

**Web Based Fleet Management System  
for  
Macksons Paint Lanka (Pvt) Ltd**

**M.A.M. ANHAR**

**November 2017**



# **Web Based Fleet Management System for Macksons Paint Lanka (Pvt) Ltd**

M. A. M. Anhar

BIT Registration Number R141611

BIT Index Number 1416111

Supervisor:

Mr. Ragukumar S Thanganathan

**November 2017**



**This dissertation is submitted in partial fulfillment of the requirement of the  
Degree of Bachelor of Information Technology (External) of the  
University Of Colombo School Of Computing**

---

# Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate:  .....

Date: 08/11/2017

Name of Candidate: M.A.M. Anhar .....

Countersigned by:

Signature of Supervisor(s)/Advisor(s):  .....

Date: 08/11/2017

Name(s) of Supervisor(s)/Advisor(s): Ragu.kumar .....

---

# Abstract

Macksons Holdings is a diversified conglomerate involved in manufacturing, retailing, property development, franchising, trading, imports and exports in household, leisure, automobile, building materials. The company handles with heavy paper work currently and which will be inefficient and time wasting. As a solution we can get an aid from the information technology. Fleet management system for delivery vehicles is emerged to reduce these problems and will increase the competency and efficiency of their work.

This system has provided many facilities such as vehicle registration, driver registration, handling vehicle request and allocation, managing delivery expenses, manage vehicle maintenance expenses, handling daily driver attendance, calculating driver Bata and incentives, vehicle maintenance information, incident tracking & generate reports to the right time etc.

The fleet management system has been developed based on the Rational Unified Process (RUP) framework. The architecture used for the system is MVC architecture, which stands for 'Model, View and Controller'. Object oriented techniques used to design and implement the system. the PHP server side scripting language has been selected as the programming language to implement this system and Netbeans has been selected as the IDE (integrated development environment). Further MYSQL server database has been selected as the database to store the data of the system.

This system has been successfully developed according to client's requirement therefore all the errors and barriers of existing system were eliminated by new system.

---

# ACKNOWLEDGEMENT

First of all I would like to express my gratitude thankful to the BIT Coordinators and academic staff of University of Colombo School of Computing (UCSC) for giving us valuable degree program and providing a good guidance continuously.

My special thanks must go to my project supervisor and lecturer Mr. Ragukumar Steven Thanganathan for the proper guidance and the excellent support which I received throughout the project.

I would like to thanks my client Mr. Salam Riyad (Group IT Manager), Mr. Chanaka (Transport Manager) and his staff of Macksons Paint Lanka (Pvt) Ltd. for giving great support and dedicated their valuable time to complete my project.

---

# Table of Contents

<b>Declaration</b> .....	<b>ii</b>
<b>Abstract</b> .....	<b>iii</b>
<b>Acknowledgement</b> .....	<b>iv</b>
<b>Table of Contents</b> .....	<b>v</b>
<b>List of Figures</b> .....	<b>viii</b>
<b>List of Table</b> .....	<b>x</b>
<b>List of Acronyms</b> .....	<b>xi</b>
<b>CHAPTER 01 – INTRODUCTION</b> .....	<b>1</b>
1.1 Introduction.....	1
1.2 Motivation of the Project.....	1
1.3 Scope of the Proposed Project.....	2
1.4 Objectives of the System.....	2
1.5 Structure of the Dissertation.....	3
<b>CHAPTER 02 – ANALYSIS</b> .....	<b>4</b>
2.1 Introduction.....	4
2.2 Existing System.....	4
2.2.1 Use case diagram for Existing System.....	4
2.2.2 Drawbacks of the Existing System.....	5
2.3 Existing Similar System.....	5
2.3.1 Vehicle Fleet Manager.....	5
2.3.2 Fleetio.....	6
2.3.3 Limitation of above off-the-shelf software.....	6
2.4 Identified System Users.....	6
2.5 Functional Requirement.....	7
2.6 Non-functional Requirement.....	9
2.7 Methodology for the Proposed System.....	10

---

<b>CHAPTER 03 – DESIGN</b> .....	<b>12</b>
3.1 Introduction.....	12
3.2 Alternative Solution.....	12
3.2.1 Use off-the-shelf software.....	12
3.2.2 Develop a Standalone System.....	12
3.3 Selected Solution.....	13
3.4 Design Architecture.....	13
3.5 Design Diagram.....	15
3.5.1 High Level Use Case Diagram for Proposed System.....	15
3.5.2 Class Diagram for Proposed System.....	16
3.5.3 Activity Diagram for Vehicle Allocation.....	17
3.5.4 Sequence Diagram for Vehicle Allocation.....	18
3.5.5 Database Design.....	19
3.6 Interface Design.....	20
3.6.1 Login Interface.....	20
3.6.2 Admin Home Page.....	21
3.6.3 Vehicle Registration Interface.....	21
3.6.4 Display Information.....	22
 <b>CHAPTER 04 – IMPLEMENTATION</b> .....	 <b>24</b>
4.1 Introduction.....	24
4.2 Implementation Environment.....	24
4.3 Major Code Segments.....	25
4.4 Re-usable Components.....	33
 <b>CHAPTER 05 – EVALUATION</b> .....	 <b>34</b>
5.1 Introduction.....	34
5.2 System Test.....	34
5.3 Test Plan and System Test case.....	34
5.3.1 Test cases for User Authentication .....	34
5.3.2 Test cases for Manage user.....	35
5.3.3 Test cases for driver registration.....	36

---

5.3.4	Test cases for Vehicle request.....	38
5.3.5	Test cases for Allocation.....	39
5.4	Test Data and Test Results.....	40
5.5	Use Acceptance Testing .....	42
<b>CHAPTER 06 – CONCLUSION.....</b>		<b>44</b>
6.1	Introduction.....	44
6.2	Future Enhancement.....	44
6.3	Lesson learnt.....	45
<b>References.....</b>		<b>46</b>
<b>Appendix A - System Documentation.....</b>		<b>47</b>
<b>Appendix B - Design Documentation.....</b>		<b>59</b>
<b>Appendix C - User Documentation.....</b>		<b>56</b>
<b>Appendix D - Management Reports.....</b>		<b>72</b>
<b>Appendix E - Test Results.....</b>		<b>75</b>
<b>Appendix F - Code Listing.....</b>		<b>79</b>
<b>Appendix G - Client Certificate.....</b>		<b>85</b>
<b>Glossary.....</b>		<b>86</b>
<b>Index.....</b>		<b>87</b>



---

# List of Figures

Figure 2.1: Use Case Diagram for the Existing System.....	4
Figure 2.2: RUP Phases.....	11
Figure 3.1: MVC Architectur.....	13
Figure 3.2: High Level Use Case Diagram for Proposed System.....	15
Figure 3.3: Class Diagram for Proposed System.....	16
Figure 3.4: Activity Diagram for Vehicle Allocation .....	17
Figure 3.5: Sequence Diagram for Vehicle Allocation.....	18
Figure 3.6: ERD Diagram for Proposed System.....	19
Figure 3.7: Login Interface for Proposed System.....	20
Figure 3.8: Home Page for Administrator.....	21
Figure 3.9: Vehicle Request Form.....	22
Figure 3.10: Display Information for Proposed System.....	23
Figure 5.1: User Evaluation Questioner.....	43
Figure B.1: Use Case Diagram for Request Vehicle.....	49
Figure B.2: Use Case Diagram for Vehicle Allocation.....	50
Figure B.3: Use Case Diagram for Close Delivery.....	52
Figure B.4: Use Case Diagram for User Management.....	54
Figure C.1: System Login Page.....	56
Figure C.2: Administrator Home Page.....	57
Figure C.3: Page Navigation.....	57
Figure C.4: Sign-out Button.....	57
Figure C.5: Data Manipulation Table.....	58
Figure C.6: Vehicle Module Icon.....	59
Figure C.7: Vehicle Registration Form.....	59
Figure C.8: Vehicle Data Manipulation Table.....	60
Figure C.9: View Vehicle .....	60
Figure C.10: Update Vehicle information.....	61
Figure C.11: Change vehicle Status.....	61
Figure C.12: Driver Module Icon.....	62
Figure C.13: Driver Registration Form.....	62

---

Figure C.14: Request Vehicle Module Icon.....	63
Figure C.15: Vehicle Request Form.....	63
Figure C.16: Cancel Vehicle Request.....	64
Figure C.17: Confirmation Message of Cancel Vehicle Request.....	64
Figure C.18: Vehicle Allocation Icon.....	64
Figure C.19: Vehicle Allocation Form.....	65
Figure C.20: Close Delivery Module Icon.....	65
Figure C.21: Close Delivery Form.....	66
Figure C.22: Delivery expenses Icon.....	66
Figure C.23: Delivery Expenses Form.....	67
Figure C.24: Vehicle Maintenance Icon.....	67
Figure C.25: Add Vehicle Maintenance Form.....	68
Figure C.26: Close Maintenance Icon.....	68
Figure C.27: Maintenance Expenses Form.....	68
Figure C.28: Fuel Management Module Icon.....	69
Figure C.29: Fuel Management Dash Board.....	69
Figure C.30: Fuel Insertion Form.....	70
Figure C.31: Fuel Pump Form.....	70
Figure C.32: User Management Icon.....	70
Figure C.33: Add User Form.....	71
Figure D.1: Last 7 Days Delivery Expenses Summery Report.....	72
Figure D.2: Customize Delivery Expenses Summery Report.....	72
Figure D.3: Detail Delivery Expenses Report.....	73
Figure D.4: De Maintenance Expenses Report.....	74
Figure G.1: Client Certificate .....	85

---

# List of Tables

4.1: Implementation Environment.....	24
5.1: Test Cases for User Authentication.....	35
5.2: Test Cases for Manage User.....	35
5.3: Test Cases for Driver Registration.....	36
5.4: Test Cases for Request Vehicle.....	38
5.5: Test Cases for Vehicle Allocation.....	29
5.6: Test Cases for Close Delivery.....	39
5.7: Test Result for Add User.....	40
5.8: Test Result for Add Driver .....	41
A.1: Hardware Requirement.....	47
A.2: Software Requirement.....	47
B.1: Use Case Description for Vehicle Request.....	49
B.2: Use Case Description for Vehicle Allocation.....	51
B.3: Use Case Description for Close Delivery.....	52
B.4: Use Case Description for Insert Delivery Expenses.....	53
B.5: Use Case Description for Add User.....	55
E.1: Test Result for User Authentication.....	75
E.2: Test Result for Request Vehicle.....	76
E.3: Test Result for Vehicle Allocation.....	77
E.4: Test Result for Close Delivery.....	78

---

# LIST OF ACRONYMS

AJAX	–	Asynchronous Javascript and XML
CSS	–	Cascading Style Sheets
DB	–	Data Base
HDD	–	Hard Disk Driver
HTML	–	Hypertext Markup Language
HTTP	–	Hypertext Transfer Protocol
JS	–	Java Script
MVC	–	Model View Controller
RAM	–	Random Access Memory
RUP	–	Rational Unified Process
SHA1	–	Secure Hash Algorithm 1
SQL	–	Structured Query Language
WWW	–	World Wide Web

---

# CHAPTER 01 – INTRODUCTION

## 1.1 Introduction

Transport service most important facts into organization for distributing their products and services to customer. An organization needs proper transport system to deliver goods and services to customers on time. A proper transport system is important role to handling distribution and maintains the company reputation with the customer. Further the company can be managed the transport cost (Fuel, driver allowance, vehicle expense etc.) and maintains vehicles details.

Macksons Holdings is a diversified conglomerate since (1983) involved in manufacturing, retailing, property development, franchising, trading, imports and exports in household, leisure, automobile, building materials. The company distributes their products and services island wide. Their documentation parts are handling with manual system. This manual system interrupts the day to day activities. Therefore, I introduce a proper transport management system. “Fleet Management System” the system will help to increase the competency and efficiency of their works.

## 1.2 Motivation of the Project

- There are many companies under Macksons holding (Pvt) limited. Therefore, introducing proper fleet management system for delivery vehicles to simplify the deliveries.
- There is no proper mechanism to manage vehicle details and driver details of the company.
- Facing many difficulties to allocate vehicles and drivers for delivery.
- There is no systematic way to generate daily, monthly reports. Because of that it will cost more time to achieve that task and the calculations won't be accurate too.
- This Project was selected for my work place which has a daunting and time consuming manual process of organizing and distributing their goods on time.

---

## 1.3 Scope of the Proposed Project

The scope of the project is mean overall functionalities expected from the proposed system. when deciding to develop a system, the scope is a most important fact which we need to consider about. The scope of this project will be as follows.

- System should be able to store vehicles information
- System should be able to store driver's information
- System should be able to request vehicle
- System should be able to allocate vehicle and driver
- System should be able to manage delivery expenses
- System should be able to manage driver payment
- System should be able to manage vehicle maintaining expenses (such as repair charges, services charges)
- System should be able to manage vehicle license & insurance information
- System should be able to manage fuel stock
- System should be able to generate reports e.g. Delivery expenses report, vehicle maintenance expenses report, etc.

## 1.4 Objectives of the System

The main objective of this system is introduce ideal method to handling daily duties of transport department. The system can be eliminated their paper works and manual process. Therefore, company will be increased more productivity and efficiency of their work through this system.

Further, system should be able to provide information for make quick decision. The system provides accurate reporting and user can be preview reports any time. The system should be able to provide safety and minimize storage.

---

## **1.5 Structure of the Dissertation**

This Dissertation provides the overall knowledge about the Transport Management System. Dissertation structure as follows

### **Chapter 02-Analysis**

This chapter describes the requirement gathering techniques, use case diagram for current system functions, existing similar systems, functional nonfunctional requirements and non- functional requirement of the system.

### **Chapter 03-Design**

This chapter describes alternative solution, selected solution, designing diagrams & user interface designs of the system.

### **Chapter 04-Implementation**

This chapter describes the implementation environment, code and module structure descriptions & reused existing components.

### **Chapter 05-Evaluation**

This chapters describes the techniques of testing, high level test plan of the system, test results and user evaluation of the system

### **Chapter 06-Conclusion**

This chapter describes the future enhancements of the system and lesson learnt of the overall project work.

# CHAPTER 02 – ANALYSIS

## 2.1 Introduction

System analysis is a one of the main phases in the software development life cycle. This involves identify the business processes, gathering client requirement, understand the client requirement and evolving solutions for overcoming the weaknesses of the existing system so as to achieve the organizational goals.

## 2.2 Existing system

### 2.2.1 Use case diagram for Existing System

Figure 2.1 shows the use case diagram of the current manual system

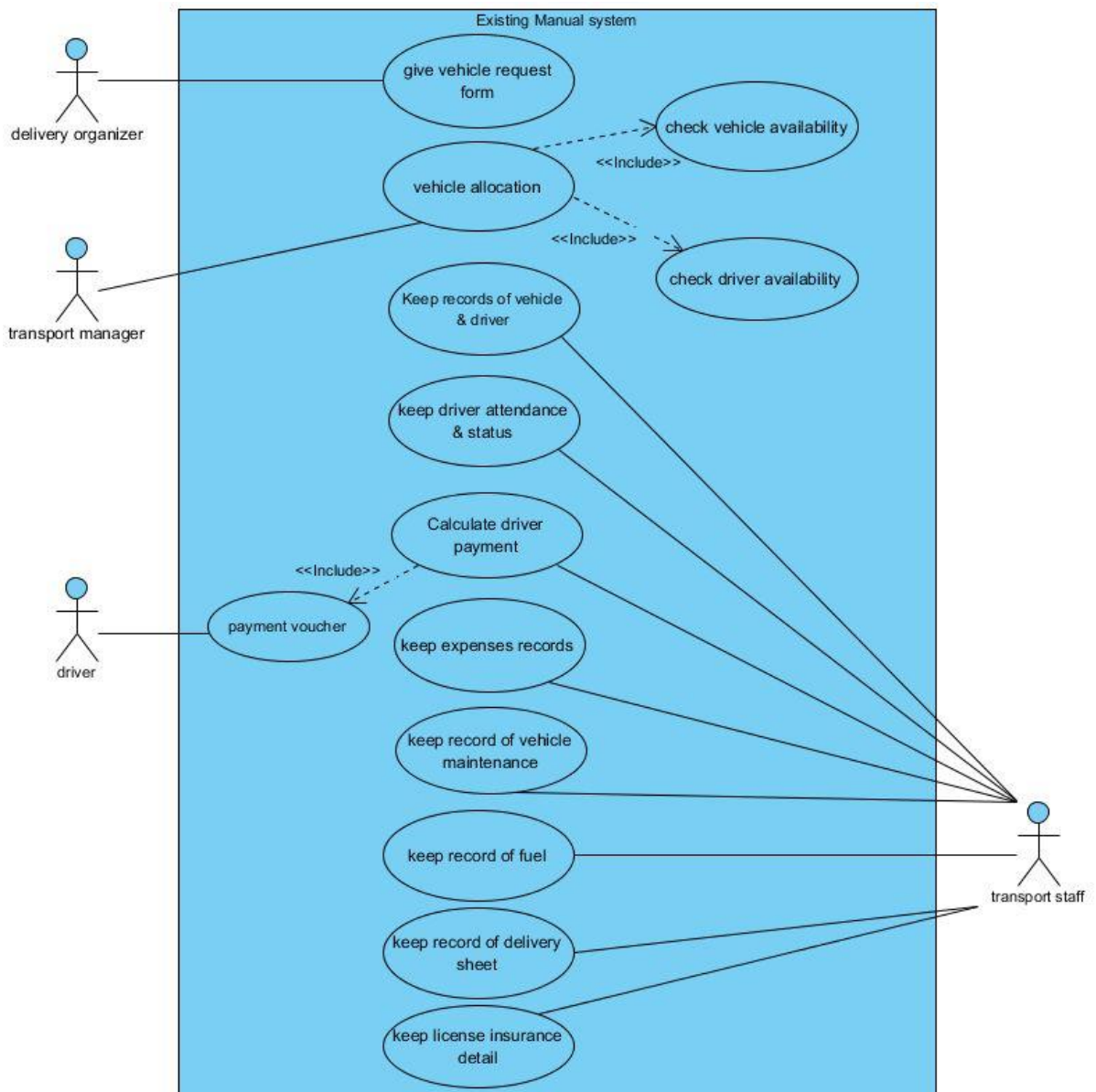


Figure 2.1 Use case diagram for existing system



---

## 2.2.2 Drawbacks of the Existing System

When vehicle information collecting taken long time period by manually

- Paper base document can be misplaced (vehicle request) therefore goods cannot be delivered on time and no evidence to prove delivery organizer for their vehicle request
- Vehicle allocation decision can take long time and sometime same vehicle can be allocated for two difference deliveries
- Vehicle license renewal date would not be identified some time when the manual system therefore penalty to be paid.
- Driver Payments are calculated by manually therefore error might be occurred
- Vehicle maintenance information are maintaining in manually eg. License expire date
- Tracking vehicle expenses is difficult

## 2.3 Existing Similar Systems

Fleet management system is important to the company controlling the transport management which system very less of software providers only supply to the market. Some fleet management software system in market

### 2.3.1 vehicle Fleet manager 4

Provide by Vinity soft. VFM is one of the leading values currently available in fleet management software. Designed to service large fleets, small

fleets, and everything in between, our fleet management solution will track vehicle and driver statistics, Inventory Management, Maintenance Scheduling, Maintenance Tracking, Mileage Tracking, Tire Management, Parts Management, maintenance records, fuel stats, and more. In-depth reporting gives you all the tools you need to make well informed decisions in a data driven world [1].



---

### 2.3.2 Fleetio



Fleetio's suite of fleet management solutions helps fleets of all sizes automate fleet operations, manage asset lifecycle and track drivers with the convenience of a smartphone and the Fleetio Go mobile app. Fleets can track, score and rank their drivers, Driver Management, Fuel Management, Inspection Management, Inventory Management, Maintenance Scheduling, Maintenance Tracking, Mileage Tracking, Parts Management, Vehicle Tracking, Work Order Management. [2]

### 2.3.3 Limitations of above off-the-shelf software

- Some expected requirements were not in above off-the self software
  - Eg: Vehicle requesting option is not available
  - Delivery expenses tracking module is not available
  - Driver payment calculation cannot be customized
- High License cost. Above fleet management software is very expansive and annual or monthly subscriptions fee included. Licenses are based on number of vehicle
- There was bulk of unnecessary functions which was not related to the proposed system.
- When Develop unique software it's the competitive advantage

So that the best option is to make bespoke software to match with the client's requirements

## 2.4 Identify system users

- Admin- Transport manager.
- Staff – transport department user or data entry operator
- Delivery organizer – vehicle requesting person
- Cashier – handover driver payment

---

## **2.5 Functional requirement**

Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.[3]

### **2.5.1 Vehicle Module**

- Transport division staff can be able to insert, view, update, and search vehicle details (registration number, cheese number, vehicle type, vehicle brand, vehicle model, manufacture year, purchase date, License and insurance expiry date, vehicle cylinder capacity, etc.)

### **2.5.2 Driver Module**

- Transport division staff can be able to insert, view, update and search driver details (driver name, employee number, NIC number, contact number, address, date of birth, driving license information etc.)
- Transport division staff can be able to update & driver status

### **2.5.3 Vehicle Request Model**

- Delivery organizer can be able to request their vehicle requirement information such as delivery type (one day delivery or outstation delivery), goods weight, delivery place, and delivery time and return date.
- Delivery organizer can be able to view their vehicle request status such as pending or processed or canceled by transport department.
- System can be allowed to user to cancel their vehicle request before vehicle reserved.

### **2.5.4 Vehicle allocation model**

- Transport manager can be able to view pending vehicle requests for delivery.
- Transport manager can be able to allocate vehicle and driver according to request.
- Transport manager can be able to cancel vehicle request.

---

### **2.5.5 Close delivery module**

- Transport staff can be able close delivery after delivery vehicle arrived.
- Transport staff can be able to view pending close delivery.
- Transport staff can be able to close delivery by insert delivery information such as depart meter, arrive meter, vehicle depart date and time, vehicle arrive date and time and fuel consumption.

### **2.5.6 Driver Payment module**

- After close the delivery system will automatically calculate driver payments eg. Bata and over time payment.
- Cashier can view payment information and issue driver payment and close payment

### **2.5.7 Delivery expenses**

- After close delivery, Transport staff can be able to insert delivery expenses such as meal expenses, room charges, fuel expenses & other expenses
- Transport staff can be able to print delivery expense.

### **2.5.8 Fuel management**

- Transport staff can be able insert & update fuel pump records.
- Transport staff can be able view fuel balance stock.
- Transport staff can be able insert Delivery fuel consumption

### **2.5.9 Maintenance module**

- Transport staff can be able insert, update & view services schedule information.
- Transport staff can be able to inset vehicle maintenance expenses  
Eg. Repairing charges, service charges, Oil changing charges  
Transport staff can be able to inset license renewal expense  
Transport staff can be able to inset insurance renewal expense  
Other expenses
- Transport staff can be able insert, update & view Oil change information
- Transport staff can be able to insert and update license information.
- Transport staff can be able to insert and update insurance information.

---

### **2.5.10 Notification module**

- System has provided facility to remind, when vehicle license is expiring
- System has provided facility to remind, when vehicle insurance is expiring
- The system provides the facility to reminder service schedule.

### **2.5.11 Reports module**

- System provides to generate daily, monthly delivery expense by Company wise.
- System provides to generate daily, weekly and monthly wise driver payment.
- System provides to generate daily, weekly and monthly wise maintenance expense.

### **2.5.12 User Module**

- Users can log into the system securely by validating username and password
- User can create new user account to access into the system
- User can be able to deactivate user account if any user will resign
- User can be able to change user password
- User can be logout from the system.

### **2.5.13 Tracking**

- User can be able to track user activities such as who access system, login time and login IP address.

## **2.6 Non-functional requirements**

Non-functional requirements defined a property and constraints must have in the system. Unless fulfill the non-functional requirements more critical than functional requirements then system is useless.

### **2.6.1 Security**

- Proposed System should be protected by passwords therefore any unauthorized personnel should not be able to access into the system.

---

## 2.6.2 Usability

- This system is able to run parallel with other system in minimum memory usage.
- User can be able to use itself after five hours training
- The system shall have icons and images that can express the idea of the function.

## 2.6.3 Reliable

- The software should be error free software before Provide to customer. Therefore, all system functions and feature should be test.

## 2.6.4 Maintainability

- The system shall be expandable for the changes in the future requirements.

## 2.7 Methodology for the Proposed System

Fleet management is fairly common domain and expected requirements also can be captured by easily. Although there are some uncertainty requirements because of this is an individual project and my first time development, also client side this is a first major project. Therefore, requirements may be changed.

Waterfall model is not suitable because Waterfall model is structural, once all requirements collected it could not be changed. RAD (Rapid Application development) also not suitable because this is my individual project (RAD) is used for team base development

Consider with above issues I have selected RUP (Rational Unified process) Methodology for the proposed system. Rup is an incremental and iterative also it supports object oriented development software development process framework.

## 2.7.1 Rational Unified process

The process is divided into four lifecycle phases, where all of them may happen in one or more iterations. Each phase usually contains five workflows: requirements, analysis, design, implementation and tests. RUP phase is depicted by figure 2.2

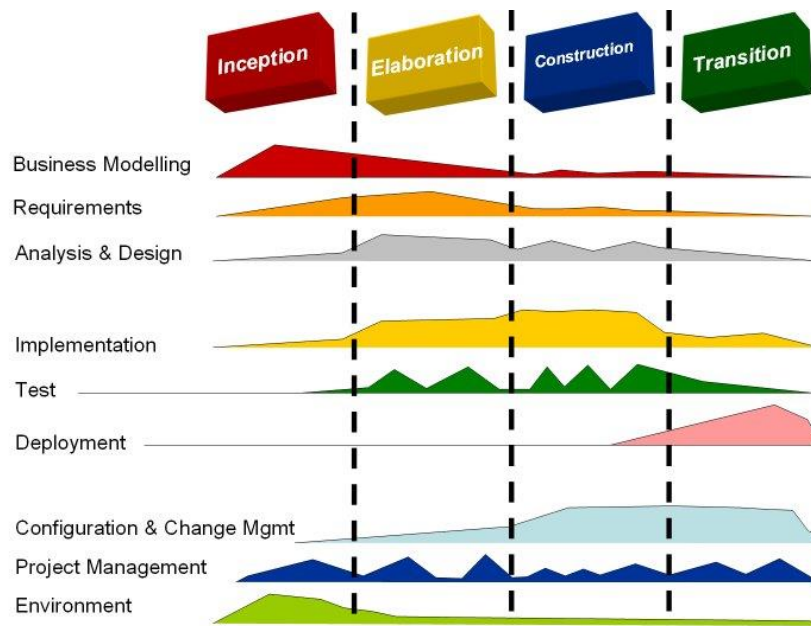


Figure 2.2 RUP phases

Therefore, RUP is the most appropriate SDLC (Software Development Life Cycle) [4].

### 1. Inception Phase

collaborating with the customer and end-users, business requirements for the software are identified and planning activities

### 2. Elaboration Phase

encompasses the planning and modeling activities of the generic process model

### 3. Construction Phase

In this phase finish development based on the baseline architecture

### 4. Transition Phase

In this phase supply the system to its end users.

---

# CHAPTER 03–DESIGN

## 3.1 Introduction

Based on the user requirements and the detailed analysis of a new system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the development of a system. The logical system design arrived at as a result of system analysis and is converted into physical system design. In the design phase the SDLC process continues to move from the what questions of the analysis phase to the how. The logical design produced during the analysis is turned into a physical design - a detailed description of what is needed to solve original problem. Input, output, databases, forms, codification schemes and processing specifications are drawn up in detail. In the design stage, the programming language and the hardware and software platform in which the new system will run are also decided. Data structure, control process, equipment source, workload and limitation of the system, Interface, documentation, training, procedures of using the system, taking backups and staffing requirement are decided at this stage. [5]

## 3.2 Alternative Solutions

### 3.2.1 Use off-the-shelf software

Buying off-the-shelf software may not fulfill client requirements as what they needs. Addition to that paying for the unnecessary components that are not related to client needs developing unique software which gives the competitive advantage.

### 3.2.2 Develop a Standalone System

Developing a standalone system is not ideal solution for this system because Number of users need to access system frequently in deferent place. Standalone system cannot be accessed from deferent place eg. Requesting their vehicle request, management need reports. Further, issues in standalone systems are platform dependent and portability issues



---

### 3.3 Selected Solution

Consider with above alternative issues, I was decided to implement customize Web based system to scratch from the beginning.

I have decided to use open source components to develop the system such as PHP server side language, MYSQL for database & apache web server as the solution is web based.

#### Benefits of Web based system

- User can be able to access the system from deferent place within an organization
- Minimum level memory requirements
- The system is able to run with any other operating system (Platform independently)
- No additional software required in client computer other than web browser
- Web based system is Centralized system therefore easy to maintain
- Lowers cost. Opens source applications are available such as MYSQL for DB, apache for web hosting

### 3.4 Design Architecture

MVC (Model View Controller) architecture has been selected for proposed system. MVC is most popular software design pattern for developing web base application. MVC architecture is depicted by figure 3.1

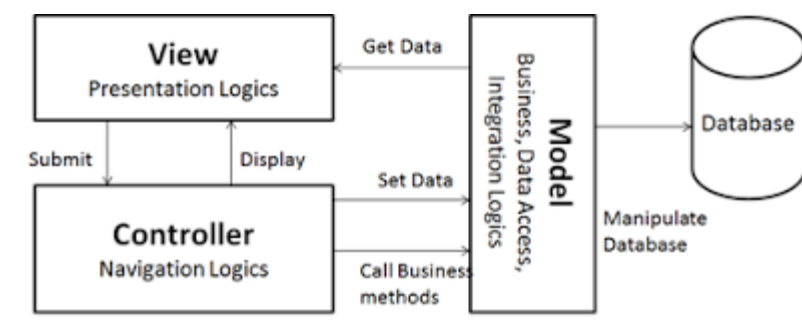


Figure 3.1 MVC architecture

- 
- **Model** - The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself..
  - **View** - A presentation of data in a particular format, triggered by a controller's decision to present the data. They are script based templating systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.
  - **Controller** The controller is responsible for responding to user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.[6]

## 3.5 Design Diagrams

### 3.5.1 The high level use case diagram for the proposed system

Use Case diagram describe a set of use cases, actors and their relationships. In other words, graphically describes who will use the system and in what ways the user expects to interact with the system. Figure 3.2 shows the high level use case diagram of the proposed system



Figure 3.2 high level use case diagram for proposed system

### 3.5.2 Class Diagram for the Proposed System

Most common diagram in modelling object oriented system. class diagram shows set of classes, interfaces and their relationships. Class diagram describe the static structure of a system.

Figure 3.3 shows the class diagram of the proposed system

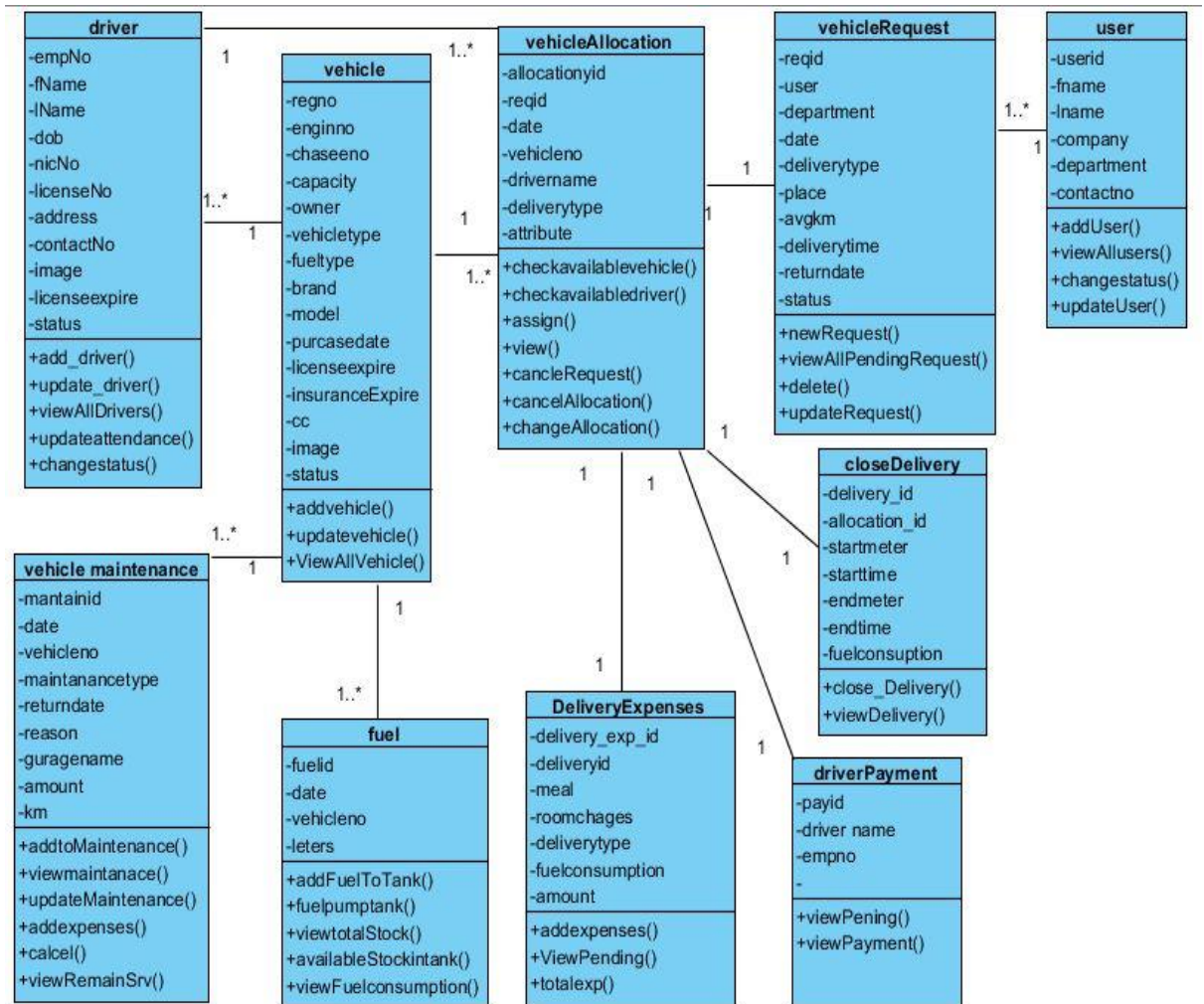


Figure 3.3. class diagram for proposed system

### 3.5.3 Activity diagram for the vehicle allocation

Activity diagram Address the dynamic view of a system. activity shows the flow from activity to activity within a system. Figure 3.4 shows the Activity diagram of the vehicle Allocation module.

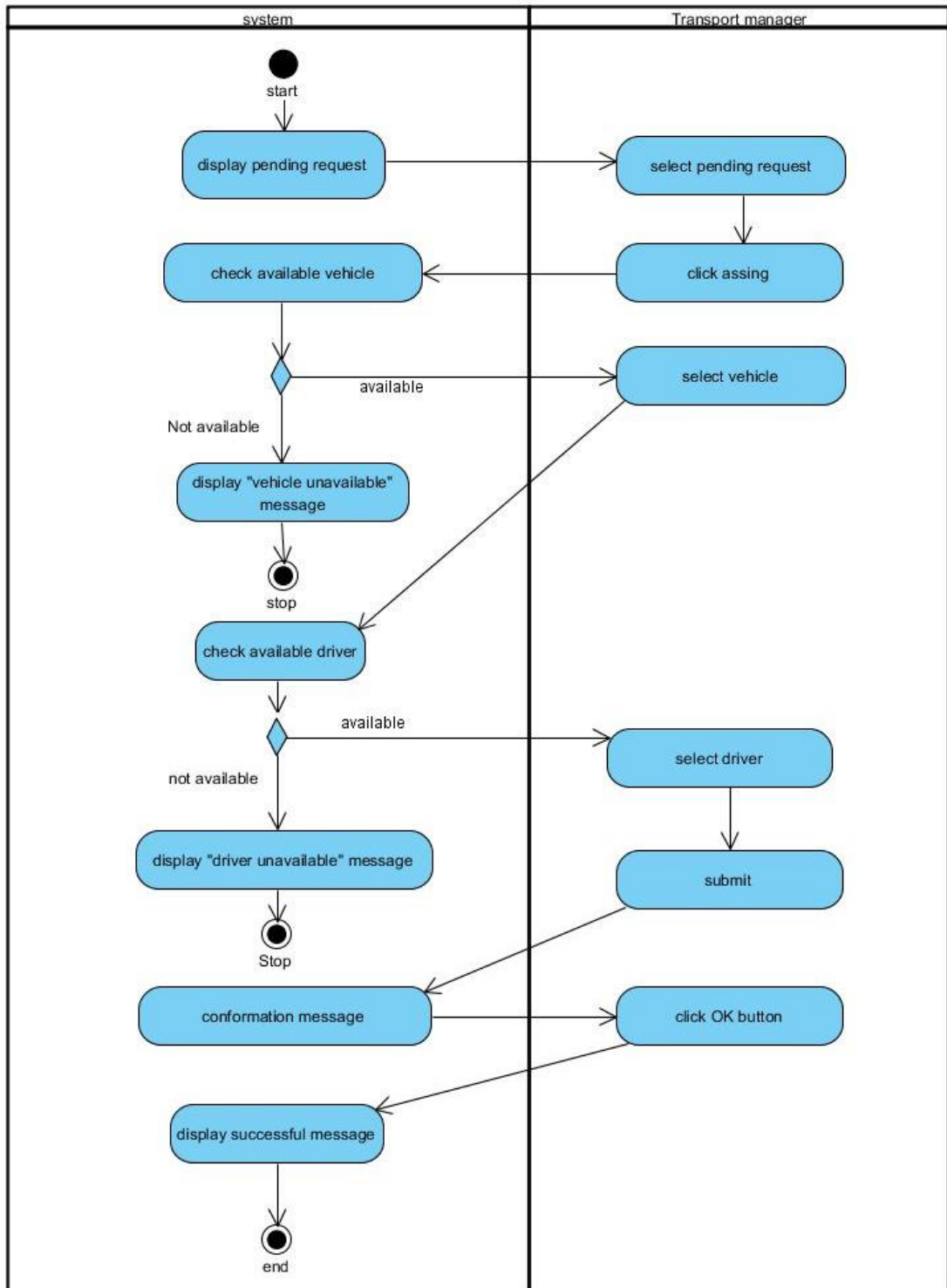


Figure 3.4 Activity diagram for vehicle allocation

### 3.5.4 Sequence diagram for vehicle allocation

sequence diagram is an interaction diagram that shows how objects interact with each other via messages in the execution of operation. Figure 3.5 shows the Sequence diagram of the vehicle Allocation system.

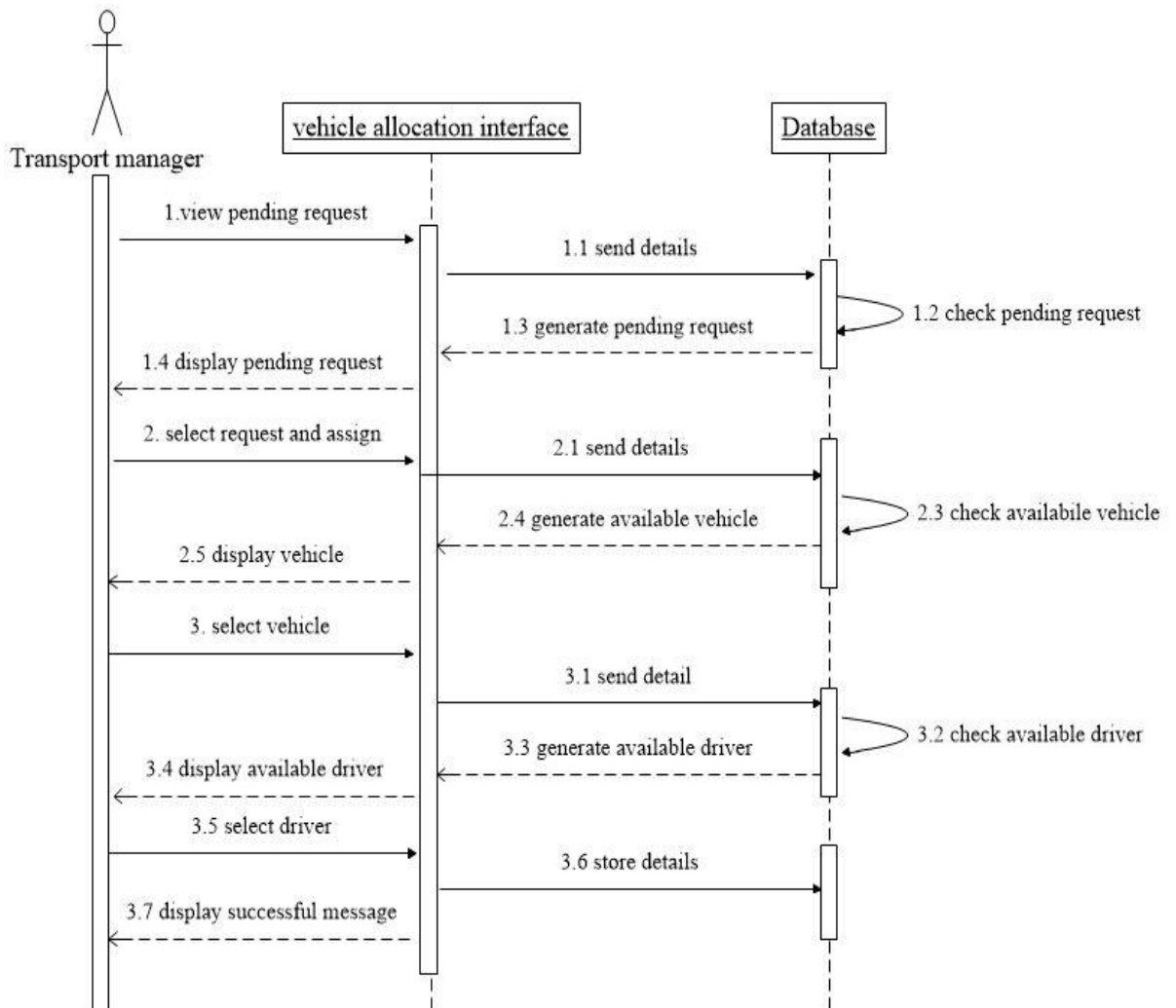


Figure 3.5 Sequence Diagram for vehicle allocation



### 3.5.5 Database Design

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems [7]. Database designing is most crucial process it will help to produce high performance, easy to maintain, improve data consistency and are cost effective in terms of disk storage space. In general database table were normalized into 3<sup>rd</sup> normal form to avoid redundancies. Figure 3.6 shows the ERD diagram of the proposed system.

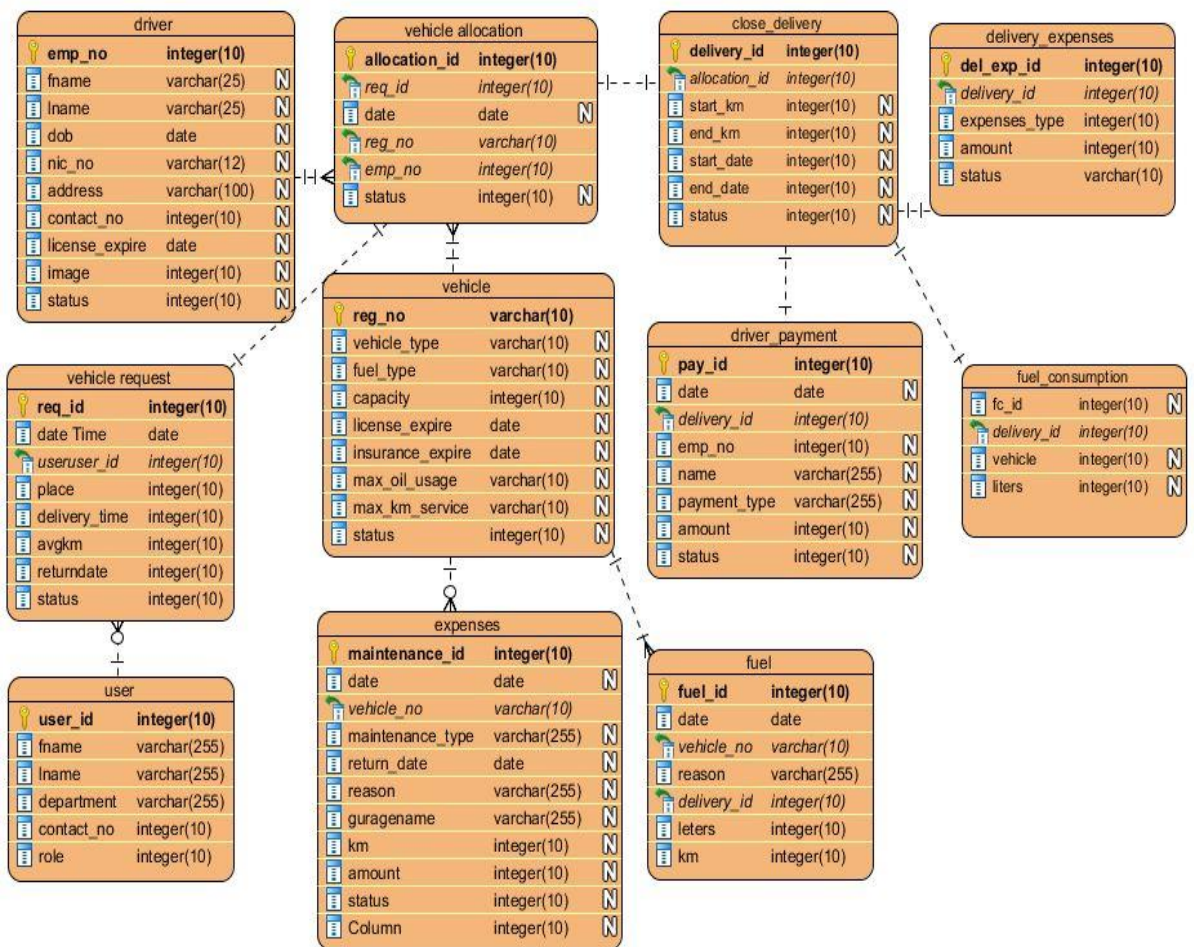


Figure 3.6 ERD diagram for proposed system

---

## 3.6 Interface Design

Interface defined a program or device communicate with user. Therefore, Interface design is an important area of developing the system. An attraction interface design will be impressed to user to work with the system. The successful interface design should be user friendly, simplicity, good colour combination, etc. all above principle of interface design applied in this system.

### 3.6.1 Login Interface

This is the system login page displayed by figure 3.7 is a pleasant look feel to the user. The System cannot be allowed unauthorized user login into the system. System will be displayed error message when unauthorized user's entry.



*Figure 3.7 Login Interface for the system*



### 3.6.2 Administrator Home Page

This is an administrator's home page. All the system modules displayed in the interface are based on icon pattern. It looks very simple and attractive to users and well-recognized icons have been used in this system.

The home page of the administrator's account is displayed by figure 3.8

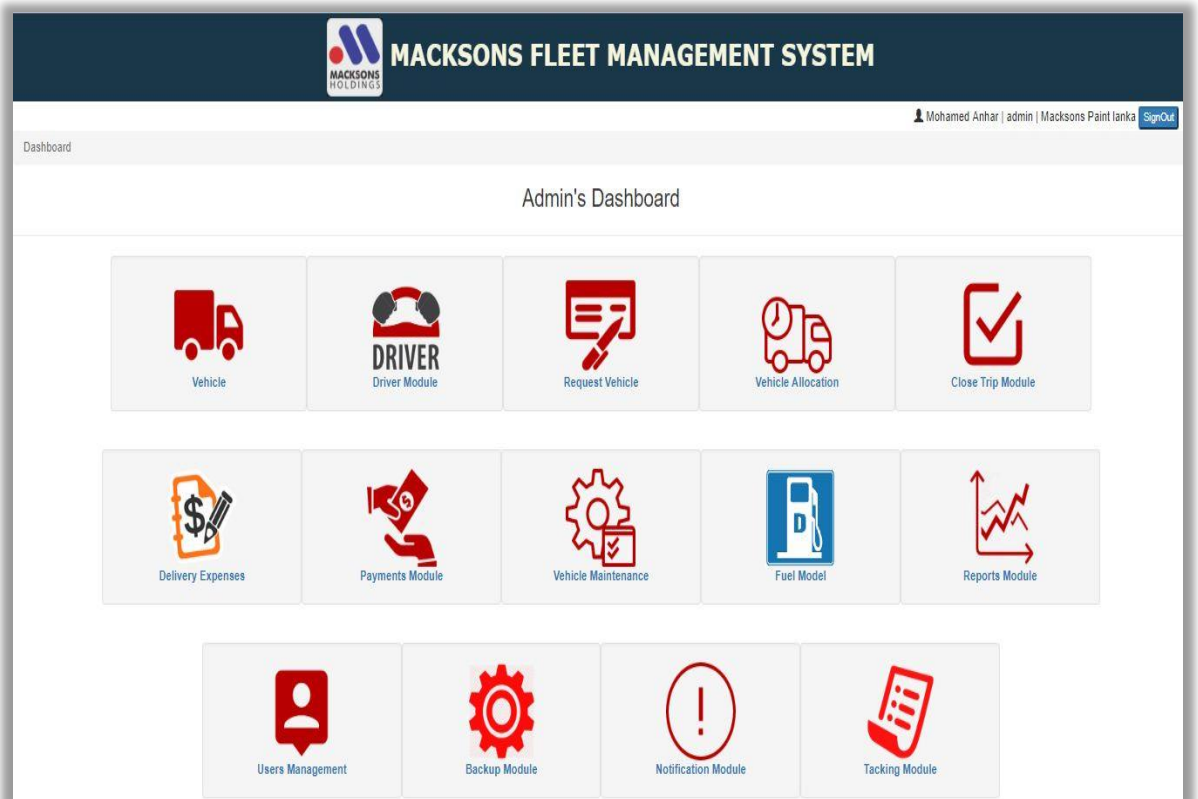


Figure 3.8 Administrator Home Page

### 3.6.3 Vehicle Registration interface

The interface which belongs to Vehicle page displayed figure 3.9. that helps to add new vehicle's information into the system. All mandatory fields are identified by star mark (\*). The system should be identified the recorded vehicle while enter the vehicle number into the cage "vehicle registration number". Here applied Ajax technology.

The image shows a web form titled "Vehicle registration" with a close button (X) in the top right corner. The form is organized into two columns of fields:

- Province \***: A dropdown menu with "Province" selected.
- Vehicle Registration Number \***: A text input field with "Vehicle Number" as a placeholder.
- Chassis Number**: A text input field with "Chassis Number" as a placeholder.
- Engine Number**: A text input field with "Engine Number" as a placeholder.
- Vehicle capacity (tons) \***: A text input field with "Capacity" as a placeholder.
- Cylinder Capacity(cc)**: A text input field with "Cylinder Capacity/ cc" as a placeholder.
- Fule type \***: A dropdown menu with "Select a Fuel Type" selected.
- Vehicle type \***: A dropdown menu with "Select Vehicle Type" selected.
- Owner \***: A dropdown menu with "Select Vehicle Owner" selected.
- Make / Brand**: A text input field with "Brand" as a placeholder.
- Model**: A text input field with "Model" as a placeholder.
- License Expire \***: A text input field with "dd-mm-yyyy" as a placeholder.
- Insurance expire \***: A text input field with "dd-mm-yyyy" as a placeholder.
- Oil Change KM \***: A text input field.
- Service schedule KM \***: A text input field.
- Year of Manufacture**: A text input field with "dd-mm-yyyy" as a placeholder.
- Vehicle Image**: A file upload field with a "Choose File" button and "No file chosen" text.

At the bottom right of the form, there are two buttons: a green "Register" button and a red "cancel" button.

Figure 3.9 vehicle registration form

### 3.6.4 Display Information

According to the client’s requirements the table view’s had created. The important and basic information are displayed to user by default in table view. When user needs more information User can *click* on “**view button**”. User can update/ edit driver information *click* on “**update button**” and also user can change driver status *click* on “**change status button**”. Therefore, this interface should be time saving and easy to use. Figure 3.10 show system display information to user.

**MACKSONS FLEET MANAGEMENT SYSTEM**

Mohamed Anhar | admin | Macksons Paint Lanka [SignOut](#)

Dashboard / Driver

Driver Module

[Add a Driver](#)

Show 10 entries Search:

Employ No	First Name	Last Name	License Expire	Contact No	status	
0	Anhar	Mohamed	0000-00-00		Available	<a href="#">View</a> <a href="#">Update</a> <a href="#">Change Status</a>
1	Arshard	Mohamed	2017-09-30	0711234567	On Delivery	<a href="#">View</a> <a href="#">Update</a> <a href="#">Change Status</a>
2	Mohamed	Anhar	2017-09-30	0713003536	Available	<a href="#">View</a> <a href="#">Update</a> <a href="#">Change Status</a>
3	Ameer	Sameem	2017-10-19	0382299601	Deactivate	<a href="#">View</a> <a href="#">Update</a> <a href="#">Change Status</a>
5	Mifras	Mahamed	0000-00-00		On Delivery	<a href="#">View</a> <a href="#">Update</a> <a href="#">Change Status</a>

Showing 1 to 5 of 5 entries Previous **1** Next

Figure 3.10 Display information

---

# CHAPTER 4: IMPLEMENTATION

## 4.1 Introduction

After completion of the design stage, implementation of the system begins according to what the design stage planned by using adequate tools and techniques. This is an important stage in the software development lifecycle, because this stage builds the system into a working system. This system has been developed as a modularized way and finally integrated all the modules as a complete system. This chapter mainly discusses about the implementation environment, the tools and techniques as well as the reusable components used to implement the system. The major code segments have been included to help the process functionalities of the system in this chapter. Code segmentations which were included in the system with a comment are meant to be used for future improvements.

## 4.2 IMPLEMENTATION ENVIRONMENT

This describes the minimum set of hardware and software configurations for the implementation environment.

Hardware Requirement	Software Requirement
Intel(R) core(TM) i5 3.6GHz Processor	Windows 10 pro
4GB Ram	PHP version 5.6
500GB Hard Disk	Apache 2.4
	Mysql Server

Table 4.1 hardware and software requirement

Although Fleet Management System was developed with above configuration, the system is fully compatible with windows XP, windows Vista, windows 7, windows 8.1.

---

## 4.3 Major Code Segments

The main code modules developed in the system have been mentioned below by briefly describing their functionality. Code modules consist with comments to identify the specific reason of a particular code line.

### Database connection Page

Web Based system work with database. system frequently connect with database server for retrieve data, insert date, update date therefore creates a common database connection php page called **dbconnection.php** .

Predefined MYSQL function called “**mysqli()**” helps to manage the connection by passing four parameters in its host name, user name, password & database name.

```
<?php
//database connection class
class dbconnection{
    public $host="localhost" ;
    public $user="root";
    public $password="";
    public $db="fleetmanagement";
//create function call connection
public function connection(){
    $connect=new mysqli($this->host,$this->user,$this->password,$this->db) or die($connect->error);
    return $connect;
    }
} ?>
```

---

## Login Controller Page

Code segment has given below gets the database connection and checks that the user exists according to the input. If exists, then check the password, if yes, check the user status, whether the person's account is active or deactive. If active, continue the process; otherwise redirect to the login page with an error message.

```
//start session
if(!isset($_SESSION)){
    session_start();
}

// Get username and password from view
$userName=trim($_POST['username']);
//Encrypt password plan to sha1
$password= sha1($_POST['password']);

//validate user name or password is Empty before send information to
server
if($_POST['username']=="" || $_POST['password']=="" ){
    //encode msg
    $_msg= base64_encode("Empty Username or password server");
    //redirect login with a msg
    header("Location:../view/login.php?msg=$_msg");
}

//include database connection page and connect database
include '../common/dbconnection.php';
$dbconobj =new dbconnection();
$connect=$dbconobj->connection();

//send user name & password to server for valid user name and
password is correct
```

```

include '../model/Loginmodel.php';

$loginobj=new login();

$result=$loginobj->loginValidate($userName, $password);

if($result->num_rows>=1){
    //valid user name and password system continues to user home page
    // some code segment at beginning (Login tracking)
    header("Location:../view/Dashboard.php");
    }
    //invalid user name and password display error message and redirect
    to login page
Else{
    $msg=base64_encode("Invalid Username or password");
    header("Location:../view/login.php?msg=$msg");
    }
}

```

Code segment has given below LoginValidation() function in model

```

function loginValidate($userName,$password){
    //super global variable from login controller page
    $connect= $GLOBALS['connect'];

    //Sql Query for check user name, password and user status "active "
    and get additional information about user to session
    $sql= "SELECT * FROM user_login l, user_info u, company c ,user_role r WHERE
    luser_name='$userName' AND l.password='$password' AND user_status='active' AND
    l.user_id=u.user_id AND u.role_id=r.role_id AND c.company_id=u.company_id" ;

    //execute query
    $result=$connect->query($sql) or die ($connect->error);
    return $result;
}

```

---

Code segment has given below creates initial required username as a session, user id, login time and gets the user's IP address which needs to be stored in the database. Then pass the login time, ip address, user id, username and an initial value to “trackIn()” function for saving the login information to “login\_track” table in database.

```
//Some code segments at the beginning
//get user information
$row=$result->fetch_array();
    $_SESSION['user_info']=$row;

//track user login information
    $login_time=date("y-m-d")." ".date("H:i:s");
    $user_id=$row['user_id'];
    $user_name=$row['user_name'];
    $ip=$_SERVER['REMOTE_ADDR'];

//push session information's to database
include '../model/Trackingmodel.php';
    $trackingobj=new tracking();
    $track_id=$trackingobj->trackIn($login_time,$ip,$user_id,$user_name);
    array_push($_SESSION['user_info'],$track_id);
```

## Sign-out Page

The function below describes about the sign-out page. After the user had successfully logged into the system this function helps the user to terminate his logged session from the system. After calling this function it requests the “trackout()” and save the logged out time of the user. If time saved successfully the session gets unset and destroyed. After session destroyed sign out page will redirect to system login page with in 5seconds.



```
date_default_timezone_set("Asia/colombo");
```

```
$user_info=$_SESSION['user_info'];
```

```
$track_id=$user_info[18];
```

```
$out=date("Y-m-d")." ".date('H:i:s');
```

```
//create database connection
```

```
include '../common/dbconnection.php';
```

```
$ob=new dbconnection();
```

```
$connect=$ob->connection();
```

```
include '../model/Trackingmodel.php';
```

```
$objso=new tracking() ;
```

```
Save the Logout Time in the database and destroy the session
```

```
$objso->trackout($track_id,$out);
```

```
unset($_SESSION['user_info']);
```

```
//redirect to login page within 5 second
```

```
header("refresh:5,url=../view/login.php");
```

```
?>
```

## Vehicle Management Function

Transport department staff carried out using this function. Adding, updating and change status of the vehicle information. Following code segment refers to the Inset Vehicle from that function.

```
//vehicle controller send information to addvehicle function
```

```
$status= strtolower($_REQUEST['status']);
```

```
switch($status){
```

```
    case "add": $vehiclearray=$_POST;
```

```
$vehicleobj->addvehicle($vehiclearray,$iname,$tmp_loc);
```

```
//get user inserted vehicle information to model via $vehicle array
```

```

from controller

public function addvehicle($vehiclearray,$iname,$tmp_loc){

    $province=$vehiclearray['province'];
    $vehicle_reg=$vehiclearray['vehicle_regno'];
    $vehicle_chasno=$vehiclearray['chassis_no'];
    $vehicle_engno=$vehiclearray['engine_no'];
    $vehicle_weigh=$vehiclearray['weigh'];
    $vehicle_cc=$vehiclearray['cc'];
    $fuel_type=$vehiclearray['fule_type'];
    $vehicle_type=$vehiclearray['vehicle_type'];
    $vehicle_owner=$vehiclearray['owner'];
    $vehicle_brand=$vehiclearray['brand'];
    $vehicle_model=$vehiclearray['model'];
    $license_exp=$vehiclearray['license_expire'];
    $insurance_exp=$vehiclearray['insurance_expire'];
    $max_oil=$vehiclearray['oil_change_km'];
    $service_km=$vehiclearray['servicekm'];
    $manu_year=$vehiclearray['year_of_manu'];

    //use controller page connection through globals variable
    $connect=$GLOBALS['con'];

    $sql="INSERT INTO vehicle (province, reg_no, chasno, engine_no, capacity_ton,
    cylinder_capa,fuel_type, vehicle_type, owner,brand,model,year_of_manufacture,status)
    values ('$province', '$vehicle_reg', '$vehicle_chasno', '$vehicle_engno',
    '$vehicle_weigh', '$vehicle_cc', '$fuel_type', '$vehicle_type', '$vehicle_owner',
    '$vehicle_brand','$vehicle_model','$manu_year','available')" ;

    $resulte=$connect->query($sql) or die ($connect->error) ;

    //upload vehicle Image

```

---

```

if($iname!=""){
    $vnewname=$vehicle_reg."_".$iname;
    $update="UPDATE vehicle SET vehicle_image='$vnewname' WHERE
    reg_no='$vehicle_reg'";
        $connect->query($update) or die($connect->error);
        $new_path="../Images/vehicle_image/".$vnewname;
// Move image from temp location to new path
        move_uploaded_file($tmp_loc,$new_path);
    }
//insert vehicle to maintenance Setup Table

$maint_setupsql="INSERT INTO maintenance_scheduler
(vehicle,license_expire,insurance_expire,max_oil,maxkm_service)
VALUES('$vehicle_reg','$license_exp','$insurance_exp','$max_oil','$service_km)";
$main=$connect->query($maint_setupsql) or die ($connect->error) ;
return $resulte;
}

```

The following code segment also belongs to the Vehicle management function and it is about the vehicle Updating process.

```

//vehicle controller get information from user input
$status= strtolower($_REQUEST['status']);
switch($status){
case "update": $vehiclearray=$_POST);
public function updatevehicle($vehiclearray,$iname,$tmp_loc){

Some code segments at the beginning
$connect=$GLOBALS['con'];

$sql="UPDATE vehicle SET chasno_no='$vehicle_chasno',
engine_no='$vehicle_engno', capacity_ton='$vehicle_weigh',
cylinder_capa='$vehicle_cc', fuel_type='$fuel_type', vehicle_type='$vehicle_type',
owner='$vehicle_owner', brand='$vehicle_brand', model='$vehicle_model',
year_of_manufacture='$manu_year' WHERE reg_no='$vehicle_reg' " ;
$result=$connect->query($sql) or die ($connect->error) ;

//update image
    if($iname!=""){
        $vnewname=$vehicle_reg."_".$iname;
        $update="UPDATE vehicle SET vehicle_image='$vnewname' WHERE
reg_no='$vehicle_reg'";
        $connect->query($update) or die($connect->error);
        $new_path="../Images/vehicle_image/".$vnewname;
// Move image from temp loc to new path
        move_uploaded_file($tmp_loc,$new_path);
    }
//update maintenance setup table
$upmaint_setupsq=$sql="UPDATE maintenance_scheduler SET
license_expire='$license_exp',insurance_expire='$insurance_exp',max_oil='$max_oil',
maxkm_service='$service_km' WHERE vehicle='$vehicle_reg' " ;
$updatemain=$connect->query($upmaint_setupsq) or die ($connect->error) ;

```

---

## 4.4 Re-usable Components

Some codes were found by referring internet and used in this project.

- **Bootstrap** - is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions [8]
  
- **Data table** - DataTables Table plug-in for jQuery. DataTables is a plug-in for the jQuery Javascript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, and will add advanced interaction controls to any HTML table [9]
  
- **Datepicker** - The datepicker is tied to a standard form input field. Focus on the input (click, or use the tab key) to open an interactive calendar in a small overlay. Choose a date [10]
  
- **Fusion chart** - provider of data visualization products (JavaScript Charts, Maps, Widgets and Dashboards. used for display fuel chart [11]

---

# CHAPTER 05- EVALUATION

## 5.1 Introduction

“Evaluation is a systematic determination of a subject's merit, worth and significance, using criteria governed by a set of standards. It can assist an organization, program, project or any other intervention or initiative to assess any aim, realizable concept/proposal, or any alternative, to help in decision-making; or to ascertain the degree of achievement or value in regard to the aim and objectives and results of any such action that has been completed. The primary purpose of evaluation, in addition to gaining insight into prior or existing initiatives, is to enable reflection and assist in the identification of future change”.[12]

## 5.2 System Test

System Testing is a level of the software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements. Software testing is an investigation conducted to provide stakeholders with

## 5.3 Test Plan and System Test Cases

The testing begins with the implementation. While developing the code the system should be tested to check for the errors. Some test cases designed for major system modules are as follows: (Please refer Appendix E for all the test cases and results).

### 5.3.1 Test Cases for User Authentication

The test cases and results of the User Authentication Module are displayed by Table 5.1

<b>Test Number</b>	<b>Test description</b>	<b>Step to Test</b>	<b>Expected Result</b>
1.	Validate User login input details	Enter correct username Enter correct password Active user status	Successfully login to the system
2.	Validate User login input details	Empty username Empty password	Display error message “oops! Empty User name and Password”
3.	Validate User login input details	Empty username Enter password	Display error message “oops! Empty User name”
4.	Validate User login input details	Enter correct username Empty password	Display error message “oops! Empty Password”
5.	Validate User login input details	Enter incorrect username Enter correct password Active user status	Display error message “Invalid Username or password”
6.	Validate User login input details	Enter correct username Enter incorrect password Active user status	Display error message “Invalid Username or password”
7.	Validate User login input details	Enter correct username Enter correct password deactive user status	Display error message “Invalid Username or password”

*Table 5.1 Test Cases for User Authentication*

### 5.3.2 Test Cases for Manage User

The test cases and results of the Manage User Module are displayed by Table 5.2

<b>Test Number</b>	<b>Test description</b>	<b>Step to Test</b>	<b>Expected Result</b>
1.	Validate Personnel Information adding	Add Information with relevant details	Display Message “Successfully ADD User”

2.	Validate Personnel Information adding	Try to add information by keeping all fields in empty	Display Error Message “empty (first mandatory)” and focus fist name
3.	Validate Personnel Information adding	Without enter required fields and submit	Display Error Message empty (first mandatory )field name and Focus that field.
4.	Validate Personnel Information adding	Enter existing email address	On keep up Display Message “Already taken” on submit Display “existing Email”
5.	Validate Personnel Information adding	Try to create Without Selecting a role	Display Error Message “Select User role”
6.	Update user information	Select user, click update button and change information and click update	Display Error Message “Update Successfully”
7.	Deactive user	Select User and click “deactive” button	Conformation popup Appears “Are you sure to deactive user?”
8.	Active user	Select User and click “active” button	Conformation popup Appears “Are you sure to active user?”

*Table 5.2 test cases for manage users*

### 5.3.3 Test cases for Driver registration

The test cases and results of the driver registration Module are displayed by Table 5.3.

Test Number	Test description	Step to Test	Expected Result
1.	Validate driver information adding	Add Information with relevant details	Display Message “Successfully ADD Driver”
2.	Validate driver information adding	Try to add information by keeping all fields in empty	Display Error Message “empty employee No” and Focus Employee



			number Field.
3.	Validate driver information adding	Enter all fields without employee number	Display Error Message “empty employee Number” and Focus Employee number Field.
	Validate driver information adding	Enter existing employee number	On keep up Display Message “Already taken” on submit Display Error Message “existing Employee number”
4.	Validate driver information adding	Enter all fields without employee number correctly	Display Error Message “Invalid employee Number”
5.	Validate driver information adding	Enter all fields without First Name	Display Error Message “empty First Name” and Focus First Name Field.
6.	Validate driver information adding	Enter all fields without Last Name	Display Error Message “empty Last Name” and Focus First Name Field.
7.	Validate driver information adding	Enter incorrect date of birth (validate Enter age below 18 and above 50)	Display Error Message if below 18 “Under age”
8.	Validate driver information adding	Enter incorrect NIC	Display Error Message “Invalid NIC Number”
9	Validate driver information adding	Register without select “driver Type”	Display Error Message “Select Driver Type”
10.	Update Driver	Select driver click update button and change information and submit	Display ”driver information Successfully Updated” Message
11.	Change driver status	Select driver and click change status button	Confirmation Popup appears “Are you sure want to [status] driver?”

*Table 5.3 test case for driver registration*

### 5.3.4 Test case for request vehicle

The test cases and results of the vehicle request Module are displayed by Table 5.4

Test Number	Test description	Step to Test	Expected Result
1.	Validate Vehicle Request form	Add Information with relevant details	Display Message “vehicle Request Sent Successfully”
2.	Validate Vehicle Request form	Try to add information by keeping all fields in empty	Display error message”Select Delivery type ” and focus the Delivery Type field.
3.	Validate Vehicle Request form	Enter all other field except “delivery Type” field	Display Error Message “Select Delivery Type” and focus the field
4.	Validate Vehicle Request form	Enter all other field except “Place” field	Display Error Message “Enter Place” and focus the field
5.	Validate Vehicle Request form	Enter all other field except “Goods weight” field	Display Error Message “Enter Goods weight” and focus the field
6.	Validate Vehicle Request form	Enter all other field except “Depart date and Time” field	Display Error Message “select Depart date and Time” and focus the field
7.	Validate Vehicle Request form	Enter all other field except “Return date” field	Display Error Message “select Return date” and focus the field
8.	Edit request	Select the request and click “edit” button and change information and click update	Display Message “Update successfully” and redirect to vehicle request page
9.	Cancel request	Select the request and click “cancel” button	Popup the Conformation dialog box “Are you sure to

			cancel Request ID ?”
--	--	--	----------------------

*Table 5.4 test cases for request vehicle*

### 5.3.5 Test case for Vehicle Allocation

The test cases and results of the vehicle allocation Module are displayed by Table 5.5

Test Number	Test description	Step to Test	Expected Result
1.	Validate Vehicle allocation form	Add Information with relevant details	Display Message “Vehicle allocated Successfully”
2.	Validate Vehicle allocation form	Try to add information by keeping all fields in empty	Display Error Empty Vehicle No and focus first field”
3.	Validate Vehicle allocation form	Select all other field except “Vehicle No” field	Display Message “ Select Vehicle No and focus the field”
4.	Validate Vehicle allocation form	Select all other field except “Driver Name” field	Display Message “ Select Driver and focus the field”
5.	Cancel request	Select the Allocation and click “cancel” button	Popup the Conformation dialog box “Are you sure to cancel the Allocation”

*Table 5.5 test cases for vehicle Allocation*

### 5.3.6 Test Cases for Close Delivery

The test cases and results of the close delivery Module are displayed by Table 5.6

Test Number	Test description	Step to Test	Expected Result
1.	Validate Close delivery form	Add Information with relevant details	Display Message “Close Delivery Successfully”
2.	Validate Close	Try to add information by	Display Empty first Message and focus

	delivery form	keeping all fields in empty	first field”
3.	Validate Close delivery form	End date time and less than Start date time	Display Message “invalid End date” and focus the field
4.	Validate Close delivery form	End Km and less than start Km time	Display Message “invalid End Km” and focus the field
5.	Validate Close delivery form	Enter all other field valid information except “Fuel consumption date” field	Display Message “Enter Fuel consumption” and focus the field

Table 5.6 test case for close delivery

## 5.4 TEST DATA AND TEST RESULTS

Some test results are given below for some test cases and the rest of the test results can be found in Appendix E

- **Manage User Test case**

The test cases with results belong to the manage user module is displayed by Table 5.7

Test Number	Step to Test	Expected Result	Status
1.	Add Information with relevant details	Display Message “Successfully ADD User”  Successfully Add User	Pass
2.	Try to add information by keeping all fields in empty	Display Error Message “empty (fist mandatory)” and focus fist name.  Empty First name Name	Pass
3.	Without enter Mandatory fields and submit	Display Error Message empty (fist mandatory)field name and Focus that field.  Empty First name Name	Pass
4.	Enter existing user name	On keep up Display Message “Already taken”  Already registered	Pass

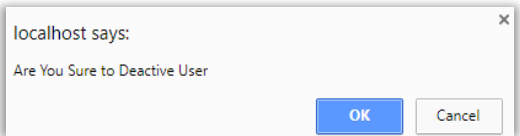
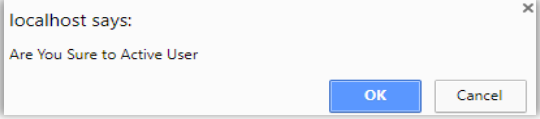
		on submit Display “existing Email” <b>Existing User Name</b>	
5.	Try to create Without Selecting a role	Display Error Message “Select User role” <b>Select Role Name</b>	Pass
6.	Select user, click update button and change information and click update	Display Error Message “User Update Successfully” User Update Successfully	Pass
7.	Select User and click “deactive” button	Conformation popup Appears “Are you sure to Deactive user?” 	Pass
8.	Select User and click “active” button	Conformation popup Appears “Are you sure to active user?” 	Pass

Table 5.7 : Test results for Manage User Test case

## • Test cases for Add Driver

The test cases with results belong to the add driver module is displayed by Table 5.8

Test Number	Step to Test	Expected Result	Status
1.	Add Information with relevant details	Display Message “Driver Registered Successfully” Driver registered Successfully	Pass
2.	Try to add information by keeping all fields in empty	Display Message “Empty Employee No” and Focus Employee Number Field. <b>Empty Employee Number</b>	Pass
3.	Enter all fields without employee number	Display Message “empty employee No” and Focus Employee number Field. <b>Empty Employee Number</b>	Pass

4.	Enter existing employee number	On keep up Display Message “Already taken” on submit Display “existing Employee No”  <b>Exisiting Employee No</b>	Pass
5.	Enter all fields without First Name	Display Message “empty First Name” and Focus First Name Field.  <b>Empty First Name</b>	Pass
6.	Enter all fields without Last Name correctly	Popup the error message “Empty Last Name”  <b>Empty Last Name</b>	Pass
7.	Enter incorrect date of birth (validate Enter age below 18 and above 50)	Display Message “Under Age”  <b>Under Age (below 18)</b>	Pass
8.	Enter incorrect NIC	Display Message “Invalid NIC”  <b>Invalid NIC</b>	Pass
9	register without select “driver Type”	Display Message “Select Driver Type”  <b>Select driver Type</b>	Pass
10.	Select driver click update button and change information and submit	Display a message ”driver information Successfully Updated”  <b>Driver Update Successfully</b>	Pass

*Table 5.8 Test results for Add driver*

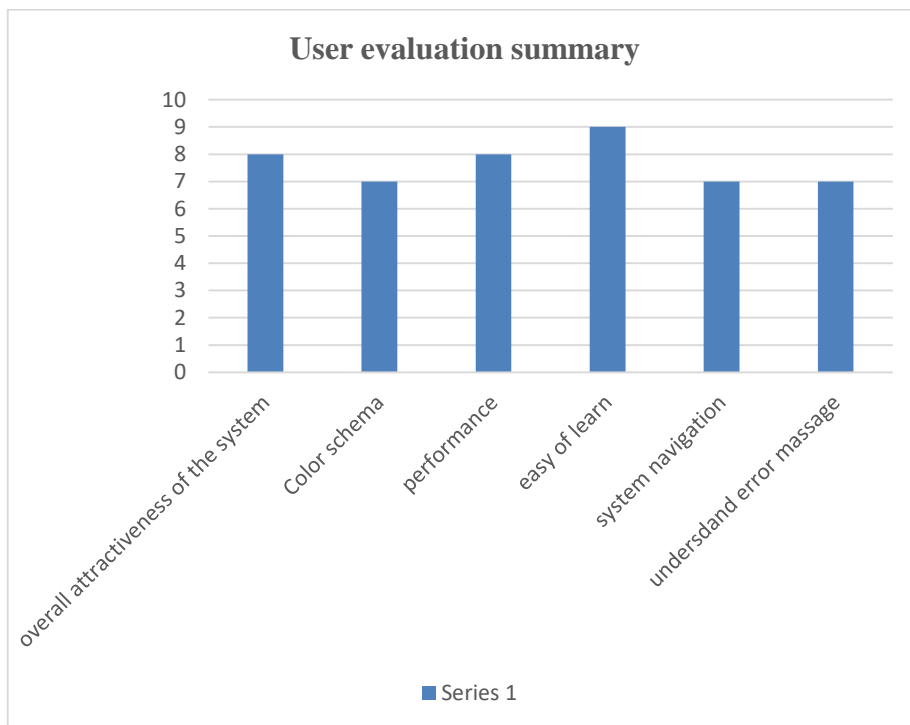
## 5.5 Use Acceptance Testing

User acceptance testing is a last phase of software testing process. After implementing the system, it was tested in the client environment to get the user acceptance testing. The system was tested by the client to identify the functionality provided by the system, whether it can satisfy the operational needs of the company. By introducing the system briefly, the client could understand how the system works. User Acceptance Testing was started by feeding the actual dataset to the developed system. There were some minor modifications required for the system.

Generally, User evaluation questionnaire was given to target population and results has summarized. Following figure 5.1 shows user evaluation questionnaire.

User Evaluation Questionnaire					
Name of User :- <i>chanaka</i>		Role of user:- <i>Transport manager</i>			
Evaluating Item	Very Good	Good	Average	Poor	Very Poor
Overall attractiveness of the system	✓				
Understandability of Error Message		✓			
Color Schema		✓			
Performance	✓				
Easy of Learning	✓				
Functionalities	✓				
Interfaces		✓			
System Navigation		✓			
Comments :					
Signature :..... <i>[Signature]</i> .....					

*Figure 5.1 User evolution questionnaire*



---

# CHAPTER 06 – CONCLUSION

## 6.1 Introduction

Macksons Holdings is a diversified conglomerate involved in manufacturing, retailing, property development, franchising, trading, imports and exports in household, leisure, automobile, building materials. They distribute goods island wide therefore they need to decide enhance their transportation system effective with new technology. Earlier they face many difficulties with manual system and day to day activities such as maintaining vehicle status, handling vehicle request, allocating vehicle and driver for vehicle request and report generating process. They have to handle lot of time to this process. Because of this Fleet management system, they can carry out their day to day activities easily and effectively.

During the requirement gathering phase functional and non-functional requirements were identified in each module. After completing client functional and non-functional requirements and system deploy into client environment some additional functionalities and features were added which client requested. Further client was satisfied about my system finally

## 6.2 Future Enhancement

This proposed system was developed within a time constraint and functionalities which were agreed by the client. Following are the some of the future enhancement of the system

- GPS Tracking for Delivery Vehicle
- Stock management for vehicle parts
- Improving reports and showing graphs by using analytical tool.



---

## 6.3 Lessons learnt

This project gave me a valuable system development experience to my life. I learnt how to use programming theories in practically. Further when we delivered the project in a time constraint, i learnt how to handle the pressure situation in a project it was a good experience for my life. Following are the some of lesson learnt things.

- Good experience was how to gathering requirement from real clients
- Since Search PHP programing codes, bootstrap codes it will help to improve my programing language knowledge
- In database design phase work with database it will improve my database knowledge and how to design a database

---

## References

- [1] Vehicle Fleet manager 4, Fleet Management System  
<https://www.vinitysoft.com/fleet-management-software-4-0/>
- [2] Fleetio Fleet Management System  
<https://www.fleetio.com/>
- [3] Wikipedia, Functional Requirement  
[https://en.wikipedia.org/wiki/Functional\\_requirement](https://en.wikipedia.org/wiki/Functional_requirement)
- [4] Wikipedia, Rational Unified Process  
[https://en.wikibooks.org/wiki/RUP\\_-\\_IBM\\_Rational\\_Unified\\_Process/Phases](https://en.wikibooks.org/wiki/RUP_-_IBM_Rational_Unified_Process/Phases)
- [5] System Design  
[http://oer.nios.ac.in/wiki/index.php/Phases\\_of\\_System\\_Development\\_Life\\_Cycle](http://oer.nios.ac.in/wiki/index.php/Phases_of_System_Development_Life_Cycle)
- [6] MCV Architecture  
[https://www.tutorialspoint.com/struts\\_2/basic\\_mvc\\_architecture.htm](https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm)
- [7] Data base design,  
[\\_https://www.guru99.com/database-design.html](https://www.guru99.com/database-design.html)
- [8] Bootstrap, [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
- [9] Data Table, <https://datatables.net/>
- [10] Date Picker, <https://github.com/xdan/datetimestpicker>
- [11] Fusion Charts, <http://jsfiddle.net/fusioncharts/RwGKa/>
- [12] Wikipedia, Evaluation <https://en.wikipedia.org/wiki/Evaluation>

---

# Appendix A – System Documentation

## System Manual

1<sup>st</sup> step:

### Prepare server

- Hardware Requirement
- Software Requirement

### Hardware Requirement

Hardware	Requirement
Processor	Intel (R) Core i3 3.4 GHz
Memory	8GB
Hard disk	500GB
Network Card	1Gb p/s
Display	1366*768 Resolution Monitor
Printer	Hp Deskjet Printer
Internet connection	ADSL Connection

*Table A.1 hardware requirement*

### Software Requirement

Software Name	Requirements
Operating System	Windows7, Windows 8.1, Windows 10 or higher versions
XAMPP	XAMPP version 3.2.1 or Separately install Apache Versions 2.4 or higher Mysql Version 5.6.26 or higher PHP Versions 5.6.12 or Higher
Web Browser	Google chrome Versions 61.0 (recommended) Firefox Versions 55.0 Microsoft edge

*Table A.2 software requirement*

---

## **2<sup>nd</sup> Step**

### **Install Xampp**

- ✓ Download xampp latest version for windows (refer Table A.2 for the Minimum Version) from below Link <https://www.apachefriends.org/download.html>
- ✓ Install xampp. Change installation Path to “D:\xampp”
- ✓ After installation complete, Verify the XAMPP is running, go to the “D:\xampp\” and open the control panel and verify whether Apache and MySQL are started.

### **Install Web Browser**

Install web Browser (refer Table A.2 for the Version and Recommended Browsers).

## **3<sup>rd</sup> Step**

### **Extract System files to hdocs directory**

- ✓ Open Project CD and copy FleetManagementSystem folder and Past it to the directory Path “ D:\xampp\htdocs

## **4<sup>th</sup> Step**

### **Database Installation**

- ✓ Open the web browser and type the URL <http://localhost/phpmyadmin/>
- ✓ Create empty database by providing name as “**fleetmanagement**” and navigate to the “**Import**” tab and click “**choose file**” button. Then browse the CD and select the “**fleetmanagement.sql**” file by opening Database folder.
- ✓ Then Press “**GO**” button located in the bottom of the page.

## **Final Step**

- ✓ Open the installed web browser and type the URL <http://localhost/fleetmanagementsystem/view/login.php> and press “Enter” button to access the system.

# Appendix B – Design Documentation

## Use case Diagram with Detailed information

- Use case diagram for Request vehicle

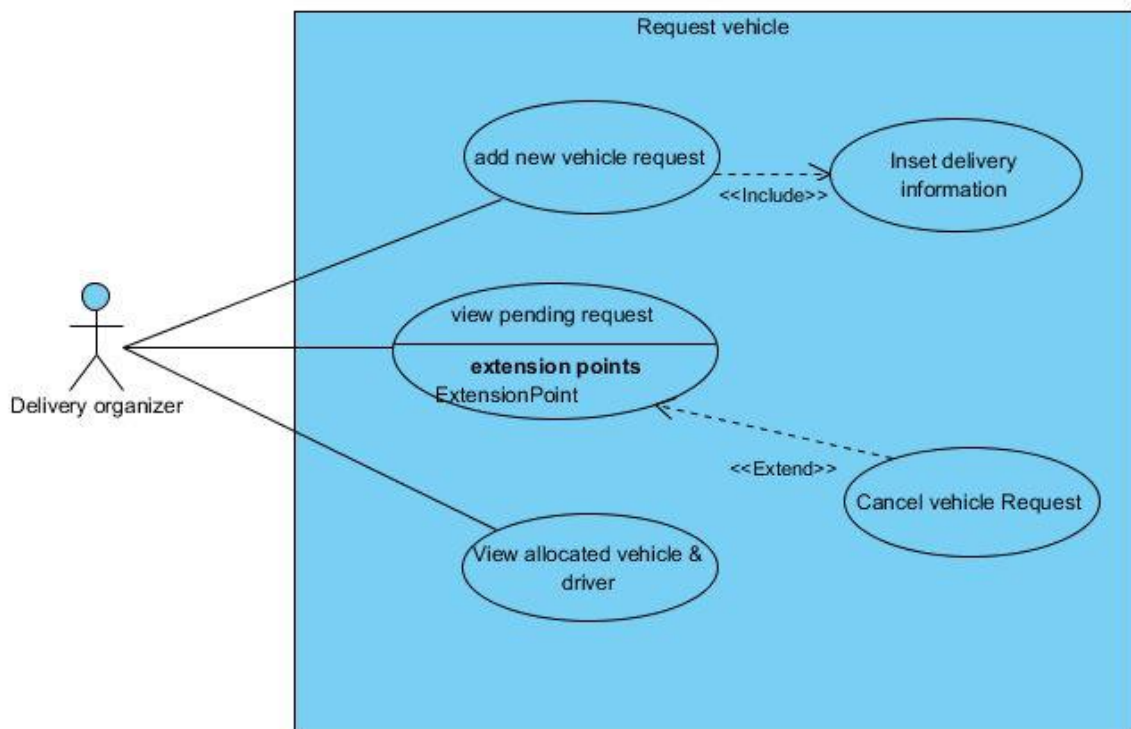


Figure B.1 : Use case diagram for vehicle request

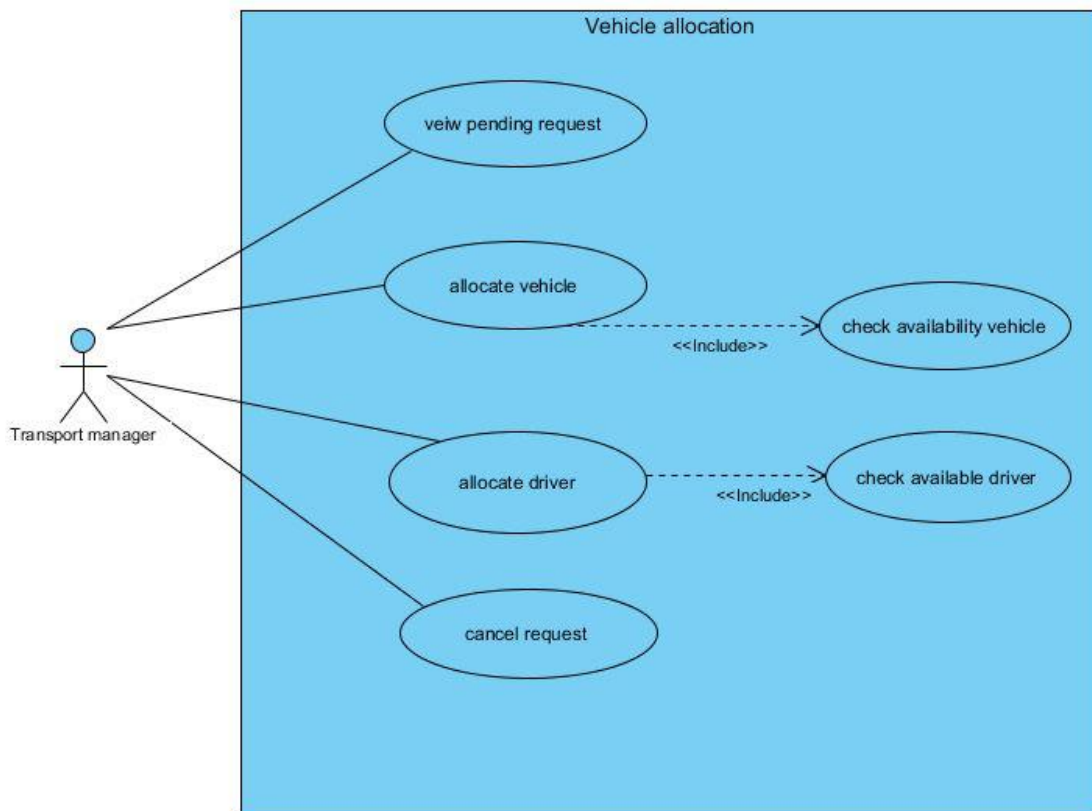
- Use case description for request vehicle

Use case name	Request vehicle	
Actors	Delivery organizer	
Description	User should be able to request vehicle based on his requirement	
Pre-condition	User should be logged into the system	
Typical course of events	Actor action	System action
	1. Select delivery type	

	2. Enter place to visit	
	3. Enter Good weight	
	4. Enter average km	
	5. Select from date & time	
	6. Select return date	
	7. Click submit button	
		8. Display successful message
Alternative courses	8. system displays the error message	
Conclusion	This use case emphasize when vehicle request process completed data are successfully stored in the database	
Post conditions	Vehicle request data save in the database	

*Table B.1: use case description for request vehicle*

- **use case diagram for vehicle allocation model**



*Figure B.2 : Use case diagram for Allocate Vehicle*

- **Use case description for vehicle Allocation**

Use case name	Vehicle Allocation	
Actors	Transport Manager	
Description	User should be able to allocate vehicle based on vehicle request	
Pre-condition	User should be logged into the system Users was already request vehicle (pending request)	
Typical course of events	Actor action	System action
		1. System will display pending vehicle request
	2. Select a relevant request	
	3. Click allocate button	
		4. Display allocation form
		5. Check available vehicle and Display available
	6. Select vehicle	
		7. Display available driver
	8. Select driver	
	9. Click allocate button	
		10. Display successfully allocated
Alternative courses	5. If there are not available vehicles system display message 7. If there are not available Driver system display message	
Conclusion	This use case emphasize when vehicle allocation process completed data are successfully stored in the database	
Post conditions	Vehicle allocation data save in the database	

*Table B.2: Use case description of vehicle allocation*

- use case diagram for close delivery model

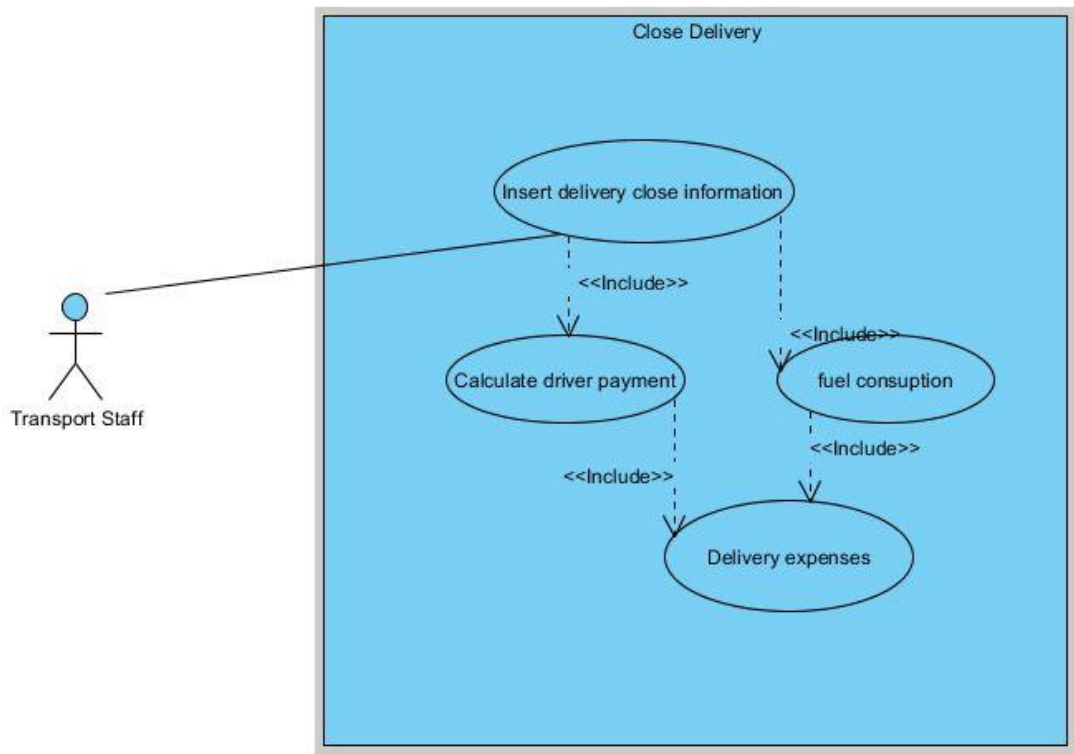


Figure B.3: Use case diagram for Close delivery model

- Use case description for Close delivery model

Use case name	Close Delivery	
Actors	Transport department staff	
Description	After delivery vehicle return to company User should be able to close delivery	
Pre-condition	User should be logged into the system Already vehicle on delivery or there is a vehicle allocate for delivery	
Typical course of events	Actor action	System action
	1. Select appropriate delivery & click close button	
		2.popup delivery close form
	3.enter vehicle start & end meter (km)	
	4.enter delivery start &	



	end date & time	
	5. enter fuel consumption	
	6. click submit button	
		7. validate entered information
		10. Calculate driver payment and insert to delivery expenses
		11. Calculate fuel consumption, fuel cost of delivery and insert into delivery expenses
		12. Display successful message
Alternative courses	7. if user does not enter All mandatory require field display error message 6. if user press cancel button “close delivery” form exit	
Conclusion	This use case emphasize when vehicle request process completed data are successfully stored in the database	
Post conditions	Processed data should properly save in the database for future assessments and for reports	

Table B.3: Use case description for close delivery

- use case description for insert delivery expenses

Use case name	Insert delivery expenses	
Actors	Transport staff	
Description	Staff enter delivery expenses to the system	
Pre-condition	User should be login to the system Transport staff should be close the particular delivery	
Typical course of events	Actor Action	System Action
		1. System will display expenses to be closed
	2. Select Particular delivery and click “close expenses” button	
		3. Display expenses form
	4. Select expenses type and enter amount	

	5. click “submit” button	
		6. System will display Successfully inserted message
Alternative courses	6.System will display error message	
Conclusion	This use case emphasize insertion of the delivery expenses	
Post condition	The data are save in database Selected delivery remove from expenses to be closed table	

Table B.4 : use case description for insert delivery expenses

- Use case diagram for user management

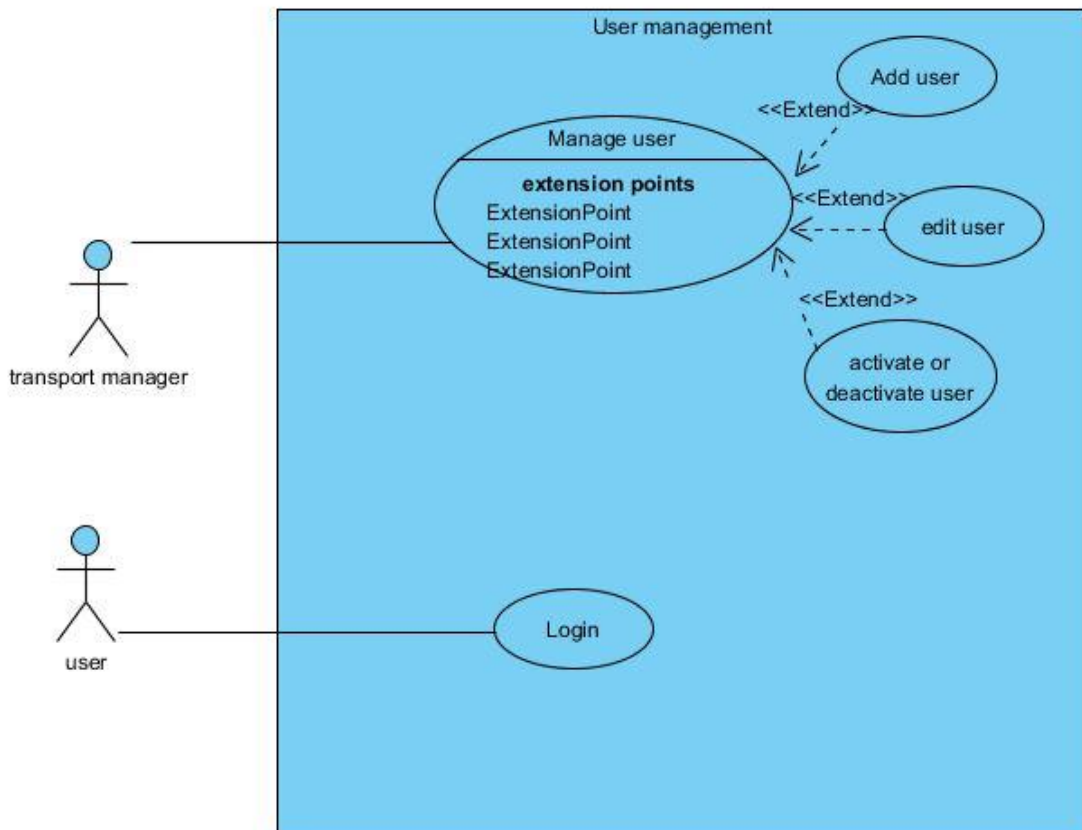


Figure B.4: use case diagram for User management

- use case description for Add User

Use case name	Add user	
Actors	Transport manager	
Description	Add user to the system	
Pre-condition	User should be login to the system	
Typical course of events	Actor Action	System Action
	1. User should <i>click “Add user” button in user management module page</i>	
		1. Display user registration form
	2. Enter user information and <i>click register button</i>	
		3. Display “add user successfully”
Alternative courses	3.System will display error message	
Conclusion	This use case emphasize insertion of the add user	
Post condition	The data are save in database	

Table B:5 Use case diagram for add user

---

# Appendix C- USER DOCUMENTATION

- **Login page**

Open web browser and type following URL <http://localhost/fleetmanagementsystem/view/login.php> and navigate. After navigation user can find Login page that is under figure C.1. after entering user name and password it will directs user to home page which under C.2. it can vary from according to user role.



*Figure C.1 : System Login Page*

Figure C.2 shows the main menu item of the main form.

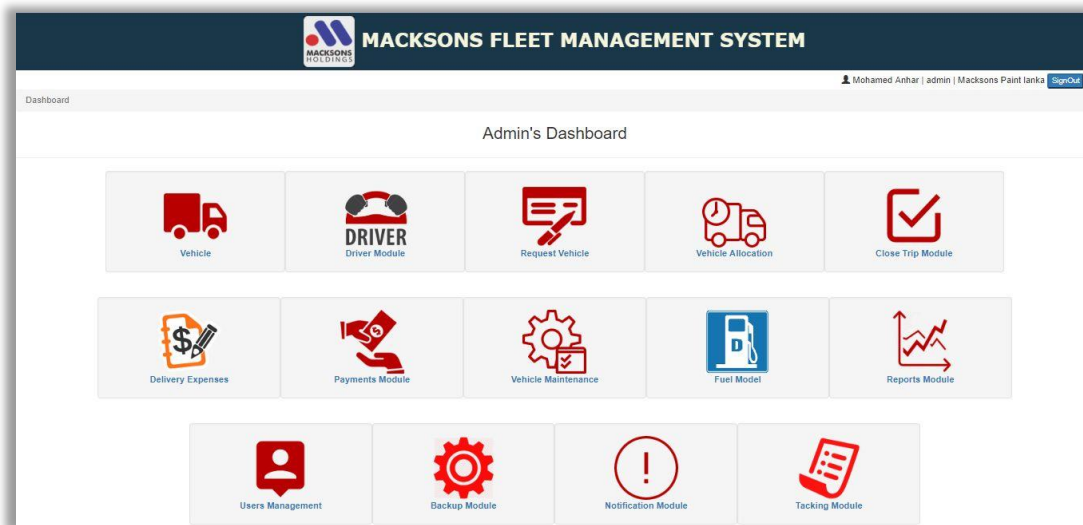


Figure C.2 Administrator Home Page

- **Pages Navigation**

Figure C.3 shows the Page navigation.



Figure C.3 Navigation Bar

- **Logout from the system**

Figure C.4 shows the Sign-out.



Figure C.4 Sign out

- **Data Manipulation in table**

This is a common operation which comes with more pages. Please see the figure C.5 to get an idea about it and the arrow heads under the figure.

(Note-: This facility is met in many pages of the system and this introduction will help to get an idea.)

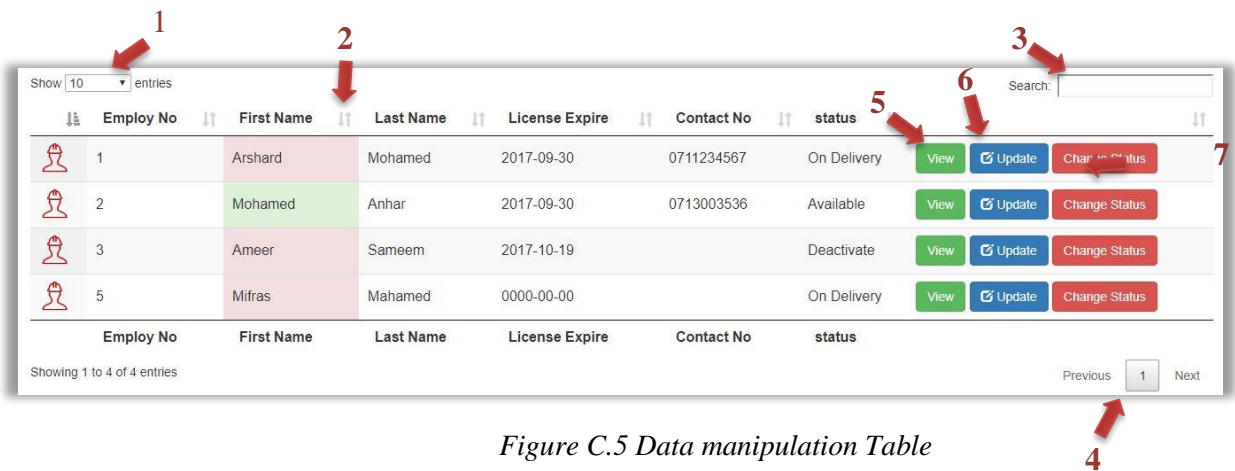


Figure C.5 Data manipulation Table

**1.Show Number of Entries -:** This option helps to manage how many records should appear at a time. If there are more records than select no of value, user can navigate to next set of items using showing figure C.6 number [4] next button.

**2.Order of the Record -:** user can be able to view data ascending and descending order by clicking Figure C.6 number [2] button.

**3.Search Box -:** user can be able to search data by typing keyword here.

**4.Pagination -:** Navigate to table pages.

**5.View Detail -:** User can be able to view Additional information belonging to a particular record.

**6.Update -:** User can be able to Change information belonging to a particular record.

**7.Change Status-** User can be able to change status of data eg. deactivate, activate.


- **Vehicle Page**

Vehicle Module is another Major system module that can be seen on user home page. Figure C.6. This Module should display only authorized user. The rest of the page is explained by the following topics.



Figure C.6 Vehicle Model

- **Add Vehicle**

Add new vehicle into system user should *click* “**Add Vehicle**”  button. All the vehicle details can be inserted via vehicle Registration. Figure shows below C.7. All the saved data displayed in the table. If user entered valid data and *click* “**register Button**” system will be displayed message as **vehicle registered successfully**. If user entered invalid data the system will be displayed the error message. All mandatory fields are identified as star mark “\*”.

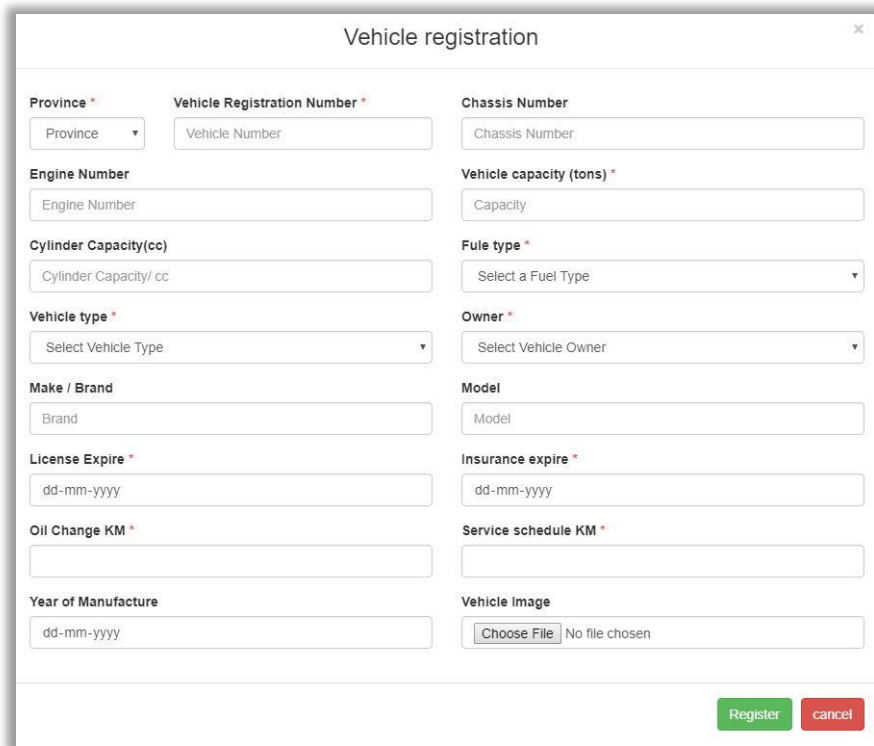


Figure TableC.7 Vehicle Registration Form

- **Vehicle Data Manipulation Table**

User can be able to View, updates or Change status of the Vehicles via figure shows below C.8 table. If user entered invalid data the system displayed the error message. User can be searched vehicle information by Registration no, Vehicle Type, Weight, License Expire date, Insurance Expire date and vehicle status. Further system provides vehicle data view to user in order of ascending and descending method.

Registration No	Vehicle Type	weight (Ton)	License Expire	Insurance Expire	status
LG - 8684	Lorry	16.5	2017-12-30	2017-12-06	available
LG-0534	Lorry	0	2017-12-30	2018-03-03	available
LH - 1548	Lorry	18.5	2017-12-30	2018-01-03	available
LH - 1783	Lorry	18.5	2017-12-30	2018-02-22	available
LH - 1824	Lorry	18.5	2017-12-30	2018-01-10	available
LH - 2001	Lorry	20	2017-12-30	2018-04-06	available
LH - 2008	Lorry	20	2017-12-30	2018-02-03	available
LH - 6273	Lorry	20	2017-12-30	2018-01-29	available
LH - 6367	Lorry	20	2017-12-30	2018-05-01	available
LH - 6368	Lorry	20	2017-12-30	2018-03-19	available

Figure C.8 Vehicle Data Manipulation Table

- **View Vehicle Details**

User can be viewed more details about the particular vehicle. Click “view” button. Figure shows C.8. & Figure C.9.

Figure C.9 View Vehicle details



- **Update Vehicle details**

User can be able to edit vehicle Information via *click* “**update**” button. Figure shows in C.8. Once it clicked vehicle update form will appear. Figure shows in C.10.

The screenshot shows a web form titled "Update Vehicle No :-LH-1548". The form is organized into two columns. The left column contains fields for Province (WP), Vehicle Registration Number (LH-1548), Engine Number, Cylinder Capacity(cc), Vehicle type (Lorry), Make / Brand (Tata), License Expire (30-12-2017), Oil Change KM (6500), and Year of Manufacture (dd-mm-yyyy). The right column contains fields for Chassis Number, Vehicle capacity (tons) (18.5), Fuel type (Diesel), Owner (Company), Model, Insurance expire (03-01-2018), and Service schedule KM (6500). At the bottom, there is a "Vehicle Image" field with a "Choose File" button and "No file chosen" text. A red car icon is displayed below the image field. At the bottom right, there are "Update" and "Cancel" buttons.

*Figure C.10 Update vehicle form*

### **Change Vehicle Status**

User should be able to change vehicle Status via *click* “**Change status**” button. Figure shows in C.8. Once it clicked change status form will be appeared Figure slows in C.11. User can be able to change vehicle status as Available, Maintenance, and Discontinued.

The screenshot shows a web form titled "Change LH-1548 Status". The form is simple and contains a "Vehicle Registration Number" field with the value "LH-1548". Below it is a "Change Vehicle Status" dropdown menu with "available" selected. At the bottom, there are "Update" and "Cancel" buttons.

*Figure C.11 Vehicle Status Change Form*

- **Driver Page**

Driver Module is another Major system module that can be seen on user home page. Figure C.12. This Module should display only authorized user. This option has appeared by default to transport manager and transport staff role. The rest of the page is explained by following topics.

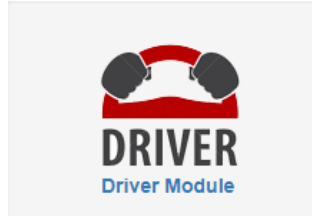


Figure C.12 Driver Module icon

- **Driver Registration Form**

Add new driver into system user should *click* “**Add Driver**” button. All driver details can be inserted via Driver Registration form shows below Figure C.13. All the saved data displayed in the table. If user entered valid data and *click* “**Register**” Button then system will be displayed message as **driver registered successfully**. If user entered invalid data the system will be displayed the error message. All mandatory fields are identified as star mark “\*”.

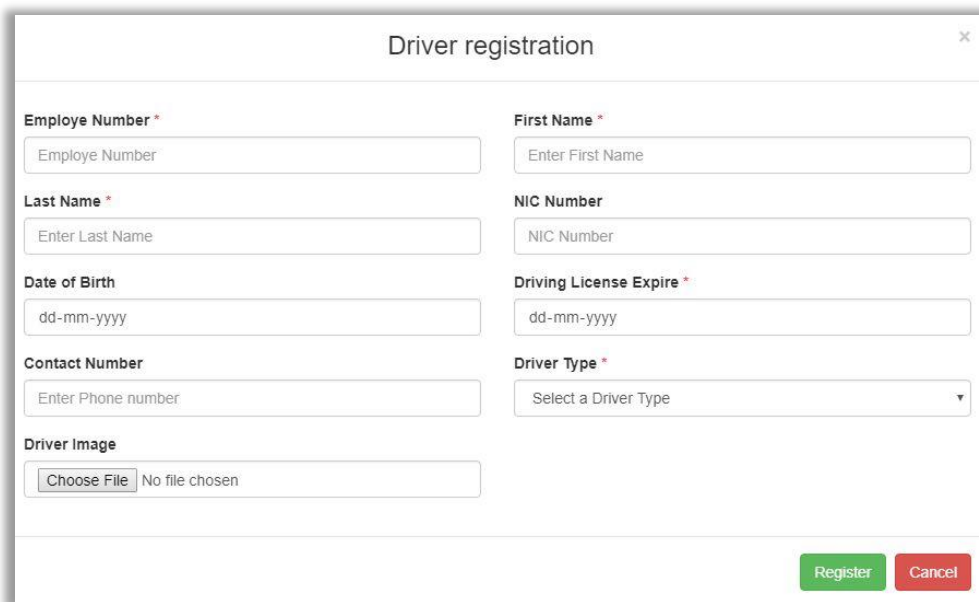
The screenshot shows a web form titled "Driver registration" with a close button (X) in the top right corner. The form is organized into two columns. The left column contains: "Employee Number \*" with a text input field containing "Employee Number"; "Last Name \*" with a text input field containing "Enter Last Name"; "Date of Birth" with a date input field containing "dd-mm-yyyy"; "Contact Number" with a text input field containing "Enter Phone number"; and "Driver Image" with a "Choose File" button and the text "No file chosen". The right column contains: "First Name \*" with a text input field containing "Enter First Name"; "NIC Number" with a text input field containing "NIC Number"; "Driving License Expire \*" with a date input field containing "dd-mm-yyyy"; and "Driver Type \*" with a dropdown menu showing "Select a Driver Type". At the bottom right of the form are two buttons: a green "Register" button and a red "Cancel" button.

Figure C.13 Driver Registration form

- **Vehicle Request model**

Request vehicle Module is another Major system module that can be seen on user home page. Figure shows in C.14. Delivery organizer should be requested their vehicle request via this module. This Module should be displayed to authorized user only. The rest of the page is explained by the following topics.

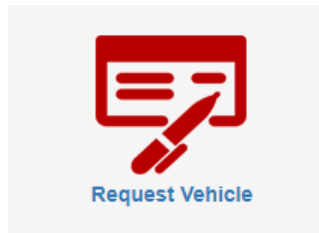


Figure C.14 request Vehicle Model icon

- **Vehicle request Form**

Add new Vehicle Request via system the user should *click* “**New Request**” button. All the vehicle request details should be inserted via Vehicle request form. Figure shows in C.15. If user entered valid data and click the “**register**” Button system will be displayed message as “**Vehicle Request Successfully Sent** “. If user entered invalid data the system will be displayed the error message. All mandatory fields are identified as star mark “\*”.

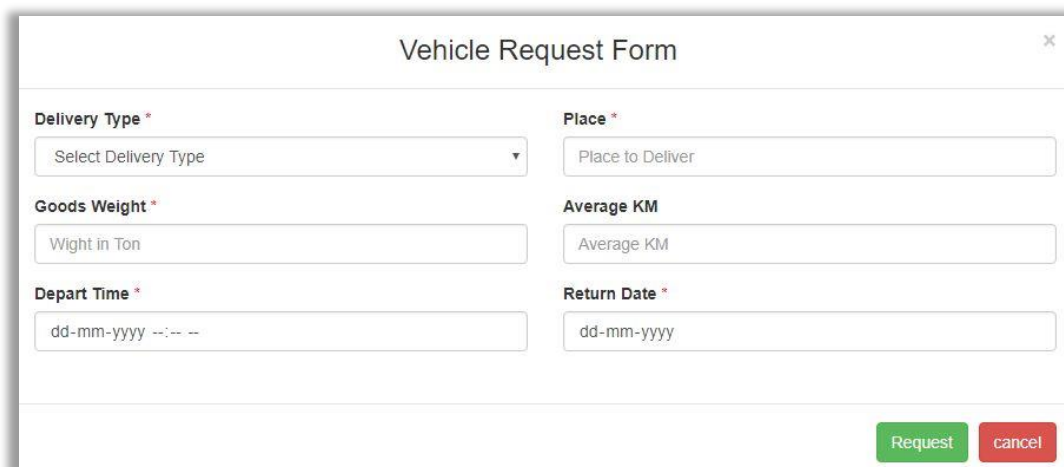
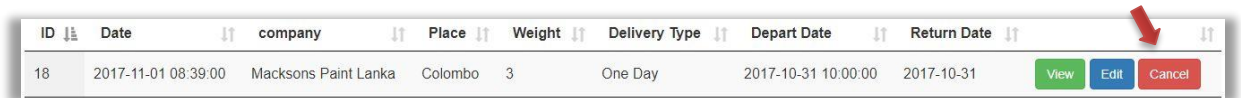
A screenshot of a web form titled "Vehicle Request Form". The form contains several input fields: "Delivery Type \*" (a dropdown menu with "Select Delivery Type"), "Place \*" (a text input field with "Place to Deliver"), "Goods Weight \*" (a text input field with "Wight in Ton"), "Average KM" (a text input field with "Average KM"), "Depart Time \*" (a date and time input field with "dd-mm-yyyy --:-- --"), and "Return Date \*" (a date input field with "dd-mm-yyyy"). At the bottom right, there are two buttons: a green "Request" button and a red "cancel" button.

Figure C.15 Vehicle Request form

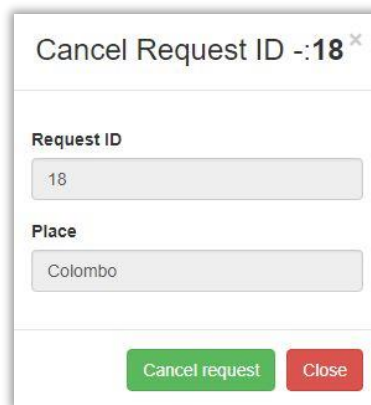
- **Cancel Vehicle Request**

Delivery organizer should be able to cancel his vehicle request by click “cancel” button (Figure shown below C.16) on particular request. After clicking cancel button system will popup confirmation message (Figure shows in C.17). Once it clicked “cancel request” request can be saved otherwise clicked “close” button request will be rejected.



ID	Date	company	Place	Weight	Delivery Type	Depart Date	Return Date	
18	2017-11-01 08:39:00	Macksons Paint Lanka	Colombo	3	One Day	2017-10-31 10:00:00	2017-10-31	<a href="#">View</a> <a href="#">Edit</a> <a href="#">Cancel</a>

Figure C.16 Cancel Request



Cancel Request ID -:18

Request ID  
18

Place  
Colombo

[Cancel request](#) [Close](#)

Figure C.17 confirmation message for vehicle request

- **Vehicle Allocation Module**

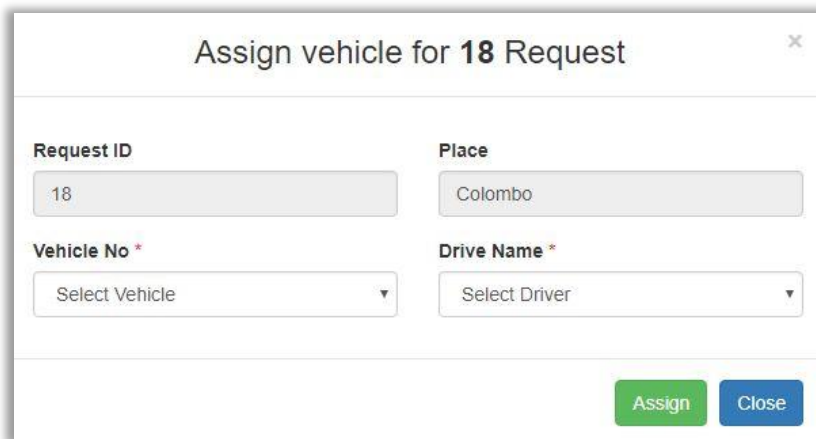
Vehicle allocation Module is another Major system module that can be seen on user home page as C.18. This Module should be displayed to authorized user only. The rest of the page is explained by the following topics.



Figure C.18 Vehicle Allocation module icon

- **New vehicle Allocation**

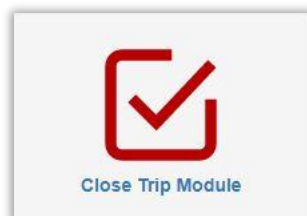
To Allocate Vehicle, user should Navigate following Path “**Vehicle Allocation**→**schedule Vehicle**” System will be displayed all vehicle requests. Select particular request and click “**Assign**” Button system will be displayed vehicle allocation form shows below figure C.19. User should be selected to vehicle no and driver name then *click* “**Assign**” button to save the transaction.



*Figure C.19 Vehicle Allocation Form*

- **Close Delivery**

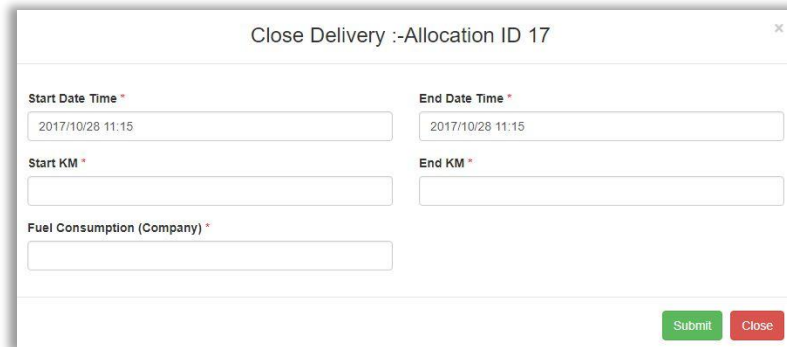
Close delivery Module is another Major system module that can be seen on user home page as C.20. After complete the delivery; transport staff should be closed the particular delivery. This Module should be displayed to authorized user only. The rest of the page is explained by the following topics.



*Figure C.20 Close Delivery icon*

- **Close Delivery Form**

System will be displayed deliveries; Transport staff should be selected particular delivery and *click* “**Close Delivery**” Button then system will be displayed close delivery form as showing below figure C.21. All mandatory fields are identified as star mark “\*”.



*Figure C.21 Close Delivery form*

- **Delivery Expenses**

Delivery expenses Module is another system module that can be seen on user home page as C.22. After **closing delivery** transport staff should be entered particular delivery expenses via *click* “**Delivery Expenses**” Icon (Figure shows in C.22) then system will be displayed list of “**Expenses to be closed**” once it selects the particular delivery and *click* “**close expense**” button then system will be displayed “**Delivery Expenses Form**” once it entered the expenses *click* “**submit**” button. This Module should be displayed to authorized user only.



*Figure C.22 Deliver Expenses*

Delivery Expenses Form			
Expense	Detail	Quantity	Amount
Select Expense ▾	<input type="text"/>	<input type="text"/>	<input type="text"/>
Total			<input type="text"/>

*Figure C.23 Close delivery Expense Form*

- **Vehicle Maintenance**

Vehicle Maintenance Module is another system module that can be seen on user home page as C.24. All maintenance expenses should be entered in this module such as repair, service, oil change, license renewal, insurance renewal etc. and also user can be able to previewed vehicle maintenance details. This Module should be displayed to authorized user only. The rest of the page is explained by the following topics.



*Figure C.24 Vehicle maintenance icon*

- **Add Vehicle maintenance form**

Add vehicle to maintenance, user should click “**Add Maintenance**” Button. Then system should be displayed Add maintenance form as bellows in figure C.25.

Figure C.25 Add Vehicle Maintenance Form

- **Close vehicle maintenance**

System will be displayed list of maintenance vehicles to be closed. Figure shows in C.26. *click* **“Close Maintenance”** Button then system will be displayed maintenance expenses form then enter maintenance expenses for particular vehicle and **“submit”** to save the transaction figure shows in C.27.

Show 10 entries

Search:

Main ID	Date	Vehicle	Maintenance Type	Garage Name	KM	
4	2017-10-06	Cv2345	Repair	AB Motors	20	<input type="button" value="EDIT"/> <input type="button" value="Close Maintenance"/> <input type="button" value="Cancel"/>

Figure C.26 Close Maintenance

Return Date

Expense Detail	Detail Description	Amount
Repair	<input type="text"/>	<input type="text"/>
Total		<input type="text"/>

+ -

Figure C.27 vehicle maintenance expenses form



- **Fuel Management**

Fuel Module is another Major system module that can be seen on user home page. Figure C.28 Company has own fuel station therefore vehicle can be fill up fuel before the delivery. This Module should display if user has appropriate privileges. By default, transport staff role should authorize to this page. The rest of the page is explained by the following topics.



Figure C.28 Fuel Module icon

- **Fuel Management Dashboard**

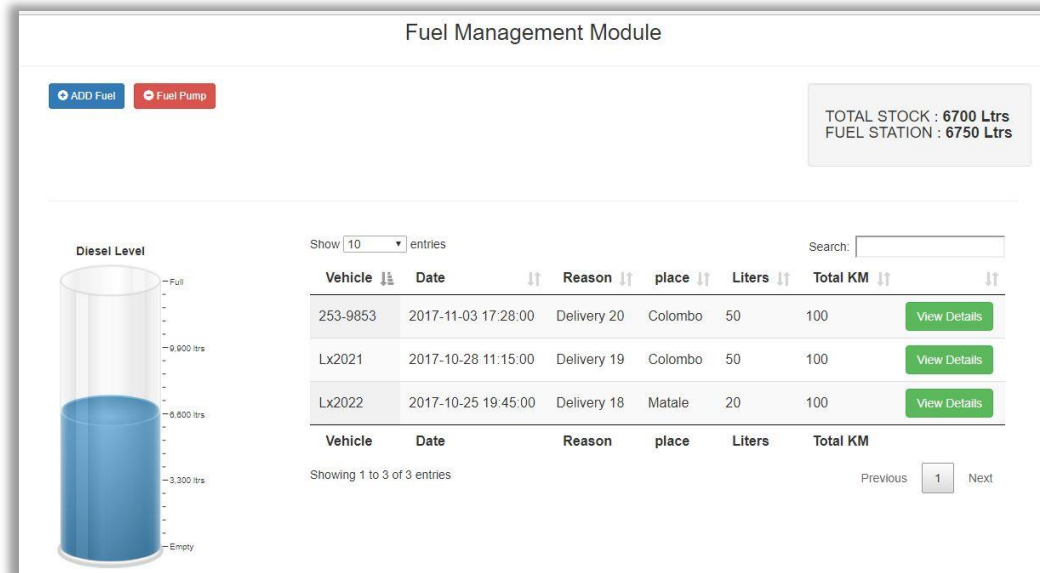
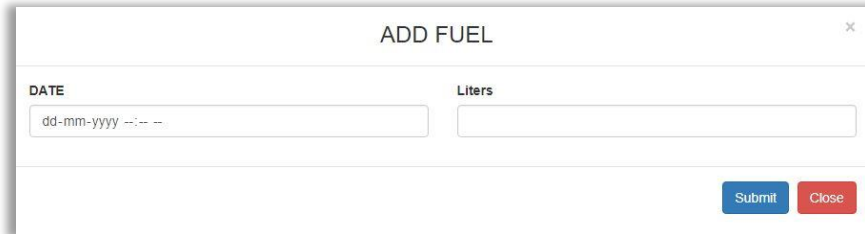


Figure C.29 fuel Dashboard

- **Fuel Pumping to Fuel station**

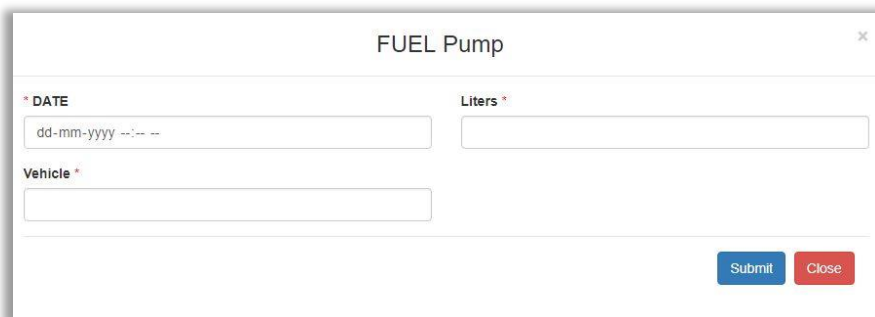
User should *click* “**ADD fuel**” button then system will be displayed Add fuel form. Figure C.30



*Figure C.30 Fuel Insert Form*

- **Pump fuel to delivery vehicle**

User should *click* “**fuel pump**” button (figure shown C.29) then system will be displayed fuel pump form. Figure C.31



*Figure C.31 Fuel pump form*

- **User Management**

User Management Module is another Major system module that can be seen on user home page as C.32.this module helps to create user login's for system user.



*Figure C.32 user management icon*

- **Create user form**

Add new user to system user should click “**Add user**” button. Figure shows below C.33. All mandatory fields are identified as star mark “\*”.

The screenshot shows a web form titled "Add a User" with a close button (X) in the top right corner. The form is organized into two columns. The left column contains: "Employe Number" (text input), "Last Name \*" (text input), "Role \*" (dropdown menu with "Select a Role" selected), "Email Address" (text input with "Email ID" placeholder), and "User Name \*" (text input). The right column contains: "First Name \*" (text input), "Company \*" (dropdown menu with "Select Company Name" selected), "Tel Number" (text input with "Tel No" placeholder), and "User Image" (file upload area with a "Choose File" button and "No file chosen" text). At the bottom right of the form, there are two buttons: a green "Register" button and a red "Cancel" button.

Figure C.33 Add user form

# Appendix D-MANAGEMENT REPORTS

## Delivery expenses report

System should be provided company wise delivery expenses report as following categories.

- Weekly Details and Summary delivery expenses report
- Monthly Details and Summary delivery expenses report
- Yearly Details and Summary expenses report

Figure D.1 show Last 7 days Company Wise Delivery Expenses Report.

Macksons Paint Lanka (Pvt) Ltd.  
78, Diggala Road Keselwatte, Panadura Sri Lanka, Sri Lanka.  
Telephone: +94-11-5530225, 5531000, 5523323  
Fax: +94-38-4281036  
Email: inquiry@multilac.com

Last 7 Days Delivery Expenses Report

From Date : 30-10-17 To Date : 06-11-17

Company	Amount
Macksons Paint lanka	6,900.00
shimmzi Plastic	10,400.00
<b>Total</b>	<b>17,300.00</b>

Figure D.1 Last 7days delivery expenses summery

Figure D.2 show customize Delivery Expenses Summery Report.

Customize Delivery Expenses Report

Delivery Type:  Company:

Date From:  To Date:

Company Name:   
Delivery Type:   
From Date: 2017/09/01 17:19 To Date: 2017/11/06 17:19

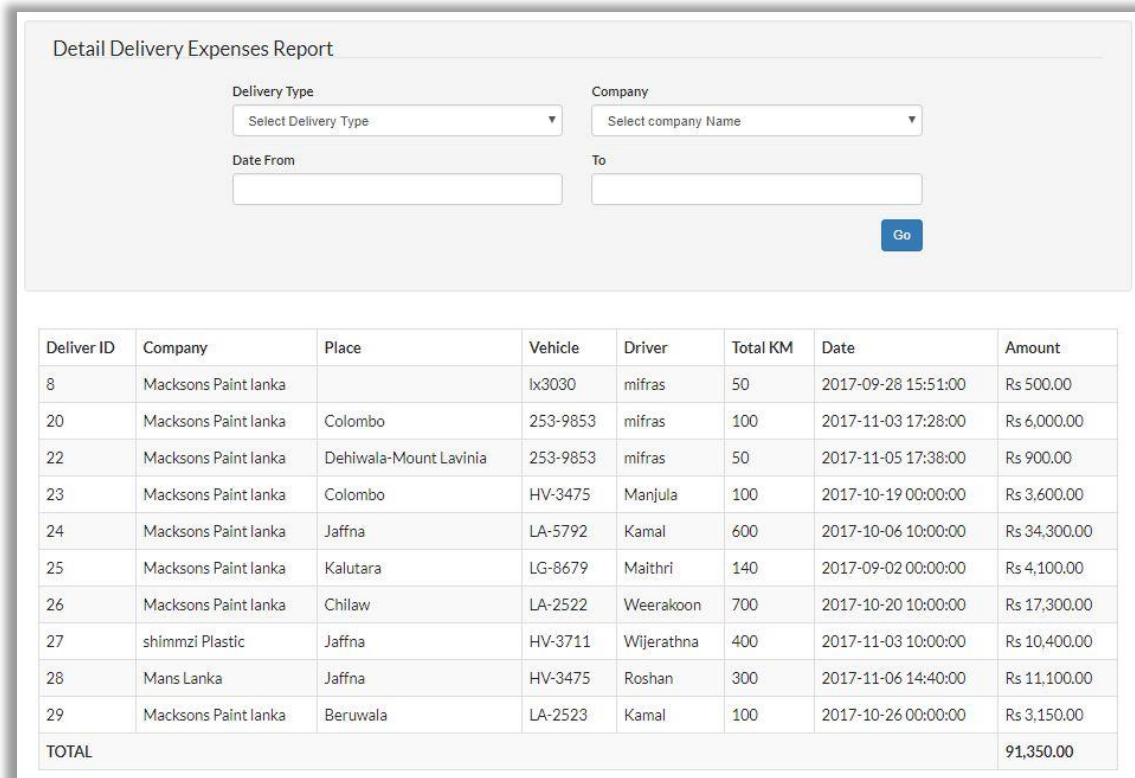
Search:

Company Name	Amount
Macksons Paint lanka	89,850.00
Mans Lanka	19,000.00
shimmzi Plastic	10,400.00
<b>TOTAL</b>	<b>119,250.00</b>

Showing 1 to 4 of 4 entries  1

Figure D.2 Customize Delivery expenses summery report

- Figure D.3 show Detail Delivery Expenses Report.



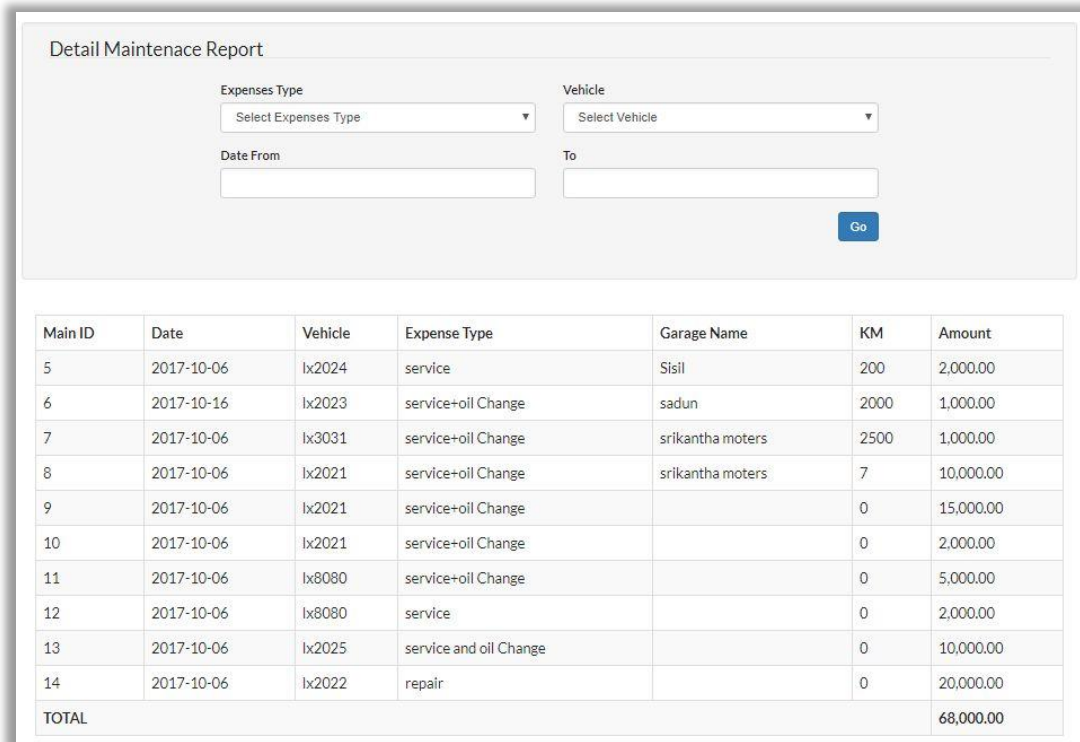
*Figure D.3 Detail Delivery Expenses Report.*

## Maintenance expenses report

System should be provided vehicle maintenance expenses report as following categories.

- Weekly Details and Summary delivery expenses report
- Monthly Details and Summary delivery expenses report
- Yearly Details and Summary expenses report

Figure D.4 show Detail Maintenance Expenses Report.



*Figure D.4 Detail Maintenance Expenses Report.*

# Appendix E – TEST RESULTS

- **Test Cases for User Authentication**

The test cases with results belong to the user Authentication Module is displayed by Table E.1

Test Number	Step to Test	Expected Result	Status
1.	Enter correct username Enter correct password Active user status	Successfully login to the system  Mohamed Anhar   admin   Macksons Paint lanka	Pass
2.	Empty username Empty password	Display error message “oops! Empty User name and Password”  oops! Empty User name and Password	Pass
3.	Empty username Enter password	Display error message “oops! Empty User name”  oops! Empty User Name	Pass
4.	Enter correct username Empty password	Display error message “oops! Empty Password”  oops! Empty Password	Pass
5.	Enter incorrect username Enter correct password Active user status	Display error message “Invalid Username or password”  Invalid Username or password	Pass
6.	Enter correct username Enter incorrect password Active user status	Display error message “Invalid Username or password”  Invalid Username or password	Pass
7.	Enter correct username Enter correct password deactive user status	Display error message “Invalid Username or password”  Invalid Username or password	Pass

Table E.1 : Test results for User Authentication

## Test case for request vehicle

The test cases with results belong to the Vehicle request module is displayed by Table E.2

Test Number	Step to Test	Expected Result	Status
1.	Add Information with relevant details	Display Message “vehicle Request Sent Successfully”  Vehicle Request Successfully Sent	Pass
2.	Try to add vehicle request by keeping all fields in empty	Display Message “Select Delivery Type” and focus the field <b>Select Delivery Type</b>	Pass
3.	Enter all other field except “delivery Type” field	Display Message “Select Delivery Type” and focus the field  <b>Select Delivery Type</b>	Pass
4.	Enter all other field except “Place” field	Display Message “Enter Place” and focus the field  <b>Empty Place</b>	Pass
5.	Enter all other field except “Goods weight” field	Display Message “Enter weight” and focus the field  <b>Empty Weight</b>	Pass
6.	Enter all other field except “Depart date and Time” field	Display Message “select Depart date and Time” and focus the field  <b>Select Depart Date &amp; Time</b>	Pass
7.	Enter all other field except “Return date” field	Display Message “select Return date” and focus the field  <b>Select Return Date</b>	Pass
8.	Select the request and click “edit” button and change information and click update	Display Message “Update successfully” and redirect to vehicle request page  Update Request Successfully	Pass
9.	Select the request and click “cancel” button	Popup the Conformation dialog box “Are you sure to cancel the request ID”	



			Pass
--	--	--	------

Table E.2 test results for vehicle request

## Test case for vehicle allocation

The test cases with results belong to the Vehicle allocation module is displayed by Table E.3

Test Number	Step to Test	Expected Result	Status
1.	Add Information with relevant details	Display Message “Vehicle allocated Successfully”  Vehicle allocated Successfully	Pass
2.	Try to add information by keeping all fields in empty	Display Error Message and focus First Mandatory field”  <b>Select the Vehicle</b>	Pass
3.	Select all other field except “Vehicle No” field	Display Message “ Select the Vehicle and focus the field”  <b>Select the Vehicle</b>	Pass
4.	Select all other field except “Driver Name” field	Display Message “ Select the Driver Name and focus the field”  <b>Select the Driver</b>	Pass
5.	Select the Allocation and click “cancel” button	Popup the Conformation dialog box “Are you sure to cancel the location”	Pass

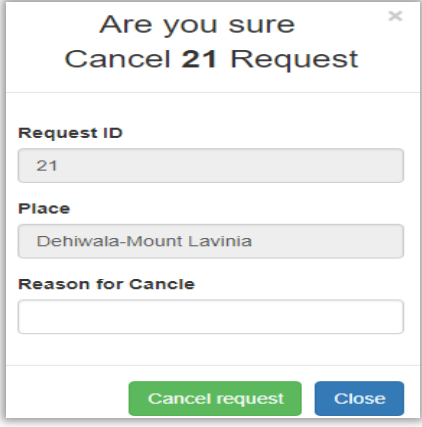
			
--	--	--	--

Table E.3 Test cases for vehicle Allocation

## Test Cases for Close Delivery

The test cases with results belong to the close delivery module is displayed by Table E.4

Test Number	Step to Test	Expected Result	Status
1.	Add Information with relevant details	Display Message “ Delivery Closed Successfully”  Delivery Closed Successfully	Pass
2.	Try to add information by keeping all fields in empty	Display error message Empty first Mandatary field name and focus first field”  <b>Empty Start KM</b>	Pass
3.	End date time and less than Start date time	Display Message “invalid End date” and focus the field  <b>Incorrect End date</b>	Pass
4.	End Km and less than start Km time	Display Message “invalid End Km” and focus the field  <b>Incorrect End KM</b>	Pass
5.	Enter all other field valid information except “Fuel consumption date” field	Display Message “Enter Fuel consumption” and focus the field  <b>Empty Fuel Consumption</b>	Pass

Table E.4 Test cases for close delivery

---

# Appendix F-CODE LISTING

## Vehicle registration Ajax Function

Using Ajax technology, validate whether vehicle already registered or not in system.

Advantage of Ajax Function

- User no needs to fill up entire form. If it is registered, system will be displayed the vehicle no while user type into the relevant cage.

```
<script>
    function checkVehicle(str)
    {
    var xmlhttp;
    if (str=="")
    {
    document.getElementById("display").innerHTML="";
    return;
    }
    if (window.XMLHttpRequest)
    { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
    }
    else
    { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function()
    {
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
```

```

document.getElementById("display").innerHTML+xmlhttp.responseText;
}
}
xmlhttp.open("GET","vehicleValidation.php?vehicle="+str,true);
xmlhttp.send();
}

</script>

+++++++ vehicleValidation.php page+++++++
//call database connection
include '../common/dbconnection.php';
$ob = new dbconnection();
$con = $ob->connection();

//create vehicle model object
include '../model/Vehiclemodel.php';
$vobj=new vehicle();

$vehicle=$_GET['vehicle'];
$pat='/^[A-Z0-9]{2,3}+[-]+[0-9]{4}+$/';
$status=0;

if(preg_match($pat,$vehicle)){
    $numrow=$vobj->validateVehicle($vehicle);
    if($numrow==0){
        echo "<span class='alert-success'>Available to use</span>";
        $status=1;
    }else{
        echo "<span class='alert-danger'>Already registered</span>";
    }
}
}

```

```
}  
?>  
<input type="hidden" name="hid" id="hid" value="<?php echo $status; ?>" />
```

## Vehicle registration Validation

User inputs of the Vehicle Registration Page are validated by the use of this function.  
Use jquery for validation function.

```
$(document).ready(function () {  
    $('#reg').submit(function () {  
        //get user input by id  
        var province=$('#province').val();  
        var reg_no=$('#vehicle_regno').val();  
        var weigh=Math.abs($('#weigh').val());  
        var fuleType=$('#fuleType').val();  
        var vehicle_type=$('#vehicle_type').val();  
        var owner=$('#owner').val();  
        var license=$('#license_expire').val();  
        var insurance=$('#insurance_expire').val();  
        var maxoil=$('#max_oil_usage').val();  
        var servicekm=$('#servicekm').val();  
        //vehicle registration number pattern  
        var vreg=/^[A-Z0-9]{2,3}[-][0-9]{4}/;
```

```

//validate is province empty
if(province==""){
    $('#msg').text("Empty Province Name");
    $('#province').focus();
    return false;
}

//Validate registration number
if(reg_no==""){
    $('#msg').text("Empty Vehicle Registration Number");
    $('#province').focus();
    return false;
}

//validate vehicle registration number pattern
if(!reg_no.match(vreg)){
    $('#msg').text("Invalid Vehicle no");
    $('#vehicle_regno').focus();
    return false;
}

var hid=$('#hid').val();
if(hid==0){
    $('#msg').text("Existing Vehicle");
    $('#vehicle_regno').focus();
    $('#vehicle_regno').select();
    return false;
}

//validate vehicle capacity
if(weigh==""){
    $('#msg').text("Empty Vehicle Capacity");
    $('#weigh').focus();
}

```

```
        return false;
    }
    //validate fuel type
    if(fuleType==""){
        $('#msg').text("Select Fuel type");
        $('#fuleType').focus();
        return false;
    }
    //validate vehicle type field is empty
    if(vehicle_type==""){
        $('#msg').text("Select Vehicle type");
        $('#vehicle_type').focus();
        return false;
    }
    //validate owner field is empty
    if(owner==""){
        $('#msg').text("Select Owner");
        $('#owner').focus();
        return false;
    }
    //validate license field is empty
    if(license==""){
        $('#msg').text("Select License Expire");
        $('#license_expire').focus();
        return false;
    }
    //validate Insurance field is empty
    if(insurance==""){
        $('#msg').text("Select Insurance Expire");
        $('#insurance_expire').focus();
        return false;
    }
}
```




---

```
//validate maximum oil usage field is empty
if(maxoil==""){
    $('#msg').text("Empty Oil Change");
    $('#max_oil_usage').focus();
    return false;
}

//validate max KM for service field is empty
if(servicekm==""){
    $('#msg').text("Empty Service Schedule");
    $('#servicekm').focus();
    return false;
}
});
});
```



# Appendix G - Client Certificate



**MACKSONS PAINTS LANKA (PVT) LTD.**  
P.O. Box 04 Keselwatta, Panadura, Sri Lanka. **MACKSONS HOUSE**  
(+94) 11 5530225 (+94) 11 5550268 No: 443, 6<sup>th</sup> Floor, Galle Road, Colombo 00300,  
Sri Lanka. (+94) 11 5348981 (+94) 11 2553248  
office@macksons.com Factory: No. 78, Diggala Road, Keselwatta,  
www.multilac.com Panadura, Sri Lanka  
Export Division : (+94) 11 5553 575

28<sup>th</sup> October 2017  
Project Examination Board,  
University of Colombo School of Computing,  
No: 221/2A, Dharmapala Mawatha,  
Colombo 07  
Dear Sir/Madam,


**LETTER OF CERTIFICATION**







I would like to inform you that Mr.M.A.M. Anhar (R141611) has productively formed and presented a lucrative system with valuable submissions for our company. Project was undertaken by him as partial fulfillment of a requirement for the Bachelor of Information Technology Degree.

I must appreciate him for using the existing resources effectively arranging and getting support from the dedicated team of our present employees. He had successfully analyzed the situation of over company and had created this excellent system which we were pleased to implement. We strongly believe that this system will help us to improve our business significantly in the future.

I would like to thank Mr. M.A.M.Anhar for his time and effort that he has extended towards the completion of this valuable system.

This certification is issued upon the request of Mr. M.A.M. Anhar

Thank you,  
Yours faithfully  
  
(Mr. Salam Riyad) 28/10/17  
(Group IT Manager - Macksons Holdings)



MULTILAC IS ONE OF THE TOP THREE GLOBAL PAINT COMPANIES TO RECEIVE THE PRESTIGIOUS LEAD SAFETY CERTIFICATION IN CONFORMITY WITH THE STRINGENT SAFETY STANDARDS AS A SAFE PAINT TO USE FROM IPEN AMERICA




Figure G.1 Client Certification

---

# GLOSSARY

**HTML** – Stands for Hyper Text Markup Language. Use to create Web pages.

**AJAX** – Stands for Asynchronous JavaScript and XML. Ajax is a set of Web development techniques using many Web technologies on the client side to create asynchronous Web applications

**Apache** – Open source web server developed by Apache Software foundation.

**CSS** – Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language.

**JavaScript** – Use to dynamic web applications. Develop by Netscape.

**MySQL** – Open source database management system.

**PHP** – Stand for PHP Hypertext Preprocessor. server side scripting language.

**RUP** – Stands for Rational Unified Process. Iterative software development methodology.

**XAMPP** – open source cross-platform web server solution stack package developed by Apache Friends. Stands for X- Cross-Platform, A- Apache, M- MariaDB, P- PHP, P- Perl

---

# Index

## A

Activities -1,2,10,12,45

Allowances – 2

Analysis-3,4,11,13

Alternative -3,13,14,35

Allocation - 3,18,19,51

Architecture -12,14

## B

Business - 4,12,15

## C

Client - 4,11,13,14,44

Company - 1,5,9,70

## D

Develop -1,2,4,6,11,12,13,14,20

Delivery - 1

Driver -1,2,3,5,6,7,8,9,23

Design - 3,5,11,13,14,16,20,21,25,34,35

Drawbacks – 5

Document - 5,13,48

## E

Expenses - 2,3,5,6,8,9,54

Efficiency - 1,2,3

Evaluation - 3,35,44

## F

Future - 3,25,35,45,54

Fuel - 1,5,6,8,31,41

## G

Generic-12

## H

Helps - 6,22,29,71

## I

Insurance - 2,7,9

Important - 1,2,5,23,25

Inventory - 5,6

information-2,3,5,7,8,9,23

## L

License - 2,5,6,7,9

Login - 10,21,27,28

Limitations - 6

## M

Management-1,2,3,5,6

Maintenance-2,5,6,9,20,32

Motivation-1

## N

Notification-2,9

---

  
**P**

Pending - 2,7,8,52

Place - 1,5,7,13,14,39

Payment - 2,5,8,9,54

**R**

Request -2,5,6,7,8,13

Requirement - 2,3,4,6,10,11,25,45

Records - 5,8,59

Report - 1,2,3,5,9,13,45

**S**

Scheduling - 5,6

System - 1,2,3,4

Service - 1,2,5,9,31

Scope – 2

**T**

Transport - 1,3,5,7,6,8

**U**

User – 6

**V**

Vehicle – 1,2,3,4,5,6