



دانشکده مهندسی کامپیوتر

پروژه Esmium

شبکه‌های تلفن همراه

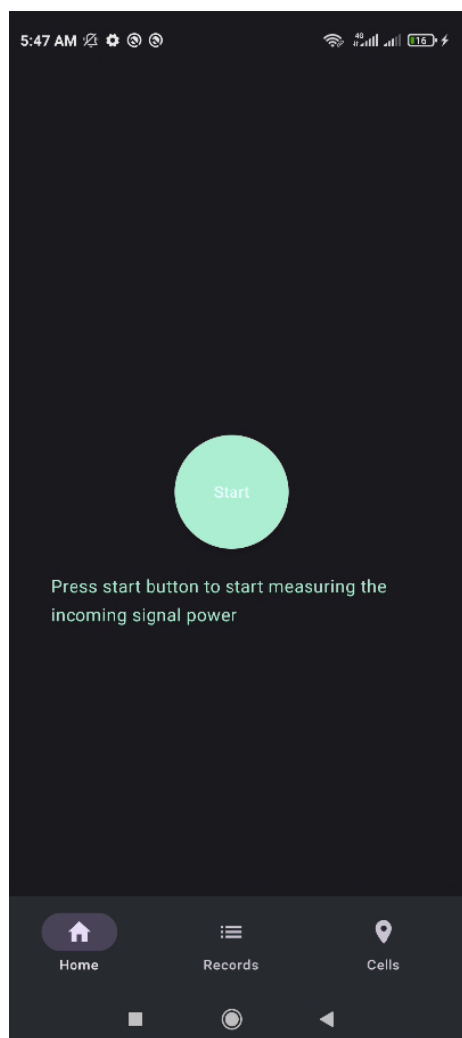
استاد درس: دکتر دیانت

نیم‌سال دوم

سال تحصیلی ۱۴۰۱-۱۴۰۲

مقدمه:

در این پروژه با استفاده از مکان کاربر و توان سیگنال دریافتی توسط UE سعی شده است تا به مختصات نسبی آنتن متصل شده دست پیدا کرد. اپلیکشن نوشته شده به زبان کاتلین است و انتظار می‌رود بر روی دستگاه‌های اندرویدی با ورژن ۹ به بالا قابل اجرا باشد.



شکل ۱: نمای کلی برنامه

مکان‌یابی:

جهت تعیین مکان آنتن مورد نظر، اول نیاز به مکان کاربر متصل داریم. برای این امر ابتدا نیاز داریم که لوکیشن کاربر فعال باشد و سپس دسترسی مورد نظر نیز به اپلیکیشن داده شود. برای مدیریت منطق‌ها و رول‌هایی مربوط به مکان کلاسی تحت عنوان `locationService` نوشته شده است.

```

۱ public fun isGpsEnabled(): Boolean {
۲     val locationManager: LocationManager =
۳         this.context.getSystemService(Context.LOCATION_SERVICE) as LocationManager
۴     return LocationManagerCompat.isLocationEnabled(locationManager)
۵ }

```

نمونه کد ۱: مشخص کردن وضعیت `location`

تابع زیر جهت درخواست دسترسی از کاربر هنگامی که دسترسی وجود نداشته باشد استفاده می‌شود.

```

۱ public fun getPermission() {
۲     ActivityCompat.requestPermissions(
۳         this.activity,
۴         arrayOf(
۵             Manifest.permission.ACCESS_FINE_LOCATION,
۶             Manifest.permission.ACCESS_COARSE_LOCATION
۷         ),
۸         1
۹     )
۱۰ }
۱۱

```

نمونه کد ۲: کد درخواست دسترسی لوکیشن

همان‌طور که مشخص است دو دسترسی برای مکان‌یابی وجود دارد. اولی `FineLocation` است که جهت دریافت محدوده کلی مکانی است و دومی جهت دریافت مختصات دقیق کاربر. برای اندرویدهای ورژن‌های نسخه ۱۲ به بالا این دو دسترسی از همدیگر تفکیک شده‌اند. در ادامه برای خواندن وضعیت سلول‌ها نیز به دسترسی `FineLocation` نیاز است.

در ادامه مختصات کاربر که شامل دو عدد `lat` و `lng` است گرفته می‌شود

۳

```
۱ if (!hasPermission()) {  
۲     getPermission()  
۳     return null  
۴ }  
۵ fusedLocationClient.lastLocation  
۶ .addOnSuccessListener { location >-  
۷     if (location != null) {  
۸         val coordinate = LocationCoordinate(location.latitude, location.longitude)  
۹         this.locationChanged.onNext(coordinate)  
۱۰     }  
۱۱ }  
۱۲ }
```

نمونه کد ۳: کد دریافت مختصات کاربر

اندازه‌گیری سیگنال ورودی:

با استفاده از کتابخانه android.telephony می‌توان اطلاعات سلول و توان سیگنال را به دست آورد. با توجه به اینکه دیتامدل سلول نسل‌های مختلف در این کتابخانه یکسان نیست، نیاز به تشخیص نسل سلول برای استخراج دیتاهای آن وجود داشت. کدهای زیر به ترتیب مشخص‌کننده اطلاعات نسل‌های ۴ تا ۲ هستند.

```
۱ if (cellInfo is CellInfoLte && cellInfo.isRegistered) {  
۲     val cellIdentityLte = cellInfo.cellIdentity  
۳     val cellSignalStrengthLte = cellInfo.cellSignalStrength  
۴     cellId = cellIdentityLte.ci  
۵     signalStrength = cellSignalStrengthLte.dbm  
۶     generation = 4  
۷     break  
۸ }
```

نمونه کد ۴: کد تشخیص آنتن نسل چهار

```
۱ if (cellInfo is CellInfoWcdma){  
۲     val cellIdentity = cellInfo.cellIdentity  
۳     val cellSignalStrength = cellInfo.cellSignalStrength  
۴     cellId = cellIdentity.cid  
۵     signalStrength = cellSignalStrength.dbm  
۶     generation = 3  
۷ }
```

نمونه کد ۵: کد تشخیص آنتن نسل سه

```

۱ if(cellInfo is CellInfoGsm){
۲     val cellIdentity = cellInfo.cellIdentity
۳     val cellSignalStrength = cellInfo.cellSignalStrength
۴     cellId = cellIdentity.cid
۵     signalStrength = cellSignalStrength.dbm
۶     generation = 2
۷ }

```

نمونه کد ۶: کد تشخیص آنتن نسل دو

در هنگام تست این بخش نتیجه‌های نسبتاً غیر منتظره‌ای به دست آمد. به عنوان مثال سیم‌کارت با اپراتور رایتل همواره به سلول‌های نسل ۴ وصل می‌شود و فرقی نمی‌کند دستگاه بر روی نسل ۳ یا نسل ۴ تنظیم شده باشد.

محاسبه مکان تقریبی سلول:

این قسمت پیاده‌سازی نشده است. روش اول

با داشتن مقدار توان دریافتی در چند نقطه و اینکه می‌دانیم توان ارسالی در مبدا چه قدر بوده است می‌توان به

محدوده حدودی آنتن در فضای سه‌بعدی دست یافت. روش دوم

با توجه به اینکه امواج الکترومغناطیسی با سرعتی نزدیک به سرعت نور حرکت می‌کنند می‌توان با توجه به

مدت زمان ارسال و دریافت سیگنال فاصله تخمینی آنتن را با کاربر پیدا کرد.