

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

SEMESTRÁLNA PRÁCA

Ľudmila Čániová

Tréningová aplikácia – „TrainIT!“

Žilina, 2021

Obsah

1	Ciele práce	3
2	Tréningové aplikácie v súčasnosti	4
2.1	STRAVA	4
2.2	Polar Flow	5
3	Návrh aplikácie	6
3.1	Návrh funkcionalít.....	6
3.2	Návrh vzhľadu obrazoviek	7
4	Implementácia.....	8
4.1	Aktivity.....	8
4.1.1	Login aktivita.....	8
4.1.2	Main aktivita.....	8
4.2	Fragmenty.....	9
4.2.1	Menu.....	9
4.2.2	Nová aktivita	10
4.2.3	Počasie	11
4.3	Knižnica Google maps	11
4.4	Knižnica okhttp3	12
4.5	Sieťová komunikácia – OpenWeatherMap	12
4.6	GPS senzor	12
4.7	Notifikácia po ukončení behu.....	14

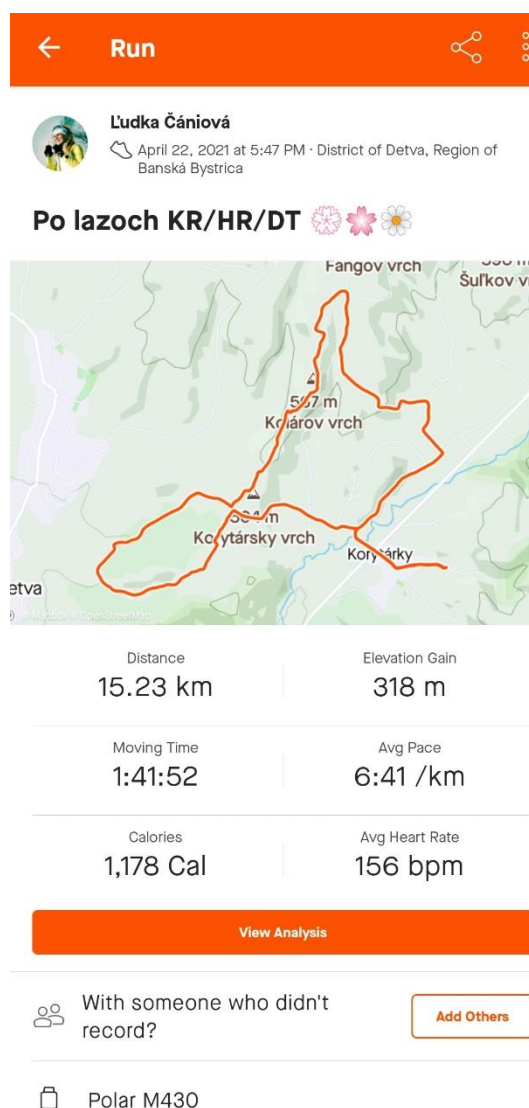
1 Ciele práce

Cieľom mojej semestrálnej práce je vytvoriť tréningovú aplikáciu pre mobilné zariadenie bežiacie na operačnom systéme Android. Táto aplikácia bude vytvorená v jazyku Kotlin. Hlavnými požiadavkami bude možnosť zaznamenávať rôzne typy tréningov a prehľad počasia v jeho okolí.

2 Tréningové aplikácie v súčasnosti

2.1 STRAVA

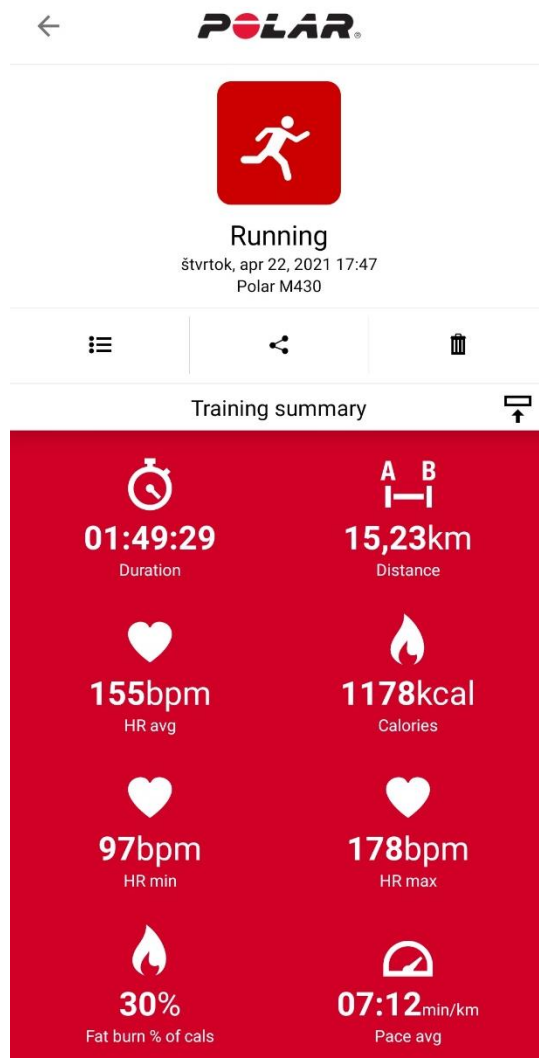
Aplikácia STRAVA, je v súčasnosti najpoužívanejšou aplikáciou na zaznamenávanie tréningov vo svete. Môžeme ju nazvať aj sociálnou sieťou pre športovcov. Športovci si do nej zaznamenávajú svoje tréningy, pocity a fotografie z nich.



Obrázok 1 - Screenshot z aplikácie STRAVA

2.2 Polar Flow

Aplikácia Polar Flow slúži na uchovávanie tréningových dát zo zariadení značky Polar. Po zaznamenaní aktivity na športtester si športovec zosynchronizuje dáta s touto aplikáciou, a následne si v nej môže prezerať všetky svoje aktivity.

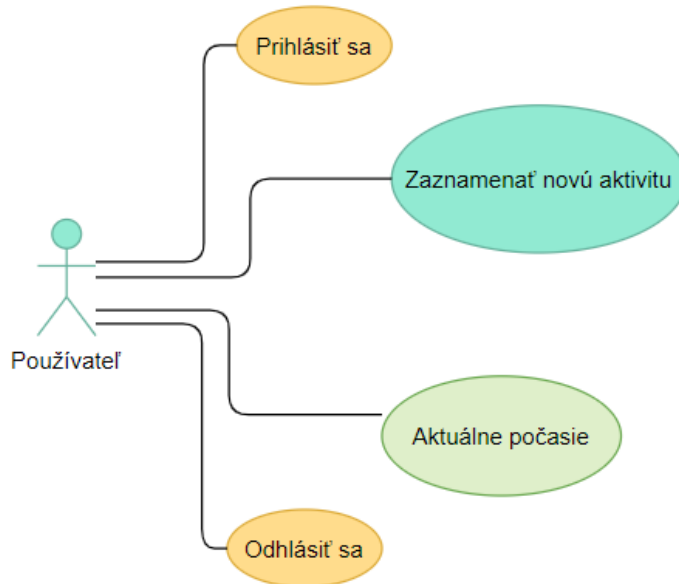


Obrázok 2 - Screenshot z aplikácie Polar Flow

Moja aplikácia sa bude líšiť od existujúcich aplikácií zjednodušeným ovládaním, príjemnejšími farbami používateľského prostredia a doplnujúcou funkciou pre získanie informácií o aktuálnom počasí.

3 Návrh aplikácie

3.1 Návrh funkcionalít

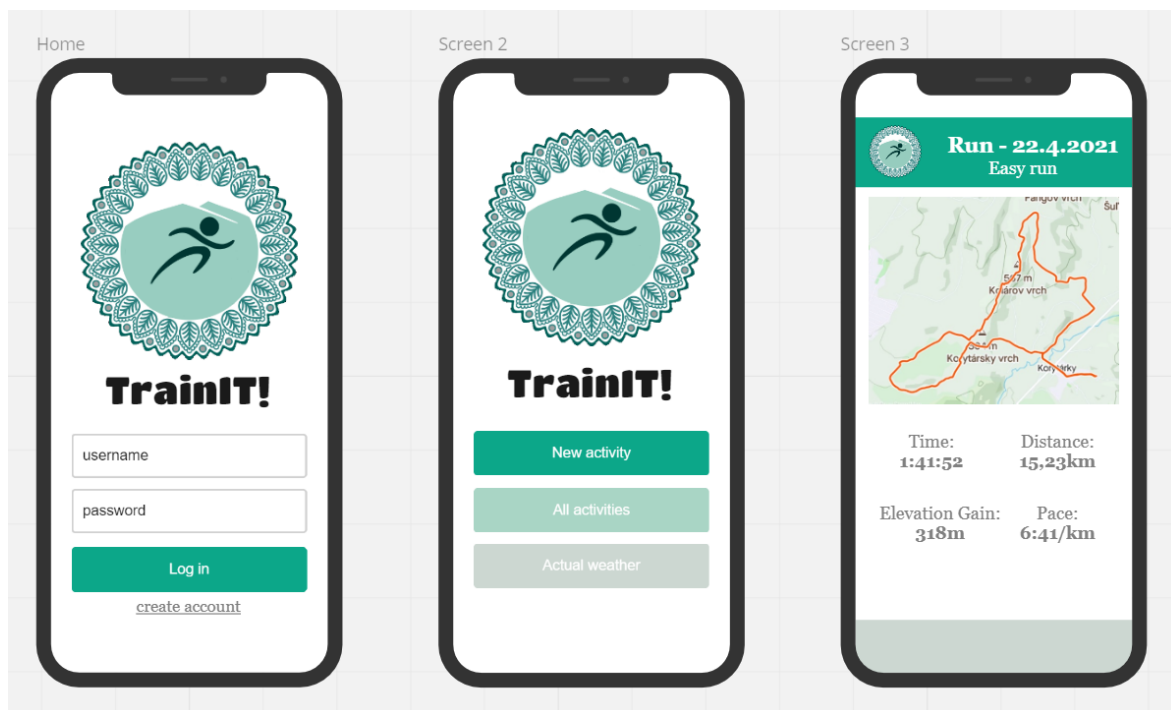


Obrázok 3 – Use case diagram

Hlavnými úlohami mojej aplikácie bude zabezpečenie konta pre každého užívateľa. Po prihlásení bude môcť používateľ zaznamenávať novú športovú aktivitu. Ďalšou z funkcií bude prehľad aktuálneho počasia na vybranom mieste.

3.2 Návrh vzhľadu obrazoviek

Na obrázku nižšie môžete vidieť ukážku návrhu obrazoviek. Na prvom obrázku je návrh domovskej obrazovky určenej na prihlásenie sa. Ďalej na druhom obrázku je zobrazené používateľské prostredie bežca. A na poslednom je pohľad na zaznamenanú aktivitu.



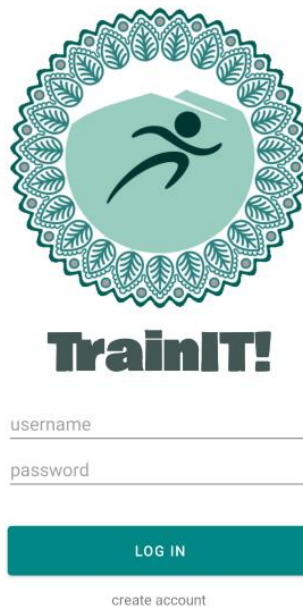
Obrázok 3 - Ukážka návrhu obrazoviek aplikácie

4 Implementácia

4.1 Aktivita

4.1.1 Login aktivita

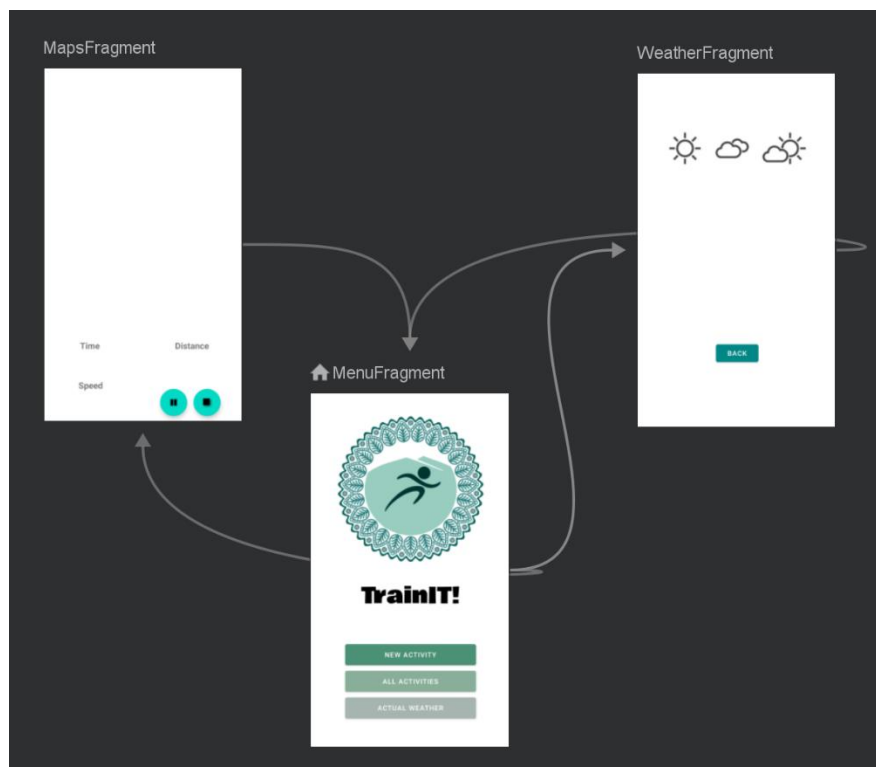
Login aktivita zabezpečuje autentifikáciu pomocou backend-u vytvoreného v jazyku GoLang. Aktivita bola vygenerovaná v Android studiu a napojená na externé api heroku (použitá aj pri semestrálnej práci z VAII). Keďže zatiaľ nie je zabezpečené vytváranie účtov v aplikácii, odporúčam použiť pre prihlásenie **username: admin a heslo: admin**.



Obrázok 4 - layout pre login aktivitu

4.1.2 Main aktivita

Main aktivita je najhlavnejšou aktivitou a obsluhuje väčšinu aplikácie. Je v nej použitá navigácia medzi tromi fragmentami, ktoré budú rozpísané ďalej.



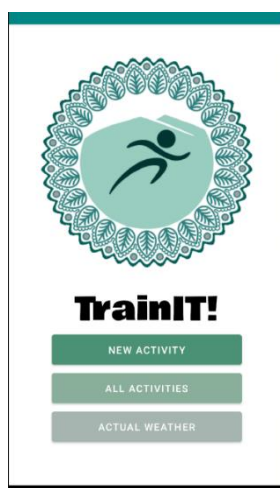
Obrázok 5 - Navigation graph

4.2 Fragmenty

Všetky fragmenty sú urobené v dvoch verziách pre portrétovú aj krajinkovú orientáciu.

4.2.1 Menu

Menu fragment obsahuje tri tlačidlá. Prvé pre vytvorenie novej aktivity. Druhé pre prehľad aktivít. Táto funkcia však nie je funkčná, je však predpripravená pre ďalšie vyvíjanie aplikácie. Posledným tlačidlom je tlačidlo pre aktuálne počasie.

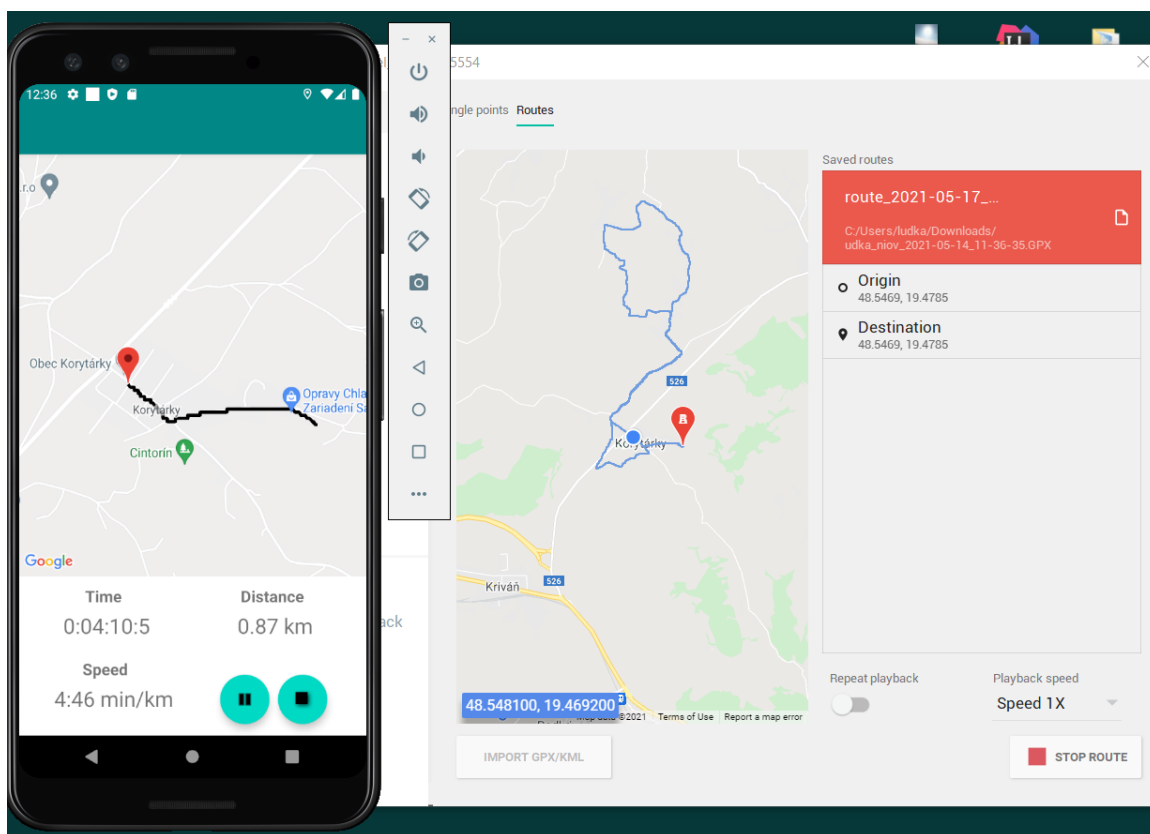


Obrázok 6 - layout pre menu fragment

4.2.2 Nová aktivita

Vo fragmente nová aktivita sa zaznamenáva pomocou gps súradníc prejdená trať. Zároveň sa vypočítava prejdená vzdialenosť, čas a rýchlosť. Aktivitu je možné pozastaviť alebo úplne stopnúť.

Testovanie aplikácie pomocou nahraného „gpx“ môžete vidieť na obrázku nižšie.



Obrázok 7 - zaznamenávanie novej aktivity

4.2.3 Počasie

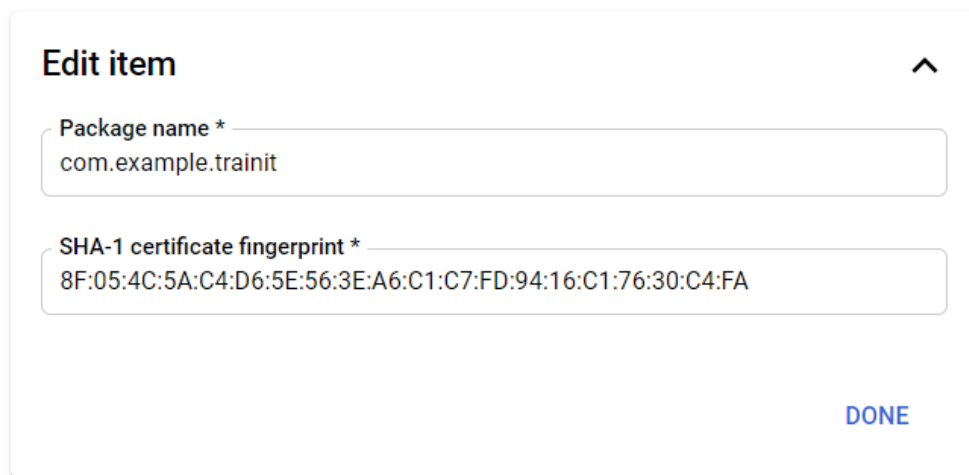
Fragment počasie zobrazuje aktuálne počasie na danom mieste.



Obrázok 8 - layout fragmentu počasie

4.3 Knižnica Google maps

Pre zobrazovanie máp je použitá knižnica Google maps. Na stránke <https://console.cloud.google.com> bolo potrebné vygenerovať si svoj vlastný API kľúč.



Obrázok 9 - nastavenia pre google api

Na základe vygenerovaného fingerprintu z androidstudia som získala API kľúč uvedený na obrázku nižšie.



Obrázok 10 - API kľúč zavedený do programu

4.4 Knižnica okhttp3

Knižnica slúži na vytvorenie spojenia potrebného na POST request pri prihlasovaní.

```
val okHttpClient = OkHttpClient()
val body = RequestBody.create(MediaType.parse("application/json; charset=utf-8"), json)
val request = Request.Builder()
    .url("https://dbtspapi.herokuapp.com/adminlogin")
    .post(body)
    .build()
val response = okHttpClient.newCall(request).execute()
```

Obrázok 11 - použitie knižnice okhttp3

4.5 Siet'ová komunikácia – OpenWeatherMap

Pri získavaní informácií o aktuálnom počasi je využitá siet'ová komunikácia a to pomocou OpenWeatherMap API. Na toto API sa autentifikujem pomocou API kľúča nižšie.

```
val API: String = "06c921750b9a82d8f5d1294e1586276f"
```

```
URL = "https://api.openweathermap.org/data/2.5/weather?lat=${location?.Latitude}&lon=${location?.Longitude}&units=metric&appid=$API"
```

Obrázok 12 - použitie OpenWeatherMap

4.6 GPS senzor

Aplikácia pracuje s GPS senzorom, ktorý je využívaný či už na získanie polohy pre aktuálne počasie na danom mieste, alebo pre aktuálnu polohu pri nahrávaní aktivity.

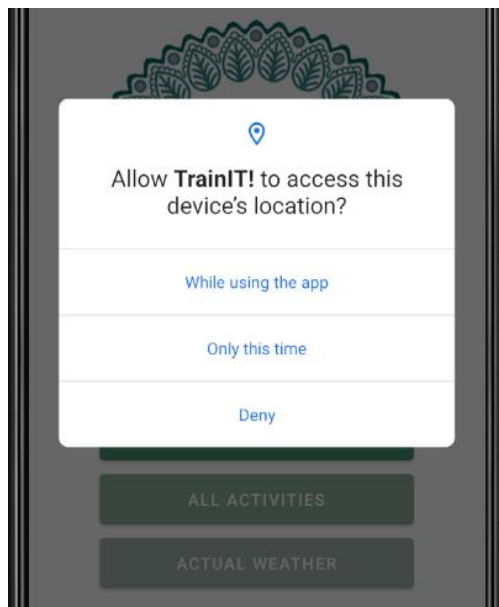
```

//skontroluje či mám povolenie na získanie polohy
private fun checkPermissions(): Boolean {
    if (ActivityCompat.checkSelfPermission(
        context: this,
        Manifest.permission.ACCESS_COARSE_LOCATION
    ) == PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(
            context: this,
            Manifest.permission.ACCESS_FINE_LOCATION
        ) == PackageManager.PERMISSION_GRANTED
    ) {
        return true
    }
    return false
}

//ak nemám povolenie na získanie polohy, tak si ho vypýtam
private fun requestPermissions() {
    ActivityCompat.requestPermissions(
        activity: this,
        arrayOf(Manifest.permission.ACCESS_COARSE_LOCATION, Manifest.permission.ACCESS_FINE_LOCATION),
        PERMISSION_ID
    )
}

```

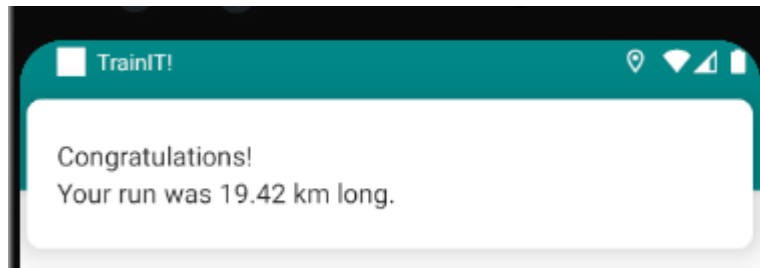
Obrázok 13 - žiadosť o zdieľanie polohy



Obrázok 14 - ukážka žiadosti o zdieľanie polohy

4.7 Notifikácia po ukončení behu

Po ukončení behu príde používateľovi notifikácia s počtom prebehnutých kilometrov.



Obrázok 15 - ukážka notifikácie

```
//nove androidy 0+(Android 8)
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
    notificationChannel = NotificationChannel(channelId, name: "notification", NotificationManager.IMPORTANCE_HIGH)
    notificationChannel.enableLights(lights: true)
    notificationChannel.lightColor = Color.GREEN
    notificationChannel.enableVibration(vibration: true)
    notificationManager.createNotificationChannel(notificationChannel)

    builder = Notification.Builder(root.context, channelId)
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setContentTitle(getString(R.string.notificationP1))
        .setContentText("Your run was ${round((SphericalUtil.computeLength(mLatLngList) / 10))/100} km long.")
        .setLargeIcon(BitmapFactory.decodeResource(this.resources, R.drawable.ic_launcher_background))
}
//stare androidy N-
else {
    builder = Notification.Builder(root.context)
        .setContentTitle(getString(R.string.notificationP1))
        .setContentText("Your run was ${round((SphericalUtil.computeLength(mLatLngList) / 10))/100} km long.")
        .setSmallIcon(R.drawable.ic_launcher_background)
        .setLargeIcon(BitmapFactory.decodeResource(this.resources, R.drawable.ic_launcher_background))
}
notificationManager.notify(id: 1234, builder.build())
```

Obrázok 16 - ukážka notifikácie v kóde