Ger`s Garage Report

Felix Edmundo Higuera Alonso

ID:2019057

CCT College

High Diploma in I.T.

January 2019

Github link: https://github.com/felixhiguera/garage.git

Abstract

Nowadays, companies are looking for a better way of having a good service in order that the customers can feel satisfied with the service, the competence is increasing and every business has to find the way how to improve the business. The reason for this project is making a website for the company Ger´s Garage in order that the customers can make an appointment for delivering the vehicles to repair them and the company can administrate the time table depending of the appointments done in the website. The customers will be able of making an appointment from the cell phone or computer because the website is responsive.

Acknowledgements

I would like to express my special thanks to my girlfriend Iky who was supporting me during this period of studying the high diploma in IT and my parents who in spite of not being with me in this country, they have been pending in my life and my progress.

Index

Table Of Content

Chapters

Appendices

Chapter 1. Introduction

As the internet has become a part of our life, companies take advantage of it to connect their products and services to people from around the world to grow their business. The most common way is via a business website. The success of Amazon starts with the website where you can buy a wide variety of products such as electronics, videogames, computers, clothes from different locations. Amazon is one of the good business examples showing people the importance of being open to everyone who has a device connected to the internet.
Ger´s Garage is a local business with a few employees, carrying out maintenance checks for all kinds of small to medium vehicles such as motorbikes, cars and small vans. Ger, the owner of the garage, realized that he will gain more business opportunities and reduce human errors by expanding the business online and having it digitalize. Therefore, this project is to fulfil his requirement by providing an online service to allow customers to book their vehicles in for a check-up or service. By doing so, the problem of appointment clashes caused by manual acts can be solved.

The project is a responsive website. One of the advantages of a website over an app is simple to use without installing and concerns of software updates. Customers can create an account with the personal information like name, phone number, email and password. After that, they are able to log in and make an appointment for the car service. The website also shows if there are any appointments available when customers make an appointment. The administrator of the website has access to all the appointments made and he can assign tasks so that mechanics specialists can know about what cars have to be repaired or need a service every day. To summarize, here are the brief project requirements. The details of the project functional requirements are listed on Chapter 3.

USERS:

- to sign up or register for making an appointment online
- to allow users to view vehicle status and history

ADMINISTRATORS

- to know there is an appointment coming in
- to access the appointment such as viewing information, assigning tasks and billing invoice

The plan for this project is first, understanding and analysing the project, followed by drafting a flow chart, e.g. for making an appointment customers have to log in first and before login, they need to sign up. key elements are written down along each step, e.g. name, phone number, email and password for sign up process. The flow chart with key elements is very important when it comes to coding the database. After understanding the flow of the process, it is important to design how the website looks. Creating wireframe with Balsamiq Mockups helps visualize the website structure and understand how it can be implemented. Finally, the frontend of the website can be built with bootstrap. Here is the list of specific objectives of the project:

- to provide customers with good user experience when browsing the website across devices whenever and wherever
- to provide the administrator with a simple-to-use interface

- to pull accurate data anytime from the database to avoid confusion from customers and administrators

The technologies that are covered in the frontend of this project is HTML for the frame, CSS for making it good-looking and Javascript for making it more dynamic with additional effects. For the backend, PHP is used to connect to the database made of MySQL.

This is a meaningful project because I see my strengths and skills are very well demonstrated in the project such as my ability to use bootstrap that is the area I want to develop in my future career. Besides, the exercise practical as all we learned during the course can be put together here. For example, in the class of web development, we learned how to make a website with HTML, CSS and Javascript. In the database class, we learned how to compile a database. In the cloud class, we learned how to make the backend and how to communicate the backend with the frontend. It displays all the knowledge we got from the 1-year course. Moreover, this project provides us hands-on experience with a realistic case study, paving the way for another real-world project.

Chapter 2. Literature Review

The section is to study and analyse the existing academic research. By comparing and contrasting different technologies and software, the most appropriate ones were selected to match the specific requirements of Ger Garage business.

After carrying out comprehensive research, this project is a website-based software instead of an App, being built with Bootstrap, PHP and MySQL. Bootstrap was used for frontend development while PHP and MySQL were selected for backend system.

The two main considerations are the budget and audience reach. Many existing studies show the similar findings (William, 2013; Michalis, 2017) that websites can reach broader audience as any mobile users can access mobile website regardless of the operating system and the web development cost is cheaper. In the situation of Ger Garage, it is a small business and business awareness is important. With the website-based tool, it is more affordable to Ger and SEO can be implemented (William, 2013), unlike an App, to bring organic traffic to the business, thus drive the awareness.

While Apps have been evolving to offer users a better experience when working with phone native features like camera and sensors, these features may not be necessary for car service booking. Another issue to consider is that Ger Garage will not expect users to use the booking system for more than 5 times a day like social media apps, so a simple-to-use and responsive website will be sufficient enough to provide a good user experience and allow people to complete the process without hassle. Based on the analysis, website is used for developing the car booking system for the Ger Garage.

Today, building a website can be very easy without extensive knowledge of coding thanks to Content Management System (CMS) (Srivastav, 2016). However, companies today still employ web developers to build websites via coding. In this project, Bootstrap, instead of CMS, is selected to be responsible for the frontend development.

Bootstrap is a HTML, CSS and JavaScript-based framework to develop responsive, mobile-first websites. It contains CSS and Javascript templates for typography, forms,

buttons, etc. It was developed by Mark Otto and Jacob Thornton with the idea of standardizing the toolset that they used when they developed for the front-end. They also developed a set of JavaScript and CSS libraries for every user of the company.

The advantages of Bootstrap are that it helps make the website frontend faster and ensures the correct design. With set CSS styles and Javascript libraries, Bootstrap brings further benefits to users:
-Maintenance and updates from Twitter.
-Elements webs packs customize.
-Logic and simple use.
-It contains sufficient documentation by levels.

CMS like Wordpress where you can drag and drop photos, videos, text and so on is simple to operate since it is not necessary to write codes, but at the same time it is difficult when it comes to editing code. For example, when adding an image in the webpage, not only does it generate the code for the image itself but also that in the library, making it difficult to locate the image in Wordpress. The website loading speed of Bootstrap is low too.
By comparing and contrasting the two open-source tools, Bootstrap suits Ger´s business more as it enables the coding to be compact and easy-to-debug. It also makes the website load faster since Bootsrap does not have much overhead or a huge number of functionalities like Wordpress.

With Bootstrap, HTML, CSS and JavaScript can be easily managed all together.

HTML (Hypertext Markup Language) is a language that is used for website development. It describes the basic structure of a page and organizes the way of how content displays. Besides, it can be added hyperlinks to other pages or documents.

CSS (Cascading Style sheet) describes how HTML elements such as fonts, colours, borders and images are displayed on the screen. It was developed by W3C (World Wide Web Consortium) and it allows us to separate the content from the documents written in HTML, XML, etc. CSS provides a simple and quick way to modify the website style. According to many studies (Waseem, 2016), CSS should be well utilized to keep the entire coding compact, make the loading speed fast and the webpages easy to maintain when files are saved separately.

When it comes to backend development, PHP is the backbone of the backend system. This is an open-source programming language and popular around the world. It can be added very easily to the HTML and thus connected to the frontend with the databse. This is one of the main reasons why the backend system was written with PHP. This language also has other benefits as follows:

- Users cannot view the code, making PHP secure.
- It is able to connect different database engines such as MySQL and PostgreSQL.
- There exist frameworks like Larabel or Symphony to make the programming work easy

- Problems of compatibility are minimal when uploading PHP-based websites to the cloud
- Developers can access assistance easily due to many resources online and the big community.

Comparing to another possible backend language JAVA, PHP has a lower development cost due to its open-source nature to match Ger's budget. PHP also has a faster page loading speed while JAVA is complex and takes a great amount of loading time.

Regarding to database management system, the most popular tool MySQL is used in the project to build Ger Garage website database. It was the most used tool in 2019 with 39% of developers using it. MySQL has some notable users like Facebook, Google, Flickr, GitHub, NASA, Netflix,etc. Some advantages over other database are:

- Data Security: It is one of the most secure technologies for the database that is why it is used for popular web applications like WordPress, Facebook or Twitter.
- Speed
- User-friendly: It is easy to install, configure and use, because the commands are in simple English
- It can be installed in different Operative Systems.

Chapter 3 System Analysis and Design

This chapter aims to describe the wireframe designs for the website and how they meet the functional requirements and data requirement of the project. This section is important for the system implementation (Chapter 4). Wireframe designs of key pages, Database tables, and relevant design diagrams such as a class diagram are included in this chapter to provide a comprehensive presentation.

As briefly explained in the chapter one, this project is to allow customers to make an appointment from cell phones or computers anytime and anywhere without, and enable administrators to access all appointment coming through.

Functional Requirements

USERS:

- to easily locate the sign up or login buttons for quickly making an appointment online
- to view if there are any available vehicle appointments
- to make an appointment and view the appointment record
- to check vehicle service status
- to view the old vehicle appointments

ADMINISTRATORS

- to know there is an appointment coming in
- to access each appointment

- to add items to each appointment (if necessary)
- to assign each appointment to specialists for conducting services
- to bill invoice easily

The function requirements are able to perform on a website in a logic manner across devices such as desktop computers, mobile phones and tablets. The functional requirements were laid out in the wire frames.

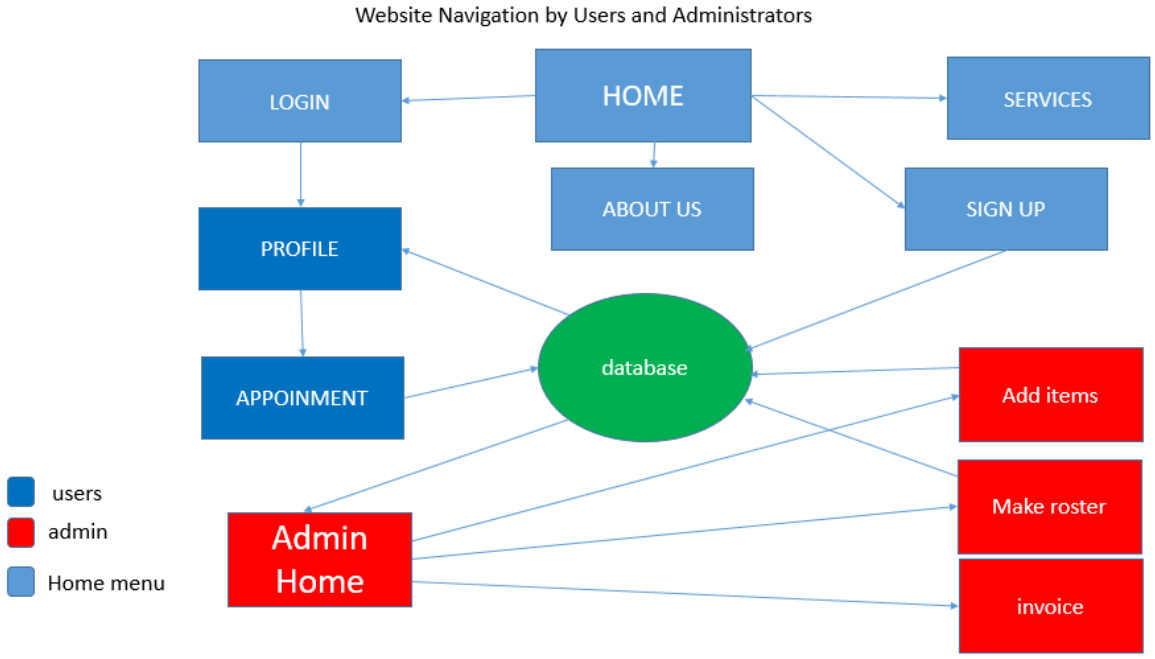


Fig.1 Website navigation by users and administrators

Before implementing the navigation on a website, wire wireframes were drafted so as to avoid a great amount of time wasted on coding on the real website, and some project requirements being ignored or mis-treated.

The wireframe design with a simple navigation allows users to clearly view the content of the website. For example, there are login and signup buttons at the top right hand corner, the similar approach to other websites. Therefore, users can get familiar with the layout and navigate the site easily. At the top left hand corner, users can see the options of home, about us and services. Next is the image. It is important to visualize what the company does with an image or even a video as these kinds of multimedia have strong stopping power to audience.

Fig.2 A Simple Design of Homepage Wire Frame

The website has three pages: Home, Services and About Us.

Home: It describes the concept of the business, where it is located, reasons why it is a good company and how visitors can contact them.

Services: It describes different kinds of services that the company has for example: Annual Service, Major Service or Repair.

About Us: it describes the history of the company and how the idea of repairing cars came from.

If the customer has not signed up, they have to register and create a user account in order to make an appointment. After having an account, they can login with the registered personal information and proceed to booking a service. If the customer has an account, they can just simply log in to request for the vehicle service.

For the Sign Up page, customers must fill in these fields:

- Name.
- Phone number.
- Email Address
- Password.
- Vehicle Type.
- Vehicle Engine.
- Driving Licence ID.
- Make Vehicle.

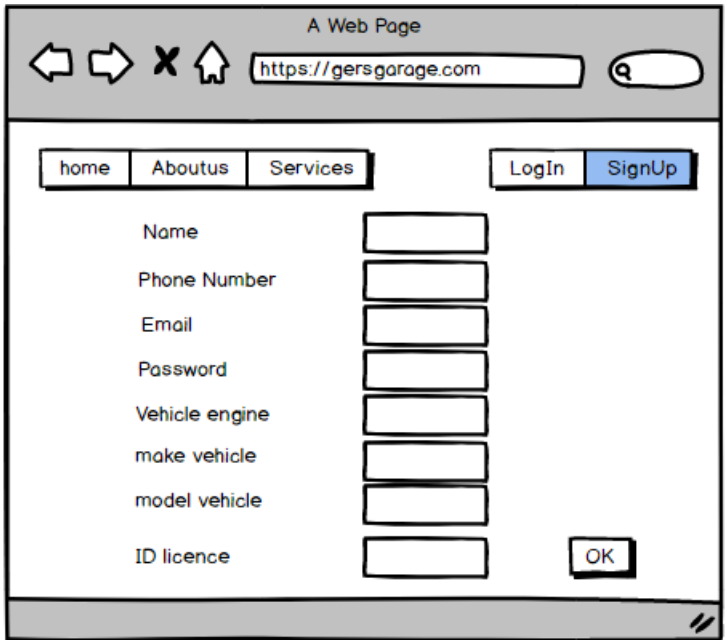


Fig.3 Sign Up Page

Once the user has an account, they can log in to access Ger Garage services..

To make an appointment, users need to select a day from the calendar and they can see whether the timeslot is available or unavailable. If the booking is successfully booked, the appointment status will be displayed in the account. All the service records are saved so that users can plan for the car maintenance easily in the future.

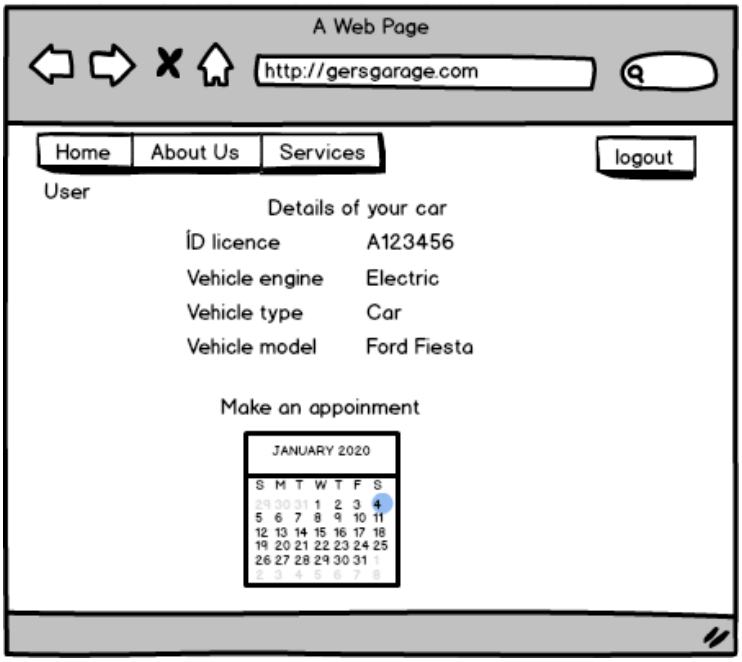


Fig.4 Making an Appointment

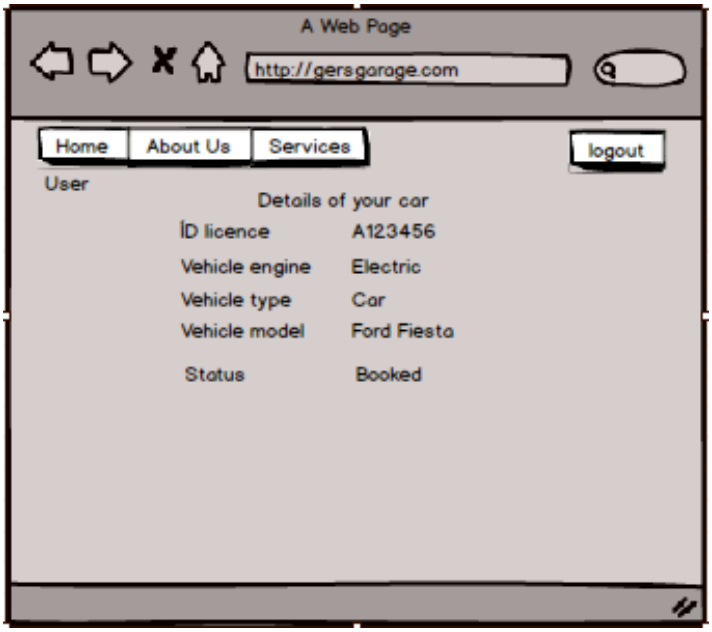


Fig.5 A Successful Appointment

After an appointment is made successfully, the administrator can see all the information from users e.g. the name, the vehicle and service.

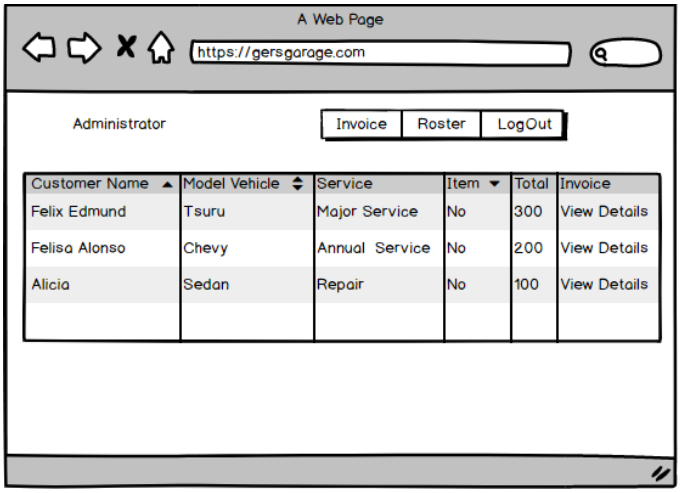


Fig.5 View the appointment details

The administrator also has options like adding items. ´Adding items´ means putting additional service items or product items in the system for each appointment. For example, if a new vehicle part is necessary to replace the old part, the administrator will add a product item to this user account.

Besides, the administrator can assign each task to the staff member who are registered and stored in MySQL database.

Since the cost of every product and service item is pulled from the database, it provides an accurate billing and saves administrators´ effort and time.

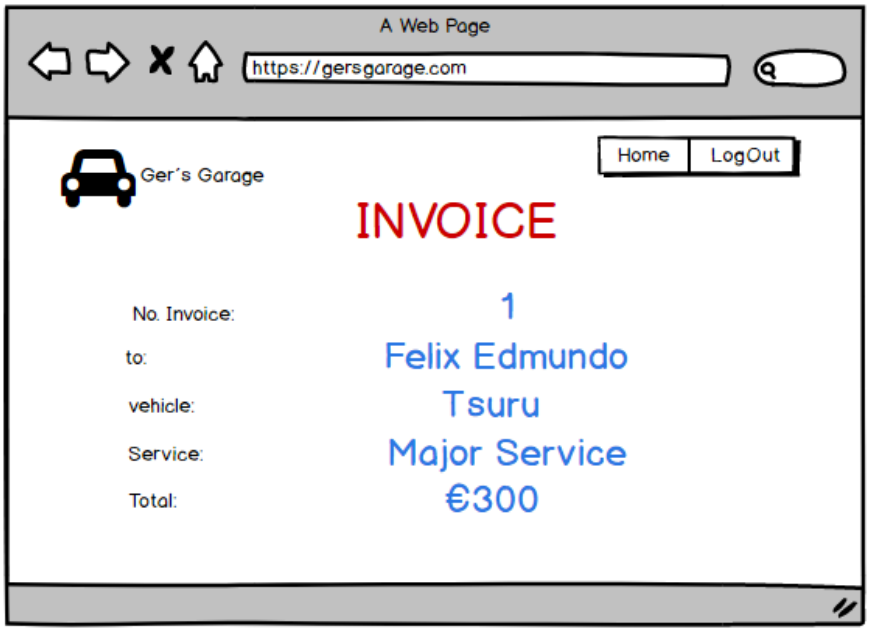


Fig.6 Invoice Billing

Data Requirements

Data requirements are the properties and the values from the properties of every object that are created in the database.

The database for this project includes some entities (objects) and every entity contains some attributes (properties). It can be seen in the following tables.

User: The entity ´User´ has ´ID´ and some other important attributes like email address, password, engine vehicle, make vehicle and model vehicle. They play an important role in login process and making an appointment.

User				
Attribute Name	Required	Type	Field Length	Default Values
ID	Yes	Number	n/a	auto increment
Name	Yes	Text	40	n/a
Email	Yes	Text	20	n/a
Password	Yes	Text	20	n/a
Phone	No	Number	12	n/a
Engine Vehicle	Yes	Text	20	n/a
Make Vehicle	Yes	Text	20	n/a
Model Vehicle	Yes	Text	20	n/a
No. licence	No	Text	20	n/a

Staff: The entity ´Staff´ has ´ID´ and name attributes only because just the name and ID are necessary for identifying staff members.

Staff				
Attribute Name	Required	Type	Field Length	Default Values
ID	Yes	Number	n/a	Auto increment
Name	Yes	Text	20	n/a

Appointments: For the ´Appointment´ entity, the class ´ID´ like every entity is important because is a single key and cannot be repeated.

Appointments				
Attribute Name	Required	Type	Field Length	Default Values
ID	Yes	Number	n/a	Auto increment
Date	Yes	Date	10	n/a
Status	Yes	Text	20	Booked
Item 1	Yes	Text	20	n/a
Item 2	Yes	Text	20	n/a
Item 3	Yes	Text	20	n/a

Services: Each service has a name and price, for example major service costs €300. Again, the ID is used for identifying the singular key, name and price.

Services				
Attribute Name	Required	Type	Field Length	Default Values
ID	Yes	Number	n/a	Auto increment
Name	Yes	Text	20	n/a
Price	Yes	Number	20	n/a

Items: The ´Items´ entity has an ID, name and price. Since every item has to be added to the service, the total cost is the ´service´ + ´item´ and ´item´ has to have a different price.

Items				
Attribute Name	Required	Type	Field Length	Default Values
ID	Yes	Number	n/a	Auto increment
Name	Yes	Text	20	n/a
Price	Yes	Number	20	n/a

Invoice: The ´Invoice´ entity has an ID, Date and Total because when a user requires an invoice it should display the date and the total of the service + item. And, the user can see all the specs of the service.

Invoice				
Attribute Name	Required	Type	Field Length	Default Values
ID	Yes	Number	n/a	Auto increment
Date	Yes	Date	10	n/a
Total	Yes	Number	20	n/a

Database tables

Entity Relationship Diagram

An entity relationship diagram shows the relationship of entity saved in a database, outlining the flow of information and communication for an accurate system implementation that is detailed in Chapter 4. An entity is an object whereas the properties of the entities are called attributes.

After making the Data requirement, the attribute(s) of every entity is identified. Every attribute is in a circle that links to the entity with a line. A verb inside a rhombus is also used to link between each attribute and demonstrates their relationships. For example, A user has many appointments.

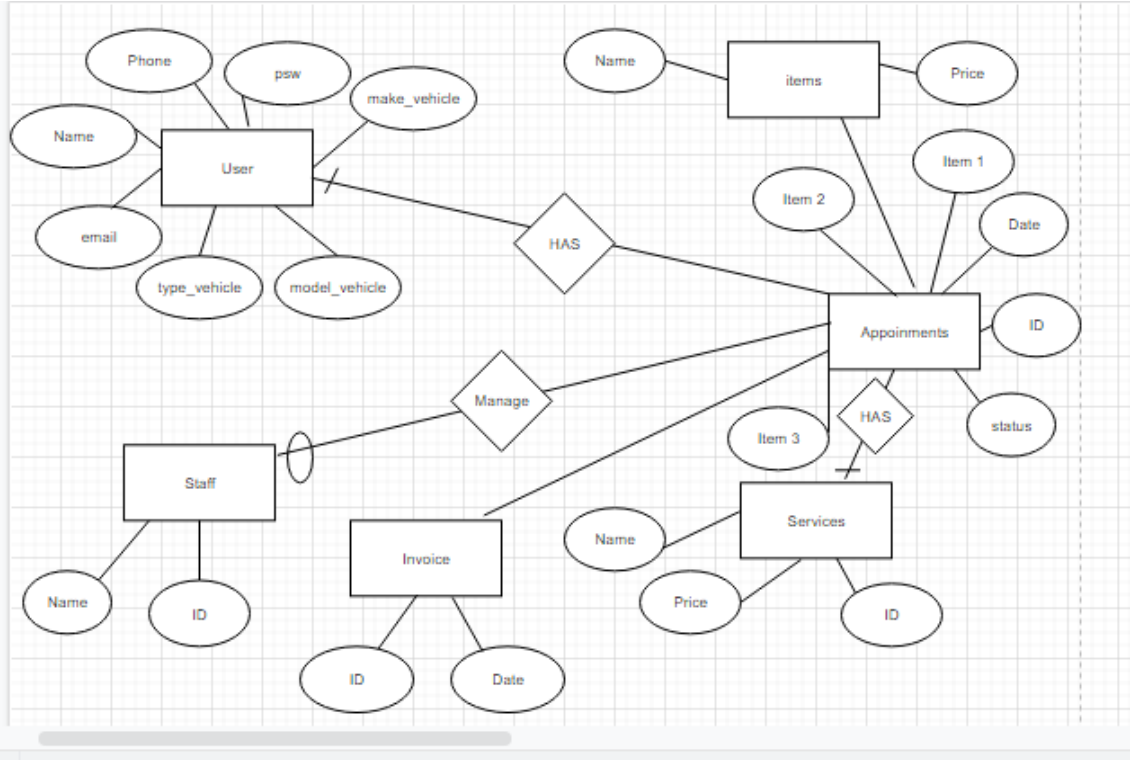


Fig.6 Entity Relationship Diagram

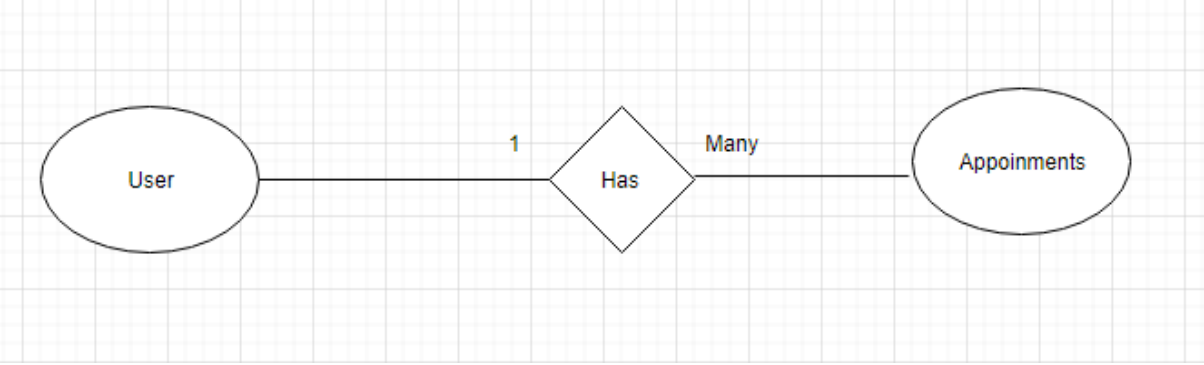


Fig.7 Relationship Attribute: Appointment and User

A user can have many appointments but each appointment only belongs to one particular user.

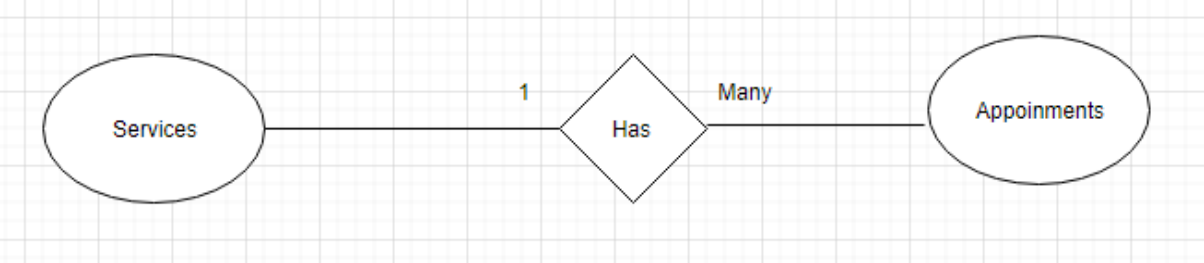


Fig.8 Relationship Attribute: Appointment and Service

A particular service can be selected in many appointments but an appointment can only have one service.

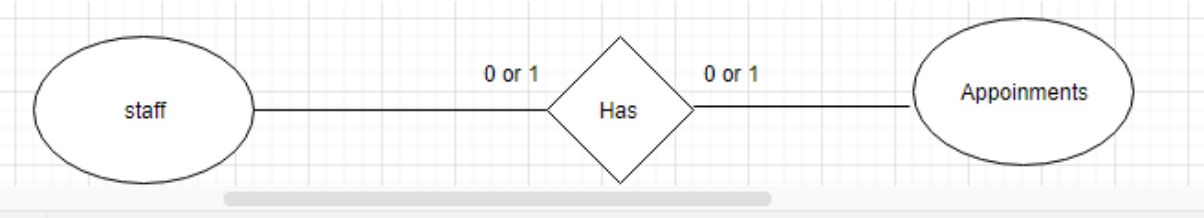


Fig.9 Relationship Attribute: Appointment and Staff

A staff member can have one or no appointment. Likewise, an appointment is assigned by one staff member or not yet.

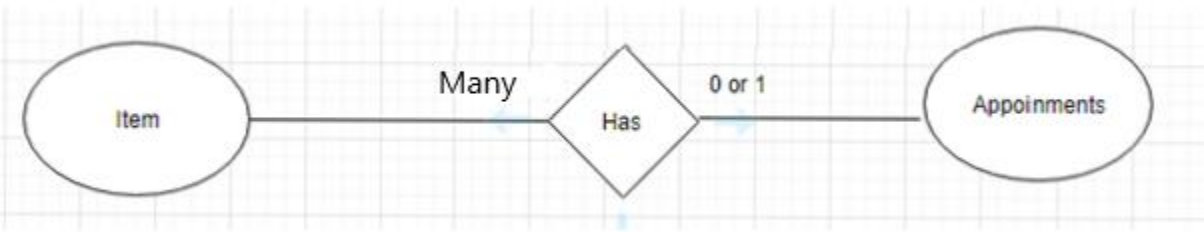


Fig.10 Relationship Attribute: Appointment and Item

An appointment can have one or no item. For example, it is not necessary to use additional parts in some vehicle services. An item can appear in many appointments because there are no appointments sometimes.

Class Diagram

A class diagram shows the content (i.e. attributes) of every entity and the relationship between the entities. Each entity has the primary key and some with at least one foreign key as well.

In this database, every entity has an ID that is the primary key and it carries a unique value while some entities also have foreign keys which are the attributes linked from another entity.

For example, when a user makes an appointment, the system will generate a unique ID Appointment for this request, like ID Appointment 001. The next Appointment ID will be ID Appointment 002 because they are the primary key that set to increase by 1 automatically.

For the system logic, a complete appointment also includes user details so a foreign key, ID User, is used to link from Appointment Entity to User Entity to pull user data like email address, names and vehicle data.

Therefore, it is important to have a foreign key in some of the entities in order to communicate with another entity to get a complete appointment record

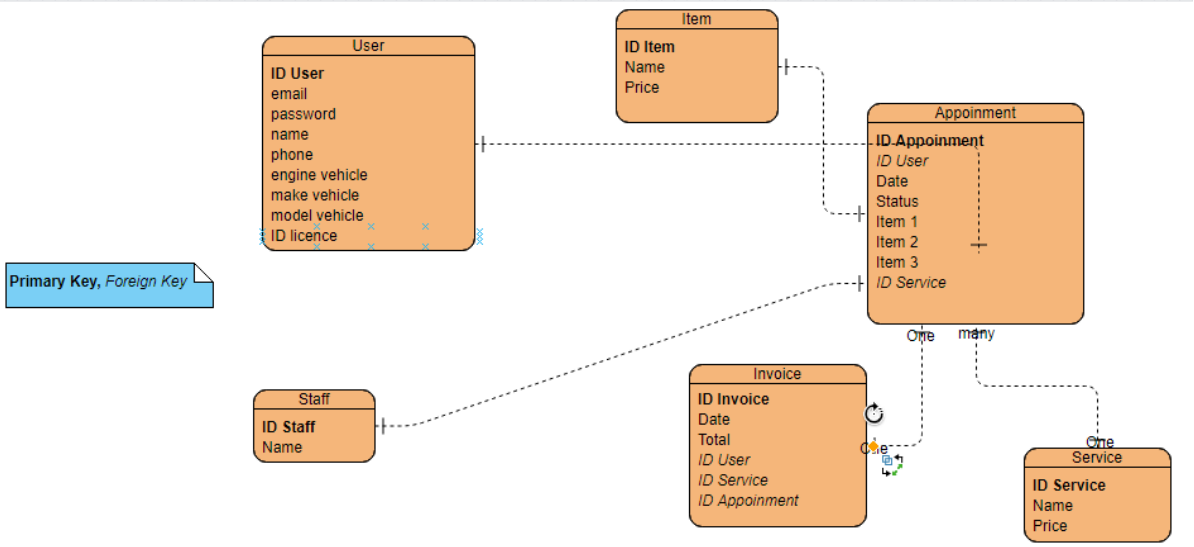


Fig.11 Class Diagram

Use Case

The use case is a technique for knowing what the possible requirements are in order to having a new system or website. Every use case mentions the different stages that the system would have with the user in order to make a task. The aim of a Use Case diagram is to allow developers to analyse the behaviour of the user or administrator on the website.

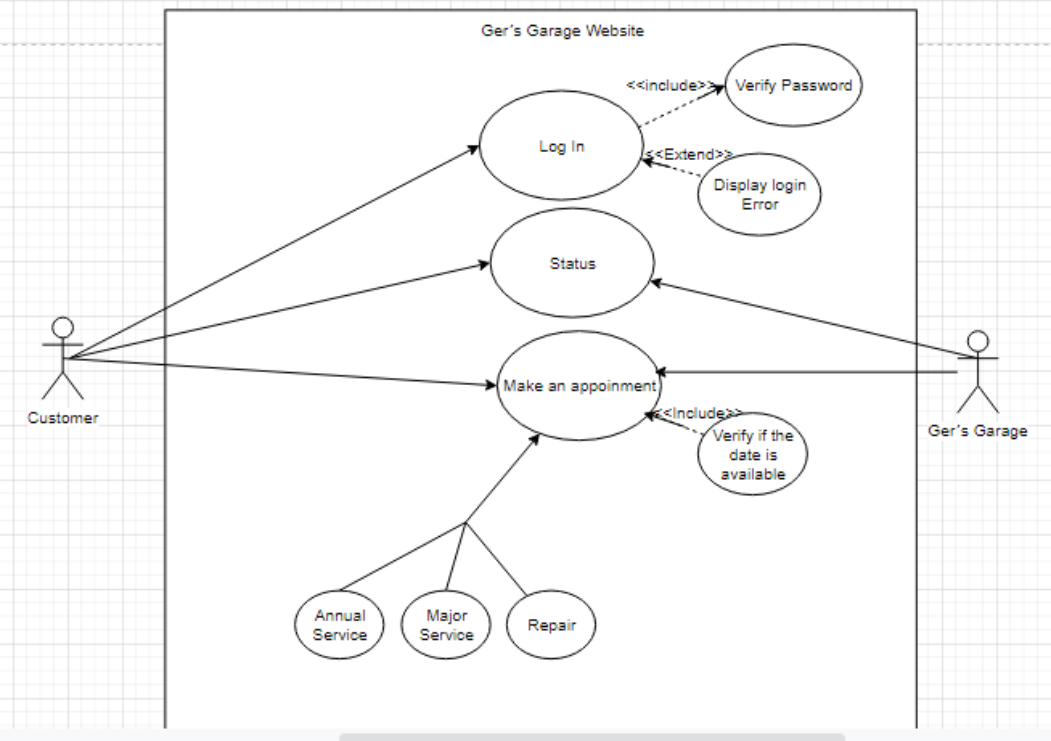


Fig.12 Use Case

When the user logs in, the software verifies the password with the database. If the user enters the wrong email address or password, the system will display an error message and "Exclude" is marked for this scenario.

A user is also able to see the appointment or vehicle service status. For example, when a user makes an appointment, the default value is "booked" but the administrator can change the value to another status such as in Service, Fixed, Collected or Unrepairable_once that the vehicle is in the Garage.

If the database says the appointment status is full, no user is allowed to book a vehicle service, meaning they have to select another date that is available (i.e. not full). In this case, "Include" is marked to indicate the information is verified and proceed to the next option.

The general case for this option is "make an appointment" that has further three choices namely "Annual Service","Major Service" and "Repair", The terms "parent" and "child" can be used.

Chapter 4 Implementation of the System

This chapter details how the project, i.e. working system together with the design, is implemented based on the requirements identified in the previous chapter. Technologies used, the implementation process and potential problems along with solutions are discussed in this section.

During the implementation, the fundamental objectives must not be forgotten – To provide users and administrator with Hassle-free experience and accurate information.

Architecture Considerations

There are some specific functional requirements that should be taken into account for the website architecture. First, users are required to log in or sign up with personal information for booking a service, and the system has to verify their data before proceeding. Therefore, a database management system along with backend programming language is one of the architecture considerations. It aims to store user and vehicle data, and communicate between the frontend system and the database.

Another requirement is for the administrator to quickly make and print the invoice. The ideal format is PDF. Therefore, a software (or a library) called FPDF is used in this project.

Technologies Used

As previously discussed, the programming languages used HTML, CSS and Javascript. They are responsible for frontend design and coded in a framework tool called Bootstrap. Backend programming language is PHP and the Database system is MySQL.

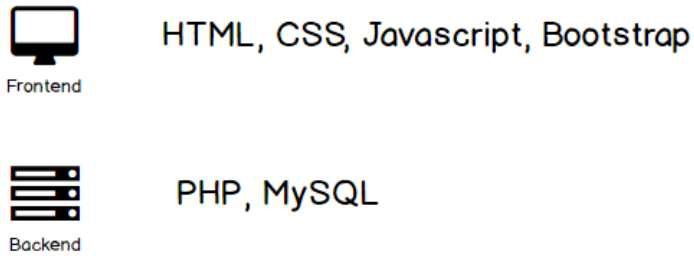


Fig.13 Technologies used for backend and frontend design

Implementation of the System

The implementation part is divided into Frontend, Communication between frontend and backend, and Backend.

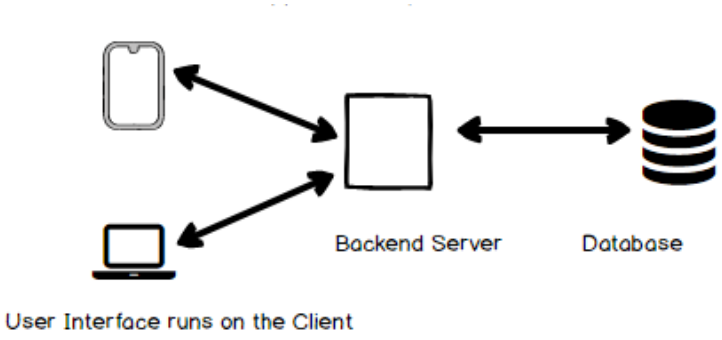


Fig.14 Simple Structure of System

Frontend

Bootstrap - Responsible for the Website Framework

The frontend framework was mainly constructed with Bootstrap template, such as the pages for visitors, users and the administrator. The template used is called “Header Blue” with a blue top menu allowing web developers to change menu names so that Login and SignUp can display as required in the project.

Editing link for each item on the menu is easy to implement in Boostrap template without heading over to coding page, and thus linking to services.php, login.php, and signup.php and so on can be easily done.

The reason for using the particular theme is because it has a simple and easy-to-read layout. Many functions like buttons, images, and textboxes can be pulled from Bootstrap library to save time on webpage design and implementation. The template selected meets user requirement 1 that they can quickly locate signup and login buttons that are isolated at the top right hand corner.

Fig.15 Theme called Header Blue.

On the login page and signup page, relevant textboxes were dragged based on what personal data is required.

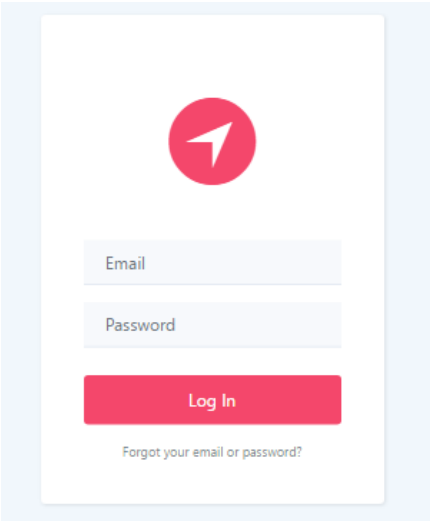


Fig.16 “Login Form Clean” used for making the login page.

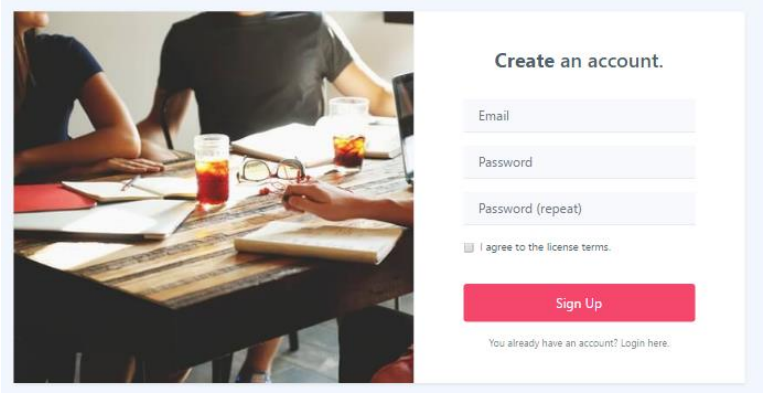


Fig.17 "Registration Form with Photo" used for the registration page.

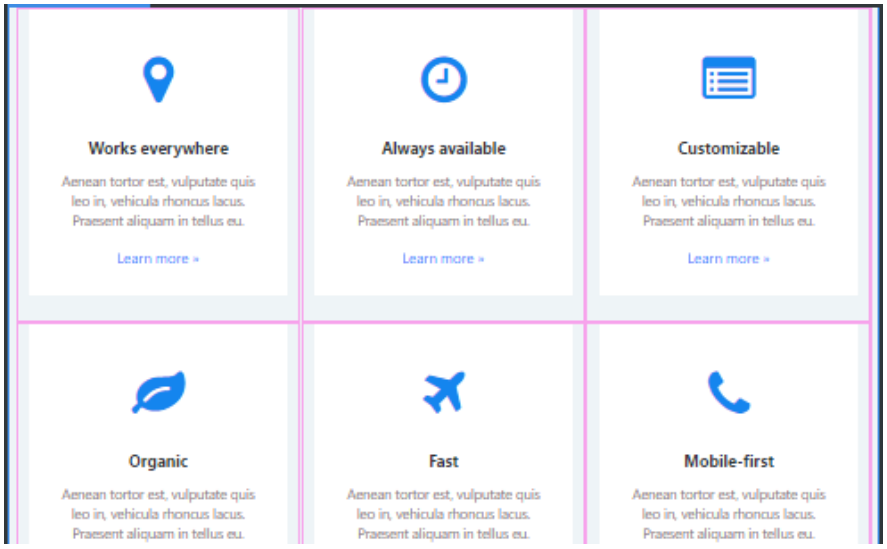


Fig.18 "Featured Boxed" used for featuring Ger Garage Advantages.

The template "SB admin" was used to create the administrator dashboard page. On the left side, there is a drop down menu to display all options required by the administrator such as adding product or service item, assigning tasks to staff members. This dashboard page is simple yet functional and Bootstrap provides basic buttons to facilitate the dashboard page setup. By dragging the elements from bootstrap library to the right webpage position, all these options are quickly well displayed without further coding. However, further coding had to be done to link from the database system to the user interface, in order to satisfy what the users and administrators need. This part is discussed in the Backend section later.

Likewise, the customer dashboard also enable users to browse multiple options e.g. viewing their profiles and selecting the services like major service and annual service.

To conclude the frontend development, all webpages were used bootstrap templates with basic elements like buttons and images to implement. However, some styles needed editing with CSS and HTML.

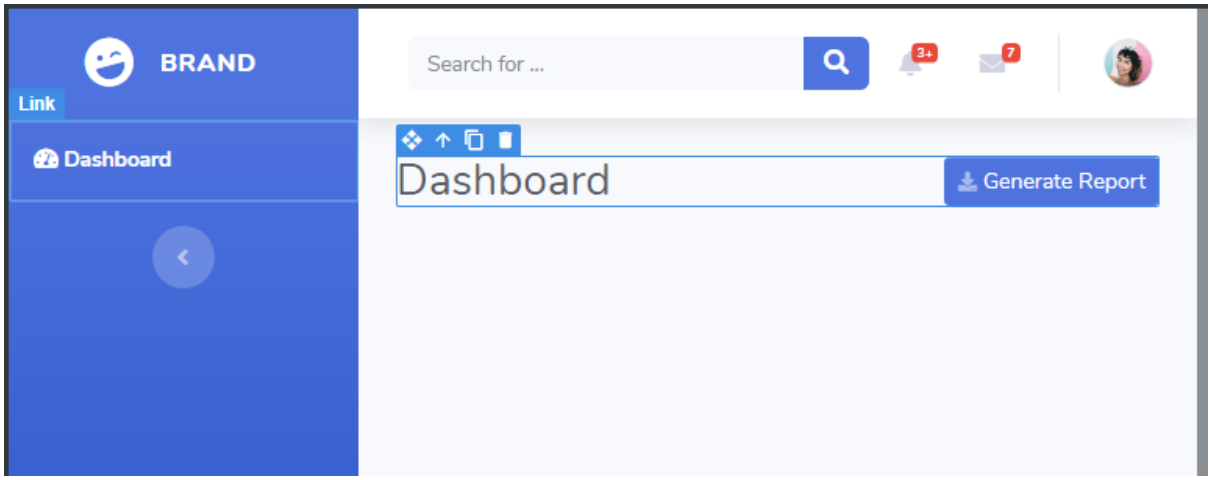


Fig.19 "SB Admin" used for Setting Up the Administrator´ Dashboard

Communication Between Frontend and Backend System

The texts and images below demonstrate how the languages are used to communicate from the frontend to the backend system.

When a user logs in, he or she has to fill personal data into textboxes and a name of each textbox has to be assigned individually, such as ´email´ for the email textbox and ´password´ for the password textbook in fig. 21. It is very important each personal data carries a name that is saved in the PHP file as a variable (fig.22), so as to communicate between backend and frontend. If this implementation section is ignored and mis-handled, the functional requirement laid out in the chapter 3 will not be met, i.e. users fail to log in to make an appointment.

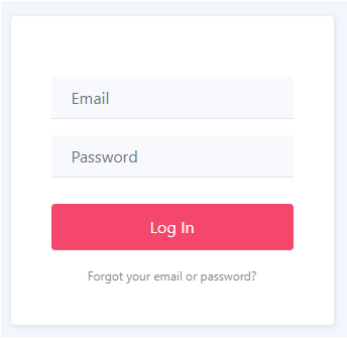


Fig.20 Textboxes on Login Page

```
<div class="login-clean">
    <form method="post">
        <h2 class="sr-only">Login Form</h2>
        <div class="illustration"><i class="icon ion-ios-navigate"></i></div>
        <div class="form-group"><input class="form-control" type="email" name="email" placeholder="Email"></div>
        <div class="form-group"><input class="form-control" type="password" name="password" placeholder="Password"></
        div>
        <div class="form-group"><button class="btn btn-primary btn-block" type="submit">Log In</button></div><a
        class="forgot" href="#">Forgot your email or password?</a></form>
```

Fig.21 Code for Textboxes on Login Page

```php
<?php
    require 'includes/database.php';
    session_start();
    if(!empty($_POST)){
        $email = mysqli_real_escape_string($conexion,$_POST['email']);
        $password = mysqli_real_escape_string($conexion,$_POST['password']);
        $encrypted_password=sha1($password);
        $sql="SELECT ID FROM users WHERE email ='$email' AND passwordU ='$encrypted_password'";
        $result = $conexion->query($sql);
        $rows=$result->num_rows;
        if($rows>0){
            $row=$result->fetch_assoc();
            $_SESSION['id_user']=$row['ID'];
            header("Location:admin/profile.php");
        }else{
            echo"<script> alert('password incorrect');
            window.location='login.php';
                </script>";
        }

    }
```

Fig.22 The user data filled in the textbox is stored as variables in PHP files, such as $email and $password

These variables in PHP contain the value from the textboxes in HTML. For example, $email has the value for the textbox “email”. After a query is made, a mapping process will activate to see if the email and password entered are the same as the database. If the data is found and matched, the user can log in, otherwise, an error message will display. That is an example how the system work from the frontend, backend to the database and vice versa.

Backend

To implement the backend system effectively, XAMPP is used to run and visualize the PHP files without internet, facilitating testing and building the system. It has a dashboard that provide an option for administrating the database and is called PhpMyAdmin.

XAMPP is a free and open source cross-platfform consisting the Apache HTTP Server, MariaDB database and interpreters for scripts written in PHP. This backend technology allows developers to communicate with the database offline, and thus save time on uploading system to the cloud. With this software, developers can focus on building and improving the database that is the heart of the project requirement.

To run this software and visualize the database, a couple of modules need activating such asI Apache, MySQL, Fillezilla and TomCat.

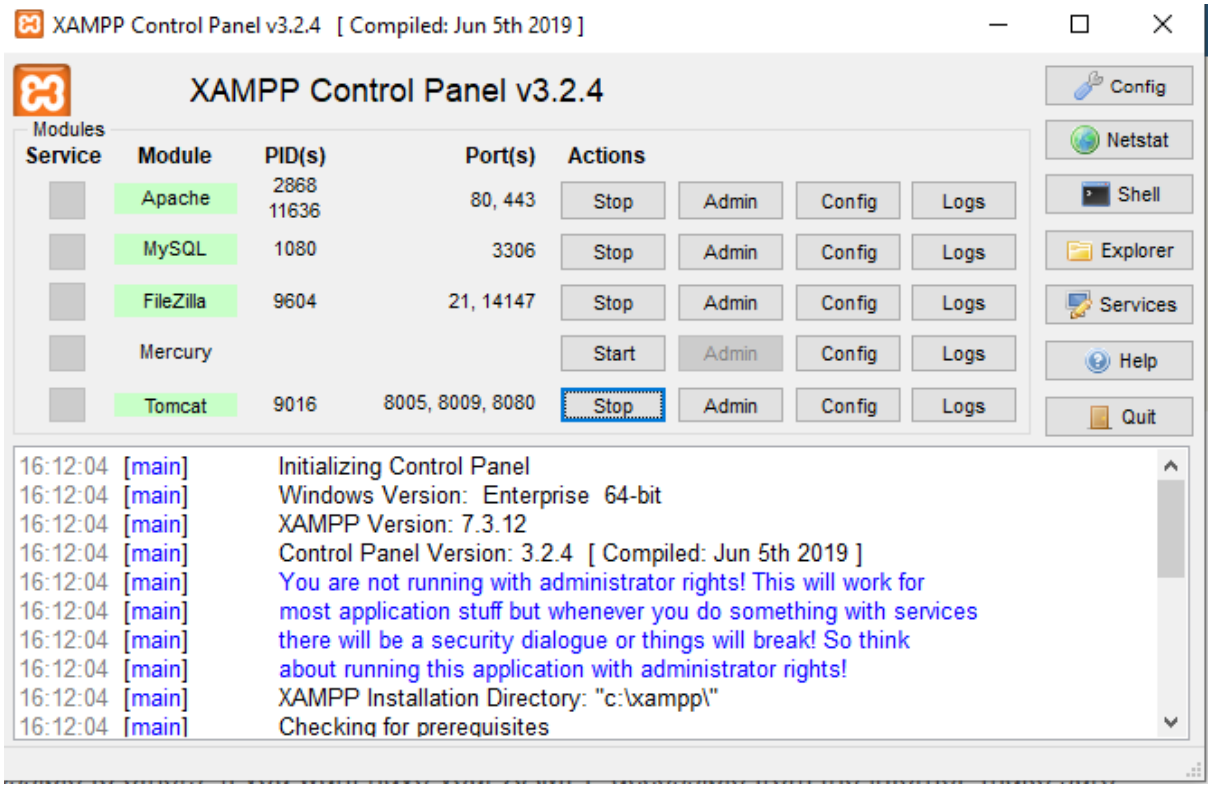


Fig.23 XAMPP Control Panel for Running Database

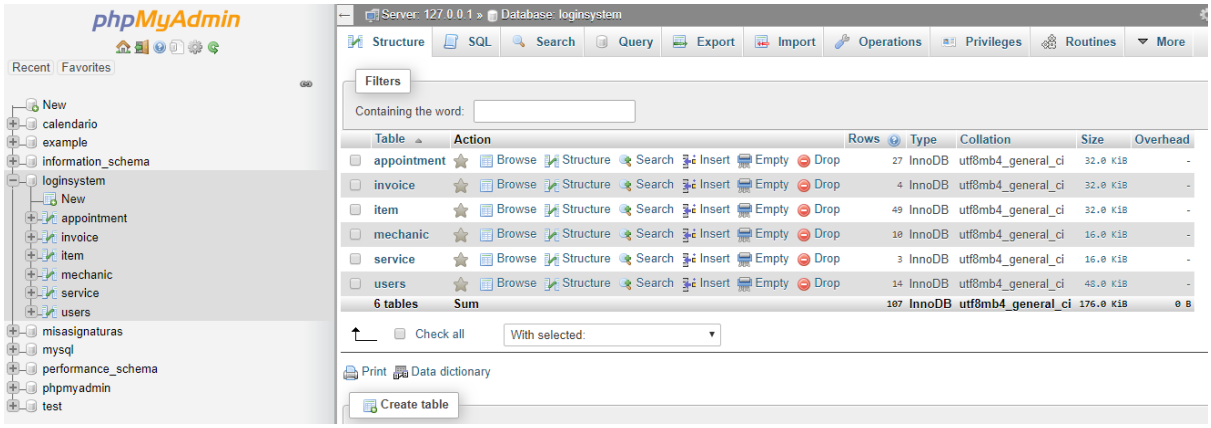
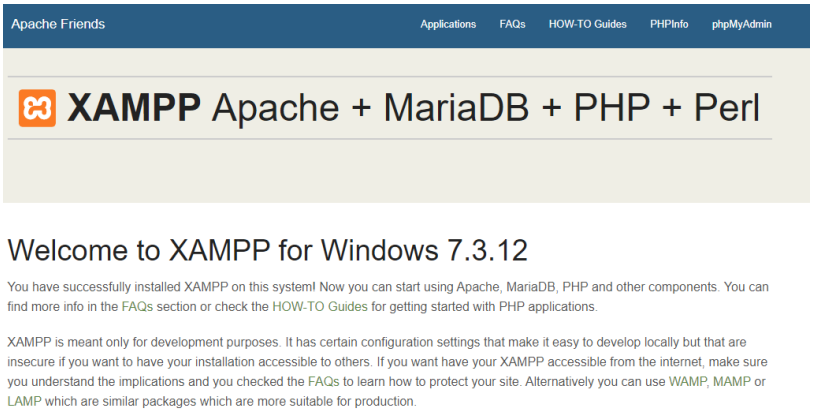


Fig.24 PhpMyAdmin to Organize Databse Tables.



FPDF Library

For invoices, FPDF library is used to quickly generate PDF with the data inserted from the database and this technology works with PHP.

It exists a manual and online tutorial {XX,XX} for knowing how to use the library. Border, background color and character string can be customized. Refer to invoicepdf.php

```
//query to bring the data from the users in order to show the Users data in the invoice pdf
$querytwo = mysqli_query($conexion, "SELECT * FROM users INNER JOIN appointment on appointment.idUser = users.idUser
AND appointment.idAppointment = '".$_GET['idAppointment']."'");

//query to bring the data from the service in order to show the service data in the invoice pdf
$querytree = mysqli_query($conexion, "SELECT * FROM service INNER JOIN appointment on service.idService =
appointment.idService AND idAppointment = '".$_GET['idAppointment']."'");
```

Fig.25 Codes for Making Invoice from Varied Data

https://www.youtube.com/watch?v=lUNwKeRygyI

This line `$pdf ->cell(90 , 5,$usersinfo['nameU'],0,1);` means that the variable pdf prints a rectangle with width: 90, height:5. The text inside of the bracket is the content of the variable $usersinfo[‘nameU’], the number 0 means no border and number 1 means the next position is in the beginning of the next line.

User Manual: How to Use the System?

The system was implemented in the way that users can easily view the status of the appointment with using their mobile phones and computer. If the car is fixed, the administrator can change the status and the user can view the update and collect the vehicle.

Fig.26 First Step: Click the SignUp button

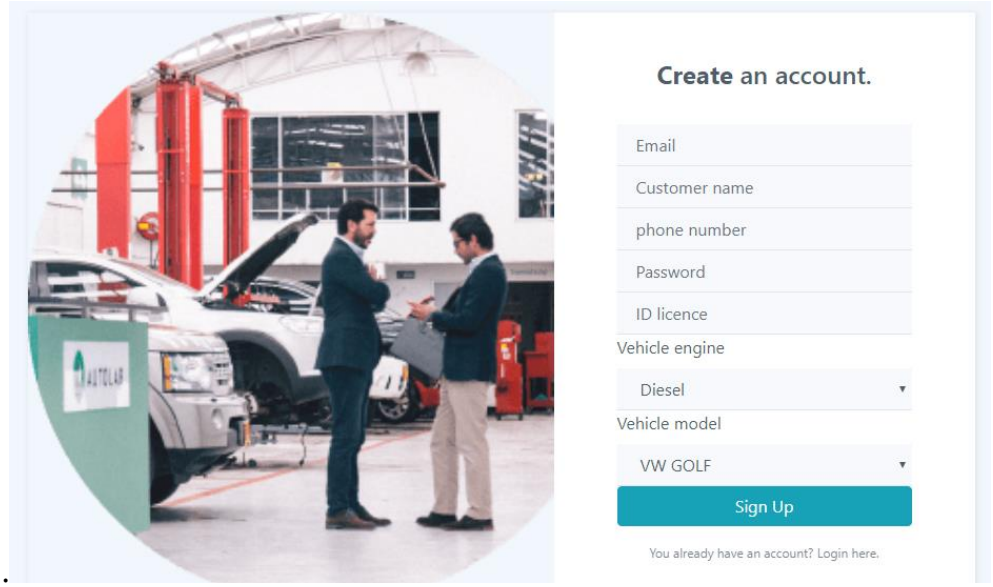


Fig.27 Second Step: Create an account

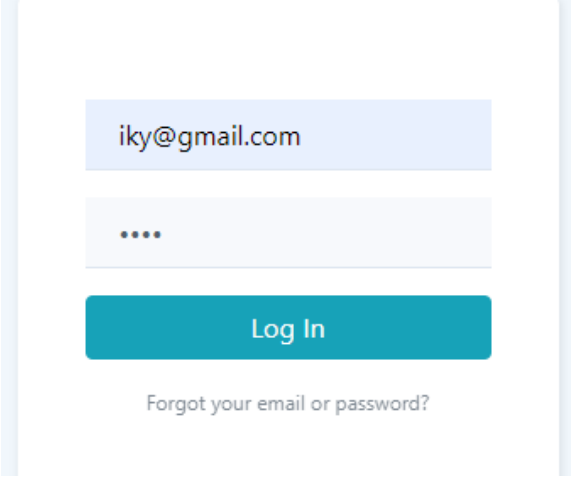


Fig.28 Third Step: Log in to the account

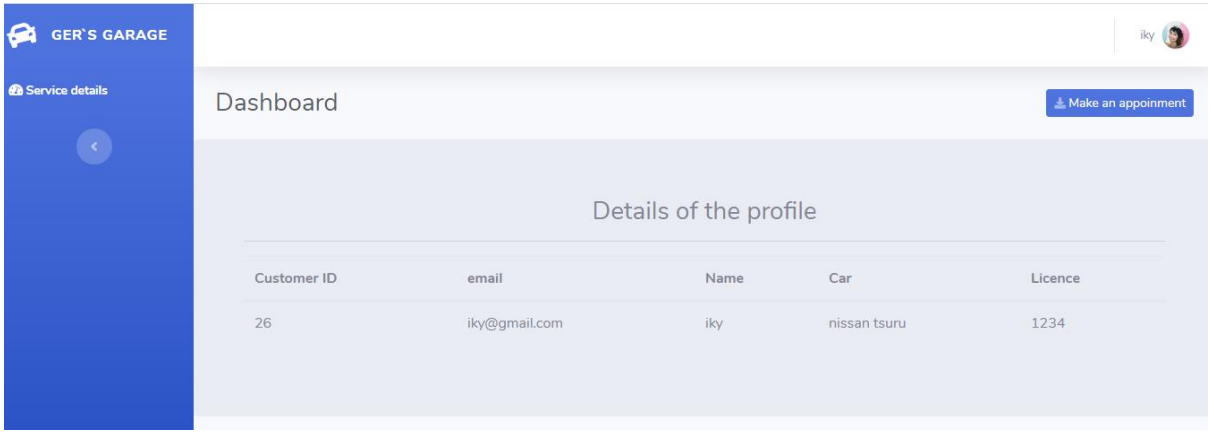


Fig.29 Fourth step: Once the user logs in, he or she can see the details and the button of making an appointment

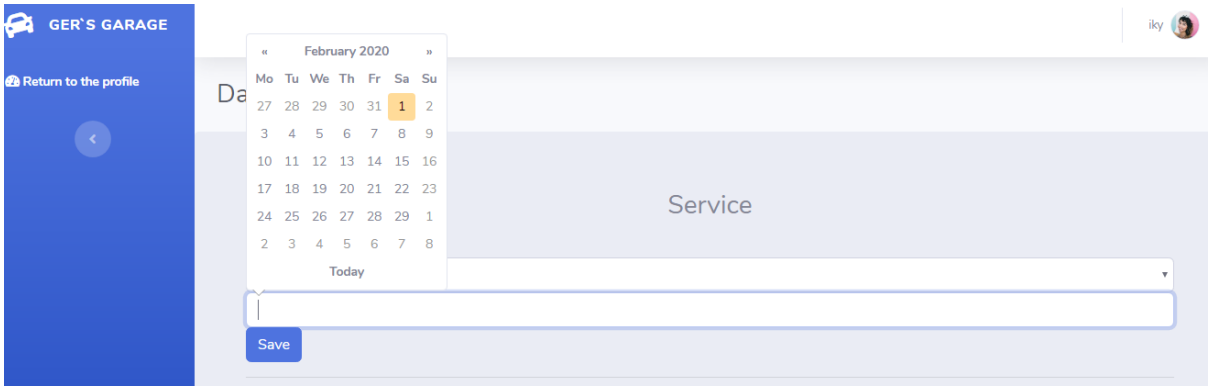
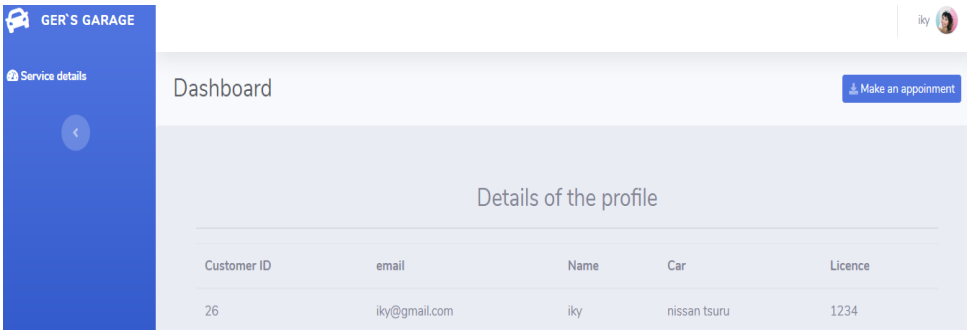


Fig.30 Fifth step: The user has to choose the service desired and the date for the vehicle service.

Once the user clicks the button save, the appointment is successfully made. If the user wants to see the appointment just made, he or she can click the option service details.



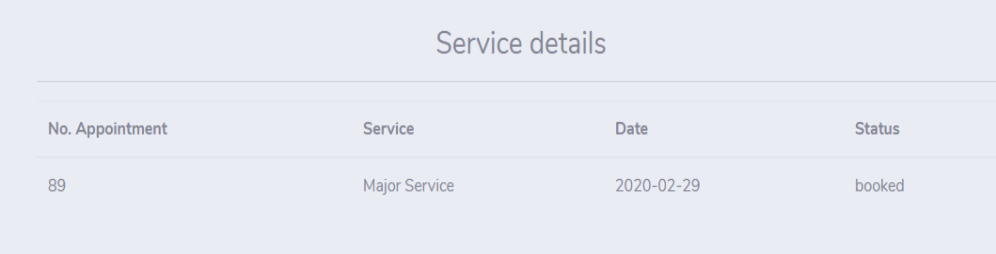


Fig.31 Sixth step: The user can see the details of the profile and service

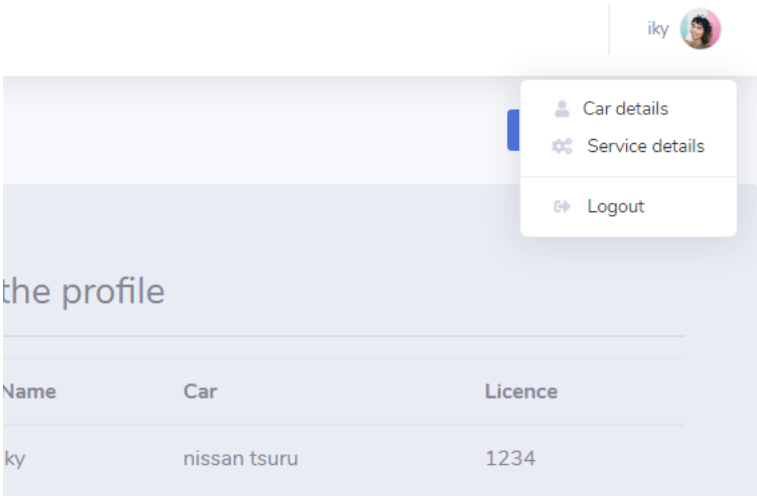


Fig.32 Sixth step: In case of that the user wants to change the car details, there exists the option

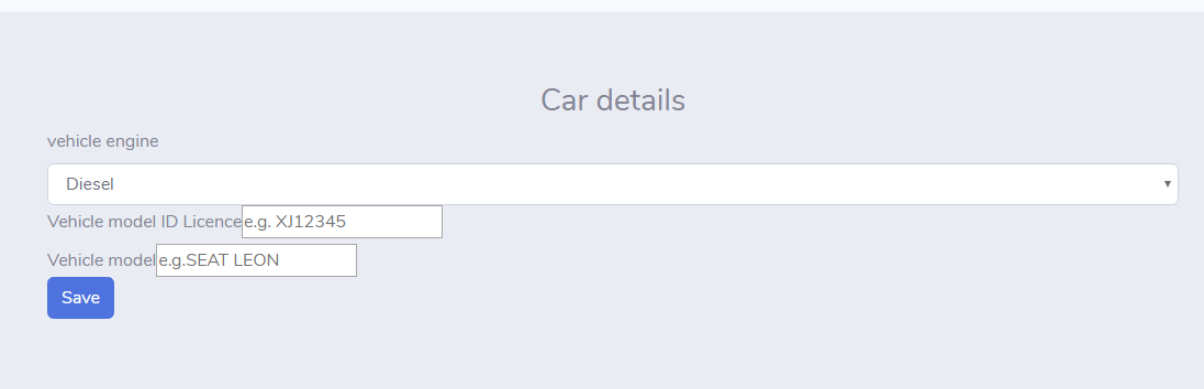


Fig.32 Seventh step: Change the details and hit Save

Administrator Manual: How to Use the System?

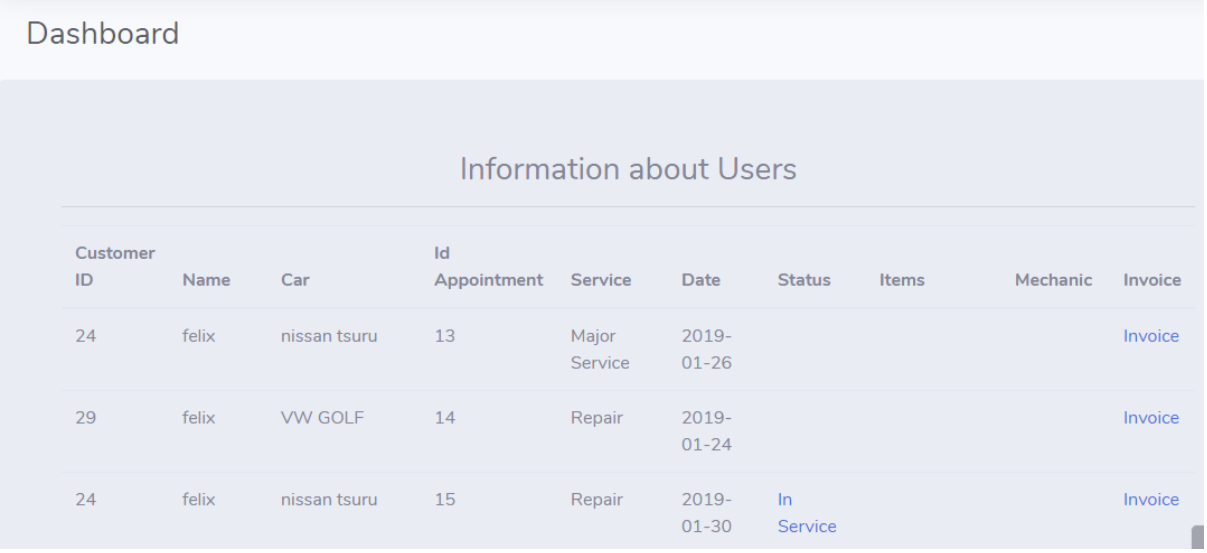


Fig.33 Administrator Dashboard Page

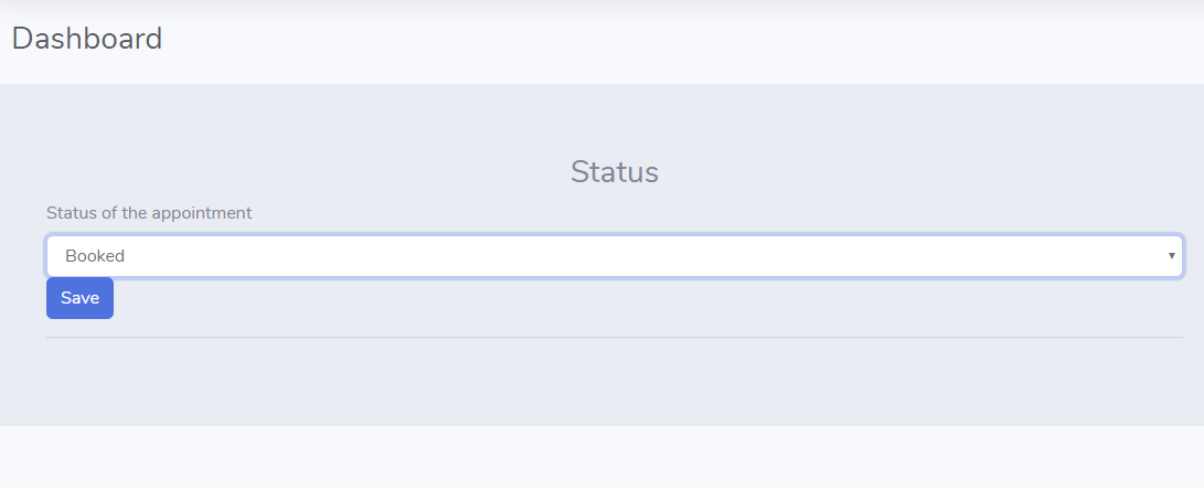


Fig.34 The admin is able to change the status of the service, e.g. booked, in service, collected, fixed or unrepairable

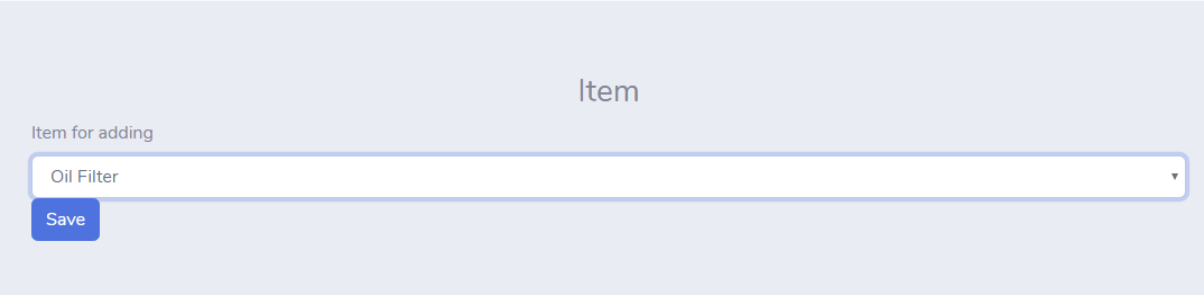


Fig.35 The admin is able to add an item used for fixing the vehicle. Every item has a fixed price in the database.

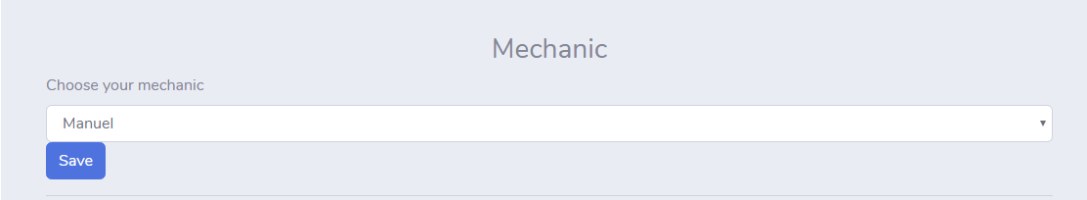


Fig.36 the admin is able to select who is responsible for each service

The administrator is also able to quickly make the accurate invoice for every customer with the aid of the data stored in the database.

Ger`s Garage
Parnel Street
Dublin,Ireland, D07XY45
phone[+12345678]
Fax [+12345678]

INVOICE

Date 1, 2, 2020
Invoice # 26
Customer ID 36

Bill to

mauricio
nissan tsuru
mauricio@gmail.com
123456
xj12345

Description	Taxable	Amount
Repair	-	100
Washer hose valves	-	15
	Total Due	$ 115

Fig.37 the admin is able to make an invoice

Problems encountered

During the implementation, some problems emerged and were difficult to fix. For example, I was struggling with updating the user table when I wanted to modify the vehicle details. I could not replace them with new vehicle details in the query that I did. The query used was called INSERT INTO and the error existed from there. After research, it was solved and the learning is when modifying the data created, it has to be used the command UPDATE users SET that was why I had problems when adding new vehicle data to a specific user.

```
$sqlvehicle = "UPDATE users SET vehicleEngine='$vehicleEngine', vehicleModel = '$vehicleType', licence='$licence'
WHERE idUser = '$iduser'";
```

Fig.38 New code that fixed the problem

I had another error when I tried to sum up the cost of the service. I made it in a query in MySQL and got the result but I found a way easier that was simply to call the price of the service and call the price of each item from the database to the page and make the sum like that:

```
$pdf ->Cell(30, 5,(int)$servicesinfo['price']+(int)$iteminfo['costItem'],1 ,1,'R'); //end of line
```

Fig.39 How to make the code simple

The last error that I had was in the calendar when I was testing the website. I was not given the exact date that I picked from the calendar. Instead, I got 1970-12-01. It took me time to find out the error – the formate of the date. The proper date format of MySQL is with hyphen and starts with the year followed by the month and day for example: yyyy-mm-dd.

```
$date1=mysqli_real_escape_string($conexion,$_POST['date']);
print_r($date1);
$date = str_replace('/', '-', $date1);
$timestamp= strtotime($date);
print_r($timestamp);
$newdate = date('Y-m-d',$timestamp);
```

Fig.40 New code that fixed the date format

Chapter 5 Testing and Evaluation

Functional correctness

The website is able to make the next tasks:

Website

- For the signup, the website has the textbox Customer name, phone, vehicle type & make, vehicle licence, vehicle engine type.

Customer

- The customer is able to make the appointment and choose the type of booking: Annual Service, Major Service and Repair.
- The customer can login as many times and have a history of the last booked.
- Customer is able to select the date of the booking, the website limit the number of bookings, max 4, I have to mention that the major service count double appointment in a day.
- Customer cannot select on Sunday the appointments.

Administrator

- The administrator is able to see the bookings.

- The administrator is able to allocate a mechanic to each vehicle, but it is not able to print the schedule
- The administrator is able to choose the item and it is able to change the price for the item, it has more than 40 items.
- The customer has more than 30 models of car, also it has the option of other brand.
- The administrator is able to change the status: booked, In service, Fixed, Collected, Unrepairable and the customer can see the current status in his own account.
- The administrator is able to make the invoice and print it in a PDF format.

Set of Input and Outputs.

Inputs:

Set of Test scripts

Description	**Expected Result**	**Result**
Customer make the appointment	The admin can change the status of the appointment	The customer is signed up, logged and make the appointment. The admin visualize the appointment of the customer. The admin change the status of the appointment.
Customer make the appointment	The admin can add items to the service	The customer is signed up, logged and make the appointment. The admin visualize the appointment of the customer. The admin can add items to the service
Customer make the appointment	The admin can print the invoice	The customer is signed up, logged and make the appointment. The admin visualize the appointment of the customer. The admin print the invoice with the information of the respective customer.

Chapter 6: Conclusion and Recommendations

During the process of making this project, I realized that there are important points that we have to follow in order to make a good website. Here are the key questions we need to ask ourselves before starting a website project.

- What the customer need to solve: It is necessary to know what the customer require in order to make an analysis of how we can resolve this problem.
- What the best technologies are to address customer's issues
- How each software can communicate with another one (FrontEnd vs Backend).
- Did we analyse different types of database: It is important to make diagrams for knowing how to implement the database.

Most of the requirement for this project were achieved but some of them still have bugs that have to be resolved.

The customer can make the appointment also the user can sign up, login, make the appointment, see the details of the service, see the details of the profile, change the model of the vehicle, One of the bugs in this area is that when I make the appointment I can make the appointment before of the day of today, this is one of the problems that I have to fix, so there still have some issues. For this side of the customer there are some suggestions for further work for example: the customer visualize the price of the service and the price of the items, also probably is important to make an individual chat in order to resolve the doubts of the customer.

The administrator can change the status of every appointment, assign a mechanic, select an item, make the invoice, change the price of the item, but the administrator cannot make the roster, in the website there are not option.

For this part of the administrator, there are some suggestions for example: every function that the administrator has to make it, it would be better to make it in just one page instead of changing from one page to another page in order to assign a value for the item, the mechanic or the status, also when the administrator change the status send an email automatically that the customer can know instantly when the vehicle is ready or was unrepairable.

Ger´s garage project gave me a vision of how really make a website in all the aspects, usually just we see the content from the frontend and we do not think what there are in the back, but the reality is that there are a lot of work in the back that it is necessary in order to make it functional, it is a good experience this project because I learnt how communicate the frontend and the backend.

Reflective Learning

I surprisingly found myself enjoying writing this over-6000-word essay in the past few weeks as I used to read and look into figures without writing that many. The project has driven me to read more and think thoroughly without getting frustrated easily. This self-understanding is very important to influence my decision on my future career, which I used to think money is more important than my passion.

However, I should have started to write the report a week earlier because I struggled much on how to implement the website. After getting stuck on my own for a while, I discussed with my classmates and lecturer who significantly refreshed my stagnant mind and brought me forwards. I should have sought advice from others earlier.

Appendix A: Code Listings

Bootstrap

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
<title>Dashboard - Brand</title>
<link rel="stylesheet" href="assets/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400i,600,600i,700,
700i,800,800i,900,900i">
<link rel="stylesheet" href="assets/fonts/fontawesome-all.min.css">
```

Jquery

```
</footer>
</div><a class="border rounded d-inline scroll-to-top" href="#page-top"><i class="fas fa-angle-up"></i></a></div>
<script src="assets/js/jquery.min.js"></script>
<script src="assets/bootstrap/js/bootstrap.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.4.1/jquery.easing.js"></script>
<script src="assets/js/theme.js"></script>
```

Gantt Chart

TASK NAME	START DATE	END DATE	DESCRIPTION
DESIGN OF THE WEBSITE (WEBSITE)	02/12/2019	08/12/2019	Design how it will show the website, how many pages.
CODING FRONTEND (WEBSITE)	09/12/2019	22/12/2019	coding using HTML, CSS, Javascript.
CODING BACKEND	16/12/2001	29/12/2019	coding using Java with Springboot
BUILDING DATA BASE	30/12/2019	12/01/2020	Building a database in MySQL.
PUT EVERYTHING TOGETHER	06/01/2020	19/01/2020	putting everything together and how it works the technologies
WRITING TECHNOLOGIES INVOLVED	20/01/2020	26/01/2020	Writing the technologies involved in the frontend, backend and database
WRITING IMPLEMENTATION OF THE SYSTEM	20/01/2020	26/01/2020	Writing the reason of using these technologies and the advantages
WRITING PROBLEMS ENCOUNTERED	20/01/2020	26/01/2020	Writing some problems that it could not resolve it
TESTING THE WEBSITE	27/01/2020	31/01/2020	Testing the website.
WRITING CONCLUSION	27/01/2020	31/01/2020	Submiting the project.

Code for the Signup

```php
if(!empty($_POST)){
    $name= mysqli_real_escape_string($conexion,$_POST['name']);
    $email=mysqli_real_escape_string($conexion,$_POST['email']);
    $phone=mysqli_real_escape_string($conexion,$_POST['phone']);
    $password=mysqli_real_escape_string($conexion,$_POST['password']);
    $licence=mysqli_real_escape_string($conexion,$_POST['licence']);
    $engine=mysqli_real_escape_string($conexion,$_POST['vehicle_engine']);
    $model=mysqli_real_escape_string($conexion,$_POST['vehicle_model']);
    $encrypted_password=sha1($password);

    $sqlemail="SELECT idUser FROM users where email ='$email'";
    $resultemail=$conexion->query($sqlemail);
    $row=$resultemail->num_rows;
    // if it find the value in email from the database, it wont be registered
    if($row>0){
        echo"<script> alert('the user already exist!');
        window.location='signup.php';
        </script>";
    }else{// in case that there are not data with the same information, it will be registered
        $sqluser = "INSERT INTO users(email, nameU, phone, passwordU,licence,vehicleEngine,vehicleModel) VALUES
        ('$email','$name','$phone','$encrypted_password','$licence','$engine','$model')";
        $resultuser=$conexion->query($sqluser);
        print_r($resultuser);
            if($resultuser>0){
                echo"<script> alert('successfully registered');
            window.location='index.php';
            </script>";
                }   else{
                echo"<script> alert('Error!');
```

Every value of each textbox is saved it in a variable in PHP, the next step is making a query to the database in order to verify if it exist these values in the database, if it does not exist, it makes another query for inserting the data in each attribute of the entity, in the image the entity is “users”.

Code for making the appointment

```php
if(!empty($_POST)){

    $valueService;
    if($_POST['service']=='Annual Service'){
       $service =1;
    }
    else if($_POST['service']=='Major Service'){
        $service =2;
    }else{
        $service =3;
    }



    // $service= mysqli_real_escape_string($conexion,$_POST['service']);
    $date1=mysqli_real_escape_string($conexion,$_POST['date']);
    print_r($date1);
   $date = str_replace('/', '-', $date1);
    $timestamp= strtotime($date);
    print_r($timestamp);
    $newdate = date('Y-m-d',$timestamp);
```

For making an appointment the customer has to enter to the page makeapp.php and he will be visualized the services like major service, annual service or repair, if the customer choose Annual service, this value in the backend (PHP) will change to 1 as in the image above.

Also in this image it saves the value of the date and it pass for some functions in PHP in order to make it ready with the correct format for MySQL.

```php
//Update the vehicle data and send to the database
$sqlservice = "INSERT INTO appointment(idService,date,idUser,status) VALUES($service,'$newdate',$iduser,
'booked')";
print_r($sqlservice);
$resultservice=$conexion->query($sqlservice);


    if($resultservice>0){
        echo"<script>  window.location='profile.php';
    </script>";
        }   else{
        echo"<script> alert('Error!');

        </script>";

    }

}
```

The next step is with these two values (date and service), it has to make a query in order to insert in the appointment table with the values.

List of References

Michalis Diamantaris (2017) *Mobile Websites vs Mobile Apps*, Republic and Canton of Geneva Switzerland: International World Wide Web Conferences Steering Committee.
Srivastav, Manoj (2016) 'WEB CONTENT MANAGEMENT SYSTEM', *International Journal of Innovative Research in Advanced Engineering (IJIRAE),* 3(3), pp. 51-61.
Waseem I. Bader (2016) 'Responsive Web Design Techniques', *International Journal of Computer Application,* 150(2), pp. 18-27.
William Jobe (2013) 'Native Apps Vs. Mobile Web Apps', *International Journal of Interactive Mobile Technologies (iJIM),* 7(4), pp. 27-32.