A MAJOR PROJECT

on

AUTOMOBILE MANAGEMENT SYSTEM

Submitted

In partial fulfillment for the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering.

By

SYED FAYAZ AHMED 20641A05K0

Under the Guidance of

Mr.P.Prudvi Krishna

Assistant professor

Department of Computer Science & Engineering

Vaagdevi College of Engineering

(UGC Autonomous, Accredited by NAAC with "A")

Bollikunta, Khila Warangal (Mandal), Warangal Urban – 506005(T.S)

(2020-2024)

VAAGDEVI COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(UGC Autonomous, Accredited by NBA, Accredited by NAAC with “A”)

Bollikunta, Khila Warangal (Mandal), Warangal Urban –506005(T.S)

CERTIFICATE

This is to certify that the major project entitled *“**AUTOMOBILE MANAGEMENT SYSTEM “*** is submitted by **SYED FAYAZ AHMED 20641A05K0** in partial fulfillment of the requirements for the award of the Degree in Bachelor of Technology in Computer Science and Engineering during the academic year 2023-2024.

Mr.P.Prudvi Krishna
Project Guide

Dr.N.Satyavathi
Head of the Department

External Examiner

ACKNOWLEDGEMENT

We wish to take this opportunity to express my sincere gratitude and deep sense of respect to beloved **Dr.K.Prakash, Principal,** Vaagdevi College of Engineering for making me available all the required assistance and for his support and inspiration to carry out this industry Oriented Major Project in the institute.

We extend our heartfelt thanks to **Dr.N.Sathyavathi, Head of the Department of CSE**, Vaagdevi College of Engineering for providing us necessary infrastructure and thereby giving us freedom to carry out the Industry Oriented Major Project.

We express heartfelt thanks to the guide, **Mr.P.Prudvi Krishna, Assistant professor**, Department of CSE for his constant support and giving necessary guidance for completion of this Industry Oriented Major Project.

We are also thankful to **Dr.K.Rekha Devi, Mr.Ch.Aravind Kumar** for their valuable suggestions, encouragement and motivations for completing this project successfully.

We are thankful to all other faculty members for their encouragement. We convey our heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

SYED FAYAZ AHMED   20641A05K0

DECLARATION

We hereby declare that this project entitled "**AUTOMOBILE MANAGEMENT SYSTEM*"*** is submitted in partial fulfillment of requirements for the award of bachelor of technology at **VAAGDEVI COLLEGE OF ENGINEERING** affiliated to Jawaharlal Nehru Technology University. The report has not been submitted either in part or full for degree earlier to this University.

SYED FAYAZ AHMED 20641A05K0

Abstract

In Today's world with so many cars, models in the market, it is hard to find out which car has a high maintenance cost/index that is authentic source Problem statement in Description o System that shows the health and Maintenance Index of various components of car models or car parts based on multiple factors. This will help new buyers to understand the maintenance costs of a certain model and probability of which car part requires more often servicing /change, OEM's to understand which part is requiring frequent change and needs to be recalled and made better in the new models. Vehicle maintenance patterns across car models by mileage, usage , age of the vehicle , regional patterns across dealers, service stations and car manufacturer. Purpose and who will benefit o Consumers so they know which car has a higher maintenance and maintenance index. Car Manufacturers , so they know which parts are getting serviced often based on the part change. How does it help the nation o Better understanding for consumers on which car to purchase with low Vehicle Maintenance Index. Govt has better understanding of car maintenance index before approving cars on the Road (ARAI Authority). Practical and reasons why this idea could be a challenge from Implementation , Data Challenge. Build a dealer/service station network , OEM, consumers who can feed data into the system – without data this solution will not work. Ability to integrate this solution easily with the current systems which can collate the data o Marketing challenge : Owner within the Govt to take this ahead and invest this product .Political : OEM's might not want such a system to be developed. Domain Bucket , Transport , Vehicles ,Technology Bucket , Integration across systems – Integration platform across systems with security model for data extraction Analytics & Data Science (if possible) - Bigdata/Hadoop, AI / ML , Data Aggregation o Visualization.

INDEX

FIGURES INDEX

Figure No: **Page No.**

CHAPTER-1

INTRODUCTION

More than two –third of automotive customers indicate that Service Convenience is a determining in selecting a brand or purchasing from a Specific dealership .Consequently, Customer Service and service management is of vital relevance to ensure ongoing Customer loyalty and retention and ultimately, dealer Profitability. Dealer Business Management enables efficient Service order Processing and billing designed for any given number of orders per day it includes Service requests and Scheduling ,Optimization of techicans,tools, and parts as well as their deployment and scheduling .User friendly interfaces provides ease of use and the integration technology ensure a seamless and smooth business process into OEM channel System, including ,job and Package Catalogs, Vehicle history files ,Warranty Systems, etc. Service monitoring and analysis increases the Visibility in Overall fixed Operations and helps increases service Capacity utilization, efficiency and decreased operating cost. Recreation Vehicle service technician inspect ,test, service, and replace every system installed in a recreation Vehicle with the exception of the dry. Both intervals are equally important for properly marinating your Vehicle Remember all Toyota dealerships offer a broad range of Parts and Service. The Driver and Vehicle Licensing Agency ,the Driving Standards Agency and the Vehicle and Operator Service Agency Provide Services for 42 million drivers.The Vehicle and Operator Services Agency(VOSA) Provides a range of licensing, testing andenforcement services with the aim of improving the roadworthiness standards of Vehicles ensuring the Compliance of operators and drivers , and Supporting the independent Traffic Commissioner.

1.1 LITERATURE SURVEY

TITLE: Supporting decision-making for solving design issues in the development phase of automotive vehicles.

Author: Timothé M. Sissoko

ABSTRACT: - The purpose of this research is to support decision making for solving design issues in the development phase of complex systems supported by numerical simulation. We conducted our studies in a multinational car manufacturer. The first part of the research was devoted to identifying the difficulties encountered in the issue resolution process, with a particular focus on decision-making issues, methods and tools. A qualitative study done with 11 experts and on 40 decision problems highlighted that the decision-makers choose from a set of process alternatives rather than artifact alternatives. The consequences of these process alternatives such as recalculating, integrating information, waiting for the technical definition of the vehicle to evolve, etc. are not explicit. We identified the lack of a rigorous framework as an opportunity for improvement. The second part was therefore to propose a framework to support design decisions. Concurrent engineering, resources constraints and project management issues have been often overlooked in the Decision Based Design literature. Attempting to bridge this gap, we designed IRDS framework. Through IRDS, we propose to make explicit the process alternatives, to gather economic data and expert forecasts in adecision model based on the prescriptive decision theory, including the maximization of the expected utility and the economic value of imperfect information. The third part of the research is related to the impact of uncertainty on the data collection process and on the overall decision outcomes. This has been done through proposing a sensitivity analysis that is performed with available data, before data gathering through elicitation process. The impacts on the decision-making process and information exchanges between stakeholders, as well as the resources consumed by the new practices we proposed have also been studied on a more superficial level. This work was in particular deployed and tested on 5 cases studies. The validation of this approach requires to collect further empirical evidence to support the hypothesis that better decisions are made on the long run. We are confident that our research will serve as a base for future studies on the design and the implementation of frameworks addressing industrial challenges.

Problem Statements :

Improving Decision-Making Processes in Automotive Design:

How can decision-making processes be enhanced to address challenges in resolving design issues during the development phase of automotive vehicles?

Enhancing Decision-Based Design Frameworks for Automotive Industry:

What improvements can be made to existing Decision-Based Design frameworks to better accommodate concurrent engineering, resource constraints, and project management issues specific to the automotive industry?

Development of a Rigorous Decision Support Framework:

What components should be included in a rigorous decision support framework to explicitly address process alternatives, gather economic data, and integrate expert forecasts for design decisions in automotive vehicle development?

Mitigating the Impact of Uncertainty on Decision Outcomes:

How can uncertainty surrounding data collection processes and decision outcomes be effectively mitigated to improve decision-making in automotive design?

Validation of an Integrated Decision Support Framework:

What empirical evidence is needed to validate the effectiveness of the Integrated Resource Decision Support (IRDS) framework in improving decision outcomes in automotive vehicle development?

TITLE: Design and implementation of Vehcile maintenance and service system for hallmark automobile service

Authors: Mario Collotta

ABSTRACT: The vehicle maintenance and service system is an automation of the various vehicle services needed by a vehicle user in a mobile application. This application provides vehicle service reminder while providing vehicle related solution. The vehicle maintenance and service system provide repair cost estimates to help mitigate the rigged system vehicle users pass through for vehicle repairs and maintenance. This project work contains a review of existing systems related to the proposed system and the prototype development

methodology used to develop the system. And also provides the design of the system using the Unified Modelling Language as well as testing of the system. The examination of the literature uncovered a number of empirical evidence areas, and the resulting discussion provides the basis for the design and implementation of vehicle maintenance and services system.

Problem Statements :

Improving Vehicle Maintenance and Service Experience for Users:

How can the vehicle maintenance and service system be designed and implemented to enhance the overall experience of vehicle users, including providing timely service reminders and repair cost estimates?

Addressing Challenges in Existing Vehicle Maintenance Systems:

What are the shortcomings of existing vehicle maintenance systems, and how can the proposed system overcome these challenges to provide more efficient and user-friendly solutions?

Enhancing Transparency and Cost-Efficiency in Vehicle Repairs:

How can the proposed system help mitigate the lack of transparency and inflated costs often associated with vehicle repairs and maintenance, thereby providing users with more accurate repair cost estimates and reducing their reliance on potentially unreliable service providers?

Optimizing System Design and Development Methodologies:

What methodologies and design approaches should be employed in the development of the vehicle maintenance and service system to ensure its effectiveness, usability, and scalability?

Integration of Unified Modeling Language (UML) in System Design:

How can the Unified Modeling Language (UML) be effectively utilized in the design of the vehicle maintenance and service system to facilitate clear communication, enhance system understanding, and streamline the development process?

TITLE: Predictive Maintenance in the Automotive Sector
Author: Marianna Ruggieri

ABSTRACT: -With the rapid advancement of sensor and network technology, there has been a notable increase in the availability of condition-monitoring data such as vibration, temperature, pressure, voltage, and other electrical and mechanical parameters. With the introduction of big data, it is possible to prevent potential failures and estimate the remaining useful life of the equipment by developing advanced mathematical models and artificial intelligence (AI) techniques. These approaches allow taking maintenance actions quickly and appropriately. In this scenario, this paper presents a systematic literature review of statistical inference approaches, stochastic methods, and AI techniques for predictive maintenance in the automotive sector. It provides a summary on these approaches, their main results, challenges, and opportunities, and it supports new research works for vehiclepredictive maintenance.

Problem Statements :

Optimizing Predictive Maintenance Strategies for Automotive Equipment:

How can advanced mathematical models and artificial intelligence (AI) techniques be effectively utilized to optimize predictive maintenance strategies for automotive equipment, leveraging condition-monitoring data such as vibration, temperature, pressure, and other electrical and mechanical parameters?

Enhancing Failure Prediction and Remaining Useful Life Estimation:

What statistical inference approaches, stochastic methods, and AI techniques can be employed to improve the accuracy of failure prediction and estimation of remaining useful life for automotive equipment, based on the analysis of big data and condition-monitoring data?

Overcoming Challenges in Implementing Predictive Maintenance in the Automotive Sector:

What are the main challenges associated with implementing predictive maintenance in the automotive sector, and how can these challenges be addressed to enable quick and appropriate maintenance actions, thereby reducing downtime and enhancing equipment reliability?

Integration of Statistical Inference, Stochastic Methods, and AI Techniques:

How can statistical inference approaches, stochastic methods, and AI techniques be integrated effectively to develop comprehensive predictive maintenance solutions for the automotive sector, considering the diverse range of data sources and parameters involved?

Identifying Opportunities for Future Research in Vehicle Predictive Maintenance:

What are the key research gaps and opportunities identified through the systematic literature review of predictive maintenance approaches in the automotive sector, and how can these findings guide new research initiatives to further advance predictive maintenance capabilities for vehicles?

1.2 EXISTING SYSTEM

Existing system is a manual one in which users are maintaining books to store the information like product details, Distributors details, purchases, sales details and accounts for every month. It is very difficult to maintain historical data.The following are the disadvantages of the existing system. It is difficult to maintain important information in books. More manual hours need to generate required reports. It is tedious to manage historical data which needs much space to keep all the previous years' ledgers, books etc. Daily sales and purchases details must be entered into books are very difficult to maintain.

DISADVANTAGES OF EXISTING SYSTEM

- Increased Repair Costs: When vehicles are not maintained regularly, small issues can become bigger problems that require expensive repairs. Over time, the cost of these repairs can add up, leading to increased expenses and reduced profits
- Decreased Safety: Vehicles that are not properly maintained are more likely to experience mechanical failures or breakdowns while on the road. This can put the safety of drivers and passengers at risk, as well as the safety of other drivers on the road.
- Reduced Vehicle Lifespan: Regular maintenance can help extend the lifespan of a vehicle. When vehicles are not maintained properly, they may require replacement sooner than expected, which can be a significant expense.

1.3 PROPOSED SYSTEM

The DISTRIBUTORS MANAGEMENT TOOL is a software application which avoids more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the distributors to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. It is very easy to record the information of online sales and purchases in the databases.

ADVANTAGES OF PROPOSED SYSTEM

- Reduced Repair Costs: By addressing small issues before they become bigger problems, organizations can help reduce repair costs over time. This can help save money and increase profitability.

- Extended Vehicle Lifespan: Regular maintenance can help extend the lifespan of vehicles, reducing the need for replacement and saving money in the long term.

- Increased Efficiency: Vehicles that are well-maintained are often more efficient, leading to lower fuel costs and increased productivity. This can help organizations save money and increase profits.

- Improved Resale Value: Vehicles that have been well-maintained and have a documented maintenance history often have a higher resale value than those that have not. This can help organizations recoup some of the costs of their fleet of vehicles when it comes time to sell them.

1.4 SYSTEM SPECIFICATION

1.4.1 HARDWARE REQUIREMENTS

System	: Pentium IV 2.4 GHz.
Hard Disk	: 40 GB.
Floppy Drive	: 1.44 Mb.
Monitor	: 15 VGA Colour.
Mouse	: Logitech.
Ram	: 1GB

1.4.2 SOFTWARE REQUIREMENTS

Operating system	: Windows 10.
Coding Language	: JAVA.
Data Base	: MS SQL

CHAPTER-2

SYSTEM STUDY

2.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

2.2 FEASIBILITY ANALYSIS

Three key considerations involved in the feasibility analysis are

- ♦ ECONOMICAL FEASIBILITY
- ♦ TECHNICAL FEASIBILITY
- ♦ SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

.

CHAPTER-3

SYSTEM DESIGN

3.1 ARCHITECTURE DIAGRAM

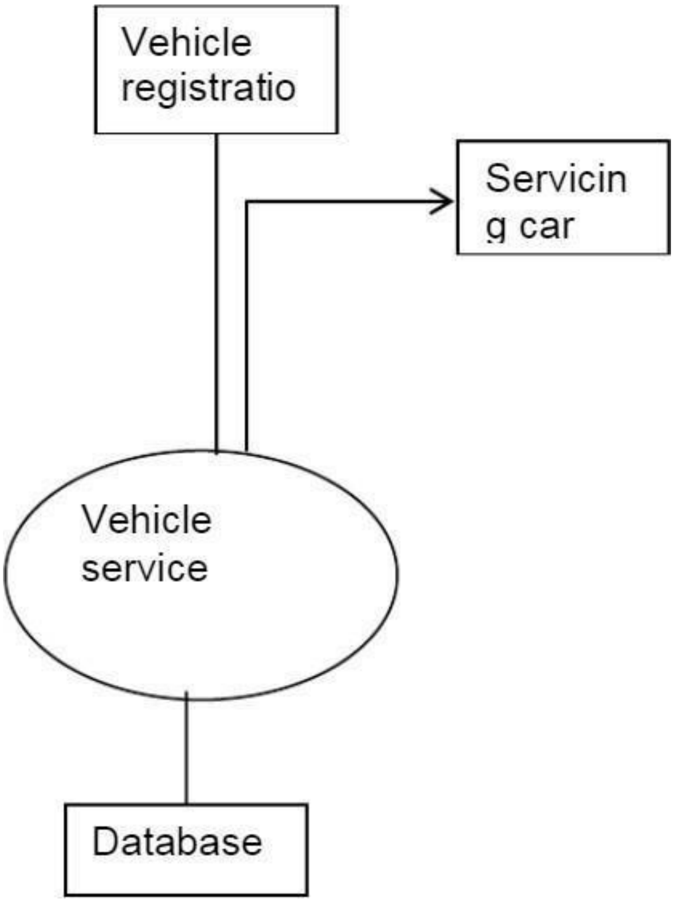


Fig 3.1 Architecture Diagram

3.2 DATA FLOW DIAGRAM

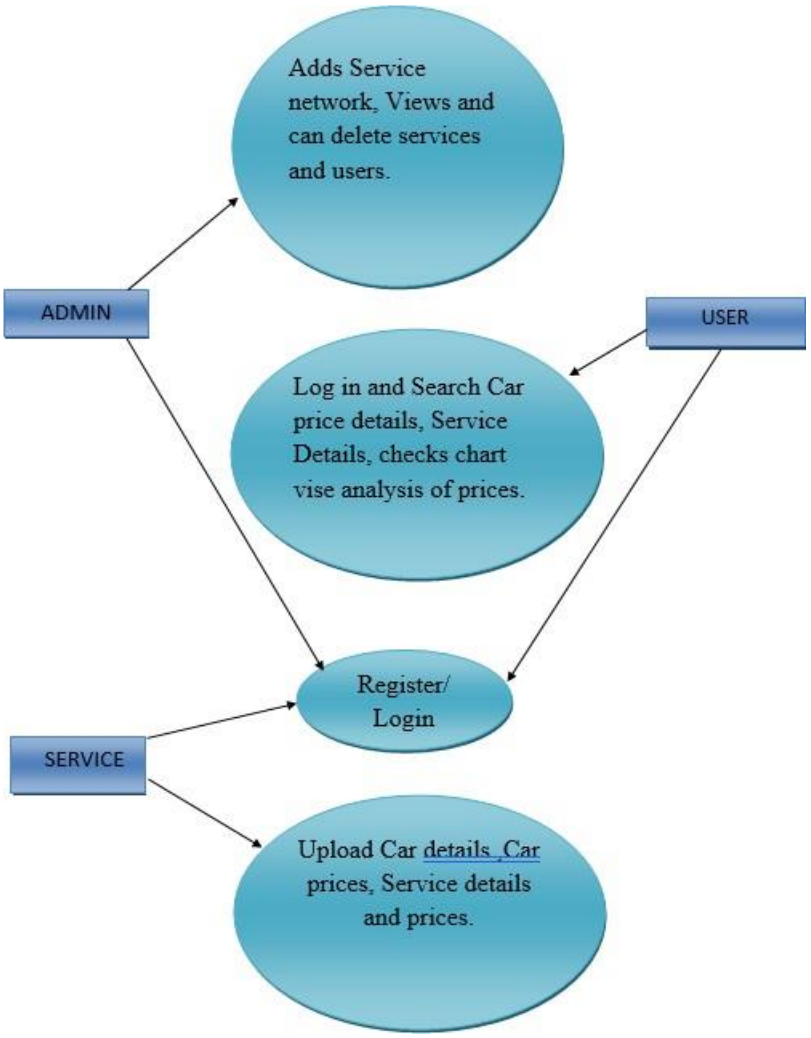

Adds Service network, Views and can delete services and users.
ADMIN
USER
Log in and Search Car price details, Service Details, checks chart vise analysis of prices.
Register/ Login
SERVICE
Upload Car details ,Car prices, Service details and prices.

Fig 3.2 Data Flow Diagram

3.3 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components:

- A Meta-model and a notation. In the future, some form of method or process may also beaddedto; or associated with, UML.
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelingand other non-software systems.
- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

3.3.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

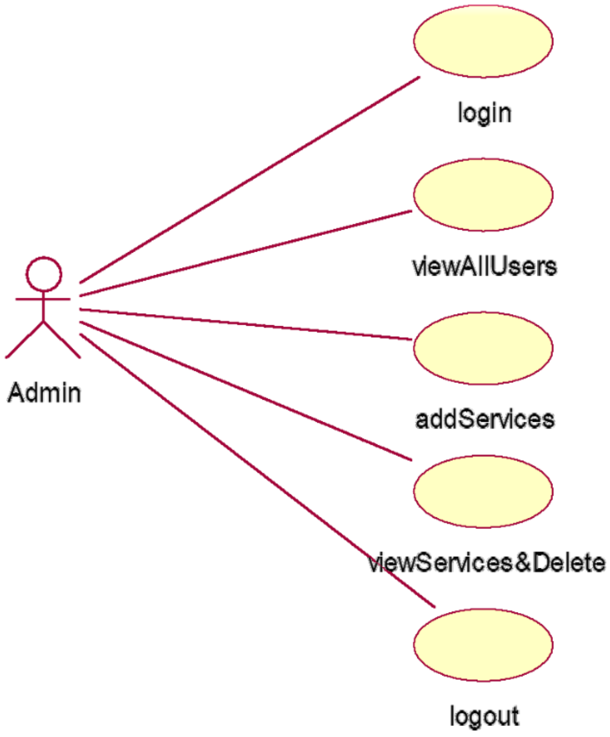


Fig 3.3 Admin Use Case Diagram

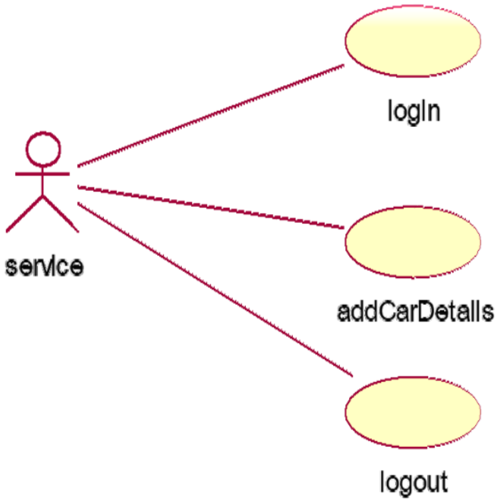


Fig 3.4 Service Use Case Diagram

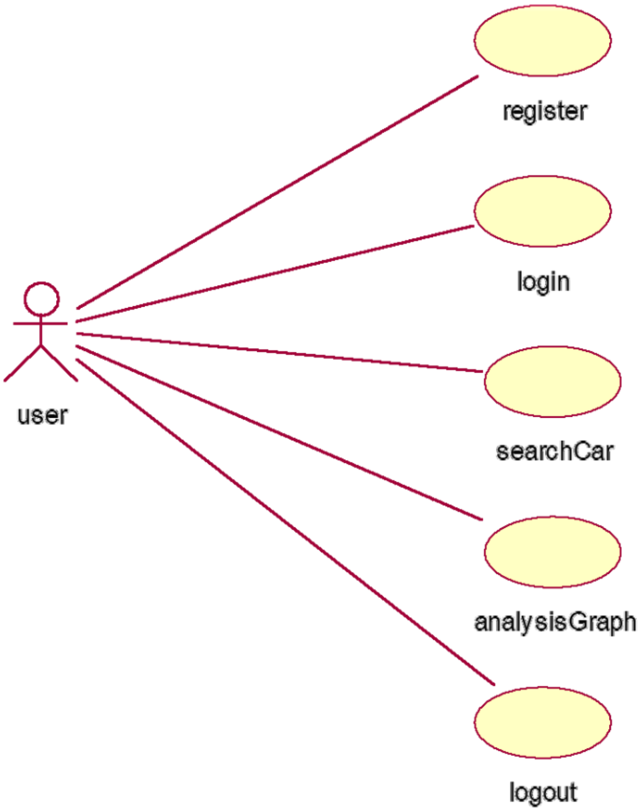


Fig 3.5 User Use Case Diagram

3.3.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

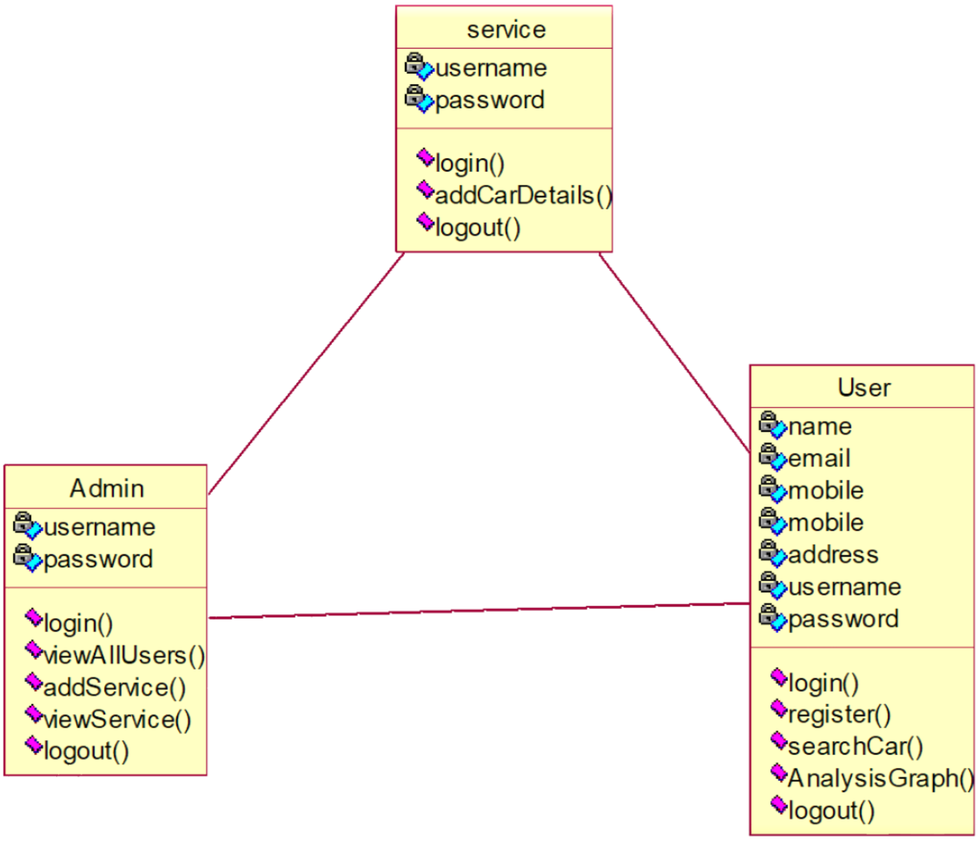


Fig 3.6 Class Diagram

3.3.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

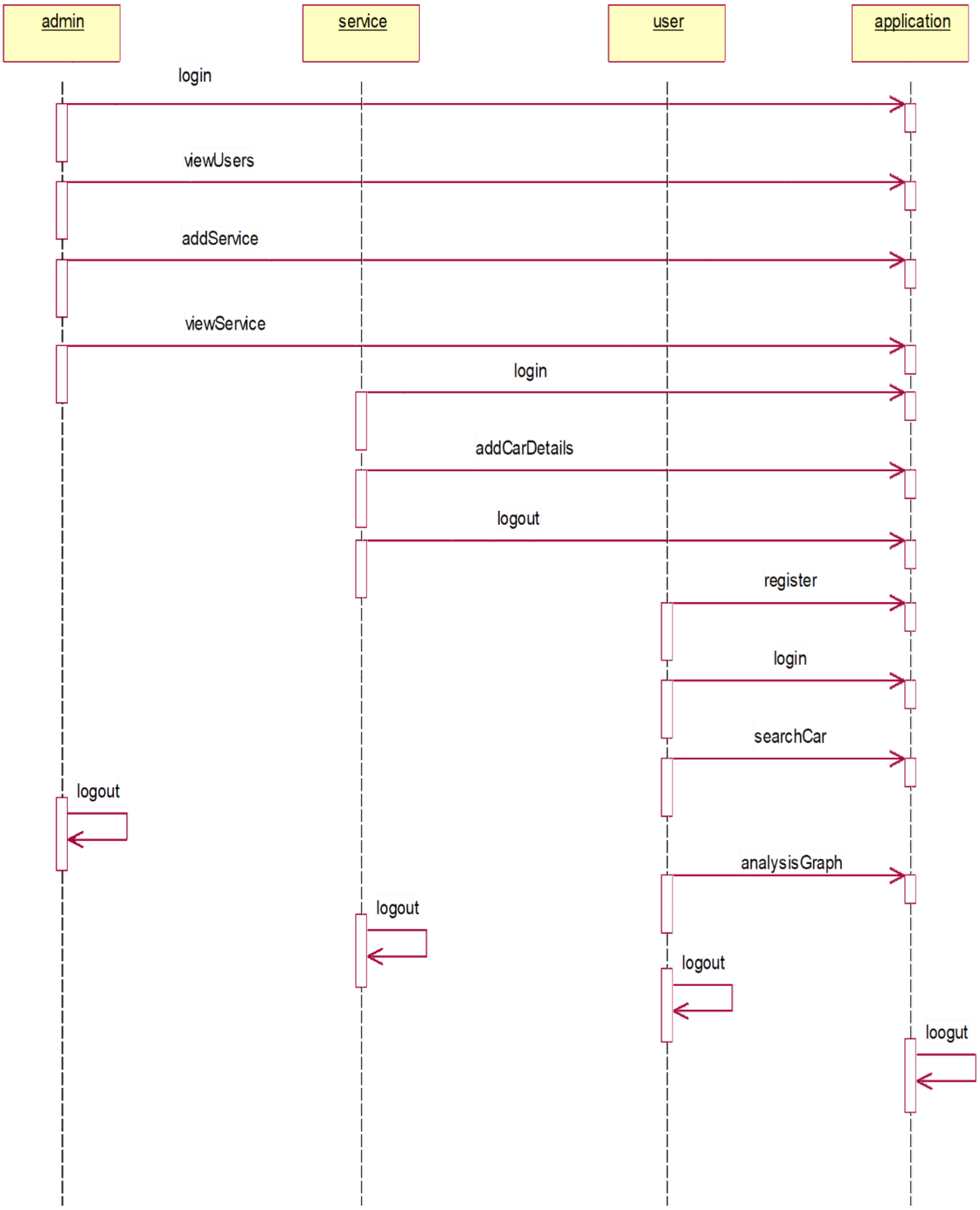


Fig 3.7 Sequence Diagram

3.3.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

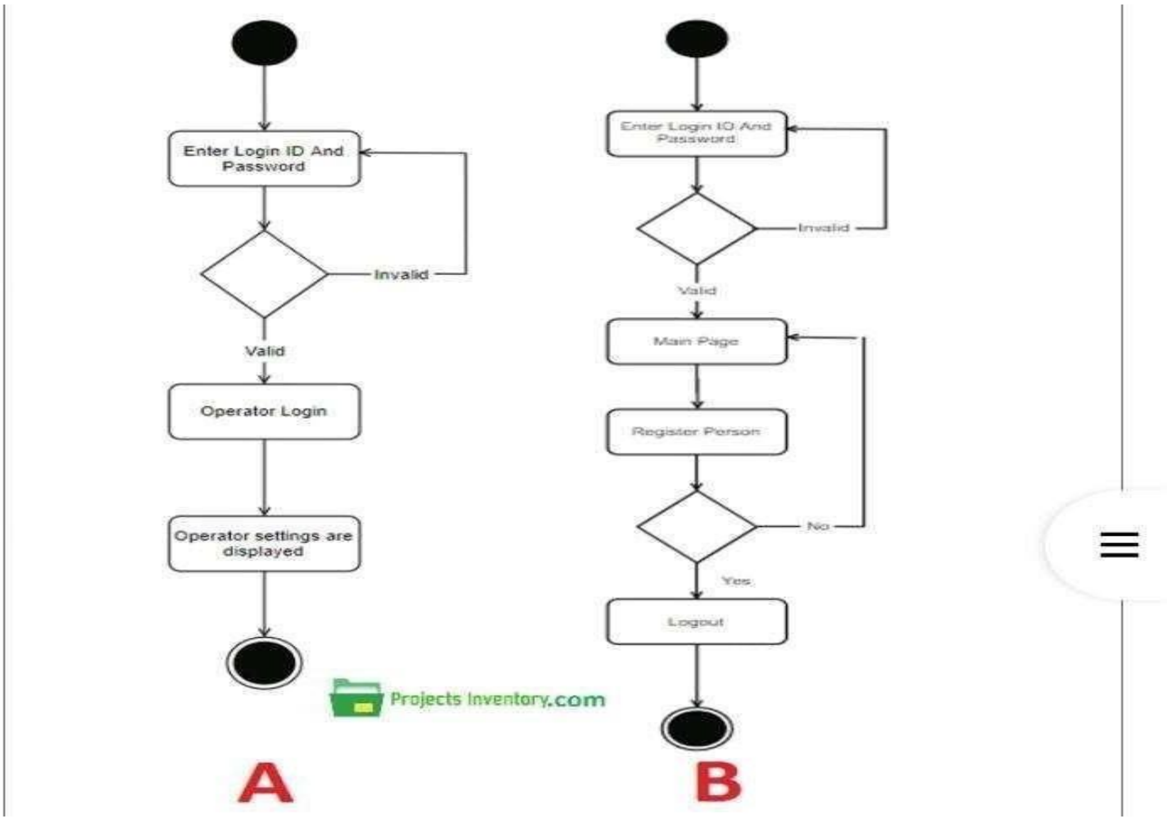


Fig 3.8 Activity Diagram

3.3.5 COLLABRATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is a visual representation of how objects in a system interact with each other. It's a behavioral diagram that shows how objects work together to achieve specific tasks or scenarios in a system. Collaboration diagrams are a fundamental part of the Unified Modeling Language (UML).

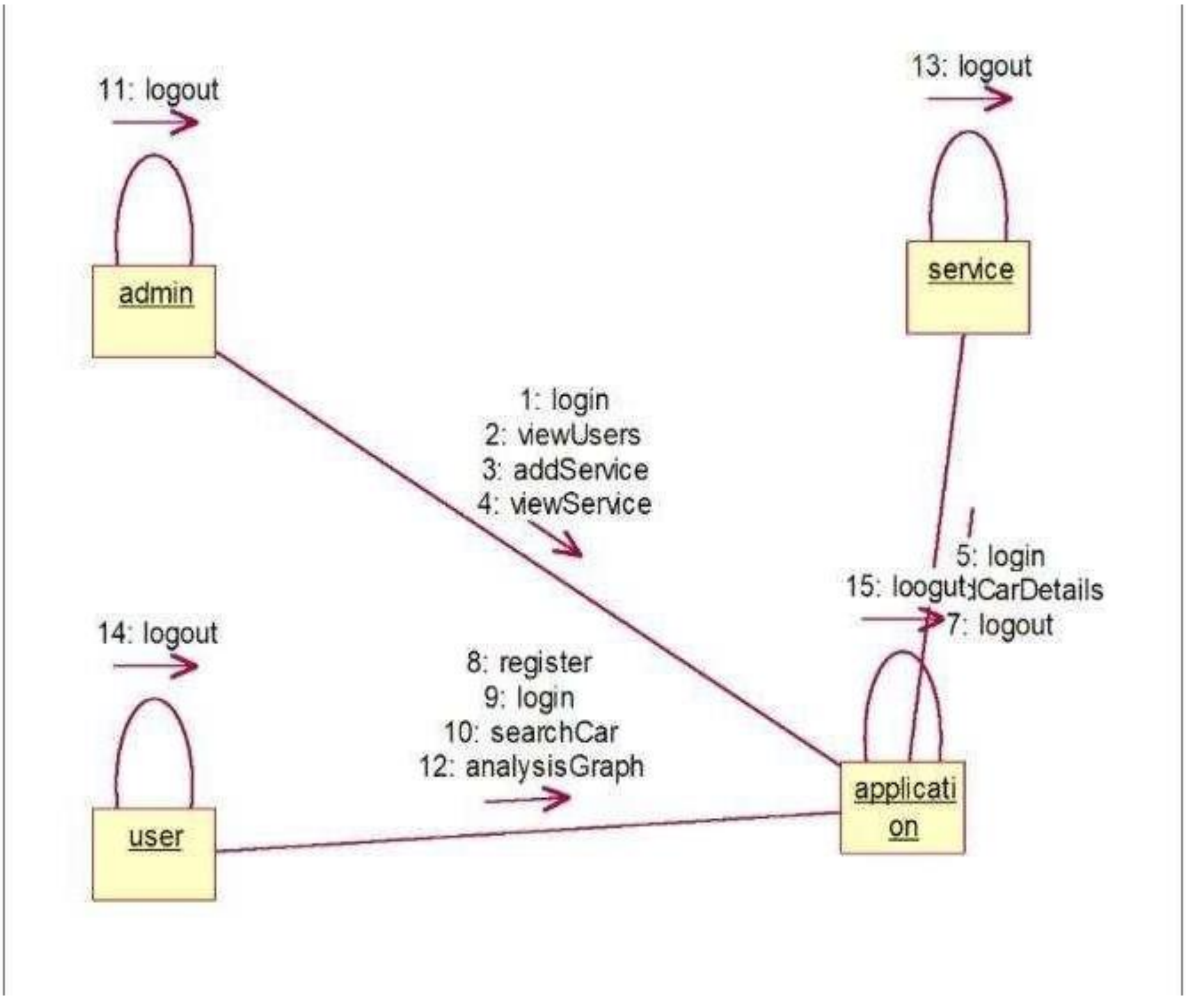


Fig 3.9 Collaboration Diagram

3.3.6 COMPONENT DIAGRAM

A component diagram is a Unified Modeling Language (UML) diagram that shows the structure of a software system by illustrating how components are wired together to form larger components

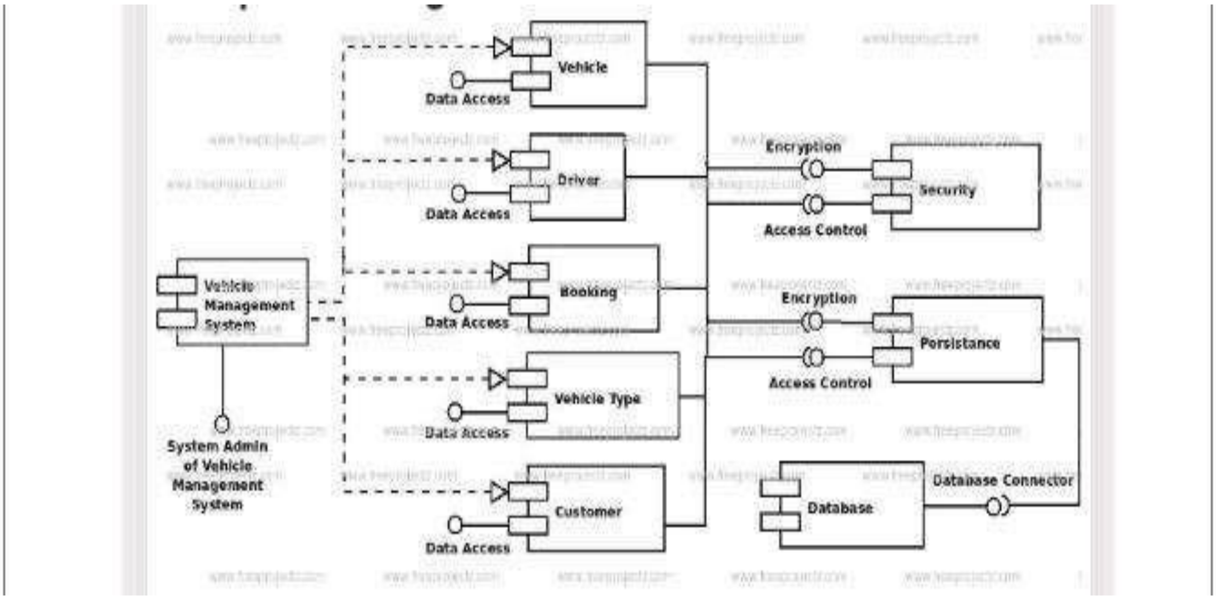


Fig 3.10 Component Diagram

3.4 E-R DIAGRAM

The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.

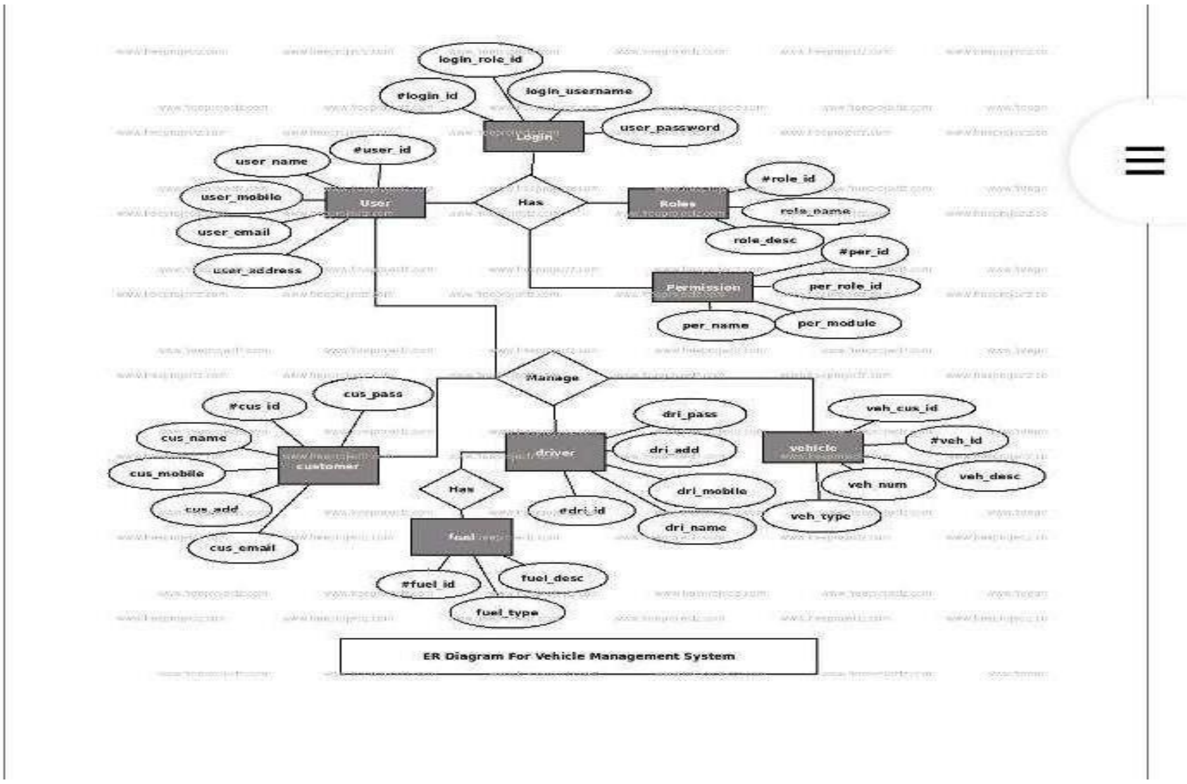


Fig 3.11 E-R Diagram

CHAPTER-4

INPUT AND OUTPUT DESIGN

4.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

4.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

CHAPTER-5

IMPLEMENTATION

5.1 MODULES

- Admin
- User
- Service

MODULES DESCRIPTION

ADMIN

In this application admin is the module, here admin can login directly with the application and after login successful admin can perform operations such as ViewAllUsers and addServices add viewServices and logout.

USER

In this application user is a module, here user should register with the application then only the user can access his home page after successful login he can perform some operations such as viewProfile, searchCarDetails, analysisGraph and logout

SERVICE

In this application services is a module, here service should be added by the admin then only the services can access his home page after successful login he can perform some operations such as addCarDetails and logout

5.2 JAVA TECHNOLOGY

Java technology is both a programming language and a platform.

The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called ***Java byte codes*** —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.

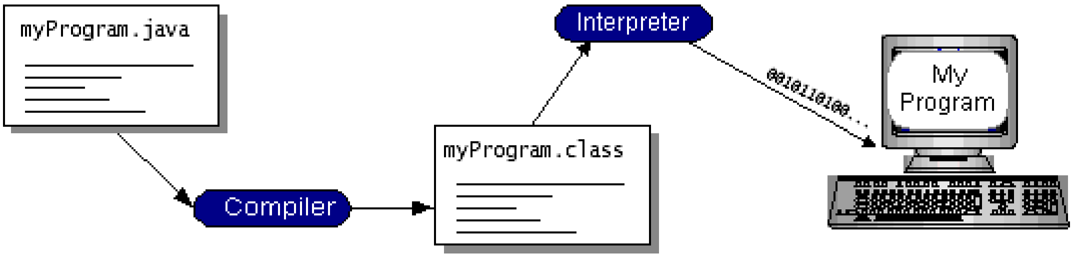
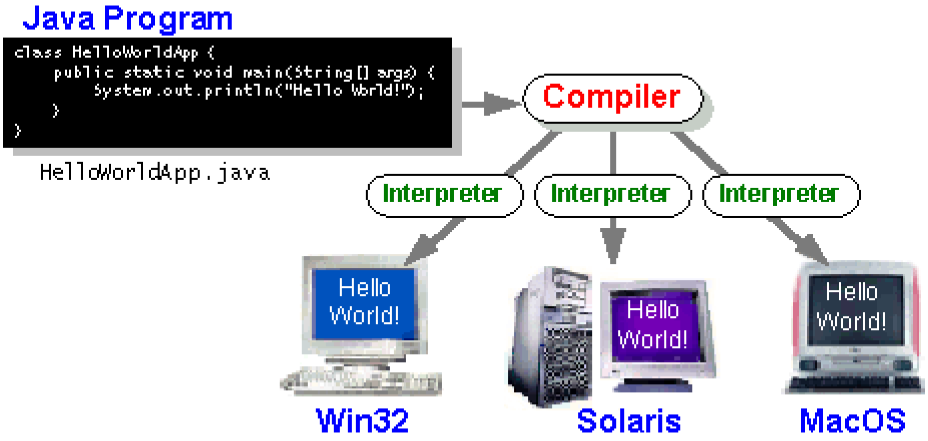


Fig 5.1 JAVA Working

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (JVM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

Fig 5.2 JAVA on different OS

The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine (Java VM)

- The Java Application Programming Interface (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as ***packages***. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

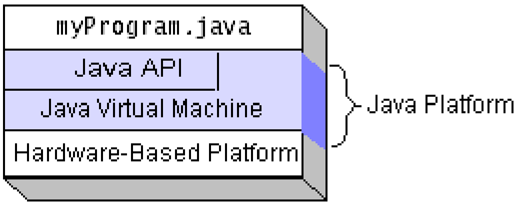


Fig 5.3 Program running on Java platform

Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a ***server*** serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a ***servlet***. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials**: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets**: The set of conventions used by applets.
- **Networking**: URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization**: Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security**: Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components**: Known as JavaBeans^{TM}, can plug into existing component architectures.
- **Object serialization**: Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC^{TM})**: Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.

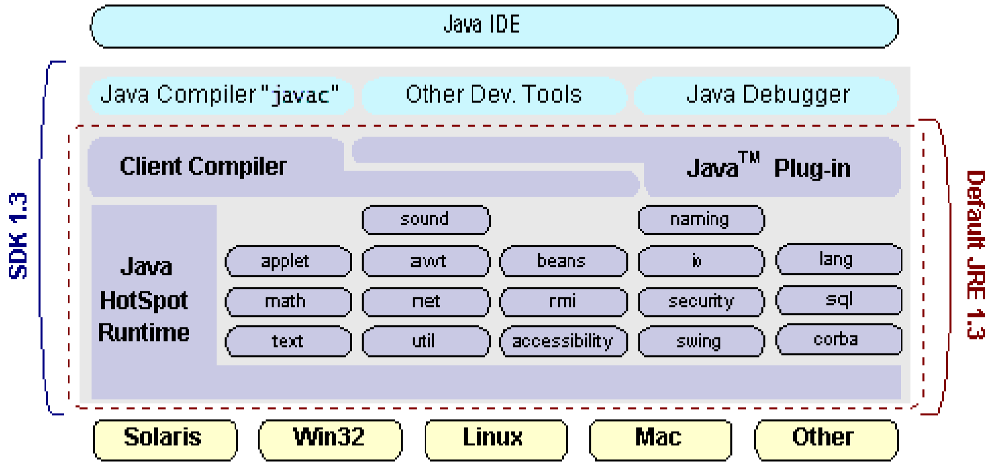


Fig 5.4 Java 2 SDK

How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

- **Get started quickly**: Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.
- **Write less code**: Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.
- **Write better code**: The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.
- **Develop programs more quickly**: Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java**: You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java^{TM} Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

- **Write once, run anywhere**: Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- **Distribute software more easily**: You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded “on the fly,” without recompiling the entire program.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn’t change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of "plug-in" database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on. To gain a wider acceptance of JDBC, Sun based JDBC's framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after. The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. **SQL Level API**

   The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to "generate" JDBC code and to hide many of JDBC's complexities from the end user.

SQL Conformance

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

2. **JDBC must be implemental on top of common database interfaces**

   The JDBC SQL API must "sit" on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

3. **Provide a Java interface that is consistent with the rest of the Java system**

   Because of Java's acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

4. **Keep it simple**

   This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

5. **Use strong, static typing wherever possible**

   Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

6. **Keep the common cases simple**

   Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible

Java has two things: a programming language and a platform. Java is a high-levelprogramming language that is all of the following

Simple
Object-oriented
Distributed
Interpreted
Robust
Secure
Architecture-neutral
Portable
High-performance
multithreaded
Dynamic

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed.

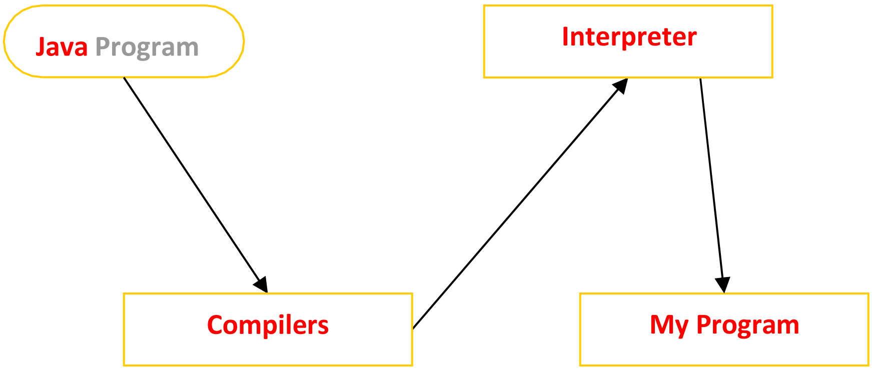


Fig 5.5 Compilation and interpretation of Java programs

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make "write once, run anywhere" possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

Networking

TCP/IP stack :

The TCP/IP stack is shorter than the OSI one:

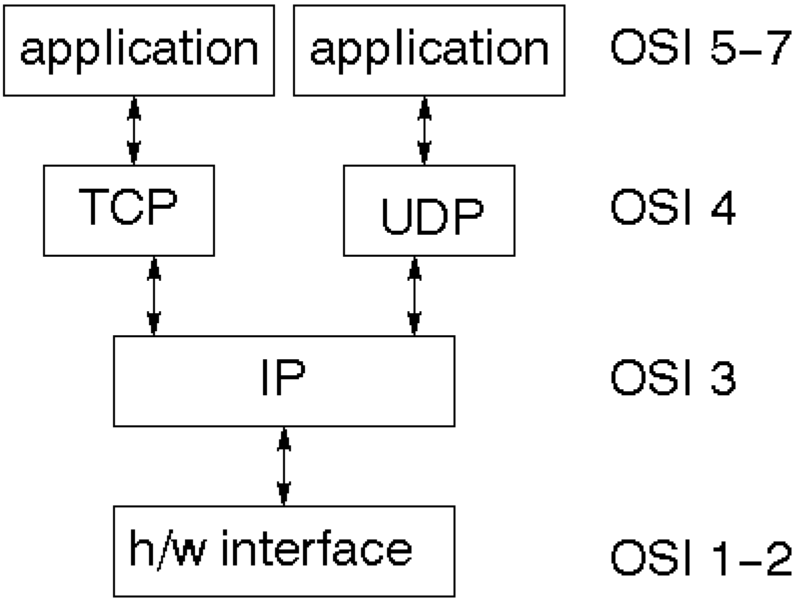


Fig 5.6 TCP/IP model

TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

Total address

Fig 5.7 IP address

The 32 bit address is usually written as 4 integers separated by dots.

Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call `socket`. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with `Read File` and `Write File` functions.

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(int family, int type, int protocol);
```

Here "family" will be `AF_INET` for IP communications, `protocol` will be zero, and `type` will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

2. Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

4. Property Editors

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

What is a Java Web Application?

A Java web application generates interactive web pages containing various types of markup language (HTML, XML, and so on) and dynamic content. It is typically comprised of web components such as JavaServer Pages (JSP), servlets and JavaBeans to modify and temporarily store data, interact with databases and web services, and render content in response to client requests.

Because many of the tasks involved in web application development can be repetitive or require a surplus of boilerplate code, web frameworks can be applied to alleviate the overhead associated with common activities. For example, many frameworks, such as JavaServer Faces, provide libraries for templating pages and session management, and often promote code reuse.

What is Java EE?

Java EE (Enterprise Edition) is a widely used platform containing a set of coordinated technologies that significantly reduce the cost and complexity of developing, deploying, and managing multi-tier, server-centric applications. Java EE builds upon the Java SE platform and provides a set of APIs (application programming interfaces) for developing and running portable, robust, scalable, reliable and secure server-side applications.

Some of the fundamental components of Java EE include:

- Enterprise JavaBeans (EJB): a managed, server-side component architecture used to encapsulate the business logic of an application. EJB technology enables rapid and simplified development of distributed, transactional, secure and portable applications based on Java technology.

- Java Persistence API (JPA): a framework that allows developers to manage data using object-relational mapping (ORM) in applications built on the Java Platform.

JavaScript and Ajax Development

JavaScript is an object-oriented scripting language primarily used in client-side interfaces for web applications. Ajax (Asynchronous JavaScript and XML) is a Web 2.0 technique that allows changes to occur in a web page without the need to perform a page refresh. JavaScript toolkits can be leveraged to implement Ajax-enabled components and functionality in web pages.

Web Server and Client

Web Server is a software that can process the client request and send the response back to the client. For example, Apache is one of the most widely used web server. Web Server runs on some physical machine and listens to client request on specific port.

A web client is a software that helps in communicating with the server. Some of the most widely used web clients are Firefox, Google Chrome, Safari etc. When we request something from server (through URL), web client takes care of creating a request and sending it to server and then parsing the server response and present it to the user.

HTML and HTTP

Web Server and Web Client are two separate softwares, so there should be some common language for communication. HTML is the common language between server and client and stands for **H**yper**T**ext **M**arkup **L**anguage.

Web server and client needs a common communication protocol, HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) is the communication protocol between server and client. HTTP runs on top of TCP/IP communication protocol.

Some of the important parts of HTTP Request are:

- **HTTP Method** – action to be performed, usually GET, POST, PUT etc.
- **URL** – Page to access
- **Form Parameters** – similar to arguments in a java method, for example user,password details from login page.

Sample HTTP Request:

```
GET /FirstServletProject/jsps/hello.jsp HTTP/1.1
Host: localhost:8080
```

3Cache-Control: no-cache

Some of the important parts of HTTP Response are:

- **Status Code** – an integer to indicate whether the request was success or not. Some of the well known status codes are 200 for success, 404 for Not Found and 403 for Access Forbidden.
- **Content Type** – text, html, image, pdf etc. Also known as MIME type
- **Content** – actual data that is rendered by client and shown to user.

MIME Type or Content Type: If you see above sample HTTP response header, it contains tag "Content-Type". It's also called MIME type and server sends it to client to let them know the kind of data it's sending. It helps client in rendering the data for user. Some of the mostly used mime types are text/html, text/xml, application/xml etc.

Understanding URL

URL is acronym of Universal Resource Locator and it's used to locate the server and resource. Every resource on the web has it's own unique address. Let's see parts of URL with an example.

http://localhost:8080/FirstServletProject/jsps/hello.jsp

http:// – This is the first part of URL and provides the communication protocol to be used in server-client communication.

localhost – The unique address of the server, most of the times it's the hostname of the server that maps to unique IP address. Sometimes multiple hostnames point to same IP addresses and web server virtual host takes care of sending request to the particular server instance.

8080 – This is the port on which server is listening, it's optional and if we don't provide it in URL then request goes to the default port of the protocol. Port numbers 0 to 1023 are reserved ports for well known services, for example 80 for HTTP, 443 for HTTPS, 21 for FTP etc.

FirstServletProject/jsps/hello.jsp – Resource requested from server. It can be static html, pdf, JSP, servlets, PHP etc.

Why we need Servlet and JSPs?

Web servers are good for static contents HTML pages but they don’t know how to generate dynamic content or how to save data into databases, so we need another tool that we can use to generate dynamic content. There are several programming languages for dynamic content like PHP, Python, Ruby on Rails, Java Servlets and JSPs.

Java Servlet and JSPs are server side technologies to extend the capability of web servers by providing support for dynamic response and data persistence.

Web Container

Tomcat is a web container, when a request is made from Client to web server, it passes the request to web container and it’s web container job to find the correct resource to handle the request (servlet or JSP) and then use the response from the resource to generate the response and provide it to web server. Then web server sends the response back to the client.

When web container gets the request and if it’s for servlet then container creates two Objects HTTPServletRequest and HTTPServletResponse. Then it finds the correct servlet based on the URL and creates a thread for the request. Then it invokes the servlet service() method and based on the HTTP method service() method invokes doGet() or doPost() methods. Servlet methods generate the dynamic page and write it to response. Once servlet thread is complete, container converts the response to HTTP response and send it back to client.

Some of the important work done by web container are:

- **Communication Support** – Container provides easy way of communication between web server and the servlets and JSPs. Because of container, we don’t need to build a server socket to listen for any request from web server, parse the request and generate response. All these important and complex tasks are done by container and all we need to focus is on our business logic for our applications.
- **Lifecycle and Resource Management** – Container takes care of managing the life cycle of servlet. Container takes care of loading the servlets into memory, initializing servlets, invoking servlet methods and destroying them. Container also provides utility like JNDI for resource pooling and management.

- **Multithreading Support** – Container creates new thread for every request to the servlet and when it's processed the thread dies. So servlets are not initialized for each request and saves time and memory.
- **JSP Support** – JSPs doesn't look like normal java classes and web container provides support for JSP. Every JSP in the application is compiled by container and converted to Servlet and then container manages them like other servlets.
- **Miscellaneous Task** – Web container manages the resource pool, does memory optimizations, run garbage collector, provides security configurations, support for multiple applications, hot deployment and several other tasks behind the scene that makes our life easier.

Web Application Directory Structure

Java Web Applications are packaged as Web Archive (WAR) and it has a defined structure. You can export above dynamic web project as WAR file and unzip it to check the hierarchy. It will be something like below image.

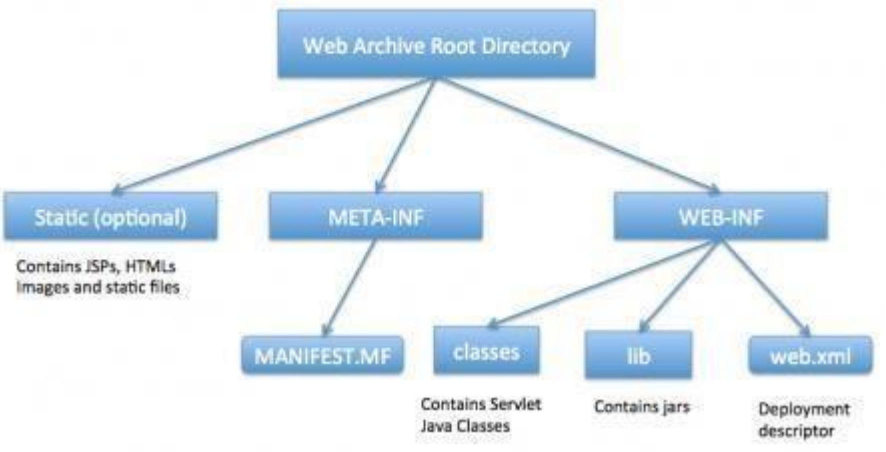

Web Archive Root Directory
Static (optional)
Contains JSPs, HTMLs Images and static files
META-INF
MANIFEST.MF
WEB-INF
classes
Contains Servlet Java Classes
lib
Contains jars
web.xml
Deployment descriptor

Fig 5.8 Web Archive

Deployment Descriptor

web.xml file is the deployment descriptor of the web application and contains mapping for servlets (prior to 3.0), welcome pages, security configurations, session timeout settings etc.

Thats all for the java web application startup tutorial, we will explore Servlets and JSPs more in future posts.

MySQL:

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and "pointers" between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL

(GNU General Public License), http://www.fsf.org/licenses/, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information (http://www.mysql.com/company/legal/licensing/).

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

You can find a performance comparison of MySQL Server with other database managers on our benchmark page.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

- **A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”), but we do not mind if you pronounce it as “my sequel” or in some other localized way.

5.3 SOURCE CODE

1. Admin.jsp

```
<%--
   Document : Admin
   Created on : 13 Apr, 2021, 12:08:42 PM
   Author     : KishanVenky
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- basic -->
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="viewport" content="initial-scale=1, maximum-scale=1">
    <!-- site metas -->
    <title>Vehicle Maintenance</title>
    <meta name="keywords" content="">
    <meta name="description" content="">
    <meta name="author" content="">
    <!-- bootstrap css -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <!-- style css -->
    <link rel="stylesheet" href="css/style.css">
```

```
         <link rel="stylesheet" href="table.css">
        <!-- Responsive-->
        <link rel="stylesheet" href="css/responsive.css">
        <!-- fevicon -->
       <!-- Scrollbar Custom CSS -->
        <link rel="stylesheet" href="css/jquery.mCustomScrollbar.min.css">
       </head>
      <!-- body -->
      <body class="main-layout">
        <!-- loader  -->
        <div class="loader_bg">
           </div>
        <!-- end loader -->
        <!-- header -->
        <header>
          <!-- header inner -->
          <div class="container-fluid">
             <div class="row">
                <div class="col-lg-3 logo_section">
                   <div class="full">
                      <div class="center-desk">
<div class="logo"> <h1>Vehicle Maintenance</h1></div>
                      </div>
                   </div>
                </div>
```

```
    <div class="col-lg-9">
      <div class="menu-area">
        <div class="limit-box">
          <nav class="main-menu">
            <ul class="menu-area-main">
              <li>
                <a href="index.html">Home</a>
              </li>
              <li class="active">
                <a href="Admin.jsp">Admin</a>
              </li>
               <li>
                <a href="Service.jsp">Service</a>
              </li>
              <li>
                <a href="User.jsp">User</a>
              </li>
            </ul>
          </nav>
        </div>
      </div>
    </div>
  </div>
</div>
<!-- end header inner -->
```

```
    </header>

    <!-- end header -->

    <!-- revolution slider -->

    <div class="banner-slider">

      <div class="container-fluid">

        <div class="row">

          <div class="col-md-7">

            <div id="slider_main" class="carousel slide" data-ride="carousel">

                <center>

          <h1 style="color:whitesmoke">Admin Login</h1>

       <form action="AdminLAction.jsp"  method="post">

      <table>

         <tr><th>UserName</th><td><input            type="text"             name="uname"
 required=""></tD></tr>

         <tr><th>Password</th><td><input            type="password"             name="pass"
 required=""></tD></tr>

         <tr><th></th><td><input type="submit" value="Login">

            <input type="Reset" value="Reset">

            </tD></tr>

      </table>

    </form>

</center

            </div>

          </div>

          <div class="col-md-5">

            <div class="full slider_cont_section">
          <h4>Vehicle</h4>
          <h3>Maintenance</h3>
          <p>
            Inspecting vehicle engine and mechanical/electrical components to diagnose issues accurately
Inspecting vehicle computer and electronic systems to repair, maintain and upgrade
Conducting routine maintenance work aiming to vehicle functionality and longevity
          </p>
          <div class="button_section">
            <a href="about.html">Read More</a>
            <a href="contact.html">Contact Us</a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<!-- end revolution slider -->
<!-- end footer -->
<!-- Javascript files-->
<script src="js/jquery.min.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.bundle.min.js"></script>
<script src="js/jquery-3.0.0.min.js"></script>
<script src="js/plugin.js"></script>
<!-- Scrollbar Js Files -->
```

```
    <script src="js/jquery.mCustomScrollbar.concat.min.js"></script>

    <script src="js/custom.js"></script>

  </body>

</html>
```

2. SERVICE.jsp

```
<%-->

   Created on : 13 Apr, 2021, 12:08:42 PM
   Author     : KishanVenky

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html lang="en">

    <!-- basic -->

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <!-- mobile metas -->

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta name="viewport" content="initial-scale=1, maximum-scale=1">

    <!-- site metas -->

    <title>Vehicle Maintenance</title>

    <meta name="keywords" content="">

    <meta name="description" content="">

    <meta name="author" content="">

    <link rel="stylesheet" href="css/bootstrap.min.css">

    <!-- style css -->
   <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="table.css">
   <!-- Responsive-->
   <link rel="stylesheet" href="css/responsive.css">
   <!-- fevicon -->
  <!-- Scrollbar Custom CSS -->
   <link rel="stylesheet" href="css/jquery.mCustomScrollbar.min.css">
</head>
<!-- body -->
<body class="main-layout">
   <!-- loader  -->
   <div class="loader_bg">
   </div>
   <!-- end loader -->
   <!-- header -->
   <header>
      <!-- header inner -->
      <div class="container-fluid">
         <div class="row">
            <div class="col-lg-3 logo_section">
               <div class="full">
                  <div class="center-desk">
                      <div class="logo"> <h1>Vehicle Maintenance</h1></div>
                  </div>
               </div>
            </div>
```

```
          <div class="col-lg-9">
            <div class="menu-area">
              <div class="limit-box">
                <nav class="main-menu">
                  <ul class="menu-area-main">
                    <li>
                      <a href="index.html">Home</a>
                    </li>
                    <li>
                      <a href="Admin.jsp">Admin</a>
                    </li>
                     <li  class="active">
                      <a href="Service.jsp">Service</a>
                    </li>
                    <li>
                      <a href="User.jsp">User</a>
                    </li>
                </nav>
              </div>
            </div>
          </div>
        </div>
      </div>
      <!-- end header inner -->
 </header><!--
end header -->
```

```
<!-- revolution slider -->
       <div class="banner-slider">
          <div class="container-fluid">
             <div class="row">
                <div class="col-md-7">
                   <div id="slider_main" class="carousel slide" data-ride="carousel">
                        <center>
                <h1 style="color:whitesmoke">Service Login</h1>
            <form action="ServiceLAction.jsp" method="post">
           <table>
               <tr><th>Email</th><td><input type="email" name="email" required=""></tD></tr>
     <tr><th></th><td><input type="submit" value="Login">
                   <input type="Reset" value="Reset">
           </table>
       </form>
         </center>
                   div>
                </div>
                <div class="col-md-5">
                   <div class="full slider_cont_section">
                      <h4>Vehicle</h4>
                      <h3>Maintenance</h3>
                      <p>
```

Inspecting vehicle engine and mechanical/electrical components to diagnose issues accurately Inspecting vehicle computer and electronic systems to repair, maintain and upgradeConducting routine maintenance work aiming to vehicle functionality and longevity

```
                    </p>
                    <div class="button_section">
                      <a href="about.html">Read More</a>
                      <a href="contact.html">Contact Us</a>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
          <!-- end revolution slider -->
          <!-- end footer -->
          <!-- Javascript files-->
          <script src="js/jquery.min.js"></script>
          <script src="js/popper.min.js"></script>
          <script src="js/bootstrap.bundle.min.js"></script>
          <script src="js/jquery-3.0.0.min.js"></script>
          <script src="js/plugin.js"></script>
          <!-- Scrollbar Js Files -->
          <script src="js/jquery.mCustomScrollbar.concat.min.js"></script>
          <script src="js/custom.js"></script>
        </body>
    </html>
```

3. USER.jsp

```
<%--
   Document : Admin
   Created on : 13 Apr, 2021, 12:08:42 PM
   Author      : KishanVenky
 --%>
 <%@page contentType="text/html" pageEncoding="UTF-8"%>
 <!DOCTYPE html>
 <html lang="en">
   <head>
     <!-- basic -->
     <meta charset="utf-8">
     <meta http-equiv="X-UA-Compatible" content="IE=edge">
     <!-- mobile metas -->
     <meta name="viewport" content="width=device-width, initial-scale=1">
     <meta name="viewport" content="initial-scale=1, maximum-scale=1">
     <!-- site metas -->
     <title>Vehicle Maintenance</title>
     <meta name="keywords" content="">
     <meta name="description" content="">
     <meta name="author" content="">
     <!-- bootstrap css -->
     <link rel="stylesheet" href="css/bootstrap.min.css">
     <!-- style css -->
     <link rel="stylesheet" href="css/style.css">
      <link rel="stylesheet" href="table.css">
     <!-- Responsive-->
```

```
  <link rel="stylesheet" href="css/responsive.css">
  <!-- fevicon -->
 <!-- Scrollbar Custom CSS -->
  <link rel="stylesheet" href="css/jquery.mCustomScrollbar.min.css">
</head>
<!-- body -->
<body class="main-layout">
  <!-- loader -->
  <div class="loader_bg">
  </div>
  <!-- end loader -->
  <!-- header -->
  <header>
    <!-- header inner -->
    <div class="container-fluid">
      <div class="row">
        <div class="col-lg-3 logo_section">
          <div class="full">
            <div class="center-desk">
              <div class="logo"> <h1>Vehicle Maintenance</h1></div>
            </div>
          </div>
        </div>
        <div class="col-lg-9">
             <div class="menu-area">
               <div class="limit-box">
              <nav class="main-menu">
```

```
                    <ul class="menu-area-main">
                      <li>
                        <a href="index.html">Home</a>
                      </li>
                      <li>
                        <a href="Admin.jsp">Admin</a>
                      </li>
                      <li>
                        <a href="Service.jsp">Service</a>
                      </li>
                      <li class="active">
                        <a href="User.jsp">User</a>
                      </li>
                     </ul>
                   </nav>
                </div>
              </div>
            </div>
          </div>
        </div>
        <!-- end header inner -->
      </header>
      <!-- end header -->
      <!-- revolution slider -->
      <div class="banner-slider">
        <div class="container-fluid">
          <div class="row">
```

```
           <div class="col-md-7">

             <div id="slider_main" class="carousel slide" data-ride="carousel">

                 <center>

           <h1 style="color:whitesmoke">User Login</h1>

         <form action="UserLAction.jsp" method="post">

        <table>

          <tr><th>UserName</th><td><input type="text" name="uname"
 required=""></tD></tr>

          <tr><th>Password</th><td><input type="password" name="pass"
 required=""></tD></tr>

          <tr><th></th><td><input type="submit" value="Login">

             <input type="Reset" value="Reset">

             <a href="Register.jsp">Register</a>

             </tD></tr>

        </table>

     </form>

       </center>

             </div>

           </div>

           <div class="col-md-5">

             <div class="full slider_cont_section">

               <h4>Vehicle</h4>

               <h3>Maintenance</h3>

               <p>

Inspecting vehicle engine and mechanical/electrical components to diagnoseissues accurately Inspecting
vehicle computer and electronic systems to repair, maintain and upgrade Conducting routine maintenance
work aiming to vehicle functionality and longevity
```

```
              </p>
              <div class="button_section">
                <a href="about.html">Read More</a>
                <a href="contact.html">Contact Us</a>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <!-- end revolution slider -->
    <!-- end footer -->
    <!-- Javascript files-->
    <script src="js/jquery.min.js"></script>
    <script src="js/popper.min.js"></script>
    <script src="js/bootstrap.bundle.min.js"></script>
    <script src="js/jquery-3.0.0.min.js"></script>
    <script src="js/plugin.js"></script>
    <!-- Scrollbar Js Files -->
    <script src="js/jquery.mCustomScrollbar.concat.min.js"></script>
    <script src="js/custom.js"></script>
  </body>
</html>
```

CHAPTER-6

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test Cases :

Admin Module:

Test Case 1: Login Functionality

Description: Verify that the admin can log in successfully with valid credentials.

Preconditions: Admin credentials are available.

Test Steps:

Enter valid admin username and password.

Click the "Login" button.

Expected Result: Admin should be logged in and directed to the admin dashboard.

Test Case 2: Add Service

Description: Ensure that the admin can add a new service with valid details.

Preconditions: Admin is logged in.

Test Steps:

Navigate to the "Add Service" section.

Enter valid service details (name, price, etc.).

Click the "Add Service" button.

Expected Result: The new service should be added successfully to the system.

Test Case 3: Delete Service

Description: Validate the admin's ability to delete a service.

Preconditions: Admin is logged in, and at least one service exists.

Test Steps:

Select a service from the list of existing services.

Click the "Delete" button.

Expected Result: The selected service should be deleted from the system.

Test Case 4: View Users

Description: Confirm that the admin can view the list of users.

Preconditions: Admin is logged in.

Test Steps:

Navigate to the "View Users" section.

Expected Result: The admin should see a list of registered users.

Test Case 5: View Services

Description: Ensure that the admin can view the list of services.

Preconditions: Admin is logged in, and at least one service exists.

Test Steps:

Navigate to the "View Services" section.

Expected Result: The admin should see a list of available services.

Service Module:

Test Case 6: Add Car Service Details

Description: Verify that service providers can add car service details.

Preconditions: Service provider is logged in.

Test Steps:

Navigate to the "Add Car Service Details" section.

Enter valid car service details.

Submit the form.

Expected Result: The car service details should be successfully added to the system.

Test Case 7: View Car Service Details

Description: Ensure that users can view existing car service details.

Preconditions: At least one car service detail exists.

Test Steps:

Navigate to the "View Car Service Details" section.

Expected Result: Users should be able to see a list of available car service details.

User Module:

Test Case 8: View Services

Description: Confirm that users can view the list of services.

Preconditions: At least one service is available.

Test Steps:

Navigate to the "View Services" section.

Expected Result: Users should see a list of available services.

Test Case 9: View Car Details

Description: Validate that users can view car details.

Preconditions: At least one car detail exists.

Test Steps:

Navigate to the "View Car Details" section.

Expected Result: Users should be able to see the details of available cars.

Test Case 10: User Authentication

Description: Test the authentication process for users.

Preconditions: User credentials are available.

Test Steps:

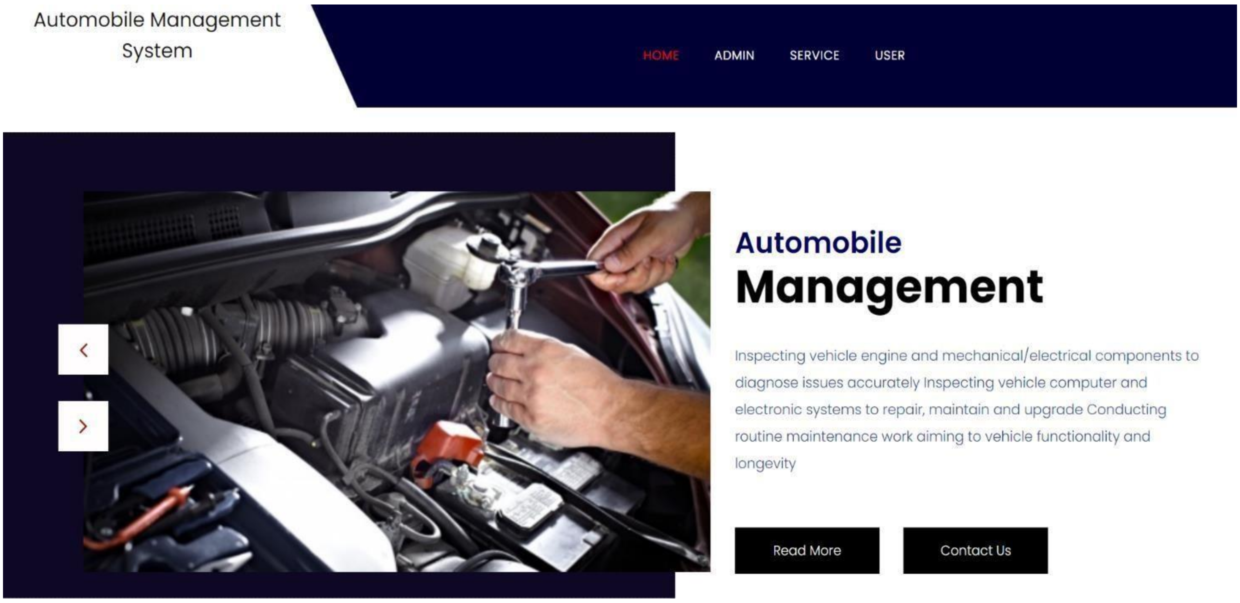
Enter valid user credentials.

Click the "Login" button.

Expected Result: Users should be logged in successfully and directed to the appropriate page.

CHAPTER-7

RESULTS

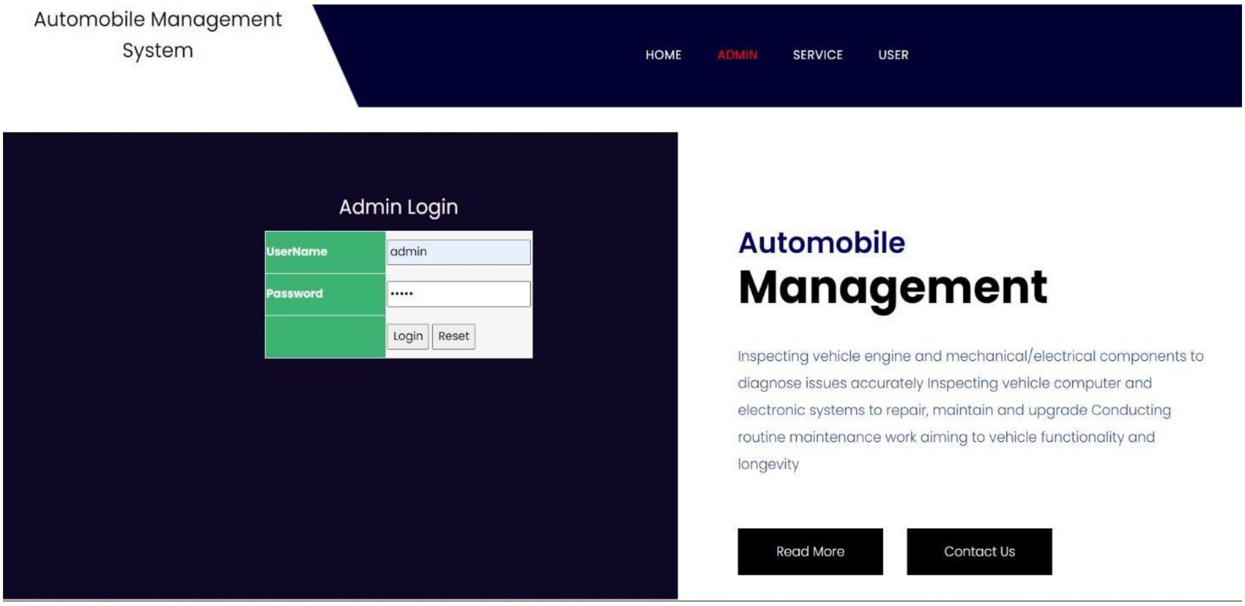


Automobile

Management

Inspecting vehicle engine and mechanical/electrical components to diagnose issues accurately Inspecting vehicle computer and electronic systems to repair, maintain and upgrade Conducting routine maintenance work aiming to vehicle functionality and longevity

Fig 7.1 Home screen



Admin Login

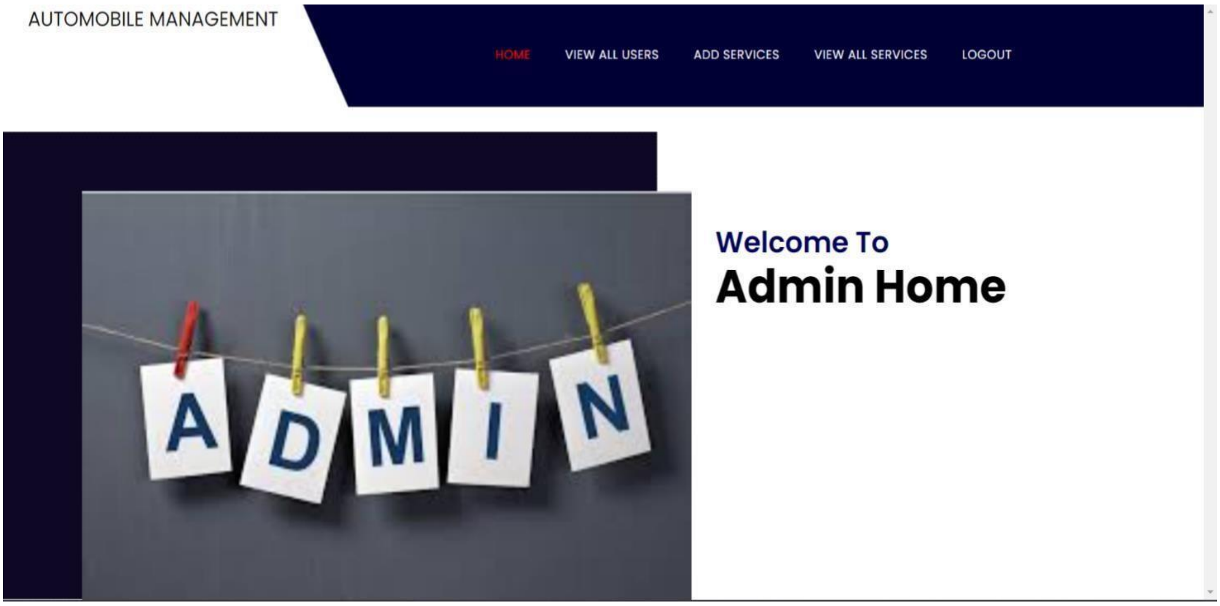
UserName	admin
Password	•••••

Automobile

Management

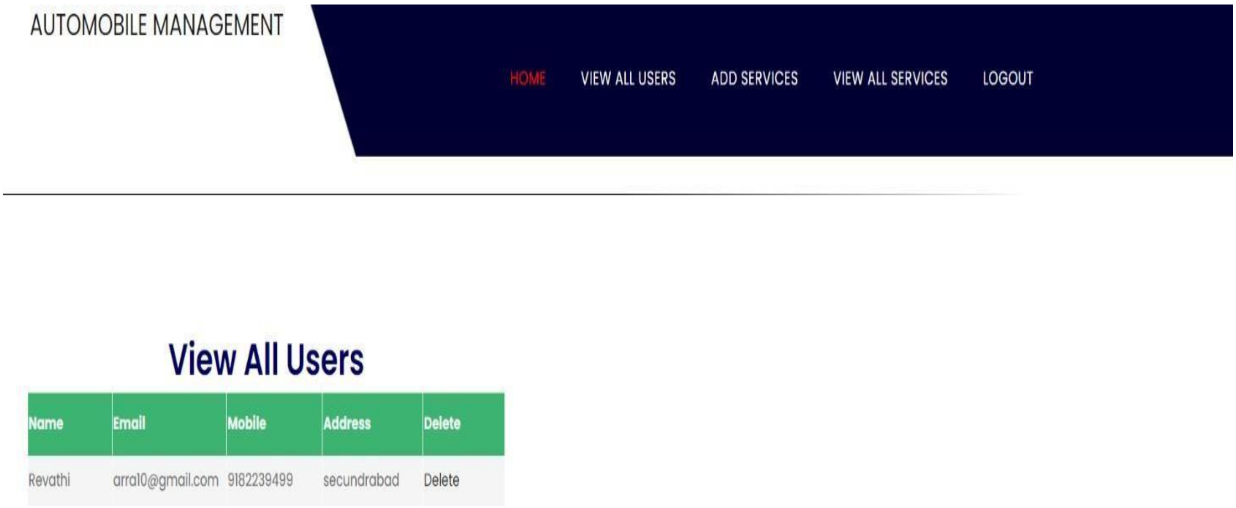
Inspecting vehicle engine and mechanical/electrical components to diagnose issues accurately Inspecting vehicle computer and electronic systems to repair, maintain and upgrade Conducting routine maintenance work aiming to vehicle functionality and longevity

Fig 7.2 Admin Login Page



Welcome To
Admin Home

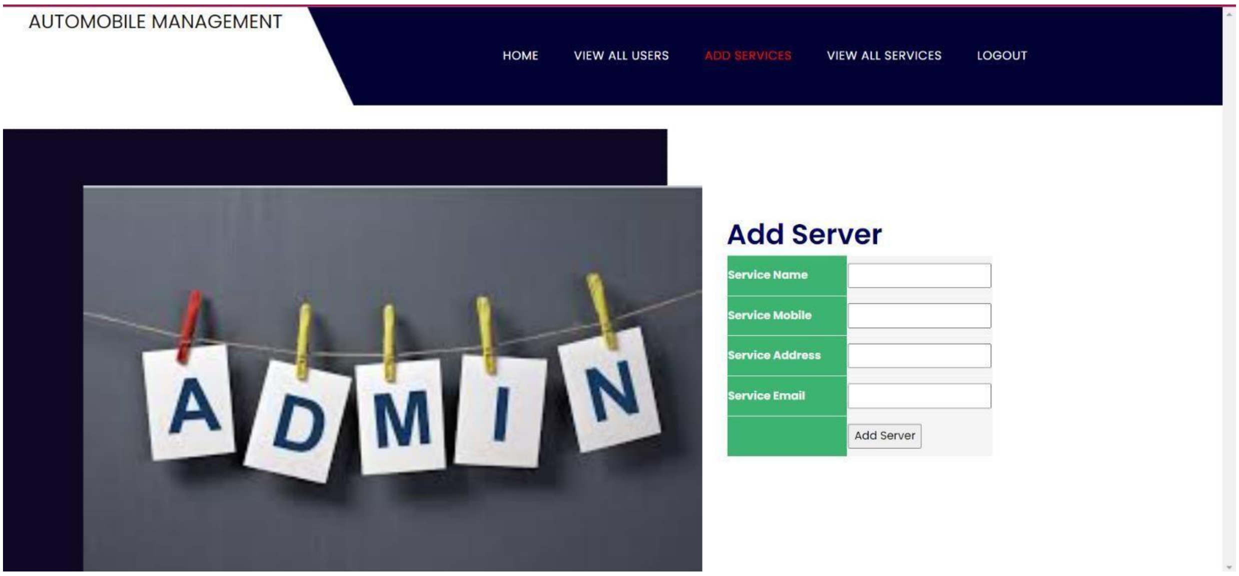
Fig 7.3 Admin Home Page



View All Users

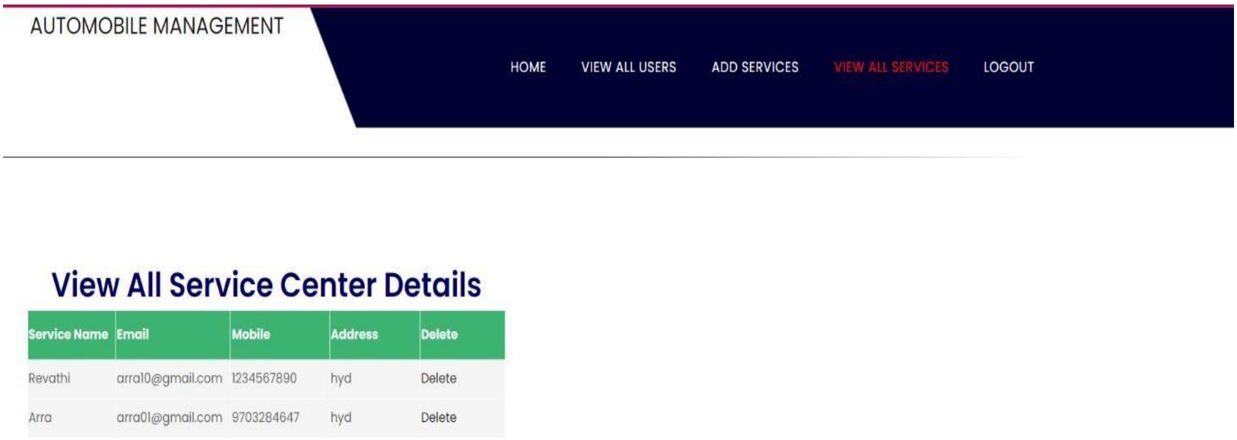
Name	Email	Mobile	Address	Delete
Revathi	arra10@gmail.com	9182239499	secundrabad	Delete

Fig 7.4 Admin Operation-View All Users



Service Name	
Service Mobile	
Service Address	
Service Email	
	Add Server

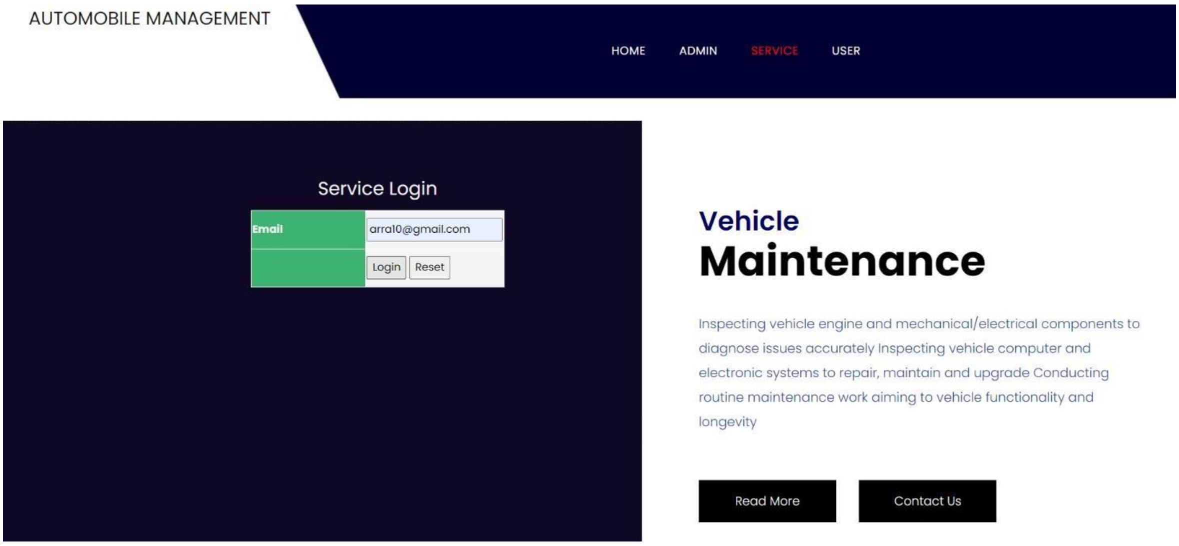
Fig 7.5 Admin Operations—Add Server



View All Service Center Details

Service Name	Email	Mobile	Address	Delete
Revathi	arra10@gmail.com	1234567890	hyd	Delete
Arra	arra01@gmail.com	9703284647	hyd	Delete

Fig 7.6 Admin Operation-View all Services



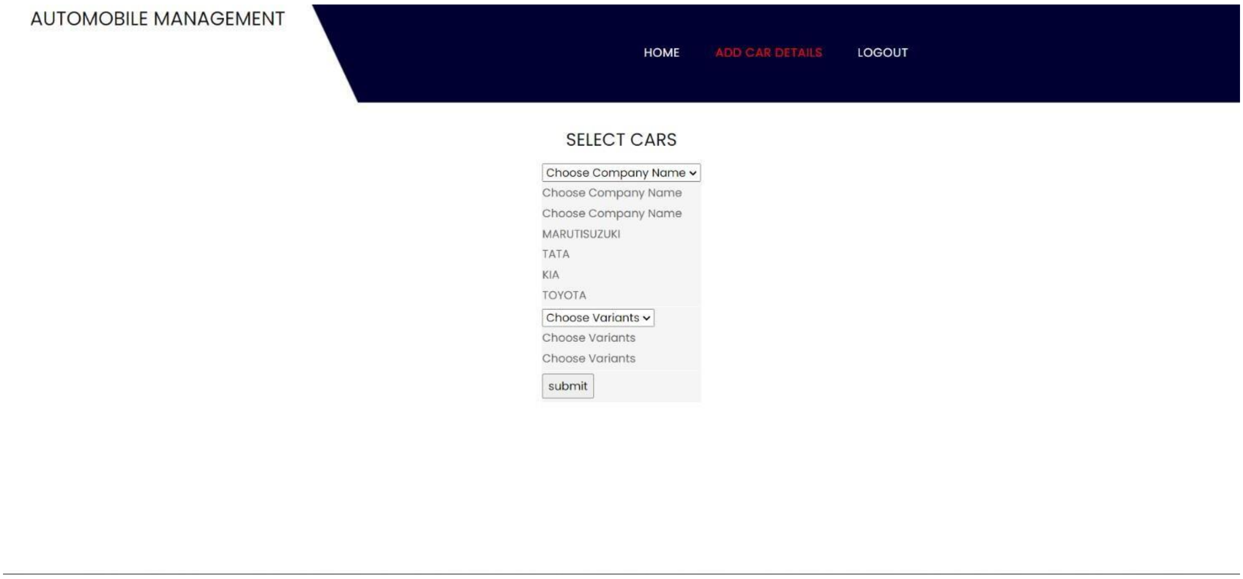
Service Login

Email: arra10@gmail.com

Vehicle
Maintenance

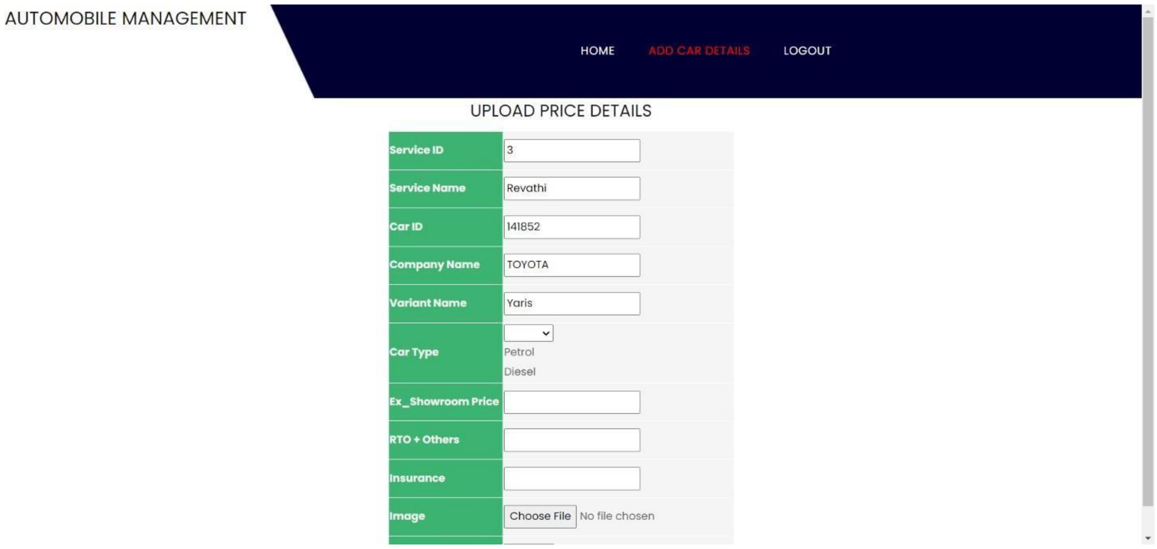
Inspecting vehicle engine and mechanical/electrical components to diagnose issues accurately Inspecting vehicle computer and electronic systems to repair, maintain and upgrade Conducting routine maintenance work aiming to vehicle functionality and longevity

Fig 7.7 Service Login Page



SELECT CARS

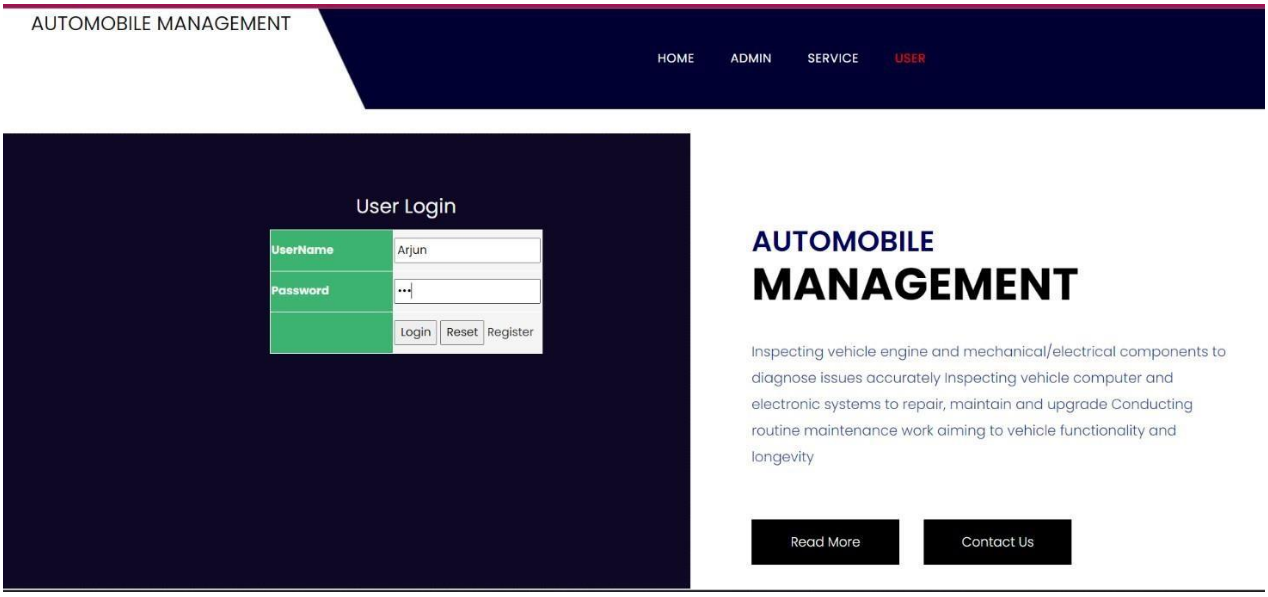
Fig 7.8 Service Operations-Select Car’s



UPLOAD PRICE DETAILS

Service ID	3
Service Name	Revathi
Car ID	141852
Company Name	TOYOTA
Variant Name	Yaris
Car Type	Petrol Diesel
Ex_Showroom Price	
RTO + Others	
Insurance	
Image	Choose File No file chosen

Fig 7.9 Service Operations-Upload Car Details



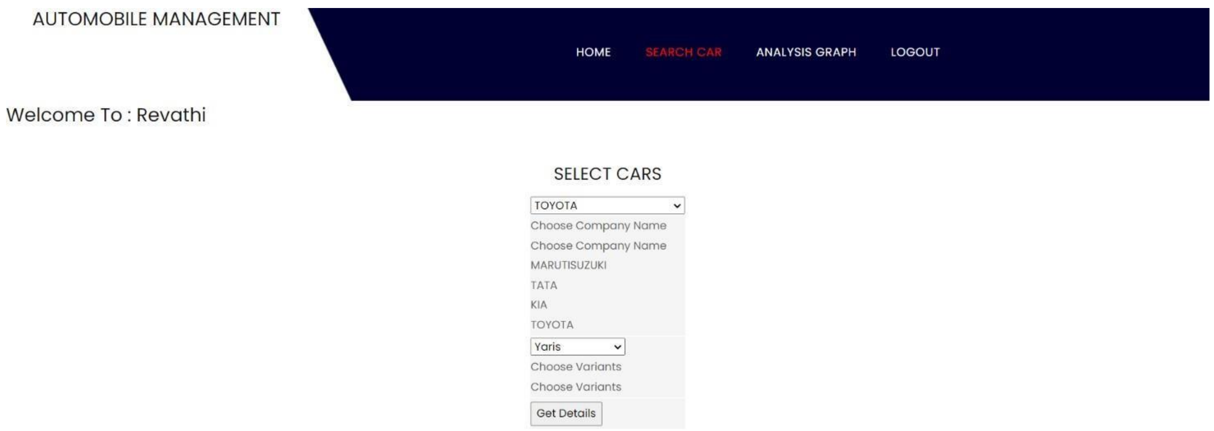
User Login

UserName	Arjun
Password	•••
	Login Reset Register

AUTOMOBILE
MANAGEMENT

Inspecting vehicle engine and mechanical/electrical components to diagnose issues accurately Inspecting vehicle computer and electronic systems to repair, maintain and upgrade Conducting routine maintenance work aiming to vehicle functionality and longevity

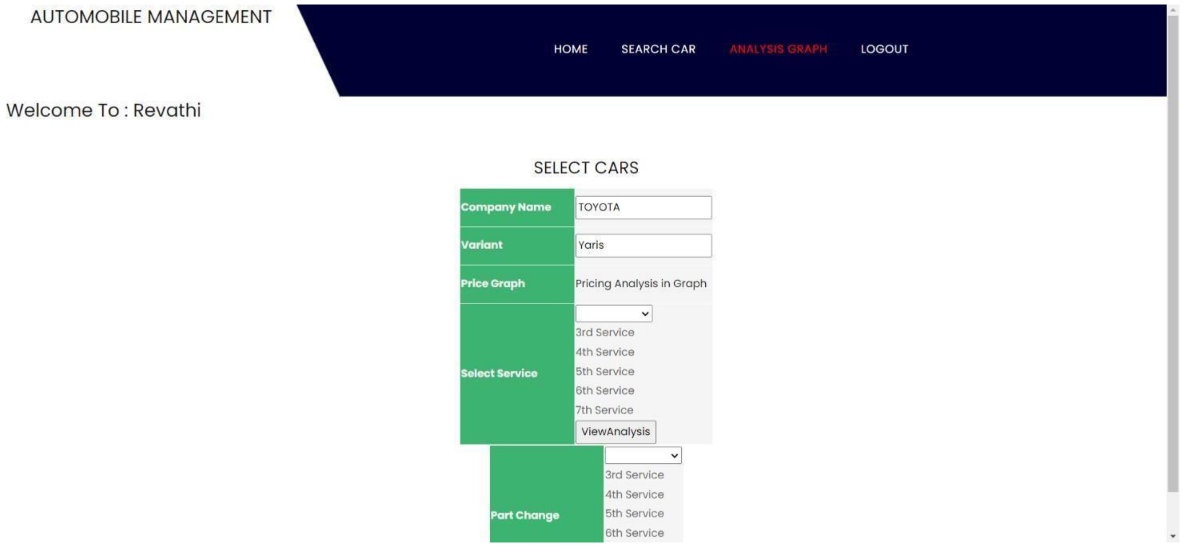
Fig 7.10 User Login Page



TOYOTA

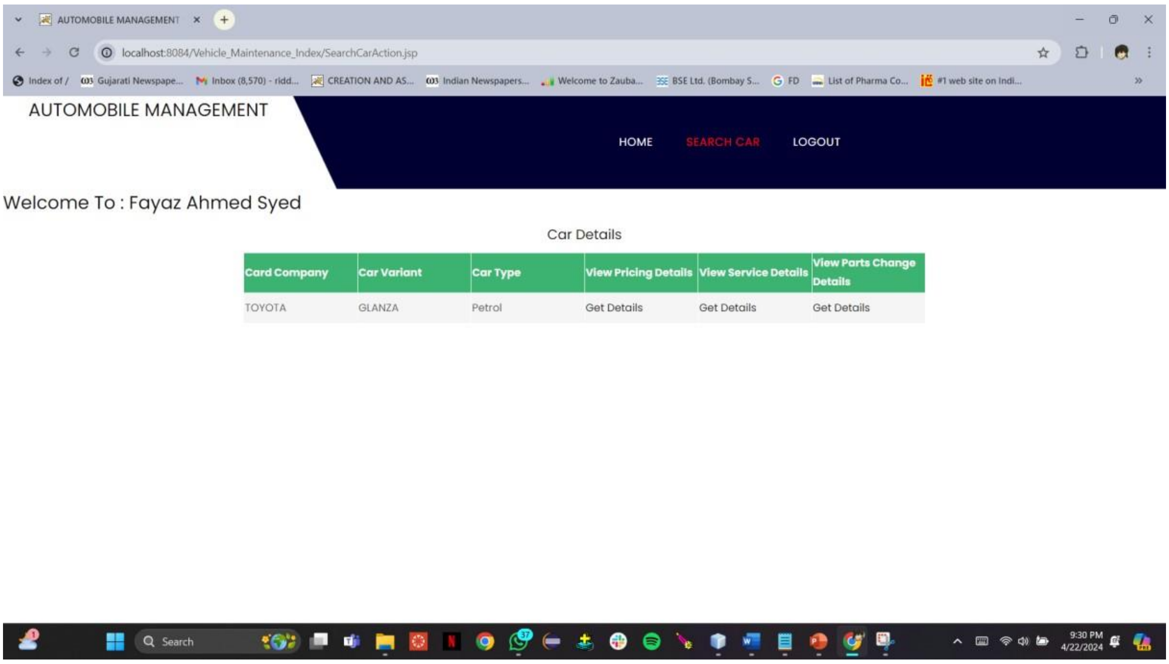
Yaris

Fig 7.11 User Operations-Select Car’s



Company Name	TOYOTA
Variant	Yaris
Price Graph	Pricing Analysis in Graph
Select Service	
Part Change	

Fig 7.12 User Operations-Select Service


AUTOMOBILE MANAGEMENT
localhost:8084/Vehicle_Maintenance_Index/SearchCarAction.jsp
Index of / | Gujarati Newspape... | Inbox (8,570) - ridd... | CREATION AND AS... | Indian Newspapers... | Welcome to Zauba... | BSE Ltd. (Bombay S... | FD | List of Pharma Co... | #1 web site on Indi...
HOME | SEARCH CAR | LOGOUT
Search
9:30 PM
4/22/2024

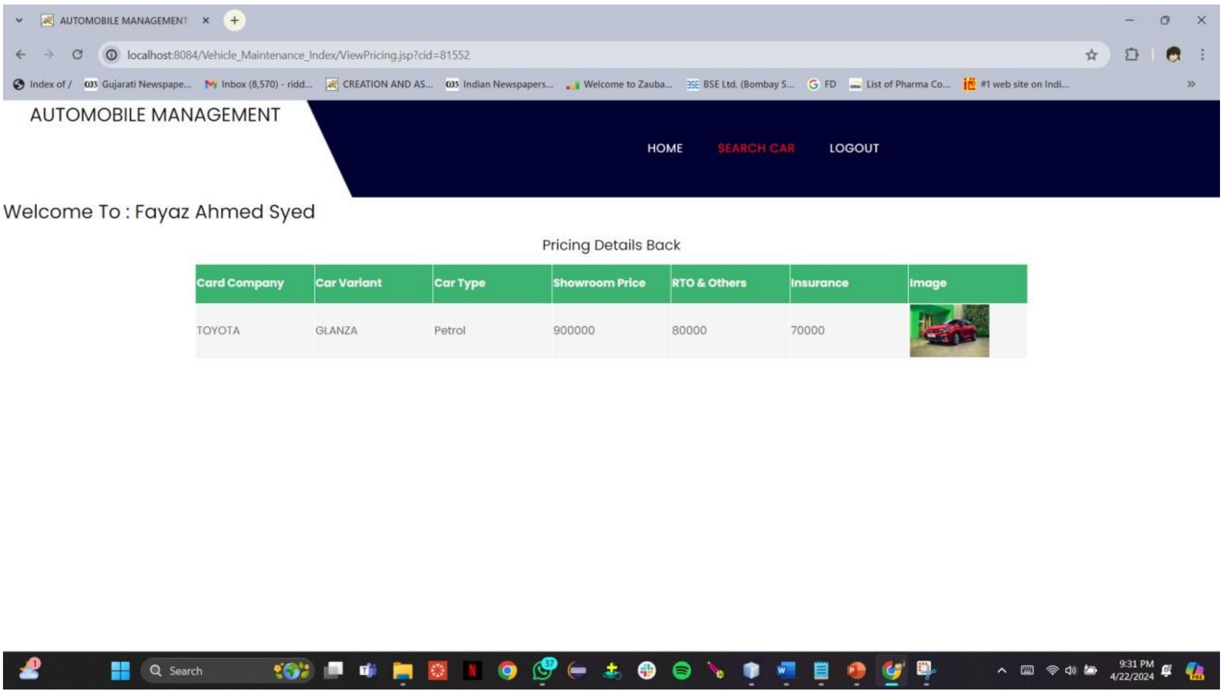
AUTOMOBILE MANAGEMENT

Welcome To : Fayaz Ahmed Syed

Car Details

Card Company	Car Variant	Car Type	View Pricing Details	View Service Details	View Parts Change Details
TOYOTA	GLANZA	Petrol	Get Details	Get Details	Get Details

Fig 7.13 View Car Details


AUTOMOBILE MANAGEMENT
localhost:8084/Vehicle_Maintenance_Index/ViewPricing.jsp?cid=81552
Index of / | Gujarati Newspape... | Inbox (8,570) - ridd... | CREATION AND AS... | Indian Newspapers... | Welcome to Zauba... | BSE Ltd. (Bombay S... | FD | List of Pharma Co... | #1 web site on Indi...
HOME | SEARCH CAR | LOGOUT
Search
9:31 PM
4/22/2024

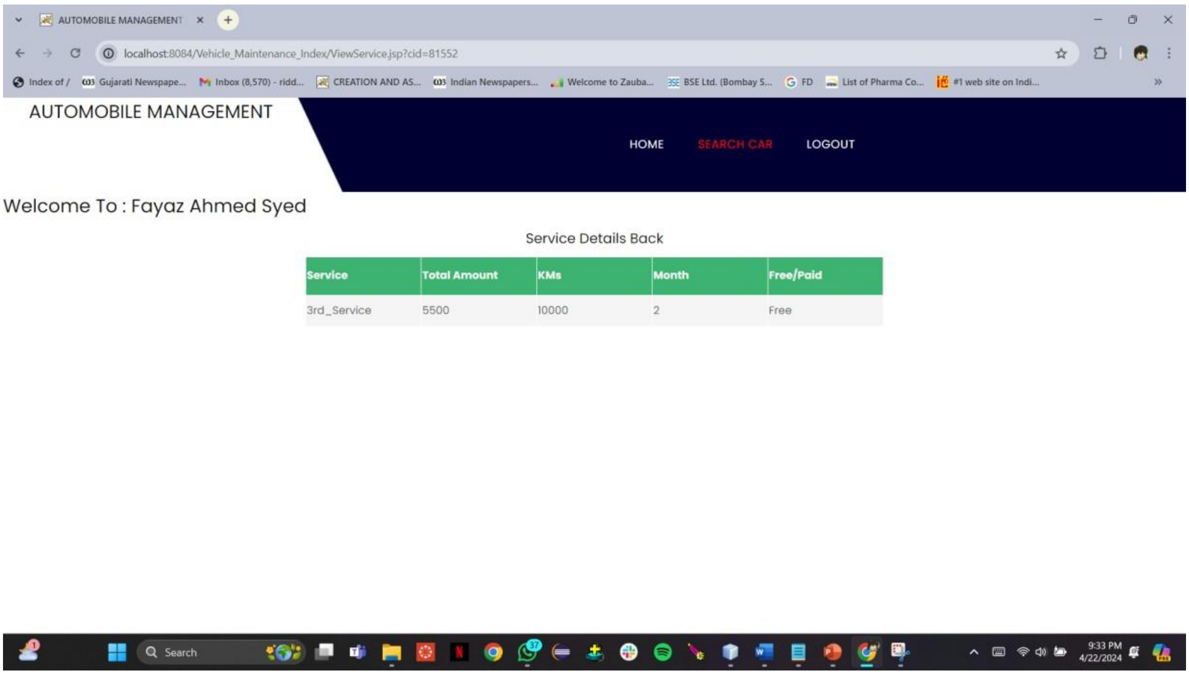
AUTOMOBILE MANAGEMENT

Welcome To : Fayaz Ahmed Syed

Pricing Details Back

Card Company	Car Variant	Car Type	Showroom Price	RTO & Others	Insurance	Image
TOYOTA	GLANZA	Petrol	900000	80000	70000	

Fig 7.14 Price Details

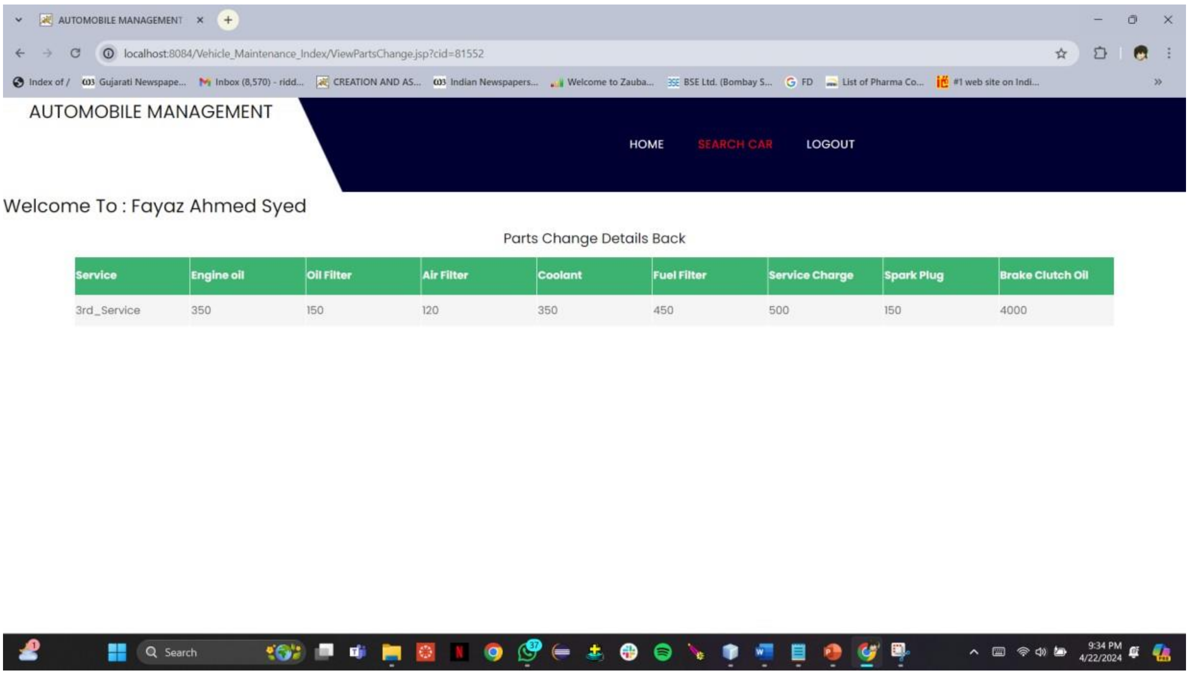


Welcome To : Fayaz Ahmed Syed

Service Details Back

Service	Total Amount	KMs	Month	Free/Paid
3rd_Service	5500	10000	2	Free

Fig 7.15 Service Details

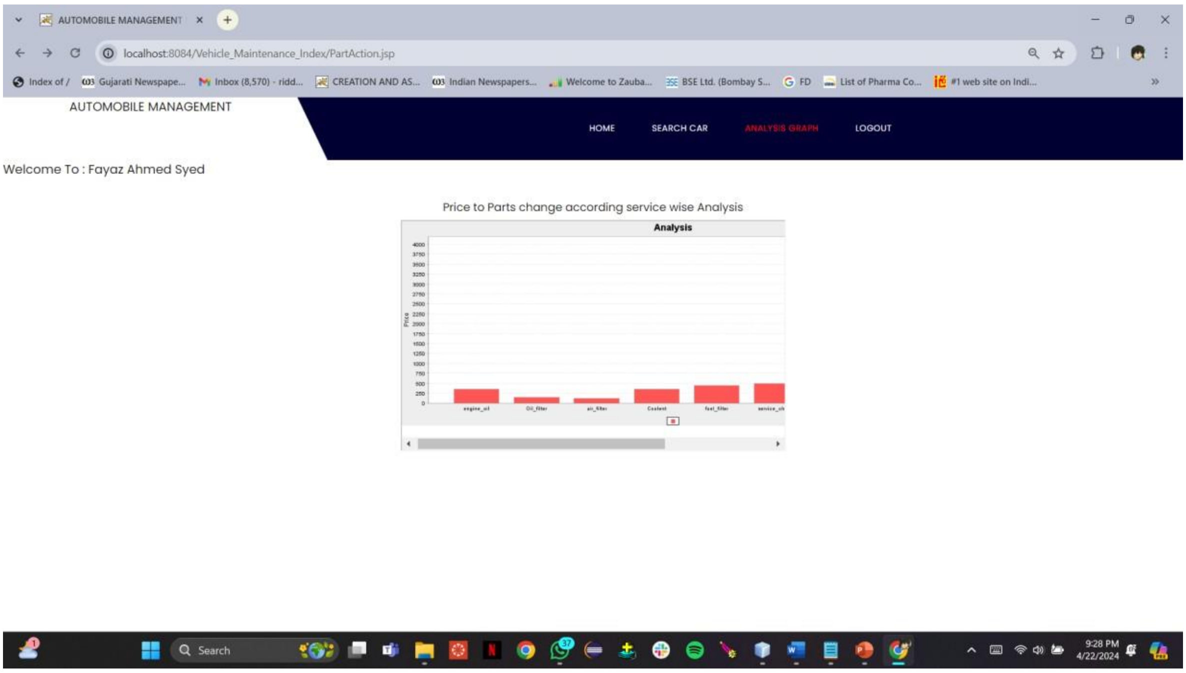


Welcome To : Fayaz Ahmed Syed

Parts Change Details Back

Service	Engine oil	Oil Filter	Air Filter	Coolant	Fuel Filter	Service Charge	Spark Plug	Brake Clutch Oil
3rd_Service	350	150	120	350	450	500	150	4000

Fig 7.16 Parts Change Details



Welcome To : Fayaz Ahmed Syed

Price to Parts change according service wise Analysis

Fig 7.17 Analysis Bar Chart

CHAPTER-8

CONCLUSION

8.1 CONCLUSION

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

- Automation of the entire system improves the efficiency
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

8.2 FUTURE SCOPE

This application avoids the manual work and the problems concern with it. It is an easy way to obtain the information regarding the various products information that are present in the Super markets. Well, I and my team members have worked hard in order to present an improved website better than the existing one’s regarding the information about the various activities. Still, we found out that the project can be done in a better way.

Primarily, when we request information about a particular product it just shows the company, product id, product name and no. of quantities available. So, after getting the information we can get access to the product company website just by a click on the product name. The next enhancement that we can add the searching option. We can directly search to the particular product company from this site. These are the twoenhancements that we could think of at present.

CHAPTER-9

Bibliography

1. Akinyemi Olasunkanmi O.*, Koyejo Oyebola T. (2011). Journal of Engineering Scienceand Technology. Priority Dispatch Scheduling in an Automobile Repair and Maintenance Workshop.
2. N. Shivasankaran and P. Senthilkumar (2014). Indian Journal of Computer Science and Engineering (IJCSE). Scheduling of Mechanics in Automobile Repair Shops UsingAnn.
3. Muhammad Kamal Arif Bin Razali (2017). Thesis Final Year Project University Sultan Zainal Abidin. Online Medical Appointment for Pusat Kesihatan Unisza Using MobileApplication.

The following books were referred during the analysis and execution phase of the project MICROSOFT .NET WITH C# Microsoft .net series

1. ASP .NET 2.0 PROFESSIONAL Wrox Publishers
2. ASP .NET WITH C# 2005 Apress Publications
3. C# COOK BOOK O reilly Publications
4. PROGRAMMING MICROSOFT ASP .NET 2.0 APPLICATION Wrox Professional Guide
5. BEGINNING ASP .NET 2.0 E-COMMERCE IN C# 2005 Novice to Professional.

WEBSITES: http://localhost:8084/Vehicle_Maintenance_Index/