Michele Gennari
n° 0000832608

February 16° 2024

Signal Doctor

An Android Noise and Signals Measurement Application

A project for Mobile Application Languages course in Computer Science degree, UniBo

Introduction

Signal Doctor is a native Android mobile application. It builds upon the project specifications for the LAM course exam. Its main uses are:

- Measuring environment noise, cellular connectivity signal strength and WiFi signal strength.
- Presenting to the user a map for each measurement type, reporting measurements quality within geographic areas.

Application Requirements

Measurements

- App can measure noise, cellular signal strength, and Wifi signal strength
- One-time measurements
- Periodic background measurements every N time interval, where N is set by the user
- Users must be notified when there are no measurements taken at their location, or when measurements are too old

Measurements Map

- Map keeps track of user location
- One map overlay per measurement type
- Overlays must divide map into areas which neither overlap or leave gaps
- Overlays paint areas with a colour representing average quality of last X measurements taken there, where X is set by the user
- Users must be notified when there are no measurements taken at their location, or when measurements are too old

Improved Application Features

Improved Application Features

Online and Offline Modes

- Measurements can be sent on a back-end server shared by clients, and map can merge offline and online data to paint areas

Fine-grained Map Area size

- Areas change their size when user zooms in and out on the map

Location search bar

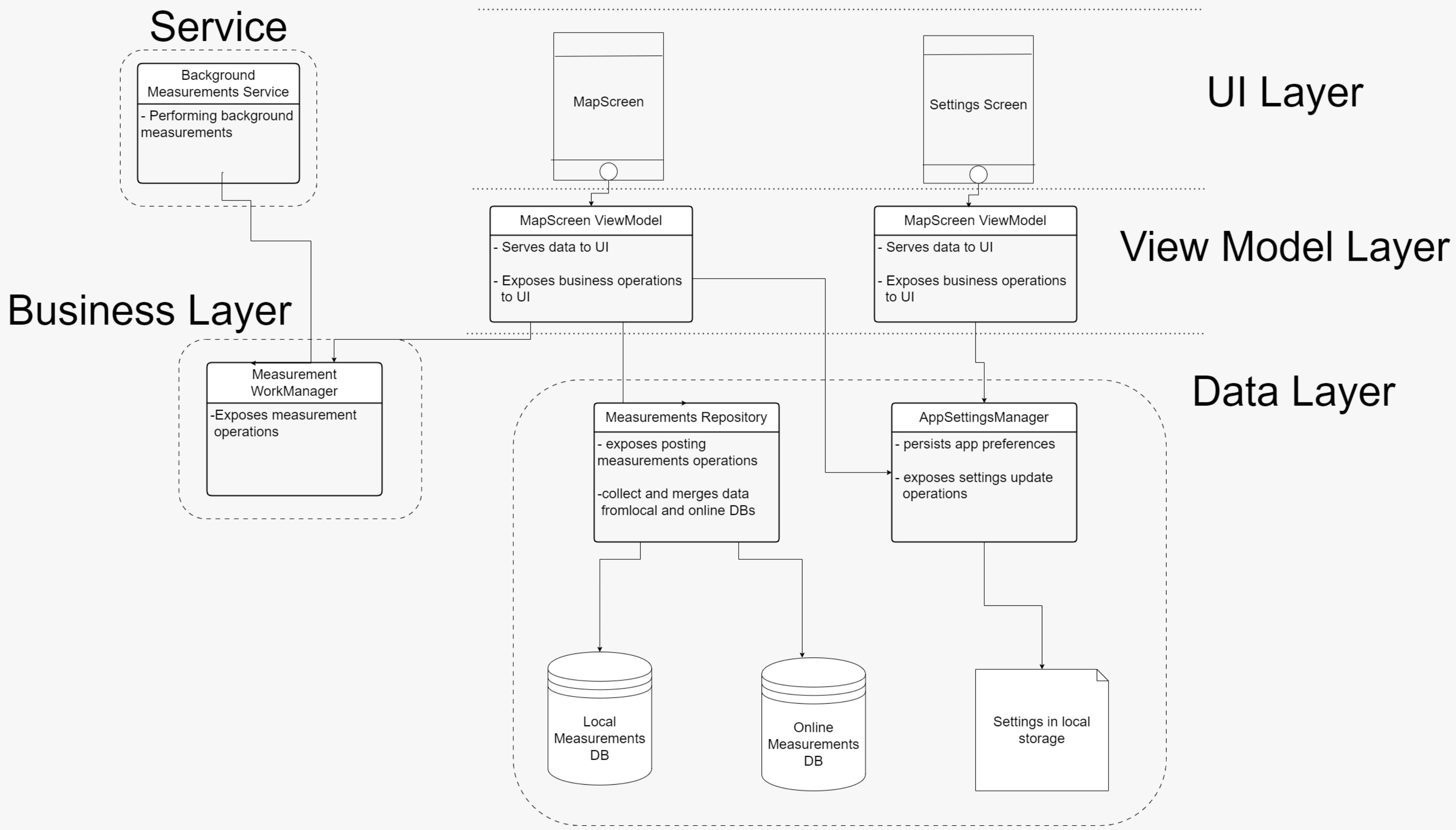
- Users can navigate to a specific location on the map

Background measurements service

- Background measurements are performed by a service that runs even when app is closed

Application architecture

Application Architecture

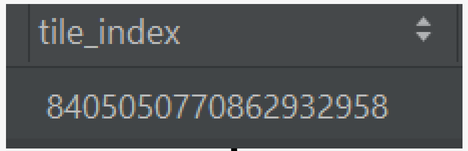

Service
Background Measurements Service
- Performing background measurements
UI Layer
MapScreen
Settings Screen
View Model Layer
MapScreen ViewModel
- Serves data to UI
- Exposes business operations to UI
MapScreen ViewModel
- Serves data to UI
- Exposes business operations to UI
Business Layer
Measurement WorkManager
-Exposes measurement operations
Data Layer
Measurements Repository
- exposes posting measurements operations
-collect and merges data fromlocal and online DBs
AppSettingsManager
- persists app preferences
- exposes settings update operations
Local Measurements DB
Online Measurements DB
Settings in local storage

Application Design & Implementations

General design choices and Implementations

Design	**Implementation**
Reactive to data updates	Kotlin Flows & Coroutines
Navigate through screens	Compose Navigation

Measurements Map with Fine-grained areas



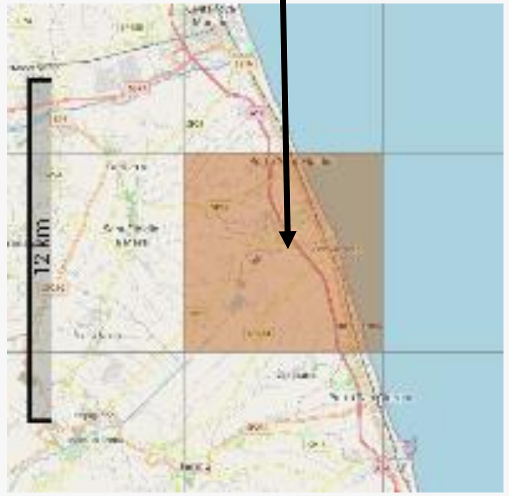
tile_index
8405050770862932958

- **Library:** OSMdroid

Check if it’s between bounds

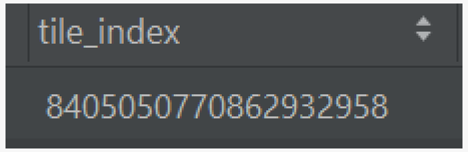
- Main idea:

- Map tiles as areas to colour
- Measurements data is stored along with the max zoom’s map tile index of where the measurement was taken
- When drawing tiles, map overlays compute tile bounds in form of tile indexes, and add to the average only those measurements that are within those bounds



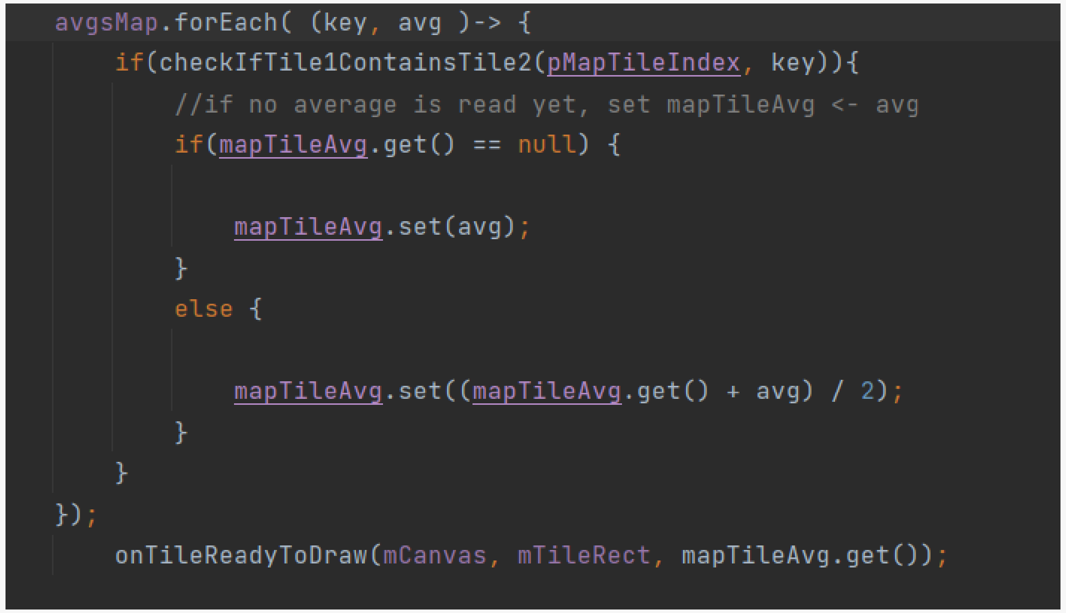
Measurements Map with Fine-grained areas

Measurement associated with map tile index



tile_index
8405050770862932958

When drawing map tile, computes average quality with measurements within tile bounds

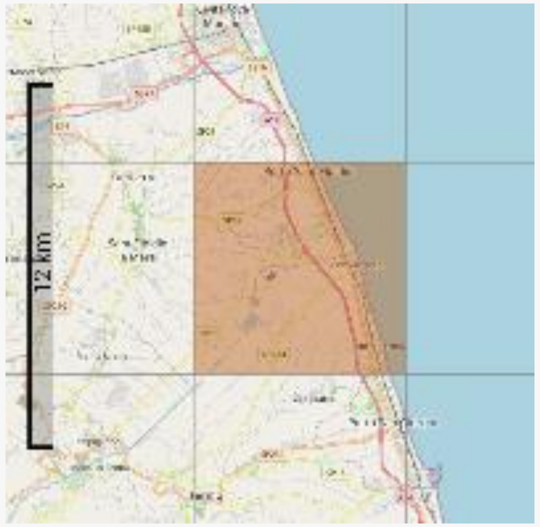


```
avgsMap.forEach( (key, avg )-> {
    if(checkIfTile1ContainsTile2(pMapTileIndex, key)){
        //if no average is read yet, set mapTileAvg <- avg
        if(mapTileAvg.get() == null) {

            mapTileAvg.set(avg);
        }
        else {

            mapTileAvg.set((mapTileAvg.get() + avg) / 2);
        }
    }
});
    onTileReadyToDraw(mCanvas, mTileRect, mapTileAvg.get());
```

Colour is a gradient between bad and good quality colours

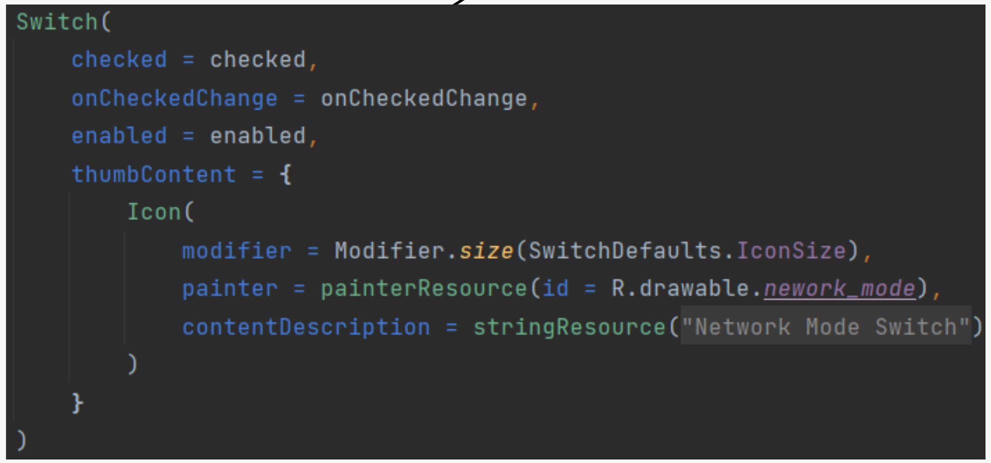


UI Screens

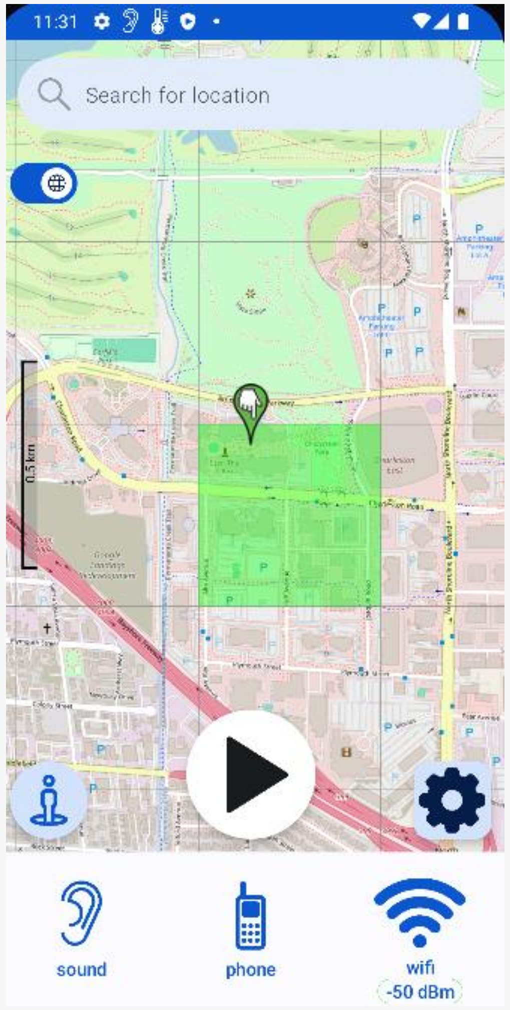
- **Framework, Library and API:** Compose, Material and Kotlin StateFlows

- Why?:

- App should look polished but minimalistic and material offers ready-to-use UI elements
- Smooth reaction to data changes



```
Switch(
    checked = checked,
    onCheckedChange = onCheckedChange,
    enabled = enabled,
    thumbContent = {
        Icon(
            modifier = Modifier.size(SwitchDefaults.IconSize),
            painter = painterResource(id = R.drawable.nework_mode),
            contentDescription = stringResource("Network Mode Switch")
        )
    }
)
```

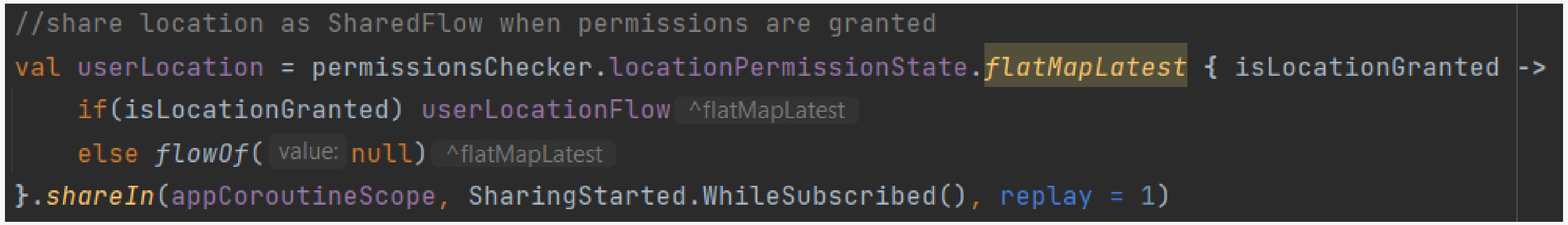


Location Updates

- **Libraries:** Kotlin SharedFlows and Google's FusedLocationProvider

- **Why?:**

  - App constantly keeps track of location to show and produce consistent information/data, and that is an expensive operation
  - SharedFlows, as their name suggest, share geo location among app components that need it, with a single location updates operation



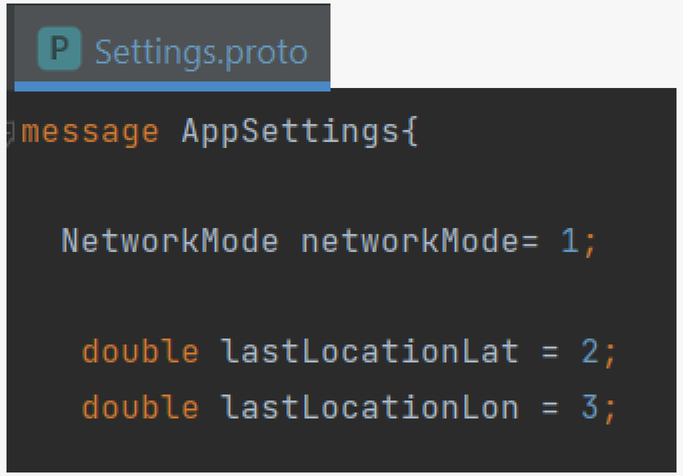
```
//share location as SharedFlow when permissions are granted
val userLocation = permissionsChecker.locationPermissionState.flatMapLatest { isLocationGranted ->
    if(isLocationGranted) userLocationFlow ^flatMapLatest
    else flowOf( value: null) ^flatMapLatest
}.shareIn(appCoroutineScope, SharingStarted.WhileSubscribed(), replay = 1)
```

App Settings

- **Libraries:** Kotlin SharedFlows and Proto DataStore

- **Why?:**

- We talked about pros of SharedFlows earlier
- Protobuff ensures type safety, since kotlin generates classes from messages definitions


Settings.proto

```
message AppSettings{

  NetworkMode networkMode= 1;

   double lastLocationLat = 2;
   double lastLocationLon = 3;
```

Measurements Repository

	Local Database	Online Database
Library	Room	Firebase Relatime DB
Why?	• Easy declarative approach	• Easy to set up server
	• One entity definition serves for both • Integrated with Flows: don't need to actively ask for data	
Base entity properties	UUID (for merging), Value, Date, Tile Index	
Queries	Post measurement, get measurements, get averages relative to tile indexes	

Measurements Operations

- **Main library:** WorkManager

- Why?: Operations survive app lifecycle

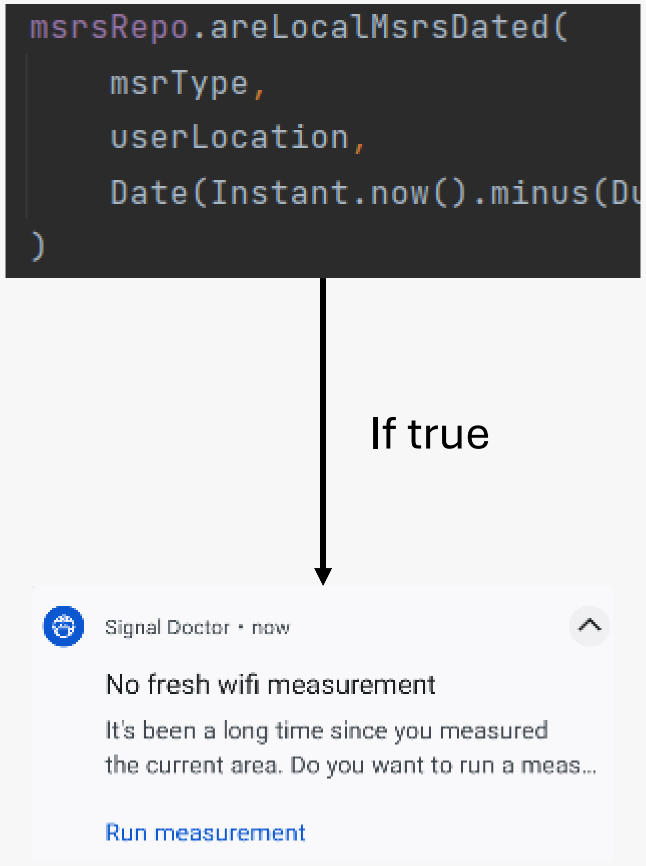
Measurement	Noise	Wifi Signal	Phone Signal
Libraries	Media Recorder, FFmpeg	Connectivity Manager	Telephony Manager
Standard used	LUFS	RSSI	

Expiration/Absence of Measurements

- Main Idea:

- Repository tells us if there are no valid measurements in current location and whenever condition it’s true, we send a notification

- Notification has a button that launches a pending intent to a broadcast receiver, which in turn ask WorkManager to run a measurement



```
msrsRepo.areLocalMsrsDated(
    msrType,
    userLocation,
    Date(Instant.now().minus(Du
)
```

Signal Doctor • now

No fresh wifi measurement

It's been a long time since you measured the current area. Do you want to run a meas...

Run measurement

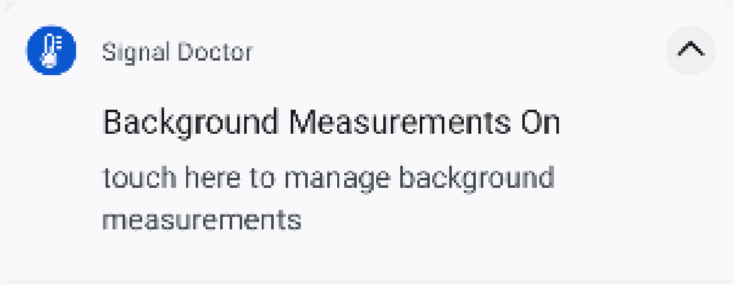
Background Measurements

-Problem

- Work manager that launches periodic work could have been enough, but users should be aware of whether background measurements are running or not

-Solution: Foreground Service

- App uses a lifecycle-aware service class, **LifecycleService,** that listens to data changes indipendently and starts/stops measurements accordingly



Signal Doctor

Background Measurements On

touch here to manage background measurements

Bibliography

—. *Kotlin coroutines on Android* . n.d. https://developer.android.com/kotlin/coroutines.

—. *Kotlin flows on Android* . n.d. https://developer.android.com/kotlin/flow.

Federico Montori, Luca Sciullo. "Virtuale.unibo." *LAM Project 2023.* n.d. https://virtuale.unibo.it/mod/resource/view.php?id=1149343.

ffmpeg.org. *FFmpeg Filters Documentation.* n.d. https://ffmpeg.org/ffmpeg-filters.html#ebur128-1.

ITU-R. *Audio levels and loudness.* 11 2023. https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770-5-202311-I!!PDF-E.pdf.

MapTiler. *Tiles à la Google Maps, Coordinates, Tile Bounds and Projection.* n.d. https://www.maptiler.com/google-maps-coordinates-tile-bounds-projection/#3/15.00/50.00.

Wikipedia. *Received signal strength indicator (RSSI).* n.d. https://en.wikipedia.org/wiki/Received_signal_strength_indicator.