



دانشکده مهندسی کامپیوتر

استاد درس: دکتر ابوالفضل دیانت

بهار ۱۴۰۳

## پروژه Rhodium

درس شبکه های تلفن همراه  
گزارش پروژه

هلیا وفایی - نگین حقیقی

۹۹۵۲۲۳۴۷-۹۹۵۲۱۲۲۶



## ۱ معرفی پروژه

هدف این پروژه، شناسایی و ثبت نقاط کور در پوشش شبکه های تلفن همراه در محیط های درون بنا با استفاده از یک برنامه تحت سیستم عامل Android است. این برنامه به کاربر امکان می دهد نقشه ای از محیط یا ساختمان را وارد کرده و مسیر حرکت خود را بر روی نقشه نمایش دهد. در طول حرکت، برنامه از سنسورهای مختلف گوشی برای نمایش مسیر دقیق کاربر بر روی نقشه استفاده می کند.

### ۱.۱ امکانات برنامه

وارد کردن نقشه محیط: کاربر می تواند نقشه محیط یا ساختمان را به برنامه وارد کند تا بر روی صفحه گوشی نمایش داده شود.

ثبت نقاط: کاربر با کلیک بر روی نقشه می تواند نقاط موردنظر را مشخص و داده های مربوط به آن ها را ثبت کند. نمایش مسیر حرکت: برنامه با استفاده از سنسورهای گوشی مسیر حرکت کاربر را بر روی نقشه نمایش می دهد و در صورت مشاهده انحراف، کاربر می تواند نقطه صحیح را مجدداً مشخص کند.

اندازه گیری پارامترهای شبکه: برنامه به صورت پیوسته پارامترهای مرتبط با توان دریافتی را برای سلول خدمتگذار و سلول های همسایه اندازه گیری و ثبت می کند. این پارامترها شامل  $CINR$ ،  $RSRQ$ ،  $RSRP$ ،  $RxLev$ ،  $C1$ ،  $C2$  برای GSM و  $RSCP$ ،  $EC/N$ ،  $0$ ،  $UMTS$  و  $LTE$  می باشد.

ثبت اطلاعات در پایگاه داده: مقادیر اندازه گیری شده به همراه مکان و پارامترهای مربوط به سلول خدمتگذار نظیر  $PLMN$ ،  $LAC$ ،  $RAC$ ،  $TAC$  و شناسه سلول در پایگاه داده ثبت می شوند.

نمایش نقشه به صورت برخط و برون خط: کاربر می تواند داده های ذخیره شده را بر روی نقشه به صورت برخط یا برون خط مشاهده کند. نقاط مختلف بر اساس توان دریافتی سلول با رنگ های مختلف (سبز، زرد، نارنجی، قرمز، سیاه) نمایش داده می شوند که رنگ سیاه به معنای عدم موفقیت در اندازه گیری است.

### ۲.۱ نکات کلیدی

استفاده بهینه از مفاهیم پایگاه داده برای طراحی مناسب و کارآمد.

تلاش برای استخراج حداکثری پارامترهای مرتبط با توان دریافتی برای نسل های مختلف شبکه ها ( $LTE$ ) ( $GSM$ ،  $UMTS$ ) بر روی نسخه های مختلف Android.

این پروژه با هدف بهبود پوشش شبکه های تلفن همراه و شناسایی نقاط کور در محیط های درون بنا طراحی شده است و به کاربران کمک می کند تا مشکلات پوشش شبکه را به صورت دقیق تر شناسایی و رفع کنند.

## ۲ بررسی فایل MainActivity.kt

هدف: فایل `MainActivity.kt` به عنوان نقطه ورود اصلی برای برنامه عمل می کند. این فایل مسئولیت راه اندازی رابط کاربری، مدیریت دسترسی ها و شروع سرویس هایی که اطلاعات شبکه و مکان را جمع آوری و نمایش می دهند، بر عهده دارد.

عناصر رابط کاربری: پیدا کردن و مقداردهی به اجزای مختلف رابط کاربری مانند `TextView` و `Button`.

پایگاه داده: مقداردهی اولیه `NetworkInfoDatabaseHelper` برای تعامل با پایگاه داده محلی.

مدیریت دسترسی ها:



بررسی و درخواست دسترسی های لازم برای مکان و وضعیت تلفن. اگر دسترسی ها داده شد، وظایف و سرویس های پس زمینه شروع می شوند.  
اجرای وظایف دوره ای:  
شی Runnable: به صورت دوره ای اطلاعات مکان و شبکه را با استفاده از LocationHelper و NetworkHelper به روزرسانی می کند. Handler: زمان بندی این به روزرسانی ها را مدیریت می کند و هر ثانیه یک بار آن ها را اجرا می کند.  
لیسنرهای دکمه:  
دکمه باز کردن نقشه: یک فعالیت جدید (MapsActivity) را برای نمایش نقشه باز می کند. دکمه توقف ورود اطلاعات: حالت ورود داده ها را تغییر می دهد و وظایف دوره ای را متوقف یا ادامه می دهد.  
مدیریت نتایج دسترسی:  
نتایج درخواست دسترسی ها را مدیریت می کند و در صورت اعطای دسترسی ها سرویس ها را شروع می کند.  
شروع و توقف سرویس ها:  
startServicesAndTasks(): وظایف دوره ای و هر سرویس اضافی مورد نیاز برای جمع آوری داده ها را شروع می کند. onDestroy(): اطمینان حاصل می کند که هنگام از بین رفتن فعالیت، handler تمام وظایف را متوقف می کند.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

شکل ۱: permissions uses

### ۳ بررسی فایل LocationHelper.kt

هدف: فایل LocationHelper.kt به منظور مدیریت مکان یابی در برنامه استفاده می شود. این فایل وظایف مربوط به دریافت آخرین مکان کاربر، درخواست به روزرسانی های مکان و نمایش آن ها در رابط کاربری را بر عهده دارد.  
دریافت FusedLocationProviderClient:  
getFusedLocationProviderClient: یک شیء FusedLocationProviderClient ایجاد می کند که برای تعامل با سرویس های مکان Google استفاده می شود.  
دریافت آخرین مکان:  
getLastLocation: تلاش می کند آخرین مکان کاربر را دریافت کند. اگر مکان موجود نباشد یا با خطا مواجه شود، به requestLocationUpdates برای دریافت به روزرسانی های مکان فراخوانی می کند.  
درخواست به روزرسانی های مکان:



`requestLocationUpdates`: در صورتی که آخرین مکان در دسترس نباشد، درخواست به روزرسانی های مکان با استفاده از `LocationRequest` و `LocationCallback` را ارسال می کند تا مکان کاربر را دریافت کند.

به روزرسانی رابط کاربری با اطلاعات مکان:

`updateLocationUI`: اطلاعات مکان را در های `TextView` رابط کاربری به روزرسانی می کند. همچنین زمان وقوع رویداد را با استفاده از `SimpleDateFormat` نمایش می دهد.

بررسی و درخواست دسترسی های مکان:

`checkLocationPermission`: بررسی می کند که آیا دسترسی های مکان کاربر اعطا شده اند یا خیر. در صورت نیاز، درخواست دسترسی های لازم را ارسال می کند.

## ۴ بررسی فایل `MapActivity.kt`

هدف: فایل `MapActivity.kt` برای نمایش مکان های شبکه و کیفیت سیگنال ها بر روی نقشه گوگل استفاده می شود. این فایل وظایف مربوط به بارگذاری نقشه، اضافه کردن مارکرها و خطوط پلی لاین بر اساس داده های ذخیره شده در دیتابیس و سفارشی سازی پنجره های اطلاعات مارکر را بر عهده دارد.

راه اندازی اولیه نقشه:

`onCreate`: تنظیمات اولیه برای فعالیت و نقشه را انجام می دهد. از طریق `SupportMapFragment`، نقشه گوگل را بارگذاری می کند و دیتابیس را مقداردهی اولیه می کند. بارگذاری داده ها بر روی نقشه:

`onMapReady`: پس از آماده شدن نقشه، داده های مکان و شبکه را از دیتابیس دریافت کرده و مارکرها را بر اساس این داده ها بر روی نقشه اضافه می کند. همچنین خطوط پلی لاین برای نمایش مسیر حرکت کاربر بر اساس مکان های ثبت شده ترسیم می شود. سفارشی سازی مارکرها و پنجره های اطلاعات:

`getInfoContents`: پنجره اطلاعات مارکر را سفارشی سازی می کند تا جزئیات بیشتری مانند مکان، `ID Cell TAC`، `LAC`، `ID`، `PLMN` و قدرت سیگنال نمایش داده شود. `setOnMarkerClickListener`: برای نمایش پنجره اطلاعات مارکر هنگام کلیک بر روی آن استفاده می شود.

تعیین رنگ مارکر بر اساس وضعیت سیگنال:

`getColorForSituation`: برای هر وضعیت سیگنال (بسیار ضعیف، ضعیف، معمولی، خوب، عالی) یک رنگ خاص تعیین می کند تا کاربران بتوانند کیفیت سیگنال را به راحتی تشخیص دهند. این فایل به کاربر کمک می کند تا با استفاده از نقشه گوگل، مکان های مختلف شبکه را مشاهده کند و اطلاعات دقیق مربوط به کیفیت سیگنال و سایر جزئیات شبکه را بررسی نماید.

```
private fun getColorForSituation(situation: String): Float {  
    return when (situation.toLowerCase()) {  
        "very poor" -> BitmapDescriptorFactory.HUE_BLACK  
        "poor" -> BitmapDescriptorFactory.HUE_RED  
        "fair" -> BitmapDescriptorFactory.HUE_ORANGE  
        "good" -> BitmapDescriptorFactory.HUE_YELLOW  
        "excellent" -> BitmapDescriptorFactory.HUE_GREEN  
        else -> BitmapDescriptorFactory.HUE_VIOLET  
    }  
}
```

شکل ۲: code colors

## ۵ بررسی فایل NetworkHelper.kt

هدف: فایل NetworkHelper.kt به منظور دریافت و نمایش اطلاعات شبکه و سیگنال های موبایل در یک اپلیکیشن اندرویدی استفاده می شود. این فایل اطلاعات مربوط به فناوری سلولی، مکان سلول و کیفیت سیگنال را جمع آوری کرده و در رابط کاربری نمایش می دهد. همچنین این اطلاعات را در پایگاه داده ذخیره می کند.

دریافت و نمایش اطلاعات شبکه:

displayNetworkInfo: این تابع اطلاعات مربوط به شبکه و سیگنال ها را از TelephonyManager دریافت کرده و در های TextView مربوطه نمایش می دهد. این اطلاعات شامل نوع فناوری سلولی، مکان سلول و کیفیت سیگنال می باشد.

استخراج اطلاعات سلول:

CellInfoGsm، CellInfoWcdma، CellInfoLte، ID Cell TAC، LAC، ID، PLMN و قدرت سیگنال استخراج می شود و در متغیرهای مربوطه ذخیره می گردد.

ذخیره اطلاعات در پایگاه داده:

saveNetworkInfoToDatabase: اطلاعات دریافت شده را به همراه زمان رویداد، مختصات جغرافیایی و کیفیت سیگنال به پایگاه داده اضافه می کند. این اطلاعات در قالب یک شی NetworkInfo ذخیره می شوند.

تشخیص نوع فناوری سلولی:

getCellTechnology: نوع فناوری شبکه (مانند ۴G، ۳G، ۲G) را بر اساس نوع شبکه TelephonyManager تشخیص داده و برمی گرداند.

محاسبه کیفیت سیگنال:

calculateSignalQuality: تابعی برای محاسبه کیفیت سیگنال بر اساس مقادیر دریافتی مانند قدرت سیگنال، RSRP، RSRQ و غیره. (در این مثال، کیفیت سیگنال به صورت ثابت "Good" برگردانده می شود). این فایل به کاربر کمک می کند تا اطلاعات دقیقی از وضعیت شبکه و سیگنال موبایل خود داشته باشد و این اطلاعات را در قالبی منسجم ذخیره و نمایش دهد.

## ۶ بررسی فایل DatabaseHelper.kt

هدف: فایل DatabaseHelper.kt برای ایجاد و مدیریت پایگاه داده SQLite به منظور ذخیره سازی و بازیابی اطلاعات شبکه موبایل در اپلیکیشن اندرویدی مورد استفاده قرار می گیرد.

تعریف کلاس: NetworkInfo

این کلاس داده های مورد نیاز برای ذخیره سازی اطلاعات شبکه را تعریف می کند، شامل زمان رویداد، مختصات جغرافیایی، فناوری سلولی، مشخصه های سیگنال مانند قدرت سیگنال، RSRP RSRQ و غیره، و وضعیت فعلی.

کلاس: NetworkInfoDatabaseHelper

این کلاس از SQLiteOpenHelper ارث بری کرده است و به منظور ایجاد و به روز رسانی پایگاه داده SQLite استفاده می شود. توابع onCreate و onUpgrade برای ساخت جدول اطلاعات شبکه و به روز رسانی آن تعریف شده اند. onCreate: جدول cellularinfo با ستون های مختلفی برای ذخیره اطلاعات شبکه ایجاد می کند.

onUpgrade: در صورت وجود نسخه قدیمی تر پایگاه داده، جدول را حذف و دوباره ایجاد می کند. توابع مدیریت داده:

insertInfo: برای درج یک شی NetworkInfo در پایگاه داده SQLite استفاده می شود. اطلاعات از طریق ContentValues درون جدول cellularinfo درج می شوند.

getAllInfo: برای دریافت تمام اطلاعات موجود در جدول cellularinfo به صورت لیست از شی های NetworkInfo استفاده می شود.

توابع کمکی:

insertNetworkInfoToDatabase: تابعی که اطلاعات مختصات جغرافیایی، فناوری سلولی و مشخصه های سیگنال را به همراه وضعیت فعلی در پایگاه داده درج می کند. calculateSituation: بر اساس قدرت سیگنال، وضعیت فعلی را محاسبه و برمی گرداند.

getSituationColor: بر اساس وضعیت فعلی، رنگ مربوطه برای نمایش گرافیکی در نقشه را باز می گرداند.

getCellTechnology: بر اساس نوع شبکه موجود، فناوری سلولی را تشخیص می دهد و برمی گرداند.

این فایل به کمک کلاس SQLiteOpenHelper و توابع تعریف شده، به اپلیکیشن اندرویدی کمک می کند تا اطلاعات شبکه موبایل را جمع آوری، ذخیره و نمایش دهد. این اطلاعات برای ارزیابی و نمایش وضعیت شبکه به کاربران ارائه می شود و همچنین برای تجزیه و تحلیل وضعیت شبکه در طول زمان مورد استفاده قرار می گیرد.