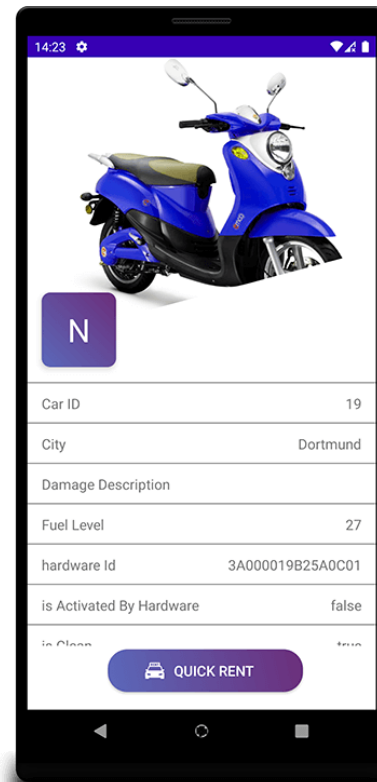
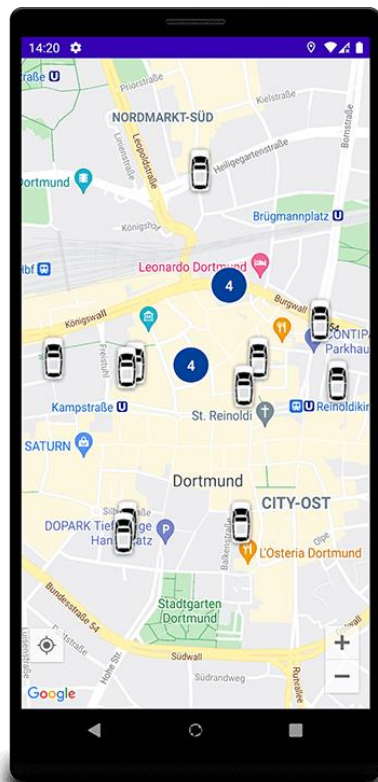


Introduction

This is an application for finding Cars on the map and renting Cars, that uses MVVM and Repository pattern, Modularization, with some principles of Clean Architecture.

In this app I was followed Google recommended [Guide to app architecture](#).



Logic and Data flow

And about Logic of app! It has a simple logic (or data-flow). At first step, you should open the app, and you will see the Map

and Cars will appear on the map in a second, and also you can see your live location on the map and if you click on location button, map camera will zoom on your location.

And also, if you click on each cluster of the cars on the map, app will zoom and open the cluster and you can Also click on a pin(car) on the map. After first click other pins will

disappear and you will see InfoBar on the car and after clicking again on that item, it will move to the detail screen. In detail screen you can see the VehicleTypeImage and scroll on the other details that shown.

You can also click on the Quick Rent and it will call the specified API and show a toast as a result.

Tech Stack

This application is written in Kotlin language.

I used Git Flow a branching model for this project, and Firebase Crashlytics for crash analytics.

Android Jetpack and Architecture Components Used almost everywhere in the app. for example you can see usage of the ViewModel, LiveData,

Lifecycles, Navigation Component and View Binding. See a complete list in "Libraries used" section.

This app does network requests via Retrofit, OkHttp and GSON.

RxJava manage network calls with simplified code and reducing needs for callbacks.

Google Map is used for map loading and Maps Cluster Manager used for clustering items on map.

Koin is used for dependency injection.

Coil (Kotlin ImageLoader) is used for image loading and Timber for logging.

Stetho is used to debugging works (like Network calls log, Database content overview, UI Hierarchy view, etc).