به نام خدا



دانشکده کامپیوتر

شبکه تلفن همراه

گزارش کار پروژه اسمیم

استاد : دکتر دیانت

تاریخ :1403/04/15

تهیه کننده : ریحانه هاشم زاده و یاسمن توکلی

مقدمه

در این پروژه با استفاده از دسترسی به موقعیت مکانی کاربر gps و سیگنال دریافتی کاربر و اندازه گیری مکرر انها برای یک سلول ( که با داشتن شناسه سلول ها انها را جدا میکنیم) موقعیت مکانی سلول را تخمین میزنیم. که میتوان از الگوریتم های متفاوتی استفاده کرد که در اینجا معرفی میکنیم.

1. روش مثلث‌سازی (Triangulation)

این روش نیاز به سه یا بیشتر سلول مختلف دارد تا بتواند موقعیت کاربر را با دقت مناسبی تعیین کند. مراحل این روش عبارتند از:

اندازه‌گیری قدرت سیگنال: قدرت سیگنال دریافتی از هر سلول اندازه‌گیری می‌شود.

تعیین فاصله: با استفاده از مدل تضعیف سیگنال، فاصله کاربر از هر سلول محاسبه می‌شود.

محاسبه موقعیت: با استفاده از مثلث‌سازی، نقاط تلاقی بین دایره‌های فرضی حول هر سلول که نشان‌دهنده فاصله تا کاربر است، موقعیت تقریبی کاربر را تعیین می‌کنند.

2. روش سه‌بعدی یا تریلاتریشن (Trilateration)

این روش مشابه مثلث‌سازی است اما برای کار کردن نیاز به سه سلول دارد و از هندسه سه‌بعدی برای تعیین موقعیت استفاده می‌کند.

3. استفاده از قدرت سیگنال دریافت شده (RSSI)

قدرت سیگنال دریافت شده (RSSI) از سلول‌های مختلف به عنوان یک معیار برای تخمین فاصله کاربر از هر سلول استفاده می‌شود. هرچند این روش ممکن است به دقت بالایی نرسد، اما می‌تواند یک تخمین اولیه از موقعیت کاربر بدهد.

4. روش‌های پیشرفته‌تر

Fingerprinting: این روش نیاز به یک پایگاه داده از سیگنال‌های ثبت شده در نقاط مختلف دارد. با مقایسه سیگنال دریافتی کاربر با این پایگاه داده، می‌توان موقعیت کاربر را تعیین کرد.

Time of Arrival (ToA): در این روش، زمان رسیدن سیگنال به سلول‌ها اندازه‌گیری می‌شود و بر اساس تفاوت زمانی بین سلول‌ها، موقعیت کاربر محاسبه می‌شود.

Angle of Arrival (AoA): در این روش، زاویه ورود سیگنال به سلول اندازه‌گیری شده و با استفاده از این زوایا و مکان‌های سلول‌ها، موقعیت کاربر تعیین می‌شود.

روش معرفی شده در صورت مسئله:

برای تعیین موقعیت مکانی کاربر با استفاده از سیگنال دریافتی از سلول متصل شده، فرآیند موقعیت‌یابی معمولاً به دو مرحله تقسیم می‌شود:

تخمین برد یا زاویه ورود سیگنال: در این مرحله، با استفاده از سیگنال دریافتی از ایستگاه پایه، تخمین‌هایی از فاصله یا زاویه ورود سیگنال به دست می‌آید.

تخمین موقعیت بر اساس قواعد هندسی: در این مرحله، از قواعد هندسی یا الگوریتم‌های خاصی استفاده می‌شود تا موقعیت دقیق کاربر تخمین زده شود.

روش‌های اصلی تخمین موقعیت که استفاده می‌شوند عبارتند از:

پهلوبندی دایره‌ای (Circular Positioning)

پهلوبندی هذلولی (Hyperbolic Positioning)

زاویهبندی (Angulation)

1. پهلوبندی دایره‌ای (Circular Positioning)

در این روش، با استفاده از اندازه‌گیری فاصله کاربر از چندین ایستگاه پایه، دایره‌هایی حول هر ایستگاه پایه با شعاعی برابر با فاصله تخمینی ترسیم می‌شود. محل تلاقی این دایره‌ها موقعیت تخمینی کاربر را نشان می‌دهد.

مراحل:

اندازه‌گیری قدرت سیگنال: قدرت سیگنال دریافتی از هر ایستگاه پایه اندازه‌گیری می‌شود.

محاسبه فاصله: فاصله هر ایستگاه پایه تا کاربر بر اساس مدل تضعیف سیگنال محاسبه می‌شود.

تعیین موقعیت: دایره‌هایی با مرکز ایستگاه پایه و شعاع فاصله تخمینی ترسیم می‌شود. محل تلاقی این دایره‌ها موقعیت کاربر را نشان می‌دهد.

2. پهلوبندی هذلولی (Hyperbolic Positioning)

این روش بر اساس اختلاف زمان رسیدن سیگنال‌ها به چندین ایستگاه پایه (TDOA) کار می‌کند. این اختلاف زمان باعث ایجاد خطوط هذلولی می‌شود و محل تلاقی این خطوط موقعیت کاربر را تعیین می‌کند.

مراحل:

اندازه‌گیری اختلاف زمان رسیدن سیگنال (TDOA): زمان رسیدن سیگنال به هر ایستگاه پایه اندازه‌گیری می‌شود.

تعیین خطوط هذلولی: بر اساس اختلاف زمان، خطوط هذلولی ترسیم می‌شود.

تخمین موقعیت: محل تلاقی خطوط هذلولی موقعیت کاربر را نشان می‌دهد.

3. زاویهبندی (Angulation)

در این روش، زاویه ورود سیگنال به ایستگاه‌های پایه (AoA) اندازه‌گیری می‌شود و بر اساس این زوایا، موقعیت کاربر تخمین زده می‌شود.

مراحل:

اندازه‌گیری زاویه ورود سیگنال (AoA): زاویه ورود سیگنال به هر ایستگاه پایه اندازه‌گیری می‌شود.

ترسیم خطوط زاویه‌ای: خطوطی با زاویه‌های اندازه‌گیری شده از هر ایستگاه پایه ترسیم می‌شود.

تخمین موقعیت: محل تلاقی این خطوط زاویه‌ای موقعیت کاربر را نشان می‌دهد.

**توضیح پروژه**

**بخش اول : dependency**

فایل Gradle نقش مدیریت وابستگی‌ها (dependencies) ایفا می‌کند. وابستگی‌ها به کتابخانه‌ها، پلاگین‌ها یا فایل‌هایی اشاره دارند که پروژه برای عملکرد صحیح به آن‌ها نیاز دارد. این وابستگی‌ها می‌توانند شامل کتابخانه‌های استاندارد، کتابخانه‌های شخص ثالث یا حتی ماژول‌های داخلی پروژه باشند.

که ما از این ها وابستگی ها استفاده کردیم:

**AndroidX**:

implementation "androidx.core:core-ktx:1.13.1"

implementation "androidx.appcompat:appcompat:1.7.0"

implementation "com.google.android.material:material:1.12.0"

implementation "androidx.constraintlayout:constraintlayout:2.1.4"

implementation "androidx.lifecycle:lifecycle-runtime-ktx:2.6.1"

**Compose :**

implementation platform("androidx.compose:compose-bom:2023.08.00")

implementation "androidx.compose.ui:ui"

implementation "androidx.compose.ui:ui-graphics"

implementation "androidx.compose.ui:ui-tooling-preview"

implementation "androidx.compose.material3:material3"

implementation "androidx.activity:activity-compose:1.7.0"

**Room :**

implementation "androidx.room:room-runtime:2.5.0"

implementation "androidx.room:room-ktx:2.5.0"

kapt "androidx.room:room-compiler:2.6.1"

**Testing :**

testImplementation "junit:junit:4.13.2"

androidTestImplementation "androidx.test.ext:junit:1.1.5"

androidTestImplementation "androidx.test.espresso:espresso-core:3.6.1"

androidTestImplementation platform("androidx.compose:compose-bom:2023.08.00")

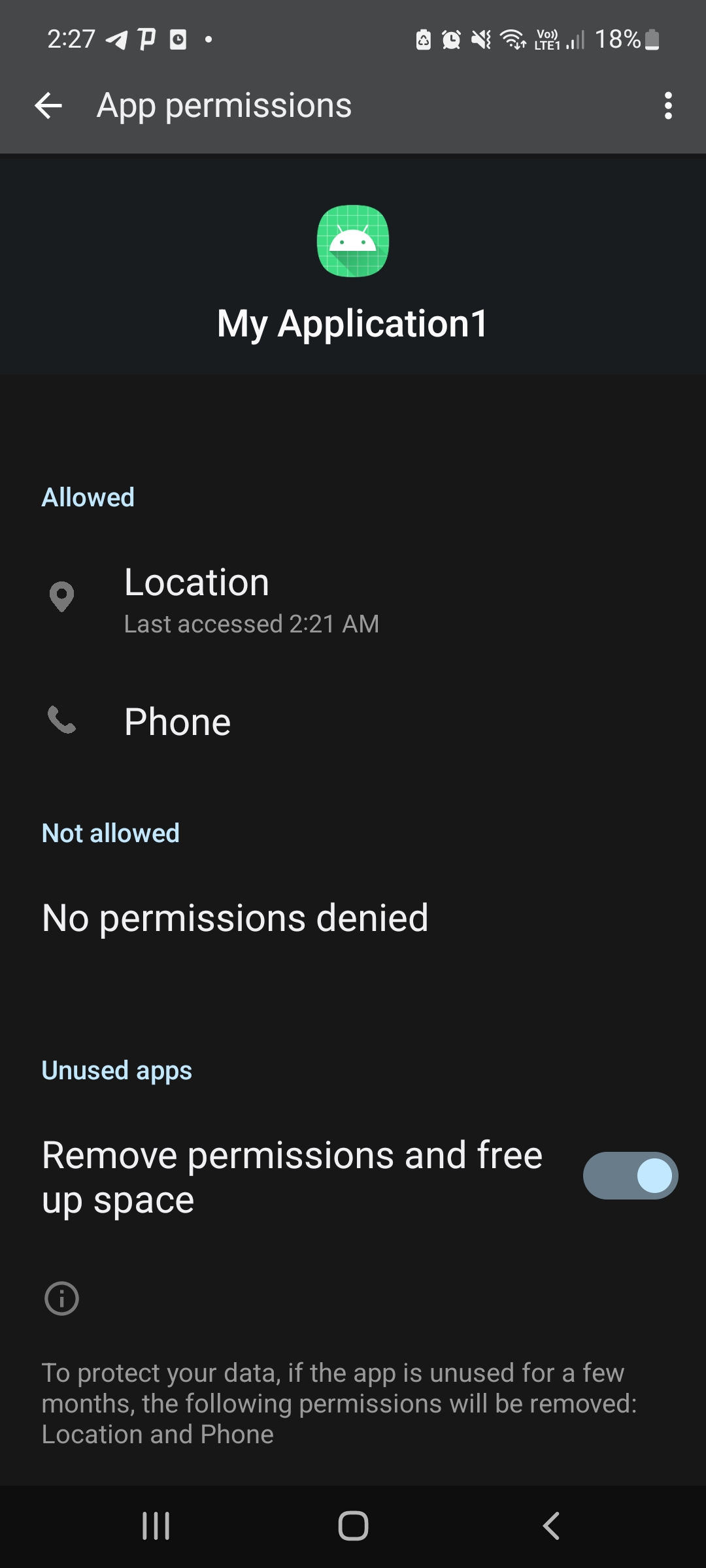
androidTestImplementation "androidx.compose.ui:ui-test-junit4

**بخش دوم:permisions**

برای احرای کد نیاز داریم یسری اجازه دسترسی را فعال کنیم. که درقسمت androidmanifest قرار دارد.

که ما برای اجازه دسترسی به مکان کاربر و وضعیت موبایل کاربر از دسترات زیر استغاده کردیم:

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />



**بخش سوم:katlin code**

برای هر صفحه که دیزاین میکنیم در این قسمت منطق برنامه را با کد کاتلین پیاده سازی میکنیم که سه صفحه داریم:

Main:

در این بخش ما یک activity جدید به نام MapsActivity.ks ایجاد می‌کنیم.

یک فایل activity\_maps.xml هم برای فرانت آن داریم.

توسعه‌ی کلاس با استفاده از تابع onCreate:

این تابع زمانی که فعالیت ایجاد می‌شود، فراخوانی می‌گردد.

تنظیمات اولیه‌ی رابط کاربری:

binding = ActivityMapsBinding.inflate(layoutInflater):

این خط برای ملزم‌کردن ویوهای موجود در فایل XML به کلاس فعالیت استفاده می‌شود.

setContentView(binding.root):

این خط نمای اصلی فعالیت را تنظیم می‌کند تا از ویوهای ملزم شده استفاده کند.

ابتدایی‌سازی FusedLocationProviderClient:

این کد از FusedLocationProviderClient برای دسترسی به موقعیت مکانی دستگاه استفاده می‌کند.

val mapFragment = supportFragmentManager.findFragmentById(R.id.map) as SupportMapFragment:

این خط SupportMapFragment را از فایل XML پیدا می‌کند.

mapFragment.getMapAsync(this):

این خط درخواست می‌دهد تا نقشه به صورت ناهمزمان (asynchronously) بارگذاری شود و زمانی که نقشه آماده استفاده شد، فراخوانی شود.

Maps: در این قسمت برای انتخاب نقشه گزینه های مختلفی را امتحان کردیم مانند map.ir , osmadnroid , google map که google map بهترین انتخاب بود.

1. کتابخانه

import android.Manifest

import android.content.pm.PackageManager

import android.location.Location

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import androidx.core.app.ActivityCompat

import com.google.android.gms.maps.CameraUpdateFactory

import com.google.android.gms.maps.GoogleMap

import com.google.android.gms.maps.OnMapReadyCallback

import com.google.android.gms.maps.SupportMapFragment

import com.google.android.gms.maps.model.LatLng

import com.google.android.gms.maps.model.MarkerOptions

import com.example.test.databinding.ActivityMapsBinding

import com.google.android.gms.location.FusedLocationProviderClient

import com.google.android.gms.location.LocationServices

در این بخش، بسته‌بندی پروژه تعریف شده و کتابخانه‌های مورد نیاز برای برنامه وارد شده‌اند، از جمله کتابخانه‌های مربوط به گوگل مپس و مکان‌یابی.

2. تعریف کلاس MapsActivity

این کلاس از AppCompatActivity ارث‌بری کرده و از OnMapReadyCallback برای کار با نقشه گوگل استفاده می‌کند.

3. متغیرهای خصوصی

در این بخش، متغیرهایی برای نقشه گوگل، بایندینگ فعالیت و سرویس مکان‌یابی تعریف شده‌اند.

4. متد onCreate

در این بخش، رابط کاربری فعالیت از طریق بایندینگ تنظیم شده و FusedLocationProviderClient برای استفاده از خدمات مکان‌یابی گوگل مقداردهی اولیه می‌شود. همچنین SupportMapFragment برای کار با نقشه گوگل آماده‌سازی می‌شود.

5. متد onMapReady

این متد زمانی که نقشه آماده استفاده است فراخوانی می‌شود. ابتدا بررسی می‌شود که آیا برنامه مجوزهای دسترسی به مکان دقیق و مکان تقریبی را دارد یا خیر. در صورت نداشتن مجوزها، درخواست مجوز داده می‌شود. اگر مجوزها فراهم باشد، مکان‌یابی فعلی کاربر انجام می‌شود و نشانگر (مارکر) بر روی مکان فعلی کاربر در نقشه اضافه می‌شود و دوربین نقشه به آن مکان حرکت می‌کند.

6. متد onRequestPermissionsResult

این متد زمانی که کاربر به درخواست مجوزها پاسخ می‌دهد فراخوانی می‌شود. اگر مجوزها داده شوند، مکان‌یابی کاربر انجام شده و نقشه به روز می‌شود. در غیر این صورت، باید مشخص شود که چه عملیاتی باید انجام شود.

7. همراهی اشیاء (Companion Object)

این بخش یک ثابت برای کد درخواست مجوز مکان تعریف می‌کند تا در بخش‌های مختلف برنامه استفاده شود.

Signal:

1. کتابخانه

import android.os.Bundle

import android.os.Handler

import android.os.Looper

import android.widget.ImageButton

import android.widget.TextView

import androidx.activity.enableEdgeToEdge

import androidx.appcompat.app.AppCompatActivity

import androidx.core.view.ViewCompat

import androidx.core.view.WindowInsetsCompat

import android.Manifest

import android.content.Context

import android.content.pm.PackageManager

import android.location.Location

import android.telephony.\*

import android.util.Log

import androidx.core.app.ActivityCompat

import androidx.core.content.ContextCompat

import com.google.android.gms.location.\*

در این بخش، بسته‌بندی پروژه تعریف شده و کتابخانه‌های مورد نیاز برای برنامه وارد شده‌اند.

**2. تعریف کلاس SignalActivity**

این کلاس از AppCompatActivity ارث‌بری کرده و به عنوان فعالیت اصلی برنامه عمل می‌کند.

**3. متغیرهای خصوصی**

در این بخش، متغیرهای مختلفی تعریف شده‌اند که برای ذخیره‌سازی اطلاعات مربوط به موقعیت جغرافیایی، قدرت سیگنال و اجزای رابط کاربری استفاده می‌شوند.

**4. متد onCreate**

در این بخش، رابط کاربری تنظیم شده و متغیرهای مربوط به اجزای رابط کاربری مقداردهی می‌شوند. همچنین رویداد کلیک دکمه مکان‌یابی تعریف شده و به بررسی مجوزها و به‌روزرسانی مکان و سیگنال می‌پردازد.

**5. بررسی مجوزها**

این بخش مجوزهای لازم برای دسترسی به مکان و اطلاعات سیگنال را بررسی می‌کند و در صورت نیاز درخواست مجوز می‌کند.

**6. به‌روزرسانی‌های مکان**

این بخش به تنظیمات به‌روزرسانی مکان و فراخوانی مکان فعلی کاربر می‌پردازد و موقعیت جغرافیایی کاربر را در TextView نمایش می‌دهد.

**7. دریافت قدرت سیگنال**این

بخش به دریافت و نمایش اطلاعات سیگنال موبایل می‌پردازد. نوع شبکه و قدرت سیگنال را به همراه شناسه سلول مربوطه در TextViewهای مربوطه نمایش می‌دهد.

**8. شروع اندازه‌گیری‌های بلادرنگ**

در این بخش، یک چرخه زمانی برای به‌روزرسانی مکان و سیگنال به صورت بلادرنگ هر 5 ثانیه تعریف شده است.

**بخش چهارم:UI**

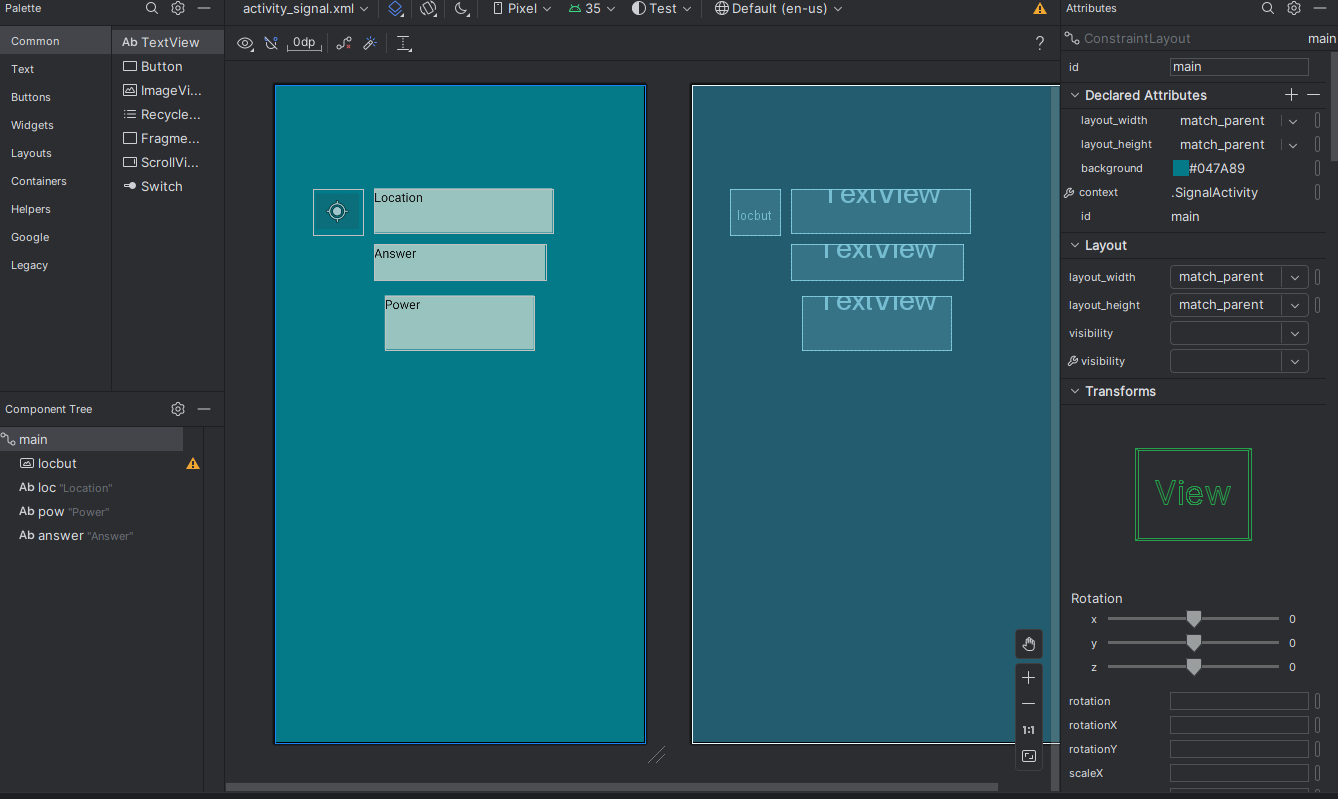
این قسمت برای تنظیم صفحه نمایش اپلیکشن که کاربر با ان در ارتباط هست ، استفاده میشود که به دو روش قابل استفاده هست : روش کد در فایل به فرمت xml که برای هر صفحه و یا در قسمت دیزاین اجزا و تنظیمات را میتوان انجام داد

در قسمت کد پارامتر های هر جز را تنظیم میتوان کرد:

مثال: یکی از اجزا صفحه اطاعات که برای نمایش مختصات مکانی کاربر هست

<TextView  
 android:id="@+id/loc"  
 android:layout\_width="199dp"  
 android:layout\_height="49dp"  
 android:layout\_marginTop="115dp"  
 android:layout\_marginEnd="102dp"  
 android:background="#FF99C3BF"  
 android:text="Location"  
 android:textColor="#141313"  
 android:textColorHighlight="#121111"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />

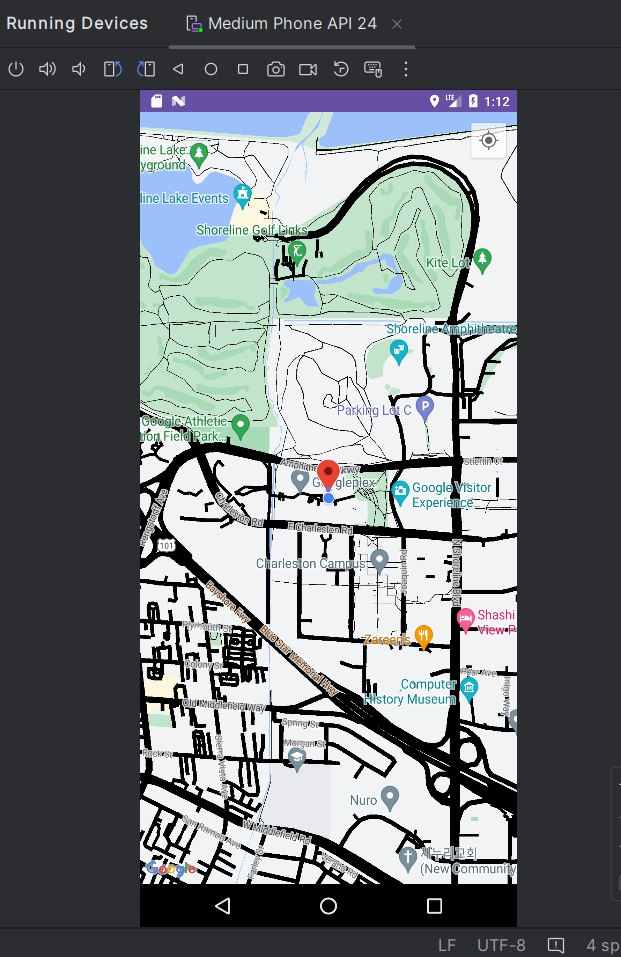
در قسمت دیزان میتوان راحت تر این بخش را مدیریت کرد (مثل موقعیت هر جزء و انتخاب نوع جزء و رنگ و ...)



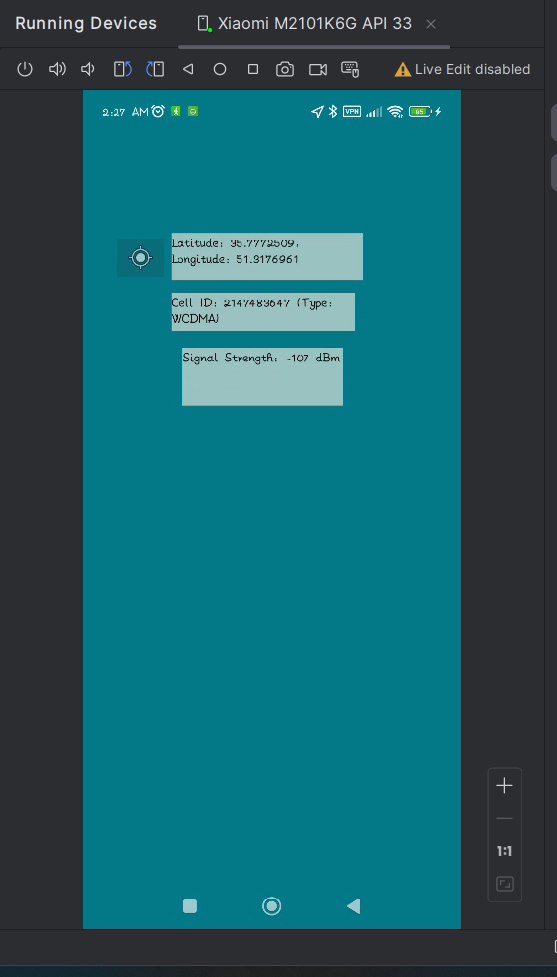
که در این قسمت ما سه صفحه داریم:

Main: صفحه اول که وارد میشویم که منو ورود به نقشه و اطلاعات هست

Maps: نمایش نقشه و موقعیت مکانی کاربر



Signsl:شامل اطلاعات مختصات مکانی کاربر ،سیگنال دریافتی از سلول, شناسه و نوع شبکه(تنولوژی نسل شبکه)

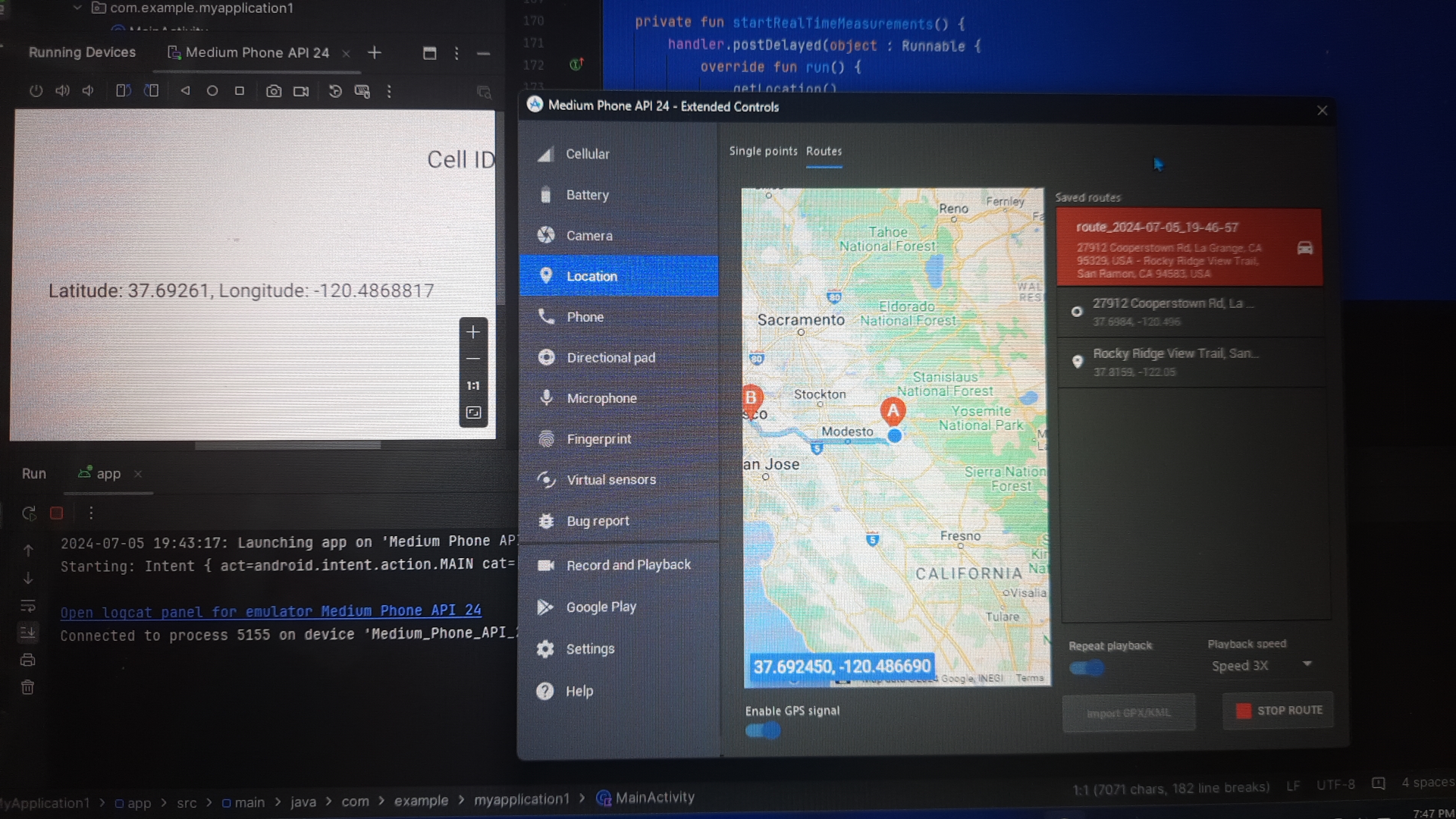


**صفحه چهارم که برای نمایش جدول هست که نمایش تمام سلول ها باشناسه و موقیت مکانی سلول خواهد بود.**

**بخش پنجم:simulate**

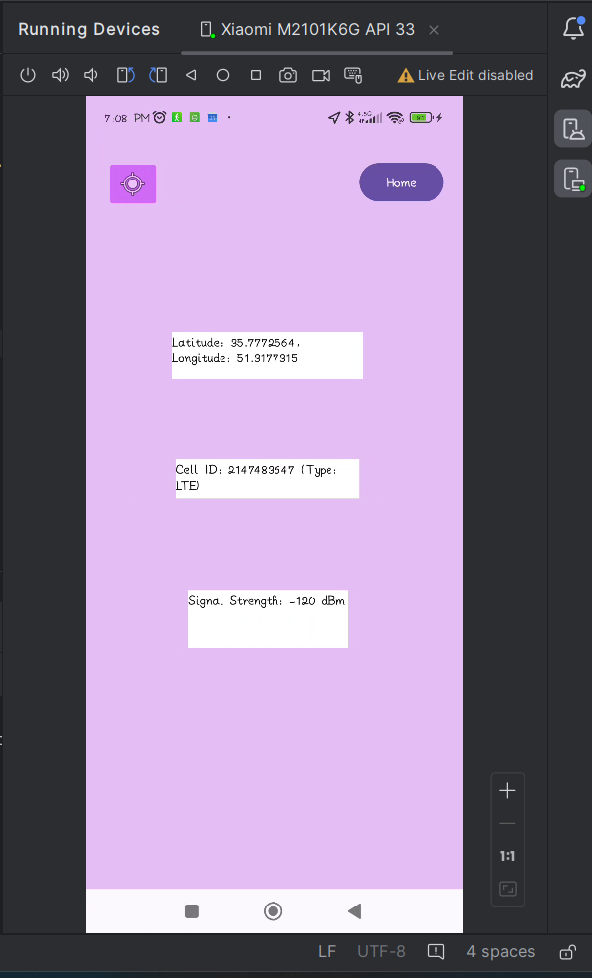
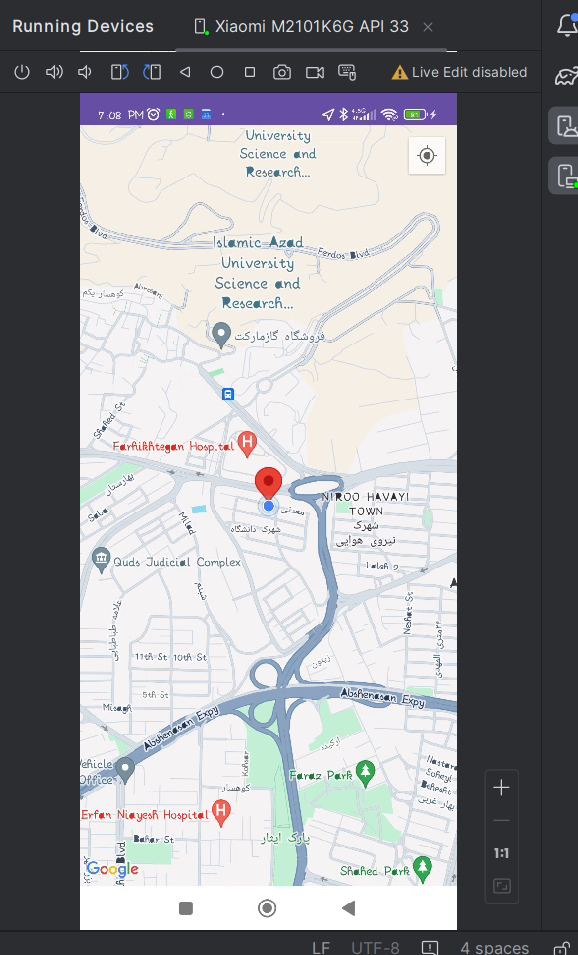
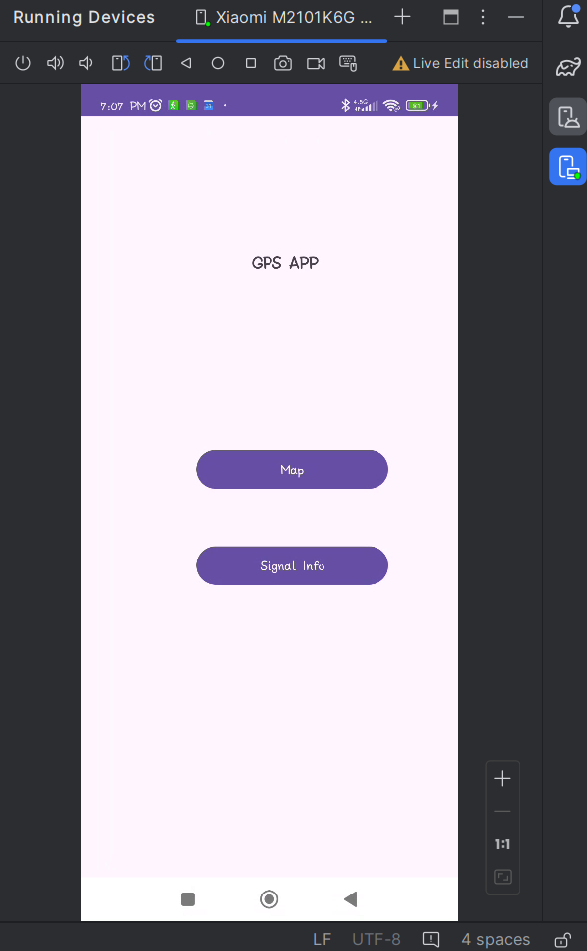
برای اینکه از برنامه ران بگیریک باید ان را در یک سیستم عامل اندریدی شبیه سازی کنیم . دو دوش دارد: استفاده از یک موبایل شبیه سازی شده و یا گوشی واقعی. که ما در اینجا از هز دو استفاده کردیم(چون گوشی شبیه سازی شده سیم کارتی نداشت تا به سلول واقعی وصل شود).

در روش اول ما ار یک گوشی با android 7.0(nougat) استفاده کردیم



روش دوم با گوشی واقعی باید این مراحل را طی کرد:

در تنظیمات گوشی به قسمت about phone به قسمت software info میرویم بعد بر روی build number 7 بار کلیک میکنیم تا dev mode فعال شود و بعد تنظیمات لازم برای وصل شدن از ظریق usb , wifi به android studio وصل میشیم و و از برنامه ران میگیریم:



بخش ششم:database

Room یک کتابخانه پایگاه داده ORM (Object Relational Mapping) است که توسط گوگل برای توسعه برنامه‌های اندرویدی ارائه شده است. این کتابخانه امکان تعامل ساده و کارآمد با SQLite را فراهم می‌کند و به توسعه‌دهندگان کمک می‌کند تا با استفاده از اشیاء جاوا یا کاتلین، داده‌ها را به پایگاه داده وارد یا از آن خارج کنند.

اجزاء اصلی Room: Entity (موجودیت) و DAO (Data Access Object) و Database (پایگاه داده) هستند.

1. Entity (موجودیت)

موجودیت‌ها کلاس‌هایی هستند که جداول پایگاه داده را تعریف می‌کنند. هر موجودیت به یک جدول در پایگاه داده نگاشت می‌شود و فیلدهای آن به ستون‌های جدول مرتبط هستند. برای تعریف یک موجودیت از Annotation @Entity استفاده می‌شود.

2. DAO (Data Access Object)

DAOها رابط‌هایی هستند که روش‌های دسترسی به پایگاه داده را تعریف می‌کنند. هر متد در DAO نشان‌دهنده یک عملیات بر روی پایگاه داده است و با استفاده از Annotationهای مختلف مانند @Insert, @Update, @Delete, و @Query تعریف می‌شوند.

3. Database (پایگاه داده)

کلاس پایگاه داده یک کلاس انتزاعی است که از RoomDatabase ارث‌بری می‌کند و شامل متدهایی برای دسترسی به DAOها می‌شود. برای تعریف این کلاس از Annotation @Database استفاده می‌شود.

ما برای اینکه بتونیم داده های توان دریافتی و مکان کاربر و شناسه سلول مربوظ به هرسلول نگه داری کنیم تا بتوانیم با کمک ان و اجرا الگوریتم مکان سلول را بدست بیاریم نیاز به یک دیتاست داریم تا بتواند دیتا هایی که به صورت realtime و به تعداد زیاد هست را دخیره کنیم که ما از room استفاده کردیم.

CellDatabaseSingleton:

این شیء یک نمونه از پایگاه داده را برای شما فراهم می‌کند. احتمالاً یک کلاس singleton است که دسترسی به پایگاه داده را مدیریت می‌کند. با استفاده از getDatabase(this)، شما یک نمونه از پایگاه داده را دریافت می‌کنید.

CellRepository:

یک مخزن است که واسط بین ViewModel و منابع داده (مانند پایگاه داده) است. احتمالاً شامل توابعی مانند درج و بازیابی داده‌ها از پایگاه داده است. شما از مخزن برای انتقال داده بین ViewModel و دیتابیس استفاده می‌کنید.

ViewModelProvider:

این کلاس ViewModel را از ViewModelProvider به عنوان یک factory ایجاد می‌کند. شما this (یعنی این Activity) و cellInfoViewFactory(repository) را به عنوان پارامترها به ViewModelProvider می‌دهید. فابریکه ViewModel (cellInfoViewFactory) برای ایجاد ViewModel از repository استفاده می‌کند.

cellView:

ViewModelی که از طریق ViewModelProvider ایجاد شده است و شامل توابعی مانند getAllCellInfo() و insert() است که با استفاده از repository به اطلاعات دسترسی دارند.

findViewById:

با استفاده از findViewById<Button>(R.id.Home)، شما یک مرجع به دکمه‌ای که R.id.Home است را دریافت می‌کنید.

setOnClickListener:

با فراخوانی setOnClickListener بر روی دکمه، یک عملکرد کلیک برای دکمه تنظیم می‌شود.

Intent:

با استفاده از کلاس Intent، یک Intent برای انتقال از این Activity به MainActivity ایجاد می‌شود.

startActivity(intent):

با فراخوانی startActivity(intent)، این Intent اجرا می‌شود و کنترل به MainActivity منتقل می‌شود.

CellInfoEntity:

یک کلاس داده است که نشان دهنده اطلاعات سلول (cell information) است. این کلاس احتمالاً به عنوان یک موجودیت (entity) در پایگاه داده تعریف شده است و شامل فیلدهایی مانند cellLocationX, cellLocationY, cellId, و signalStrength می‌باشد.

**currentLatitude** **و currentLongitude**:

متغیرهایی هستند که مختصات مکانی فعلی دستگاه را نگه می‌دارند. این مقادیر معمولاً از GPS یا سرویس‌های مکان‌یابی دیگر گرفته می‌شوند.

currentNodeId:

متغیری که ID سلول فعلی (cell ID) را نگه می‌دارد. این مقدار احتمالاً از سرویس‌های شبکه موبایل یا اطلاعات سلولی دستگاه گرفته می‌شود.

currentSignalStrength:

متغیری که قدرت سیگنال فعلی را نگه می‌دارد. این مقدار نیز از اطلاعات سلولی دستگاه گرفته می‌شود.

مراحل:

ایجاد یک شیء CellInfoEntity:

با استفاده از مقادیر currentLatitude, currentLongitude, currentNodeId, و currentSignalStrength یک شیء CellInfoEntity ایجاد می‌کنید.

درج اطلاعات در پایگاه داده با استفاده از ViewModel:

با فراخوانی cellViewModel.insert(cellInfo)، شیء CellInfoEntity را به ViewModel ارسال می‌کنید تا در پایگاه داده درج شود. ViewModel از Repository برای درج داده‌ها در پایگاه داده استفاده می‌کند.