# Kars - Project Documentation

## 1. Project Overview

We designed this application in order to help people on certain problems that arise because of our busy day-to-day life.

The development was driven by the fact that people forget about things they don't like or are obligated to do.

The aim of our application is to help manage some of the maintenance chores for our cars. It is very easy to forget when you have to drive your car to the service for the periodical technical inspection (or ITP, as we call it) or to buy a new vignette.

Kars figures that out for you!

You only have to insert the dates when you last did your ITP and when you bought your vignette. When the time comes, our application will remind you that you have only one week left until you have to take your car to the service again or buy a new vignette. You will never have to worry about them again.

In addition to these, we also added a car-finding feature which can prove itself very useful in the moments when you forget where you parked your car. When that happens, the location of your car will be only one touch away, by accessing your car’s saved location, which can be easily saved every time you park it.

Kars development team consists of Badea Adrian, Andrei Arnautu and Trepteanu Narcis Lucian, members of group 235.

## 2. User stories, backlog creation

Being the first time developing an application like this for all of us, we needed a way to split our tasks and share the progress each of us has made.

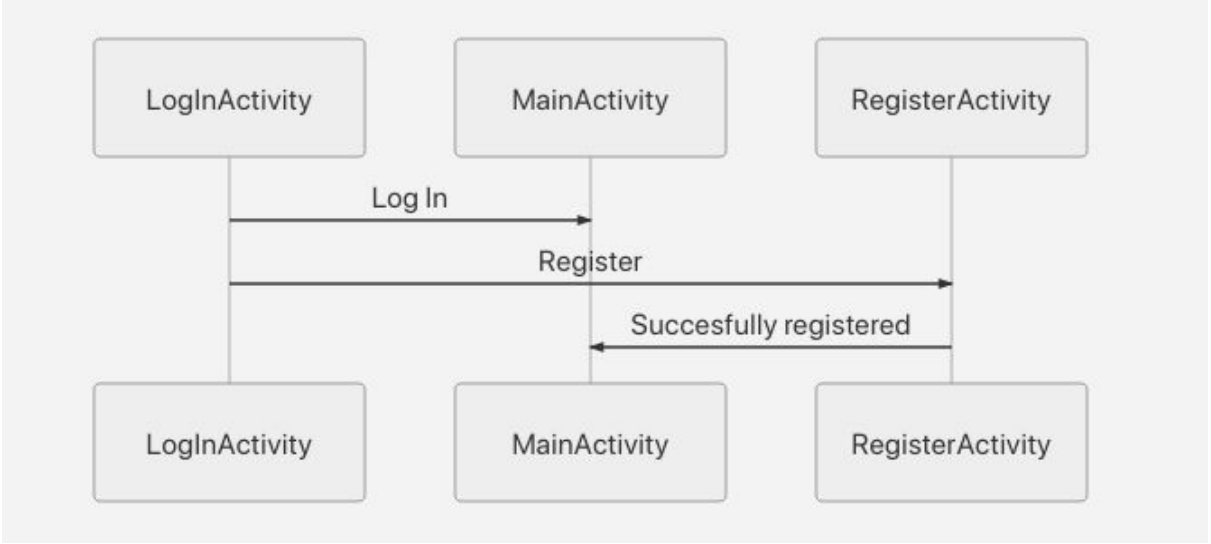
We kept track of the tasks we had to do using Trello:
https://trello.com/b/rZlAhM7P/mds

This platform made it easy to split the tasks between us, to visualize the parts of the application where we made progress as well as ToDo-lists, adding comments and setting deadlines. It made the whole development process a lot cleaner and faster, cutting out the need to make a video conference every couple days.

## 3. Design & architecture, UML

The design was mainly built on the Android Studio platform, with emphasis on communication, source control between team members and collaboration.

We also used Mermaid JS to outline several relations between activities, as explained further:




## 4. Source control

For the source control we decided to use github with its collaboration tools, which include forking, pull requests and branching. We also used local branching as outlined, and then the pull requests were resolved by the team project administrator.

Our project's github repository: Kars project.

## 5. Bug reporting

All our commits were merged on github, as well as bug fixes by team members, which when needed, provided an external perspective on some parts of code written by another member.

## 6. Build Tool

Kars uses Gradle from Google's Android Studio. This build tool encompasses certain design features such as automatic xml creation from java files and overrides Activity interfaces. We also used certain services and frameworks, the most important one being Google's Firebase, which we used for storing the data for each user.

## 7. Refactoring & code standards

We used certain refactoring methods, such as branching by abstraction and composing methods.

Some code standards from Java and Gradle were our guidelines, which improved the afterwards testing and some functions.

## 8. Design patterns

In the development period we used certain design patterns which are more practical than those taught in class, such as : MVVM Authentication, and some other singletons integrated in Android development in which we stored static methods and database-related data.

## 9. Tests

The tests we used were android-driven examples, available from the integrated development environment, as well as the firebase console testing, in which we had to pull some code from the developer platform to test the connection (and also use some ssh-keys to connect Android Studios to the online in-time database console)

We hope you will download our app in the future! (When it will be available on Google Apps)