**Abstract**

The Vehicle Service Management System is integral to providing quality customer service. It helps to measure public satisfaction and is a useful source of information and feedback for improving services. User-friendliness is the key aspect of this system and user-friendly GUI is deployed in this system.

There are 2 main modules amalgamated in this system: -

1. User Module
2. Admin/ Supervisor Module

The following objectives have been achieved by implementing this system: -

1) Smooth flow of data without any hurdles.

2) Adequate validation checks for data entry.

3) Adequate security of data.

4) Facility to update data from time to time.

5) Prompt and specific retrieval of data.

6) Flexibility in the system according to the changing environment.

7) Accuracy, timeliness, and comprehensiveness of the system output.

8) Stability and operability by people of average intelligence.

|  |  |  |
| --- | --- | --- |
| **Table of Contents** | | |
| **Chapter no.** | **Topic** | **Page. no.** |
| **1.** | **Chapter 1 - Introduction**  1.1 Objectives and Scope of Project  1.2 Theoretical Background | **10**  **10-11** |
| **2.** | **Chapter 2 – Review of Literature**  2.1 Need of System  2.2 Proposed System | **13**  **13-14** |
| **3.** | **Chapter 3 – System Requirement Analysis**  3.1 Process Analysis  3.2 SRS Methodology Adopted  3.3 System Requirements | **16-20**  **21**  **22** |
| **4.** | **Chapter 4 – System Design**  4.1 Use Case Diagram  4.2 DFD Diagram  4.3 E-R Diagram  4.4 Class Diagram  4.5 Sequence Diagram  4.6 Collaboration Diagram  4.7 Activity Diagram  4.8 State Chart Diagram  4.9 Database Design | **24**  **25-26**  **27**  **28**  **28-29**  **29**  **30**  **31**  **31-33** |
| **5.** | **Chapter 5 – System Development**  5.1 Programme Development  5.2 Testing | **35-36**  **37-39** |
| **6.** | **Chapter 6 – System Implementation**  6.1 Acquisition  6.2 Implementation Output Screens | **41**  **42-47** |
| **7.** | **Chapter 7 – Summary and Conclusion**  7.1 Result and Outcome of the System  7.2 Limitations of the System | **49**  **50** |

**Acronyms-**

1. **VSMS-** Vehicle Service Management System
2. **SRS-** Software Requirement Specification
3. **ER-** Entity Relationship
4. **DFD-** Data Flow Diagram
5. **PHP-** Hypertext Preprocessor
6. **HTML-** Hypertext Markup Language
7. **CSS-** Cascading Style Sheet

**List of Tables-**

Table1: Hardware Requirements

Table2: Software Requirements

Table3: tbladmin MySQL Table

Table4: tblcategory MySQL Table

Table5: tblenquirytype MySQL Table

Table6: tblenquiry MySQL Table

Table7: tblmechanics MySQL Table

Table8: tbluser MySQL Table

Table9: tblservicerequest MySQL Table

Table10: User Registration Test Case

Table11: User Authentication Test Case

Table12: Service Request Test Case

Table13: Update Mechanic Details Test Case

Table14: Hardware System Requirements

Table15: Server System Requirements

**List of Figures-**

Figure1: Protype Model

Figure2: Use Case Diagram

Figure3: Context Diagram

Figure4: Level 1 DFD

Figure5: Level 2 DFD (Customer)

Figure6: Level 2 DFD (Supervisor)

Figure7: ER Diagram

Figure8: Class Diagram

Figure9: Sequence Diagram (User Login)

Figure10: Sequence Diagram (Check Status)

Figure11: Collaboration Diagram (Login)

Figure12: Activity Diagram (Register Service Request)

Figure13: Activity Diagram (Update Service Status)

Figure14: State Chart Diagram (Service Update and Invoice Generation)

Figure15: State Chart Diagram (Update Mechanic Details)

Figure16-28: System Implementation Output Screens

CHAPTER 1

**Introduction**

* 1. **OBJECTIVES AND SCOPE OF THE PROJECT**

The Vehicle Service Management System is an integral to providing quality customer service. It helps to measure public satisfaction and is a useful source of information and feedback for improving services. In this system, the users or customers can book appointments for their vehicle service at any time, at any place with the help of any computing device with web access. The user friendliness is the key aspect of this system and user-friendly GUI is deployed in this system.

**The following objectives have been set:**

1) Smooth flow of data without any hurdles.

2) Adequate validation checks for data entry.

3) Adequate security of data.

4) Facility to update data from time to time.

5) Prompt and specific retrieval of data.

6) Flexibility in the system according to the changing environment.

7) Accuracy, timeliness, and comprehensiveness of the system output.

8) Stability and operability by people of average intelligence.

* 1. **THEORETICAL BACKGROUND**

In this era of technological advancement where everything is being computerized and the use of paperwork has reduced to minimal extend, and most of the works nowadays are done on internet. So, this project is developed and designed to a user-friendly website/ portal based on PHP as front end and MySQL as back end tool to help users to have Vehicle Service Management System.

This system is a user-friendly system with integration of HTML, CSS, Bootstrap, PHP and MySQL. The webpage is divided into two parts i.e. User Section and Supervisor/ Administrator Section, which you can notice in the webpage. The users can request the system to service their vehicles, enquire about any queries or service status. The supervisor module has a dashboard where they can manage the requests made by the customer, manage mechanics, inventory, generate reports etc.

All precautions have been taken in creation of this website so that it is easy for the users to use it. Every page of the website is created after detailed understanding on every minute aspect which would make the user’s work freer and easier.

CHAPTER 2

**Review of Literature**

**2.1 NEED FOR SYSTEM**

The existing system of vehicle service management system is done manually and there is requirement of personnel to maintain customer records and record service schedules, the customers had to make a service appointment telephonically before physically bringing vehicle or availing doorstep assistance for service. The implemented system can receive request for service through the implemented web portal which can handle the details without any difficulty & with a little bit of effort. As the work was done manually before, so it will be very time consuming & required a large effort to maintain the files. By computerizing the system these files can be handled with a small effort & in less time.

* 1. **PROPOSED SYSTEM**

The proposed system has a responsive and dynamic homepage where users/ customers can sign-in the system securely and the users are greeted with an informative style homepage. The users can see their user panel which contains their service history, service requests etc. The users can change their existing passwords with new ones. The users can enquire about availability of parts or lodge their grievance in the enquiry/ feedback section. The users can request for the periodical service of their vehicle through service request section. The users can track their service and print the invoice for their vehicle service.

There is a separate portal for this system for the supervisors where they can login this system securely by entering their login credentials and they are greeted with an analytical style homepage where they can see diagrammatical style of information/ report for their convenience. The supervisor handles mechanics information and customers. They assign customer service requests to mechanics and updates service request status; they also handle enquiry made by the customers.

**Advantages of Proposed System**

The proposed system has some advantages over the existing one which are listed below:

1. **Enhancement**: The main objective of Vehicle Service Management System is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with the computer-based system.
2. **Automation**: This system automates each and every activity of the manual system and increases its throughput. Thus, the response time of the system is very less and it works very fast.
3. **Accuracy**: This System provides the user a quick response with very accurate information regarding the customer records.
4. **User-Friendly**: This System has a very user-friendly interface. Thus, the users will feel very easy to work on it. The software provides accuracy along with a pleasant interface. Make the present manual system more interactive, speedy and user friendly.
5. **Availability**: The transaction reports of the system can be retried as and when required. Thus, there is no delay in the availability of any information, whatever needed, can be captured very quickly and easily.

CHAPTER 3

**System Requirement**

**Analysis**

**3.1 Process Analysis**

The purpose of this system is to ease the procedure of vehicle service for the customers in an effective and efficient manner. This system provides services such as vehicle service booking, service status, invoice generation, mechanic and user management, enquiry form.

These are the following functionality/ processes of the modules integrated with this system: -

1. **User Registration**
2. **Service Booking**
3. **Mechanic Management**
4. **Vehicle Service Request Management**
5. **Enquiry Form**
6. **Enquiry Management**
7. **User Registration:**

The User fills the registration form by giving the personal information and successfully registers with the website.

* 1. **Provisions:**
* Full Name
* Mobile Number
* Email Address
* Password
* Repeat Password
  1. **Functionalities**
* New User is created
* User details inserted
* Each user is allotted with unique User-ID
  1. **Validation Rules**
* User already exists
* Please fill the required fields

1. **Service Booking:**

The User fills the service registration form by giving the vehicle information and successfully registers with the website.

* 1. **Provisions:**
* Vehicle Category
* Vehicle Name
* Vehicle Model
* Brand
* Registration Number
* Service Date and Time
* Delivery Type
  1. **Functionalities**
* Service request created
* Service details inserted
  1. **Validation Rules**
* Please fill the required fields

1. **Mechanic Management:**

The Supervisor manages mechanic working under him/her and inserts new workers or updates the details of existing workers.

* 1. **Provisions:**
* Mechanic Name
* Phone Number
* Email Address
* Residential Address
  1. **Functionalities**
* Mechanic details inserted
* Mechanic details updated
  1. **Validation Rules**
* Please fill the required fields

1. **Vehicle Service Request Management:**

The Supervisor manages the service requests submitted by the customer and updates the status of the service requests.

* 1. **Provisions:**
* Service Number
* Approval Status
* Mechanic Assigned
* Total Bill
* Service Status
  1. **Functionalities**
* Servicing request completed
* Servicing request rejected
  1. **Validation Rules**
* Please fill the required fields

1. **Vehicle Service Request Management:**

The Supervisor manages the service requests submitted by the customer and updates the status of the service requests.

* 1. **Provisions:**
* Service Number
* Approval Status
* Mechanic Assigned
* Total Bill
* Service Status
  1. **Functionalities**
* Servicing request completed
* Servicing request rejected
  1. **Validation Rules**
* Please fill the required fields

1. **Enquiry Form:**

The user fills the enquiry form by filling required information in the system.

* 1. **Provisions:**
* Enquiry Type
* Description
  1. **Functionalities**
* Enquiry Submitted
  1. **Validation Rules**
* Please fill the required fields

1. **Enquiry Management:**

The supervisor resolves the enquiry of the customers and updates remarks by filling required information in the system.

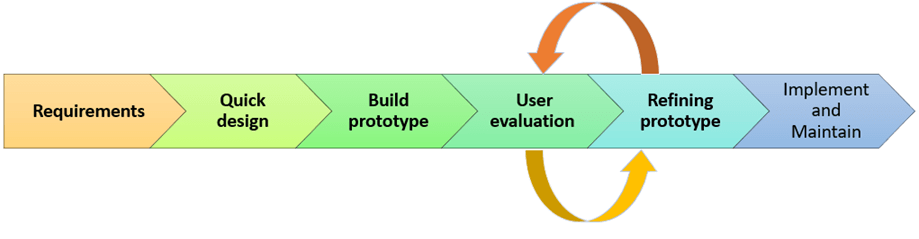
* 1. **Provisions:**
* Status Update
* Response
* Date and Time
  1. **Functionalities**
* Enquiry Resolved
  1. **Validation Rules**
* Please fill the required fields

**3.2 SRS Methodology Adopted**

Prototype methodology is defined as a Software Development model in which a prototype is built, test, and then reworked when needed until an acceptable prototype is achieved. It also creates a base to produce the final system.

Software prototyping model works best in scenarios where the project's requirement is not known. It is an iterative, trial, and error method which take place between the developer and the client.

**Phases of Prototype Model**

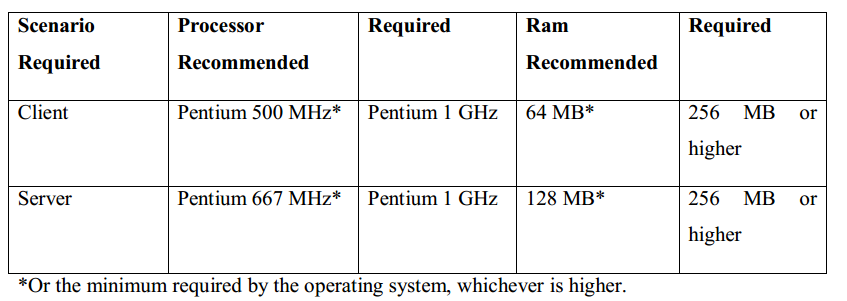
****

**Figure 1. Protype Model Phases**

1. **Requirement gathering and analysis**
2. **Quick Design**
3. **Build Prototype**
4. **User Evaluation**
5. **Redefine Prototype**
6. **Implement and Maintain**

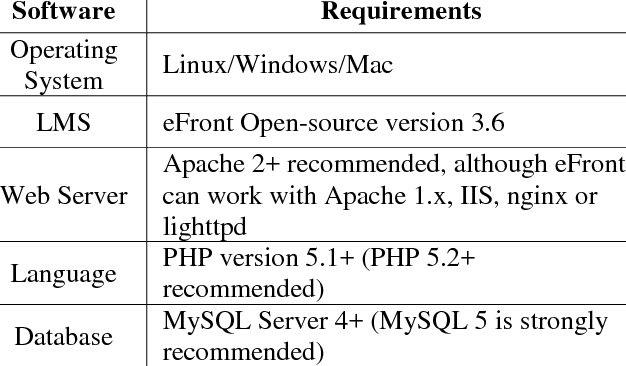
**3.3 System Requirements**

**3.3.1 Hardware Requirements**

****

**Table 1: Hardware Requirements**

**3.3.2 Software Requirements**



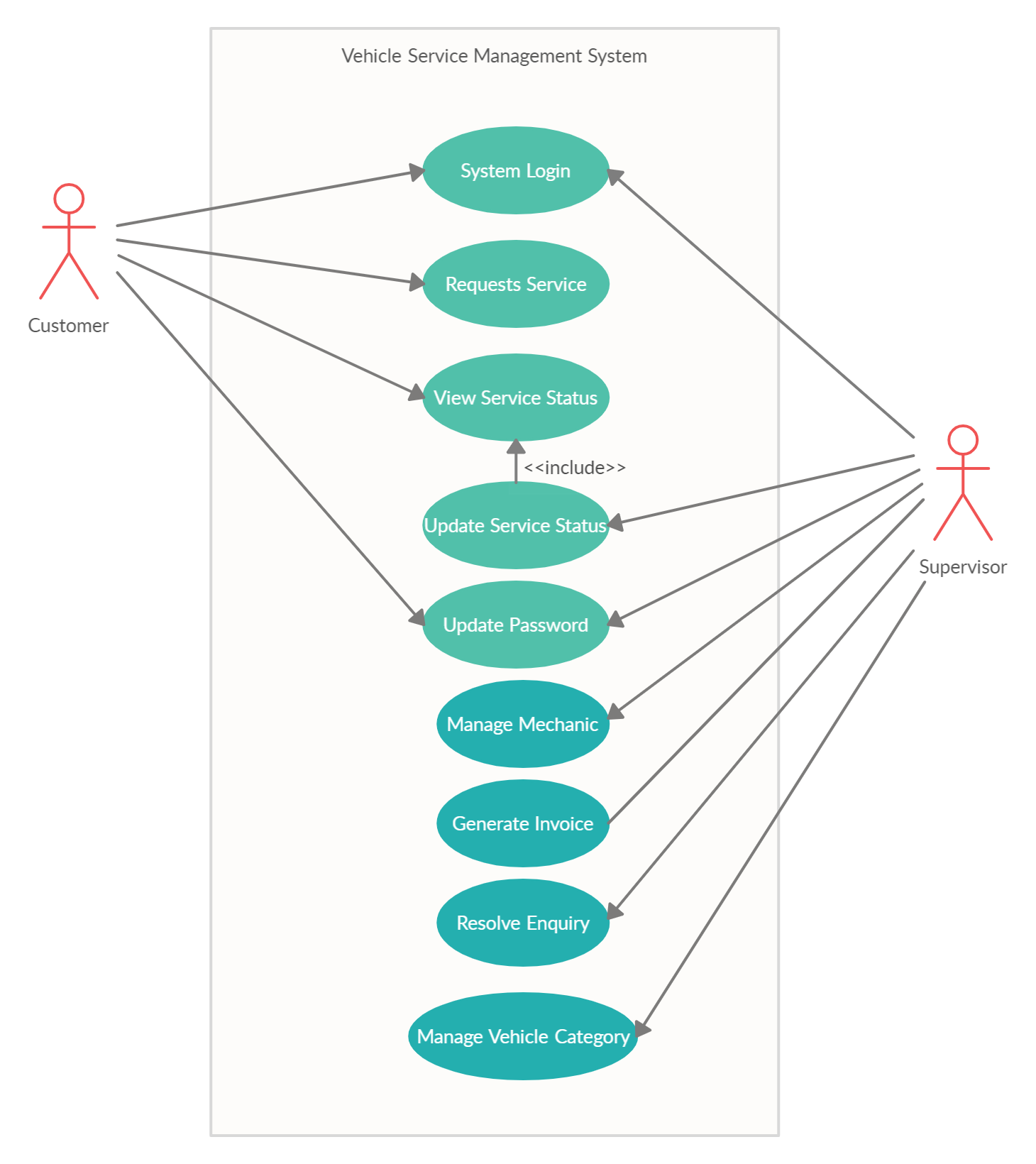
**Table 2: Software Requirements**

|  |
| --- |
|  |
|  |

CHAPTER 4

**System Design**

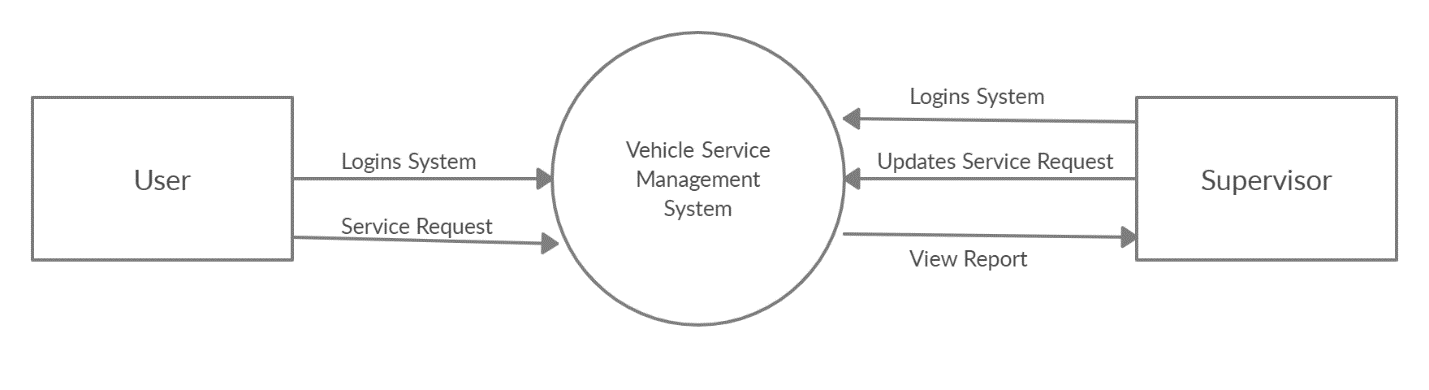
**4.1 Use Case Diagram**



**Figure 2: Use Case Diagram of VSMS**

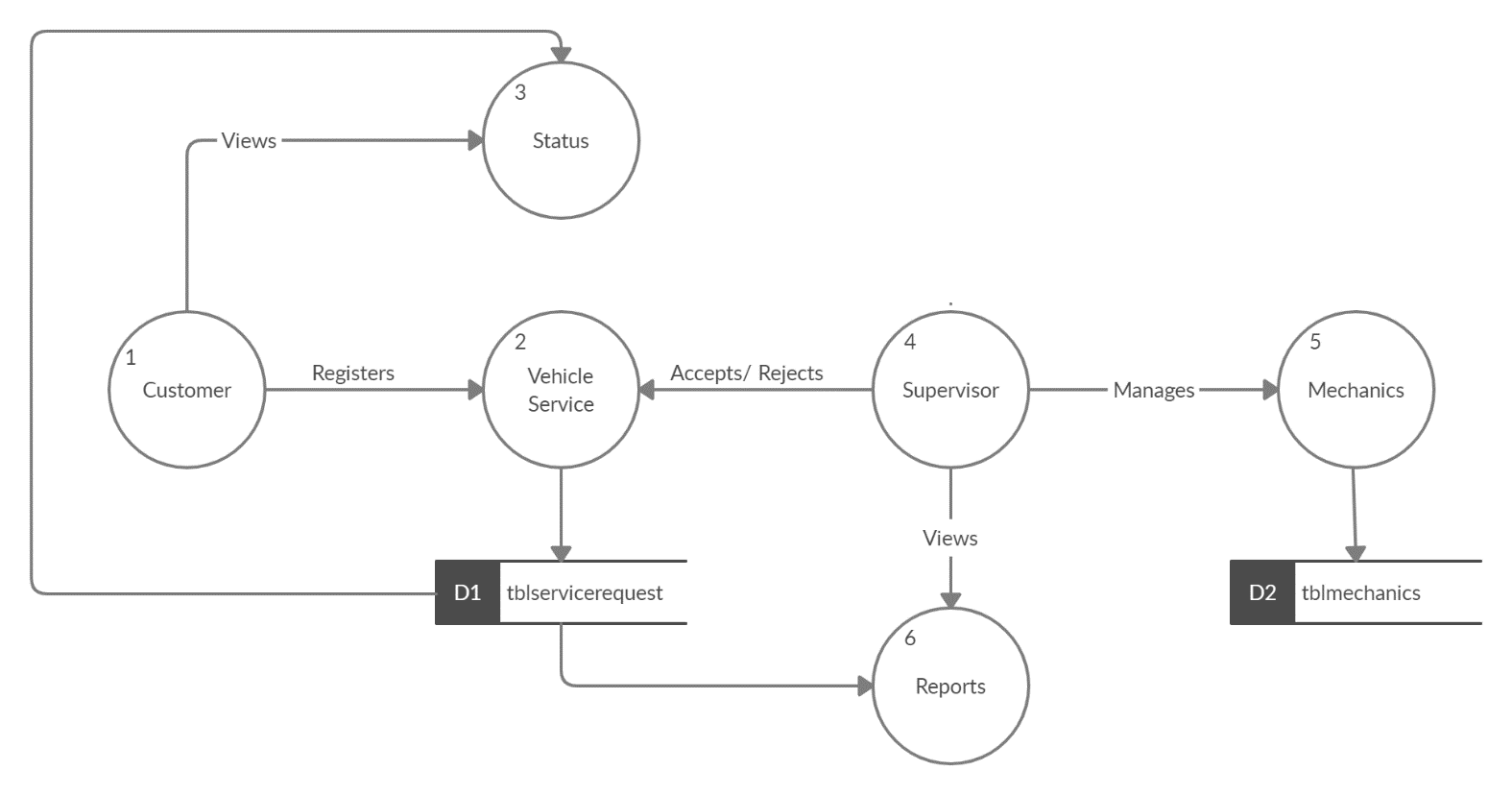
**4.2 Data Flow Diagram**

**4.2.1 Level-0 DFD or Context Diagram**



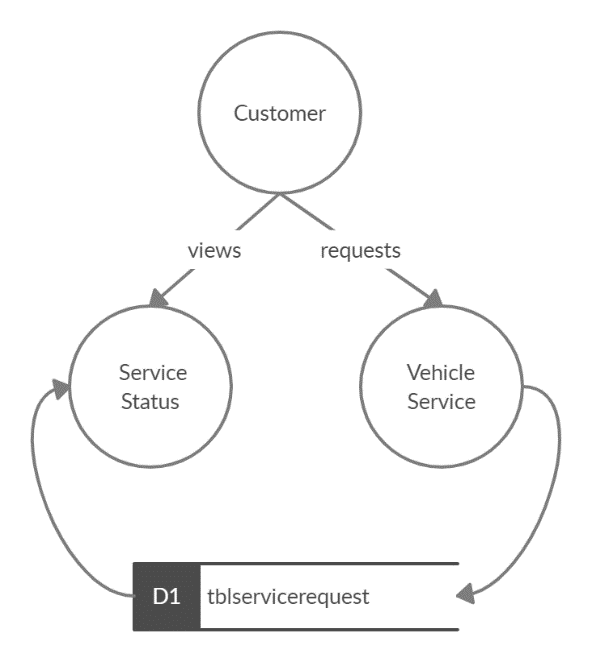
**Figure 3: Context Diagram of VSMS**

**4.2.2 Level 1 DFD**

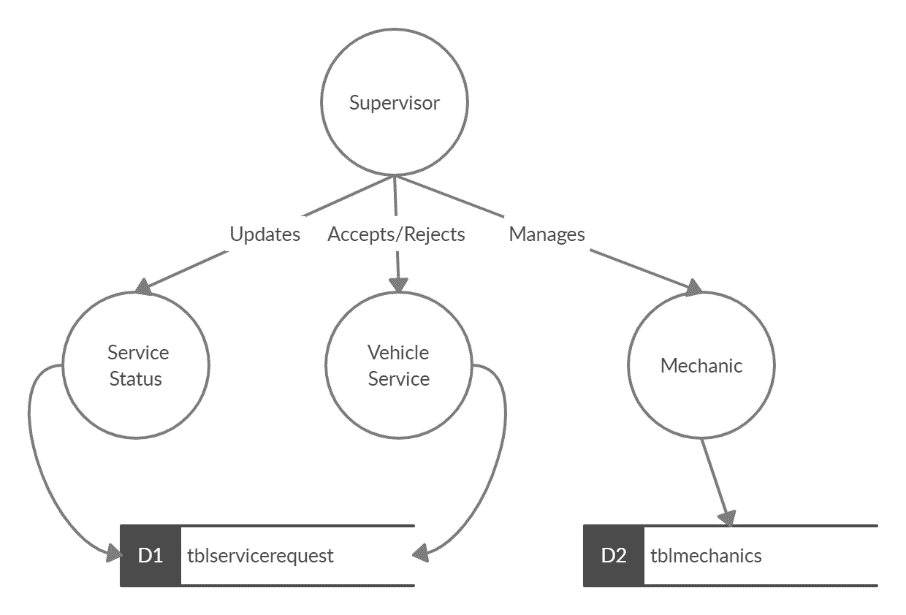


**Figure 4: Level 1 DFD**

**4.2.3 Level 2 DFD**

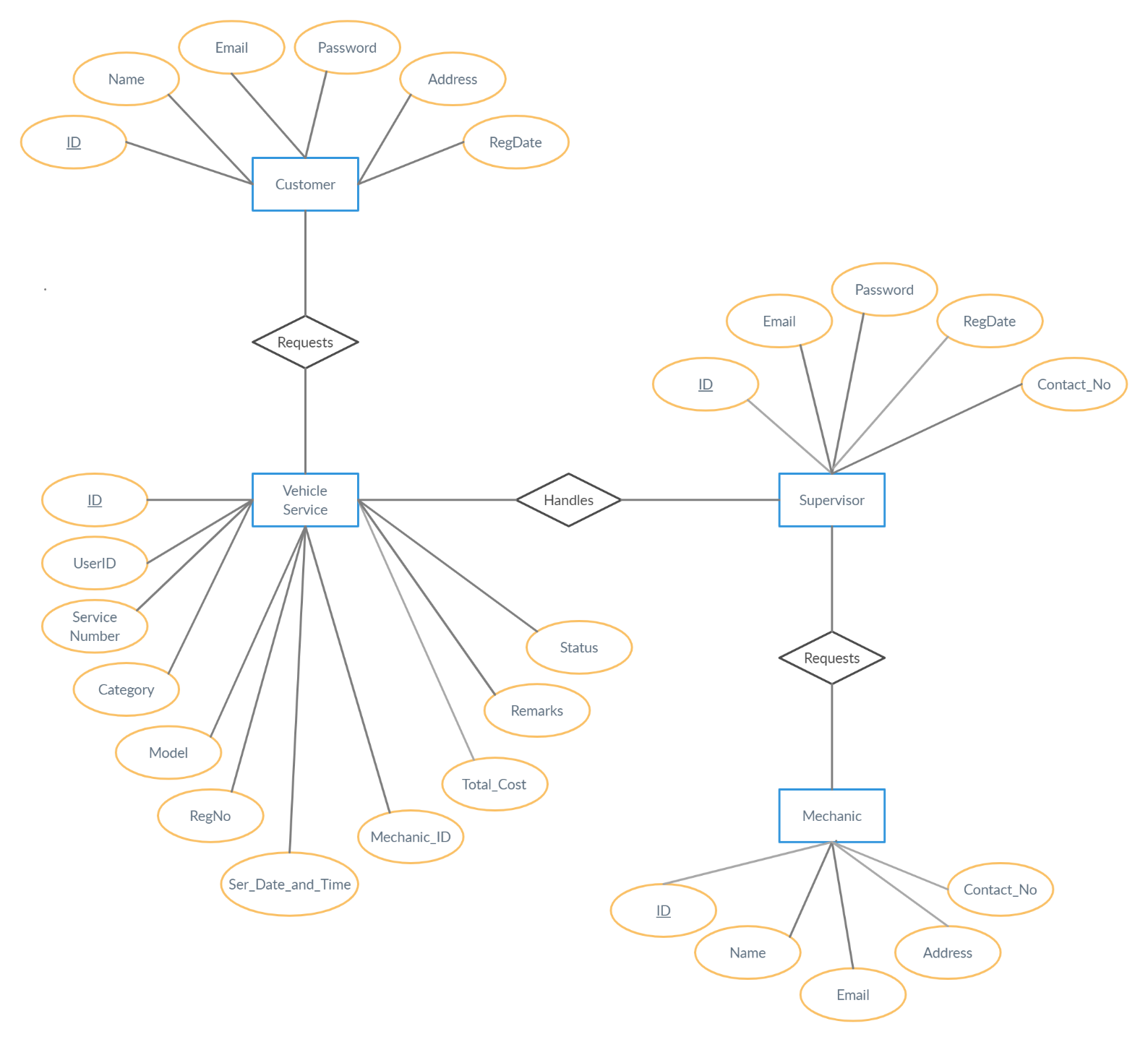


**Figure 5: Level 2 DFD (Customer)**



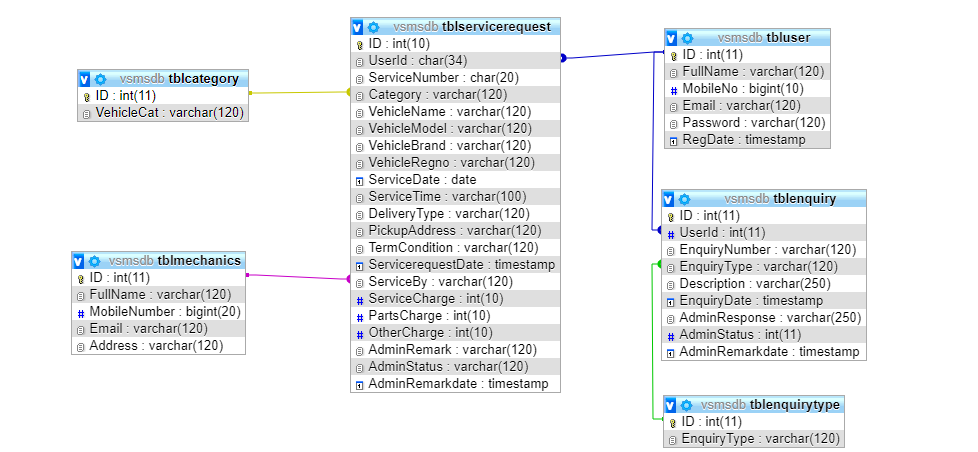
**Figure 6: Level 2 DFD (Supervisor)**

**4.3 ER Diagram**



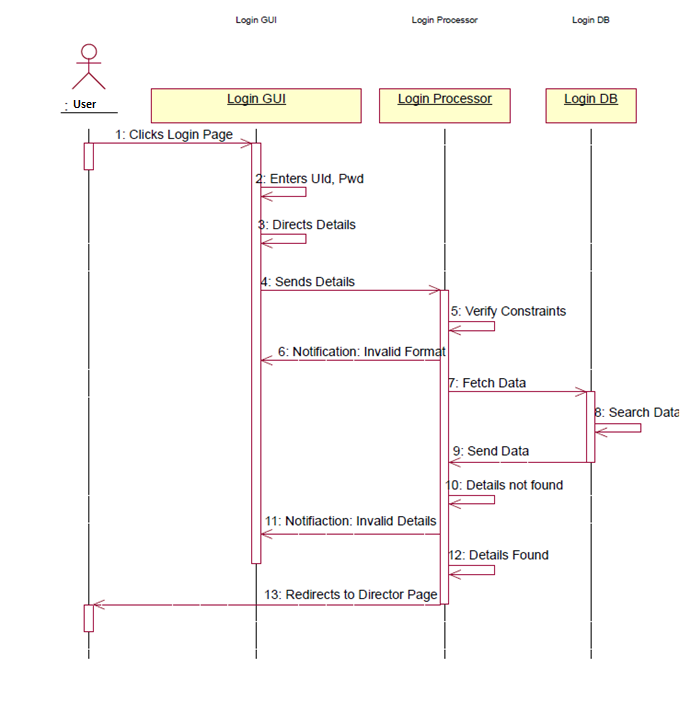
**Figure 7: ER Diagram of VSMS**

**4.4 Class Diagram**

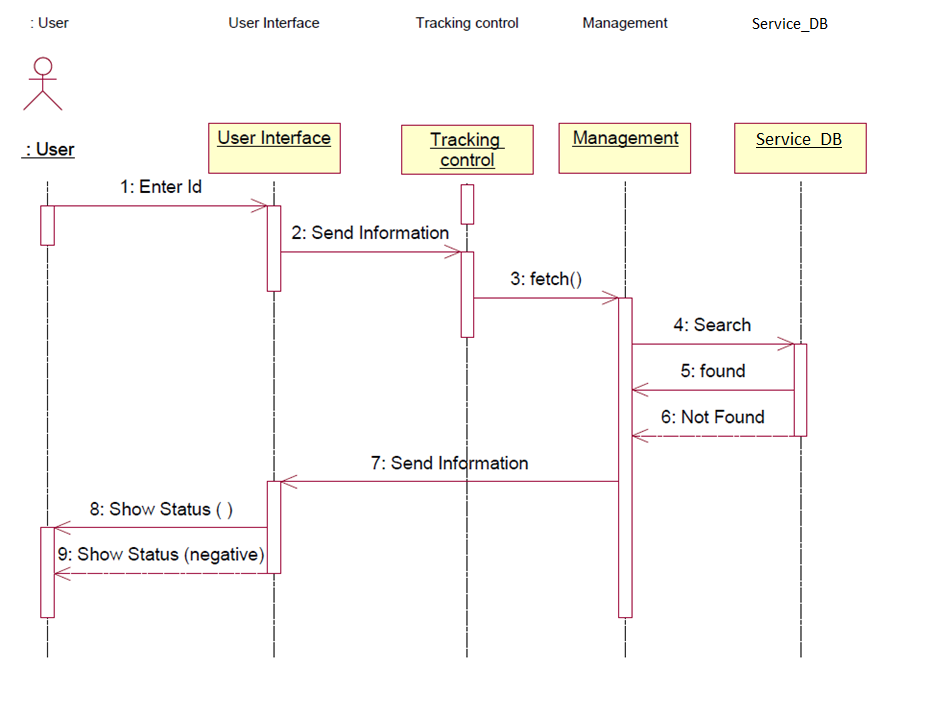


**Figure 8: Class Diagram**

**4.5 Sequence Diagram**

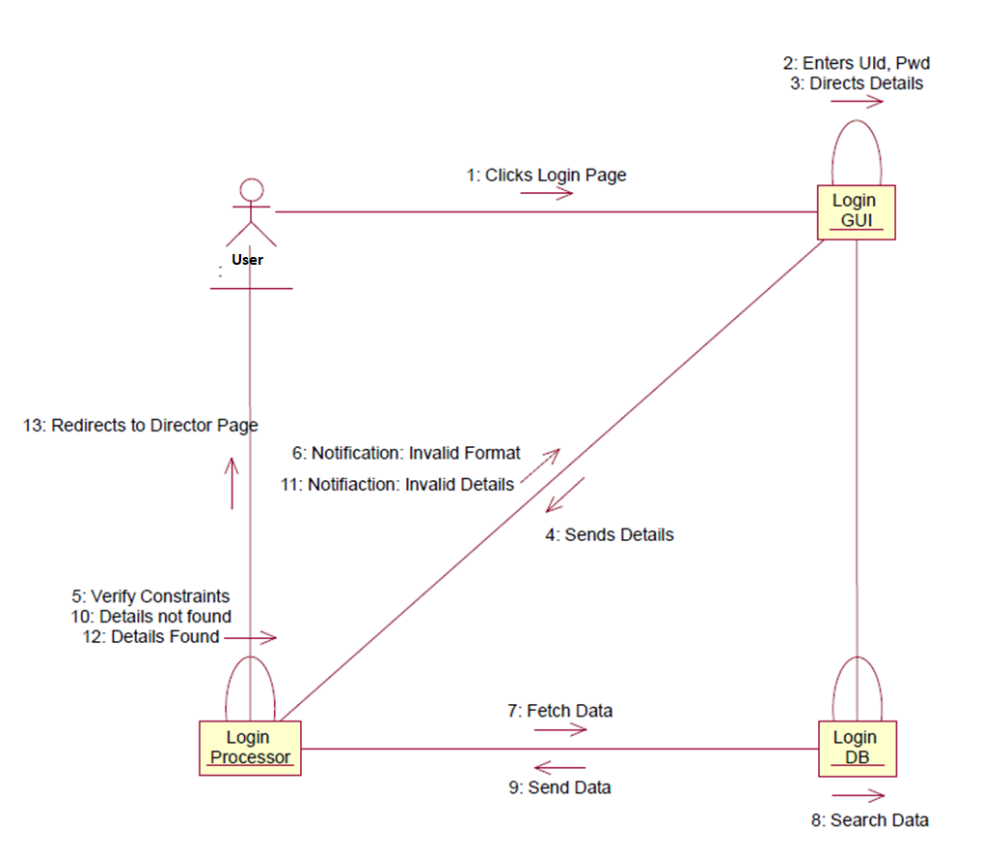
****

**Figure 9: User Login**

****

**Figure 10: Check Status**

**4.6 Collaboration Diagram**

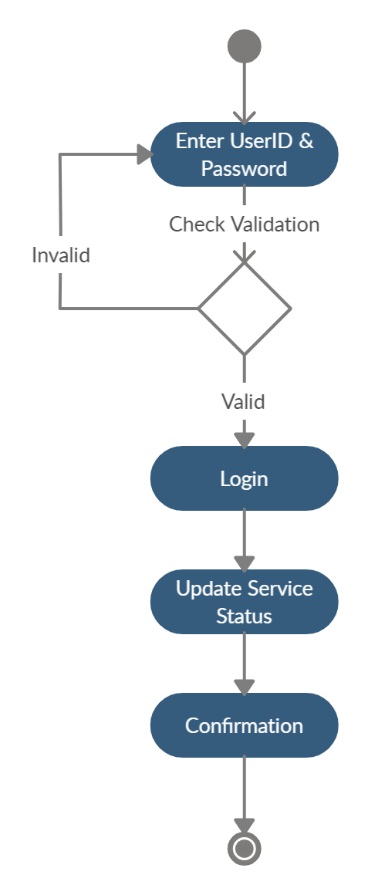
****

**Figure 11: User Login**

**4.7 Activity Diagram**

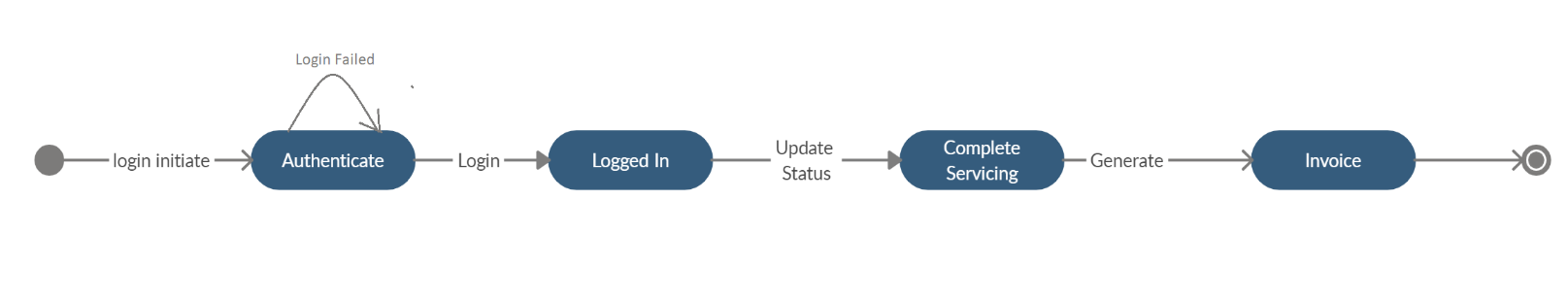


**Figure 12: Register Service Request**

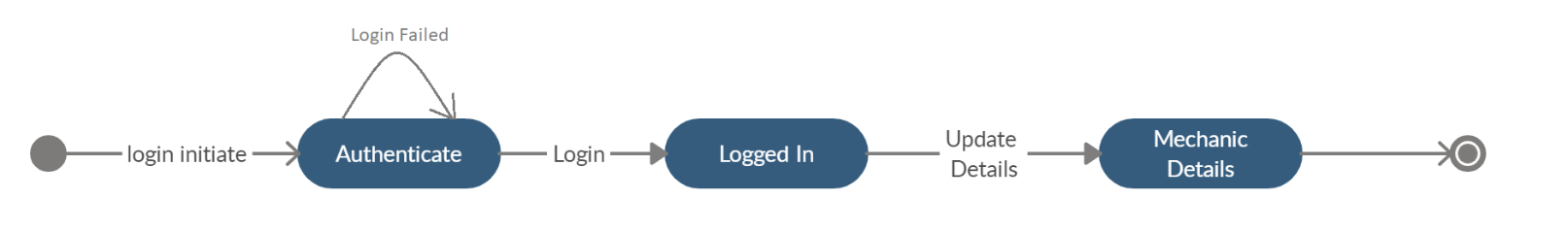


**Figure 13: Update Service Status**

**4.8 State Chart Diagram**

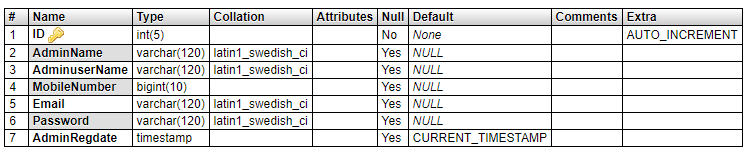
****

**Figure 14: Service Update and Invoice Generation**

****

**Figure 15: Update Mechanic Details**

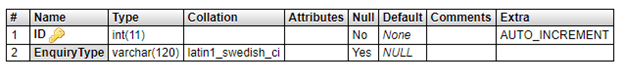
**4.9 Database Design**

****

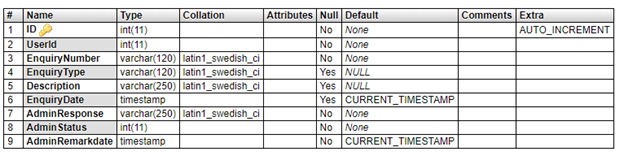
**Table 3: tbladmin MySQL Table**

****

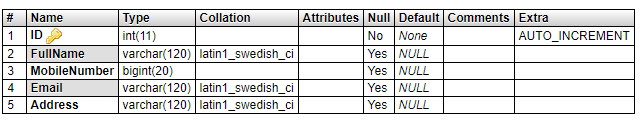
**Table 4: tblcategory MySQL Table**

****

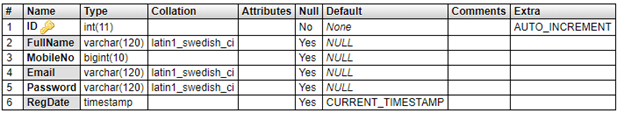
**Table 5: tblenquirytype MySQL Table**

****

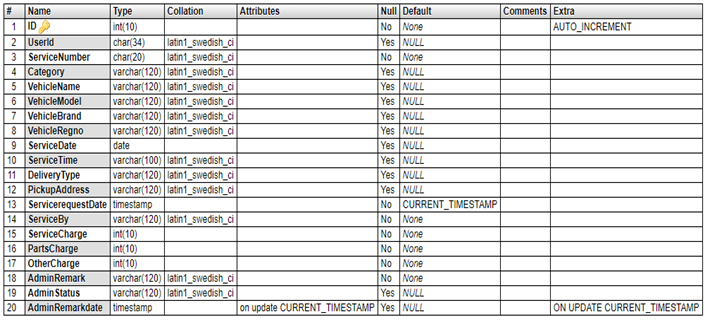
**Table 6: tblenquiry MySQL Table**



**Table 7: tblmechanics MySQL Table**

****

**Table 8: tbluser MySQL Table**

****

**Table 9: tblservicerequest MySQL Table**

CHAPTER 5

**System Development**

* 1. **Programme Development**
     1. **Front End Software Tool- Hypertext Preprocessor (PHP)**

PHP: Hypertext Preprocessor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as a standalone interpreter on most operating systems and computing platforms.

**ADVANTAGES TO PHP:**

* Open Source, readily available (you can be using it today) and dual-   
  licensed - if you are doing non-profit work or not licensing, there is no cost.
* Very easy to understand Syntax, some really cool features (arrays   
  are something else!)
* Interfaces very easily with Apache/MySQL
* Server side.
* Lots of good source code out there to use and/or learn from, as well   
  as many useful libraries for working with PDFs, graphics, etc.
  + 1. **Back End Software Tool- MySQL**

MySQL is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source [client–server model](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) RDBMS.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](https://en.wikipedia.org/wiki/LAMP_%28software_bundle%29) open source web application software stack (and other "[AMP](https://en.wikipedia.org/wiki/List_of_AMP_packages)" stacks). LAMP is an acronym for "[Linux](https://en.wikipedia.org/wiki/Linux), [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](https://en.wikipedia.org/wiki/Perl)/[PHP](https://en.wikipedia.org/wiki/PHP)/[Python](https://en.wikipedia.org/wiki/Python_%28programming_language%29)." [Free-software](https://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL.

* + 1. **XAMPP Client (For testing and run system locally)**

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

* 1. **Testing**

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.

**Unit Testing**

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

|  |  |  |
| --- | --- | --- |
| **Test case 1**: User Registration | | **Priority (H, L):** High |
| **Test Objective:** For Verifying Registration | | |
| **Test Description:** “User enters the required fields, presses register button”, client program contacts with server, server contacts with the database, database updates and sends result to the user. | | |
| **Requirements Verified:** Yes | | |
| **Test Environment:**  Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program. | | |
| **Test Setup/Pre-Conditions:** Apache server should be in running state. All the mandatory fields must be entered. | | |
| Actions | Expected Results | |
| The user will register to access application. | Displays the respective Pages | |
| **Pass: Yes Conditions pass: Yes**  **Fail**: No | | |
| **Problems / Issues:** NIL | | |
| **Notes**: Successfully registered | | |

|  |  |  |
| --- | --- | --- |
| **Test case 2**: **Verifying Authentication.** | | Priority (H, L): High |
| **Test Objective**: For Verifying Authentication. | | |
| **Test Description**: “User enters username and password and presses submit button”, client program contacts with server, server contacts with the database, database checks for authentication and sends result as a valid user. | | |
| **Requirements Verified**: Yes | | |
| **Test Environment**: Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program. | | |
| **Test Setup/Pre-Conditions**: Apache and Database server should be in running state. Username and Password fields should be entered. | | |
| Actions | Expected Results | |
| The user presses submit button. | “Valid User”, “. Displays Main Menu. | |
| Pass: yes Conditions pass: yes Fail: No | | |
| Problems / Issues: NIL | | |
| Notes: User Verified successfully | | |

* + 1. **Test Cases**

**Table 10: User Registration Test Case**

**Table 11: User Authentication Test Case**

|  |  |  |
| --- | --- | --- |
| **Test case 3: Service Requests** | | **Priority (H, L):** High |
| **Test Objective:** For Inserting Requests | | |
| **Test Description:** “User enters the required fields, presses Submit button”, client program contacts with server, server contacts with the database, database updates and sends result to the user. | | |
| **Requirements Verified:** Yes | | |
| **Test Environment:**  Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program. | | |
| **Test Setup/Pre-Conditions:** Apache server should be in running state. All the mandatory fields must be entered. | | |
| Actions | Expected Results | |
| The user will Select the product and sends the enquiry request | Displays the respective Pages | |
| **Pass: Yes Conditions pass: Yes**  **Fail**: No | | |
| **Problems / Issues:** NIL | | |
| **Notes**: Request is placed successfully | | |

**Table 12: Service Request Test Case**

|  |  |  |
| --- | --- | --- |
| **Test case 4: Update Mechanics** | | **Priority (H, L):** High |
| **Test Objective:** For updating mechanic details. | | |
| **Test Description:** “User Selects the required fields, presses update button”, client program contacts with server, server contacts with the database, database updates and sends result to the user. | | |
| **Requirements Verified:** Yes | | |
| **Test Environment:**  Apache and Database server must be in running state, Database Should contain appropriate table and link must be established between server and client program. | | |
| **Test Setup/Pre-Conditions:** Apache server should be in running state. All the mandatory fields must be entered. | | |
| Actions | Expected Results | |
| The user will select a product and clicks delete | Displays the respective Pages | |
| **Pass: Yes Conditions pass: Yes**  **Fail**: No | | |
| **Problems / Issues:** NIL | | |
| **Notes**: product is deleted successfully. | | |

**Table 12: Update Mechanic Test Case**

CHAPTER 6

**System Implementation**

* 1. **Acquisitions**
     1. **Hardware Requirements for System Development**

|  |  |
| --- | --- |
| **CPU** | **1 Ghz. or higher AMD or Intel Based SOC** |
| **Memory** | **1 GB for 32 Bit System and 2 GB for 64 Bit System.** |
| **Storage** | **16 GB + 100 MB for 32 Bit System and**  **32 GB + 100 MB for 64 Bit System** |
| **GPU** | **DirectX-9 or later based integrated GPU or external GPU.** |
| **Display** | **800x600 or higher resolution display output monitor.** |

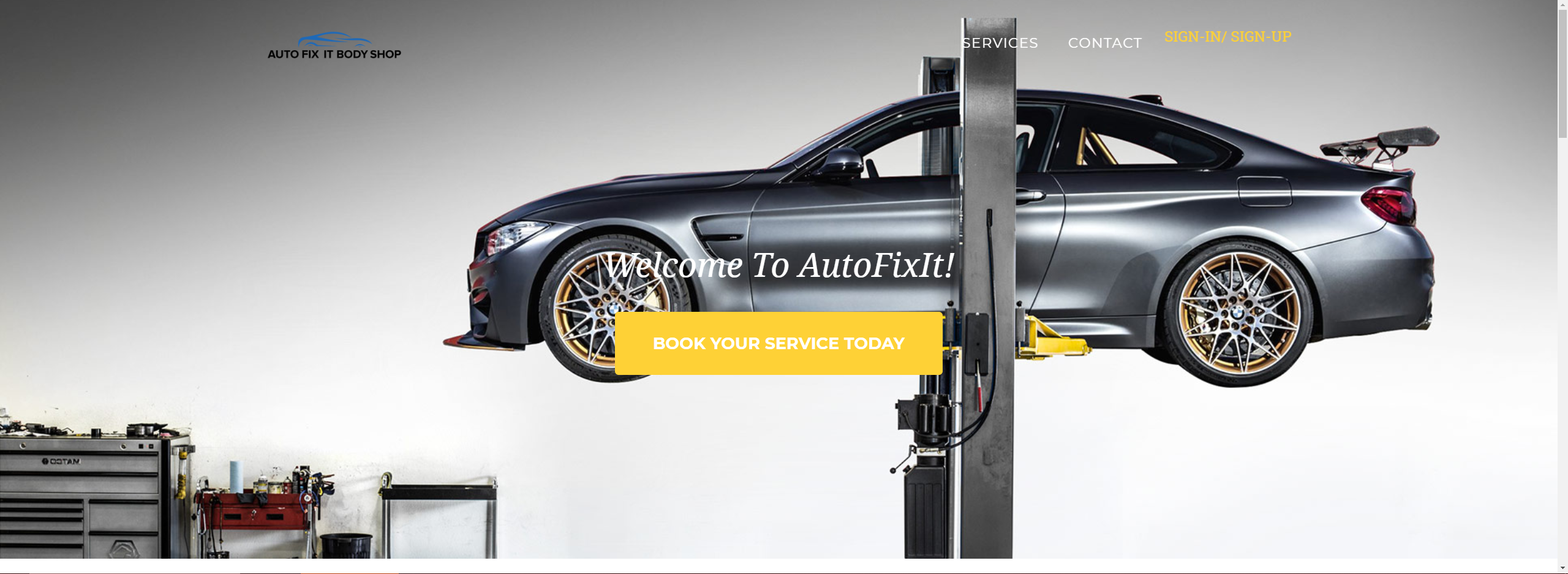
**Table 14: Hardware Specification for VSMS development**

* + 1. **Server Requirements and Cost Estimation for System Development**

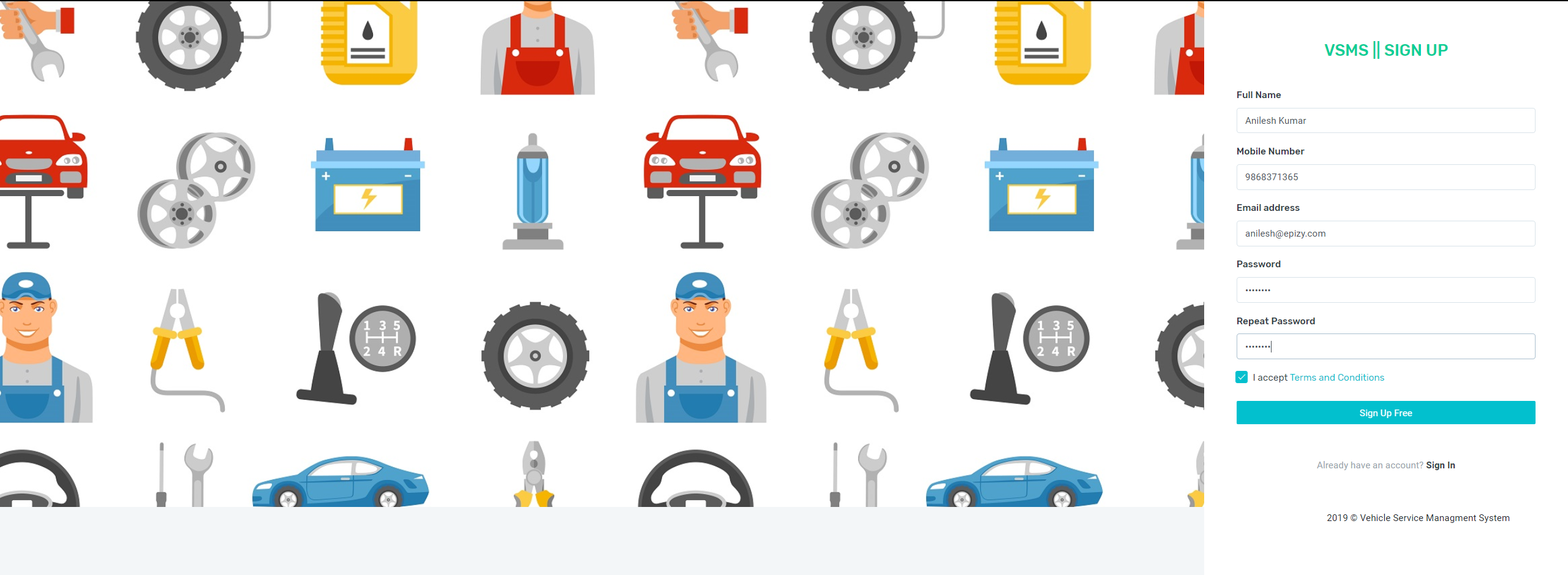
|  |  |
| --- | --- |
| **Server** | **Apache based server** |
| **FTP** | **Monsta FTP or Similar** |
| **MySQL Server** | **CPanel or Similar** |
| **Server Cost** | **10$ and above for domain and 3-7 $ per month for hosting.** |

**Table 15: Server Specification for VSMS development**

* 1. **System Implementation Output Screens**



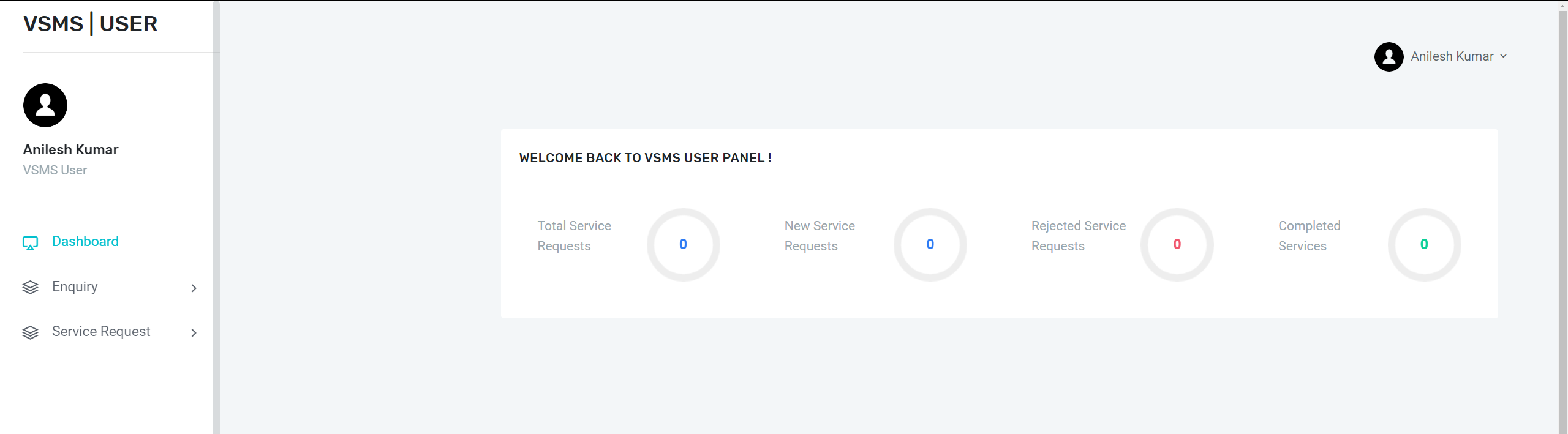
**Figure 16: VSMS Homepage**



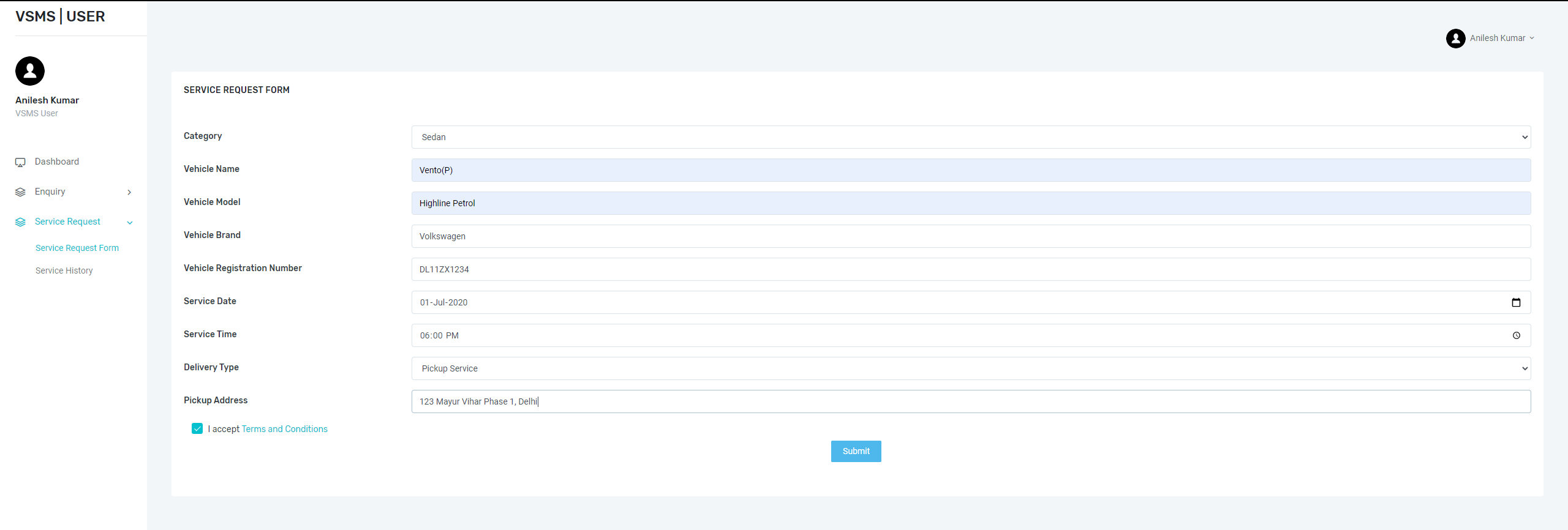
**Figure 17: Customer Sign-Up**



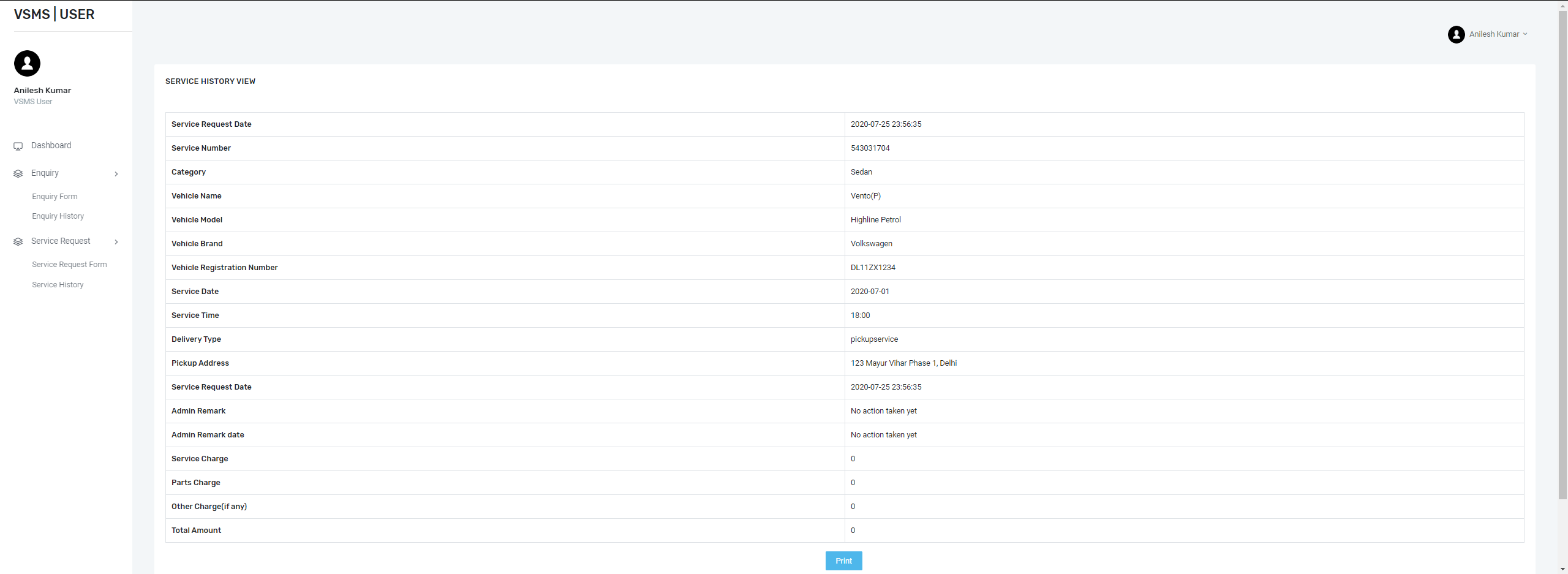
**Figure 18: Customer Log-In**



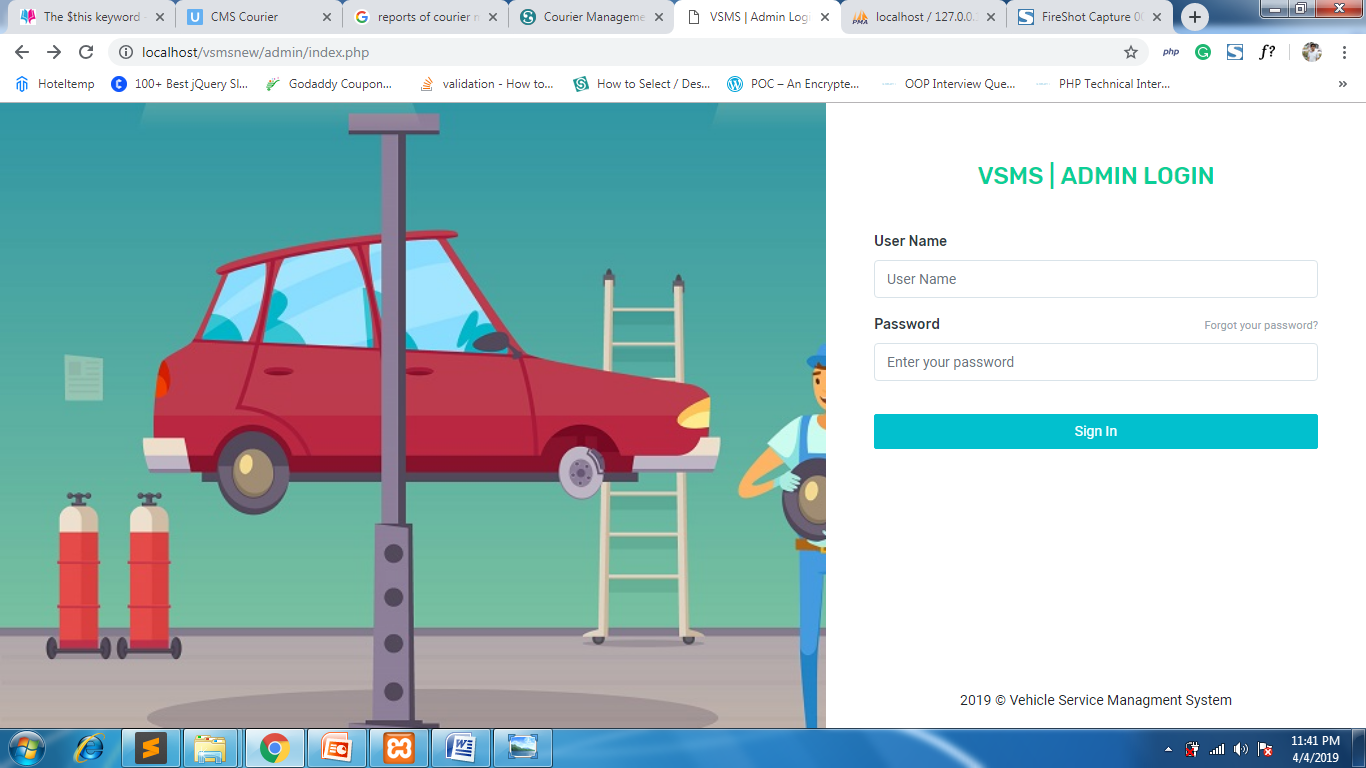
**Figure 19: Customer Homepage**



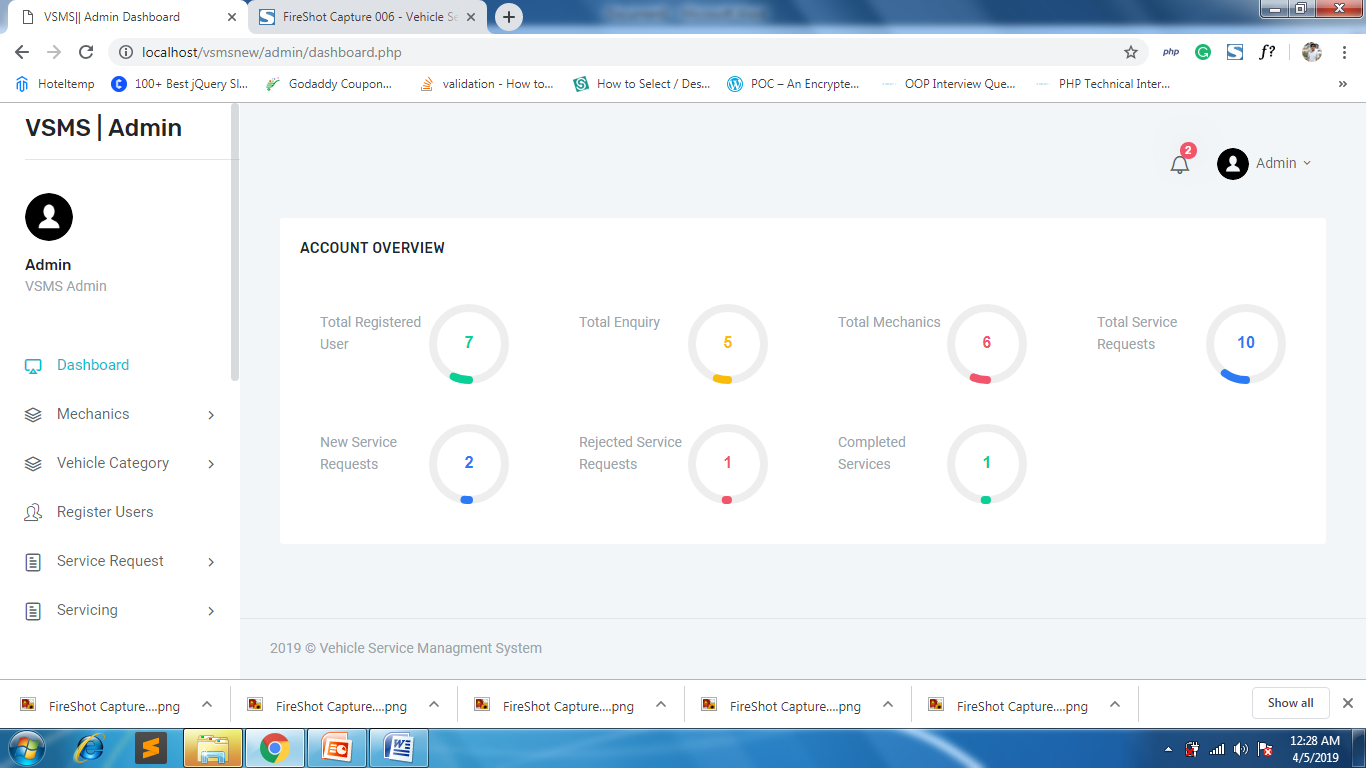
**Figure 20: Service Registration**



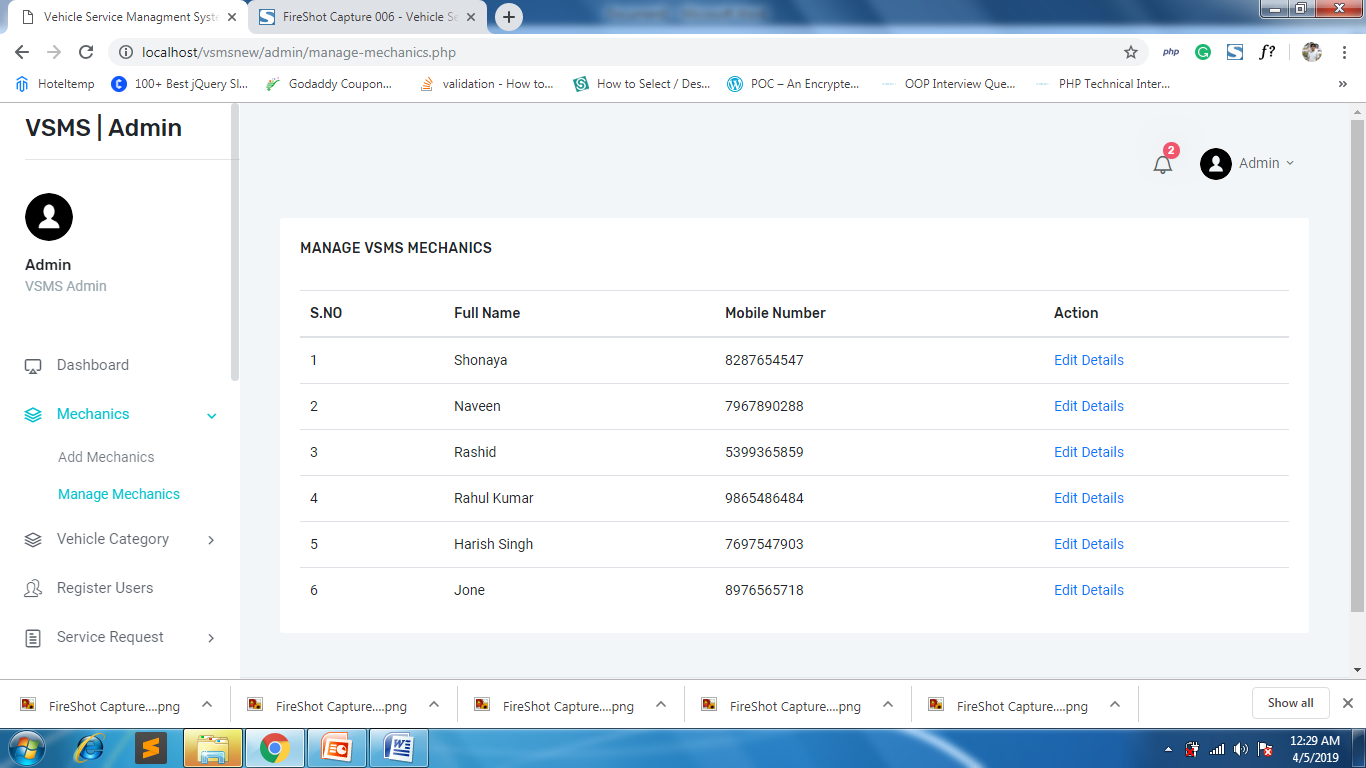
**Figure 21: Service History**



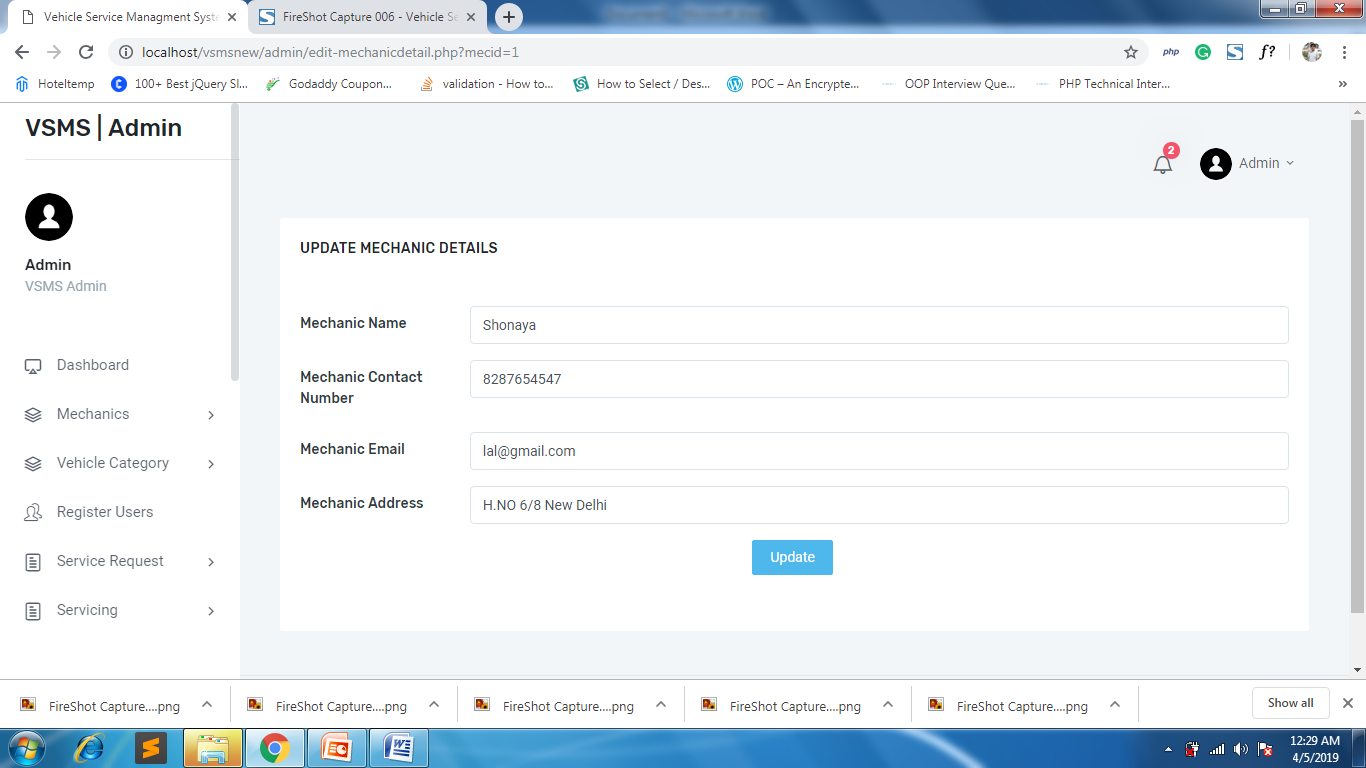
**Figure 22: Supervisor/ Admin Login**



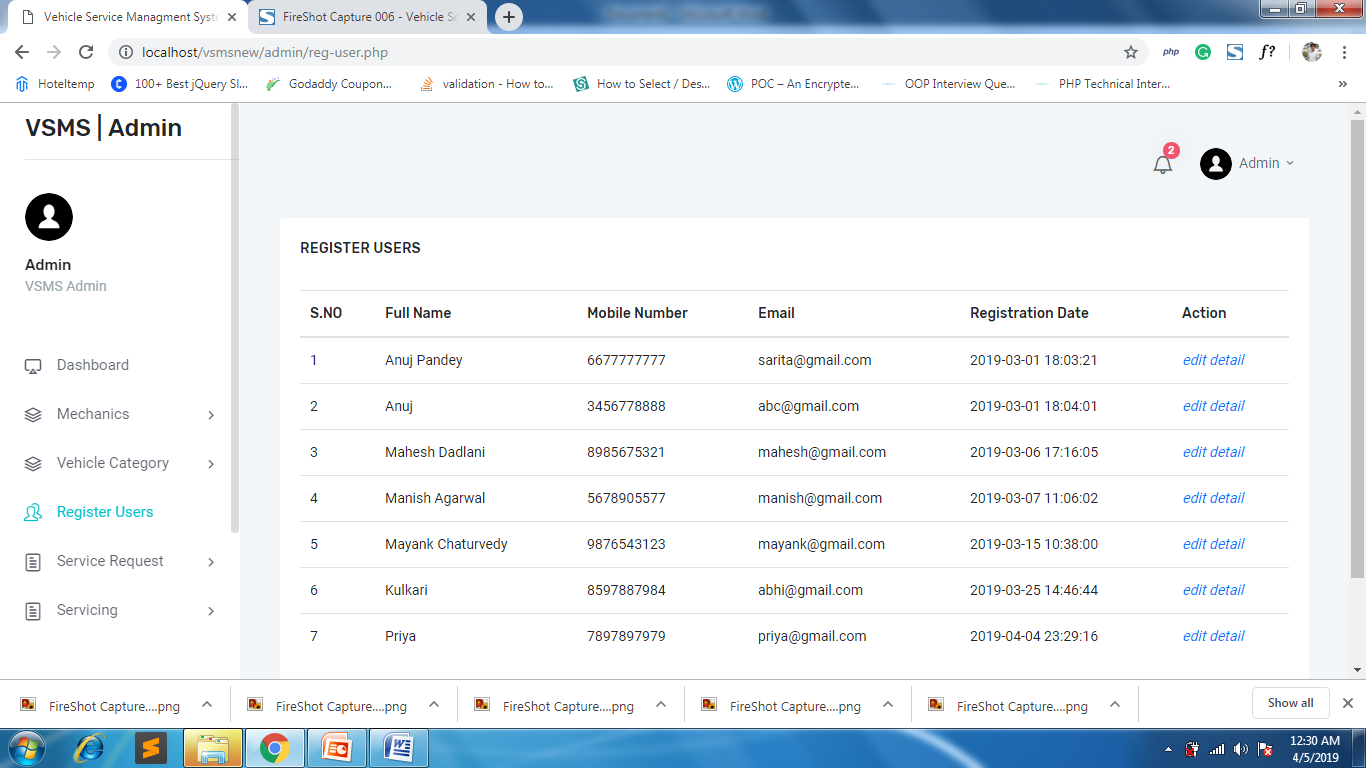
**Figure 23: Supervisor/ Admin Dashboard**



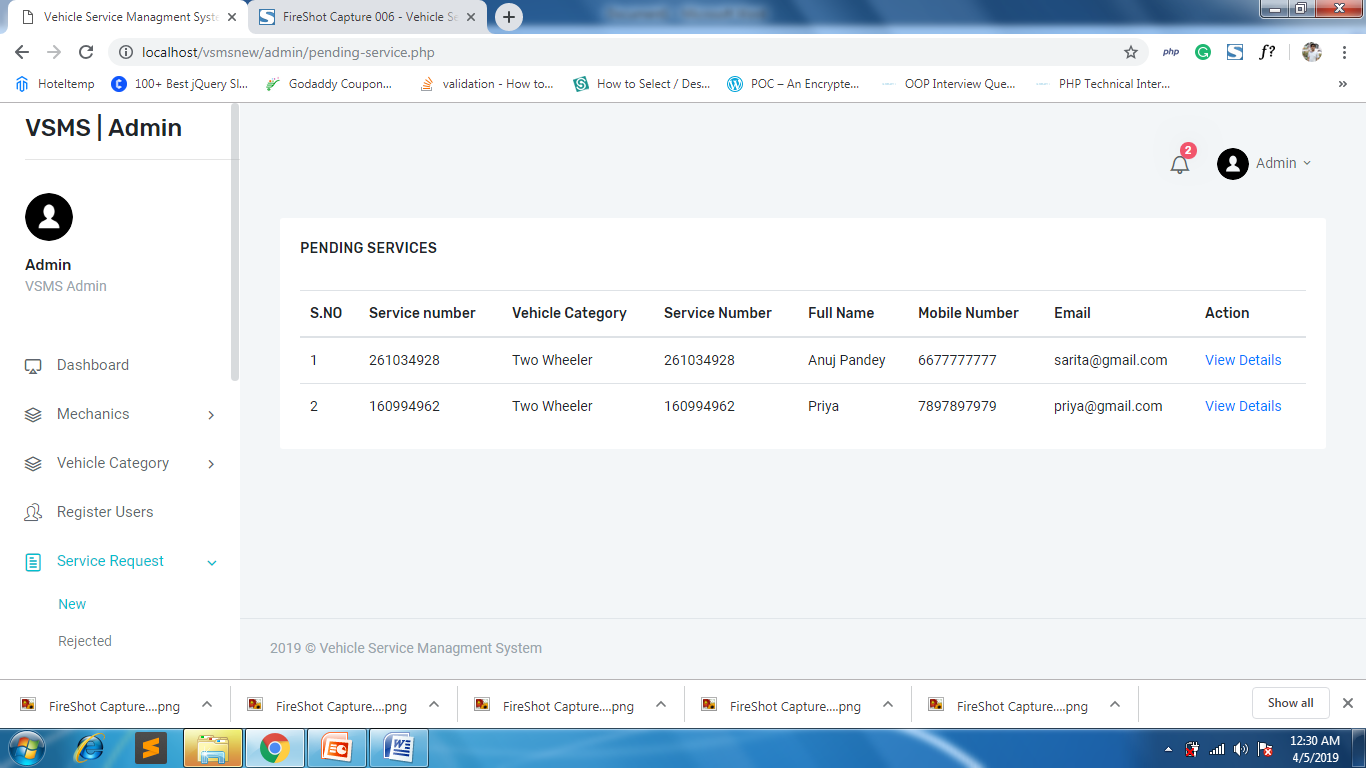
**Figure 24: Manage Mechanics**



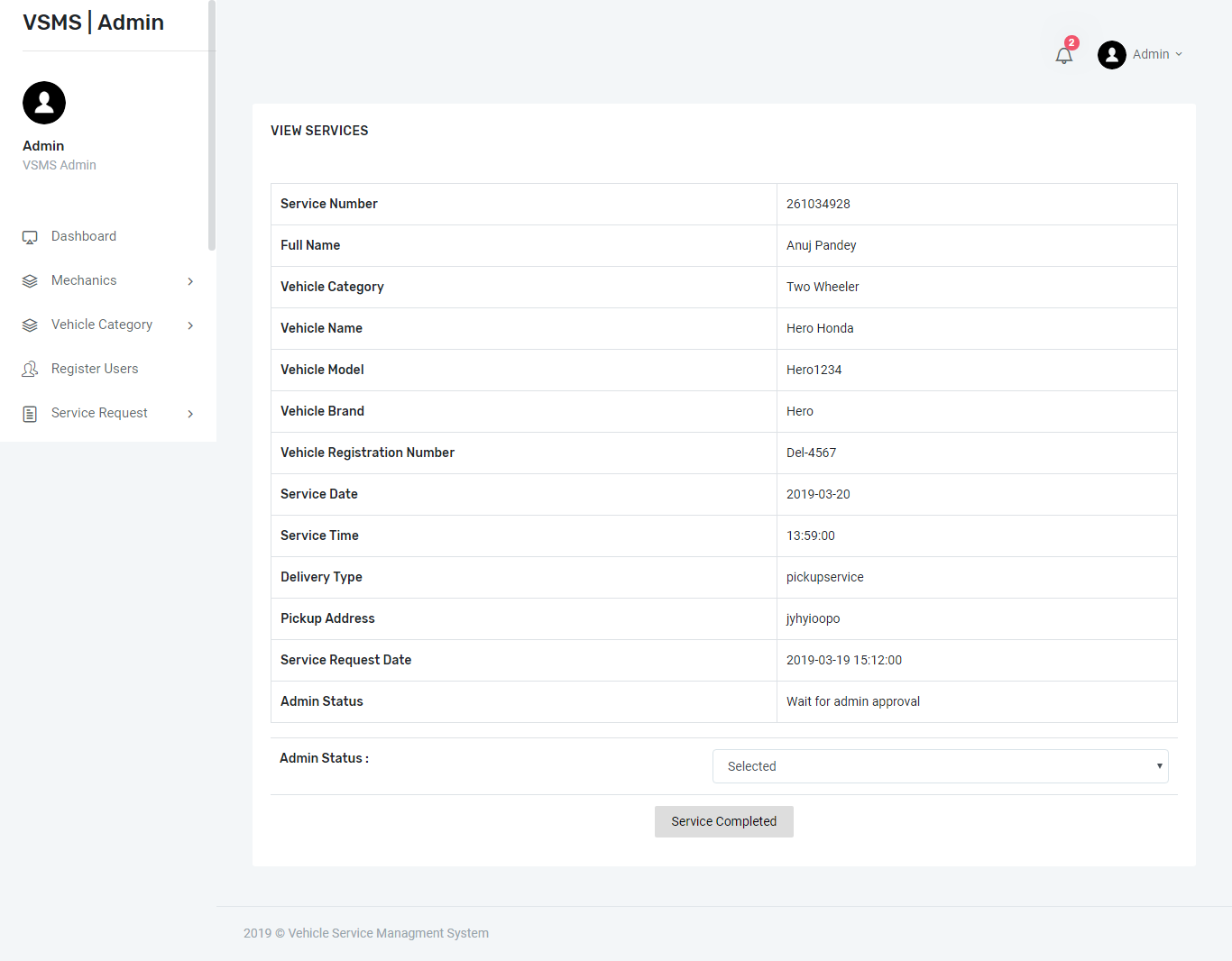
**Figure 25: Update Mechanic Details**



**Figure 26: User Management**



**Figure 27: Pending Service Requests**



**Figure 28: Service Management and Invoice Generation**

CHAPTER 7

**Summary**

**&**

**Conclusion**

**7.1 Result of the Proposed System**

This development has given everybody a software package that will provide user-friendly environment, which is very easy to work with, even for people with very little knowledge of computer.

Management of various tasks is incorporated in the package and will deliver the required information in a very easy to use and easy to access manner.

This package will provide accuracy, efficiency, speed and easiness to the end user. Since the system is verified with valid as well as invalid data and is run with an insight into the necessary modifications that may require in the future, it can be maintained successfully without any problems.

This system gives option to users to book their vehicle service request at anytime and anywhere part of this world. At this time every user has shortage of time so Users can book services online this helps to save traveling time, fuel, money and other kinds of resources.

The following results have been achieved by implementing this system: -

1. Smooth flow of data without any hurdles.
2. Adequate validation checks for data entry.
3. Facility to update data from time to time.
4. Flexibility in the system according to the changing environment.
5. Accuracy, timeliness and comprehensiveness of the system output.
6. Stability and operability by people of average intelligence.
7. With the amalgamation of bootstrap mobile computability of webpage has been achieved.

**7.1 Limitations of the System**

* The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
* Lack of OTP based authentication due to time constraints.

**References:**

1. **Bootstrap Examples-** <https://getbootstrap.com/docs/4.5/examples/>
2. Gamrat B., "PHP and preprocessed Web pages," Dr.Dobb'S Journal, January 2006, 31(1), pp. 46-48
3. Hasin Hayder, J.P.Maia, L.Gheorge, "Smarty PHP Template Programming And Applications," Packt Publishing, 2006, pp. 23-33
4. Veglis A., Leclercq M., Quema V., "PHP and SQL made simple, Distributed Systems Online," Aug 2005, 6(8),pp. 15-22