WEB SERVICE PROJECT REPORT

Github: https://github.com/romainducrocq/Web-Service-Project

Build and test

To build the project from the sources, please refer to the "**Build project from scratch**" section in the Github Readme, and test the project with the values provided in the "**Test values**" section.

Here is an overview of the git repository tree:

- Javadoc : the generated java documentation
- WS_project :
  - Bank : the bank web service
  - IfsCarsService : the IfsCars web service
  - IfsCarsServiceClient : the IfsCars web service client
  - RentalProject : the rental RMIs
  - SQL files for database initialization
- README.md
- Report.pdf

Overall architecture

The project is divided in 2 main parts that we can link to 2 different use cases : the rental part and the sale part.

The rental application will communicate with 2 RMI objects : one is dedicated to the employees management and the other one to the rental management.

The sale application communicates with a Web Service called IfsCarService that is in charge of handling the sale process. It calls the Web Service BankManager that is in charge of the banking process (checking the customer is solvent and making the payment). The WS BankManager uses real-time exchange rates by contacting the WS CurrencyServer (provided by the site http://fx.currencysystem.com).

3 MySql databases are used :

1. one for the employees, used by the RMI Employee management object;
2. one for the vehicles, used by the RMI Rental management object and the WS IfsCarService;
3. one for the bank accounts, used by the WS BankManager.

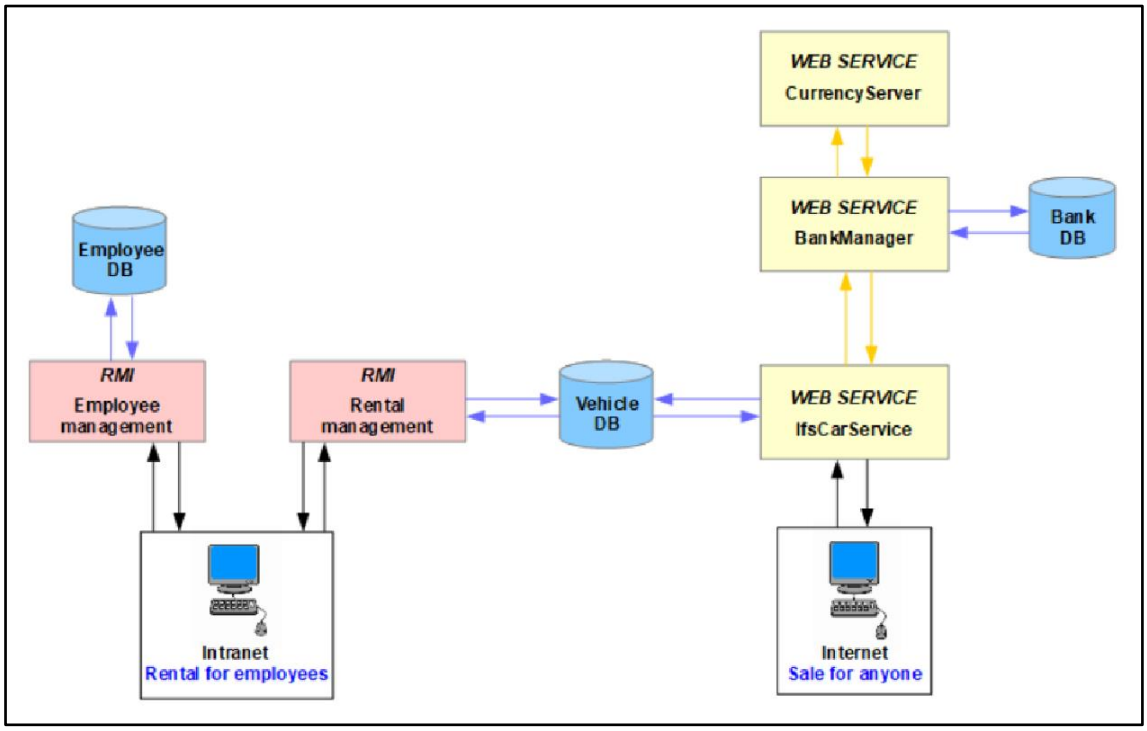


Figure 1 : architecture of the project

Interface Architecture

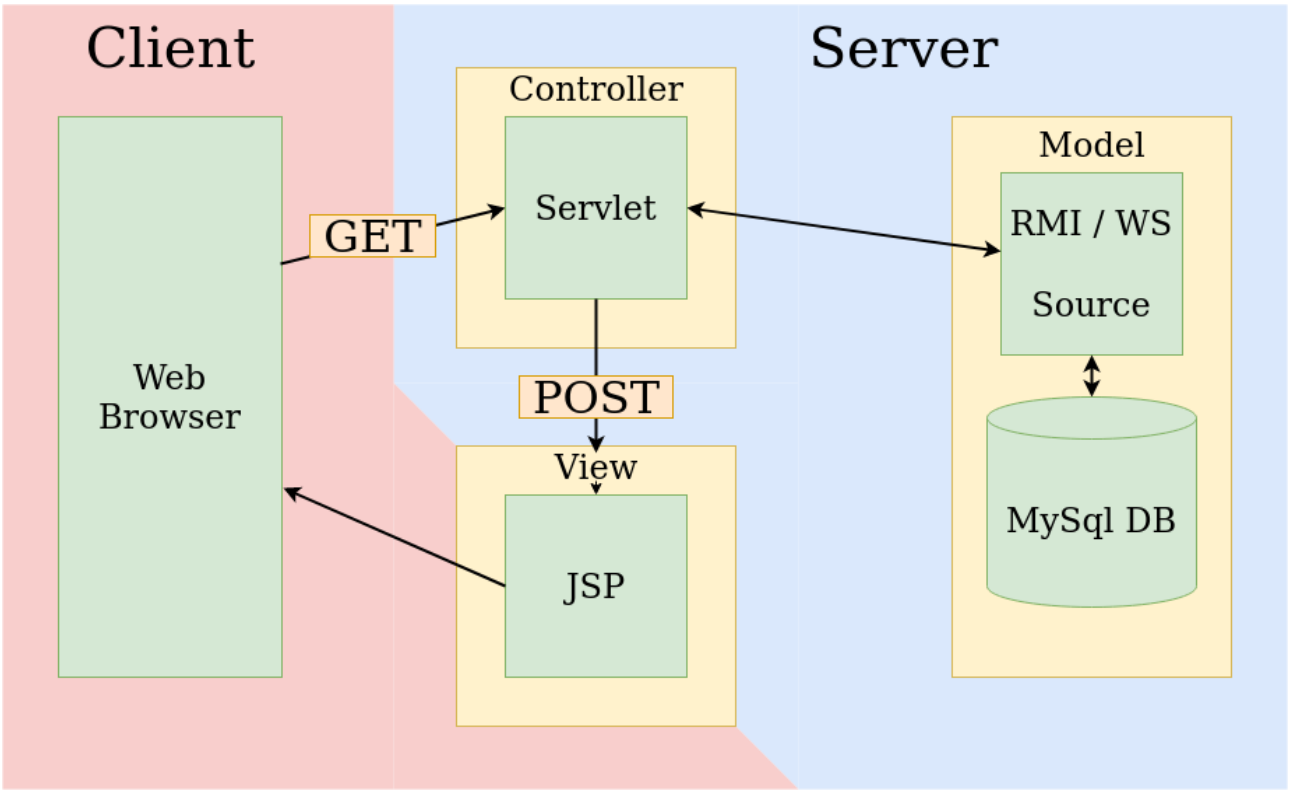


Figure 2 : architecture of the interfaces

We have implemented web interfaces for both our Rental RMI and our Sales WS, using Servlets and JSPs. These technologies are designed to run Web Sites on Apache Tomcat Servers with Java controllers.

The Model of our applications is designed in the RMI / WS sources and the MySql Database, and works without a graphical interface with console API requests. Here, we added a Controller and a View on top of it.

The Servlets account for the Controllers in the architecture. The Servlet is a Java program which makes the link between the Model and the View. On the one hand, it makes API calls to the source to gather data and perform operations. On the other hand, it handles HTTP requests by receiving GET requests from the browser and sending data to the View through POST requests. Therefore, each servlet is associated with a single Web Page and a single View, and performs the server logic of the page.

The JSPs are the Views of the architecture. They are the server pages stored at the root of the Web Site. The JSP files are augmented HTML pages incorporating JSP tags. They contain the classical Web technologies, which are HTML, CSS and JavaScript, and include Java code in the JSP tags, used to receive the data posted by the servlet at page load. Once the page is loaded on server-side, it is published in the browser by the JSP. Since no java code can be executed after page load, we parse the data into JavaScript objects to manipulate it within the client-side logic. Finally, when the page is changed, the client-side Javascript performs a new HTTP POST request to send data to the servlet of the next web page.

A good way to understand this is to say that Servlets and JSPs allows to run Web Pages with Java on an Apache Tomcat Server just like ASPs allows to run C# on an IIS Server, or PHP on an Apache HTTP Server.

Detailed operation

Rental application

In the rental part, the employees of the company can rent some vehicles.
For this purpose, we have developed a kind of intranet by using Servlets and JSP technologies.
The employee is first asked to authenticate. The authentication is validated by calling the RMI object handling the employees. It loads the employees and their attributes from a MySQL database using the JDBC API.
If the user is correctly authenticated, he can choose either to rent a vehicle or to return a vehicle.

If he wants to rent a vehicle, the available vehicles are displayed by asking the RMI object handling the rental park (previously loaded from a MySQL database). The user chooses one vehicle. Depending on the availability of the vehicle, a message is displayed and a mail is sent to the user (using the JavaMail API).

When the user wants to return a vehicle, he is asked to enter the vehicle identifier, a note and the condition of return. The RMI rental manager handles the return by making the car available. If somebody was waiting for this car, an email is sent to inform him that the vehicle is now rented by him.

Sale application

In the sale part, any user can buy some vehicles. For this purpose, we have developed a web site by using Servlets and JSP technologies.
The internet user will select a car in a list of cars available for sale. The web application asks the web service IfsCarsService that requests the vehicles database to get the list of available cars.
Then the user adds the vehicle(s) to the cart. The cart is handled by the WS IfsCarsService also.
The user can select the currency in which the price must be displayed. The WS IfsCarsService is the access point for currency handling. It then calls the Web Service BankService that calls the WS currency server.
The payment validation process is done by the WS IfsCarsService that uses Web Service BankService. This WS accesses a MySQL database to check that the user is solvent.

Basic scenario

For testing, we have created several entries in the different databases. Here are 2 scenarios, one for each application that show the different features of our project.

Rental application

Order	Use case	Steps	Expected result
1	Natacha rents the Clio of the year 2019.	Load the page http://localhost:8080/RentalProject/authenticate. Username : ngrumbach Password : badpwd Back to the previous page. Username : ngrumbach Password : password Choose « Rent a car » Choose the Clio of year 2019 (first car in the list) and validate.	Authentication error message Validation message displayed in the web page Validation email sent to Natacha
2	Romain wants to rent the Clio of the year 2019.	Load the page http://localhost:8080/RentalProject/authenticate. Username : rducrocq Password : password Choose « Rent a car » Choose the Clio of year 2019 (first car in the list) and validate.	Warning message displayed in the web page : the car is already rented. Romain has been added to the waiting list. Warning email sent to Romain
3	Alexandre wants to rent the Mustang.	Load the page http://localhost:8080/RentalProject/authenticate. Username : atherond Password : password Choose « Rent a car » Choose the Mustang and validate.	Validation message displayed in the web page Validation email sent to Alexandre
4	Natacha returns the Clio of the year 2019.	Load the page http://localhost:8080/RentalProject/authenticate. Username : ngrumbach Password : password Choose « Return a car » Enter the id 1, the note 4 and validate.	Validation message displayed in the web page Validation email sent to Romain : he is now renting the car.
5	Alexandre wants to see the Clio that can be rented.	Load the page http://localhost:8080/RentalProject/authenticate. Username : atherond Password : password Choose «Rent a car »	The note of the Clio of year 2019 should be 4.

Sale application

Order	Use case	Steps	Expected result
1	Janis Joplin hesitates between the Peugeot 5008 and the Ford Mustang. Finally, she wants to buy the Ford Mustang.	Load the page http://localhost:8080/IfsCarsServiceClient/index Click on the center panel. Add to cart the Peugeot 5008 vehicle. Click on the Cart tab. Remove the car from the cart. Click on the Shop tab. Add the Ford Mustang to the cart. Click on the cart tab. Proceed to checkout. Change the currency and choose USD. Enter the payment informations : Name: Janis Joplin Card number : 0000 1111 2222 3333 CVV : 4444 Expiracy date: May 2025 Click on “Confirm and Pay”	 The Peugeot 5008 disappears of the list of available cars. It appears in the cart. The cart becomes empty. The Peugeot 5008 is available in the list of cars. The Ford Mustang disappears from the list of available cars. The Ford Mustang appears in the cart with a price in Euros. The price is displayed in USD. Error message (Janis has not sufficient funds).
2	Jimi Hendrix wants to offer the Mustang to Janis Joplin.	Load the page http://localhost:8080/IfsCarsServiceClient/index Click on the center panel. Add the Ford Mustang to the cart. Click on the cart tab. Proceed to checkout. Enter the payment informations : Name: Jimi Hendrix Card number : 0123 4567 8910 1112 CVV : 1314 Expiracy date: December 2023 Click on “Confirm and Pay”	 The Ford Mustang disappears from the list of available cars. The Ford Mustang appears in the cart with a price in Euros. The price is displayed in EUR. A validation message is displayed.

Additional features

We have written comments in source code and generated the documentation in HTML format thanks to JAVADOC tool.

To work in a collaborative way, we have used the Github platform.

Difficulties encountered

The development of the RMI objects and the Web Services have been done quite easily.
We have spent a lot of time in creating the user interfaces because we did not know the Servlets and JSP technologies.

At some moments, we had doubts on the architecture : what is the pertinence of the WS IfsCarsService whereas we have a user interface ?

During the development, we were also faced with compatibility problems between the different technologies we used (JSP, JDBC...).

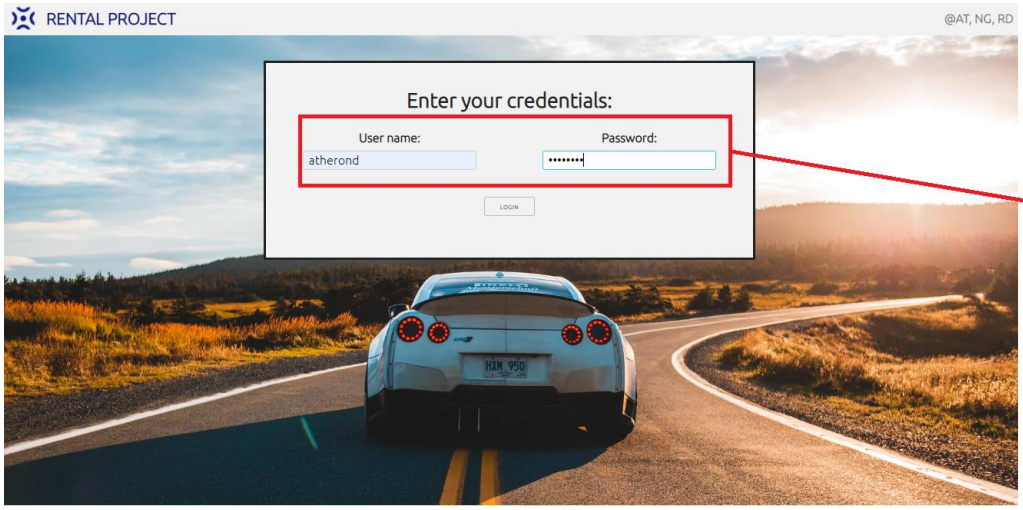
User manual

Summary:

1. How can I rent a car?

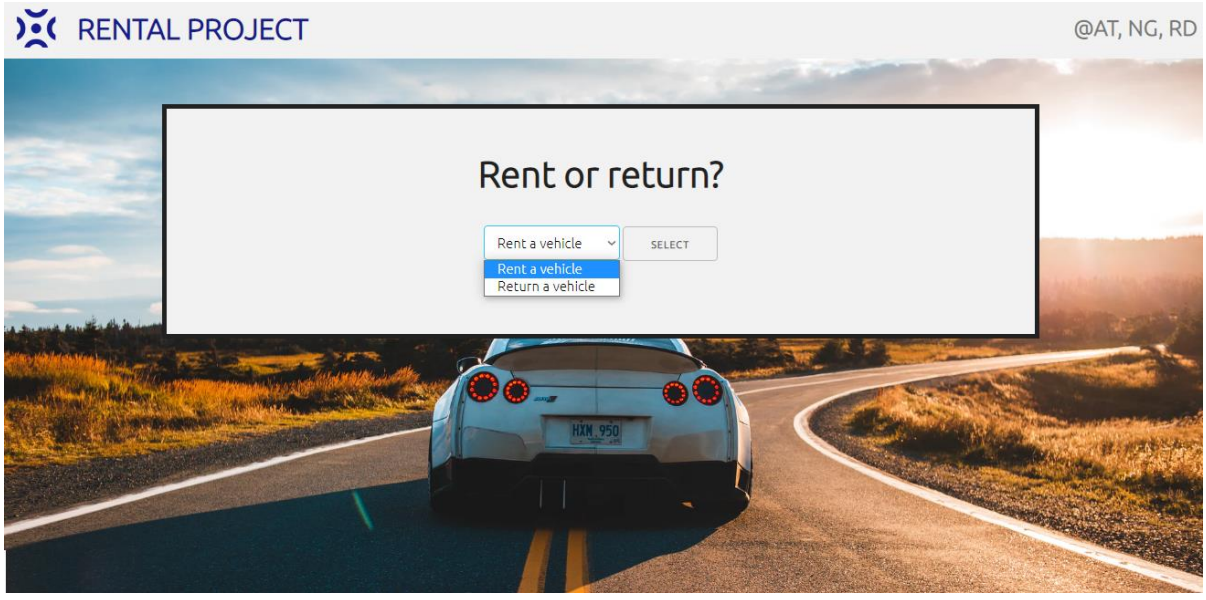
1.1 Connection to the service

By going to this url: http://localhost:8080/RentalProject/authenticate you can access the rental car service.
You will first arrive in the login page where you'll be asked to enter your user name and password.



1.2 Rent or return a car?

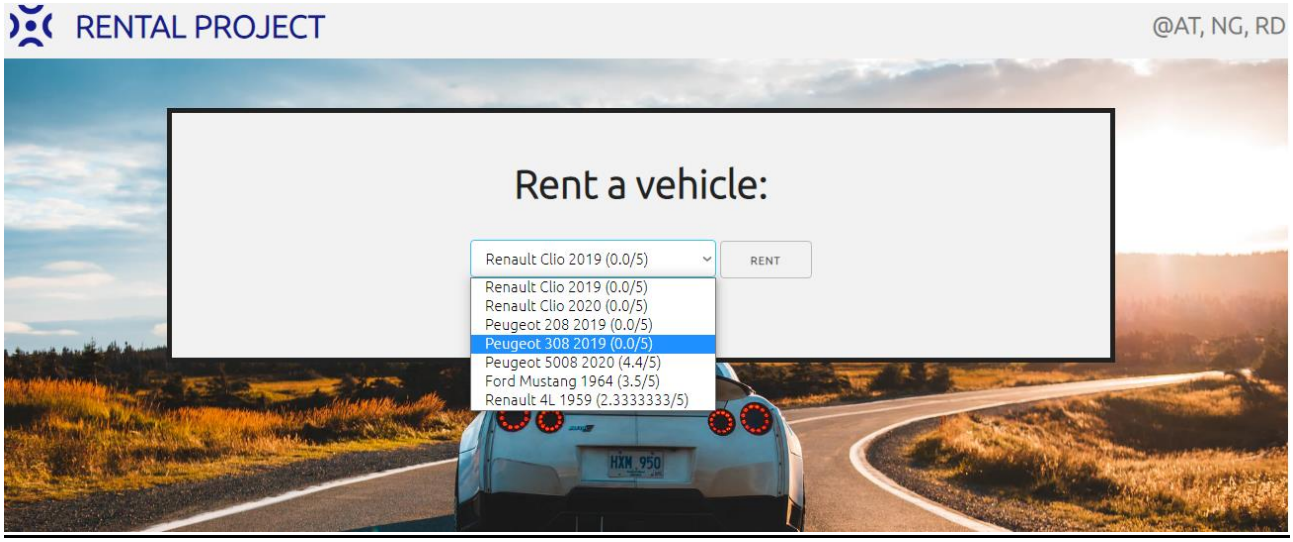
Once you've put your employee credentials you will be asked if you want to rent a new car or return a vehicle.
If you want to rent a car, select the first choice of the dropdown list and click on select. Then go to point 1.3 to see how to rent a car.
If you already have rented a car and you want to return it, select the second option of the dropdown list and click on select. Then go to point 1.4 to see the steps on how to return it.



1.3 How to rent a car

1.3.1 Choose your vehicle

If you selected the rent option on point 1.2, you'll arrive on this page, asking you to select the vehicle you want to rent:



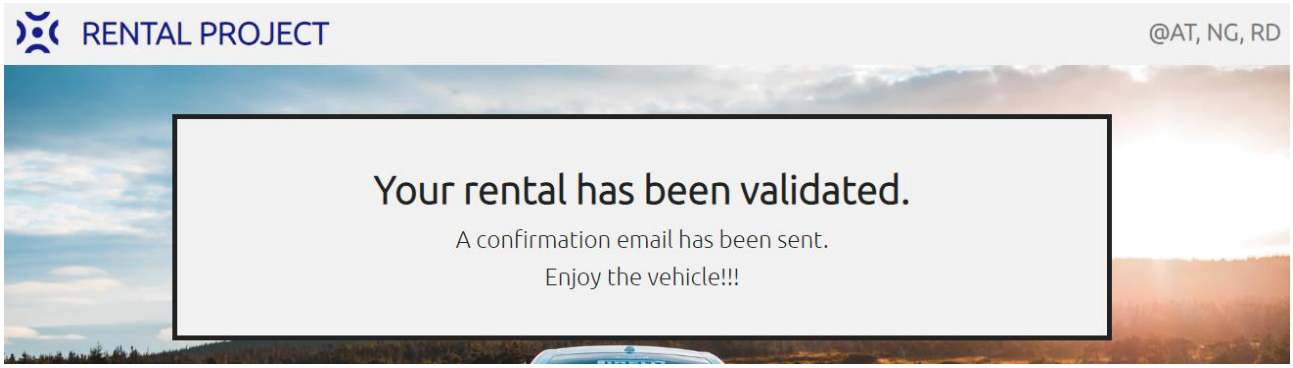
The dropdown list shows all the vehicles that you can rent.
The informations of each vehicle are:

- Brand
- Model
- Year
- Rating

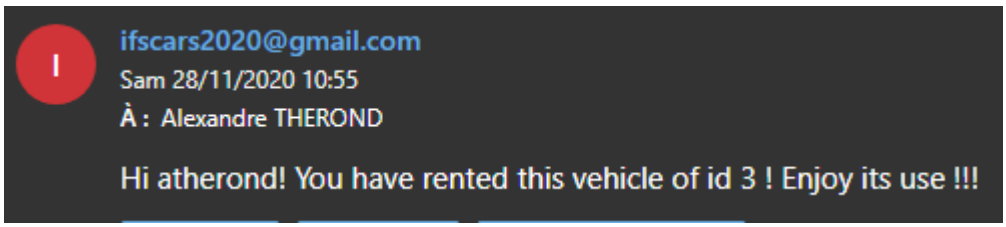
Once you have made your choice, click on the rent button.

1.3.2 Directly have access to your vehicle

If no one else is renting the selected car, this information message should appear:



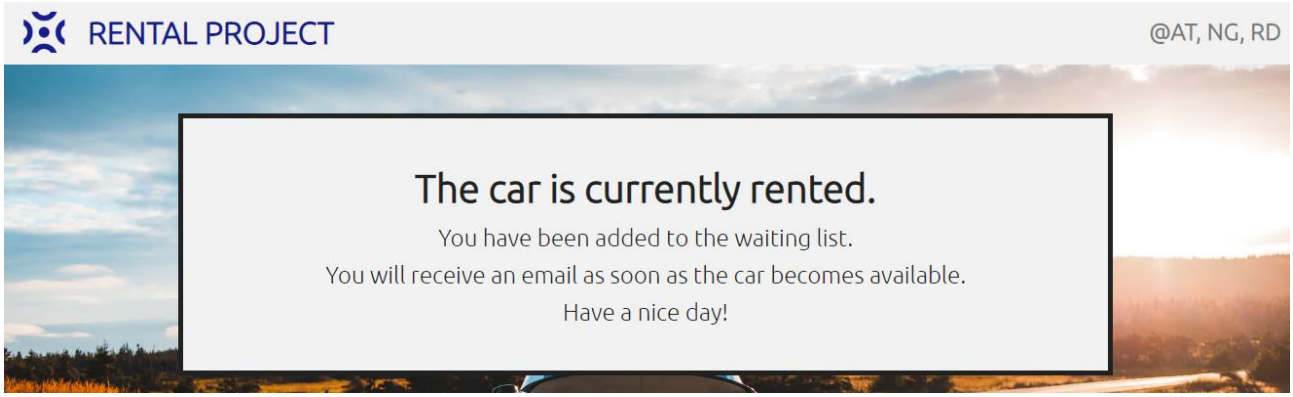
Verify your mail-box, you should have received a confirmation of renting, showing the id associated to the vehicle from the ifscars2020@gmail.com mail:



PLEASE KEEP THIS EMAIL.
You will need the id of the vehicle to return it.
But for now, enjoy the car!

1.3.3 You have been put in the waiting list

If after selecting a vehicle to rent, this message appear:



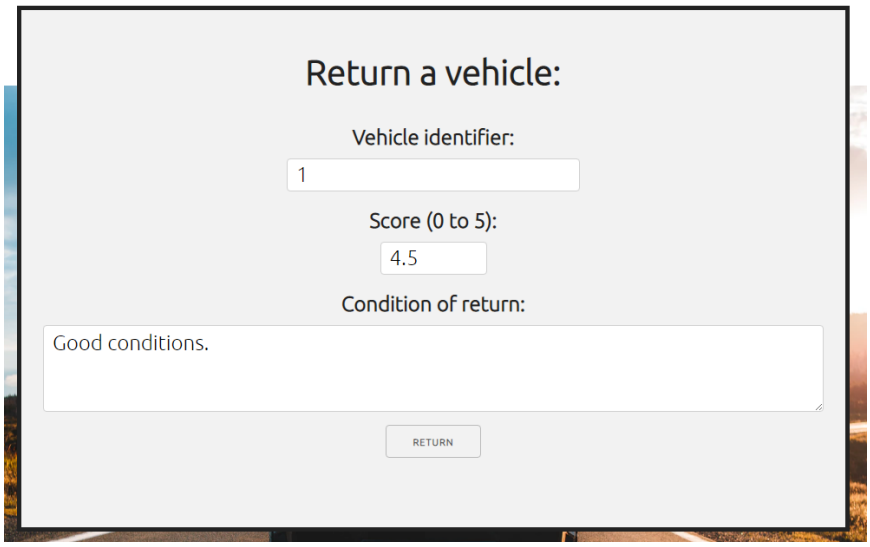
The car you had selected is not currently available and is already being rented by another employee.
You will receive an information email once the car is ready to use.

Once you receive it, please keep the id you have been associated with and enjoy the car!

1.4 How to return a car

You finish using your car and you want to return it? Here's how to do it.

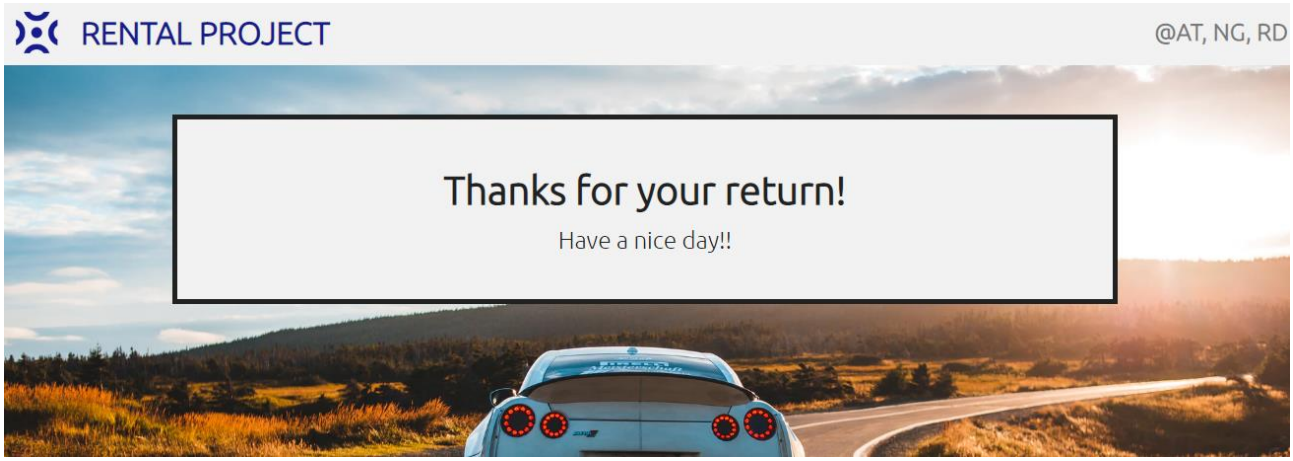
Once you have selected the second option of point 1.2, you will arrive on this page:



Please indicate:

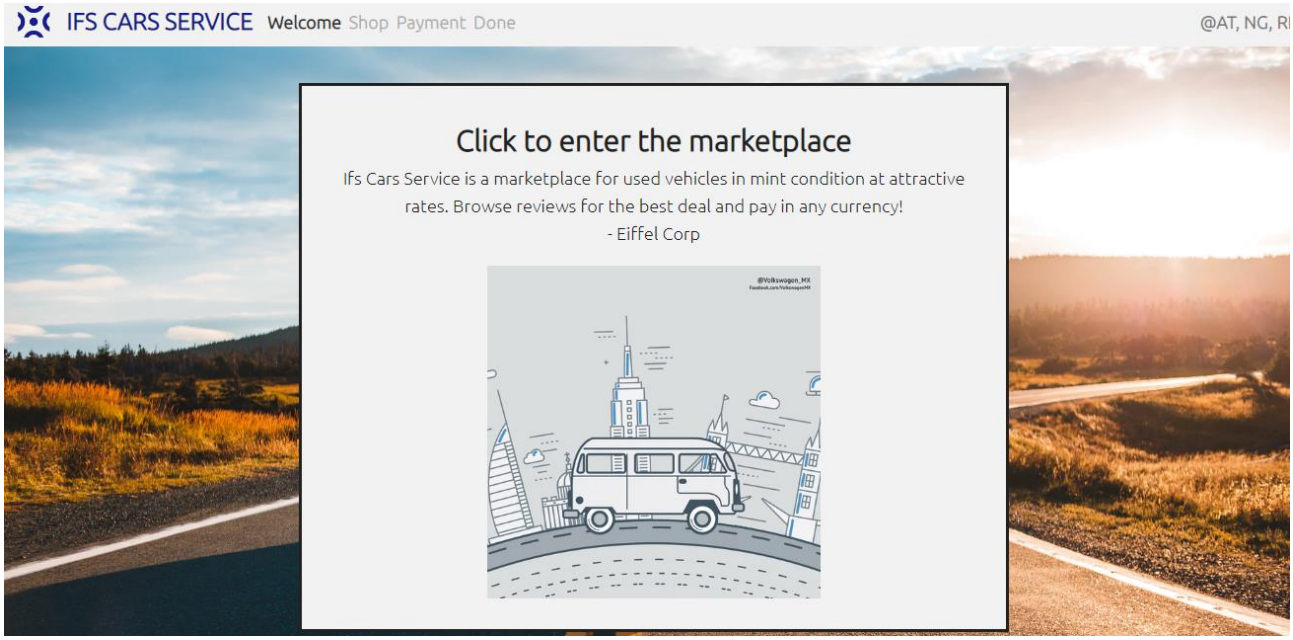
- The car id you have been granted, you can find it in your mail-box from ifscars2020@gmail.com
- The score you give to this vehicle
- A message about how is the car condition on your return

If this message appears, you have successfully returned the car!



2. How can I buy a car?

To buy a car, you need to access the IFS Cars Service, accessible by this url:
http://localhost:8080/IfsCarsServiceClient/index



And simply click on the image to access the service.

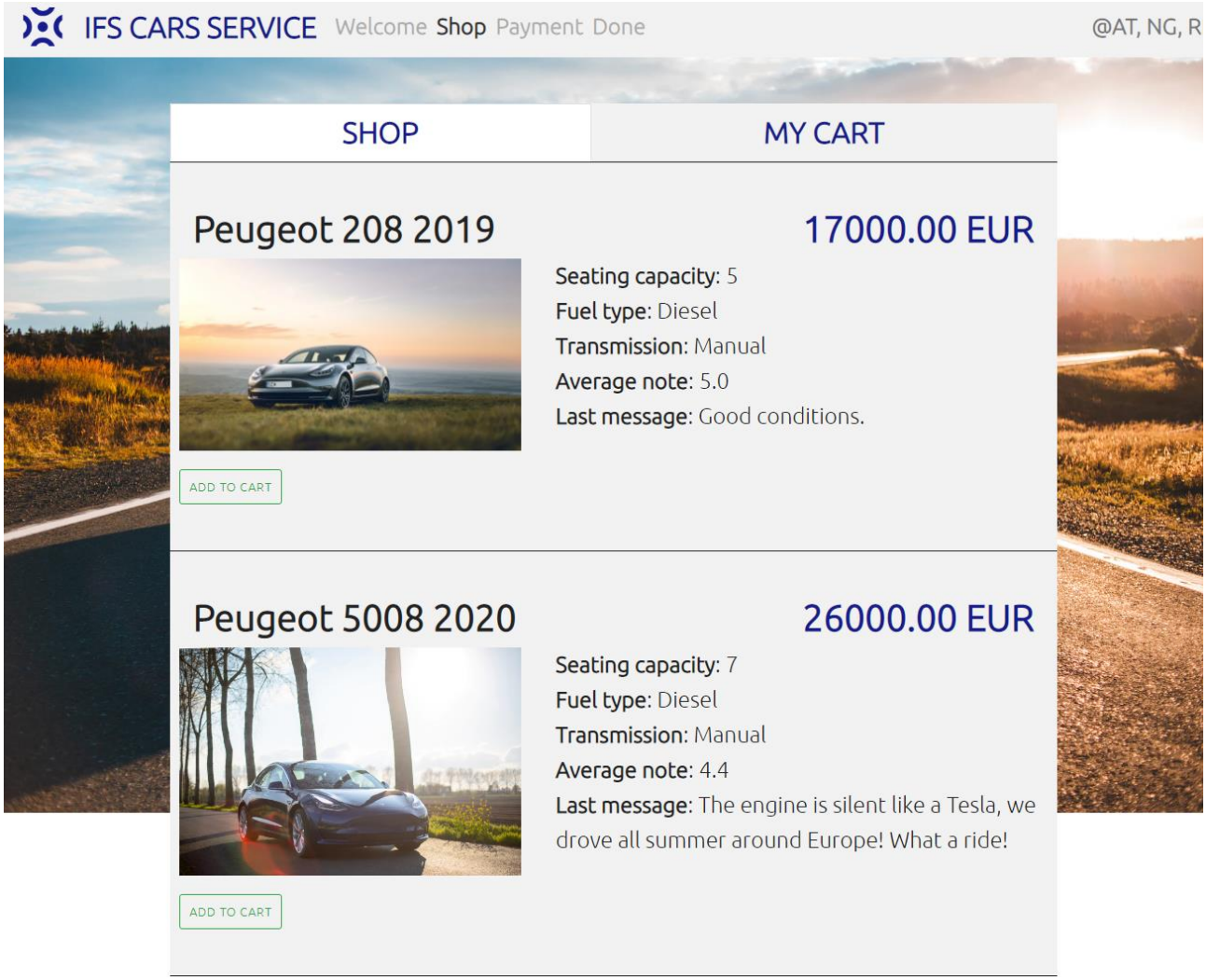
2.1 Choose your car

After clicking on the image you will arrive in the shop section of the service.

2.1.1 Cars information in SHOP section

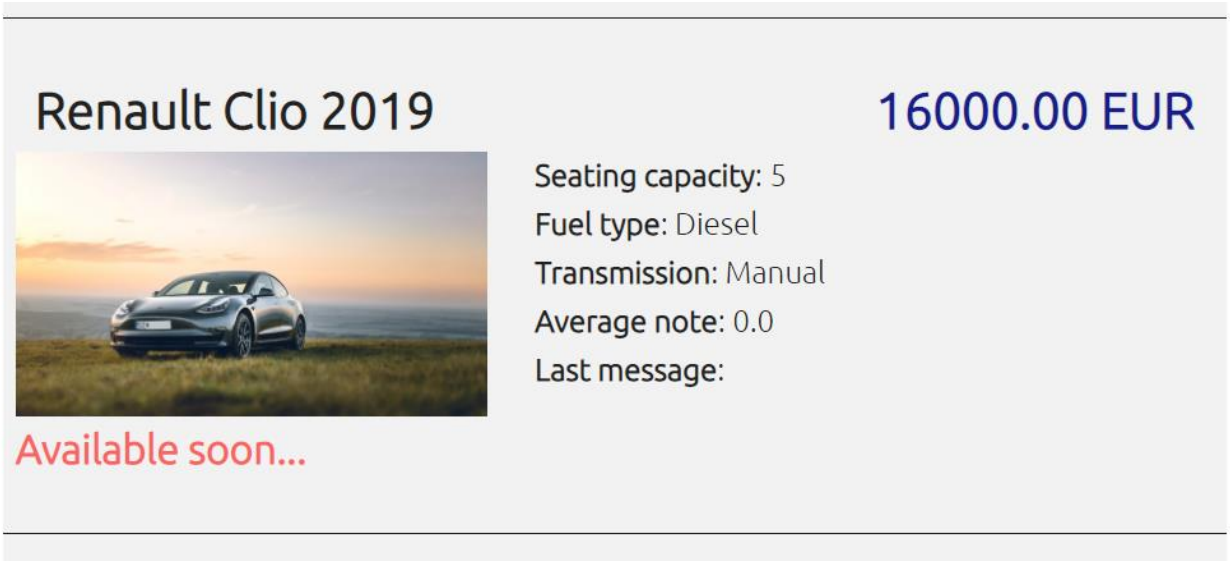
Here you can look at all the cars. Be careful about the car situation. Indeed only already rented cars can be purchased. You can see the availability of a car if the button “ADD TO CART” is indicated below the car picture.

You can note that the car given as an example in this manual has been updated below:



The last message and the note correspond to the message given when the car has been returned.

An example of unavailable car:



For every vehicle you can see this informations:

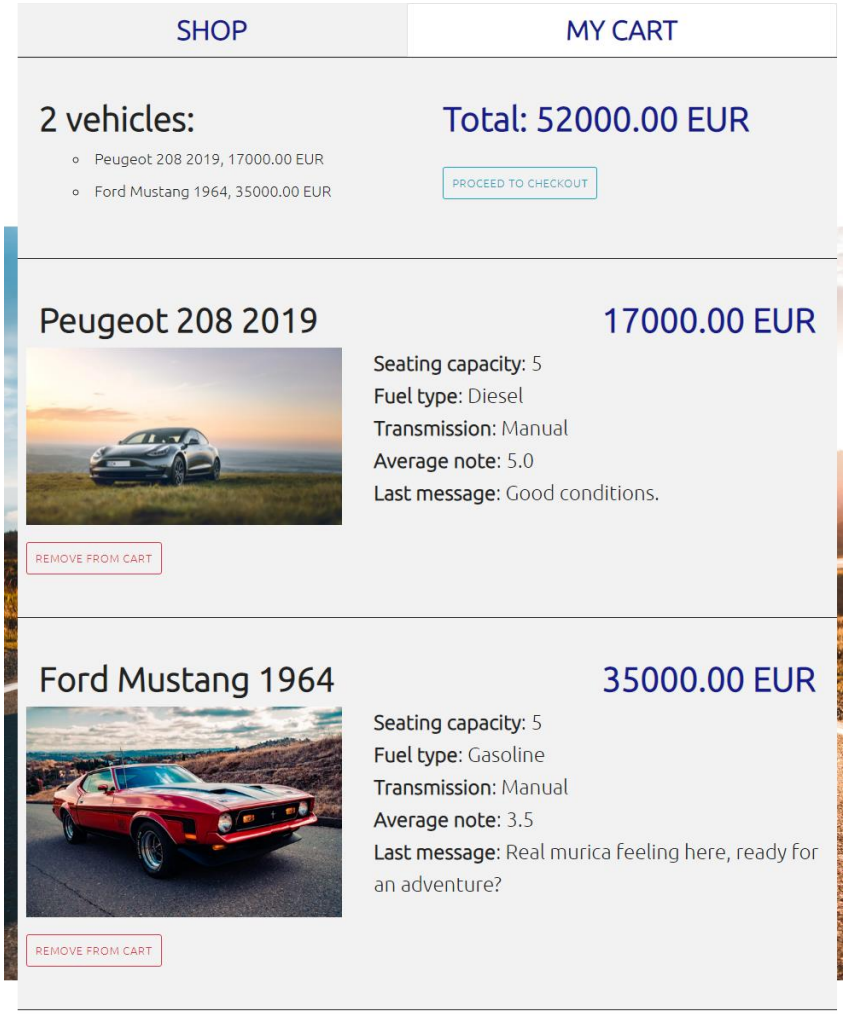
- Brand
- Model
- Year
- Price
- Seating capacity
- Fuel type
- Transmission type
- Average of renting notes
- Last message after being rented
- Availability

2.1.2 Select a car

If you want to buy a car, simply click on the “ADD TO CART” button.
The selected car will disappear from the SHOP section and will be added to your cart.

2.1.3 MY CART

The MY CART section displays the car(s) you have chosen to buy.



This section shows:

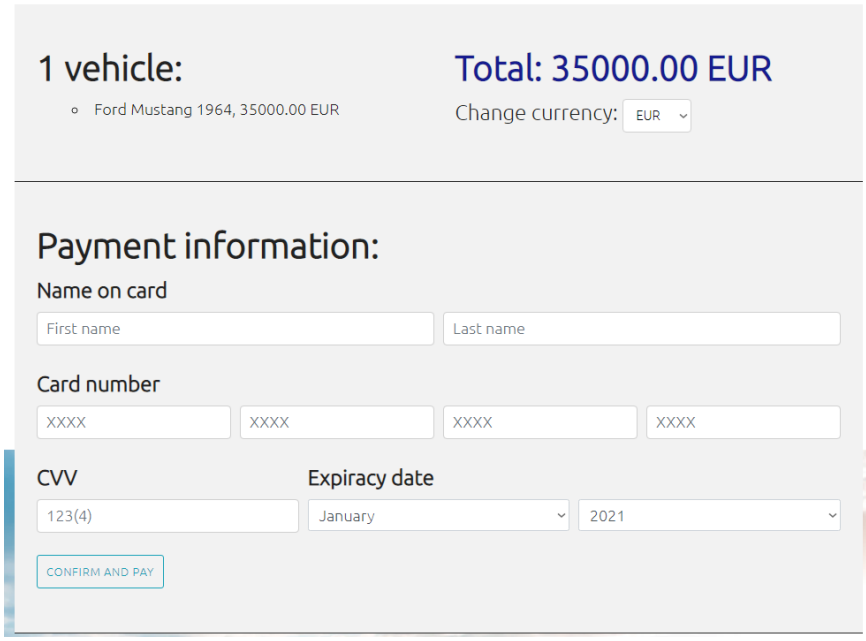
- How many vehicles you have chosen
- Their characteristics (brand, mode, year and price)
- Total price
- Car's information like in the SHOP section

You can at any time remove one car by simply clicking on the REMOVE FROM CART button.

Once you are ready, click on the PROCEED TO CHECKOUT button

2.2 Payment

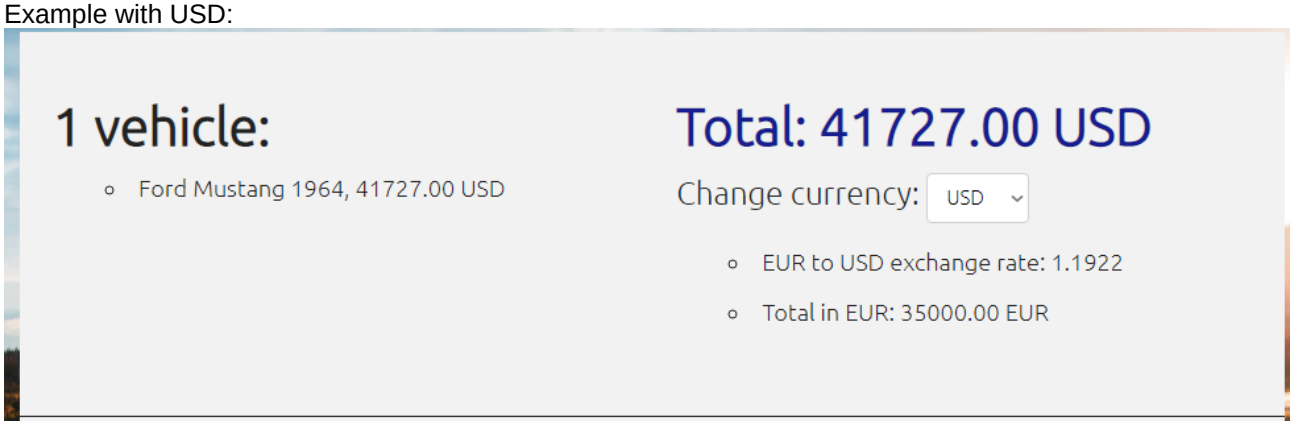
On the payment page, you will see different kinds of information.



2.2.1 Price currency

Vehicle prices are in euro but you can simulate how much the vehicle chosen will cost you in your prefered currency.
Use the dropdown list to select your prefered currency.

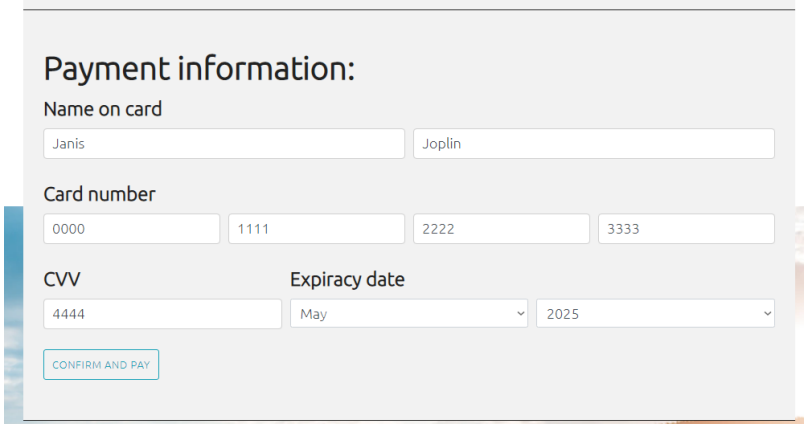
Example with USD:



We can see that the USD price of the vehicle is now displayed. Also, the change rate of 1 EUR to USD and the real price (the one in euro) are also shown.

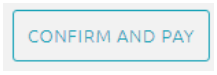
2.2.2 Bank informations

The second part of the Payment section is related to your bank information.



Please indicate your credit card information to proceed to the payment:

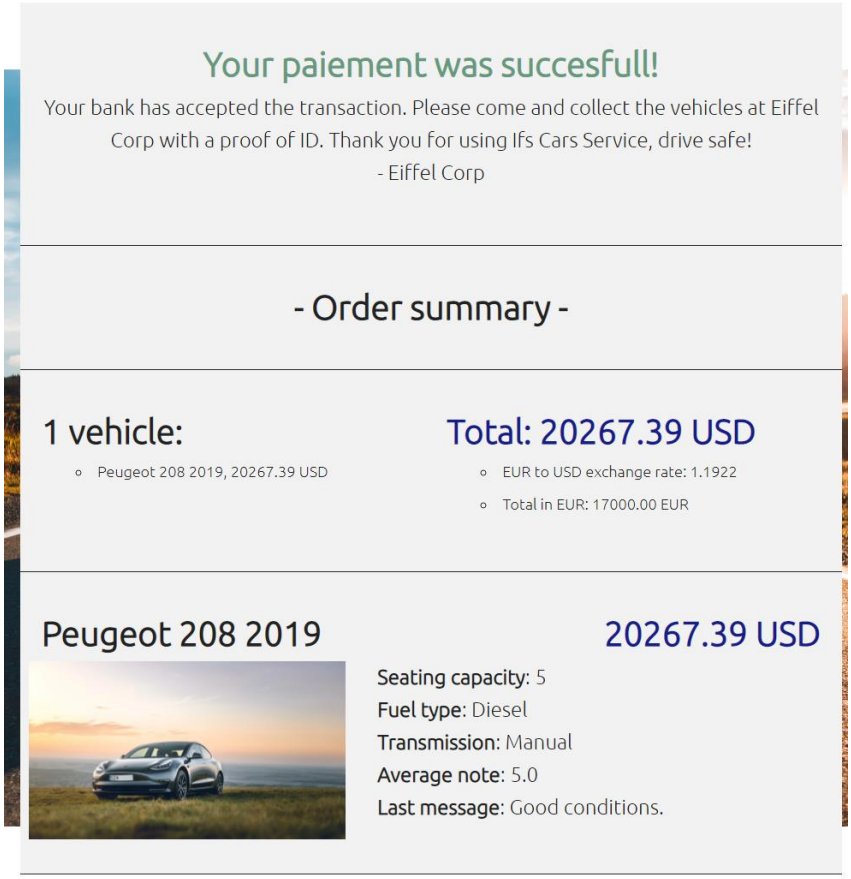
- First and Last name of card owner
- Card number
- CVV
- Expiration date

Then click on 

to validate the payment.

2.2.3 Payment successful

If no problem occur during the transaction, you should arrive to a page similar to this one:



Here you are given the order summary with the car information and car price in your bank currency and in euro.

To pick up the car, please come to Eiffel Corp with your proof of ID.

2.2.4 Payment error

Unfortunately, if you see this page, this means that the payment didn't occur. Make sure you have entered the right credit card information or verify that you have enough funds available for the car(s) chosen.