

Last Mile Delivery Management System



By
Syed Ahtsham Ul Hassan

Supervised by
Dr. Khalid Saleem

Department of Computer Science,
Quaid-I-Azam University, Islamabad, Pakistan.
Session (2018-2022)

ACKNOWLEDGEMENT

Firstly, I would like to express my gratefulness to Allah Almighty, Whose blessings I have seen during my journey at Quaid-i-Azam University in the form of achievements, success and learning. In addition, I am obligated to several people for their sustenance and generous contribution. I especially express my sincere gratitude to my advisor Dr. Khalid Saleem for guiding me throughout the process. I could not have imagined having a better advisor and mentor for my project. I am also thankful to my family and friends for their continuous support during this journey.

ABSTRACT

Last Mile Delivery Management System (LMDMS) is a web application that will help the courier managers to manage the vehicles and deliveries in a simpler and easier way. The database will keep track of the consignments (yet to be delivered, delivered and returned), drivers, vehicles, fuel allocations to the drivers for the vehicles, and vehicle assignments to the drivers etc.

LMDMS will help the supervisors to automatically generate the delivery sheets from consignments for specific areas. This automatic process will be according to the consignments' weights and volumes. At first consignments will be separated on the basis of their area zones (postal codes/zip code). Secondly, Consignments with less than or equal to 2kg weight will be separated from consignments with greater than 2kg weight and are added to the delivery sheet to be assigned to a bike and a rider. And >2kg will be added to the delivery sheets for the vehicles. And this will be done while considering the vehicle total weight and volume carrying capacity. This process will be near optimal in assigning the vehicle a delivery sheet so that the vehicle would carry the maximum number of deliveries per trip. Moreover, the fuel is allocated to the drivers while considering the area they have to go for the deliveries. This area will come from the delivery sheet that has just been created. For the areas which requires extra fuel, such areas will get extra fuel automatically. The manager can edit the areas which requires extra fuel from his dashboard. LMDMS uses MySQL.

Table of Contents

Table of Figures -----	j
Table of Tables -----	iii
1. Software Project Management Plan-----	1
1.1 Introduction-----	1
1.1.1 Project Overview-----	1
1.1.2 Project Deliverables-----	2
1.2 Project Organization-----	2
1.2.1 Software Process Model-----	2
1.2.2 Roles and Responsibilities -----	3
1.2.3 Tools and Techniques-----	4
1.3 Project Management Plan-----	5
1.3.1 Tasks -----	5
1.3.2 Description -----	5
1.3.2.1 Requirement and Analysis Phase -----	5
1.3.2.2 Design Phase -----	5
2. Software Requirements Specification-----	7
2.1 Introduction-----	7
2.1.1 Purpose -----	7
2.1.2 Mission Statement -----	7
2.1.3 Definitions, Acronyms and Abbreviations -----	9
2.1.4 Overview -----	9
2.2 Overall Description-----	10
2.2.1 Product Perspective -----	10
2.2.1.1 System Interfaces -----	10
2.2.1.2 User Interfaces -----	10
2.2.1.3 Hardware Interfaces -----	10
2.2.1.4 Software Interfaces -----	10
2.2.1.5 Communication Interfaces -----	10
2.2.2 Product Functions -----	10
2.2.3 User Characteristics -----	11
2.2.4 Constraints -----	11
2.2.5 Assumptions and Dependencies -----	11
2.3 Specific Requirements -----	11
2.3.1 External Interfaces -----	11
2.3.3 Performance Requirements -----	11
2.3.4 Software Quality Attributes -----	12
2.3.4.1 Reliability -----	12
2.3.4.2 Availability -----	12
2.3.4.3 Security -----	12
2.3.4.4 Portability -----	12

2.3.5 Database Requirements-----	13
2.3.5.1 Conceptual Data Model-----	14
2.3.5.2 Logical Data Model -----	13
2.4 Functional Requirements-----	16
2.4.1 Use Cases List -----	16
2.4.2 Use Case Diagram-----	17
2.4.3 Use Case Descriptions -----	18
2.4.3.1 UC-1 Login -----	18
2.4.3.2 UC-2 Add Staff-----	19
2.4.3.3 UC-3 View Profile-----	20
2.4.3.4 UC-4 Edit Staff's profile-----	21
2.4.3.5 UC-5 Delete Staff's profile-----	22
2.4.3.6 UC-6 View Fuel information -----	23
2.4.3.7 UC-7 Request Fuel-----	24
2.4.3.8 UC-8 View Fuel Consumption-----	26
2.4.3.9 UC-9 Add a Vehicle -----	27
2.4.3.10 UC-10 Edit Vehicle Information-----	28
2.4.3.11 UC-11 Delete Vehicle-----	29
2.4.3.12 UC-12 View Vehicle-----	30
2.4.3.13 UC-13 Assign Vehicle -----	31
2.4.3.14 UC-14 Edit Vehicle Assignment -----	32
2.4.3.15 UC-15 View Vehicle Assignment -----	34
2.4.3.16 UC-16 Un-Assign Vehicle -----	34
2.4.3.17 UC-17 View Fuel Request History-----	36
2.4.3.18 UC-18 View Consignments-----	36
2.4.3.19 UC-19 Generate Delivery Sheet-----	37
2.4.3.20 UC-20 Edit Delivery Sheet-----	39
2.4.3.21 UC-21 Delete Delivery Sheet-----	40
2.4.3.22 UC-22 View Delivery Sheet-----	41
2.4.3.23 UC-23 Check out Delivery Sheet-----	42
2.4.3.24 UC-24 View Areas-----	43
2.4.3.25 UC-25 Edit Areas-----	44
2.4.3.26 UC-26 Logout-----	45
2.5 System Sequence Diagrams (SSDs)-----	46
2.5.1 Login SSD -----	46
2.5.2 Edit Staff Profile SSD-----	46
2.5.3 Edit Vehicle Assignment SSD -----	47
2.5.4 Edit Area SSD-----	48
2.5.4 Edit Delivery Sheet SSD -----	49
2.6 Domain Model-----	50
2.7 Data Flow Diagrams-----	50
2.7.1 Delivery Sheet Generation Process DFD Level 0 -----	50
2.7.1a. Delivery Sheet Generation Process DFD Exploded -----	50
2.7.2 Fuel Allocation Process DFD Level 0 -----	50

2.7.2a. Fuel Allocation Process DFD Exploded-----	50
2.7.3 Delivery Sheet Generation for Bike and Vehicle DFD Level 0 -----	50
2.7.3a. Delivery Sheet Generation for Bike and Vehicle DFD Exploded -----	50
2.7.4. Vehicle Assignment Process DFD Level 0-----	50
2.7.4a. Vehicle Assignment Process DFD Exploded-----	50
3. Software Design Description -----	61
 3.1 Introduction-----	61
 3.1.1 Purpose-----	61
 3.1.2 Design Overview-----	61
 3.2 Software Architecture Design -----	61
 3.2.1 Chosen System Architecture-----	61
 3.2.2 Discussion of Alternative Designs-----	63
 3.2.3 System Interface Description-----	63
 3.3 User Interface Design -----	63
 3.3.1 Description of the User Interface-----	63
 3.3.2. Screen Images-----	63
 3.3.2.1 Log In Screen-----	64
 3.3.2.2 Home Screen-----	64
 3.3.2.3 View Vehicles Screen-----	65
 3.3.2.4. Vehicle Assignments Screen -----	65
 3.3.2.5 Delivery Sheets List Screen-----	66
 3.3.2.6 View Delivery Sheet Screen-----	66
 3.3.2.7 Request Fuel Screen-----	67
 3.3.2.8 Add Staff Member Screen-----	67
 3.3.3 Sequence Diagrams-----	68
 3.3.3a Objects and Action -----	68
 3.3.3.1 Login Sequence Diagram -----	68
 3.3.3.2 Generate Delivery Sheet Sequence Diagram-----	69
 3.3.3.3 Request Fuel Sequence Diagram -----	69
 3.3.3.4 Assign Vehicle Sequence Diagram-----	69
 3.4 Class Diagram-----	71
	71
4. Software Test Documentation -----	94
 4.1 Introduction-----	94
 4.1.1 System Overview -----	94
 4.1.2 Test Approach-----	94
 4.2 Test Plan -----	94
 4.2.1 Features to be tested-----	94
 4.2.2 Features not to be tested-----	94
 4.2.3 Testing tools and environment-----	94
 4.3 Test Cases-----	95

4.3.1 TC-1: Login -----	95
4.3.2 TC-2: Add new Staff Profile -----	95
4.3.3 TC-3: Request Fuel -----	96
4.3.4 TC-4: Add Consignment to Delivery Sheet -----	97
4.3.5 TC-5: Generate Delivery Sheet -----	97
4.3.6 TC-6: Add Vehicle -----	98
References-----	99
Appendix -----	100

Table of Figures

Figure 1. Software Process Model (Incremental Model) -----	3
Figure 2. Gantt Chart -----	6
Figure 3. LMDMS Scope Diagram -----	7
Figure 4. Relational Database Model -----	13
Figure 5. Conceptual Data Model -----	14
Figure 6. Logical Data Model -----	15
Figure 7. Use Case Diagram -----	17
Figure 8. Login SSD -----	46
Figure 9. Edit Staff Profile SSD -----	46
Figure 10. Edit Vehicle Assignment SSD -----	47
Figure 11. Edit Area SSD -----	48
Figure 12. Edit Delivery Sheet SSD -----	49
Figure 13. Domain Model -----	50
Figure 14. Delivery Sheet Generation Process DFD – Level 0 -----	51
Figure 15. Delivery Sheet Generation Process DFD – Exploded -----	52
Figure 16. Fuel Allocation Process DFD Level 0 -----	53
Figure 17. Fuel Allocation Process DFD - Expanded -----	54
Figure 18. Delivery Sheet Generation for Bike and Vehicle – DFD Level 0 -----	55
Figure 19. Delivery Sheet Generation for Bike and Vehicle – DFD Exploded -----	56
Figure 20. Vehicle Assignment Process – DFD Level 0 -----	57
Figure 21. Vehicle Assignment Process – DFD Exploded -----	58
Figure 22. LMDMS Architecture Diagram -----	62
Figure 23. Log In screen -----	64
Figure 24. Home screen -----	64
Figure 25. View Vehicles Screen -----	65
Figure 26. Vehicle Assignments Screen -----	65

Figure 27. Delivery Sheets List Screen -----	66
Figure 28. View Delivery Sheet Screen -----	66
Figure 29. Request Fuel Screen -----	67
Figure 30. Add new Staff Screen -----	67
Figure 31. Login Sequence Diagram -----	68
Figure 32. Generate Delivery Sheet Diagram -----	69
Figure 33. Request Fuel SD -----	69
Figure 34. Assign Vehicle -----	70
Figure 35. Class Diagram -----	71

Table of Tables

Table 1. Tools and Techniques -----	4
Table 2. Definitions, Acronyms and Abbreviations -----	9
Table 3. UC-1 Login -----	18
Table 4. UC-2 Add staff member -----	19
Table 5. UC-3 View Profile -----	20
Table 6. UC-4 Edit staff's profile -----	21
Table 7. UC-5 Delete Staff Profile -----	22
Table 8. UC-6 View fuel information -----	23
Table 9. UC-7 Request Fuel -----	24
Table 10. UC-8 View Fuel Consumption -----	26
Table 11. UC-9 Add a vehicle -----	27
Table 12. UC-10 Edit Vehicle Information -----	28
Table 13. UC-11 Delete Vehicle -----	29
Table 14. UC-12 View Vehicle -----	30
Table 15. UC-13 Assign Vehicle -----	31
Table 16. UC-14 Edit Vehicle Assignment -----	32
Table 17. UC-15 View Vehicle Assignment -----	33
Table 18. UC-16 Un-Assign Vehicle -----	34
Table 19. UC-17 View Fuel Request History -----	35
Table 20. UC-18 View Consignments -----	36
Table 21. UC-19 Generate Delivery Sheet -----	37
Table 22. UC-20 Edit Delivery Sheet -----	39
Table 23. UC-21 Delete Delivery Sheet -----	40
Table 24. UC-22 View Delivery Sheet -----	41
Table 25. UC-23 Check out delivery sheet -----	42
Table 26. UC-27 View Areas -----	43
Table 27. UC-28 Edit Areas -----	44

Table 28. UC-29 Logout -----	45
Table 29. TC-1 Login -----	95
Table 30. TC-2 Add new staff profile -----	95
Table 31. TC-3 Request Fuel -----	96
Table 32. TC-4 Add Consignment to Delivery Sheet -----	97
Table 33. TC-5 Generate Delivery Sheet -----	97
Table 34. TC-6 Add Vehicle -----	98

Chapter 1

Software Project Management Plan

(SPMS)

1. Software Project Management Plan

1.1 Introduction

In the past, people just used to post letters and other important official documents via post offices. And it may take days or even weeks for the deliveries to reach their intended destination address. It was a slow, time taking, and not a reliable process/way of sending the important documents. When the internet came, and e-commerce turned common, there arose a need of fast, secure, and reliable shipments. Nowadays people prefer buying products online more often. Since, it takes less time and there is a large online market available around the world.

Therefore, the courier services came with a huge infrastructure of their fleets, routes, accessible points, and delivery personnel. This helped the processing of shipment of a delivery fast, reliable, secure and trustworthy. But something more challenging for the courier managers.

Today, in Pakistan there are many courier services working e.g., TCS, Leopards, DHL etc. TCS is one of the leading courier services of Pakistan that delivers consignments on the doorstep. It is estimated that TCS delivers around 300,000 consignments per day worldwide. And it has a large fleet containing motorbikes, vans, semi-trucks, mini trucks, trailers, cargo planes, etc. And it has a very large number of staff working at different depots and express centers in Pakistan. And it's difficult to manage such a huge system without a proper database management system, therefore they have automated some parts of their system. The parts automated in the system I am developing are not fully automated at TCS.

Courier Managers could not manage their fleets, staff, deliveries/consignments, fuel allocations, and effective processing of deliveries as they grow larger. Therefore, they need a user friendly, easier to use, a comprehensive, and an effective system through which they can simply manage their assets and deliveries. Such a system can be a web application, a desktop application, a mobile application or a combination of both a mobile and web application through which the courier manager can track the daily routine tasks. And the system should ease the manager by carrying out frequent tasks automatically such as generation of delivery sheet, assignment of vehicle to the drivers, and allocation of fuel to the drivers.

This will help the manager to manage without much human intervention through his/her dashboard, which will make it secure, will maximize the profit through optimal decisions.

1.1.1 Project Overview

Last Mile Delivery Management is a hassle full task if done manually. There is a need for a management system with a comprehensive database and user-friendly graphical user interface through which a courier manager can easily manage the consignments, fleet, fuel allocations, and staff.

Last Mile Delivery Management System (LMDMS) which is a web application will help the courier managers to manage the vehicles and deliveries in a simpler and easier way. The database will keep track of the consignments (yet to be delivered, delivered and returned), drivers, vehicles (contractual and company owned), fuel allocations to the drivers for the vehicles, and vehicle assignments to the drivers etc.

LMDMS will allow the supervisors to automatically generate the delivery sheets from consignments for specific areas. This automatic process will be according to the consignments' weights and volumes. At first consignments will be separated on the basis of their area zones (postal codes/zip code). Secondly, Consignments with less than or equal to 2kg weight will be separated from consignments with greater than 2kg weight and are added to the delivery sheet to be assigned to a bike and a rider. And >2kg will be added

to the delivery sheets for the vehicles. And this will be done while considering the vehicle total weight and volume carrying capacity. This process will be near optimal in assigning the vehicle a delivery sheet so that the vehicle would carry the maximum number of deliveries per trip.

Moreover, the fuel is allocated to the drivers while considering the area have to visit for the deliveries. This area will come from the delivery sheet that has just been created. For the areas which requires extra fuel, such areas will get extra fuel automatically. The manager can edit the areas which requires extra fuel from his dashboard.

LMDMS also provide the features of generating the reports on the fuel consumption, vehicle assignments, deliveries made by the drivers on daily, weekly, monthly basis.

The aim of this project is to develop a system that will automate vehicle assignment, optimal delivery sheet generation, allocate the fuel to the drivers on the basis of area zones of deliveries.

At Tranzum Courier Service (TCS), some part of the process is automatic, and some semi-automatic and other manual. My project will be focused on the parts that are manual and need automation. For example:

Fully Automated Systems:

Consignments Management System, Vehicle tracking system, consignment tracking system

Semi-Automated Systems:

Delivery Sheet Generation using Barcode reader, Delivery Management System

Manual Systems:

Vehicle Management System, Drivers Management System, Delivery Sheet Management System, Delivery Sheet Assignment to vehicles, Fuel Allocation System

LMDMS will be focused on the manual and semi-automated systems. And will try to accomplish maximum automation under scope.

1.1.2 Project Deliverables

There are the three deliverables for this project.

- Software Project Management Plan (SPMP)
- Software Requirements Specifications (SRS)
- Software Test Documentation (STD)
- Implementation of the project with complete documentation

1.2 Project Organization

Project organization consists of software process model, roles and responsibilities and tools and techniques.

1.2.1 Software Process Model

I will use Incremental Model as a software process model for this project due to following reasons:

- This model is flexible, less costly to change scope and requirements.
- New increments can be done in low cost.
- It is easier to test and debug during a smaller iteration.
- Easier to manage tasks.
- It is easy to check the progress after each iteration.
- Generate working software earlier during software lifecycle.

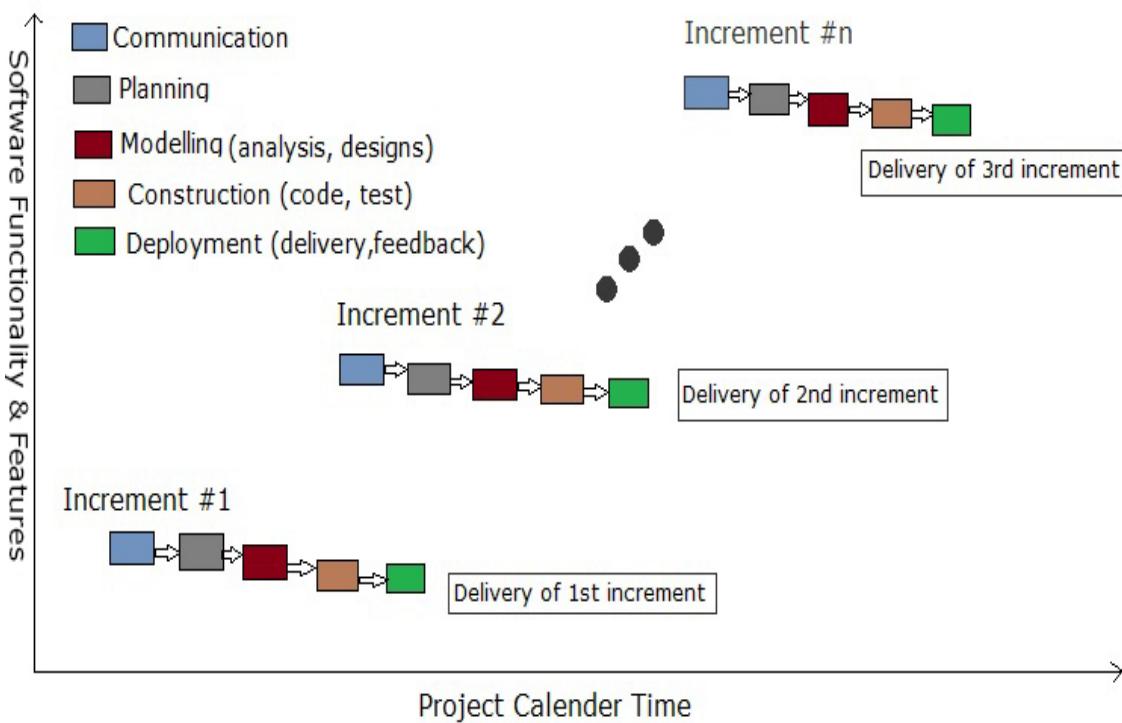


Figure 1. Software Process Model (Incremental Model)

1.2.2 Roles and Responsibilities

I am single developer of this project so there is no division of roles and responsibilities.

1.2.3 Tools and Techniques

Following are the tools used in this project:

Table 1. Tools and Techniques

Tools	Source	Logo
MS-Word	https://products.office.com/en/word	
Project Libre	https://www.projectlibre.com	
StarUML	https://staruml.io/	
Sublime Text	https://www.sublimetext.com/	
Mamp Server	mamp.info/en/mac/	
MySQL	https://www.mysql.com/downloads/	
HTML	https://www.w3schools.com/html/html_intro.asp	
CSS	https://www.w3schools.com/css/css_intro.asp	

1.3 Project Management Plan

Following is the description of project management plan for this project. It explains how time and resources are managed throughout the life cycle of this application.

1.3.1 Tasks

There are two phases of project plan. First is the requirement and analysis and second is the design phase of this Last Mile Delivery Management System. In requirements and analysis phase, the major tasks are to identify scope, requirements, feasibility report, define use cases, develop analysis model, develop SRS and review SRS. In the second phase, the major tasks are to develop a design using Object Oriented Approach, design model of user input, validate input, develop models and evaluate design.

1.3.2 Description

Following is the description of major tasks of both analysis and design phases.

1.3.2.1 Requirement and Analysis Phase

- **Identify Requirements**
The main goal is to review case study and define requirements.
- **Feasibility Report**
The main goal of the report is to check if the project is feasible or not.
- **Define Use Cases**
Define use cases and make a use case diagram, and often analysis models.
- **Develop SRS**
Define functional and nonfunctional requirements and develop software requirement specification document. It includes all other details of product like scope, purpose and introduction.

1.3.2.2 Design Phase

- **Design Architecture**
Refers to the development of the structure of the project.
- **Make User Interfaces**
Extracts the view of the project.
- **Make Sequence Diagram**
Document the system's requirements and to flushes out its design by providing the interaction logic between the objects in the system in the time order that the interactions has taken place.
- **Make Class Diagram**
Describe the attributes and operations of each entity associated with the system and the constraints imposed on the system.

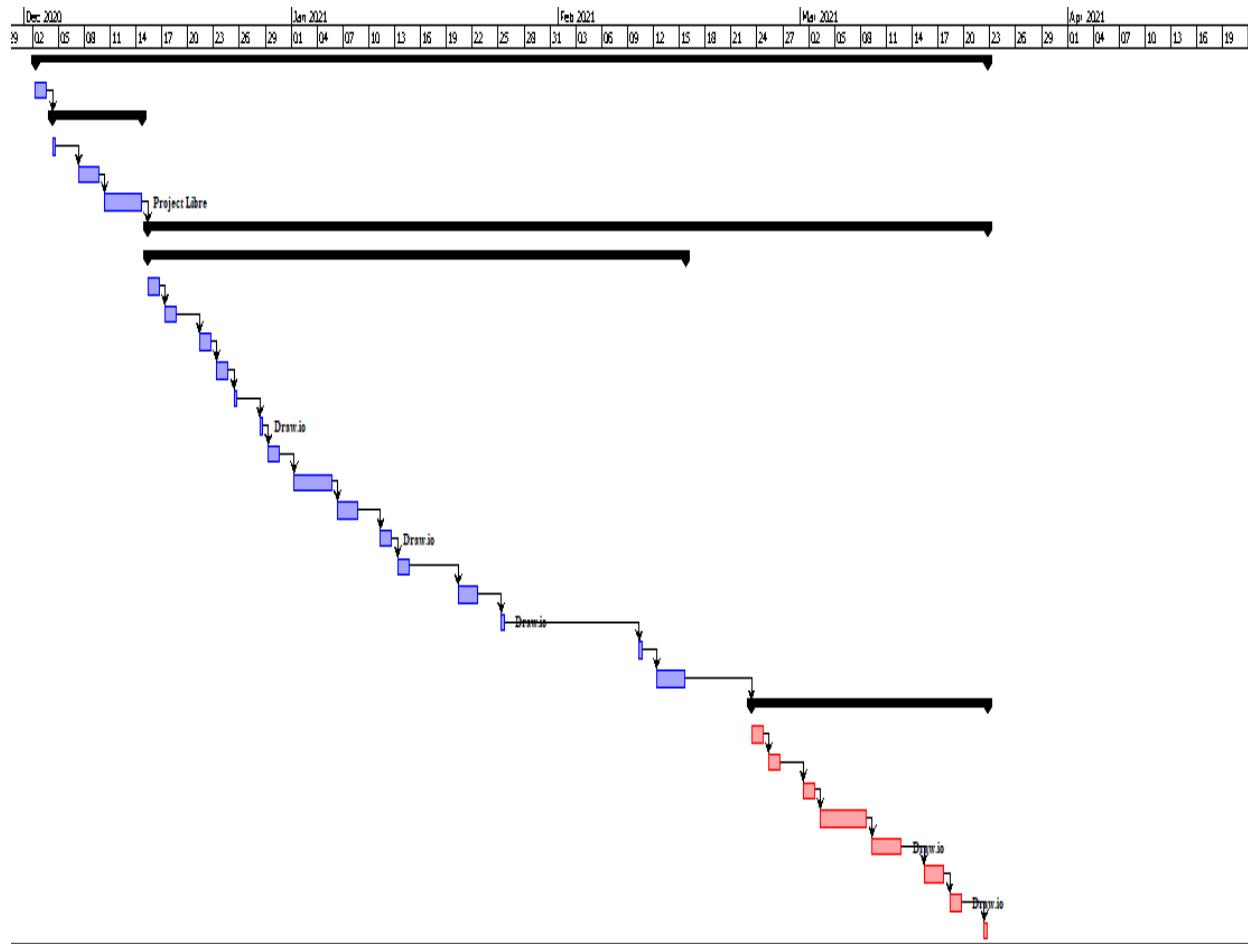


Figure 2. Gantt Chart 1

Chapter 2

Software Requirements Specifications

(SRS)

2. Software Requirements Specification

2.1 Introduction

This section is mainly associated with the overall view of everything included in this SRS document and scope description of the project. Other than that, the purpose of this document is described in this section.

2.1.1 Purpose

The purpose of this SRS document is to clearly define the system under development, namely the Last Mile Delivery Management System (LMDMS). This document covers the functional and non-functional requirements for LMDMS that happen to be essential features of the system that are known, fixed and agreed to be delivered. The intended audience of this document includes the owner of the courier service company. Other intended audience includes the development team or developer.

2.1.2 Mission Statement

Last Mile Delivery Management System (LMDMS) is an information system based on a web application. LMDMS is intended to automate the vehicle assignment, fuel allocation, delivery sheet generation, and consignments reports etc. This will be done by storing the data on the above entities in a comprehensive database and applying various business rules to accomplish the objectives. This application will mainly be used to ease the courier manager and manage the courier operations in a smart and efficient way.

At Tranzum Courier Service (TCS), some part of the process is automatic, and some semi-automatic and other manual. My project will be focused on the parts that are manual and need automation. For example:

Fully Automated Systems:

Consignments Management System, Vehicle tracking system, consignment tracking system

Semi-Automated Systems:

Delivery Sheet Generation using Barcode reader, Delivery Management System

Manual Systems:

Vehicle Management System, Drivers Management System, Delivery Sheet Management System, Delivery Sheet Assignment to vehicles, Fuel Allocation System

LMDMS will be focused on the manual and semi-automated systems. And will try to accomplish maximum automation under scope.

LMDMS for Multiple branches:

The current version of LMDMS is to be developed for a single branch level management. For multi branch setup we need to tailor the LMDMS's logical and conceptual models and subsequent models to fulfill the needs of multiple branches data and their respective users. The branch table is shown in the logical model with its attributes named as branchID, branchName, address, contact, email.

Each branch's data is stored locally for that branch. And after 24 hours all the data from each branch will be archived in a single archive which will then be used for the data visualization of all the branches.

Moreover, a log file will be maintained for each branch that will contain the update queries/changings in the already existing data in a single branch. And this log file will then be concatenated with the other branches' log files.

System Scope Diagram:

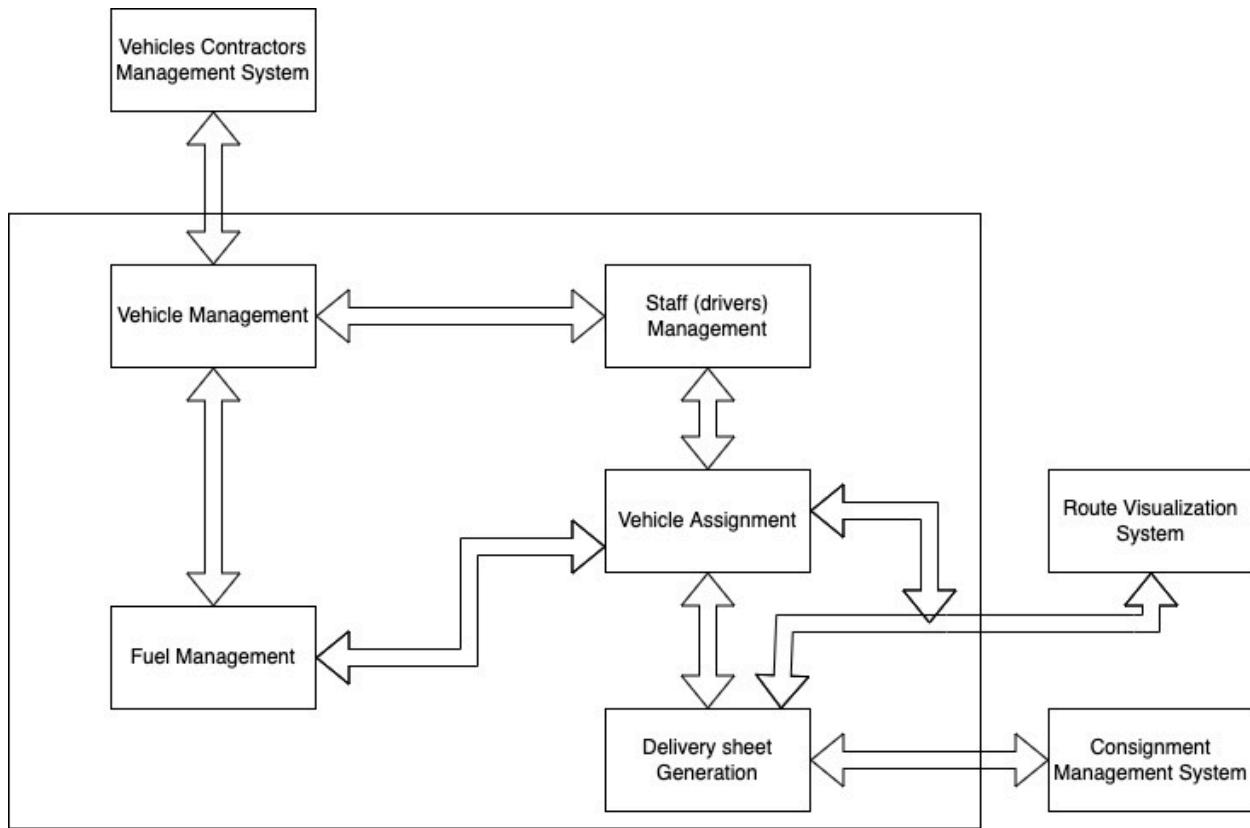


Figure 3. LMDMS Scope Diagram

2.1.2a Mission Objectives

- Add/View/Edit/remove Vehicle Assignment (assigning vehicle to a driver)
- Allocation of fuel on the basis of vehicle
- Allocation of fuel on the basis of area
- Separation of Consignments on the basis of their weights
- Separation of Consignments on the basis of their destination areas
- Separation of Consignments on the basis of their volume
- Generation of Consignment Delivery Sheet on the basis of area
- Optimally assigning the consignments to the vehicles by creating a delivery sheet so that maximum number of consignments can be delivered per trip
- Extra allocation of fuel to the areas that require extra fuel to visit
- Alerting the manager about the remaining fuel if it's lesser than a threshold
- Allowing the manager to request fuel
- Allowing the manager to add/view/edit/remove a staff member
- Allowing the manager to add/view/edit/remove a vehicle

- Allowing the supervisor to generate/check out (verify delivery sheet taken by the driver) /check in (verify delivery sheet submitted by the driver) /edit a delivery sheet
- Allowing the manager to edit the areas as extra fuel area or not
- Allowing drivers to view their profiles
- Allowing the drivers to view the delivery sheet
- Allowing the drivers to view the fuel allocated to them
- Allowing the drivers to submit the delivery sheet and give remarks on the delivery sheet

Constraints

- Internet availability
- Device availability
- Web Application based

2.1.3 Definitions, Acronyms and Abbreviations

Table 2. Definitions, Acronyms and Abbreviations

Term	Definition/Abbreviation
Manager	A person who can control and manage whole system
Delivery Sheet	The consignments list that is generated on the basis of specific area zone and is taken by the driver along with consignments in his vehicle to deliver them
LMDMS	Last Mile Delivery Management System (the system under development)
Supervisor	This role in LMDMS supervises the daily routine tasks like, delivery sheet generation, assignment of vehicle and drivers etc.
Vehicle	A vehicle can be a bike, truck, car, van, semi-truck, hilux etc.
Driver	A driver can be a bike rider, truck driver, car driver, van driver, hilux driver etc.
Fleet	Fleet contains all the vehicles as assets

2.1.4 Overview

Section 1 of the SRS provides a brief introduction about the system to be developed. Section 2 describes the product in more details. It includes its functions, characteristics of users and many more. Section 3 describes the specific requirements in more details.

2.2 Overall Description

2.2.1 Product Perspective

LMDMS is an information system which is a web-based application with database. The system will interface with an external Consignment Management System. It will take information and data regarding the consignments from an external system to create the delivery sheets from them and assign these delivery sheets [1] to the drivers so that the drivers would deliver the consignments to the specified addresses. And it is not fully independent. It is dependent on consignments data. The system provides easy and user-friendly environment to manage the courier operations.

2.2.1.1 System Interfaces

This system interface is a consignment management system from which it pulls consignments data and perform different operations on it such as delivery sheet creation.

2.2.1.2 User Interfaces

The users are the manager of courier service, supervisor, and drivers at courier service company. Users have access to application via their accounts. Manager has the special rights i.e., requesting fuel, managing vehicles, staff etc. Whereas the supervisor can manage the delivery sheet. And the drivers can just view the delivery sheet, their profiles, and submit the delivery sheet with remarks against each delivery.

- Front-end: HTML, CSS, Bootstrap, JavaScript/jQuery
- Back-end: php, MySQL

2.2.1.3 Hardware Interfaces

The hardware interface used for accessing the LMDMS are smart phone or computer.

2.2.1.4 Software Interfaces

The Operating System can be windows or Mac having a working latest web browser which supports HTML5 with proper internet connection.

2.2.1.5 Communication Interfaces

The communication interface is the Graphical user interface (GUI) based web pages through which the manager, supervisors and drivers will communicate with system's database.

2.2.2 Product Functions

Following are the main functions of LMDMS

- User can login from the login interface in the application.
- After signing in all options will be displayed according to the role of the user just logged in.
- Manager can add/view/edit/remove staff accounts and profiles
- Manager can add/view/edit/remove vehicles
- Manager can view/edit/remove/add vehicle assignment
- Manager can view/request Fuel
- Manager can generate reports on Fuel consumption day wise/week wise/month wise
- Manager can generate reports on consignments delivered and returned by drivers
- Manager can view and get report on delivery sheets
- Supervisor can view/edit/remove/generate a delivery sheet
- Supervisor can check out/check in delivery sheet
- Manager can view/edit the areas including areas requiring extra fuel

- Drivers can view their profiles/delivery sheet/submit delivery sheet giving remarks
- Users can log out

2.2.3 User Characteristics

The system expects the user to have basic knowledge of using a smart phone or a computer. The user must be able to understand basic English to use LMDMS.

2.2.4 Constraints

- Internet availability
- Computer or a mobile phone availability
- Web application based

2.2.5 Assumptions and Dependencies

This application needs valid id and login to use application.

2.3 Specific Requirements

2.3.1 External Interfaces

The external interfaces for LMDMS include are smart phone or computer. Mobile touch screen or Computer's keyboard is used for taking inputs and output will be displayed on the screen of the computer or mobile.

2.3.2 Functions

Following are the main functions of LMDMS

- User can login from the login interface in the application.
- After signing in all options will be displayed according to the role of the user just logged in.
- Manager can add/view/edit/remove staff accounts and profiles
- Manager can add/view/edit/remove vehicles
- Manager can view/edit/remove/add vehicle assignment
- Manager can view/request Fuel
- Manager can generate reports on Fuel consumption day wise/week wise/month wise
- Manager can generate reports on consignments delivered and returned by drivers
- Manager can view and get report on delivery sheets
- Supervisor can view/edit/remove/generate a delivery sheet
- Supervisor can check out/check in delivery sheet
- Manager can view/edit the areas including areas requiring extra fuel
- Drivers can view their profiles/delivery sheet/submit delivery sheet giving remarks
- Users can log out

2.3.3 Performance Requirements

LMDMS supports more than 10s of users at a time. It can handle GBs of information smartly. The app must be interactive, and the delays involved must be less. So, in every action response of the application there are no immediate delays. In case of opening different screens and saving the settings there is delay much below 2 seconds. Also, whenever we perform any change on database the change appears in the database instantly within 2 seconds. Pulling large amount of data from database should not crash the system.

2.3.4 Software Quality Attributes

2.3.4.1 Reliability

System should be reliable. There should be no occurrences of failure. The system should be able to work properly all the time. There is only 1% chance of error. System should give 100% performance in any circumstances without any failure.

2.3.4.2 Availability

System must be available to user at any time (24/7). User must be connected with internet to use the system.

2.3.4.3 Security

While using application, there should not be any threats from other software applications. Only authenticated user will have access to the system. Only user with special rights will be able to manage staff, vehicles and fuel data.

2.3.4.4 Portability

Portable with all mobile phones and computer with latest version of the internet web browser which supports HTML5.

2.3.5 Database Requirements

MySQL database may be used by this system to store all the information at the backend, so that the tables can communicate and store data facilitating instantaneous searching, organizing, and reporting. The Last Mile Delivery Management System (LMDMS) will use a relational data model to allow a high degree of data independence.

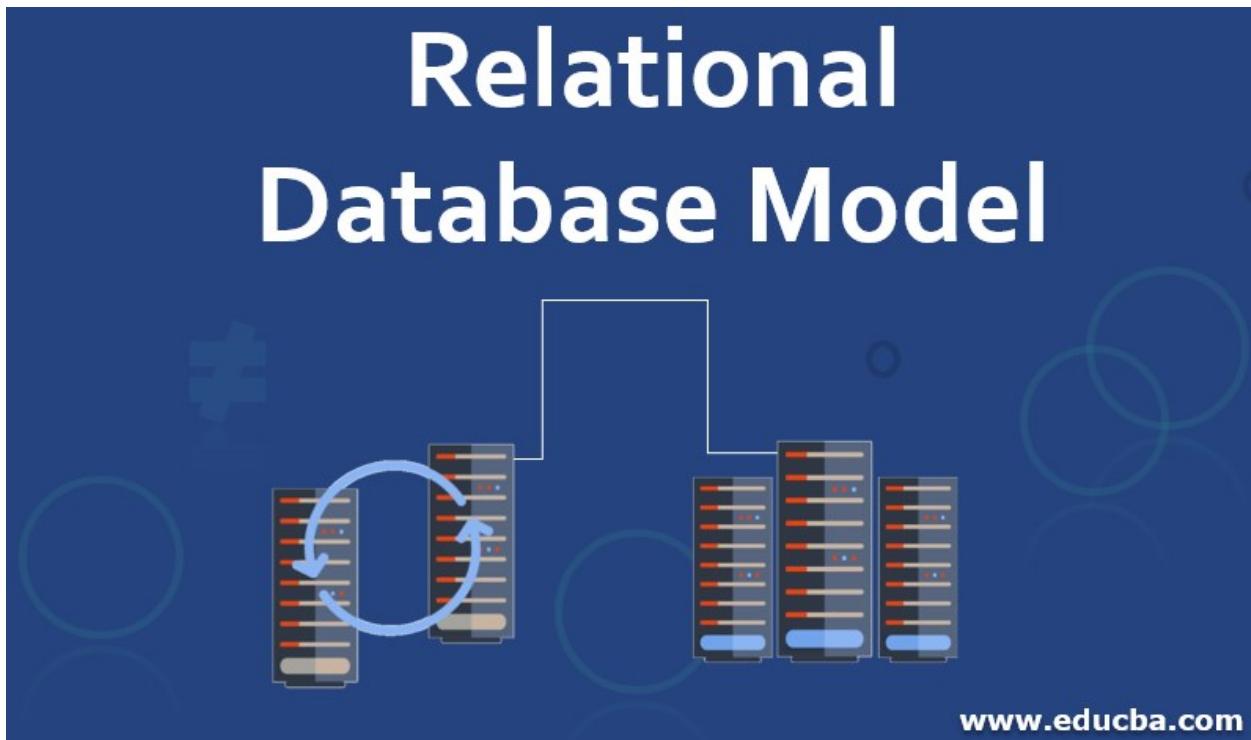


Figure 4. Relational Database Model

2.3.5.1 Conceptual Data Model

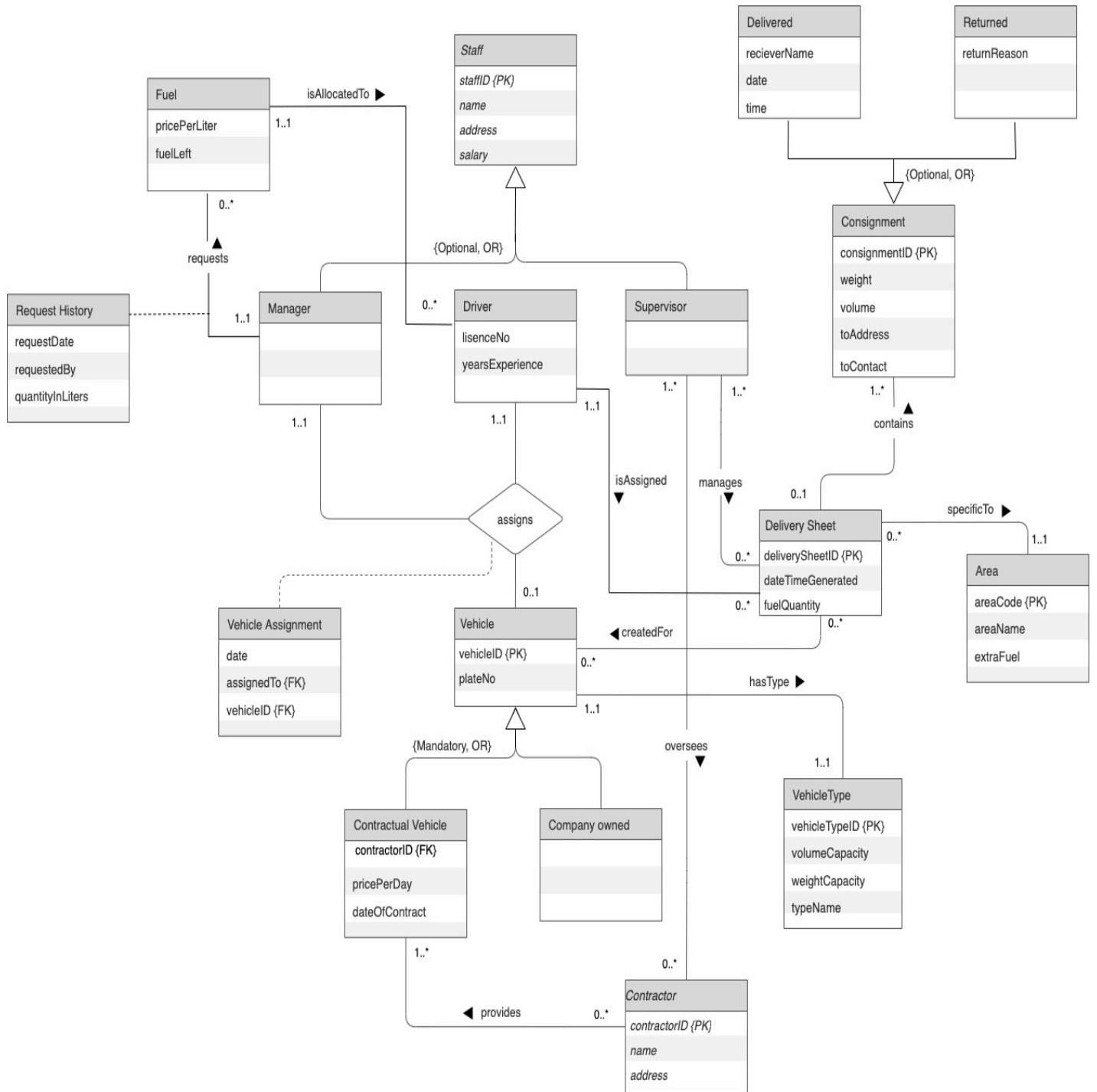


Figure 5. Conceptual Data Model

2.3.5.2 Logical Data Model

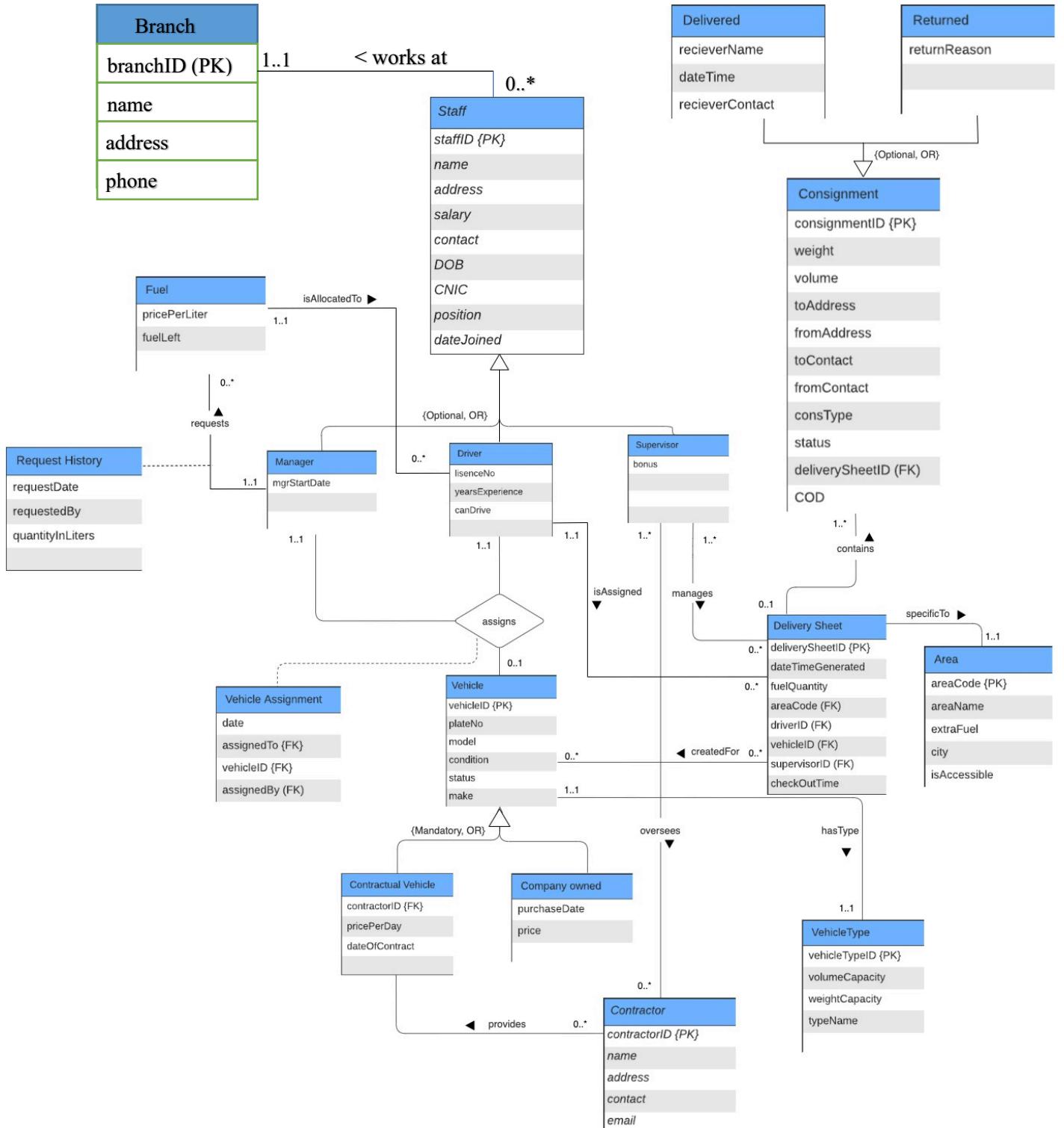


Figure 6. Logical Data Model

2.4 Functional Requirements

2.4.1 Use Cases List

USE CASE ID	USE CASE Name
UC1	Login
UC2	View profile
UC3	Add new staff member's profile
UC4	Edit staff's profile
UC5	Delete staff's profile
UC6	View fuel information
UC7	Request Fuel
UC8	View Fuel consumption
UC9	Add a new vehicle
UC10	Edit vehicle info
UC11	Delete vehicle
UC12	View vehicles
UC13	Assign vehicle to a driver
UC14	Edit vehicle assignment
UC15	View vehicle assignment
UC16	Un-assign vehicle
UC17	View Fuel Request History
UC18	View Consignments
UC19	View Delivery Sheet
UC20	Generate delivery sheet
UC21	Edit delivery sheet
UC22	Delete delivery sheet
UC23	Checks out delivery sheet
UC24	View Areas
UC25	Edit Areas
UC26	Logout

2.4.2 Use Case Diagram

Figure 7. Use Case Diagram



2.4.3 Use Case Descriptions

Following are the use case descriptions.

2.4.3.1 UC-1 Login

Table 3. UC-1 Login

ID	UC-1
Name	Login
Primary Actors	User (Manager, Supervisor, Driver)
Stakeholders	Developer of the app, User, Admin
Description	The use case, “Login” is performed by the User to log in to the system to perform various functionalities using Last Mile Delivery Management System (LMDMS).
Pre-condition	User/Admin must be authorized and registered.
Post-condition	<ul style="list-style-type: none"> • User logged in. • Access to the application is provided
Main Success Scenario	<ol style="list-style-type: none"> 1. User opens the web application. 2. System shows the Log in screen 3. User enters username. System verifies it. 4. User enters password. System verifies it. 5. User successfully logs into the system and is taken to the respective home page.
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. User/Admin restarts the application. 1a. The application crashed. 4a. Incorrect password. <ol style="list-style-type: none"> 1. Application allows to re-enter password. 4b. Forget password. <ol style="list-style-type: none"> 1. Application allows to reset password. 4a. System is down, therefore account can't be accessed.
Special Requirements	<ul style="list-style-type: none"> • After entering username and password, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Few times a day, unless cookies are maintained at the user's browser.

Technology and Data Variation List	There should be no other way to login in the system.
Priority	High

2.4.3.2 UC-2 Add Staff

Table 4. UC-2 Add staff member

ID	UC-2
Name	Add staff member
Primary Actors	Manager
Stakeholders	Manager, staff member
Description	The use case, “Add staff member” is performed by the Manager after which staff member will be able to use the system
Pre-condition	Manager must be authorized and authenticated.
Post-condition	Staff member added successfully.
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager clicks on the “Add Staff Member” button. 2. The system opens a form to enter the data for the new staff member 3. The manager enters username 4. Manager enters the password 5. Manager re-enters the password 6. Manager enters name, email, address, contact, salary of the staff member 7. Manager selects a role for the staff member’s account (manager, supervisor, driver) 8. Manager presses “Add” button 9. The system confirms the new entry of a staff member profile
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ol style="list-style-type: none"> 1. Admin restarts the application. 3a. Manager leaves the username field empty <ol style="list-style-type: none"> 1. System tells the manager that this field cannot be empty 3b. Manager enters the wrong format in the username field <ol style="list-style-type: none"> 1. System shows an error message 2. System asks for the username again 3c. Manager enters a username already taken

	<p>1. System tells the manager that username is already taken</p> <p>4a. Manager leaves the password field empty</p> <p>4b. Manager enters a weak password</p> <p> 1. System allows to re-enter password</p> <p>5a. The manager enters a password that doesn't match the already entered password in the re-enter password field</p> <p> 1. System tells that manager that the password doesn't match with the set password</p> <p>9a. System shows an alert to fill the empty fields</p> <p>9b. System shows an alert to enter the correct type and format of data</p>
Special Requirements	<ul style="list-style-type: none"> On every action, system responds within 30 seconds 90% of the time. Language internationalization on the text displayed.
Frequency	Occasionally
Technology and Data Variation List	There should be no other way to add a new staff member to the system
Priority	Low

2.4.3.3 UC-3 View Profile

Table 5. UC-3 View Profile

ID	UC-3
Name	View Profile
Primary Actors	Staff members (Manager, Supervisor, Driver)
Stakeholders	Developer of the app and Users
Description	View profile use case is performed by the users of the system to view their profiles.
Pre-condition	User must be authenticated i.e., logged in to the system
Post-condition	User has viewed his/her profile successfully
Main Success Scenario	<ol style="list-style-type: none"> User log in to the system. System shows user's username in the left corner. User press his username.

	4. System shows all the information stored on the user. 5. User views his information.
Alternate Scenario or Extensions	*a. At any time, system may go down. 1. User restarts the application. 1a. User forgot his/her password. 1. System shows the forgot password option. 2. User proceeds to the reset password use case. 3. System updates the password of the user.
Special Requirements	<ul style="list-style-type: none"> On button presses, system responds within 30 seconds 90% of the time. Language internationalization on the text displayed.
Frequency	Few times a week
Technology and Data Variation List	There should be no other way to view the user profile. Only an authenticated user can view his/her profile. And only manager can look and view the user's profiles.
Priority	Medium

2.4.3.4 UC-4 Edit Staff's profile

Table 6. UC-4 Edit staff's profile

ID	UC-4
Name	Edit staff's profile
Primary Actors	Manager
Stakeholders	Developer of the app and Manager
Description	Edit staff profile use case is carried out by the manager to update the profile of a staff member like his name, dob, password, address etc.
Pre-condition	Manager must be logged in and the staff member's profile exists in the system.
Post-condition	Profile is updated successfully with the changes
Main Success Scenario	1. Manager selects the staff member whose profile has to be updated. 2. System shows the staff member's profile and edit option. 3. Manager selects the edit option. 4. System shows the editable fields.

	<p>5. Manager enters the information to be updated.</p> <p>6. Manager submits the updated profile.</p> <p>7. System updates the profile and stores the updated information of the user.</p> <p>8. System acknowledges the manager that the profile has been updated successfully and shows the updated profile to the manager.</p>
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <p>1. Manager restarts the application.</p> <p>5a. Manager enters the incorrect data w.r.t datatypes.</p> <p>1. System alerts the manager to enter the correct type of data and format.</p>
Special Requirements	<ul style="list-style-type: none"> On button press, system responds within 30 seconds 90% of the time. Language internationalization on the text displayed.
Frequency	Rarely
Technology and Data Variation List	There should be no other way to edit the staff profile, only manager can do this and from his account.
Priority	Medium

2.4.3.5 UC-5 Delete Staff's profile

Table 7. UC-5 Delete Staff Profile

ID	UC-5
Name	Delete Staff's profile
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	Only manager can perform this use case when any staff member's leaves the company, or an incorrect staff is entered into the system.
Pre-condition	The staff profile must exist in the system already
Post-condition	Staff profile is deleted successfully
Main Success Scenario	<ol style="list-style-type: none"> Manager selects the staff member whose profile has to be deleted System shows the staff member's profile and delete option Manager selects the delete option System asks for the confirmation of deletion of the profile System asks the manager to enter his password

	<p>6. Manager enters his password and press confirm deletion</p> <p>7. System deletes the staff profile and shows a confirmation message that the profile has been deleted successfully.</p>
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. Manager restarts the application. <p>4a. Manager selects the do not delete option.</p> <ol style="list-style-type: none"> 1. System returns back to the staff profiles' window. 2. Manager views the staff member's profiles in a tabular form <p>5a. Manager enters a wrong password.</p> <ol style="list-style-type: none"> 1. System asks the manager to enter the password again as the password entered is wrong. 2. Managers enters the password again. <p>6a. Manager press cancel button</p> <ol style="list-style-type: none"> 1. System goes back to previous page and doesn't perform the action of deletion
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Rarely
Technology and Data Variation List	There should be no other way to delete a staff member's profile. Only manager can perform this use case.
Priority	High

2.4.3.6 UC-6 View Fuel information

Table 8. UC-6 View fuel information

ID	UC-6
Name	View Fuel Information
Primary Actors	Driver, Manager
Stakeholders	Developer of the app, Manager, Driver
Description	This use case is carried out by the driver and manager to view the fuel information like price, quantity allocated etc.

Pre-condition	User is logged in.
Post-condition	Fuel information is viewed successfully.
Main Success Scenario	<ol style="list-style-type: none"> 1. User logs in to the system. 2. System shows different options including Fuel option in the menu. 3. User selects the Fuel option. 4. System shows the fuel information including price, quantity allocated, date and time etc. 5. User views the information.
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ol style="list-style-type: none"> 1. Restart the application. 4a. If the user is manager, then he/she can request the fuel. <ol style="list-style-type: none"> 1. System shows the request fuel option. 2. Manager proceeds with the request fuel use case.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Sometimes in a day
Technology and Data Variation List	There should be no other way to view the fuel information. Only logged in users can view their fuel allocation.
Priority	Low

2.4.3.7 UC-7 Request Fuel

Table 9. UC-7 Request Fuel

ID	UC-7
Name	Request Fuel
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	In this use case, the manager requests the fuel from the upper management by giving his/her credentials and the quantity needed for his floor.
Pre-condition	Manager is logged in, fuel quantity is below a threshold.
Post-condition	Fuel has been given successfully.

Main Success Scenario	<ol style="list-style-type: none"> 1. Manager views the fuel information. 2. System checks if the fuel left in the floor right now is less than the threshold quantity. If it's less then it enables the request button. 3. Manager selects this option. 4. System shows the input fields for the quantity to be requested, and any description. 5. Managers enters the quantity which is again can't exceed a limit. 6. Managers press the send request option. 7. System asks for the manager's account password. 8. Manager enters the password and submits the request. 9. System sends this request to another system or to the owner of the LMDMS and asks for confirmation of allowance of fuel to the manager. 10. Once confirmed, the manager is notified about the fuel quantity given by the system. 11. System updates the fuel quantity allocated to the manager for the floor and shows the new quantity.
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ul style="list-style-type: none"> 1. Admin restarts the application. 2a. If the remaining quantity is not less than the threshold. <ul style="list-style-type: none"> 1. System keeps the request fuel option disabled. 5a. Manager enters incorrect quantity or description. <ul style="list-style-type: none"> 1. System alerts the manager to enter the correct data. 7a. If not confirmed, manager can send a reminder with reference to the request. <ul style="list-style-type: none"> 1. System shows a send a reminder option. 2. Manager selects this option. 3. System sends a reminder to the owner of the LMDMS on the request. 7b. If the request is refused. <ul style="list-style-type: none"> 1. System notifies the manager about the refusals and the reasons if any. 2. System allows the manager to request again. 8a. Manager enters a wrong password. <ul style="list-style-type: none"> 1. System asks the manager to enter the password again. 2. Manager re-enters the password.

Special Requirements	<ul style="list-style-type: none"> On button press, system responds within 30 seconds 90% of the time. Language internationalization on the text displayed.
Frequency	Once in a month.
Technology and Data Variation List	There should be no other way to request fuel. Only manager can request fuel by giving his/her credentials.
Priority	High

2.4.3.8 UC-8 View Fuel Consumption

Table 10. UC-8 View Fuel Consumption

ID	UC-8
Name	View Fuel Consumption
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	Manager wants to view the fuel consumption by drivers day-wise, week-wise, month-wise
Pre-condition	Manager must be logged in, and fuel must be consumed
Post-condition	Manager has viewed the fuel consumption by the drivers
Main Success Scenario	<ol style="list-style-type: none"> Manager logs into the account. System shows dashboard with a menu containing an option of Fuel Management Manager selects the Fuel Management Option System shows a drop-down menu containing different options related to Fuel Management. Manager selects the Fuel Consumption Option. System shows the Fuel Consumption by each driver in the tabular form along with the delivery sheets information. Manager views the Fuel Consumption; manager can also filter the results.
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> Restart the application.

	<p>7a. Manager selects the month-wise report of consumption.</p> <ol style="list-style-type: none"> 1. System shows the month-size fuel consumption by the drivers. <p>7b. Manager selects the week-wise report of consumption.</p> <ol style="list-style-type: none"> 1. System shows the week-size fuel consumption by the drivers.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed. • System should generate printable report in MS-Excel Format.
Frequency	Occasionally
Technology and Data Variation List	There should be no other way to view the fuel consumption. Only manager can view the fuel consumptions by the drivers.
Priority	Low

2.4.3.9 UC-9 Add a Vehicle

Table 11. UC-9 Add a vehicle

ID	UC-9
Name	Add a vehicle
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	Manager wants to add a new vehicle (truck, car, van, shahzore, bike) into the fleet in LMDMS
Pre-condition	Manager must be logged in.
Post-condition	Manager has successfully added a vehicle into the fleet
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager logs into the system. 2. Manager selects the vehicle management option from the menu bar. 3. System shows a drop down sub menu for the vehicle management. 4. Manager selects the add vehicle option. 5. System shows the form to enter the new vehicle information and submit it. 6. Manager enters the data into the fields and hit submit button. 7. System checks if all the required fields have been filled with the correct type of data.

	<p>8. System adds a new vehicle record into the database.</p> <p>9. System shows a confirmation message that the vehicle has been added to the fleet successfully.</p>
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. Restart the application. <p>6a. Manager doesn't enter the correct datatype data into the fields.</p> <ol style="list-style-type: none"> 1. System alerts the manager to enter the correct type of data into the field. <p>7a. Manager leaves a required field empty.</p> <ol style="list-style-type: none"> 1. System asks the manager to fill the required fields. 2. Manager performs the step 6 of Main success scenario.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Occasionally
Technology and Data Variation List	There should be no other way to add a new vehicle into the fleet, and only manager can perform this use case.
Priority	Medium

2.4.3.10 UC-10 Edit Vehicle Information

Table 12. UC-10 Edit Vehicle Information

ID	UC-10
Name	Edit Vehicle information
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	Manager wants to edit the vehicle information that exists in the fleet
Pre-condition	Manager is logged in and has opened the vehicle profile page
Post-condition	The vehicle information has successfully updated
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager selects the edit vehicle option 2. System shows the editable fields of the vehicle 3. Manager enters the updated information 4. Manager press submit button

	<p>5. System updates the record of the vehicle</p> <p>6. System shows a confirmation message that the record is successfully updated for the vehicle</p>
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <p>1. Restart the application.</p> <p>3a. Manager enters any wrong type of data into the field</p> <p>1. System alerts the manager to enter the correct type of data</p> <p>4a. Manager press cancel button</p> <p>1. System goes to the previous page without updating the record of the vehicle</p>
Special Requirements	<ul style="list-style-type: none"> On button press, system responds within 30 seconds 90% of the time. Language internationalization on the text displayed.
Frequency	Occasionally
Technology and Data Variation List	There should be no other way to update/edit the vehicle information and only manager can perform this action.
Priority	Medium

2.4.3.11 UC-11 Delete Vehicle

Table 13. UC-11 Delete Vehicle

ID	UC-11
Name	Delete Vehicle
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	Manager wants to delete a record of a vehicle
Pre-condition	Manager is logged in, the vehicle exists in the system and he is on the vehicle profile page
Post-condition	Vehicle is deleted from the fleet successfully
Main Success Scenario	<ol style="list-style-type: none"> Manager selects the vehicle whose profile has to be deleted System shows the vehicle profile and delete option Manager selects the delete option

	<ol style="list-style-type: none"> 4. System asks for the confirmation of deletion of the profile 5. System asks the manager to enter his password 6. Manager enters his password and press confirm deletion 7. System deletes the vehicle profile 8. System shows a confirmation message that the profile has been deleted successfully.
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. 2. Manager restarts the application. 4a. Manager selects the do not delete option. 3. System returns back to the vehicles' profile window. 4. Manager views vehicles' profiles 6a. Manager enters a wrong password. 3. System asks the manager to enter the password again as the password entered is wrong. 4. Managers enters the password again. 6b. Manager press cancel button 1. System goes back to previous page and doesn't perform the action of deletion
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Rarely
Technology and Data Variation List	There should be no other way to delete the profile of any vehicle and only manager does this.
Priority	High

2.4.3.12 UC-12 View Vehicle

Table 14. UC-12 View Vehicle

ID	UC-12
Name	View Vehicle
Primary Actors	User (Manager, Driver, Supervisor)

Stakeholders	Developer of the app, Manager, driver supervisor
Description	The user wants to view a vehicle profile
Pre-condition	User is logged in to the system
Post-condition	User has viewed profile of a vehicle successfully
Main Success Scenario	<ol style="list-style-type: none"> 1. User logs in to the system 2. User goes to the view vehicle option 3. System shows the vehicle profile 4. User views the vehicle profile
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ol style="list-style-type: none"> 1. Restart the application. 3a. In case if the user is supervisor or a manager <ol style="list-style-type: none"> 1. System shows all the vehicle in the tabular form 2. System gives option to view a single vehicle profile 3b. In case if the user is a driver <ol style="list-style-type: none"> 1. System shows the vehicle profile that's assigned to the
Special Requirements	NULL
Frequency	Many times in a day
Technology and Data Variation List	There should be no other way to turn lights off automatically in the system.
Priority	High

2.4.3.13 UC-13 Assign Vehicle

Table 15. UC-13 Assign Vehicle

ID	UC-13
Name	Assign Vehicle
Primary Actors	Admin, User, Manager
Stakeholders	Developer of the app, Manager
Description	Manager wants to assign an unassigned vehicle to a driver

Pre-condition	Manager must be logged in and the vehicle must not be already assigned to any driver
Post-condition	Vehicle has been assigned to a driver successfully
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager opens the vehicle management tab from the main menu 2. System shows the vehicle that are not assigned yet to any of the driver 3. Manager selects the vehicle that is to be assigned to any driver 4. System shows the vehicle profile and a button to assign to a driver 5. Manager presses the button to assign 6. System recommends the suitable available drivers based on the experience of driving a particular vehicle 7. Manager selects the driver to be assigned to and hit save 8. System saves the assignment along with time stamp, manager id who assigned, and assignment information 9. System shows a confirmation message to the manager that the vehicle is successfully assigned to the driver.
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. Admin restarts the application. <p>6a. In case if there is no driver available who is not assigned any vehicle yet.</p> <ol style="list-style-type: none"> 1. System shows the message that there is no driver available
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Sometimes in a month
Technology and Data Variation List	There should be no other way to assign a vehicle to a driver, and this can only be done by the manager
Priority	Medium

2.4.3.14 UC-14 Edit Vehicle Assignment

Table 16. UC-14 Edit Vehicle Assignment

ID	UC-14
Name	Edit vehicle assignment
Primary Actors	Manager
Stakeholders	Developer of the app, manager

Description	Manager wants to edit the vehicle assignment information like date, time, the driver assigned and the vehicle assigned
Pre-condition	Manager is logged in and has opened the vehicle assignment tab
Post-condition	The vehicle assignment has been successfully updated
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager goes to the vehicle assignment tab 2. System shows all the vehicle assignments and edit button corresponding to each vehicle assignment 3. Manager press the edit option 4. System shows the editable fields of the vehicle assignment 5. Manager updates the fields and hit save 6. System checks and save the updated information about the vehicle assignment 7. System shows a confirmation message that the vehicle assignment has been successfully updated
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ol style="list-style-type: none"> 1. Admin restarts the application. 5a. Manager enters the wrong type of data or incorrect data <ol style="list-style-type: none"> 1. System alerts the manager to enter the correct type of data 2. System allows the manager to enter the data 3. Manager enters the data and continues from step 5 of main success scenario 5a. Manager presses the cancel button instead of save button <ol style="list-style-type: none"> 1. System goes back to the previous window without updating the record
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Few times in a week
Technology and Data Variation List	There should not be any other way to update the vehicle assignment.
Priority	Medium

2.4.3.15 UC-15 View Vehicle Assignment

Table 17. UC-15 View Vehicle Assignment

ID	UC-15
Name	View Vehicle Assignment
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	Manager wants to view the vehicle assignments to the drivers
Pre-condition	Manager is logged in and
Post-condition	Manager has viewed the vehicle assignments
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager goes to the vehicle management menu 2. System shows different options including vehicle assignments 3. Manager selects the vehicle assignment option 4. System shows all the vehicle assignments to the drivers 5. System shows the vehicles and corresponding assigned drivers in a tabular form with different filters related to the type of the vehicle, and date 6. Manager views the vehicle assignments
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. Manager restarts the application.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Sometimes in a day
Technology and Data Variation List	There should be no other way to view the vehicle assignments to the drivers
Priority	Low

2.4.3.16 UC-16 Un-Assign Vehicle

Table 18. UC-16 Un-Assign Vehicle

ID	UC-16
Name	Un-Assign Vehicle
Primary Actors	Manager
Stakeholders	Developer of the app, Manager

Description	Manager wants to unassign a vehicle assigned to a driver
Pre-condition	Manager is logged in to the LMDMS
Post-condition	Vehicle has been un assigned successfully and can be assigned to a new driver
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager goes to the vehicle management option 2. System shows different options including vehicle assignment 3. Manager goes to the vehicle assignment 4. System shows the vehicle assignments and options to un-assign the vehicle 5. Manager press the un-assign button for a particular vehicle that he wants 6. System updates the database with this un assignment and the vehicle is now ready to be assigned to some new driver 7. System shows the confirmation message after updating that the vehicle is un—assigned successfully.
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ol style="list-style-type: none"> 1. Manager restarts the application. 5a. Manager doesn't find the vehicle assignment that he was looking <ol style="list-style-type: none"> 1. System has given a search option in the top bar 2. Manager searches for the particular vehicle in the search bar 3. System shows the results for the particular vehicle 4. Manager continues from step 5 of the main success scenario 5b. In case if the manager mistakenly un assign a wrong vehicle <ol style="list-style-type: none"> 1. System allows the manager to undo
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Occasionally
Technology and Data Variation List	There should not be any other way to un assign the vehicle, and only manager can perform this action
Priority	Medium

2.4.3.17 UC-17 View Fuel Request History

Table 19. UC-17 View Fuel Request History

ID	UC-17
Name	View Fuel Request History
Primary Actors	Manager
Stakeholders	Developer of the app, Manager
Description	Manager wants to view the fuel request history
Pre-condition	Manager is logged in and in the fuel management section of LMDMS
Post-condition	Manager has viewed the fuel request history
Main Success Scenario	<ol style="list-style-type: none"> 1. Manager press the fuel request history option in the menu 2. System shows the fuel request history with different filters like time period, request by etc. 3. System allows the manager to generate a report on fuel request history in tabular form 4. Manager views the fuel request history
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. Manager restarts the application.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Sometimes in a month
Technology and Data Variation List	There should not be any other way to view the fuel request history and only authorized person and only a manager can view this.
Priority	Low

2.4.3.18 UC-18 View Consignments

Table 20. UC-18 View Consignments

ID	UC-18
Name	View Consignments
Primary Actors	User (Manager, Supervisor)
Stakeholders	Developer of the app, Manager, Supervisor

Description	Manager or a supervisor wants to view the consignments data
Pre-condition	User is logged in
Post-condition	User has viewed the consignments data in a tabular form
Main Success Scenario	<ol style="list-style-type: none"> 1. User goes to the Consignments option 2. System shows all the consignments in a tabular form with status as yet to be delivered, delivered, or returned 3. User views the consignments
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. User restarts the application.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Many times in a day.
Technology and Data Variation List	There should be no other way to view the consignments data.
Priority	Medium

2.4.3.19 UC-19 Generate Delivery Sheet

Table 21. UC-19 Generate Delivery Sheet

ID	UC-19
Name	Generate Delivery Sheet
Primary Actors	User (Manager, Supervisor)
Stakeholders	Developer of the app, Manager, Supervisor
Description	User wants to generate the delivery sheet for a particular area
Pre-condition	User must be logged in and there must be consignments that are not delivered for a delivery sheet to be created
Post-condition	Delivery sheet has been created successfully
Main Success Scenario	<ol style="list-style-type: none"> 1. User goes to the delivery sheet management option of LMDMS 2. System shows different options including generate delivery sheet 3. User press the generate delivery sheet option

	<ol style="list-style-type: none"> 4. System shows option to select the area code for which the delivery sheet is to be generated and to select the vehicle 5. User selects an area code and hits next 6. System generates the delivery sheet on the basis of the consignments that are not delivered yet, weight of consignments 7. System presents a delivery sheet and asks for confirmation 8. User confirms the delivery sheet 9. System shows the delivery sheet to the driver for whom it is created for 10. System saves the delivery sheet ids against the consignments and it also saves the generator, and confirmed by.
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ul style="list-style-type: none"> 1. User restarts the application. 5a. User selects an area code for which there is no consignment available to be delivered <ul style="list-style-type: none"> 1. System shows a message that there is no consignment for this area 8a. User doesn't confirm the delivery sheet <ul style="list-style-type: none"> 1. System asks the user to confirm again, if still not 2. System deletes the delivery sheet 3. System allows the user to generate a new delivery sheet 7a. User press the edit delivery sheet option <ul style="list-style-type: none"> 1. Includes use case <i>Edit Delivery Sheet</i>
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Many times a day.
Technology and Data Variation List	There should not be any other way to create delivery sheet.
Priority	High

2.4.3.20 UC-20 Edit Delivery Sheet

Table 22. UC-20 Edit Delivery Sheet

ID	UC-21
Name	Edit Delivery Sheet
Primary Actors	User (Manager, Supervisor)
Stakeholders	Developer of the app, Manager, Supervisor
Description	User wants to edit a delivery sheet by allocating a driver, changing vehicle, editing the consignments in the delivery sheet, and other editing operations on the delivery sheet
Pre-condition	User must be logged in, and a delivery sheet is created
Post-condition	Delivery sheet successfully updated
Main Success Scenario	<ol style="list-style-type: none"> 1. User views the delivery sheet. 2. System shows options including 'Edit delivery sheet' on the viewing page 3. User goes to the edit delivery sheet option. 4. System shows editable driver assignment, change vehicle, allocate fuel, edit consignment etc. 5. User performs the edit operation accordingly and hit save or cancel. 6. System checks if the edited information is correct and updates the delivery sheet. 7. System shows a confirmation message that the delivery sheet has been updated successfully.
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ol style="list-style-type: none"> 1. User restarts the application. 5a. User press cancel button. <ol style="list-style-type: none"> 1. System goes back to the previous page without updating the delivery sheet. 6a. The entered information is incorrect or have wrong type of data. <ol style="list-style-type: none"> 1. System alerts the user about the above situation 2. System allows the user to enter the correct type of data. 3. User continues from step 4 of Main success scenario.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Sometimes in a day

Technology and Data Variation List	There should not be any other way to edit the delivery sheet, and only manager and supervisor can perform this use case.
Priority	Medium

2.4.3.21 UC-21 Delete Delivery Sheet

Table 23. UC-21 Delete Delivery Sheet

ID	UC-21
Name	Delete Delivery Sheet
Primary Actors	User (Manager, Supervisor)
Stakeholders	Developer of the app, User
Description	User wants to delete a generated delivery sheet
Pre-condition	<ul style="list-style-type: none"> • The delivery sheet must be created for any area • The user must be logged in • Delivery sheet is not yet confirmed and checked out by the supervisor
Post-condition	User has successfully deleted the delivery sheet
Main Success Scenario	<ol style="list-style-type: none"> 1. User is viewing the delivery sheet 2. System is showing different options including 'Delete delivery sheet' 3. User presses delete delivery sheet 4. System asks for confirmation 5. User confirms the deletion 6. System deletes the delivery sheet and confirms with a message
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <p>3. Restart the application.</p> <p>5a. User doesn't confirm the deletion</p> <ol style="list-style-type: none"> 1. System goes back to the viewing delivery sheet page without deleting the delivery sheet
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Occasionally
Technology and Data Variation List	There should not be any other way to delete a delivery sheet
Priority	Medium

2.4.3.22 UC-22 View Delivery Sheet

Table 24. UC-22 View Delivery Sheet

ID	UC-22
Name	View Delivery Sheet
Primary Actors	User (Manager, driver, supervisor)
Stakeholders	Developer of the app, User (Manager, driver, supervisor)
Description	User wants to view a delivery sheet that is generated by the system
Pre-condition	User must be logged in and the delivery sheet must be generated
Post-condition	User has viewed the delivery sheet successfully
Main Success Scenario	<ol style="list-style-type: none"> 1. User goes to the delivery sheet management section of LMDMS 2. System shows generated delivery sheets in a table 3. User presses the view delivery sheet to view a specific delivery sheet in the table 4. System takes user to the next page and shows the full delivery sheet with consignments that are in the delivery sheet, area code, driver and vehicle assigned and the fuel that is to be allocated for this trip 5. User views the delivery sheet
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. User restarts the application.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Rarely
Technology and Data Variation List	There should be no other way to check Sensor status in the system.
Priority	Low

2.4.3.23 UC-23 Check out Delivery Sheet

Table 25. UC-23 Check out delivery sheet

ID	UC-23
Name	Check out delivery sheet
Primary Actors	User (Manager, Supervisor)
Stakeholders	Developer of the app, User (Manager, Supervisor)
Description	User wants to check out the delivery sheet which means he/she has to confirm and proceed the delivery sheet forward towards the driver for them to view it and go on the route to deliver the consignments
Pre-condition	User is logged in and the delivery sheet exists
Post-condition	The delivery sheet is checked out successfully
Main Success Scenario	<ol style="list-style-type: none"> 1. User views the delivery sheet 2. System shows options including 'Check out Delivery sheet' 3. User proceeds with the check-out option 4. System saves the information of the checker and date and time of check out 5. System checks out the delivery sheet and shows it to the driver with a notification that the delivery sheet has been generated and checked out for the deliveries 6. System shows the confirmation message that the delivery sheet has been checked out successfully.
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. User restarts the application. <p>3a. User presses the cancel button or doesn't proceed.</p> <ol style="list-style-type: none"> 1. System doesn't proceed and next time when user will generate another delivery sheet for the same area, it'll show the previously generated sheet to be checked out. 2. User continues with the step 3 of Main success scenario.
Special Requirements	<ul style="list-style-type: none"> • On button press, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Many times a day
Technology and Data Variation List	There should be no other way to check out the delivery sheet

Priority	Medium
-----------------	--------

2.4.3.24 UC-24 View Areas

Table 26. UC-24 View Areas

ID	UC-24
Name	View Areas
Primary Actors	User (Manager, Supervisor)
Stakeholders	Developer of the app, User (Manager, Supervisor)
Description	User wants to view the areas that the current depot is handling for deliveries
Pre-condition	User must be logged in to the system and there must be entries in the areas to view
Post-condition	User has successfully viewed the areas under the depot
Main Success Scenario	<ol style="list-style-type: none"> 1. User goes to the areas section 2. System shows all the areas' information in a tabular form including the zip code, name, and whether an extra fuel is required for the area or not 3. User views the areas
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 1. User restarts the application.
Special Requirements	<ul style="list-style-type: none"> • After pressing the button, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Few times a day
Technology and Data Variation List	There should be no other way to submit the delivery sheet
Priority	High

2.4.3.25 UC-25 Edit Areas

Table 27. UC-25 Edit Areas

ID	UC-28
Name	Edit Areas
Primary Actors	User (Manager, Supervisor)
Stakeholders	Developer of the app, User (Manager, Supervisor)
Description	User wants to edit the areas under the depot
Pre-condition	User is logged in and there exists areas that is to be edited
Post-condition	User has successfully edited the areas
Main Success Scenario	<ol style="list-style-type: none"> 1. User clicks the edit areas option from the menu 2. System allows the user to edit the name, zip code, or whether the area requires fuel or not, etc. 3. User enters the information about the area that is to be updated and hits save button 4. System checks and saves the updated information 5. System shows the confirmation message that the area has been updated successfully.
Alternate Scenario or Extensions	<ul style="list-style-type: none"> *a. At any time, system may go down. <ol style="list-style-type: none"> 1. User/Admin restarts the application. 3a. User enters the wrong type of data in the field <ol style="list-style-type: none"> 1. System alerts the user that the entered data is incorrect 2. User enters again 3b. User clicks the cancel button <ol style="list-style-type: none"> 1. System goes back to the previous state without changing the area records
Special Requirements	<ul style="list-style-type: none"> • After pressing the button, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Rarely
Technology and Data Variation List	There should be no other way to edit the areas
Priority	Medium

2.4.3.26 UC-26 Logout

Table 28. UC-26 Logout

ID	UC-29
Name	Logout
Primary Actors	User (Manager, Supervisor, Driver)
Stakeholders	Developer of the app, User (Manager, Supervisor, Driver)
Description	User wants to logout of the system
Pre-condition	User is logged in the system
Post-condition	User has successfully logged out of the system
Main Success Scenario	<ol style="list-style-type: none"> 6. User clicks his username or user icon 7. System opens a popup menu with an option of logout 8. User clicks the “logout” option 9. The system ends the user’s session and logs out the user.
Alternate Scenario or Extensions	<p>*a. At any time, system may go down.</p> <ol style="list-style-type: none"> 2. User restarts the application
Special Requirements	<ul style="list-style-type: none"> • After pressing the Logout button, system responds within 30 seconds 90% of the time. • Language internationalization on the text displayed.
Frequency	Depends upon the need of user
Technology and Data Variation List	There should be no other way to log out from the system
Priority	High

2.5 System Sequence Diagrams (SSDs)

2.5.1 Login SSD

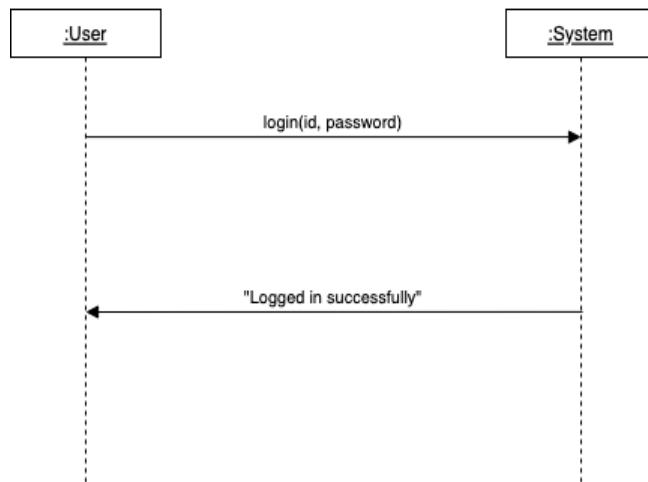


Figure 8. Login SSD

2.5.2 Edit Staff Profile SSD

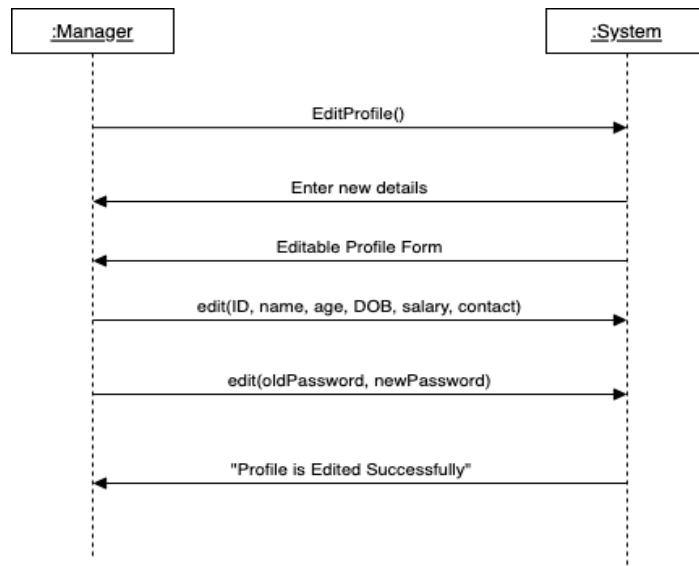


Figure 9. Edit Staff Profile SSD

2.5.3 Edit Vehicle Assignment SSD

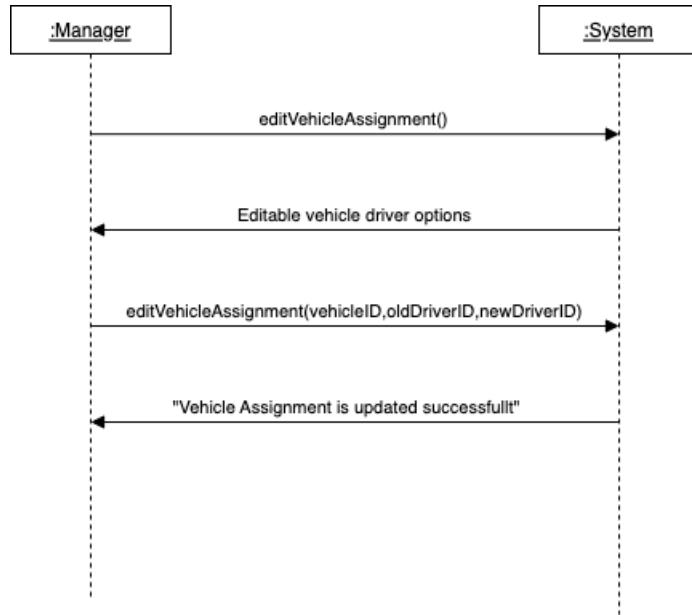


Figure 10. Edit Vehicle Assignment SSD

2.5.4 Edit Area SSD

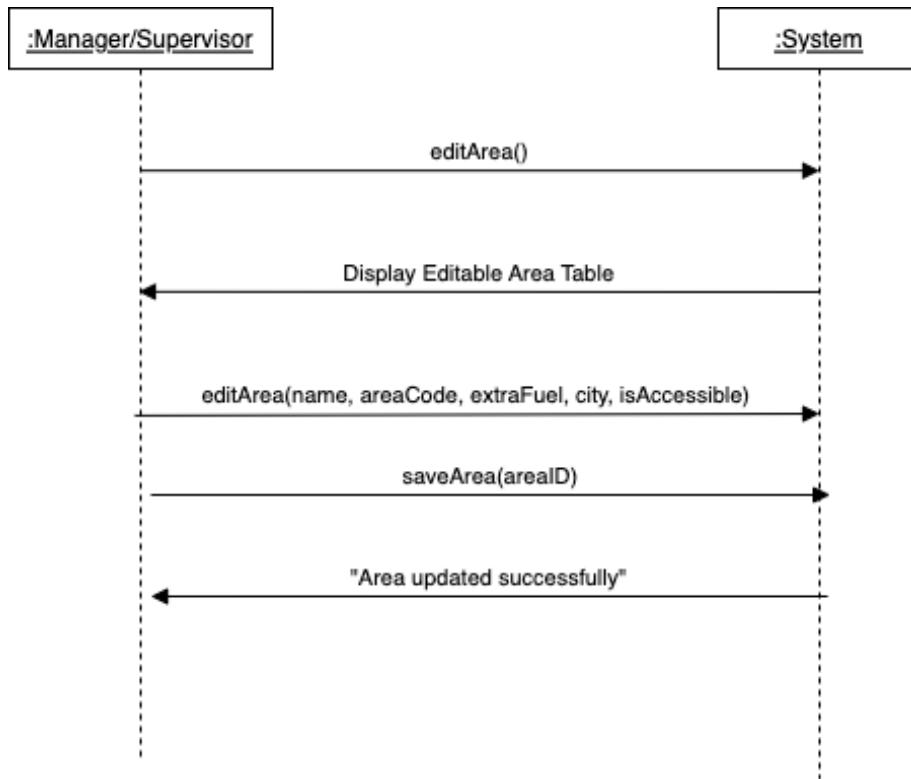


Figure 11. Edit Area SSD

2.5.4 Edit Delivery Sheet SSD

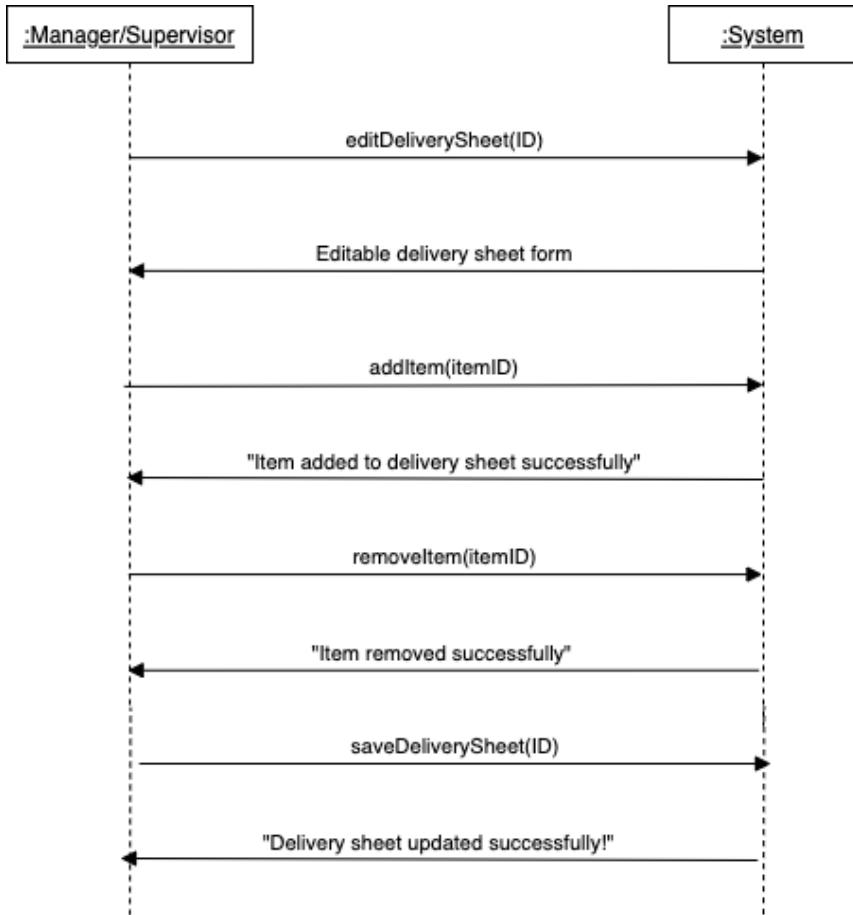


Figure 12. Edit Delivery Sheet SSD

2.6 Domain Model

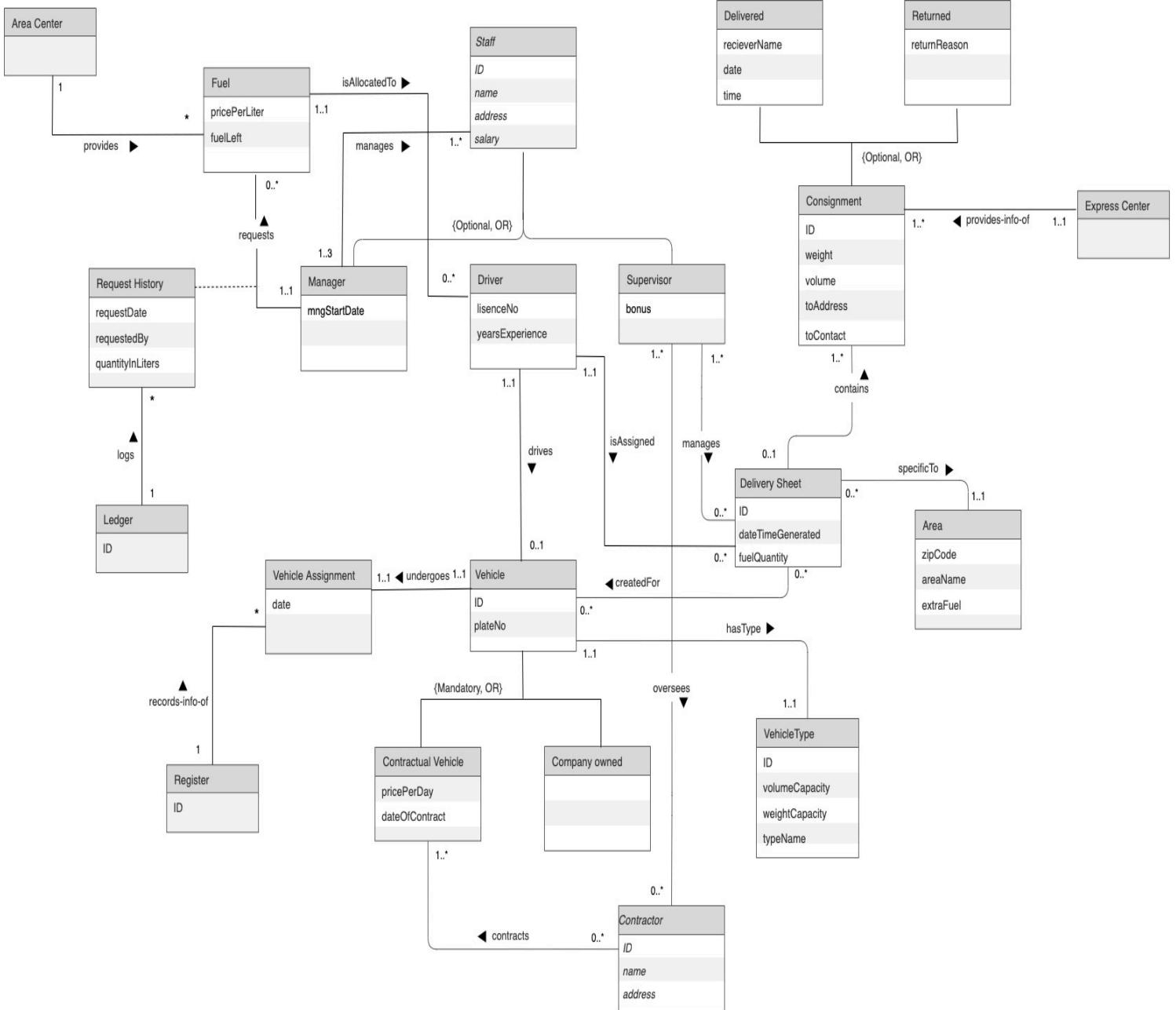


Figure 13. Domain Model

2.7 Data Flow Diagrams

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.

2.7.1 Delivery Sheet Generation Process DFD

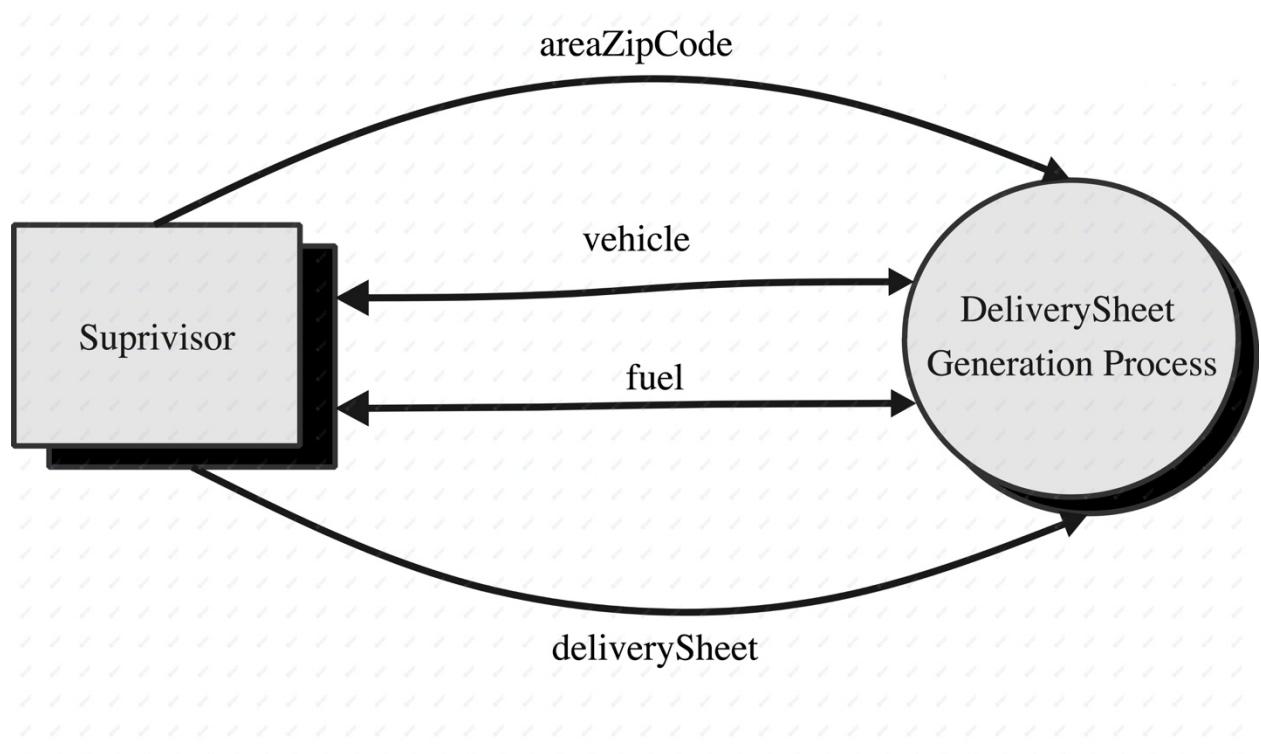


Figure 14. Delivery Sheet Generation Process DFD – Level 0

2.7.1a. Delivery Sheet Generation Process DFD

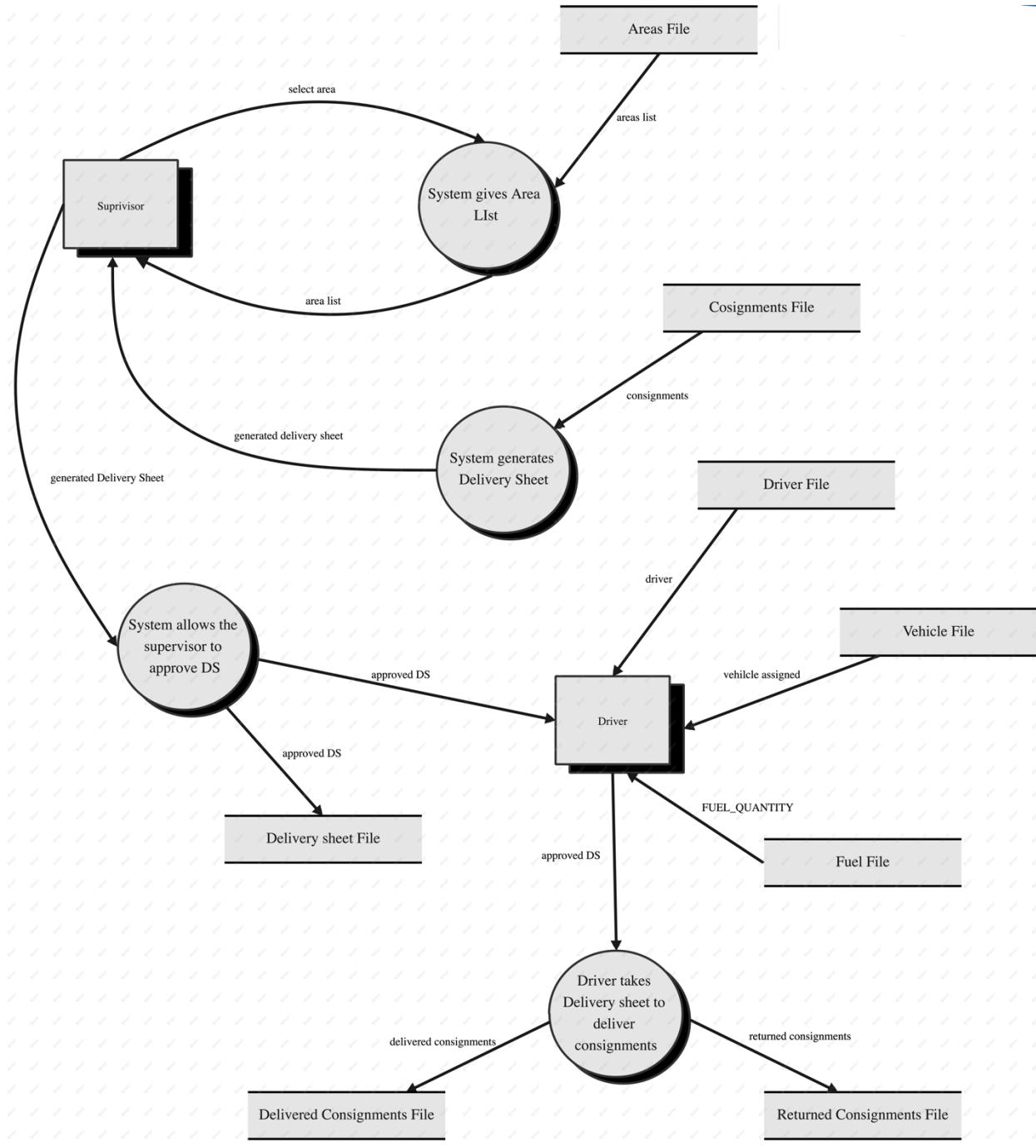


Figure 15. Delivery Sheet Generation Process DFD – Exploded

2.7.2 Fuel Allocation Process DFD

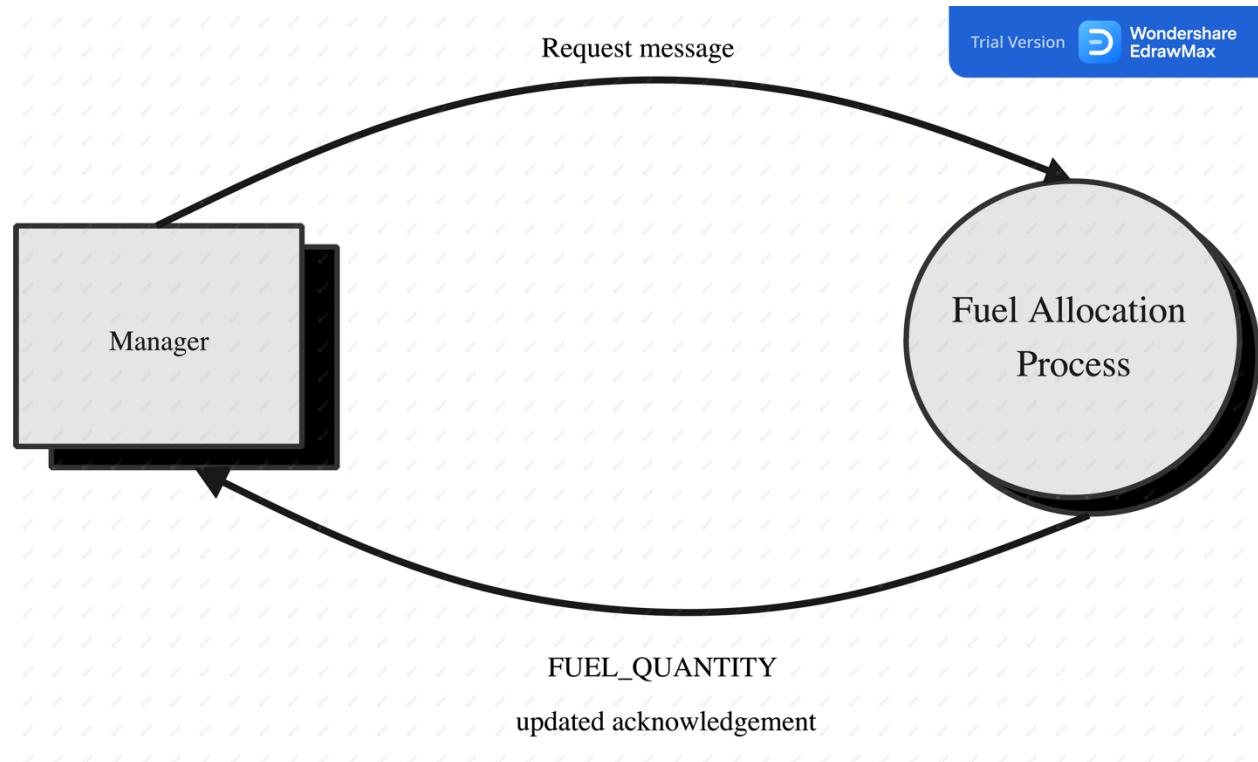


Figure 16. Fuel Allocation Process DFD Level 0

2.7.2a. Fuel Allocation Process DFD

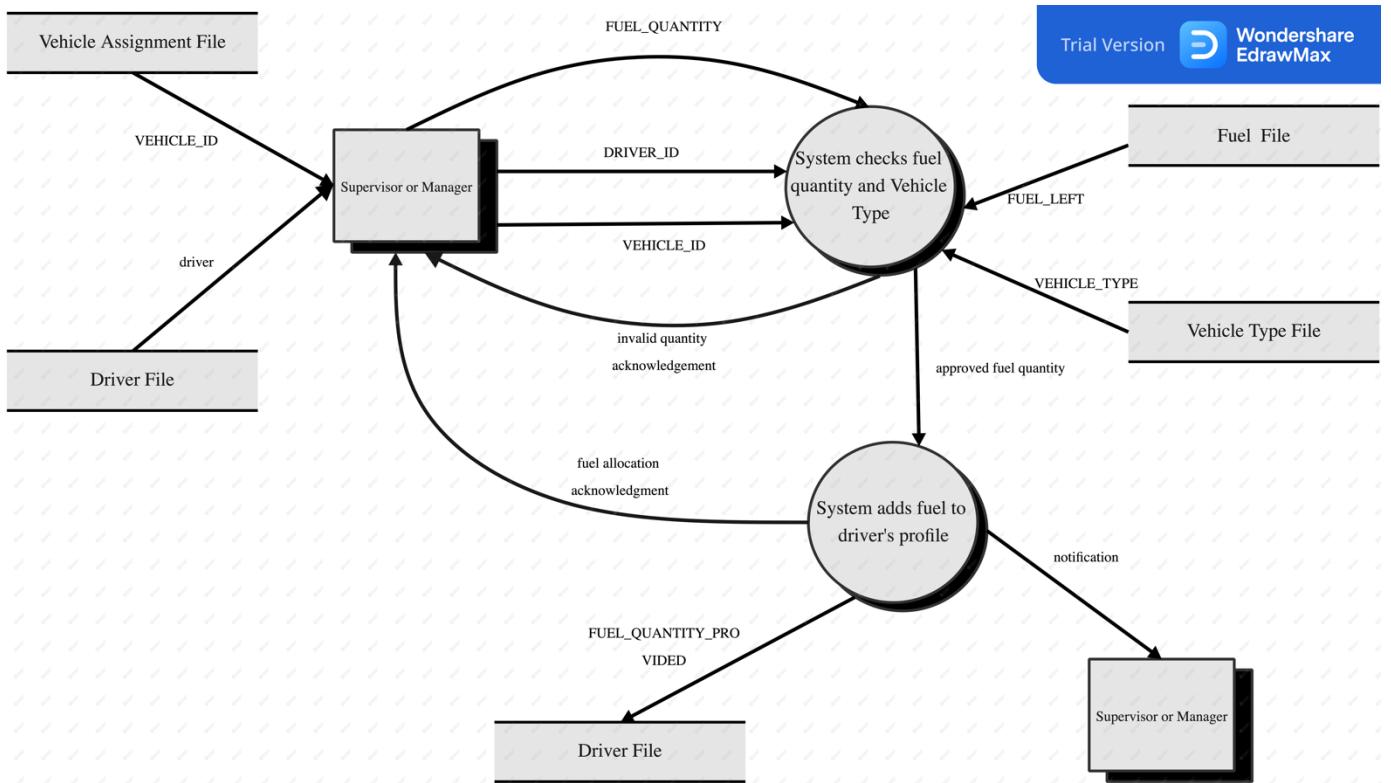


Figure 17: Fuel Allocation Process DFD - Expanded

2.7.3 Delivery Sheet Generation for Bike and Vehicle – DFD

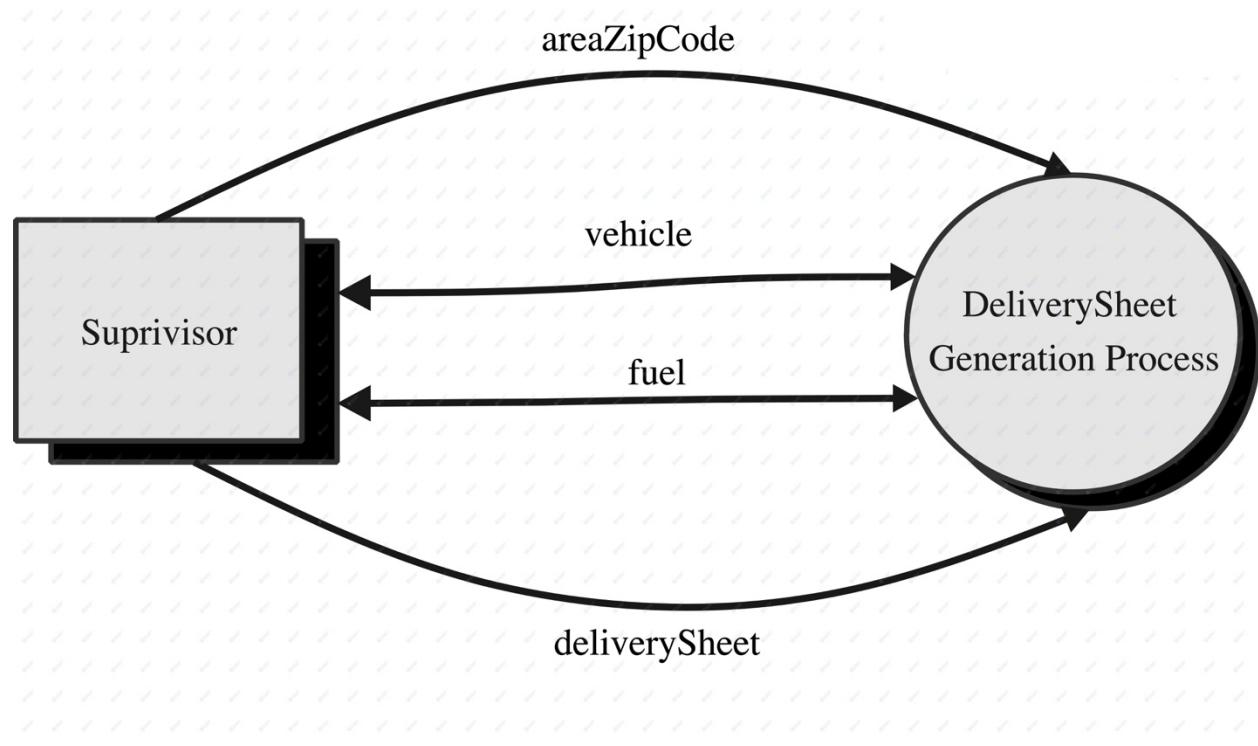


Figure 18. Delivery Sheet Generation for Bike and Vehicle – DFD Level 0

2.7.3a. Delivery Sheet Generation for Bike and Vehicle – DFD

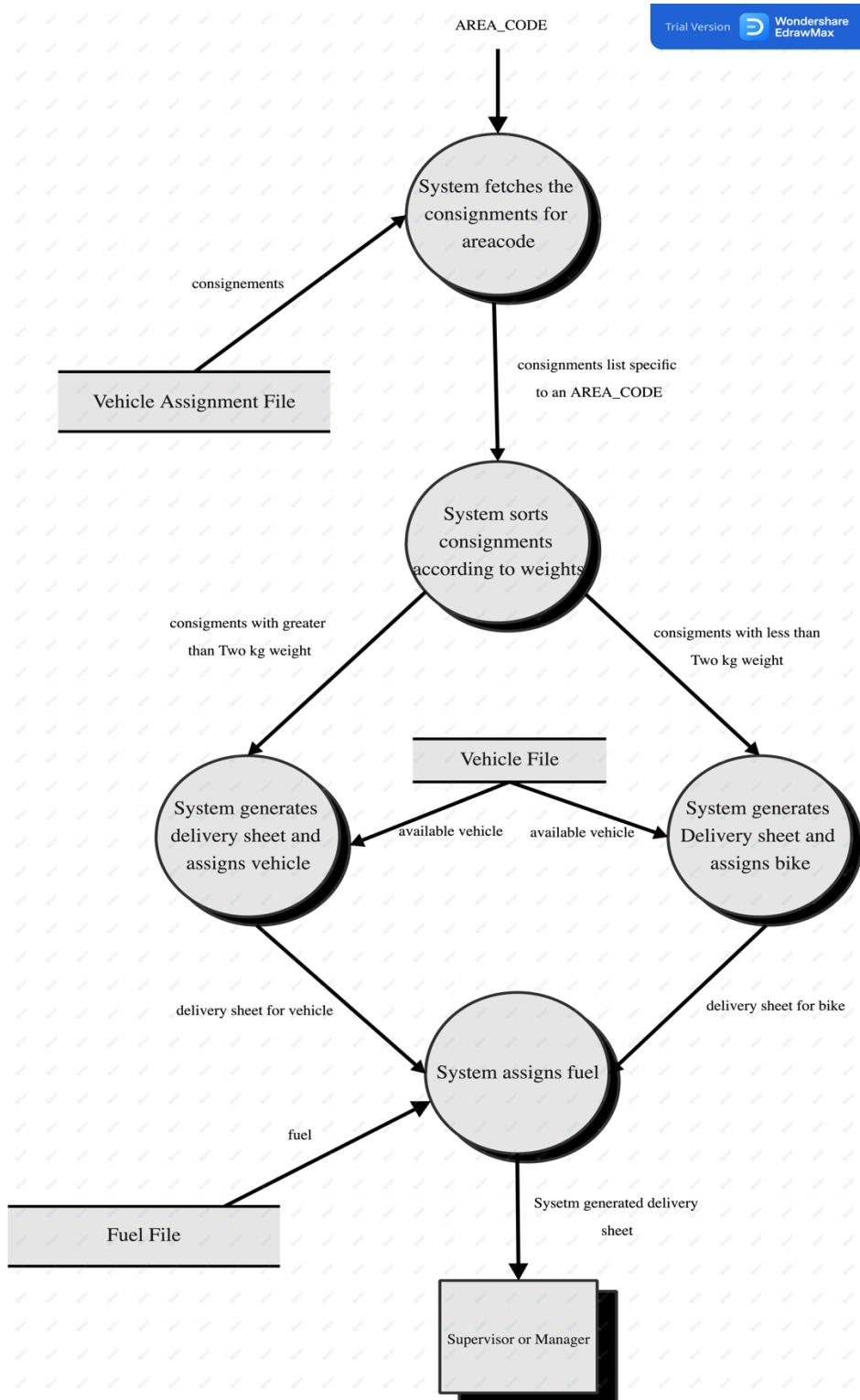


Figure 19. Delivery Sheet Generation for Bike and Vehicle – DFD Exploded

2.7.4. Vehicle Assignment Process – DFD

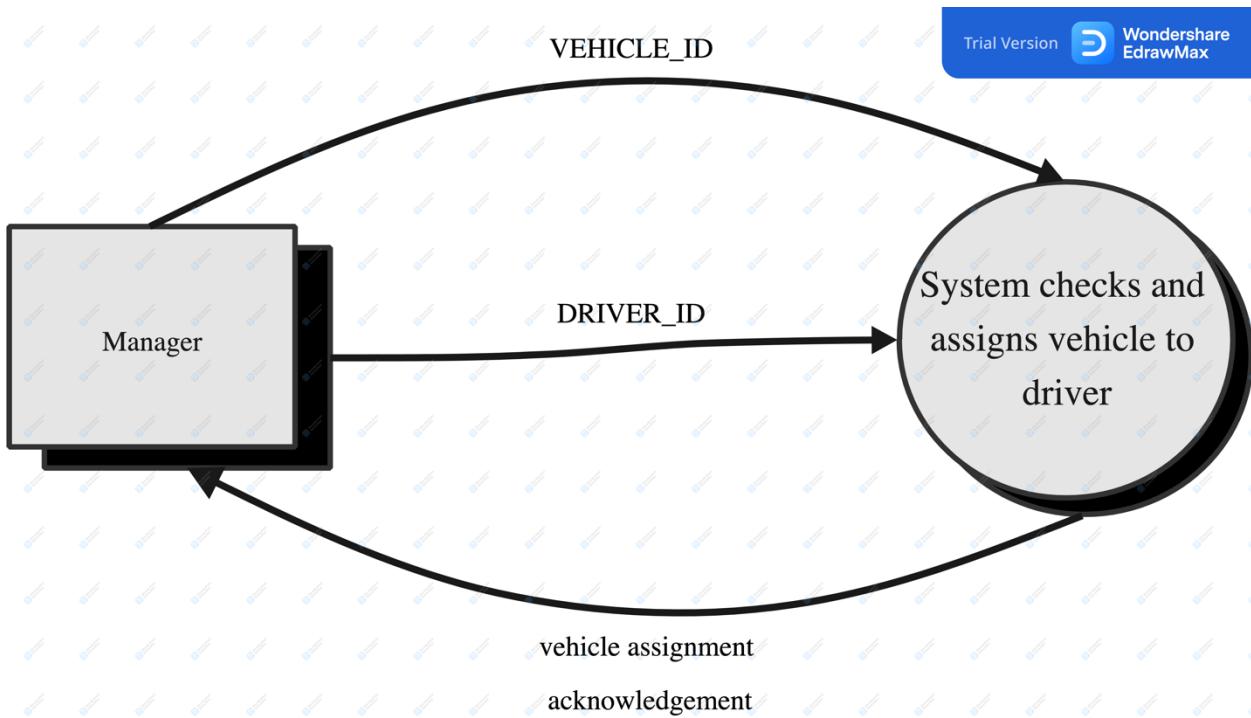


Figure 20. Vehicle Assignment Process – DFD Level 0

2.7.4a. Vehicle Assignment Process – DFD

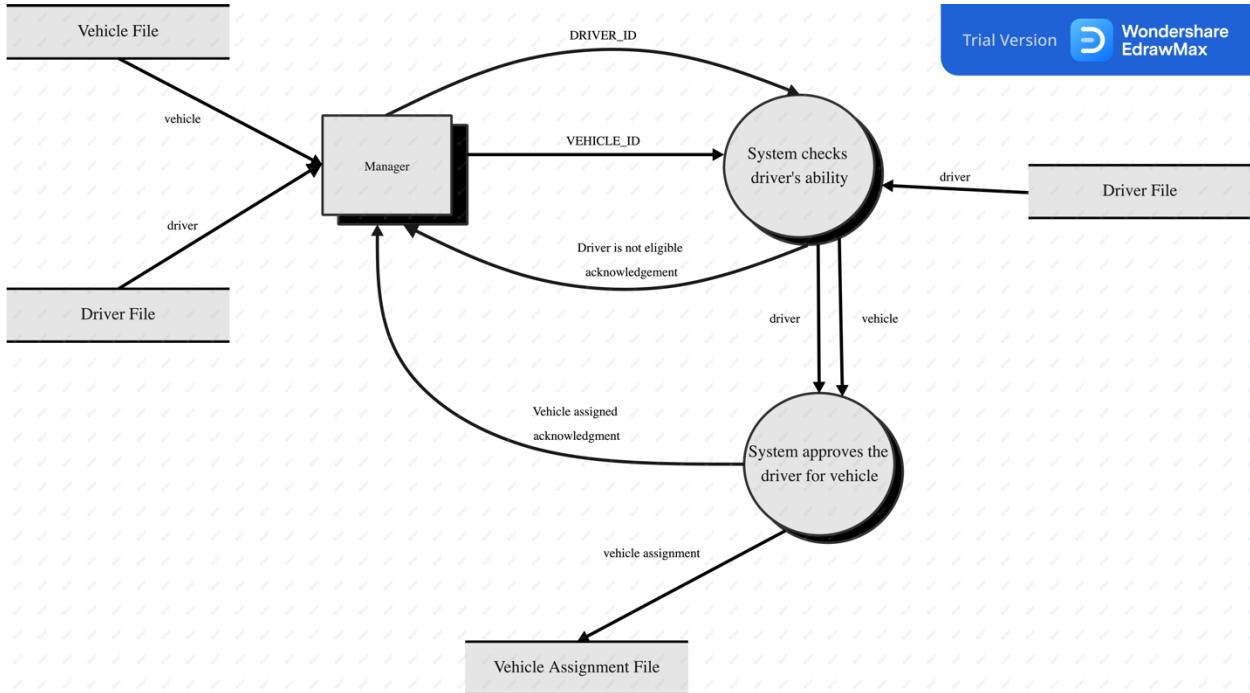


Figure 21. Vehicle Assignment Process – DFD Exploded

Chapter 3

Software Design Description

(SDD)

3. Software Design Description

3.1 Introduction

Software Design Description (SDD) is the representation of a software design which is used for communicating design information of a system to its stakeholders. It shows how the software system will be structured to satisfy the requirements. The SDD document contains the interface design, architecture diagram, sequence diagrams and class diagram. Within the Software Design Document narrative and graphical documentation of the software design for the project include use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information. It includes the description of how the software will meet the requirements.

3.1.1 Purpose

The purpose of the Software Design Document is to provide a description of the design of Last Mile Delivery Management System (LMDMS) to allow for software developer to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

3.1.2 Design Overview

The software design document provides design details of the Last Mile Delivery Management System (LMDMS). The document contains a complete low-level description of the system, providing insight of the structure and design of each component. A system architecture is a conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

3.2 Software Architecture Design

Architectural design defines the relationship between major structural elements of the software. It defines the design patterns that can be used to satisfy the requirements that have been defined for the system. Architecture design entails the way these components interact and the structure of data that are used by the components. Components or modules are generalized to represent major system elements and their interactions.

3.2.1 Chosen System Architecture

In this section, we describe chosen architecture of the system. The basic architecture of this system is 3-tier model. A 3-tier model uses the client/server computing model. It has three layers.

- Presentation Layer
- Business Layer
- Data Layer

3.2.1.1 LMDMS Architecture Diagram

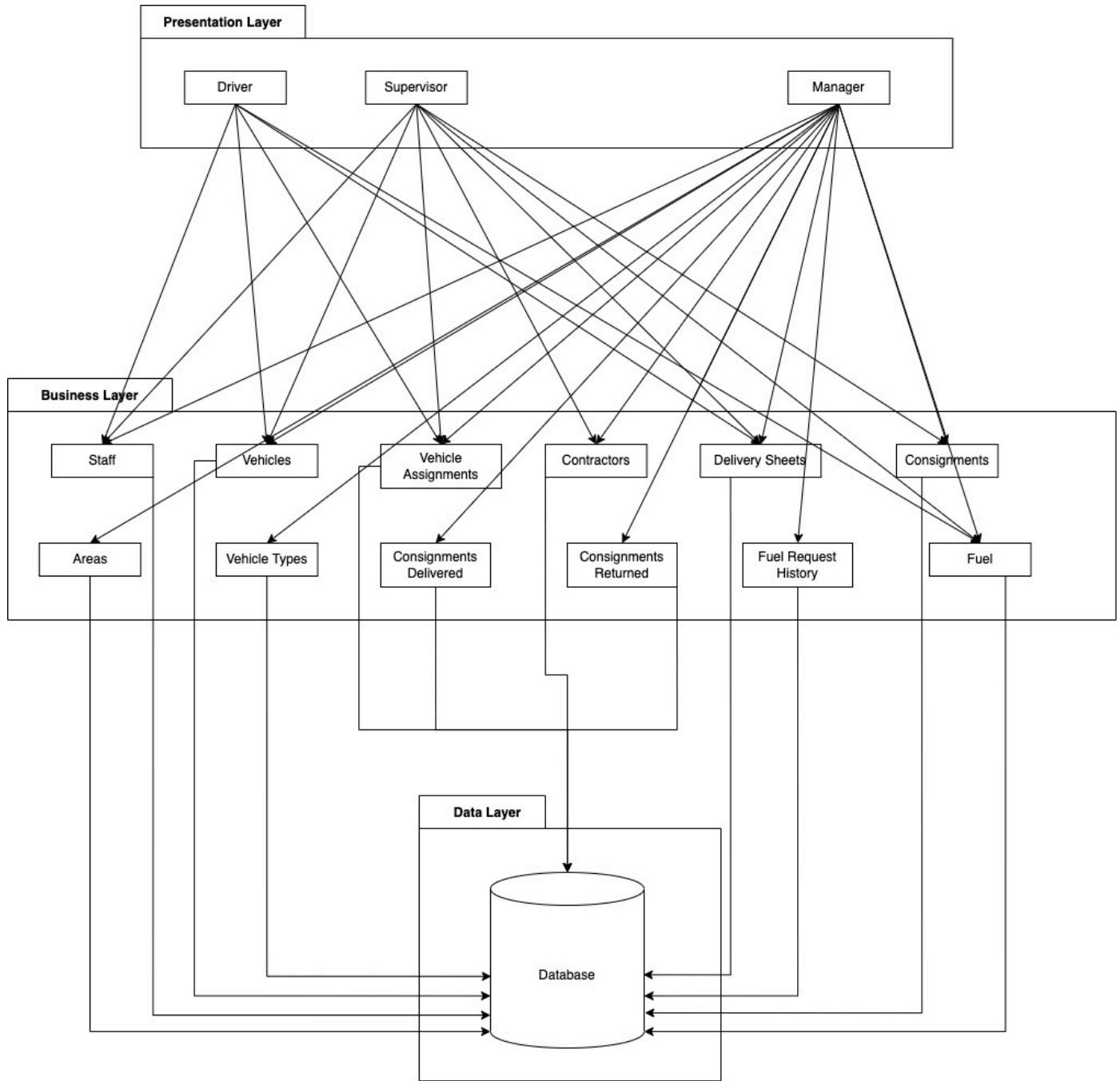


Figure 22. LMDMS Architecture Diagram

3.2.2 Discussion of Alternative Designs

We are going to use the 3-Tier architecture instead of 2-Tier architecture which isolates the Application logic from the Data access tier. But for my application devices data and security is more important than time. Therefore, in our scenario 3-tier architecture will be better.

3.2.3 System Interface Description

System interface describes the flow of resources. It is the logical characteristics of each interface between the software product and the hardware components of the system.

3.3 User Interface Design

User interface is the logical characteristics of each interface between the software product and its users. In this section user interface of LMDMS is discussed.

3.3.1 Description of the User Interface

In LMDMS user can interact with the system by using mouse and keyboard. When user opens the website login screen will appear with all the login requirements, and when the user entered his login details, he will be entered in to the system. If the user entered manager's login details, he would get the manager's home page, and if supervisor's account, then supervisor's homepage, and if driver's account then the home page for driver will open with limited operations. After that according to the user requirement he/she can choose the options available.

3.3.2. Screen Images

Following are the screen images of LMDMS

3.3.2.1 Log In Screen

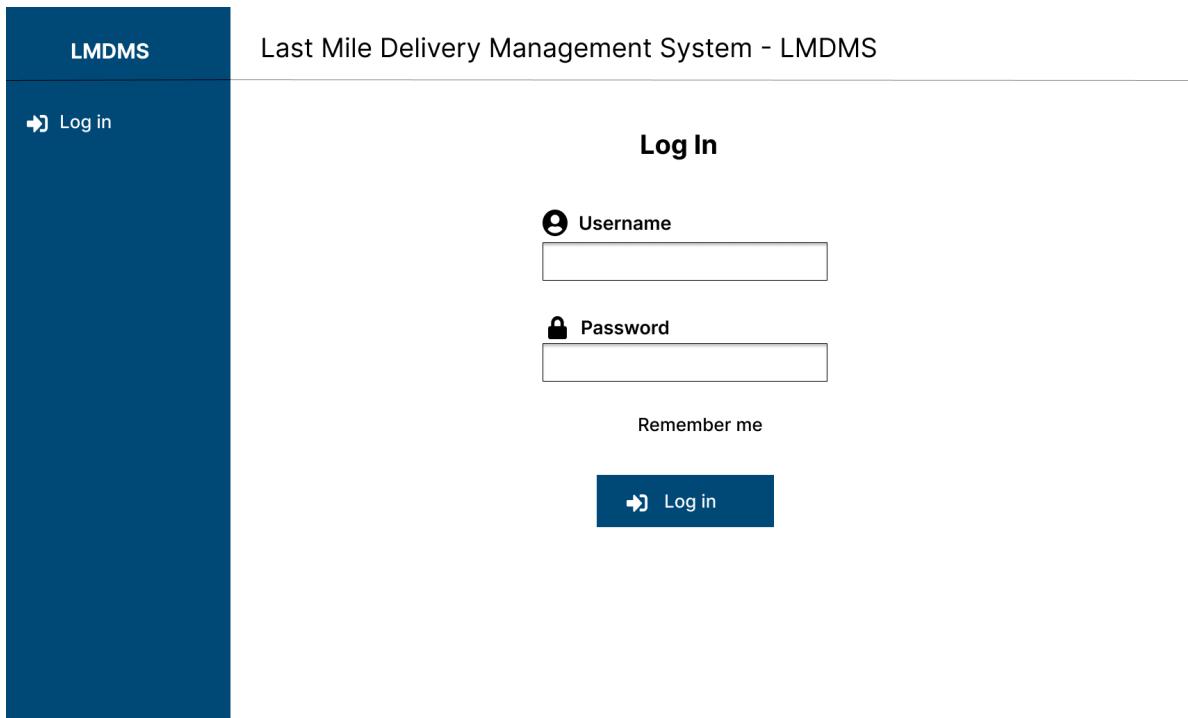


Figure 23: Log In screen

3.3.2.2 Home Screen

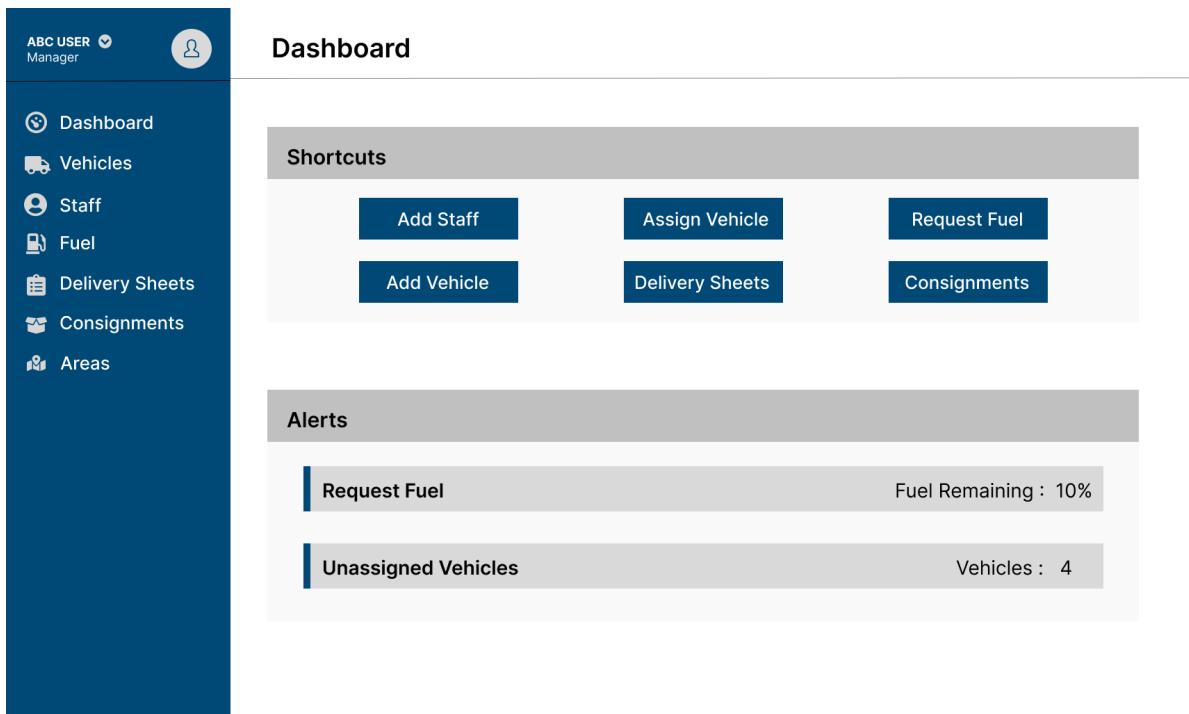


Figure 24: Home screen

3.3.2.3 View Vehicles Screen

The screenshot shows the 'Vehicles' screen with a sidebar menu on the left. The sidebar includes links for Dashboard, Vehicles (selected), Staff, Fuel, Delivery Sheets, Consignments, and Areas. The main area is titled 'Vehicles' and contains a table of assigned vehicles. The table has columns for Name, Plate No., Model, Make, Status, and Driver. Three rows of data are shown, each with edit and delete icons.

Name	Plate No.	Model	Make	Status	Driver
Toyota Hilux Vigo	LEC-214	2016	Toyota	Active	Sajid Ali
Toyota Hilux Vigo	LEC-214	2016	Toyota	Active	Sajid Ali
Toyota Hilux Vigo	LEC-214	2016	Toyota	Active	Sajid Ali

Figure 25: View Vehicles Screen

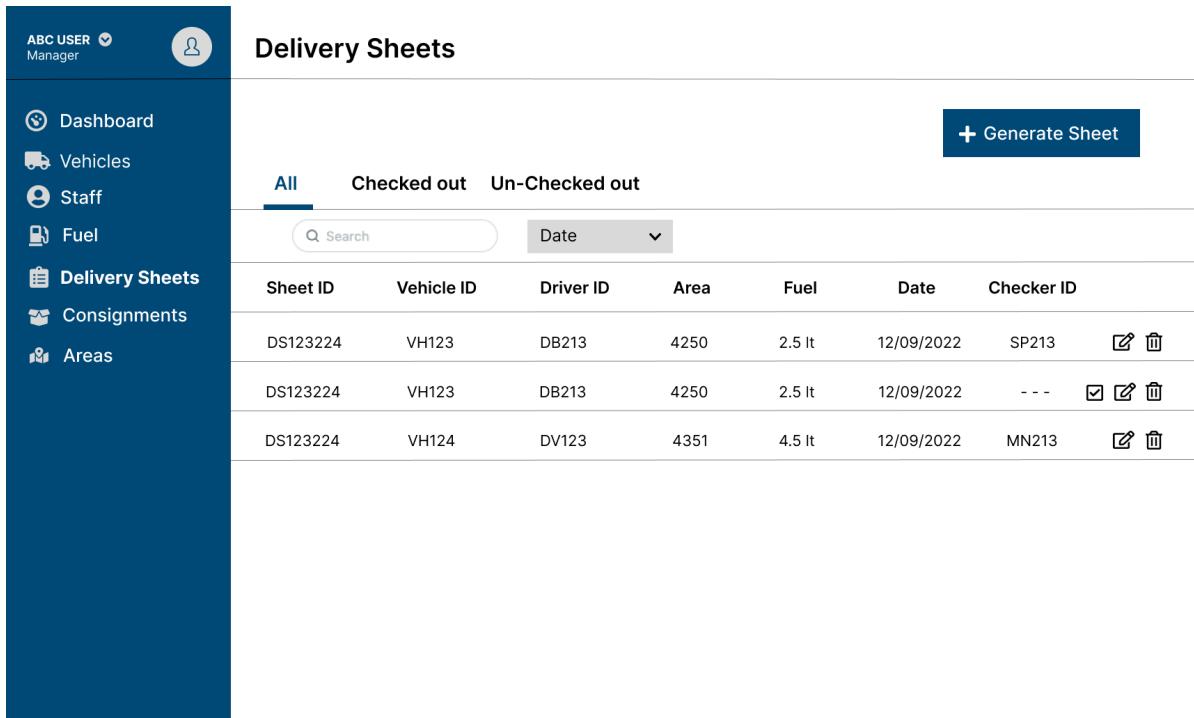
3.3.2.4. Vehicle Assignments Screen

The screenshot shows the 'Vehicle Assignments' screen with a sidebar menu on the left. The sidebar includes links for Dashboard, Vehicles (selected), Staff, Fuel, Delivery Sheets, Consignments, and Areas. The main area is titled 'Vehicle Assignments' and contains a table of assigned drivers. The table has columns for Vehicle, Driver, Assign Date, and Assigned By. Three rows of data are shown, each with edit and delete icons.

Vehicle	Driver	Assign Date	Assigned By
Toyota Hilux Vigo	Sajid Ali	10/3/22	Umar Khan
Toyota Hilux Vigo	- - - - -	- - - - -	- - - - -
Suzuki Carry	Shahid Khan	10/3/22	Kashif Shah

Figure 26: Vehicle Assignments Screen

3.3.2.5 Delivery Sheets List Screen



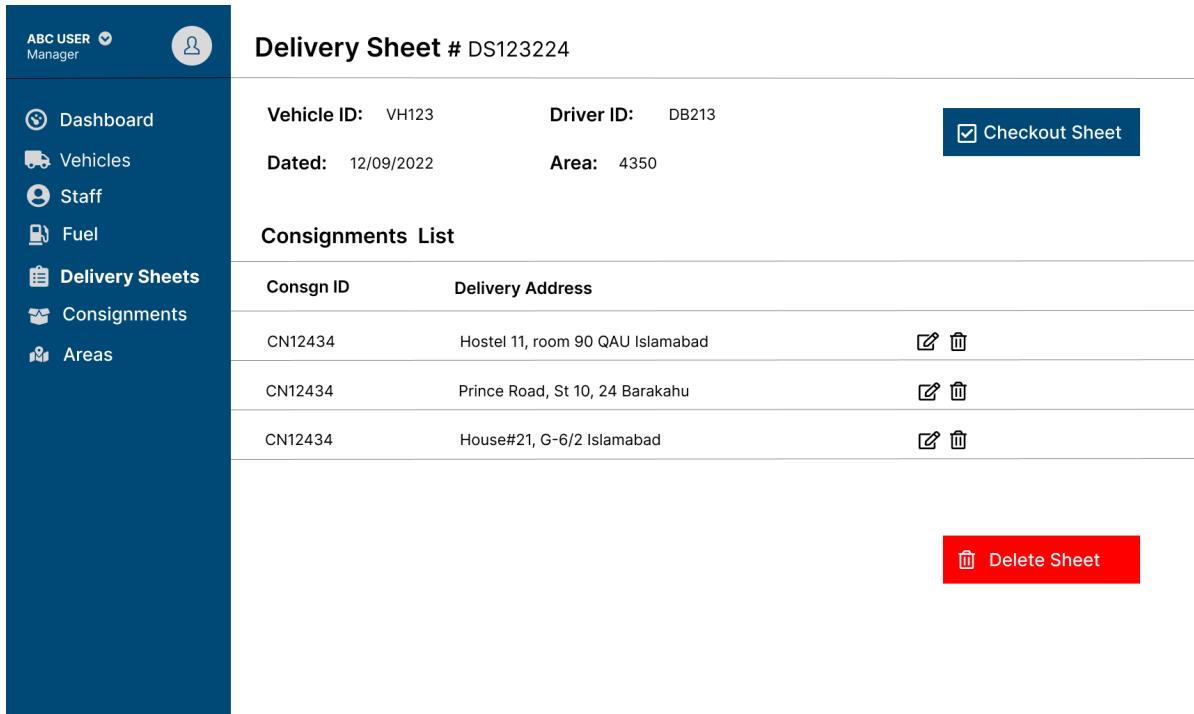
The screenshot shows the 'Delivery Sheets' list screen. At the top right, there is a user profile icon labeled 'ABC USER Manager'. Below it, a header bar has three tabs: 'All' (selected), 'Checked out', and 'Un-Checked out'. To the right of the tabs is a blue button with a plus sign and the text 'Generate Sheet'. A search bar with a magnifying glass icon and a dropdown menu for 'Date' are also present. The main area contains a table with the following columns: Sheet ID, Vehicle ID, Driver ID, Area, Fuel, Date, and Checker ID. The table lists three rows of data:

Sheet ID	Vehicle ID	Driver ID	Area	Fuel	Date	Checker ID
DS123224	VH123	DB213	4250	2.5 lt	12/09/2022	SP213
DS123224	VH123	DB213	4250	2.5 lt	12/09/2022	- - -
DS123224	VH124	DV123	4351	4.5 lt	12/09/2022	MN213

Each row has edit and delete icons at the end.

Figure 27: Delivery Sheets List Screen

3.3.2.6 View Delivery Sheet Screen



The screenshot shows the 'View Delivery Sheet' screen for delivery sheet # DS123224. At the top right, there is a user profile icon labeled 'ABC USER Manager'. Below it, a header bar shows the delivery sheet number 'Delivery Sheet # DS123224'. The main area is divided into two sections: 'Vehicle Details' and 'Consignments List'.

Vehicle Details:

Vehicle ID:	VH123	Driver ID:	DB213
Dated:	12/09/2022	Area:	4350

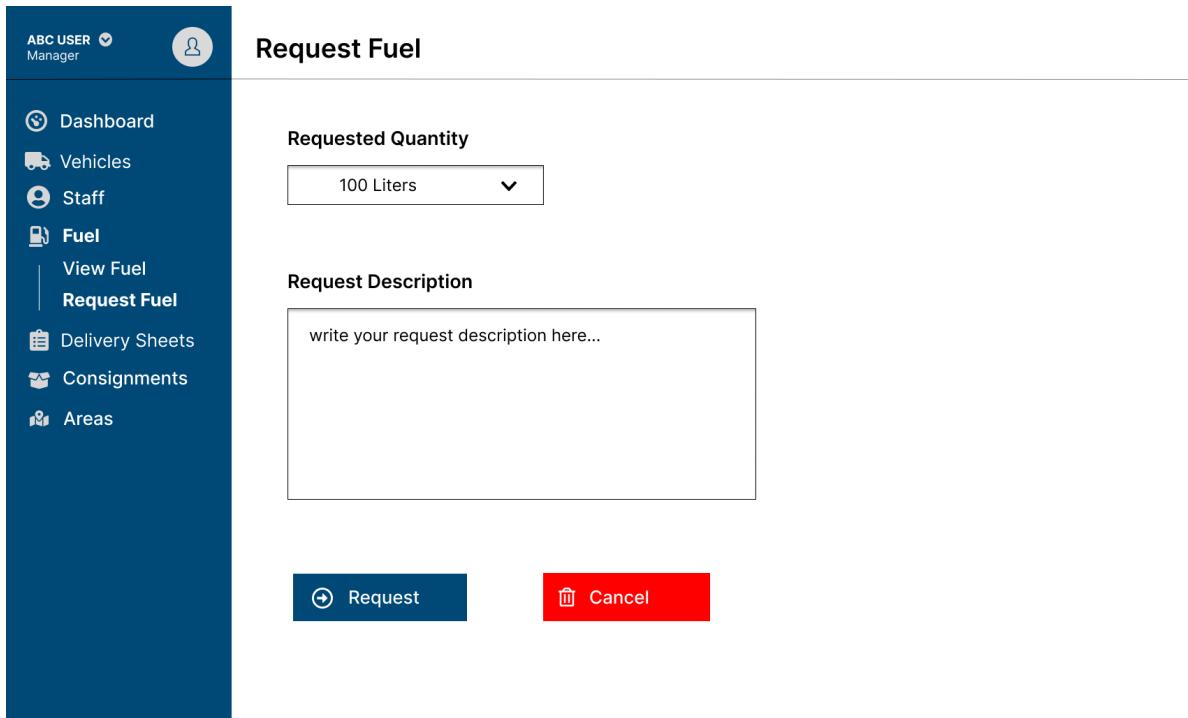
Consignments List:

Consgn ID	Delivery Address	Actions
CN12434	Hostel 11, room 90 QAU Islamabad	
CN12434	Prince Road, St 10, 24 Barakahu	
CN12434	House#21, G-6/2 Islamabad	

A red button at the bottom right says 'Delete Sheet' with a trash icon.

Figure 28: View Delivery Sheet Screen

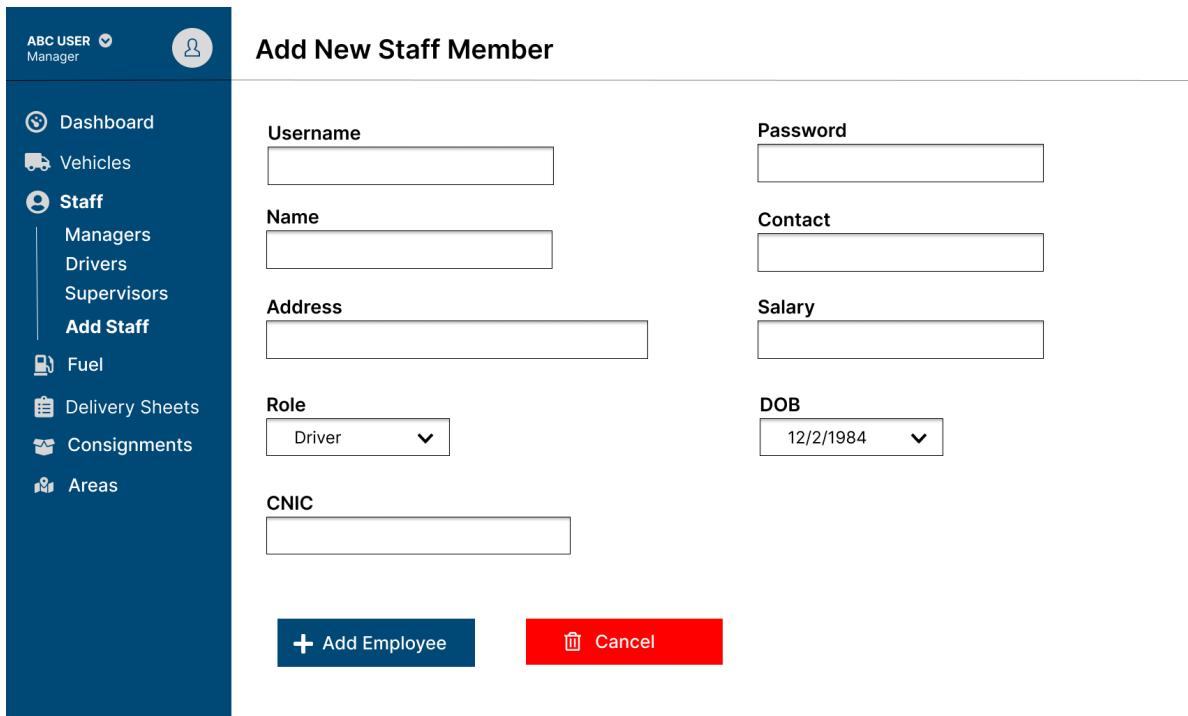
3.3.2.7 Request Fuel Screen



The screenshot shows the 'Request Fuel' screen. At the top left is a user profile for 'ABC USER Manager'. The main title is 'Request Fuel'. On the left sidebar, under the 'Fuel' section, 'Request Fuel' is selected. The main area has two input fields: 'Requested Quantity' (set to '100 Liters') and 'Request Description' (with placeholder text 'write your request description here...'). Below these are two buttons: a blue 'Request' button and a red 'Cancel' button.

Figure 29: Request Fuel Screen

3.3.2.8 Add Staff Member Screen



The screenshot shows the 'Add New Staff Member' screen. At the top left is a user profile for 'ABC USER Manager'. The main title is 'Add New Staff Member'. On the left sidebar, under the 'Staff' section, 'Add Staff' is selected. The main area contains several input fields: 'Username' (empty), 'Password' (empty), 'Name' (empty), 'Contact' (empty), 'Address' (empty), 'Salary' (empty), 'Role' (set to 'Driver'), 'DOB' (set to '12/2/1984'), and 'CNIC' (empty). Below these are two buttons: a blue '+ Add Employee' button and a red 'Cancel' button.

Figure 30: Add Staff Member Screen

3.3.3 Sequence Diagrams

3.3.3a Objects and Action

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are developed against the use cases.

3.3.3.1 Login Sequence Diagram

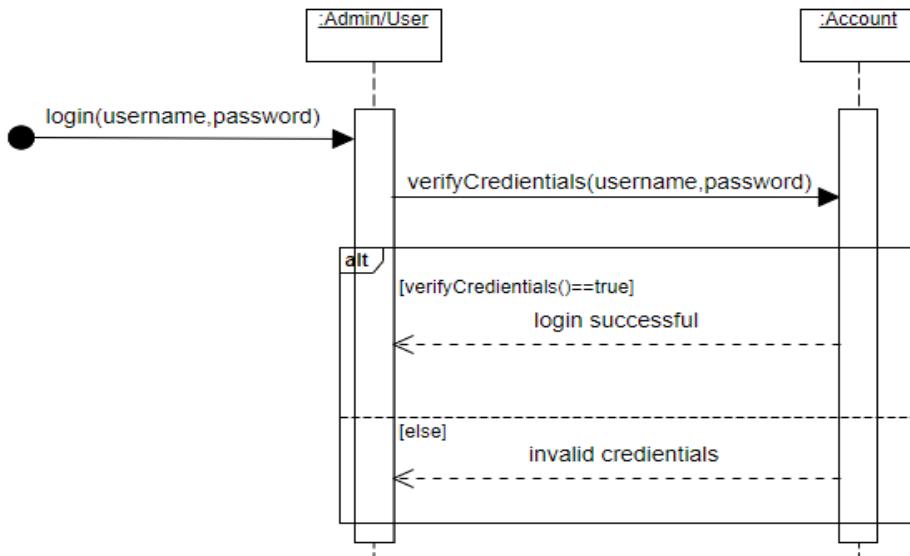


Figure 31. Login Sequence Diagram

3.3.3.2 Generate Delivery Sheet Sequence Diagram

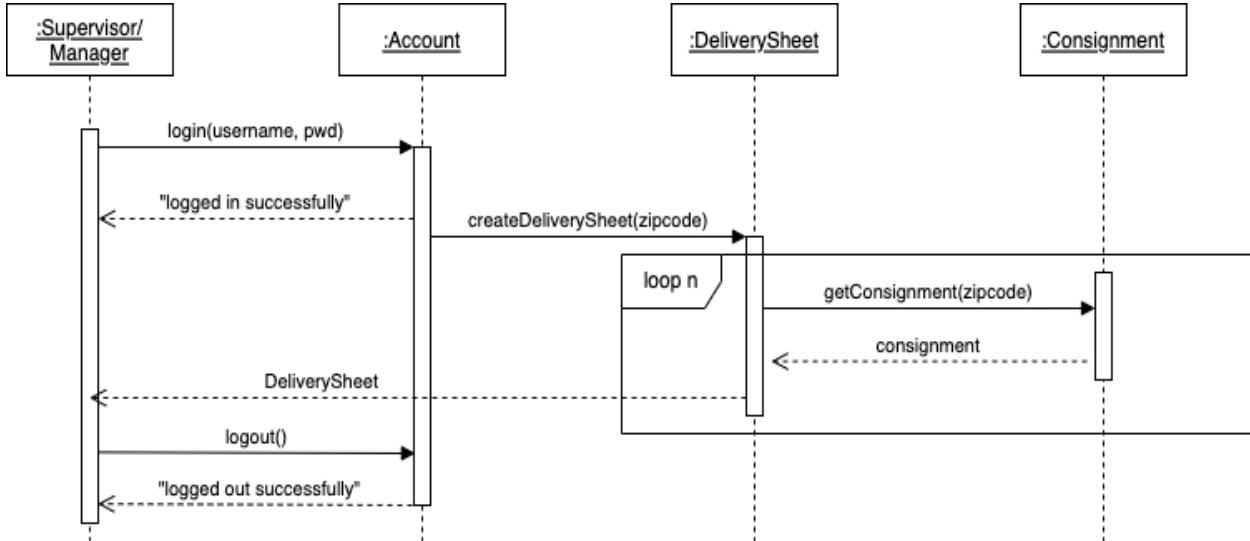


Figure 31. Generate Delivery Sheet Diagram

3.3.3.3 Request Fuel SD

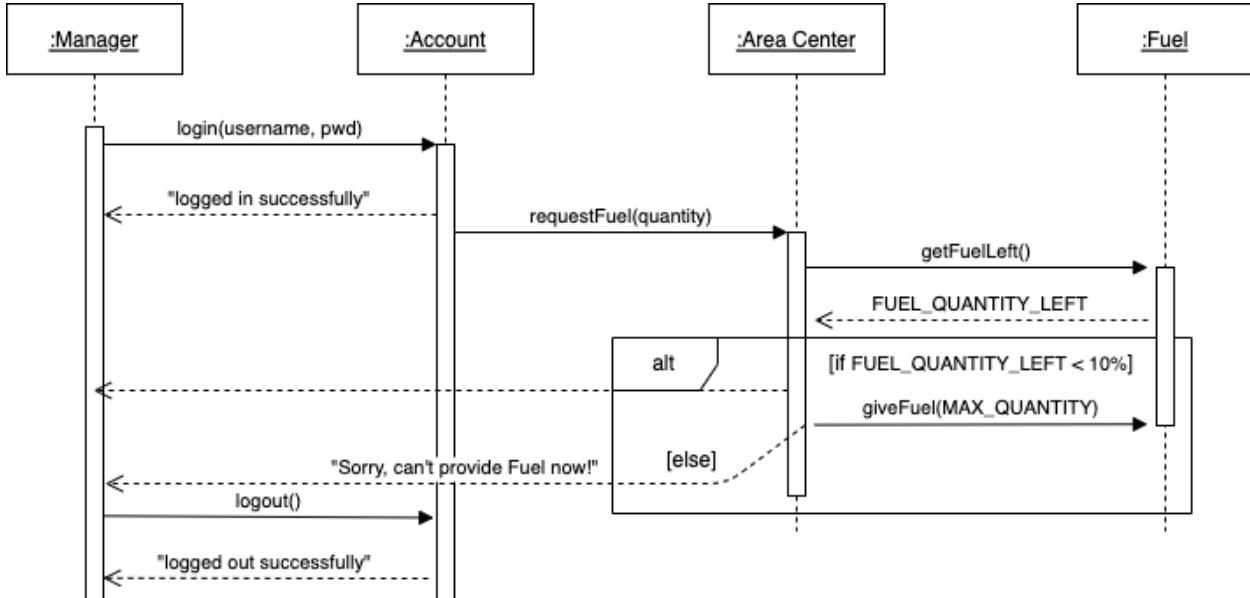


Figure 33. Request Fuel SD

3.3.3.4 Assign Vehicle

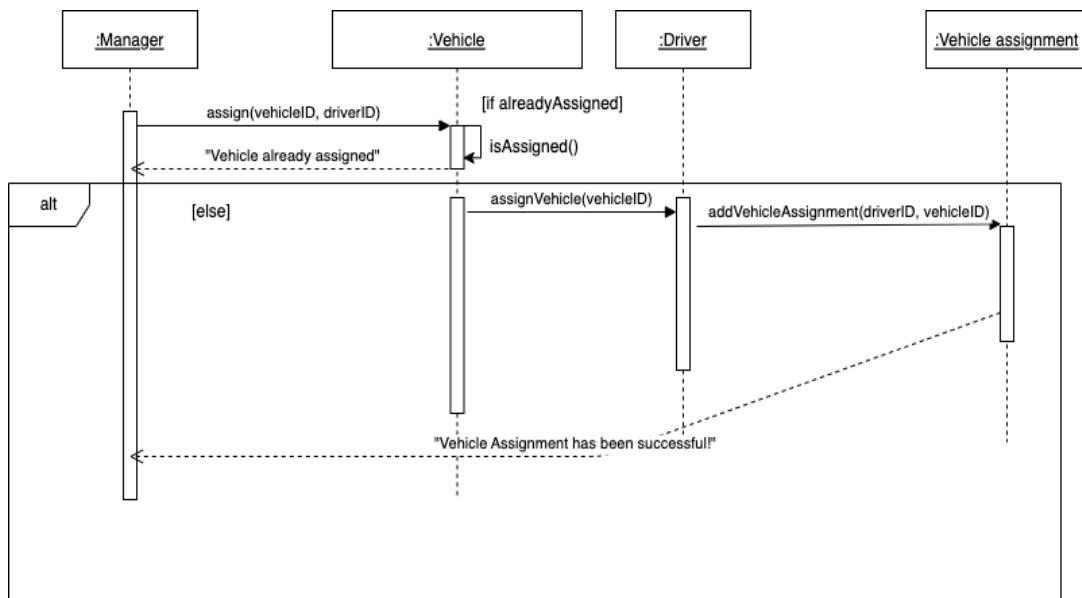


Figure 34. Assign Vehicle

3.4 Class Diagram

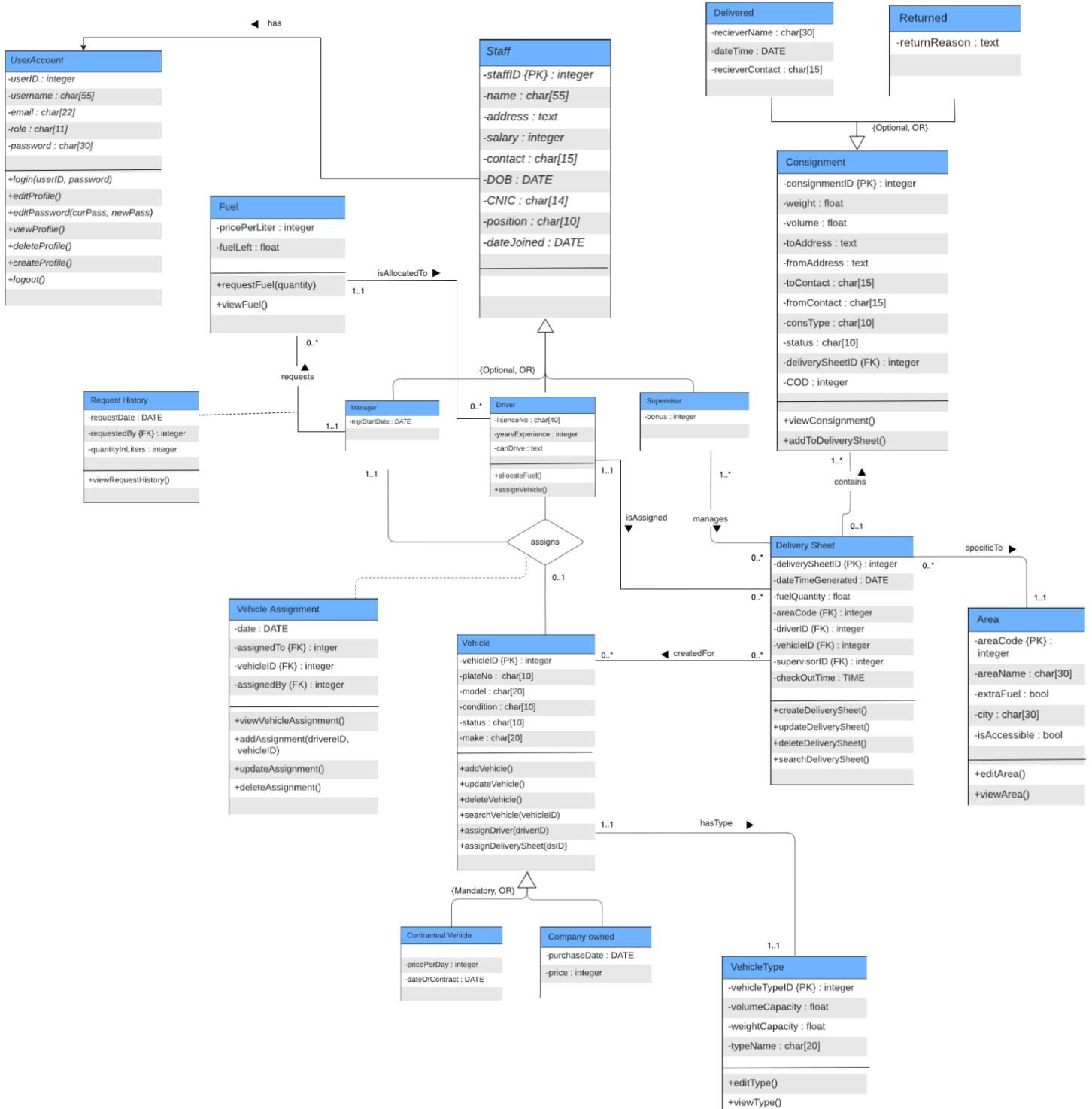


Figure 35. Class Diagram

Chapter 4

Software Test Documentation

(STD)

4. Software Test Documentation

4.1 Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Software testing is a very important process that should be done during the development process because it is very useful to assess the quality of the product.

4.1.1 System Overview

LMDMS will help the supervisors to automatically generate the delivery sheets from consignments for specific areas. This automatic process will be according to the consignments' weights and volumes. At first consignments will be separated on the basis of their area zones (postal codes/zip code). Secondly, Consignments with less than or equal to 2kg weight will be separated from consignments with greater than 2kg weight and are added to the delivery sheet to be assigned to a bike and a rider. And >2kg will be added to the delivery sheets for the vehicles. And this will be done while considering the vehicle total weight and volume carrying capacity. This process will be near optimal in assigning the vehicle a delivery sheet so that the vehicle would carry the maximum number of deliveries per trip. Moreover, the fuel is allocated to the drivers while considering the area they have to go for the deliveries. This area will come from the delivery sheet that has just been created. For the areas which requires extra fuel, such areas will get extra fuel automatically. The manager can edit the areas which requires extra fuel from his dashboard.

4.1.2 Test Approach

I will be using the User acceptance testing (UAT) to test my LMDMS which is also called beta testing or end user testing, consist of a process of verifying that a solution works for the user. It is not system testing (ensuring software does no crash and meets documented requirements), but rather is there to ensure that the solution will work for the user i.e. test the user acceptance the solution. In software development, UAT as one of the final stages of a project often occurs before a client or customer accepts the new system. Users of the system perform test in line with what occur in real life scenarios.

4.2 Test Plan

4.2.1 Features to be tested

Features to be tested are all according to user prospective. For example

4.2.2 Features not to be tested

Features not to be tested are from the developer's point of view. For example

- How much power is consumed by the processor?
- How much memory is consumed by the LMDMS application?
- Software risk factor
- Maintainability of the system

4.2.3 Testing tools and environment

As this is beta testing (testing by the user) so no specific tools and environment is required. All a user need is a web browser with working internet connection.

4.3 Test Cases

Following are the test cases of LMDMS.

4.3.1 TC-1: Login

Table 29. TC-1 Login

ID	TC-1
Description	Check the user can login to the system using valid credentials.
Setup	<ol style="list-style-type: none">1. The web application is running.2. The log in page is opened.
Inputs	<ol style="list-style-type: none">1. username: abc123, password: qweqwe2. username: zad123, password: 121213
Expected Output	<ol style="list-style-type: none">1. Login successful2. Alert: Incorrect username or password
Observed Output	<ol style="list-style-type: none">1. Login successful2. Alert: Incorrect username or password
Verdict	Pass

4.3.2 TC-2: Add new Staff Profile

Table 30. TC-2 Add new staff profile

ID	TC-2
Description	Check the manager can add a new staff profile to the system using valid username.
Setup	<ol style="list-style-type: none">1. The web application is running.2. The Add new staff page is opened.
Inputs	<p>A: Edit profile</p> <ol style="list-style-type: none">1. Name: Muhammad Ali, ID: asd123, email: mali@gmail.com <p>B: Edit password</p> <ol style="list-style-type: none">1. Current password: qweqwe, new password: 123123, re-enter the new password: 1231232. Current password: abcdef, new password: admsdm, re-enter the new password: admsdn

Expected Output	A: Edit profile 1. Profile updated B: Change Password 1. Password updated 2. Alert: New password doesn't match with the re-typed password
Observed Output	A: Edit profile 1. Profile updated B: Change Password 1. Password updated 2. Alert: New password doesn't match with the re-typed password
Verdict	Pass

4.3.3 TC-3: Request Fuel

Table 31. TC-3 Request Fuel

ID	TC-3
Description	Check the manager can request fuel with the valid remaining fuel quantity
Setup	1. The web application is running. 2. The request fuel page is opened.
Inputs	1. Requested quantity: 100 Liters while the remaining quantity hasn't reached the less than 10% limit. 2. Requested quantity: 100 Liters while the remaining quantity: 5% 3. Request Quantity>100 liters whereas maximum request allowed is 100 liters
Expected Output	1. Alert: The Remaining quantity hasn't reached the < 10% hence Fuel is not allocated 2. Fuel is allocated successfully 3. Alert: The requested quantity exceeds limit
Observed Output	1. Alert: The Remaining quantity hasn't reached the < 10% hence Fuel is not allocated 2. Fuel is allocated successfully 3. Alert: The requested quantity exceeds limit
Verdict	Pass

4.3.4 TC-4: Add Consignment to Delivery Sheet

Table 32. TC-4 Add Consignment to Delivery Sheet

ID	TC-4
Description	Check if the supervisor or manager can add a >2kg consignment into the bike's delivery sheet
Setup	<ol style="list-style-type: none"> 1. The web application is running. 2. The Edit Delivery Sheet page is opened.
Inputs	<ol style="list-style-type: none"> 1. Add Consignment with weight < 2kg to the delivery sheet for the bike 2. Add consignment with weight > 2kg to the delivery sheet for the vehicle 3. Add consignment with weight > 2kg to the delivery sheet for the bike
Expected Output	<ol style="list-style-type: none"> 1. New Consignment is added to the delivery sheet 2. New Consignment is added to the delivery sheet 3. Error: The consignment could not be add to the delivery sheet because it's > 2kg and the delivery sheet is for Bike
Actual Output	<ol style="list-style-type: none"> 1. New Consignment is added to the delivery sheet 2. New Consignment is added to the delivery sheet 3. Error: The consignment could not be add to the delivery sheet because it's > 2kg and the delivery sheet is for Bike
Verdict	Pass

4.3.5 TC-5: Generate Delivery Sheet

Table 33. TC-5 Generate Delivery Sheet

ID	TC-5
Description	Check if the supervisor or manager can generate a delivery sheet with wrong area code
Setup	<ol style="list-style-type: none"> 1. The web application is running. 2. The Generate Delivery Sheet page is opened.

Inputs	<ol style="list-style-type: none"> 1. AreaCode = 4320 which doesn't exist in the areasList 2. AreaCode = 4520 which exists in the areasList
Expected Output	<ol style="list-style-type: none"> 1. Alert: The given area doesn't exist in the system 2. Delivery sheet generated successfully
Actual Result	<ol style="list-style-type: none"> 1. Alert: The given area doesn't exist in the system 2. Delivery sheet generated successfully
Verdict	Pass

4.3.6 TC-6: Add Vehicle

Table 34. TC-6 Add Vehicle

ID	TC-6
Description	Check if the manager can add a new vehicle successfully with correct data
Setup	<ol style="list-style-type: none"> 1. The web application is running. 2. The Add Vehicle page is opened.
Inputs	<ol style="list-style-type: none"> 1. Vehicle name: Toyota Hilux, vehicle type: Jeep, make: Toyota 2. Vehicle name: Suzuki Carry, vehicle type: <empty>, make: Suzuki
Expected Output	<ol style="list-style-type: none"> 1. Vehicle has been added successfully 2. Alert: The vehicle type field can not be empty
Actual Result	<ol style="list-style-type: none"> 1. Vehicle has been added successfully 2. Alert: The vehicle type field can not be empty
Verdict	Pass

References

1. IEEE Std 830-1998 (Revision of IEEE Std 830-1993), IEEE Recommended Practice for Software Requirements Specifications.
2. Software Requirements Specification Document Sunil & Pradhan.
3. Software Engineering, a Practitioners Approach by Rogers Pressman, 7th Edition.
4. “Introduction”, Scribd, [Online]. Available:
<https://www.scribd.com/document/391032851/IOT-Based-Smart-Home>. [Accessed 01 11 2019]
5. “Software Process Model”, Google, [Online]. Available:
<https://www.tutorialride.com/software-engineering/prescriptive-process-models.htm>.
[Accessed 05 11 2019]
6. IEEE recommended practice for software requirements specifications (Approved 25 June 1998, IEEE-SA Standards Board).
7. Craig Larman, “Applying UML and Patterns”, 3rd ed, [20 10 2004]

Appendix

- Delivery Sheet - Manually generated TCS Delivery Sheet photo taken from TCS delivery person

TCS Express & Logistics		Shipment Delivery Record					
DELIVERY DATE:	08/04/2014	LHE			ROUTE		GWJ14
SHEET NUMBER:	LHP73711	STATION:			COURIER OFFICER:		43216
THE CONSIGNEE DECLARIES THAT HE/SHE HAS RECEIVED THE SHIPMENT (S) IN GOOD ORDER AND CONDITION							
SR	CONSIGNMENT DETAILS	RECEIVER'S NAME	SIGN / COMMENTS	SR	CONSIGNMENT DETAILS	RECEIVER'S NAME	SIGN / COMMENTS
1	4561084803 MR SHABBIR'S 1	Muzamil AMP Abbas	M(✓)	11	306013685700	MRS TAZ 1009	
2	CNIC # 858-C 4760713197 AAA	0		12	CNIC # 350-m / 0337-165 HASAN KHI .5 1	HASSAN Riaz	✓✓
3	CNIC # 858-C 14000944643	0	M. (✓)	13	CNIC # 334-0 ANEES	ANEES	✓✓
4	CNIC # 333-R (0322-7223404) 4561074659 MS ATIQA BAT MUX .5 1	12:15 Rep Zameer		14	CNIC # 428-L FAISAL BHAI FSD 1.5 1	MUSTAFA B	✓✓
5	CNIC # 425-R 6060347574 NARGIS DIYAL SWL .5 1	SKM Deep 12:09		15	CNIC # 157-L MUSHTAQ LHE .5 1	NAREEM	✓✓
6	CNIC # 425-R 1460932591 ELEGANCE THE GJT .5 1	YASIR		16	CNIC # 944-L (3530-2047)		✓✓
7	CNIC # 47-L 30043618320 SYED AJMAL KHI .5	13:58 CINIA		17	CNIC # 2298-R 10:00		✓✓
8	CNIC # 641-L 4760711826	641-L (2298-R) Mohsin Mohsin		18	CNIC # 1127-O 5061233984 YAWAR KHAN RWP .5 1	RATI	✓✓
9	CNIC # 258-R 306013841581	12:05 KURRAM K		19	CNIC # 1127-O 259-R (0321-9905805)		✓✓
10	CNIC # 1127-O 5061233984 YAWAR KHAN RWP .5 1	12:04 Musallat		20	CNIC # 1127-O 259-R (0321-9905805)		✓✓
TOTAL SHIPMENTS: 15				TOTAL PIECES: 15		Debriefed By: MUHAMMAD IRFAN (51571) Date: 8/4/14 Time: 2:17 Delivered _____ Undelivered _____ Sing: PAKWHEELS.COM	
Courier Officer				Security Officer		Operation Controller	