ABSTRACT

Based on Ger’s requirements, where he runs a small garage and carries out maintenance checks and he is supported by few staff. It led to Garage Web System. This application is to help Ger to manage his garage, fixes, bookings and employees. Ger used to have manual bookings, where the clients would come or call to the garage and booking their services.

The new application has been being developed to support all stakeholders. Also, the web system helps Ger to manage his garage, clients to book their appointments and the staff with the services. Moreover, the application offers login for staff and customer, staff and customer register, booking, invoice and stock.

CHAPTER 1:

INTRODUCTION

PROJECT CONTEXT

Ger manages a small team of 4 mechanics. All those employees are able to maintain different types of vehicle and perform any service or repair. In some cases, maintenance check will require parts or supplies (e.g. fluids, oil, tyres) which the staff will look for in the stock of supplies from the garage and invoice the customer for.

It is a small garage which carries out any type of services and repairs on cars, motorbikes, small vans and small buses. However, due to the high demand of bookings, Ger faced, back in the past, some issues which is going to be solved with the new application. The Web System allows customers and staff to login and register their details and customer can make appointment to their vehicles. The staff are able to view all booking assigned then for the current day. Ger will also be able to manage and view all bookings and register new staff.

This web system is designed to assign automatically a booking to staff after customer making an appointment. Besides, the application allows Ger to print the schedule for any particular date, allocate costs to each booking or a basic fixed cost.

Taking into consideration the above scenario, the main services that need to be provided are:

Customer creates account and book an appointment. After signing up and registering the customer can visualise the last booking. Also the customer will be able to update details:

User and password, contact details and vehicle details

The staff will be able to check the schedule for the day, update booking status and complete information about services and prices.

Administrator (Ger) will be allowed to create new staff, check and update bookings and stock control.

Once registered, the customer can log-in and book a slot.

When booking the customer need to provide some extra information related to appointment day, vehicle details and issue description, type of service and give out details about the vehicle issue. While booking the customer can’t make an appointment on Sunday and also there a limit of services per day.

The mechanical needs to update booking status (Booked, In Service, Fixed / Completed, Collected and Unrepairable/Scrapped).

Once the booking has been changed to Completed the application sends the invoice by email to the customer or print on the screen.

Booking daily limit: The garage currently works with 4 staff which can carries out any kind of service. Each employee works at maximum on 4 vehicles per day. By consequence, the garage supports 16 services per day. Total amount of services per day is the math between total of staff multiplied by limit of staff services per day.

WHY IS A GOOD PROJECT?

Developing a web-based application makes us go beyond the concepts we have learnt. Also, it gives us the opportunity to integrate all technologies we have studied during the course and and others stacks which have been searched to implementing this Ger Garage Web System.

The application has been built to serve small garages but can also attend to others businesses which the work flow is to offer services (e.g hairdresser, computer maintenance, health consulting and others). In additional, as it is web service based, new modules can be build and bigger garage business will be able as well.

The system is a web service based architecture. It makes easier create new functionalities and integrate to others web application. Also, troubleshooting and fixes become simpler because it’s possible isolate the problem. New modules can be added too (e.g. payments and e-invoice) with minimum effect, since they are functionalities apart from each other. A plus, the front-end is built using ionic calling the endpoints. In other words, to change the View layer is just create a new one and connect to the same endpoints, without affecting the back-end.

Main goal:

Create a web-based application where customers can book their vehicles to a garage service.

Objectives:

Research of technologies and review previous concepts;

Define architecture and design diagrams;

DB connection;

Map JPA entities;

Connect the Front-end with Back-end;

Final report;

Areas to cover

Regarding to this web-based application see below technologies and methodologies list.

* Java 8, object oriented language;
* Java Persistence API, as Java software component to map the database using annotations, and manage queries on the DB;
* Spring Boot as a Java framework in order to create web services.
* Tomcat as HTTP web server provider.
* Gradle is an open-source build automation system based on Apache Ant and Apache Maven.
* Postman is a tool used to test RESTful APIs.
* Ionic Framework for building desktop apps using web technologies integrated to Angular.

In terms of the project itself, the following areas will be covered:

* SCRUM as Project Management Agile Methodology.

Project planning techniques such as Gantt Charts;

Personal challenges

It project development intends to address the following challenges:

Front-end development

PROJECT PLAN:

Methodology:

Due to the complexity of working with and integrating a variety of stack technologies and also considering the size of the project, it has been managed using Scrum framework which is lightweight, simple to understand but difficult to master, according to (Schwaber and Sutherland, 2017).

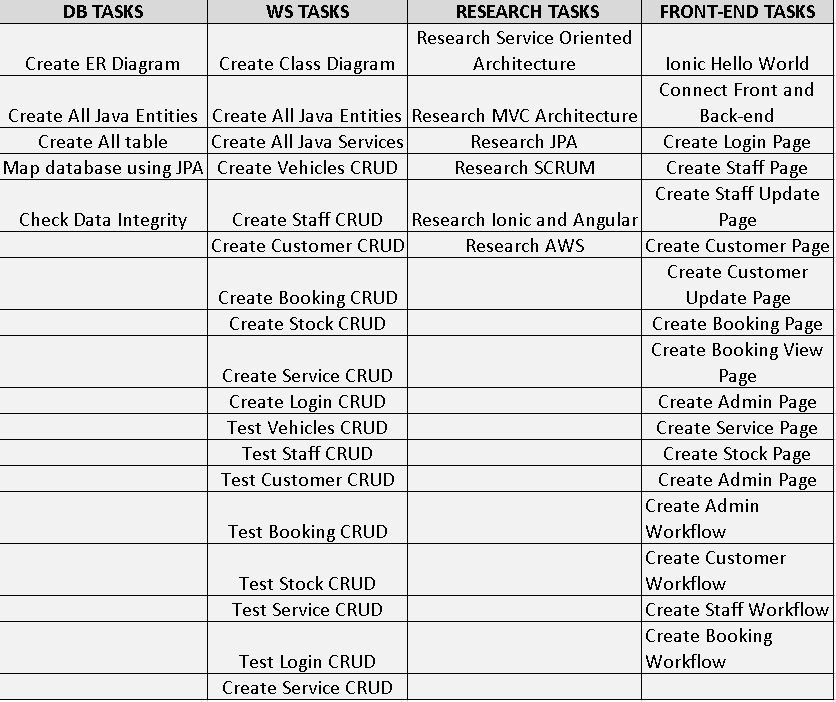
Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. It is for developing, delivering, and sustaining complex products and its definition consists of Scrum’s roles, events, artifacts, and the rules that bind them together (Schwaber and Sutherland, 2017).

There are three types of team stakeholders. Product Owner (PO), Scrum Master and Developer. As the Scrum creators stated the complexity of mastering it, this project has been managed by Product Owner and Developer role.

The PO role has produced the followings artifacts:

* Product Backlog items;
* Ordering Product Backlog to best achieve goals;

See below list of product backlog items:

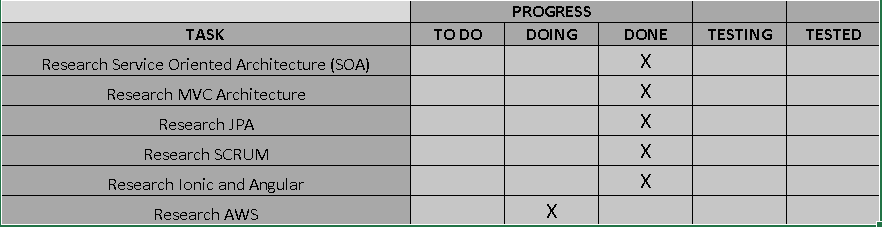


Product Backlog Item

Regarding to the Development Team, it is the role which the stakeholder s work of delivering a potentially releasable Increment of “Done” product at the end of each Sprint (Schwaber and Sutherland, 2017).

At part of the project, the developers has been supported for a Kanban board. According to (Scrum, 2020): “Kanban is a strategy for optimizing flow. The practices in the Kanban Guide for Scrum Teams help enhance and complement the Scrum framework and its implementation.”

See below list of activities on the Kanban board which the development team has used to follow and release the product backlog.



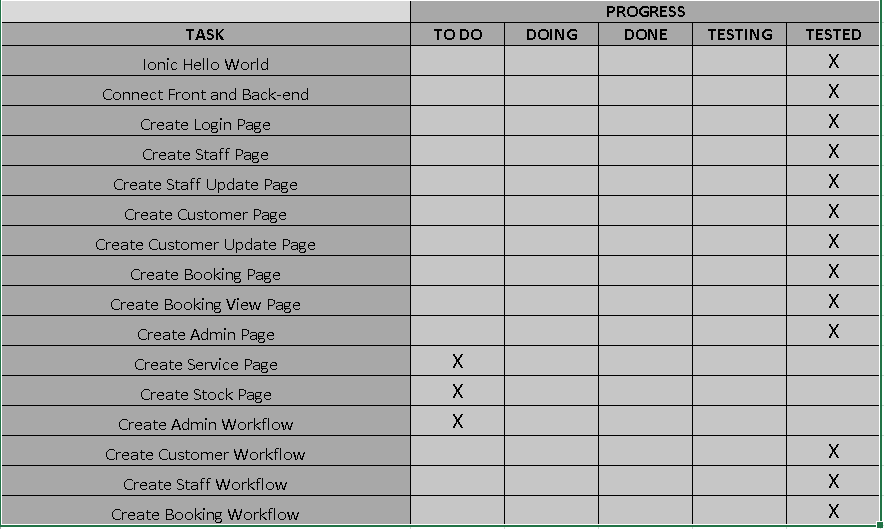
Kanban Board: Research Tasks



Kanban Board: Database Tasks



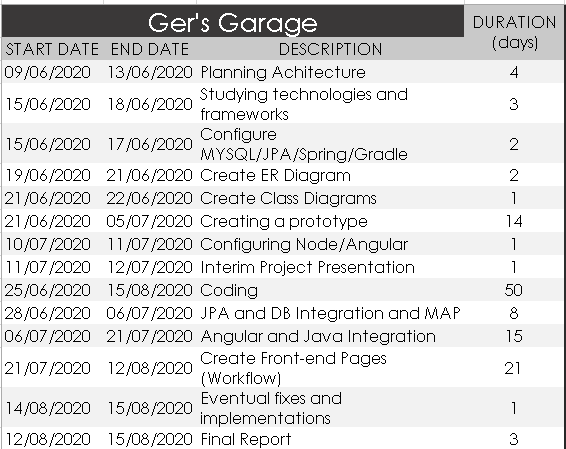
Kanban Board: Back-end Tasks



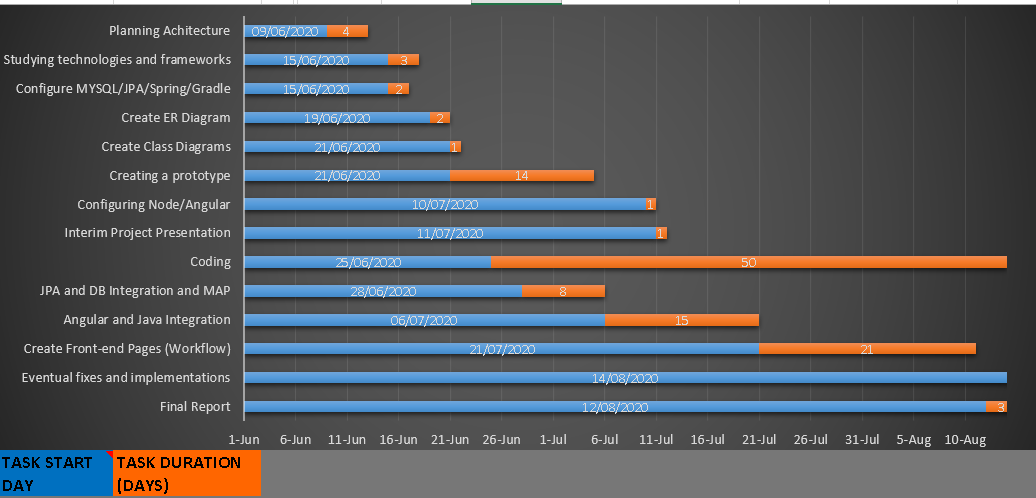
Kanban Board: Front-end Tasks

A Gantt Chart helped at the planning stage to outline and illustrate project schedule. It gives an overview of a project, tasks and deadline as well.

Check below:



Gantt Chart: Timetable contents



Gantt Chart: Bar Graph

In additional, this system has utilised sprint of 1 week and use the Weekly Reflective Journal as a Sprint Review and Sprint Retrospective, which has given the possibility of revising Product Backlog and the Sprint Retrospective to create a plan for improvements.

CHAPTER II:

LITERATURE REVIEW

The second chapter presents the academic research reviewed and used on this web application.

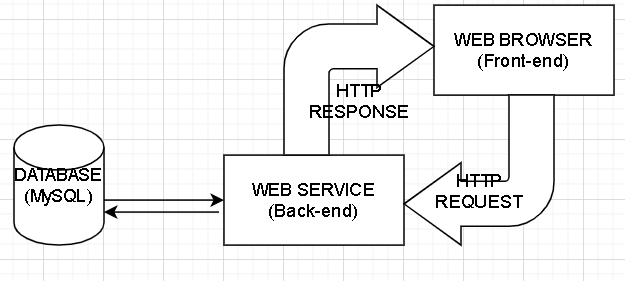
To achieve the project objectives, the following studies areas has been researched:

* REST Architecture;
* Ionic / Angular;

Although other topics and areas has been studied, those are the most challenging and which required a considerable research time to complete this project.

Representational State Transfer (REST)

REST is an architectural style which defines constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability and modifiability that enable services to work best on the Web (Oracle, 2013).



REST Architecture

Each service in this architecture keeps its own code and are able to execute a complete functionality. Taking the Customer workflow as a reference, it is possible visualise functionalities apart from each other. Through HTTP Request Methods (GET, PUT, POST and DELETE) and using JSON. For instance, customer register is a functionality where the front-end (Browser) ask to back-end (Web Service) the method to register and it has been exposed through an endpoint.

IONIC

Ionic is an open source, cross-platform framework used to develop web applications and hybrid mobile apps.

Ionic is a framework which helps to the frontend UX and UI interaction of an app, UI controls, interactions, gestures and animations. It is easy to integrate with other libraries or frameworks, such as Angular or React. Alternatively, it can be used standalone without any frontend framework using a simple script include (Ionic, 2020).

By integrating this framework with Angular libraries, using the command line (@ionic/angular), gives all Angular core components, what improves the development process.

CHAPTER III:

SYSTEM ANALYSIS & DESIGN

This web system is designed to meet Ger’s mechanical garage needs. It is a small garage which carries out any type of services and repairs on cars, motorbikes, small vans and small buses.

Ger manages a small team of 4 mechanics. All those employees are able to maintain different types of vehicle and perform any service or repair. In some cases, maintenance check will require parts or supplies (e.g. fluids, oil, tyres) which the staff will look for in the stock of supplies from the garage and invoice the customer for it.

The third chapter will describe the application functional requirements and functional design. Also, it will show the diagrams that support either back-end and front-end.

FUNCTIONAL REQUIRENMENTS

The application implies functional requirements for administrators, customers and mechanicals. Moreover, pre-set up is required.

PREREQUISITES

* At least one administrator and one staff need to set up;

MAIN REQUIREMENTS

* The customer must sign up on the system towards booking their vehicles.
* After signing up, the customer must make an appointment for their vehicle service.
* The staff actor need to update the booking status. Also, add description of extra service, use supplies from stock and charge for it if necessary.
* Either Staff or Administrator must print the invoice to the customer.

Requirements per subsystem:

ACCESS TO THE SYSTEM

Customer Register

The customer must access the web system and create an account in order to be able to make an appointment to their vehicles. By going through the register page the customer will find a form where is required some details which he needs to fill up.

Required information on the form:

* Username and login;
* First and last name;
* Email, phone number and PPSN;
* Vehicles details;
  + Type;
  + Maker;
  + Model;
  + Fuel;
  + Manufacture;
  + Licence number;

Customer Log-in

To access the application the customer needs to provide the following:

* Username;
* Password;

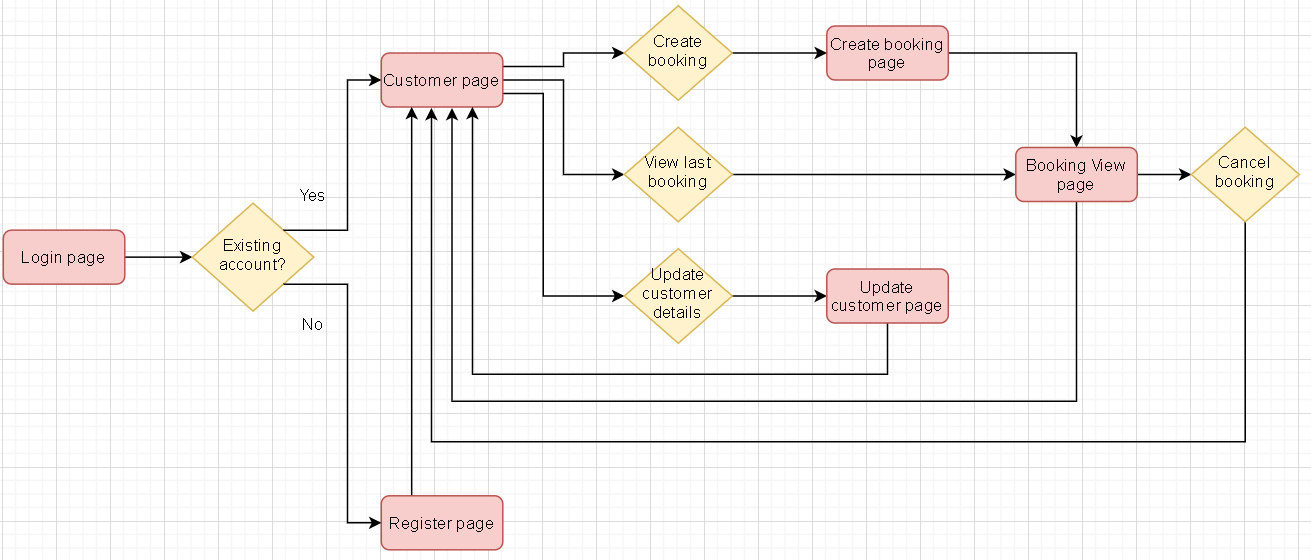
Customer Booking

In order to book their vehicles the customer must go to the booking page and inform some details. See below:

* Booking date (Appointment);
* Type of Service: Annual Service, Major Service, Repair / Fault and Major Repair);
* Report the issue details;
* The system will automatically load the customer and vehicle details.

Customer Booking View

By accessing this page, the customer can visualise the last booking with the given details, plus the booking status and the mechanical responsible for the vehicle service.



Customer Workflow

Staff Register

The staff will be created by the administrator with no need of going to the register page such as on the customer workflow. Towards creating an account the staff is required some details which he needs to inform to the administrator.

Required information on the form:

* Username and login;
* First and last name;
* PPSN;

Staff Log-in

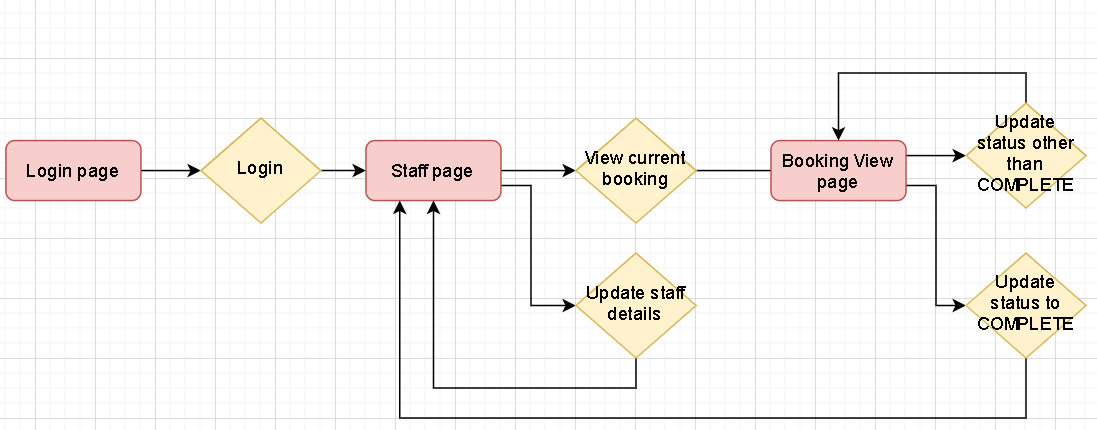
To access the application the employee needs to provide the following:

* Username;
* Password;

Staff Booking View

On the booking page the staff must update booking status for the services he is responsible for. In case the service implies to use stock supplies, the staff can add it on the notes and also add the value description for the extra service: See below:

* Booking Status (Mandatory): Booked, In Service, Fixed, Collected and Unrepairable/Scrapped;
* Description Extra Service (Optional)
* Extra Service Price (Optional);



Staff Workflow

Administrator Register

As described on the prerequisites section, the web system admin will be set up at the very beginning through a database dump or a simple DB insert.

In relation to an admin account, the following details will be required:

Required information on the form:

* Username and login;
* First and last name;
* PPSN;

Administrator Log-in

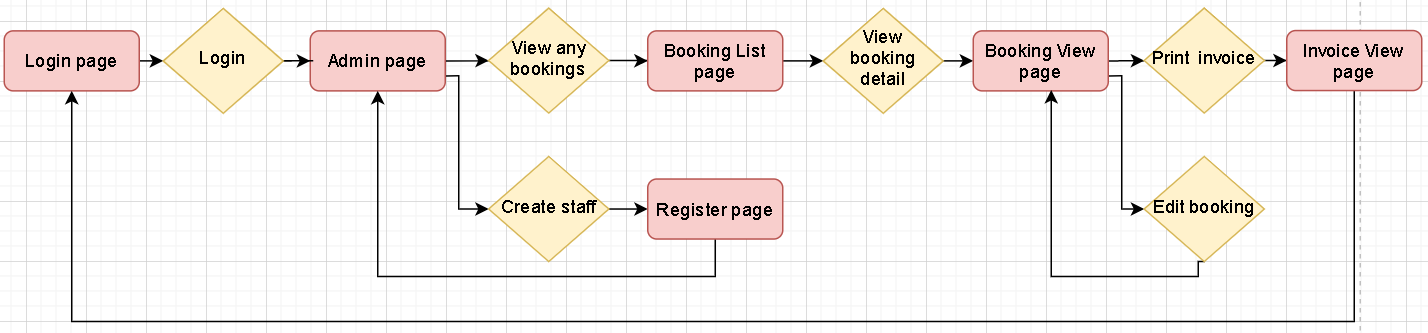
To access the application the admin role needs to provide the following:

* Username;
* Password;

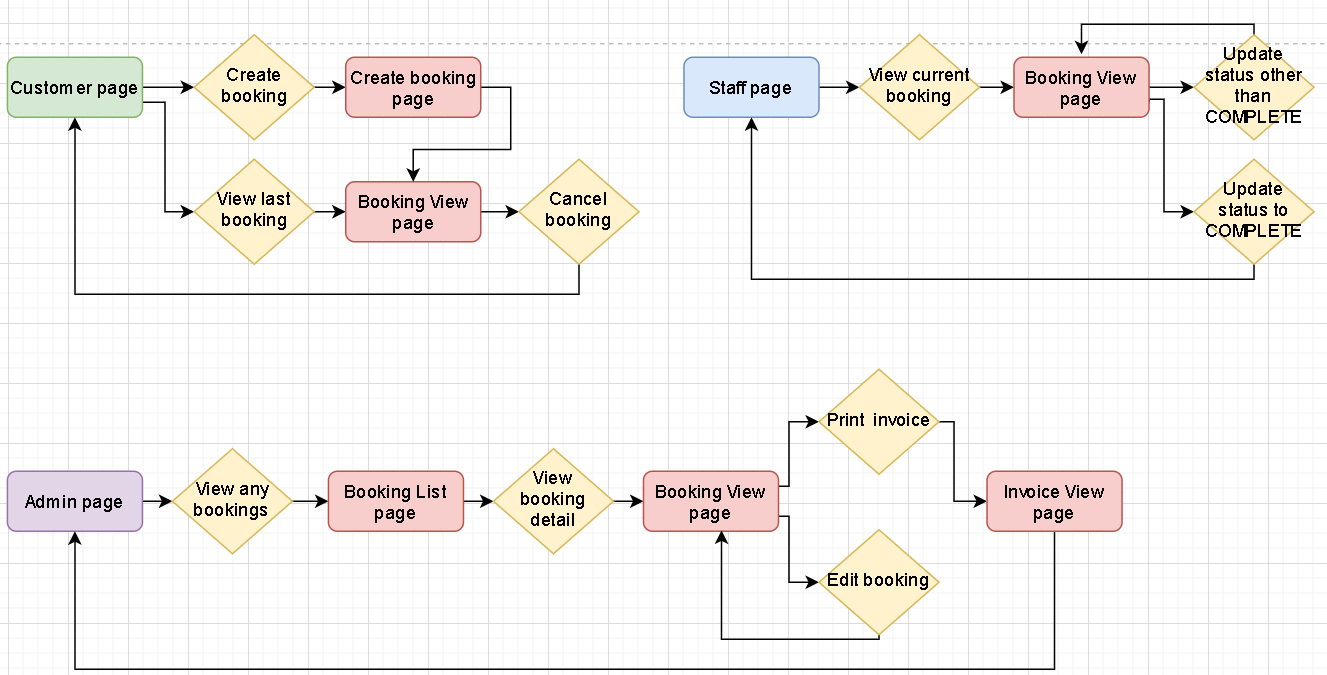
Administrator Booking View

On the booking page the administrator can access any booking. In addition, he is able to update booking status. In case the service implies to use stock supplies, the admininstrator can add it on the notes and also add the value description for the extra service: See below:

* Booking Status (Mandatory): Booked, In Service, Fixed, Collected and Unrepairable/Scrapped;
* Description Extra Service (Optional)
* Extra Service Price (Optional);
* Print Invoice (Mandatory);



Administrator Workflow



Booking Workflow: Customer, staff and adminstrator

Wireframes

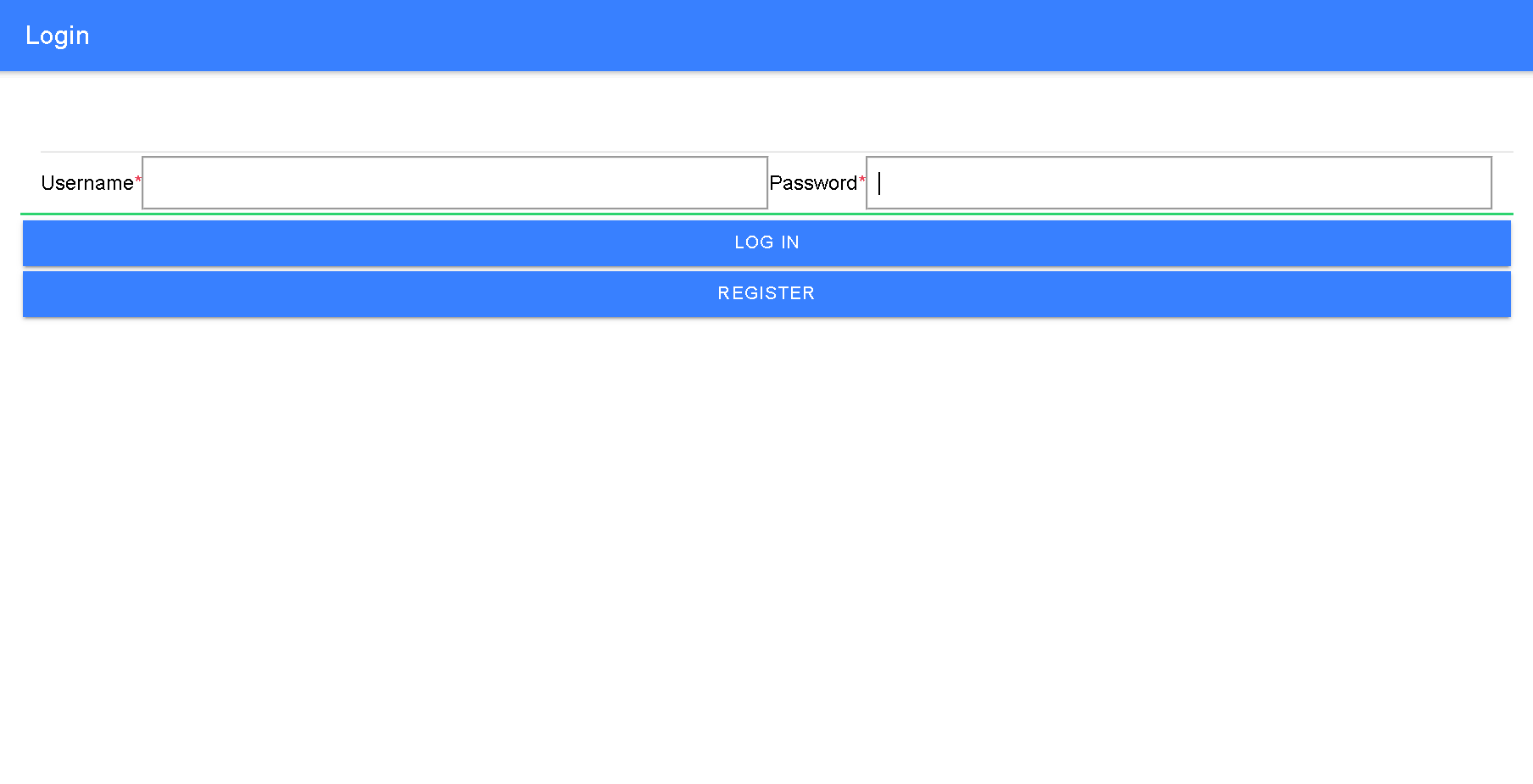
This section will depict the above workflows. For it, all requirements described on the previous section will be illustrated through wireframes.

PUBLIC PAGE

The Log-in and Register are public pages which are not necessary to access using username and password.

Log-in Page

It allows customers to register and staff, administrator and customer to sign in.



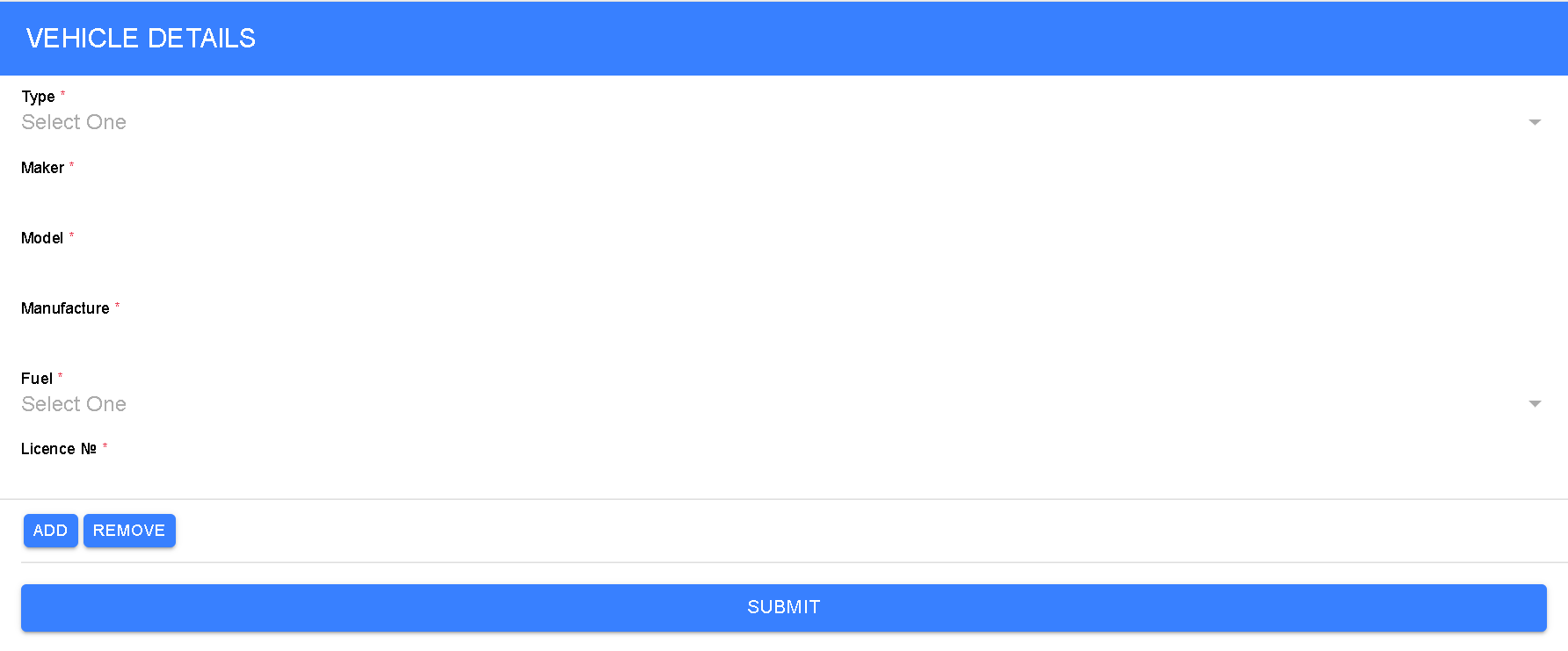
Login Page

Register Customer Page

Creation of a new customer account by filling up required fields and optional information.



Customer Register Page: User and Customer Details Section



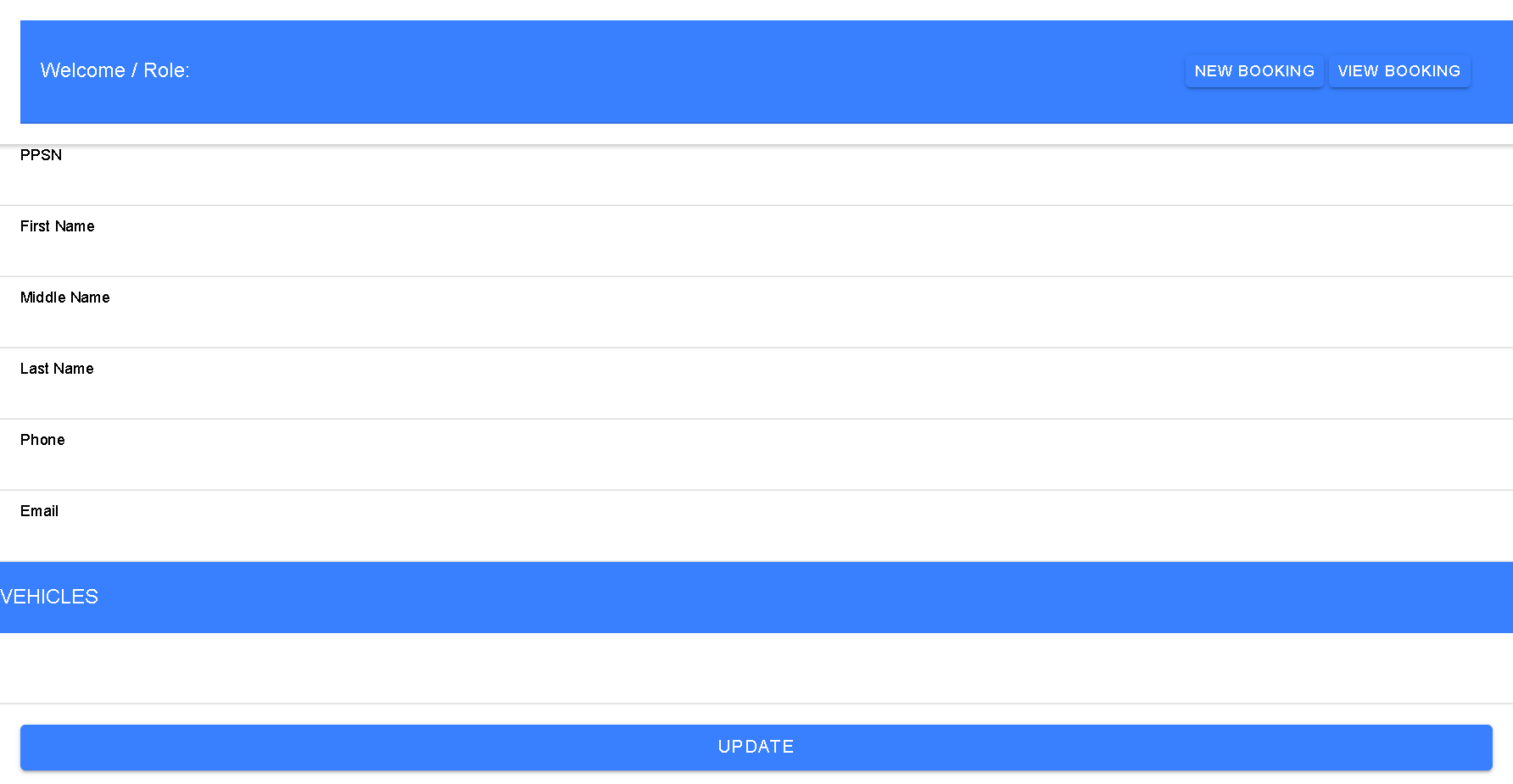
Customer Register Page: Vehicle Details Section

PRIVATE PAGE

Customer:

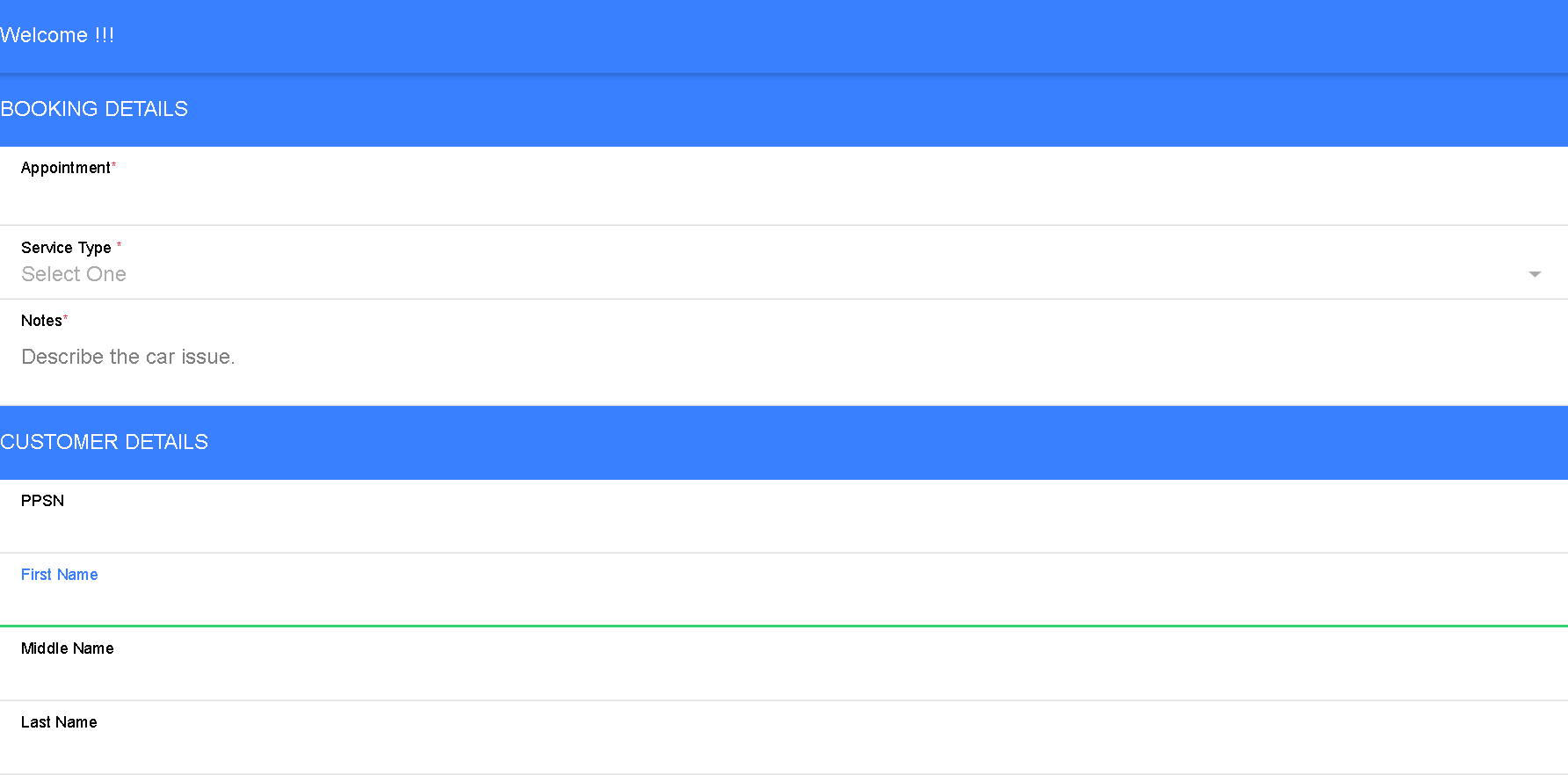
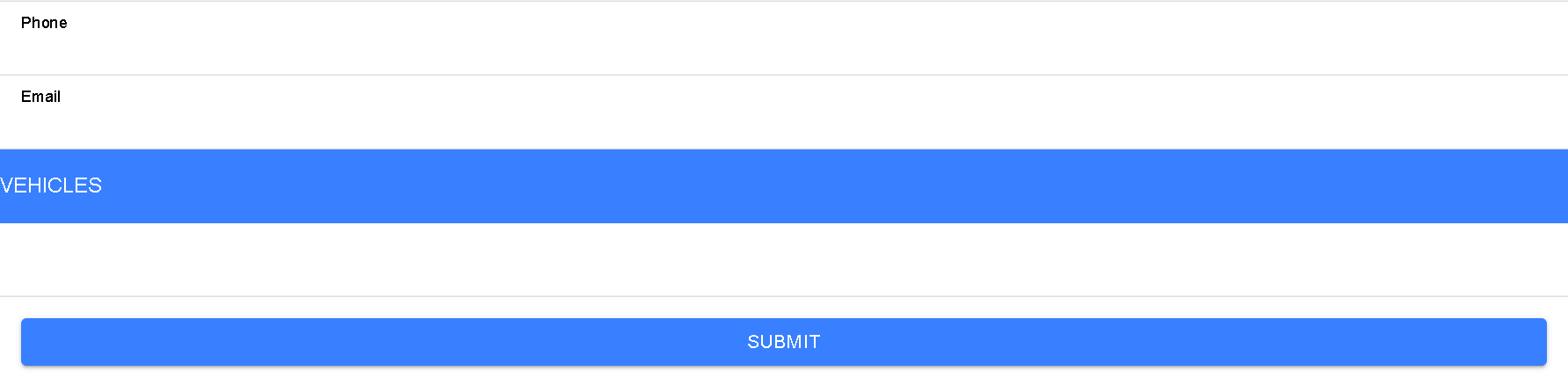
After creating an account, the customer can access the private page by using username and password on the log-in page. The customer page view will be loaded with their personal information and vehicle details.

At this point, the customer is able to update personal and vehicle details, make an appointment for a new booking and view the last booking.



Customer Page

If the customer decides to access the new booking page, his details will be automatically populated and other details will be required on the booking form.

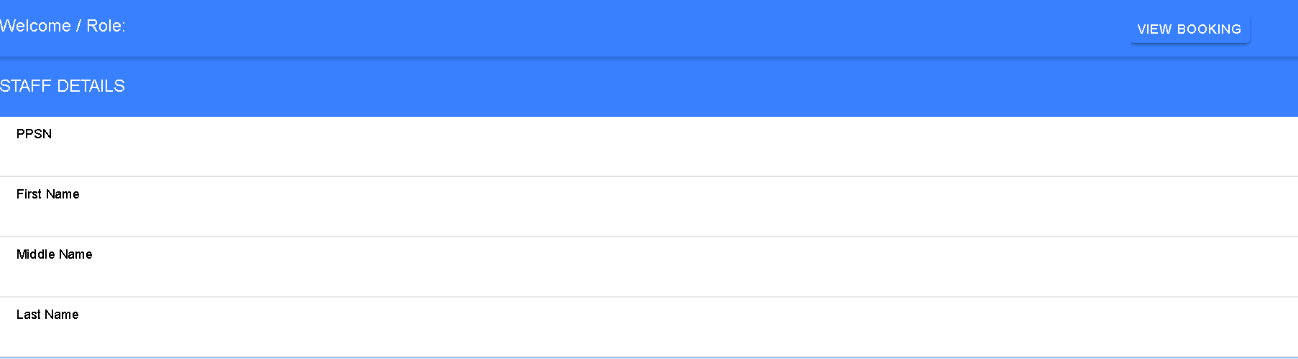
New Booking Page

In case they have already created a booking, customer can view the last booking page, which is very similar to the new booking page but brings additional information such as booking status and staff name responsible for their vehicle service.

Staff:

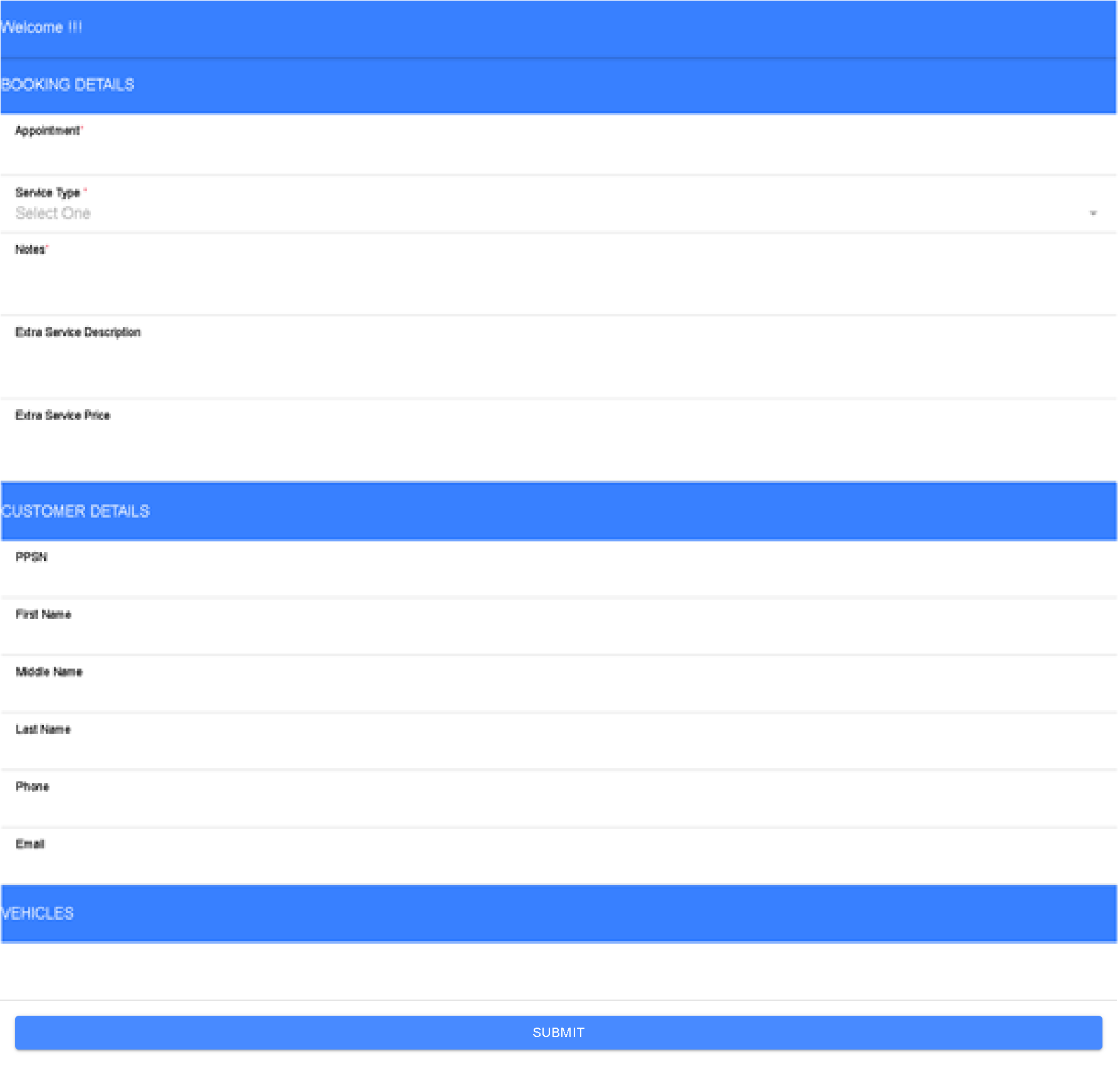
After having their accounts by the system admin, the staff will access the private page by using username and password on the log-in page and it leads to the staff page view where all personal data will be populated while the page is being loaded.

On this page, the mechanical is able to change personal details by using the button update. Also, will be possible to view all assigned booking for the current day.



Staff Page

If the staff decides to access the booking page, it will show a booking list for the current day and he can access a specific booking where the mechanical can see the details for the current booking, and change status. This page is very similar to the booking page illustrated on the customer section. The small changes are: Booking status drop-down will available for changing and extra inputs (Extra service description and extra service price.)



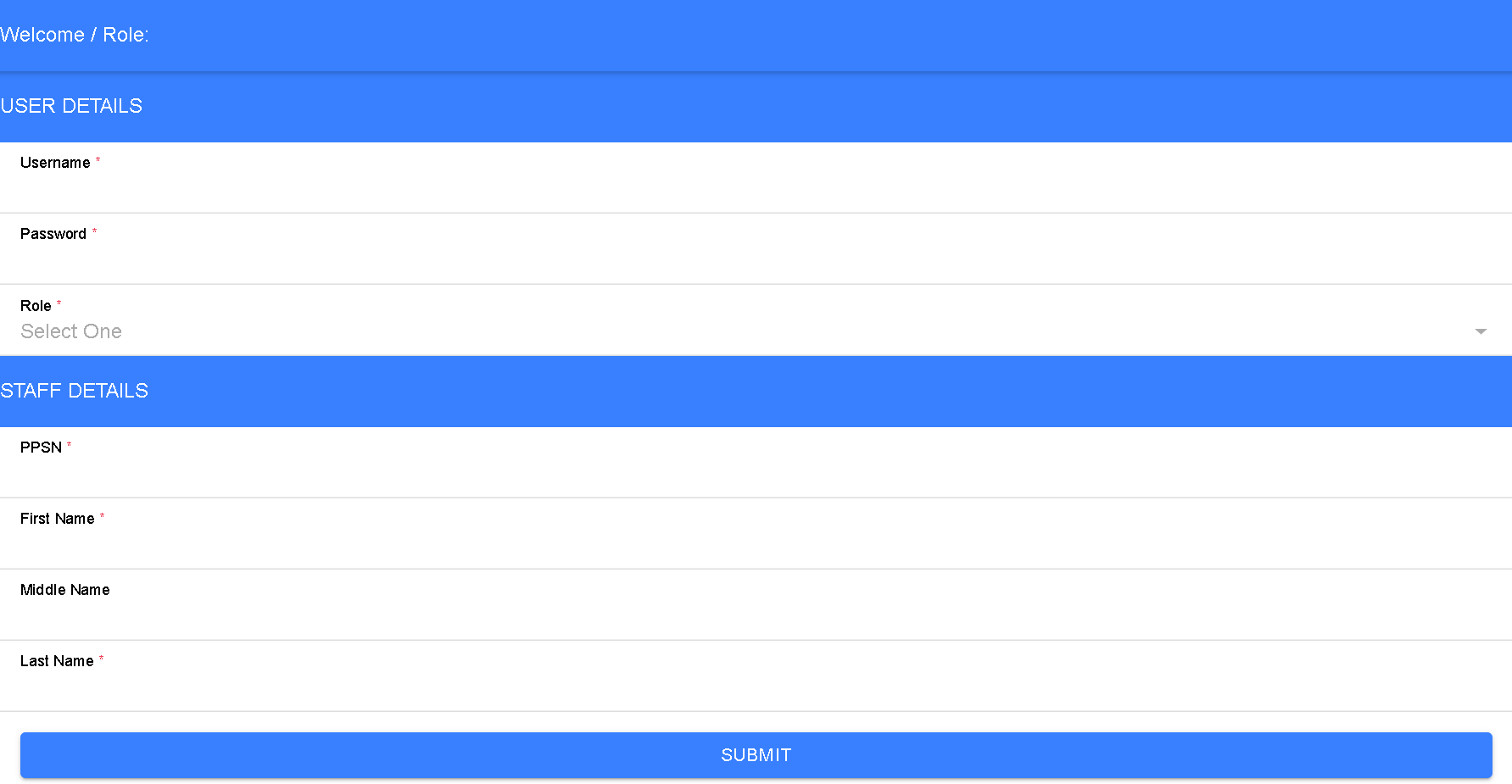
Booking View Page

Staff can also print the invoice to the customer.

Administrator:

After having an accounts set up at the very beginning, the administrator will access the private page by logging-in. It will bring to the administrator page view where all personal data will be populated while the page is being loaded.

On this page, the system admin is able to create others staff and view any booking. When creating new staff on the system, the staff register page will be loaded where the admin will type the staff details as seen on the image below.



Staff Register Page

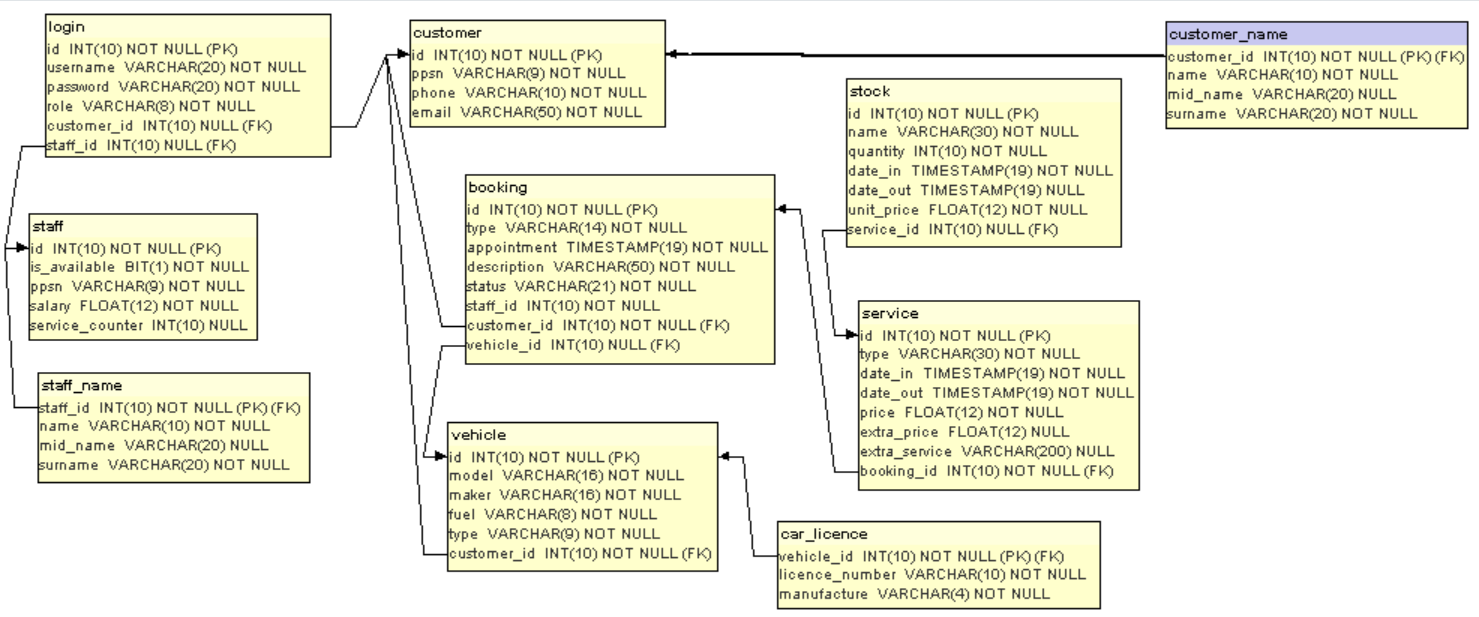
If instead of creating a new staff the administrator decides to view bookings, it leads to a list of bookings where it is possible also search per appointment date. By choosing one of them, the administrator will access the selected booking. It shows the same page and gives the possibility of updating booking status and all the others mandatory and optional fields on the staff booking view page section.

DATABASE

A relational database (MySQL) has been used to build this project in order to store all relevant data about Ger’s Garage Web System. Using this tool, required and additional tables have been created to a better working system. And to support the project with creation and drop tables and queries, a MySQL Workbench tool has been used.

Diagrams

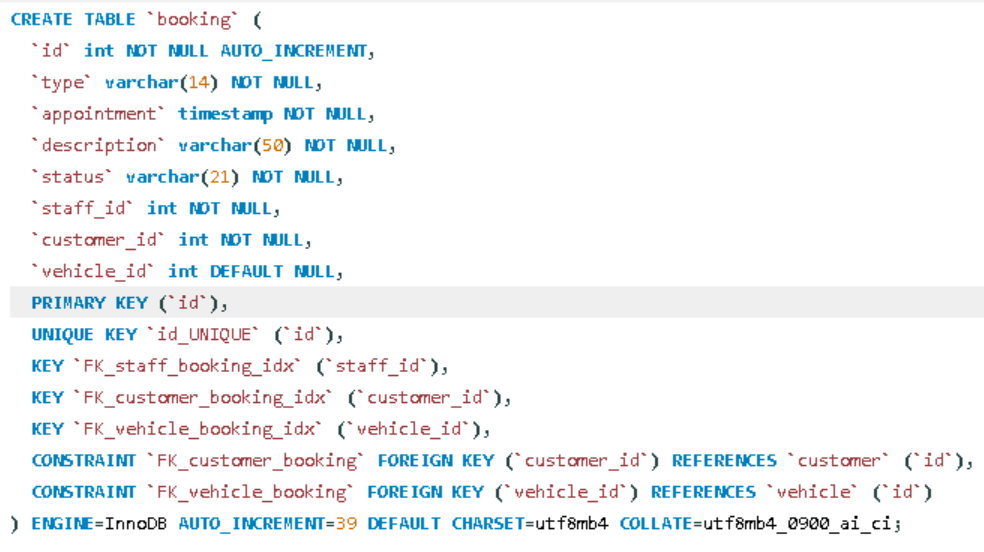
The project started been developed from the Entity Relation Diagram (ER). It depicts the database structure itself and help on the development side. See the diagram below and attributes, entities and relations.



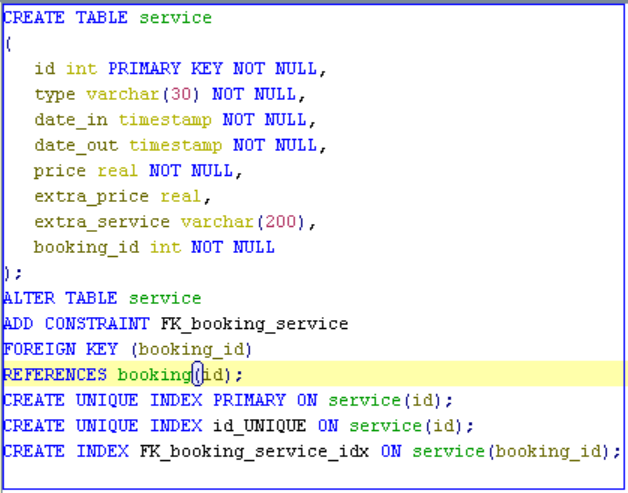
Entity Relational Diagram

Database Data Definition Language (DDL)

DDL defines structures such as database, schemas, constraints and others. It helps to create and make changes on the database. See below images of the DDL of this project:



Booking Table DDL

It is possible to create a script for the table structure as well. It helps when replication the database to another environment.

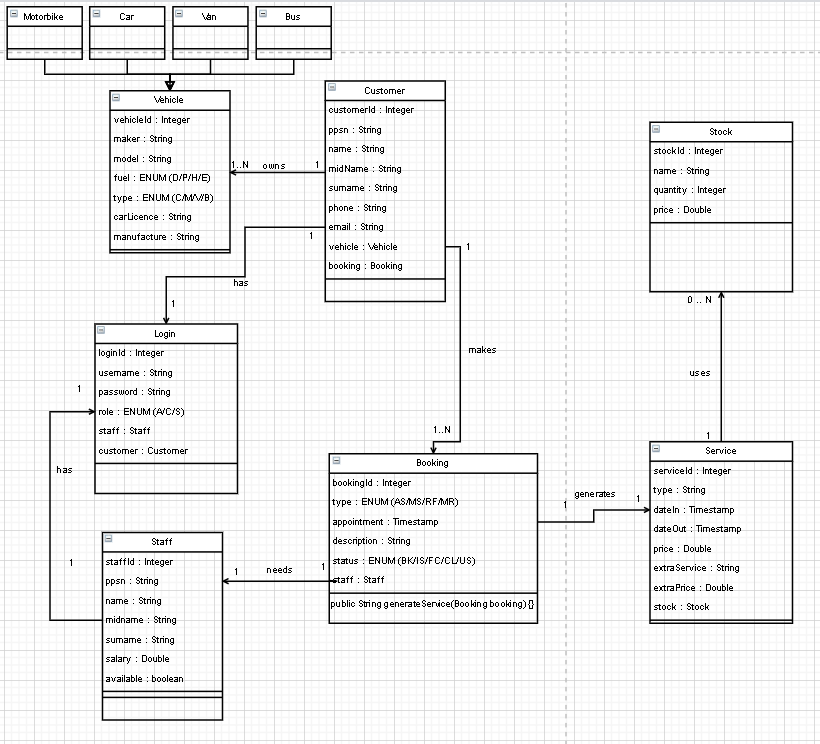
DB Script

BACK-END

Java is an Object Oriented (OO) language. The decision of Java is due to it is widely spread with big companies building their system and many forums and communities to support. Also, there are many different frameworks to support Java development.

Diagrams

In parallel with the ER diagram, a class diagram also has been produced to make easier to view all classes, attributes, methods and relationships.



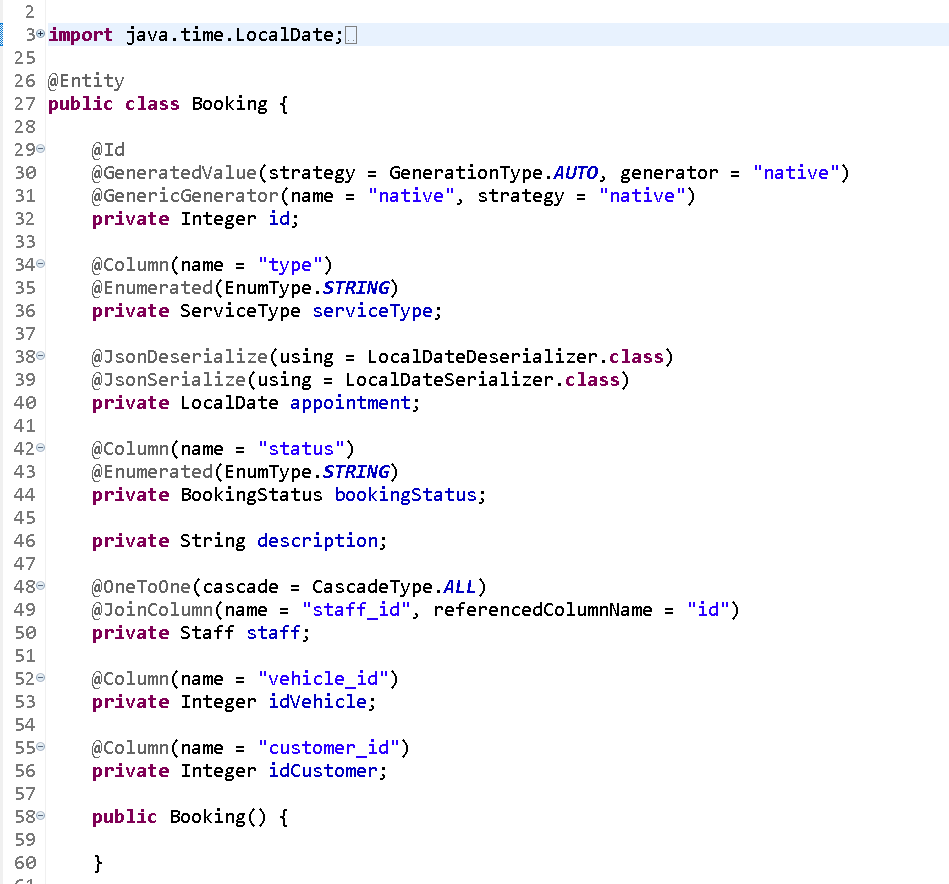
Class Diagram

Also, this project will show some of the most known and used Java frameworks.

Java Persistence API

Previously described on the Chapter 1, JPA helped on the Java development dealing with queries and database connection.

This API supports on mapping the by using annotations. See below Booking class being mapped with those annotations. Notice that it reflects the database.



Booking Class using JPA annotations

Gradle Framework

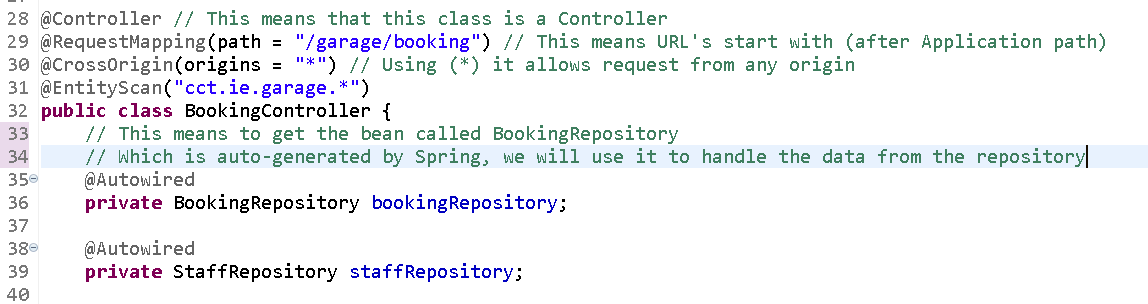
The back-end development started through this framework. It as a build automation tool to control and download project dependencies. With Gradle, by adding dependencies on the “build.gradle” file, similar to POM.xml from Maven, the project download all libraries which will be used to develop.



Spring Framework

It is a framework which supports on building web services. Springs annotations supports on the development side. With those annotation, it becomes simple to create HTTP Request and Response.

See below Booking Controller with some required annotation to build the web service.



Spring Annotations

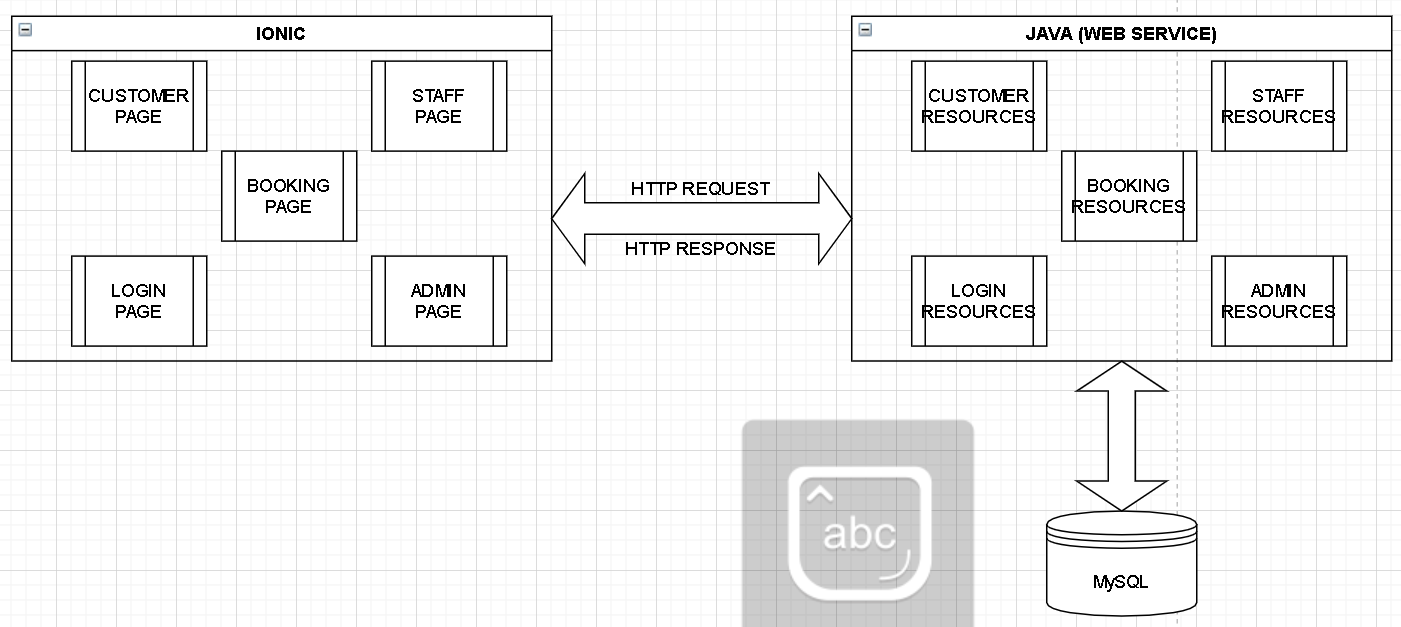
Also, see below the Java Project Structure.



FRONT-END

Ionic is front-end language used to build this application and integrated with Angular gives a powerful tool to front-end development.

The front layer has been built in an indepent way, what makes easier to customise the front-end. Towards this, it is just necessary build another front-end and call the endpoints, with no effect on the back-end side.



Garage System Architecture

CHAPTER IV:

SYSTEM IMPLEMENTATION

As seen on Gantt chart through the specified tasks or on the product backlog on the previous chapters, the project life cycle has been split into 3 main phases:

* Management and Design;
* Development;
* Testing;

The previous chapters has explained about the first part of the life cycle and now, the development phase of the project will be shown.

ARCHITECTURE

The project architecture is based on Restful and uses Java 8 to build the web service and make the resource available. Besides Java, JPA manages the database access, Spring to support to create the endpoints, MySQL as a data store and Ionic to create pages to user accessibility.

TECHNOLOGIES IMPLEMENTED

FRONT END

* Ionic;
* HTML;
* CSS;
* JavaScript;
* Angular;
* HTTP Methods;
* JSON;

BACK END

* Java;
* JPA;
* Tomcat;
* Spring Framework;
* Gradle;

DATABASE

* MySQL;

DEVELOPMENT

The following table summarizes all tasks on the development life cycle and includes issues, troubleshooting and solution accomplished on this project.

