**CCT College Dublin**

Camila Cerqueira dos Santos

2019060

**Guided Technology Project**

Ger’s Garage

Dublin - Ireland

2020

# Abstract

This Guided Technology Project aims to develop a web system for the management of a garage. This document presents the analysis, the design and the development of this system. In order to put into practice everything that was learned during the Higher Diploma in Science in Computing course at CCT College Dublin and also as a way of learning new technologies, an application was developed using the PHP programming language, MySQL database as a basis and other tools and technologies necessary for the development of the system. The system allows the customer to search and book services online also to consulting the order online through a login and password. It allows the manager to control the services provided, control their activities and the activities of staffs, among others.

Keywords: Guided Technology Project. Garage. Web System. PHP. MySQL.

# Acknowledgements

First of all, I thank God for always giving me the strength to overcome difficult moments. My family and friends for always encouraging me to pursue my goals. To all staff, lectures and colleagues at CCT College Dublin for all learning, patience and friendship.

List of Figures

Figure 1 – Proposed Timeline 10

Figure 2 – Use Case Diagram 13

Figure 3 – Entity-Relationship Diagram 15

Figure 4 – Class Diagram 16

Figure 5 – Wirefrme – Main Page 17

Figure 6 – Wireframe – Admin 17

Figure 7 – Wireframe – Customer 18

Figure 8 – Test Cases 35

List of Tables

Table 1 – Functional Requirements 12

Table 2 – Non-Functional Requirements 12

Table 3 – Use Case Diagram Description 13

Table 4 – Data Design 29

Table of Contents

[Abstract 2](#_Toc34694295)

[Acknowledgements 3](#_Toc34694296)

[1: Introduction 8](#_Toc34694297)

[1.1: Purpose 8](#_Toc34694298)

[1.2: Scope of Project 8](#_Toc34694299)

[1.3: Required knowledge 9](#_Toc34694300)

[1.4: Initial proposed plan 9](#_Toc34694301)

[1.5: ‘Why’ this is a good project? 10](#_Toc34694302)

[1.6: Novel aspects 10](#_Toc34694303)

[1.7: Acronyms and abbreviations 10](#_Toc34694304)

[2: Literature Review 11](#_Toc34694305)

[3: System Analysis and Design 11](#_Toc34694306)

[3.1: Detailed description of the functionality of the proposed system 11](#_Toc34694307)

[3.1.1: Functional Requirements 11](#_Toc34694308)

[3.1.2: Use Case Diagram 12](#_Toc34694309)

[3.1.3: Entity-Relationship Diagram 14](#_Toc34694310)

[3.1.4: Class Diagram 15](#_Toc34694311)

[3.2: Wireframes 16](#_Toc34694312)

[3.3: User Interface Design 18](#_Toc34694313)

[3.3.1: Main Page: 18](#_Toc34694314)

[3.3.2: Register 20](#_Toc34694315)

[3.3.3: Log In 21](#_Toc34694316)

[3.3.4: Logging in as a Customer 21](#_Toc34694317)

[3.3.5: Logging in as an Admin 25](#_Toc34694318)

[3.4: Data Design 29](#_Toc34694319)

[4: Implementation of the system 30](#_Toc34694320)

[4.1: Technologies used 31](#_Toc34694321)

[4.2: Implementation of the system 33](#_Toc34694322)

[4.3: Problems encountered 34](#_Toc34694323)

[5: Testing and Evaluation 34](#_Toc34694324)

[6: Conclusions 35](#_Toc34694325)

[Appendix A: Code Listings 36](#_Toc34694326)

[List of References 36](#_Toc34694327)

# 1: Introduction

## 1.1: Purpose

The main purpose of this document is to present a detailed description of the website Ger’s Garage and its functionalities. It explains all requirements and provides all the necessary information for system design, software implementation, testing and system approval.

## 1.2: Scope of Project

Increasingly, companies are seeking to replace the processes carried out on a daily basis manually with automated systems. The use of these systems aims to reduce costs and considerably streamline these processes, i.e. in such a way that as little effort as possible is required from people.

Web-type applications are examples of this, as more and more tasks, from sending messages to paying bills, can be performed online, using only one computer with Internet access. What used to be not very accepted by the population because it was considered dangerous, nowadays has become frequent in most people’s lives, who use these applications for common day-to-day tasks, such as buying products, scheduling services, among others.

According to this principle, a new website was created for Ger’s Garage in order to automate the garage services as well as advertise the garage and its services.

Based on the specifications provided, the website created allows customers to register on the site and book a service for their vehicles which can be Annual Service, Major Service, Repair / Fault or Major Repair among others.

Customers are able to select the date they want for the booking but the system has some particularities: The website limits the number of bookings allowed per day. If there is no available space for a day, then the customer is not able to book a service on that day and have to select another day. Also Ger’s garage is closed on Sundays for service bookings, therefore the system not allows customers to select a Sunday.

Once registered, customers are able to login on future occasions and the system remembers their details such as details of the vehicle they last booked in to the garage. Also the system allows the customer to track the status of the requested service online, just by logging in with the registered user and password.

When scheduling a service or repair the customer must provide some personal data and also vehicle details such as: Customer name and contact details, Vehicle type and make, Vehicle licence details, Vehicle engine type (diesel / petrol / hybrid / electric) Booking Required (i.e. Annual Service, Major Service, Repair / Fault or Major Repair) Customer Comments (this field allows customer add any notes they want to, such as a description of the problem).

The manager has an admin access that allows him to view schedules for any day or week allowing him to plan his work schedule and his staffs’ rostering. With an admin access, the manager is also able to allocate a mechanic to each vehicle for any type of service; however each mechanic can carry out at most 4 services/repairs in one day. If the booking is a Major Repair then this counts as a double service. Also the manager is able to print the schedule for any particular date.

The manager can specify costs for the bookings as well as add extra costs for any item that needs to be repaired or replaced. For example, if a tyre needs to be replaced then the cost of the tyre is added and the cost of carrying out a “wheel balancing” is added. The system provides a list of at least 40 different parts/items for the garage.

The customer must provide details of the type and make of car (e.g. Car – Ford Feista). The system allows appointments to motorbikes, cars, small vans and small buses also it allows the user to choose “other” in case their vehicle is not in the list.

The manager is able to set each booking to one of 5 possible statuses:

1) Booked – this is the default status when a booking is made.

2) In Service – when the vehicle arrives at the garage.

3) Fixed / Completed – when the vehicle is ready for collection.

4) Collected – when the customer has taken the vehicle away and paid their bill.

5) Unrepairable / Scrapped – when the fault cannot be fixed; in this case the car has either been taken away by the customer or has been sent for scrap.

The manager is able to print an “invoice” or bill for each customer when the service/repair is complete. The system provides an itemised bill for the customer.

## 1.3: Required knowledge

To build this website the project will require a developer that have some knowledge of HTML that is used to structure content and codes to define how website elements are displayed on the screen. PHP among other functions, is also used to perform communication with the server. With it, it was possible to interact with the database and other applications, implementing dynamic and complex functions on the website. MySQL that is a relational database, especially used for websites built on PHP. CSS (Cascading Style Sheets) is used for website design and style. Bootstrap responsive to adapt the site to any type of screen. JavaScript is used to add interactivity and other dynamic features to the webpage.

## 1.4: Initial proposed plan

The initial proposed plan for this project is to computerize the entire garage service by creating a website that allows customers to book online their vehicles for a particular service. It also allows the owner to manage their garage and plan their work and their staff’s work. And thus advertising your business and services. Thereby optimizing the service and also advertising their business.

That is my initial proposed timeline for this project:

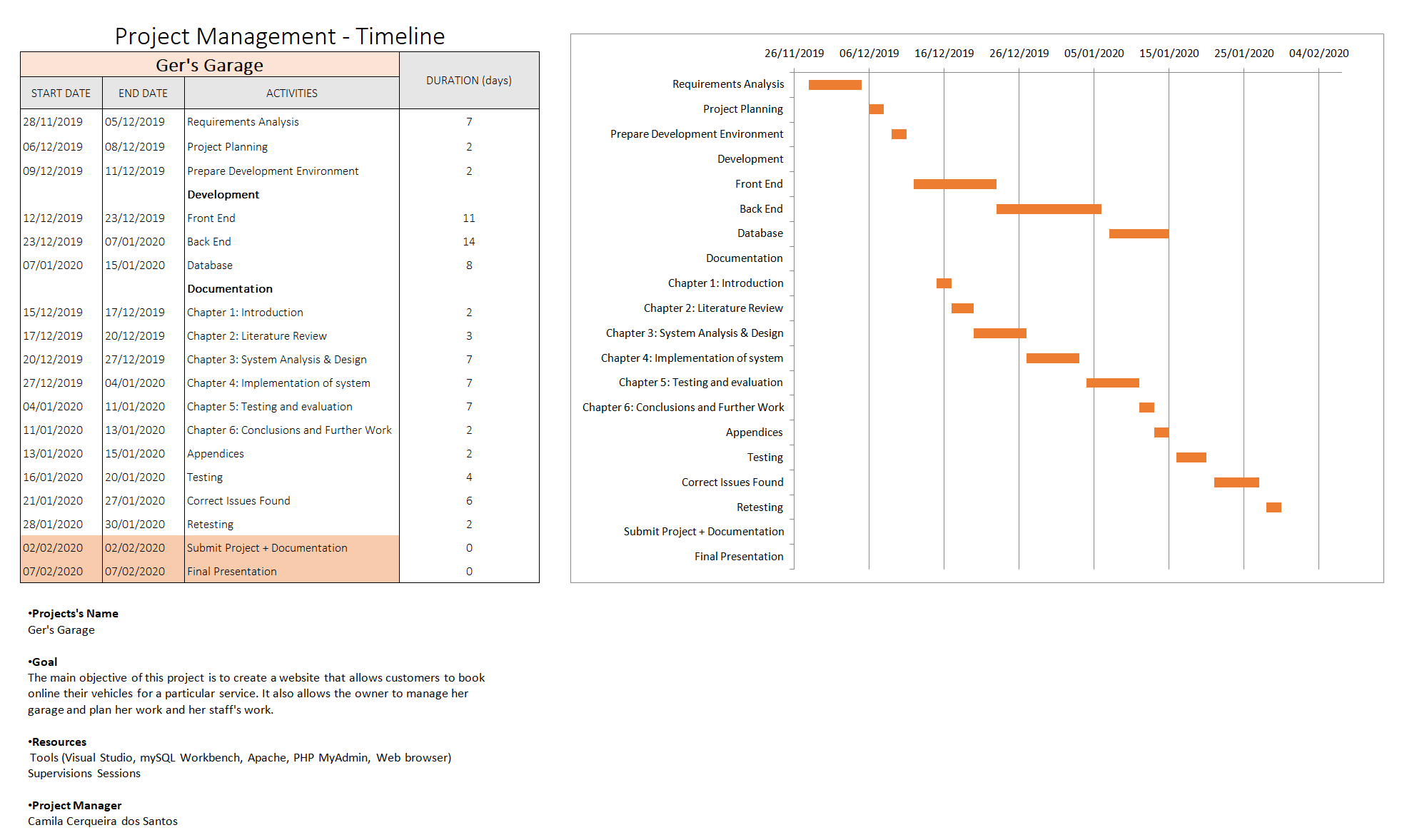


Figure 1

## 1.5: ‘Why’ this is a good project?

In choosing which project to work on, I considered the tools and technologies I felt most comfortable with. I opted for the web development project because besides being interested in it I will have the chance to put into practice all the contents learned during the first and second semester of the course and have the chance to study and put into practice new technologies that may not have been addressed during the course.

## 1.6: Novel aspects

Working on this project, besides being able to put into practice all the contents learned during the course and also improving my skills I will be able to understand how projects in a real world are and how to work on them. Surely working on this project will add value to my professional career and will help me to improve my skills and also to acquire new knowledge.

## 1.7: Acronyms and abbreviations

HTML: Hypertext Markup Language

PHP: Hypertext Preprocessor

PDO: PHP Data Objects

CSS: Cascading Style Sheets

FR: Functional Requirement

NFR: Non-Functional Requirement

AJAX: Asynchronous Javascript And XML

XML: Extensible Markup Language

HTTP: Hypertext Transfer Protocol

URL: Uniform Resource Locator

SQL: Structured Query Language

JSON: JavaScript Object Notation

API: Application Programming Interface

DOM: Document Object Model

PK: Primary Key

FK: Foreign Key

# 2: Literature Review

For the development of this project I used a lot of information collected on the website w3schools.com also watching videos on youtube.com, which was very important in the choice of which technologies to use and in the decision to choose one technology instead of another. All the technologies used as well as the description of each one are in chapter 4 of this document. All visited sites are listed in the Reference List at the end of this document.

# 3: System Analysis and Design

This chapter presents the result of this work, which is the implementation of a system for the management of a garage. The description of the system with its functionalities, requirements gathering, use case diagram, class diagram and the implementation of the system with the results obtained are presented.

## 3.1: Detailed description of the functionality of the proposed system

The main purpose of this project is to develop a website in order to computerize the management of a garage where the customer can easily book a service for their vehicles and also to optimize the manager’s work and their staffs.

The system provides control of customers, vehicle parts, vehicles repaired and management of services provided. It also has access control to the system, in which only a previously registered user who has a password can access the system.

### 3.1.1: Functional Requirements

For the development of this system, the necessary requirements were analyzed and raised in order to identify the functionalities that the system must have to meet the needs of users and what are the restrictions that exist on these functionalities.

The requirements specification is separated into two types of requirements which are the Functional Requirements (FR) and Non-Functional Requirements (NFR). Functional Requirements define the functionalities of the system, that is, what the system does, while Non-Functional Requirements define the quality characteristics that the system must have and which are related to its functionalities, that is, the technical requirements of an environment for example aspects of system security, performance, failure prevention etc.

Table 1 lists the Functional Requirements of the system and Table 2 lists Non-Functional Requirements.

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| FR1 | The system should allow the customer to register on the website. |
| FR2 | The system should allow the customer to log in to the website. |
| FR3 | The system should allow the customer to book a service or repair. |
| FR4 | The system should allow booking for motorbikes, cars, small vans and small buses. |
| FR5 | The system should not allow bookings on Sundays. |
| FR6 | The system should allow the customer to enter personal and vehicle data. |
| FR7 | The system should allow the customer to check the status of a service. |
| FR8 | The system should remember the details of a previously registered customer. |
| FR9 | The system should allow the manager to have an admin access. |
| FR10 | The system should allow the manager to log in to the website. |
| FR11 | The system should allow the manager to register new employees on the website. |
| FR12 | The system should allow the manager to register services on the website. |
| FR13 | The system should allow the manager to view booked services. |
| FR14 | The system should allow the manager to allocate staff to a specific service. |
| FR15 | The system should allow the manager to view the schedule for a specific date. |
| FR16 | The system should allow the manager to print an invoice or bill. |

Table 1

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| NFR1 | The system must be run in any web browser. |

Table 2

### 3.1.2: Use Case Diagram

The figure below shows the Use Case Diagram of the proposed system, based on the requirements raised previously.

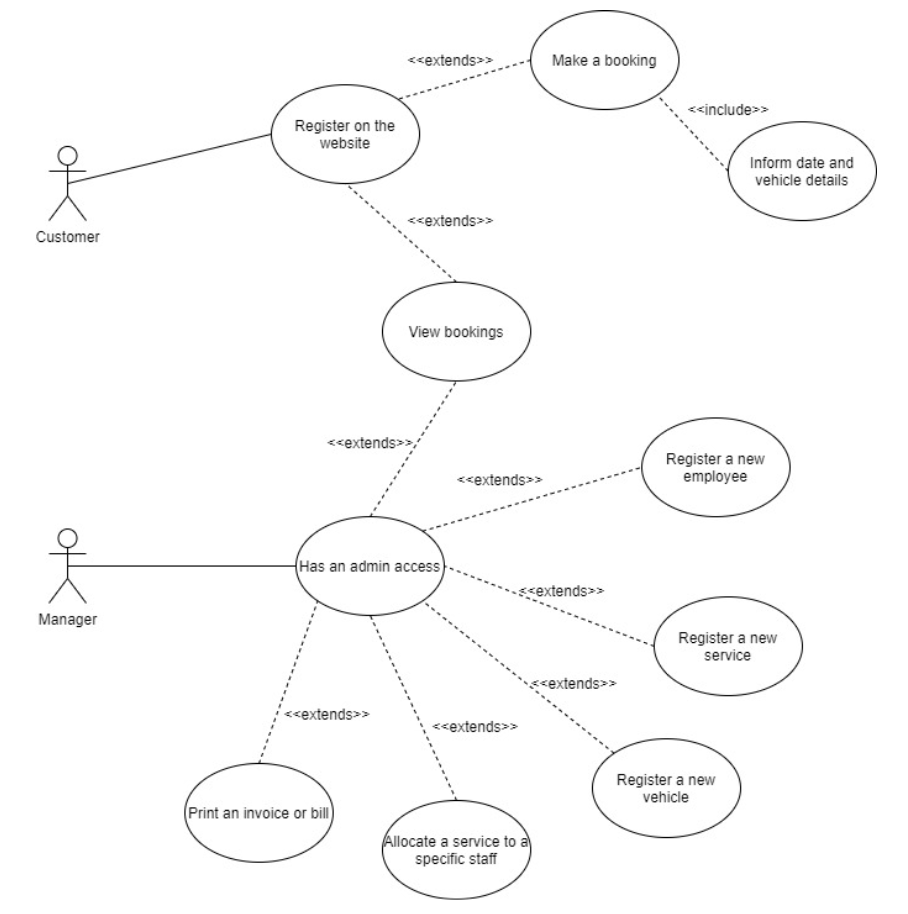


Figure 2

Table 3 presents the description of the use cases presented in the diagram of Figure 1.

|  |  |
| --- | --- |
| Use Case | Register on the website |
| Actor | Customer |
| Description | The system allows the customer to register and log in to the website. |
| Use Case | Make a booking |
| Actor | Customer |
| Description | The system allows the customer to book a service or repair. |
| Use Case | Inform date and vehicle details |
| Actor | Customer |
| Description | The system allows the customer to inform a date and vehicle details. |
| Use Case | View bookings |
| Actor | Customer |
| Description | The system allows the customer to view and check the status of previous bookings. |
| Use Case | Has an admin access |
| Actor | Manager |
| Description | The system allows the manager to log in to the website as an admin. |
| Use Case | View bookings |
| Actor | Manager |
| Description | The system allows the manager to view client’s bookings. |
| Use Case | Register a new employee |
| Actor | Manager |
| Description | The system allows the manager to include, change and delete an employee on the website. |
| Use Case | Register a new service |
| Actor | Manager |
| Description | The system allows the manager to include, change and delete a service on the website. |
| Use Case | Register a new vehicle |
| Actor | Manager |
| Description | The system allows the manager to include, change and delete a vehicle on the website. |
| Use Case | Allocate a service to a specific staff |
| Actor | Manager |
| Description | The system allows the manager to assign a task to a specific staff. |
| Use Case | Print an invoice or bill |
| Actor | Manager |
| Description | The system allows the manager to print an invoice or bill. |

Table 3

### 3.1.3: Entity-Relationship Diagram

Figure 3 presents the Entity Relationship Diagram of the developed system, in which it is possible to view the distribution of attributes in the database entities with their relationships between the tables. The relationship entity model is the beginning to develop and design the system database. It is the description of the structure of a database.

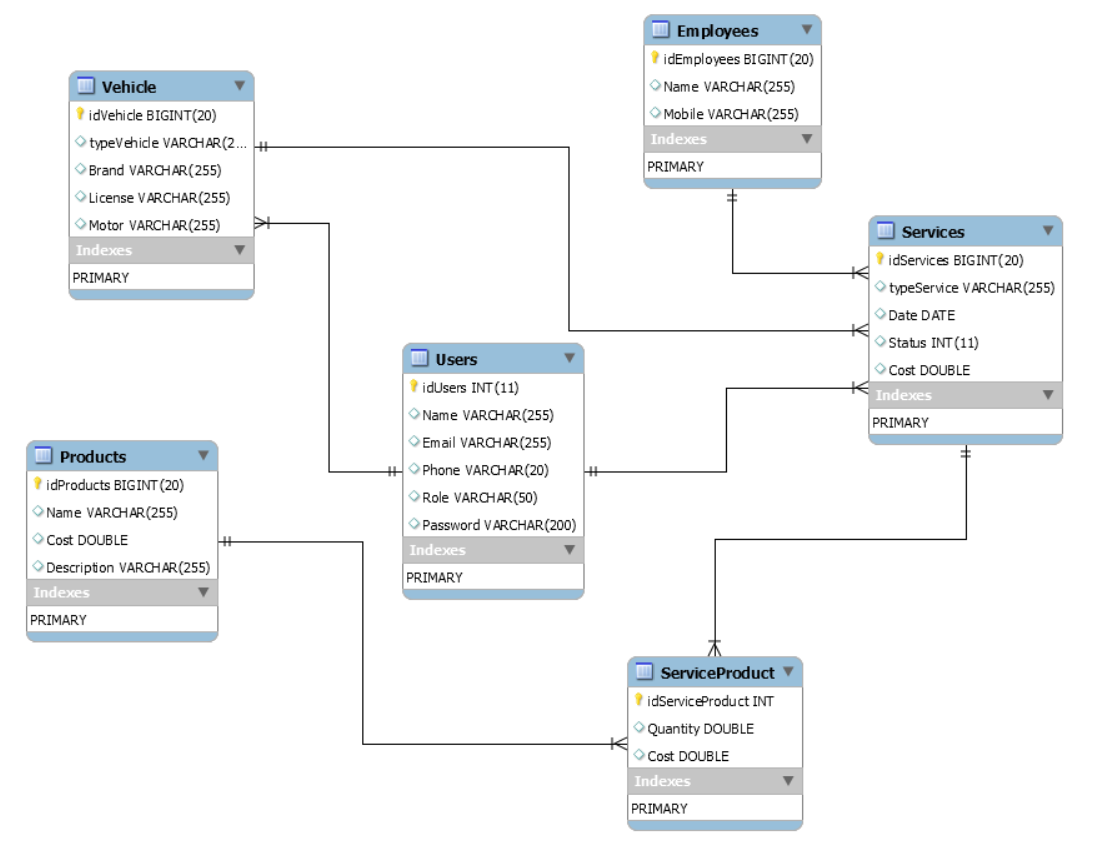


Figure 3

### 3.1.4: Class Diagram

The class diagram supports most other diagrams. It defines the class structure of the system and establishes how the classes relate.

Figure 4 presents the classes, with the attributes that are necessary in the project, providing an overview of how the system should work.

## 

Figure 4

## 3.2: Wireframes

In this session is presented the initial prototypes of the main screens in the system. For this work I used the tool InVision App, a web tool that allows you to create system prototypes quickly and with a pleasant and easy to use interface.

Main page:

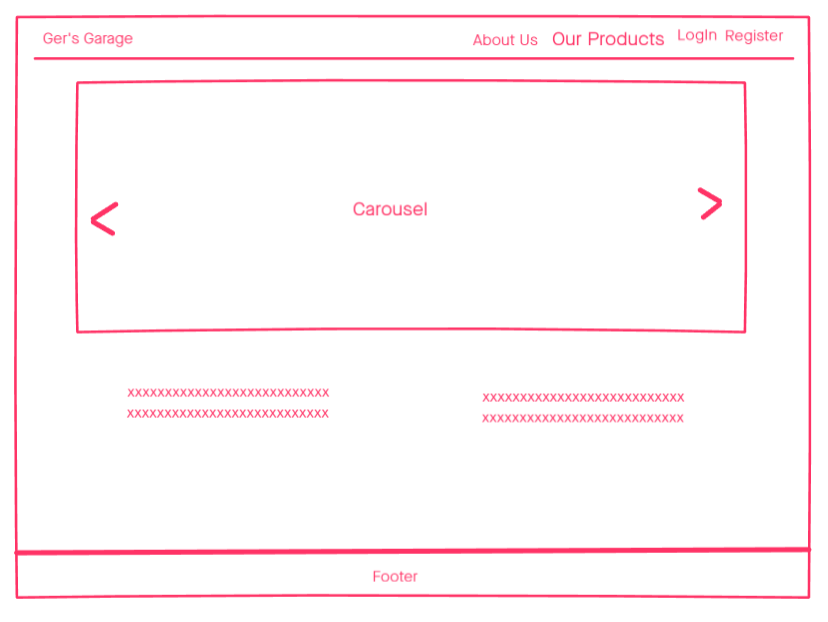


Figure 5

Logging as an Admin

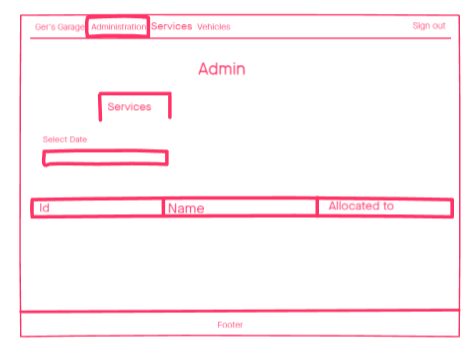
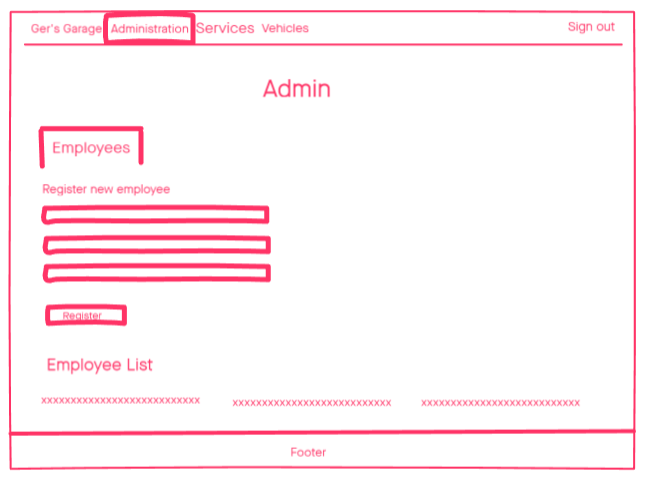


Figure 6

Logging in as a Customer

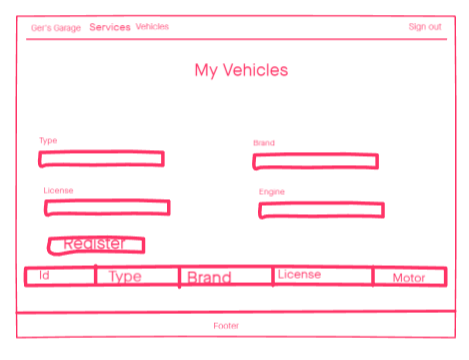




Figure 7

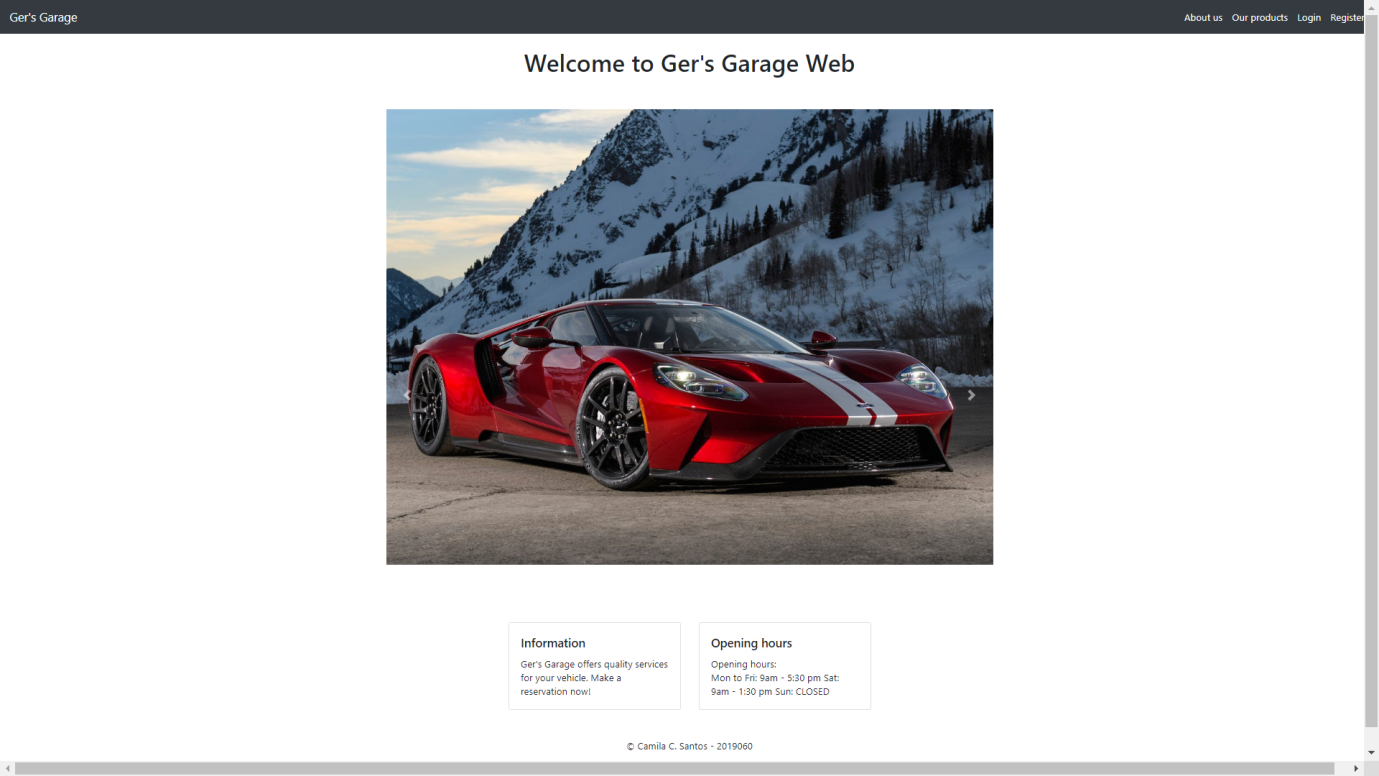
## 3.3: User Interface Design

In this session is presented the way of using the system through its screens, that is, user interaction. This description is made through the main functionalities of the system and is presented below.

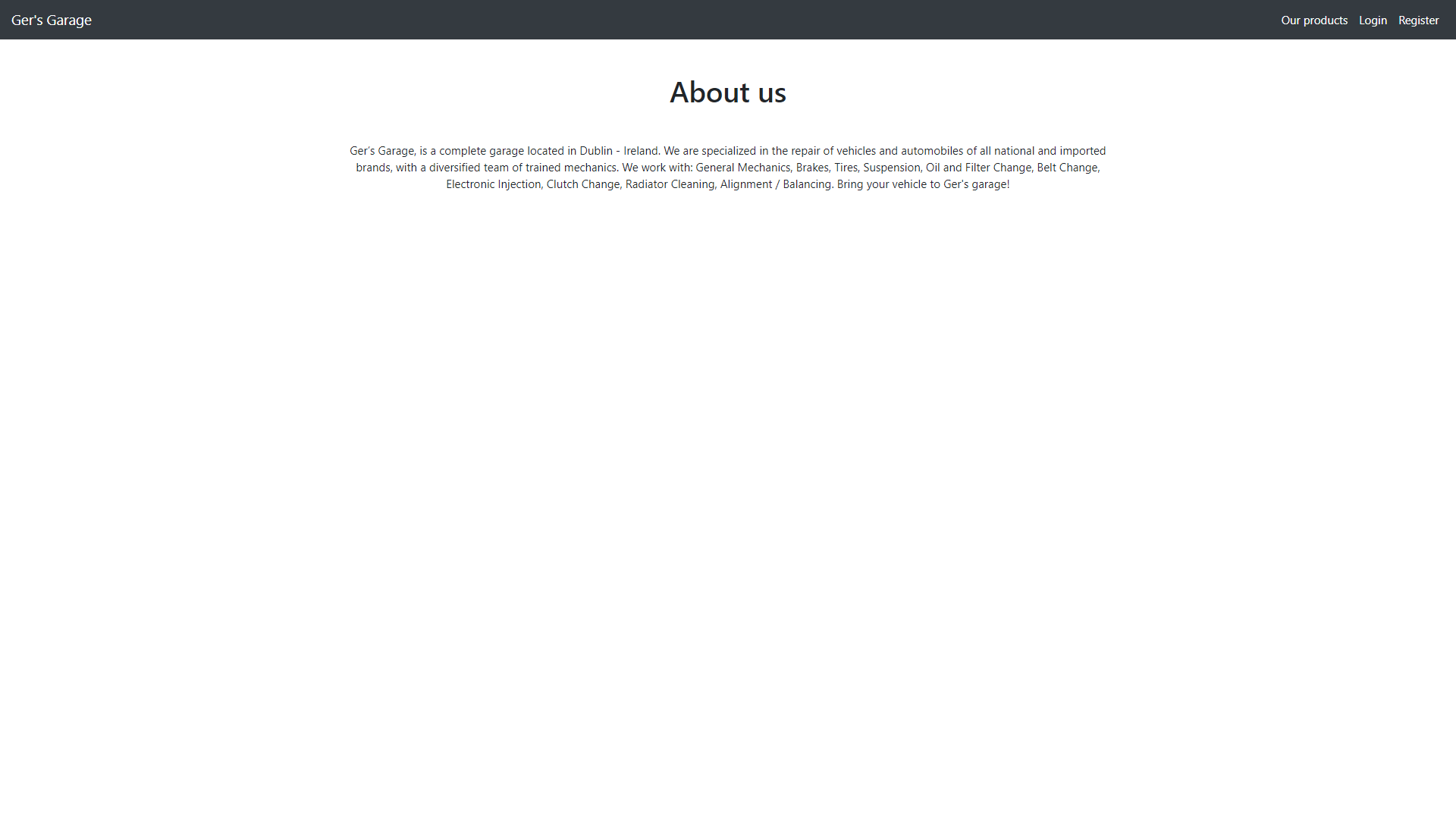
### 3.3.1: Main Page:

On the main page the user will be able to access the page “About Us” where there is a brief description of the company and “Our Products” where there is a list of the products offered by the company.

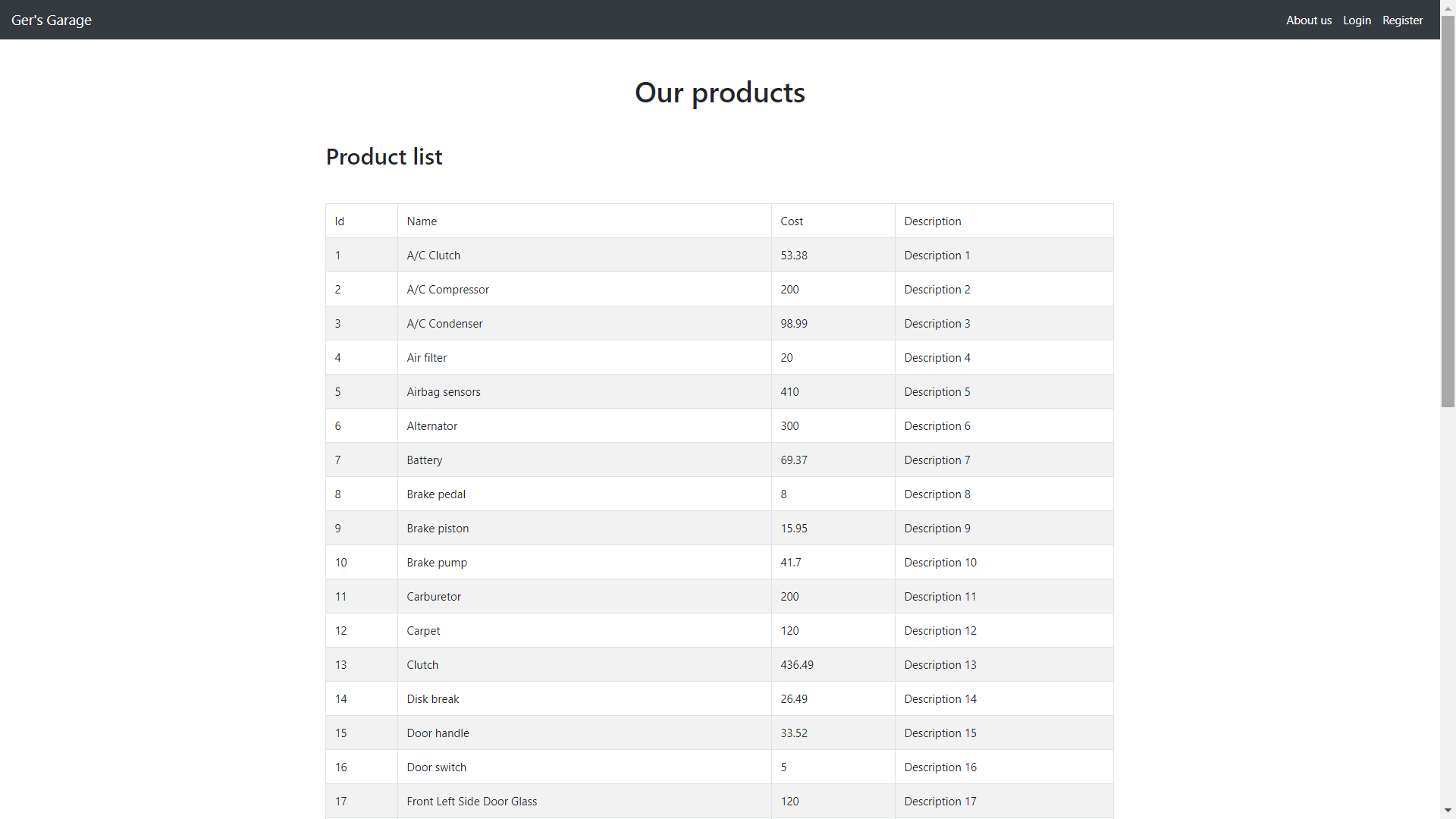
It is also on the main page that the user can register and login to the system.



About Us:

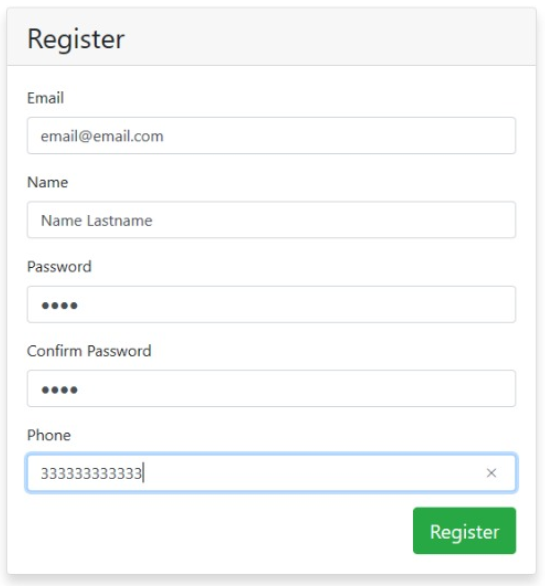


Our Products:



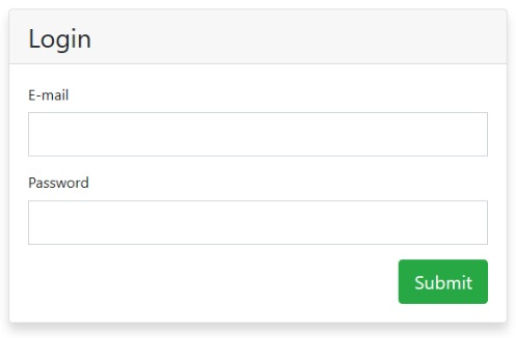
### 3.3.2: Register

To be able to log into the system the user must first register. For this, the user must inform a valid email, name, password and phone.



### 3.3.3: Log In

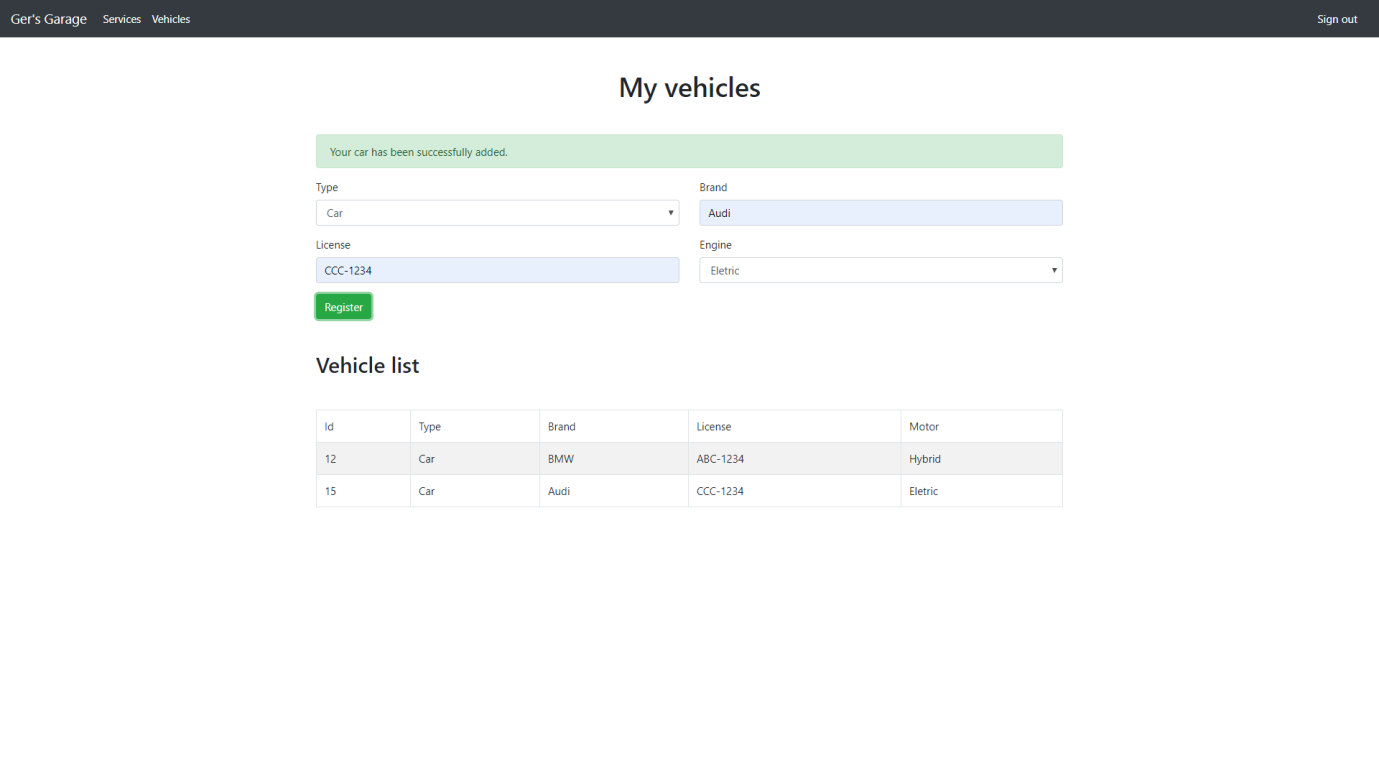
Once registered, to access the functionalities of the system, the user must log in with a registered email and password.



### 3.3.4: Logging in as a Customer

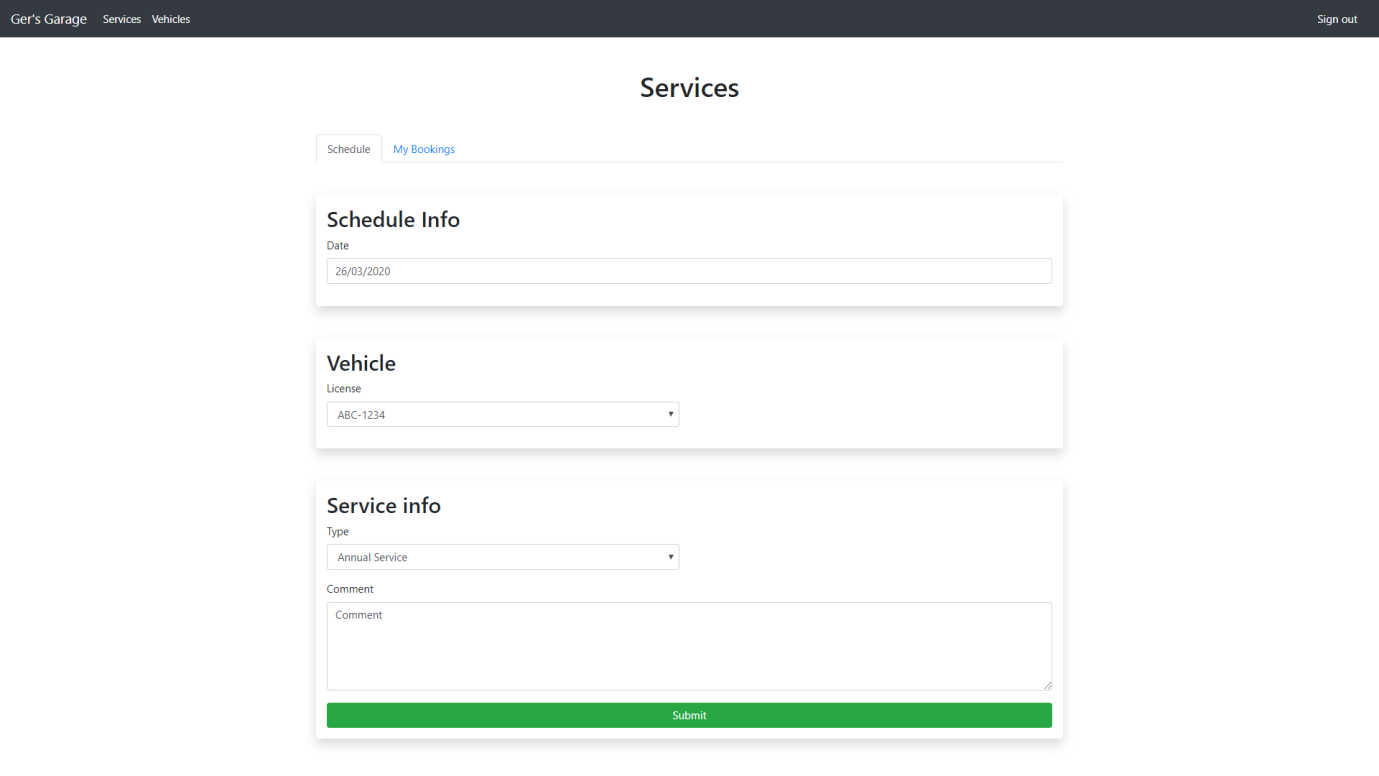
Vehicles:

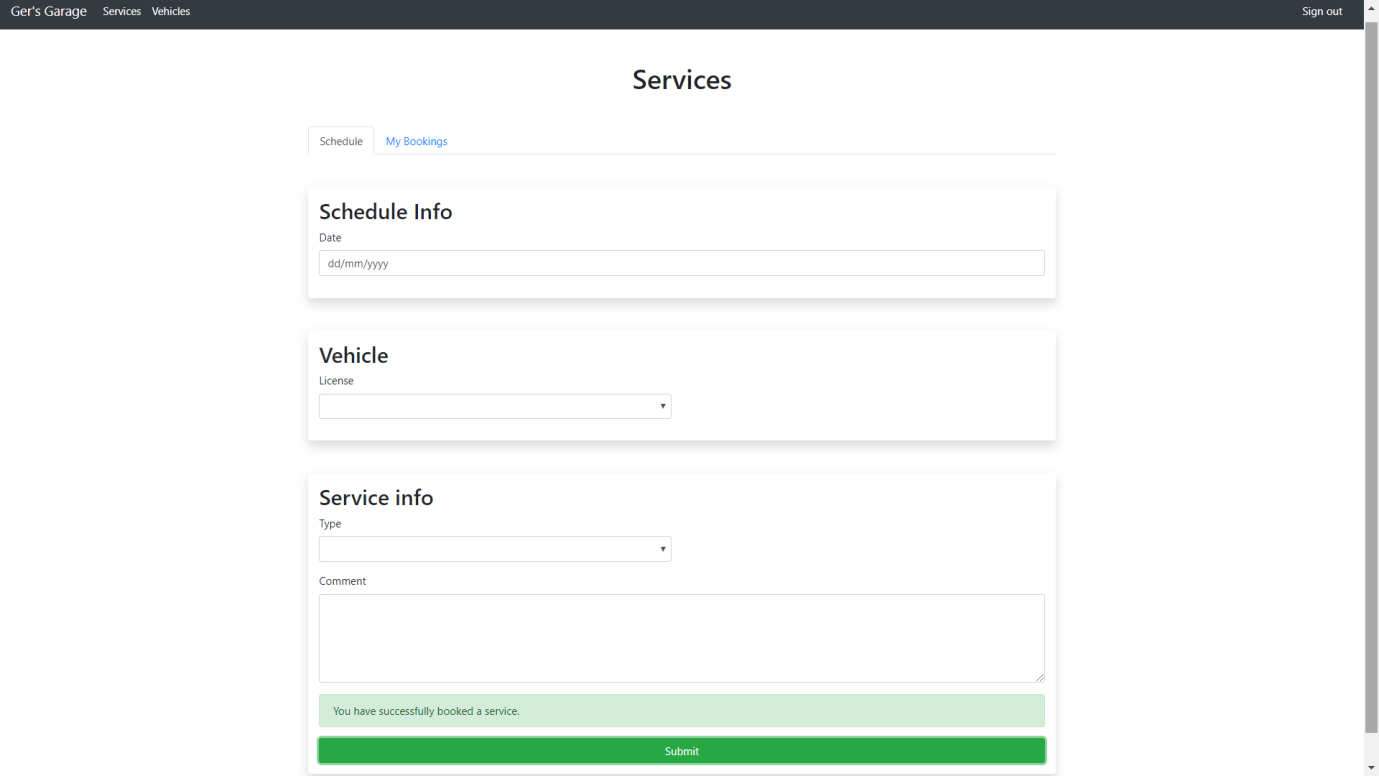
In the Vehicles tab the customers are able to register the vehicles that they want to book a service. They must inform the vehicle informations and when clicking on the Register button the system displays a success message and the added vehicle is shown in the list of vehicles below.



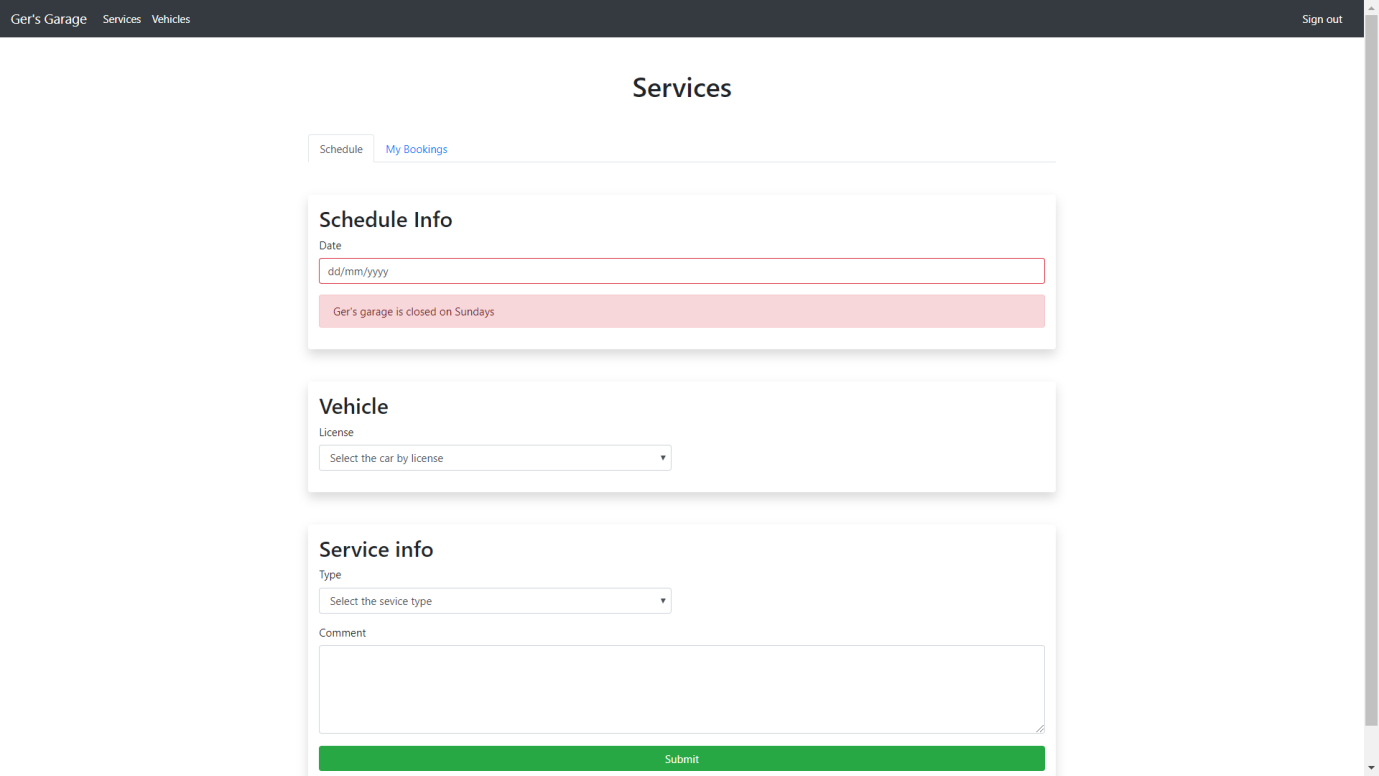
Services > Schedule:

In this tab the customers are able to schedule a service for the registered vehicle. They must inform a date, select the registered vehicle, the type of service they want to schedule and any comments. When clicking on the Submit button, the system displays a message stating that the scheduling was successful.

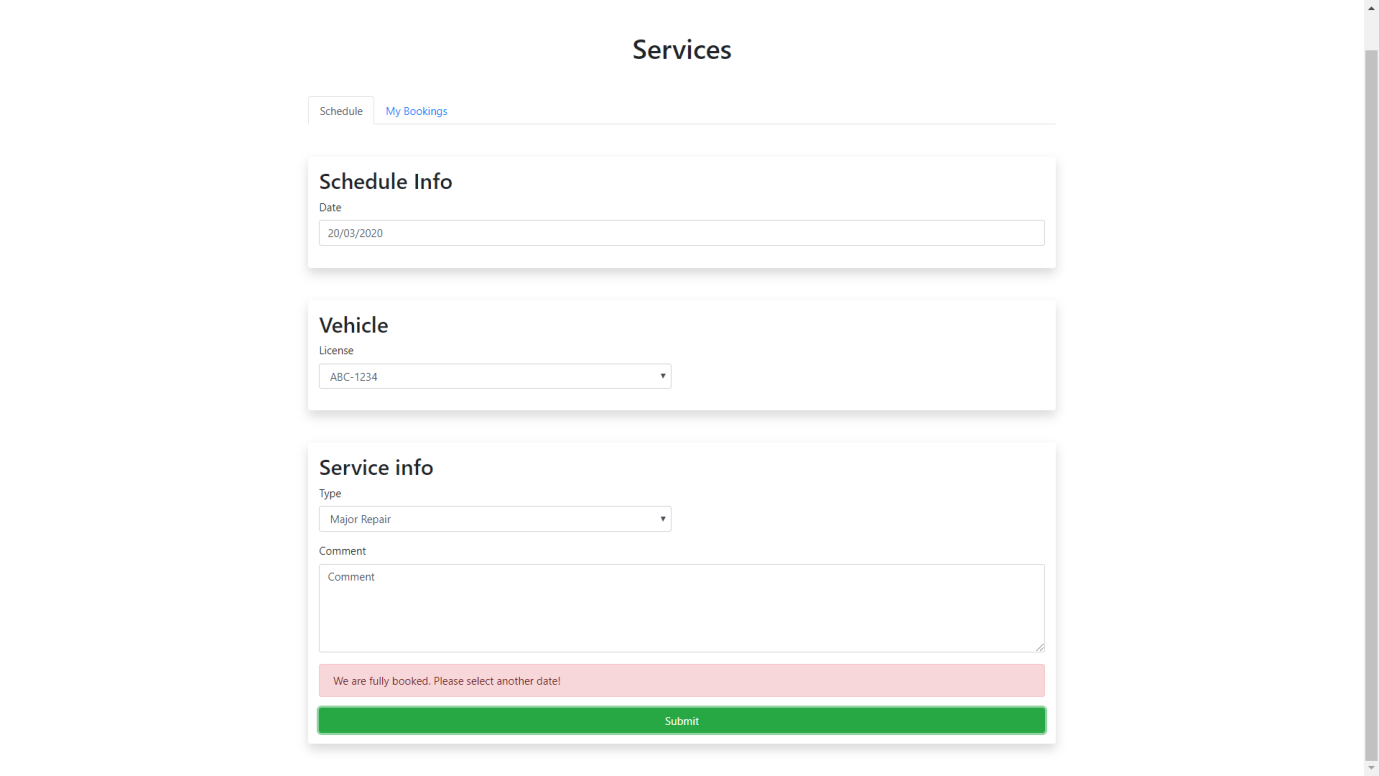




Note that the customer is not able to book a service on Sundays.

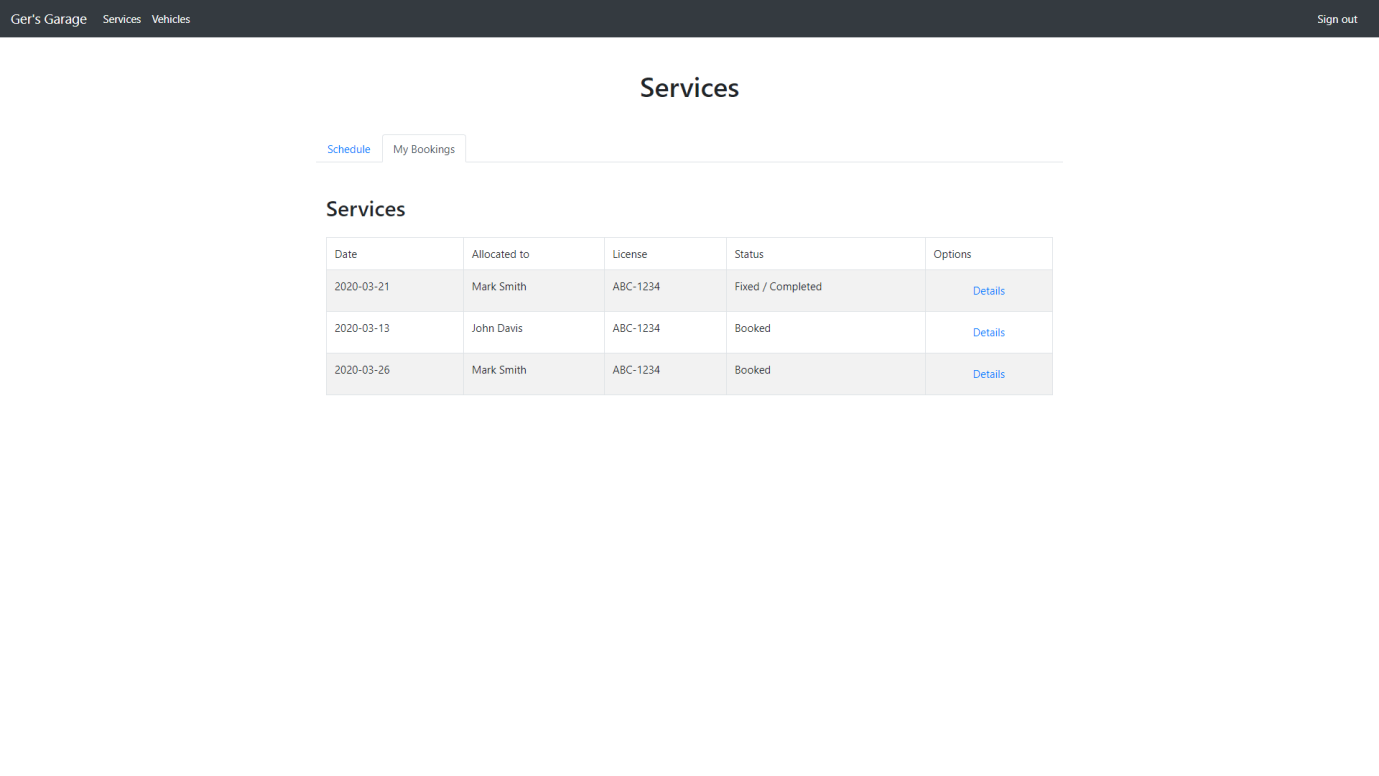


Also, the system allows only 4 bookings per day. It shows a message informing the customers that they must choose another day for their booking.



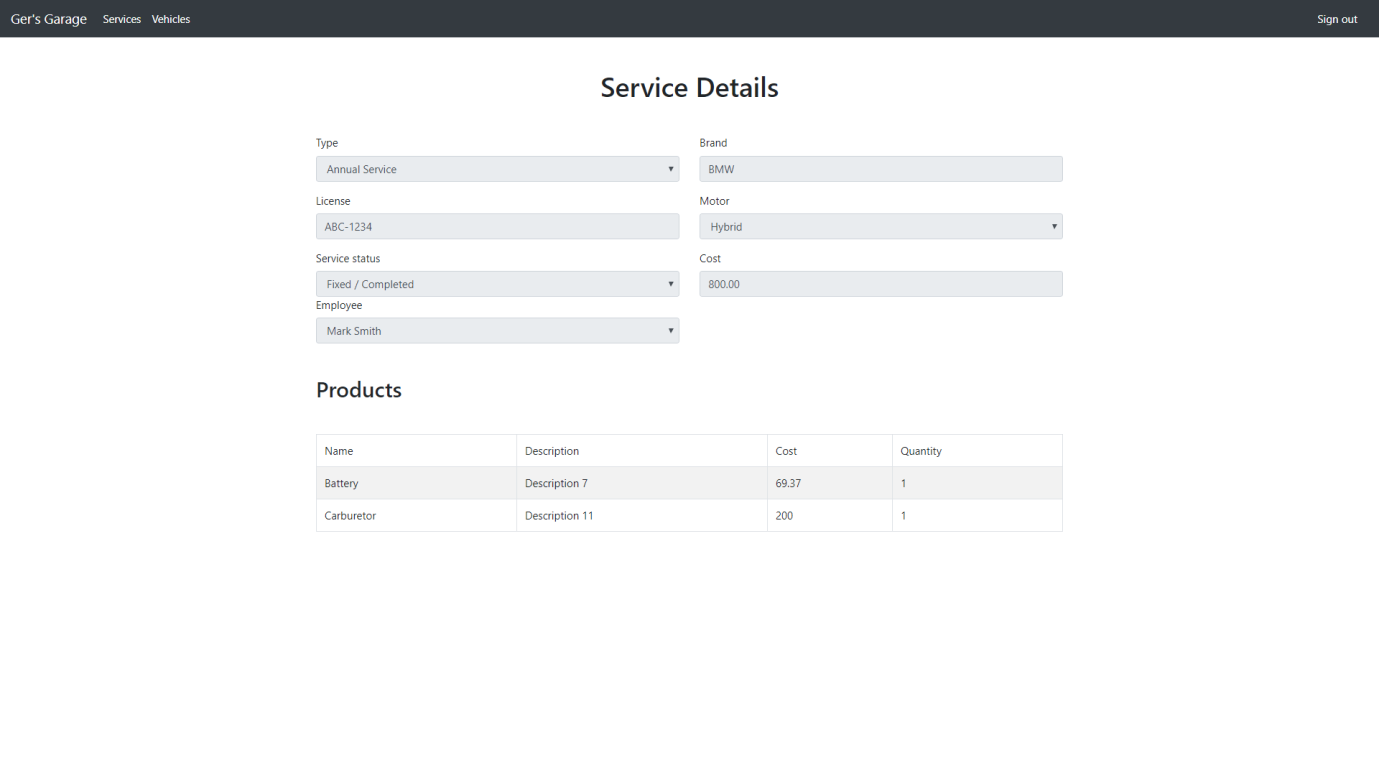
Services > My Bookings

In this tab the customers are able to consult their bookings.



Services > My Bookings > Details

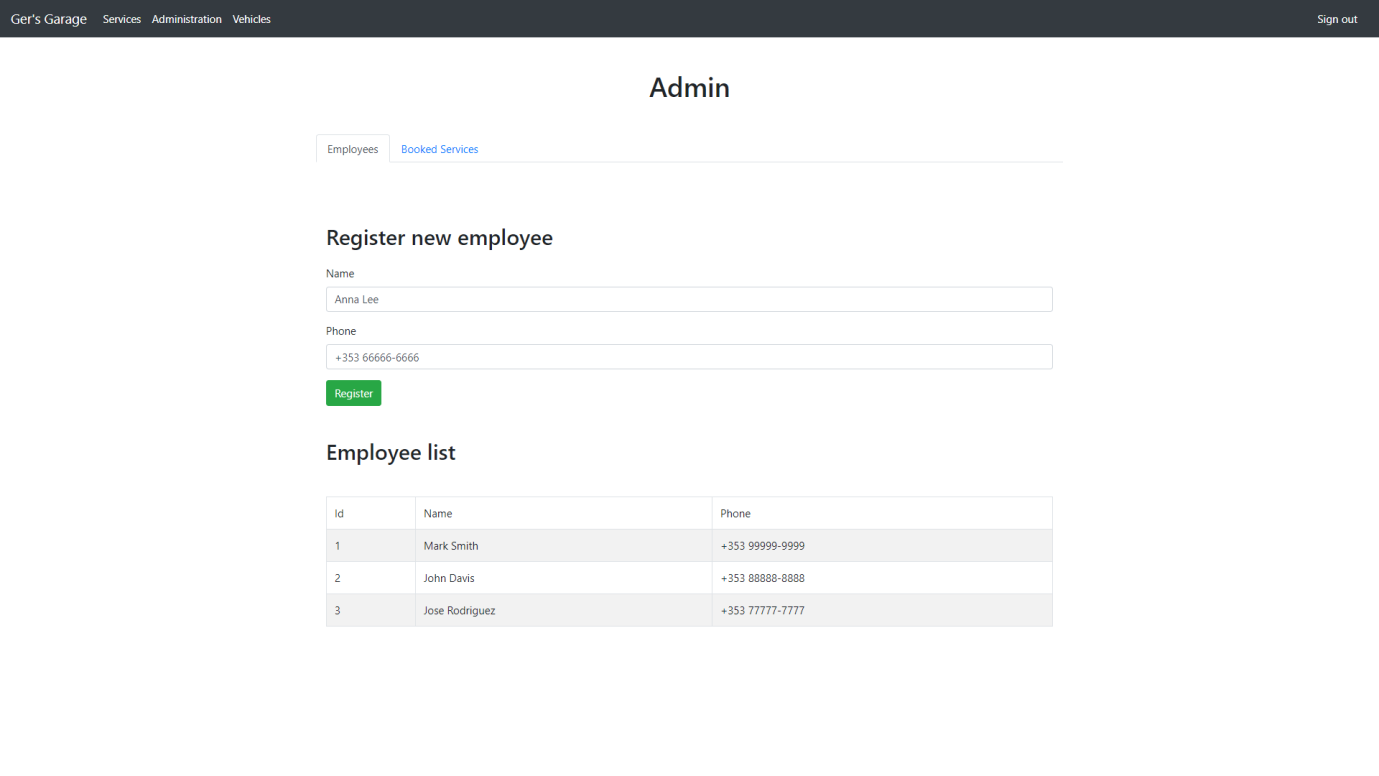
By clicking on the Details option, the customers are able to view details of their booking such as status, cost, employee who performed the service and extra parts / costs added to the service.

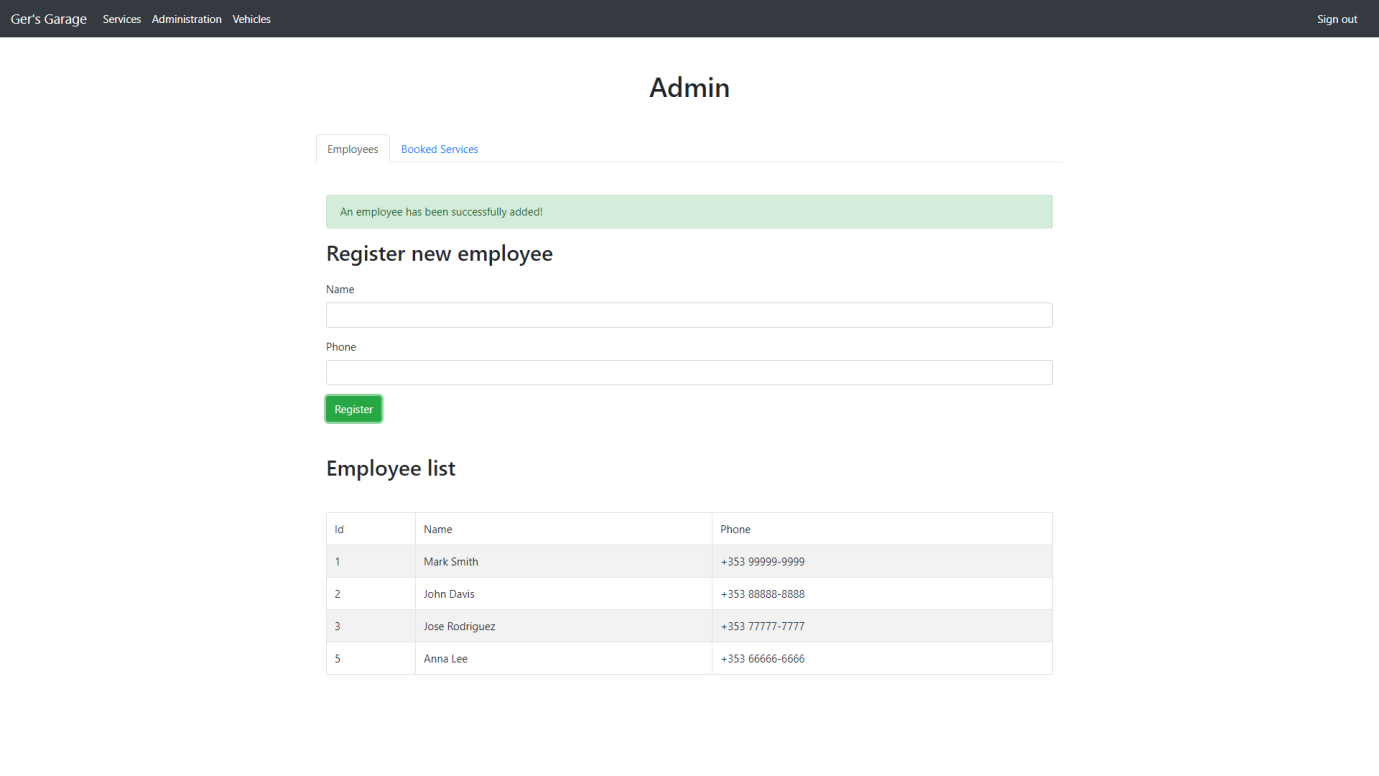


### 3.3.5: Logging in as an Admin

Administration > Employees:

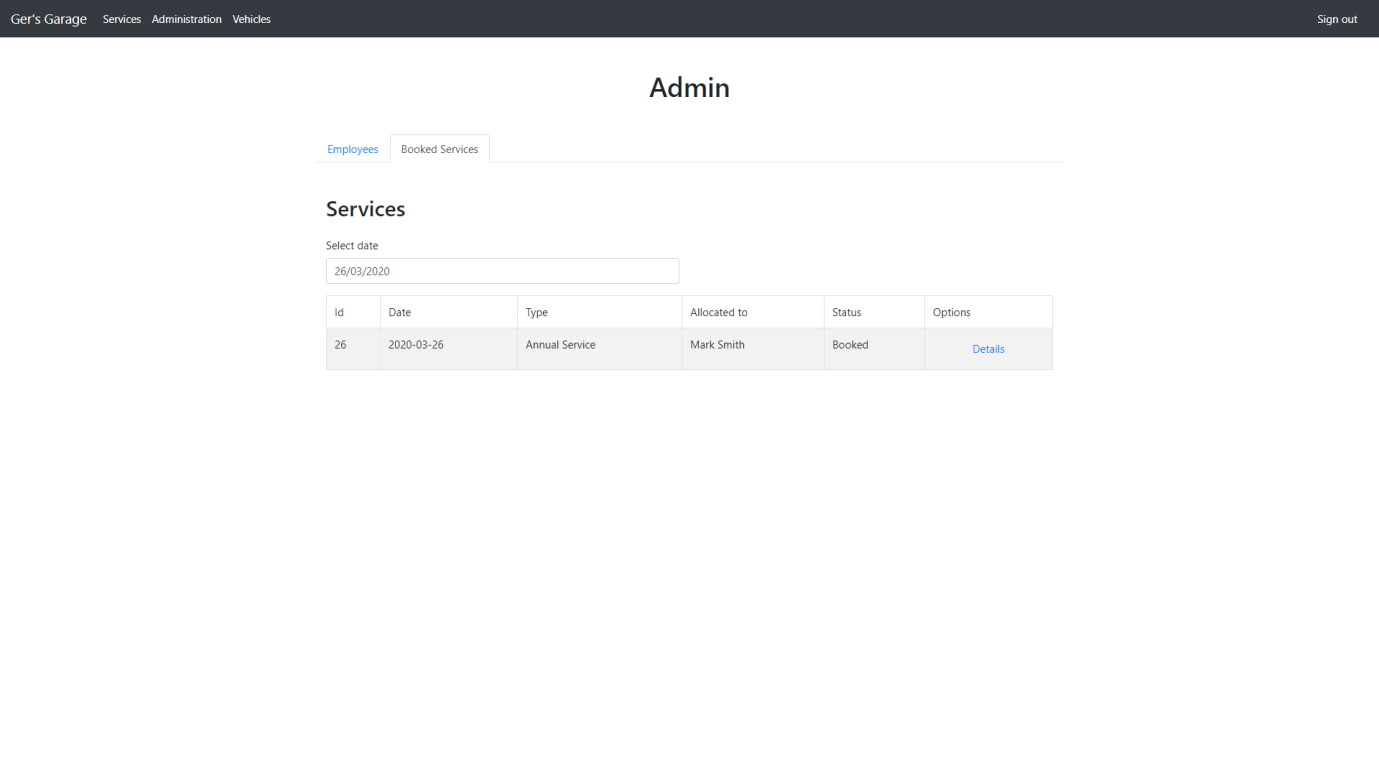
In this tab, the Admin is able to register new employees to the system. When entering the information and clicking on the Register button, the system displays a message that the employee has been successfully created and the new employee is displayed on the list of employees below.





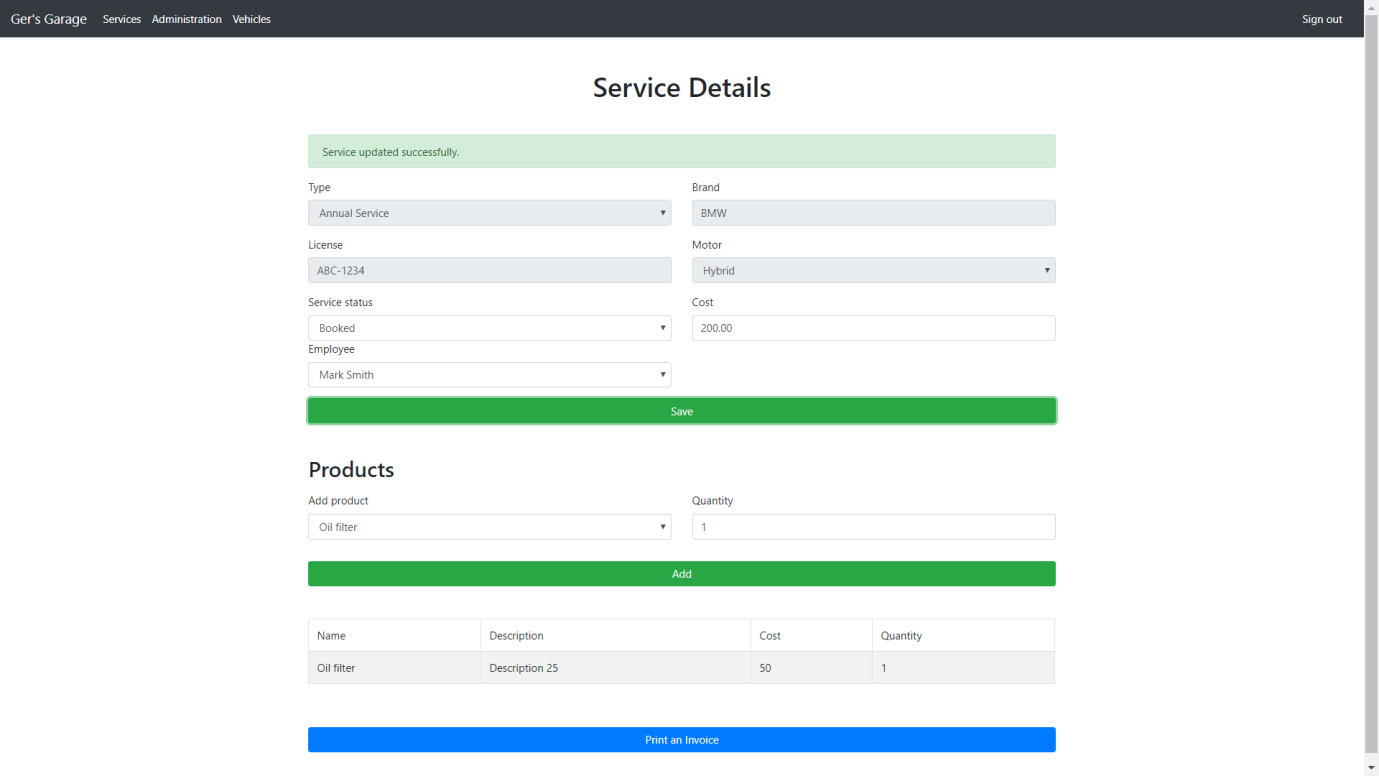
Administration > Booked Services:

In this tab, the Admin is able to view the schedules for a specific date. To do this, he must select the desired date and the system will display the schedules for that date, if any.

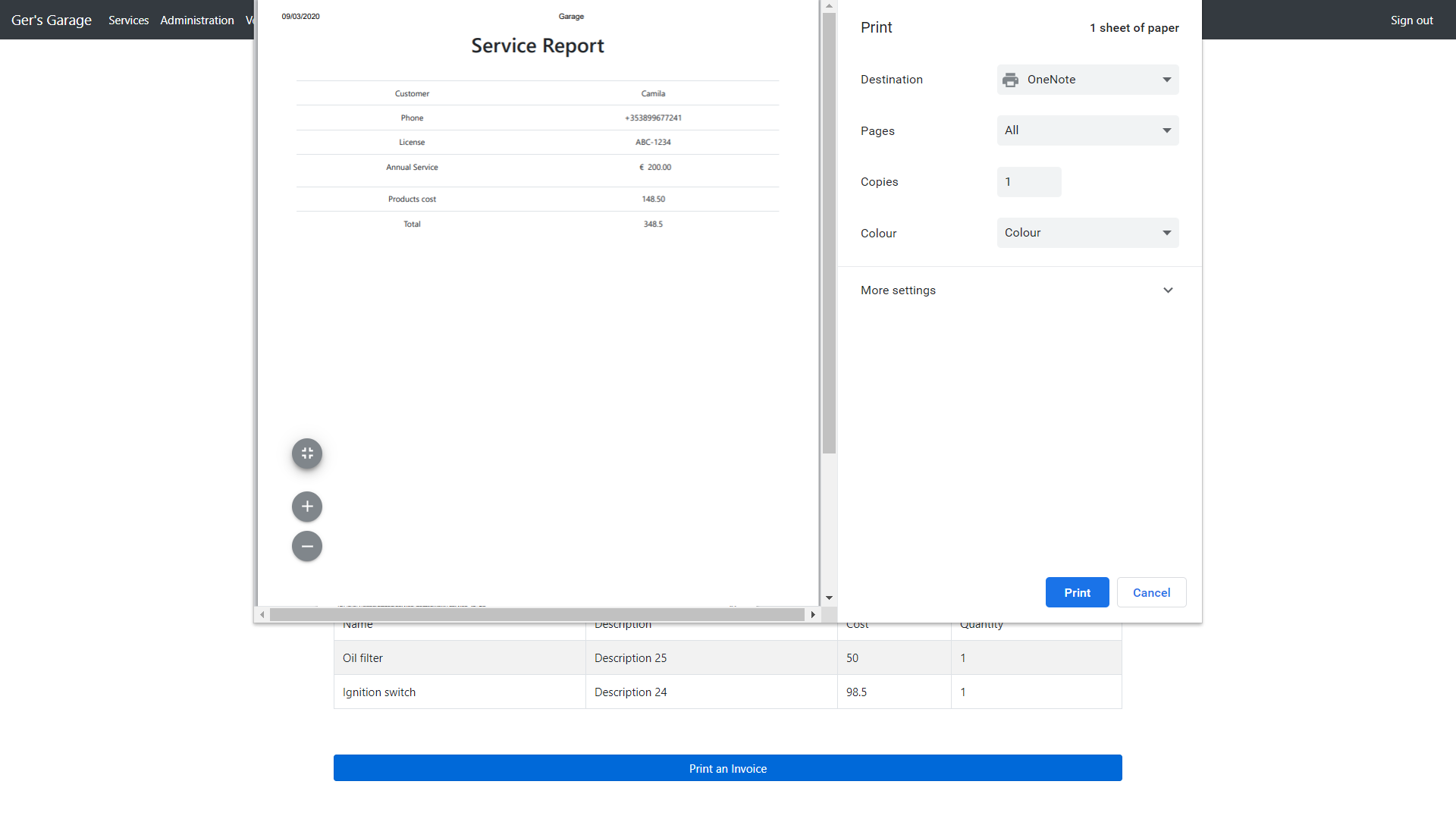


Administration > Booked Services > Details

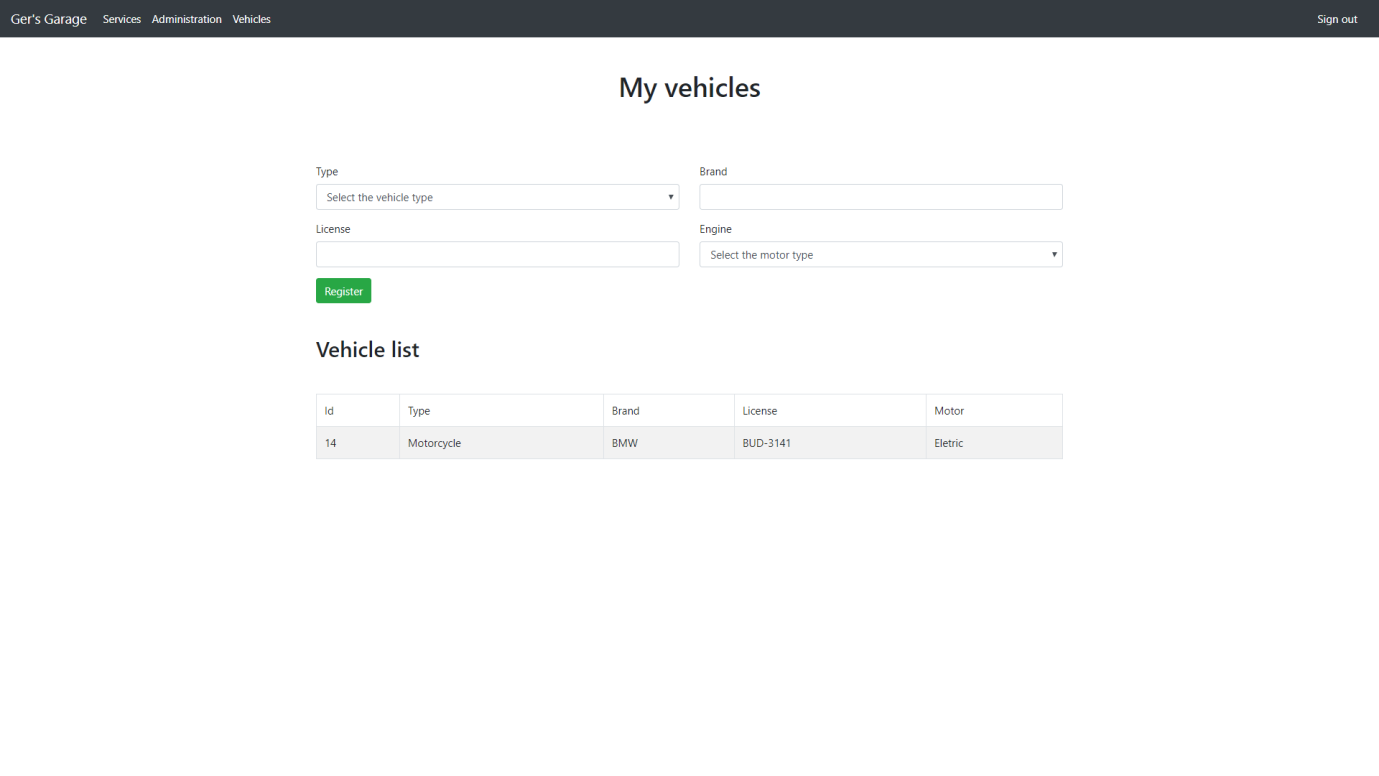
By selecting the Details option, the Admin can change the status of the service, allocate an employee to perform the service, inform the value of the service and add extra products to the service.

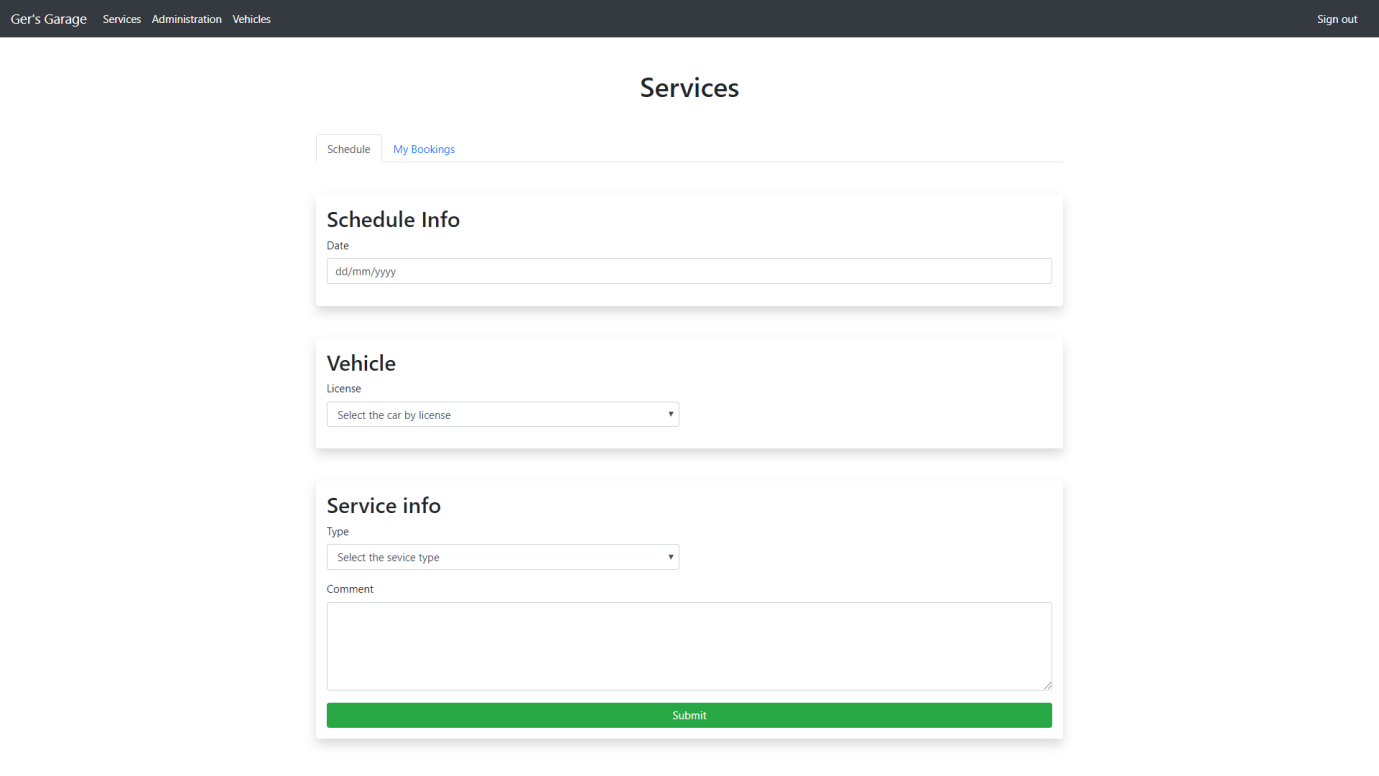


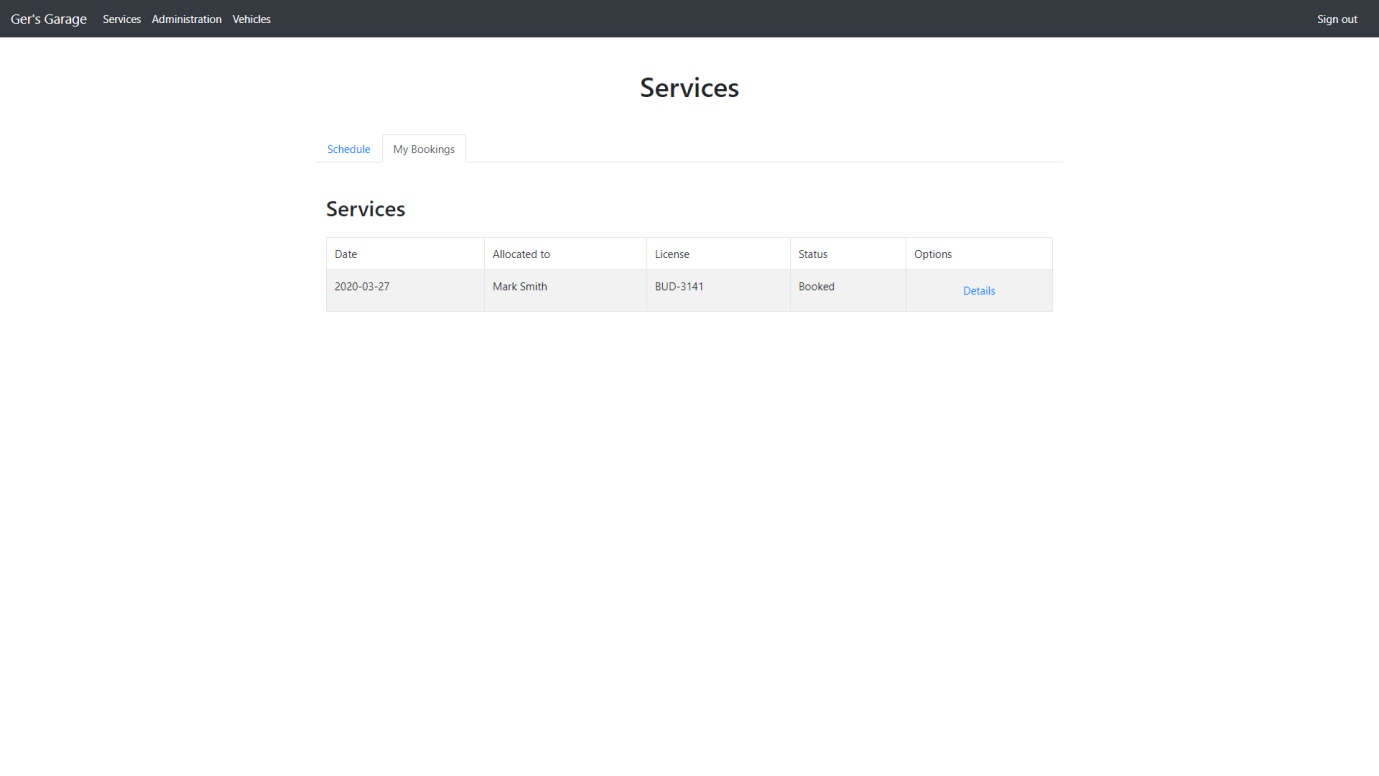
In this tab the admin can also print an invoice with a description of the services.

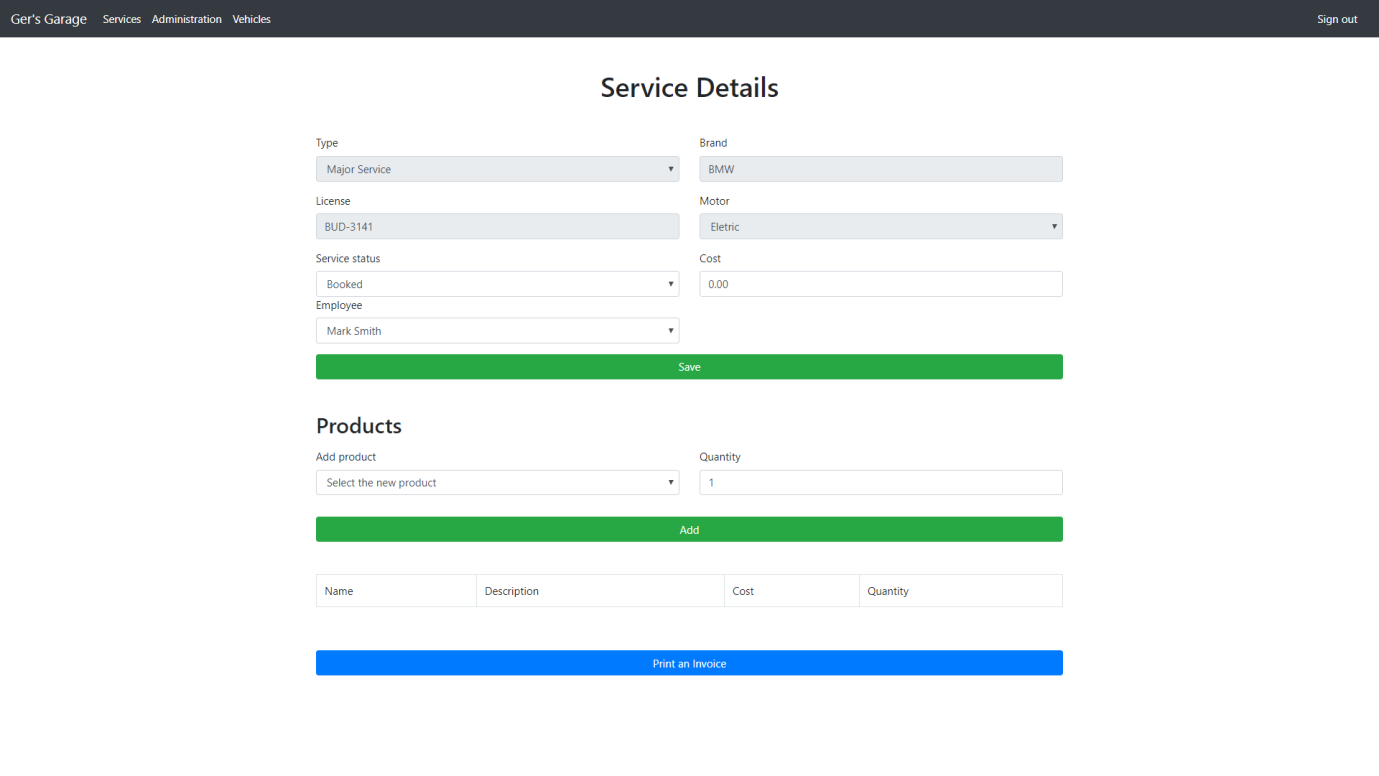


As well as a customer, the Admin is able to register a vehicle and schedule a service by accessing the Vehicles and Services tabs.









## 3.4: Data Design

Here are described the database tables and the relationships between them according to the Entity-Relationship Diagram that can be seen in figure 3 of this document.

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| Employees | - PK: ID  - Name  - Email  - Mobile  - Post | This table stores all staffs and their data. |

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| Services | - PK: ID  - Name  - Email  - Mobile  - Post | This table stores all services booked by a client. |

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| Vehicle | - PK: ID  - Type  - Brand  - License  - Motor  - FK: idCustomers | This table stores all vehicles registered by a client. |

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| Products | - PK: ID  - Name  - Cost  - Description | This table stores all products in the garage. |

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| Users | - PK: ID  - Name  - Email  - Phone  - Role  - Password | This table stores all registered users on the system. |

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| ServiceProduct | - PK: ID  - Cost  - FK: idServices  - FK: idProducts | This table stores all extra products that may be added to the service. |

Table 4

# 4: Implementation of the system

This section will describe how the system was developed. The system was developed using PHP as the main language, Javascript was also used to manipulate some elements as well as user animations in the frontend.

This chapter presents the tools, technologies and method used to develop the system.

## 4.1: Technologies used

This is a project with the application of technologies such as HTML, JavaScript, CSS, PHP among others where each one was used in its best specificity.

HTML5, CSS3 and JavaScript were used to develop the Front End. For the Back End, PHP was used for business rules, the JQuery library for communication between the client and the server. For the model layer, MySQL database manager and PHP PDO were used as the connection mode to assist in data persistence.

Below we see a more detailed description of each technology.

Database:

**MySQL** – It is considered one of the most popular databases. MySQL has numerous advantages, such as compatibility with various operating systems, including Linux, Windows and Mac. It consists of a Database Management System (DBMS), which works with the SQL language. This way, to interact with the MySQL database, it is necessary to know the SQL commands.

The acronym SQL stands for "Structured Query Language", used as standard in several relational model databases. In this model, data is recorded in tables, which relate to each other.

Computer Langauges:

**HTML** – It is a markup language that serves to give meaning and structure to web content, such as defining headers, paragraphs, making references or inserting images and videos.

As it is a markup language, the data is encapsulated or "tagged" into the HTML elements, which define and describe its purpose on the web page.

The browser then reads the HTML code, which tells it which parts correspond to the headings, paragraphs, links, etc. The HTML describes the data for the browser, which displays it according to the instructions.

**CSS** – It is a style sheet language, which has the role of making a page presentable on the web, directly related to design and appearance. What it does is to separate the written structural part, that is, the content, from the visual part, that is, the graphic design of this same page.

With CSS it is possible to create animations, visual effects, dynamic websites, etc., besides making the code less polluted and more organized.

**PHP** – It is a programming language especially suitable for web development and can be embedded in HTML documents.

The main differential of PHP is to run alongside the server, promoting excellent database interactions. The programming is specially indicated for the development of systems, applications and sites that need to deal with confidential data, with enhanced security.

For this project, among other functions it was also used to perform communicaton with the server. It was possible to interact with the database and other applications, implementing dynamic and complex functions on the website.

**JavaScript** – It is a behavior programming language that allows the creation of dynamic content, media control and animations to make a website more interactive and interesting.

With JavaScript, you can display messages and other interesting information, make checks or dynamically change the visual presentation of the pages, depending on the behavior you want your page (or application) to have.

For this project it was used to add interactivity and other dynamic features to the webpage.

Frameworks:

**Jquery** – It is the most famous JavaScript library. It was created under the “Write less, do more” mantra and that's exactly why it is so surprising, with a few lines of code you can make the most varied effects that used to cost dozens of lines of code with pure Javascript or a few hours of work in Flash.

You can use jQuery to create more dynamic pages, make AJAX requests or even changes to the page after it loads.

For this project it was used to add interactivity and dynamism to the website.

**AJAX** – Stands for Asynchronous JavaScript and XML. It is a set of web-oriented development techniques that allows applications to work asynchronously, processing any request to the server in the background.

JavaScript is a well-known programming language. Among its features, is the ability to manage dynamic content on a website and allow dynamic interaction with the user. XML (eXtensible Markup Language), as the name suggests, is a variation of HTML-style markup language. While HTML is used to display data, XML stores and transmits it.

Both JavaScript and XML work asynchronously in AJAX. Thus, any application that uses AJAX can send and receive data from the server without having to reload the entire page.

**Bootstrap responsive** – It is a front-end framework that facilitates the development of websites with mobile technology (responsive) without having to type a line of CSS to "make and happen".

Besides that, Bootstrap has a diversity of components (plugins) in JavaScript (jQuery) that help the designer to implement: tooltip, menu-dropdown, modal, carousel, slideshow, among others without the least difficulty, just adding some settings in the code, without the need to create scripts and more scripts.

The main and logical goal of Bootstrap is to increase productivity, decreasing the time in Front-End development.

Perhaps one of the most interesting features of this framework is the fact that it has an extensive library of components, such as icons, text boxes, panels and link colors.

Besides standardizing a series of parameters, Bootstrap is one of the easiest ways to make a site responsive, that is, to make a site adaptable to different types of screens, such as cell phone, tablet or computer.

**JSON** – It is a data structure in javascript. It is a standard for formatting data that will be transmitted between two applications of different languages, that is, JSON is mainly used to allow the exchange of data between languages, among other functions.

**Waterfall methodology**

When bilding a website it is extremely important to choose a development methodology that meets the project requirements.

This choice helps the developer both in the organization and in the prioritization of activities.

For this project was chosen Waterfall methodology that has sequential elements that guide the software development, allowing all functions to be structured correctly.

The main objective of this system is that the different stages of development follow a sequence: The first stage is directed to the second and this one moves to the third and so on.

The activities that must be executed are gathered into tasks and executed sequentially, where a task only starts when the previous one has been completely finished.

## 4.2: Implementation of the system

The system was developed using the PHP language for implementation on the server and HTML, CSS, JavaScript, Bootstrap 4 e jQuery technologies were used to develop the screens and system validations. For the communication between PHP and the Front End, Ajax (JQuery) was used.

After the user logs into the system, the logged in user is returned to the client and the front-end application keeps the permissions and everything the user has (except the user's password), after that the entire part of the user's authorization is done via javascript, some buttons are invisible depending on the user's permission.

All information in the System is dynamic, it is possible to register products, register mechanics, all under the control of the administrator.

Users who register to the System will be able to request repair services and any mechanic who is available can start the service. When starting a service, the mechanic can register parts that were necessary for the repair and / or extra services that were necessary to do.

The Invoice Form printing was done via Javascript, as soon as the user clicks on the "Print an Invoice" button, the Javascript via jQuery "hears the click on the button" and begins to set up the report table. The report elements already exist and have been implemented using Bootstrap 4 classes that show the elements when in print mode and hide them outside that mode. When the entire report is ready, the window.print () function is called.

## 4.3: Problems encountered

The first problem encountered was the short time to develop a relatively large system.

Besides that, the biggest difficulty found was in getting the API to receive JSON (which is common in REST web services), unfortunately there were many complications when it came to receiving the body in JSON format and deserializing it, so the alternative solution used was to receive data through POST forms to register items.

As only JQuery was used to manipulate the DOM of HTML pages, it was necessary to seek knowledge about the bootstrap in its documentation https://getbootstrap.com/docs/4.0/getting-started/introduction/ and the JQuery documentation https: / /api.jquery.com/ to facilitate the maintenance of the DOM.

# 5: Testing and Evaluation

This section describes how tests were performed on the system. Informal tests were carried out by the programmer herself, as the system was implemented.

With the defects discovered, corrections and improvements were made in the code in order to eliminate errors and flaws found.

The evidence for these tests can be seen in chapter 3.3 of this document.

# 

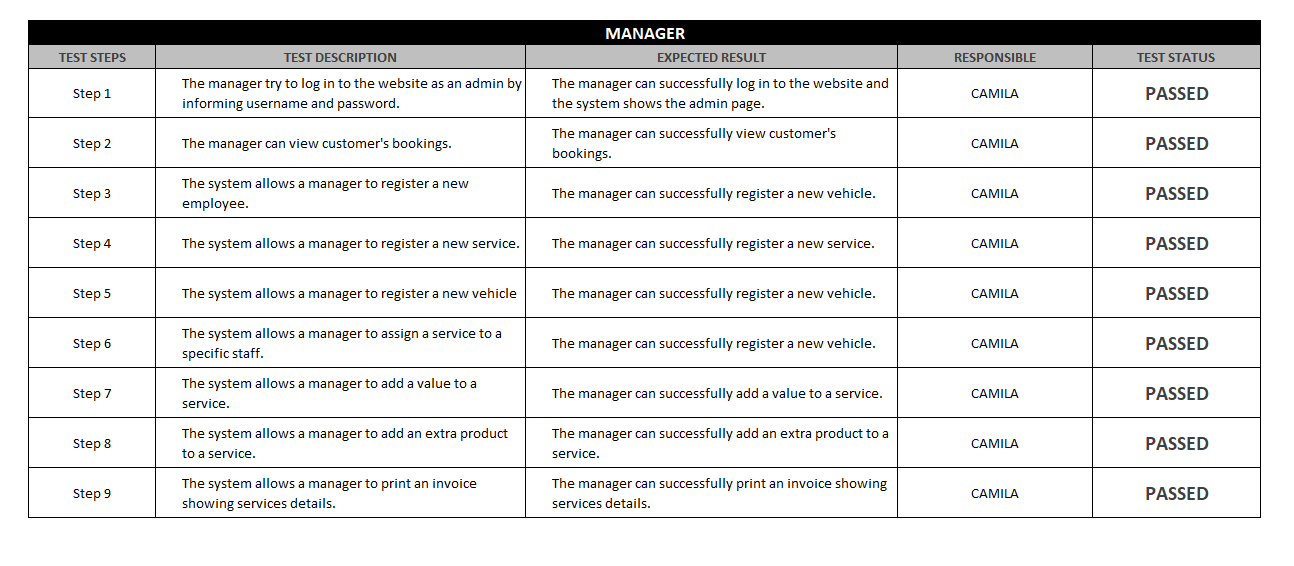


Figure 8

# 6: Conclusions

This work aimed the development of a Web Application for the management of Ger’s Garage.

The use cases necessary for the development of this system were elaborated from the requirements survey, in which the brief description of the project provided by CCT College was analyzed to identify which functionalities are necessary for the system.

To facilitate the understanding of the system were also developed the use case diagram, class diagram and the entity relationship diagram, thus allowing an overview of the system.

After having the documentation ready and defined the technologies to be used the system was developed using techniques and concepts such as object orientation, client/server model among others.

The system was properly tested, to ensure the correct operation of what was coded, the defects were corrected and improvements were made in the implementation.

The use of this system allowed the client to computerize and improve the processes within its garage in an efficient and practical way, saving time in customer service and controlling the services provided in a safe way.

As presented it can be seen that the objective of this work was successfully achieved, which was to carry out the analysis and development of a Web Application for the management of Ger’s Garage.

# Appendix A: Code Listings

This project can be found on GitHub:

<https://github.com/camilacsantos85/Ger-sGarage>

# List of References

() Bootstrap Tutorial, Available at: https://www.w3schools.com/bootstrap/default.asp (Accessed: 29th February 2020).

() Bootstrap, Available at: https://getbootstrap.com/ (Accessed: 29th February 2020).

() jQuery Effects - Animation, Available at: https://www.w3schools.com/jquery/jquery\_animate.asp (Accessed: 29th February 2020).

() What is jQuery?, Available at: https://jquery.com/ (Accessed: 29th February 2020).

() PHP Connect to MySQL, Available at: https://www.w3schools.com/php/php\_mysql\_connect.asp (Accessed: 29th February 2020).

() PHP, Available at: https://www.php.net/ (Accessed: 29th February 2020).

() PHP & MySQL Tutorial. Learn how to use PHP & MySQL, Available at: https://www.siteground.com/tutorials/php-mysql/ (Accessed: 29th February 2020).

() , Available at: https://www.youtube.com/ (Accessed: 29th February 2020).

() What is Waterfall Model in SDLC? Advantages & Disadvantages, Available at: https://www.guru99.com/what-is-sdlc-or-waterfall-model.html (Accessed: 29th February 2020).

(2019) What Is SDLC Waterfall Model?, Available at: https://www.softwaretestinghelp.com/what-is-sdlc-waterfall-model/ (Accessed: 29th February 2020).

() CSS Introduction, Available at: https://www.w3schools.com/css/css\_intro.asp (Accessed: 29th February 2020).

() MySQL Tutorial, Available at: https://www.mysqltutorial.org/ (Accessed: 29th February 2020).

() What is MySQL?, Available at: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html (Accessed: 29th February 2020).

() What is Unified Modeling Language (UML)?, Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/ (Accessed: 29th February 2020).

() JSON - Introduction, Available at: https://www.w3schools.com/js/js\_json\_intro.asp (Accessed: 29th February 2020).

() Introducing JSON, Available at: https://www.json.org/json-en.html (Accessed: 29th February 2020).

(2020) Working with JSON, Available at: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON (Accessed: 29th February 2020).

() MySQL Functions (PDO\_MYSQL) , Available at: https://www.php.net/manual/en/ref.pdo-mysql.php (Accessed: 29th February 2020

() (The only proper) PDO tutorial, Available at: https://phpdelusions.net/pdo (Accessed: 29th February 2020).

() jQuery.ajax(), Available at: https://api.jquery.com/jquery.ajax/ (Accessed: 29th February 2020).