



دانشکده مهندسی کامپیوتر

## پروژه درس

درس شبکه‌های تلفن همراه

محمد اصولیان، نوید ابراهیمی و سینا علی‌نژاد

نیم سال دوم  
سال تحصیلی ۱۴۰۲-۱۴۰۳

## مقدمه

در پروژه باریم، قرار است یک کلاینت و یک سرور داشته باشیم. برنامه‌ای که بر روی کلاینت قرار دارد، باید همواره به وضعیت سلول خدمتگذار گوش کند و توان دریافتی از آن را بدست آورد. اگر مقدار توان دریافتی از یک حدی کاهش پیدا کرد، پیامی را برای سرور ارسال کند. این پیام باید حاوی اطلاعاتی از سلول خدمتگذار مثل منطقه مکانی (LAC)، تکنولوژی سلول، شناسه سلول و برخی موارد دیگر باشد.

## ساختار کلی پروژه

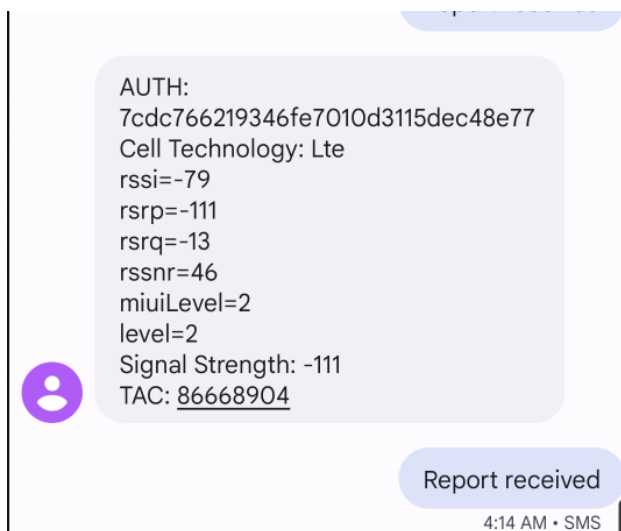
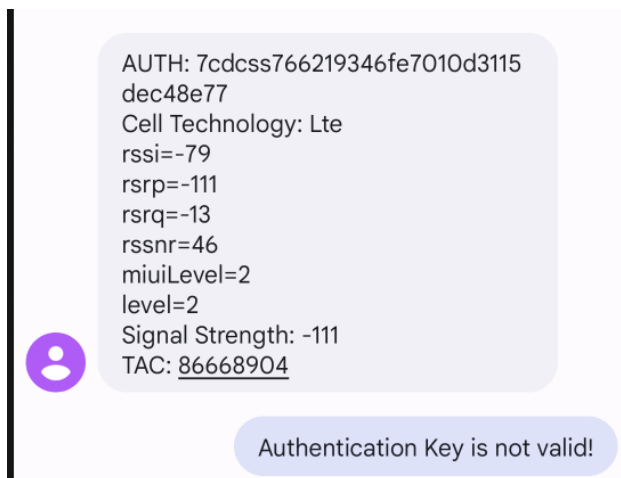
عمده‌ی کد پروژه در سمت کلاینت است، زیرا هم باید اطلاعات سلول خدمتگذار را بخواند و هم باید ارسال پیام انجام دهد. کد پروژه از دو سرویس اندروید تشکیل شده است. یکی همان سرویس اصلی یا main است که در درون آن با زدن دکمه Start یک سرویس دیگر که SignalMonitoring است شروع میشود. این یک سرویس foreground است، به این معنا که سیستم نمیتواند به دلایل مختلف مثل کمبود حافظه این سرویس را قطع کند زیرا ارزش آن از سرویس‌های background بیشتر است. هنگامی که کاربر روی دکمه شروع میزند، سرویس SignalMonitoring شروع میشود. درون این سرویس یک Listener تعریف میشود. کار این Listener این است که حواسش به تغییرات قدرت سیگنال باشد و هنگامی که این مقدار تغییر کرد، اطلاعات جدید را از سلول جمع بکند. پس از جمع‌آوری اطلاعات، مقادیر notification نمایشی را تغییر می‌دهد و سپس بررسی میکند که آیا مقدار توان دریافتی از یک حدی کمتر شده است یا خیر. اگر کمتر شده بود، اطلاعات جمع‌آوری شده را در قالب یک پیام برای سرور ارسال می‌کند.

```
private val signalStrengthListener = object : PhoneStateListener() {
    @RequiresApi(Build.VERSION_CODES.Q)
    override fun onSignalStrengthsChanged(signalStrength: SignalStrength) {
        super.onSignalStrengthsChanged(signalStrength)
        val cells = signalStrength.cellSignalStrengths
        if (cells.isNotEmpty()){
            val cellSignal = cells[0]
            val strength = getSignalStrengthForCell(cellSignal)
            val threshold = getThresholdForCell(cellSignal)
            val tech = getTechnologyForCell(cellSignal)
            updateNotification(text = "Cell Technology: $tech ---- strength: $strength")

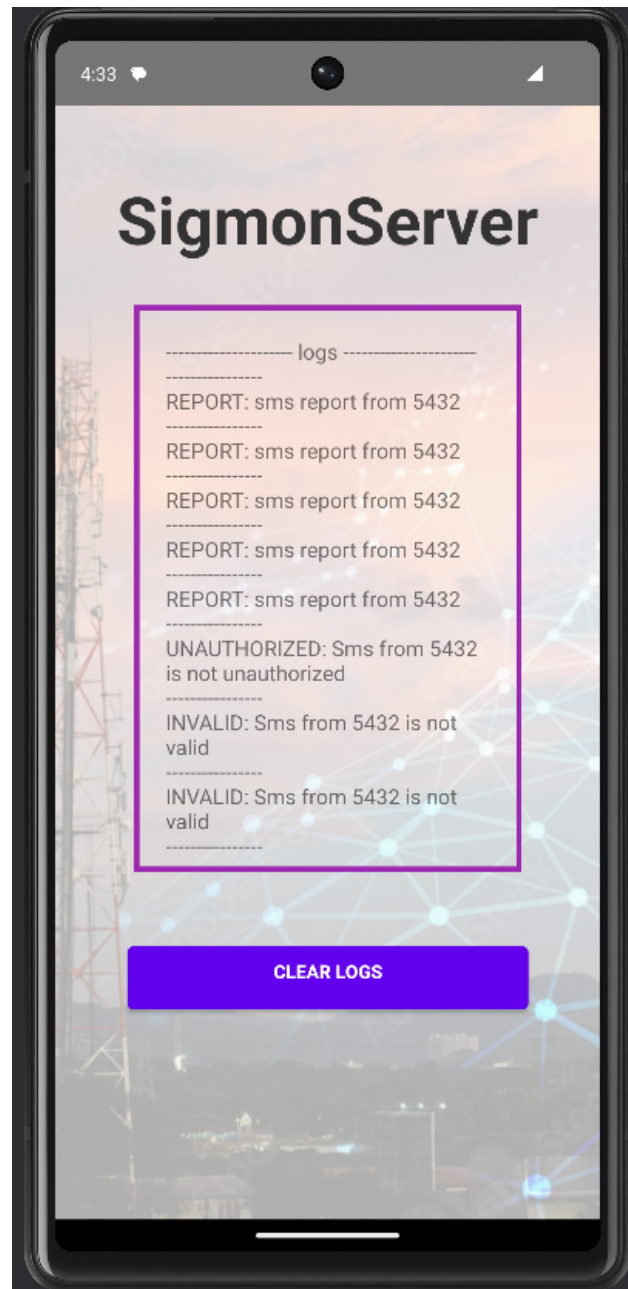
            if (strength < threshold){
                sendSmsToServer(message = get-serving-cell-info())
            }
        }
    }
}
```

همچنین در سمت سرور، با اجرای برنامه و درخواست دسترسی‌ها، سرور شروع به شنیدن پیامک‌ها میکند. در صورتی که پیامک دریافتی در قالب پروتکل مشخص شده نباشد، سرور پاسخی به آن نمیدهد. در صورتی که رمز هش شده درست نباشد،

سرور پیام اشتباه بودن رمز را میفرستد و در صورتی که رمز هش شده صحیح باشد، سرور پیام دریافت گزارش را میفرستد

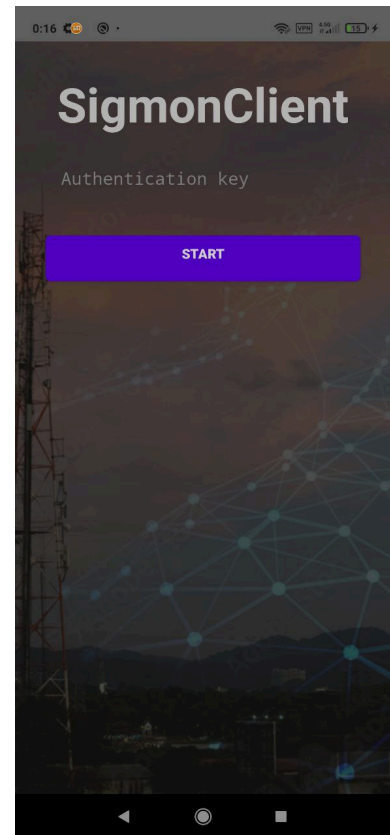


در برنامه سرور قابلیت مشاهده و ذخیره سازی لاگ ها هم فراهم شده. کاربر میتواند وضعیت درخواست هایی که به سرور میشود را مشاهده و در صورت لازم لاگ ها را پاک کند.

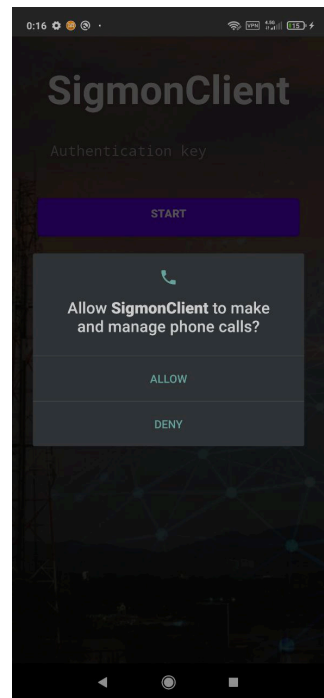
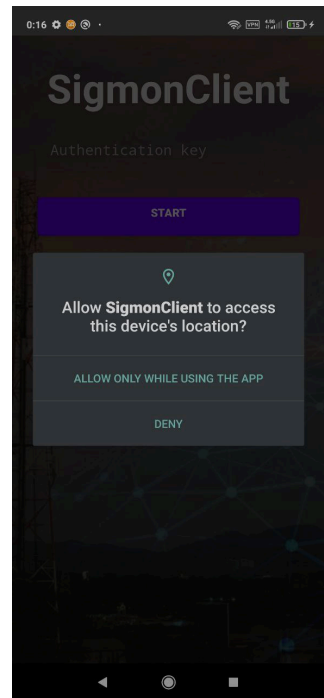


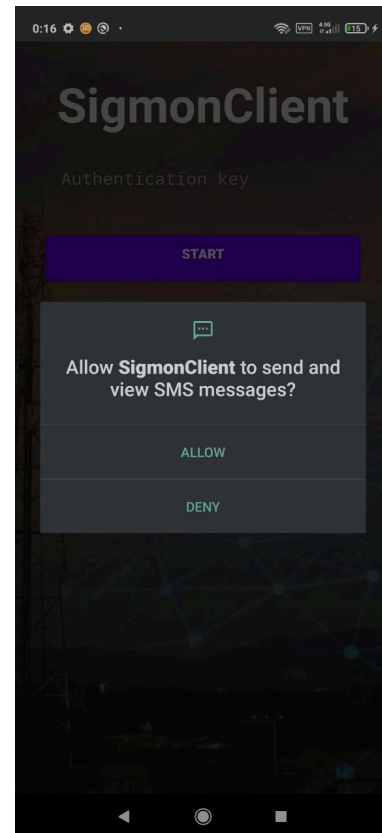
## راهنمای نرم افزار

برای بخش کلاینت یک رابط کاربری ساده نیز زده شد. عکس زیر صفحه اصلی برنامه کلاینت را نشان می‌دهد.

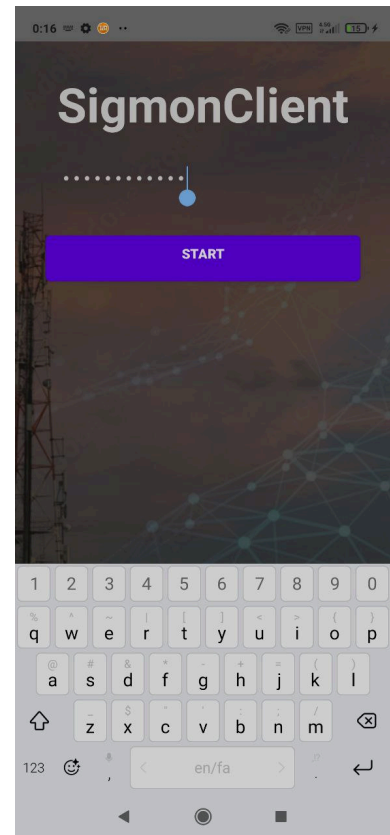


پس از ورود به برنامه، باید اجازه‌های لازم را برای ارسال و دریافت پیام و همچنین گرفتن اطلاعات سلول و location بدهید.



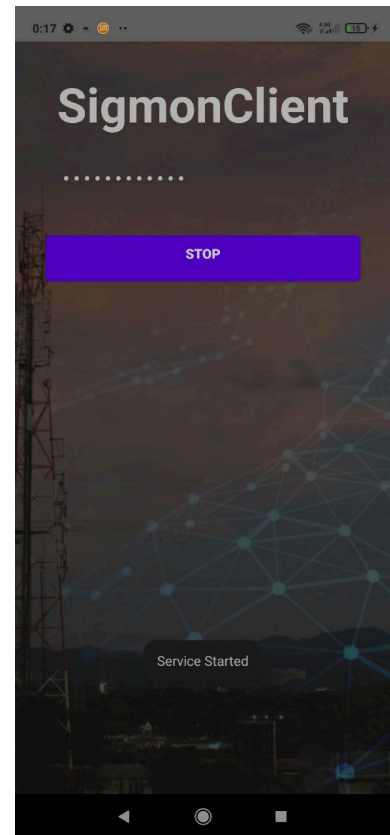


ابتدا کاربر باید عملیات register را انجام دهد. پس از آن، سرور یک رمز را به صورت hash شده به کاربر پیامک می‌کند و کاربر می‌تواند از آن برای شروع برنامه استفاده کند. تصویر زیر وارد کردن پسورد را در برنامه نشان می‌دهد.

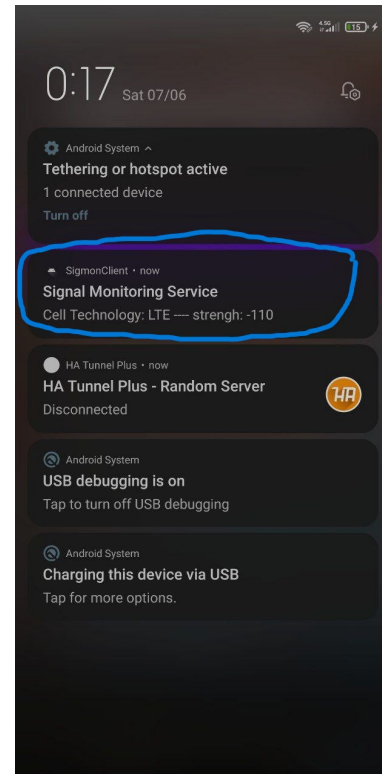


پس از زدن دکمه Start نوشته‌ی روی دکمه به Stop تغییر می‌کند و یک پیام به صورت toast در پایین صفحه شروع فرایند را اعلام می‌کند.

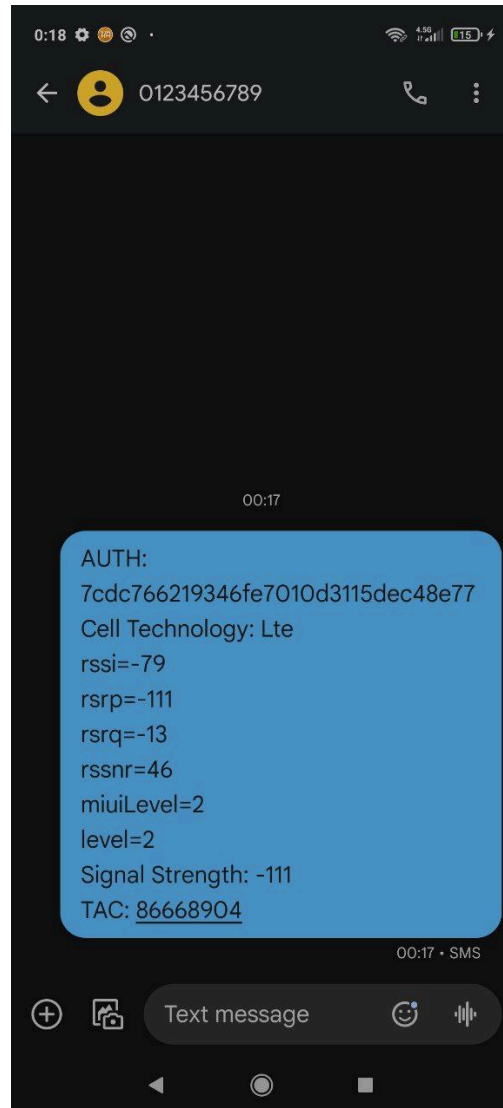




در این حالت وضعیت شبکه، یعنی تکنولوژی سلول و توان دریافتی از آن به صورت یک notification و به صورت live نمایش داده می‌شود.



همانطور که از عکس مشخص است، تکنولوژی سلول خدمتگذار LTE یا همان 4G است. همچنین توان سیگنال دریافتی برابر با -110 dbm است. اگر این توان دریافتی از -100 کمتر شود. پیامی را به مانند زیر برای سرور ارسال میکند.



همانطور که مشخص است، این اطلاعات شامل Authentication Key به صورت hash شده با الگوریتم md5 است. همچنین تکنولوژی سلول و قدرت سیگنال و مواردی که خاص نسل چهارم میباشند مثل TAC و rsrp و rsrq . در نهایت با زدن دکمه Stop این فرایند متوقف میشود.

