Web based Fleet Management System

A dissertation submitted for the Degree of Master of Information Technology

HDN SILVA

University of Colombo School of Computing

2018

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: HDN SILVA

Registration Number: 2015/MIT/054

Index Number: 15550545

_____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr. /Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: Dr MGNAS FERNANDO

_____

Signature: Date:

Abstract

Ceylon Electricity Board (CEB) is the national power utility and is a statutory body established by the Act Number 17 of 1969, with reflecting a monopoly in electricity power sector in Sri Lanka. CEB core business processes include power generation, transmission, distribution and retailing. CEB operates with nearly 6.1 million customers as of 2016 and responsible for provide electricity continuously to the customers without any issues.

At present, the management and monitoring of the vehicles in CEB is done manually. Thus problems such as data duplication, lack of integrity, data inconsistency and missing data has become a major problem. To overcome the above mentioned problems web based system was implemented.

Requirements were gathered referring web resources, onsite observation and conducting interviews with CEB management and staff members. To enhance the more features similar systems were used. Based on the above facts gathering techniques, users of the system were identified. The entire system was divided as Vehicle Registration Management, Vehicle Allocation, Booking Details, Running & Fuel Management, Vehicle Management and Report Generation. This was completely replaced by the current manual system and provided an advanced closed monitoring and proper management of vehicles.

The three-tier architecture with XAMPP, MySQL workbench 6.0 community edition and .NET technology were used to develop this system. The object oriented concept was used to draw use case diagrams, sequence diagrams, class diagrams, etc. Microsoft Visio 2013 were used as the modeling software at the analyzing and designing stage of the project. After preparing a test plan, testing was initiated. Unit test was developed in the implementation for the solution and further solution was done against test cases. Acceptance testing was carried out after implementing the onsite. And questionnaire was given to the selected users in the evaluation.

The aim of the project is to provide a web based Fleet Management System (FMS) to increase efficiency and productivity of the vehicle management activities in a smooth manner in CEB.

Acknowledgements

While taking the opportunity of presenting this study illustrating and implementing the Web based Fleet Management System for Ceylon Electricity Board. I wish to convey a note the assistance given to me various people who came across me during my tenure of completing this study. Therefore it is my unbound duty to offer my appreciation with great pleasure to the following personnel.

Specially to Dr M G N A S Fernando, senior lecturer of the University of Colombo School of Computing, who imparted his valuable knowledge and enabling me to enhance my knowledge and for giving me immeasurable guidance and support to successfully completing this report as my supervisor. I wish to express my gratitude to the director, all academic staff of the University Of Colombo School Of Computing, for the advices given and all the help rendered.

Further thanks also Mr. Kanishka Jayasekara (former DGM/IT), Dr Rohan Abeysekara (DGM/IT) and all the staff members of the Ceylon Electricity Board. Without those help and support I could not have manage and complete this study on time.

My colleagues for sharing their knowledge and views on this problematic area. Last but not least, I would like to convey my gratitude to my parents and my friends for their inspiration and guidance during this period of study.

Table of Contents

List of Figures

Figure No Description Page Number

List of Tables

Table No Description Page Number

List of abbreviations

AGM	Additional General Manager
AGM/AM	Additional General Manager/Asset Management
CC	Chief Clerk
CEB	Ceylon Electricity Board
CEO	Chief Executive Officer
DEO	Data Entry Operator
DGM	Deputy General Manager
DGM/AM	Deputy General Manager/Asset Management
DGM/IT	Deputy General Manager (Information Technology)
FMS	Fleet Management System
GM	General Manager
GPRS	General Packet Radio Service
GPS	Global Positioning System
RFID	Radio Frequency Identification
UI	User Interface
UML	Unified Modeling Language

Chapter 1 : Introduction

1. Background

Ceylon Electricity Board (CEB) is the national power utility and is a statutory body established by the Act Number 17 of 1969, with reflecting a monopoly in electricity power sector in Sri Lanka. CEB core business processes include power generation, transmission, distribution and retailing. Electricity demand is currently met by nine thermal power stations, fifteen large hydroelectric power stations, and fifteen wind farms, with a smaller share from small hydro facilities, a coal power station and other renewables such as solar[1]. CEB operates with nearly 6.1 million customers as of 2016.

All functions are operated in island wide nearly 800 CEB workstations. Each workstation comes under division, branch, unit or sub unit and also can be named as a cost centre. Each cost centre has number of vehicles.

There are many divisions function under by Chief Executive Officer (CEO). Each division functions separately and CEO is responsible for all divisions. The hierarchy of the CEB is attached (Annexure 1).

The vehicles are monitored by head of each hierarchy level in CEB and Additional General Manager/Asset Management (AGM/AM) is responsible for monitor all vehicles. As CEB does not have a proper system to identify or monitor total number of CEB vehicles, vehicle types, hired vehicles, fuel consumption, maintenance cost, ownership of the vehicle, vehicle parking location, vehicle transfer history etc. AGM/AM could not been able to get exact information on vehicles.

Some vehicles are restricted to specific projects, some are strictly organizational driver driven and others self-staff based on administration policies related to CEB. The function of vehicle is dependent on CEB policies and structures.

1.1. Motivation

CEB is the licensee which is responsible for generating, transmitting, distribution and sale of electrical energy in Sri Lanka. It is responsible for provide electricity continuously to the customers without any issues. The system will optimize the use of vehicle in accordance with the organizations operating needs, enhance the quality of the service and develop efficiencies to make the service more effectively.

1.2. Aims & Objectives

The main aim of the project is to develop a web based Fleet Management System (FMS) to increase efficiency and productivity of the CEB vehicle management activities in a smooth manner except the emergency situation activities (natural disaster situation, breakdown services etc.).

Further this system will provide the facilities such as identification information of the vehicle, details of vehicle location, allocation to officer(s) and driver(s), vehicle transfer history, vehicle booking, monitor the details of manage renting dates of hired vehicles, revenue license and insurance date etc. Also possible to determine the number of excess or shortage of vehicles in unit wise, sub unit wise etc.

In addition to the above functionalities, the system will generate following reports.

- Summary of refueling by vehicle and average fuel consumption
- Summary mileage of each vehicle
- Services and maintenance
- Accident details.

The main objectives of this project will generate information in connection with the decision making activities in the followings.

- To determine the maintenance cost whether is it high or low.
- To determine whether fuel consumption is economical or not.
- To provide the recommendation whether the vehicle is further maintenance or dispose.
- To provide statistics details like total fuel consumption, maintenance cost (unit wise, sub unit wise, etc.) and required information for the next year budget estimation.

1.3. Methodology

Fact finding method is an important activity in a system design. There are various methods use for fact findings. Referring web resources, onsite observation, conducting interviews and studying the similar systems were used as fact gathering techniques in this system.

There are several methodologies available in software development. They are agile software development, waterfall method, incremental model, spiral method, rapid application development etc. To develop this system, incremental model was used.

1.3.1. Waterfall Model

Waterfall Model is referred to as a linear-sequential life cycle model. It is very simple to understand and use. In this model, each phase must be completed before the next phase begin. Therefore any phase in the development process begins only if the previous phase is complete and the outcome of one phase acts as the input for the next phase sequentially [2]. The sequential phases are requirement analysis, system design, implementation, testing, development and maintenance.

The following Figure 1.1 shows the waterfall life cycle model.

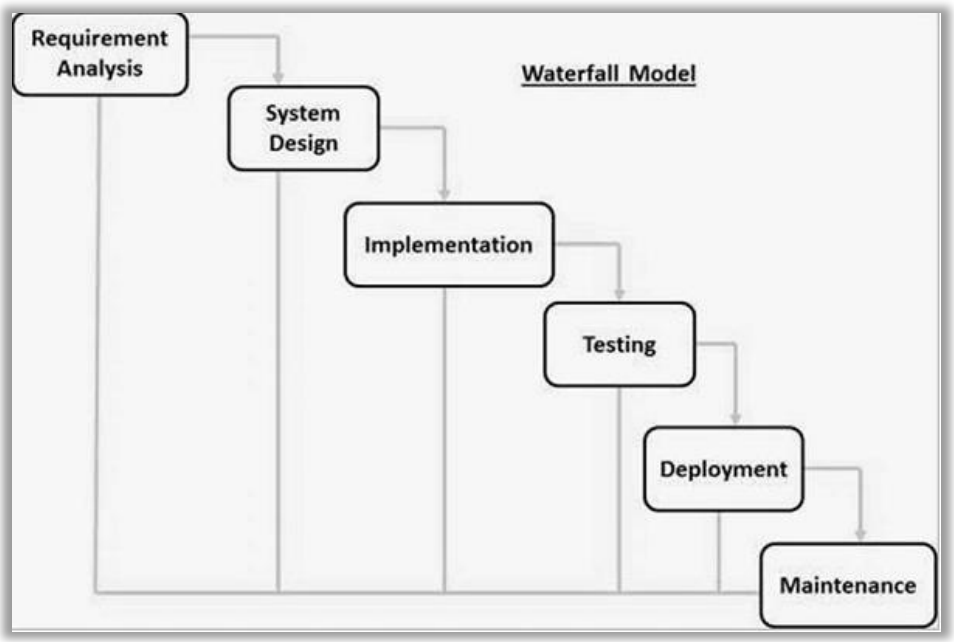


Figure 1.1: Waterfall life cycle model [2]

1.3.2. Incremental Model

Incremental Model is a process of software development. In this mode, requirements are broken down into small multiple builds [3]. This each build passes through the requirements, design, coding and testing phases until the product is finished.

The following Figure 1.2 represents the Incremental life cycle model.

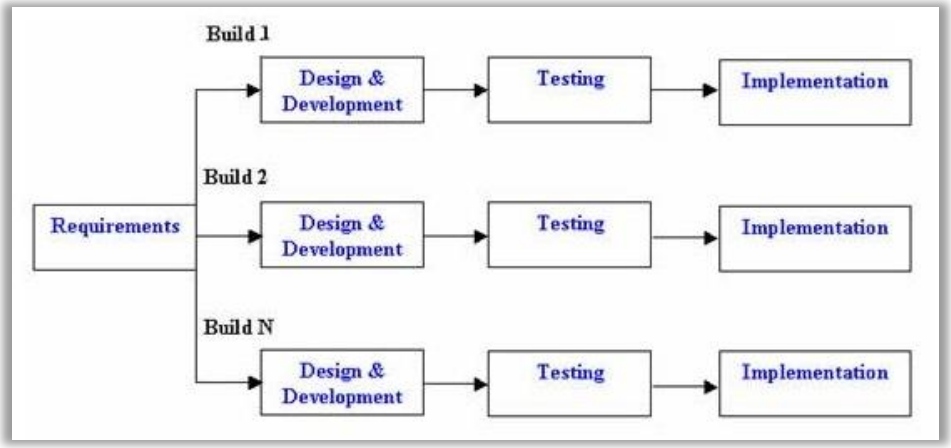


Figure 1.2: Incremental life cycle model [3]

Waterfall method is good when requirements are very well documented, clear and fixed. And also have are no ambiguous requirements. To develop the system incremental model was used because it helped to generates working software quickly and early during the software life cycle, more flexible, less costly to change scope and requirements and easier to test and debug during a smaller iteration. It is easier to manage risk because modules are handled separately and risky pieces are identified and handled during its cycle. During the incremental life cycle, developer can closely work with customer.

1.4. Scope of the project

The system is able to capture the followings.

- Vehicle Registration Management
    - CEB own Vehicle Data Management (vehicle number, year, make, model, etc.)
    - Hired Vehicle Data Management (vehicle number, year, make, model, Contract Period, Rate per Km, etc.)

- Vehicle Allocation
    - Allocation to the Officer/ Officer In Charge
    - Vehicle Transfer and Allocation Parking
- Booking Details
- Running & Fuel Management
    - Fuel Usage and Running Chart Management
- Vehicle Management
    - Services
    - Accident Details
    - Spare Part Details
    - License & Insurance
- Reports
    - Vehicle Details
    - Summary Mileage of each Vehicle
    - Statistics and budget estimation

1.5. Limitations of the project

As the limitations of the project period the following identified areas are also not included in this project.

- Payments
- Vehicle disposal
- GPS (Global Positioning System)
- Mobile/Android application
- CEB letters & forms
- CEB vehicle usage related circulars

Assumptions:

All vehicle details are registered in Fleet Management System after vehicle purchasing (Purchasing methods and procedures are not included here) for this project.

1.6. Structure of Dissertation

The second chapter consists of similar systems and literature review. This clearly shows the comparison between existing systems and proposed system. The third chapter describe requirement specifications (Functional and Non Functional requirement). The fourth chapter provides design part with contains use cases, class diagrams, etc. The fifth chapter consists of Implementation of the project. The sixth chapter consists of testing and evaluation.

Chapter 2 : Background

2. Introduction

There are various online similar systems and journal publications related to the fleet management system. But similar systems and after studies related to the CEB are very difficult to find. But this review extended to the study of the existing systems and journal publications used by similar organizations to analyze the features that were exposed through them. This review helped to formulate the functional and non-functional requirements based on the need and availability of technical resources.

2.1.Similar Systems

2.1.1. Vehicle Fleet Manager

Vinity Soft Business Solutions is a Canadian company, provides a solution for vehicle management system. Their unique approach to developing products is based on real world experience. This fleet management solution provides vehicle and driver maintenance records, fuel maintenance, accident detail maintenance, inventory management, maintenance scheduling, Parts Management, track vehicle expenses, track vehicle expenses and depreciation, etc.

There is a detailed module for reports generated, which includes 3 categories as vehicle, personal and vendors [4].

Some of the key features are

- Service remainders – Create new remainders or updating existing ones.
- Service history – log all maintenance as performed.
- Fuel log – fuel history details.
- Parts inventory – keep track of spare parts inventory and get reminded when parts needs to be re-ordered.
- Notes & issue tracking – operators and mechanics can quickly report issues on vehicles and equipment during inspections.
- Documents – store and view important documents related to particular vehicle, driver, service or work order such as insurance records or scanned invoices.

- Work orders – create work orders and keep track of their progression. Convert them to ‘completed services’ once they are closed.

The following Figure 2.1 shows the main menu of the Vehicle Fleet Manager system. It shows registered vehicle details and other menus such as vehicle payments, insurance payments, service schedule, work orders, etc.

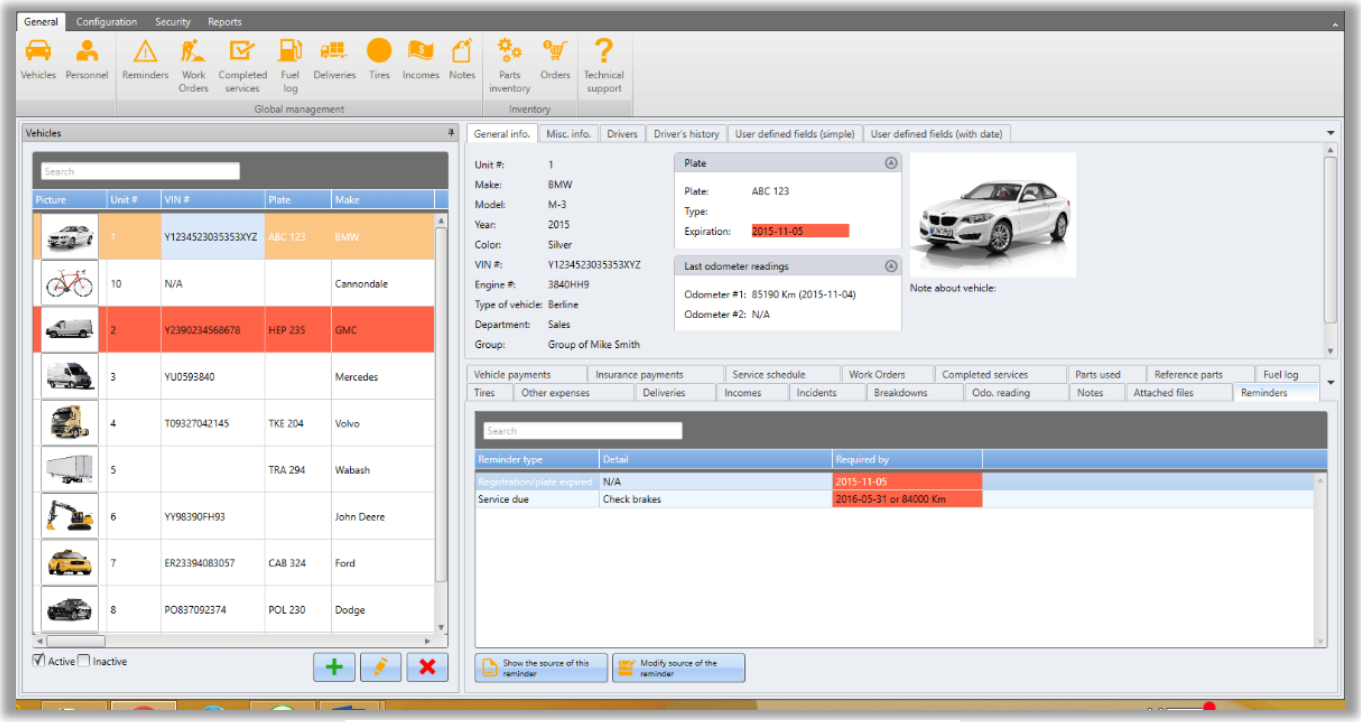


Figure 2.1: Vehicle Fleet Manager [4]

The Figure 2.2 shows Vehicle registration details entry form in the Vehicle fleet manager system.

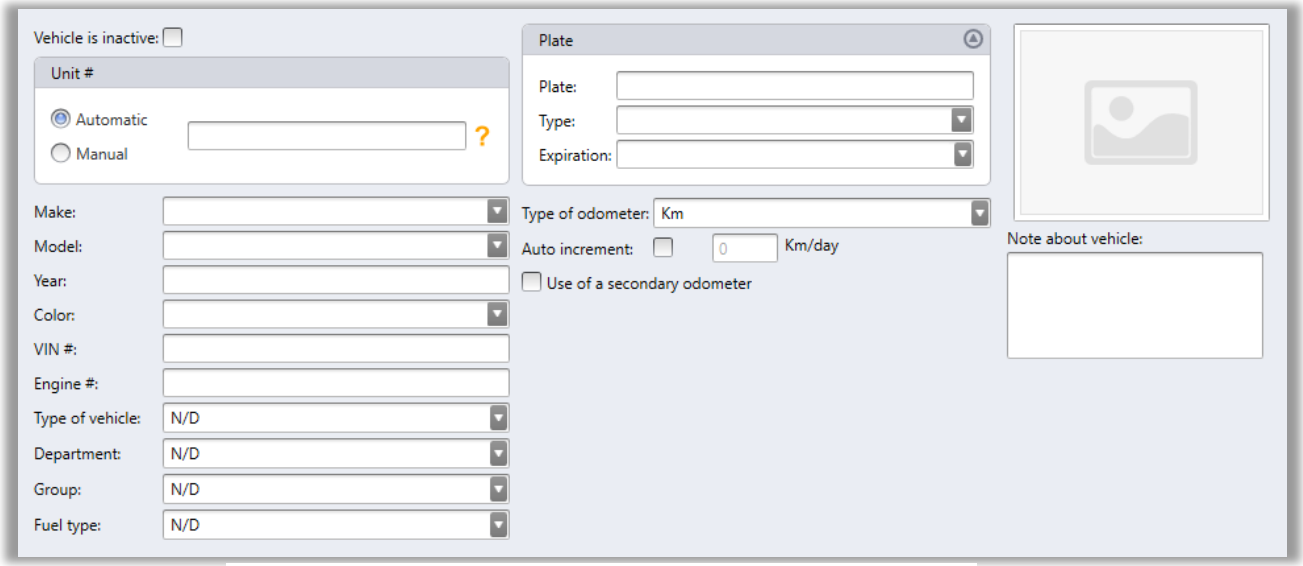


Figure 2.2: Vehicle Fleet Manager – Vehicle registration form [4]

The following Table 2.1 elaborate the current situation against the CEB.

Fleet Management Software	**Fleet Management System for CEB**
Provides manage inventory management	No such inventory management module
Unable to maintain history maintenance module (hired vehicle history, vehicle transfer details, etc.)	Provide History records maintenance
Unable to maintain luxury tax and emission test module	Maintain luxury tax and emission test module
No facility to maintain vehicle transfer records	Maintain vehicle transfer records
Provides manage parts management	No such parts management module

Table 2.1: Differences between Fleet Mgt. software system and Fleet Mgt. for CEB system

As a summary, this system has much more functionalities such as parts inventory, notes & issue tracking, work orders, deliveries, etc. which are not relevant to the users present expectation. But most of the features which are suitable for the future development system, hired vehicle history maintenance, vehicle transfer from one workstation to another workstation with or without officer or/and driver, emission test, luxury tax maintenance, etc. not available in this Fleet management software system.

2.1.2. Fleet Maintenance Pro

Innovative Maintenance Systems is a USA company, provides a solution for vehicle maintenance management software. Fleet Maintenance Pro provides fleet inventory tracking, preventative maintenance, repair maintenance, history recording, fuel tracking, parts inventory, and reporting.

Fleet Maintenance Pro will track an unlimited number of fleet vehicles and equipment. It can track most of the descriptive details that need to manage fleet, including the vehicle specifications, purchase details, loan/lease details and most of the information. Even have user-definable fields for tracking additional equipment facts that are important [5].

Some of the features are

- Repair maintenance – Keeping track and monitoring trends in repair maintenance.
- Preventative Maintenance – schedule repairs for unexpected problems that drivers encounter.
- History recording – maintain history details such as maintenance, fuel, expenses, accident, inspections, etc.
- Fuel tracking – Monitoring fuel consumption.
- Equipment details
- Work orders – generate work orders for each piece of equipment based on this due maintenance.
- Tire inventory
- Parts inventory – provide a complete set of parts inventory features, including stock management, part associations, reorder notifications, and a built-in purchase order system.

The Figure 2.3 shows Vehicle registration details entry form in the Fleet maintenance pro system.

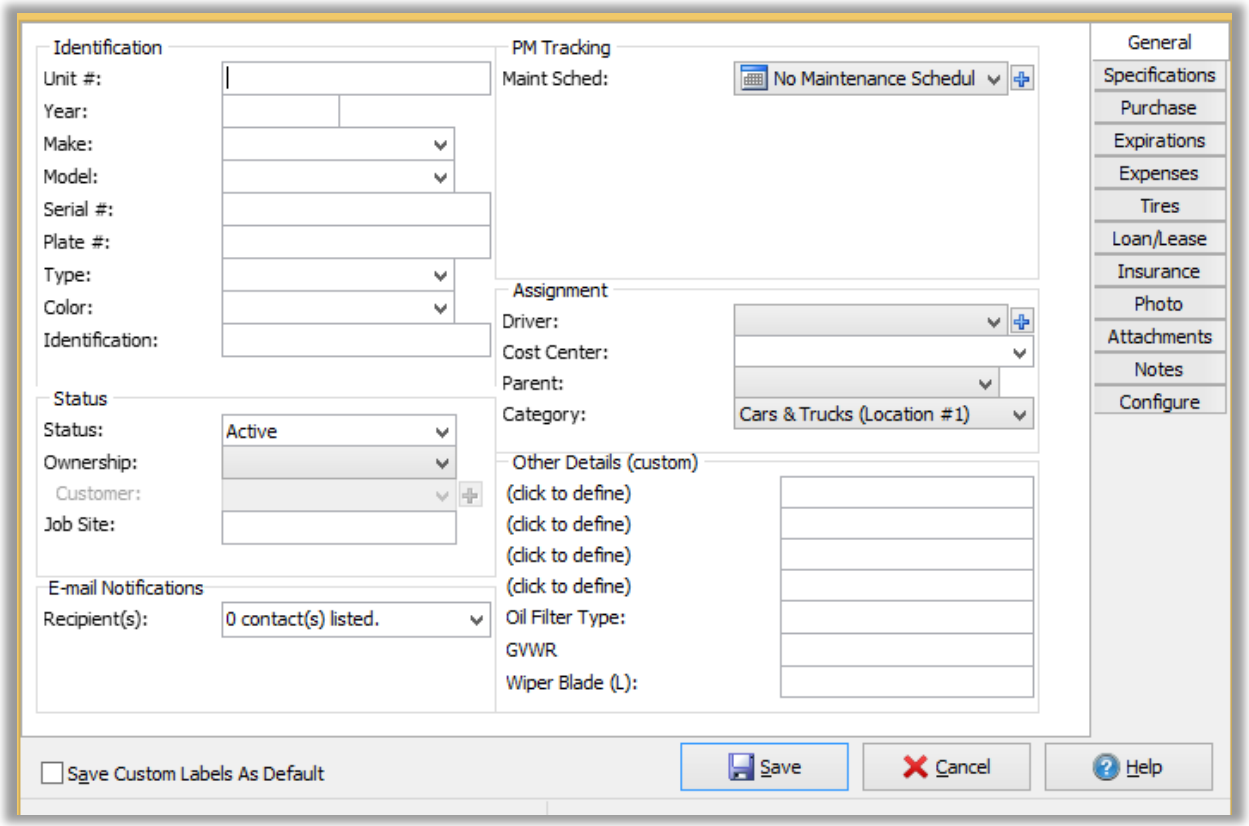


Figure 2.3: Fleet maintenance pro – Vehicle registration form [5]

The following Table 2.2 elaborate the current situation against the CEB.

Fleet Maintenance Pro	Fleet Management System for CEB
Provides inventory tracking module	No such inventory tracking module
Provides preventative maintenance module	No such preventative maintenance module
Unable to maintain luxury tax and emission test module	Maintain luxury tax and emission test module
No facility to maintain vehicle transfer records	Maintain vehicle transfer records
Provides parts inventory module	No such parts inventory module

Table 2.2: Differences between Fleet Mgt. Pro system and Fleet Mgt. for CEB system

This is closer to the proposed system but this has much more functionalities such as parts inventory, tire inventory, Preventative Maintenance, work orders, etc. which are not relevant to the proposed system. But most of the features which are suitable for the proposed system, hired vehicle history maintenance, vehicle transfer from one workstation to another workstation with or without officer or/and driver, emission test, luxury tax maintenance, etc. not available in this Fleet management pro system.

2.1.3. Fleetio Manage

Fleetio is an Alabamian company launched in January 2012, and today thousands of people use Fleetio to manage hundreds of thousands of vehicles, equipment, parts, drivers and more. Fleetio Manage gives everything need to manage a fleet, whether have 10 or 10,000 vehicles and equipment. Fleet operations are complex and difficult to manage. People start out trying to use spreadsheets but quickly outgrow them. Many companies do not actively manage their fleet operations at all, hoping people will drive safely and be responsible for their vehicles [6].

Some key features are

- Inspections
- Issues – Capture issues as soon as they arise.
- Remainders
- Services – Keep detailed service.

- Vehicle management
- Fuel management
- Contacts & users – Manage information and documents about everyone that touches fleet
- Parts & inventory
- Manager-level control – Keep detailed service records and identify which vehicles are regularly in the shop as well as their associated maintenance costs.
- Work orders – Plan, schedule and carry out detailed maintenance tasks and costs.
- Viewer access – Allow specific Users to view specific data in Fleetio but prevent them from being able to create or edit records.

The Figure 2.4 shows Main menu & Vehicle registration details entry form in the Fleetio manage system. It shows menus such as dashboard, work orders, inspections, issues, etc.

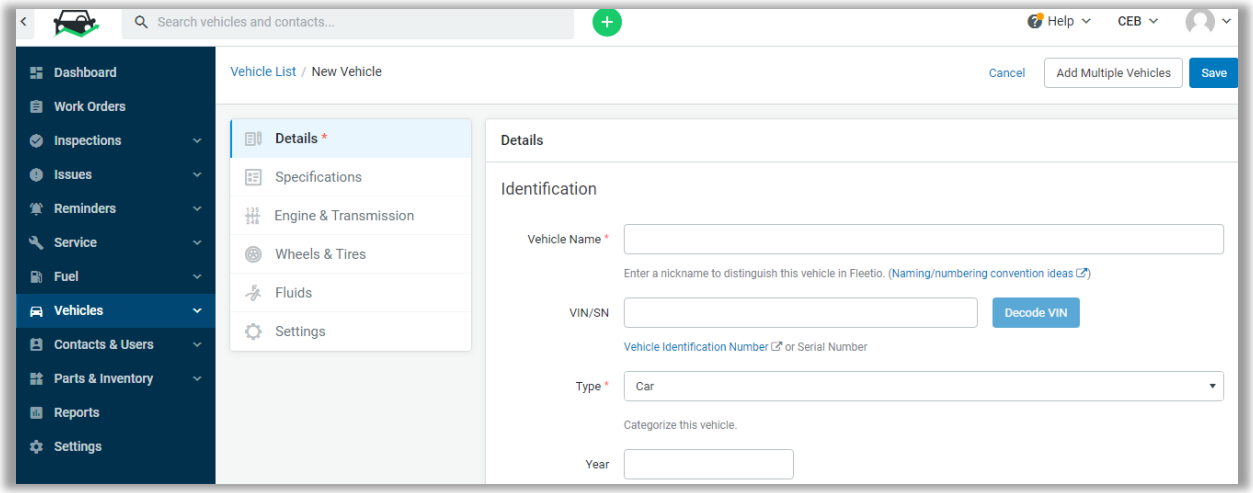


Figure 2.4: Fleetio manage Vehicle registration screen [6]

Figure 2.5 shows one of fuel consumption report for a vehicle. It's clearly indicate average usage, price, and total fuel cost, etc.

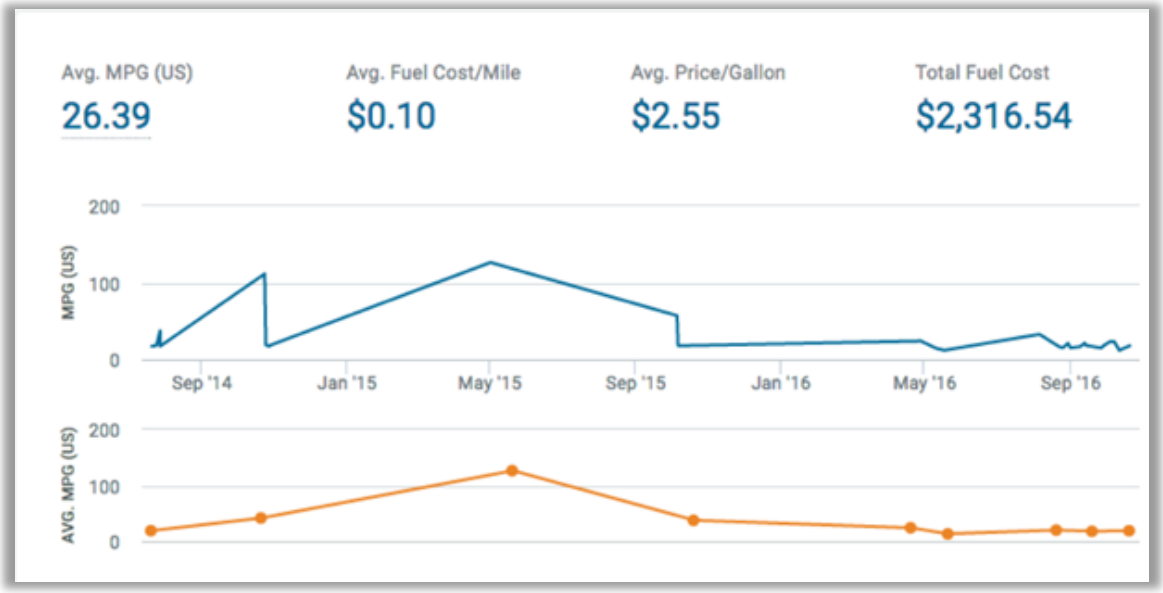


Figure 2.5: Fleetio manage Fuel management report [6]

The following Table 2.3 elaborate the current situation against the CEB.

Fleetio Manage	**Fleet Management System for CEB**
Provides work orders module	No such work orders module
Unable to maintain luxury tax and emission test module	Maintain luxury tax and emission test module
No facility to maintain vehicle transfer records	Maintain vehicle transfer records
Provides parts & inventory module	No such parts & inventory module

Table 2.3: Differences between Fleetio manage system and Fleet Management system for CEB

This has much more functionalities such as parts inventory, work orders, manage GPS devices, inspections, etc. which are not relevant to the proposed system and some of the basic information of the vehicle cannot identify in this system. Some important information about vehicle such as hired vehicle history maintenance, vehicle transfer from one workstation to another workstation with or without officer or/and driver, emission test, luxury tax maintenance, etc. not available in this Fleetio manage system.

2.1.4. Fuel consumption and collision prevention systems modules

According to the journal paper [7], it described about Fuel consumption and collision prevention systems modules. This uses GPS based technology, General Packet Radio Service (GPRS) and cloud computing infrastructure. The vehicle used GPS device with a GPRS module, sensors for identifying fuel level/status, driver identification and collision prevention system. This paper discussed different kind of embedded devices used to capture data. It includes sending, receiving, storing and processing of measured information and all kinds of actions needed for controlling devices remotely.

The sensors are used to monitor the fuel level of the vehicle, how much fuel was filled into the tanks, set alerts to specify fuel levels & sends information to the device about the fuel level changes in the tank. When the fuel level reaches critical level it notifies the driver and the administrator of the system.

In driver identification module, it prevents unauthorized usage of the fleet that is owned by the company. When an unauthorized usage detected the system notifies the owner which vehicle and the location of the vehicle is being used [7]

Followings are some of the key of this paper:

- Vehicle management
- Fuel management
- Driver identification
- Collision management
- GPS (tracking & navigation)

The Figure 2.6 shows overview of fleet management system, combining machine to machine with cloud computing.

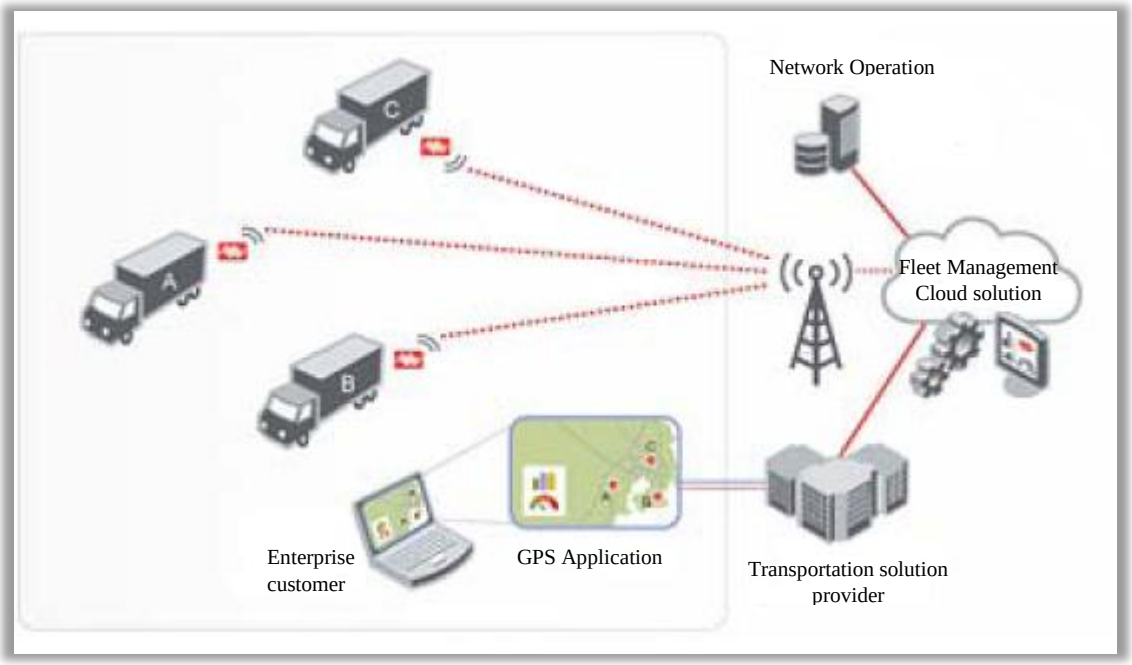


Figure 2.6: Overview of fleet management system [7]

2.1.5. Introduction to the Basis of Fleet Management Technology

According to the research paper [8], there are different types of fleets as motor vehicles, aircraft, marine vessels and trains. Basically this paper focus on about fuel savings via GPS tracking (monitor idling, speed and other vehicle performance), productivity improvements with route optimization tools (appointments and deliveries across the service area), driver accountability with GPS tracking and geo-fencing tools and automatic repair/maintenance alerts. This is used GPS, RFID (Radio Frequency Identification) and telematics devices to gather information and data.

In [8], it is described the usage and importance of embedded devices. GPS or RFID provide time and location data. Telematics provides more data than GPS or RFID devices such as engine temperature, oil pressure, engine speed in rpm, engine warning lights, etc. This also focus on a variety of real and hypothetical case studies of transportation (ground and air), retails services, home healthcare, agriculture and field services.

The following main benefits are described in the paper.

- Vehicle maintenance and safety
- Efficient operations

- Fuel economy statistics
- Staff management
- Customer service

According to "W.G Wenzel (2016) [8] report", it is mainly focus on the advanced computing technology to help make save money. This used embedded devices to capture data for different opportunities to improve operations, maintain vehicles, enhance safety and driver performance, manage and train staff, serve customers and analyze big economic picture. This paper described much more functions about the fleet management but the proposed system needs to identify basic functions such as hired vehicle history maintenance, vehicle transfer from one workstation to another workstation with or without officer or/and driver, emission test, luxury tax maintenance, etc.

2.1.6. Best practice fleet management and priority actions

In this research paper [9], discussed about the workplace road safety policies and procedures. Some of the selected company managers reported that road trauma is a social, human, financial, reputation, efficiency and operational risk to an organization.

The main key features are

- Workplace safety and road safety management
- Vehicle safety
- Driver safety
- Road traffic safety management systems
- Corporate road safety programs
    - Road traffic safety management systems
    - Driving for better business
    - National road safety partnership program

According to "Best practice fleet management and priority actions[9] report" ,it is mainly focus on the workplace road safety policies and procedures. But the functions described in this paper is not related to the proposed system. The proposed system needs to identify basic functions of

vehicle, driver, history maintenance, vehicle transfer from one workstation to another workstation with or without officer or/and driver, emission test, luxury tax maintenance, etc.

According to the expectations of similar systems and research papers in the internet, unable to identify the CEB hierarchy. Therefore a user login authority levels and module permissions cannot be created according to CEB hierarchy. Also Work flow path for the recommendations and approvals cannot be defined. The above systems provide basic understanding of the requirements and help to clarify certain functions of the system. These functions of the following systems can be used for the proposed system.

Chapter 3 : Requirement Analysis

3. Information Gathering Method

In this study, requirement gathering is a very comprehensive process due to the existing manual adhoc procedure of CEB Fleet Management System. As the ways of fact gathering techniques interviews and onsite observation techniques were used in this study. Interviewing the Deputy General Manager/Asset Management (DGM/AM), other Deputy General Managers (DGMs) and staff members who are handling vehicle related document files were supported to gather information which were needed for analyzing the requirement. After meeting managers and staff members observation method was used to understand the manual process.

Based on the outcome of the above data gathering techniques, all the administrative functions were identified.

3.1. Current System

The current system is manually operated. It is very complex, inefficient, time consuming due to data duplication, lack of data integrity and data inconsistency, etc. also it is decentralized and flat file method is being used.

The process of the present situation

- All vehicle master details are documented.
- An officer or officer in-charge is allocated to the vehicle.
- Parking location is arranged. (It may be a CEB parking location or a private parking location.)
- When transferring a vehicle, document file is transferred to the new location.
- Every employee who has a CEB vehicle has to manage the running chart. (Running chart should be maintained daily or monthly.)
- Data maintenance done using an excel sheet.

3.2. Analysis

According to the present situation, the system is categorized under following modules. It was as Vehicle Registration Management, Vehicle Allocation, Booking Details, Running & Fuel Management, Vehicle Management and Report Generation.

AGM/AM is responsible for all CEB vehicles and other senior managers (DGMs) are responsible for vehicles in their own cost centers.

At present, no method to determine excess vehicle or shortage in a cost centre. Also no acceptable method to book vehicles. Therefore it is hard to comprehend a schedule or to cancel a journey in the current situation.

3.3. Present users of the system

In this system, the current users of the system as follows.

- Subject Clerk/Data Entry Operator (DEO) – person who handles the vehicle details in the system
- Chief Clerk (CC) – person who checks the vehicle details
- Driver
- Authorized Officer
- GM/AGMs/DGMs – person who views management reports
- Administrator – person who administrates the system
- Officers who are allocated for CEB Vehicle

3.4. Functional Requirements

3.4.1. Vehicle Registration Management

Under the vehicle registration, following task should be provided.

- Vehicle number is the primary key will help to identify a record uniquely.
- Editing details should be updated in master table and history record should be maintained.
- View facility should be provided hired vehicle historical details such as contract period details, fuel rates, etc.
- Only single record should be added to the master table.
- Facility should be provided for add, edit, delete, validate, approve and search a vehicle.

3.4.2. Vehicle Allocation

Followings should be provided to manage vehicle allocation information on officers and parking location.

- Any vehicle should be allocated to the officer/ Officer-In-Charge.
- Vehicle should be transferred from one cost centre to another.
- Allocation parking for a vehicle.
- Add, edit, cancel, view facility should be provided.

3.4.3. Booking Details

Should be allowed to user to book a vehicle from a pool to which they belong or have been tagged to.

- View facility should be provided for booking details.
- Only pool vehicle can be booked.
- The user who booked a journey could be cancelled a booking.
- Booking can be done only for future.
- Add, edit, cancel, view facility should be provided.

3.4.4. Running & Fuel Management

Fuel expense management requires careful scrutiny to take every opportunity for minimizing cost and mitigating risk. Under the running and fuel management, the following task to be done.

- Officer, subject clerk or driver could be entered running chart details.
- Fuel entry details should be enter officer or subject clerk.
- Running chart details should be enter daily or month wise in summary format.
- View facility should be provided for running chart details and fuel entry details.
- Add, edit, cancel facility should be provided.

3.4.5. Vehicle Management

The modules should be categorized as Vehicle Maintenance, Accident Details, Spare Part Details, Accessory Details and License & Insurance

- Subject clerk or Chief Clerk can be entered details.
- History details should be maintained.
- Add, edit, cancel, view facility should be provided.

3.4.6. Report Generation

Reports should be provided for all modules separately. This reduces the work load of management, analysis to make decision effectively about fleet utilization, maintenance, etc. Following reports should be generated in vehicle registration.

- CEB own vehicles
  - Branch wise vehicle details, Class wise vehicle details, Running chart details, Accident details, Service & maintenance details
- Hired vehicles
  - Details of hired vehicles, Running chart details
- View facility should be provided

3.5. Non Functional Requirement

- Accuracy – report generation, data output should be accurate in the system.
- Accessibility- system should have accessible to the identified users.
- Extensibility – system should be easily customizable at the later stage.
- Usability – user friendliness of the system should be very high.

- Resource constraints (memory, disk space) – system should have proper response time and processing time without delay. Also should have proper back procedure on the system.

3.6. Present gap against the requirements

The present system is done by a manual process in the adhoc manner based on each cost centre. Therefore, data is decentralized and flat file method has been used. In this system, all vehicle details were gathered to the excel sheet. As nearly 800 cost centres in CEB, it is difficult to collect details in proper time in appropriate way. Therefore, it was realized that data duplication, lack of integrity, inconsistency, etc. are arising.

In this study, it was found in difficult to identify vehicle history details in several years. It was identified some valuable data and records are missing at this stage (ex. Hired vehicle details, running charts, etc.).

Following are the justification for each state,

- Details were documented as hard copies.
- Unavailability of some cost centres (ex: name changed has been done of cost centres, associated with another cost centre, terminated)
- No one has analyzed history details

According to the above considerations it was realized this process is inefficient, complex and consumes duration of time.

Chapter 4 : Design of Solution

4. Introduction

As the way of eliminating the existing difficulties of the system and to achieve above mentioned objectives, the new system can be designed as follows. To design the new system, the following techniques used in this study. They are main architecture diagram, use cases, sequence diagrams, activity diagrams and class diagram. Fleet Management System (FMS) architecture diagram and it's related users can be represented using the following Figure 4.1 and Figure 4.2.

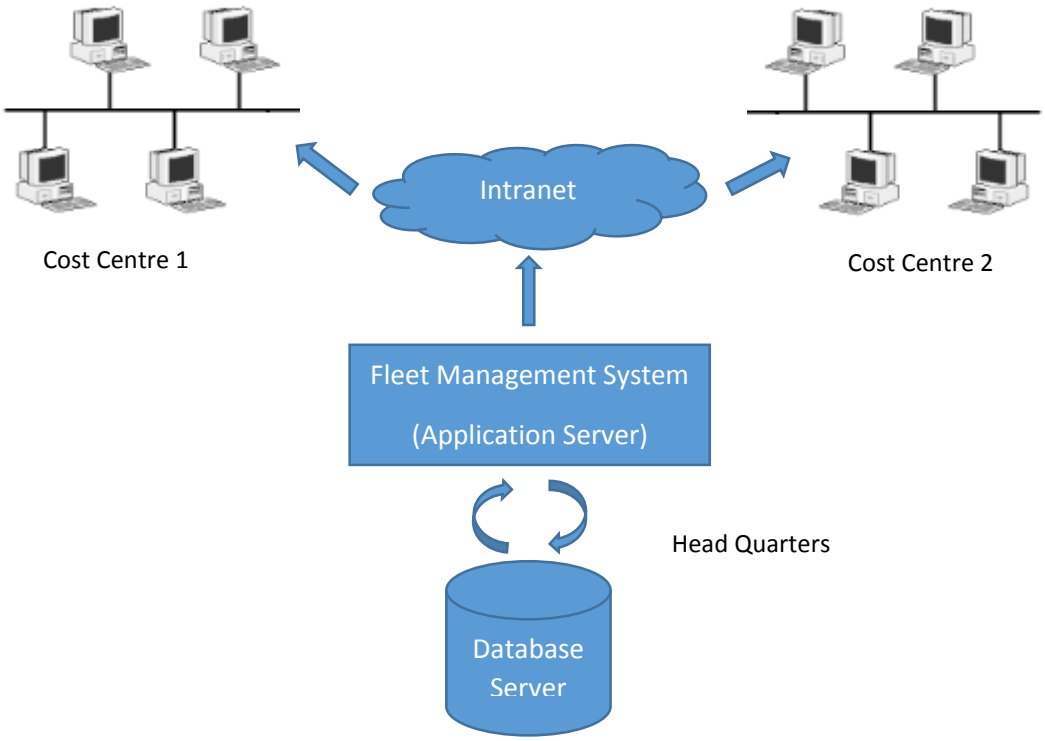


Figure 4.1: System Architecture

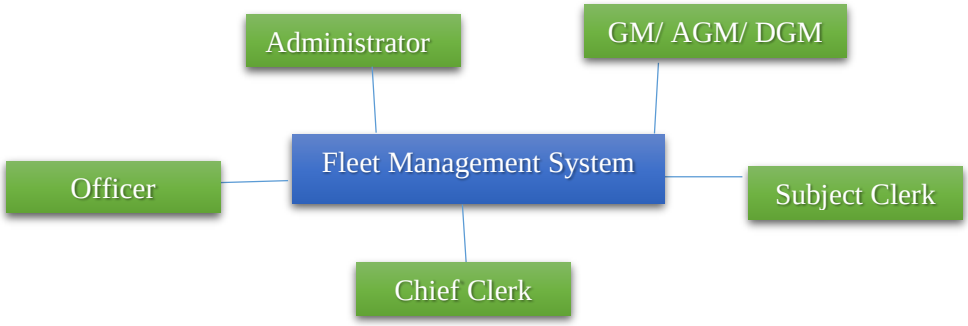


Figure 4.2: Users of the system

4.1. Use Case Diagrams

Use case diagrams are high level user requirement analysis of a system. This can be used to gather system requirement, get an outside view of a system, identify the external and internal factors influencing the system and identify the actors, etc. [10].

Use case diagrams are useful to represent presentations to stakeholders.

4.1.1. Login

Following use case diagram (Figure 4.3) & description (Table 4.1) helps to identify the user login process of the system.

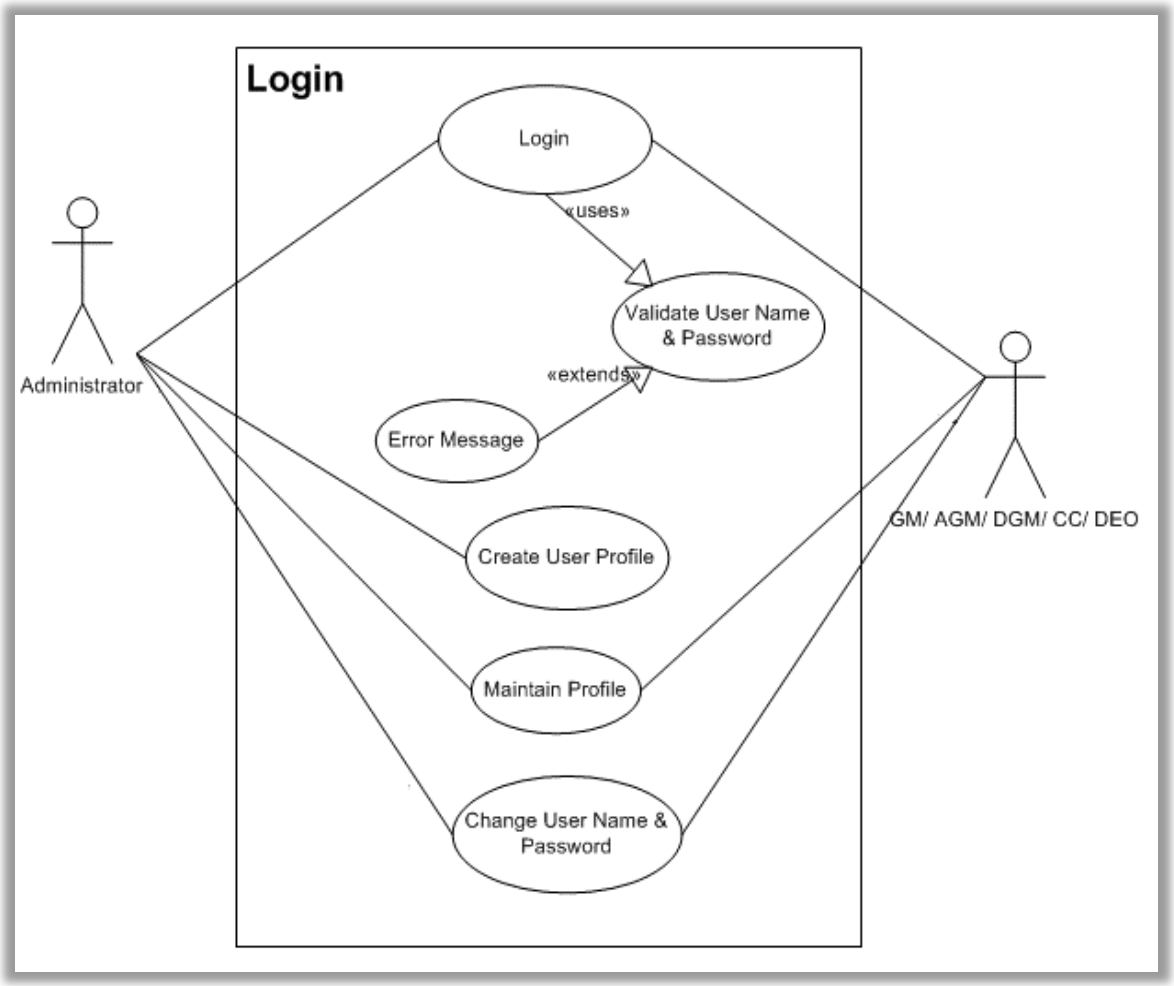


Figure 4.3: Login process use case diagram

Use case Name	**Login**
Users	Administrator, GM, AGM, DGM, CC, DEO, Drivers, Officers
Description	• Create user profile
Precondition	• Administrator create user profile • Users can change their User Name and Password
Typical Course of Events	• Type User Name & Password • System check whether User Name & Password is already exist If User Name & Password are correct Log in to the system Else Display error message
Alternate Course	• If invalid User Name or Password system will display an error message
Post Condition	-

Table 4.1: Use case description of login process

4.1.2. Fleet Management System

Following use case diagram (Figure 4.4) & description (Table 4.2) helps to identify the Fleet management system process of the system.

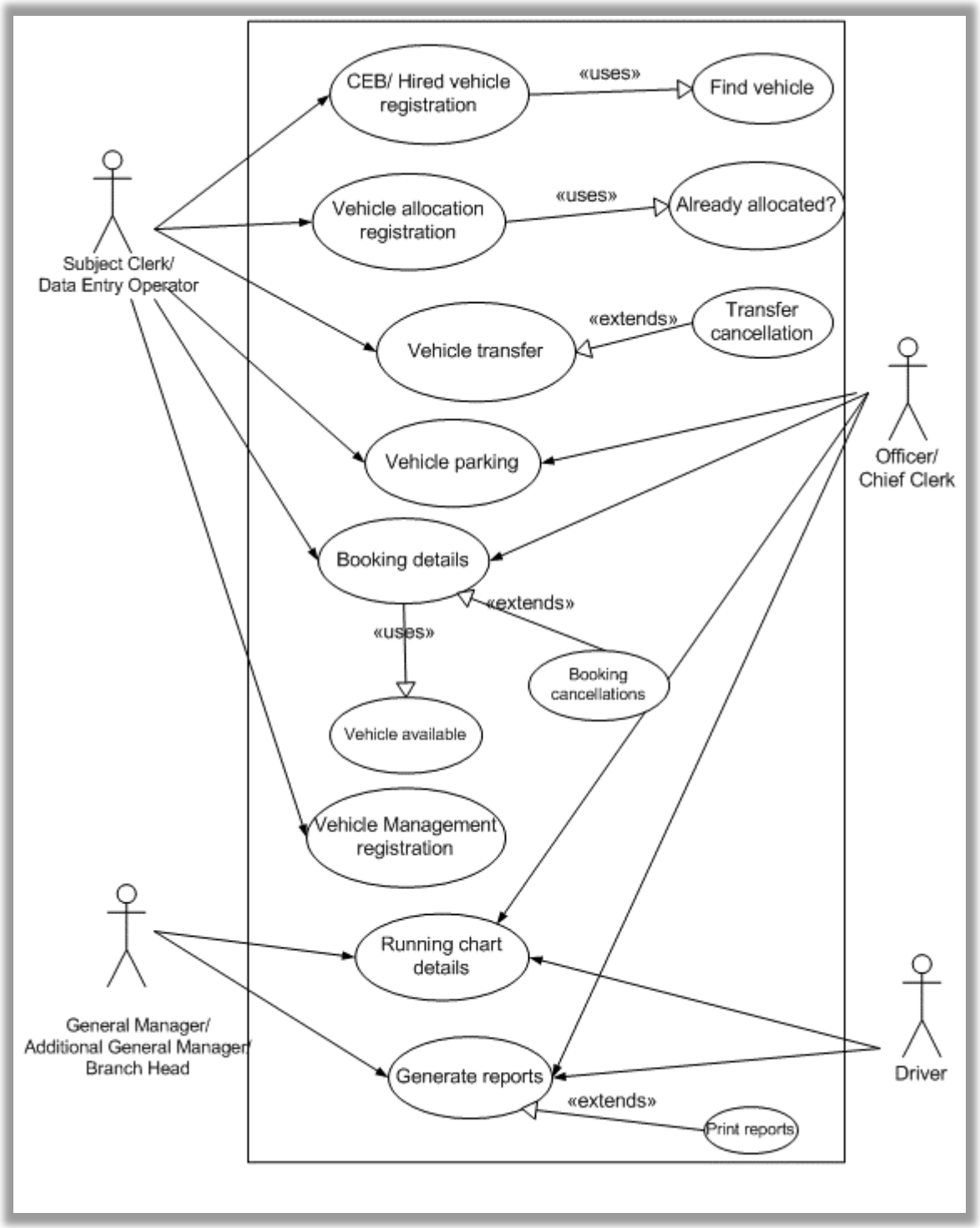


Figure 4.4: Fleet management system

Use case Name	**Fleet Management System**
Users	Administrator, GM, AGM, DGM, CC, DEO, Drivers, Officers
Description	• Process of Fleet Management System
Precondition	• Create user profile • Create user login
Typical Course of Events	• Input CEB or Hired vehicle details • Vehicle allocation to an officer or driver • Vehicle transfer • Vehicle parking • Vehicle booking • Vehicle management registration • Running charts • Report generate
Alternate Course	• vehicle Already exits message • Data invalid messages • Record inserted message
Post Condition	-

Table 4.2: Use case description of Fleet management system

4.1.3. Vehicle Registration

Following use case diagram (Figure 4.5) & description (Table 4.3) helps to identify the vehicle registration process of the system.

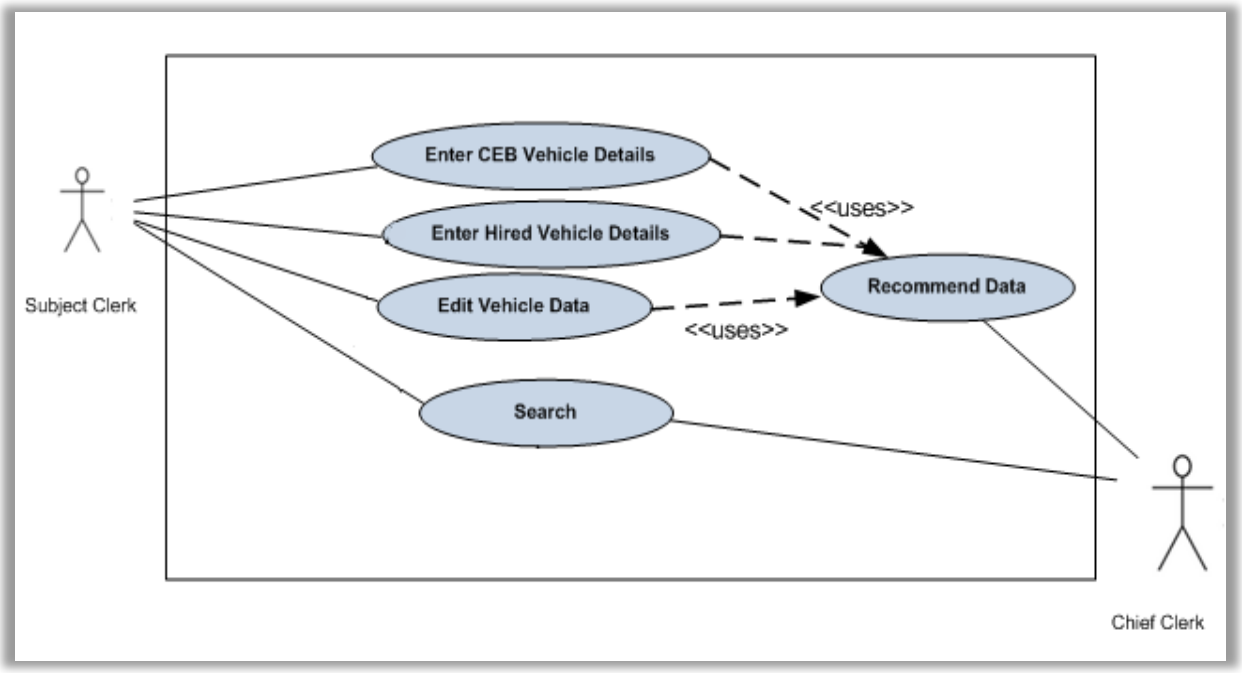


Figure 4.5: Vehicle registration use case diagram

Use case Name	**Vehicle Registration**
Users	Subject Clerk
Description	• Input a CEB or Hired vehicle details such as registration number, make, class, manufactured year, etc. • When registration number entered to the system verified whether it's already exists. • Validate vehicle details
Precondition	• Create user profile • Create user login
Typical Course of Events	• Enter registration number • System check whether registration number is already exists If registration number exists Display vehicle details Else Enter vehicle details Check data validity If valid insert record Else Display error message
Alternate Course	• Data invalid messages • Record inserted message
Post Condition	User can view vehicle running details according to particular vehicle

Table 4.3: Use case description of Vehicle registration

4.1.4. Vehicle Allocation

Following use case diagram (Figure 4.6) & description (Table 4.4) helps to identify the vehicle allocation process of the system.

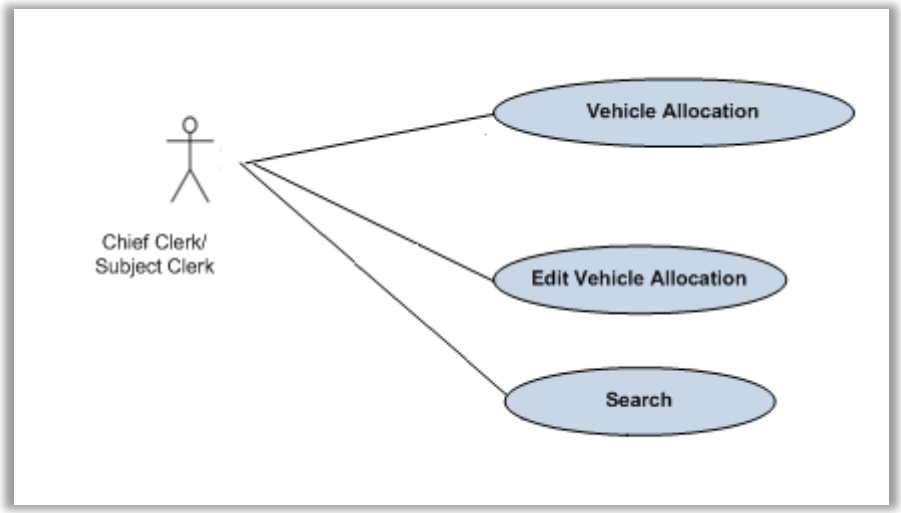


Figure 4.6: Vehicle allocation use case diagram

Use case Name	**Vehicle Allocation**
Users	Subject Clerk, Chief Clerk
Description	• Allocate a CEB or Hired vehicle to an officer or officer-in-charge.
Precondition	• Create user profile • Create user login • Vehicle already exists in the system
Typical Course of Events	• Enter vehicle registration number • System check whether vehicle is already allocated If allocated Display details Else Allocate vehicle to the officer/office-in-charge Check data validity If valid insert record Else Display error message
Alternate Course	• Vehicle already allocated message • Data invalid messages • Record inserted message
Post Condition	User can view vehicle allocation details according to particular vehicle

Table 4.4: Use case description of Vehicle allocation

4.1.5. Vehicle Booking

Following use case diagram (Figure 4.7) & description (Table 4.5) helps to identify the vehicle booking process of the system.

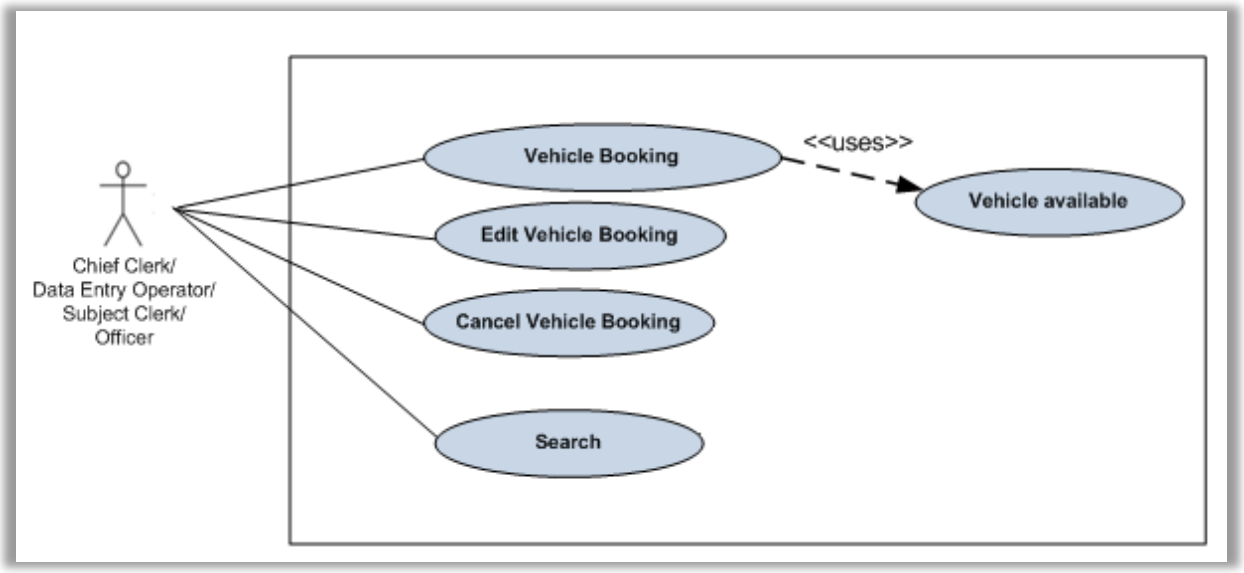


Figure 4.7: Vehicle booking use case diagram

Use case Name	**Vehicle Booking**
Users	Administrator, GM, AGM, DGM, CC, DEO, Drivers, Officers
Description	• Process of vehicle booking
Precondition	• Create user profile • Create user login • Vehicle already exists in the system
Typical Course of Events	• Enter date and vehicle number • Check whether booking is already exists If booking exists Display booking details If cancellation Cancel a booking Else Enter booking details Check data validity If valid insert record Else Display error message
Alternate Course	• Booking Already exits message • Data invalid messages • Record inserted message
Post Condition	User can view vehicle booking details according to particular

Table 4.5: Use case description of Vehicle booking

4.1.6. Running & Fuel Management

Following use case diagram (Figure 4.8) & description (Table 4.6) helps to identify the running and fuel management process of the system.

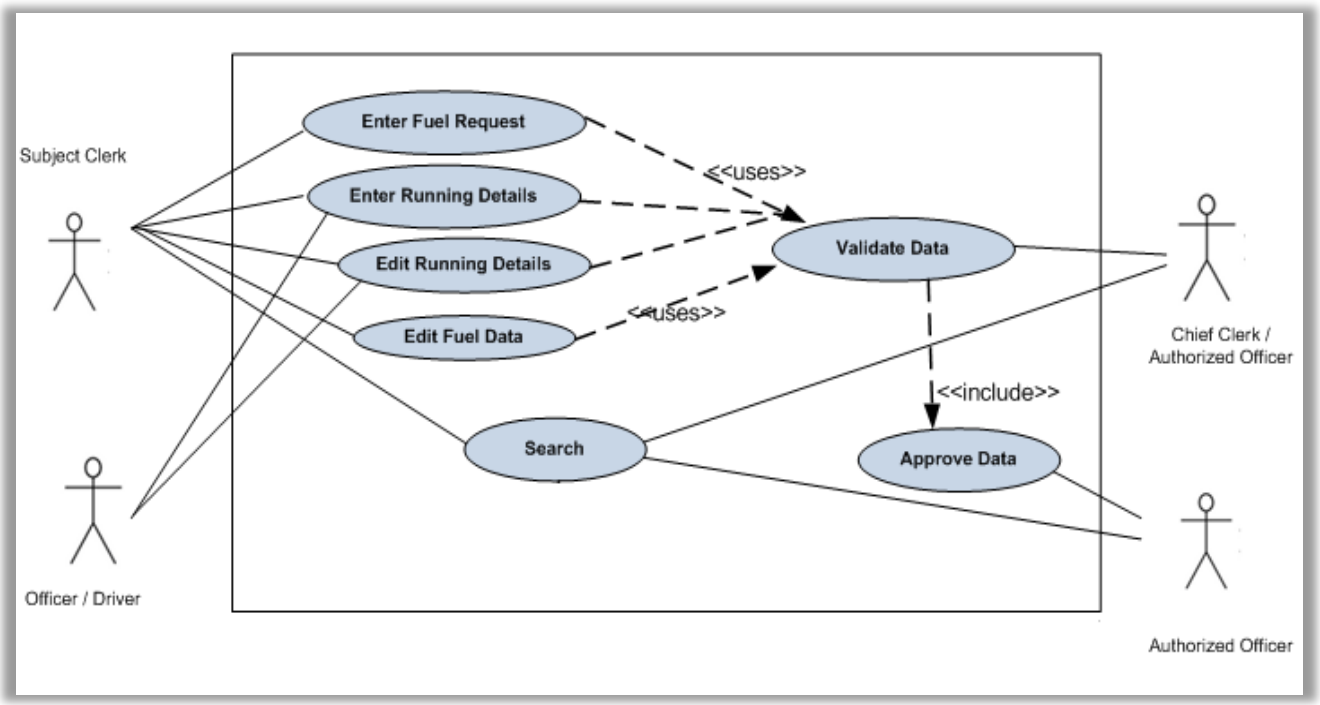


Figure 4.8: Running & Fuel management use case diagram

Use case Name	**Running & Fuel Management**
Users	GM, AGM, DGM, Subject clerk, officer, driver
Description	• Process of running & fuel management
Precondition	• Create user profile • Create user login • Vehicle already exists in the system
Typical Course of Events	• Fuel request • Daily running chart • Summary sheet
Alternate Course	• Booking Already exits message • Data invalid messages • Record inserted message
Post Condition	User can view vehicle booking details according to particular vehicle and particular date.

Table 4.6: Use case description of Running & fuel management

4.1.7. Report Generation

Following use case diagram (Figure 4.9) & description (Table 4.7) helps to identify the report generation process of the system.

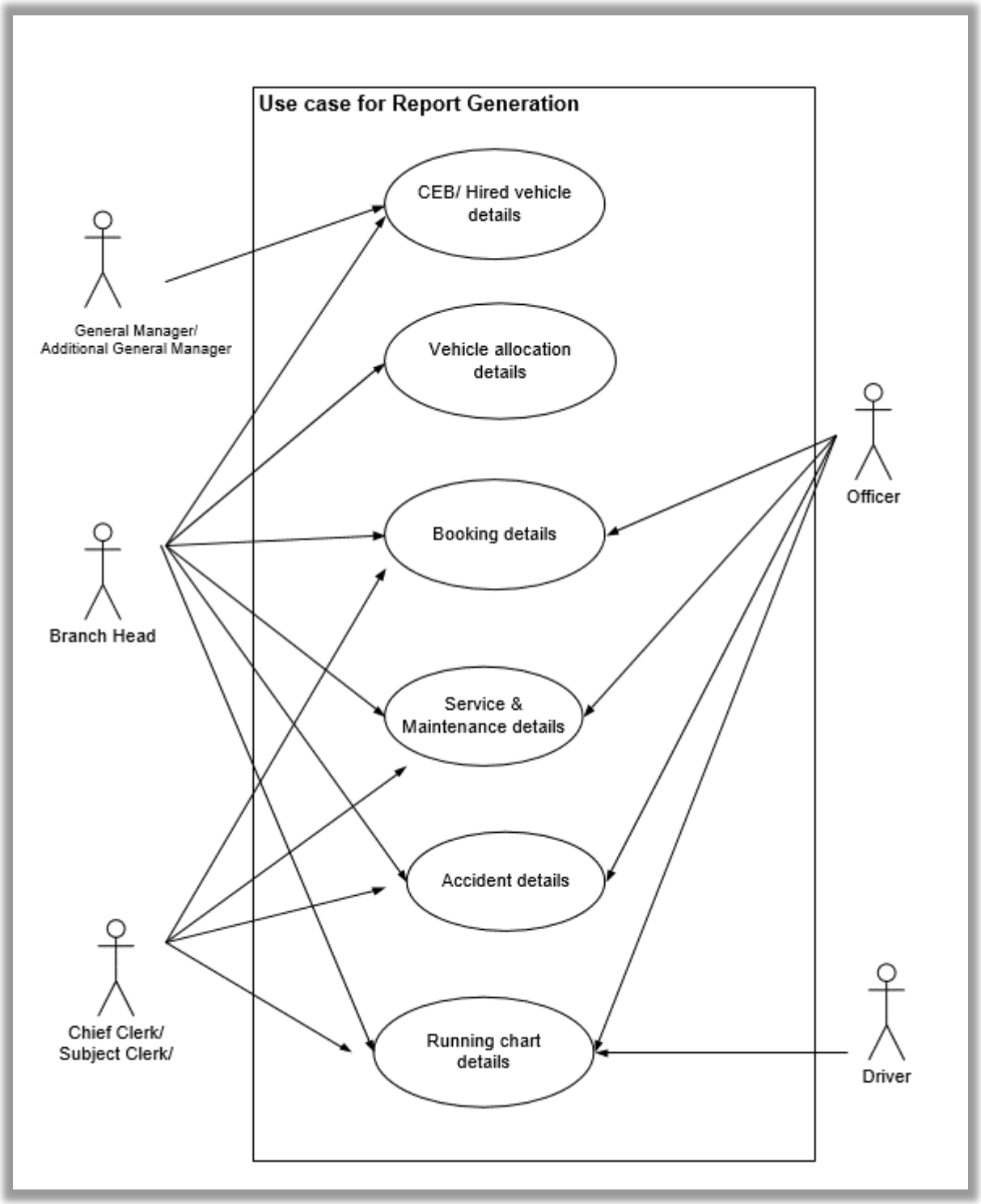


Figure 4.9: Report generation use case diagram

Use case Name	**Report Generation**
Users	Any user
Description	• Process of report generate
Precondition	• Create user profile • Create user login • Vehicle details already exist in the system
Typical Course of Events	• Vehicle details • Vehicle allocation details • Booking details • Service & maintenance details • Accidents details • Running chart details
Alternate Course	-
Post Condition	User can view reports.

Table 4.7: Use case description of Report generation

4.2. Sequence Diagrams

Sequence diagrams in UML is described interactions among classes in terms of an exchange of messages over time. This also called as event diagrams. This help to visualize and validate various runtime scenarios and predict how a system will behave and to discover responsibilities a class [11].

The following shows the sequence diagrams of vehicle registration (Figure 4.10), vehicle allocation (Figure 4.11) and, vehicle booking (Figure 4.12).

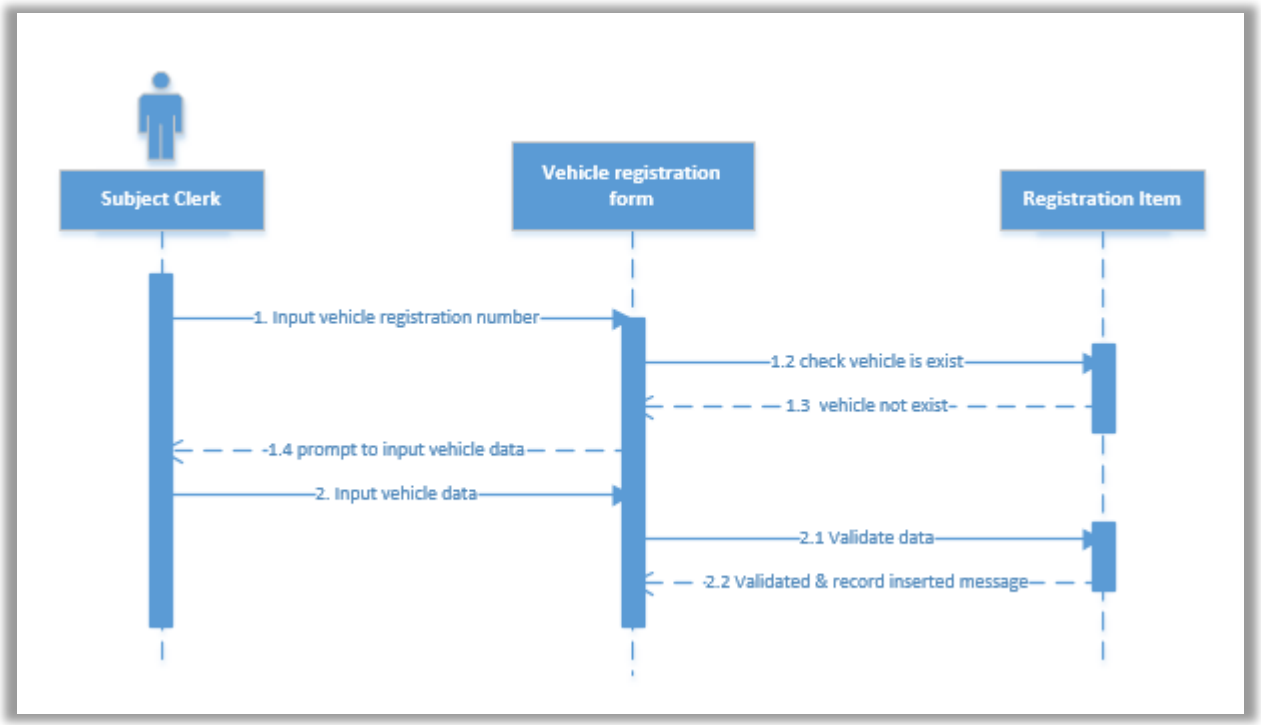


Figure 4.10: Sequence diagram of Vehicle Registration

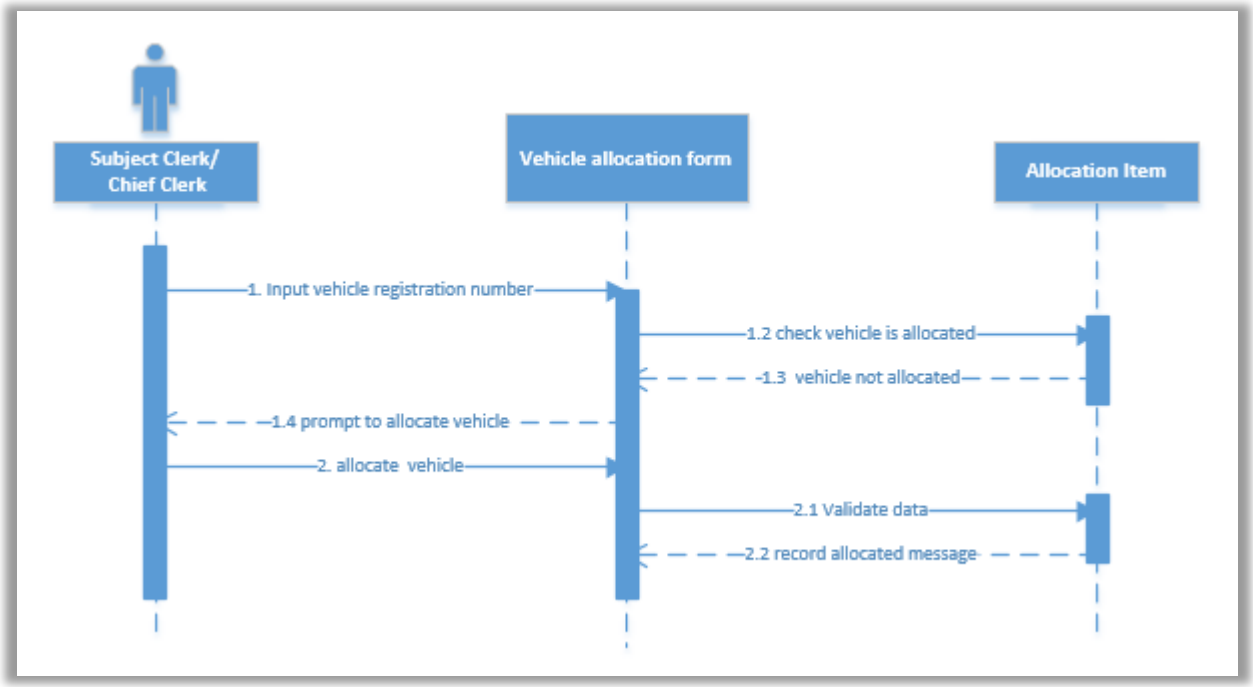


Figure 4.11: Sequence diagram of Vehicle Registration

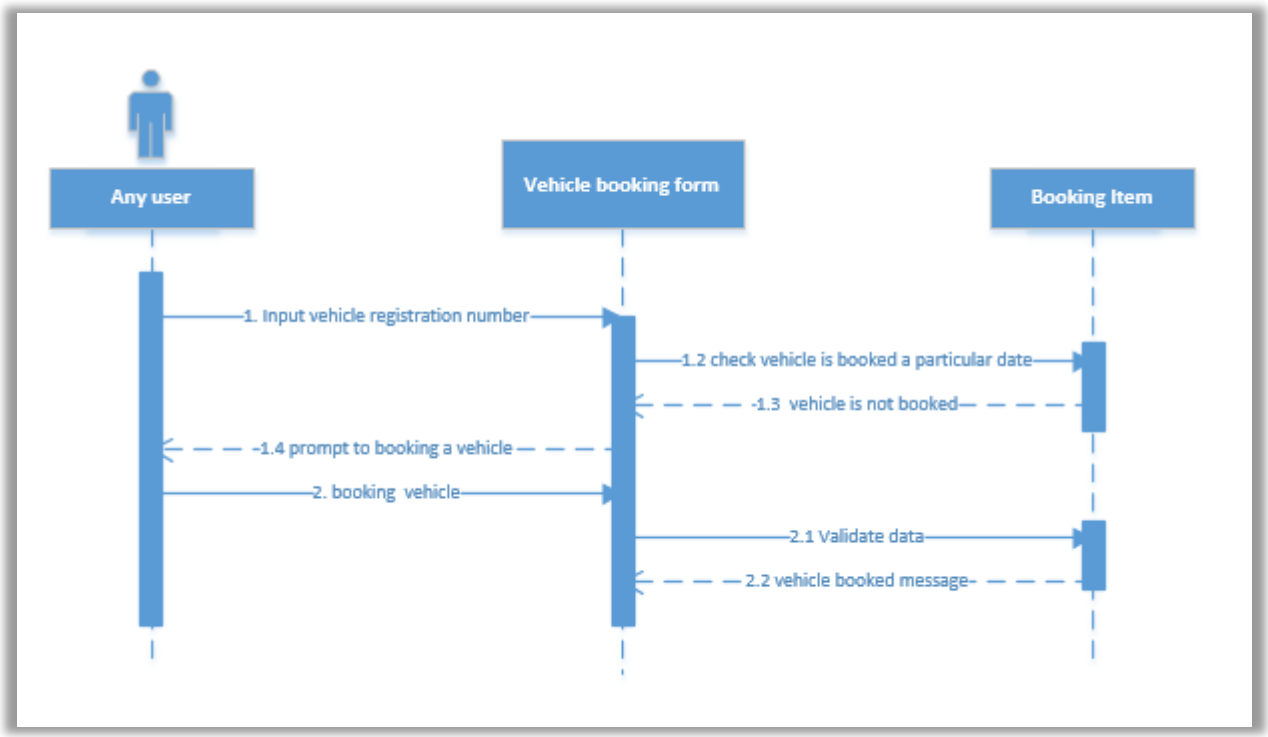


Figure 4.12: Sequence diagram of a Vehicle booking

4.3. Activity Diagrams

Activity diagrams in UML is a flowchart to represent the flow from one activity to another activity. Activity is an operation of the system and it captures the dynamic behavior [12].

Following activity diagrams show the process of CEB vehicle registration (Figure 4.13), hired vehicle registration (Figure 4.14) and vehicle transfer (Figure 4.15).

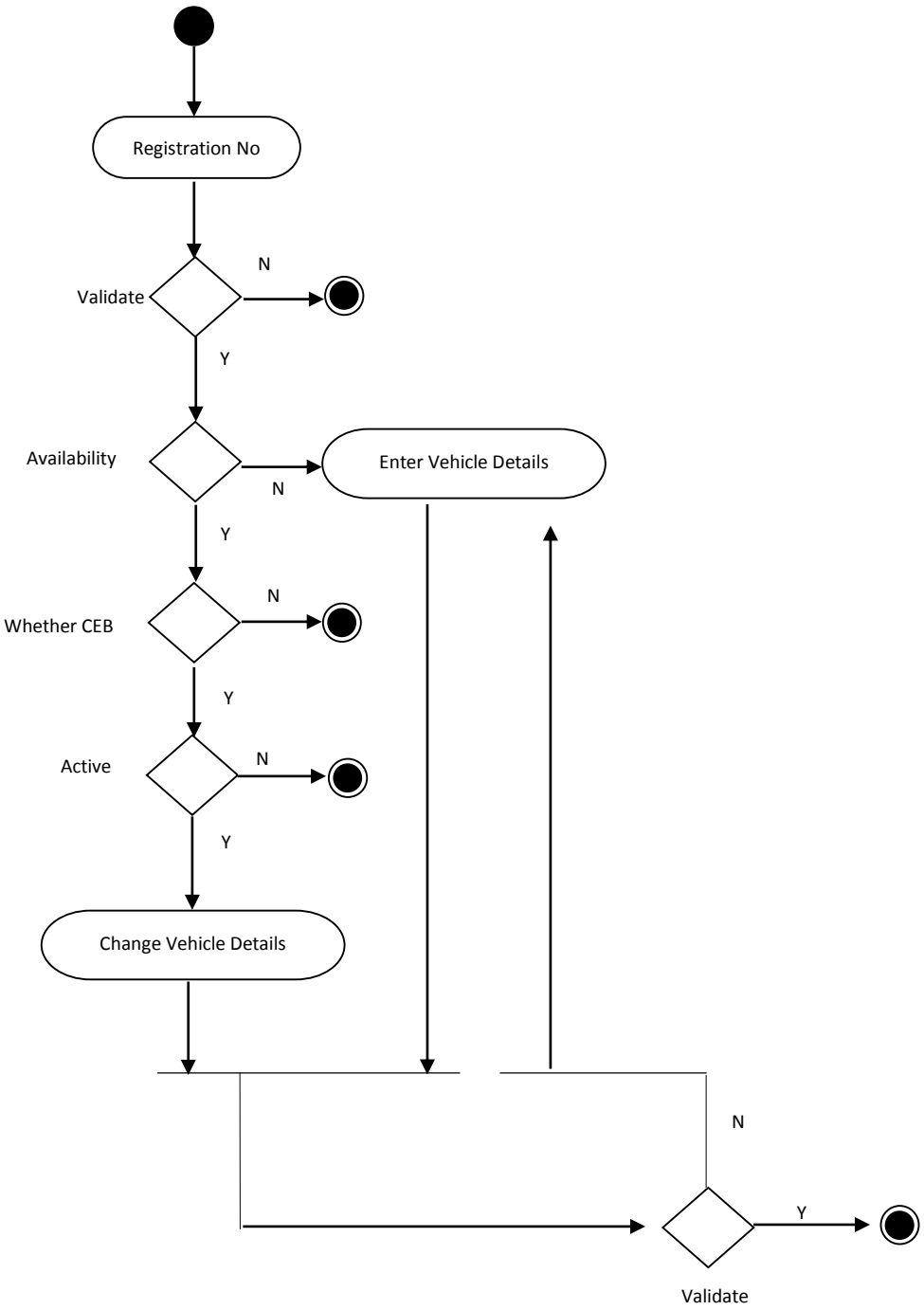


Figure 4.13: Activity diagram of CEB vehicle registration

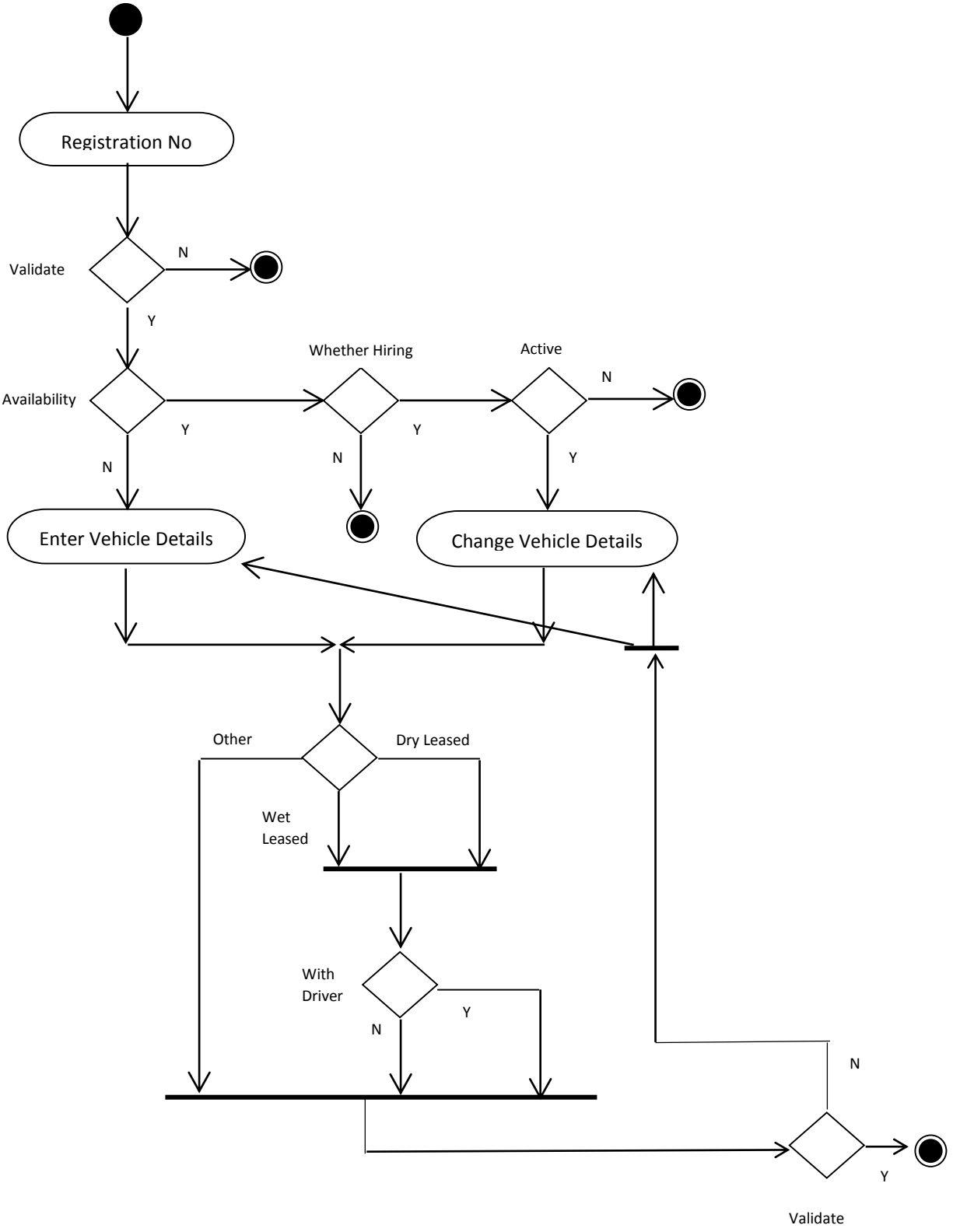


Figure 4.14: Activity diagram of Hired vehicle registration

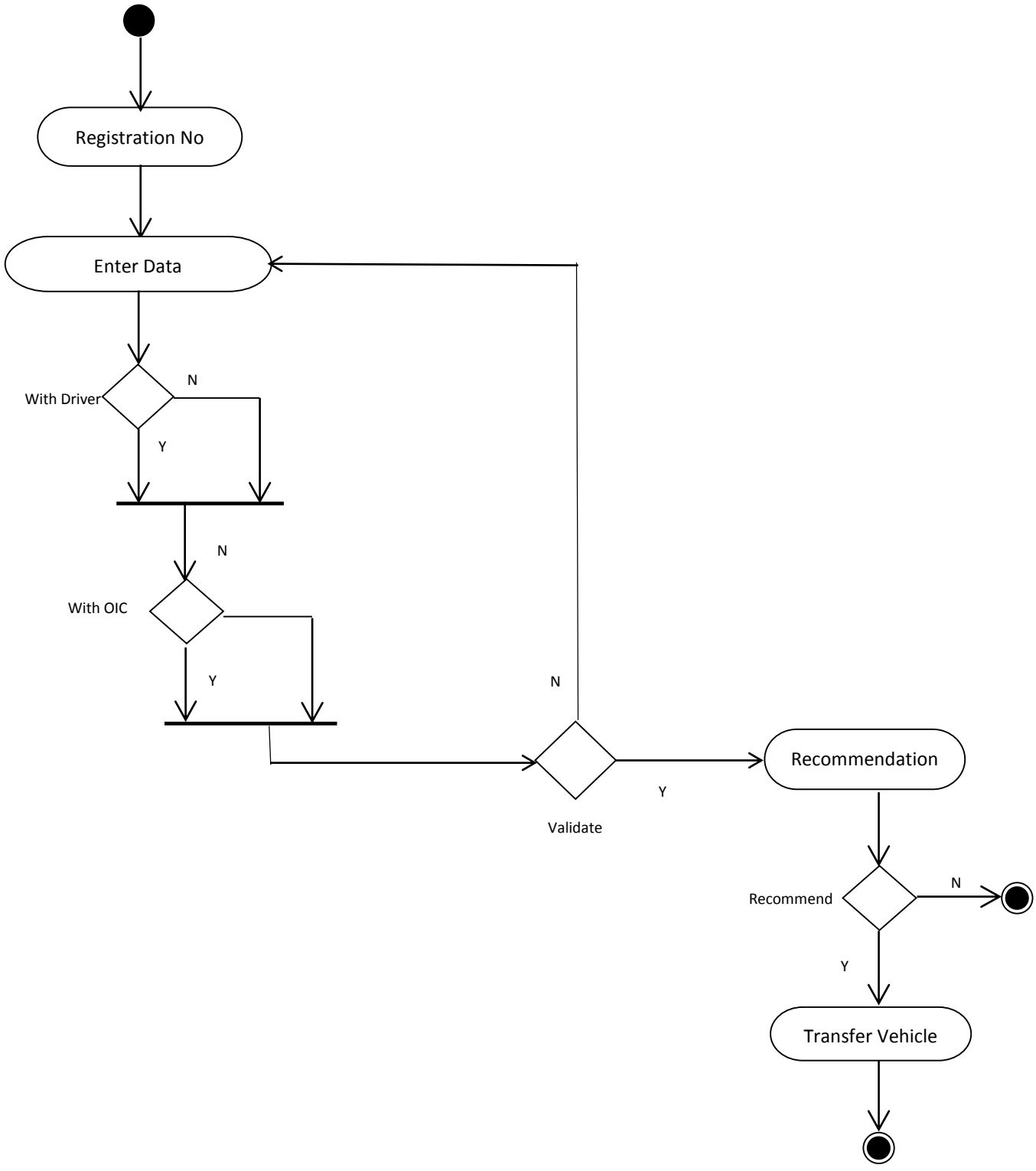


Figure 4.15: Activity diagram of Vehicle transfer

4.4. Class Diagrams

Class diagrams in Unified Modeling Language (UML) is used for construction of software applications and represents the static view of an application. Class diagram describes the attributes and operations of a class on the system. This is very useful for analysis and design of the static view of an application and describe the responsibilities of the system [13].

The following diagram show the database relationship of the proposed Fleet management system.

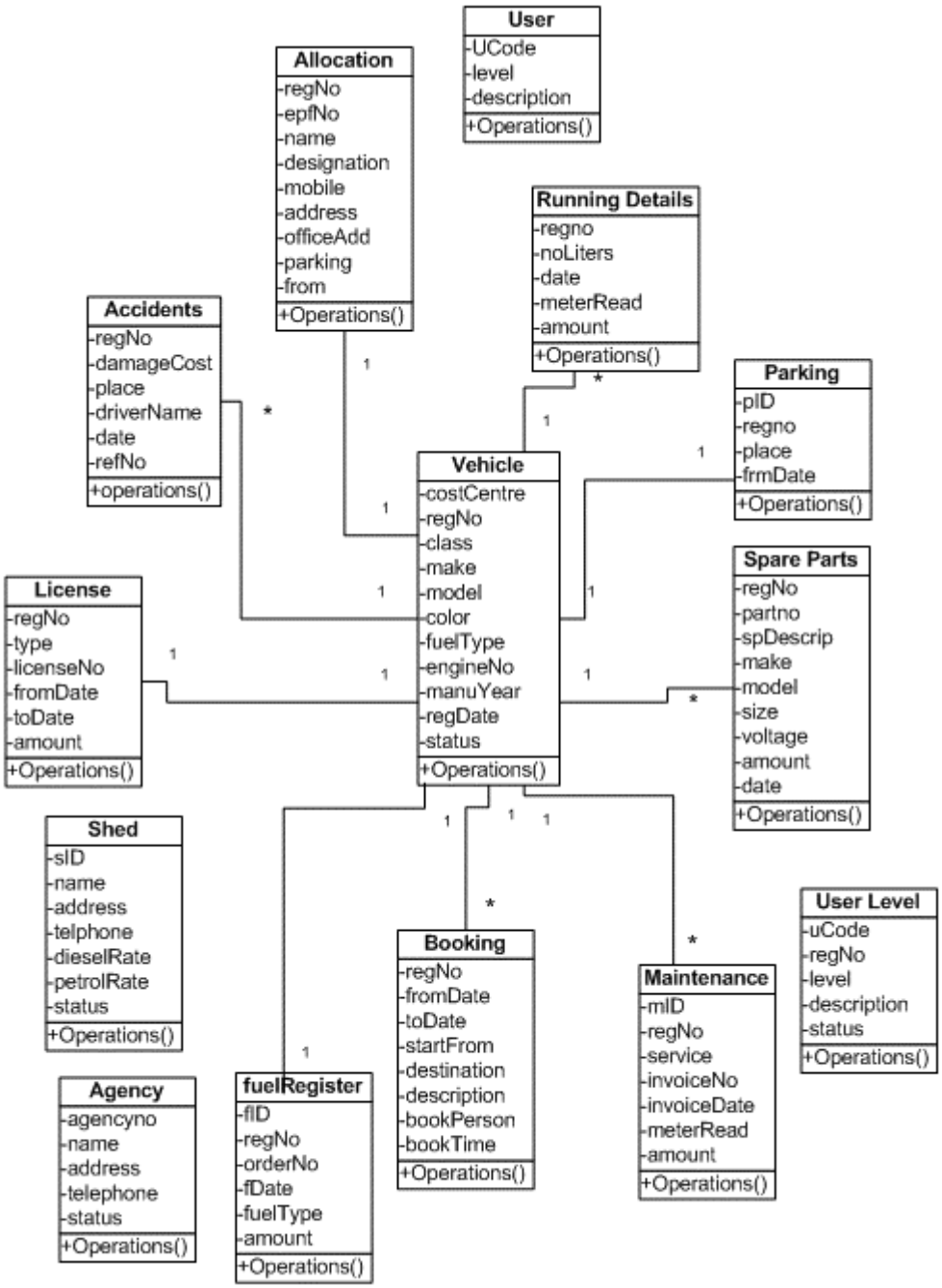


Figure 4.16: Class diagram of Fleet management system

4.5. User Interface Design

User interface (UI) design is a major part of the design process. It is the front-end application view to which user interacts in order to use the software. By using UI user can manipulate and control the software and hardware. UI can be graphical, text-based, audio-video based, or hardware and software combination or both [14].

User interface design is very important. If the design is not clear then cost of software development and training will be high. The steps of user interface design in proposed FMS is as follows.

- All interfaces are web form based
- All interfaces should have a meaningful heading
- Information should be clear, simple and understandable
- Interfaces should be attractive
- Items should be grouped in proper way (eg: option buttons, buttons, etc.)
- Error messages for incorrect data
- All the interfaces are designed according to the functional requirements.

Chapter 5 : Implementation

5. Introduction

This chapter describes the implementation of the proposed system.

5.1. Development Environment

The following software was used for the development for the proposed solution.

- Operating System: Microsoft Windows 8 and above
- Web server : Microsoft IIS
- Web browsers: Internet Explorer, Chrome
- Microsoft .NET Framework 4.5
- Microsoft Visual Studio 2015 Community Edition
- MySQL Workbench 6.0 Community Edition
- XAMPP

Microsoft ".NET Framework" is really a collection of Application Programming Interfaces (APIs) and use shared library of code that developers can call when developing applications. The shared library can perform all kinds of different functions. Instead of writing code, user can use code from the library performs that function. Using a framework of shared code it provides some standards between applications.

Microsoft visual studio 2015 Community Edition is the main programming language presently used in CEB. It is an open source free programming language. Source codes and tutorials are freely available in the internet. Visual studio is familiar version to anyone and can use simple code with live assistance for develop projects. The object structure of code easily with inline object browsing in the Solution Explorer and easily search files in the project.

In this study, MySQL workbench 6.0 community edition was used as the database. It was more flexible to installed and work with .net environment. Presently used in CEB and users are more familiar and it is a requirement of a client. MySQL act as a simple database implementation. It was easy to redesign, rewriting application code. This is an open source software. Using MySQL developers and DBAs can quickly and easily convert existing applications to run on

MySQL both on Windows and other platforms. Migration also supports migrating from earlier versions of MySQL to the latest releases.

XAMPP is an open source software and it is easy to handle and installed. Most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

Source code for .net connectivity with MySQL

con.ConnectionString = ("server=127.0.0.1; port=3306; Database=vms; Username=root;")
con.Open()

5.2. Implemented modules and source codes

In this section describes the description of the implemented modules including the source codes.

5.2.1. CEB vehicle registration

Add vehicle data into the system

Before entering the vehicle details, the system checked the validation of vehicle number. The following sample of the source code represents the vehicle number validation. Vehicle information are added into the system by subject clerk or chief clerk.

The sample code of vehicle information addition is shown in Figure 5.1.

```
  connection.Open()
   sql = "INSERT INTO vehicle_master (costcode, prov_code, reg_no, id, sid, v_class,
load_capacity, t_class, make, model, color1, color2, type_of_fuel, chassis_no,
engine_no, c_capacity, s_capacity, tyre_size, unladen_weight, gross_weight,
year_of_manu, usage,date_of_reg, date_of_purchase, purchase_price, type,
status,remarks, exist, enuserid, endatime) Values
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

                Dim com As OleDbCommand = New OleDbCommand(sql, connection)
                com.Parameters.AddWithValue("costcode", costcentre)
                com.Parameters.AddWithValue("prov_code", pcode)
                com.Parameters.AddWithValue("reg_no", regno)
                com.Parameters.AddWithValue("id", vid)
```

```
                com.Parameters.AddWithValue("sid", vsid)
                com.Parameters.AddWithValue("v_class", vcode)
                com.Parameters.AddWithValue("load_capacity", lccode)
                com.Parameters.AddWithValue("t_class", tclass)
                com.Parameters.AddWithValue("make", mcode)
                com.Parameters.AddWithValue("model", model)
                com.Parameters.AddWithValue("color1", ccode1)
                com.Parameters.AddWithValue("color2", ccode2)
                com.Parameters.AddWithValue("type_of_fuel", fcode)
                com.Parameters.AddWithValue("chassis_no", chassis)
                com.Parameters.AddWithValue("engine_no", engineno)
                com.Parameters.AddWithValue("c_capacity", cylinder)
                com.Parameters.AddWithValue("s_capacity", seating)
                com.Parameters.AddWithValue("tyre_size", tyre_size)
                com.Parameters.AddWithValue("unladen_weight", unladen)
                com.Parameters.AddWithValue("gross_weight", grossweight)
                com.Parameters.AddWithValue("year_of_manu", manuyear)
                com.Parameters.AddWithValue("usage", ucode)
                If regdate1 = "" Then
                    com.Parameters.Add("date_of_reg", OleDbType.Char).Value = regdate1
                Else
                    com.Parameters.Add("date_of_reg", OleDbType.Date).Value = regdate
                End If

                If purchasedate1 = "" Then
                    com.Parameters.Add("date_of_purchase", OleDbType.Char).Value =
purchasedate1
                Else
                    com.Parameters.Add("date_of_purchase", OleDbType.Date).Value =
purchasedate
                End If
                com.Parameters.AddWithValue("purchase_price", purchaseprice)
                com.Parameters.AddWithValue("type", vtype)
                com.Parameters.AddWithValue("status", "C")
                com.Parameters.AddWithValue("remarks", remarks)
                com.Parameters.AddWithValue("exist", "Y")
                com.Parameters.AddWithValue("enuserid", UID)
                com.Parameters.AddWithValue("endatime", CurrentDate)

                com.ExecuteNonQuery()
                lblMessage.Visible = True
                lblMessage.Text = "Record Inserted......"
                connection.Close()
```

Figure 5.1: Sample code of vehicle information addition

Other selected source code segments related to this system is provided in Appendix B.

Microsoft Visual Studio 2015 Community Edition and MySQL Workbench 6.0 Community Edition used for the entire system. The entire selected code segments are included in Appendix B and described their importance.

Chapter 6 : Testing, Evaluation and Results

6.1.Testing

6.1.1. Introduction

A proper test plan is needed for software testing. In this stage, all possibilities have been checked and errors were identified and corrected.

6.1.2. Module structure

The following Figure 6.1 shows the basic module structure of the system.

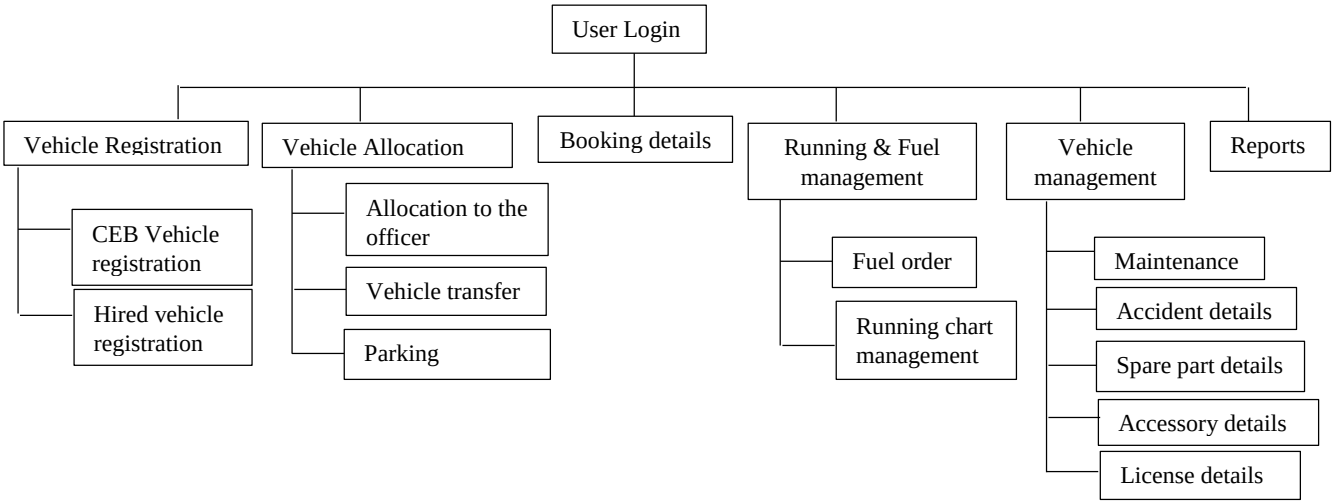


Figure 6.1: Module structure of the system

6.1.3. Test Plan

The complete system tested is based on the test plan. It is mainly targeted to user interface and validation according to the CEB system requirement.

Prepare test cases – This is based on customer requirement specification

Prepare test matrix – This maps test cases to respective requirement.

Creating test data – Test data create according to requirement.

Executing test cases – Execute test case and the result is documented.

The sampling test case format is as follows.

Test	Test case	Expected result	Actual result

Table 6.1: Sampling test case format

Unit Testing – The first test in the development process is the unit test. The source code is manually divided into several modules. These each module have a specific function. Unit test ensure that each module perform the function correctly to the documented specification.

System Testing – Once all the modules are integrated, system testing is done at this stage. This testing ensures that all the entire integrated points meet requirement.

Alpha & Beta Testing – These tests are done under the real usage of customer environment.

Acceptance Testing – This is the final stage of the testing process. This testing was done in live environment after releasing the customer site.

6.1.4. Functions to be tested

The following functions are tested in this section.

- Login process
- CEB vehicle registration
- Vehicle allocation
- Vehicle transfer
- Vehicle parking
- Fuel request management
- Running chart management
- Accident details
- Reports

6.1.5. Testing process overview

The steps of the testing process overview as follows.

- Understanding requirement
- Prepare test cases
- Creating test data
- Executing test cases

6.1.6. Software testing process

The basic steps taken of the testing process as follows. Software testing process is an iterative process and all the testing were done by according to the test plan.

- All the test cases were defined.
- Run test cases. The process will be run, until the errors found. Errors are addressed in the next process of the system.
- All successful and failure test cases should be mentioned in the test cases. In the test case the nature of the failure should be clearly mentioned.
- Failure test cases should be corrected, tested & mentioned.

6.1.7. Testing in the project lifecycle

Once the test plan is created, unit test was started as starting point. Integration testing can be performed after unit test of the separate modules. It is based on the functional specification of the software. After this separate module testing, system testing was done. It included functional and non-functional requirements. Acceptance testing in the system was done after implementation of customer site. Customer can analyze the system and accept to the test plan. When entering data into the system, alpha & beta testing were conducted in this stage.

Validation messages

Login Page

Login Id should be EPF number of the each employee. EPF number should contain 6 digits. Figure 6.2 represented the user id validation message.

Figure 6.2: Invalid User Id message

Vehicle Registration

All vehicle numbers are checked according to the following format. In the number formats X denotes alphabetical character and 9 denotes as number. All validations are done by according to the format.

- XX9999 (ex: NB5698)
- XXX9999 (ex: CAT5698)
- 99-9999 (ex: 53-5698)
- 999-9999 (ex: 253-5698)
- 99SRI9999 (ex: 56SRI5698)

Data entry enter form (Vehicle addition) validation messages are shown in the following Figure 6.3 and Figure 6.4.

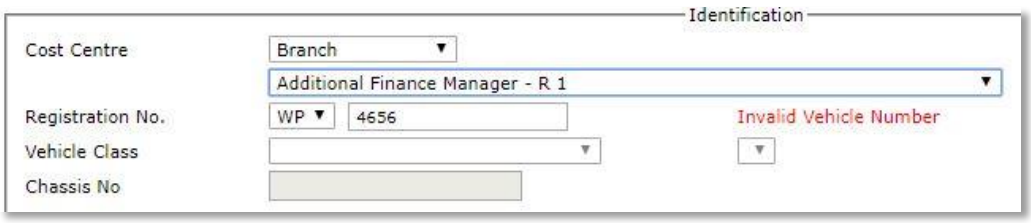


Figure 6.3: Invalid vehicle number message

This Figure 6.4 represented the date validation for vehicle parking location.

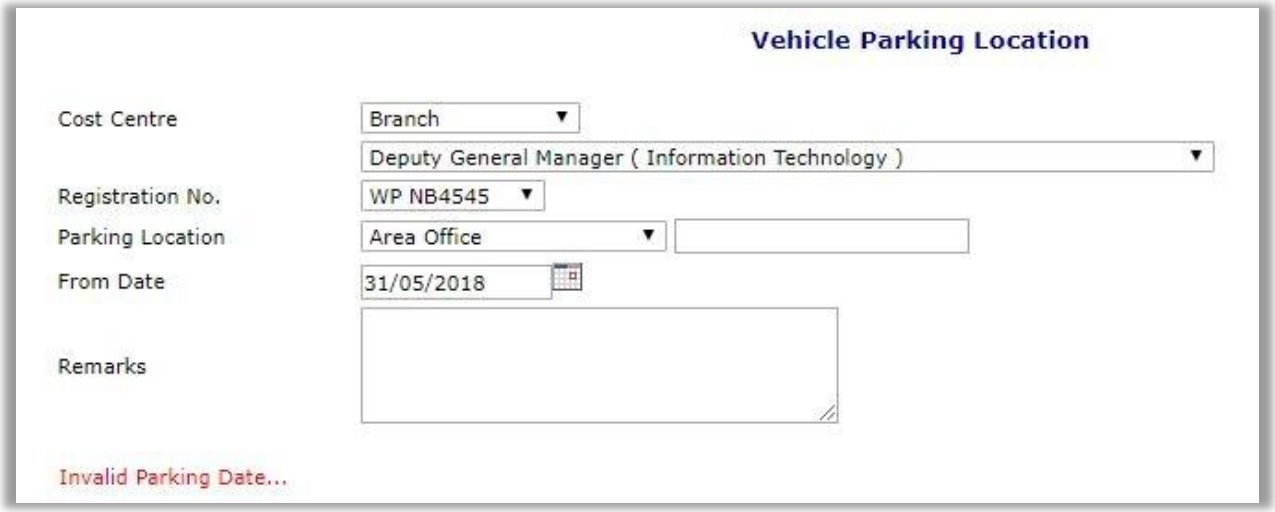


Figure 6.4: Invalid date message

Successful messages

After all data validation, record is inserted or updated into the database.

The Figure 6.5 and Figure 6.6 represented the "Data updated" and "Data Insert" message.

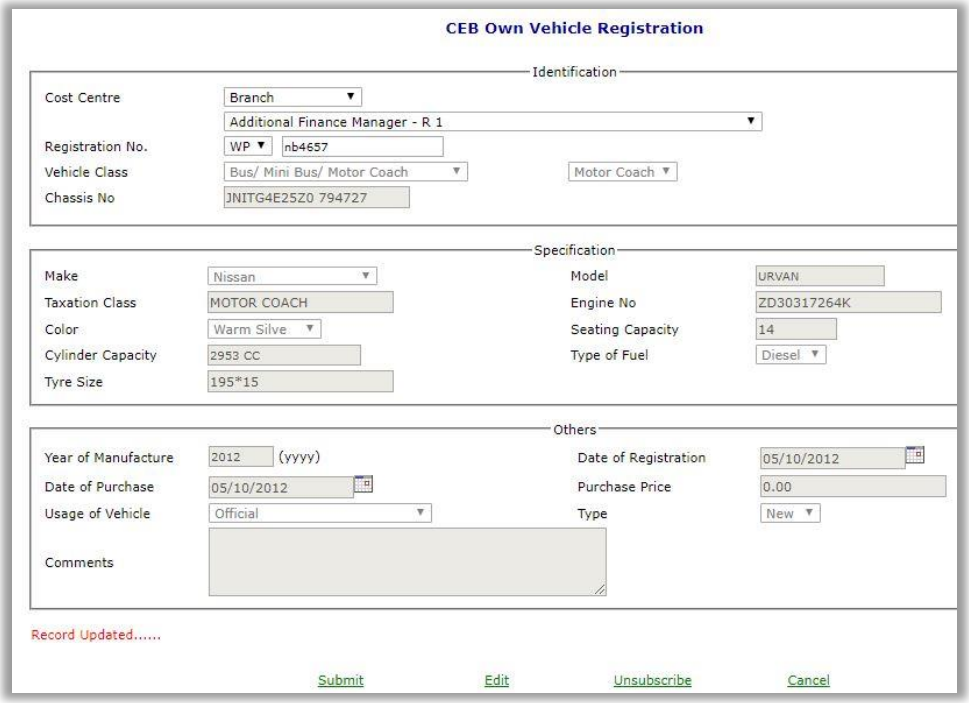


Figure 6.5: Data Updated message

Figure 6.6: Data Insert validation message

6.1.8. Test Cases

Test case included test, test case, test data, expected result and actual result. Test data were being carried out throughout the test. After executing, the expected result checked with the actual result and actual result was filled after executing the test.

Test case for Login

The login test cases are displayed below Table 6.2.

Test	Test Case	Expected result	Actual result
Login by user	Enter correct user name and password	Login to the system successfully	Login to the system successfully
	Enter incorrect username or password	Username or password invalid message	Username or password invalid message
	Username or password is empty	Both user name & password empty message	Both user name & password empty message

Table 6.2: Test case for login

Test case for CEB vehicle registration

The test cases for CEB vehicle registration are displayed below Table 6.3.

Test	Test Case	Expected result	Actual result
Add vehicle details	Enter vehicle registration number	Enable to enter vehicle details	Enabled to enter vehicle details
	Click “Submit” button without selecting	Message should display “Class cannot be blank…”	Message displayed “Class cannot be blank…”

	vehicle class from dropdown		
	Click "Submit" button without entering vehicle manufacture year	Message should display "Manufacture year cannot be blank..."	Message displayed "Manufacture year cannot be blank..."
	Fill all vehicle details and press "Submit" button	Message should display "Record inserted..."	Message displayed "Record inserted..."
Edit vehicle details	Enable to edit vehicle details	Should enable to enter all vehicle details	Enabled to enter all vehicle details
Delete vehicle details	Click "Unsubscribe" button to behave as inactive record.	Message should display "Record deleted..."	Message displayed "Record deleted..."
Cancel vehicle details	Click "Cancel" button to add new record	Reset all textboxes and dropdowns except cost center and vehicle registration	Reset all textboxes and dropdowns except cost center and vehicle registration

Table 6.3: Test case for CEB vehicle registration

Other test cases are presented in Appendix C.

6.2. Evaluation

Acceptance testing is performed by the customer after system testing. The evaluation is useful for this system which is suited for the CEB. This measures the customer satisfaction of the implemented system. In this evaluation, following users were used to enter data. There were different 5 types of users access the system with different login permission. Most of them were data entry operators and staff members. The selected candidates answered the questionnaire.

The following Table 6.4 shows the questionnaire of the user evaluation.

Fleet Management System

Questionnaire for user evaluation

Please respond the each of followings.

1 – Entire CEB 2 – Divisional level 3 – Only Branch 4 – Only Province

S – Super User

User category

1	2	3	4	S

1 – Very poor 2 – Poor 3 – Satisfy 4 – Good 5 – Excellent

id	Functionality	Rating				
		1	2	3	4	5
1	Is the system concerned on major functionalities?					
2	Is the system relevant to your work?					
3	Do you get benefits other than manual work?					
4	Is the system functionalities enough to your work?					
5	Are the system user interfaces help to understand quickly?					
6	Are the system design is help to understand quickly?					
7	Are the system functionalities helpful to your work?					
8	Can effectively complete your work using this system?					
9	Is it feel comfortable using this system?					
10	Are the system give error messages that clearly how to fix problems?					
11	When you make mistakes, can recover easily and quickly?					
12	Is the information provided for the system easy to understand?					
13	Is the organization of information on the system screen is clear?					

Continued next page

Continued previous page

14	Is the interface of this system is pleasant?					
15	Overall rating to the system					
16	Is the transaction time reasonable?					
	Remarks					

Table 6.4: Questionnaire for user evaluation

6.2.1. Evaluation methodology

In the above questionnaire (Table 6.4), there is a rating method which is range from 1 to 5. The rating value 5 is the highest and the rating value 1 is the lowest. User can evaluate the system in various ways as its functionality, appearance, usability etc.

Average score:

$$S = \frac{\sum_{i=1}^{n} S_i}{n}$$

S - Score value for each question: $0 \leq S \leq 5$

n – Number of questions related to the functionality

Explore

This is a function of SPSS. The produces detailed univariate statistics and graphs for numeric scale variables.

6.2.2. Evaluation analysis

During the analyzing of the evaluation, the total average of the above functionalities of the questionnaire was weighted between rating value 3 to 4. Following shows the evaluation analysis using Explore function in SPSS.

Case Processing Summary

		Cases					
		Valid		Missing		Total	
	id	N	Percent	N	Percent	N	Percent
rate	1	27	100.0%	0	.0%	27	100.0%
	2	25	100.0%	0	.0%	25	100.0%
	11	23	85.2%	4	14.8%	27	100.0%

Figure 6.7: Case Processing Summary Table

In the case processing table (Figure 6.7), shows the valid values and missing values of the sample. Id represented the id number of the above questionnaire.

- Id 1 – Is the system concerned on major functionalities?
- Id 2 – Is the system relevant to your work?
- Id 11- When you make mistakes, can recover easily and quickly?

In descriptive table (Figure 6.8), it shows the mean value of the id 1 is 3.93, id 2 is 3.64 and id 11 is 3.09.

Descriptives

	id			Statistic	Std. Error
rate	1	Mean		3.93	.140
		95% Confidence Interval for Mean	Lower Bound	3.64	
			Upper Bound	4.21	
		5% Trimmed Mean		3.92	
		Median		4.00	
		Variance		.533	
		Std. Deviation		.730	
		Minimum		3	
		Maximum		5	
		Range		2	
		Interquartile Range		1	
		Skewness		.116	.448
		Kurtosis		-1.013	.872
	2	Mean		3.64	.181
		95% Confidence Interval for Mean	Lower Bound	3.27	
			Upper Bound	4.01	
		5% Trimmed Mean		3.66	
		Median		4.00	
		Variance		.823	
		Std. Deviation		.907	
		Minimum		2	
		Maximum		5	
		Range		3	
		Interquartile Range		1	
		Skewness		-.272	.464
		Kurtosis		-.514	.902
	11	Mean		3.09	.153
		95% Confidence Interval for Mean	Lower Bound	2.77	
			Upper Bound	3.40	
		5% Trimmed Mean		3.10	
		Median		3.00	
		Variance		.538	
		Std. Deviation		.733	
		Minimum		2	
		Maximum		4	
		Range		2	
		Interquartile Range		1	
		Skewness		-.139	.481
		Kurtosis		-1.008	.935

Figure 6.8: Descriptive table of the Explore function

Histograms

Histograms provide a visual interpretation of numerical data by indicating the number of data points that lie within a range of values.

The following histograms (Figure 6.9, Figure 6.10, and Figure 6.11) show the frequency of the categorical data against in id.

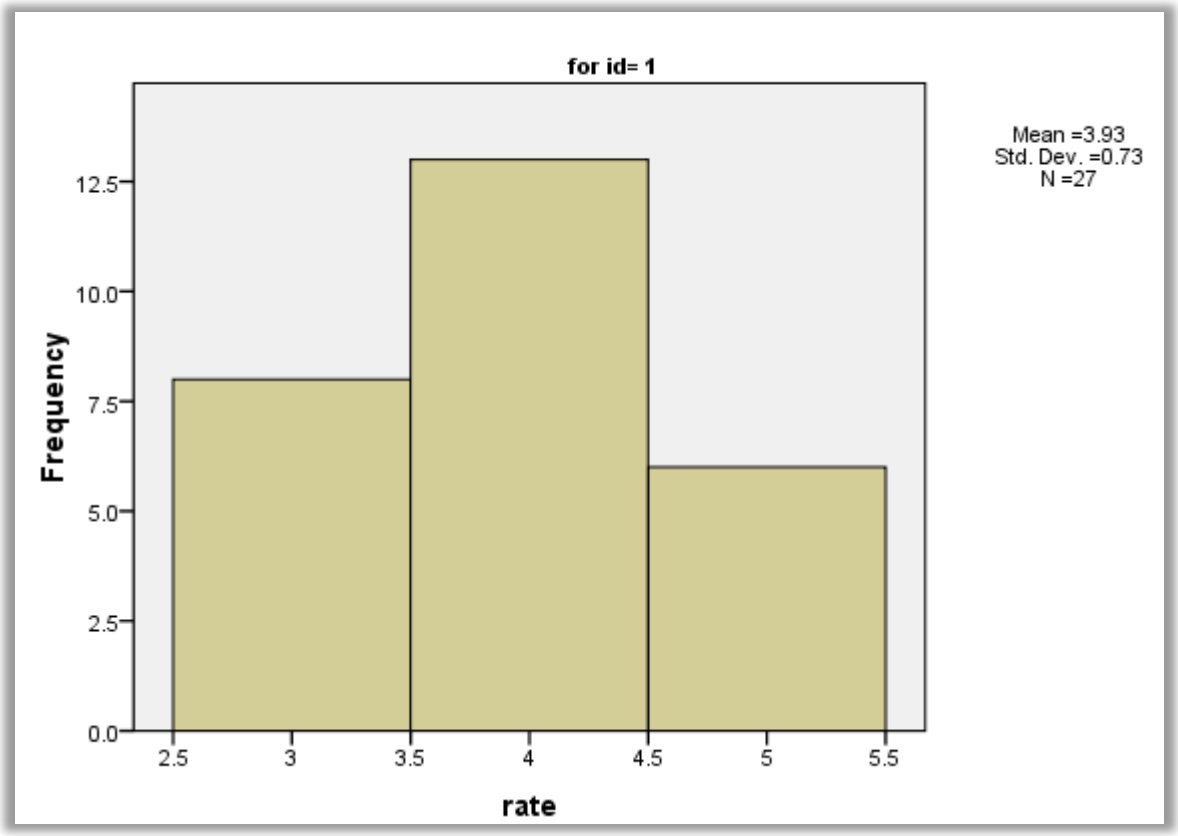


Figure 6.9: Histograms of id 1

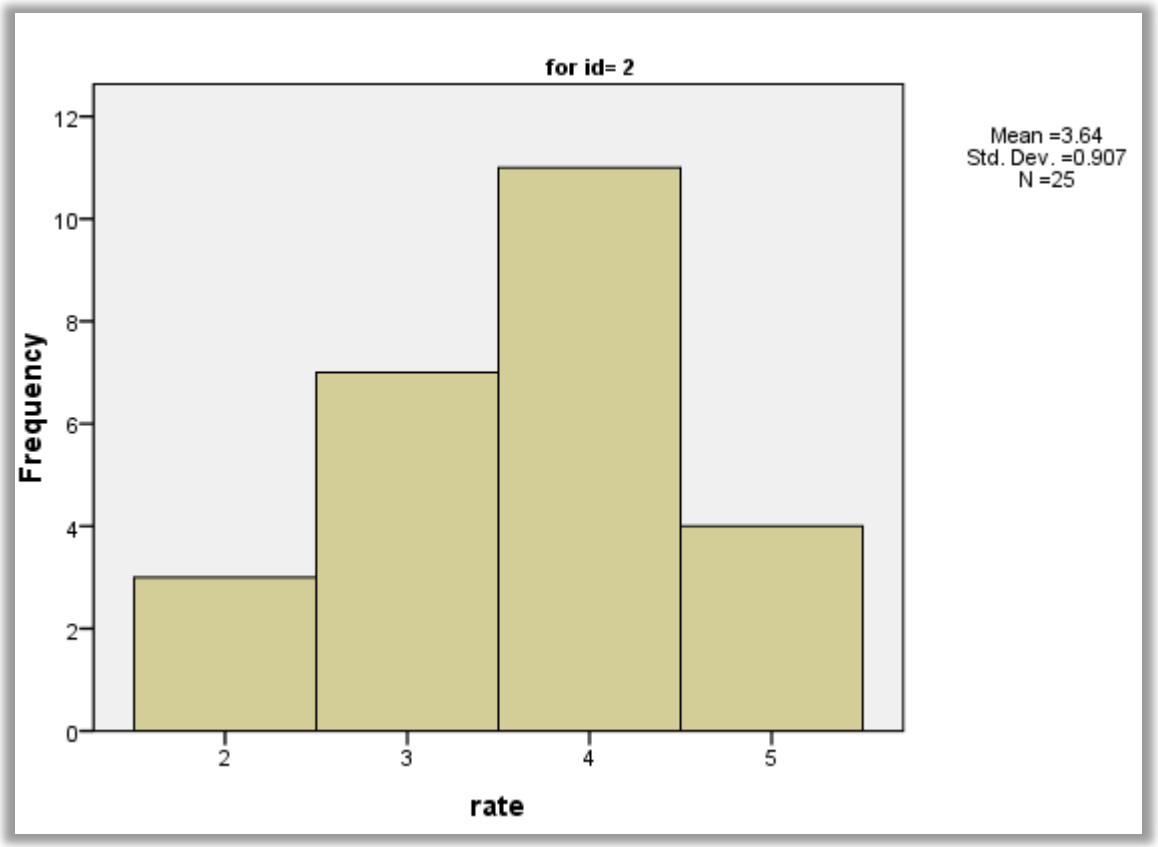


Figure 6.10: Histograms of id 2

Box – plot

The box-plot representing distribution of feedback of the functionality. The minimum rate is 2 and maximum rate is 5. The following Figure 6.11 presented the distribution of selected id's in given above questionnaire. The X-axis presented the id and Y-axis presented the rate of the id.

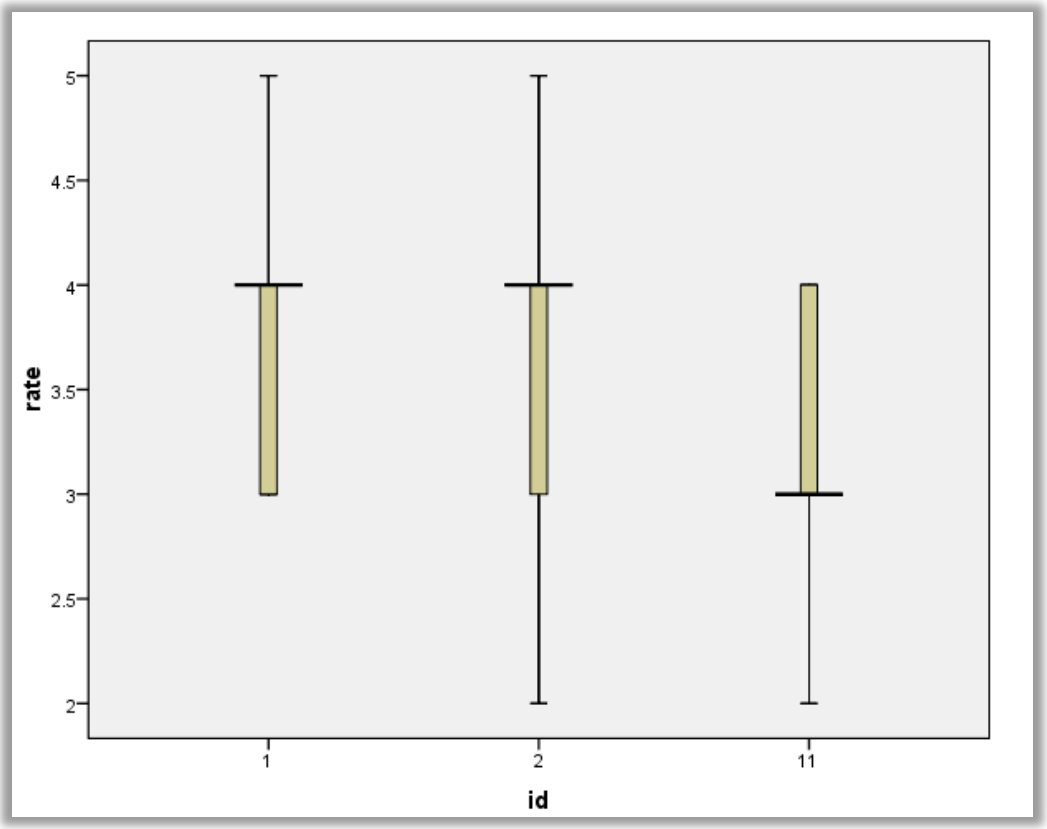


Figure 6.11: Box-plot representation of selected id's

6.3. Results

Administrator, staff officers and subject clerks can view reports. The following reports can generate data as Summary of vehicle expenditure (Figure 6.12 and Figure 6.13). This shows records in detailed view and graphical representation.

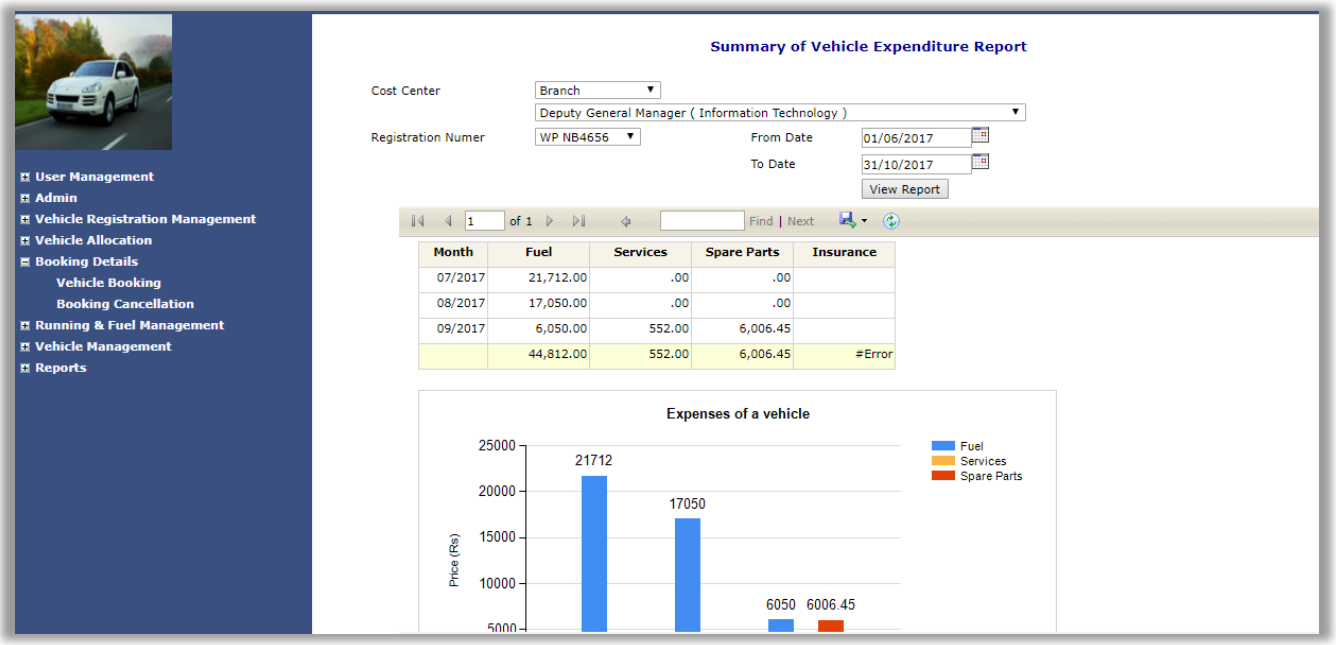


Figure 6.12: Summary of vehicle expenditure report

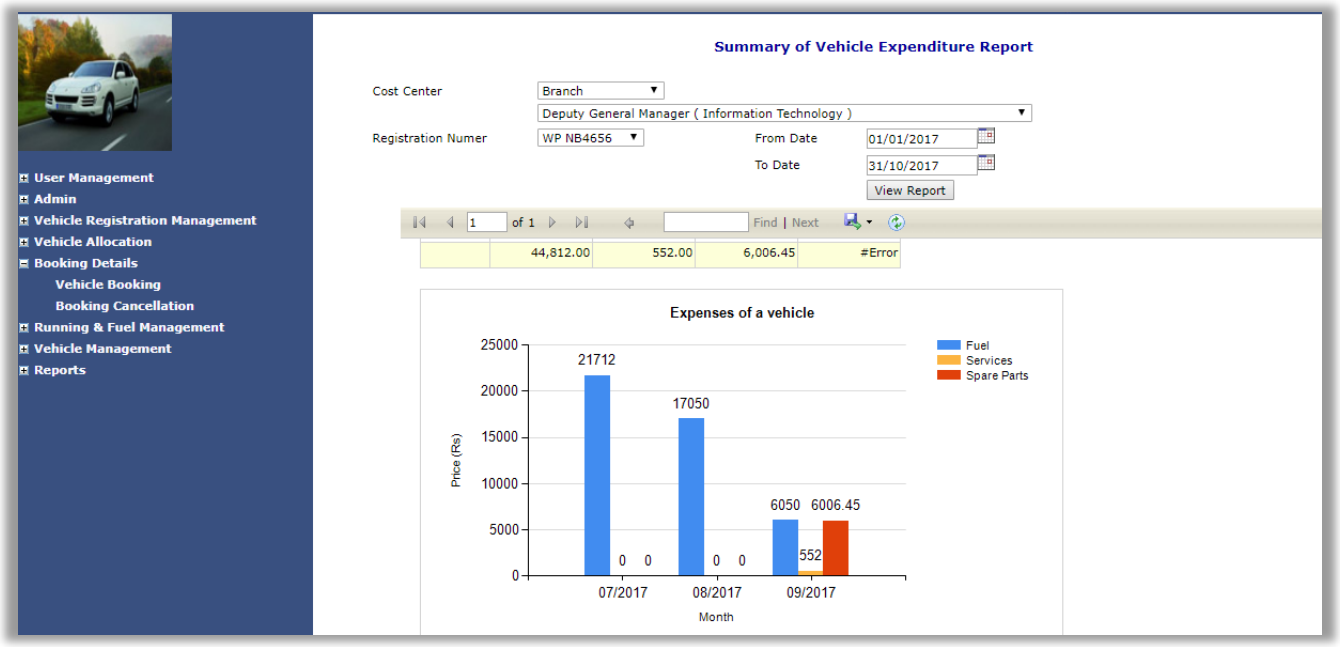


Figure 6.13: Summary of vehicle expenditure in graphical representation

Other imported results are provided in Appendix E.

Chapter 7 : Discussion

7. Introduction

Fleet management system was successfully completed, all the objectives of the system achieved. Almost all the functional requirements were completed.

All the required functions were implemented according to the customer needs. The system was verified & validated.

7.1. Achievements of Objectives

This study has been solved many issues faced in CEB. Using this system, CEB management could be identified vehicle details, number of vehicle in a cost center, division wise vehicle details and total expenses. Furthermore this could be provided the facilities such as identification of vehicle location, allocation to officer(s), vehicle transfer history, vehicle booking, monitor the revenue license date etc. Also possible to determine the number of excess or shortage of vehicles in unit wise, sub unit wise.

In addition to the above functionalities, the system is provided following reports.

- Details of fuel consumption & running chart of a vehicle
- Summary mileage of each vehicle
- Services and maintenance
- Accident details.

The main achievement of this project is to generate information in connection with the decision making activities in the followings.

- To determine the maintenance cost whether is it high or low.
- To determine whether fuel consumption is economical or not.
- To provide the recommendation whether the vehicle is further maintenance or dispose.
- To provide statistics details like total fuel consumption, maintenance cost (unit wise, sub unit wise, etc.) and required information for the next year budget estimation.

7.2. Lesson learnt

The entire project was a lesson to the developer. Developer learnt how to gather requirements, analyze, design and implementation large organizations such as CEB. It was very difficult to gather requirements because there are 10 divisions under CEB that functions separately and different operation (ex: Distribution divisions, generation, projects, etc).

User Interface simplification is very important to CEB. Developer was very concern about the user interface simplification due to number of software system trainings (other than FMS). This study was mainly targeted to insert data to the system and generate reports with graphical representation.

The system has connect with MySQL database to insert data, retrieve data, and transfer data, etc. When implementing the system, design and implement the CEB structure and handling cost centers was learnt. It was done at the very first stage. Also user identification and user roles were defined at different levels.

The developer learnt how to analyze the data and view information in statistically. It was mainly highlighted to view information and get the decisions very easily.

7.3. Future improvements

The user interfaces could be improved. In order to improve this study, the following identified areas are also be developed.

- Payments handling of a vehicle
- Vehicle disposal
- GPS (Global Positioning System) tracker
- Mobile/Android application
- CEB letters & forms
- CEB vehicle usage related circulars

When booking a vehicle, sending a SMS message to driver is accepted. And also better to handle minimum flow paths for letter approvals. Error and warning messages displayed by this study could be improved. The system should be improved whether database connection is available or offline.

References

[1] "CEB," Ceylon Electricity Board, [Online]. Available: http://www.metroceb.lk/. [Accessed 4 2017].

[2] "Tutorial Points," 2017. [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm. [Accessed 06 2017].

[3] "Wikipedia," July 2017. [Online]. Available: http://en.wikipedia.org/wiki/Incremental_build_model. [Accessed 14 July 2017].

[4] "Vehicle Fleet and Asset Management Solutions," Vinitysoft Business Solutions, 2017. [Online]. Available: www.vinitysoft.com/fleet-management-software-4-0/. [Accessed 05 2017].

[5] "Fleet Maintenance Pro," Innovative Maintenance Systems, 1994 - 2017. [Online]. Available: http://www.mtcpro.com. [Accessed 05 2017].

[6] "Fleetio," 2012. [Online]. Available: https://www.fleetio.com. [Accessed 06 2017].

[7] D. Istrefi and B. Çiço, "Fleet Management Solution," *Fuel consumption and collision prevention system modules,* Vols. 3,No.3, no. June, 2013.

[8] W. G. Wenzel, "Using Fleet Management to Make and Save Money," *Introduction to the Basis of Fleet Management Technology,* pp. 1-18, 2016.

[9] Bailey, Lydon, Davern, *Best practice fleet management and priority actions,* no. 28th - 30th August, Brisbane, Queensland., 2013.

[10] "UML - Use Case Diagrams," tutorialspoint, 2017. [Online]. Available: https://www.tutorialspoint.com/uml/uml_use_case_diagram.htm. [Accessed August 2017].

[11] "Sequence Diagram," SmartDraw software, LLC, 1994-2017. [Online]. Available: https://www.smartdraw.com/sequence-diagram/. [Accessed August 2017].

[12] "UML - Activity Diagrams," tutorialspoint, 2017. [Online]. Available: https://www.tutorialspoint.com/uml/uml_activity_diagram.htm. [Accessed August 2017].

[13] "UML - Class Diagram," tutorialspoint, 2017. [Online]. Available: https://www.tutorialspoint.com/uml/uml_class_diagram.htm. [Accessed August 2017].

[14] "Software User Interface Design," tutorialspoint, 2017. [Online]. Available: https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm. [Accessed Aug 2017].

Appendix A– Database Design

1. CEB vehicle registration

The Figure A.1 shows the CEB vehicle table that was design to enter all CEB vehicle details.

Table Name: cebvehicle Schema: vms

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
costCode	VARCHAR(6)	☐	☑	☐	☐	☐	☐	☐	
provCode	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	NULL
regNo	VARCHAR(12)	☑	☑	☐	☐	☐	☐	☐	
classId	INT(11)	☐	☐	☐	☐	☐	☐	☐	NULL
vClass	VARCHAR(6)	☐	☐	☐	☐	☐	☐	☐	NULL
tClass	VARCHAR(30)	☐	☐	☐	☐	☐	☐	☐	NULL
make	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL
model	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	NULL
color	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL
typeFuel	VARCHAR(7)	☐	☐	☐	☐	☐	☐	☐	NULL
chassisNo	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	NULL
engineNo	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	NULL
cCapacity	VARCHAR(10)	☐	☐	☐	☐	☐	☐	☐	NULL

Figure A.1: CEB vehicle table

2. Vehicle allocation table

The Figure A.2 shows the vehicle allocation table that was design to enter all vehicle allocation details.

Table Name: vehicle_allocation Schema: vms

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	INT(11)	☑	☑	☐	☐	☐	☐	☑	
provCode	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	NULL
regNo	VARCHAR(12)	☐	☐	☐	☐	☐	☐	☐	NULL
officerName	VARCHAR(60)	☐	☐	☐	☐	☐	☐	☐	NULL
designation	VARCHAR(50)	☐	☐	☐	☐	☐	☐	☐	NULL
type	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL
isPool	VARCHAR(1)	☐	☐	☐	☐	☐	☐	☐	NULL
remarks	VARCHAR(50)	☐	☐	☐	☐	☐	☐	☐	NULL
fromDate	DATE	☐	☐	☐	☐	☐	☐	☐	NULL
isActive	VARCHAR(1)	☐	☐	☐	☐	☐	☐	☐	NULL
enuserid	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL
endatime	DATETIME	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	

Figure A.2: Vehicle allocation table

3. Vehicle booking table

The Figure A.3 shows the vehicle booking table that was design to enter all vehicle booking details.

Table Name: vehicle_booking Schema: vms

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	INT(11)	☑	☑	☐	☐	☐	☐	☑	
regNo	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL
purpose	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
jnyBegin	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
jnyDestination	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
startDate	DATE	☐	☐	☐	☐	☐	☐	☐	NULL
returnDate	DATE	☐	☐	☐	☐	☐	☐	☐	NULL
passenger	VARCHAR(75)	☐	☐	☐	☐	☐	☐	☐	NULL
timePicked	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
timeReturn	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
isActive	VARCHAR(1)	☐	☐	☐	☐	☐	☐	☐	NULL
remarks	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	NULL
enuserid	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL

Figure A.3: Vehicle booking table

4. Fuel management table

The Figure A.4 shows the vehicle fuel management table that was design to enter all fuel management details.

Table Name: fuel_register Schema: vms

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	INT(11)	☑	☑	☐	☐	☐	☐	☑	
provCode	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	NULL
regNo	VARCHAR(12)	☐	☐	☐	☐	☐	☐	☐	NULL
orderNo	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL
odometer	INT(11)	☐	☐	☐	☐	☐	☐	☐	NULL
fuelType	VARCHAR(10)	☐	☐	☐	☐	☐	☐	☐	NULL
volume	DECIMAL(5,2)	☐	☐	☐	☐	☐	☐	☐	NULL
fDate	DATE	☐	☐	☐	☐	☐	☐	☐	NULL
price	DECIMAL(10,2)	☐	☐	☐	☐	☐	☐	☐	NULL
station	VARCHAR(20)	☐	☐	☐	☐	☐	☐	☐	NULL
remarks	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
isActive	VARCHAR(1)	☐	☐	☐	☐	☐	☐	☐	NULL
enuserid	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL

Figure A.4: Fuel management table

5. Running chart management table

The Figure A.5 shows the vehicle running chart management table that was design to enter all running details.

Table Name: running_details Schema: vms

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	INT(11)	☑	☑	☐	☐	☐	☐	☑	
provCode	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	NULL
regNo	VARCHAR(12)	☐	☐	☐	☐	☐	☐	☐	NULL
rDate	DATE	☐	☐	☐	☐	☐	☐	☐	NULL
miloStart	INT(11)	☐	☐	☐	☐	☐	☐	☐	NULL
miloEnd	INT(11)	☐	☐	☐	☐	☐	☐	☐	NULL
runningKm	DECIMAL(5,2)	☐	☐	☐	☐	☐	☐	☐	NULL
kmPerLit	DECIMAL(5,2)	☐	☐	☐	☐	☐	☐	☐	NULL
inTank	DECIMAL(5,2)	☐	☐	☐	☐	☐	☐	☐	NULL
isActive	VARCHAR(1)	☐	☐	☐	☐	☐	☐	☐	NULL
remarks	VARCHAR(100)	☐	☐	☐	☐	☐	☐	☐	NULL
enuserid	VARCHAR(15)	☐	☐	☐	☐	☐	☐	☐	NULL
endatime	DATETIME	☐	☐	☐	☐	☐	☐	☐	NULL

Figure A.5: Running chart management table

6. Accident details management table

The Figure A.6 shows the vehicle accident details table that was design to enter all accident management details.

Table Name: accident_details Schema: vms

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	INT(11)	☑	☑	☐	☐	☐	☐	☑	
provCode	VARCHAR(2)	☐	☐	☐	☐	☐	☐	☐	NULL
regNo	VARCHAR(12)	☐	☐	☐	☐	☐	☐	☐	NULL
incidentPlace	VARCHAR(60)	☐	☐	☐	☐	☐	☐	☐	NULL
accDate	DATE	☐	☐	☐	☐	☐	☐	☐	NULL
officerName	VARCHAR(60)	☐	☐	☐	☐	☐	☐	☐	NULL
designation	VARCHAR(50)	☐	☐	☐	☐	☐	☐	☐	NULL
accNature	VARCHAR(60)	☐	☐	☐	☐	☐	☐	☐	NULL
damageCost	DECIMAL(10,2)	☐	☐	☐	☐	☐	☐	☐	NULL
policeRef	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	NULL
recoverFrom	VARCHAR(60)	☐	☐	☐	☐	☐	☐	☐	NULL
isActive	VARCHAR(1)	☐	☐	☐	☐	☐	☐	☐	NULL
remarks	VARCHAR(75)	☐	☐	☐	☐	☐	☐	☐	NULL

Figure A.6: Accident details management table

Appendix B - Source Codes

1. CEB vehicle registration

1.1. Record insertion

```
Try

connection.Open()

sql = "INSERT INTO cebvehicle (costCode, provCode, regNo, classId,  vClass,  tClass,
make, model, color, typeFuel, chassisNo, engineNo, cCapacity, sCapacity, tyreSize,
yearManu, usage1,dateReg, datePurchase, purchasePrice, type, remarks, isActive,
enuserid, endatime) Values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"

       Dim com As MySqlCommand = New MySqlCommand(sql, connection)
       com.Parameters.AddWithValue("costCode", costcentre)
       com.Parameters.AddWithValue("provCode", prov)
       com.Parameters.AddWithValue("regNo", regno)
       com.Parameters.AddWithValue("classId", vid)
       com.Parameters.AddWithValue("vClass", vcode)
       com.Parameters.AddWithValue("tClass", tclass)
       com.Parameters.AddWithValue("make", make)
       com.Parameters.AddWithValue("model", model)
       com.Parameters.AddWithValue("color", color1)
       com.Parameters.AddWithValue("typeFuel", fuel)
       com.Parameters.AddWithValue("chassisNo", chassis)
       com.Parameters.AddWithValue("engineNo", engineno)
       com.Parameters.AddWithValue("cCapacity", cylinder)
       com.Parameters.AddWithValue("sCapacity", seating)
       com.Parameters.AddWithValue("tyreSize", tyre_size)
       com.Parameters.AddWithValue("yearManu", manuyear)
       com.Parameters.AddWithValue("usage1", usage)
       com.Parameters.AddWithValue("dateReg", regdate1)
       com.Parameters.AddWithValue("datePurchase", purchasedate1)
       com.Parameters.AddWithValue("purchasePrice", purchaseprice)
       com.Parameters.AddWithValue("type", vtype)
       com.Parameters.AddWithValue("remarks", remarks)
       com.Parameters.AddWithValue("isActive", "Y")
       com.Parameters.AddWithValue("enuserid", UID)
       com.Parameters.AddWithValue("endatime", CurrentDate)

       com.ExecuteNonQuery()connection.Close()

Catch ex As Exception
        lblMessage.Visible = True
        lblMessage.Text = ex.Message
End Try
```

2. Vehicle transfer

```
Try
    objdb_connection.db_connection()
    trans = objdb_connection.con.BeginTransaction()

    If RadioButtonList1.SelectedValue = "A" Then
         TransType = "T"                    '----Only Vehicle Transfer

         sql = "Insert INTO trans_history(provCode, regNo, fromBranch, toBranch,
fromDate, type, remarks, enuserid, endatime) Values (?,?,?,?,?,?,?,?,?)"

    ElseIf RadioButtonList1.SelectedValue = "B" Then
         TransType = "A"                    '----Active & Transfer record

         sql = "Insert INTO trans_history(provCode, regNo, toBranch, fromDate, type,
remarks, enuserid, endatime) Values (?,?,?,?,?,?,?,?)"
    End If

    '-------Insert transation details-----------

    comm = New MySqlCommand(sql, objdb_connection.con)
    comm.Transaction = trans
    comm.Parameters.AddWithValue("provCode", provCode)
    comm.Parameters.AddWithValue("regNo", Regno)
    If RadioButtonList1.SelectedValue = "A" Then
        comm.Parameters.AddWithValue("fromBranch", OldBranch)
    End If
    comm.Parameters.AddWithValue("toBranch", NewBranch)
    comm.Parameters.AddWithValue("fromDate", transDate)
    comm.Parameters.AddWithValue("type", TransType)
    comm.Parameters.AddWithValue("remarks", remarks)
    comm.Parameters.AddWithValue("enuserid", UID)
    comm.Parameters.AddWithValue("endatime", CurrentDate)
    comm.ExecuteNonQuery()


    '------Update ceb vehicle details table----------
     Update_VehicleMasterTable(NewBranch, provCode, Regno, v_exist)

     trans.Commit()
     lblMessage.Visible = True
     lblMessage.Text = "Record Inserted......"
     ddlRegno.SelectedItem.Text = ""
     objdb_connection.con.Close()
Catch ex As Exception
     trans.Rollback()
     lblMessage.Visible = True
     lblMessage.Text = ex.Message
End Try



Sub Update_VehicleMasterTable(ByVal NewBranch As String, ByVal provCode As String,
ByVal regno As String, ByVal v_exist As String)
        sql = "Update cebvehicle set costCode=?, isActive=? where provCode=? and
regNo=?"
        comm = New MySqlCommand(sql, objdb_connection.con)
        comm.Transaction = trans
        comm.Parameters.AddWithValue("costCode", NewBranch)
```

```
        comm.Parameters.AddWithValue("isActive", v_exist)
        comm.Parameters.AddWithValue("provCode", provCode)
        comm.Parameters.AddWithValue("regNo", regno)
        comm.ExecuteNonQuery()
    End Sub
```

3. Booking cancellation

```
If Not String.IsNullOrEmpty(row.Cells(1).Text) And (row.RowState And
DataControlRowState.Edit) > 0 Then
            Dim ID As String = row.Cells(1).Text

            '----DB values----
            objdb_connection.db_connection()
            objvar.sql = "Select * from vehicle_booking where id='" & ID & "'"
            objdb_connection.openquery(objvar.sql)
            If objdb_connection.rs.Read Then
                If objdb_connection.rs.HasRows Then
                    e_flag = objdb_connection.rs("isActive")
                End If
            End If
            objdb_connection.con.Close()

            '----Datagrid Values------
            Dim flag As String = (CType((row.Cells(10).Controls(0)), TextBox)).Text

            Try
                objdb_connection.db_connection()
                trans = objdb_connection.con.BeginTransaction()

                sql = "Update vehicle_booking set isActive=? where id=?"
                Using cmd As New MySqlCommand(sql, objdb_connection.con)
                    cmd.Transaction = trans
                    cmd.Parameters.Add(New MySqlParameter("@isActive",
MySqlDbType.VarChar, 1, "isActive")).Value = flag
                    cmd.Parameters.Add(New MySqlParameter("@id", MySqlDbType.VarChar,
5, "id")).Value = ID
                    cmd.ExecuteNonQuery()
                End Using

                trans.Commit()
                objdb_connection.con.Close()

                lblMessage.Text = "Record Updated......"
                GridView1.EditIndex = -1
            Catch ex As Exception
                lblMessage.Text = ex.Message
            End Try
            lblMessage.Visible = True
        End If
```

4. Report of Expenses of a vehicle

```
dt = New VehicleExpenditure.DataTable4DataTable

        dt = New DataTable
        dt.Columns.Add(New DataColumn("Date"))
        dt.Columns.Add(New DataColumn("Fuel"))
        dt.Columns.Add(New DataColumn("Services"))
        dt.Columns.Add(New DataColumn("SpareParts"))
        dt.Columns.Add(New DataColumn("Insurance"))

        i = 0
        yearMonth(0) = ""

        Try
            objdb_connection.db_connection()
            sql = "Select * from fuel_register where regNo='" & regNo & "' and
(fDate>='" & fromDate1 & "' and fDate<='" & toDate1 & "') order by fDate, odoMeter"
            objdb_connection.openquery(sql)

            While objdb_connection.rs.Read
                If objdb_connection.rs.HasRows Then
                    dtrow = dt.NewRow()

                    Dim fDate As Date = objdb_connection.rs("fDate")
                    '---for graph----
                    Dim year As String = Mid(fDate.ToString("dd/MM/yyyy"), 4, 7)
                    yearMonth(i) = year
                    If (i = 0) Then
                        totPrice(i) = objdb_connection.rs("price")
                    ElseIf (yearMonth(i) <> yearMonth(i - 1)) Then

                        '-----Expenses for Services--------
                        k = 0
                        objdb_connection.db_connect()
                        sql = "Select * from service where regNo='" & regNo & "' and
(serviceDate>= '" & fromDate1 & "' and serviceDate <='" & toDate1 & "') order by
serviceDate,regNo"
                        objdb_connection.openquery1(sql)
                        While objdb_connection.rs1.Read
                            If objdb_connection.rs1.HasRows Then
                                Dim serviceDate As Date =
objdb_connection.rs1("serviceDate")
                                Dim year1 As String =
Mid(serviceDate.ToString("dd/MM/yyyy"), 4, 7)

                                If yearMonth(i - 1) = year1 Then
                                    If k = 0 Then
                                        servicePrice(k) =
objdb_connection.rs1("payAmount")
                                    Else
                                        servicePrice(k) = servicePrice(k - 1) +
objdb_connection.rs1("payAmount")
                                    End If
                                End If
                            End If
                            k = k + 1
                        End While
                        objdb_connection.con1.Close()
```

```
                        '-----Expenses for spare parts--------
                        j = 0
                        objdb_connection.db_connect()
                        sql = "Select * from spare_parts where regNo='" & regNo & "'
and (purDate>= '" & fromDate1 & "' and purDate <='" & toDate1 & "') order by
purdate,regNo"
                        objdb_connection.openquery1(sql)
                        While objdb_connection.rs1.Read
                            If objdb_connection.rs1.HasRows Then
                                Dim purDate As Date = objdb_connection.rs1("purDate")
                                Dim year1 As String =
Mid(purDate.ToString("dd/MM/yyyy"), 4, 7)

                                If yearMonth(i - 1) = year1 Then
                                    If j = 0 Then
                                        partsPrice(j) =
objdb_connection.rs1("totPrice")
                                    Else
                                        partsPrice(j) = partsPrice(j - 1) +
objdb_connection.rs1("totPrice")
                                    End If
                                End If
                            End If
                            j = j + 1
                        End While
                        objdb_connection.con1.Close()
                        '----------------------------------------

                        dtrow = dt.NewRow()
                        dtrow(0) = yearMonth(i - 1)
                        dtrow(1) = totPrice(i - 1).ToString("###,###.#0")
                        dtrow(2) = servicePrice(k - 1).ToString("###,###.#0")
                        dtrow(3) = partsPrice(j - 1).ToString("###,###.#0")
                        dtrow(4) = ""
                        dt.Rows.Add(dtrow)
                        servicePrice(k - 1) = 0
                        partsPrice(j - 1) = 0
                        Dim totalPrice As Decimal = objdb_connection.rs("price")
                        totPrice(i) = totalPrice.ToString("###,###.#0")
                    Else
                        totPrice(i) = totPrice(i - 1) + objdb_connection.rs("price")
                    End If
                End If
                i = i + 1
            End While
            objdb_connection.con.Close()
        Catch ex As Exception
            Message_label(ex.Message)
        End Try
```

Appendix C – Test Cases

Test case for Vehicle allocation

The test cases for vehicle allocation are displayed below Table C.1.

Test	Test Case	Expected result	Actual result
Add vehicle allocation details	Press "Submit" button without entering officer name	Message should display "Cannot be blank officer name..."	Message displayed "Cannot be blank officer name..."
	Press "Submit" button without entering "From date"	Message should display "Invalid date..."	Message displayed "Invalid date..."
	"From date" is greater than current date	Message should display "Invalid date..."	Message displayed "Invalid date..."

Table C.1: Test case for vehicle allocation

Test case for Vehicle transfer

The test cases for CEB vehicle transfer are displayed below Table C.2.

Test	Test Case	Expected result	Actual result
Add vehicle transfer details	Press "Submit" button without entering "Transfer date"	Message should display "Invalid date..."	Message displayed "Invalid date..."
	"Transfer date" is greater than current date	Message should display "Invalid date..."	Message displayed "Invalid date..."

Table C.2: Test case for vehicle transfer

Test case for Vehicle parking

The test cases for CEB vehicle parking are displayed below Table C.3.

Test	Test Case	Expected result	Actual result
Add parking details	Press "Submit" button without entering "Transfer date"	Message should display "Invalid date..."	Message displayed "Invalid date..."
	"Transfer date" is greater than current date	Message should display "Invalid date..."	Message displayed "Invalid date..."

Table C.3: Test case for vehicle parking

Test case for Fuel order

The test cases for fuel order are displayed below Table C.4.

Test	Test Case	Expected result	Actual result
Add fuel order details	Click "Submit" button without entering fuel order number	Message should be display "Fuel order number cannot be blank…"	Message should be display "Fuel order number cannot be blank…"
	Click "Submit" button without entering fuel order date	Message should be display "Fuel order date is invalid…"	Message should be display "Fuel order date is invalid…"
	Click "Submit" button without entering fuel volume	Message should be display "Cannot be '0' fuel volume…"	Message should be display "Cannot be '0' fuel volume…"
	Click "Submit" button without entering fuel price	Message should be display "Cannot be '0.00' fuel price…"	Message should be display "Cannot be '0.00' fuel price…"
Cancel vehicle details		Reset all textboxes and dropdowns except cost center and vehicle registration	Reset all textboxes and dropdowns except cost center and vehicle registration

Table C.4: Test case for fuel order

Test case for Running chart management

The test cases for vehicle running chart management are displayed below Table C.5.

Test	Test Case	Expected result	Actual result
Add running chart details	Click "Submit" button without entering milo meter reading (End)	Message should be display "Milo meter (End) cannot be '0' meter reading…"	Message should be display "Milo meter (End) cannot be '0' meter reading…"
	Click "Submit" button without entering running Km	Message should be display "Running Km cannot be '0' …"	Message should be display "Running Km cannot be '0' …"
	Click "Submit" button without entering Km per liters	Message should be display "Km per liters cannot be '0' …"	Message should be display "Km per liters cannot be '0' …"
	Click "Submit" button without entering running date	Message should be display "Invalid date …"	Message should be display "Invalid date …"

Table C.5: Test case for running chart management

Appendix D – User Documentation

The purpose of this document is to provide all essential information for a user to assist and make comprehensive use of administration application. This document includes most important description of interfaces and its functionality.

1. Login Page

Login page is the first user interface to log in to the system. By typing the valid user name and password user can log in to the system. The following Figure D.1 shows the login page.

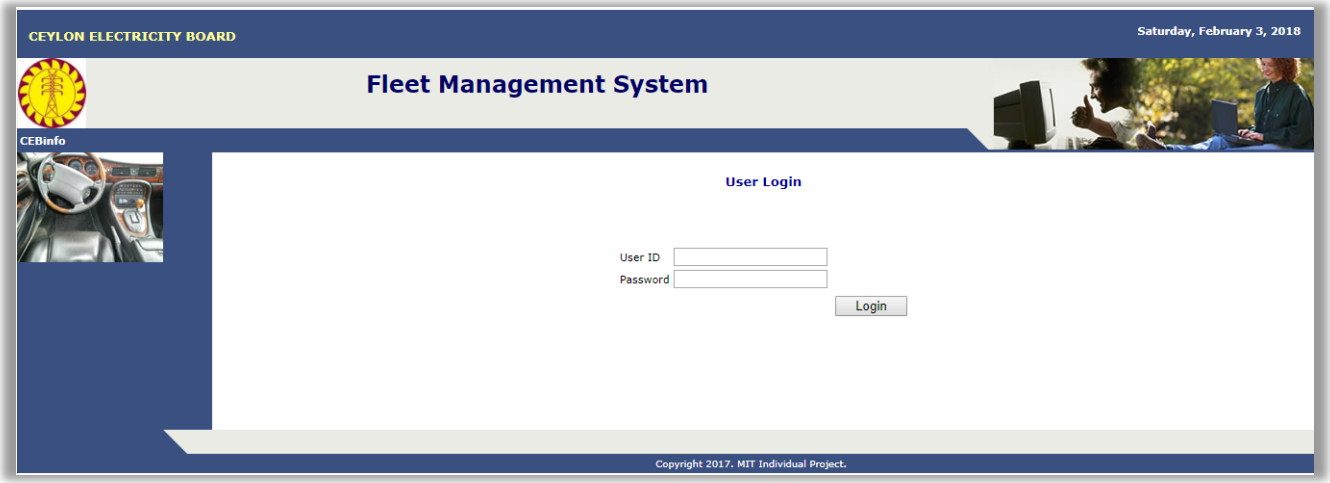


Figure D.1: Login Page

After Login to the system, Home page will be displayed. According to the user login users are categorized in to different user levels.

User Levels

- Administrator – person who administrates the system
- Subject Clerk/Data Entry Operator (DEO) – person who handles the vehicle details in the system
- GM/AGMs/DGMs – person who views management reports
- Driver

Username: enter your EPF Number with leading zero (6 number text)

The following Figure D.2, figure D.3 and figure D.4 show the dynamic menu according to the user profile.

❖ Admin Home Page

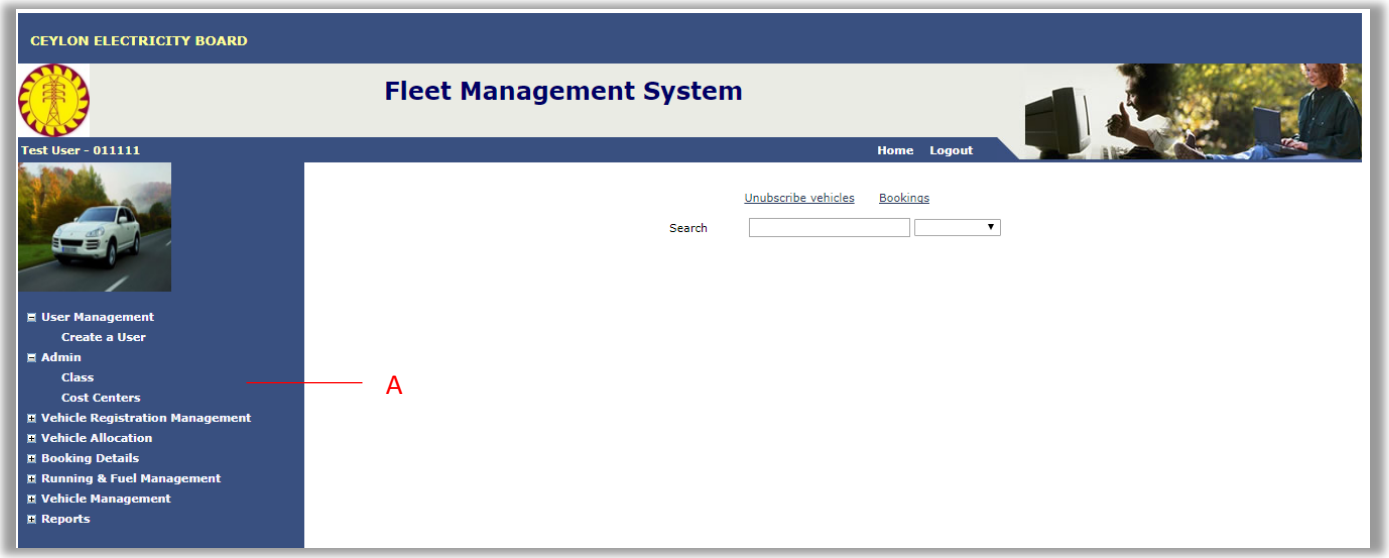


Figure D.2: Admin home page

A- Dynamic menu according to the user profile.

❖ Subject Clerk/Data Entry Operator

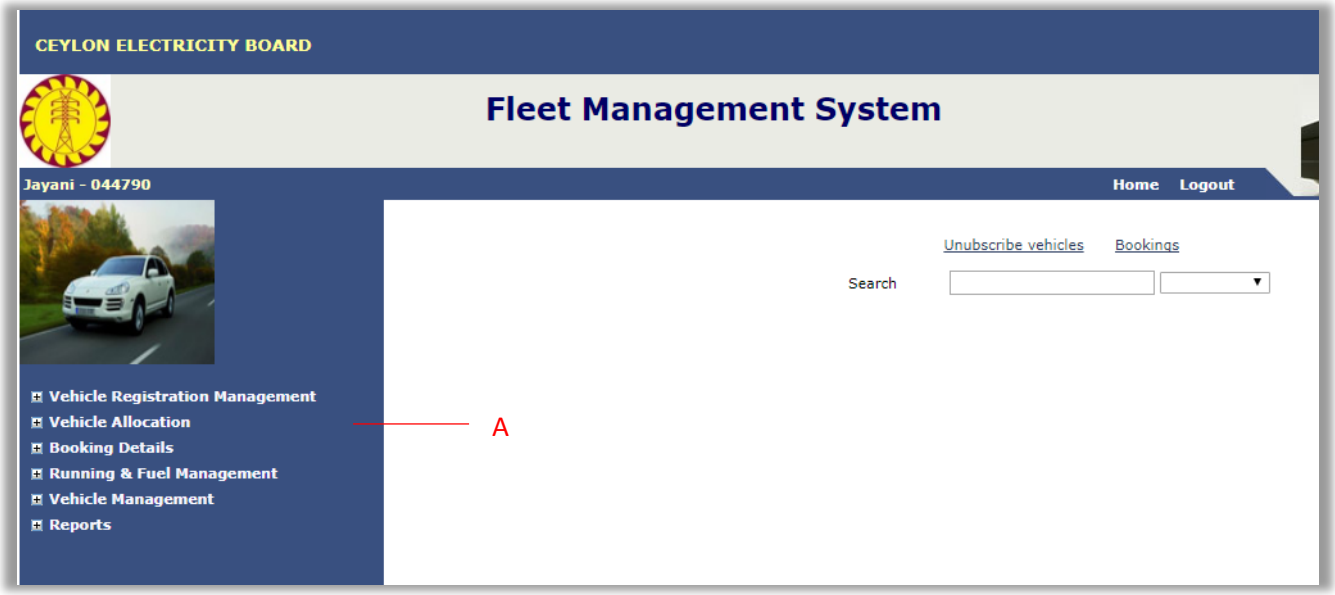


Figure D.3: Subject Clerk/Data Entry Operator's home page

❖ Driver

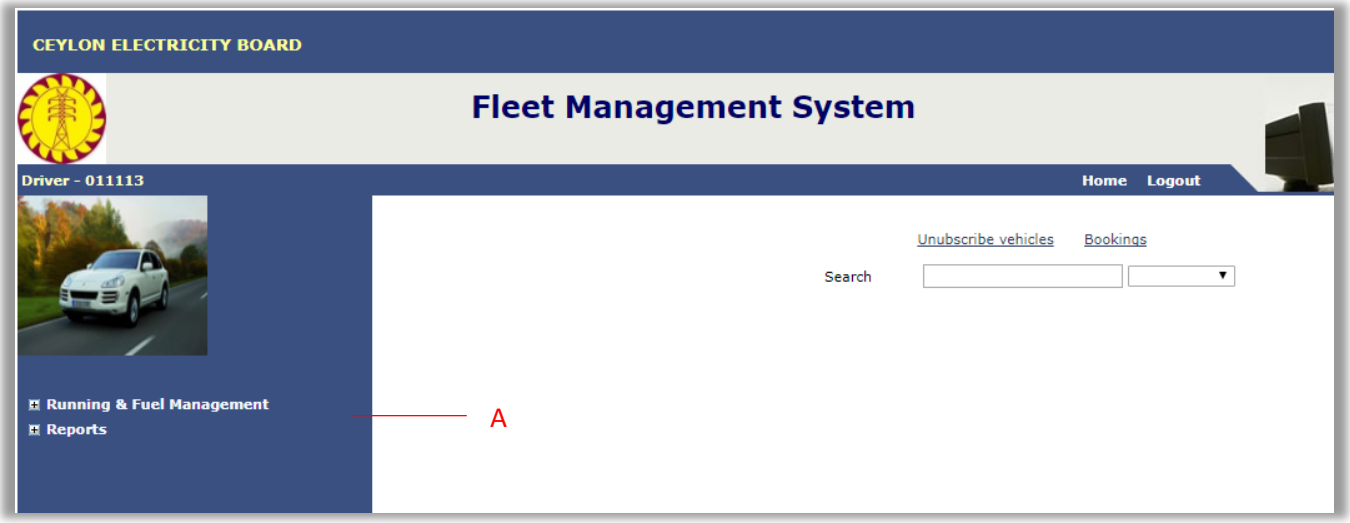


Figure D.4: Driver's home page

2. CEB vehicle details

2.1 Add details

This is a subject clerk or data entry function which is used to add vehicle details to the system.

Steps to add vehicle details

1- Select hierarchy from the dropdown.
2- Select related cost center from the dropdown.
3- Select vehicle registered province from the dropdown.
4- Type registration number.

Vehicle number should be according to the following format.

- XX9999 (ex: NB5698)
- XXX9999 (ex: CAT5698)
- 99-9999 (ex: 53-5698)
- 999-9999 (ex: 253-5698)
- 99SRI9999 (ex: 56SRI5698)

5- Select vehicle class and sub class from the dropdown.
6- Type chassis number.
7- Select vehicle make from the dropdown.
8- Type taxation class
9- Select color from the dropdown.
10- Type cylinder capacity

11- Type tyre size

12- Type model

13- Type engine number

14- Type seating capacity (should be only integer values)

15- Select type of fuel from the dropdown

16- Type vehicle manufacture year (should be 4 digit integer value)

17- Select date of purchase from the calendar icon.

18- Select usage of vehicle from the dropdown.

19- Select date of registration from the calendar icon.

20- Select type from the dropdown

21- Type any comments if have

22- Select "Submit" button to add record to the database.

23- A successful message will display if data is correctly entered otherwise an error will be displayed.

The following Figure D.5 and Figure D.6 shows the CEB vehicle registration form.

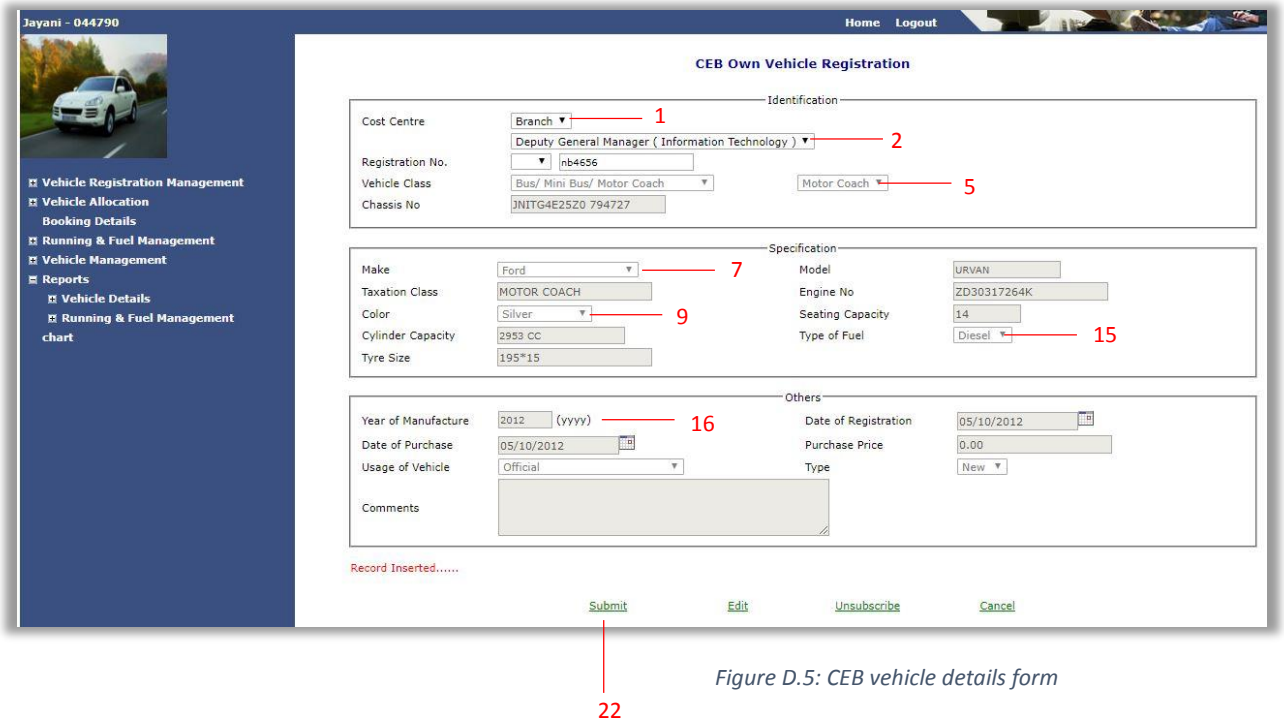


Figure D.5: CEB vehicle details form

2.2 Edit CEB vehicle details

Using this function, available record details can be altered.

Steps

1. Select hierarchy from the dropdown.

2. Select related cost center from the dropdown.
3. Select vehicle registered province from the dropdown.
4. Type registration number (if record is available details is displayed on the screen)
5. Click on “Edit” button (text boxes and dropdowns are enabled).
6. Changed details if you want.
7. Click on “Submit” button

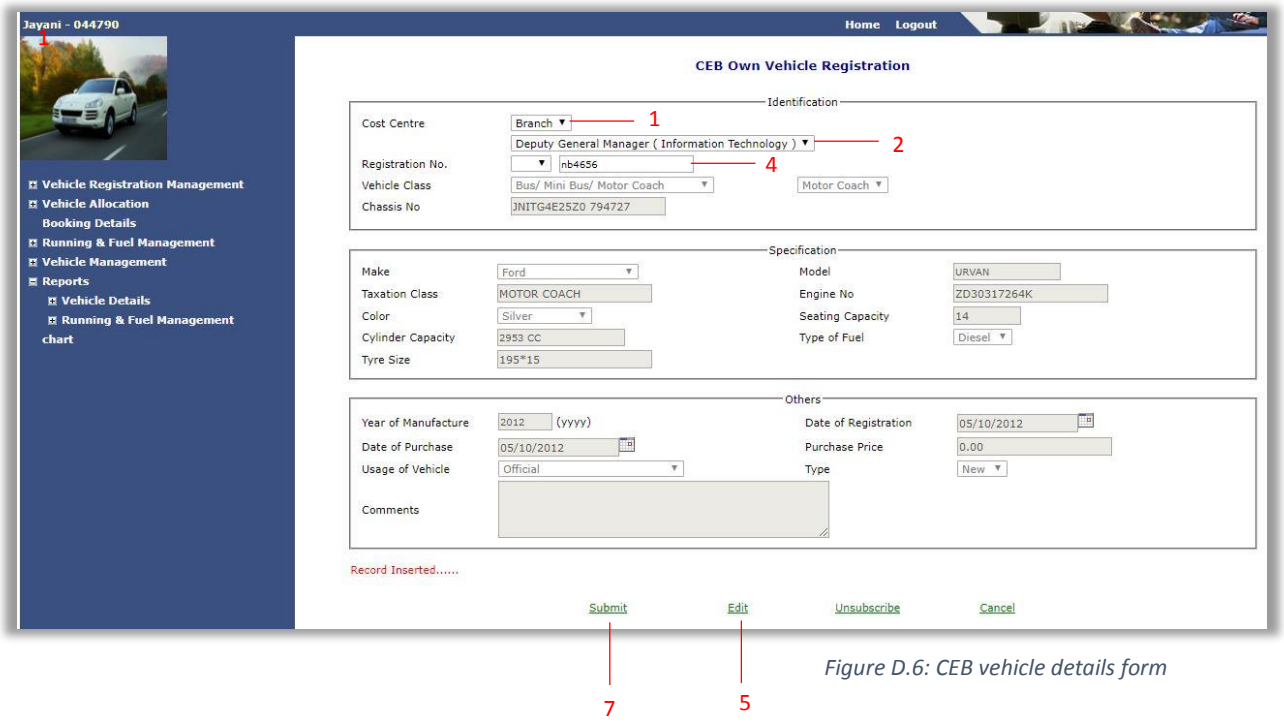


Figure D.6: CEB vehicle details form

3. Vehicle Allocation

This is a subject clerk or data entry function which is used to add vehicle allocation details to the system.

Steps to add vehicle allocation details

1- Select cost center related to CEB hierarchy from the dropdown.
2- Select vehicle registration number from the dropdown.
3- Type name of the officer.
4- Select designation of the officer from the dropdown.
5- Select vehicle allocation type to the officer from the dropdown.
6- Select whether vehicle type is pool or not.
7- Select or type parking location.
8- Select vehicle allocation date to the officer from the calendar icon.

9- Type remarks if available

10- Click on "Submit button.

Vehicle allocation form (Figure D.7) shows as follows.

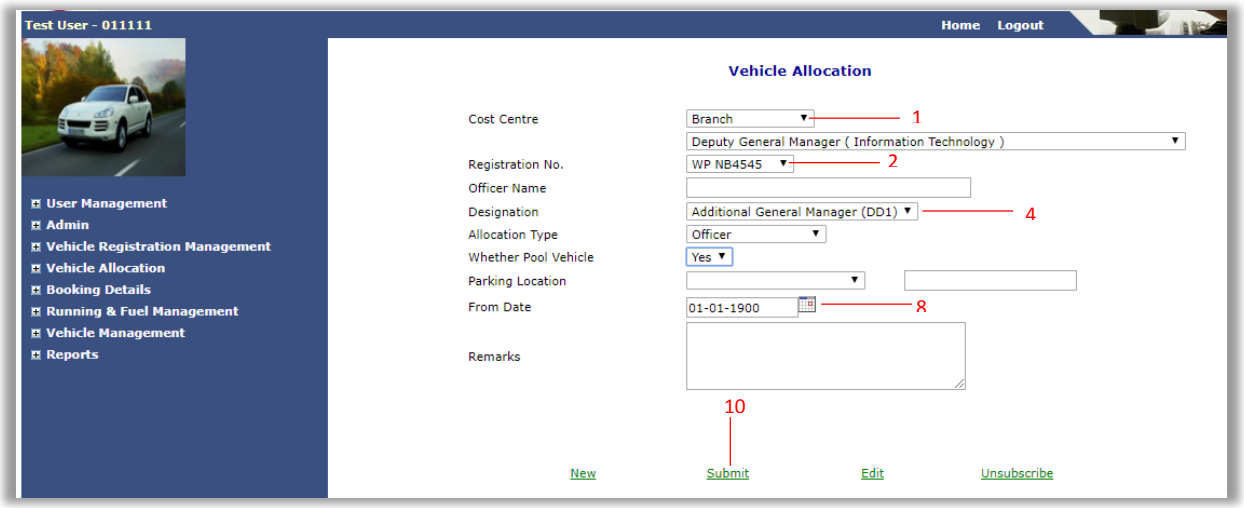


Figure D.7: Vehicle allocation form

4. Reports

4.1 CEB vehicle details

1- Select type from the dropdown (All – all details of given authority level, Only branch – view vehicle details of selected or authority level cost center, province- view only province level vehicle details)

2- Select cost center or province according to given type in above.

3- Click on view button.

CEB vehicle detail report (Figure D.8) shows as follows.

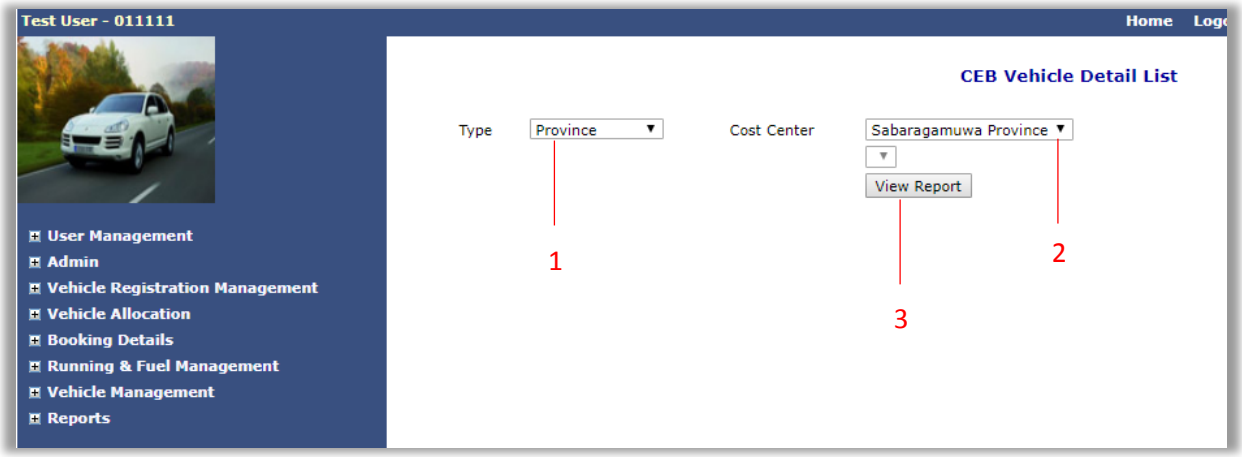


Figure D.8: CEB vehicle details report form

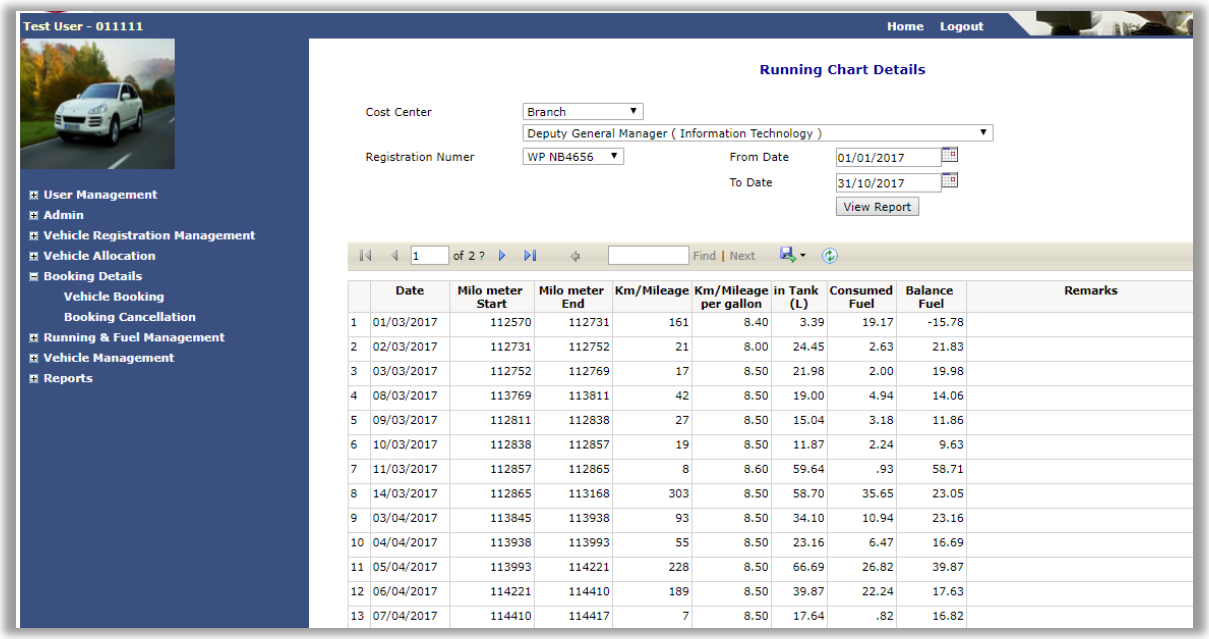
5. Logout

In the fleet management system, on the top of the home page there will be a logout button with the text of "Logout" (Figure D.9). By clicking logout button user can logout from the system.

Figure D.9: Logout button

Appendix E – Results

The following Figure E.1 and Figure E.2 shows in detailed view and graphical representation of running chart in the given period.



	Date	Milo meter Start	Milo meter End	Km/Mileage	Km/Mileage per gallon	in Tank (L)	Consumed Fuel	Balance Fuel	Remarks
1	01/03/2017	112570	112731	161	8.40	3.39	19.17	-15.78	
2	02/03/2017	112731	112752	21	8.00	24.45	2.63	21.83	
3	03/03/2017	112752	112769	17	8.50	21.98	2.00	19.98	
4	08/03/2017	113769	113811	42	8.50	19.00	4.94	14.06	
5	09/03/2017	112811	112838	27	8.50	15.04	3.18	11.86	
6	10/03/2017	112838	112857	19	8.50	11.87	2.24	9.63	
7	11/03/2017	112857	112865	8	8.60	59.64	.93	58.71	
8	14/03/2017	112865	113168	303	8.50	58.70	35.65	23.05	
9	03/04/2017	113845	113938	93	8.50	34.10	10.94	23.16	
10	04/04/2017	113938	113993	55	8.50	23.16	6.47	16.69	
11	05/04/2017	113993	114221	228	8.50	66.69	26.82	39.87	
12	06/04/2017	114221	114410	189	8.50	39.87	22.24	17.63	
13	07/04/2017	114410	114417	7	8.50	17.64	.82	16.82	

Figure E.1: Running chart detail report

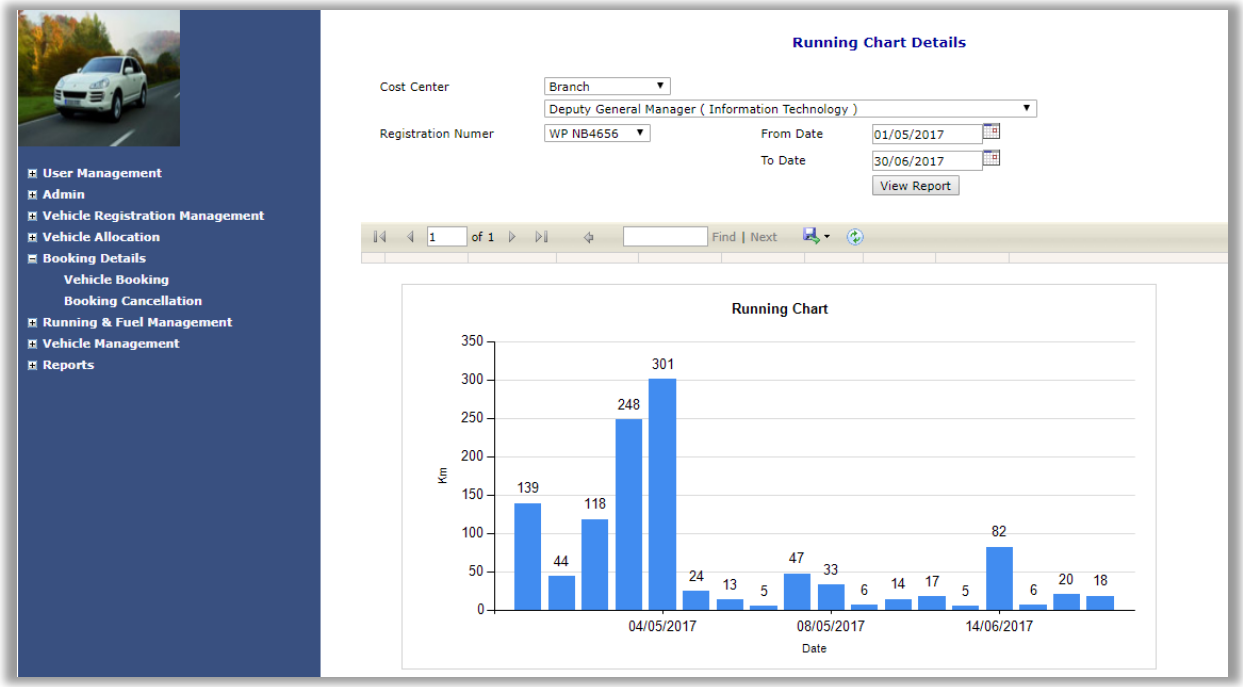


Figure E.2: Running chart detail in graphical representation

The following Figure E.3 shows the total fuel expenditure of the cost center and Figure E.4 shows the fuel expenditure of a vehicle in a given period.

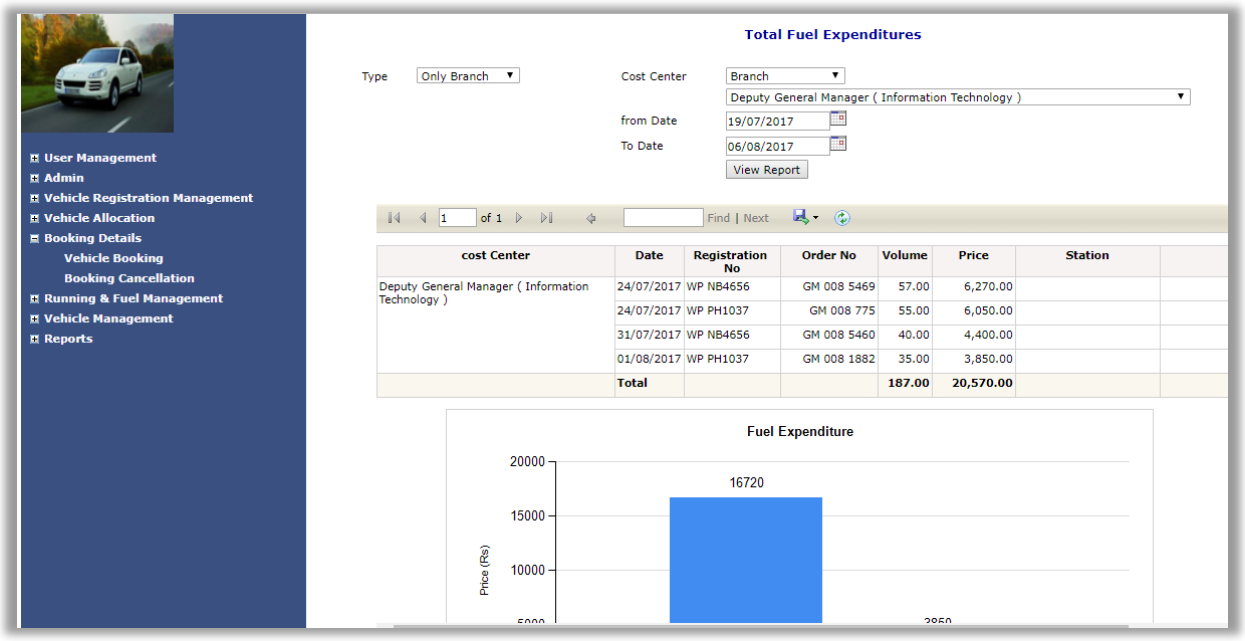


Figure E.3: Total fuel expenditure report

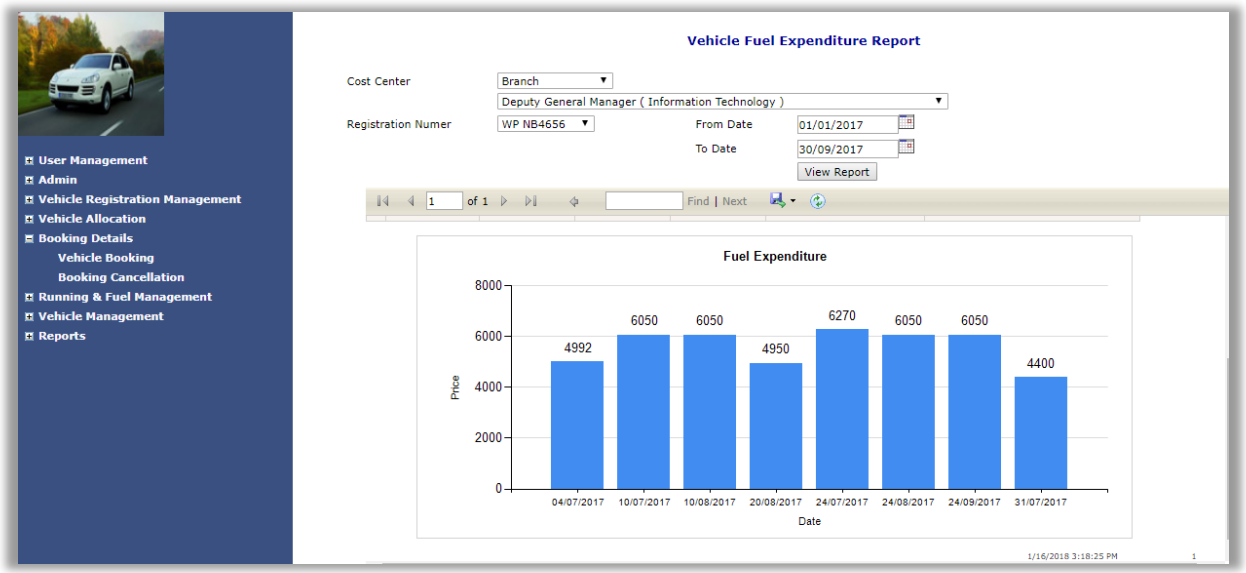
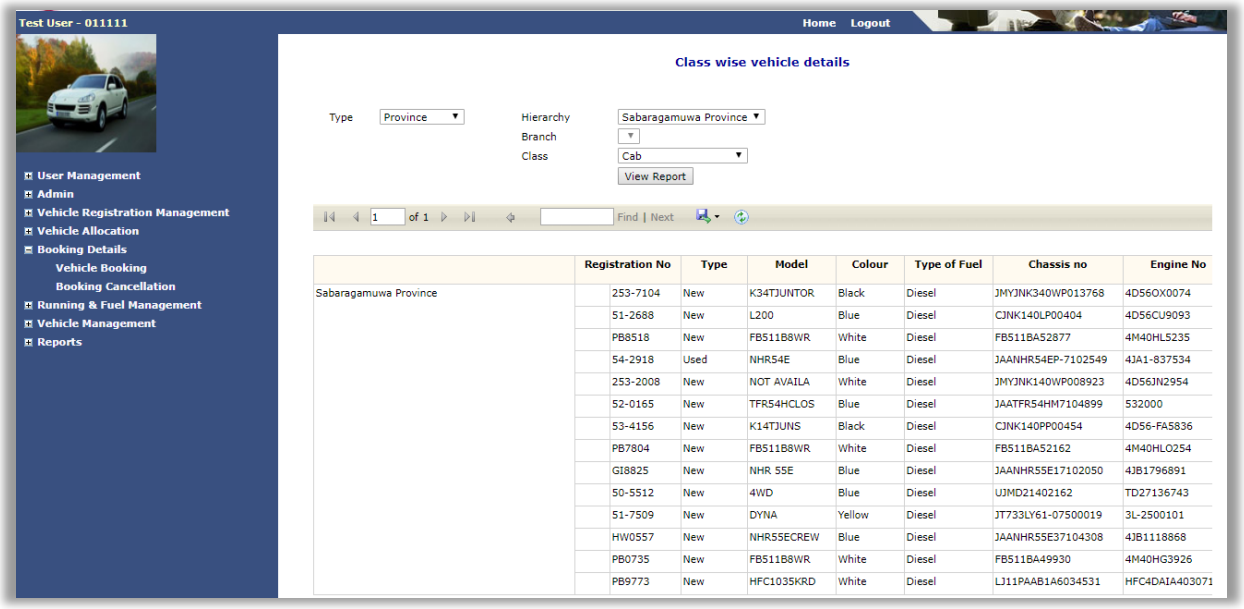


Figure E.4: Graphical representation of fuel expenditure

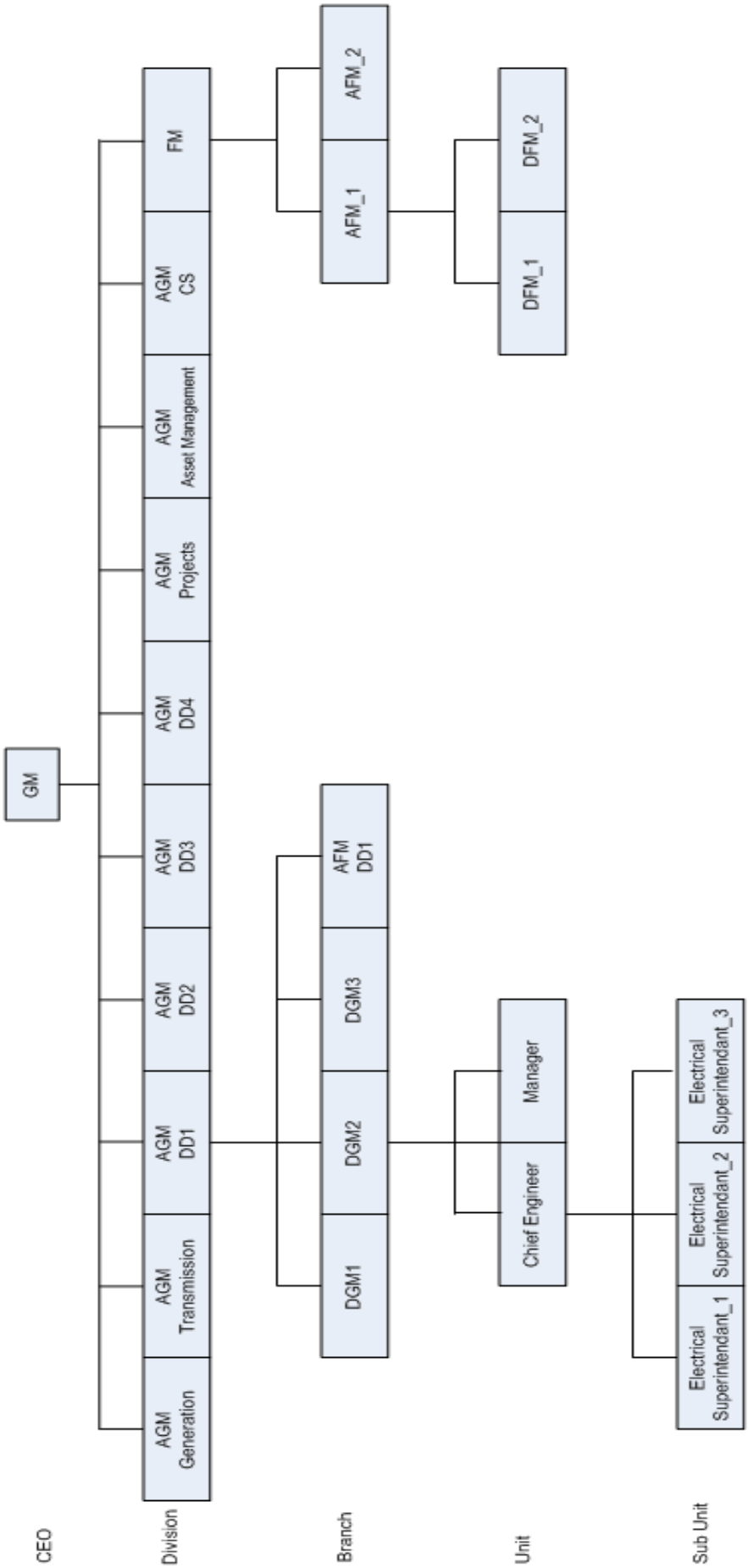
The following Figure E.5 shows the class wise vehicle details report.



	Registration No	Type	Model	Colour	Type of Fuel	Chassis no	Engine No
Sabaragamuwa Province	253-7104	New	K34TJUNTOR	Black	Diesel	JMYJNK340WP013768	4D56OX0074
	51-2688	New	L200	Blue	Diesel	CJNK140LP00404	4D56CU9093
	PB8518	New	FB511B8WR	White	Diesel	FB511BA52877	4M40HL5235
	54-2918	Used	NHR54E	Blue	Diesel	JAANHR54EP-7102549	4JA1-837534
	253-2008	New	NOT AVAILA	White	Diesel	JMYJNK140WP008923	4D56JN2954
	52-0165	New	TFR54HCLOS	Blue	Diesel	JAATFR54HM7104899	532000
	53-4156	New	K14TJUNS	Black	Diesel	CJNK140PP00454	4D56-FA5836
	PB7804	New	FB511B8WR	White	Diesel	FB511BA52162	4M40HLO254
	GI8825	New	NHR 55E	Blue	Diesel	JAANHR55E17102050	4JB1796891
	50-5512	New	4WD	Blue	Diesel	UJMD21402162	TD27136743
	51-7509	New	DYNA	Yellow	Diesel	JT733LY61-07500019	3L-2500101
	HW0557	New	NHR55ECREW	Blue	Diesel	JAANHR55E37104308	4JB1118868
	PB0735	New	FB511B8WR	White	Diesel	FB511BA49930	4M40HG3926
	PB9773	New	HFC1035KRD	White	Diesel	LJ11PAAB1A6034531	HFC4DAIA403071

Figure E.5: Class wise vehicle detail report

Annexure 1


CEO
Division
Branch
Unit
Sub Unit
GM
AGM Generation
AGM Transmission
AGM DD1
AGM DD2
AGM DD3
AGM DD4
AGM Projects
AGM Asset Management
AGM CS
FM
AFM_1
AFM_2
DFM_1
DFM_2
DGM1
DGM2
DGM3
AFM DD1
Chief Engineer
Manager
Electrical Superintendant_1
Electrical Superintendant_2
Electrical Superintendant_3