

Laboratory 7

Car-Sharing (2 / 2)

Agenda

Introduction	2
Preconditions	5
Tasks	6
Evaluation	8

Introduction

This laboratory aims to implement more of the features from car-sharing application.

Feature: Sending car information to backend

Usually, the following information is sent to backend:

- Fuel consumption
- Location of the vehicle
- Various information regarding state of the car (based on sensors) - Remote diagnosis

This information may be used by customer, by the company (providing the car-sharing service) or both.

The fuel consumption is relevant for both.

- The fuel consumption is shown in the phone app. Would you, as a driver, want to rent a car without fuel or with battery on 0? Definitely not!
- The company also tracks this in order to send employees to fuel up the cars.
- Some companies also offer some discounts at the final prices for electric cars when the cars are left connected to a charging station.

The location of the vehicle is also used by both.

- The company must know at any point of time where the vehicle is.
- Many companies offer the car-sharing services within a limited area (example: inside a city). The backend will use this information and not allow the user to park the car and end the rental unless the car is within that specified geolocation.
- The user will also want to know where the car is so that he can reserve a car that is in his / her proximity and follow the map until (s)he reaches the car.

The remote diagnosis is useful for the company to know when the car needs service, take it there and update the car's status as 'in service' inside the phone application.

Feature: Registration of new user

In order to be able to use the car-sharing service, a person must create an account using a phone application.

The mandatory requirements are usually:

- Name
- Identification details
- Valid driver license – photograph is usually required to be uploaded in the application
- Payment method / Credit card
- Email address
- Mobile Phone Number
- Agreement of the 'Terms and Conditions' provided by the company

The account is validated by the backend by performing various checks on the data received. Based on this validation, the backend will authorize or not the driver for renting the car.

Please be aware that the authorization may also be revoked in various cases specified in the 'Terms and Conditions'. One example would be severe damage to the car during a car-drive.

The backend may also keep a history of the driver's car rides.

Feature: Validation at end rental

In order to make sure that the vehicle is left secured, the following are usually checked:

- All windows, sunroof and doors are closed
- The interior lights are closed
- Doors are locked
- Car is parked within the allowed location

When receiving the request of end-rental, the backend checks this information. In case something is not as expected, the driver will receive a notification on the phone application and will need to take corrective action. Only once the backend confirms that everything is in order, the end-rental will be marked as successful.

In case the trip has ended not successfully, the driver may be liable for any damage to the vehicle that may result.

Feature: Calculation of the price

There are multiple methods of payment for a car-sharing ride. The most common are the following.

- Trip package => The driver chooses from various trip packages offered by the company. Examples can be: daily trip packages, fixed kilometers trip packages etc. In case the driver exceeds the benefits from the trip packages, he will pay the extra costs as specified in the description of the trip package (usually, one of the below methods).

- Per-Minute Vehicle Fee => The driver is charged based on the duration of the entire drive. This option usually has also limitations like a 24 hour time limit and a maximum number of kilometers.
- Per-kilometer Vehicle Fee => The driver is charged based on the number of kilometers of the entire drive. This option usually has also limitations like a 24 hour time limit and a maximum number of kilometers.

The price is calculated after the rental has ended and it is visible in the phone application for car-sharing. Some companies also provide an invoice by mail to the driver.

Please be aware that certain companies may also offer some special fees customized by the car-sharing habits of the driver (based on the history of the driver's car rides).

Preconditions

The tasks for today continue laboratory 1. Thus, you will need to have the code from laboratory 1.

Tasks

Add the following functionalities:

Exercise 1

This point aims to implement the feature: sending information to backend.

Inside “Car [Client]”, the information received from sensors inside car (other Electronic Control Units / ECUs) will be simulated by reading a JSON file “car_info.json” containing:

- Fuel consumption
- Location of the vehicle => latitude & longitude
- State of the lights => On / Off
- State of the Doors => Open / Closed / Locked

Please send the car information from car client to server.

Note:

- You must modify L03_car_carSharingClient.py. Check where “Exercise 1” is specified. You have there a description of what must be done.
- car_info.json is already parsed and available in self.car_info.
- Response from „Car [Client]” to Server must be in the format „car-info <type-of-info> <info-value>” for the server to be able to parse it correctly.

```

Carsharing Server
(2022-02-23 18:18:55) : Parsing Json File...
(2022-02-23 18:18:55) : key=restricted
(2022-02-23 18:18:55) : [restricted] = ['5422', '6533', '9877', '6678', '1564', '2221']
(2022-02-23 18:20:02) : Started.
(2022-02-23 18:20:04) : 48b6e175-94c4-11ec-a0dd-44e51728cb42 register-client IS21KSM
(2022-02-23 18:20:04) : received command from client =48b6e175-94c4-11ec-a0dd-44e51728cb42 register-client IS21KSM
(2022-02-23 18:20:04) : client_id=48b6e175-94c4-11ec-a0dd-44e51728cb42
(2022-02-23 18:20:04) : command_id=register-client
(2022-02-23 18:20:04) : acknowledge-register "IS21KSM"
(2022-02-23 18:20:04) : handle message end
(2022-02-23 18:20:17) : request-info latitude
(2022-02-23 18:20:17) : 48b6e175-94c4-11ec-a0dd-44e51728cb42 car-info latitude 47.1585
(2022-02-23 18:20:17) : received command from client =48b6e175-94c4-11ec-a0dd-44e51728cb42 car-info latitude 47.1585
(2022-02-23 18:20:17) : client_id=48b6e175-94c4-11ec-a0dd-44e51728cb42
(2022-02-23 18:20:17) : command_id=car-info
(2022-02-23 18:20:17) : received car info
(2022-02-23 18:20:17) : car-info latitude 47.1585
(2022-02-23 18:20:17) : handle message end

```

Connected Start Send Clear Exit

Fig.1: Server receiving latitude value

- The steps for testing are

Server -> Start

Car [Client] -> Connect

Car [Client] -> Register

Server -> Send „IS21KSM#request-info latitude”

Exercise 2

This point aims to implement the feature: registration of new user.

Before phone app (client) sends a ‘register-client’ request to server, it must check that user has introduced the mandatory information (ID / CNP, age, driver license, email, phone).

For this academic application, no regex is necessary to verify that the input is in the correct form.

Please add the needed code inside method `register_client` in order to do such check and to register this client to Server (similar to registration done in “Car [Client]”).

Note:

- You must modify `L03_phone_carSharingClient.py`.
- The method for introducing the mandatory user info and storing them already exists: `register_user`.

- The steps for testing are

Server -> Start

Phone App [Client] -> Connect

Phone App [Client] -> Register User

Phone App [Client] -> Register

Exercise 3

This point aims to implement the feature: validation at end rental.

The phone app client will request ‘end-rental’ from Server.

When Server receives this request, Server will request the Car Client to ‘car-end-rental’.

When Car Client receives this request, Car Client will check if the doors are closed and if the lights are off (this information is stored in `self.car_info`, please check Exercise 1). If they are, it will respond to server “response-end-rental success”. Otherwise, it will respond to server “response-end-rental success”. Server will forward this status to Phone App Client.

Please modify the “Car [Client]” in order to check that doors are closed and lights are off.

Notes:

- In order not to interfere with previous laboratory, the ‘start-rental’ function is not used in this laboratory. Thus, if there is someone who missed the previous laboratory, (s)he can recover.
- You must modify L03_car_carSharingClient.py
- The steps for testing are

Server -> Start

Phone App [Client] -> Connect

Phone App [Client] -> Register User

Phone App [Client] -> Register

Car [Client] -> Connect

Car [Client] -> Register

Phone App [Client] -> server-end-rental IS21KSM

Evaluation

Score:

- a) 2 point(s)
- b) 2 point(s)
- c) 2 point(s)