Mobile Computing Report

Find my Car

Student Dorian Dascalu

# About the project

The project presented in this report is representing by an application dedicated for car localization and guidance of the driver.

The application will provide to the user the possibility to save the location of the car, in order to be easily found and recognized inside difficult areas.

The location will be provided by the GPS and guidance to the location by maps services. Additionally to this, on board camera and accelerometer will be used to have more flexible ways to save the location of the car.

All the data will be saved on a cloud in order to have a history of locations available.

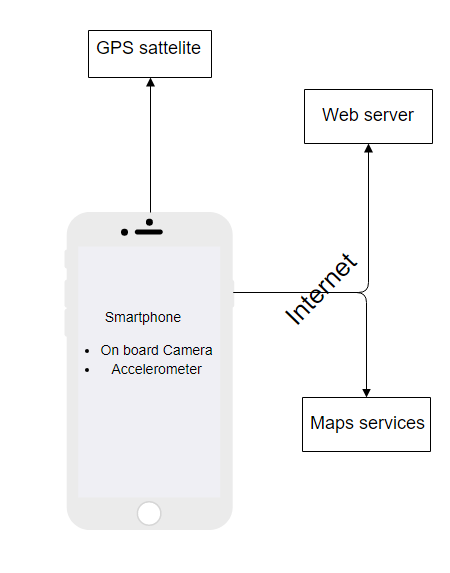
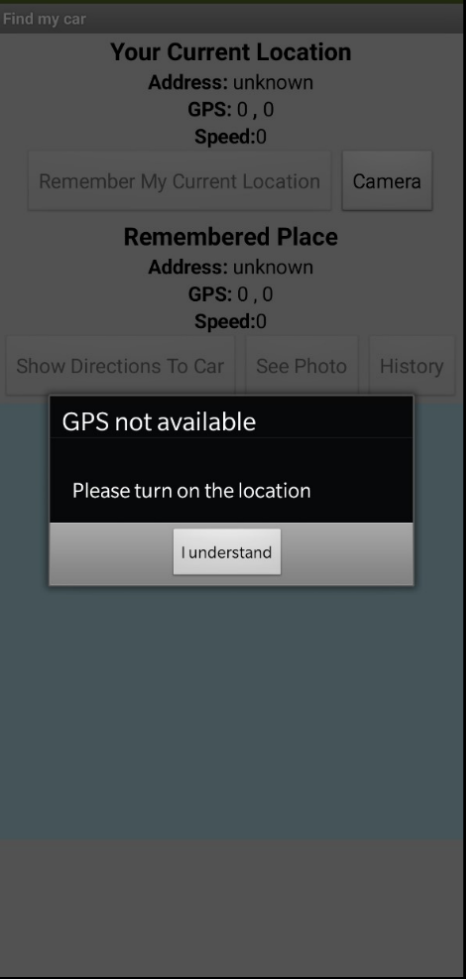
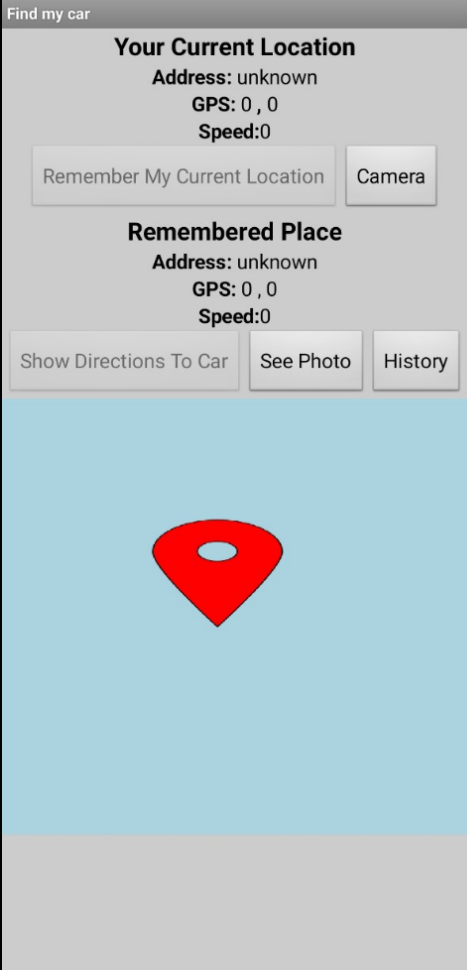
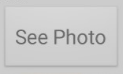


Fig 1 System high overview with interactions

# 2. Application wireframe

Keeping in mind the possibility of losing GPS signal due to different factors, the application will provide mechanism of saving the position in the both scenarios (GPS signal On/Off)



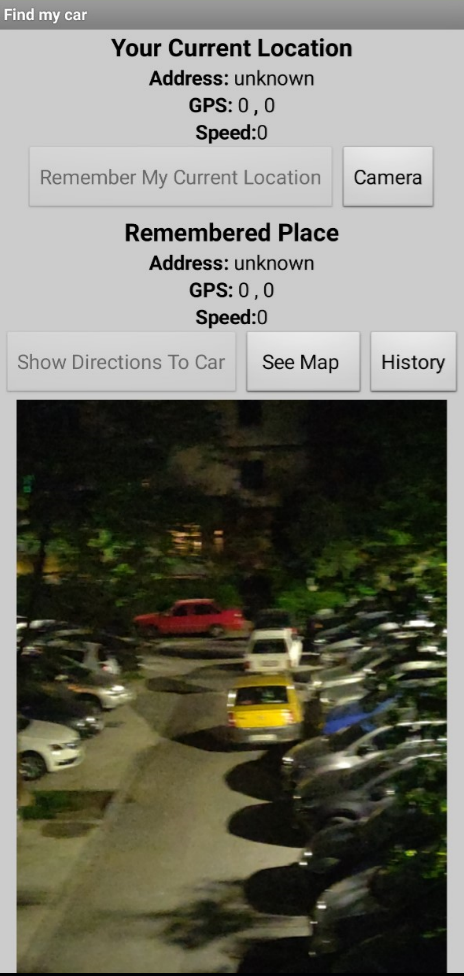
Save a photo with car position is the only functionality available in this scenario

Pop up notification to make visible the limitation of missing GPS signal

Start with no GPS signal available



Once the photo was made, Save photo functionality become available (not grey out)

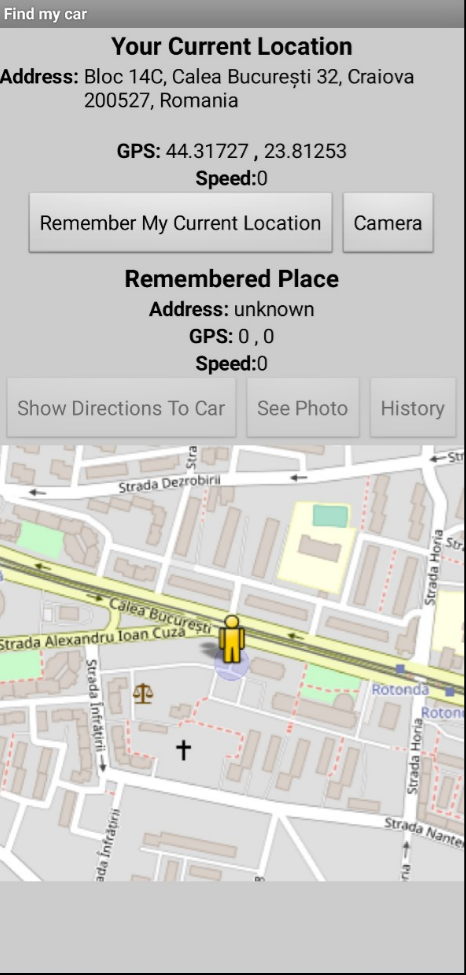
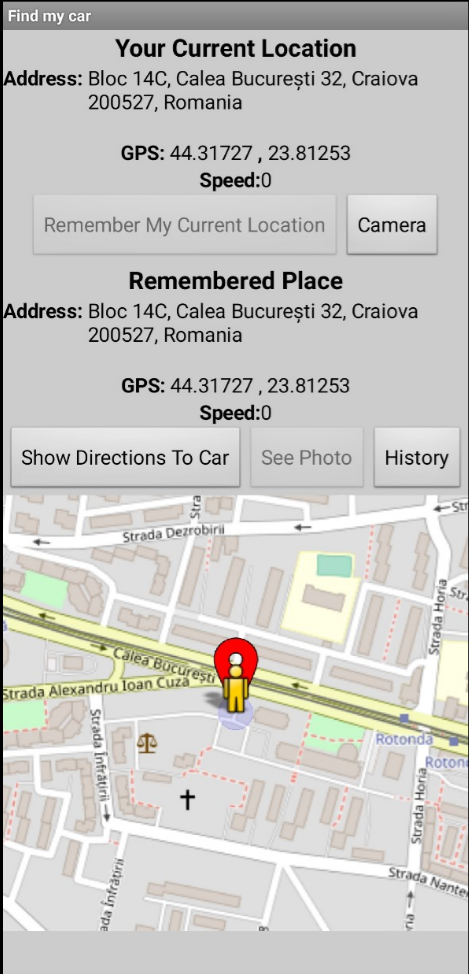


Start with GPS signal available

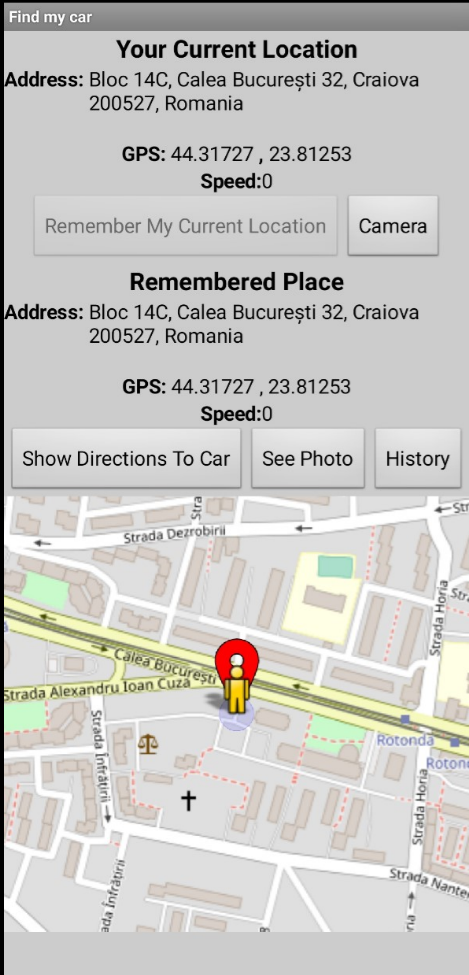
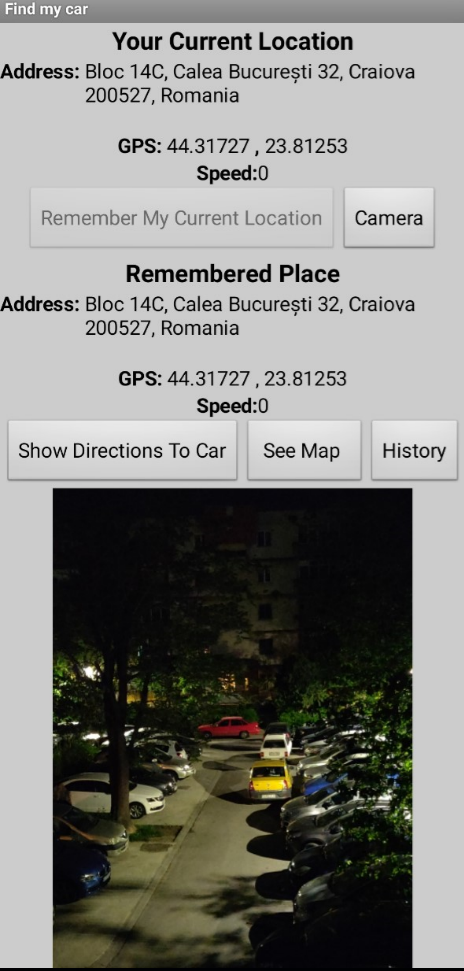
User can switch between Photo or Map perspective

User can return or clear the entries

History will not provide useful information since location is unknown



Once the location is available, driver can save his location, leading to :  
- saving and placing a marker on the map  
- allowing to show direction with maps services  
- saving the location on web data bases



Save photo functionality now saves also the location automatically

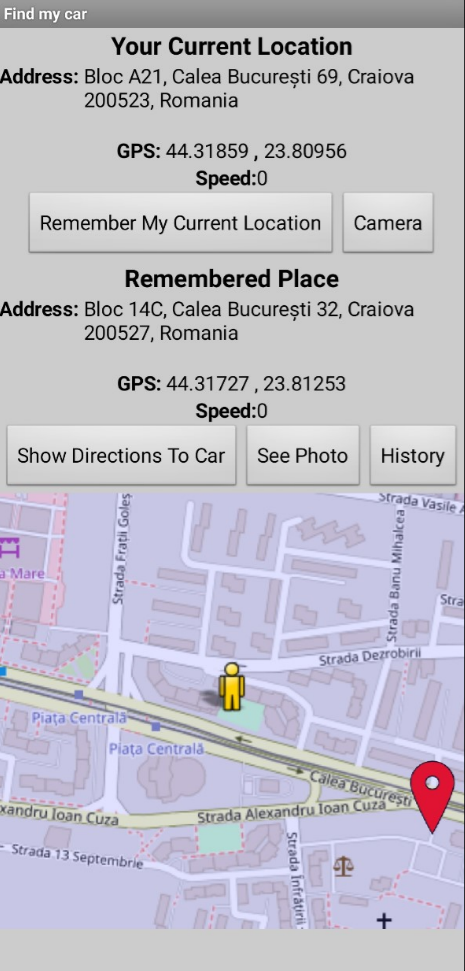
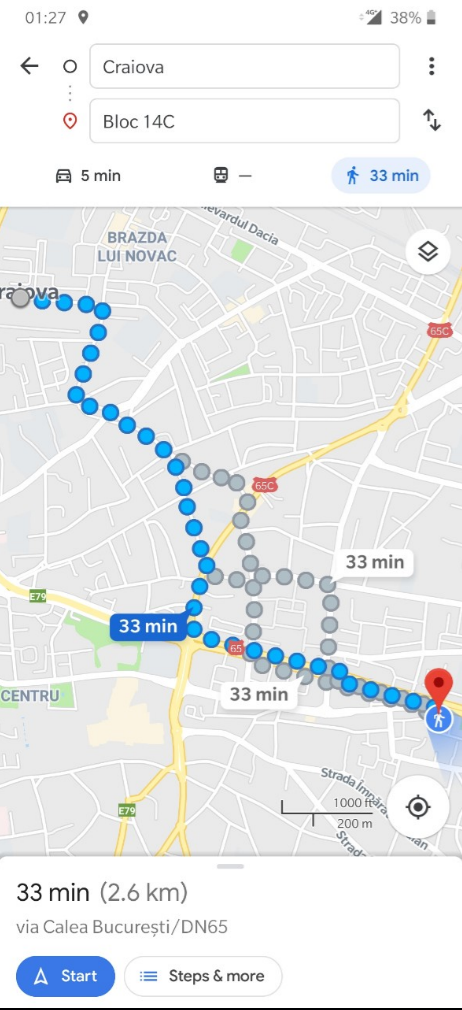
Possibility to swap as described above

Back from where the software came

Two location saved (one normal and one from camera)

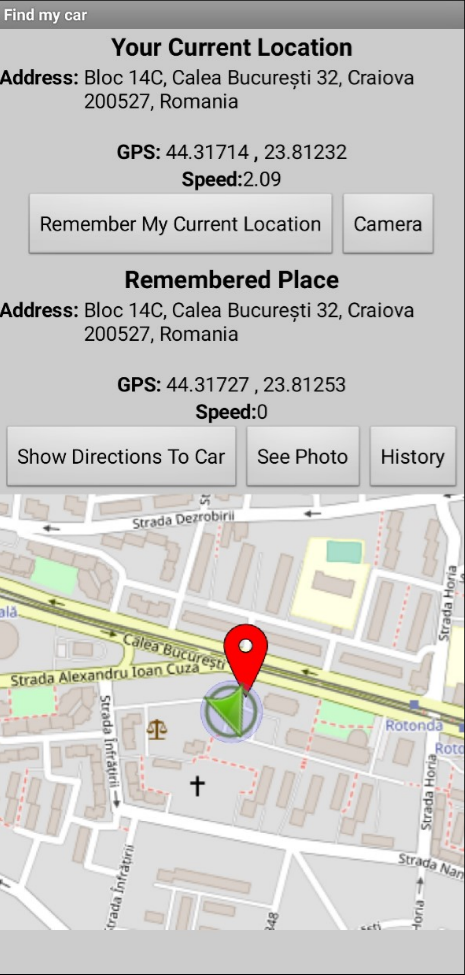


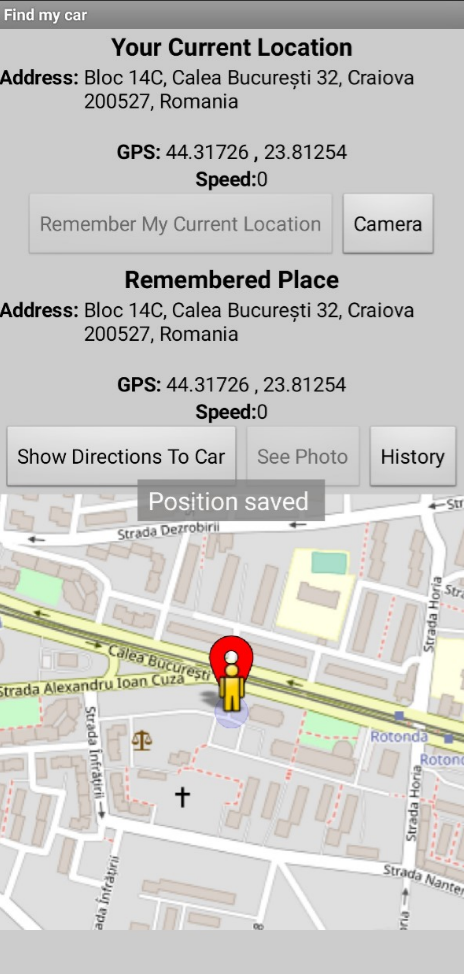
Clear entities will erase the data



Location changed

Show direction is using google maps functionality in order to provide guidance and traffic options





Application is capable to measure also the speed of current/saved location

Position will be saved if the device will be shaken

Fig 2 Application wireframe

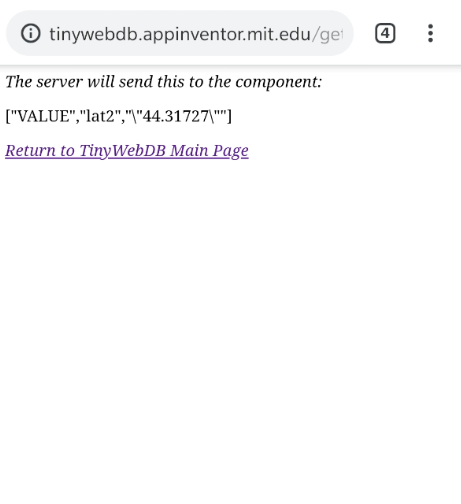
# 3. Server Side Functionality

Implementation of cloud based web server consist in MIT App Inventor included plugin called “Tiny Web DB“ with API : http://tinywebdb.appinventor.mit.edu

Example of handling the requests

On web handling of database is done as following :





The number after general Tags (Address, long, lat, speed) is referring to the general number of Tags in order to do not overwrite data

# This framework has the following APIs :

 sd Save the value “0” with tag “lat” concatenated with

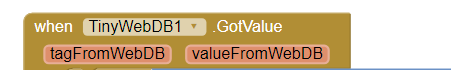
“global tag” to have a unique

value

Get value from tag “lat” concatenated

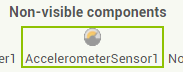
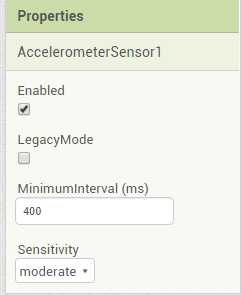
with “global tag”. This method will invoke a

a function trigger base which will return the value

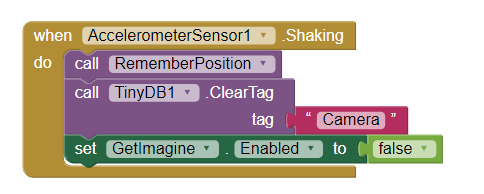


# 4. Movement

In order to use Accelerometer sensor, a non-visible component shall be added.

 The application will start 400 ms refresh configuration with a moderate sensitivity.

The role of the accelerometer in the application is to have a more friendly way to save the location. When shaking the device, an event is triggered and we start to save location.

 When accelerometer detects a shaking

action, it will call functionality to save position, and also clearing the current photo since is obsoleted

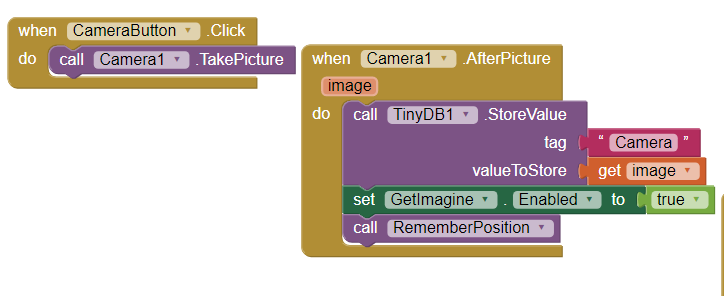
# 6. Images

In order to use Camera, a non-visible component shall be added.

 It does not have any initialization configuration.

The role of the Camera is to provide an alternative to saving the position of the car.

Additionally to the situation when GPS signal is not available, Camera will also save the position when the location is available.

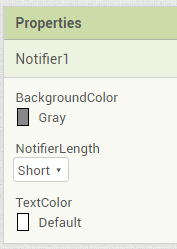
 When the button is pressed, camera is taking a picture. After the photo is saved, and event will be triggered to get the data and save to database

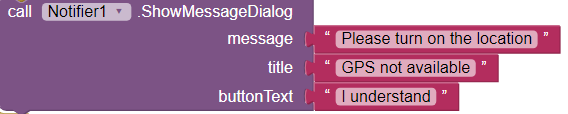
# 7. Wifi Listener

The application is not using Wifi module.

# 8. Notifications

In order to use Notifications, a non-visible component shall be added.

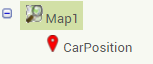
 Default configuration will consist in font color and the duration.

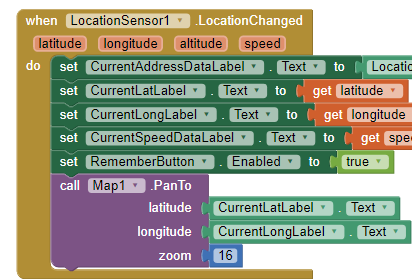
 To raise a notification, you have to select proper notification format and fill the messages box



# 9. Maps and Directions

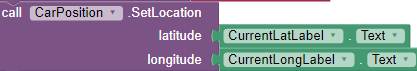
In order to use Maps, visible component shall be added.

 Default configuration will consist in initial position and customize of interface.

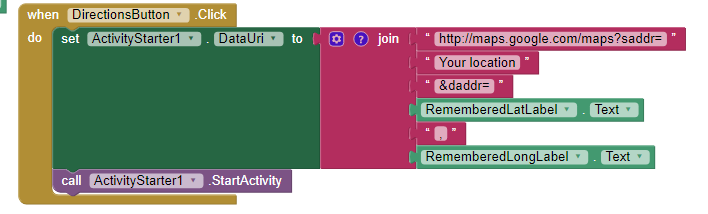
The application will show current position on mini-map from the screen when a new location is detected.

When location is changed, an event is raised where we are saving all the new values. Furthermore, we are setting a new location with latitude and longitude parameter and also zoom.

The marker is added when a new position is remembered in order to have the position of the car.

 The marker will be moved to the new current position.

In order to have the directions to the position, the application will invoke Google Maps services.



When the button is pressed, we are creating a string with current position as starting address “Your position” and destination address.

After that, application will launch the Activity and Google maps services will calculate the tracks.

Project location and information:

<https://github.com/DascaluDorian/Projects/tree/master/Mobile%20Computing%20Report>