

Activity Tracker App

Laboratorio di applicazioni mobili



Scopo

Permettere all'utente di tracciare le sue attività durante la giornata e consultarne le statistiche

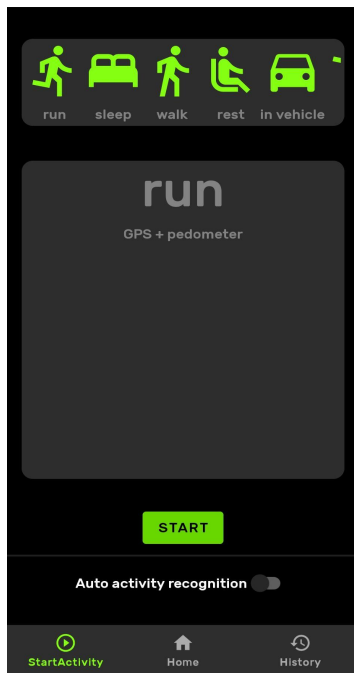
Per avere un'esperienza migliore possibile l'utente deve cercare di registrare continuamente le sue attività

Funzionamento

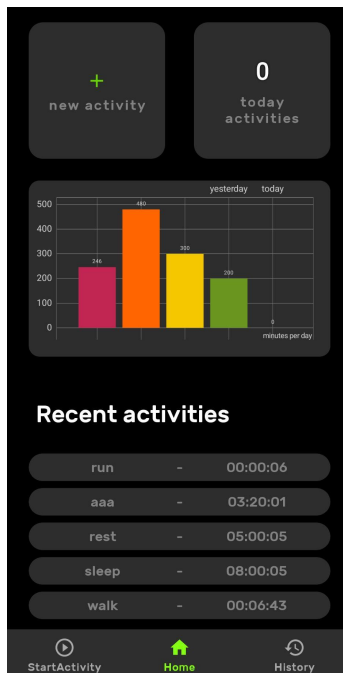
- (se non esiste ancora, l'utente crea l'attività che vuole fare)
- l'utente inizia l'attività o abilita il riconoscimento automatico
- l'utente ferma l'attività che viene salvata nel DB
- l'utente può consultare le statistiche dell'attività

Struttura

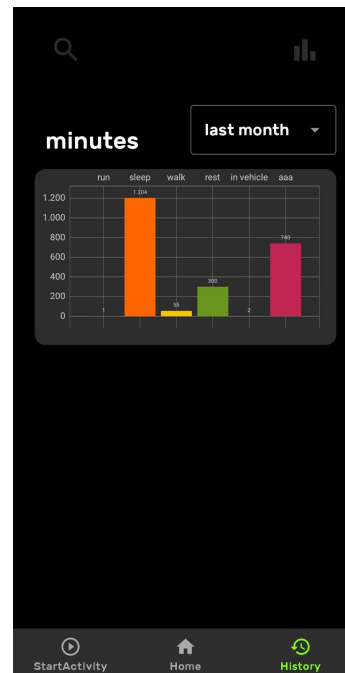
StartActivity



Home



History



Implementazione

- applicazione nativa in kotlin
- target sdk 34, min sdk 26
- design pattern: MVVM
- single activity application
- Android Views

Database: Room

ActivityType
Table

activityType_table						
<input type="checkbox"/> Live updates						
	id	name	desc	iconSrc	tools	
1	1	run		ic_activitytype_run	11	
2	2	sleep	time spent in bed	ic_activitytype_sleep	0	
3	3	walk		ic_activitytype_walk	10	
4	4	rest		ic_activitytype_rest	0	
5	5	in vehicle	car,bus, bicycle, ship, plane	ic_activitytype_car	1	
6	6	none		ic_activitytype_generic	0	
7	7	work		ic_activitytype_work	0	

activityRecord_table						
<input type="checkbox"/> Live updates						
	id	type	startTime	finishTime	toolData	
1	1	in vehicle	1720261019447	1720261064150	{"positions":[{"latitude":44.4091739,"longitude":11.6634715}],	
2	2	walk	1720261077866	1720261082882	{"positions":[],"steps":100}	
3	3	in vehicle	1720262716263	1720262812667	{"positions":[{"latitude":44.3644681,"longitude":11.7845392}],	
4	4	walk	1720469827464	1720472709499	{"positions":[],"steps":1000}	
5	5	sleep	1720443907464	1720472914482	{}	

ActivityRecord
Table

timestamp

JSON

Background: Services

Usati per registrare i dati delle attività:

- `ActivityRecognitionService`: riconoscimento automatico con Activity Recognition API
- `BaseService`: tiene traccia del tempo, classe padre
- `GpsService`: riceve gli aggiornamenti sulla posizione
- `PedometerService`: tiene traccia dei passi

I dati registrati vengono anche usati per aggiornare la UI come LiveData

Background: Broadcast Receivers

AlarmReceiver: gestisce i task schedulati

- notifiche periodiche
- none activity, registra i vuoti di attività di ogni giorno (trasparente all'utente)

ActivityRecognitoinBroadcastReceiver:

Gestisce il riconoscimento automatico di un'attività

Salva le attività terminate sul DB (> 1min)

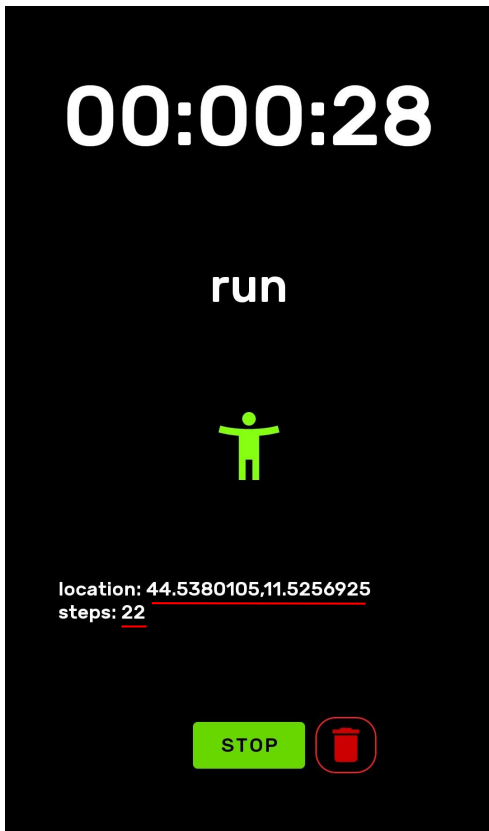
Notifica le attività registrate

Sensori

Contapassi:

Il Service registra un listener al
SensorManager

Implementa l'interfaccia
SensorEventListener per ottenere gli
aggiornamenti con una callback



Posizione:

Con FusedLocationProvider

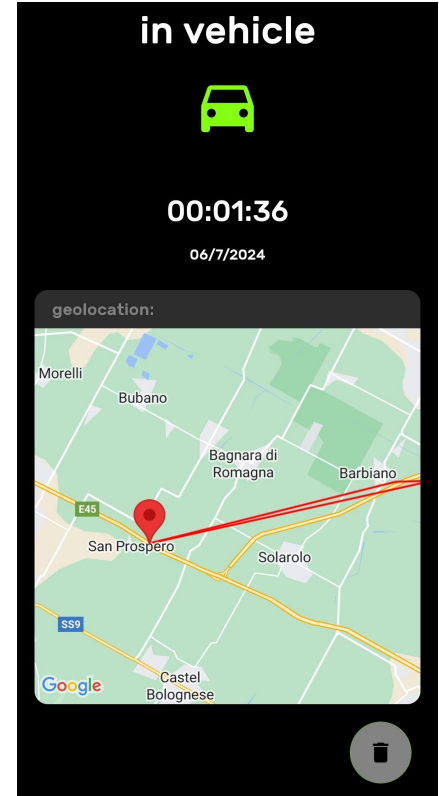
Ogni volta che viene registrata una
posizione lancia una callback

Posizioni salvate come array di
LatLng

API: Maps

SupportMapFragment fa le chiamate all'API e lancia una callback quando la mappa è pronta.

Alla mappa vengono aggiunte le posizioni registrate, come polyline; la posizione iniziale è indicata anche con un marker.



API: Activity Recognition

Activity Transition API non utilizzabile.

Ogni 1000ms richiede di elaborare i dati a disposizione per trovare l'attività in corso.

Richiede più risorse ed è meno adatta.

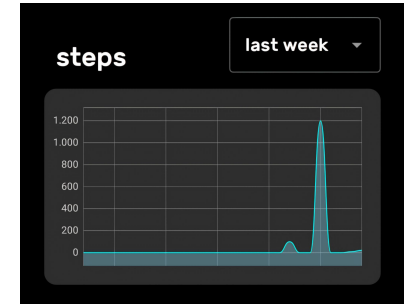
Grafici

1) minuti di attività al giorno divisi per tipo



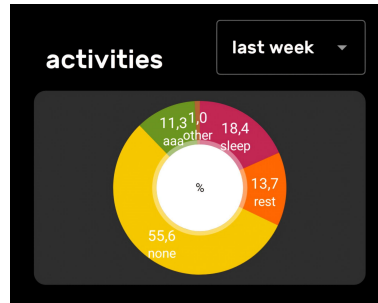
Bar Chart

2) passi giornalieri



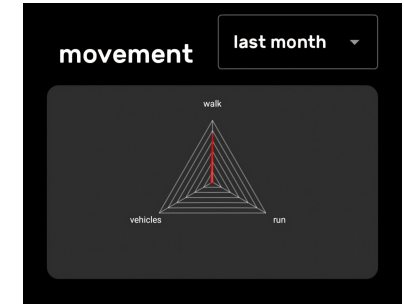
Line Chart

3) divisione delle attività nella giornata



Pie Chart

4) metodi di spostamento



Radar Chart