต้องมีการแสดงตนก่อนใช้งานแอปของเรา มีหลายวิธีหลายรูปแบบ หนึ่งในวิธีที่ได้รับความนิยมในปัจจุบันก็คือ อาศัยการแสดงตนจากบัญชีของ Google Account ซึ่งเป็นบัญชีที่เราได้มาตอนสมัคร Gmail โดยอัตโนมัติ

การสร้างแอปขอใช้บริการ Google Sign-in

ผู้เขียนต้องการสร้างแอปที่ผู้ใช้งานต้องยืนยันตัวเองด้วยบัญชี Google Account ก่อนใช้งานเท่านั้น เมื่อยืนยันเรียบร้อยแล้ว ก็จะเข้าสู่หน้าจอที่สร้างขึ้นมาสำหรับสมาชิกโดยเฉพาะ

ตัวอย่างที่ 14-1 การสร้างแอปขอใช้บริการ Google Sign-in มีขั้นตอน ดังนี้

1. สร้างโปรเจ็กต์แบบ Empty Activity ตั้งชื่อว่า UsingGoogleAuth
2. ให้ผู้อ่านไปที่ไฟล์ AndroidManifest.xml เพื่อตรวจสอบชื่อ Package ของโปรเจ็กต์ปัจจุบัน ให้ผู้อ่าน Copy เก็บไว้ก่อน เพราะเราต้องใช้ชื่อ Package นี้ ขอใช้บริการ Google Sign-in ในภายหลังที่ไฟล์ AndroidManifest.xml นี้ ให้ผู้อ่านขอสิทธิ์ใช้อินเทอร์เน็ตด้วย ดังรูปที่ 14-1

สคริปต์ XML ที่ 14-1 การสร้างแอปขอใช้บริการ Google Sign-in (บางส่วน) (\app\manifests\AndroidManifest.xml)

```
<uses-permission android:name="android.permission.INTERNET" />
```

บทที่ 15

Jetpack Compose

การพัฒนา Android Apps โดยอาศัยภาษา Kotlin ยังมีอีกรูปแบบหนึ่งที่ Google พัฒนาขึ้นมา เรียกว่า **Jetpack Compose** ประกอบไปด้วยรายการชิ้นส่วนที่อยู่ในหมวด androidx.compose มีลักษณะแตกต่างไปจากที่เราเรียนรู้ก่อนหน้านี้พอสมควร ทำหน้าที่สร้างส่วนแสดงผล (UI) ด้วยวิธีการเขียนโค้ด เราจะมาศึกษากันว่ามีประเด็นใดบ้างที่น่าสนใจ

พื้นฐานการสร้างโปรเจ็กต์โดยอาศัย Jetpack Compose

การสร้างโปรเจ็กต์ Android Apps ที่ใช้รูปแบบจากฟีเจอร์ Jetpack Compose มีขั้นตอน ดังนี้

1. ที่หน้าจอเลือกสร้างโปรเจ็กต์ ให้ผู้อ่านเลือกสร้างโปรเจ็กต์แบบ Empty Compose Activity เพื่อให้งานฟีเจอร์ Jetpack Compose ตั้งแต่เริ่มต้นโปรเจ็กต์ ตั้งชื่อว่า UsingJetpack Compose ดังรูปที่ 15-1

บทที่ 17

การอัพโหลดแอปขึ้น Google Play Store

หลังจากที่เราพัฒนา Android Apps จนถึงขั้นตอนที่แอปของเราพร้อมใช้งาน ก็จะเข้าสู่ขั้นตอนสุดท้ายนั่นคือ การอัพแอปของเราขึ้น Google Play Store เพื่อเผยแพร่แอปของเราให้คนอื่นโหลดใช้งานก็คือ เนื้อหาที่จะนำเสนอในบทนี้

ขั้นตอนการนำ Signed APK ขึ้น Play Store มีการอัพเดตเปลี่ยนแปลงอยู่เสมอ ส่งผลให้ขั้นตอนต่างๆ ที่นำเสนอในบทนี้อาจจะตรงหรือไม่ตรงกับปัจจุบันของผู้อ่าน ขอให้ผู้อ่านยึดถือขั้นตอนปัจจุบันที่ผู้อ่านเห็นเป็นหลัก โดยที่ผู้เขียนมีข้อแนะนำในขั้นต้นอยู่ 3 อย่างคือ

1. **เตรียมไฟล์ Signed APK** ให้พร้อมก่อน โดยที่ชื่อ Package ต้องไม่มีคำว่า com.example
2. **ข้อมูลที่เป็นข้อความธรรมดา** ผู้อ่านต้องป้อนในช่องที่มีเครื่องหมาย * ให้ครบตามที่ผู้อ่านเห็น
3. **ไฟล์รูปภาพ** ไฟล์รูปภาพแต่ละขนาดจะถูกนำไปแสดงใน Play Store

การสมัครบัญชีนักพัฒนา Android Apps

มีขั้นตอน ดังนี้

1. ผู้อ่านต้องมีอีเมลของ Gmail ก่อน สมัครฟรีได้ที่เว็บ www.gmail.com เพื่อให้ได้บัญชี Google Account
2. ให้ผู้อ่านไปที่เว็บ https://play.google.com/apps/publish/signup/ เพื่อสมัครบัญชีนักพัฒนาของ Google เสียค่าใช้จ่ายครั้งเดียวคือ 25 เหรียญ (ค่าใช้จ่ายอาจจะเท่าหรือไม่เท่าก็ได้ ตรวจสอบตามช่วงเวลาของผู้อ่านเป็นหลัก)