Faculty of Science and Technology

Computer Engineering Department

Engenharia de Serviços

Applying service design techniques to model a real-world service

2020/2021

Work done by the group:

António Maria 2017265346

Cátia Semedo 2020207998

Rui Reis 2013134606

Šimon Smrček 2020188489

Susana Meireles 2020207801

Table of content

Images

Introduction

The assignment was to implement a prototype for the service that was previously modeled in Part 1. This service gives the car companies the opportunity to continue with their services during the COVID-19 pandemic by providing delivery of an online-bought car directly to the designated address.

This process starts with a client browsing the Car Dealership website, from which he/she chooses the desired car (by choosing model and color), and as of last but not least places the order.

With users' car specifications, the Dealership contacts the Country Distributor, using an API to find out if the desired car is available, and after receives an answer with the confirmation or rejection of the fact.

In case the car's availability is confirmed, the Dealership reserves it and receives a confirmation with the vehicle identification number (VIN). Afterward, the Dealership confirms the client's order to him/her by an email including the VIN.

The Dealership, with all the given information (e.g. VIN), proceeds to request a license plate from the National Auto Registry, using a specific API again.

When the car is prepared for delivery, the Dealership sends an e-mail to the client with all the important information (the VIN, plate number, delivery date, and a confirmational QR code).

The logistics company will later deliver the car to the client's address and will ask to scan the QR code that the client received by email as proof of acceptance of the car.

The last part of the process is the transmission of the information to the Car Dealership so it can conclude and finish the sale.

Architecture

After analyzing the problem described before, we implemented an architecture from the beginning of the purchase to the delivery of the car. In figure 1, we can see the architecture of the project developed.

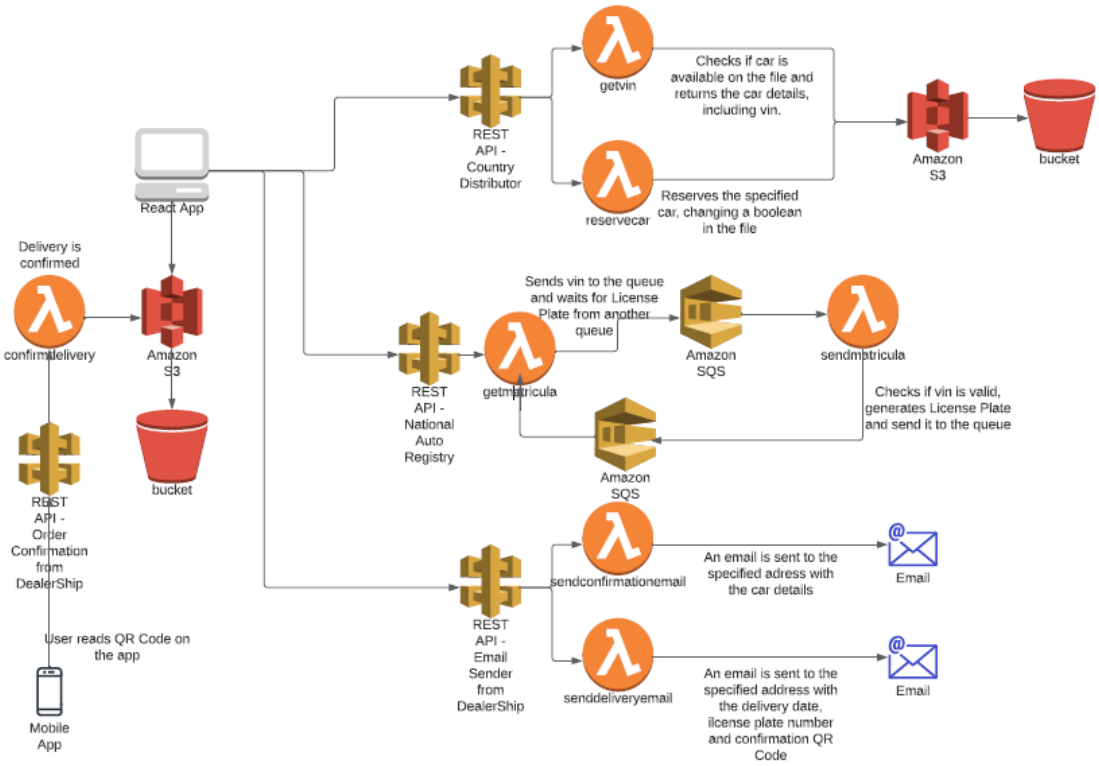


Imagem 1 - Architecture

We have a React Application that is hosted on AWS S3 which calls several APIs to do the operations.

The API from the CountryDistributor has two resources with one method each, one uses a lambda function to get all the cars of a given model and color, which the react application will use to select the first one available to reserve, and the other uses another lambda function to reserve the car and returns the VIN of the chosen vehicle. All this data is stored on AWS S3 in a file and is manipulated when a car is reserved so it doesn't appear anymore when other clients select the same model and color.

The API from the National Auto Registry uses a resource integrated with a lambda function that sends a message to an SQS queue, and this queue is also linked with a lambda function that validates the VIN number that he receives and returns a License Plate to another queue so It can be read by the first lambda which is also receiving, and

consequently deleting, messages from the second queue. In the end, the returned message is a license plate or an invalid VIN number warning.

We also used an API for sending emails, so it has more flexibility in where we could use it in the future development of this prototype, instead of calling it consecutively after the client makes his order as it is now. The API has two resources, one for each email we have to send. The first one confirms the order of the client, sending the car model, the color, and the VIN of the chosen car. The second lambda function sends an email with the delivery date, the VIN number, the license plate and generates a QR Code that is linked to another API that confirms the delivery with the Dealership.

The last API is called when the client scans the QR Code with our application, this QR activates the lambda function that returns a message saying that the delivery was confirmed.

WebApplication of the Dealership

We created a web application of the Dealership built using React, that allows the users to choose the car model and color they desire, and then place the order. They also have to fill in their names and addresses.

This application also contacts the user by email confirming the order. Later the user receives another email with the VIN of the car, plate number, and confirmational QR code, and this one is stored in AWS S3.

http://cardealership-group4.s3-website-us-east-1.amazonaws.com/

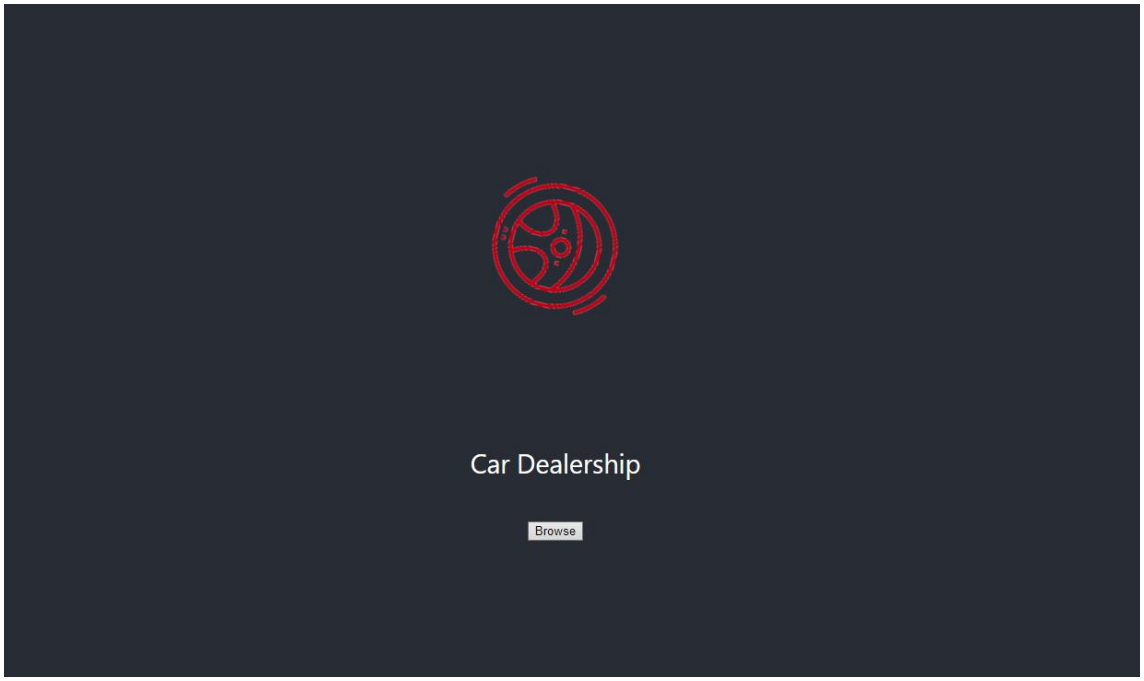


Imagem 2 Application of the Dealership

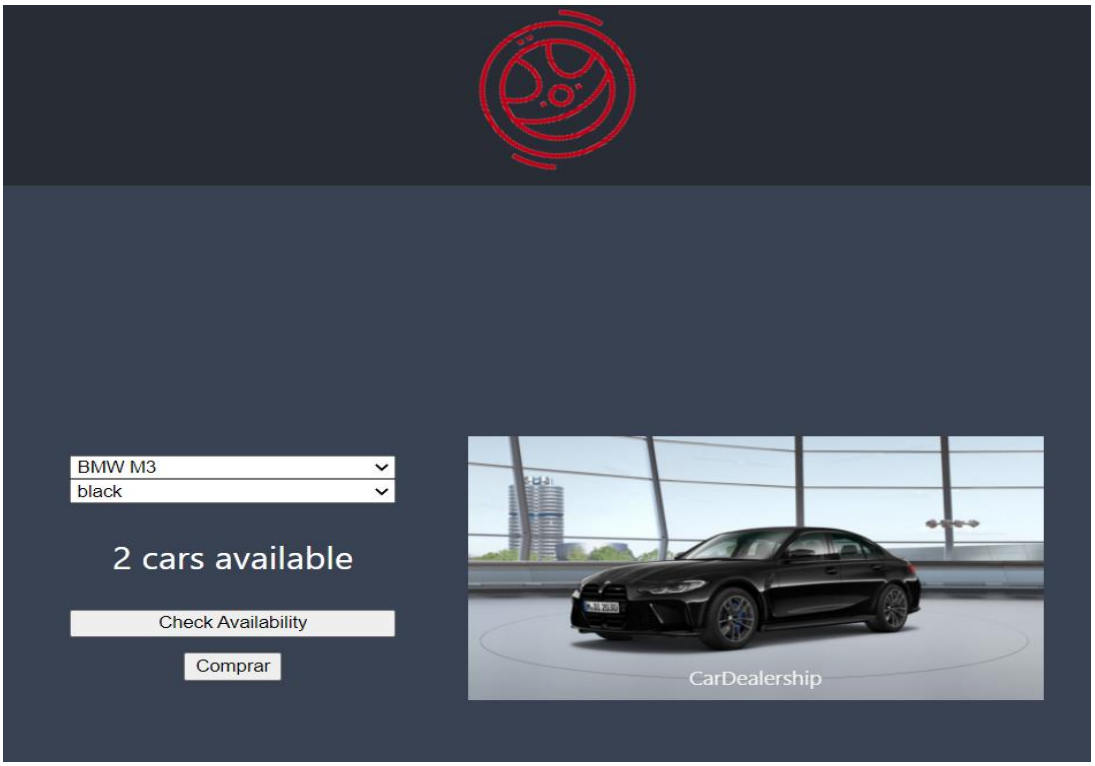


Imagem 3 - Cars available

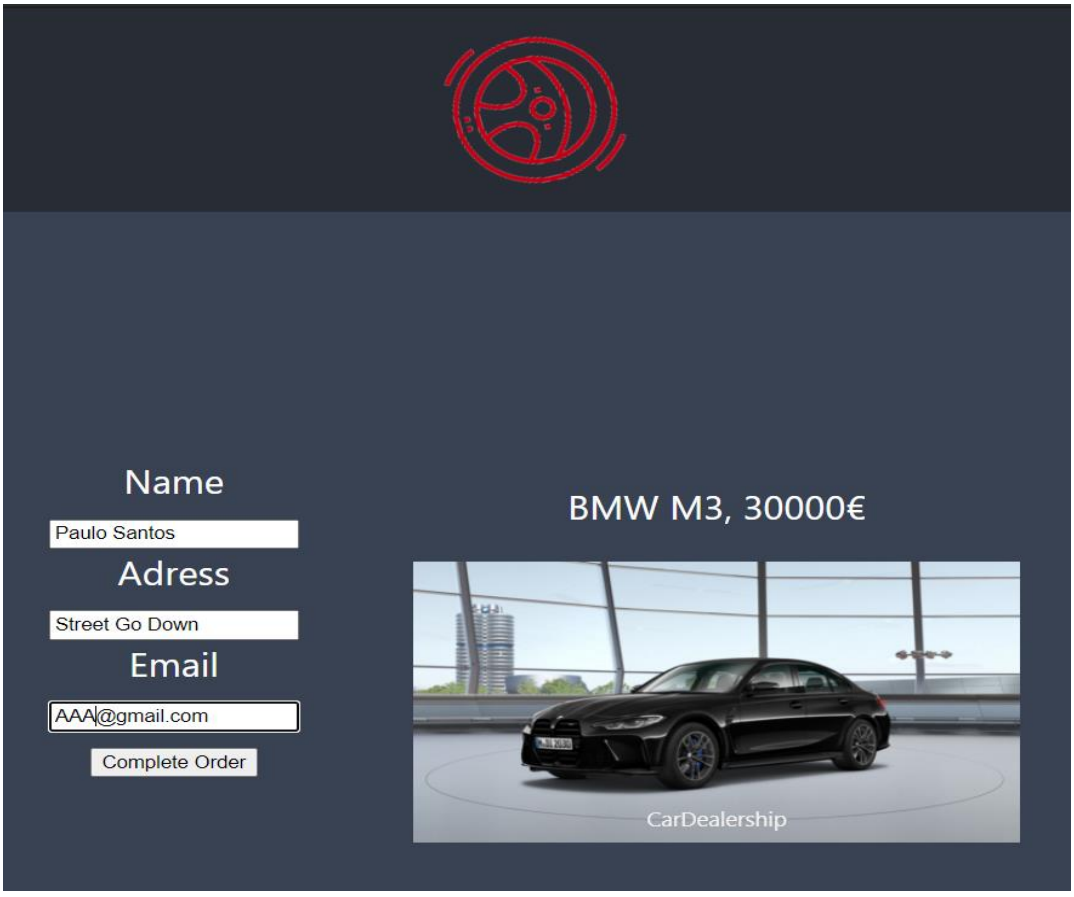


Imagem 4 - Complete Order

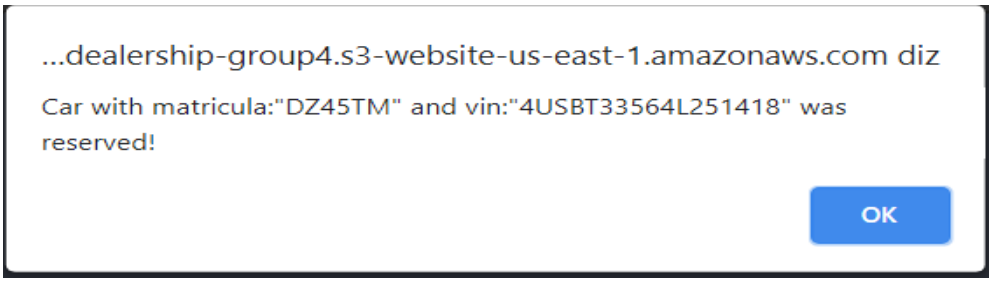


Imagem 5 - Car was reserved

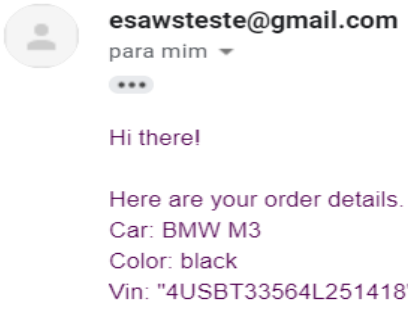


Imagem 6 - Order Details

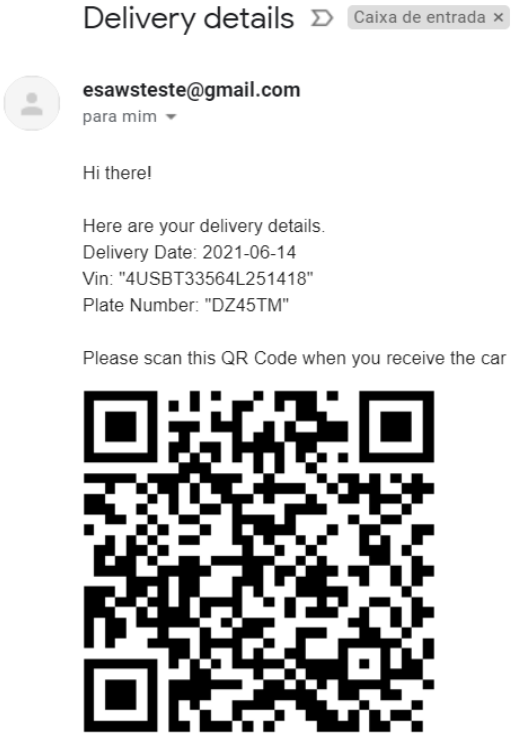


Imagem 7 - QR code

REST API of the Country Distributor

To allow the Country Distributor to get the list of the available cars (within the models and colors available) and also to allow him to reserve an available vehicle and receive a given VIN (vehicle identification number), we implemented the following API'S that were suggested:

CountryDistributor:

Initially, it was necessary to create a database where some car data were saved, containing the following attributes (ID, Model, Price, Colour, Reserved, and VIN) as shown in figure 8.

ID, Model, Price, Color, Reserved, VIN

```
1,BMW M3,30000,black,False,4USBT33564L251418
2,BMW M3,30000,black,False,WAUWGCFB0A0015984
3,Tesla Model S,60000,white,False,WBY1Z4C55FV202299
4,Mercedez Classe C,30000,black,False,4USBT33505LR69760
5,Volkwasgen Golf,20000,yellow,False,WBAAM5330YEJ40718
6,Nissan Micra,15000,white,False,WBXPC93418UJ05839
7,Volkwagen Polo,12000,red,False,WBALM1C56GE634647
8,Audi A5,35000,black,False,WBAWV13577PK49872
```

Imagem 8 - Database

Available vehicles:

This returns all cars available for a given model and color that aren't reserved (as we can see in figure 9).

https://kt4iskt5c9.execute-api.us-east-1.amazonaws.com/Test/nomes?modelo={param0}&cor={param1}

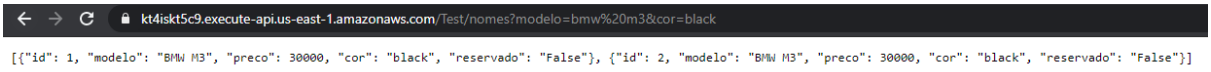


Imagem 9 - Details

The unavailability of a car is indicated in figure 10 as well.

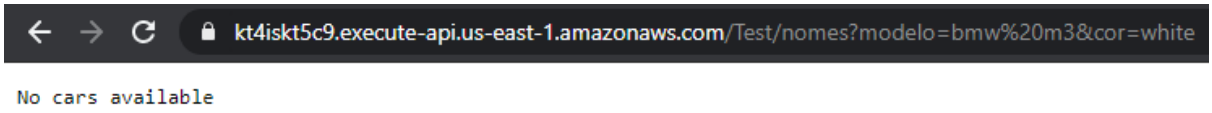


Imagem 10 - No cars available

Given VIN:

Gives VIN returns the VIN of the reserved car just using an id, as we can see in figure 11.

https://kt4iskt5c9.execute-api.us-east-1.amazonaws.com/Test/reservecar?id={param0}

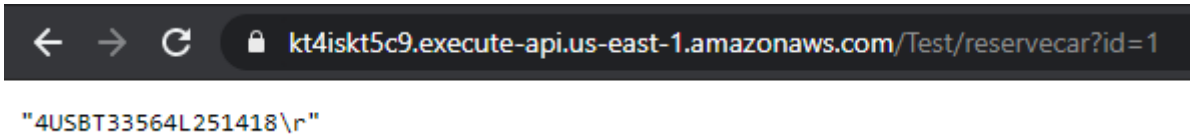


Imagem 11 - Vin

In case the car is already reserved or doesn't exist will be concluded on the page as well, as we can see in figure 12.

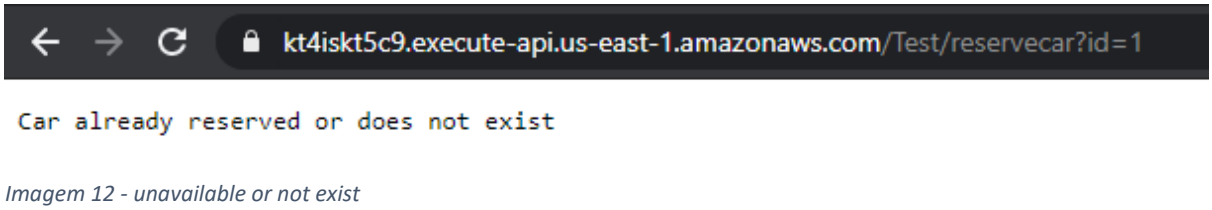


Imagem 12 - unavailable or not exist

REST API of the National Auto Register

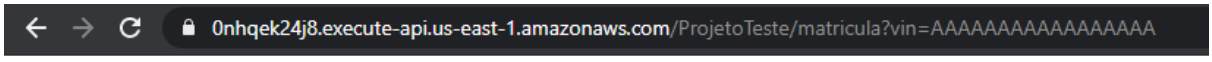
After the Country Distributor sends the VIN confirmation, it is necessary to request the license plate from the National Auto Registry with the validated VIN. The request received is processed using AWS SQS.

National Auto Registry:

License plate:

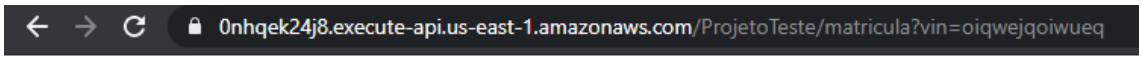
This returns the license plate of the reserved car if the VIN is proven valid, as is possible to see in figure 13 In the case the VIN is not valid, the user receives information reporting the invalidity, as visible can see in figure 14.

https://0nhqek24j8.execute-api.us-east-1.amazonaws.com/ProjetoTeste/matricula?vin={param0}



"NH22CL"

Imagem 13 - Lincense Plate



"Vin not valid!"

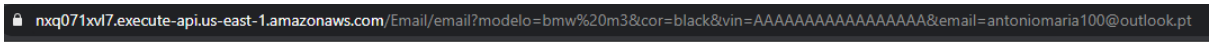
Imagem 14 - Vin not valid

EmailSender:

When the car availability is confirmed, the dealership sends an email that includes the model, color, and the VIN of the car, as visible in figure 15.

Order Confirmation:

https://nxq071xvl7.execute-api.us-east-1.amazonaws.com/Email/email?modelo={param0}&cor={param1}&vin={param2}&email={param3}



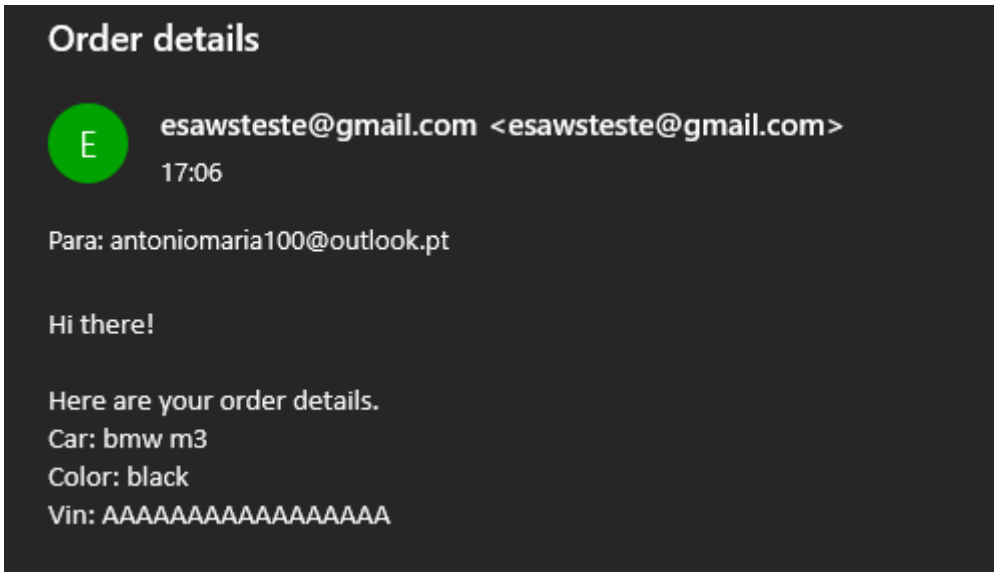
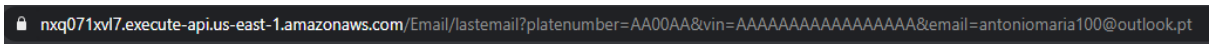


Imagem 15 - Order Confirmation

Delivery Details:

When the car is ready to be delivered, the dealership sends an e-mail to the client with the VIN, plate number, delivery date, and a confirmation QR code, as possible to see in figure 16.

https://nxq071xvl7.execute-api.us-east-1.amazonaws.com/Email/lastemail?platenumber={param0}&vin={param1}&email={param2}



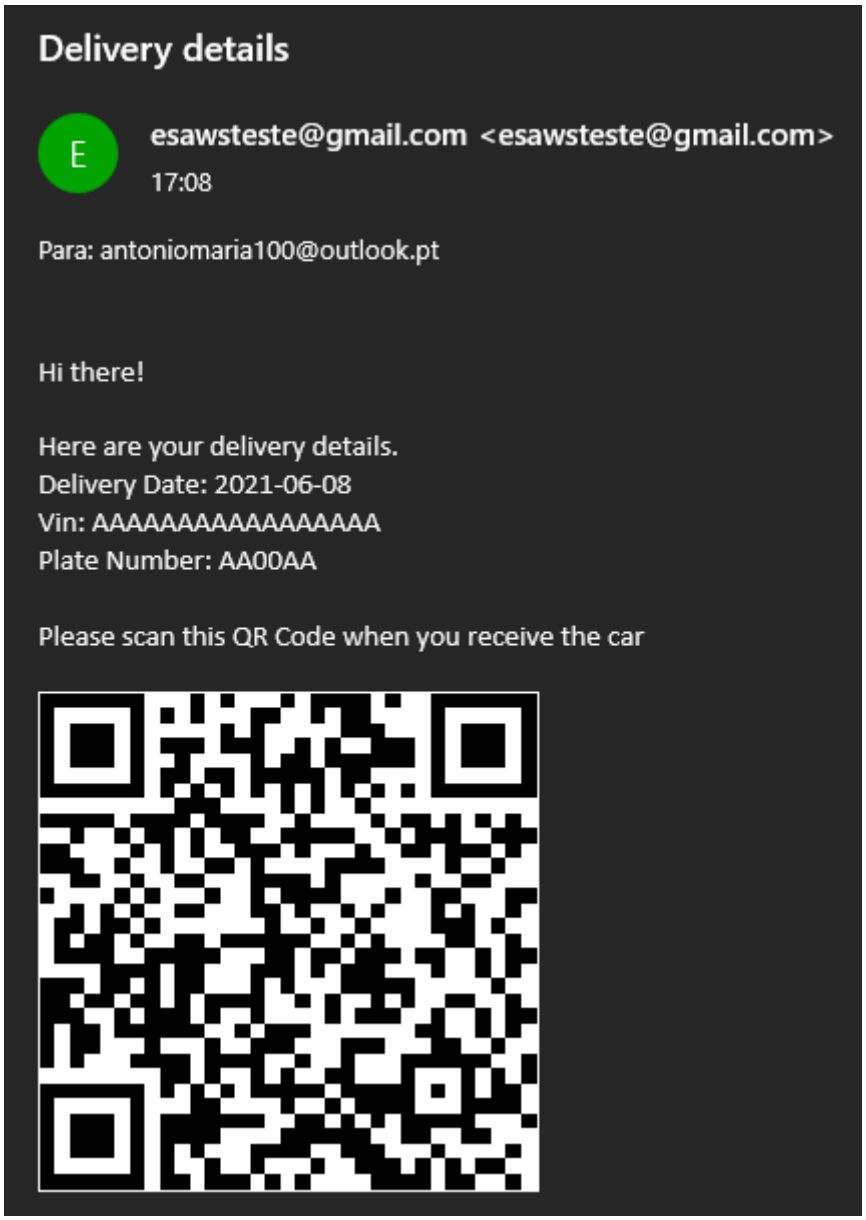


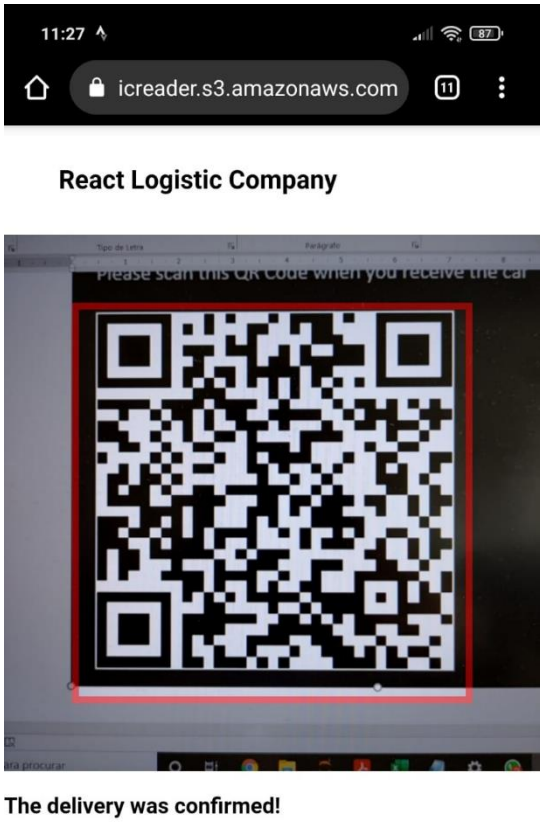
Imagem 16 - QR Code

Web application of the Logistics Company

The Logistics Company web application allows the user to scan the QR Code and contact the Dealership to confirm the car delivery.

By allowing access to the camera, the user is given the possibility to validate the QR Code that was sent to him/her by email, as possible to see in figure 17.

https://logisticreader.s3.amazonaws.com/build/index.html



Conclusion

With the conclusion of this work, we fulfilled everything that was requested in the assignment, putting into practice all the material taught throughout the semester in class.

In this way, we arrive at a final prototype of the application that allows the user to choose the car they want and purchase it, after which they receive a purchase confirmation email with all the necessary data so that they can later receive the car showing only a QR Code.