AUTOHUB

A WEB BASED SPARE PART MANAGEMENT SYSTEM

By

H.A. Perera

IM/2012/042

A report submitted in fulfillment of the requirements for the degree of Bachelor of Science in Management and Information Technology (B.Sc. MIT)

Name of the supervisor: Dr. Ruwan Wickramarachchi

B.Sc. in Management and Information Technology-2012/2013

Department of Industrial Management

University of Kelaniya

Sri Lanka

2017

Declaration

I hereby certify that this project and the all the artefacts associated with it is my own work and it has not been submitted before nor is currently being submitted for any other degree Programme.

Full name of the student: Heshan Anupama Perera

Student No: IM/2012/042

Signature of the student: Date:

Name of the supervisor(s): Dr. Ruwan Wickramarachchi

Signature of the supervisor:…….. Date:…….

Acknowledgement

I would like to thank my initial project supervisor, Mr. Amila Withanarachchi, and the current supervisor Dr. Ruwan Wickramarachchi for the continuous guidance given, initially for supporting me in preparing for the poster presentation and then to successfully complete the system analysis and design stage of the project. I sincerely appreciate the encouragement and support given by Dr. Shantha Jayalal, project coordinator during the system analysis and design stage. Also, I thank all other staff members. The comments I had from them, during the initial stages were very important in determining the functionalities and the scope of the project.

I would also like to extend my gratitude to Mr. Rohan Fernando the owner of St. Anne's Auto Spares, Wennappuwa for giving me the continuous support by giving information about the way sales and other business related things work in the field of automobile spare parts industry.

Last but certainly not least I owe many thanks to my fellow batch mates who helped me out in numerous ways in the system analysis and design stage of the project.

Heshan Anupama Perera
19/09/2016

Abstract

Automobile spare parts industry is one of the most growing industries in the world and as well as in Sri Lanka. Unlike in many other countries we don't have any common marketplace to buy these auto spares. Basically, customers and as well as retailers find many different disadvantages of the current manual process. Taking this matter in to consideration my main objective is to develop a web based E-commerce website which will eventually bring retailers and customers in to one marketplace.

Once the project is completed customer can access multiple different functionalities. AutoHub facilitates customers to compare spares of different retailers all around Sri Lanka, order different spares, buy multiple items through shopping cart, use secure payment methods, make enquiries and feedbacks etc.

Retailer on the other hand can register to the website after agreeing on an agreement. They can then manage their account and update their products via the website so that they appear to the customers to be purchased. They can also get the order details about a certain purchase and also get the ability make different kinds of reports related to sales that supports in decision making.

Table of contents

List of Figures

List of Tables

1 Introduction

This chapter gives a brief introduction of the AutoHub spare parts system Web solution, need of the applications and user base. It also describes the scope of the project, aims and objectives of the proposed system. At the end, this chapter explains the feasibility and tools.

1.1 Introduction

Information Technology is initiating change in every aspect of how we live our day today lives. With the development of the Internet had an immediate effect the life of common people and started to use all the facilities and enjoy using them. E-commerce concept is emerged as a result of Internet and other computer networks, where people buying and selling products or services over electronic systems. As a result, Internet shopping became very popular because that it can be brought in a single click.

Automobile spare part industry is a highly developing industry in the present world. However, searching spare parts manually from different shops for the automobiles is currently a time consuming and a hard process in the customer's perspective. They find many difficulties while doing so.

There are many variations in prices from shops to shops. Customer often find difficulties while accessing product information of some products through different shops. Sometimes they won't even find the matching spare for their vehicle which lead to a high rate of turnover and exchanging. Practically the owners of rare vehicle brands and models are facing much more difficulties than other commonly used vehicles by these processes. Above all, customer often face and feel that their time being wasted.

I hope to implement a solution for this problem by an E-commerce web based system where a customer can search the relevant spare parts and also allowing them to search through flexible methods like categorize spares according to the different categories like engine parts, electrical parts, body parts etc.

1.2 Need Analysis

1.2.1 Problem Definition

According to the current system all the processes are done manually in most of the spare part retail shops, where a customer comes to a spare part shop, ask for a spare part and the product is searched from the inventory and then he buy the product if it matches the correct spare. This process is highly time consuming and the retailers would not be able to provide information concurrently to number of customers. Eventually this will result in providing incorrect information about spare parts, customer dissatisfaction etc.

Most of the shops are only opened for around 9 hours per day while closed throughout the Sundays and other holidays. Customer are not having the opportunity to access the services from the retail shops at desired times.

For the retailers, according to the current system they have to maintain an inventory in the shop itself and this will lead to an increment of inventory cost and eventually the profit margins will be different. And also, there is no proper way to analyze the sales during past periods and they have to go through each and every record to find out the spare parts with a high demand during a particular time period.

1.2.2 Existing manual system

- Customer ask for a spare part by directly coming to the shop or by calling to the shop.
- Retailer will go through the catalog or database (if present) to check whether spare is available with them and reply to the customers that seek for information.
- If customer need the spare retailer will find the spare from the inventory after checking whether the desired quantities are found.
- Retailer agrees on a discount and a warranty period if applicable.
- Payment is done and a receipt is handed over to the customer.
- Sales records are recorded manually by retailer.
- In case of an exchange, they are handled at the shops depending on the warranties and condition of the item.

1.2.3 Drawbacks of the current process.

- Inability to access services throughout 24x7.
- Wastage of Time.
- Price and quality variations from shop to shop.
- Unable to get reliable product information.
- High rate of exchanging.
- Excess amount of manual recording.
- Difficulties in analyzing sales.
- Cost of inventory.

1.3 Benefits of the Proposed System

The proposed system has considered and analyzed the needs in the perspective of both the customer and the retailer. So, with the implementation of the system both the customer and the retailer will be benefited.

Benefits to Customers.

Enabling customer to make search on the item by himself via the website.

With just a few steps customer can search a particular spare via many different flexible searching methods.

Ability to make enquiries.

Customers can use the service to enquire about the status and information about spare parts if they are unable to find spares by searching.

Having the facility to make complains.

It's always a common defect in an e-commerce website where customers didn't receive the products, product received may be changed or not in the condition as described. So, clarify this sort of problems customer can easily file a complaint against an order which will be redirected back to the particular retailer.

Enabling customers to make ratings and reviews on the purchased products.

This is one of the main benefit of the new proposed system. This will help the customers to decide on buying a certain product from a certain retailer. There will be a certain rating for the retailers calculated by previous customer added ratings and reviews and from it whenever a customer buys a product he can verify whether the seller has a good rating by reading the reviews and ratings that have already placed.

Customers can compare spare parts with different retailers.

As the website is interconnected with number of retailers, for a one spare part there can be number of retailers.

Secured online payments.

For an E-commerce website, it's always common to include payments via online. For a website to be successful this payment must be through a secured way so in this system I am hoping to implement payment via PayPal which is the most famous way of online payment.

24x7 Access

Customer can consume the services whenever he likes as there are no any limitations regarding the time.

Benefits to the Retailers.

24x7 open.

Not only for the customer, retailers also like to extend the time period of shops opened so that a customer can buy spares for a long period of time. With the system retailer, can make their shop visible to the customers throughout 24x7.

Face Higher competition.

As there are many retailers bought together to a single marketplace through this website there can be many sellers offering same spares which will eventually lead to a competition among retailers where every retailer would try to do his/her best to attract more and more customers.

Decrease in the inventory costs.

Not like a real-time purchasing retailers do not need to keep higher quantities of the spare parts in their stocks/inventories where their costs of inventory will go down.

Generation of reports.

Almost in every spare shop they are using manual records whenever they need to analyze the sales during a particular time period. But this system is supposed to have a report generation section where all the details of the sales are taken in to consideration and reports will be generated accordingly.

Source of Marketing.

Almost every business in the present world that are having websites has the facility to market the business without any costs. So, retailers that would subscribe to AutoHub website would have a way of marketing their business through the system.

1.4 Scope of the project

As the main goal of the project is to facilitate a high-level assistance for shopping spare parts by bringing number of retailers and customers to a single marketplace. Though the main focus of the system is to achieve the best to the customer, system is not only customer oriented there many functionalities and advantages that a retailer achieves with this system. The functionalities and the scope is summarized in the following figure.

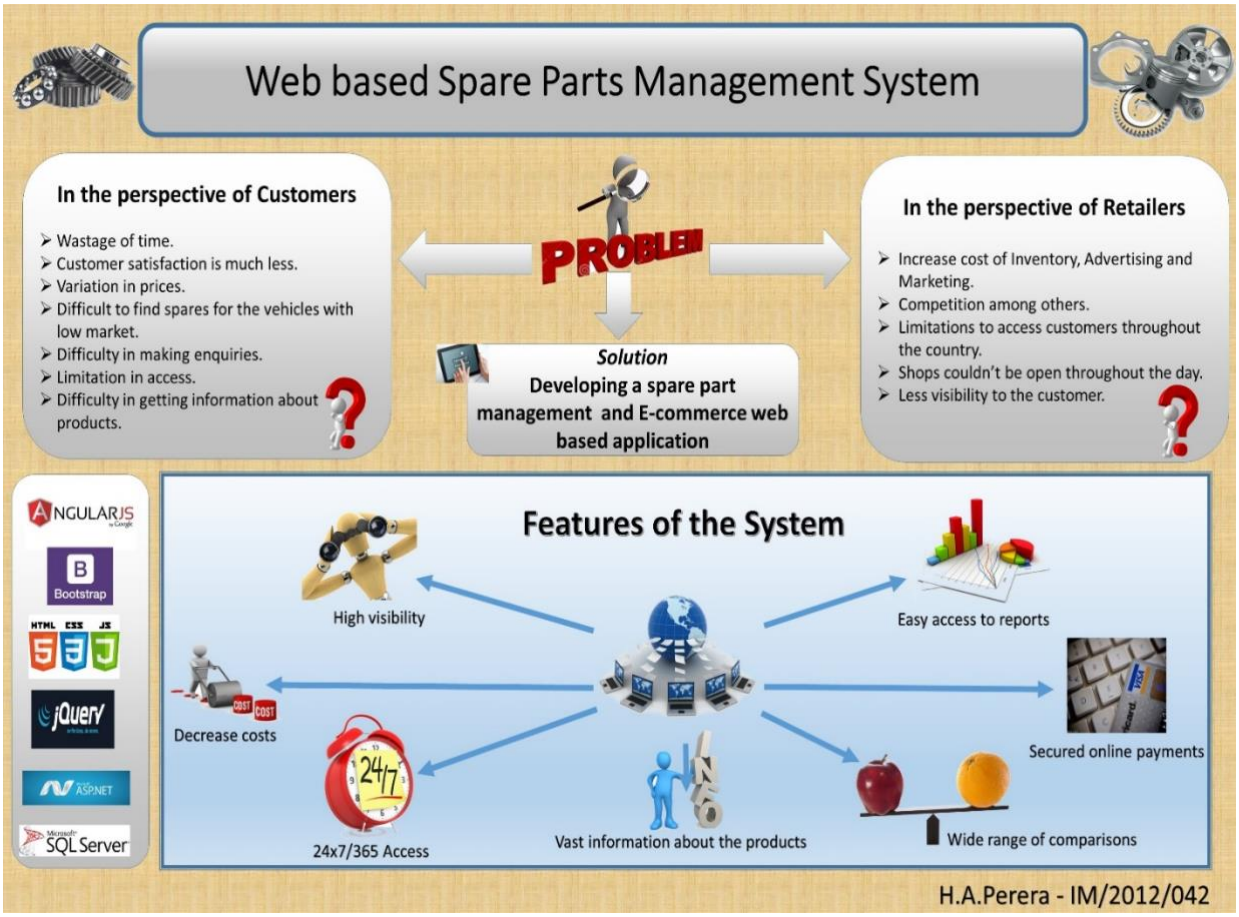


FIGURE 1.4.1-SCOPE OF THE PROJECT

1.5 Objectives

From this project, for a customer it is expected to support and facilitate the online shopping experience of Automobile spare parts and accessories via a web application. Furthermore, this will take online shopping in to a whole new level by making it more efficient through effective features. In the other side, for a particular retailer can use the web solution of the AutoHub to feed the data into the inventory (database) and retrieve sales and other indexes to make better decisions in the near future. Since this is an individual project, the developer will also be exposed to the requirement analysis as well. Though AutoHub system not have a particular client as it is made with a combination of number of sellers it is a must to communicate with at least three of the retailers throughout this development to clarify problems and to understand how this spare part industry works practically in the present world. This will allow gain experience on designing the system ‘s interfaces with including functionalities that are capable of higher usability.

1.6 Tools and Feasibility

Since there is no client involved in the project, gathering requirements and other relevant information will be difficult. For the marketing information, surveys and questionnaires can be used. The web development is to be done using JavaScript, AngularJS, JQuery, Ajax, Bootstrap, SQL and ASP.net. Since this is an E -commerce based online shopping experience, there will be no issues with the implementation. Users will be familiar with the concept and it will take them to a whole new level in online shopping and they will enjoy a lot.

Project Feasibility

Since there are around three retailers to communicate with it is very easy to understand the requirement gatherings and sorting out misunderstandings with the help of them.

Technical Feasibility

There are many sources to understand and learn powerful languages and frameworks and styling sheets like JavaScript, AngularJS, JQuery, Ajax, Bootstrap SQL and ASP.net

Economic Feasibility

Though most of the tools that are used in the development are free, we need to pay for a server to host the website after development.

2 System Analysis

This chapter describes the current scenario of the problem situation. It describes the current process through the use of case diagrams. This chapter also contains some activity diagrams further explaining some use cases.
Additionally, this chapter contains a discussion on possible business system options followed by a requirement specification. Finally, the possible BSO's are evaluated against the requirement catalogue and justification of choosing the best BSO through the proposed system is provided.

Outline of the chapter

2.1 Overall use case diagram for current system
2.2 Activity diagrams elaborating some use cases further
2.3 System Requirement analysis and Requirement Specification
2.4 Business System Options
2.5 Cost Benefit Analysis for BSOs
2.6 Selected BSO with a Sound Justification

2.1 Overall Use case diagram for current manual system.

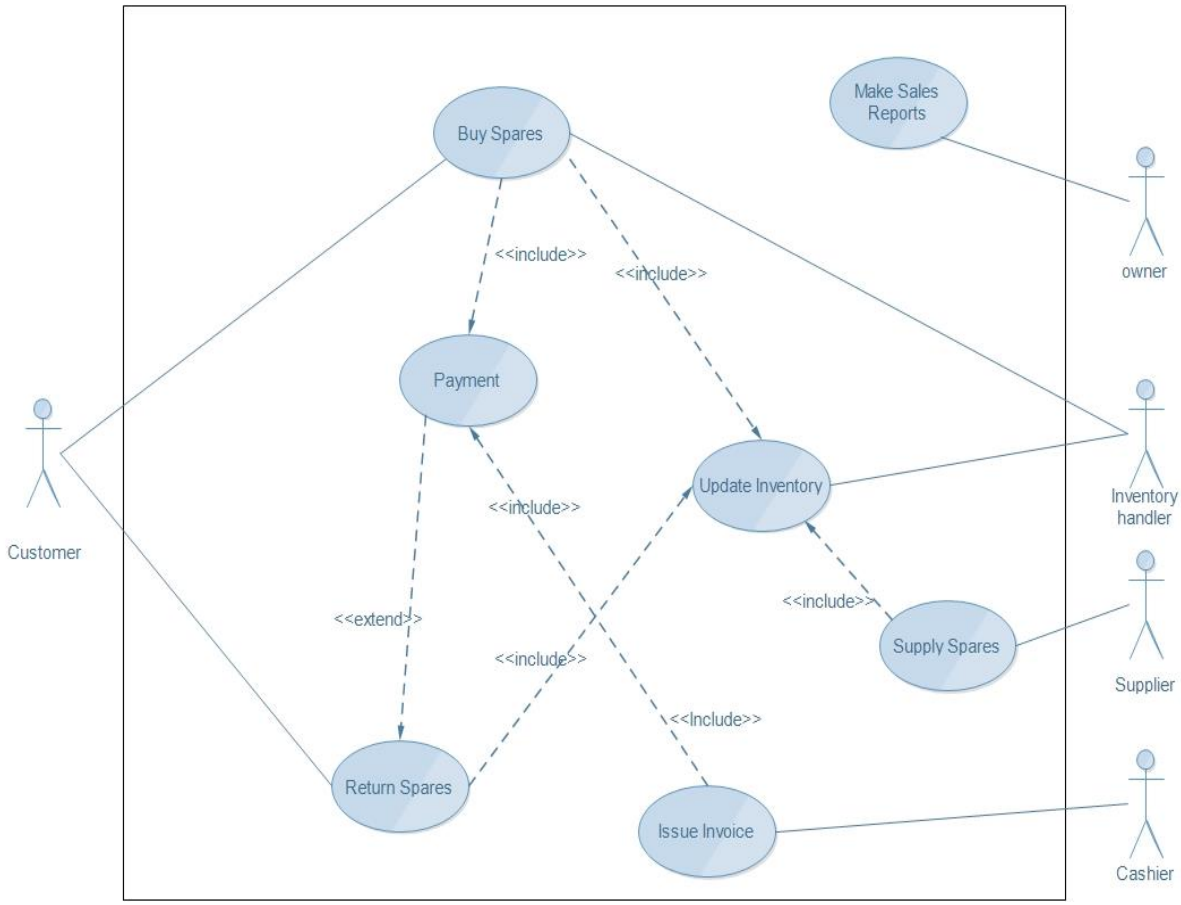


FIGURE 2.1.1 OVERALL USE CASE FOR CURRENT SYSTEM

This use case diagram shows the current processes in a typical spare part shop. The main roles are performed by cashier, supplier, inventory handler, owner and the customer. The use cases represent the activities which takes place in the shop in the above figure.

2.2 Activity diagrams for some of the use cases in the use case diagram

The figure 2.2-1 illustrates the activity diagram to describe the functionality of the buying a spare through a spare part shop identified in the existing use case in the manual process.

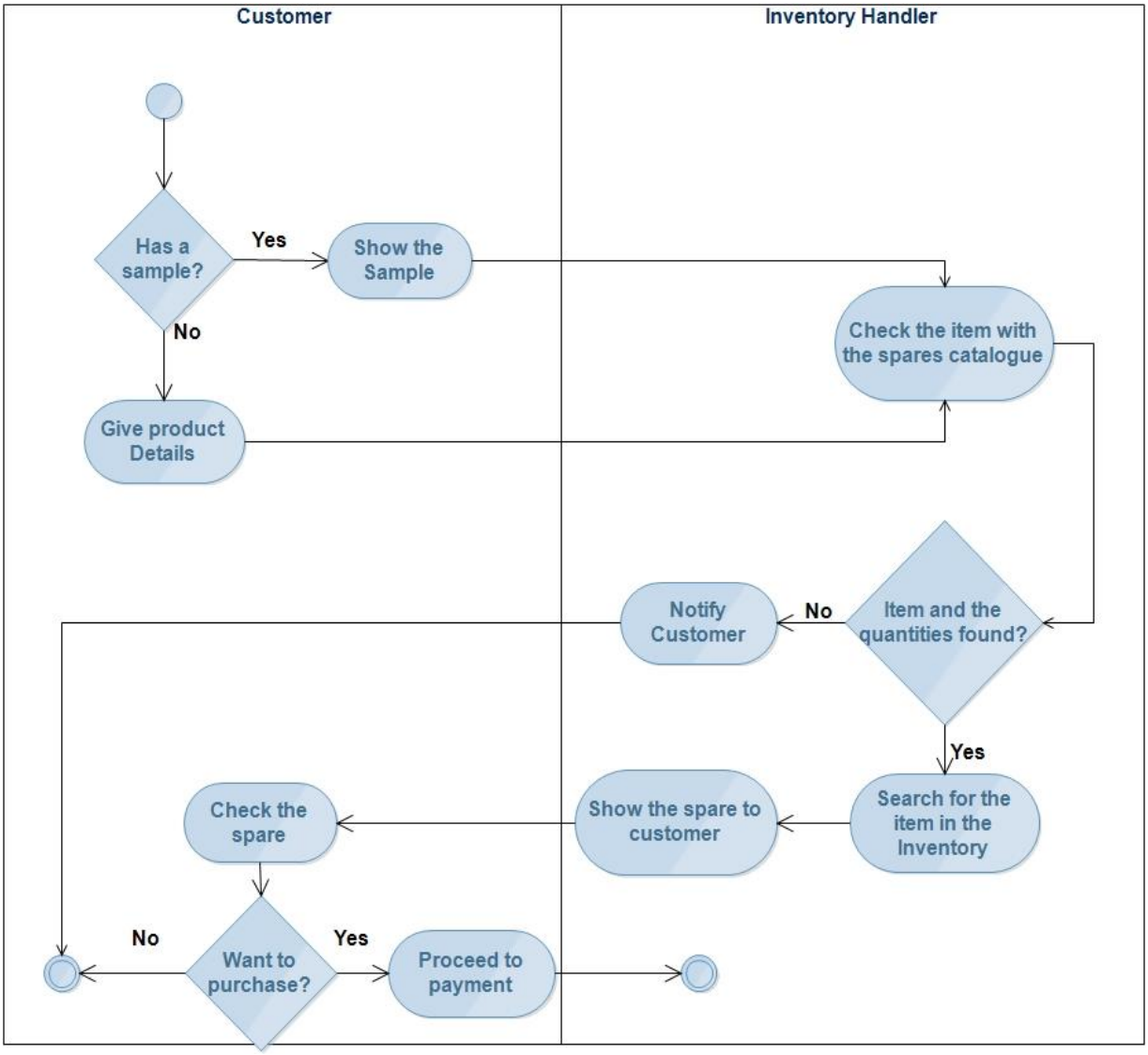


FIGURE 2.2.1 ACTIVITY DIAGRAM FOR EXISTING BUYING PROCESS

The figure 2.2-2 illustrates the activity diagram to describe the functionality of the payment use case identified in the existing use case in the manual process.

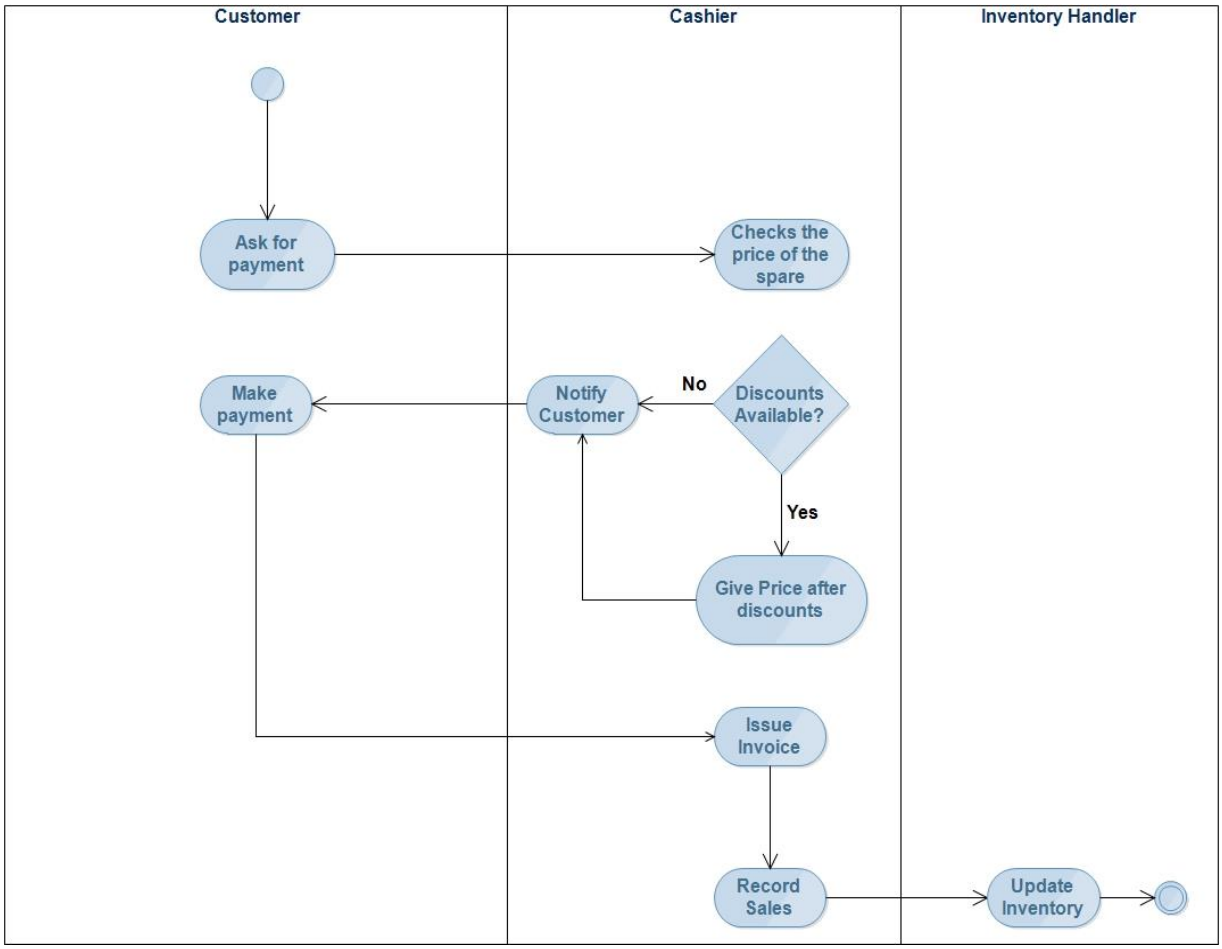


FIGURE 2.2.2 ACTIVITY DIAGRAM FOR PAYMENT IN EXISTING SYSTEM

The figure 2.2.3 illustrates the activity diagram to describe the functionality of the exchanging a spare use case identified in the existing use case in the manual process.

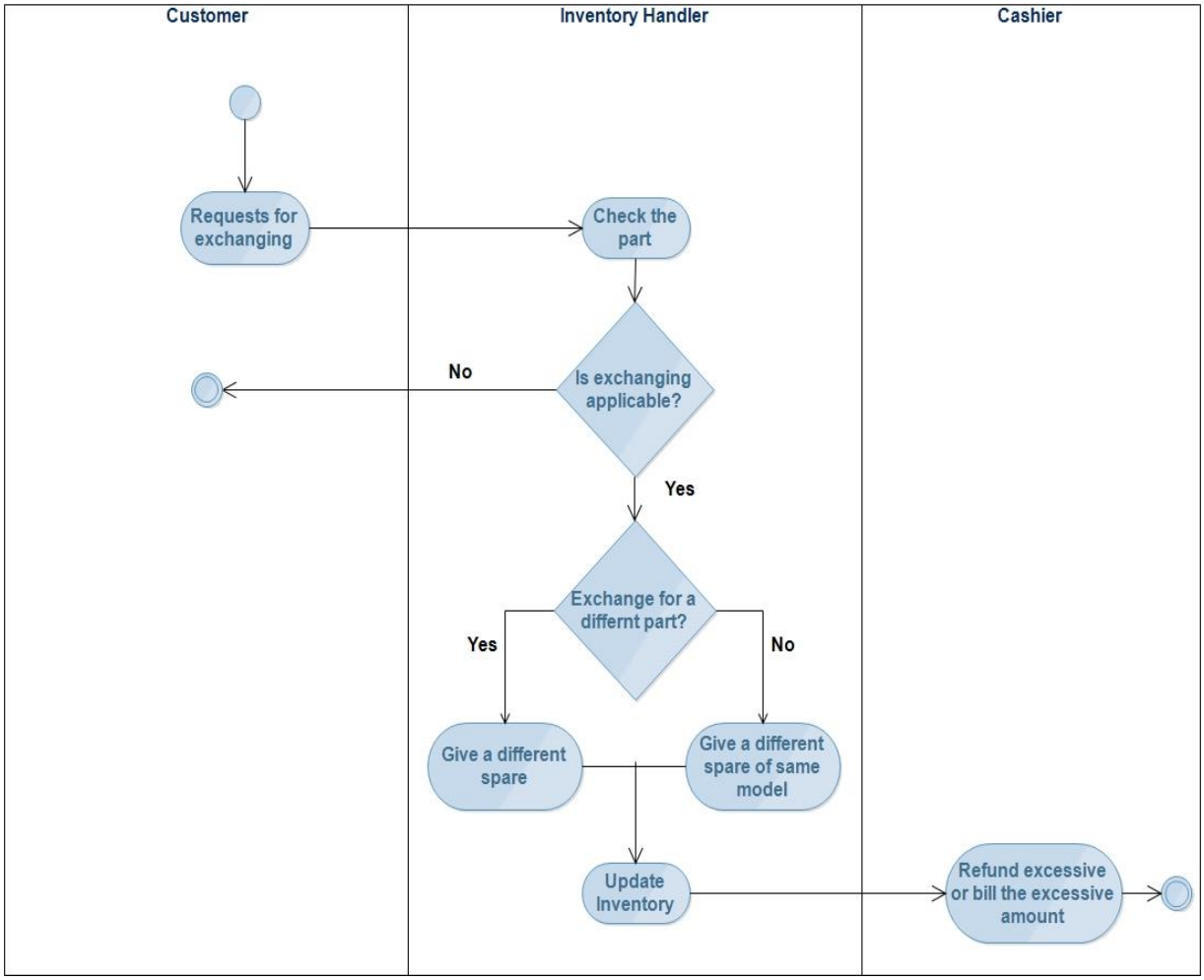


FIGURE 2.2.3 ACTIVITY DIAGRAM FOR EXCHANGING A SPARE IN EXISTING SYSTEM

2.3 System Requirement analysis and Requirement Specification

2.3.1 Functional Requirements

Functional requirements describe what the system should be able to do in order to satisfy the main objective of it. Mainly these requirements describe what are the tasks or functions which the system can perform after implementation.

Following table will show functional requirements for administrator/retailer side

1	Shall be able to administrator, retailer to login or logout of the system.
2	Shall be able to administrator to register retailers to the system.
3	Shall be able to administrator to insert, delete or update vehicle brands and models.
4	Shall be able administrator to view requests for new brands and models by retailers.
5	Shall be able to administrator to add new spare parts category to the system.
6	Shall be able to update and delete retailers accounts.
7	Shall be able to retailer to login/logout of the system.
8	Shall be able to retailer to update his/her account.

10	Shall be able to retailer to add new requests to add vehicle models and brands.
11	Shall be able retailer to view his/her current inventory.
12	Shall be able retailer to add items to his/her inventory.
13	Shall be able retailer to update the quantities of spares in his/her inventory
14	Shall be able retailer to view the enquiries of the customers.
15	Shall be able to retailer to response back to the customers.
16	Shall be able to retailer to view orders placed by customers.
17	Shall be able to retailer to update the order status.
18	Shall be able to retailer to generate sales related reports.
19	Should be able to provide retailer better decision making.
20	Should be able to add product information based on different spare categories.

TABLE 2.3.1 FUNCTIONAL REQUIREMENTS FOR ADMINISTRATOR/RETAILER

Following table shows functional requirements for customer.

1	Shall be able to create accounts and update customer basic account.
2	Shall be able to log in to the system log out of the system.
3	Shall be able to customer to set default delivery address.
4	Shall be able to browse spares according to different categories.
5	Shall be able to apply number of filters for browsing.
6	Shall be able to check availability of products.
7	Shall be able to insert products into shopping cart and edit and remove form shopping cart.
8	Shall be able to make enquiries.
9	Shall be able to online pay for purchased items.
10	Shall be able to view order status.
11	Shall be able to compare prices and products of different retail shops.
12	Shall be able to rating products quality after the delivery and purchase of a specific order.
13	Shall be able to add reviews on products.

14	Shall be able to access product information of spares.
15	Shall be able to view the ratings/reviews placed by other customers on spares and retailers.
16	Shall be able to ask questions from retailers.
17	Should be able to receive alerts on recommended products and offers.
18	Should be able to view large size images of products (zoom).
19	Should display location of shop and retailer contact details for a particular product.

TABLE 2.3.2 FUNCTIONAL REQUIREMENTS FOR CUSTOMER

2.3.2 Non-Functional Requirements

Non-functional requirements describe the behavior of the system other than the main functionalities of it. Hence, they will cover the requirements which are not included in section Non-functional requirements describe the usability, reliability, performance, maintainability and other similar aspects of the system. These set of requirements may not be directly related to the main functionality but they are of extreme importance to the proper functioning of the system. Following table will show Non- functional requirements for the AutoHub system.

1	Shall be able to present user friendly interfaces.
2	Shall be able to handle around thousand users concurrently.
4	Shall be able to access via smartphones.
5	Shall be able to provide the safety and security, system
6	Shall be able to provide security in online payments.
7	Should be able to secure information from external parties
8	Should be responsive immediately to customer requests
9	Should be able to real-time update changes immediately.

TABLE 2.3.3 NON-FUNCTIONAL REQUIREMENTS

2.4 Business System Options

Since this project does not have an actual business client, in fact after the implementation of the system there would be number of retailers connected, the business system options do not have to be focused on satisfying retailer's requirements thoroughly. The features which the BSOs are offering should satisfy mainly the customer's online shopping requirements and retailer's inventory update and other analyzing requirements. According to these requirements all the BSOs must provide Rich Internet Applications. Because any spare part retailer that have subscribed to the system around Sri Lanka should be able to access into the web site and the other hand any interested customer should be able to access to browse and order products. It should be feasible with the developer as well.

When it comes to technical aspects, these options can be varied according to the Operating System versions used (OS), client end, server end (web server), web services, use of open source software, mobile applicability and the platform dependencies etc.

In this section three BSOs will be presented and at the end they will be evaluated to come up with the best option. Each BSO will be consisted of overlapping features as well as different features. Hence each BSO will be focused on different aspects of the AutoHub system. Evaluation of the BSOs will be done by comparing them against the functional and non-functional requirements of the system.

2.4.1 BSO 1: Customer side and retailer side both integrated into one web based solution

This BSO is regarding on introducing a new web-based application for both seller and buyer. Depending on the login username and password user will be redirected to the relevant dashboards. Accessibility is available from anywhere where there is an internet connection along with a device with internet accessibility

Functionalities:

This will basically include all the online shopping activities that are related to the customers and data entry and other activities related to a retailer.

Customer side: User can create their profiles through the website, logged to the site with securities, order items, use shopping cart, view images and documents, compare products, check product availability, pay securely via PayPal, check product reviews and ratings and compare retailer ratings will integrated with this BSO. All the other functional requirements specified in 2.3.1 can satisfy through the solution. User also can access to web interface using their mobile phone when Internet access is enable.

Retailer side: Spare parts and accessories sellers in Sri Lanka can access the website after the administrator register them in the system and provide them with a username/password which can be changed later according to their preferences. Web site and can input product details into the system. And all the other requirements specified in 2.3.1 can fulfil using the integrated web solution.

Justification: This BSO satisfies almost all of the functional requirements and the non-functional requirements. Since this is an Internet solution it is give flexibility to both customer and merchant to use it access from anywhere.

Benefits to both customer and retailer:

- Nothing to download/install - avoid the burden in deploying in each retailer's/customer's machine.
- Updates are easier – No need to update the versions regularly.
- Attractive and responsive GUI – Unlike windows and mobile applications this will provide a better view and offer more friendly UIs.
- Can access from anywhere – Access from anywhere where there is internet and a device which is capable of connecting to internet
- Platform independent – The customers/retailers operating systems and hardware doesn't really matter.

Approximate Cost Estimation:

Assuming merchants and customer's computers are available.

	Estimated Cost(LKR)
Server Computer	Rs 200000.00
Hosting Service	Rs. 15000.00
Internet cost	Rs. 5000.00
Routers and other networking hardware	Rs. 8000.00
Kaspersky Internet Security	Rs. 10000.00
Electricity	Rs. 5000.00
Total Cost	**Rs. 243,000.00**

TABLE 2.4.1 BSO1 COST

2.4.2 BSO 2: Retailer side windows application and customer a web based solution

Functionalities:

This is concerning on developing a windows application to the retailers while customers could access the functionalities through a web based solution.

Customer side: User can create their profiles through the website, logged to the site with securities, order items, use shopping cart, view images and documents, compare products, check product availability, pay securely via PayPal, check product reviews and ratings and compare retailer ratings will integrated with this BSO. All the other functional requirements specified in 2.3.1 can satisfy through the solution. User also can access to web interface using their mobile phone when Internet access is enable.

Retailer side: Developed windows application must be installed in to every retailer's PC who are willing to be connected to the website. Application is capable of providing almost all of the functional requirements but there are some of the non-functional requirements that couldn't be provided.

Benefits to customers:

- Nothing to download/install – Customers doesn't need to install anything it's just a simple surf through the internet.
- Updates are easier – No need to update the versions regularly.
- Attractive and responsive GUI – Unlike windows and mobile applications this will provide a better view and offer more friendly UIs.
- Can access from anywhere – Access from anywhere where there is internet and a device which is capable of connecting to internet.
- Platform independent – The operating systems and hardware doesn't really matter.

Issues related to retailers:

- Application should be installed in every retailer's computer.
- Platform Dependent.
- Current resources should be compatible with the application.
- User interfaces are not much attractive and responsive.

Approximate Cost Estimation:

Assuming merchants and customer's computers are available.

	Estimated Cost(LKR)
Server Computer	Rs. 200000.00
Hosting Service	Rs. 15000.00
Internet cost	Rs. 5000.00
Routers and other networking hardware	Rs. 8000.00
Kaspersky Internet Security	Rs. 10000.00
Electricity	Rs. 5000.00
Desktop Application Installation and maintenance cost	Rs. 5000.00
Total Cost	**Rs. 248,000.00**

TABLE 2.4.2 BSO2 COST

2.4.3 BSO 3: Web based retailer side and mobile based customer side

Functionalities:

This BSO is concerning on introducing an android application for buyer and web solution for sellers. This will be integration of android application and web site with added functionalities. Accessibility is available from anywhere anytime there is an internet access**.**

Retailer side: Spare parts and accessories sellers in Sri Lanka can access the website after the administrator register them in the system and provide them with a username/password which can be changed later according to their preferences. Web site and can input product details into the system. And all the other requirements related to retailers specified in 2.3.1 can fulfil using the integrated web solution.

Customer side: User can create their profiles through the mobile application, logged to the site with securities, order items, use shopping cart, view images and documents, compare products, check product availability, pay securely via PayPal, check product reviews and ratings and compare retailer ratings will integrated with this BSO. All the other functional requirements specified in 2.3.1 can satisfy through the solution. But there are some of the non-functional requirements that this BSO would not be able to cover.

Benefits to the retailers:

- Nothing to download/install - avoid the burden in deploying in each retailer's machine.
- Updates are easier – No need to update the versions regularly.
- Attractive and responsive GUI – Unlike windows and mobile applications this will provide a better view and offer more friendly UIs.
- Can access from anywhere – Access from anywhere where there is internet and a device which is capable of connecting to internet
- Platform independent – The retailers operating systems and hardware doesn't really matter.

Issues related to customers:

- Incompatibility – Unlike a website can reach users across many different types of mobile devices, native apps require a separate version to be developed for each type of device.
- Non-Immediacy - Require the customers to first download and install the app from an app marketplace before the content or application can be viewed.
- Difficult in upgrading - Changes are not immediately visible customers should update the app each time when there's a new version which then must be downloaded on each type of device.

Approximate Cost Estimation:

Assuming merchant's computers and customer's Mobiles are available.

	Estimated Cost(LKR)
Server Computer	Rs. 200000.00
Hosting Service	Rs. 15000.00
Internet cost	Rs. 5000.00
Routers and other networking hardware	Rs. 8000.00
Kaspersky Internet Security	Rs. 10000.00
Electricity	Rs. 5000.00
Application Installation and maintenance cost	Rs. 5000.00
Google play store account	Rs. 3500.00
Total Cost	**Rs. 251,500.00**

TABLE 2.4.3 BSO3 COST

2.5 Evaluation of the BSOs

2.5.1 Functional Requirements vs BSO's

ID	Description	BSO1	BSO2	BSO3
1	Shall be able to administrator, retailer to login or logout of the system.	✓	✓	✓
2	Shall be able to administrator to register retailers to the system.	✓	✓	✓
3	Shall be able to administrator to insert, delete or update vehicle brands and models.	✓	✓	✓
4	Shall be able administrator to view requests for new brands and models by retailers.	✓	✓	✓
5	Shall be able to administrator to add new spare parts category to the system.	✓	✓	✓
6	Shall be able to update and delete retailers accounts.	✓	✓	✓
7	Shall be able to retailer to login/logout of the system.	✓	✓	✓
8	Shall be able to retailer to update his/her account.	✓	✓	✓
9	Shall be able retailer to view and access currently present spare categories, vehicle brands and models.	✓	✓	✓

10	Shall be able to retailer to add new requests to add vehicle models and brands.	✓	✓	✓
11	Shall be able retailer to view his/her current inventory.	✓	✓	✓
12	Shall be able retailer to add items to his/her inventory.	✓	✓	✓
13	Shall be able retailer to update the quantities of spares in his/her inventory	✓	✓	✓
14	Shall be able retailer to view the enquiries of the customers.	✓	✓	✓
15	Shall be able to retailer to response back to the customers.	✓	✓	✓
16	Shall be able to retailer to view orders placed by customers.	✓	✓	✓
17	Shall be able to retailer to update the order status.	✓	✓	✓
18	Shall be able to retailer to generate sales related reports.	✓	✓	✓
19	Should be able to provide retailer better decision making.	✓	✓	✓
20	Should be able to add product information based on different spare categories.	✓	✓	✓

1	Shall be able to create accounts and update customer basic account.	✓	✓	✓
2	Shall be able to log in to the system log out of the system.	✓	✓	✓
3	Shall be able to customer to set default delivery address.	✓	✓	✓
4	Shall be able to browse spares according to different categories.	✓	✓	✓
5	Shall be able to apply number of filters for browsing.	✓	✓	✓
6	Shall be able to check availability of products.	✓	✓	✓
7	Shall be able to insert products into shopping cart and edit and remove form shopping cart.	✓	✓	✓
8	Shall be able to make enquiries.	✓	✓	✓
9	Shall be able to online pay for purchased items.	✓	✓	✓
10	Shall be able to view order status.	✓	✓	✓
11	Shall be able to compare prices and products of different retail shops.	✓	✓	✓
13	Shall be able to add reviews on products.	□	□	□
14	Shall be able to access product information of spares.	✓	✓	✓

15	Shall be able to view the ratings/reviews placed by other customers on spares and retailers.	✓	✓	✓
16	Shall be able to ask questions from retailers.	✓	✓	✓
17	Should be able to receive alerts on recommended products and offers.	✓	✓	✓
18	Should be able to view large size images of products (zoom).	✓	✓	✓
19	Should display location of shop and retailer contact details for a particular product.	✓	✓	✓

TABLE 2.5.1 FUNCTIONAL REQUIREMENT'S VS BSO'S

2.5.2 Non-functional requirements vs BSO's

ID	Description	BSO1	BSO2	BSO3
1	Shall be able to present user friendly interfaces.	✓		
2	Shall be able to handle around thousand users concurrently.	✓	✓	✓
3	Shall be able to up and running 24 hours and available to customers at all times.	✓	✓	✓
4	Shall be able to access via smartphones not depending on their O/S.	✓		
5	Shall be able to provide the safety and security, system	✓	✓	✓
6	Shall make security in online payments.	✓	✓	✓
7	Should be able to secure information from external parties	✓	✓	✓
8	Should be responsive immediately to customer requests	✓	✓	✓
9	Should be able to real-time update changes immediately.	✓	✓	✓

TABLE 2.5.2 NON-FUNCTIONAL REQUIREMENT'S VS BSO'S

2.6 Summary of the chapter

With the above analysis, it is clear that BSO1 all the functional and as well as non-functional requirements. With the cost analysis section, the three BSOs will provide a range of identical costs but there is a small cost differences i.e. BSO2 being the highest and BSO3 the next and the least will be BSO1. So it is clear that the best option which was BSO 1 is selected as the further designing and development process.

3 System Design

This chapter includes the proposed system's design. It describes the proposed system and its functionalities in more details by utilizing use case diagrams and activity diagrams. By the end of the chapter, database design will show the tables which will be used in the system. and finally, the chapter will conclude with the Graphical User Interfaces.

Outline of the chapter

3.1 Use case

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

3.1.1 Overall use case for the current system.

Diagram shows the use case diagram for proposed system this include all activities related to both customers and retailers.

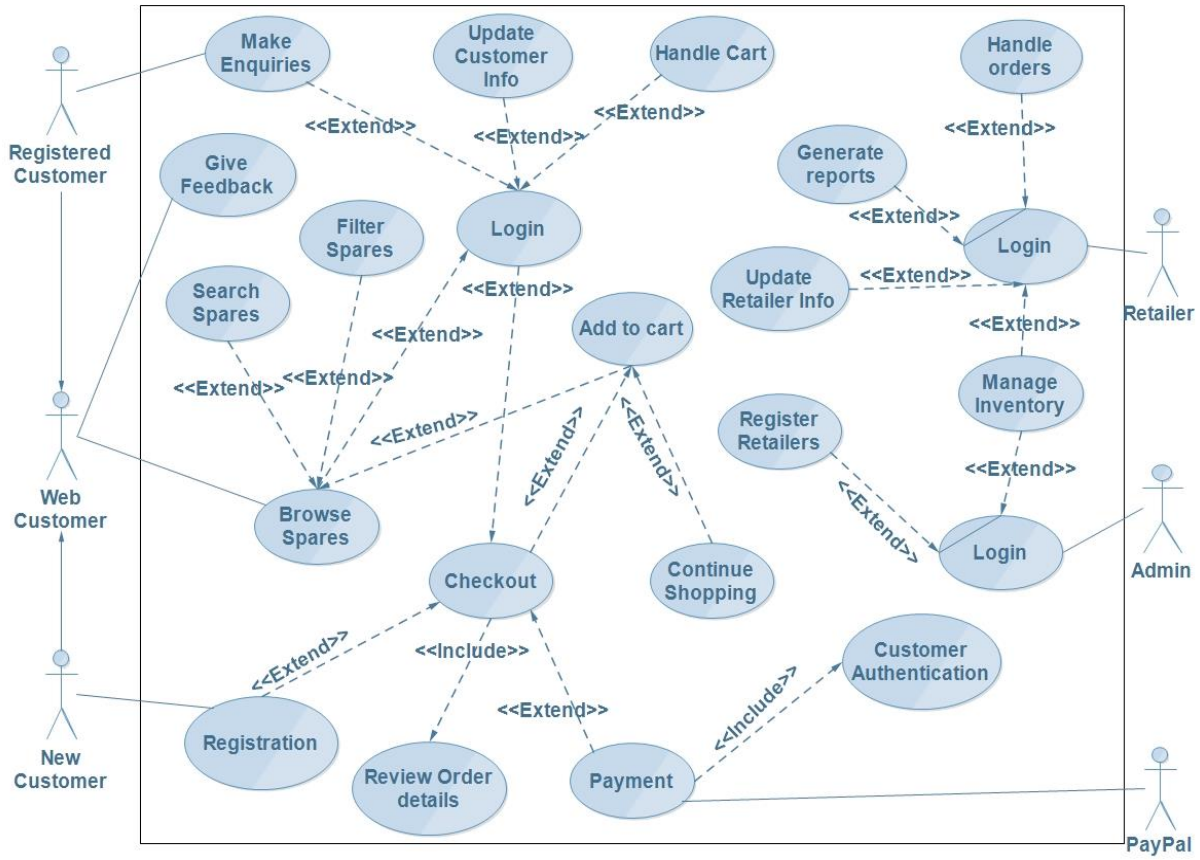


FIGURE 3.1.1 OVERALL USE CASE FOR THE CURRENT SYSTEM

3.1.2 Granular level use case diagram for 'Manage Inventory' use case

Above diagram shows the granular level use case diagram for proposed system this include all activities related to both customers and retailers.

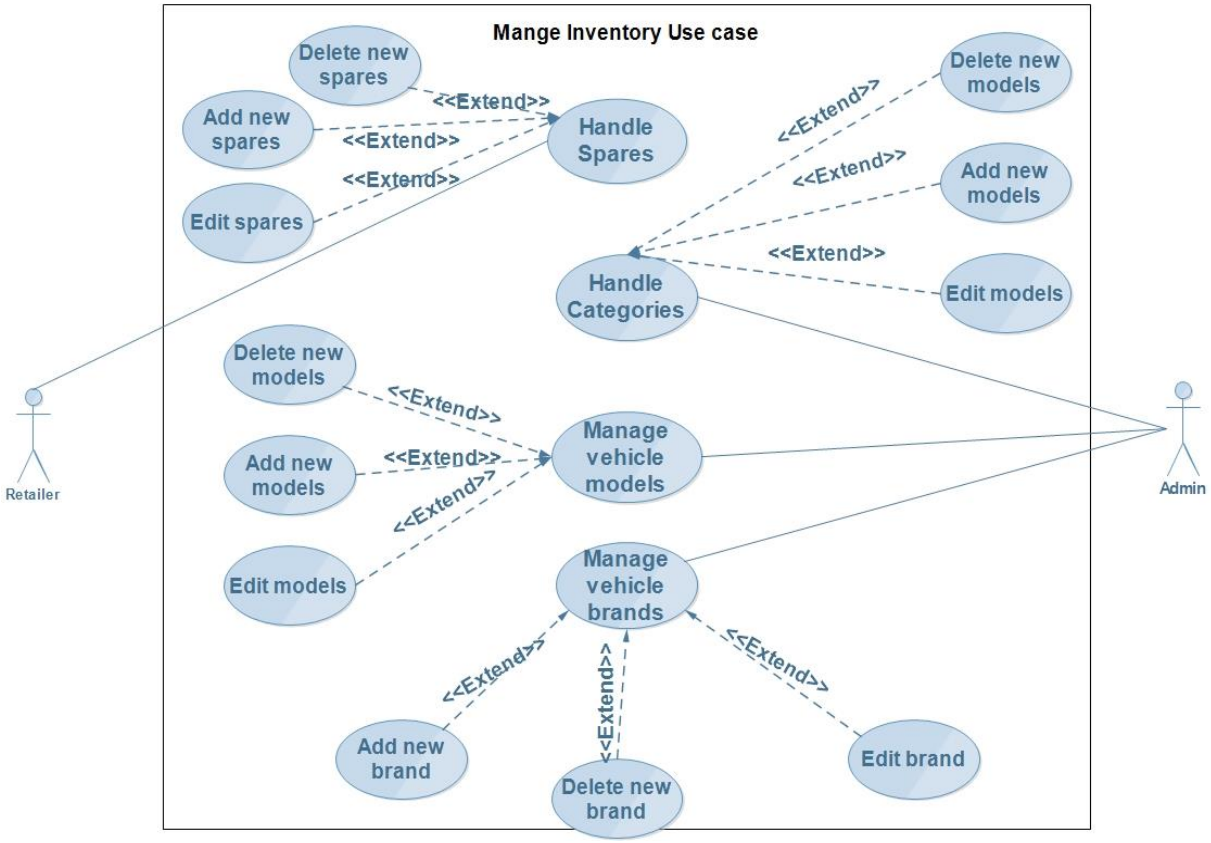


FIGURE 3.1.2 GRANULAR USE CASE DIAGRAM FOR MANAGE INVENTORY

3.1.3 Use Case Descriptions

3.1.3.1 Login

'Login' use case in use case diagram is further illustrated in the use case description below.

Use Case	Login
Use Case ID	UC -01
Actors	Registered Customer, Retailer
Description	This use case describes how a user logs into the website.
Basic Flow	1. The system requests that the actor to enter his/her name and password. 2. The actor enters his/her name and password. 3. The system validates the entered name and password and logs the actor into the system.
Alternative Flows	If in the *Basic Flow* the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the *Basic Flow* or cancel the login, at which point the use case ends.
Pre-Conditions	He must register to the website
Post- Conditions	Successfully log into the system
Exceptions	-

TABLE 0.1 'LOGIN' USE CASE DESCRIPTION

3.1.3.2 Registration

'Registration' use case in use case diagram is further illustrated in the use case description below.

Use Case	Registration
Use Case ID	UC -02
Actors	New Customer
Description	This use case describes how a customer registers into the website.
Basic Flow	1. The system requests that the actor to enter his/her credentials. 2. The actor fills the registration form and submit it. 3. The system validates the entered details and register a new user to the system.
Alternative Flows	If in the *Basic Flow* the user enters an invalid information the system displays an error message. The actor can choose to either return to the beginning of the *Basic Flow* or cancel the registration, at which point the use case ends.
Pre-Conditions	-
Post- Conditions	Successfully register into the system
Exceptions	-.

TABLE 0.2 REGISTRATION USE CASE DESCRIPTION

3.1.3.3 Update Customer Info.

'Update Customer Info' use case in use case diagram is further illustrated in the use case description below.

Use Case	Update Customer Info.
Use Case ID	UC -03
Actors	Registered Customer
Description	This use case describes how a user can change his information.
Basic Flow	1. Actor click my account button. 2. The actor edits the information and save. 3. The system updates the user details.
Alternative Flows	If in the *Basic Flow* the user enters an invalid information the system while editing displays an error message. The actor can choose to either return to the beginning of the *Basic Flow* or cancel the editing information, at which point no any update of previous data doesn't take place and the use case ends.
Pre-Conditions	-
Post- Conditions	Successfully updates account.
Exceptions	-.

TABLE 0.3 'UPDATE CUSTOMER INFO.' USE CASE DESCRIPTION

3.1.3.4 Search Spares

'Search Spares' use case in use case diagram is further illustrated in the use case description below.

Use Case	Search Spares
Use Case ID	UC -04
Actors	Web Customer
Description	This use case describes how a Customer searches the spare items in the website.
Basic Flow	1. The actor enters the details that he need to search in the search textbox. 2. The system will search the phrases in the textbox to find the matching items that are in the database. 3.System will then load the items in to the spare browsing page.
Alternative Flows	If in the *Basic Flow* if there are no results found the actor can choose to either return to the beginning of the *Basic Flow* which is to search again or cancel the searching, at which point the use case ends.
Pre-Conditions	-
Post- Conditions	Customer get the intended results or if not he can search again.
Exceptions	-.

TABLE 0.4 'SEARCH SPARES' USE CASE DESCRIPTION

3.1.3.5 Filter Spares

'Filter Spares' use case in use case diagram is further illustrated in the use case description below.

Use Case	Filter Spares
Use Case ID	UC -05
Actors	Web Customer
Description	This use case describes how a Customer broaden his search by setting different set of filters. This will categorize the search.
Basic Flow	1. The actor go to the advance settings page and apply set of filters he intends and then start to search. 2. The system will retrieve the spares from the database depending on the filters and display results.
Alternative Flows	If in the *Basic Flow* if there are no results found the actor can choose to either return to the beginning of the *Basic Flow* or cancel the filtering, at which point the use case ends.
Pre-Conditions	-
Post- Conditions	Customer get the intended results or if not he can apply filtering again.
Exceptions	-.

TABLE 0.5 'FILTER SPARES' USE CASE DESCRIPTION

3.1.3.6 Add to Cart

'Add to Cart' use case in use case diagram is further illustrated in the use case description below.

Use Case	UC -06
Use Case ID	Add to cart
Actors	Web Customer
Description	This use case describes how a user can add item into the shopping cart.
Pre-condition	The item should be selected.
Basic Course	1. The customer navigates to the item he/she wishes to buy. Then select quantity and clicks on the "Add to Cart" button. 2. The system displays the Cart Screen with the all the old items and the newly added item. The subtotal field displays the total cost of the shopping cart. 3. The customer repeats steps 1 and 2 for all the items he/she wants to add to the cart.
Alternative Course	1. If the user enters a non-positive or non-integer quantity the system displays an appropriate error message. 2. 'Refresh cart' feature is available for resetting the cart.
Exceptional course	If quantity fields are not filled system display error message.
Post- Conditions	Items are added to the cart.

TABLE 0.6 'ADD TO CART' USE CASE DESCRIPTION

3.1.3.7 Handle Cart

'Handle Cart' use case in use case diagram is illustrated in the use case description below.

Use Case	Handle Cart
Use Case ID	UC -07
Actors	Web Customer
Description	This use case describes how an actor can modify items in the shopping cart.
Basic Flow	1. The user navigates to the item he wishes to add to the cart and clicks on the "Add to Cart" button. 2. The system displays the Cart Screen with the all the old items and the newly added item. The total field displays the total cost of the shopping cart. 3. The user repeats steps 1 and 2 for all the items he wants to add to the cart. 4. If user need to modify the quantity then he enters the needed quantity in the quantity box of the cart. 5. The system updates the new quantity and displays the modified line item totals and sub-total to the user. 6. The user clicks the "Remove" button to remove any of the items in the cart. 7. The system deletes the item from the cart and adjusts the sub-total accordingly.
Alternative Flows	1. User proceeds to modifying cart without logging in. 2. If the user enters a non-positive or non-integer quantity the system displays an appropriate error message.
Pre-Conditions	-
Post- Conditions	-
Exceptions	-.

TABLE 0.7 'HANDLE CART' USE CASE DESCRIPTION

3.1.3.8 Checkout

'Checkout' use case in use case diagram is further illustrated in the use case description below.

Use Case	Checkout
Use Case ID	UC -08
Actors	Web Customer
Description	This use case describes how a Customer request to checkout.
Basic Flow	1. The actor click on proceed to check out in the shopping cart 2. The system will redirect to checkout page and the page will show the summary of address of shipping and the items. 3. If the details are correct customer can proceed to payment option else he can edit the details.
Alternative Flows	In the step 2 of Basic Flow If the actor has not logged in to the system he can either choose to continue as a guest, login or register.
Pre-Conditions	-
Post- Conditions	Actor can successfully continue to PayPal page
Exceptions	-.

TABLE 0.8 'CHECKOUT' USE CASE DESCRIPTION

3.1.3.9 Payment

'Payment' use case in use case diagram is further illustrated in the use case description below.

Use Case	Payment
Use Case ID	UC -09
Actors	Web Customer
Description	This use case describes how the User of the system can make payments for the purchases.
Basic Flow	1. User clicks on "Proceed to payment". 2. System displays the "PayPal" login page. 3. User logs on to PayPal. 4. System displays the review of the order. 5. User click on 'Confirm'.
Alternative Flows	In the step 2 of Basic Flow If the actor has not logged in to the system he can either choose to continue as a guest, login or register.
Pre-Conditions	-
Post- Conditions	- User is able to pay online successfully. - An Order Id is generated by the system.
Exceptions	Entering of wrong PayPal password/username.

TABLE 0.9 'PAYMENT' USE CASE DESCRIPTION

3.1.3.10 Give Feedback

'Give Feedback' use case in use case diagram is further illustrated in the use case description below.

Use Case	Give Feedback
Use Case ID	UC -10
Actors	Web Customer
Description	This use case describes how the customer can rate, add comments or add complains for the purchases.
Basic Flow	1. User navigate to feedback page where he finds all the orders that belongs to him with dates. 2. Then he/she will click on a relevant order where it will load all the items in that order. 3. User select an item and comment/rate/complain on a product. 4. Click on save.
Alternative Flows	In the step 2 of *Basic Flow* If the actor enters a wrong ID then system will pop up a message and allow them to retry.
Pre-Conditions	User must have an Order ID (A purchase should have been made)
Post- Conditions	-
Exceptions	-

TABLE 0.10 'FEEDBACK' USE CASE DESCRIPTION

3.1.3.11 Handle Orders

'Handle Orders' use case in use case diagram is further illustrated in the use case description below.

Use Case ID	UC -11
Use Case	Handle orders
Actors	Retailer
Description	This use case describes how the retailer get the information about a corresponding order placed by a customer. This includes many functionalities related to orders such as changing delivery status etc.
Pre-Conditions	A retailer should login to the system.
Basic Course	1. View all the orders with relevant order details corresponding. 2. Choose a particular order. 3. If administrator wants to deliver this order handover it to courier service & update order status as "Pending Order". 4. If administrator receive delivery success receipt, update order status as "Delivered order".
Alternative Course	-
Exceptions Course	-
Post- Conditions	Changes save in the system.

TABLE 0.11 'HANDLE ORDERS' USE CASE DESCRIPTION

3.1.3.12 Handle Spares

'Handle Spares' use case in granular use case diagram 'Manage Inventory' is further illustrated in the use case description below.

Use Case ID	UC -12
Use Case	Handle Spares
Actors	Retailer
Description	This use case describes how the retailer get the ability to update the details including names, prices quantities and descriptions about a corresponding spare part. It also features the retailer to delete a spare and adding new product to the database so that it will appear in the website.
Pre-Conditions	A retailer should login to the system.
Basic Course	1. Retailer chooses the spare tab. 2. System will load the spares that belong to that particular retailer. 3. He can find the necessary spare part that needed to be updated by searching 4. Then Choose add/delete/update buttons as needed. 5. Edit the relevant information that belong to particular spare and save it in the system.
Alternative Course	-
Exceptions Course	-
Post- Conditions	Changes save in the system.

TABLE 0.12 'HANDLE SPARES' USE CASE DESCRIPTION

3.1.3.13 Handle Categories

'Handle Categories' use case in granular use case diagram 'Manage Inventory' is further illustrated in the use case description below.

Use Case ID	UC -13
Use Case	Handle Categories
Actors	Admin
Description	This use case describes how the admin get the ability to update the details including names, sub categories about a corresponding category.
Pre-Conditions	An admin should login to the system.
Basic Course	1. Admin chooses the categories tab. 2. System will load the categories. 4. Then Choose add/delete/update buttons as needed. 5. Edit the relevant information that belong to particular category and save it in the system.
Alternative Course	-
Exceptions Course	-
Post- Conditions	Changes save in the system.

TABLE 0.13 'HANDLE CATEGORIES' USE CASE DESCRIPTION

3.1.3.14 Handle Vehicle Models

'Handle Vehicle Models' use case in granular use case diagram 'Manage Inventory' is further illustrated in the use case description below.

Use Case ID	UC -14
Use Case	Handle Vehicle Models
Actors	Admin
Description	This use case describes how the admin get the ability to update the details including Transmission type, Year of Manufacture, Model Name and Fuel Type about a corresponding vehicle model
Pre-Conditions	An admin should login to the system.
Basic Course	1. Admin chooses the vehicle models tab. 2. System will load the vehicle model. 3.Then Choose add/delete/update buttons as needed. 4. Edit the relevant information that belong to particular vehicle model and save it in the system.
Alternative Course	-
Exceptions Course	-
Post- Conditions	Changes save in the system.

TABLE 0.14 'HANDLE VEHICLE MODELS' USE CASE DESCRIPTION

3.1.3.15 Handle Vehicle Brands

'Handle Vehicle Brands' use case in granular use case diagram 'Manage Inventory' is further illustrated in the use case description below.

Use Case ID	UC -15
Use Case	Handle Vehicle Brands
Actors	Admin
Description	This use case describes how the admin get the ability to update the details including brands names, country made about a corresponding vehicle brands.
Pre-Conditions	An admin should login to the system.
Basic Course	1. Admin chooses the vehicle brand tab. 2. System will load the vehicle brand. 3. Then Choose add/delete/update buttons as needed. 5. Edit the relevant information that belong to particular vehicle brand and save it in the system.
Alternative Course	-
Exceptions Course	-
Post- Conditions	Changes save in the system.

TABLE 0.15 'HANDLE VEHICLE BRANDS' USE CASE DESCRIPTION

3.1.3.16 Register Retailer

'Register Retailer' use case in use case diagram is further illustrated in the use case description below.

Use Case	Register Retailer
Use Case ID	UC -16
Actors	Admin
Description	This use case describes how a retailer registers into the website.
Basic Flow	1. Retailer ask the admin that they need to register to the system. 2. Admin log in to the site. 3. Admin enter the details into the forms and registers to the system.
Alternative Flows	If in the *Basic Flow* the admin enters an invalid information the system displays an error message. The actor can choose to either return to the beginning of the *Basic Flow* or cancel the registration, at which point the use case ends.
Pre-Conditions	Admin should log in to the system.
Post- Conditions	Successfully register into the system.
Exceptions	-

TABLE 0.16 'REGISTER RETAILER' USE CASE DESCRIPTION

3.2 Activity Diagrams

Activity diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another.

3.2.1 Login Activity.

Below activity diagram is the further illustration of actions occurred in use Case 'Login'.

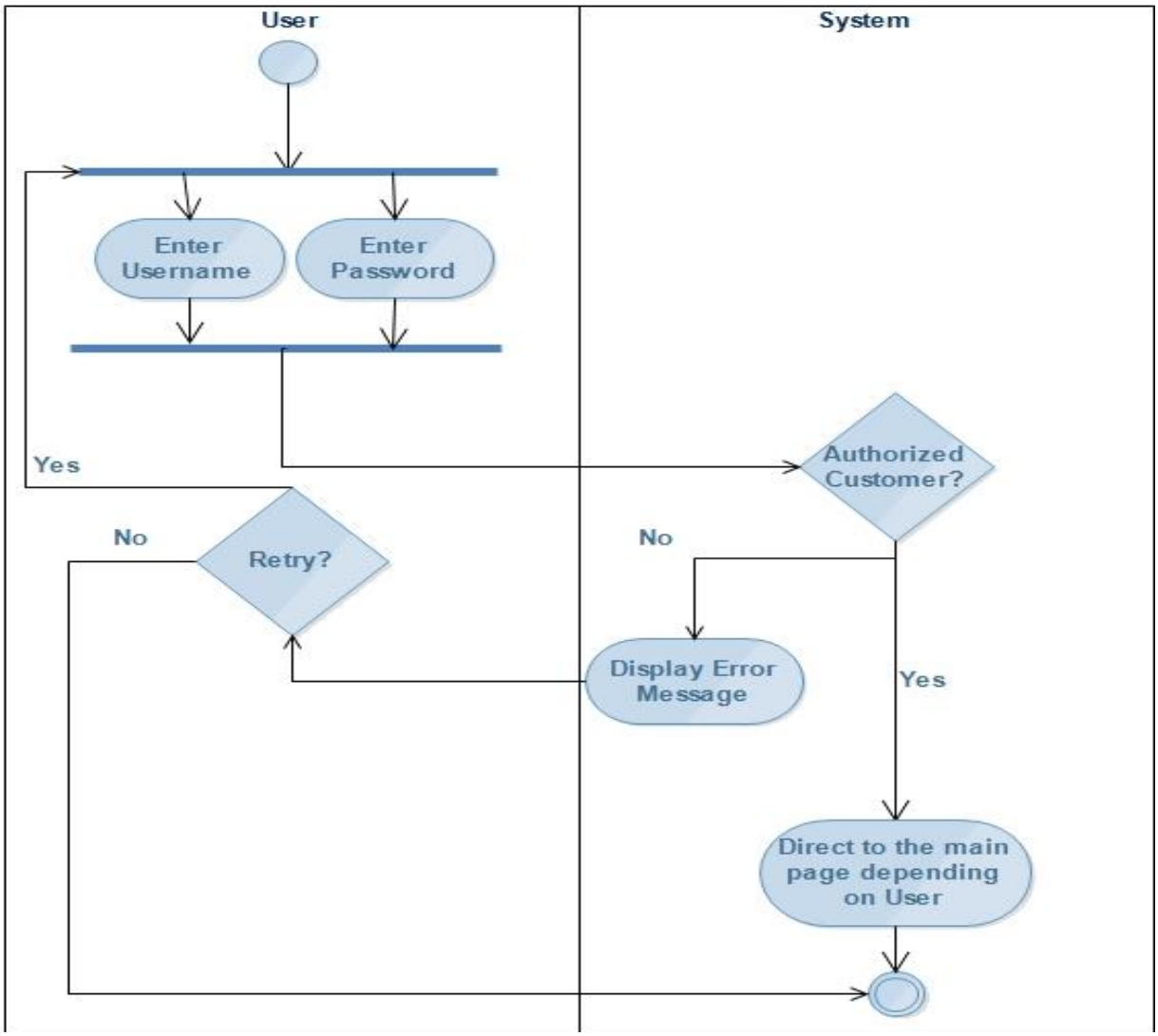


FIGURE 3.2.1 ACTIVITY DIAGRAM FOR LOGIN

3.2.2 Registration Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Registration’.

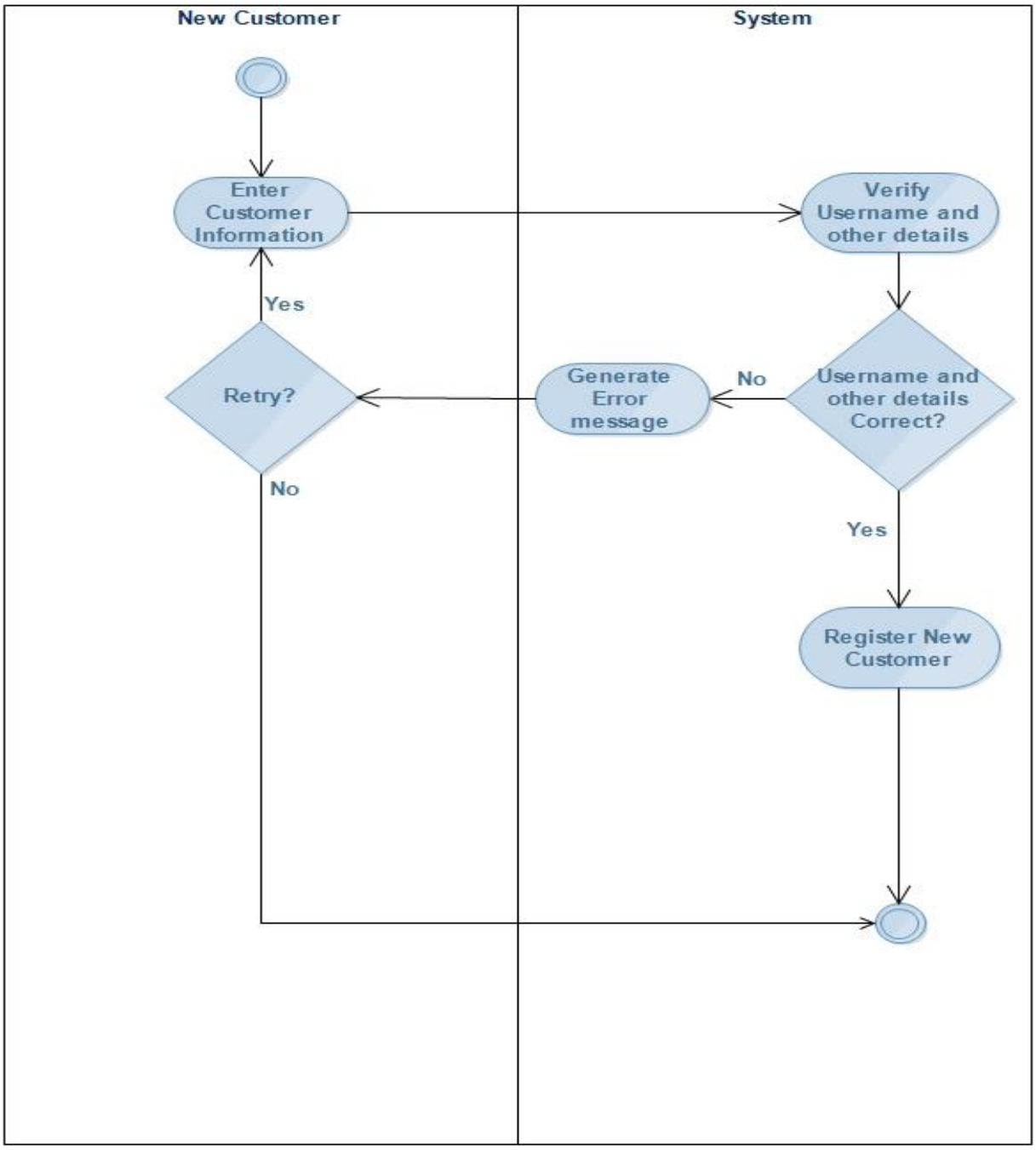


FIGURE 3.2.2 ACTIVITY DIAGRAM FOR 'REGISTRATION'

3.2.3 Browse Spares Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Browse Spares’.

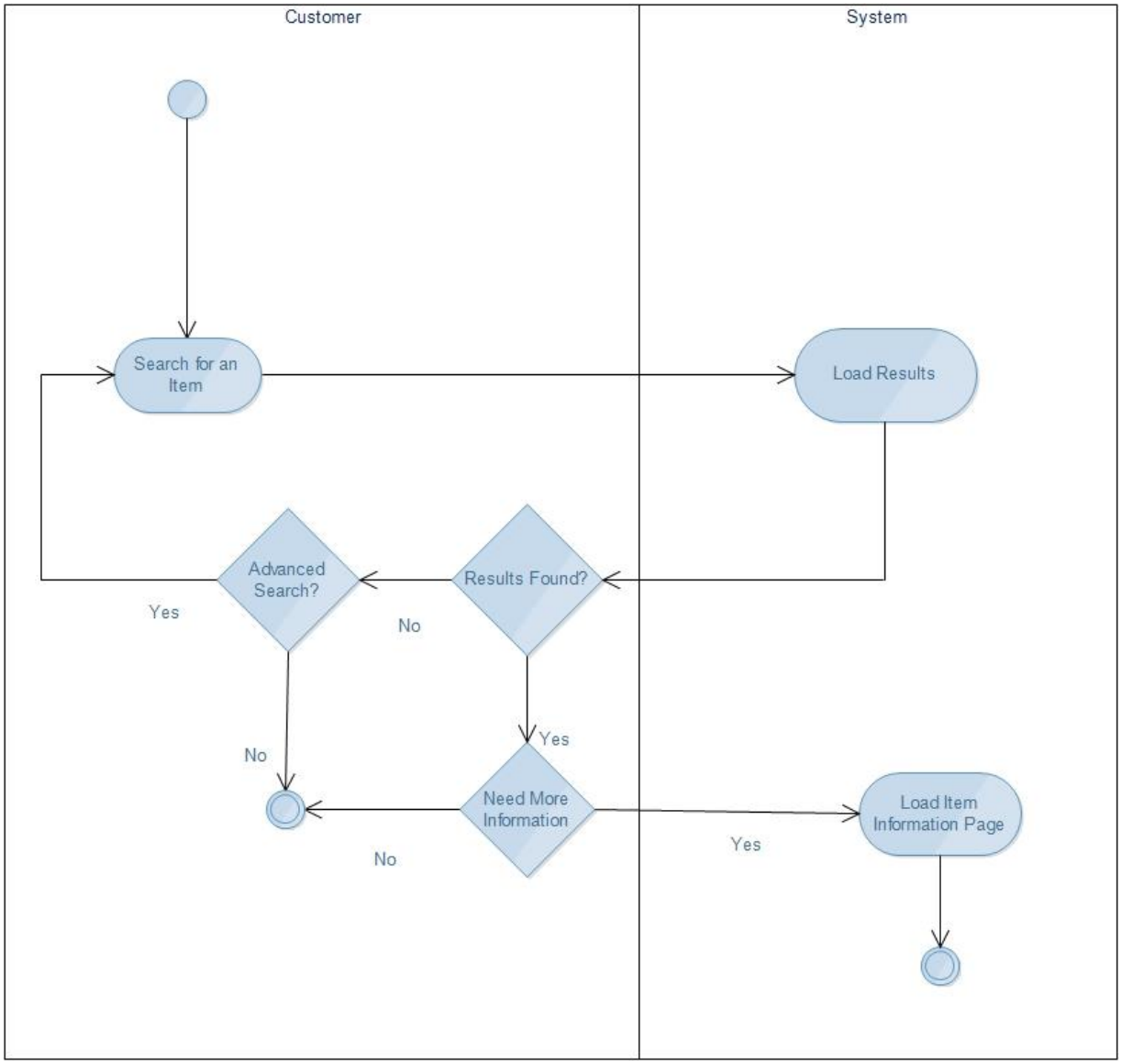


FIGURE 3.2.3 ACTIVITY DIAGRAM ‘BROWS SPARES’

3.2.4 Handle Cart Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Handle Cart’.

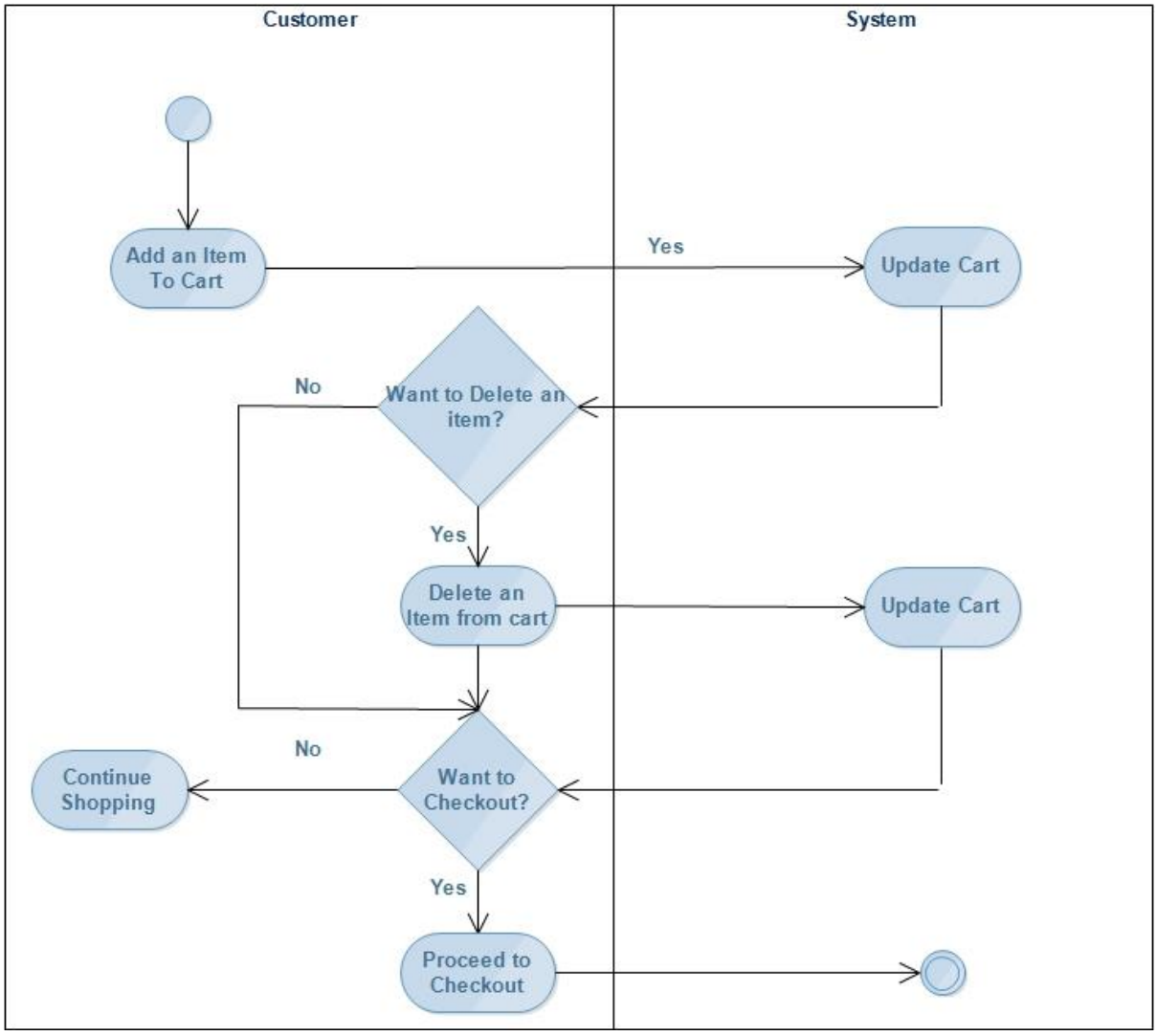


FIGURE 3.2.4 ACTIVITY DIAGRAM ‘HANDLE CART’

3.2.5 Add to Cart Activity

Below activity diagram is the further illustration of cause of actions occurred in use case 'Add to Cart'.

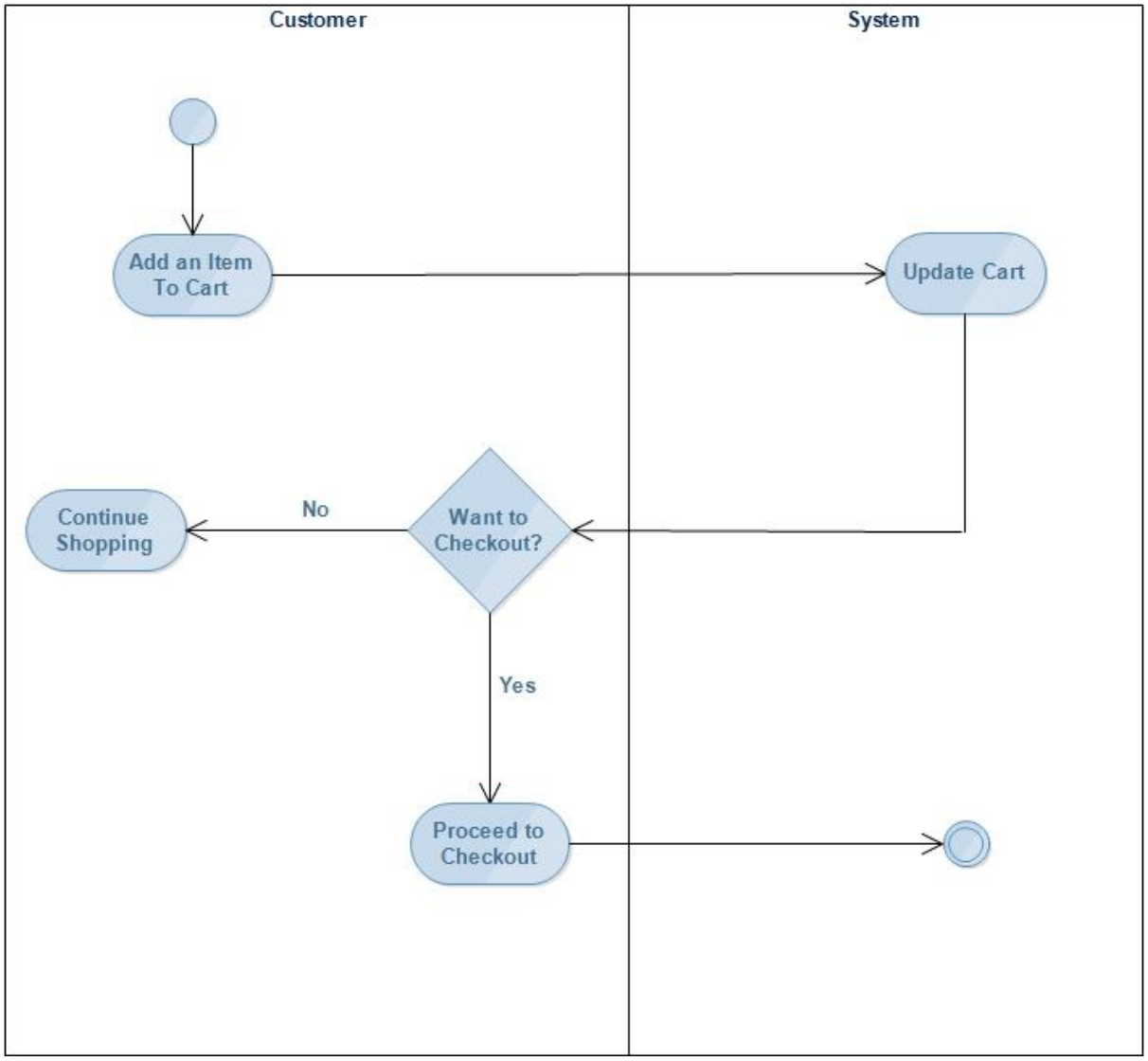


FIGURE 3.2.5 ACTIVITY DIAGRAM 'ADD TO CART'

3.2.6 Checkout Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Checkout’.

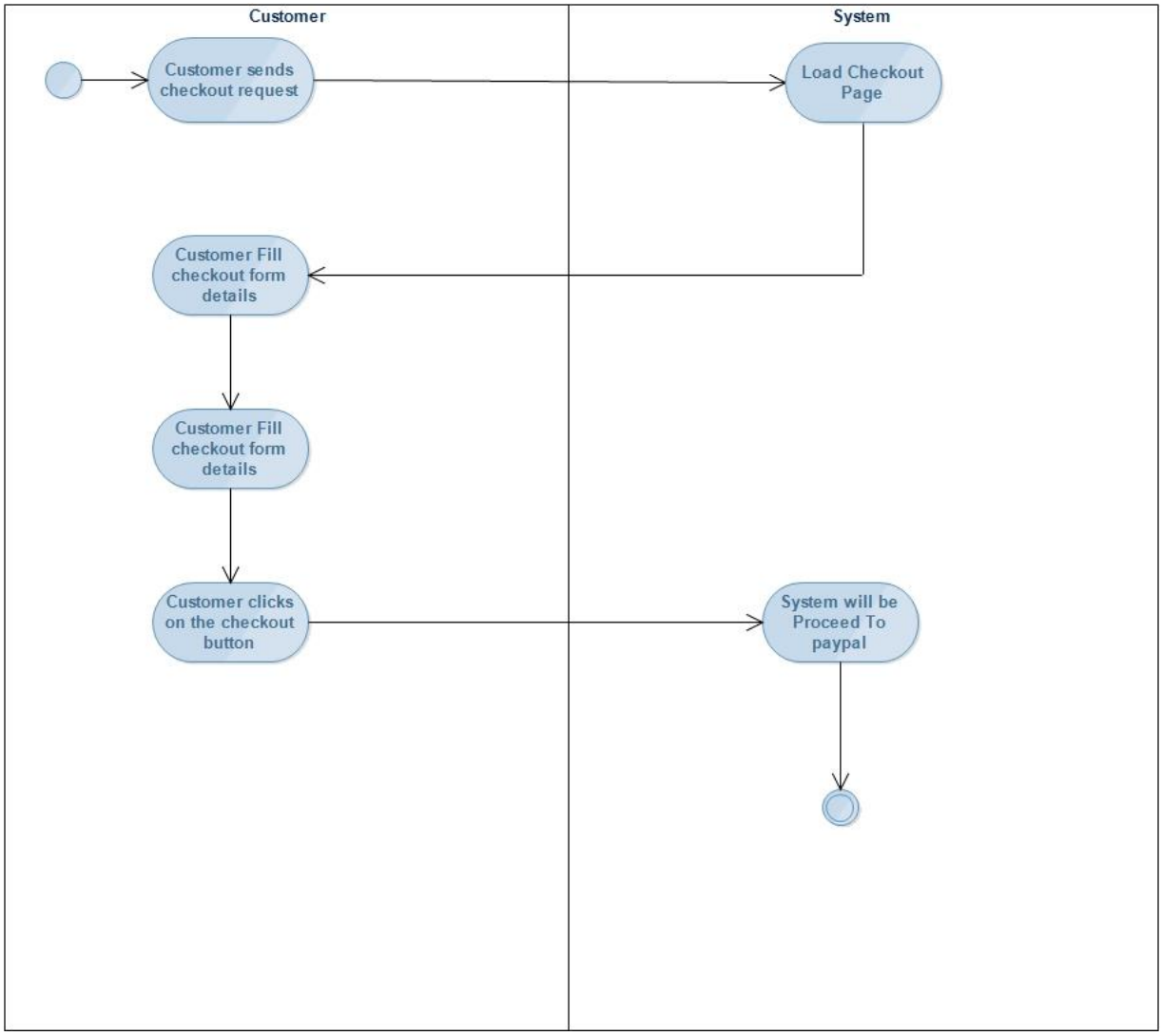


FIGURE 3.2.6 ACTIVITY DIAGRAM ‘CHECKOUT’

3.2.7 Payment Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Payment’.

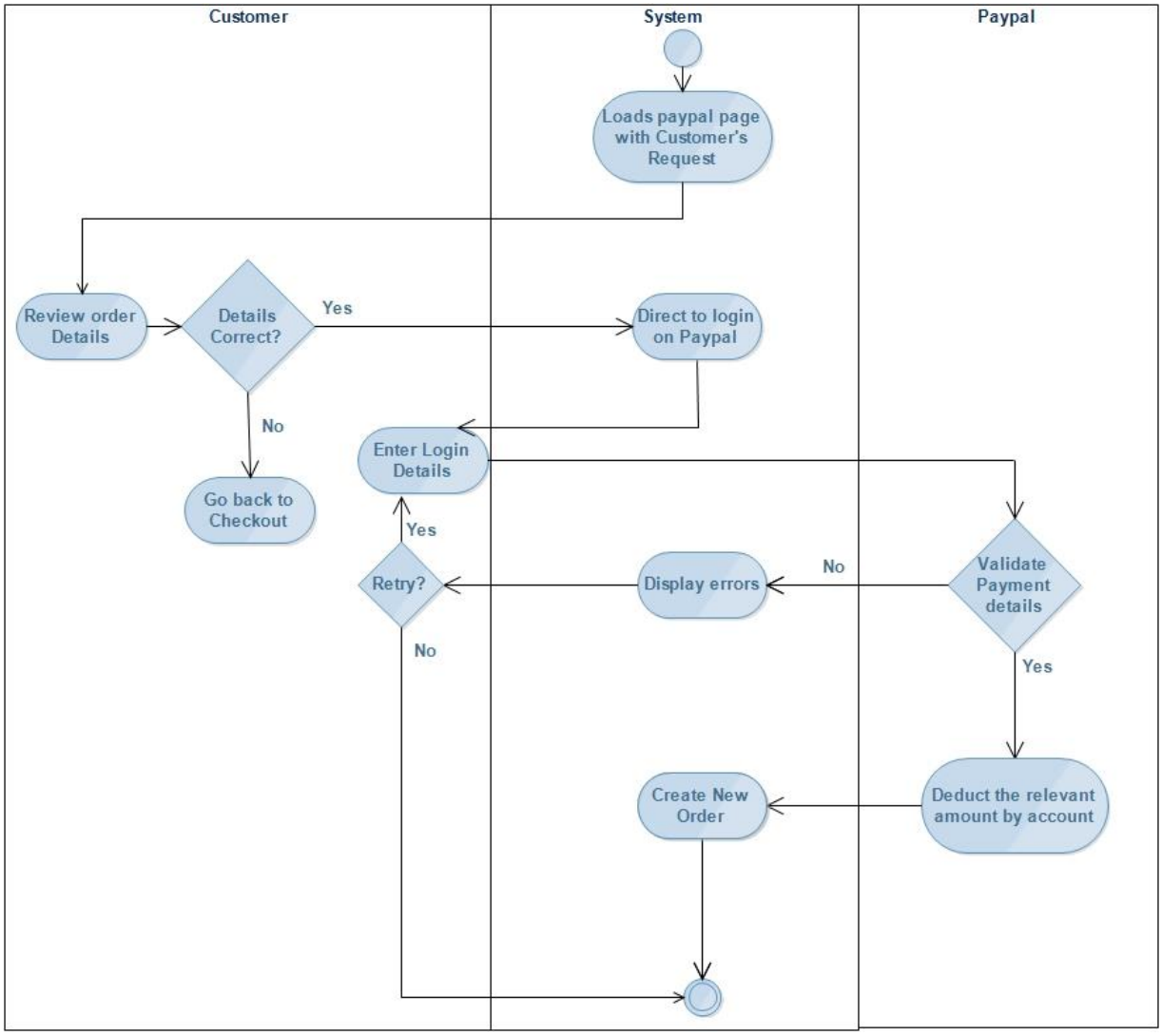


FIGURE 3.2.7 ACTIVITY DIAGRAM ‘PAYMENT’

3.2.8 Enquiry Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Make Enquiries’.

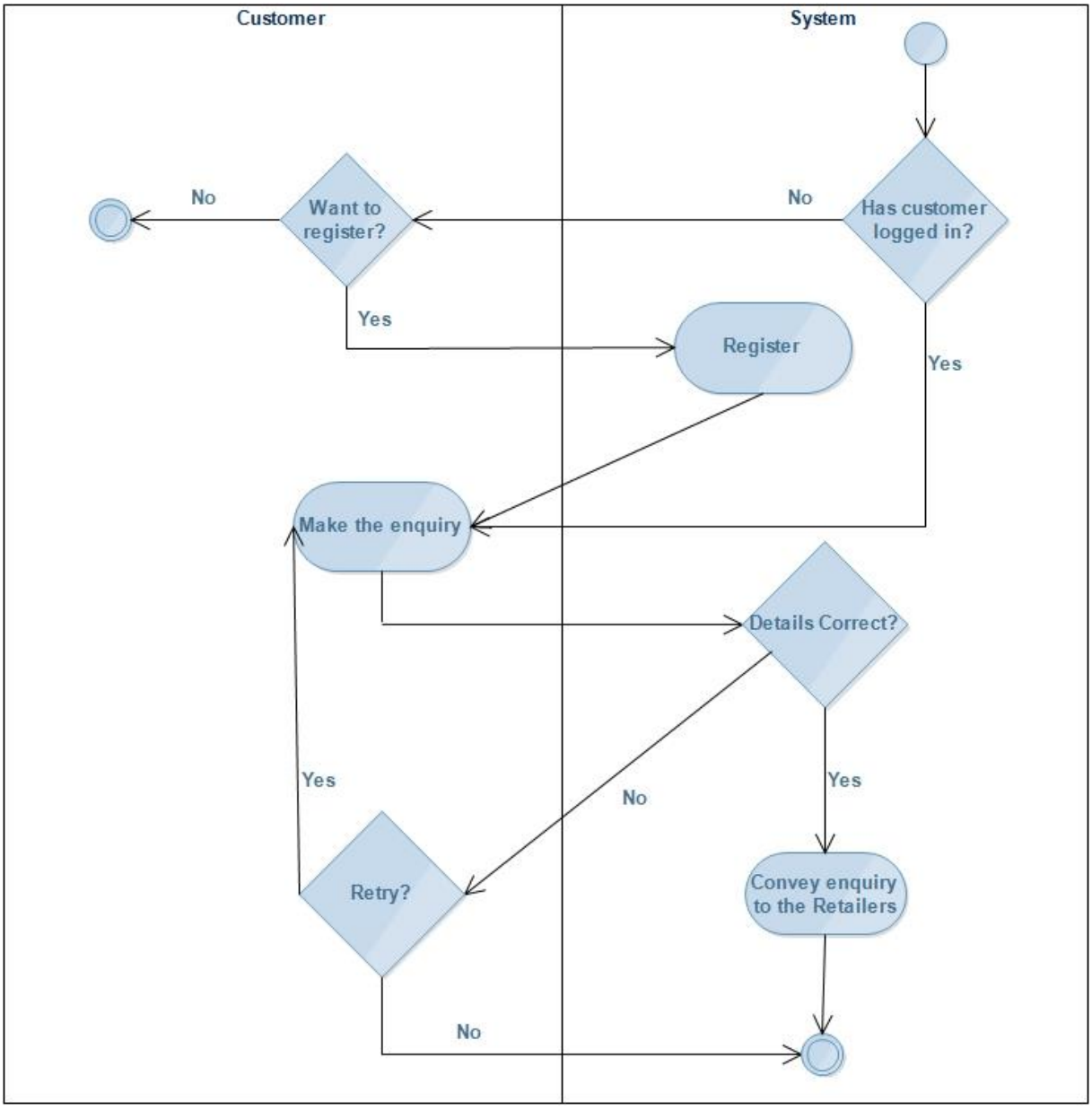


FIGURE 3.2.8 ACTIVITY DIAGRAM ‘MAKE ENQUIRIES’

3.2.9 Give Feedback Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Give Feedback.

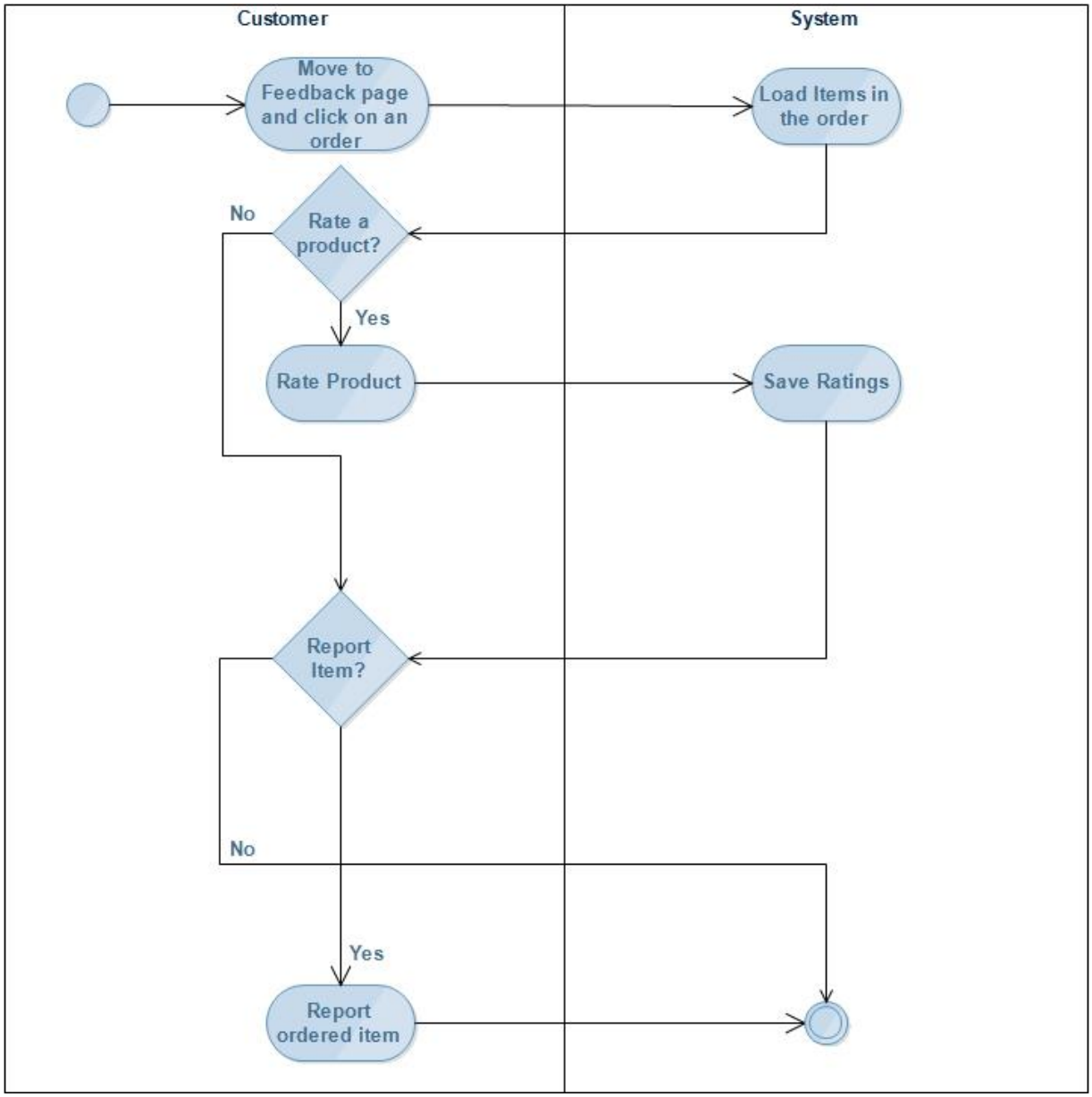


FIGURE 3.2.9 ACTIVITY DIAGRAM ‘GIVE FEEDBACK’

3.2.10 Generate report Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Generate Report’ activity which is done by retailer.

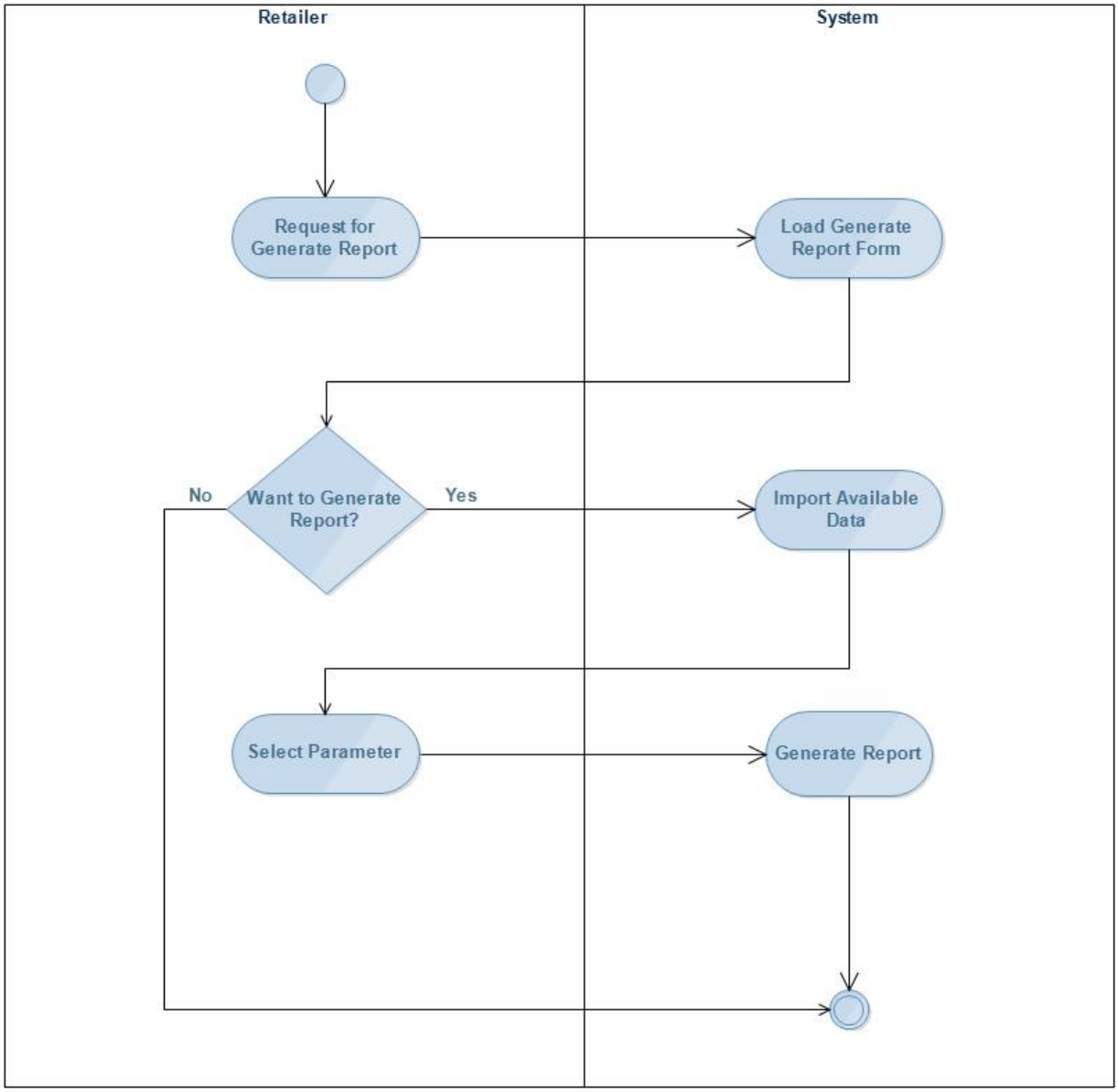


FIGURE 3.2.10 ACTIVITY DIAGRAM ‘GENERATE REPORT’

3.2.11 Handle Brands Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Handle Brands’ activity which is done by Admin.

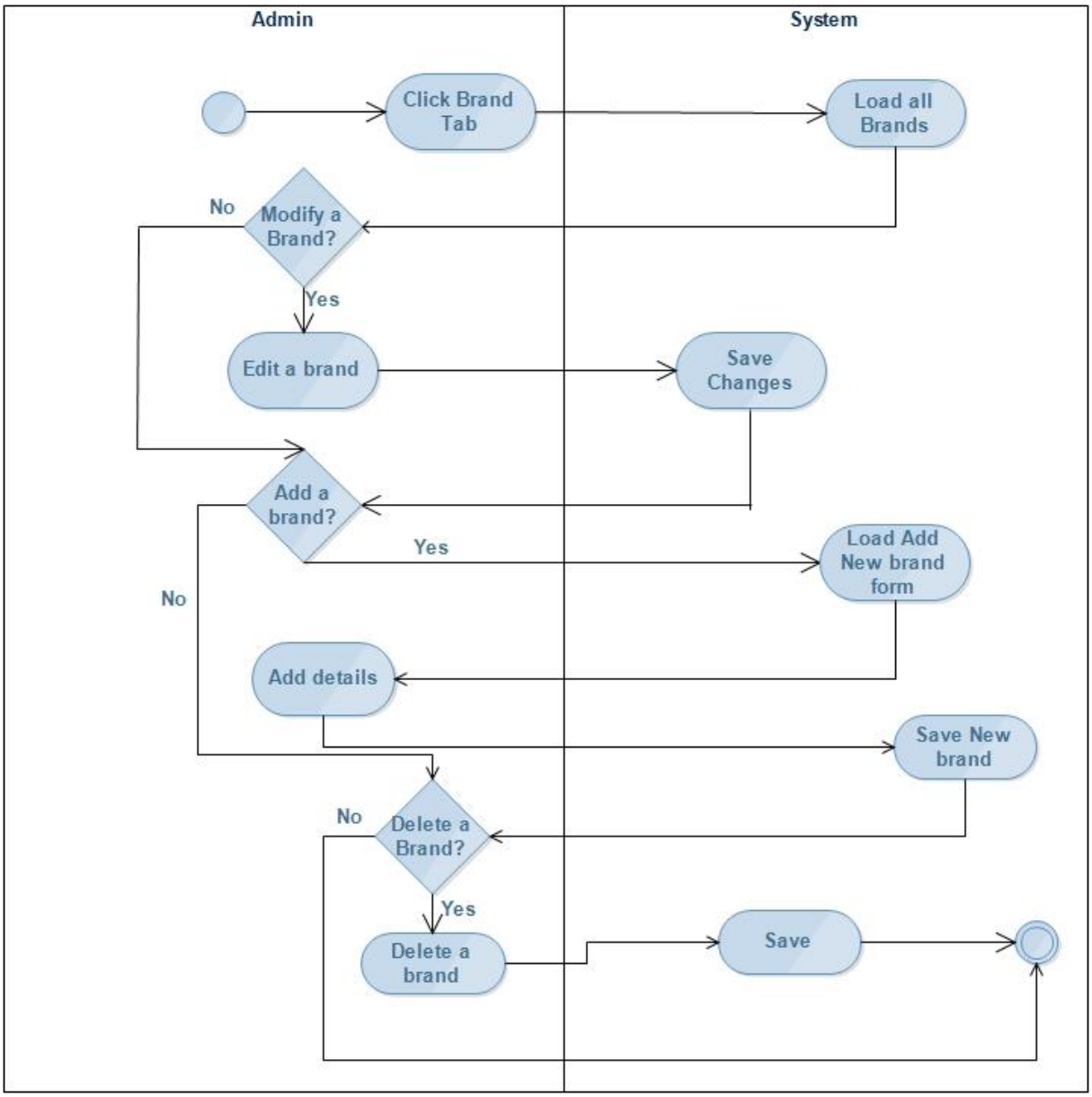


FIGURE 3.2.11 ACTIVITY DIAGRAM ‘HANDLE BRANDS’

3.2.12 Handle Categories Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Handle Categories’ which is done by Admin.

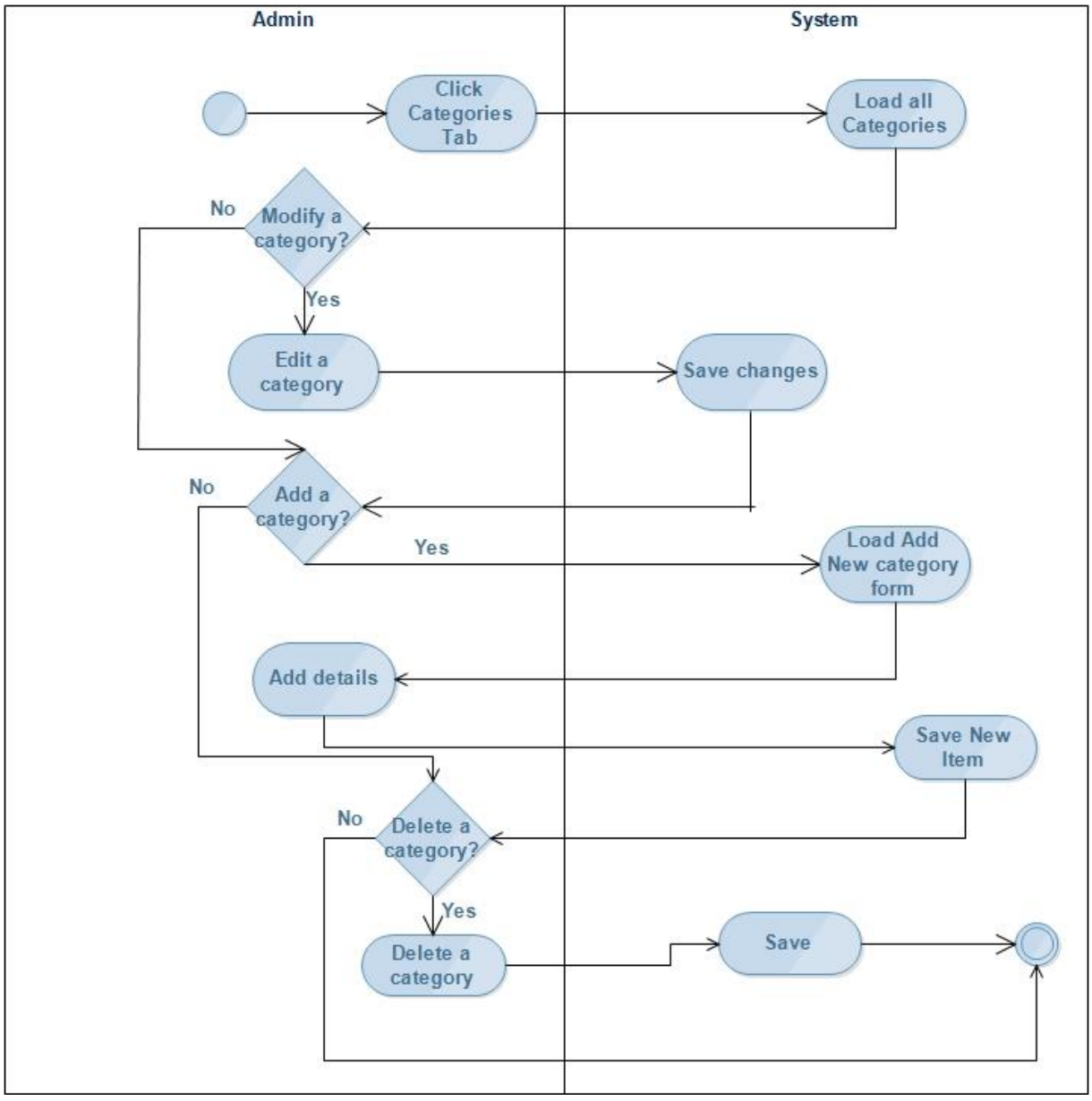


FIGURE 3.2.12 ACTIVITY DIAGRAM ‘HANDLE CATEGORIES’

3.2.13 Handle Models Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Handle Models’ which is done by Admin.

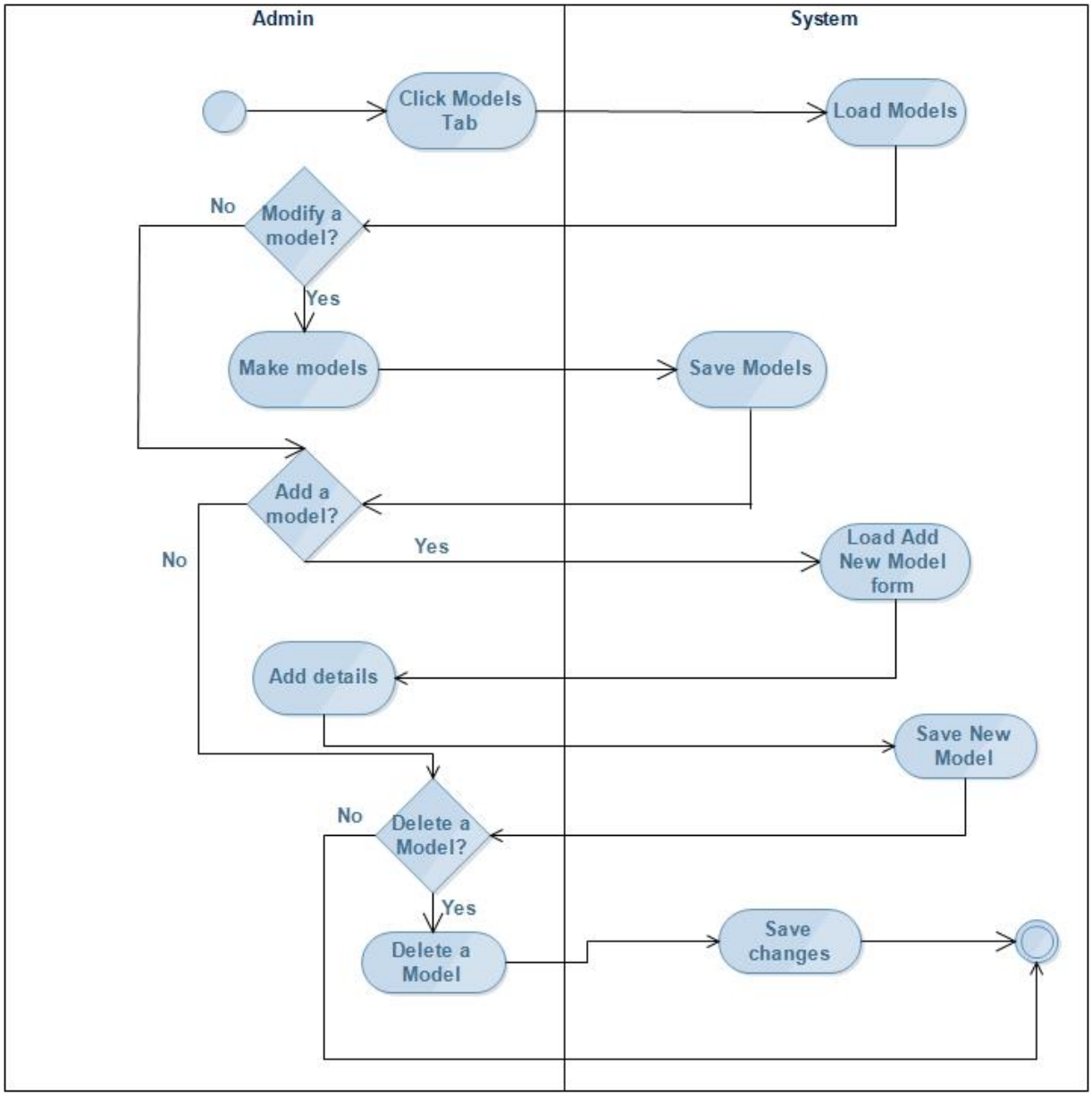


FIGURE 3.2.13 ACTIVITY DIAGRAM ‘HANDLE MODELS’

3.2.14 Handle Spares Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Handle Spares’ which is done by Retailer.

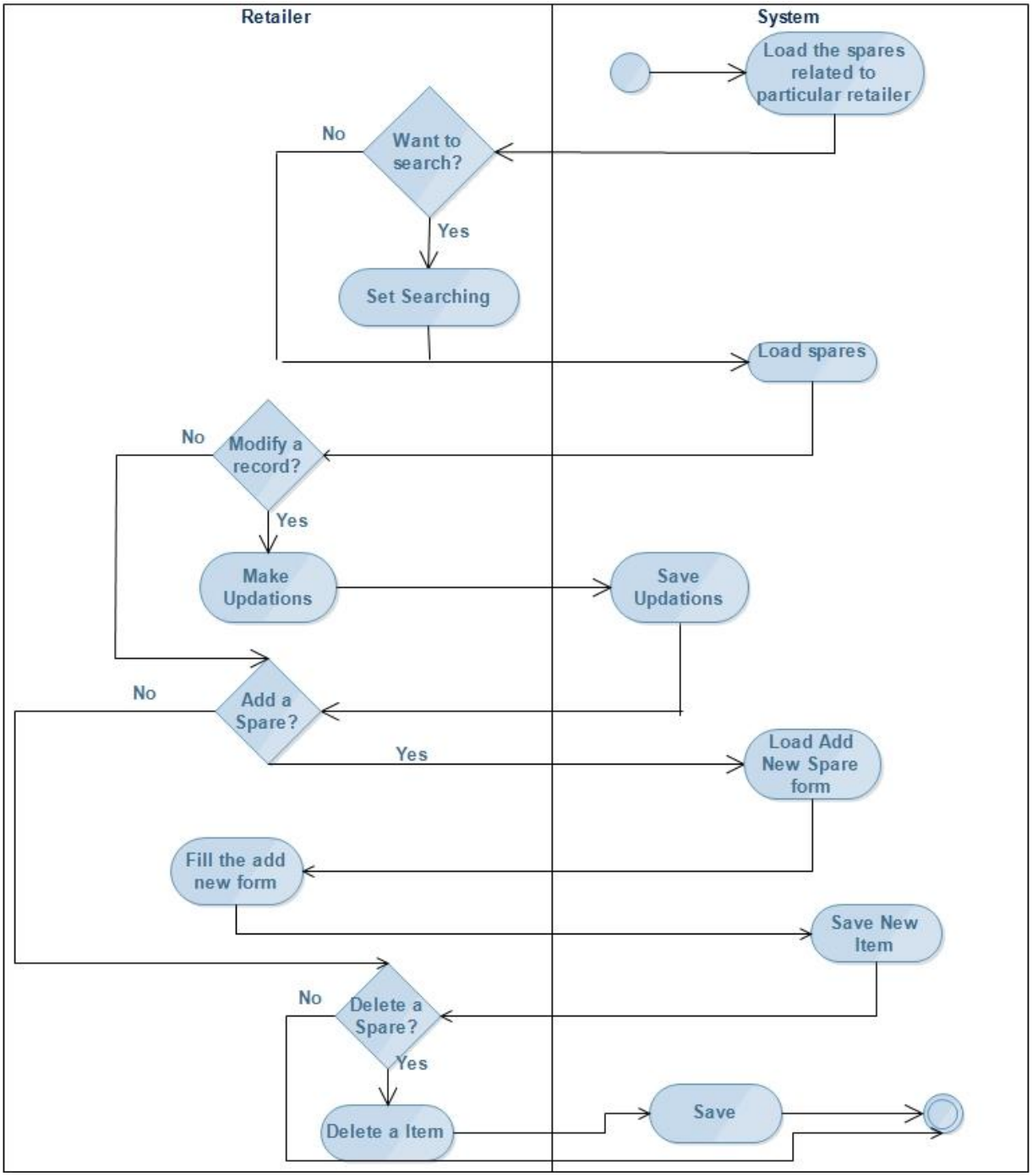


FIGURE 3.2.14 ACTIVITY DIAGRAM ‘HANDLE SPARES’

3.2.15 Register Retailer Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Register Retailer’ which is done by Retailer.

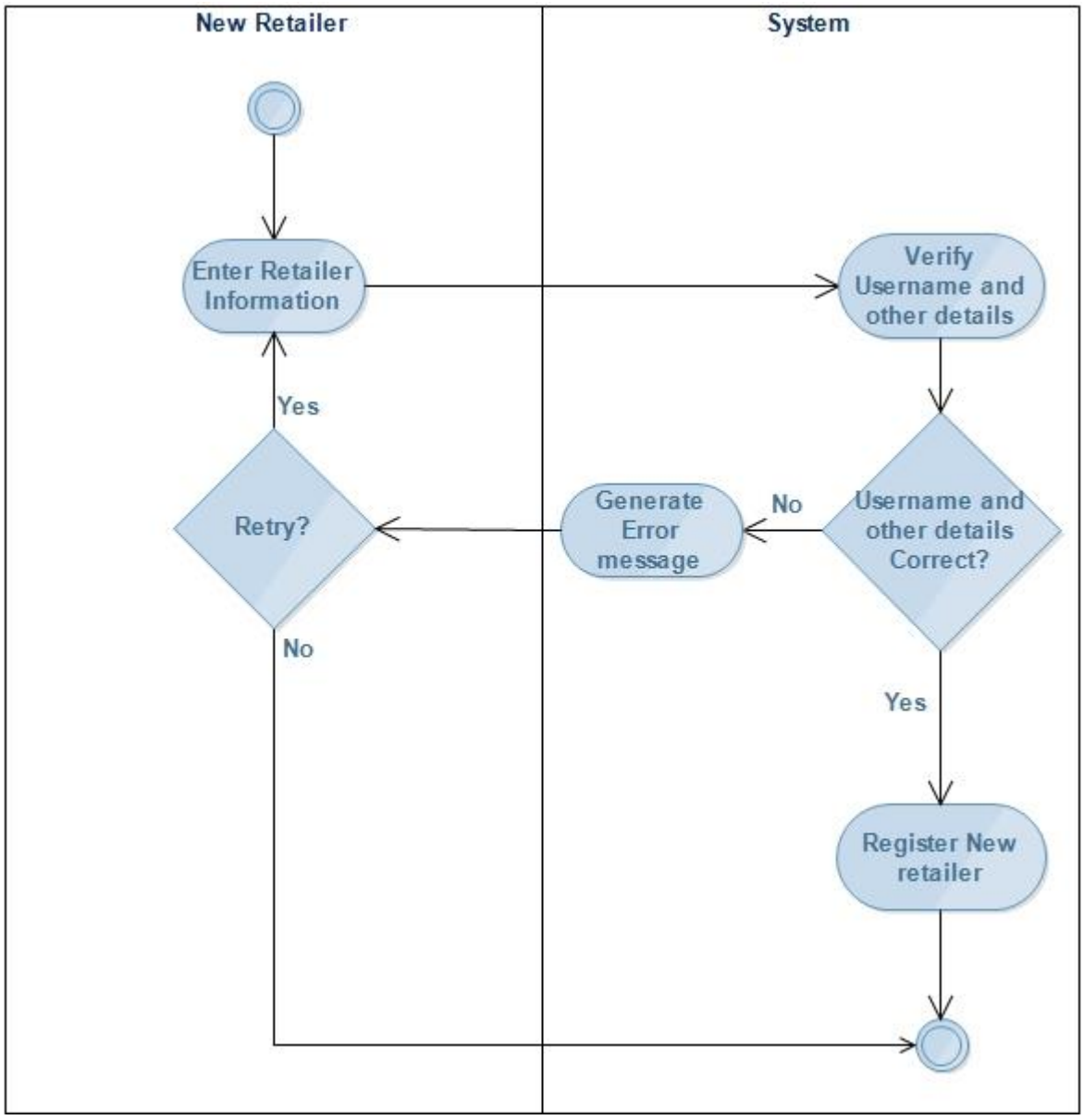


FIGURE 3.2.15 ACTIVITY DIAGRAM ‘REGISTER RETAILERS’

3.2.16 Update Customer Info Activity.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Update Customer’ which is done by customer.

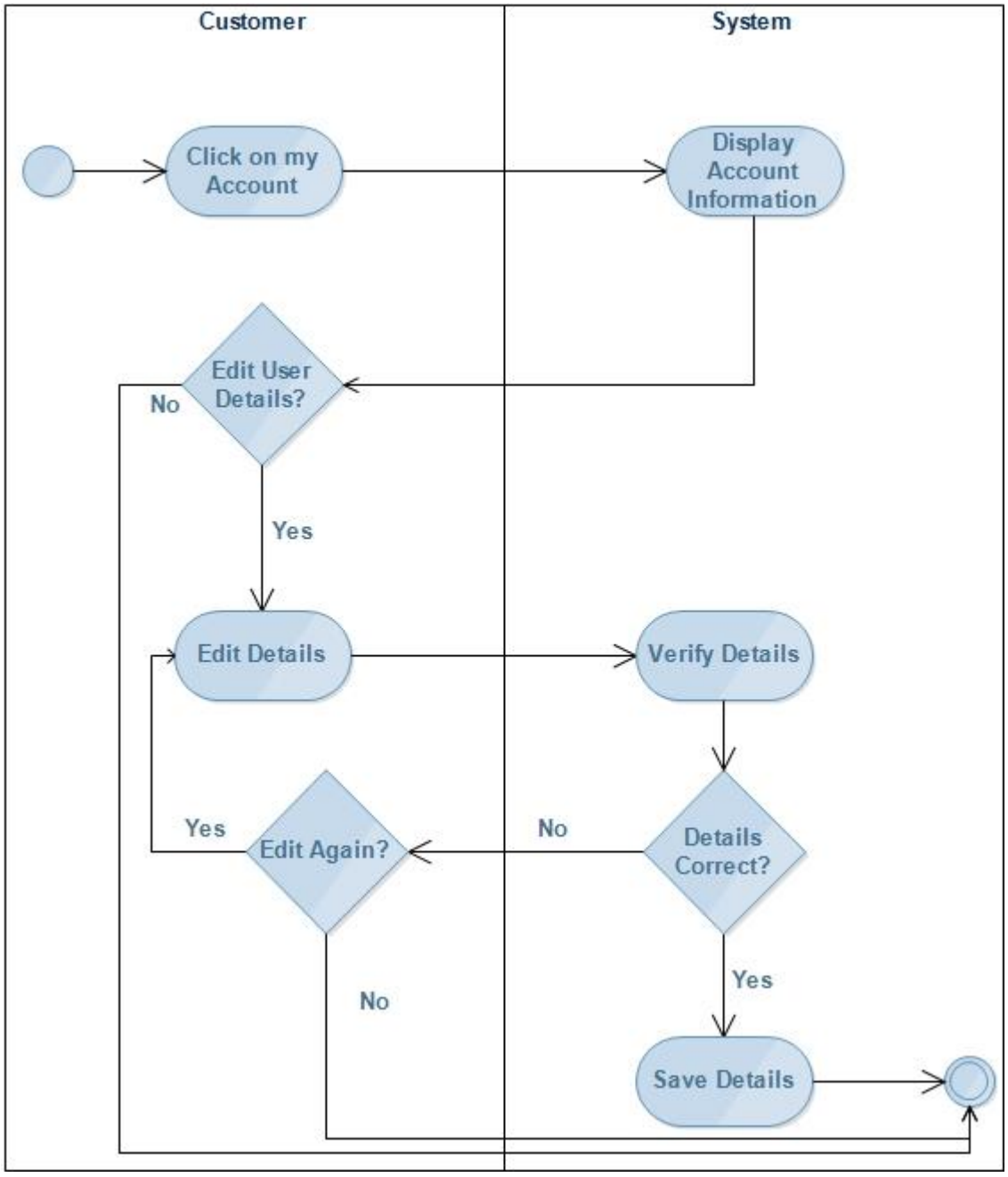


FIGURE 3.2.16 ACTIVITY DIAGRAM ‘UPDATE CUSTOMER INFO’

3.2.17 Handle Orders.

Below activity diagram is the further illustration of cause of actions occurred in use case ‘Handle Orders’ which is done by Retailer.

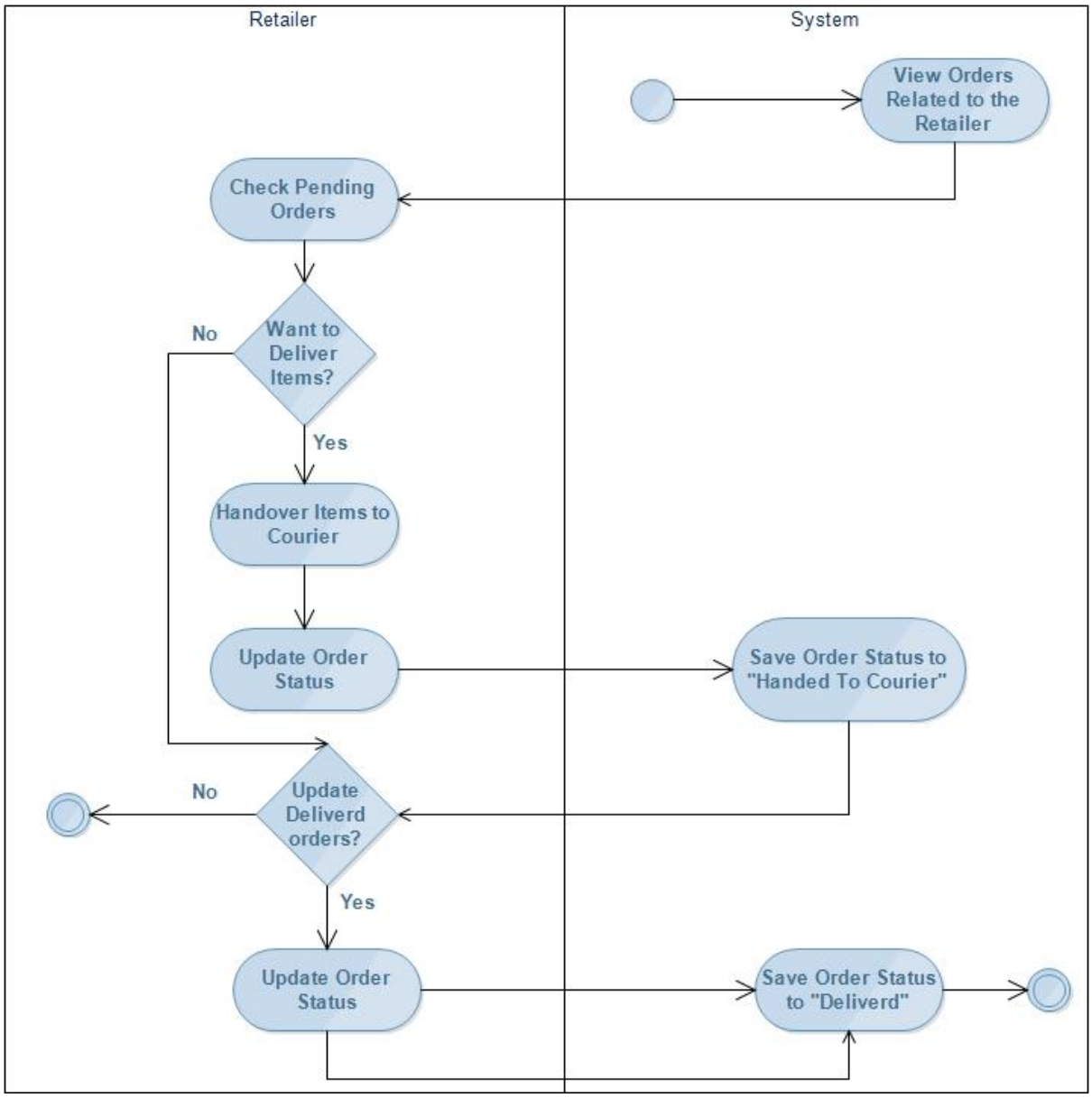


FIGURE 3.2.17 ACTIVITY DIAGRAM ‘HANDLE ORDERS’

3.3 Sequence Diagrams

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. Following are some of the main sequence diagrams for the system.

3.3.1 Login

Sequence of actions for User Login is illustrated in below sequence diagram.

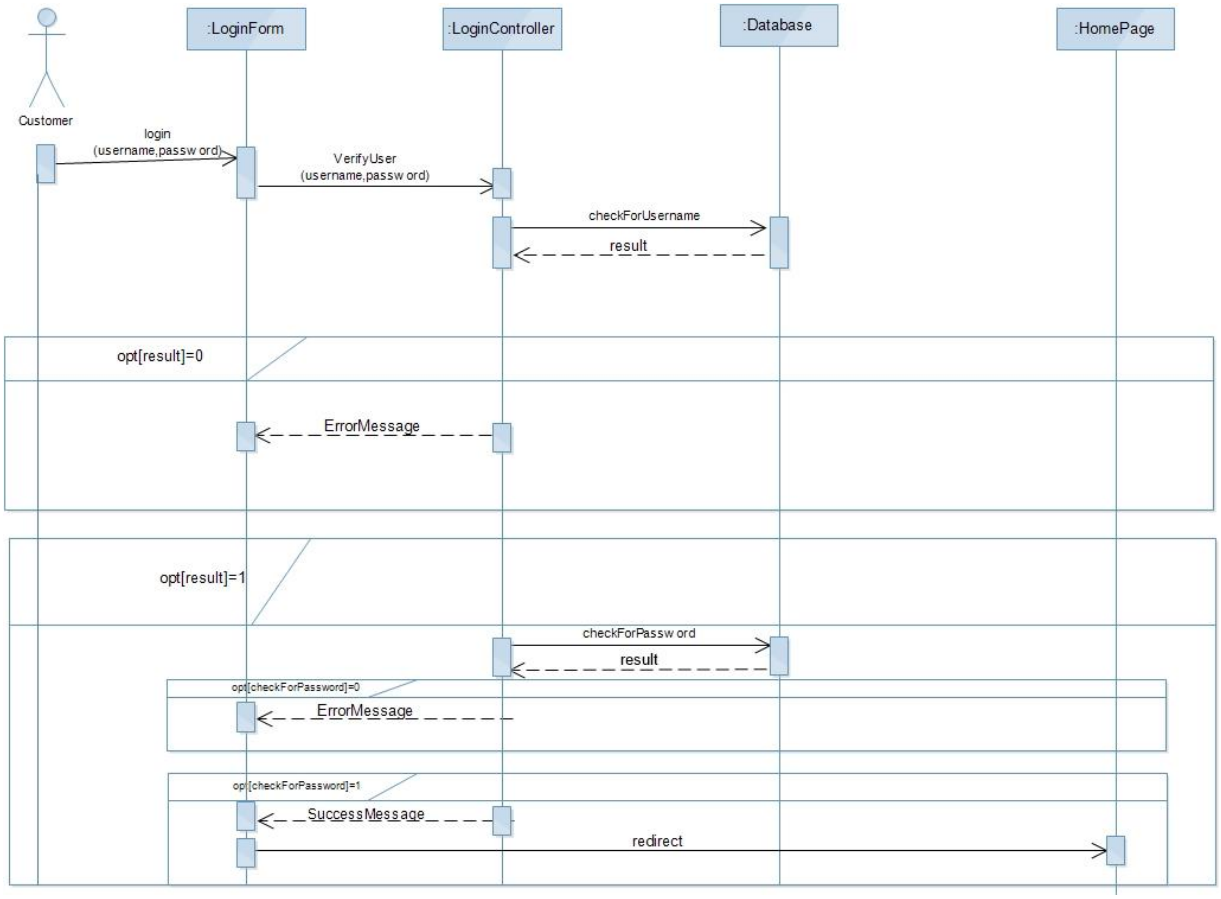


FIGURE 3.3.1 – SEQUENCE DIAGRAM 'USER LOGIN'

3.3.2 Search Items

Sequence of actions for Search Item is illustrated in below sequence diagram.

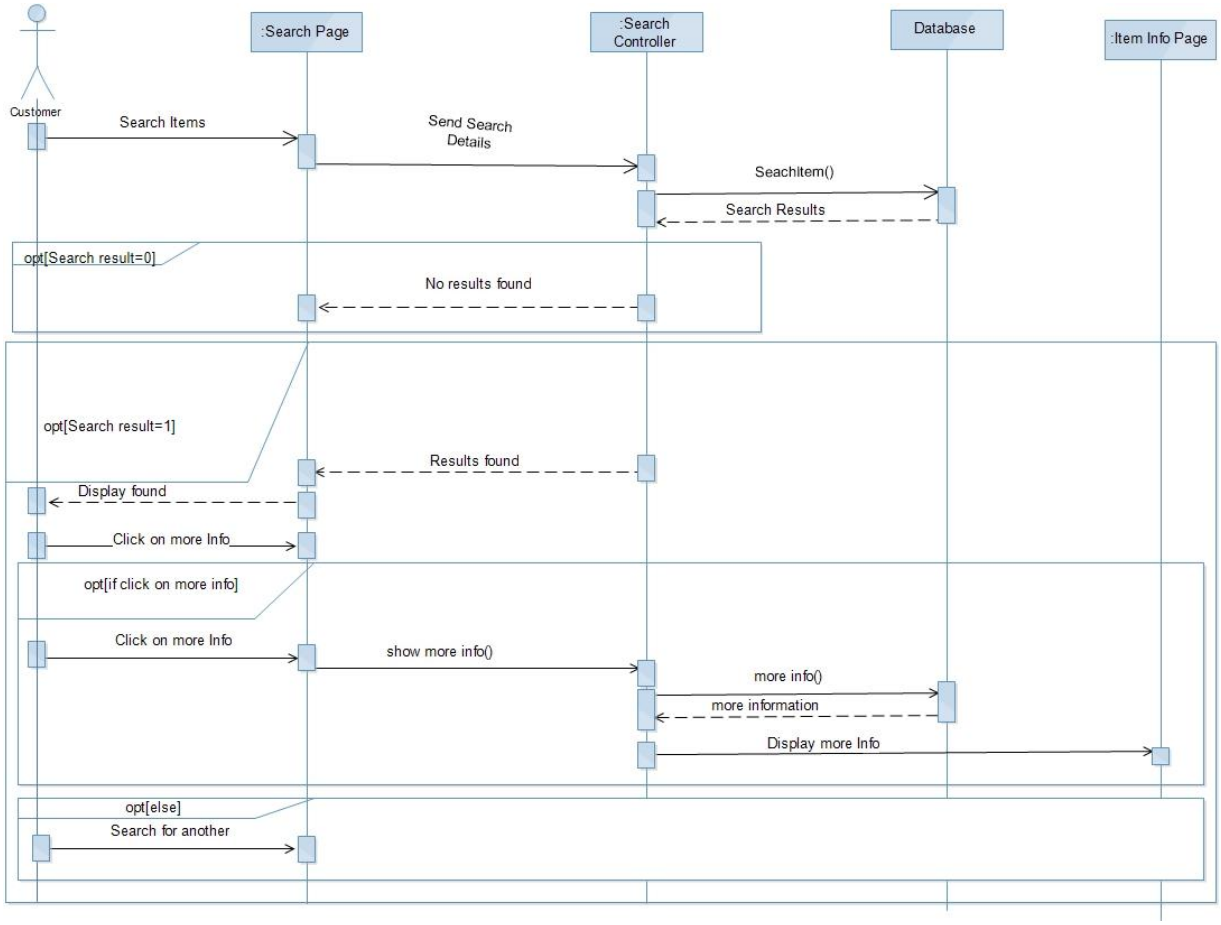


FIGURE 3.3.2 SEQUENCE DIAGRAM 'SEARCH ITEM'

3.3.3 Add to Cart

Sequence of actions for Add to Cart is illustrated in below sequence diagram.

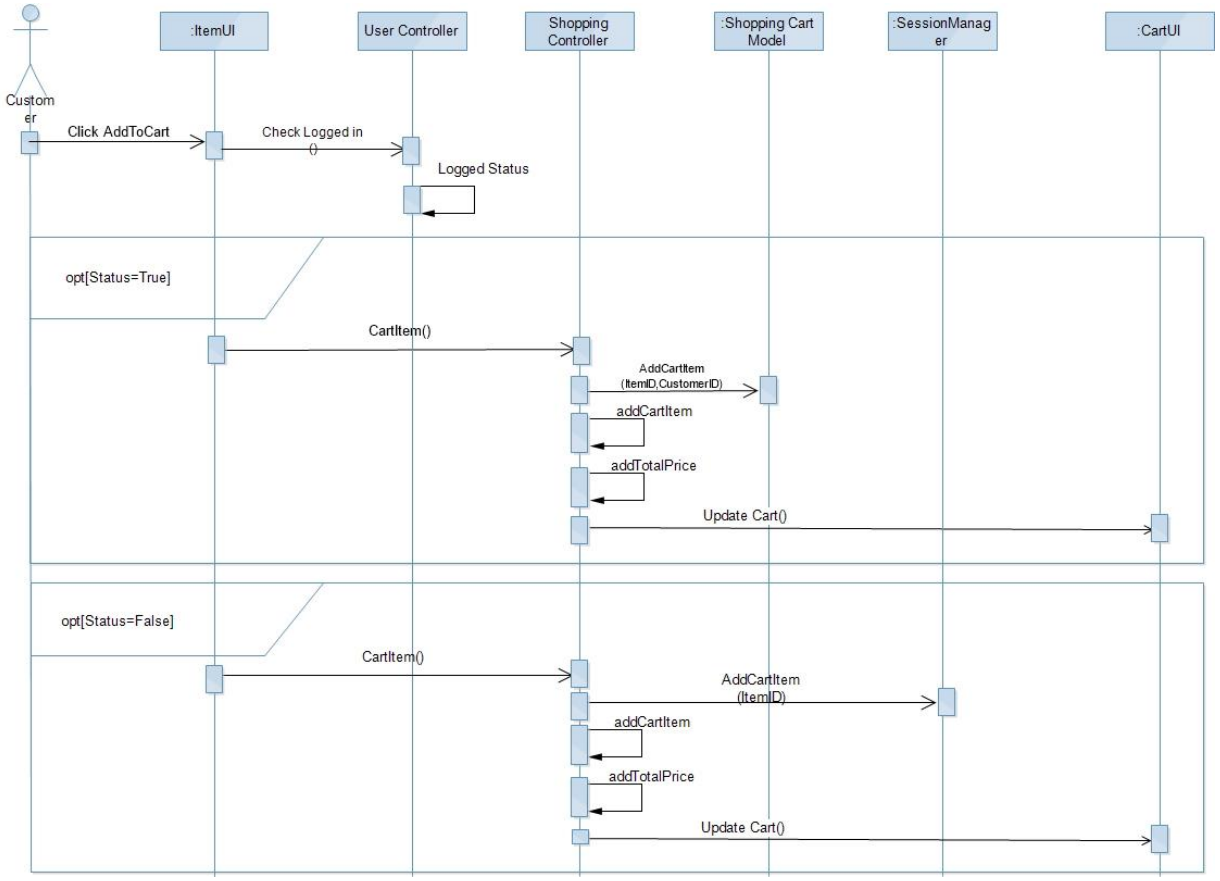


FIGURE 3.3.3 SEQUENCE DIAGRAM 'ADD TO CART'

3.3.4 Payment

Sequence of actions for Payment is illustrated in below sequence diagram.

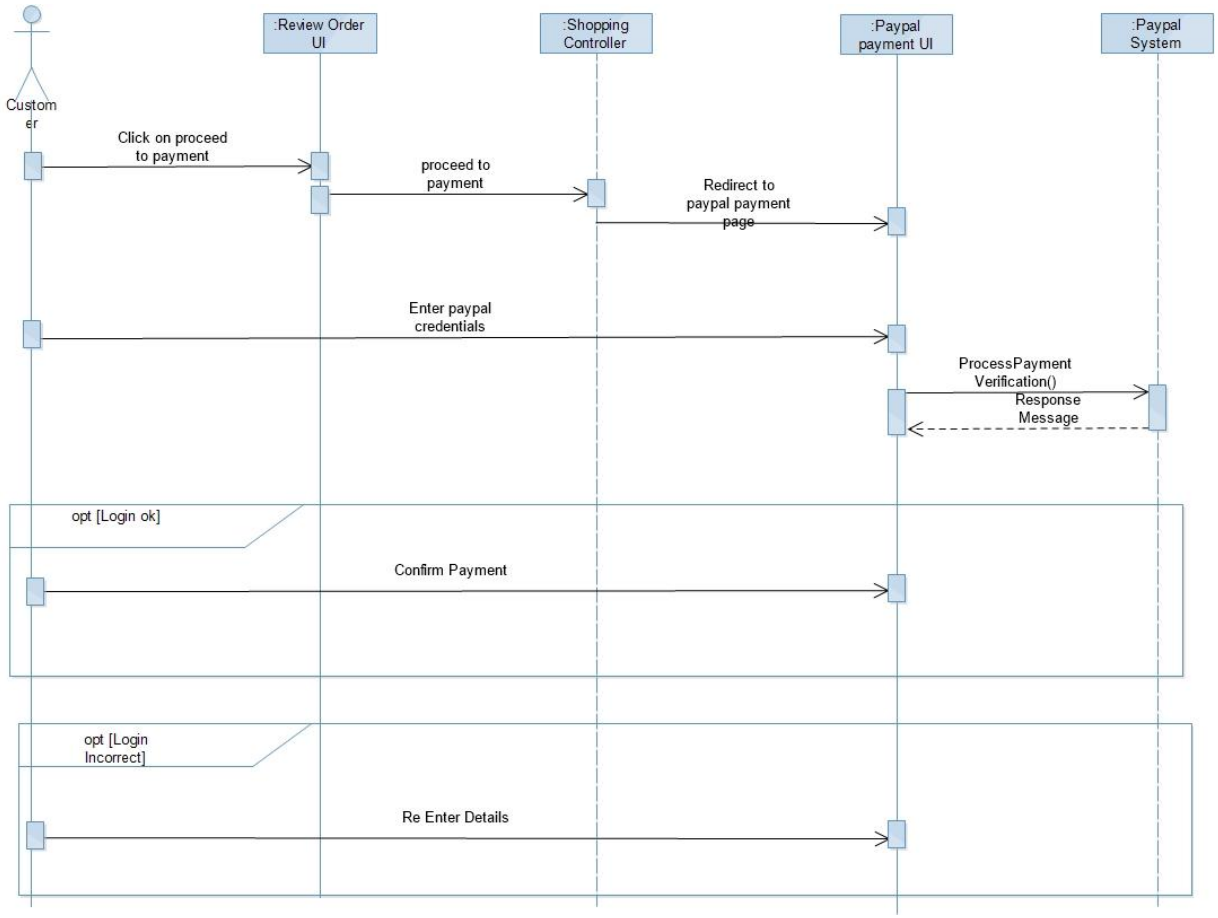


FIGURE 3.3.4 SEQUENCE DIAGRAM 'PAYMENT'

3.3.5 Update Inventory

Sequence of actions for Update Inventory is illustrated in below sequence diagram.

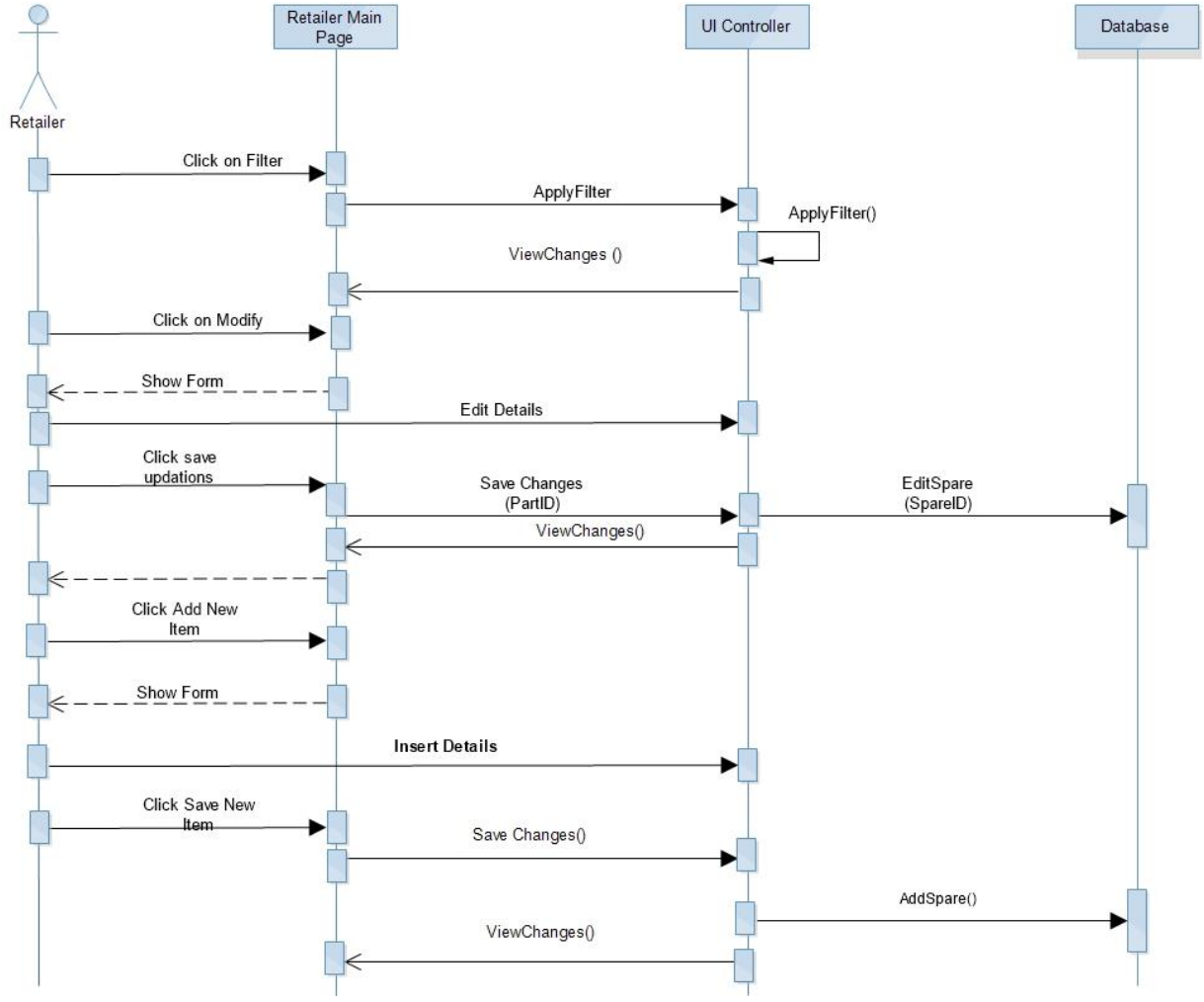


FIGURE 3.3.5 SEQUENCE DIAGRAM 'UPDATE INVENTORY'

3.3.6 Customer Registration

Sequence of actions for Customer Registration is illustrated in below sequence diagram.

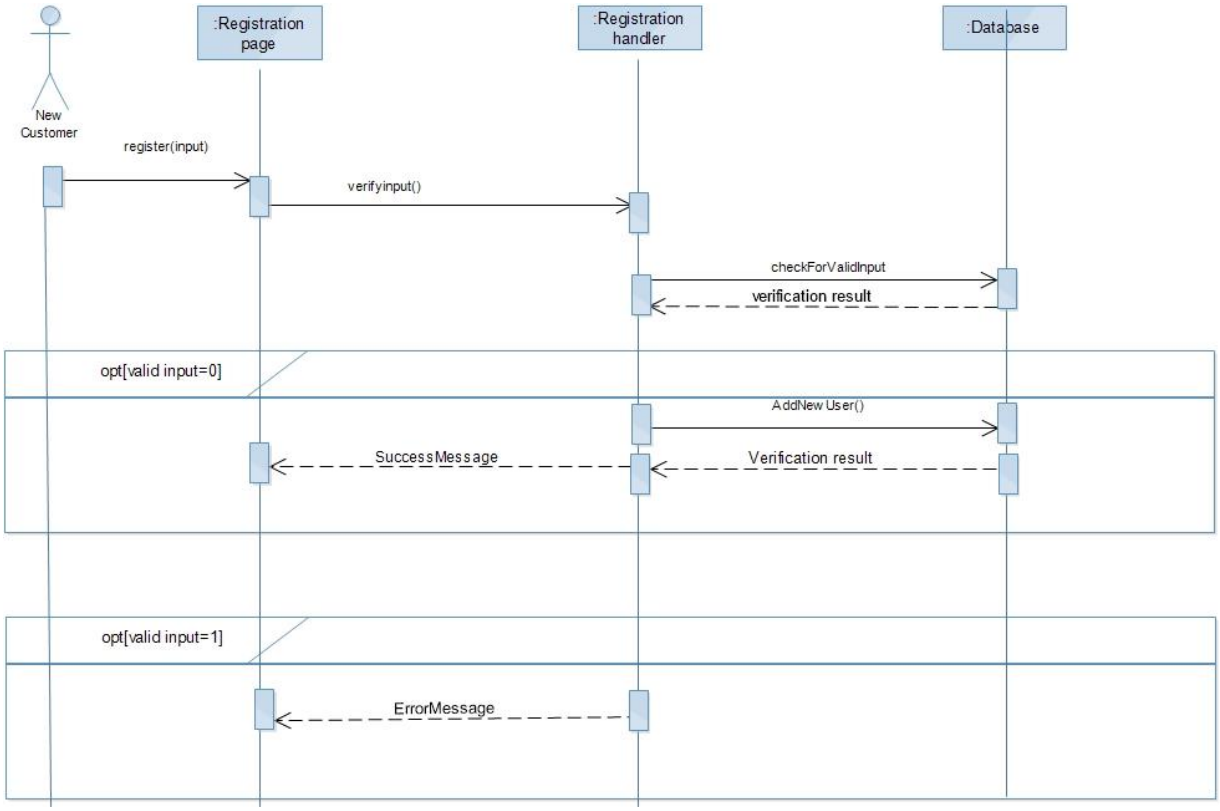


FIGURE 3.3.6 SEQUENCE DIAGRAM 'CUSTOMER REGISTRATION'

3.4 Class Diagram

Class diagram below describes the structure of a proposed system by showing the system's classes, their attributes, operations, and the relationships among the classes.

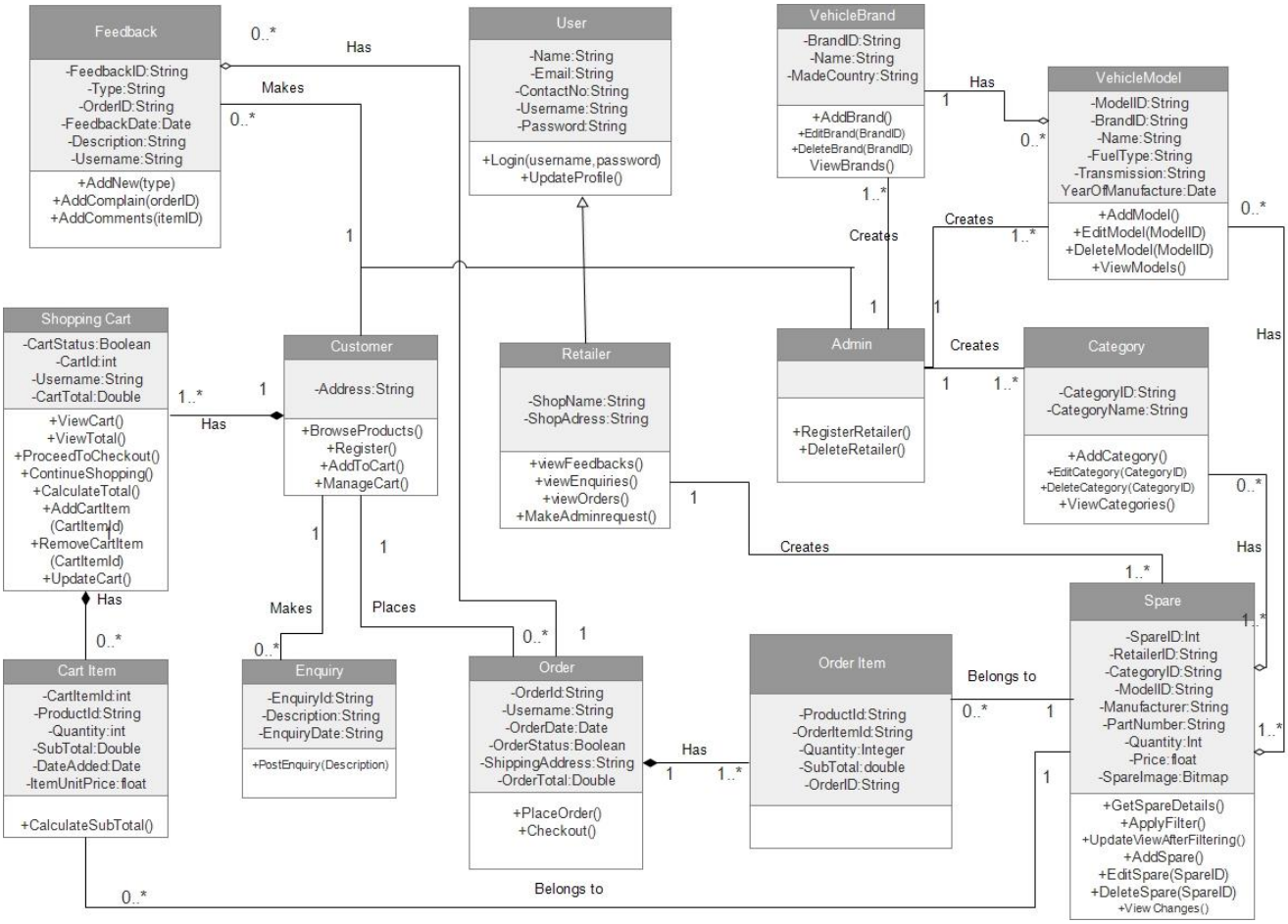


FIGURE 3.4.1 CLASS DIAGRAM

3.5 Database Design

3.5.1 ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

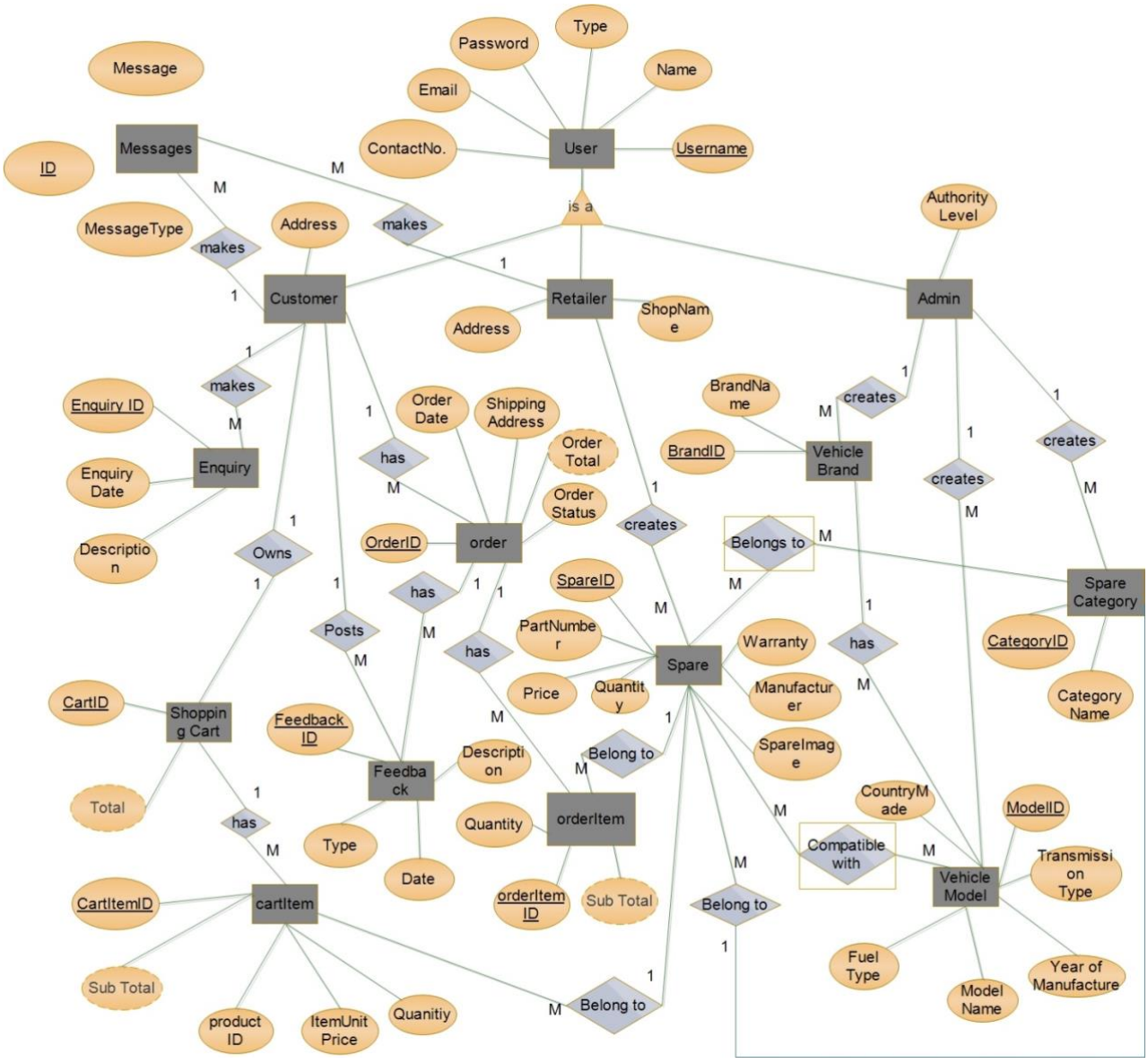


FIGURE 3.5.1 ER DIAGRAM

3.5.2 Database Design

This diagram illustrates the relationship between the tables in the database graphically.

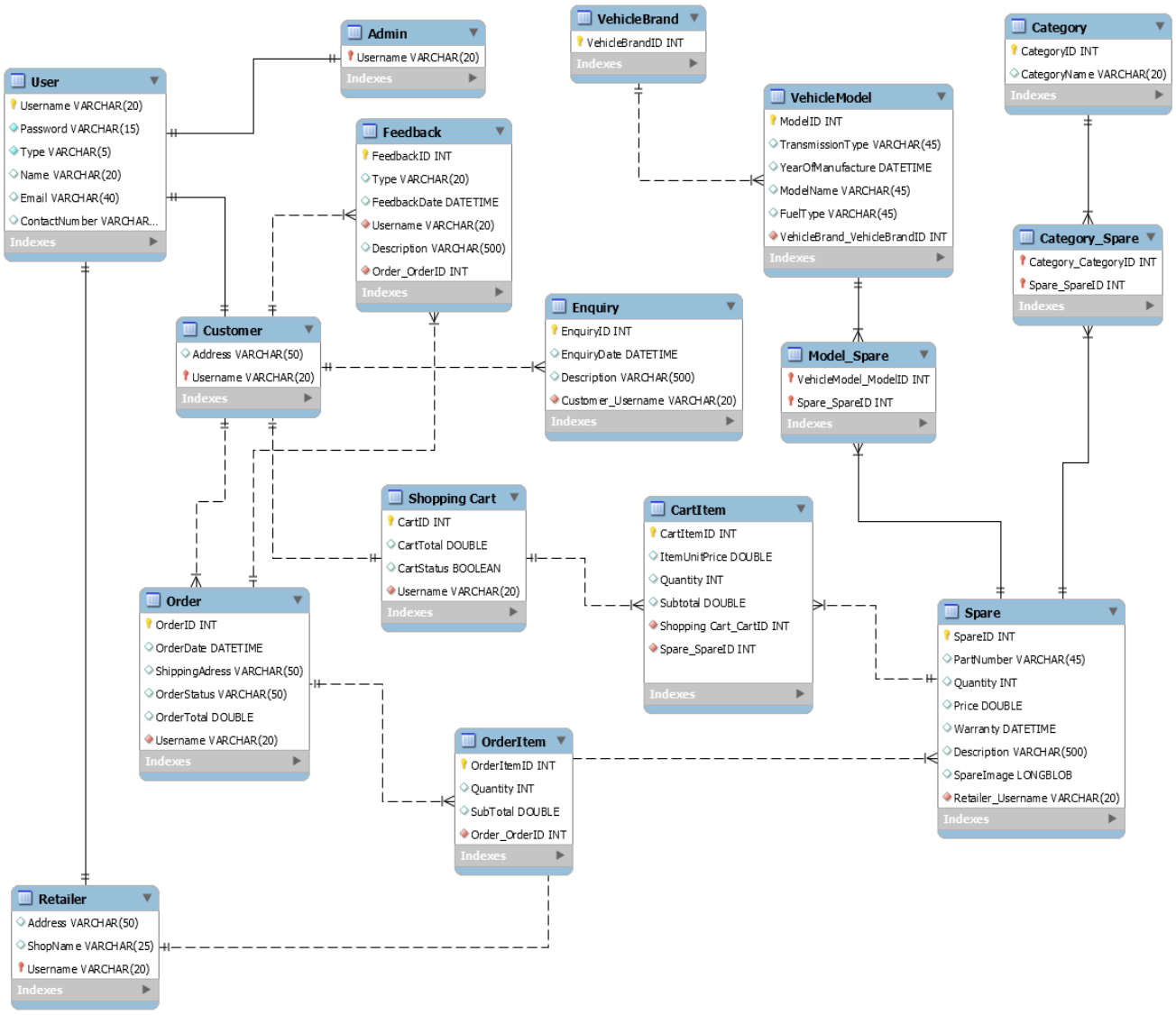


FIGURE 3.5.2 DATABASE DESIGN DIAGRAM

3.6 Interfaces

3.6.1 Login Form

Following figure shows the login form of the AutoHub system.

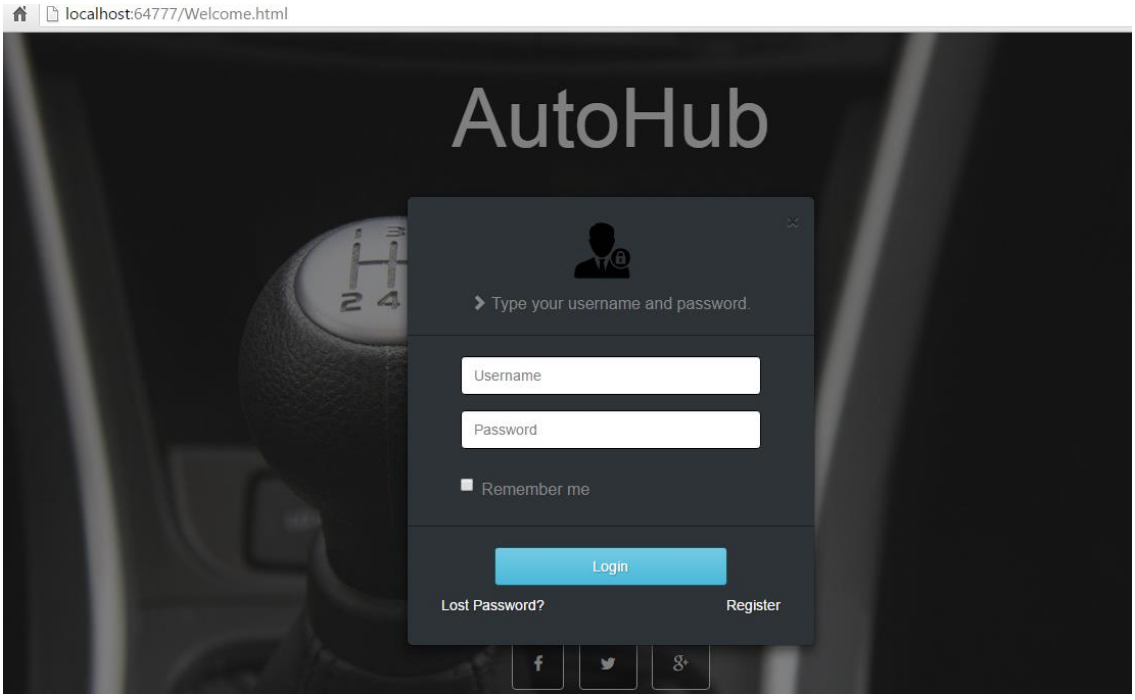


FIGURE 3.6.1 LOGIN FORM

User can enter his/her details and once he clicks on the login button then the system will check the validity of the username and password and redirect back to the particular dashboard.

3.6.2 Registration Form

Following figure shows form that is used to register customers to the system.

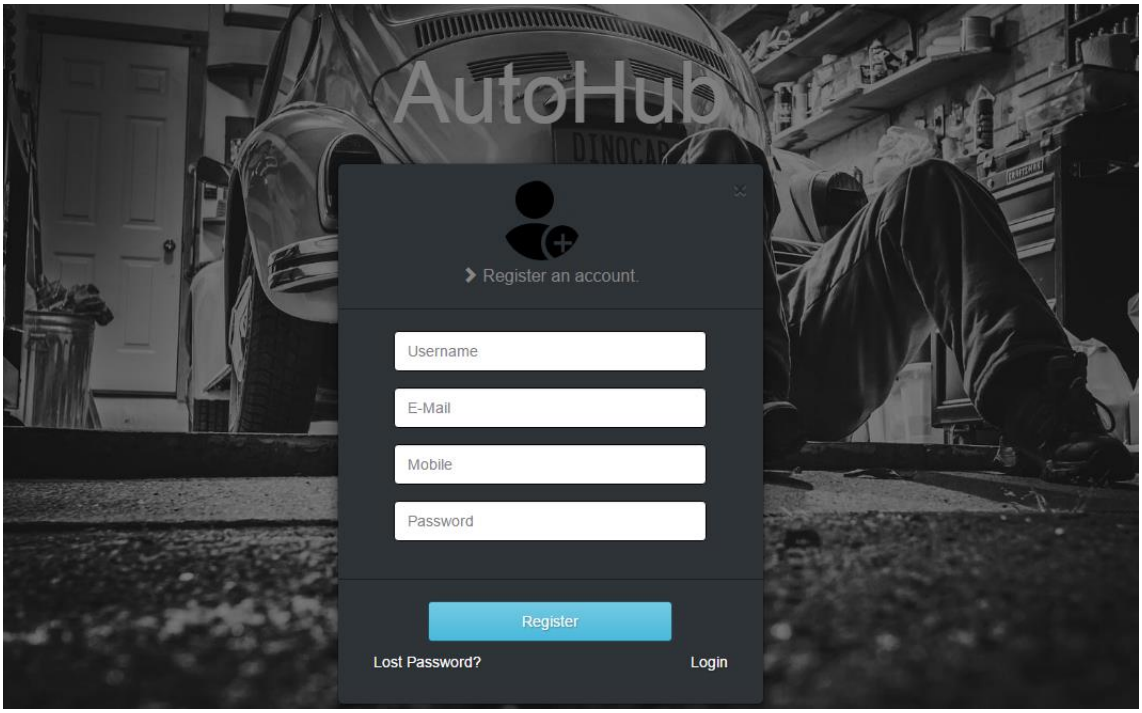


FIGURE 3.6.2 REGISTRATION FORM

If a particular customer is interested in using the system, they need to first register in the system. They need to fill out the above details and click on the register button and then a new user will be registered in to the system.

3.6.3 Welcome Page

Following figure shows the welcome screen of the system.

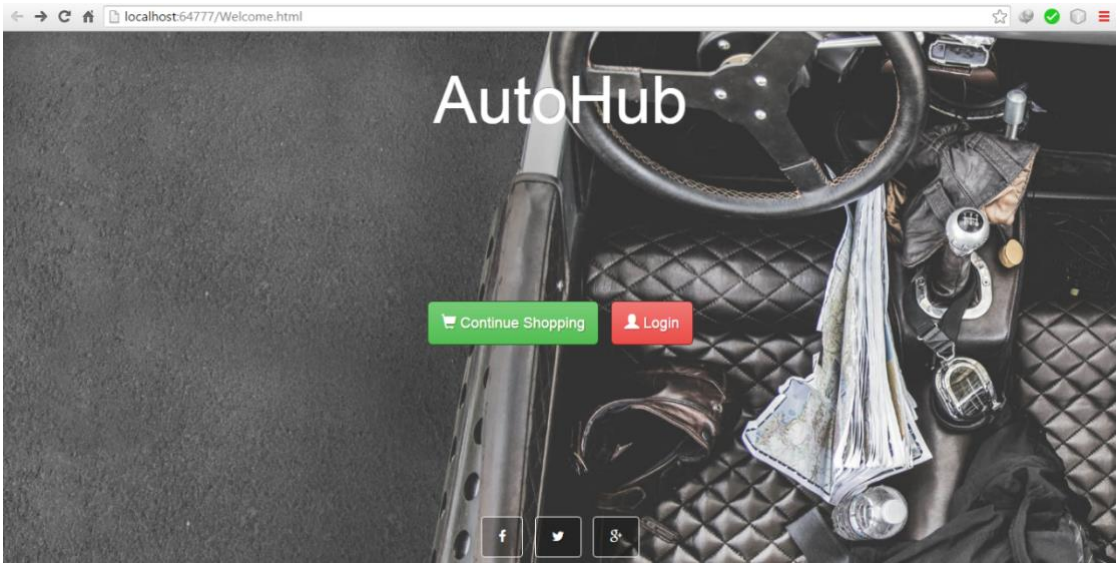


FIGURE 3.6.3 WELCOME PAGE

Customer can either directly use continue shopping button or else login and then access the services. But for the retailers and the admin initial login is necessary.

3.6.4 Home Page

Following figure shows the customer side home page.

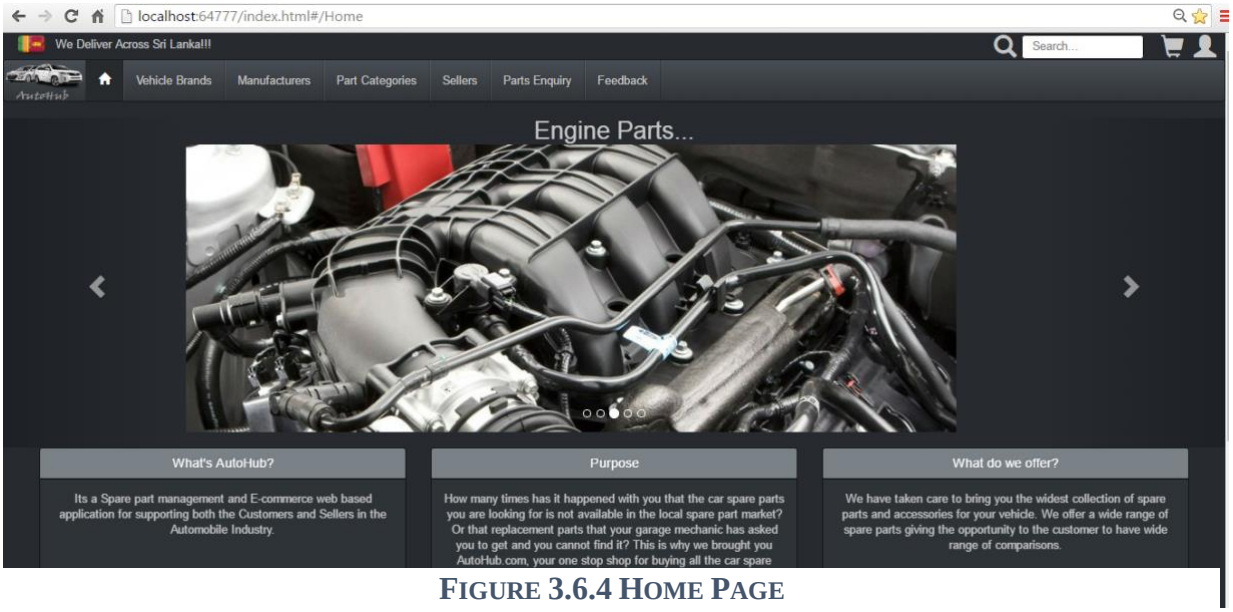


FIGURE 3.6.4 HOME PAGE

A user can navigate to any of the services in the tabs and access the services of the system quickly through this view.

CHAPTER 4

4 Development

4.1 Programming Languages and Development Tools

AutoHub is a web based spare parts management system. When selecting the programming languages and development technologies, compatibility with existing resources such as software, hardware, human and organizational capabilities have been more focused on. GUI design should be attractive and support form filling controls for most of the GUIs. In the native mobile application, also the modules are interacting with database through web service.

4.1.1 MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements

MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web-based applications and online publishing and is an important component of an open source enterprise stack called LAMP. LAMP is a Web development platform that uses Linux as the operating system, Apache as the Web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

4.1.2 Php

PHP stands for PHP: Hypertext Preprocessor. Hypertext refers to files linked together using hyperlinks, such as HTML (Hyper Text Markup Language) files. Preprocessing is executing instructions that modify the output. Taken directly from PHP's home, PHP.net, "PHP is an HTML-embedded scripting language. Web solution server side implementations are written on PHP scripting language. Within web solution PHP used as server side validation scripting language Mobile.

4.1.3 JavaScript

JavaScript enable programmatic access to computational objects within a host environment and that is implemented as part of a web browser in order to provide enhanced user interfaces and dynamic websites. In the Study Room the JavaScript is used to enhance the usability through providing more interactive, speedy environment and to implement the client side validation as well.

4.1.4 CSS

Short for Cascading Style Sheets, a new feature being added to HTML that gives both Web site developers and users more control over how pages are displayed. By using CSS one could separate HTML content from its appearance, distinguishing style from structure. CSS give the following advantages.

- Control layout of many documents from one single style sheet.
- More precise control of layout
- Apply different layout to different media-types.

4.1.5 HTML

HTML is the most basic building block of the Web. It describes and defines the content of a webpage. Other technologies besides HTML are generally used to describe a webpage's appearance/presentation (CSS) or functionality (JavaScript).
"Hypertext" refers to links that connect webpages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

4.1.6 Laravel

Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar.

4.1.7 MVC architecture

The Model represents your data structures. Typically, your model classes will contain functions that help you retrieve, insert, and update information in your database

The View is the information that is being presented to a user. A View will normally be a web page, but in Laravel, a view can also be a page fragment like a header or footer. It can also be an RSS page, or any other type of "page".

The Controller serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

Within the AutoHub web solution has been used Laravel MVC framework to enhance the development capabilities.

4.1.8 PhpStorm IDE

PhpStorm is a lightweight and smart PHP IDE focused on developer productivity that deeply understands your code, provides smart code completion, quick navigation and on-the-fly error checking. It is always ready to help you shape your code, run unit-tests or provide visual debugging.

This is a community for sharing information, tips, and tricks regarding the PHP IDE from JetBrains.

4.1.9 XAMPP

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), Maria DB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (Maria DB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.

4.1.10 AJAX

AJAX is a latest web development technology which allows dynamic web page contents with scripting and data manipulation with JSON/XML objects. Asynchronous JavaScript and XML is a group of interrelated web development methods used on the client-side to create asynchronous web applications. With Ajax, this web application can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behavior of the existing page. Data is retrieved using the HttpRequest object using objects and arrays

4.1.11 JQuery

This is cross browser JavaScript library designed to simplify client side scripting of HTML web pages. jQuery is free, open source software, jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developer create more dynamic web applications. JQuery is used to implement image gallery, and interactive web pages with toggle option.

4.1.12 PayPal

PayPal is an online service allowing users to transfer money without sharing financial information with the recipient. Users pay for transactions through credit/debit cards, bank accounts, PayPal accounts or financing.

4.1.13 NodeJs

An important thing to realize is that Node is not a webserver. By itself it doesn't do anything. It doesn't work like Apache. There is no config file where you point it to you HTML files. If you want it to be a HTTP server, you have to write an HTTP server (with the help of its built-in libraries). Node.js is just another way to execute code on your computer. It is simply a JavaScript runtime. NodeJs has been used to connect the admin and the retailer to an external private chat.

Chapter 5

5 Testing

Software testing aspects of the Collaborator is discussed under this chapter. Under this chapter First it discussed the strategies and types of testing used, why and how they were employed to test the system. Then it will discuss about the sample test cases, followed by the test reports of the system and results illustrating the severity of the bugs identified and possible solutions.

Outline of the chapter.

5.1 Test Plan and Test Strategy

5.2 Sample Test Cases

5.3 Test Report

5.1 Test Plan and Test Strategy

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The functionalities and functions are tested in order to improve the quality of the product and improve the functionalities of the product too. The testing plan defines the items to be tested and the functions are selected bases on the importance of the functions, and the risk of the functions on the users view point. Then the test cases were designed corresponding with the use case descriptions. They were executed manually and the results were recorded. The bugs identified were corrected and tested again. In the testing process of this system both black box testing and white box testing has used.

Most important use cases identified by the user and the developer were added to a test plan. Then test cases were designed corresponding with use case descriptions. Those test cases were executed and results were recorded. Identified bugs were corrected and tested again. Both black box testing and white box testing were used in testing.

White box testing was carried out while the functionality was developing. The code was reviewed after one part of code was finished to identify logic errors and programming errors that could lead to bugs in certain cases. If any errors were found those were corrected. This ensures that the block of code is bug free before carrying on to the next block of code. This is similar to unit testing.

Black box testing is used to test the software without knowing the internal structure of code or program. Most likely this testing method is what most of tester actual perform and used the majority in the practical life. Basically, software under test is called as “Black-Box”, we are treating this as black box & without checking internal structure of software we test the software.

User interface testing basically means testing the UI by considering parameters such as consistency, usability, visibility, accessibility and compatibility. Many developers have a tendency to place too heavy an emphasis on the functional aspects, underlying structure and technical brilliance of software.

5.2 Sample Test cases

ID	Test Case Name	Description	Input	Expected Output	Result
1.1	Email Address Validity	Insert random text as Email input violating the email format	“111aaa”	Doesn’t let the user to submit if email format is incorrect.	pass
1.2	Email format	Insert a properly formatted email but not a registered email.	“heshana @gmail.com”	Give an alert “User Credentials does not match”	pass
1.3	Email empty	Keep the email input empty	“ ”	Show that field is empty while focusing on it.	pass
1.4	Password empty	Keep the password input empty	“ ”	Show that field is empty while focusing on it.	pass
1.5	Incorrect password	Insert a Wrong Password	“fdgsdg”	Give an alert “User Credentials does not match”	pass

TABLE 5.2.1 TEST CASE USER LOGIN

ID	Test Case Name	Description	Input	Expected Output	Result
2.1	Email Address Validity	Insert random text as Email input violating the email format	“111aaa”	Doesn’t let the user to register if email format is incorrect.	pass
2.2	Email address reuse	Insert an email that has been already registered	“heshananupama@gmail.com”	The email has already been taken.	pass
2.3	Email null	Keep the email input empty	“ ”	Show that field is empty while focusing on it.	pass
2.4	Password Validity	Keep the password input empty	“ ”	Give an alert “invalid date”	pass
2.5	Password confirmation	Insert a Wrong Confirm Password	“abc” and “ab”	Give an alert “Password Confirmation does not match”	pass
2.6	Password Length	Password length	“anu”	Password should least 6 characters’ alert pops up.	pass

TABLE 5.2.2 TEST CASE REGISTER

ID	Test Case Name	Description	Input	Expected Output	Result
3.1	Brand Null	Keeping brand Dropdown to null	“ ”	Show error on submitting “The brand name field is required”	pass
3.2	Model Null	Keeping model input to null	“ ”	Show error on submitting “The model name field is required”	pass
3.3	Country null	Keep the country input empty	“ ”	Show error on submitting “The country field is required”	pass
3.4	Transmission Type null	Keep the Transmission type dropdown null	“ “	Show error on submitting “The transmission type field is required”	pass
3.5	Fuel Type Null	Keep the Fuel type dropdown null	“ ”	Show error on submitting “The transmission type field is required”	Pass
3.6	Year Not Formatted	Keeping the	“gvg”	Show error on submitting “The year does not match the format”	pass
3.7	Country not formatted	Keep the password input empty	“ ”	Give an alert “invalid date”	pass

TABLE 5.2.3 TEST CASE ADD VEHICLE MODEL

ID	Test Case Name	Description	Input	Expected Output	Result
4.1	Name Null	Keeping retailer name null	“ ”	Show error on submitting “The name field is required.”	pass
4.2	Shop name Null	Keeping Shop Name input empty	“ ”	Show error on submitting “The model name field is required”	pass
4.3	Address null	Keep the address field empty	“ ”	Show error on submitting “The address field is required”	pass
4.4	Password null	Keep the password field empty	“ “	Show error on submitting “The password field is required”	pass
4.5	Email Null	Keep the email field empty	“ ”	Show error on submitting “The email field is required”	Pass
4.6	Password confirmation	Insert a Wrong Confirm Password	“abc” and “ab”	Give an alert “Password Confirmation does not match”	pass
4.7	Retailer Image	Insert a wrong file type	“alert.pdf “	Give an alert “The retailer image must be a file of type: jpeg, png, jpg, gif, svg.”	pass

TABLE 5.2.4 TEST CASE REGISTER RETAILERS

ID	Test Case Name	Description	Input	Expected Output	Result
5.1	Brand Null	Keeping brand name null	“ ”	Show error on submitting “The Brand name field is required.”	pass
5.2	Existing Brand Name	Adding an existing Brand Name.	“Toyota”	Show error on submitting “The brand name is already taken”	pass

TABLE 5.2.5 TEST CASE ADD VEHICLE BRAND

ID	Test Case Name	Description	Input	Expected Output	Result
6.1	Category Null	Keeping category name null	“ ”	Show error on submitting “The category name field is required.”	pass
6.2	Category Name	Adding an existing category Name.	“Engine”	Show error on submitting “The category name is already taken”	pass

TABLE 5.2.6 TEST CASE ADD CATEGORY

ID	Test Case Name	Description	Input	Expected Output	Result
7.1	Message Null	Keeping message text area null	“ ”	Show error on submitting “The text area field is required.”	pass
7.2	Contact number null	Putting contact number with string.	“2fsfsd0000 ”	Show error on submitting “The contact number must be a number”	pass
7.3	Contact number length	Contact number less than 10 numbers	“555”	Show an error on submitting “contact number must be at least 10”	pass

TABLE 5.2.7 TEST CASE ADD ENQUIRY

ID	Test Case Name	Description	Input Data	Expected Output	Status:
8.1	Product quantity	Add New Product with higher quantity	100	Show a message in a toast ―only 2 units remaining.	Pass
8.2	Adding an item that is already in cart	1. Select Product 2. Select quantity,	2	Cart will be successfully update quantity on the relevant spare	Pass

TABLE 5.2.8 TEST CASE ADD PRODUCT INTO CART

5.3 Test Report

Testing was simultaneously done as the development was progressed. Throughout the development code units were unit tested. For a particular process the code was tested depending on various inputs. Then modules were unit tested. Then integration testing was done.

The reason for testing throughout the year was to identify bugs early as possible because it will be too costly and time taking to correct if bugs were found at the end of development.

Tests were performed iteratively. Number of bugs detected by test cases and test data was reduced with each iteration. Some bugs took more time than expected to fix. But fixing those bugs was important to system to work properly.

Bugs which were found during the tests were fixed immediately. If it could not be fixed then and there, it was written down and retried later.

Methods used to find bugs when testing and developing are as follows.

I. Laravel dd () method were used to test functionality of server side operations.
II. JavaScript console.log function was used to identify errors in client side.
III. Google Chrome developer tools console was used to monitor logs added by JQuery and JavaScript.
IV. Postman was used to debug the network requests.

5.3.1 Testing Environment

Hardware:	Intel Core i7 2.4GHz
	8 GB RAM
Software:	Windows 8.1 Operating System (64 bit)
	Google Chrome web browser.
	Postman.

6 SYSTEM IMPLEMENTATION

This chapter discusses about implementing Collaborator. Minimum software and hardware requirements for running the application, installation guide is included in the chapter. User manual which explains how to use the system is also included.

Outline of the chapter

6.1 System Requirements

6.2 Installation Guide

6.3 User Manual

6.1 System requirements

6.1.1 Hardware requirements

Hardware Requirements of the Client machine.

The machine should be in the following configuration.

i. Intel core i3 1.8 or similar
ii. 2GB or above RAM
iii. 500 MB or above Hard Disk

6.1.2 Software requirements

Operating System

- Windows 7/8/8.1/10

Backend Software

- Laravel 5.3
- XAMPP/WAMP
- NodeJs

6.2 Installation Guide

6.2.1 Installation of website development tools

First, XAMPP Server needed to be installed. Version must be 3.2 or above.

i. Download the XAMPP Server installer by choosing among 32-bits or 64-bits version depending on the Windows version.
ii. Run the downloaded installer to initiate the setup. Complete the setup by following all wizard instructions until the end. (XAMPP Server will require around 350MB+ space on the disk). It is recommended to install XAMPP on *C:/XAMPP*.
iii. Move 'AutoHub' folder (project folder) to ―C:\Xampp\htdocs folder
iv. Open web browser and follow this address 'http://localhost/phpmyadmin/'
v. Create database 'autohub'.
vi. Go to Import section of PhpMyAdmin.
vii. Browse autohub.sql file in ―C:\xampp\htdocs\ autohub folder and upload that file in to Import section.
viii. Click on go button. Then database will be created.

Then, configure Laravel on your system

The Laravel framework has a few system requirements. Of course, all of these requirements are satisfied by the Laravel Homestead virtual machine, so it's highly recommended that you use Homestead as your local Laravel development environment.

However, if you are not using Homestead, you will need to make sure your server meets the following requirements:

- PHP >= 5.6.4
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension
- Tokenizer PHP Extension
- XML PHP Extension

6.2.2 Installing Laravel

Laravel utilizes Composer to manage its dependencies. So, before using Laravel, make sure you have Composer installed on your machine through https://getcomposer.org/.

6.2.2.1 Alternative: Via Laravel Installer

First, download the Laravel installer using Composer:

composer global require "laravel/installer"
Make sure to place the `$HOME/ composer/vendor/bin` directory (or the equivalent directory for your OS) in your $PATH so the `laravel` executable can be located by your system.

Additional System Configurations

Go to c://windows/System32/Drivers/etc/hosts and add the ‘127.0.0.1 autohub.com’ to the file and save (you need windows authorization while doing so).

Then go to your XAMPP folder /Apache/conf/extra/httpd-vhost.conf file and copy

```
<VirtualHost autohub.com:80>
    DocumentRoot "D:/xampp/htdocs/Autohub/public"
   ServerName autohub.com
</VirtualHost>
```

and paste the above lines of codes.

6.3 User Manual

This user manual will guide you through the system functionality enabling you, easily manage and standardize AutoHub activities. This software is created by H.A. Perera, with special customized features to satisfy the requirements of Sri Lankan spare part industry and accessories customers. Three user categories are authorized to access the system. They are,

I. Admin
II. Customers
III. Retailers

AutoHub is facilitates a high-level assistance for customers of Sri Lanka who shopping spare parts and accessories. Number of features will be given to the user to be a member of the system and use the web application in a productive manner.

Admin Side:

- Add, update, delete brands, categories and models.
- Register Retailers.
- Identify the summary of the AutoHub system.

Retailer Side:

- Add, update, delete a spare.
- View/Reply enquiries.
- View Complains.
- View/Change order status.
- Make/View various reports and charts.

Customer side mobile application:

- Product comparison among different shops.
- Order items.
- Purchase Items.
- Shopping Cart.
- Reviewing system and ratings.
- Advance search option.
- Filtering search results.
- Make enquiries.
- Make complains.

6.3.1 Admin Side

6.3.1.1 Login

Once user enter details and click login button they can logging to the necessary dashboard.

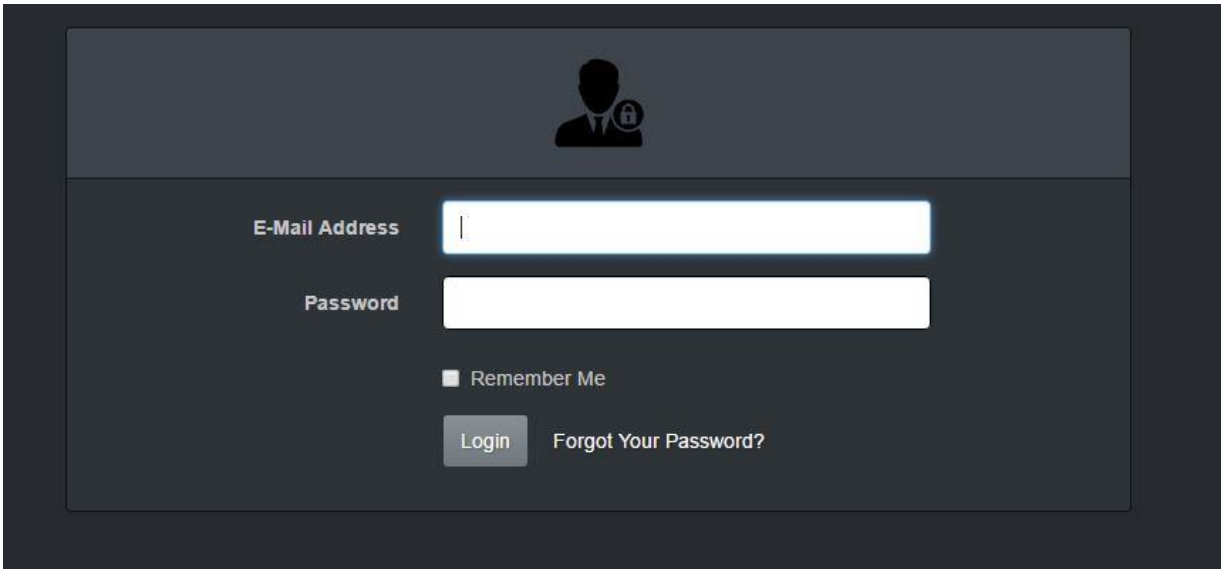


FIGURE 6.3.1 LOGIN

Steps:

1. Enter email address.
2. Enter Password.
3. Click on login button.
4. If user forgot password, click on forgot password link.

6.3.1.2 Registration

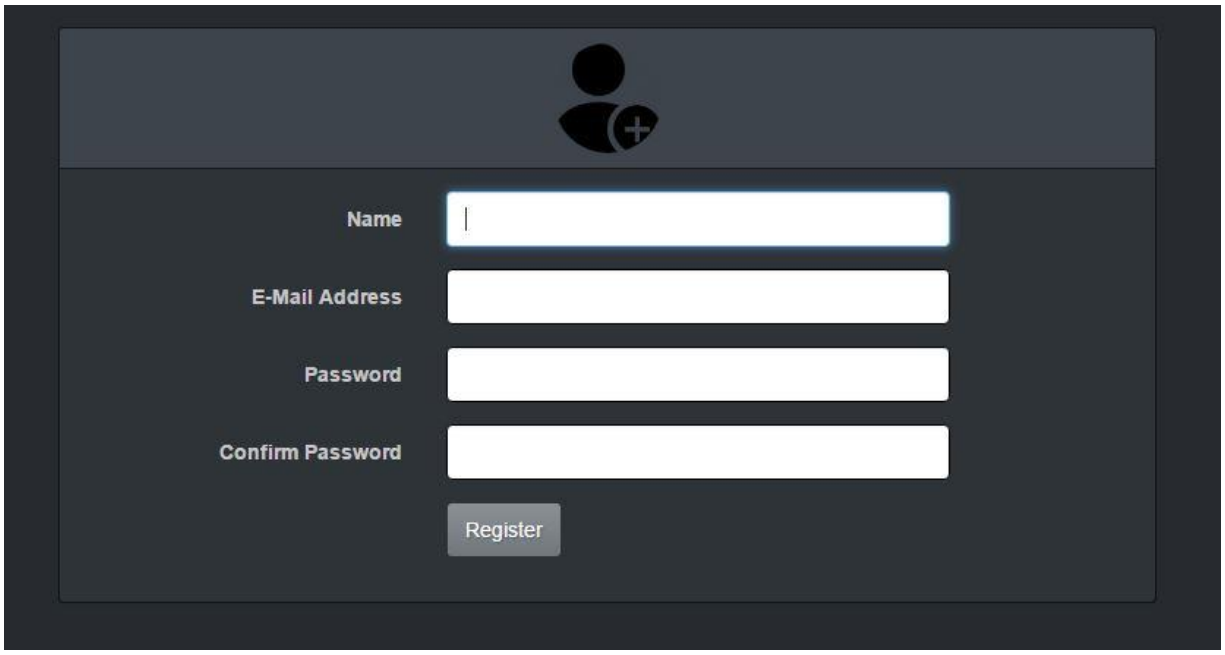


FIGURE 6.3.2REGISTRATION

Steps:

1. Enter name.
2. Enter email address.
3. Enter password and confirm password.
4. Click on register button

6.3.1.3 Admin home page

Three main user groups are customer, retailer and admin.

After the admin make a successful login, this home page will appear. This is admin's view.

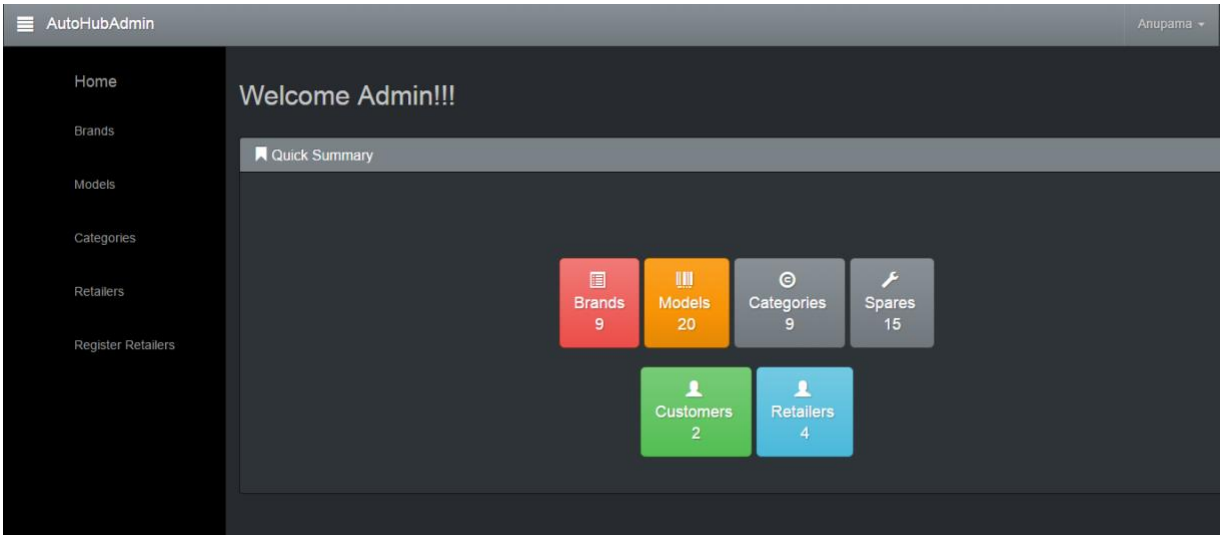


FIGURE 6.3.3 ADMIN HOME PAGE

6.3.1.4 Manage Brands

Admin can add/edit/delete vehicle brands through this view.

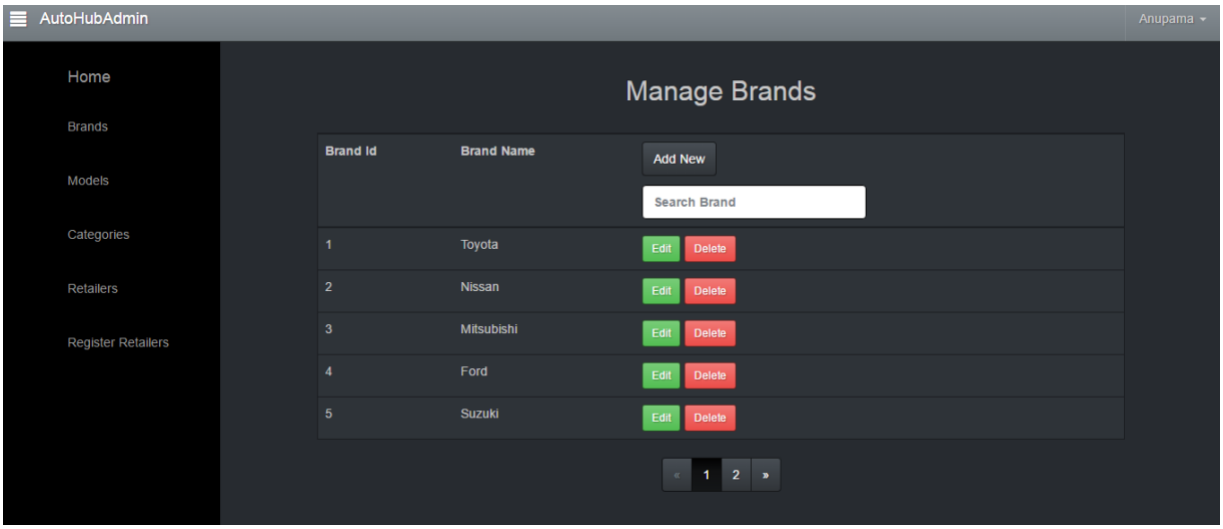


FIGURE 6.3.4 MANAGE BRANDS

Steps:

1. Click on Brand side bar tab to view brand table.
2. If Admin
3. wants to add new brand, click on Add new button.
4. Enter brand name on modal and click save button.
5. If Admin wants to edit brand, click on Edit button and reenter the brand name and save it.
6. If Admin wants to delete brand, click on Delete button and confirm delete.

6.3.1.5 Manage Models

The following figure shows the table of vehicle models in the admin section.

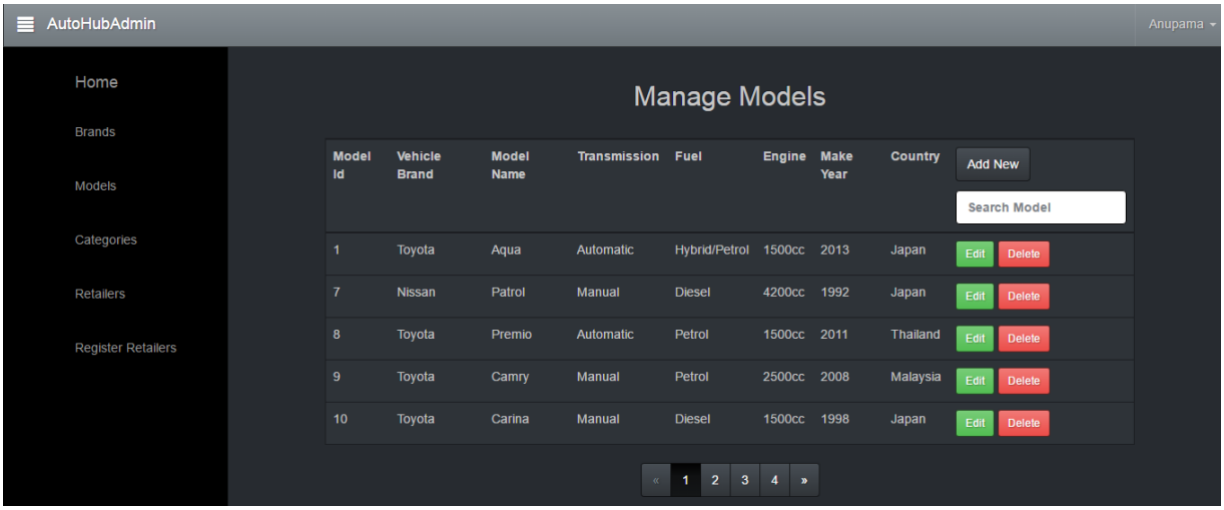


FIGURE 6.3.5 MANAGE MODELS

If Admin wants to manage models in the inventory, click on Models tab and view the Models table. Admin can add/edit/delete vehicle models through this view by clicking Add New, Edit and Delete buttons.

6.3.1.6 Manage Categories

Following figure shows the manage vehicle categories tab.

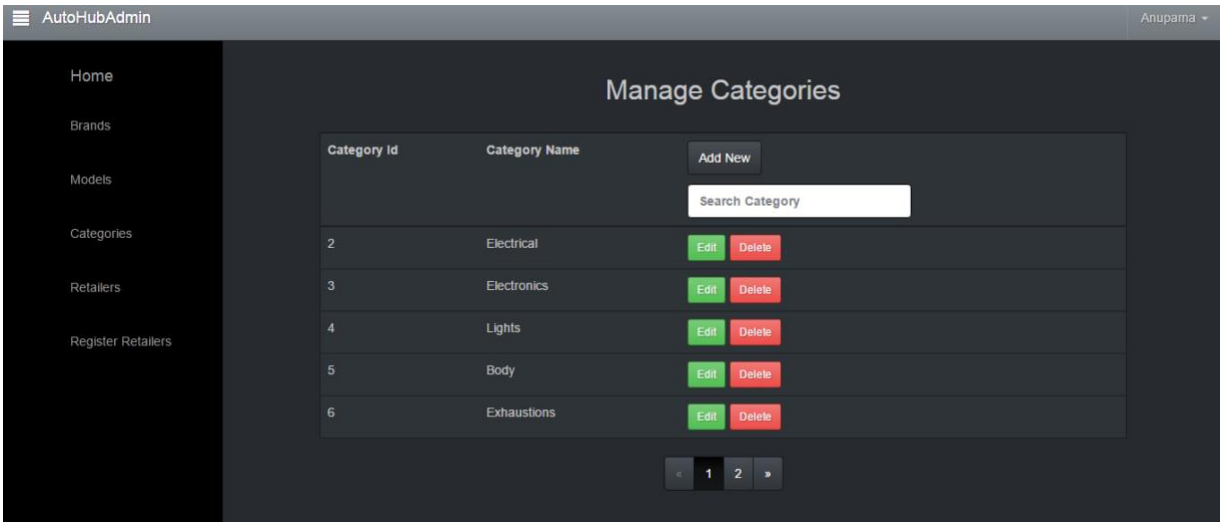


FIGURE 6.3.6 MANAGE CATEGORIES

If Admin wants to manage categories in the inventory, click on Models tab and view the Models table. Admin can add/edit/delete vehicle models through this view by clicking add new, edit and delete buttons

6.3.1.7 Retailer view

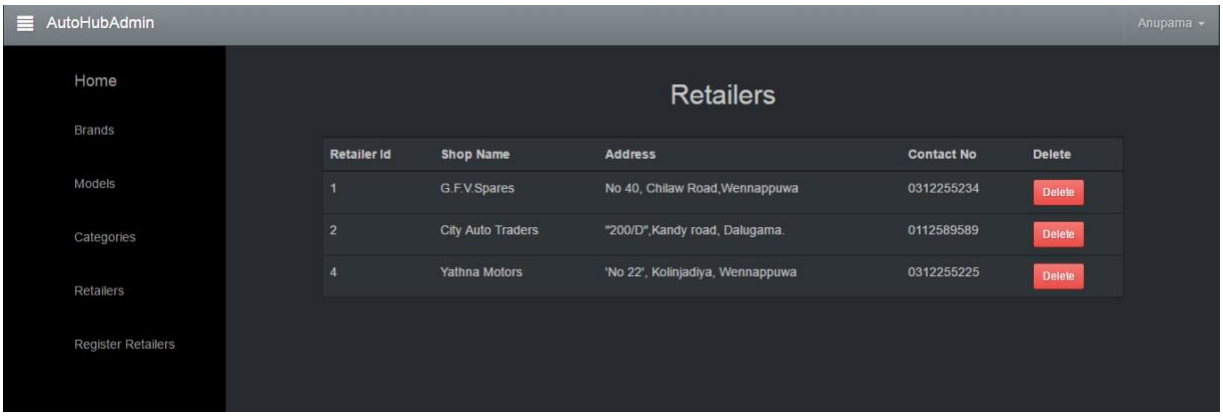


FIGURE 6.3.7 RETAILERS VIEW

Admin can view existing retailers through this view they can also remove a retailer from the system.

6.3.1.8 Retailer registration view

Admin can register a new retailer to the AutoHub system through this view. Following is the form used to add new retailer.

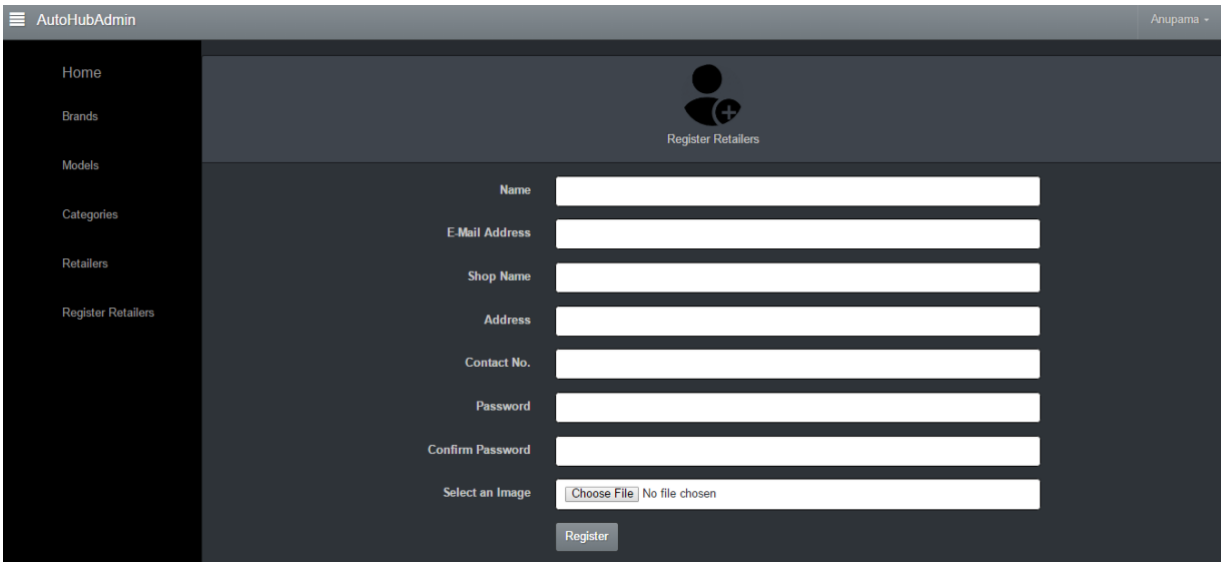


FIGURE 6.3.8 - REGISTER RETAILERS

Steps:

1. Click on Register Retailer tab.
2. Fill out retailer's details.
3. Click Register button

6.3.2 Retailer Side

6.3.2.1 Retailer Dashboard view

Following figure shows the dashboard of the retailer every retailer is having a separate dashboard for them.

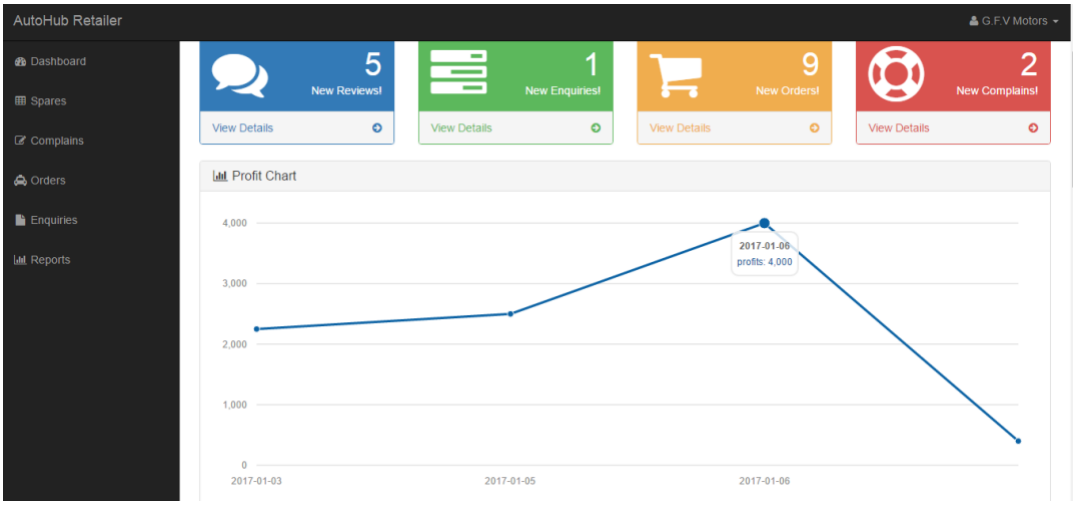


FIGURE 6.3.9 - RETAILER DASHBOARD-SALES CHART

After a retailer make a successful login system will redirect the retailer to their own dashboard. Dashboard is included with a summary of enquiries, orders, complains and reviews that belong to a particular retailer. It also shows the number of graphs which shows the profit, sales, orders and order item status that are related to logged in retailer

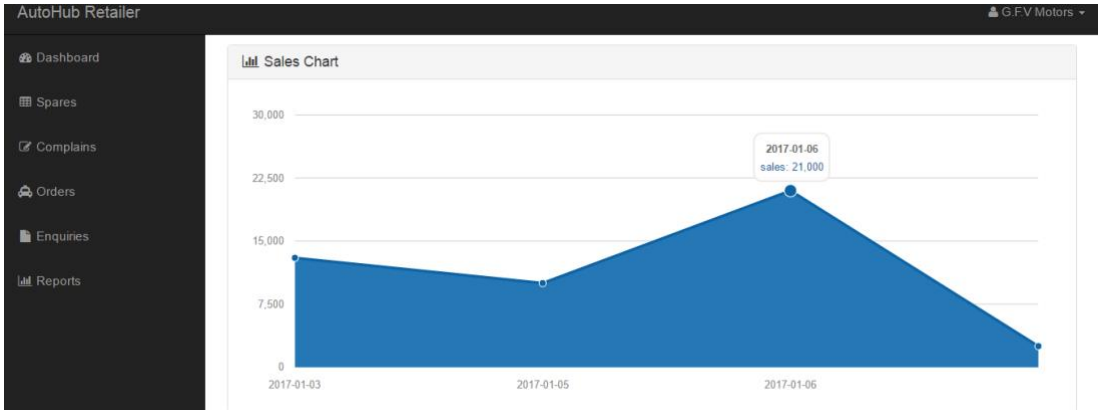


FIGURE 6.3.10 RETAILER DASHBOARD-PROFIT CHART

Following graph shows the total number of orders that are related to a particular retailer with dates.

FIGURE 6.3.11 RETAILER DASHBOARD-ORDERS CHART

The following show the current order item status and most recent orders that are related to a particular retailer.

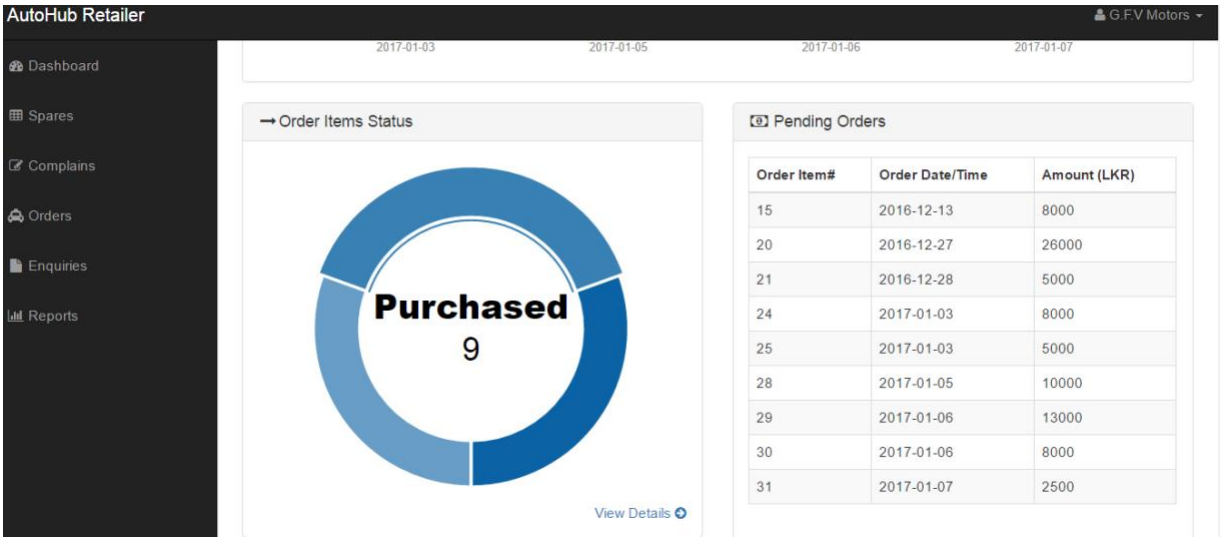


FIGURE 6.3.12 RETAILER DASHBOARD-ORDER ITEMS CHART & PENDING ORDERS

6.3.2.2 Manage Spares view

If retailer wants to manage spare item in their inventory, click on spares tab on the side bar and view spare table.

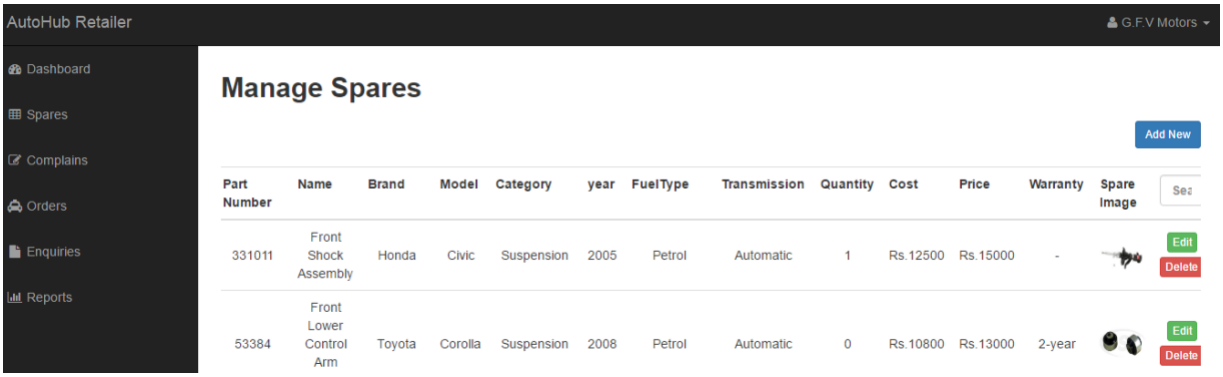


FIGURE 6.3.13 MANAGE SPARES

If retailer wants to delete spare details, click on Delete button and ok the delete confirmation message.

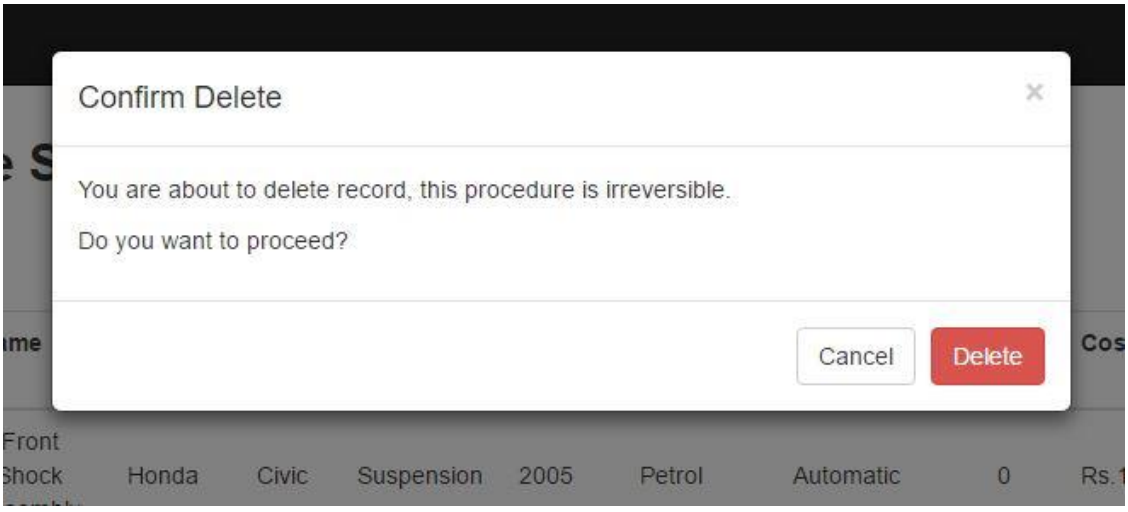


FIGURE 6.3.14 DELETE SPARE MODAL

If the retailer needs to add a spare to the system, he need to enter the information through the following form.

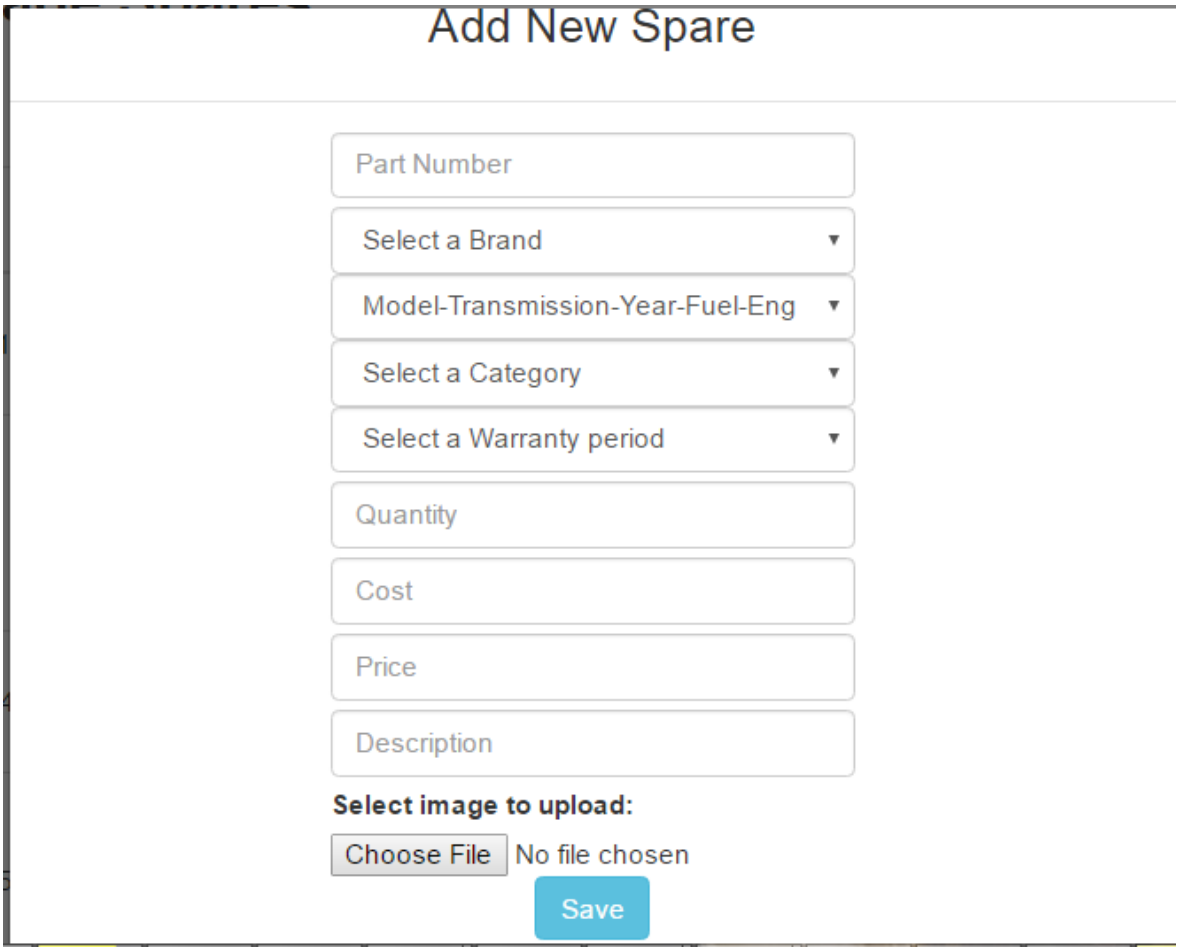


FIGURE 6.3.15 – ADD NEW SPARE MODAL

Steps:

1. If retailer wants to add new spare to the inventory, click on Add New button.
2. Fill the details of new spare.
3. Click on save button.

If retailer wants to edit spare details, click on Edit button and edit the spare details.

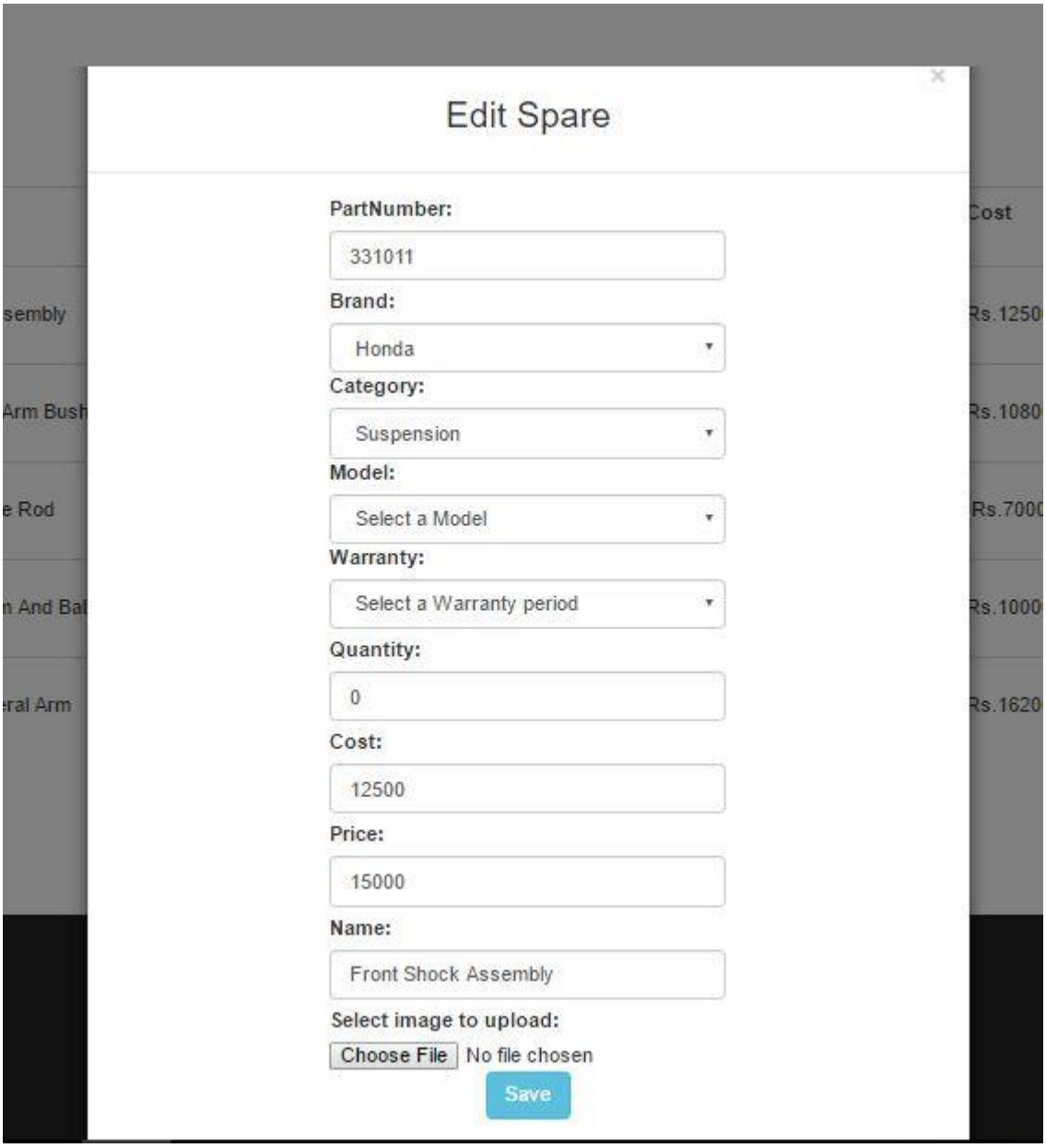


FIGURE 6.3.16 EDIT SPARE MODAL

6.3.2.3 Manage Orders view

Retailer can view an existing order that belongs to him. He can also update the order item status so that retailer can know the current order status.

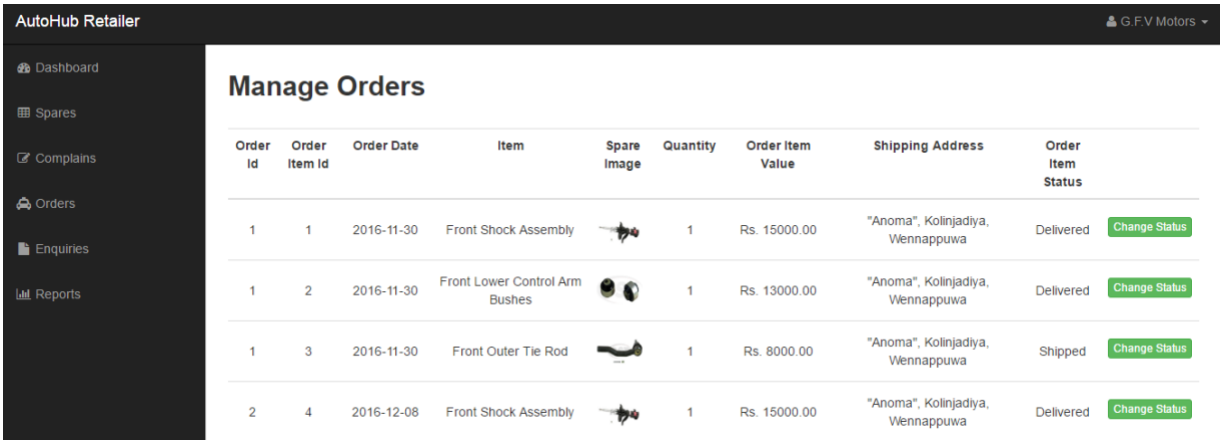


FIGURE 6.3.17 - MANAGE ORDERS

Steps:

1. Click on orders tab and view order details of particular retailer.
2. If retailer wants to update order status click on Order Status button.
3. Then update status as shipped or delivered according to order item.
4. Click save button.

Retailer can change a purchased item status through following form.

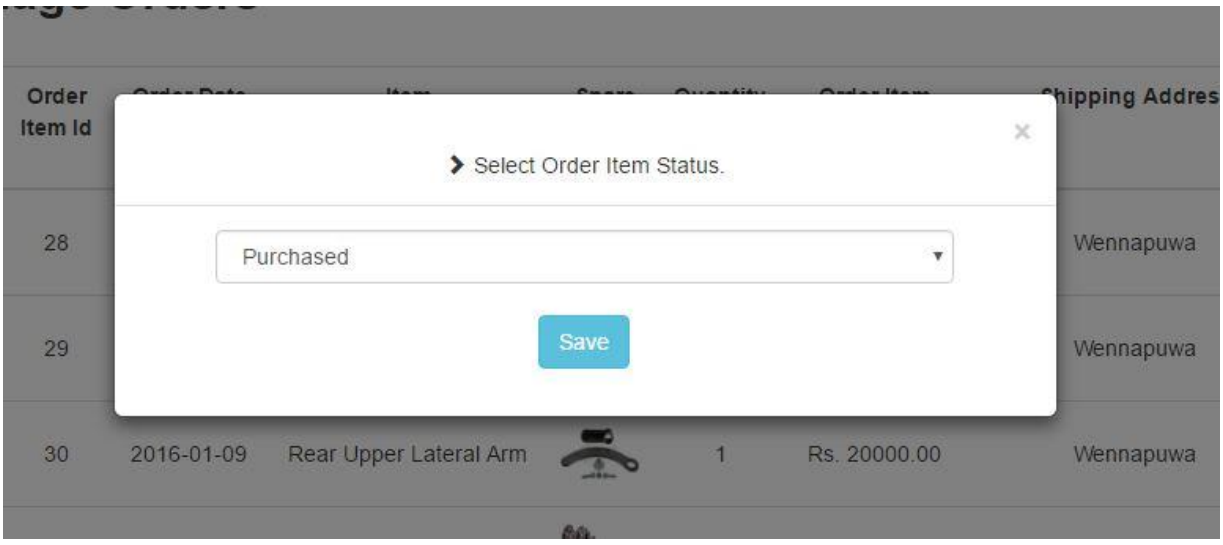


FIGURE 6.3.17 ORDER STATUS UPDATE MODAL

6.3.2.4 Manage Enquiries

Retailer can view the enquiries that have been placed by a customer. He can also reply back to the customer if he willing to do so.

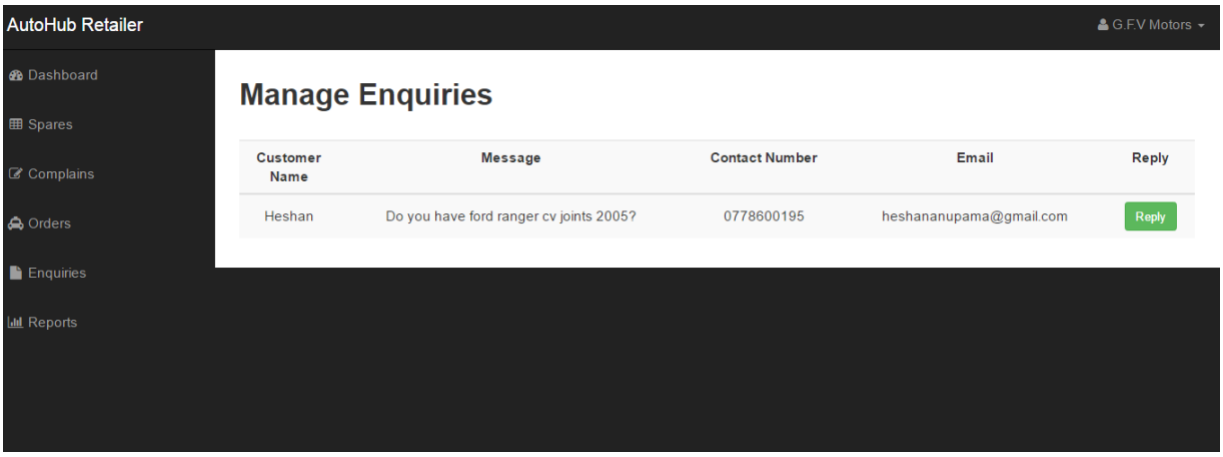


FIGURE 6.3.18 MANAGE ENQUIRIES

6.3.2.5 Manage Complains

Retailer can view the complains that have been placed by a customer.

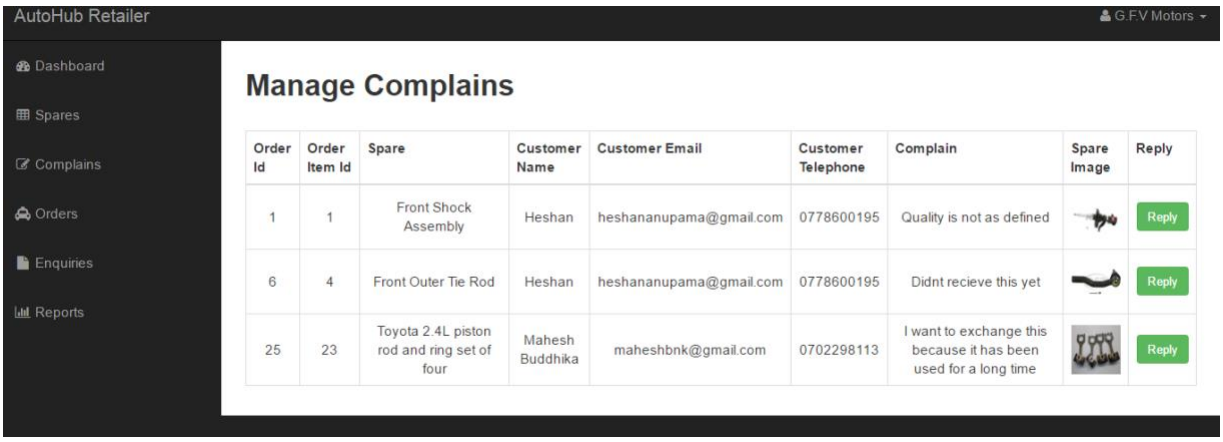


FIGURE 6.3.19 MANAGE COMPLAINS

He can also reply back to the customer and these messages will be shown at the relevant customer's inbox.

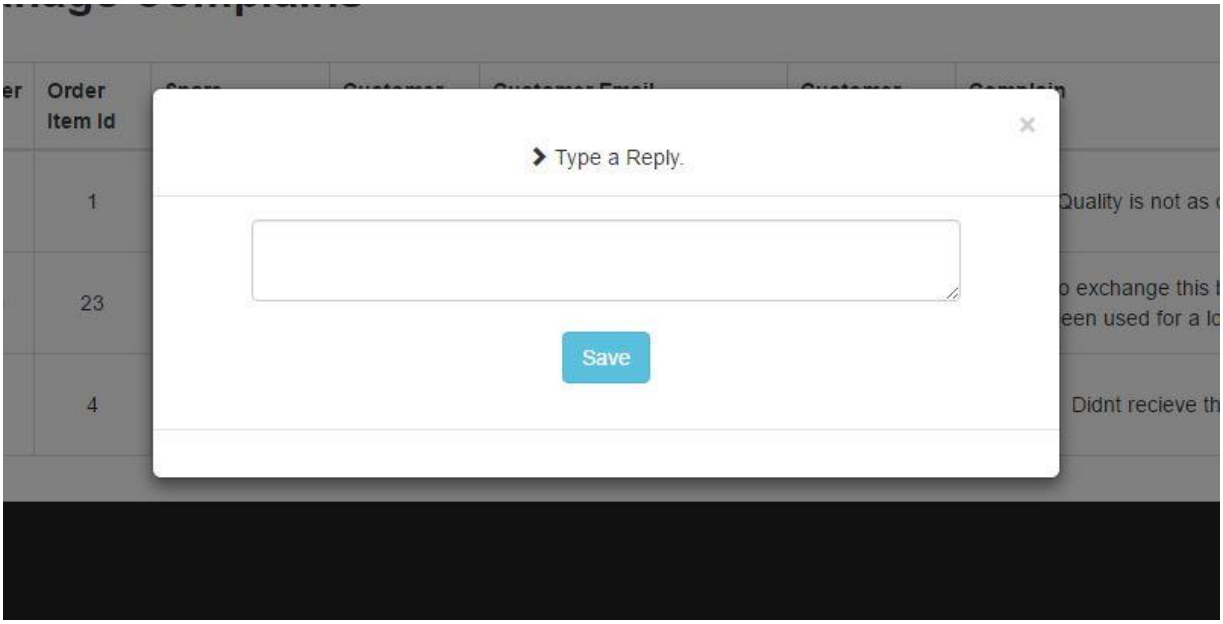


FIGURE 6.3.20 MAKE REPLY TO COMPLAIN

6.3.2.6 Report Generation

Retailer can generate different kinds of reports as sales, orders, profits and inventory based on different generating frequencies as daily, weekly monthly and yearly.

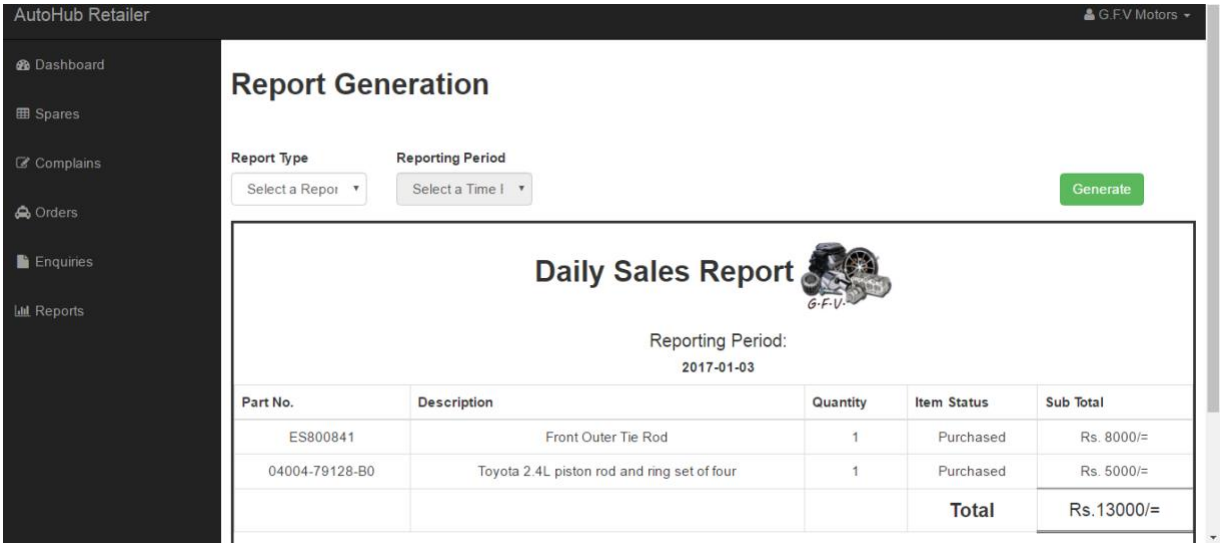


FIGURE 6.3.21 DAILY SALES REPORT

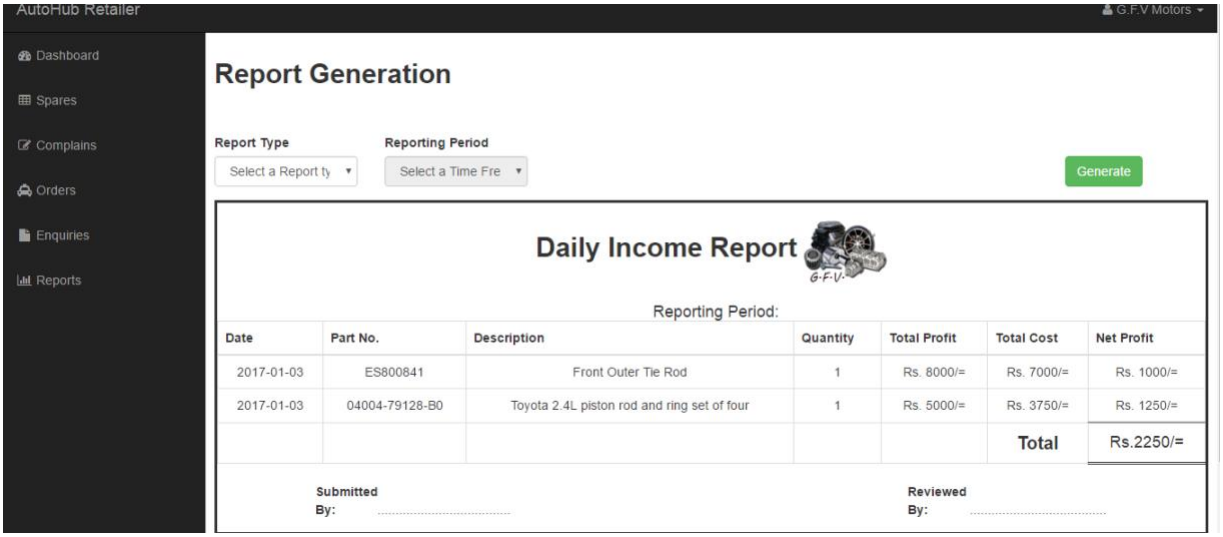


FIGURE 6.3.22 DAILY INCOME REPORT

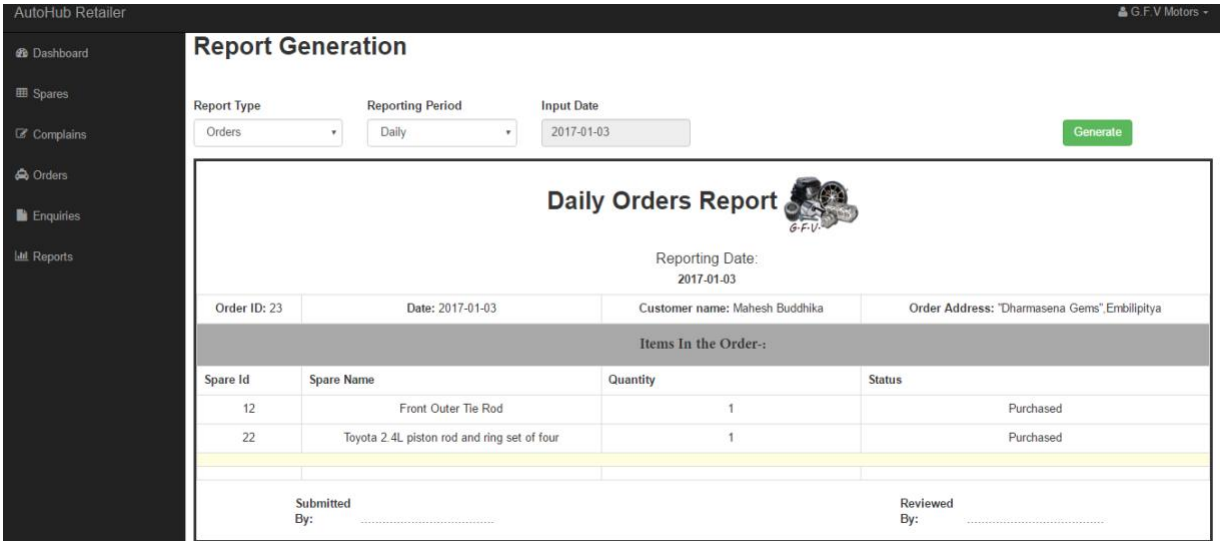


FIGURE 6.3.23 DAILY ORDERS REPORT

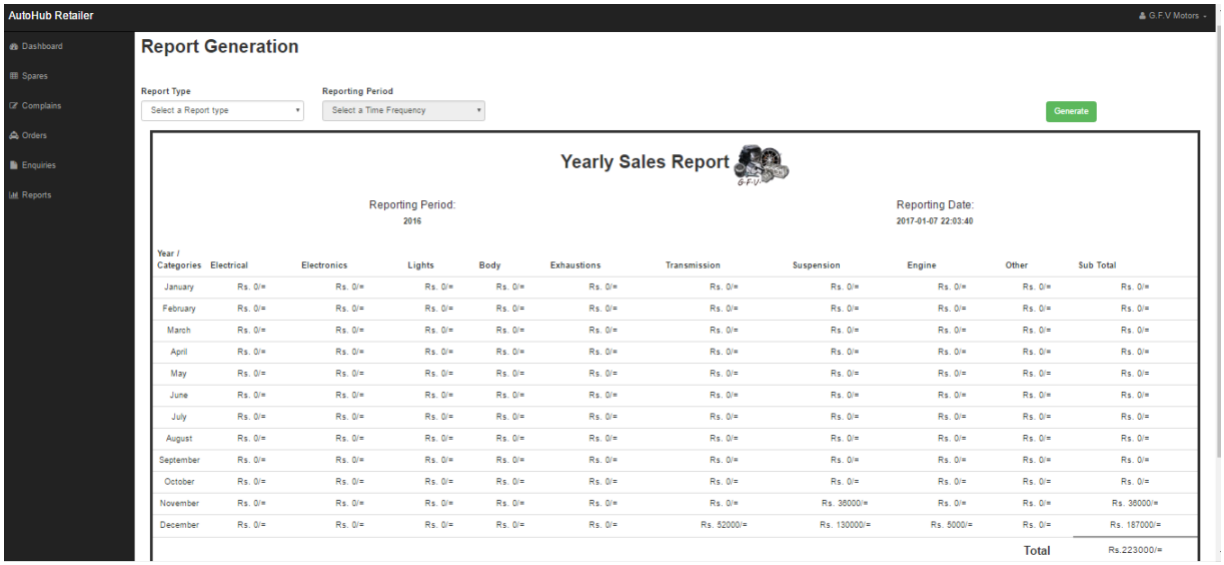


FIGURE 6.3.24 YEARLY SALES REPORT

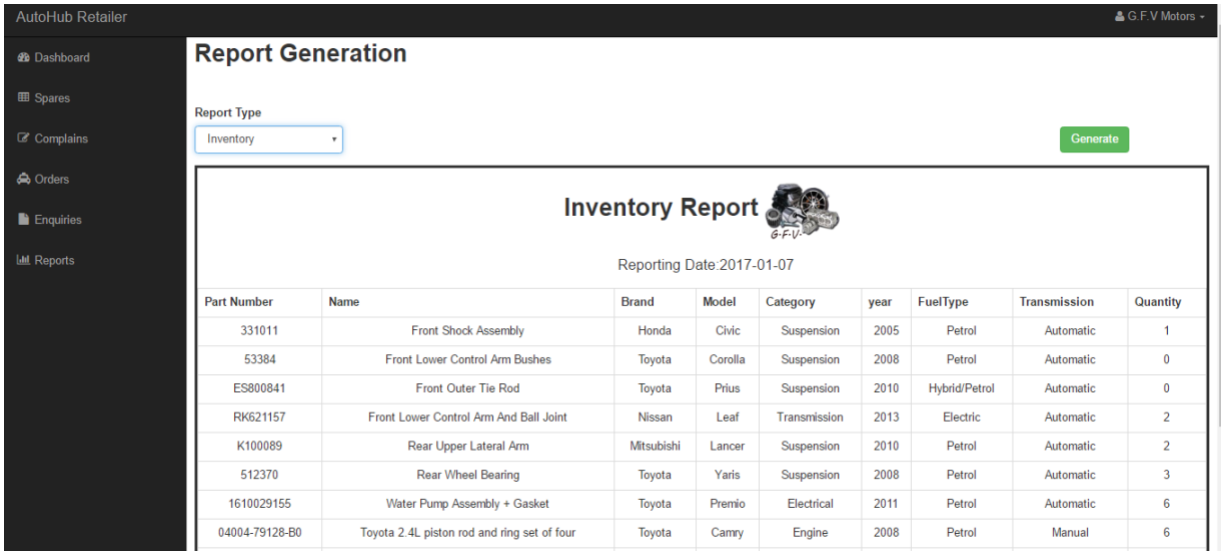


FIGURE 6.3.25 INVENTORY REPORT

6.3.2.7 Analytical Charts

This view, it consists of charts with last year's profits, sales with related to different vehicle brand and spare Categories. User can access to this section using 'Charts' option in the sidebar.

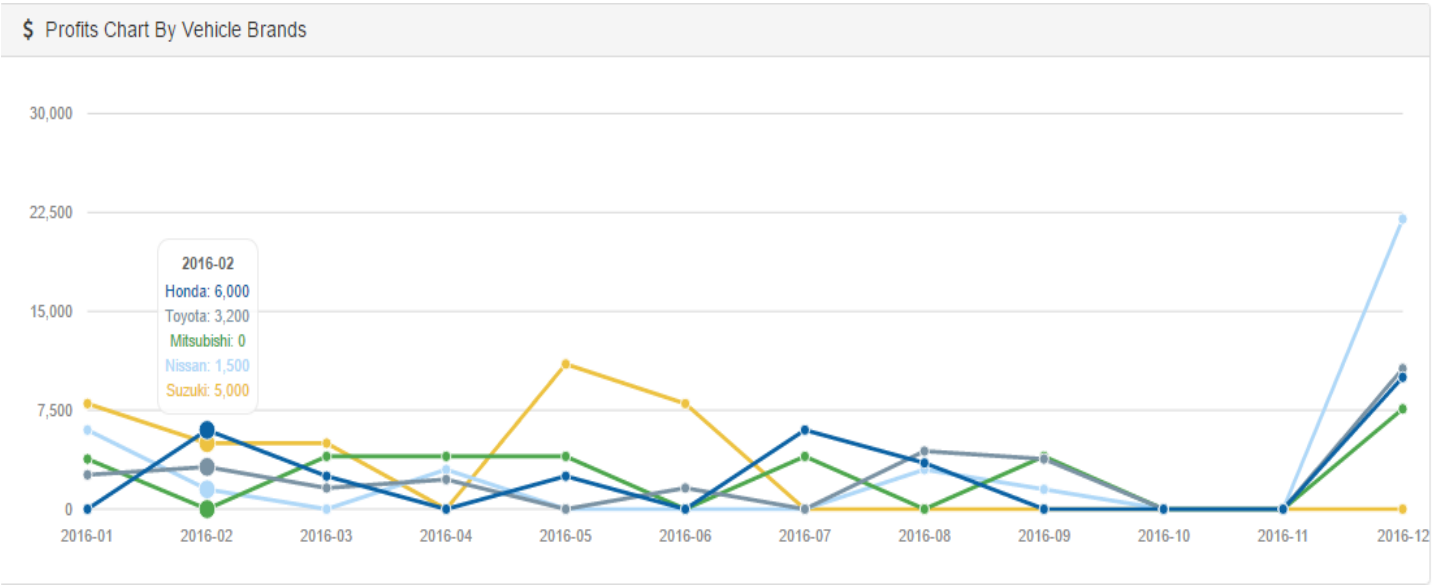


FIGURE 6.3.26 PROFIT BY VEHICLE BRAND

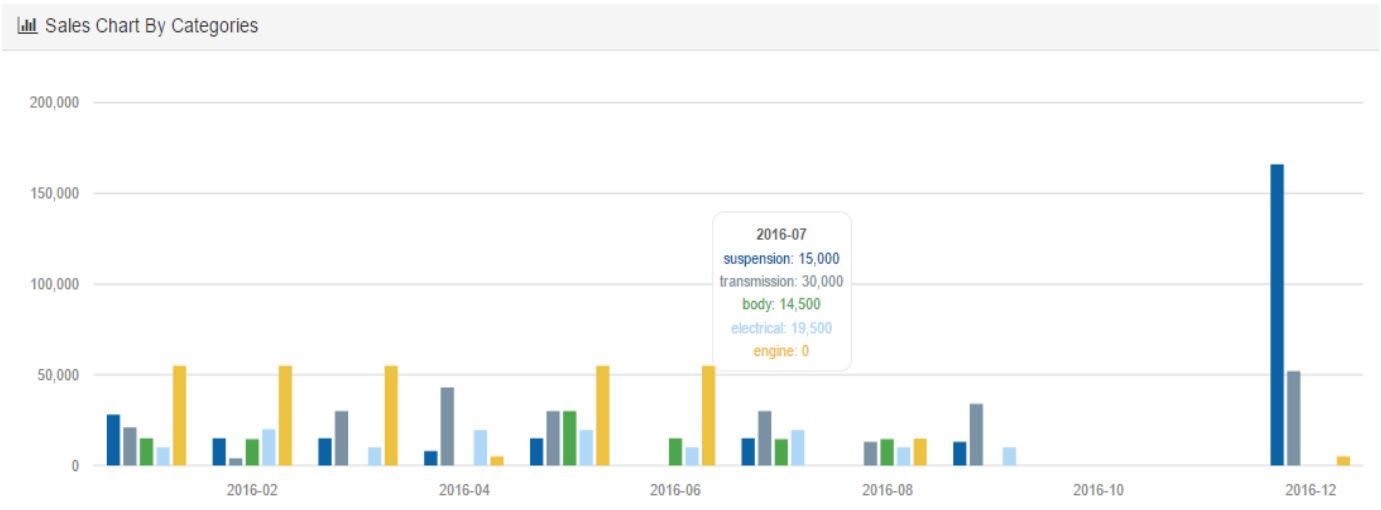


FIGURE 6.3.27 SALES BY SPARE CATEGORIES

6.3.3 Customer Side

6.3.3.1 Customer Home Page

Following figure shows the customer side home page.

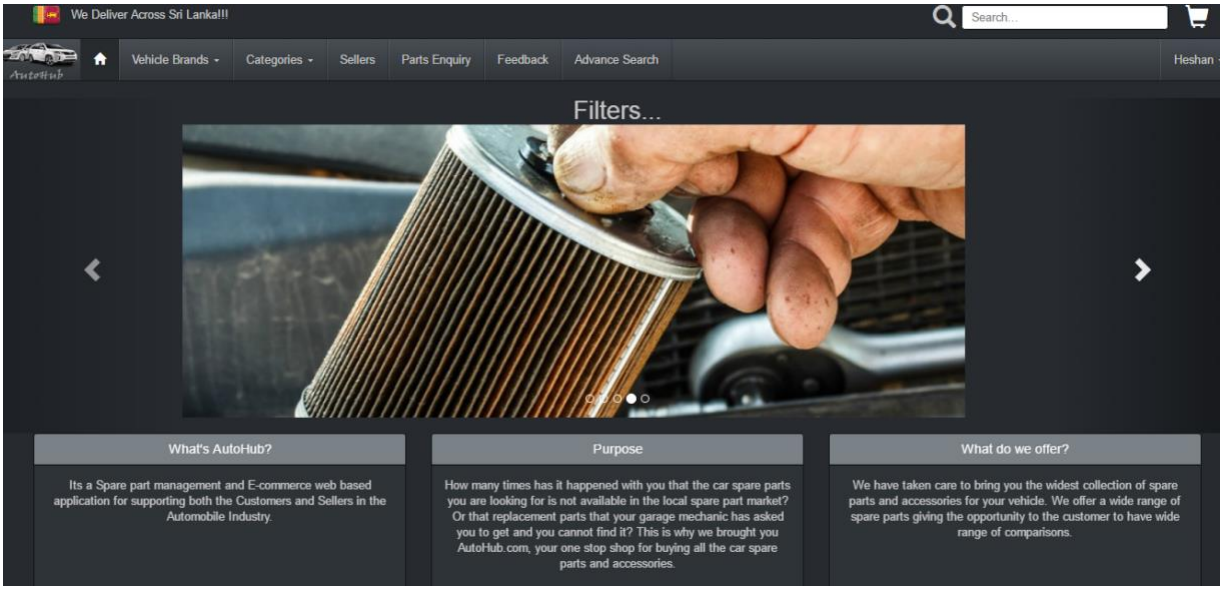


FIGURE 0.1 CUSTOMER HOMEPAGE

After a customer make a successful login system will redirect this page however customer can go to this page without login. But they couldn't access some of the services. They can navigate to different services with the shortcuts provided in the above view.

6.3.3.2 Enquiries

Following figure shows the form that is used by a customer to make enquiries.

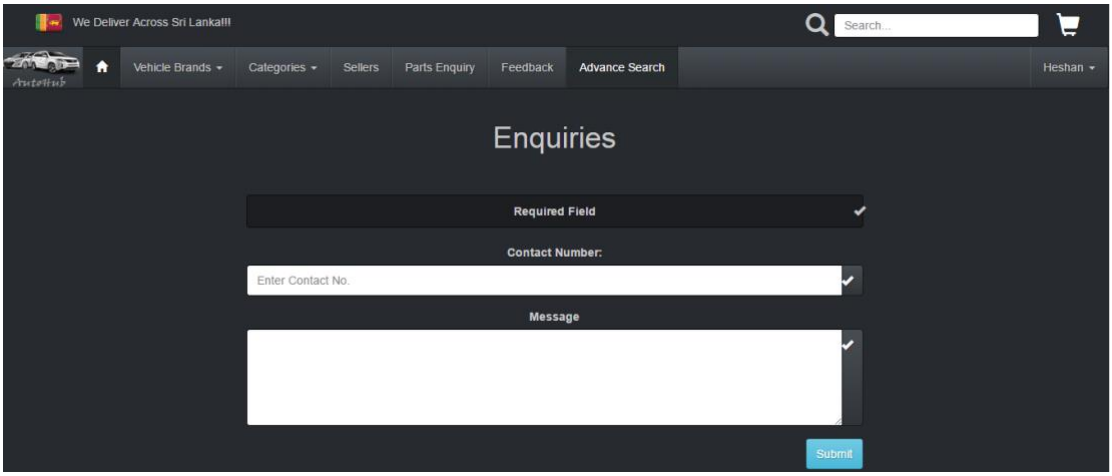


FIGURE 0.2 ENQUIRIES

Customer need to login in order to place an enquiry After a customer make a successful login system will redirect this page however customer can go to this page without log in. But they couldn't access some of the services.

6.3.3.3 Browse Spares

Customer can search for the spares that he is looking for using any kind of keyword. System will search by the part number or by the spare name.

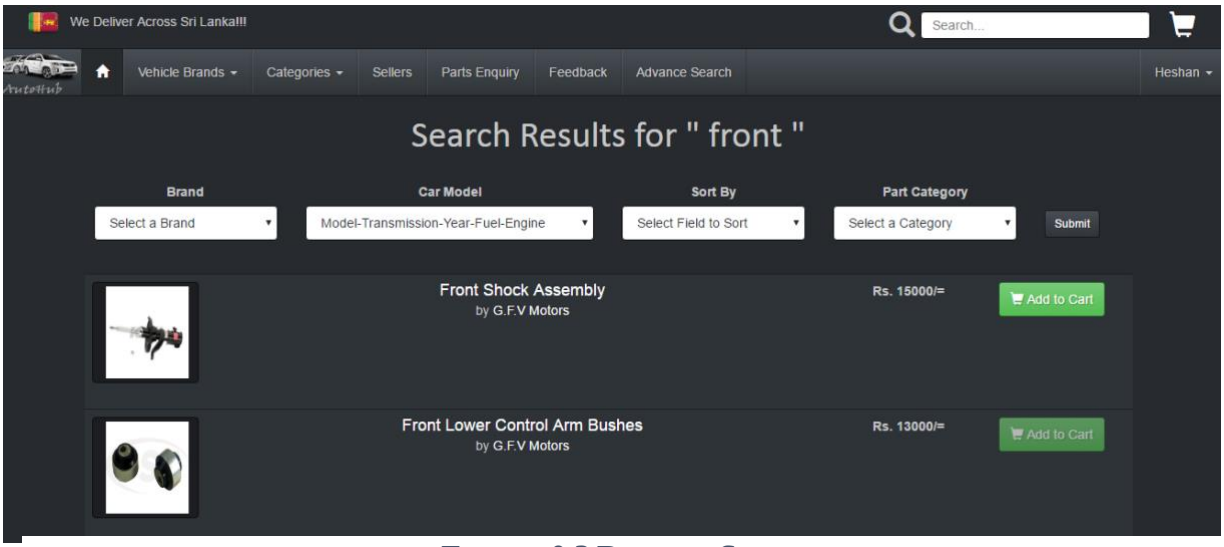


FIGURE 0.3 BROWSE SPARES

If he doesn't find the necessary spares, then he can search further by applying different filters as below.

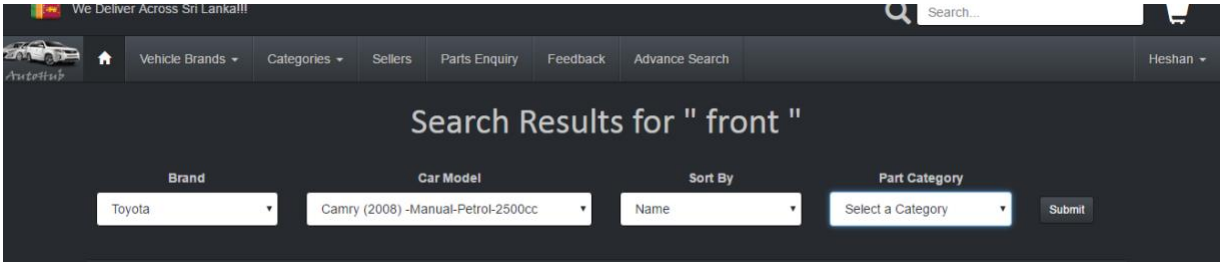


FIGURE 0.4 BROWSING PANEL

6.3.3.4 Advanced Search

Following figure shows the advanced searching function of the system.

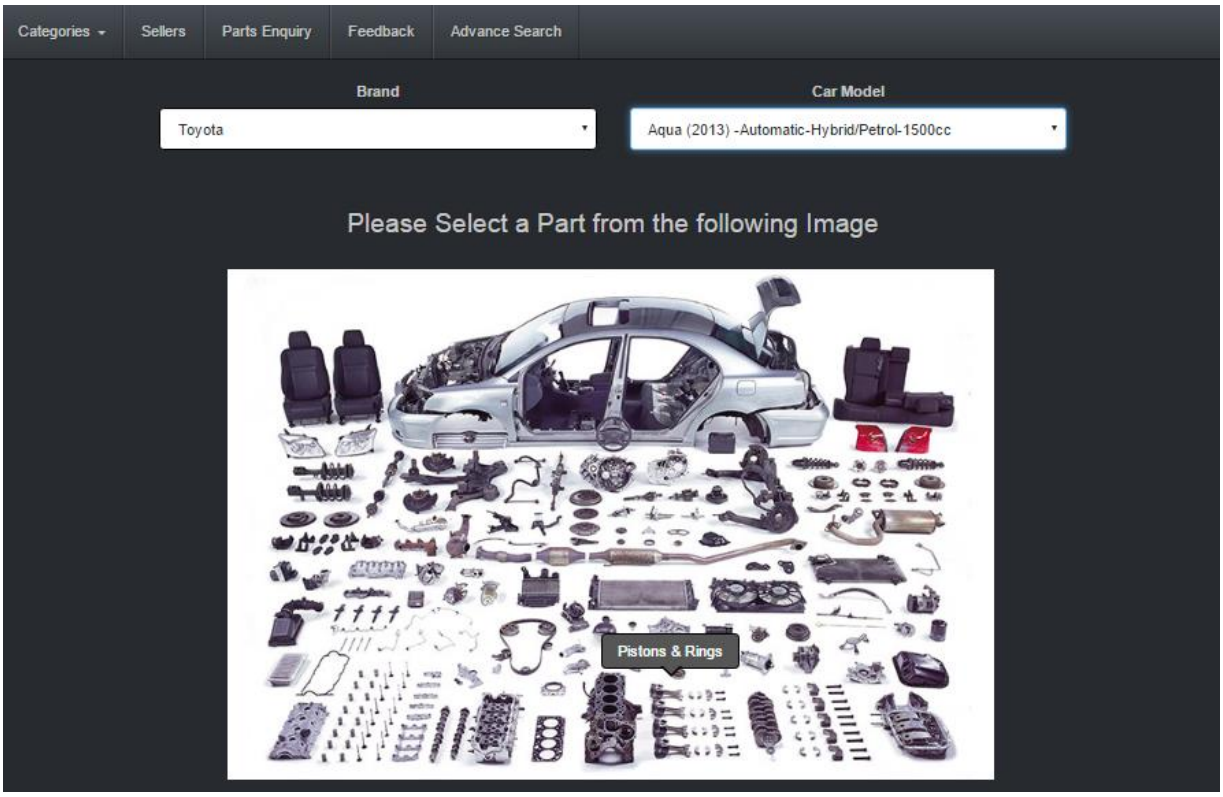


FIGURE 0.5 ADVANCE SEARCH

This is one of the main feature of the system a user can select a car model that he owns and click on any part on the image. Following image show 'Pistons & Rings' mouse over when the user click there it will automatically search the items

6.3.3.5 Product Details

If customer click on name or image, system will redirect to the product more info page.

Steps:

1. Enter quantity that required.
2. If customer wants to add to shopping cart, click on add to shopping cart button
3. If customer wants to continue shopping, click on continue shopping button.

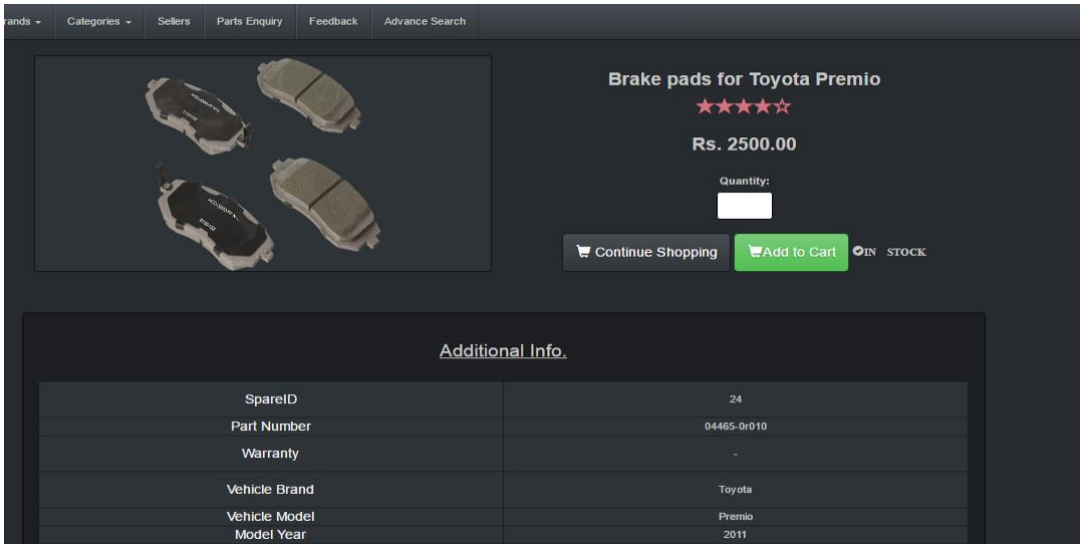


FIGURE 0.7 PRODUCT DETAILS

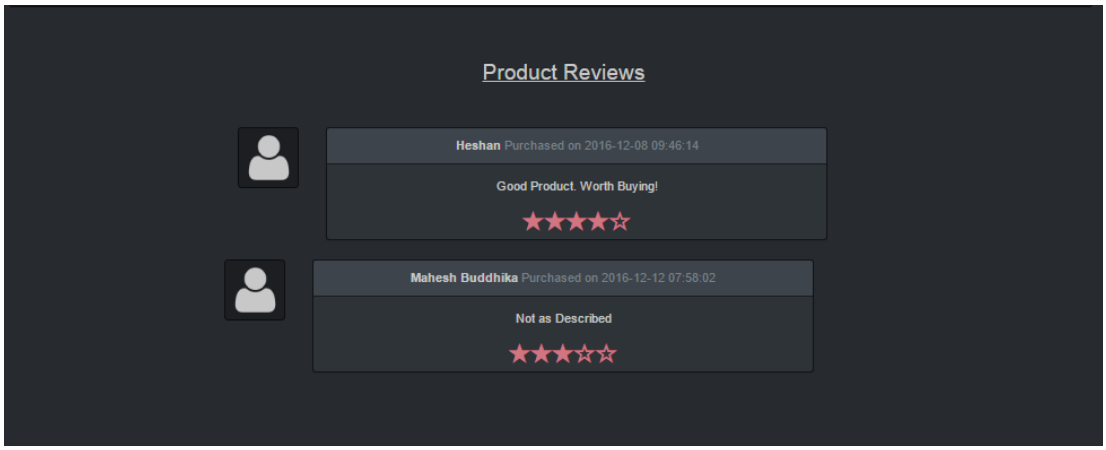
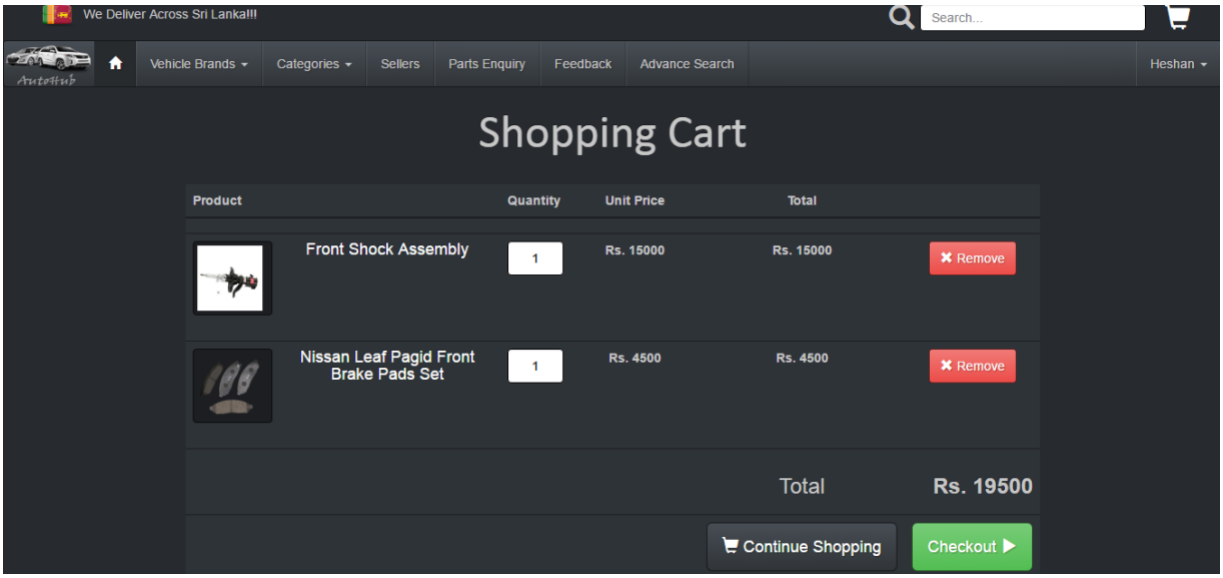


FIGURE 0.6 PRODUCT DETAILS-REVIEWS

User can add reviews after you purchase a product. Then those reviews will be shown in relevant item's product detail section so that another customer can review when purchasing an item.

6.3.3.6 Shopping Cart

After customers added items to the cart they can view the shopping cart which shows all the items alone with quantities and totals. If customer wants to delete cart item, click on remove button.



6.3.3.7 Checkout

After a customer clicks on the checkout button system will direct to the following view.

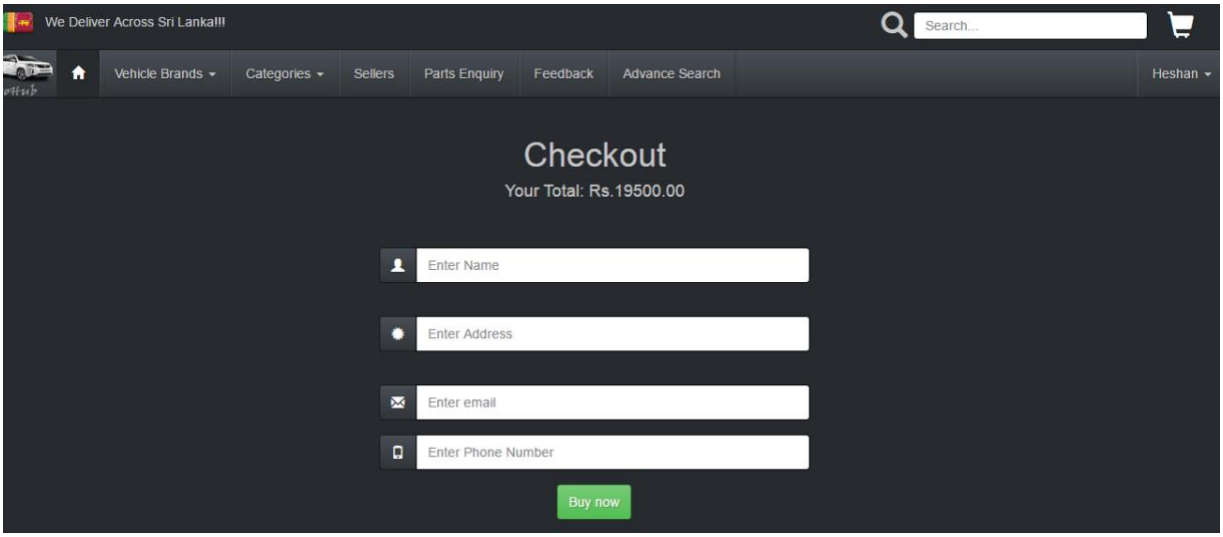


FIGURE 0.8 CHECKOUT

Customer can proceed to PayPal upon the clicking of 'Buy now' button.

6.3.3.8 Payment

If a customer confirms the purchase, system will redirect to the following PayPal page. Customers can then continue the payment from their PayPal accounts.

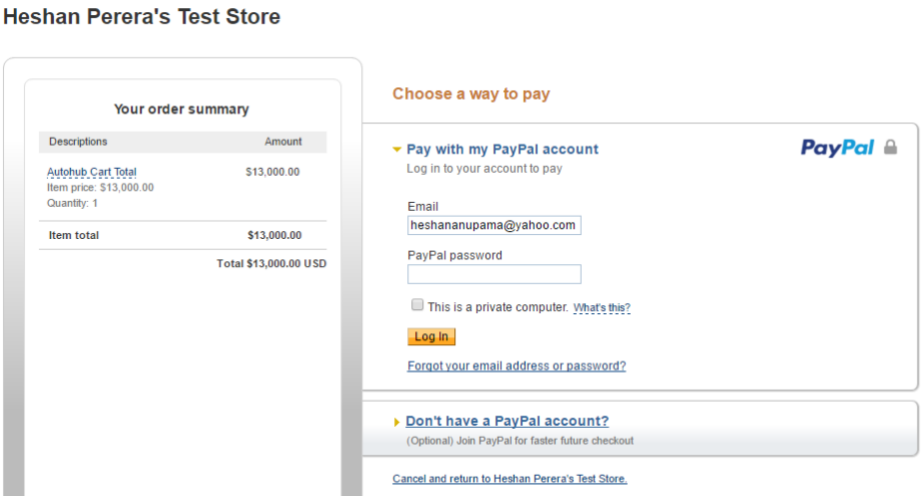


FIGURE 0.9 PAYMENT

6.3.3.9 Feedback

Following figure shows the feedback view of the system.

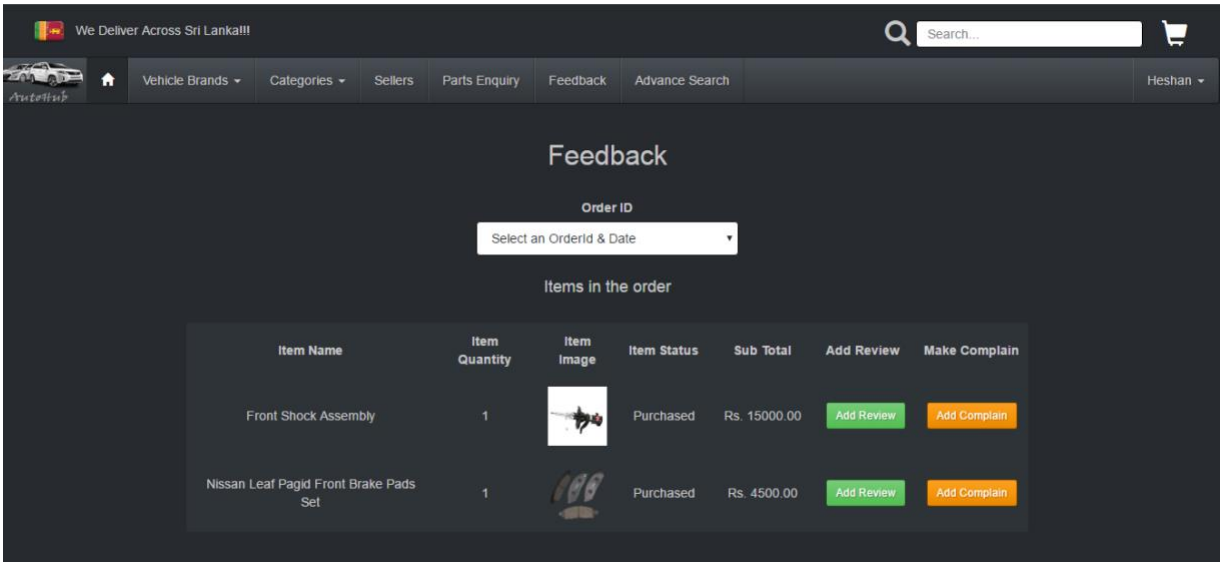


FIGURE 0.10 FEEDBACKS

Customer can add feedbacks to the items that he had purchased. It can be either a complaint or a review. Complains are shown to the relevant retailers so that they can resolve it as soon as possible.

Reviews

Customer can add review to a purchased item. But a review can be placed only once for an order item.

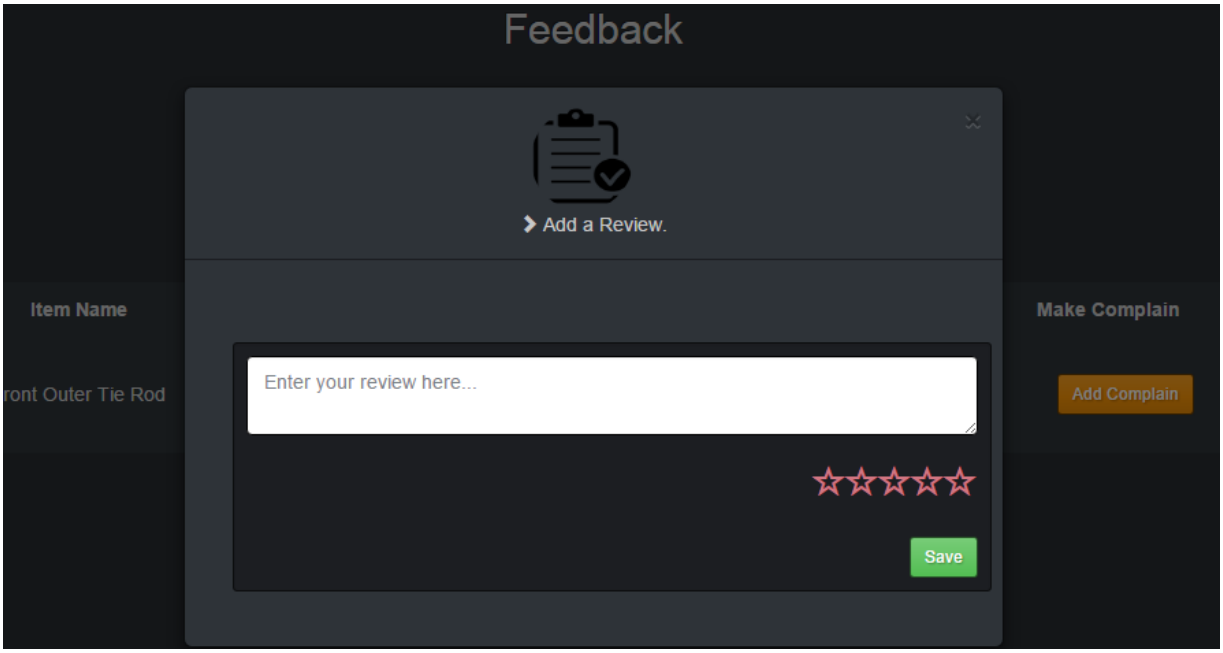


FIGURE 0.11 REVIEW MODAL

Complains

Customer can place a complain if he finds any defects of the purchased item.

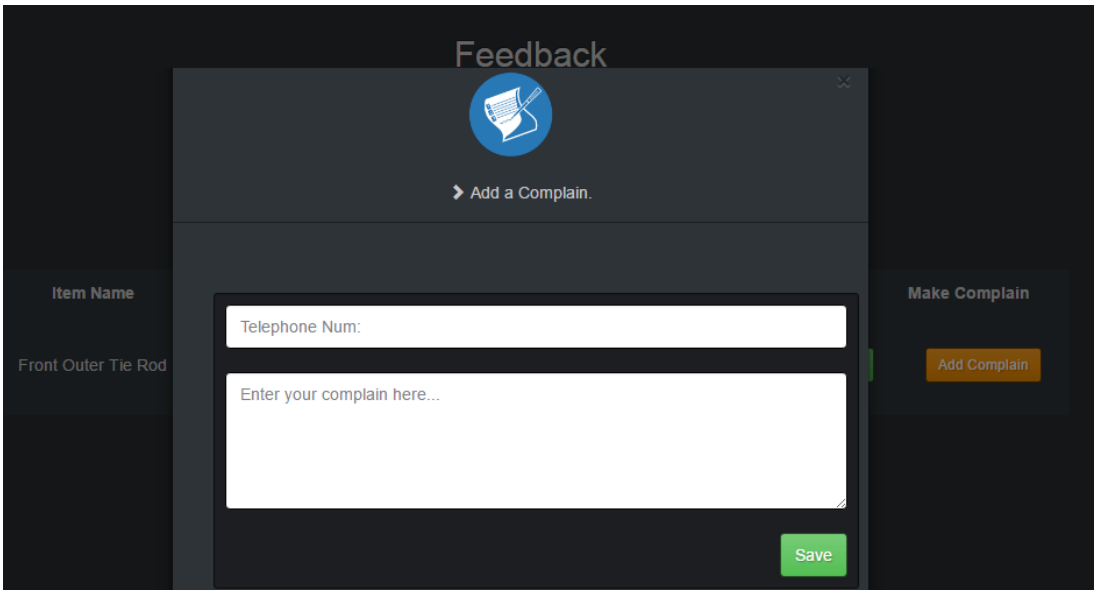


FIGURE 0.12 COMPLAIN

6.3.3.10 Inbox

Following figure shows the inbox of a particular customer. Each customer is having an unique inbox.

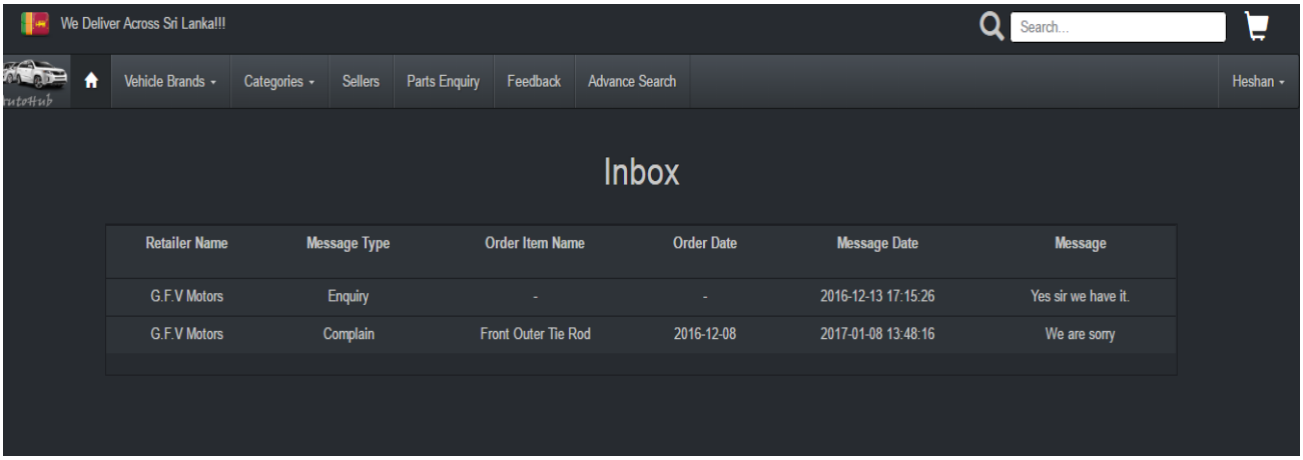


FIGURE 0.13 CUSTOMER INBOX

Logged in customer can view the messages that a particular retailer had sent to him using this view. Message could be two types either a reply to an enquiry or a reply to a complain.

Chapter 7

7 Conclusion and Discussion

This chapter is included with the degree of objectives met after implementing the project, usability of System, Drawbacks and Limitations along with the further development of the project.

7.1 Degree of objectives met

For the Customer -:

Enabling customer to make search on the item by himself via the website.

With just a few steps customer can search a particular spare via many different flexible searching methods. Along with advanced search ability where a user can click on an image of spare parts to search by just a click.

Ability to make enquiries.

Customers can use the service to enquire about the status and information about spare parts if they are unable to find spares by searching and the retailer can reply the customer with information that he need.

Having the facility to make complains.

It's always a common defect in an e-commerce website where customers didn't receive the products, product received may be changed or not in the condition as described. So, clarify this sort of problems customer can easily file a complaint against an order which will be redirected back to the particular retailer.

Enabling customers to make ratings and reviews on the purchased products.

This is one of the main benefit of the new proposed system. This will help the customers to decide on buying a certain product from a certain retailer. There will be a certain rating for the retailers calculated by previous customer added ratings and reviews and from it whenever a customer buys a product he can verify whether the seller has a good rating by reading the reviews and ratings that have already placed.

Customers can compare spare parts with different retailers.

As the website is interconnected with number of retailers, for a one spare part there can be number of retailers. This allow the customer to compare and contrast between different retailers.

For the retailer: -

Face Higher competition.

As there are many retailers bought together to a single marketplace through this website there can be many sellers offering same spares which will eventually lead to a competition among retailers where every retailer would try to do his/her best to attract more and more customers.

Decrease in the inventory costs.

Not like a real-time purchasing retailers do not need to keep higher quantities of the spare parts in their stocks/inventories where their costs of inventory will go down.

Generation of reports and charts.

Almost in every spare shop they are using manual records whenever they need to analyze the sales during a particular time period. But this system is supposed to have a report generation section where all the details of the sales are taken in to consideration and reports will be generated accordingly. Retailers also can generate different charts in which the management could use those to take strategic decisions.

Take strategic decisions.

By analyzing charts drawn according to different categories and brands, management could take strategic decisions in order to the future success of the business.

7.2 Usability of the System

7.2.1 Compatibility with Heuristic Evaluation Guidelines

Navigation

All the pages contain navigation bar where the navigational path is indicated with link buttons. This Primary navigation is located as highly visible to users providing user guidance. Through main menu any user can navigate across the web pages which he is given the access and go back to the previous page visited, or use the sign out link button to sign out from the system at any given time. This helps to minimizing user waiting time.

Content, Form design and Language

With the objective of minimizing user's memory load, interfaces were created as simple, efficient and avoiding information overload. Text style and sizes are in a very simply readable format. Background content, interfaces were made by using simple colors and styles according to system theme colors.

Use Meaningful links

Rather than using generic words such as ―Click here, System has differentiated links by using terms related to place where user is redirected to.

Consistency & Standards

System maintain the consistency throughout the project using standard styles through all pages avoiding surprising users.

User Authentication

If password is forgotten user can reset new password.

Error prevention

Mostly select box have been used as the user doesn’t have to type anything and there are no any validations required while passing data through select boxes. Before sending input boxes data via network requests the data have been validated through html5 and laravel data requests. Regex have also been used to validate the user inputs.

Help users recognize, diagnose, and recover from errors

Error messages are expressed in simple language that precisely define the problem and simply suggest a solution that can be used to resolve this issue.

7.3 Limitations and Drawbacks

Difficulty in hosting the web application

Web application couldn't be hosted as it is cost to buy a domain name and host the web application. Due to this fact system currently runs on localhost powered by apache.

Difficulty of using the PayPal API for the real Accounts

When developing the online payments in the application currently have used sandbox Buyer and Merchant Account. By doing so we couldn't get the full functionality of PayPal API such as providing notifications back to the web application.

Difficulty in providing free deliveries

Retailers don't agree to provide free deliveries as it cost more to deliver the product. Due to this fact customers need to pay the courier the delivery chargers.

Handling returns and damage items.

Most of the retailers will not accept the returns without a hesitation. They try to convince the customers that there is no any error from their side.

7.4 Future Modifications, improvements and extensions possible

- Hosting the web application in a server.
- Chat function, among different customers and retailers have to be implemented
- Provide an android application to the customers.
- SMS notifications when purchasing products and the retailer delivers the product could be implemented.
- Development of the advanced search option in to a 3D model with the 360-degree rotation.
- Enhancing system into a more responsive an compatible web while accessing through mobiles.

7.5 Conclusion

As the final conclusion, from scratch to the very end the project has kept me occupied in an immense learning. From the system analysis phase to the system implementation everything that has been learnt had provided a very broad edge in a software development scenario which I believe will be beneficial for me in future. Especially the chance I got to work and learn in an e-commerce shopping environment has added more value to the development experience.

I have learnt that AutoHub has an effective service to its audience- a mix of both retailers and customers. Though there are some limitations that I had mentioned earlier in the chapter in proximity, I think that AutoHub is an effective web application to make the shopping of vehicle spare parts in a more convince and attractive manner.

Apart from the knowledge I have obtained in working in an e-commerce shopping system, the utilization of other tools, technologies and programming languages had also included a profound learning curve which has been a precious experience for me to preserve for the future benefit.

8 References

- ✓ Laravel documentation - https://laravel.com/docs/5.3
- ✓ Bootstrap tutorials - http://www.bootsnip.com
- ✓ Stack Overflow – Question and answer site for professional and enthusiast programmers http://www.stackoverflow.com/
- ✓ Laravel.io-Forums on laravel www.laravel.io
- ✓ http://youtube.com