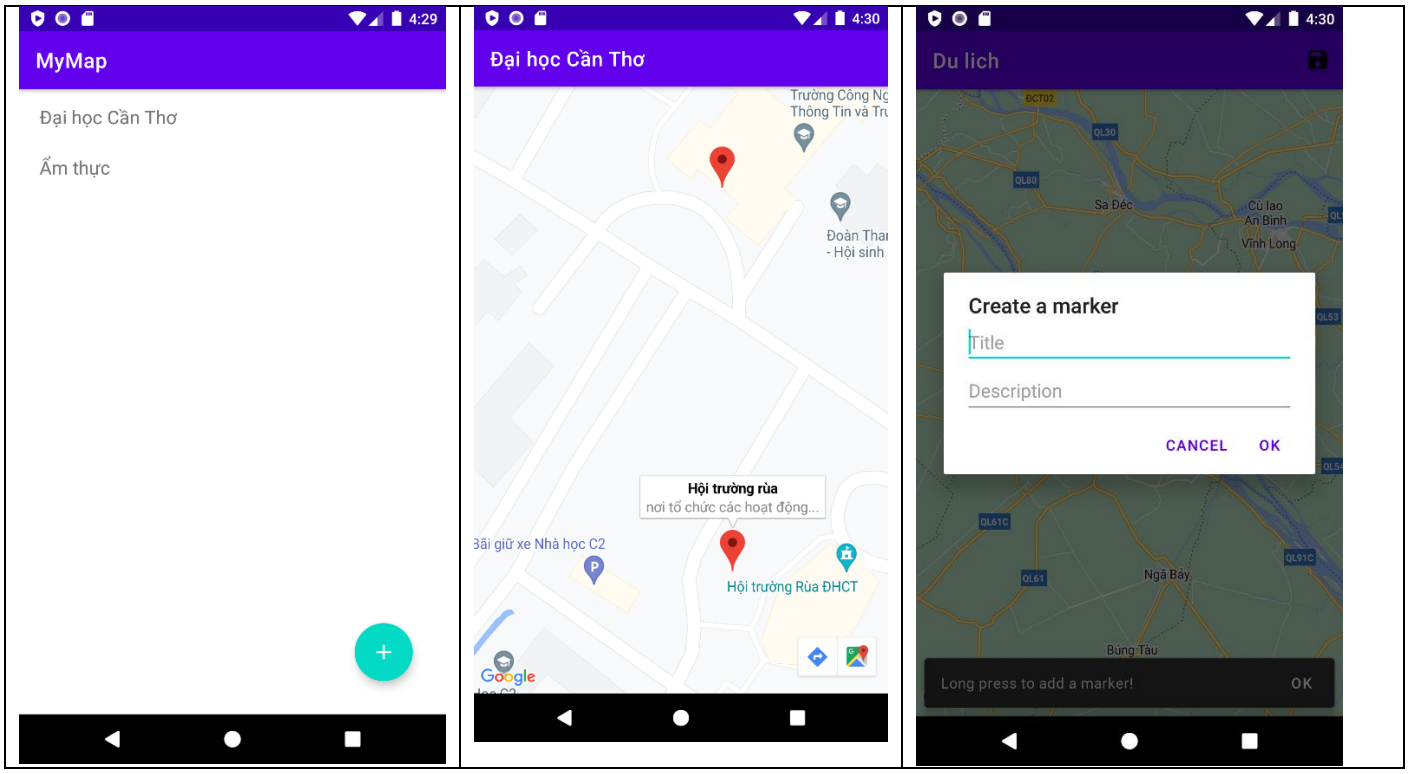
THỰC HÀNH ANDROID KOTLIN – TUẦN 14

ỨNG DỤNG ĐỊA ĐIỂM YÊU THÍCH



I. Mục tiêu

1. Hướng dẫn lập trình Google Map trên Android
2. Hiển thị danh sách địa điểm bằng RecyclerView
3. Sử dụng truyền dữ liệu Serializable qua Intent
4. Sử dụng AlertDialog

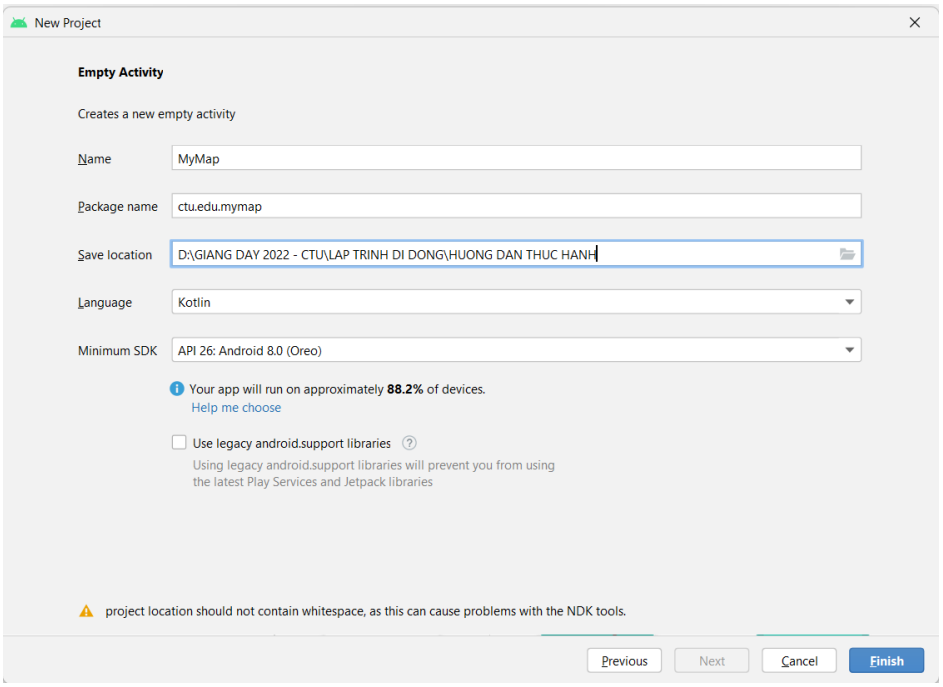
II. Hướng dẫn

0. Config Gradle

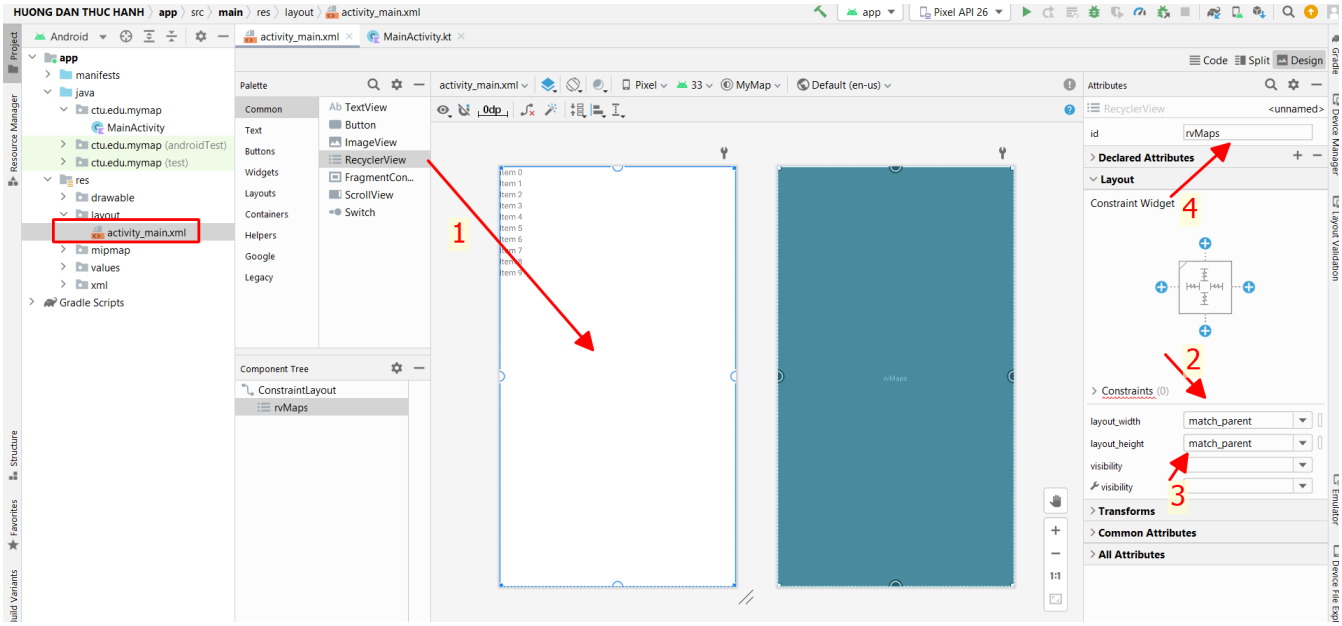
```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'com.google.android.libraries.mapsplatform.secrets-gradle-plugin'
}

android {
    compileSdk 32

    defaultConfig {
        applicationId "ctu.edu.mymap"
        minSdk 26
        targetSdk 32
```

```
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }

    buildFeatures {
        viewBinding true
    }
}

dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.android.gms:play-services-maps:18.1.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
}
```

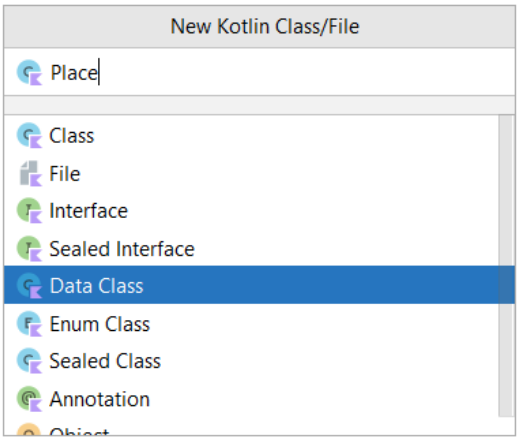
1. Tạo new project → Empty Activity → Next → Finish



2. Tạo RecyclerView (để hiển thị danh sách địa điểm)
   - Click vào chỉnh sửa file activity_main.xml.
   - Xóa TextView "Hello World" mặc định.
   - Kéo thả RecyclerView vào layout.
   - Chỉnh layout_width và layout_height của RecyclerView thành **match_parent**.
   - Đặt id cho RecyclerView thành **rvMaps**.



3. Tạo class UserMap ("lưu bản đồ người dùng") và Place ("địa điểm")
   - Tạo mới package: New → Package → Đặt tên "models"



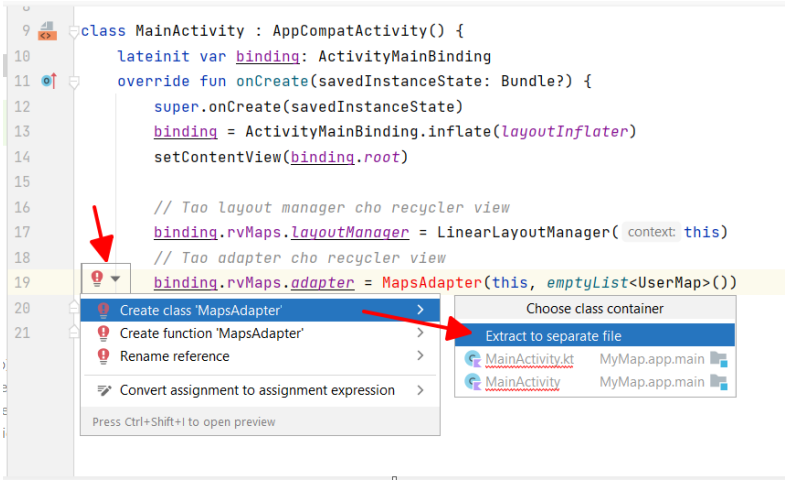
- Tạo data class Place: Chuột phải vào package models: **New** → **Kotlin Class/File** → Đặt tên **Place** và chọn **data class**.
- Trong data class **Place** có các thuộc tính: Tiêu đề (title), Mô tả (description), Vĩ độ (latitude), Kinh độ (longitude).

```
data class Place(
    val title: String,
    val description: String,
    val latitude: Double,
    val longitude: Double
)
```

- Tương tự, tạo data class UserMap: Chuột phải vào package models: **New** → **Kotlin Class/File** → Đặt tên **UserMap** và chọn **data class**.
- Trong data class UserMap có các thuộc tính: Tiêu đề (title), danh sách địa điểm (places)

```
data class UserMap(
    val title: String,
    val places: List<Place>
)
```

4. Viết code cho RecyclerView "rvMaps"
   - MainActivity: **setup binding như các project trước**. (lateinit var binding: ActivityMainBinding, binding = ActivityMainBinding.inflate(layoutInflater), setContentView(binding.root)).
   - Tạo layout manager cho rvMaps:
     binding.rvMaps.layoutManager = LinearLayoutManager(this)
   - Tạo adapter cho rvMaps:
     binding.rvMaps.adapter = **MapsAdapter**(this, emptyList<UserMap>())
   - Click vào gợi ý fix lỗi chỗ MapsAdapter → Chọn create class → Extract to separate file → OK



- Tạo layout row cho RecyclerView: chuột phải vào layout: New → Layout Resource File → đặt tên row_place.xml và chỉnh giao diện như sau:

```
<?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="wrap_content"
    android:padding="10sp"
    android:id="@+id/row_layout"
    >
    <TextView
        android:id="@+id/tv_place"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Đại học Cần Thơ"
        android:textSize="18sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginStart="10dp"
        />
</androidx.constraintlayout.widget.ConstraintLayout>
```

- Viết code cho MapsAdapter

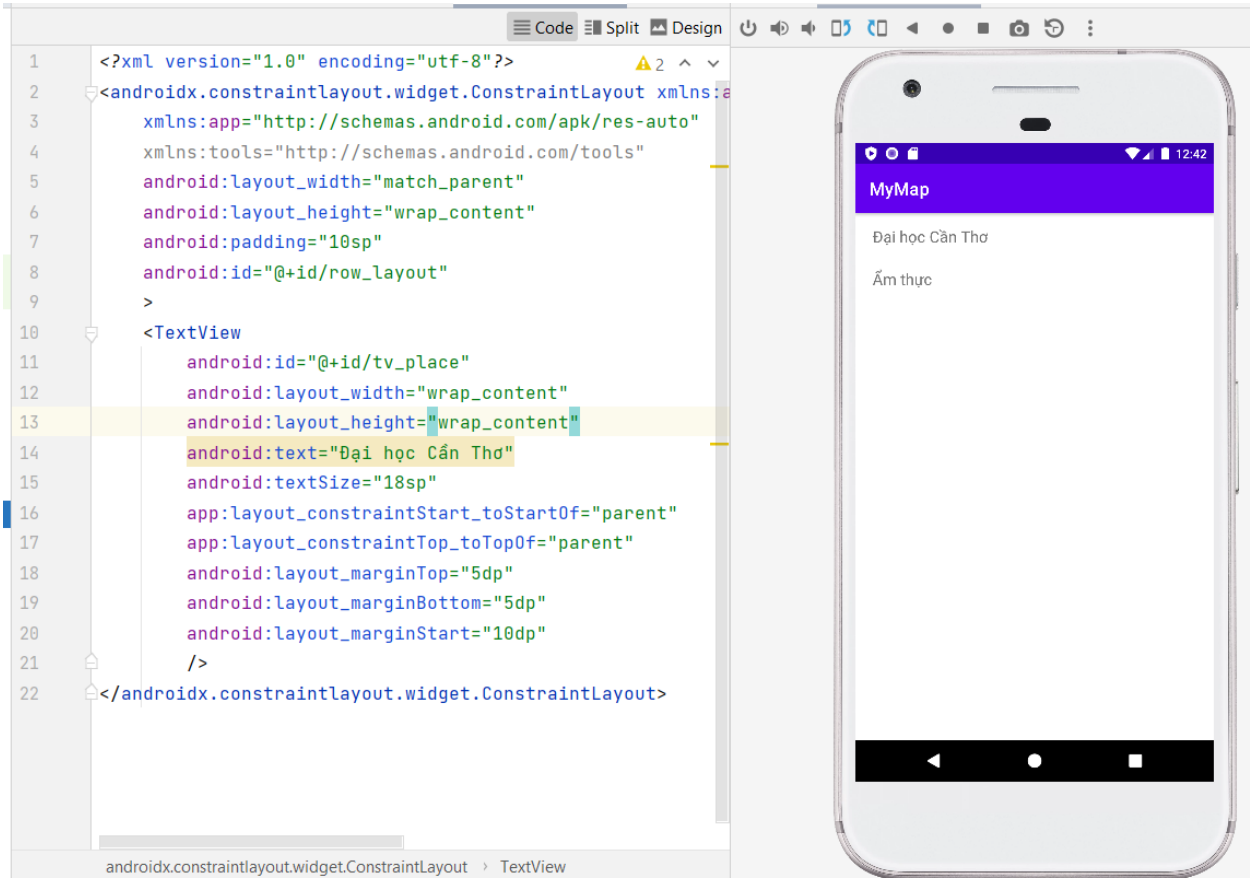
```
class MapsAdapter(val context: Context, val userMaps: List<UserMap>) :
RecyclerView.Adapter<MapsAdapter.MyViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
MyViewHolder {
        val view = LayoutInflater.from(context).inflate(R.layout.row_place ,
parent, false)
        return MyViewHolder(view)
    }

    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
```

```
        val userMap = userMaps[position]
        val tvTitle = holder.itemView.findViewById<TextView>(R.id.tv_place)
        tvTitle.text = userMap.title
    }

    override fun getItemCount():Int = userMaps.size

    class MyViewHolder(itemView: View):RecyclerView.ViewHolder(itemView){
    }
}
```

- Viết hàm sinh dữ liệu để kiểm tra hiển thị rvMaps: Tạo hàm generateSimpleData() trong MainActivity như sau:

```
private fun generateSimpleData(): List<UserMap>{
    return listOf(
        UserMap("Đại học Cần Thơ",
            listOf(
                Place("Trường CNTT&TT", "thuộc ĐH Cần Thơ", 10.0308541,
105.768986),
                Place("Trường Nông nghiệp", "thuộc ĐH Cần Thơ",
10.0302655,105.7679642),
                Place("Hội trường rùa", "nơi tổ chức các hoạt động...",
10.0293402,105.7690273)
            )
        ),
        UserMap("Ẩm thực",
            listOf(
                Place("The 80's icafe", "Đường Mạc Thiên Tích",
10.0286827,105.7732964),
                Place("Trà Sữa Tigon", "Đường Mạc Thiên Tích",
10.0278105,105.7718373),
                Place("Cafe Thủy Mộc", "Đường 3/2", 10.0273775,105.7704913)
            )
        )
    )
}
```

- Run app kiểm tra dữ liệu hiển thị đúng chưa.



- Tiếp tục viết xử lý sự kiện click cho từng item của rvMaps bằng cách chỉnh lại MapsAdapter như sau (bổ sung thêm interface OnClickListener):

```
private const val TAG = "MapsAdapter"
class MapsAdapter(val context: Context, val userMaps: List<UserMap>, val
onClickListener: OnClickListener) :
RecyclerView.Adapter<MapsAdapter.MyViewHolder>() {

    interface OnClickListener {
        fun onItemClick(position: Int)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
MyViewHolder {
        val view = LayoutInflater.from(context).inflate(R.layout.row_place ,
parent, false)
        return MyViewHolder(view)
    }

    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        val userMap = userMaps[position]
        val tvTitle = holder.itemView.findViewById<TextView>(R.id.tv_place)
        tvTitle.text = userMap.title
        holder.itemView.setOnClickListener {
```

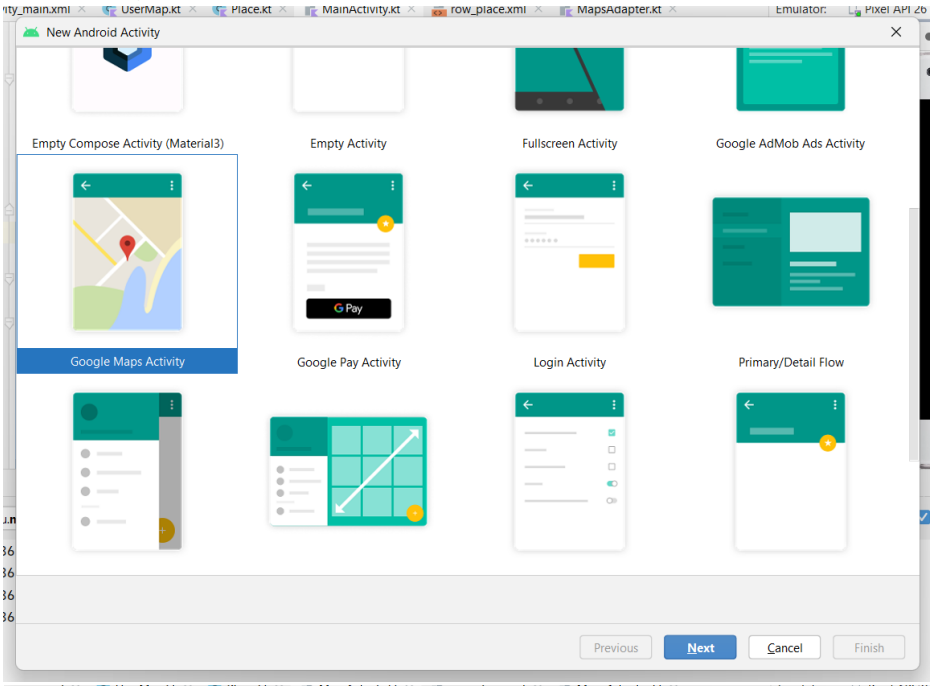
```
            Log.i(TAG, "Click on position $position")
            onClickListener.onItemClick(position)
        }
    }

    override fun getItemCount():Int = userMaps.size

    class MyViewHolder(itemView: View):RecyclerView.ViewHolder(itemView){
    }
}
```

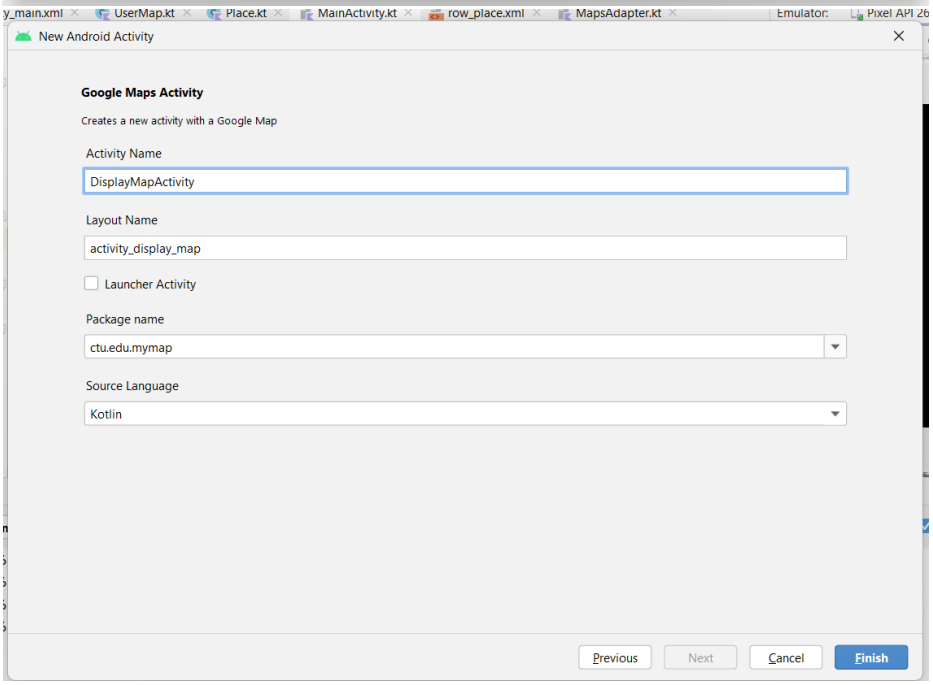
- Chỉnh lại code khai báo bên phần MainActivity

```
private const val TAG = "MainActivity"
class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Tao layout manager cho recycler view
        binding.rvMaps.layoutManager = LinearLayoutManager(this)
        // Tao adapter cho recycler view
        binding.rvMaps.adapter = MapsAdapter(this, generateSimpleData(),
object: MapsAdapter.OnClickListener{
            override fun onItemClick(position: Int) {
                Log.i(TAG, "onItemClick $position")
            }
        })
    }

    private fun generateSimpleData(): List<UserMap>{
        return listOf(
            UserMap("Đại học Cần Thơ",
                listOf(
                    Place("Trường CNTT&TT", "thuộc ĐH Cần Thơ", 10.0308541,
105.768986),
                    Place("Trường Nông nghiệp", "thuộc ĐH Cần Thơ",
10.0302655,105.7679642),
                    Place("Hội trường rùa", "nơi tổ chức các hoạt động...",
10.0293402,105.7690273)
                )
            ),
            UserMap("Ẩm thực",
                listOf(
                    Place("The 80's icafe", "Đường Mạc Thiên Tích",
10.0286827,105.7732964),
                    Place("Trà Sữa Tigon", "Đường Mạc Thiên Tích",
10.0278105,105.7718373),
                    Place("Cafe Thủy Mộc", "Đường 3/2",
10.0273775,105.7704913)
                )
            )
        )
    }
}
```

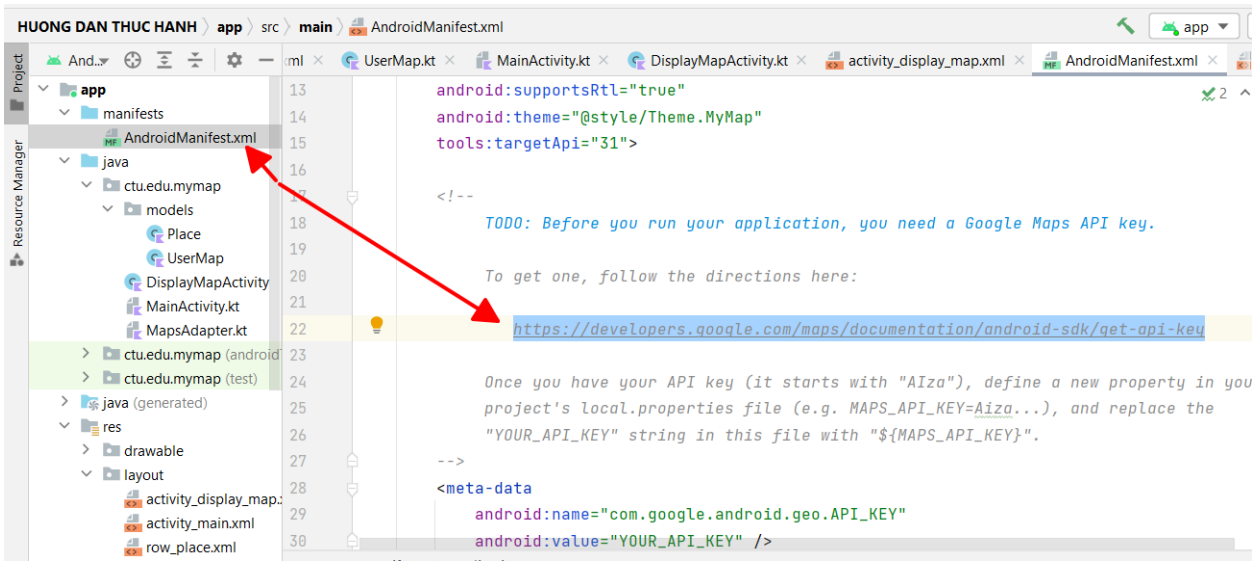
- Chạy app lại thử, kiểm tra sự kiện onClick đã hoạt động chưa (check ở phần Logcat)

5. Tạo activity map để hiển thị danh sách địa điểm
   - New → Activity → Google Maps Activity → Next → Đặt tên "DisplayMapActivity" → Finish (Đợi gradle build...)

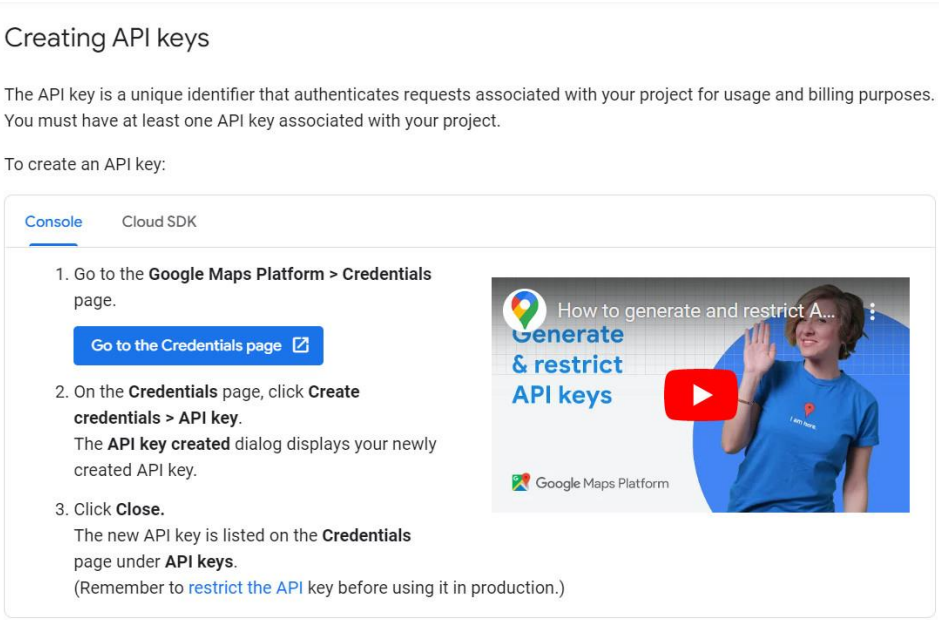




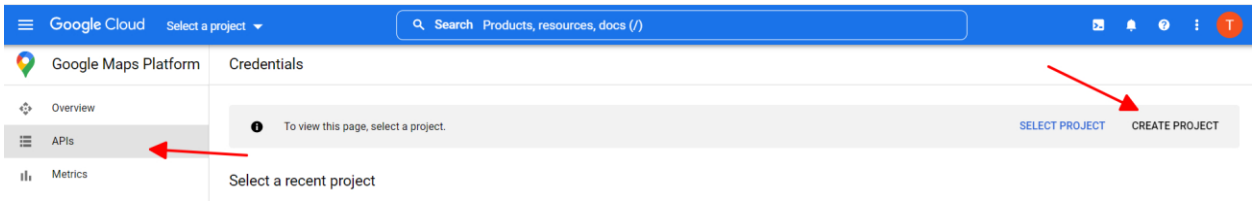
- Mở file AndroidManifest.xml: các bạn sẽ thấy đường link để tạo API Key (bắt buộc phải có khi sử dụng GoogleMap)



- Các bạn mở link bằng trình duyệt (Chrome) và làm theo hướng dẫn để tạp API Key

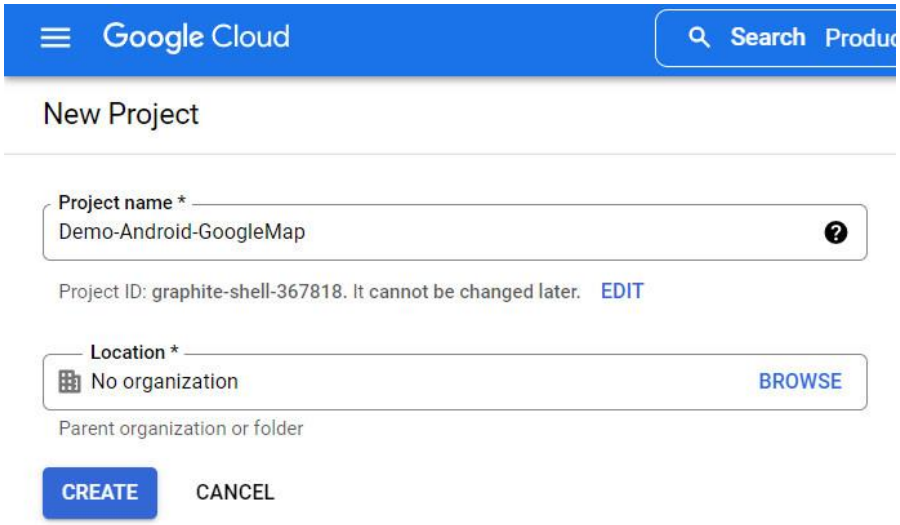


Click vào “Go to the Credentials page”



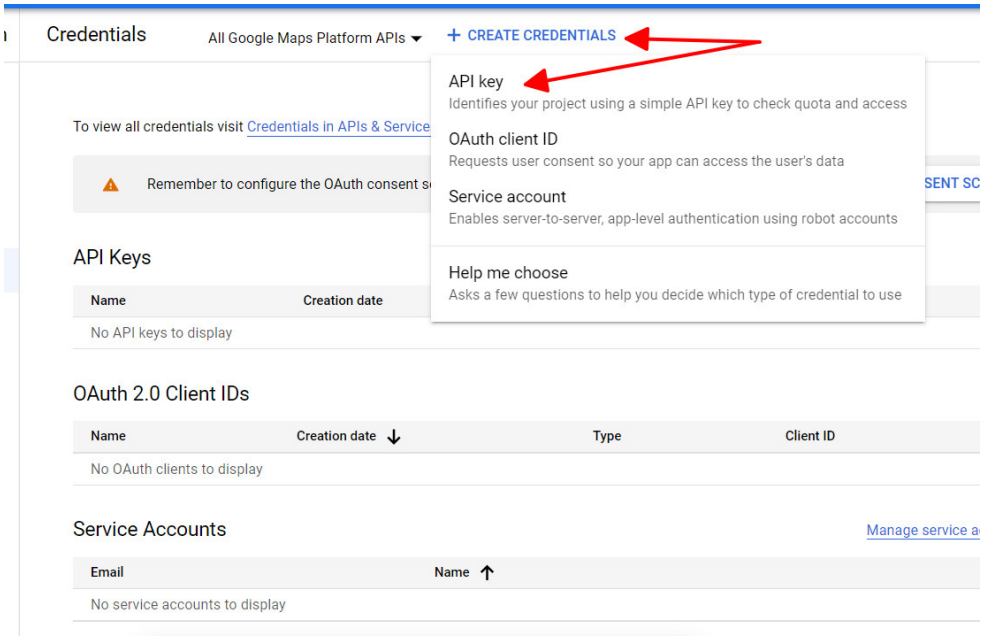
Click vào APIs → CREATE PROJECT

- Tạo project:



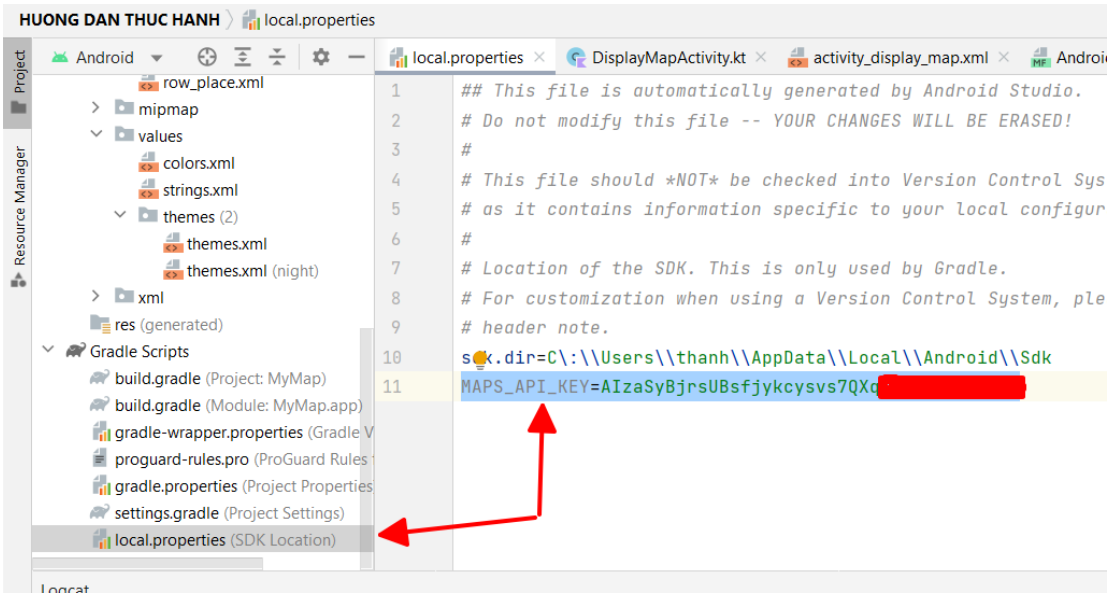
Đặt tên project và click vào CREATE

- Tạo API Key

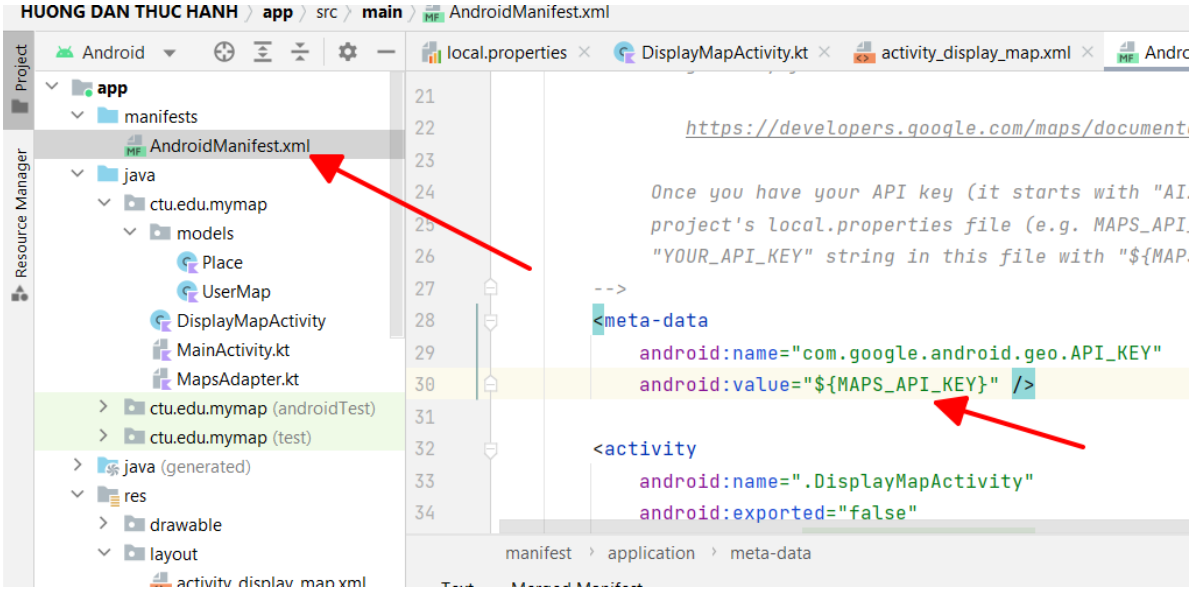


Click vào “CREATE CREDENTAILS” → API key

- Sau khi tạo API key xong, mở file local.properties thêm vào như hình sau:



Chỉnh sửa thêm key vào local.properties

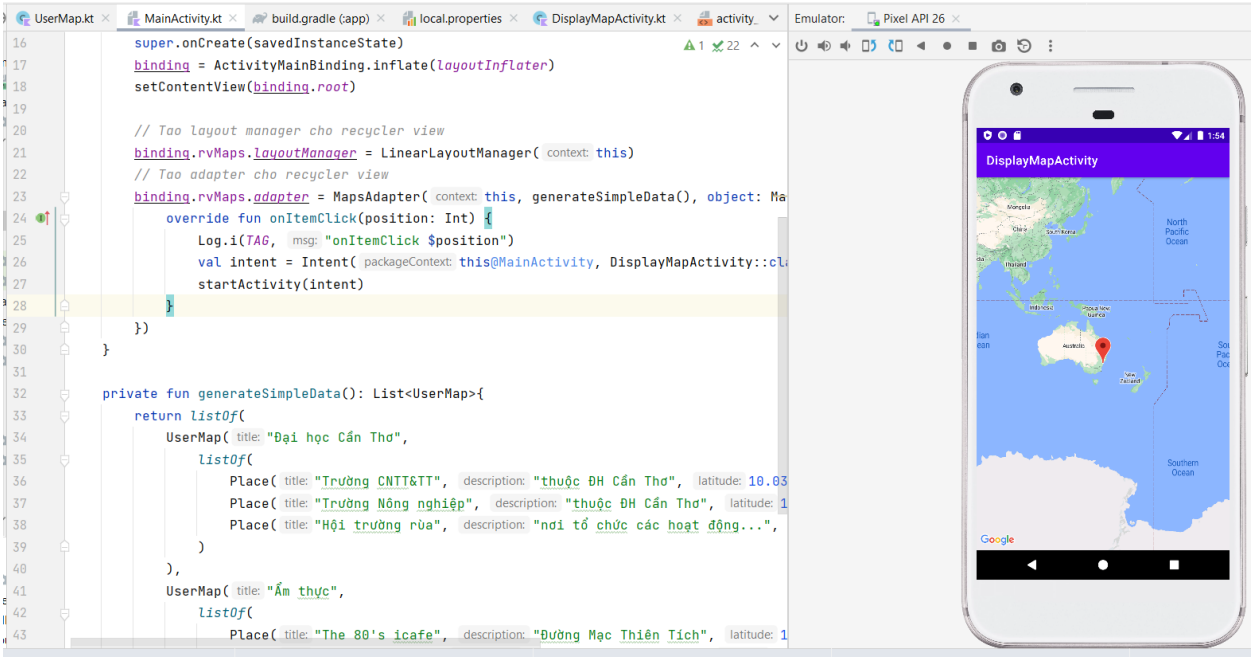


Chỉnh lại API key trong AndroidManifest.xml như hình trên

- Thêm code gọi Intent DisplayMapActivity như sau:

```
// Tao layout manager cho recycler view
binding.rvMaps.layoutManager = LinearLayoutManager( context: this)
// Tao adapter cho recycler view
binding.rvMaps.adapter = MapsAdapter( context: this, generateSimpleData(), object: MapsAdapter.On
    override fun onItemClick(position: Int) {
        Log.i(TAG, msg: "onItemClick $position")
        val intent = Intent( packageContext: this@MainActivity, DisplayMapActivity::class.java)
        startActivity(intent)
    }
})
}
```

- Chạy thử app và click vào item bất kỳ xem hiển thị được Map chưa.



- **Code để hiển thị danh sách UserMap:**
  a. Tạo thêm file Utils (New → Kotlin Class/File → Object → Đặt tên Utils)

```
object Utils {
    const val EXTRA_USER_MAP = "EXTRA_USER_MAP"
}
```

Dùng để lưu các giá trị cần thiết: cụ thể ở đây là KEY để gọi Intent

  b. Thêm Serializable vào Place và UserMap (để có thể truyền dữ liệu class này qua Intent)

```
data class Place(
    val title: String,
    val description: String,
    val latitude: Double,
    val longitude: Double
):Serializable

data class UserMap(
    val title: String,
    val places: List<Place>
): Serializable
```

  c. Chỉnh lại hàm onItemClick

```
private const val TAG = "MainActivity"
class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        // Khởi tạo dữ liệu
```

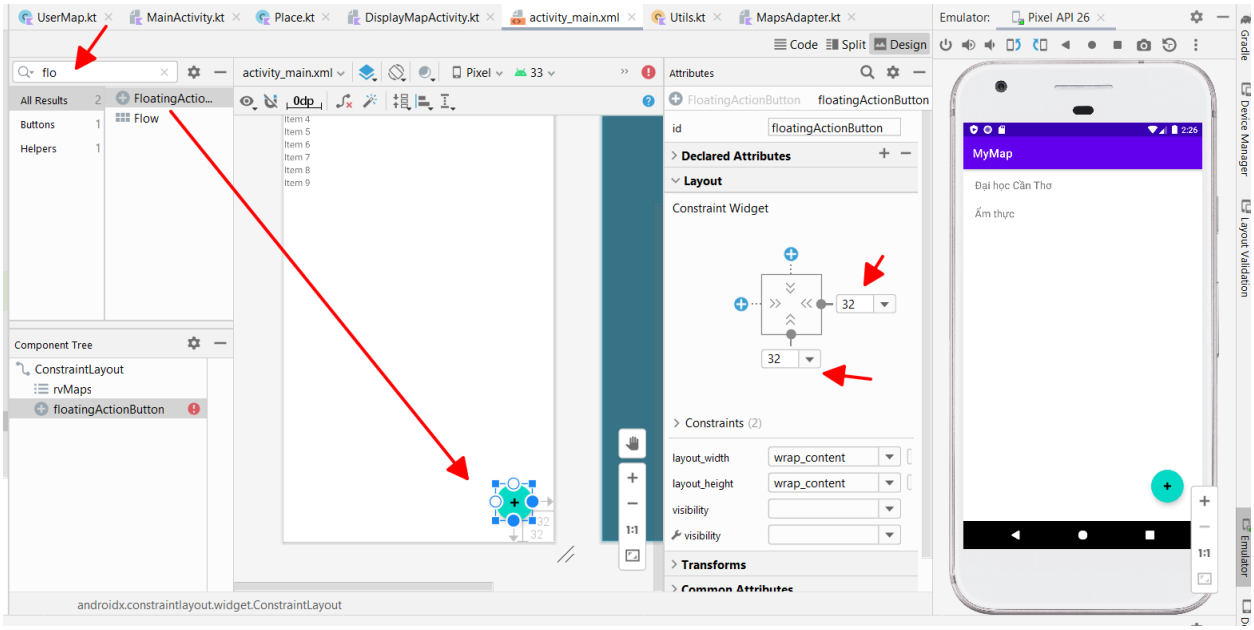
```
        val userMaps = generateSimpleData()
        // Tao layout manager cho recycler view
        binding.rvMaps.layoutManager = LinearLayoutManager(this)
        // Tao adapter cho recycler view
        binding.rvMaps.adapter = MapsAdapter(this, userMaps, object:
MapsAdapter.OnClickListener{
            override fun onItemClick(position: Int) {
                Log.i(TAG, "onItemClick $position")
                val intent = Intent(this@MainActivity,
DisplayMapActivity::class.java)
                intent.putExtra(Utils.EXTRA_USER_MAP, userMaps[position])
                startActivity(intent)
            }
        })
    }

    private fun generateSimpleData(): List<UserMap>{
        return listOf(
            UserMap("Đại học Cần Thơ",
                listOf(
                    Place("Trường CNTT&TT", "thuộc ĐH Cần Thơ", 10.0308541,
105.768986),
                    Place("Trường Nông nghiệp", "thuộc ĐH Cần Thơ",
10.0302655,105.7679642),
                    Place("Hội trường rùa", "nơi tổ chức các hoạt động...",
10.0293402,105.7690273)
                )
            ),
            UserMap("Ẩm thực",
                listOf(
                    Place("The 80's icafe", "Đường Mạc Thiên Tích",
10.0286827,105.7732964),
                    Place("Trà Sữa Tigon", "Đường Mạc Thiên Tích",
10.0278105,105.7718373),
                    Place("Cafe Thủy Mộc", "Đường 3/2",
10.0273775,105.7704913)
                )
            )
        )
    }
}
```

d. Chỉnh sửa code cho DisplayMapActivity: Đọc dữ liệu truyền vào từ Intent và hiển thị lên bản đồ, chỉnh lại tên cho ActionBar dựa vào userMap

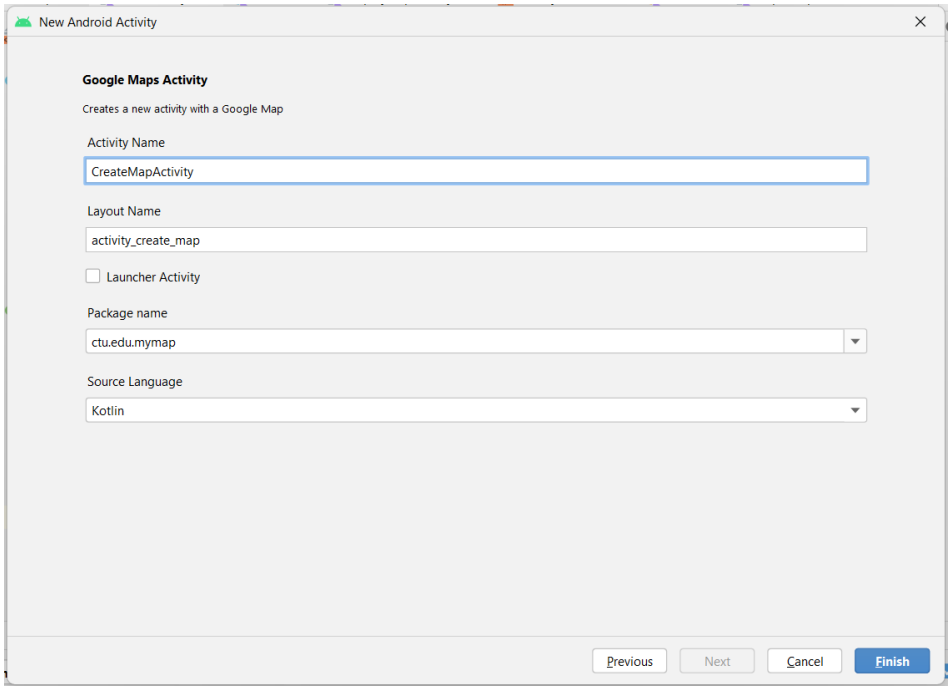
```
private const val TAG = "DisplayMapActivity"
class DisplayMapActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var binding: ActivityDisplayMapBinding
    private lateinit var userMap: UserMap
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityDisplayMapBinding.inflate(layoutInflater)
        setContentView(binding.root)
```

```
        userMap = intent.getSerializableExtra(Utils.EXTRA_USER_MAP) as
UserMap
        supportActionBar?.title = userMap.title
        // Obtain the SupportMapFragment and get notified when the map is
ready to be used.
        val mapFragment = supportFragmentManager
            .findFragmentById(R.id.map) as SupportMapFragment
        mapFragment.getMapAsync(this)
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the
camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will
be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered
once the user has
     * installed Google Play services and returned to the app.
     */
    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap
        Log.i(TAG, "map: ${userMap.title}")
        val boundsBuilder = LatLngBounds.builder()
        for(place in userMap.places){
            val latLng = LatLng(place.latitude, place.longitude)
            boundsBuilder.include(latLng)

mMap.addMarker(MarkerOptions().position(latLng).title(place.title).snippet(pl
ace.description))
        }

mMap.animateCamera(CameraUpdateFactory.newLatLngBounds(boundsBuilder.build(),
1000, 1000, 0))
//
mMap.moveCamera(CameraUpdateFactory.newLatLngBounds(boundsBuilder.build(),
1000, 1000, 0))
//        val ctu = LatLng(10.031452976258134, 105.77197889530333)
//        mMap.addMarker(MarkerOptions().position(ctu).title("Trường ĐH Cần
Tho"))
//        mMap.moveCamera(CameraUpdateFactory.newLatLng(ctu))
    }
}
```

6. Viết code xử lý thêm địa điểm mới: Tạo floating button, khi click vào nút floating sẽ hiển thị activity Map để cho chọn địa điểm mới…
   - Thêm FloatingActionButton vào activity_main.xml như các project trước.



- Tạo activity map mới (tương tự ở trên): New → Activity → Google Maps Activity → Next → Đặt tên "**CreateMapActivity**" → Finish



- Viết code xử lý sự kiện click cho FloatingActionButton: Khi click vào sẽ gọi intent có kết quả trả về, kết quả trả về sẽ được xử lý ở hàm getResult.

```
binding.floatingActionButton.setOnClickListener {
    val intent = Intent(this@MainActivity, CreateMapActivity::class.java)
    intent.putExtra(Utils.EXTRA_MAP_TITLE, "New map name")
    getResult.launch(intent)
}
```

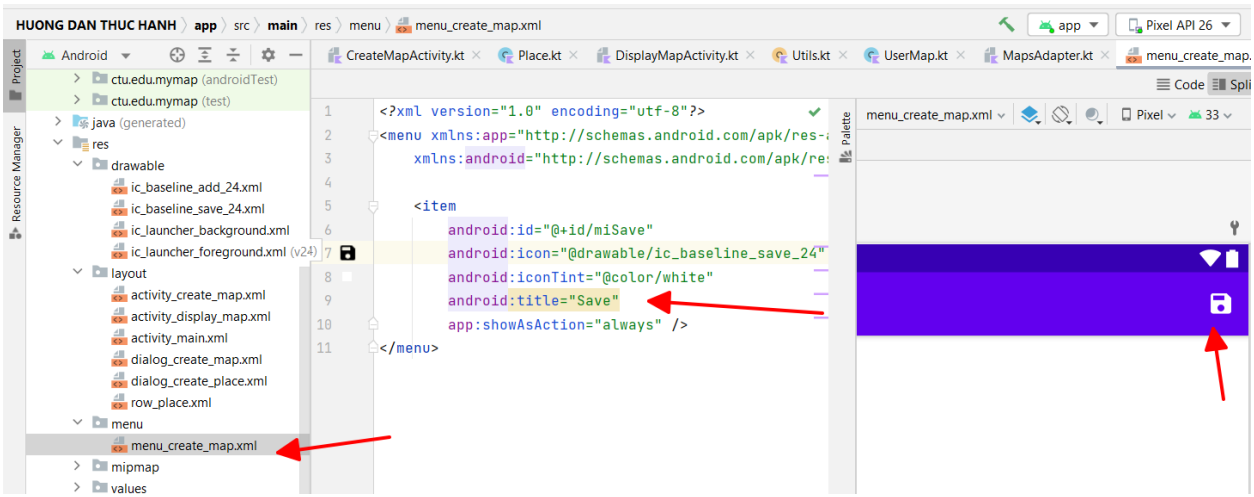
```
// Receiver
private val getResult =
registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
    if (it.resultCode == Activity.RESULT_OK) {
        val value = it.data?.getStringExtra("input")
    }
}
```

- Viết code cho **CreateMapActivity**: Khi người dùng giữ chuột trái lâu trên bản đồ sẽ hiển thị ra dialog cho người dùng nhập vào Name và Description, click OK sẽ tạo ra marker mới trên bản đồ. (Sử dụng snackbar để ra thông báo cho người dùng biết cách sử dụng)

```
private const val TAG = "CreateMapActivity"
class CreateMapActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var binding: ActivityCreateMapBinding
    private var markers: MutableList<Marker> = mutableListOf<Marker>()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityCreateMapBinding.inflate(layoutInflater)
        setContentView(binding.root)
        val title = intent.getStringExtra(Utils.EXTRA_MAP_TITLE)
        supportActionBar?.title = title
        // Obtain the SupportMapFragment and get notified when the map is
ready to be used.
        val mapFragment = supportFragmentManager
            .findFragmentById(R.id.map) as SupportMapFragment
        mapFragment.getMapAsync(this)
        mapFragment.view?.let {
            Snackbar.make(it, "Long press to add a marker!",
Snackbar.LENGTH_INDEFINITE)
                .setAction("OK", {})
                .setActionTextColor(ContextCompat.getColor(this,
R.color.white))
                .show()
        }
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the
camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will
be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered
once the user has
     * installed Google Play services and returned to the app.
     */
    override fun onMapReady(googleMap: GoogleMap) {
```

```
        mMap = googleMap
        mMap.setOnInfoWindowClickListener {
            marker ->
            Log.i(TAG,"setOnInfoWindowClickListener - Delete")
            markers.remove(marker)
            marker.remove()
        }
        mMap.setOnMapLongClickListener { latLng ->
            Log.i(TAG,"setOnMapLongClickListener")
            val placeFormView =
LayoutInflater.from(this).inflate(R.layout.dialog_create_place, null)
            AlertDialog.Builder(this).setTitle("Create a marker")
                .setView(placeFormView)
                .setNegativeButton("Cancel", null)
                .setPositiveButton("OK"){
                    _,_ ->
                    val _title =
placeFormView.findViewById<EditText>(R.id.et_title).text.toString()
                    val _description =
placeFormView.findViewById<EditText>(R.id.et_description).text.toString()
                    if (_title.trim().isEmpty() ||
_description.trim().isEmpty()){
                        Toast.makeText(this, "Fill out title & description",
Toast.LENGTH_SHORT).show()
                        return@setPositiveButton
                    }
                    val marker = mMap.addMarker(

MarkerOptions().position(latLng).title(_title).snippet(_description)
                    )
                    markers.add(marker!!)
                }
                .show()
        }
        // Add a marker in CTU and move the camera
        val ctu = LatLng(10.031452976258134, 105.77197889530333)
        mMap.addMarker(MarkerOptions().position(ctu).title("Trường ĐH Cần
Thơ"))
        mMap.moveCamera(CameraUpdateFactory.newLatLng(ctu))
    }
}
```

7. Tạo menu



Code xử lý menu trong CreateMapActivity

```
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.menu_create_map, menu)
    return super.onCreateOptionsMenu(menu)
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
}
```

8. **Hoàn thiện ứng dụng** (bổ sung thêm: khi click tạo map thì cho người dùng nhập vào title để lưu trữ danh sách map, click vào menu Save để trả kết quả về Intent parent)

MainActivity

```
private const val TAG = "MainActivity"
class MainActivity : AppCompatActivity() {
    lateinit var binding: ActivityMainBinding
    lateinit var userMaps: MutableList<UserMap>
    lateinit var mapAdapter:MapsAdapter
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        // Khởi tạo dữ liệu
        userMaps = generateSimpleData().toMutableList()
        // Tao layout manager cho recycler view
        binding.rvMaps.layoutManager = LinearLayoutManager(this)
        // Tao adapter cho recycler view
        mapAdapter = MapsAdapter(this, userMaps, object:
MapsAdapter.OnClickListener{
            override fun onItemClick(position: Int) {
                Log.i(TAG, "onItemClick $position")
                val intent = Intent(this@MainActivity,
DisplayMapActivity::class.java)
```

```
                intent.putExtra(Utils.EXTRA_USER_MAP, userMaps[position])
                startActivity(intent)
            }
        })
        binding.rvMaps.adapter = mapAdapter
        binding.floatingActionButton.setOnClickListener {
            val mapFormView =
LayoutInflater.from(this).inflate(R.layout.dialog_create_map, null)
            AlertDialog.Builder(this).setTitle("Map Title")
                .setView(mapFormView)
                .setNegativeButton("Cancel", null)
                .setPositiveButton("OK"){
                        _,_ ->
                    val _title =
mapFormView.findViewById<EditText>(R.id.et_title_map).text.toString()
                    if (_title.trim().isEmpty()){
                        Toast.makeText(this, "Fill out title",
Toast.LENGTH_SHORT).show()
                        return@setPositiveButton
                    }
                    val intent = Intent(this@MainActivity,
CreateMapActivity::class.java)
                    intent.putExtra(Utils.EXTRA_MAP_TITLE, _title)
                    getResult.launch(intent)
                }
                .show()
        }
    }

    // Receiver
    private val getResult =
registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
        if (it.resultCode == Activity.RESULT_OK) {
            val userMap = it.data?.getSerializableExtra(Utils.EXTRA_USER_MAP)
as UserMap
            userMaps.add(userMap)
            mapAdapter.notifyItemInserted(userMaps.size - 1)

            Log.i(TAG, userMap.title)
        }
    }


    private fun generateSimpleData(): List<UserMap>{
        return listOf(
            UserMap("Đại học Cần Thơ",
                listOf(
                    Place("Trường CNTT&TT", "thuộc ĐH Cần Thơ", 10.0308541,
105.768986),
                    Place("Trường Nông nghiệp", "thuộc ĐH Cần Thơ",
10.0302655,105.7679642),
                    Place("Hội trường rùa", "nơi tổ chức các hoạt động...",
10.0293402,105.7690273)
                )
            ),
            UserMap("Ẩm thực",
                listOf(
```

```
                    Place("The 80's icafe", "Đường Mạc Thiên Tích",
10.0286827,105.7732964),
                    Place("Trà Sữa Tigon", "Đường Mạc Thiên Tích",
10.0278105,105.7718373),
                    Place("Cafe Thủy Mộc", "Đường 3/2",
10.0273775,105.7704913)
                )
            )
        )
    }
}
```

CreateMapActivity

```
private const val TAG = "CreateMapActivity"
class CreateMapActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var binding: ActivityCreateMapBinding
    private var markers: MutableList<Marker> = mutableListOf<Marker>()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityCreateMapBinding.inflate(layoutInflater)
        setContentView(binding.root)
        val title = intent.getStringExtra(Utils.EXTRA_MAP_TITLE)
        supportActionBar?.title = title
        // Obtain the SupportMapFragment and get notified when the map is
ready to be used.
        val mapFragment = supportFragmentManager
            .findFragmentById(R.id.map) as SupportMapFragment
        mapFragment.getMapAsync(this)
        mapFragment.view?.let {
            Snackbar.make(it, "Long press to add a marker!",
Snackbar.LENGTH_INDEFINITE)
                .setAction("OK", {})
                .setActionTextColor(ContextCompat.getColor(this,
R.color.white))
                .show()
        }
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu_create_map, menu)
        return super.onCreateOptionsMenu(menu)
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        if (item.itemId == R.id.miSave){
            Log.i(TAG, "Clicked on Save!")
            if (markers.isEmpty()){
                Toast.makeText(this, "There must be at least one marker on
the map", Toast.LENGTH_SHORT).show()
                return true
            }
            val places = markers.map{
```

```
                it -> Place(it.title!!, it.snippet!!, it.position.latitude,
it.position.longitude)
            }
            val userMap =
UserMap(intent.getStringExtra(Utils.EXTRA_MAP_TITLE)!!, places)
            val data = Intent()
            data.putExtra(Utils.EXTRA_USER_MAP, userMap)
            setResult(Activity.RESULT_OK, data)
            finish()
            return true
        }
        return super.onOptionsItemSelected(item)
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the
camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will
be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered
once the user has
     * installed Google Play services and returned to the app.
     */
    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap
        mMap.setOnInfoWindowClickListener {
            marker ->
            Log.i(TAG,"setOnInfoWindowClickListener - Delete")
            markers.remove(marker)
            marker.remove()
        }
        mMap.setOnMapLongClickListener { latLng ->
            Log.i(TAG,"setOnMapLongClickListener")
            val placeFormView =
LayoutInflater.from(this).inflate(R.layout.dialog_create_place, null)
            AlertDialog.Builder(this).setTitle("Create a marker")
                .setView(placeFormView)
                .setNegativeButton("Cancel", null)
                .setPositiveButton("OK"){
                    _,_ ->
                    val _title =
placeFormView.findViewById<EditText>(R.id.et_title).text.toString()
                    val _description =
placeFormView.findViewById<EditText>(R.id.et_description).text.toString()
                    if (_title.trim().isEmpty() ||
_description.trim().isEmpty()){
                        Toast.makeText(this, "Fill out title & description",
Toast.LENGTH_SHORT).show()
                        return@setPositiveButton
                    }
                    val marker = mMap.addMarker(

MarkerOptions().position(latLng).title(_title).snippet(_description)
                    )
```

```
                    markers.add(marker!!)
                }
                .show()
        }
        // Add a marker in CTU and move the camera
        val ctu = LatLng(10.031452976258134, 105.77197889530333)
        mMap.addMarker(MarkerOptions().position(ctu).title("Trường ĐH Cần
Tho"))
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(ctu, 10f))
    }
}
```

Utils

```
object Utils {
    const val EXTRA_USER_MAP = "EXTRA_USER_MAP"
    const val EXTRA_MAP_TITLE = "EXTRA_MAP_TITLE"
}
```

Place

```
data class Place(
    val title: String,
    val description: String,
    val latitude: Double,
    val longitude: Double
):Serializable
```

UserMap

```
data class UserMap(
    val title: String,
    val places: List<Place>
): Serializable
```

DisplayMapActivity

```
private const val TAG = "DisplayMapActivity"
class DisplayMapActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var binding: ActivityDisplayMapBinding
    private lateinit var userMap: UserMap
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityDisplayMapBinding.inflate(layoutInflater)
        setContentView(binding.root)

        userMap = intent.getSerializableExtra(Utils.EXTRA_USER_MAP) as
UserMap
        supportActionBar?.title = userMap.title
        // Obtain the SupportMapFragment and get notified when the map is
ready to be used.
        val mapFragment = supportFragmentManager
            .findFragmentById(R.id.map) as SupportMapFragment
```

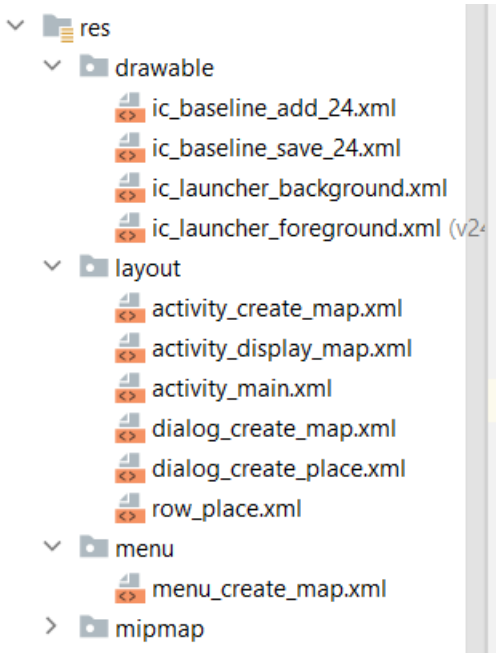
```
        mapFragment.getMapAsync(this)
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the
camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will
be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered
once the user has
     * installed Google Play services and returned to the app.
     */
    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap
        Log.i(TAG, "map: ${userMap.title}")
        val boundsBuilder = LatLngBounds.builder()
        for(place in userMap.places){
            val latLng = LatLng(place.latitude, place.longitude)
            boundsBuilder.include(latLng)

mMap.addMarker(MarkerOptions().position(latLng).title(place.title).snippet(pl
ace.description))
        }

mMap.animateCamera(CameraUpdateFactory.newLatLngBounds(boundsBuilder.build(),
1000, 1000, 0))
//
mMap.moveCamera(CameraUpdateFactory.newLatLngBounds(boundsBuilder.build(),
1000, 1000, 0))
//        val ctu = LatLng(10.031452976258134, 105.77197889530333)
//        mMap.addMarker(MarkerOptions().position(ctu).title("Trường ĐH Cần
Tho"))
//        mMap.moveCamera(CameraUpdateFactory.newLatLng(ctu))
    }
}
```

MapsAdapter

```
private const val TAG = "MapsAdapter"
class MapsAdapter(val context: Context, val userMaps: List<UserMap>, val
onClickListener: OnClickListener) :
RecyclerView.Adapter<MapsAdapter.MyViewHolder>() {

    interface OnClickListener {
        fun onItemClick(position: Int)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
MyViewHolder {
        val view = LayoutInflater.from(context).inflate(R.layout.row_place ,
parent, false)
        return MyViewHolder(view)
```

```
    }

    override fun onBindViewHolder(holder: MyViewHolder, position: Int) {
        val userMap = userMaps[position]
        val tvTitle = holder.itemView.findViewById<TextView>(R.id.tv_place)
        tvTitle.text = userMap.title
        holder.itemView.setOnClickListener {
            Log.i(TAG, "Click on position $position")
            onClickListener.onItemClick(position)
        }
    }

    override fun getItemCount():Int = userMaps.size

    class MyViewHolder(itemView: View):RecyclerView.ViewHolder(itemView){
    }
}
```

Giao diện



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rvMaps"
```

```
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="1dp" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/floatingActionButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="32dp"
        android:clickable="true"
        android:src="@drawable/ic_baseline_add_24"
        android:tint="@color/white"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:tint="@color/white"
        tools:ignore="SpeakableTextPresentCheck" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

dialog_create_map.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingStart="20dp"
    android:paddingEnd="20dp">

    <EditText
        android:id="@+id/et_title_map"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Title"
        android:inputType="textPersonName" />
</LinearLayout>
```

diaglog_create_place.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingStart="20dp"
    android:paddingEnd="20dp">

    <EditText
        android:id="@+id/et_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
        android:ems="10"
        android:hint="Title"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/et_description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Description"
        android:inputType="textPersonName"
        android:minHeight="48dp" />
</LinearLayout>
```

row_place.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="wrap_content"
    android:padding="10sp"
    android:id="@+id/row_layout"
    >
    <TextView
        android:id="@+id/tv_place"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Đại học Cần Thơ"
        android:textSize="18sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginStart="10dp"
        />
</androidx.constraintlayout.widget.ConstraintLayout>
```

menu_create_map.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/miSave"
        android:icon="@drawable/ic baseline save 24"
        android:iconTint="@color/white"
        android:title="Save"
        app:showAsAction="always" />
</menu>
```