NAME – SAKSHAM JAIN ID – 201851106 GROUP - 4

END – SEM PROJECT REPORT

CS 363 (Software Engineering LAB)

Automobile Service Management

The Aim of our project is to solve the problem of our end users of standing in long queues and waiting for their turn to come for getting their vehicle serviced ; By developing a web application for fast and efficient functioning of the various processes involved in vehicle servicing.

With the help of our web application a user can book a service appointment for his vehicle as per his flexible date and time slot . It isn't necessary for them to take their vehicle at the service centre . As, in case of emergencies rather they can opt for scheduling a pickup where a Tow-Er will be assigned by the Service Center for picking and delivering the vehicle to the centre.

With the Track Nearest Center Locator , it is easy for a client to track the service centres which are nearby him . While the most important thing a user seeks is to check the service progress of his vehicle whenever he wants to do it. For this , the Track Progress feature is available using which one can check for updates of the status of the service at any time.

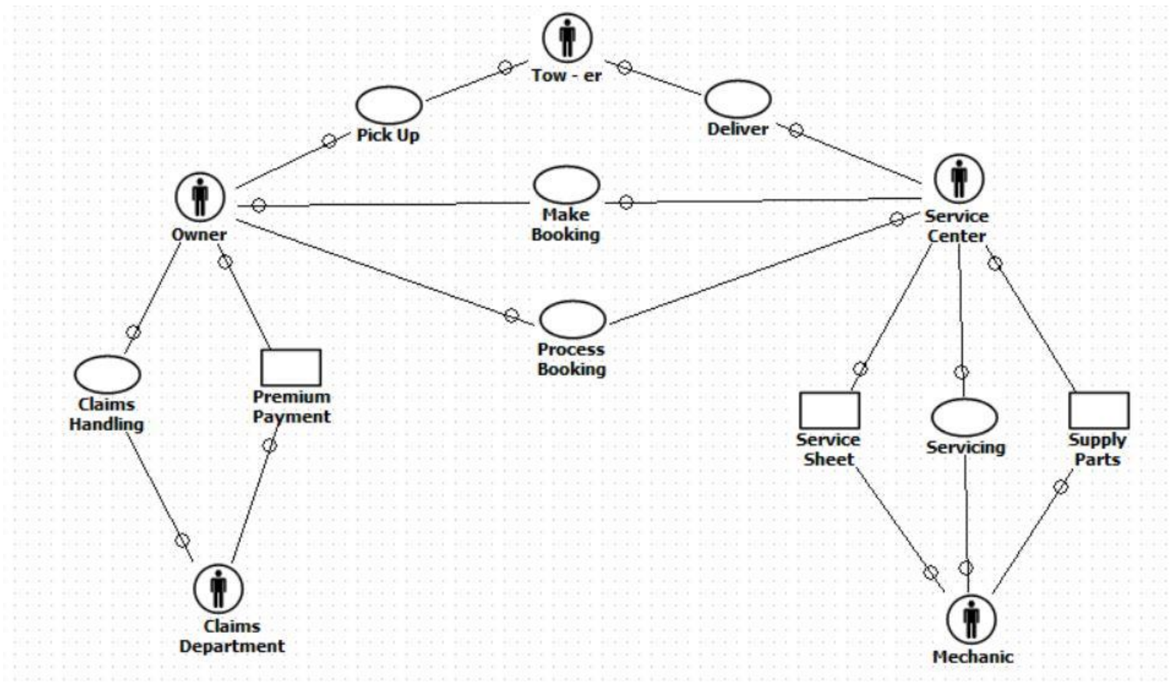
Customers can also check the details of their Vehicle Insurance by requesting it from the Service Center which further sends a request to the Claims Department . Authentication is an important aspect for this system which is done by using id and password. The real power of this project lies not in direct selling of products, but in the creation of tighter relationships with customers and delivering of a high level of service and support, which in turn improves organization sales and its goodwill.

Further we have presented the idea of our project in the form of a Strategic Dependency Model and various Strategic Rationale Models within it .

NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**

Strategic Dependency Model

The given figure is the SD Model of our projects which includes Various Agents and Dependencies among them.

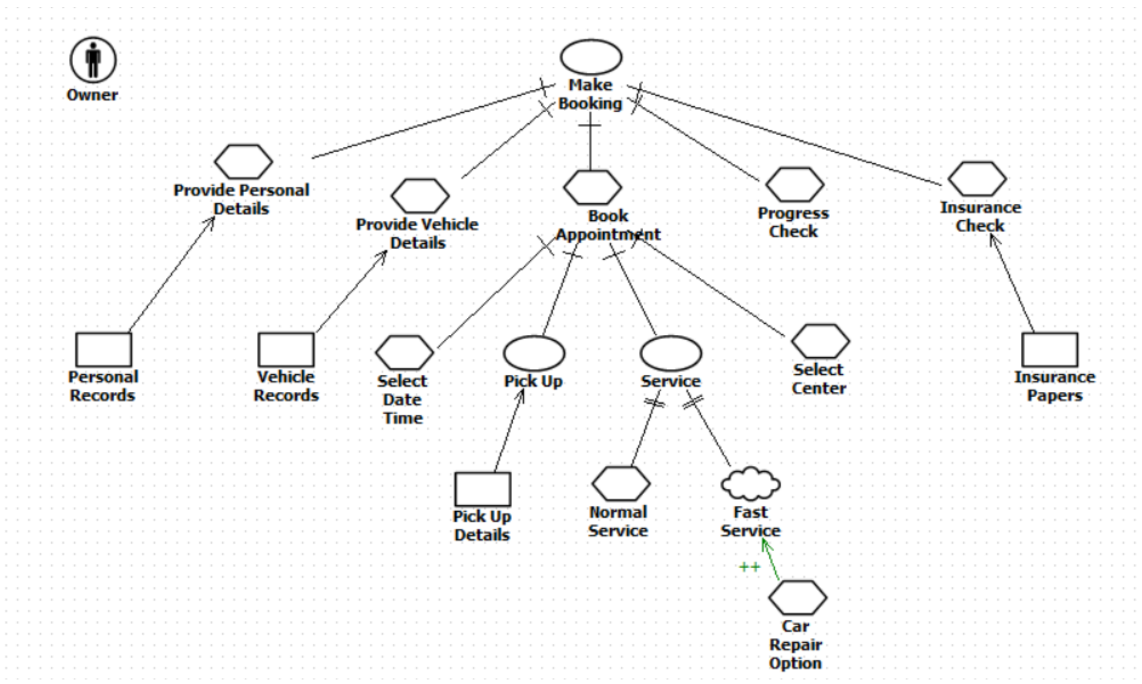


1. **Service Center** depends on **Owner** to Make Booking (Goal Dependency) using the web app
2. **Owner** depends on **Service Center** to Process the Booking (Goal Dependency)
3. **Owner** depends on **Tow-Er** to Pick Up the Vehicle (Goal Dependency)
4. **Service Center** depends on **Tow-Er** to Deliver the Vehicle (Goal Dependency)
5. **Service Center** depends on **Mechanic** to do the Servicing of Vehicle (Goal Dependency)
6. **Service Center** depends on **Mechanic** to provide the Service Sheet (Resource Dependency) in order generate bill for the service
7. **Mechanic** depends on **Service Center** to Supply parts (Resource Dependency) to perform servicing of vehicle
8. **Owner** depends on **Claims Department** for Claim Handling (Goal Dependency)
9. **Claims Department** depends on **Owner** for Premium Payment (Resource Dependency)

NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**

Strategic Rationale Models

Owner acts as the end user in our system and it the most important actor in the SD model. The main Goal of Owner is to **Make a booking** which is demonstrated in given SR diagram.



SR Model for Make Booking

In order to make a booking first we have to execute the following subtasks –

Provide Personal Details – Details of personal is provided by owner in form of a resource.

Provide Vehicle Details – Details of the vehicle is provided by owner in form of a resource.

Book Appointment – In order to book an appointment there are various subtasks and subgoals.

Select Date & Time (Sub Task) **Pick Up** (Sub Goal – Further Depends on Tow-Er)

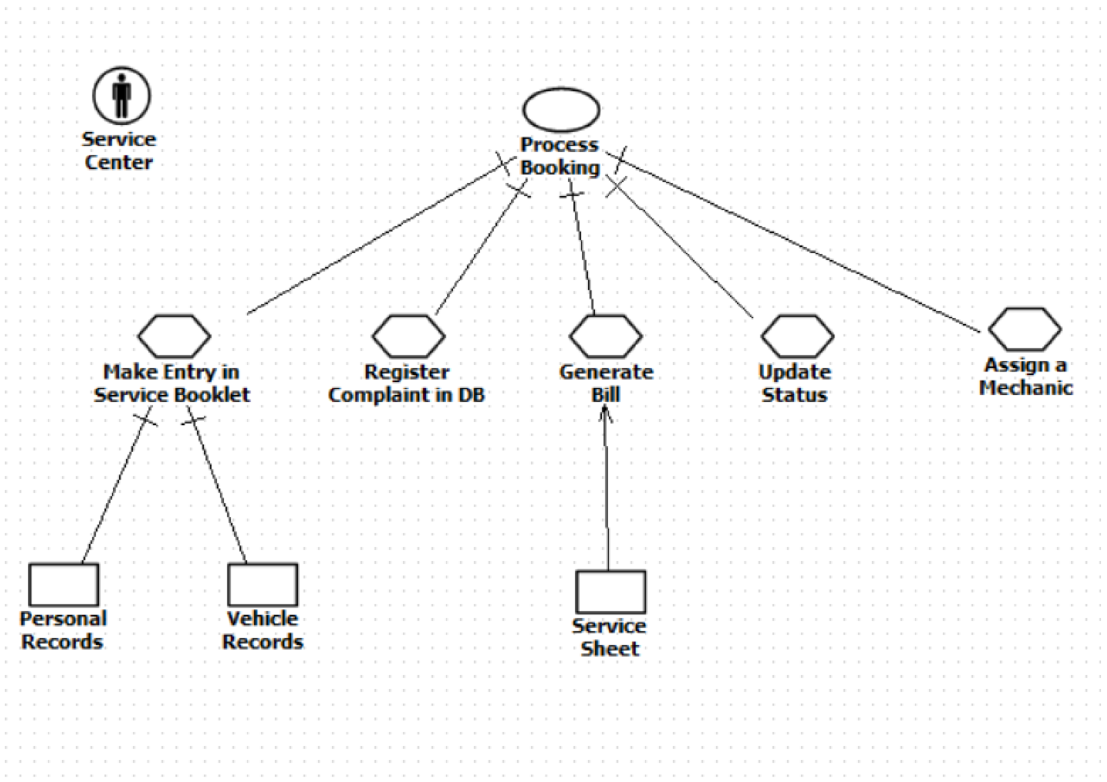
Select Center (Sub Task) **Service Type** (Sub Goal) – Can be Normal or Fast

Progress Check – Check the Status of Progress of Servicing which further depends on Service Center.

Insurance Check – Check the validity of Vehicle Insurance by Insurance Papers (Resource)

Fast Service is a NFR Soft goal and using the Car Repair Option helps in achieving this Goal

Service Center is one of the key actor in our system. The main Goal of the Service Center is to **Process Booking** which is demonstrated in given SR diagram.



<u>SR Model for Process Booking</u>

In order to process a booking first we have to execute the following subtasks –

<u>Make Entry in Service Booklet</u> – Details of the vehicle and the owner has to be recorded in the Center Database so as to maintain proper records of services . For this , Personal & Vehicle Records are provided by the owner as a resource.

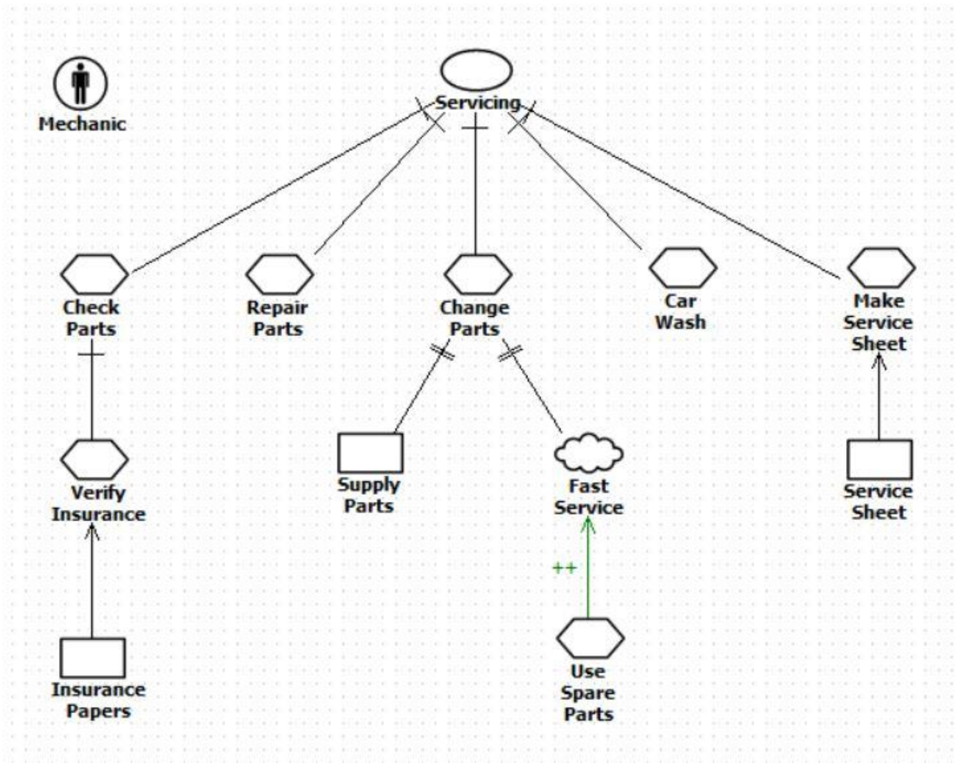
<u>Register Complaint in Database</u> – Registering Queries of the clients and answering them is a crucial part of processing a booking of vehicle service.

<u>Update Status</u> – It is the responsibility of Service Center to update the status of services time to time so that owner can check the Status of Progress of his Vehicle service.

<u>Assign a Mechanic</u> – Assigning a Mechanic as an actor so that servicing can be done is a key task of the centre.

<u>Generate Bill</u> – Making the Bill for the vehicle servicing with the help of Service Sheet (Resource) which is provided by mechanic after successful completion of vehicle service.

Servicing of the Vehicle is the basic aim of our system for which is the responsibility of **Mechanic** actor.



<u>SR Model for Servicing</u>

In order to process a get servicing done , first we have to execute the following subtasks –

<u>Check Parts</u> – Mechanic checks the vehicle for the fault in various parts. When the parts are found he verifies by using Vehicle Insurance Papers (Resource) that whether the faulty part is covered in the insurance (in case of accident) or not.

<u>Repair Parts</u> – Perform repair for the faulty parts found in the vehicle.

<u>Change Parts</u> – Mechanic change the faulty parts if they cannot be fixed. This can be achieved by two ways either by using Supplied Parts (Resource) from the Service Center OR by using Spare Parts which are already available in order to get the parts changed faster.

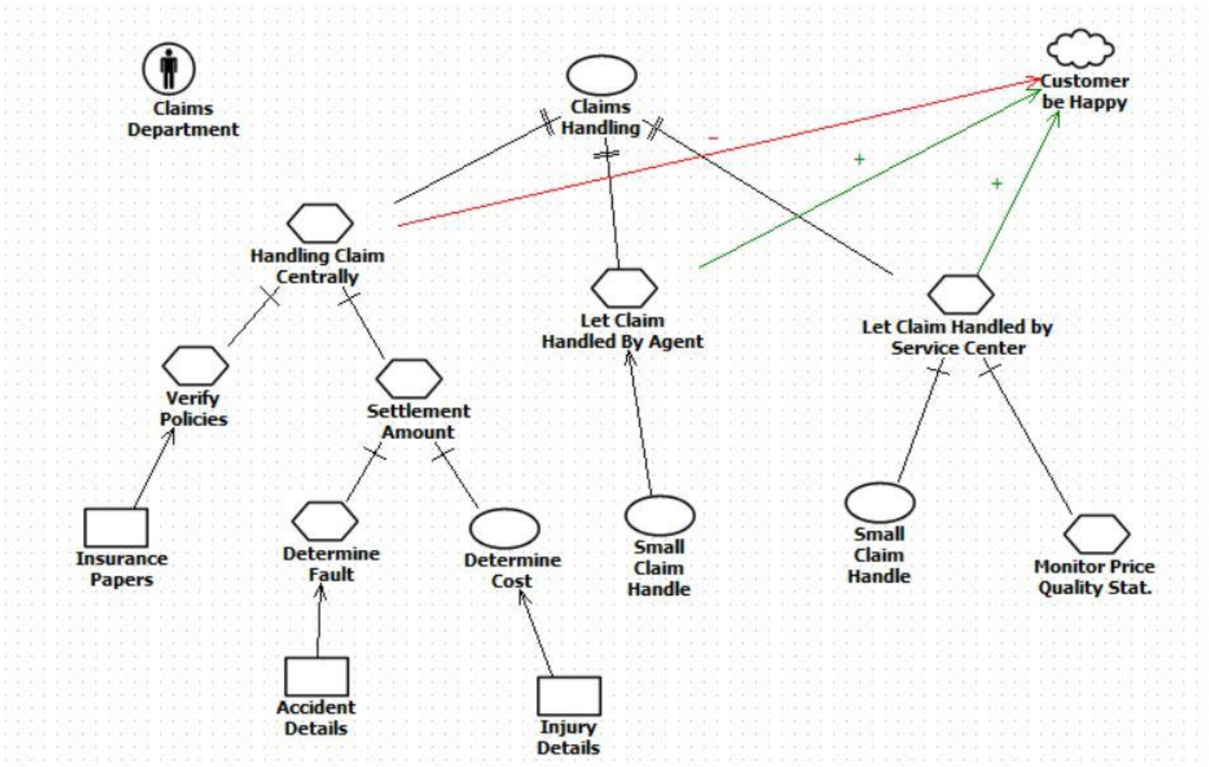
<u>Service Sheet</u> – Preparing the Service Sheet after performing all the repairs by specifying the repairs done in it and further providing it to the Service Center for Bill Generation.

<u>Car Wash</u> – As the last part of Servicing get the vehicle clean before returning to the owner.

Fast Service is a NFR Soft goal and using Spare Parts helps in achieving this Goal

NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**

Claims Handling in case of an accident of the Vehicle is performed by the **Claims Department** which is also responsible for maintaining the insurance of the vehicle.



SR Model for Claims Handling

There are 3 possible ways in our system by which the claim settlement can be done –

Handling Claim Centrally – This is performed by the Claims Manager where he first verifies the vehicle policies with the help of Insurance Papers (Resource) then does the amount settlement by Determining the Fault and the Cost using Accident Details and Injury Details as resources respectively.

Let Agent Handle the Claim – Another option for claim settlement is to get it settled by an Agent in case it is a small claim handle.

Let Service Center Handle the Claim – Claim can also be handled by the Service Center in case it is small by monitoring the price quality stat without the involvement of unnecessary agents.

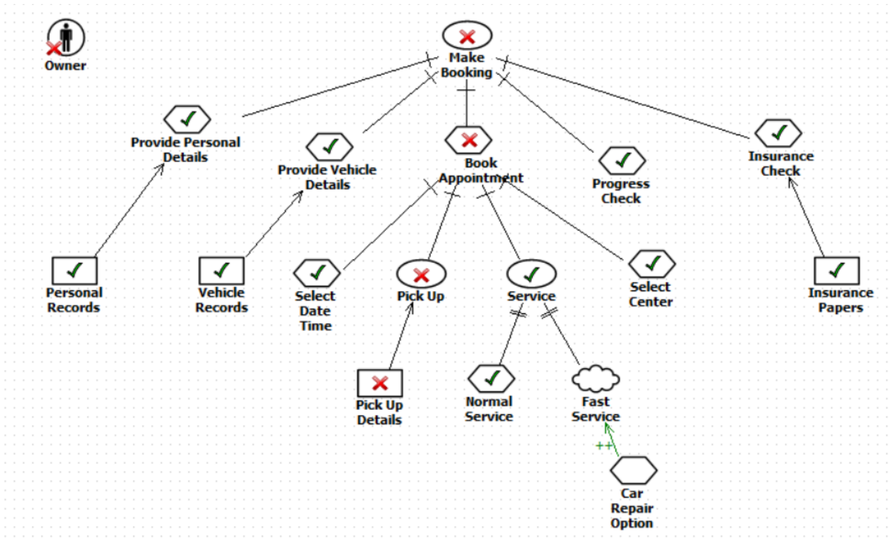
Customer be Happy is a NFR Soft goal . In case of third party involvement in claim settlement it hurts the customer as when more people are involved in the chain process then it gets delayed.

NAME – SAKSHAM JAIN ID – 201851106 GROUP - 4

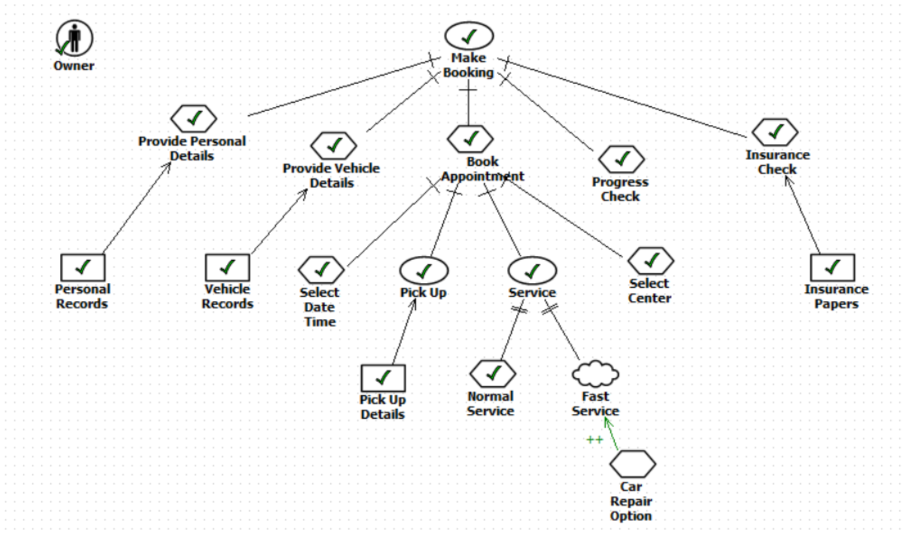
Satisfiability Analysis (Owner SR Model)

In a **Closed World** System we assume that when something not known to be true then it must be false. While in an **Open World** we assume that when something is not known to be true then it is considered unknown.

Here , there are 5 subtasks in order to achieve the hard goal. We can see that as soon as any subtask is denied , our hard goal is denied and is not achievable. Here we deny giving the Pick Up details so the appointment is not booked and so we fail to make a booking for the service of vehicle.



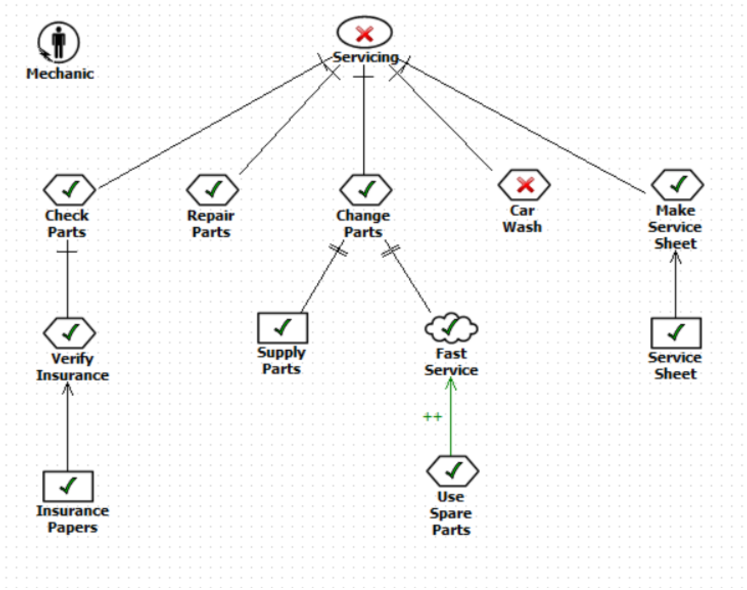
As soon as we satisfy the Pick Up Details Resource in the SR model , the changes are propagated through the levels making all the conditions satisfiable and hence our hard goal of **Make Booking** is satisfied.



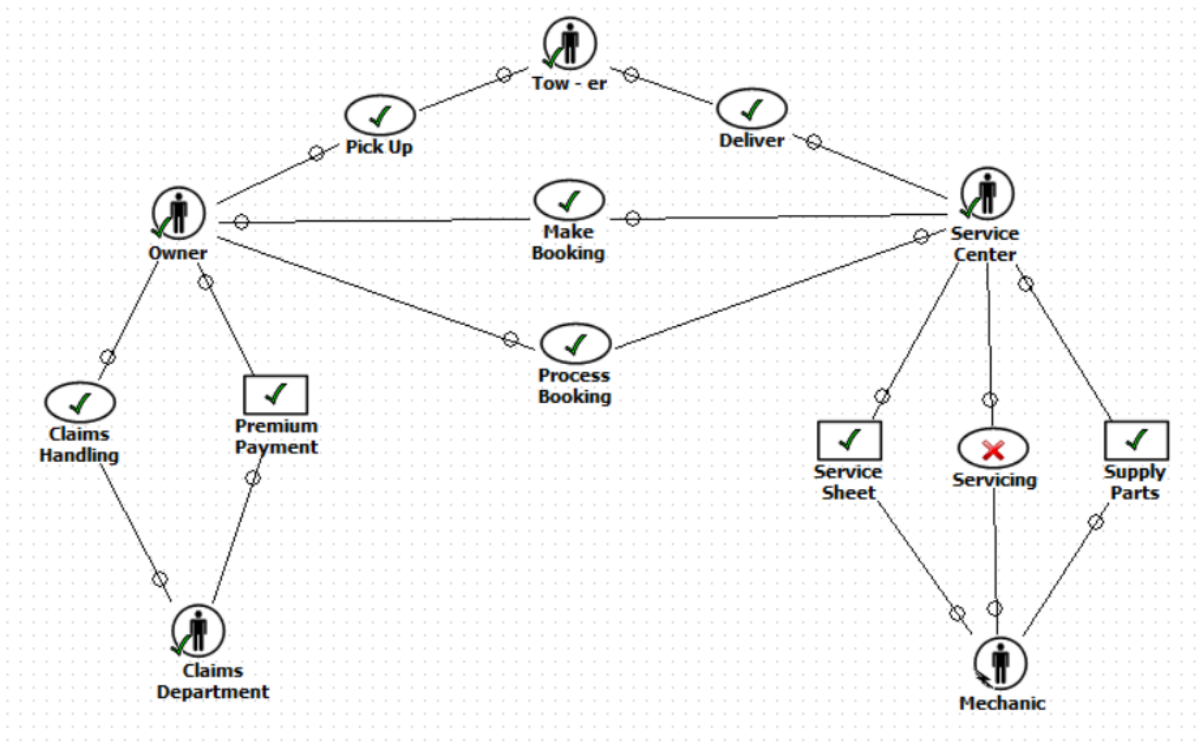
NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**

(Mechanic SR Model)

Similar to prior case , in this SR model there are 5 subtasks in order to achieve the hard goal. We can see that as soon as any subtask is denied , our hard goal is denied and is not achievable. Here we deny Car Wash so the Servicing cannot be considered satisfied.

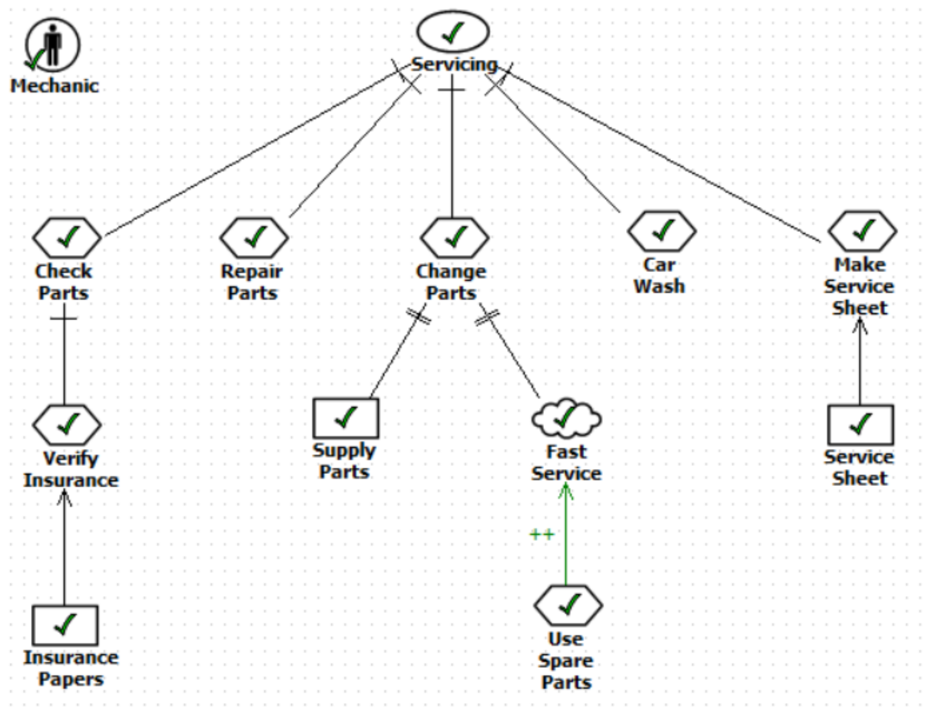


Further in case of every SR model , the changes made in SR diagrams are also visible in SD Diagram. As here the hard goal of Servicing is Denied and hence the state of Mechanic is Conflicting.

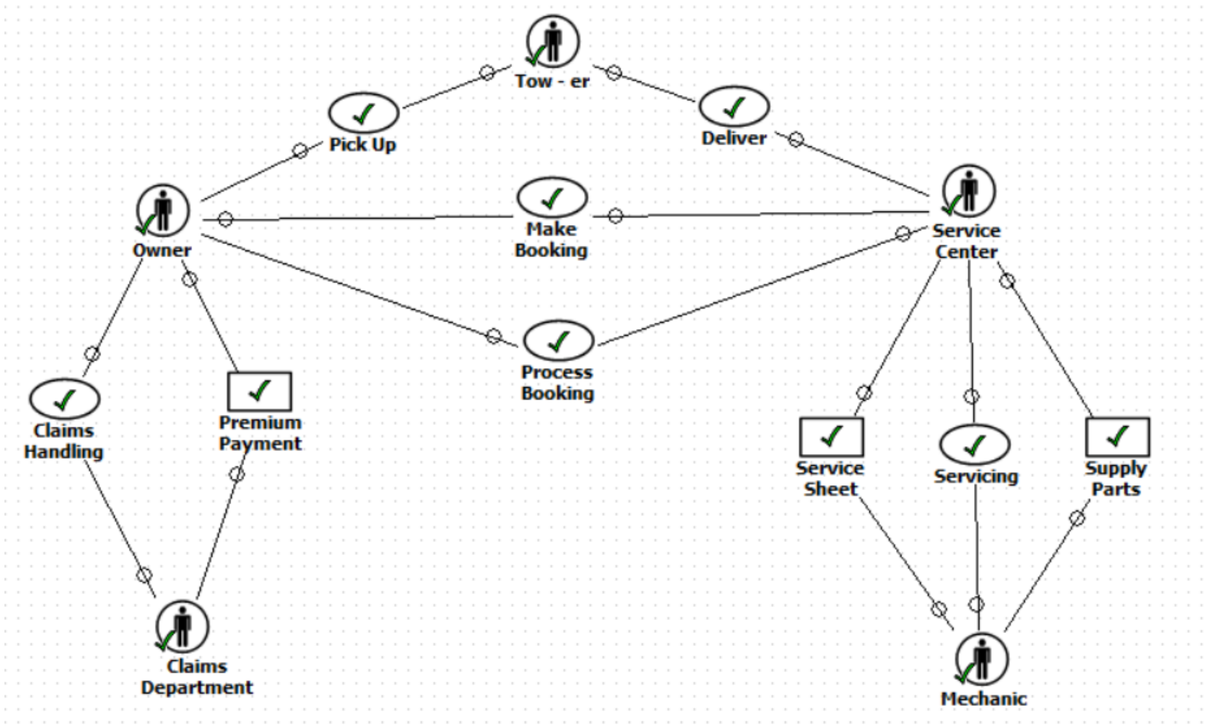


NAME – SAKSHAM JAIN ID – 201851106 GROUP - 4

As soon as we satisfy Car Wash in the SR model , the changes are propagated through the levels making all the conditions satisfiable and hence our hard goal of **Servicing** is satisfied.



Also the changes are reflected in SD Model too.

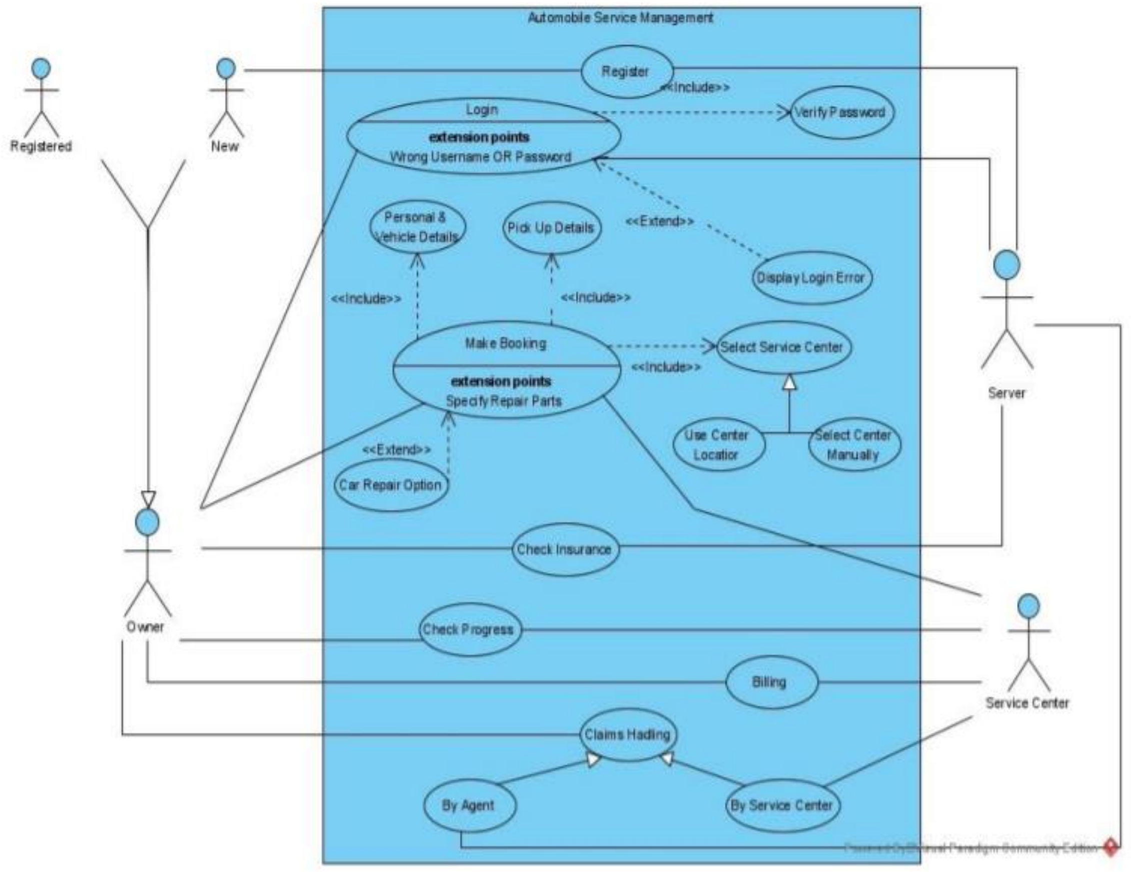


MID SEM REPORT END

NAME – SAKSHAM JAIN ID – 201851106 GROUP - 4

Visual ParaDigm Diagrams

Use Case Diagram



Automobile Service Management is the System of the diagram which has the given set of actors i.e. Owner (New Or Registered) , Server (System Admin) and Service Center.

Initially an Owner Logs in (Use case) into the system . It has an include relation with Verfiy password depicting that password needs to be verified. Hence , The include relationship adds an additional functionality which is not specified in the base use case here.

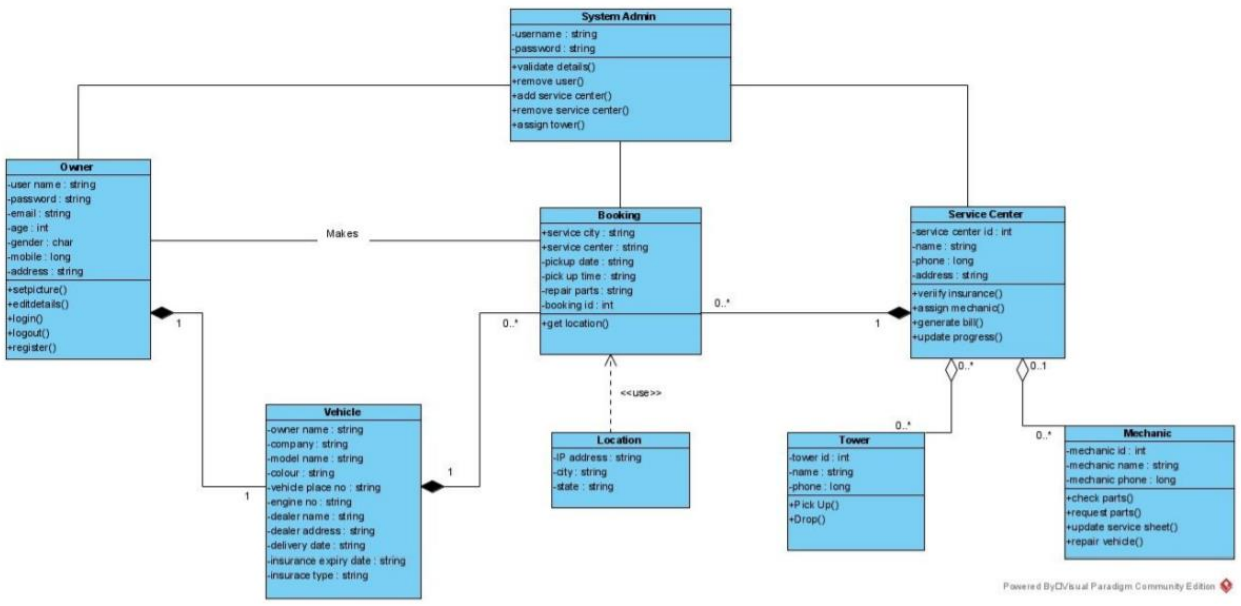
Display login error is an extend use case of Login which shows the optional system behviour in case the flow reaches the given extension points.

Then the owner makes a booking. This use case has include relation with other use cases that are Person , Vehicle and Pick Up details along with selection of Service Center.Car repair option is an extend use case of make a booking whose usage is optional.

Select Service Center case has 2 child cases specifying that either of them must be used to ensure to Parent Use case.Futher Check Insurance use case can be used by owner to verify his vehicle isurance from server.

Other use cases include Progress check , Billing and Claims Handling and these are observed between Service Center and the Owner.

Class Diagram



Above is the class diagram of our mini project containing 6 classes. It provides a basic notation for other structural diagrams.

Owner – Attributes of this class are the personal details of the car owner along with it's username and password. Basically the details which one needs to provide during registration. Further the class contains several methods which are the services provided within it like logging in and out , editing details etc.

Vehicle – This class doesn't contain any method as it doesn't have any service of its own. Attributes within the class are the details of the vehicle which the user needs to provide in order to do a booking for servicing.

System Admin – Some of the most Crucial methods within our project are provided by this class. This class provides method for adding or removing an Owner as well as a Service Center within the system.

Booking – Attributes of this class are the details required in order to make a booking like Service center location and pick up drop off time. It has a method get location which is used to detect owner's location

Location – As a subclass of booking class it has attributes like IP address , city and state which are used for fetching the location

Service Center – Attributes of this class are unique details required for uniquely identifying a centre within the system. It contains the core services of our project as the following methods -Assigning Mechanic , verifying insurance , generating bill and Update status of progress

Booking has a composition relation with the 2 classes that are vehicle and Service center which means if any of these 2 classes are removed there is no significance Booking class. A similar relation exists between owner and vehicle.

Further Service Center aggregates 2 classes-

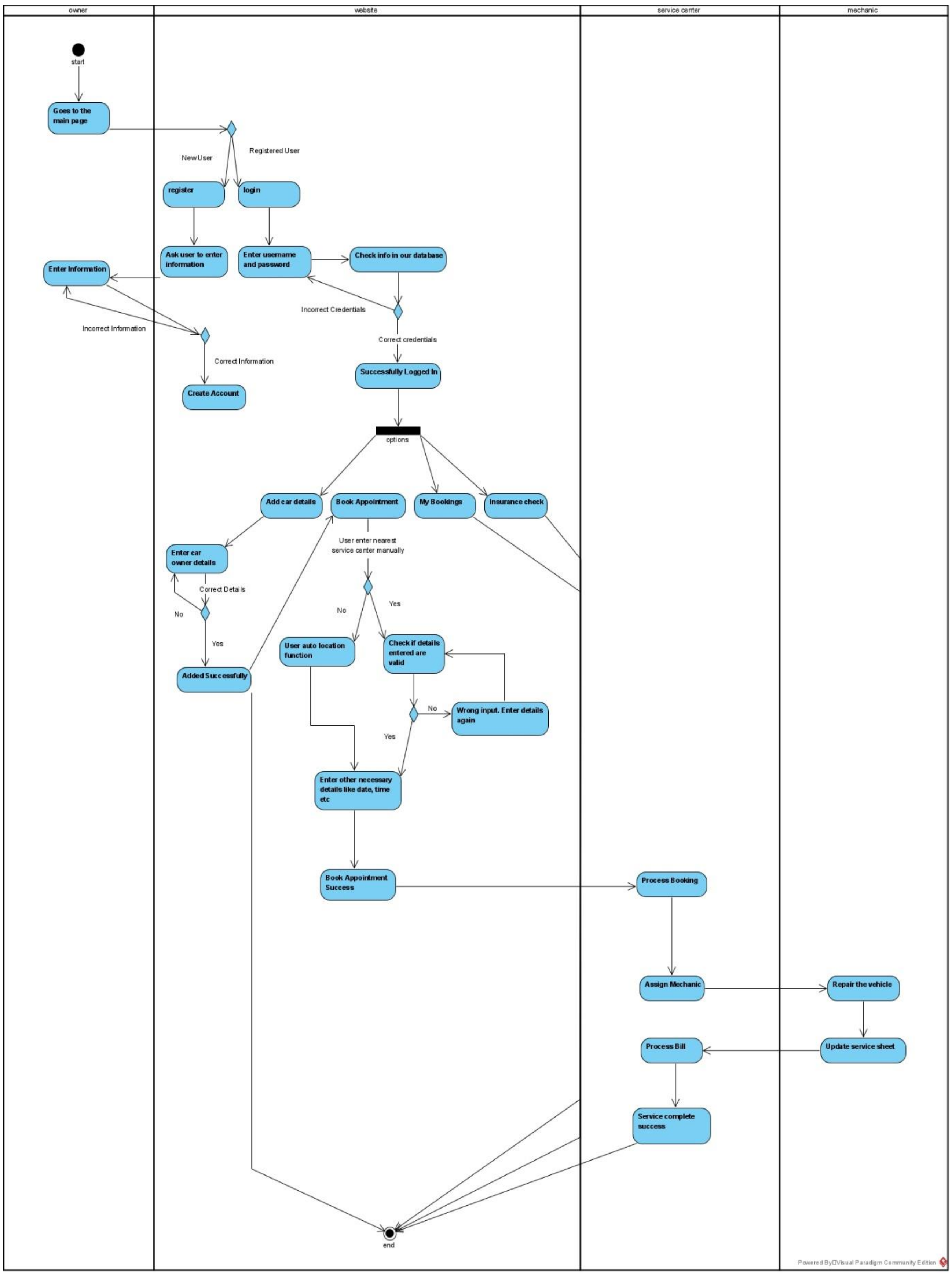
Tower – Attributes here , are unique personal details and this class provides Pick Up and Drop Off services as methods .

Mechanic – Attributes here , are unique personal details and this class provides methods involving services that are check and repair of parts of vehicle along with updating the service sheet.

Multiplicity between the classes is displayed in the above figure.

NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**

Activity diagram



Activity diagrams describe how activities are coordinated to provide a service at different levels of abstraction. It also gives a bigger perspective on what will be the basic workflow of our website.

So, activity diagram helps us to be clear in what process is to be followed if suppose we want to perform a particular task on website like for example viewing car details, booking an appointment etc.

We have divided our flowchart into four swim lanes namely owner, website, service center, mechanic.

First step here is to go to the main page of website. After that we have a decision node where user will have two options whether to login or to register. If user selects to register on the website then he will be asked to enter information and after the user has entered information then credibility of information is checked , in case not proper user will have to enter the same again. Similar Kind of process takes place for log in.

As User gets to the Dashboard 3 parallel options are displayed which are add car details, my bookings section and Insurance check. User can select any of these. Suppose he selects add car details option then in the next step he will be asked to enter car details. If the entered details are correct the car will be added successfully else he will have to re-enter car details.

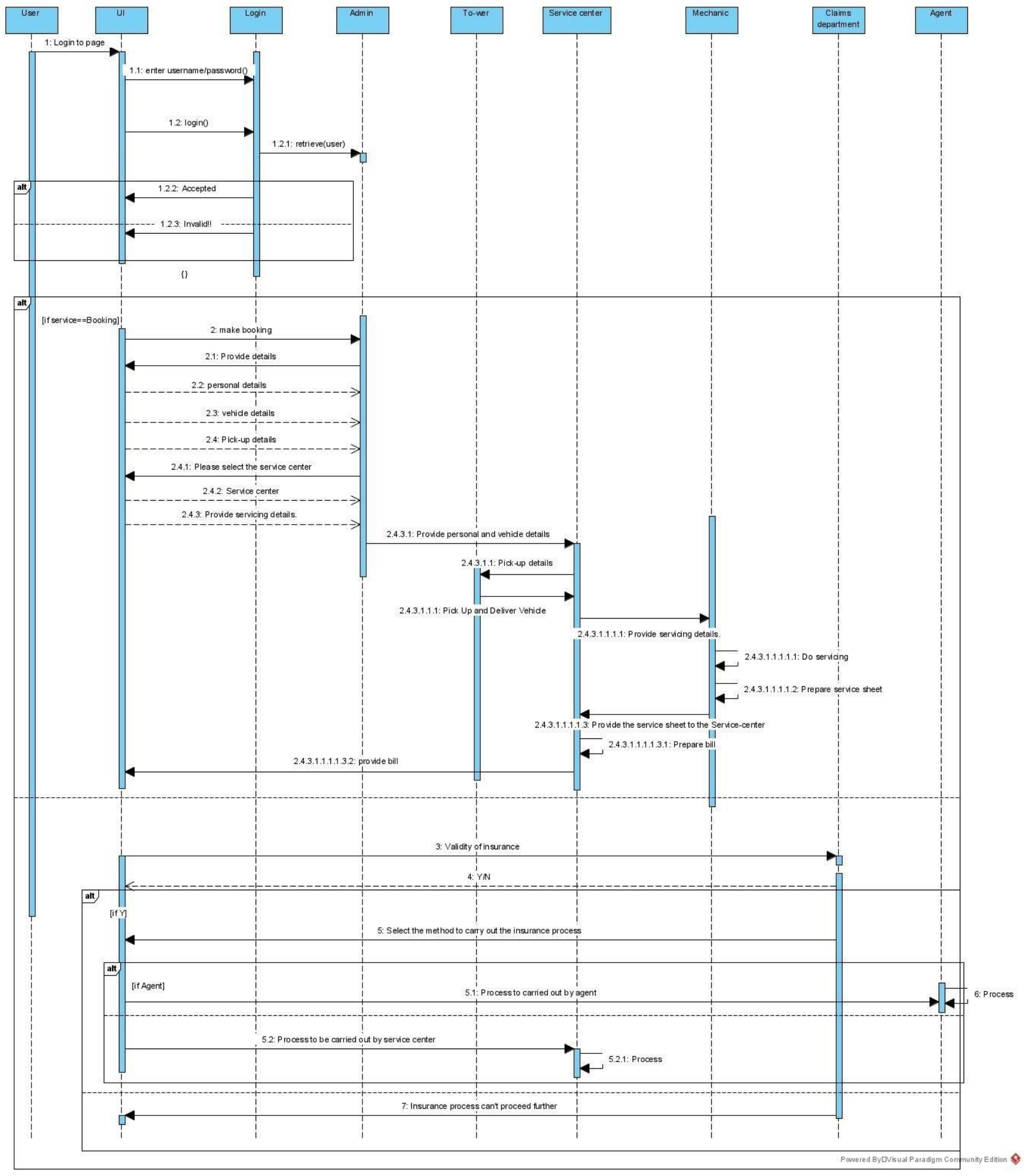
After successfully adding car details, user can book an appointment as the next step. He provides his current location and may either select nearest service center locator or select a center manually. The he enters other necessary in order to book appointment successfully.

Then the service center receives the booking and its next task is to process the booking. A mechanic is the assigned who repairs the vehicle and updates the service sheet in which is sent to the service center which to process the bill .

So in this flowchart user can reach end point by opting any of the options under the fork node and by opting any of these steps user can end the process.

NAME – SAKSHAM JAIN ID – 201851106 GROUP - 4

Sequence diagram


User
UI
Login
Admin
To-wer
Service center
Mechanic
Claims department
Agent
1: Login to page
1.1: enter username/password()
1.2: login()
1.2.1: retrieve(user)
alt
1.2.2: Accepted
1.2.3: Invalid!!
{}
alt
[if service==Booking]
2: make booking
2.1: Provide details
2.2: personal details
2.3: vehicle details
2.4: Pick-up details
2.4.1: Please select the service center
2.4.2: Service center
2.4.3: Provide servicing details.
2.4.3.1: Provide personal and vehicle details
2.4.3.1.1: Pick-up details
2.4.3.1.1.1: Pick Up and Deliver Vehicle
2.4.3.1.1.1.1: Provide servicing details.
2.4.3.1.1.1.1.1: Do servicing
2.4.3.1.1.1.1.2: Prepare service sheet
2.4.3.1.1.1.1.3: Provide the service sheet to the Service-center
2.4.3.1.1.1.1.3.1: Prepare bill
2.4.3.1.1.1.1.3.2: provide bill
3: Validity of insurance
4: Y/N
alt
[if Y]
5: Select the method to carry out the insurance process
alt
[if Agent]
5.1: Process to carried out by agent
6: Process
5.2: Process to be carried out by service center
5.2.1: Process
7: Insurance process can't proceed further
Powered By Visual Paradigm Community Edition

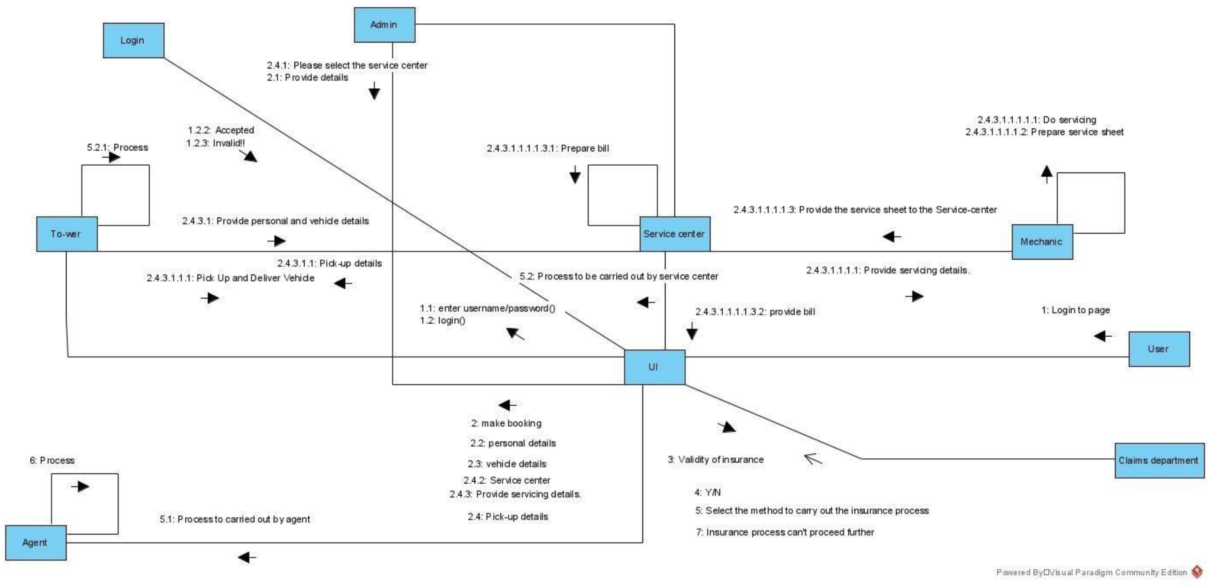
A sequence diagram shows the interaction between objects in a sequential order. The horizontal axis shows the elements that are involved in the interaction. The vertical axis represents time proceedings. The thin blue rectangle on the vertical axis shows the period when the entity is active.

Simple arrows are the call messages whereas the arrow with dashes are return messages. The block in the diagram represents the if-else condition. If the condition is True then the order of interactions is as that of the upper block else it is of the below block.

Here the system provides two facilities i.e. servicing and insurance. So they are shown in two separate blocks. Servicing and preparing service sheet is the task of the mechanic itself, hence it is shown by the self-message.

If the user selects the insurance option, then first the validity of the insurance is checked. Here, there could be two conditions i.e insurance can be valid as well as invalid .If the valid, then all the steps of the upper block are followed in the sequential manner. There is a nested block showing that the user has 2 options i.e to get insurance process done by the agent or by the service center.

Communication diagram

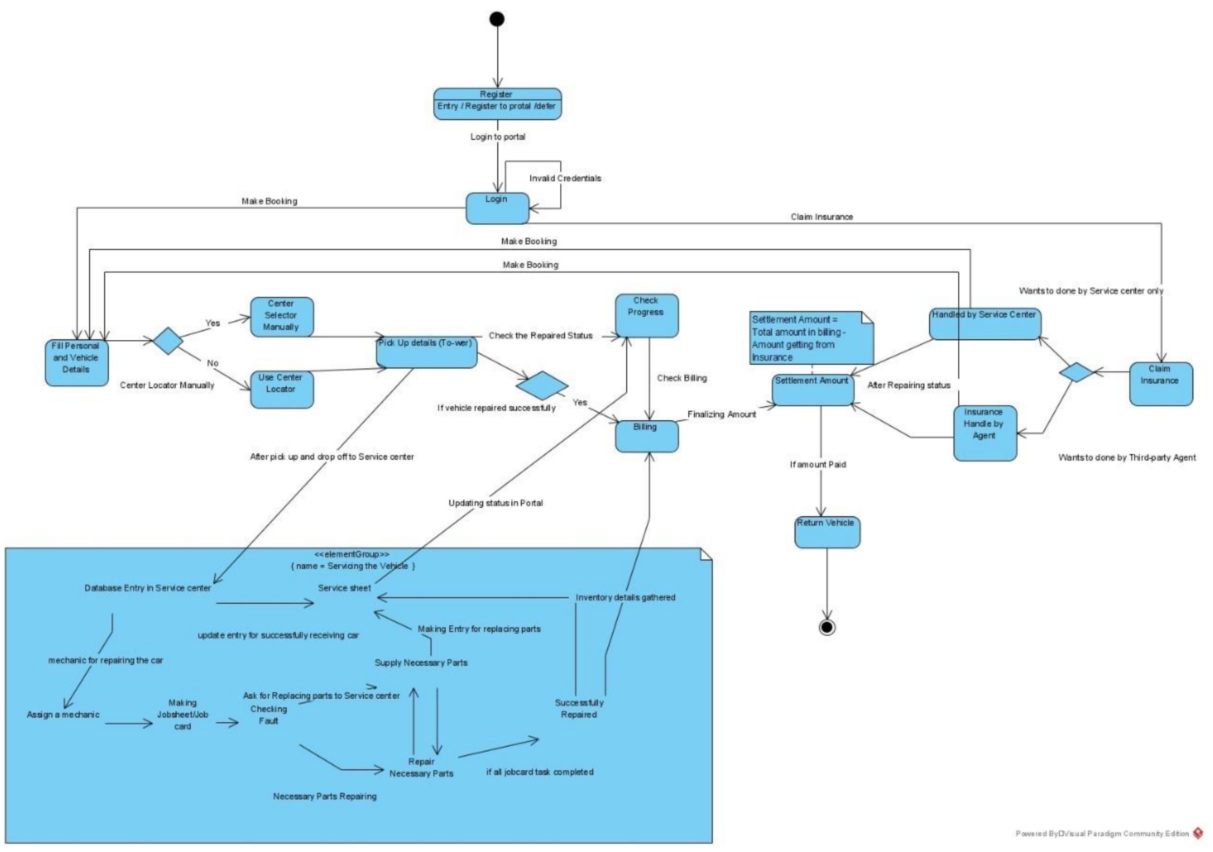

Login
Admin
2.4.1: Please select the service center
2.1: Provide details
1.2.2: Accepted
1.2.3: Invalid!!
5.2.1: Process
2.4.3.1.1.1.1.3.1: Prepare bill
2.4.3.1.1.1.1.1: Do servicing
2.4.3.1.1.1.1.2: Prepare service sheet
To-wer
2.4.3.1: Provide personal and vehicle details
Service center
2.4.3.1.1.1.1.3: Provide the service sheet to the Service-center
Mechanic
2.4.3.1.1: Pick-up details
2.4.3.1.1.1: Pick Up and Deliver Vehicle
5.2: Process to be carried out by service center
2.4.3.1.1.1.1: Provide servicing details.
1.1: enter username/password()
1.2: login()
2.4.3.1.1.1.1.3.2: provide bill
1: Login to page
UI
User
2: make booking
2.2: personal details
2.3: vehicle details
2.4.2: Service center
2.4.3: Provide servicing details.
2.4: Pick-up details
6: Process
3: Validity of insurance
Claims department
4: Y/N
5: Select the method to carry out the insurance process
7: Insurance process can't proceed further
Agent
5.1: Process to carried out by agent
Powered By Visual Paradigm Community Edition

Communication Diagram is a kind of interaction diagram which shows how objects interact. It is similar to the Sequence diagram. The only difference is that sequence diagram emphasizes more on the time ordering of the messages whereas the communication diagram emphasizes more on the structural relationships among the objects interacting.

Messages passed between objects are represented by labelled arrows while the messages that objects send to themselves are indicated by loops. For example - the agent and the service center does the process itself and hence the message is indicated as a loop. Messages are composed of message text prefixed by a sequence number. This sequence number indicates the time-ordering of the message.

<u>State Machine Diagram</u>

State machine diagram is used to describe state-dependent behaviour for an object. **An object responds differently to the same event depending on what state it is in**. State machine diagrams are usually applied to objects but can be applied to any element that has behaviour to other entities such as: actors, use cases, methods, subsystems systems and etc. and they are typically used in conjunction with interaction diagrams (usually sequence diagrams). It also helps to determine that the present state is feasible to achieve through some path (from Initial and Final state).


Register
Entry / Register to protal /defer
Login to portal
Invalid Credentials
Login
Make Booking
Claim Insurance
Make Booking
Make Booking
Fill Personal and Vehicle Details
Yes
No
Center Selector Manually
Use Center Locator
Center Locator Manually
Pick Up details (To-wer)
Check the Repaired Status
Check Progress
Check Billing
If vehicle repaired successfully
Yes
Billing
Finalizing Amount
Settlement Amount = Total amount in billing - Amount getting from Insurance
Settlement Amount
After Repairing status
Handled by Service Center
Wants to done by Service center only
Claim Insurance
Insurance Handle by Agent
Wants to done by Third-party Agent
If amount Paid
Return Vehicle
After pick up and drop off to Service center
Updating status in Portal
<<elementGroup>>
{ name = Servicing the Vehicle }
Database Entry in Service center
Service sheet
Inventory details gathered
update entry for successfully receiving car
Making Entry for replacing parts
mechanic for repairing the car
Supply Necessary Parts
Ask for Replacing parts to Service center
Assign a mechanic
Making Jobsheet/Job card
Checking Fault
Successfully Repaired
Repair Necessary Parts
if all jobcard task completed
Necessary Parts Repairing
Powered By Visual Paradigm Community Edition

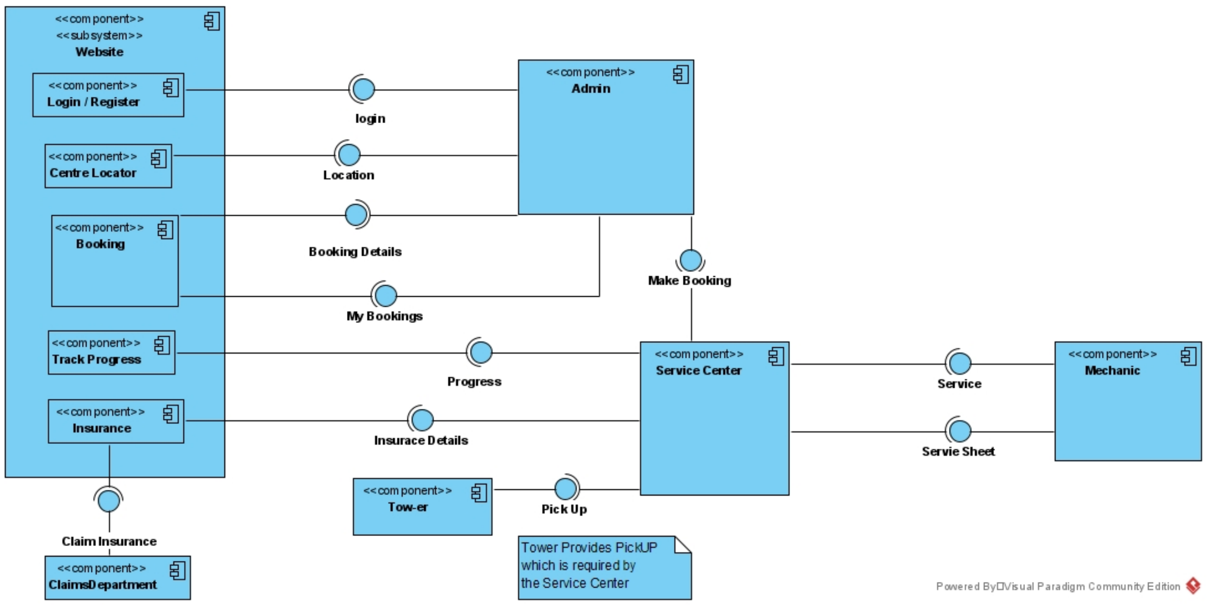
Main states: -

1. Login:
    a. Incoming Transitions [Entry]
        i. through registering in portal
    b. Outgoing Transitions [Exit]

i. In same state if Invalid Credentials
ii. Can make booking
iii. Can also go for Claim Insurance

2. Center Locator Manually (Decision state)
    a. Incoming Transitions
        i. After filling up personal and vehicle details
    b. Outgoing Transitions
        i. If manually then Center Selector manually
        ii. Else use the Center Locator API feature
3. Return Vehicle (Final state)
    a. Incoming Transitions
        i. If amount paid
    b. Outgoing Transitions
        i. End state

Component Diagram


<<component>> <<subsystem>> Website
<<component>> Login / Register
<<component>> Centre Locator
<<component>> Booking
<<component>> Track Progress
<<component>> Insurance
login
Location
Booking Details
My Bookings
Progress
Insurace Details
Claim Insurance
<<component>> ClaimsDepartment
<<component>> Admin
Make Booking
<<component>> Service Center
Service
Servie Sheet
<<component>> Mechanic
<<component>> Tow-er
Pick Up
Tower Provides PickUP which is required by the Service Center
Powered By Visual Paradigm Community Edition

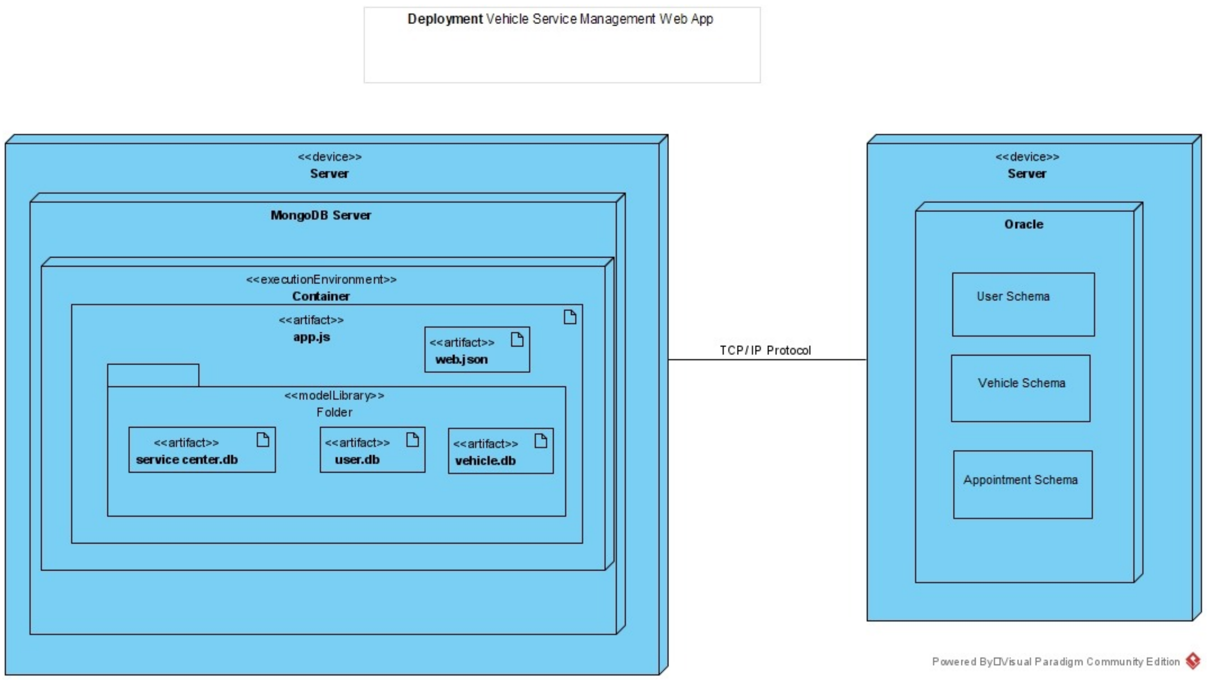
A component diagram breaks down the actual system under development into various high levels of functionality. Each component is responsible for one clear aim within the entire system and only interacts with other essential elements on a need-to-know basis.

Deployment Diagram

Deployment diagram is a part of physical view. It is UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Specification level deployment diagram shows overview of deployment of artifacts to deployment targets, without referencing specific instances of artifacts or nodes.

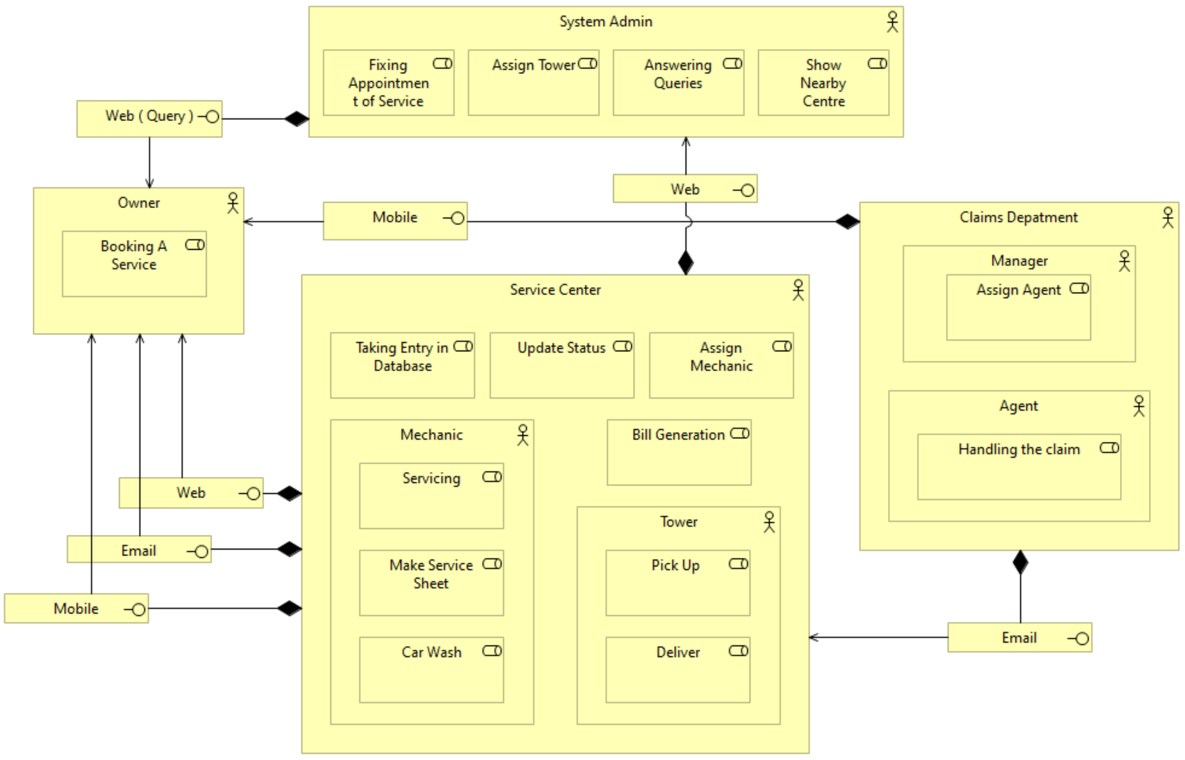


Here we have 2 servers, one server which contains all the artifacts and contain actual execution environment with all frontend files and database files, another is the database server from which we can interact through our database for fetch, update and delete the data of schemas.

Both the servers are interacting / communicating through the TCP/IP protocol.

ArchiMate Diagrams

Organisation Viewpoint (Business Layer)



Organization viewpoint is a part of Business Layer which focuses on the (internal) organization of a company, or an organizational entity.

It shows the structure of an enterprise in terms of roles, departments, etc. Models in this viewpoint can be presented as nested block diagrams. This viewpoint is very useful in identifying competencies, authority, and responsibilities in an organization.

Various actors are identified in order to contrast the viewpoint and are referred as a business actor - an entity that is capable of performing some behavior.

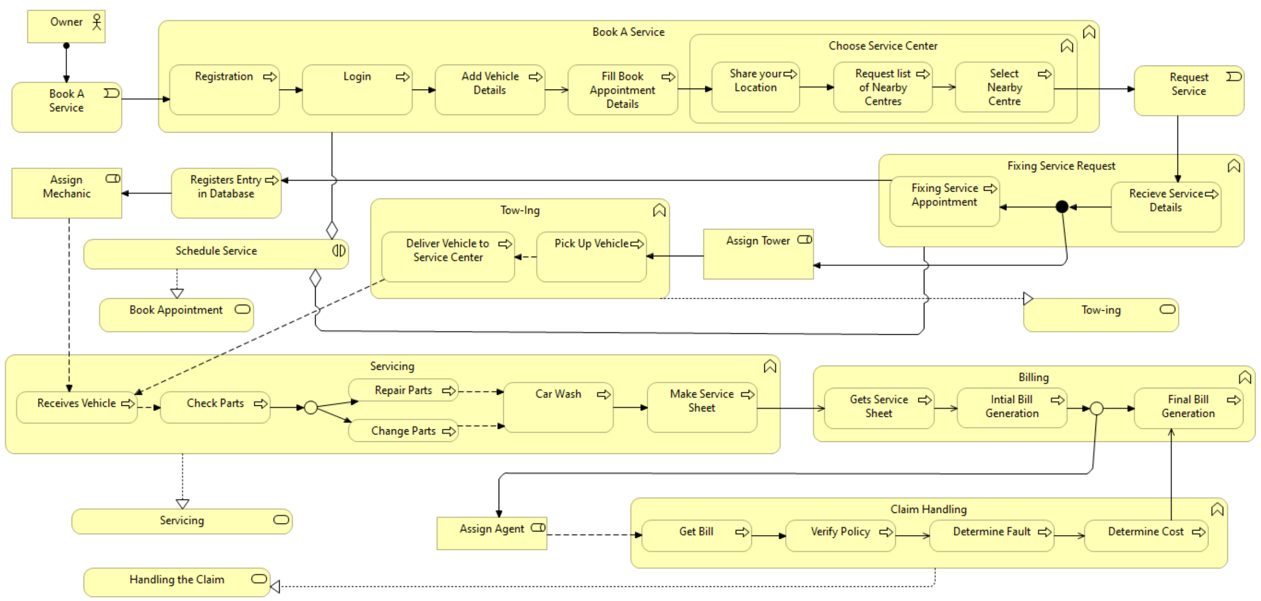
Each of these actors have either single or multiple roles that represents the responsibility for performing some specific behaviour. Roles are connected with Actor by Assignment Relation.

An Actor can be within an actor (as organization) depicting that he is the part of that entity and they are connected by Aggregation Relation showing the dependability.

Finally all the actors as independent actors are connected by several different business interfaces like web , email and mobile (also known as channels) that represent the point of access where a business service is made available to the environment.

NAME – SAKSHAM JAIN ID – 201851106 GROUP - 4

Business Process Cooperation Viewpoint (Business Layer)



Business process cooperation viewpoint is a part of Business Layer which is used to show the relationships of one or more business processes with each other or within their environment. It can be used both to create a high-level design of business processes within their context and to provide an operational manager responsible for one or more such processes with insight into their dependencies.

Initially Owner triggers Book a service (Event) which further triggers the Function– Book a Service. A set of process are included in this function which takes place sequentially by triggering the next process in the sequence. Then , Request Service Event happens which triggers the Fix Request and Towing Function. Database entry registration process takes place and a mechanic is assigned.

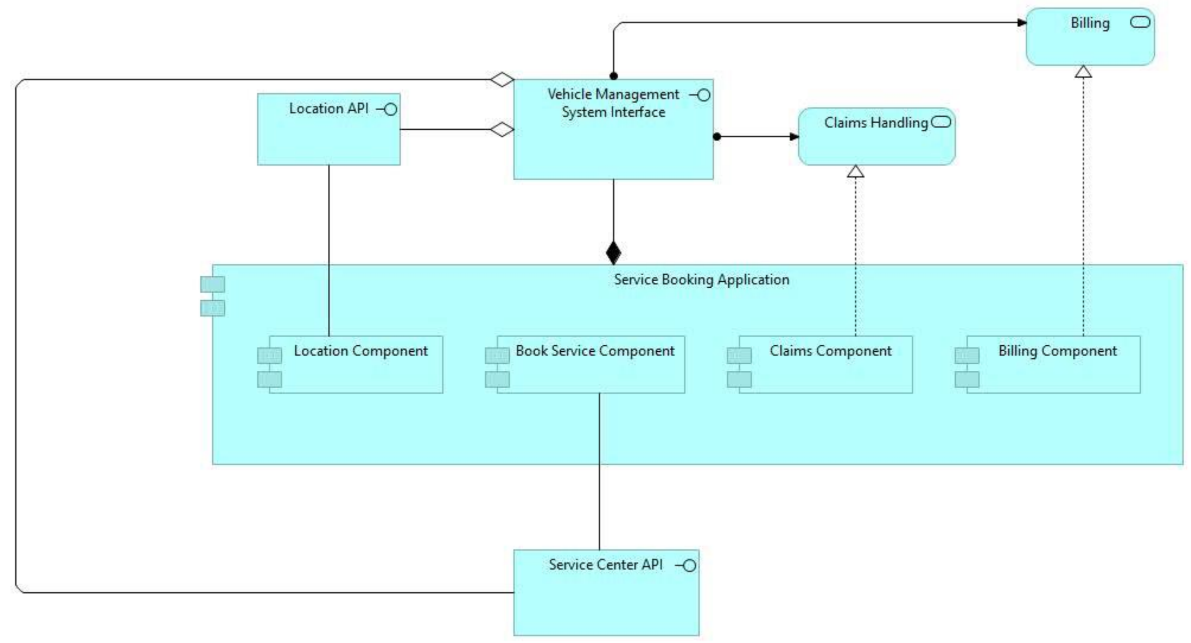
Schedule service is a Collaboration that represents a unit of collective business behaviour performed by the constituting 2 functions. Book Appointment is a service which is realized by this collaboration.

Further the flow goes to Servicing Function where various processes occur i.e. Checking and Repairing of Vehicle , Service sheet generation etc. Servicing as a business service realizes this function.

Next , Billing Function is executed. Claims handling function can come into this flow in case the vehicle service is due to some accident hence an OR junction is provided after Initial Bill Generation as if this is the case first an agent is assigned for the Claim Handling process after which the Final Bill is generated.

NOTE – All the process within a function have an aggregation relation with the function.

Application Cooperation Viewpoint



Application cooperation viewpoint describes the relationships between application components in terms of the information flowing between them, or in terms of the services they offer and use .It is used to create an overview of the application landscape of an organization.

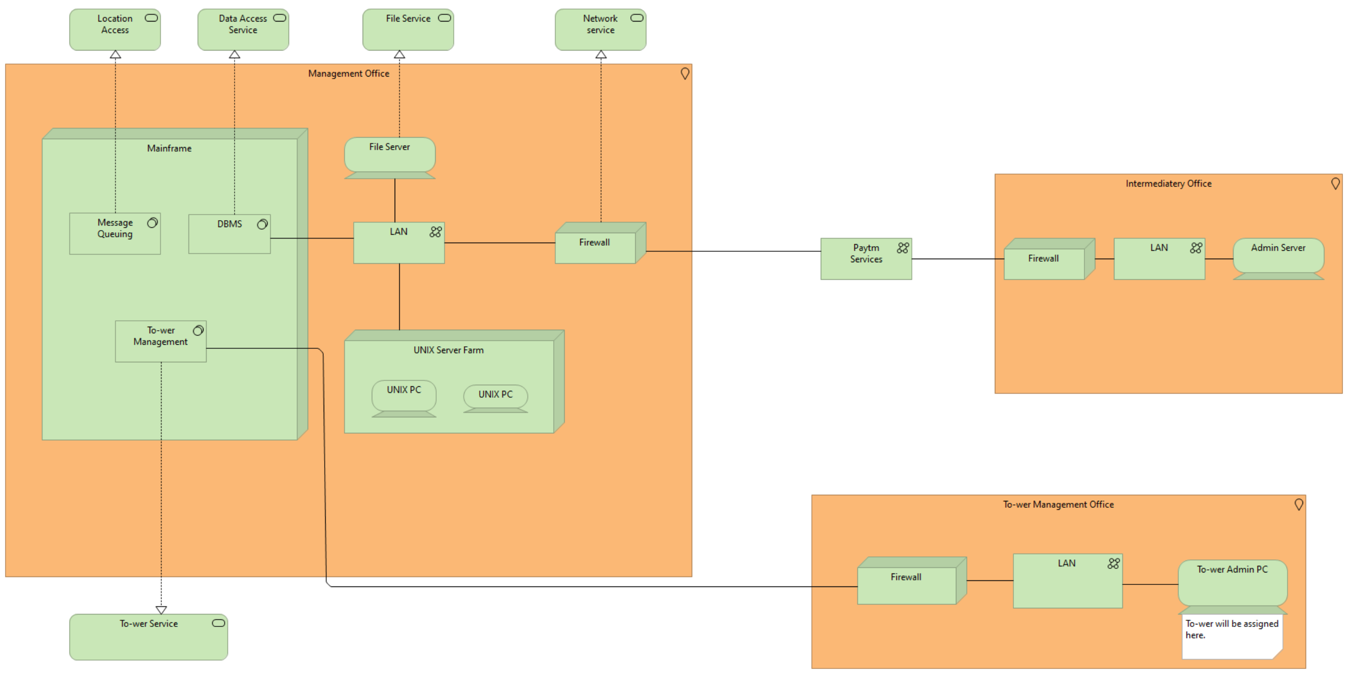
The main application component here is Service Booking Application which consists of many other small components as its parts like location component, book service component, claims component and billing component.

These four components determine working of service booking application. Now next thing which we depict using application cooperation view is how all these component connect with the application interface to make it fully operational application.

In order to connect location component to interface we use location which tells us the current location of the user. Similarly book service component is connected to interface via service center API that tells us nearest service center to user current location.

Claims component is connected to interface via claims handling service which manages all the claims handling part needed by the user. And similarly billing component is connected to interface via billing service which basically generates final service bill for the user.

Technology Viewpoint



It shows how application layer is supported by software and hardware technology elements like Physical devices, networks, or system software (e.g., O/S, databases, and middleware).

Here departments of company are distributed over different locations. We have 3 different locations which can be used for accessing resources which is shown in orange background.

1. **Management office**

   It is the server office, which can be deployed using public or private cloud. In office we have mainly divide into three nodes.

a. **Mainframe**

   It is providing access for all necessary services. Message Queuing service software which can be load balancer in technical term, which helps to minimize the traffic on server. DBMS system software is database management system, which is used for storing the data in efficient manner. Tower management is a system software, which handles the service of allocating the tower on the basis of user demand. Location access service is the API which will be linked to the external API for fetching the current location of user.

b. **File server**

   It is a separate sever which is specially made for accessing important file which can be redundant of actual database, in case of failure.

c. **UNIX Server Farm**

   It contains actual server which will manage the mainframe and file server through UNIX systems.

All the servers are inter-connected through the LAN for communicating purpose.

And a firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. A firewall typically establishes a barrier between a trusted network and an untrusted network, such as the Internet.

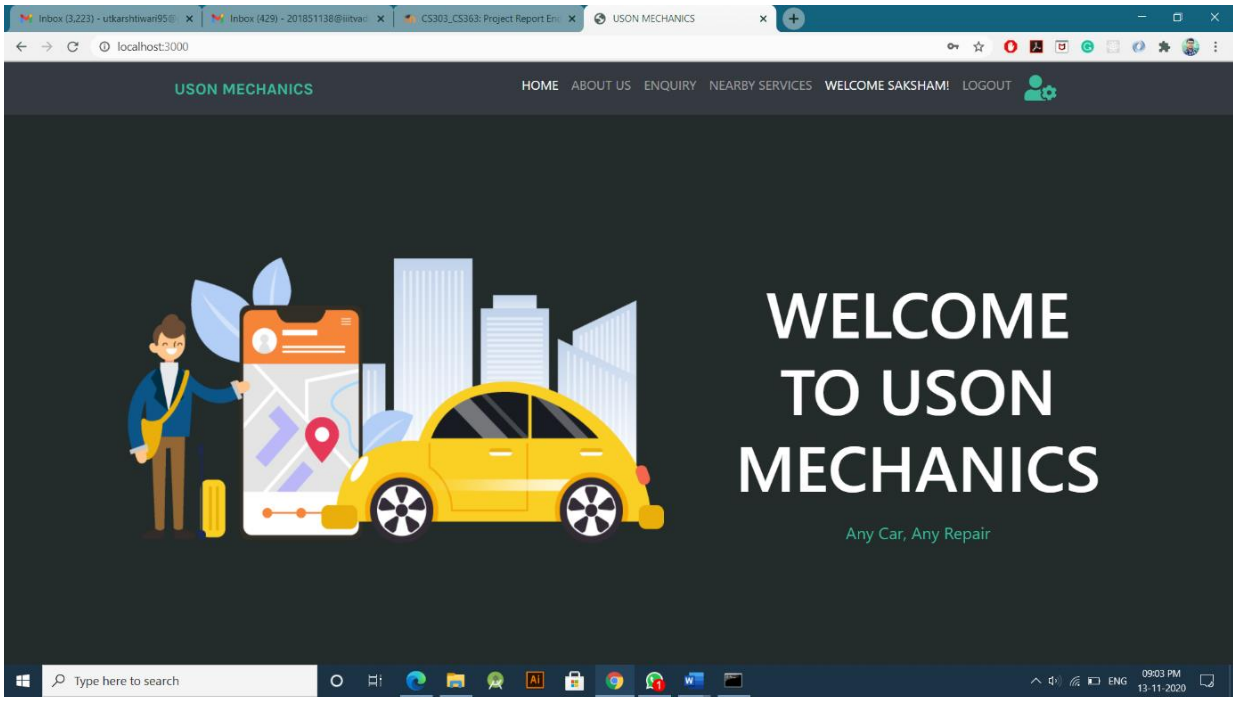
2. **Intermediary office**
   An intermediary office is a third-party office that offers intermediation services between two parties, which involves conveying messages between principals in a dispute, preventing direct contact and potential escalation of the issue.

3. **To-wer management office**

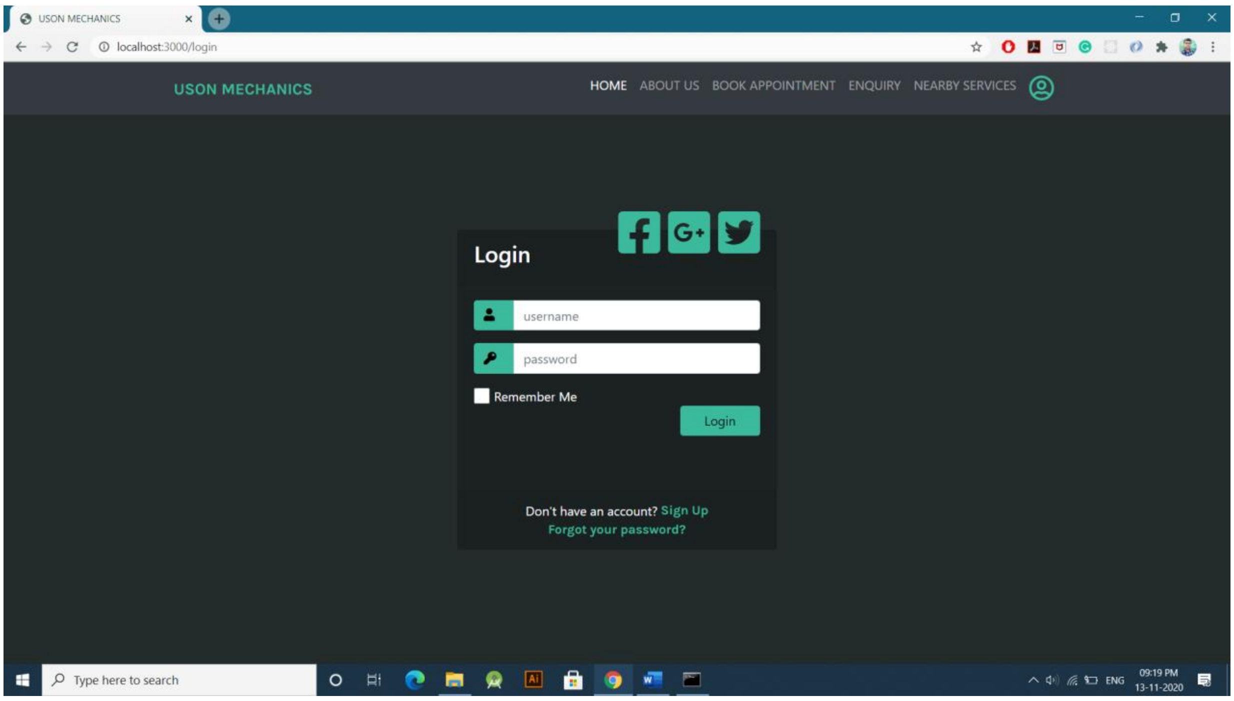
   To-wer management office is a tower-management system software linked office which manages when and how to assign the towers in an efficient manner.

NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**

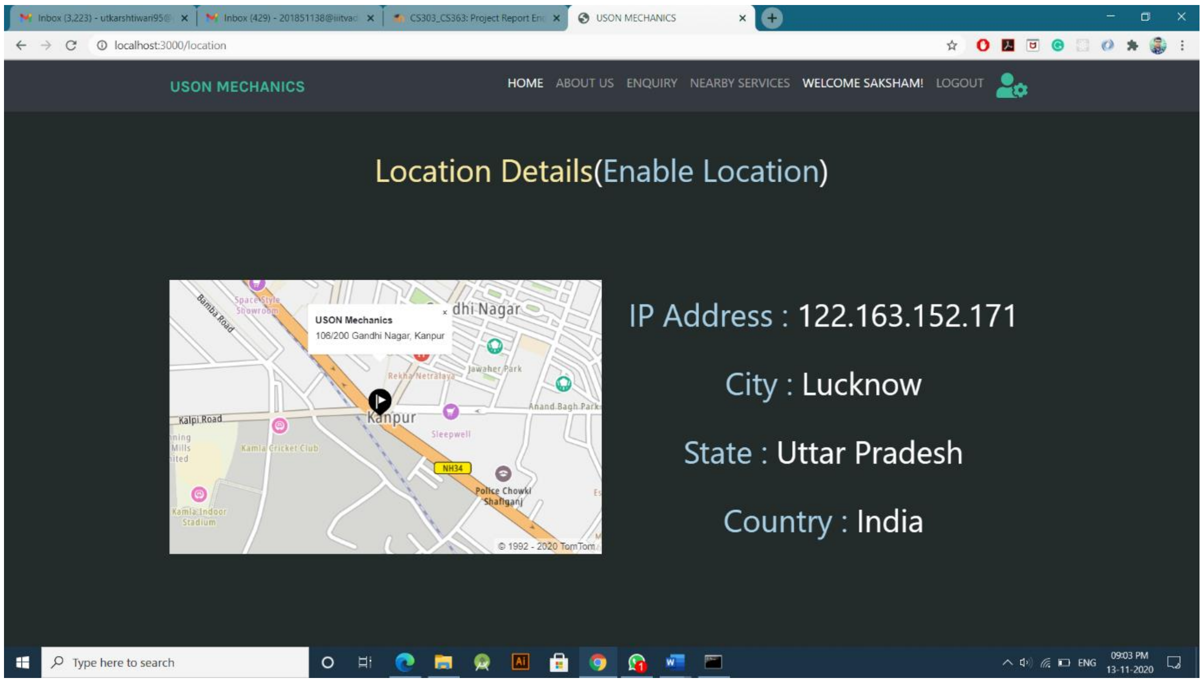
Website Modules



The page shown above is our homepage of the website.



Login PAGE


USON MECHANICS
HOME ABOUT US ENQUIRY NEARBY SERVICES WELCOME SAKSHAM! LOGOUT
USON Mechanics
106/200 Gandhi Nagar, Kanpur

Location Details(Enable Location)

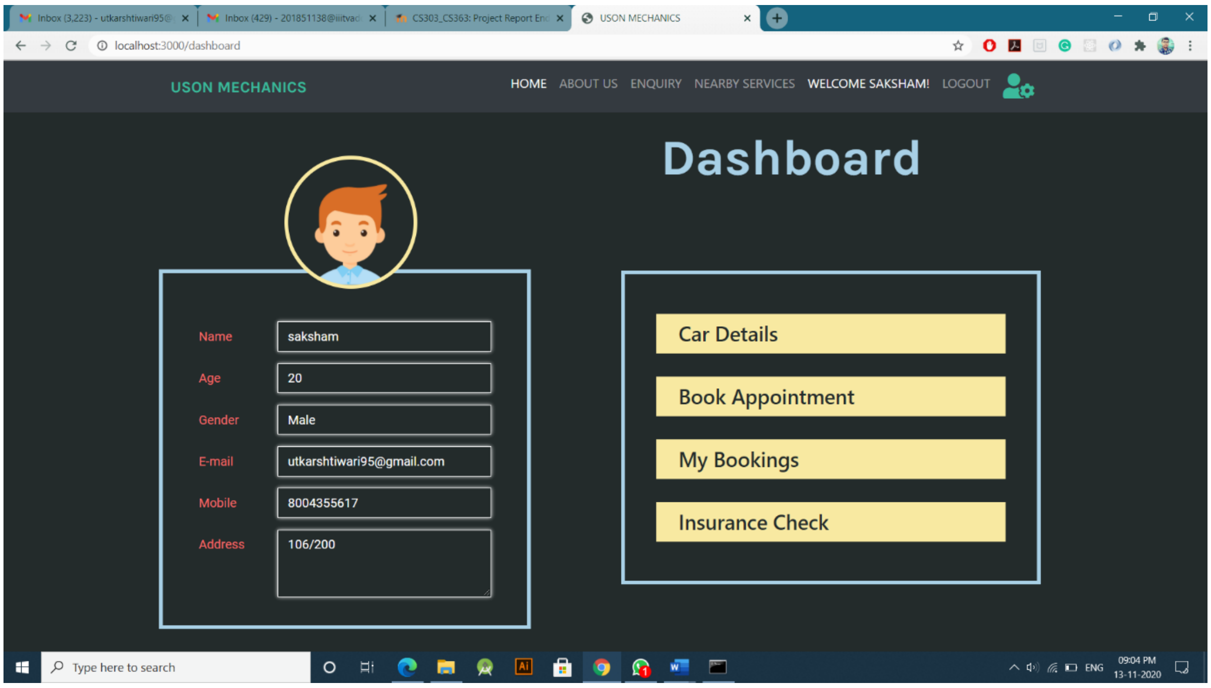
IP Address : 122.163.152.171

City : Lucknow

State : Uttar Pradesh

Country : India

Next module we created is locate your nearest service centre page. This page generates user IP ADDRESS through an API and the data through here is taken and given as a input to another API using geolocation and gives us all the details about user city and state.

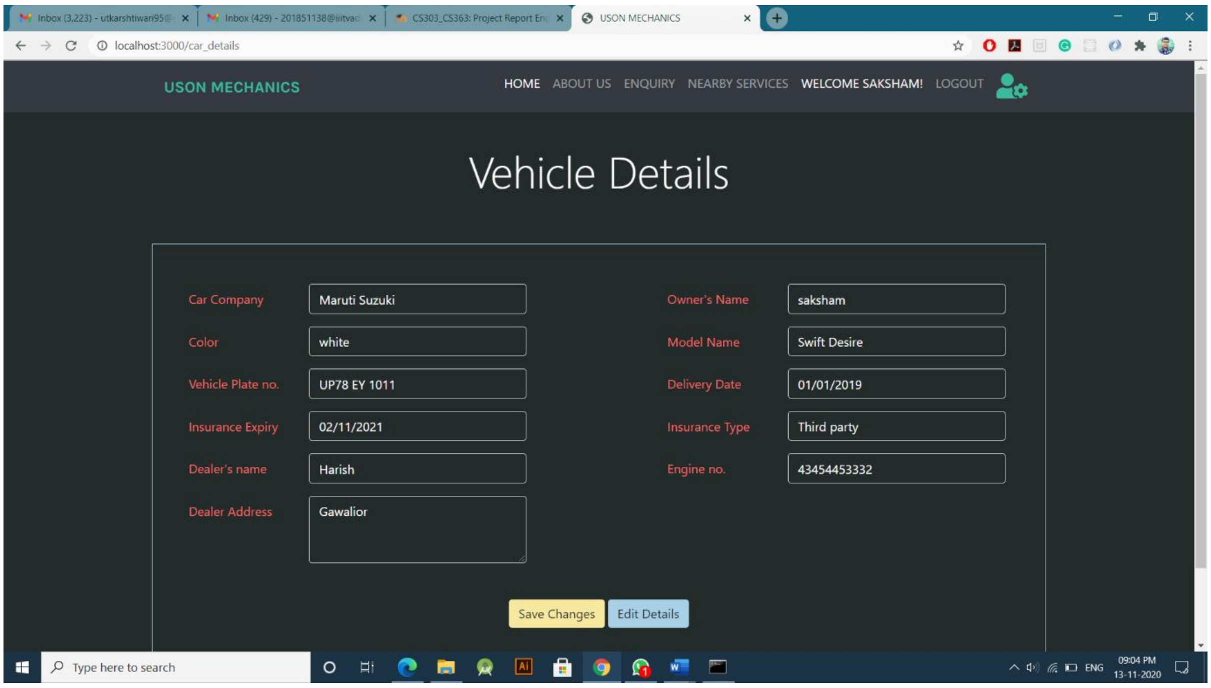


Dashboard

Name	saksham
Age	20
Gender	Male
E-mail	utkarshtiwari95@gmail.com
Mobile	8004355617
Address	106/200

The page shown above is dashboard module of our application, it gives different parallel options to user like user can view car details, or book an appointment, and later can view it in my bookings section, and at last can check the validity of their insurance.

NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**

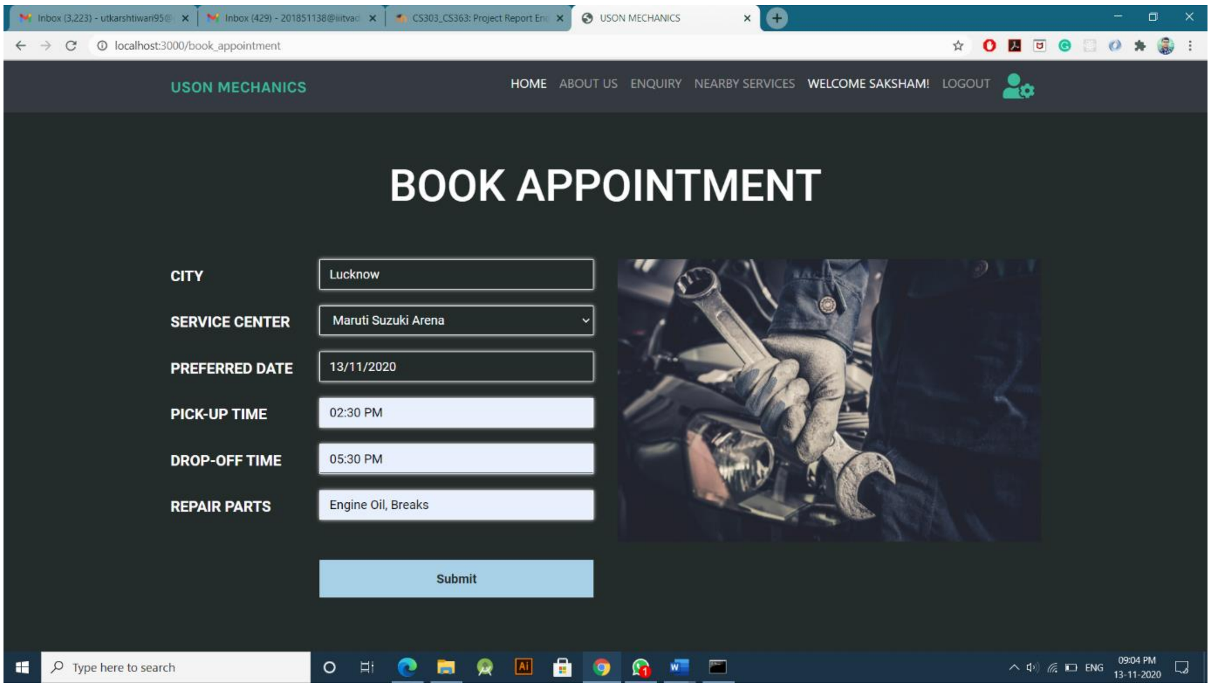


localhost:3000/car_details

Vehicle Details

Car Company	Maruti Suzuki	Owner's Name	saksham
Color	white	Model Name	Swift Desire
Vehicle Plate no.	UP78 EY 1011	Delivery Date	01/01/2019
Insurance Expiry	02/11/2021	Insurance Type	Third party
Dealer's name	Harish	Engine no.	43454453332
Dealer Address	Gawalior		

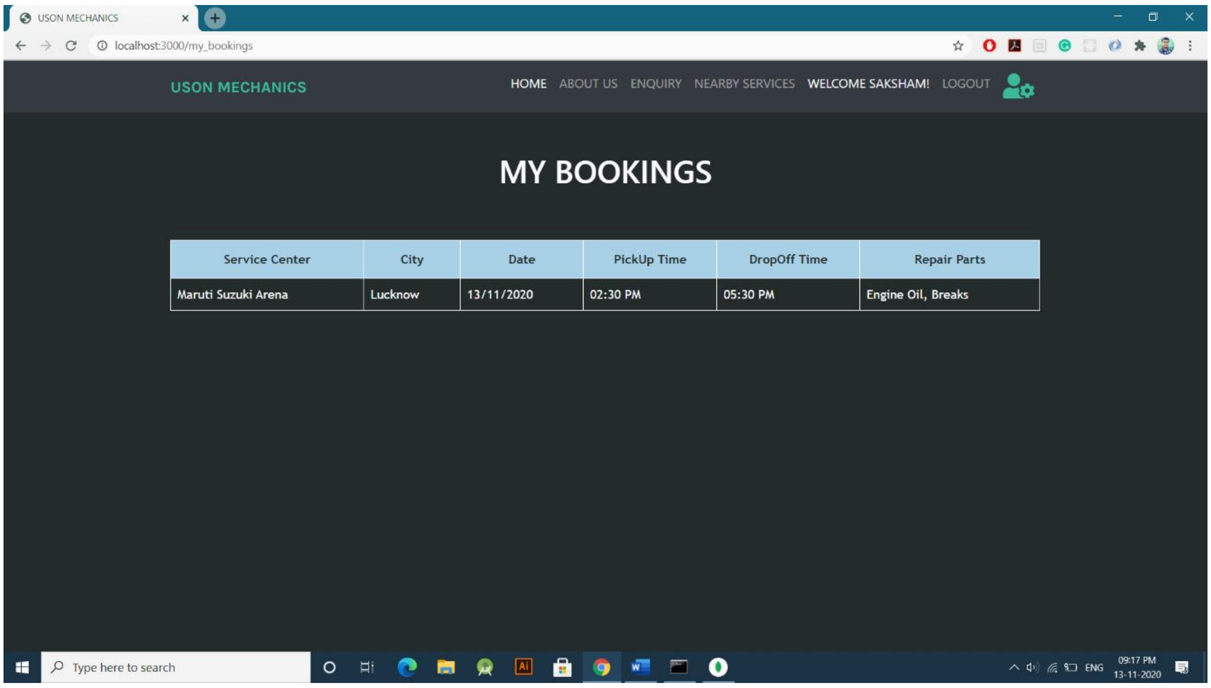
In Book appointment module city is autofilled when user open nearest service center module, users current city is shown here and service centers present in that area are also listed to the user.



localhost:3000/book_appointment

BOOK APPOINTMENT

CITY	Lucknow
SERVICE CENTER	Maruti Suzuki Arena
PREFERRED DATE	13/11/2020
PICK-UP TIME	02:30 PM
DROP-OFF TIME	05:30 PM
REPAIR PARTS	Engine Oil, Breaks


USON MECHANICS
localhost:3000/my_bookings
USON MECHANICS
HOME ABOUT US ENQUIRY NEARBY SERVICES WELCOME SAKSHAM! LOGOUT
MY BOOKINGS

Service Center	City	Date	PickUp Time	DropOff Time	Repair Parts
Maruti Suzuki Arena	Lucknow	13/11/2020	02:30 PM	05:30 PM	Engine Oil, Breaks

The above module lists all the recent appointment made by the user along with the pickup and drop off time selected by the user.

Selenium Web Testing

We prepared 3 scripts in JAVA (Eclipse IDE) for Automated web testing of Our Mini Project.

(NOTE - Thread.sleep() is used in each line of code so that changes are visible for some time)

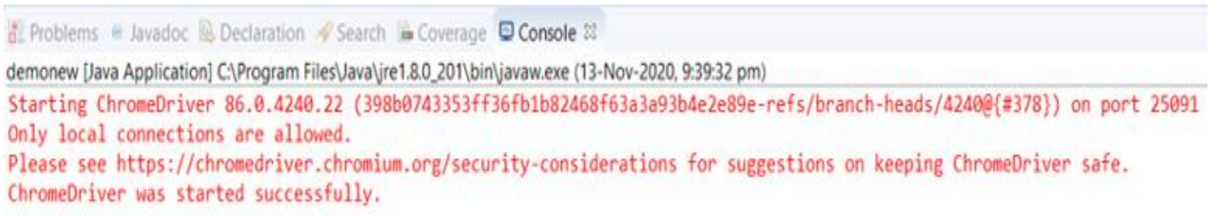
1). Registration Test – When we run the following Script a new user registration takes place on our web application.

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.concurrent.TimeUnit;
public class demonew //Registration Automated TEST
{
   public static void main(String[] args) throws InterruptedException
   {
      // declaration and instantiation of objects/variables
      System.setProperty("webdriver.chrome.driver", "G:\\chromedriver.exe");
      WebDriver driver=new ChromeDriver();

      driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
            // In Case of Exception , show it with delay of 5 seconds in terminal

      driver.navigate().to("http://localhost:3000/");
            // navigate to homepage
```

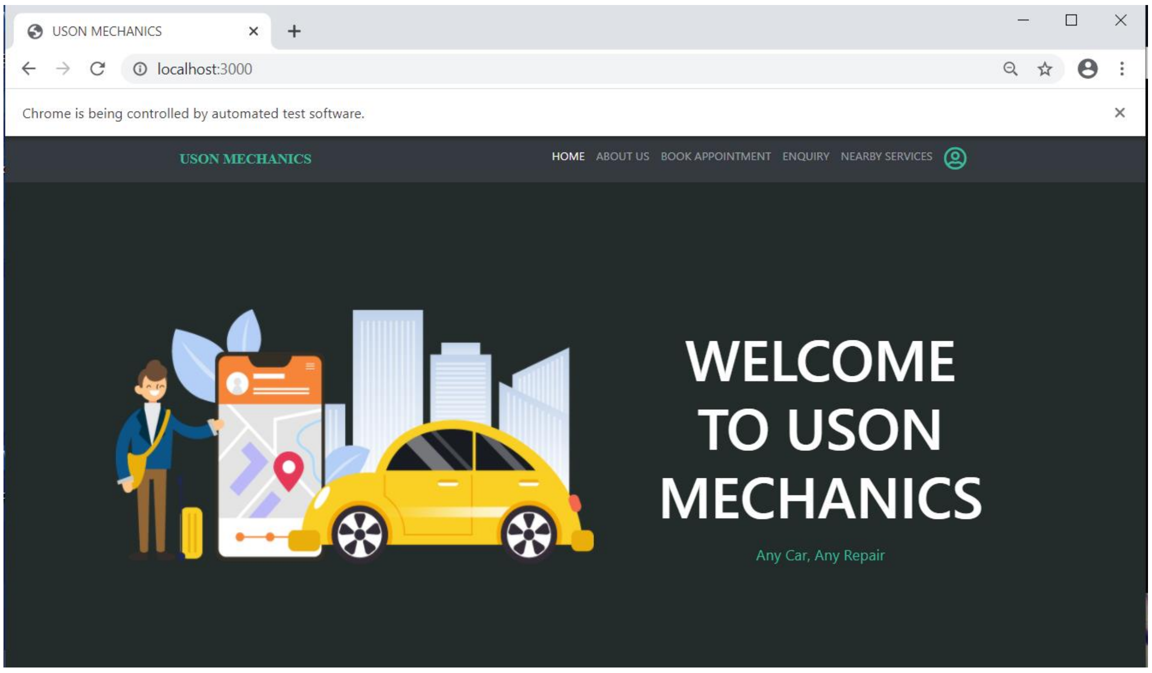
```
        Thread.sleep(2000);
        driver.manage().window().maximize(); // maximize the window
        Thread.sleep(2000);
        driver.findElement(By.id("login1")).click();
            // Click on User Icon
        Thread.sleep(2000);
        driver.findElement(By.name("signup1")).click();
            // For new user Registration click on Sign Up and Enter all the
            Required Fields
        Thread.sleep(2000);
        driver.findElement(By.name("username")).sendKeys("saksham1099");
        Thread.sleep(2000);
        driver.findElement(By.name("password")).sendKeys("mkonjibhu");
        Thread.sleep(2000);
        driver.findElement(By.name("email")).sendKeys("saksham909@gmail.com");
        Thread.sleep(2000);
        driver.findElement(By.name("age")).sendKeys("21");
        Thread.sleep(2000);
        driver.findElement(By.name("gender")).sendKeys("male");
        Thread.sleep(2000);
        driver.findElement(By.name("mobile")).sendKeys("9425307306");
        Thread.sleep(2000);
        driver.findElement(By.name("address")).sendKeys("Windsor Hills");
        Thread.sleep(2000);
        driver.findElement(By.id("register1")).click();
            // After entering all Values Click Register
        Thread.sleep(2000);
        driver.findElement(By.id("d1")).click(); // go to Dashboard
        Thread.sleep(20000);
        driver.close();
            // Registration will be done and user will be logged in then - close
            the window after 20 seconds
    }
}
```

As we have done the automated testing using Chrome Web Driver – a new tab opens in chrome and the process in the above code takes place in sequencial order.

Output in the Terminal



```
demonew [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (13-Nov-2020, 9:39:32 pm)
Starting ChromeDriver 86.0.4240.22 (398b0743353ff36fb1b82468f63a3a93b4e2e89e-refs/branch-heads/4240@{#378}) on port 25091
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
```

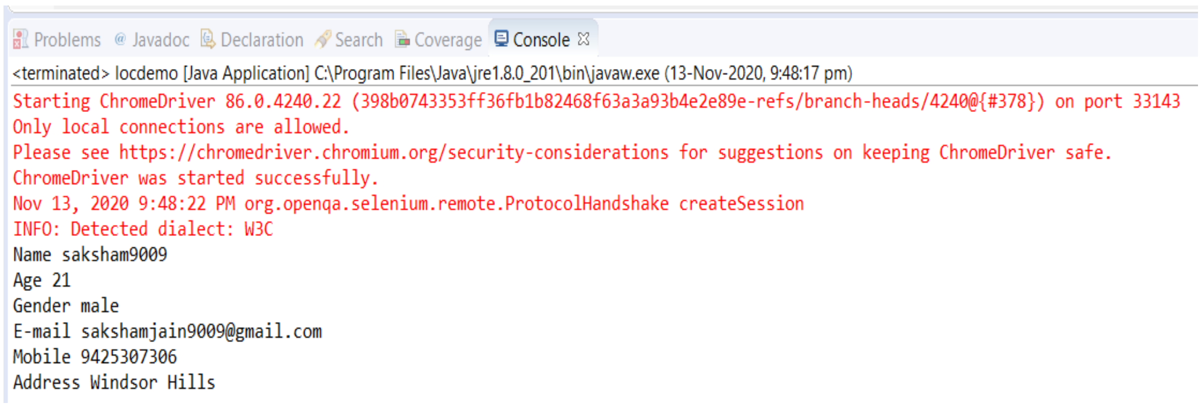


2). **Location and Details Test** – When we run the following Script the Location details of the Logged -In User are fetched using a Location Detector API . Vehicle Details are Displayed on the browser and User Personal Details are listed on the terminal.

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import java.util.List;
import java.util.concurrent.TimeUnit;
public class locdemo // LOCATION AND DETAILS AUTOMATED TEST
{
  public static void main(String[] args) throws InterruptedException
  {
    // declaration and instantiation of objects/variables
    System.setProperty("webdriver.chrome.driver", "G:\\chromedriver.exe");
    WebDriver driver=new ChromeDriver();

    driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
        // In Case of Exception , show it with delay of 5 seconds in terminal

    driver.navigate().to("http://localhost:3000/");// navigate to homepage
    Thread.sleep(2000);
    driver.manage().window().maximize(); // maximize the window
    Thread.sleep(2000);
    driver.findElement(By.id("login1")).click();
        // Click on User Icon to login and give details
    Thread.sleep(2000);
    driver.findElement(By.name("username")).sendKeys("saksham9009");
```

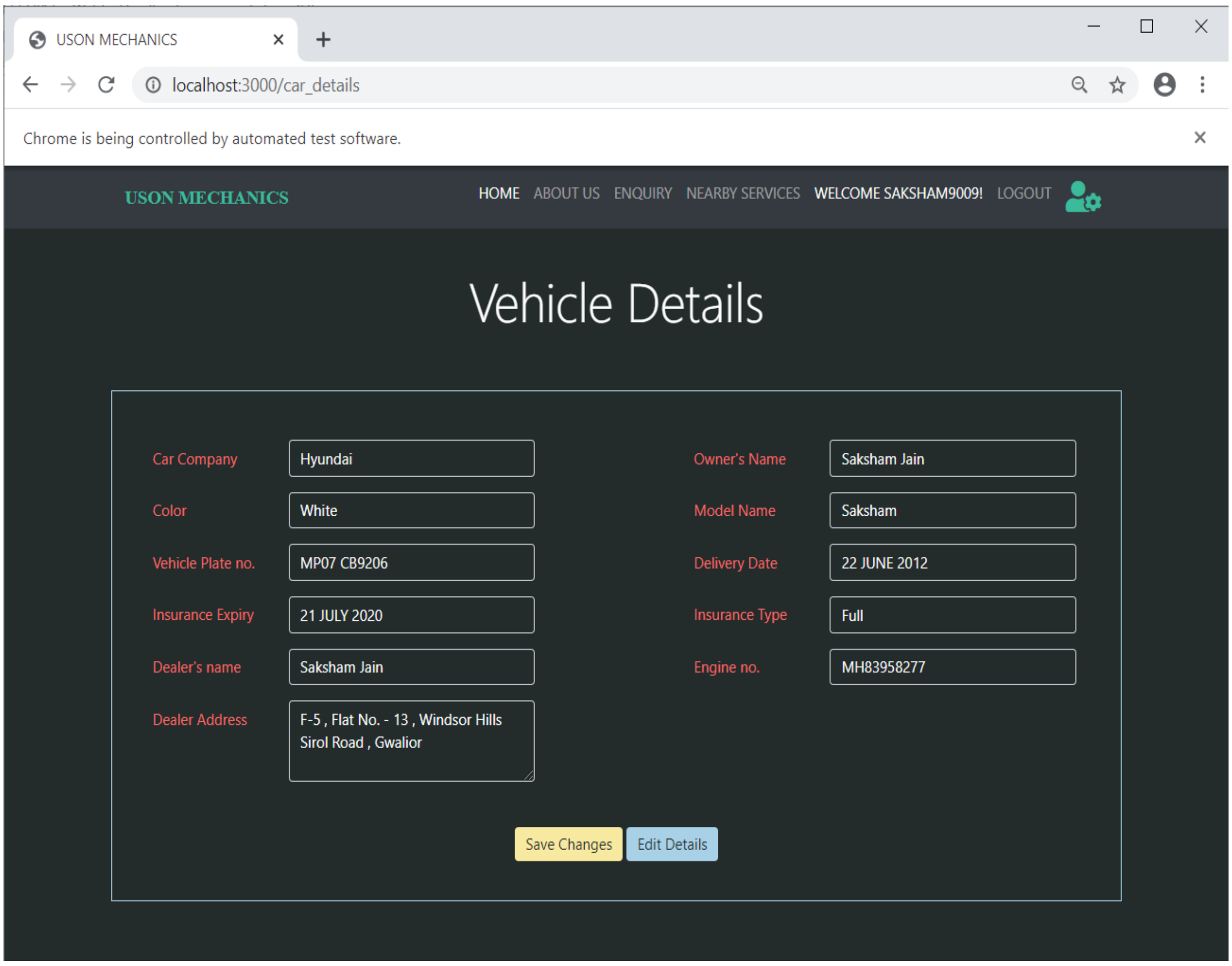
```
        Thread.sleep(2000);
        driver.findElement(By.name("password")).sendKeys("mkonjibhu");
        Thread.sleep(2000);
        driver.findElement(By.id("clilog")).click(); // Click on login
        Thread.sleep(2000);
        driver.findElement(By.id("loc2")).click(); // click on nearby services
        Thread.sleep(2000);
        driver.findElement(By.id("enb")).click(); // Click on enable location
        Thread.sleep(10000);
        driver.findElement(By.id("d1")).click();
            // After knowing your Location return to dashboard
        Thread.sleep(3000);
        driver.findElement(By.id("cardet")).click(); // Check car details
        Thread.sleep(3000);
        driver.navigate().back();
        Thread.sleep(3000);

        List <WebElement> listOfElements = driver.findElements
            (By.xpath("//*[@id=\"enq\"]"));
        // Generate a list for displaying the text and input field on terminal
        List <WebElement> input = driver.findElements
            (By.cssSelector("input[type='text']"));
        for(int i=0; i<input.size();i++)
        {
          System.out.println(listOfElements.get(i).getText()+" "
            +input.get(i).getAttribute("value"));
        }
        Thread.sleep(3000);
        driver.findElement(By.id("out")).click(); // logout
        Thread.sleep(3000);
        driver.close();
    }
}
```

Terminal Output



```
<terminated> locdemo [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (13-Nov-2020, 9:48:17 pm)
Starting ChromeDriver 86.0.4240.22 (398b0743353ff36fb1b82468f63a3a93b4e2e89e-refs/branch-heads/4240@{#378}) on port 33143
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Nov 13, 2020 9:48:22 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Name saksham9009
Age 21
Gender male
E-mail sakshamjain9009@gmail.com
Mobile 9425307306
Address Windsor Hills
```

NAME – SAKSHAM JAIN **ID – 201851106** **GROUP - 4**


USON MECHANICS

localhost:3000/car_details

Chrome is being controlled by automated test software.

USON MECHANICS HOME ABOUT US ENQUIRY NEARBY SERVICES WELCOME SAKSHAM9009! LOGOUT

Vehicle Details

Car Company	Hyundai	Owner's Name	Saksham Jain
Color	White	Model Name	Saksham
Vehicle Plate no.	MP07 CB9206	Delivery Date	22 JUNE 2012
Insurance Expiry	21 JULY 2020	Insurance Type	Full
Dealer's name	Saksham Jain	Engine no.	MH83958277
Dealer Address	F-5 , Flat No. - 13 , Windsor Hills Sirol Road , Gwalior		

3). **Book Appointment Test** – When we run the following Script an appointment is booked of the vehicle of logged in user by selection of the desired centre and giving other details.

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
import java.util.concurrent.TimeUnit;
public class demo3 // BOOKING AUTOMATED TEST
{
   public static void main(String[] args) throws InterruptedException
   {
      // declaration and instantiation of objects/variables
      System.setProperty("webdriver.chrome.driver", "G:\\chromedriver.exe");
      WebDriver driver=new ChromeDriver();

      driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
          // In Case of Exception , show it with delay of 5 seconds in terminal
```

```java
        driver.navigate().to("http://localhost:3000/");// navigate to homepage
        Thread.sleep(2000);
        driver.manage().window().maximize(); // maximize the window
        Thread.sleep(2000);
        driver.findElement(By.id("login1")).click();
            // Click on User Icon to login and give details
        Thread.sleep(2000);
        driver.findElement(By.name("username")).sendKeys("saksham9009");
        Thread.sleep(2000);
        driver.findElement(By.name("password")).sendKeys("mkonjibhq");
        Thread.sleep(2000);
        driver.findElement(By.id("clilog")).click(); // Click on login
        Thread.sleep(2000);
        driver.findElement(By.id("d1")).click(); // Go to dash board
        Thread.sleep(3000);
        driver.findElement(By.id("bookapp")).click();// go to book appointment
        Thread.sleep(3000);

        Select citylist = new Select(driver.findElement
            (By.name("service_center")));
            // City name automatically appea select 3rd element from drop down menu
        citylist.selectByIndex(3);

        driver.findElement(By.name("date")).sendKeys("4/11/2020");
            // enter the other values
        Thread.sleep(2000);
        driver.findElement(By.name("pickup_time")).sendKeys("8:00 AM");
        Thread.sleep(2000);
        driver.findElement(By.name("dropoff_time")).sendKeys("3:00 PM");
        Thread.sleep(2000);
        driver.findElement(By.name("repair_parts")).sendKeys("Engine");
        Thread.sleep(2000);
        driver.findElement(By.name("button")).click();
            // Click Submit to make booking
        Thread.sleep(3000);
        driver.findElement(By.id("mybook")).click();
            // Check the Updated BOOKINGS
        Thread.sleep(20000);
        driver.close(); // close webdriver
    }
}
```