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 assign automatically booking to the staff after customer make 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 to service, type of service and type 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). A plus is the front-end is built using ionic calling the endpoints from the back-end. In other words, to change the View layer is just create a new one and connect to the same endpoints.

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 list.

*  Java 8, object oriented language widely spread and therefore many professionals, forums and guides are available for help and guidance;

 Service Oriented Architecture (SOA) as architectural pattern;

 Relational Database (MySQL);

 Java APIs such as JDBC API in order to access to the DB;

*  Java Persistence API, as Java software component to map the database using annotations, and manage queries on the DB;
* Spring: This framework used to build micro services.
* Tomcat: It provides a HTTP web server environment which Java code can run.
* Gradle: This is an open-source build automation system based on Apache Ant and Apache Maven.
* Ionic: Framework for building mobile and desktop apps using web 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;