

Car Trading Information Management System

A project

Submitted to the Department of Computer Science and Engineering Bangladesh University Of Business And Technology (BUBT), Dhaka In partial fulfillment of the requirements for the degree of

**BACHELOR OF SCIENCE
IN
COMPUTER SCIENCE AND ENGINEERING
SEMESTER: Fall 2018-19**

SUBMITTED BY:

14152103082 Prodipta Promit Mukherjee

14152103003 Minia Rahman Bristy

14152103030 Sanchita Paul

SUPERVISED BY:

Mr. Milon Biswas

Assistant Professor

Department of CSE

Bangladesh University of Business and Technology (BUBT)

BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY (BUBT)
RUPNAGAR, MIRPUR-2, DHAKA-1216 , BANGLADESH

May 2019

Certificate

This is to certify that this is a bonafide record of the project presented by the students whose names are given below in partial fulfilment of the requirements of the degree of Bachelor of Science in Computer Science and Engineering.

ID No	Names of Students
14152103082	Prodipta Promit Mukherjee
14152103003	Minia Rahman Bristy
14152103030	Sanchita Paul

Mr. Milon Biswas

Supervisor

Assistant Professor

Department of CSE, BUBT.

Mr. Prof. Dr. M. Ameer Ali

Professor & Chairman

Department of CSE, BUBT.

DEDICATION

We dedicate this project to God Almighty our creator, our strong pillar, our source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout this program and on His wings only we have soared. We also dedicate this work to our parents who has encouraged us all the way and whose encouragement has made sure that we give it all it takes to finish that which we have started. We dedicate my dissertation work to my family and many friends. A special feeling of gratitude to our loving parents, whose words of encouragement and push for tenacity ring in my ears. We also dedicate this dissertation to my many friends who have supported me throughout the process.

APPROVAL

This **Project Report** Submitted by Prodipta Promit Mukherjee bearing ID No. 14152103082, Minia Rahman Bristy bearing ID No. 14152103003, Sanchita Paul bearing ID No. 14152103030 in partial fulfillment of Final Project Submission for BSc. Engg. in CSE degree has been examined and accepted for further process.

Approved:

Mr. Milon Biswas

Supervisor

Assistant Professor

Department of CSE, BUBT.

ABSTRACT

Car Trading is one of the largest business field in Bangladesh. Day by day the demand of personal car is growing and this field of business contributes huge in our economy. By this time, the trading becomes more competitive

At present, almost every car trader executing their business through traditional paper based system. The problem arise in executional time and cost is not optimized according to this modern competitive age. There are too many unusual expenses which occurring some tangible and intangible losses which is not accepted by trader. By following this current approach of this business field and modern world, we decided to make this following digital system solution for car traders. Our attempt is not only bounded to this system's development but also provide services in spite of maintaining this system with the progress of modern age.

Our goal is to provide an effective, efficient and satisfactory system and service solution to car traders to make their business more smart and rich. By following this, we are conscious to store and access information, safely and securely in remote machine so that the probability of destroying or steeling of traders valuable business data can be reduce as much as possible. Rich components for business management will be available. Another and the most important section of this business is "Sales and Marketing". Whereas this issue is comparatively sensitive, we'll provide a rich and modern web application for a great representation and support service to customers of the car trader. It'll be great to maintain own hardware and technologies in user's office or work place because this approach provides highest security of data but hiring of hardware and technology is not risky and comparatively it's much expense reducing and user friendly. We will provide our service on the basis of traders demand and choice and their satisfaction will our achievement.

ACKNOWLEDGMENTS

Praise to Allah, the most magnificent and the most merciful, without whose patronage and blessing this project would not have been successfully completed. He gave us zeal, confidence, power of determination and courage and vanquished all the stumbling hardness that we faced on the way.

It is an auspicious occasion for us as students of Department of Computer Science and Engineering, one of the prestigious academic centers of the Bangladesh University of Business and Technology(BUBT), to express our deep feelings of gratitude to the department . We are immensely indebted to our supervisor, Mr. Milon Biswas, Lecturer, Department of Computer Science and Technology, for his wonderful guidance, inspiration, encouragement and also for through review and correction of this dissertation work that could not be finalized without his astute supervision. We would also thank to our honorable Chairman of Department of CSE for his support and giving us permission to use the computer labs whenever we needed.

We pay profound regard to all of our teachers of the department for their very valuable directives and special attention. We would thank to our departmental staff for their service. Our parents are very much keen and hopeful in the best performance of the dissertation we are going to submit. We wish we could fulfill their aspiration. We also pay regards to our friends in the department who, through their interest and work, are our contestant source of inspiration.

With Best Regards,

Prodipta Promit Mukherjee

Minia Rahman Bristy

Sanchita Paul

DECLARATION

We hereby declare that the project entitled Car Trading Information Management System V4.0 submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering in the Faculty of Computer Science and Engineering of Bangladesh University of Business and Technology (BUBT) is our own work and that it contains no material which has been accepted for the award to the candidate(s) of any other degree or diploma, except where due reference is made in the text of the project. To the best of our knowledge, it contains no materials previously published or written by any other person except where due reference is made in the project.

Prodipta Promit Mukherjee
ID: 14152103082

Minia Rahman Bristy
ID: 14152103003

Sanchita Paul
ID: 14152103030

Contents

Title	i
Certificate	i
Dedication	ii
Approval	iii
Abstract	iv
Acknowledgements	v
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Existing Systems	4
1.2.1 Prism ERP	5
1.2.2 VMS-Vehicle Management Solution	5
1.2.3 AutoSoft	6
1.3 Motivation	8
1.3.1 Targeted Audiences	8
1.3.2 Demand Analysis	9
1.3.3 State of the Art	9
1.3.4 Investments	9
1.3.5 Expected Revenue	9
1.3.6 Strategic Benefits	10
1.4 Objectives	10
1.5 Contribution	11
1.6 Organaization of Project Report	13
1.7 Conclusion	14
2 LITERATURE REVIEW	16
2.1 Introduction	16

2.2	State of the Art	17
2.3	Existing Systems	18
2.3.1	Inventory	18
2.3.2	Sales	19
2.3.3	Human Resources	19
2.3.4	Accounts	20
2.3.5	Customer Relationship	20
2.4	Supporting Literature	21
2.4.1	Process of Requirement Analysis	21
2.4.2	Database Design	21
2.4.3	Data Flow Diagram	28
2.4.4	Use Case Diagram	29
2.4.5	Activity Diagram	32
2.4.6	Sequence Diagram	34
2.4.7	Class Diagram	36
2.4.8	State Chart Diagram	39
2.5	The Car Trading Management System Project	40
2.5.1	Features	40
2.5.2	Stategic Benifits	41
2.5.3	Disadvantge of Digital System for Car Traders	42
2.6	Conclusion	42
3	SYSTEM ANALYSIS & REQUIRMENT	43
3.1	Introduction	43
3.2	System Analysis	44
3.3	Information Gathering	46
3.4	Feasibility Study	47
3.4.1	Technical Feasibility	48
3.4.2	Operational Feasibility	48
3.4.3	Economic Feasibility	49
3.5	Result of System Analysis	50
3.5.1	Data Store	50
3.5.2	Data Insecurity	50
3.5.3	Slower Business Processes	50
3.5.4	Lack of Transparency	50
3.5.5	Accounts Issues	51
3.5.6	Cost Ineffective	51
3.5.7	Old Style of Marketing and CRM	51
3.5.8	No Decision Support System	51

3.6	Requirment Analysis	52
3.6.1	System Requirment	53
3.7	Conclusion	55
4	METHODOLOGY	56
4.1	Introduction	56
4.2	Methodology	57
4.3	Developing the Report Generation and Analytical Functionalities .	57
4.4	Use of Object Oriented Analysis & Design	57
4.4.1	What is Object-Oriented Analysis	58
4.4.2	What is Object-Oriented Design	59
4.4.3	Advantages and Disadvantages of Object-Oriented Analysis and Design	60
4.4.4	Why we choose this methodology	60
4.5	A brief look on the phases of OOSAD	61
4.5.1	Define Use case model	61
4.5.2	Drawing Activity Diagrams	61
4.5.3	Drawing Sequence Diagrams	61
4.5.4	Develop class diagrams	62
4.5.5	Draw State Chart Diagram	62
4.5.6	Modify Diagrams & Complete Specifications	62
4.5.7	Develop & Document The System	62
4.6	Contribution	62
4.7	Conclusion	63
5	PROJECT DESIGN	64
5.1	Introduction	64
5.2	Use Cases	64
5.3	Activity Diagrams	86
5.4	Sequence Diagrams	96
5.5	Class Diagrams	105
5.6	State Chart Diagram	114
5.7	Database Design	115
5.7.1	Schema Diagram for our database	115
5.7.2	Entity Relationship Diagram for our database	116
5.8	Database Object Design	117
5.8.1	Accounts	117
5.8.2	AccountsReceivables	117
5.8.3	Expenses	118

5.8.4	Customers	118
5.8.5	Images	118
5.8.6	Make	119
5.8.7	Model	119
5.8.8	Sales Record	120
5.8.9	Stuff	120
5.8.10	Users	121
5.8.11	Vehicle	121
5.9	Data Flow Diagram	121
5.9.1	Data Flow Diagram (0 Level)	121
5.9.2	Data Flow Diagram (1 Level)	122
5.10	Implementation	123
5.11	Result Analysis	124
5.12	Conclusions	124
6	DEVELOPMENT	125
6.1	Introduction	125
6.2	Starting the development cycle	125
6.3	Building a proof of concept	126
6.4	Developing the solution components	126
6.5	Testing	126
6.5.1	White Box Testing	127
6.5.2	Black Box Testing	128
6.6	Building the solution	129
6.7	Closing the Developing Phase	129
6.8	Conclusions	130
7	USER MANUAL	131
7.1	Introduction	131
7.2	Hardware requirements	131
7.2.1	Database Server	132
7.2.2	Web Server	133
7.2.3	End User	133
7.3	Software requirements	135
7.3.1	Database Server	135
7.3.2	Web Server	135
7.3.3	Client	135
7.4	Interfaces	136
7.4.1	Desktop Application	136

7.4.2	Web Site	147
7.5	Conclusions	153
8	CONCLUSIONS	154
8.1	Conclusions	154
8.2	Future Enhancement	155
8.2.1	Smart Phone App	155
8.2.2	Data Security	156
8.2.3	Artificial Intelligence in Car Shop	156
8.2.4	Blockchain	156
References		158
Appendices		165
A	Microsoft Visual Studio 2017	167
B	JetBrains Resharper	168
C	SQL Server Management Studio	169
D	Microsoft Blend for Visual Studio 2017	170
E	TexWorks	171
F	Adobe Illustrator	172
G	Microsoft Visio 2013	173

List of Figures

2.1	One-to-One	26
2.2	One-to-Many	26
2.3	Many-to-One	27
2.4	Many-to-Many	27
2.5	Data Flow Diagram Notations	28
2.6	Use Case Diagram Notations	31
2.7	Activity Diagram Notations	33
2.8	Sequence Diagram Notation	35
2.9	Class Diagram Notations	37
2.10	State Chart Diagram Notations	39
4.1	Object Oriented Analysis & Development Model	58
5.1	Use Case Diagram(Sky View) of the System	65
5.2	Use Case Diagram of Authentication Component	66
5.3	Use Case Diagram of Home Navigation Component	68
5.4	Use Case Diagram of Inventory Component	70
5.5	Use Case Diagram of Sales Component	72
5.6	Use Case Diagram of Manage Stuff Component	74
5.7	Use Case Diagram of Expenditures Component	76
5.8	Use Case Diagram of Decision Support or Dashboard Component .	78
5.9	Use Case Diagram of Owner Area Component	80
5.10	Use Case Diagram of Support Component	82
5.11	Use Case Diagram of Web Site	84
5.12	Use Activity Diagram of Authentication Component	86
5.13	Use Activity Diagram of Home Navigation Component	87
5.14	Use Activity Diagram of Inventory Component	88
5.15	Use Activity Diagram of Sales Component	89
5.16	Use Activity Diagram of Manage Stuff Component	90
5.17	Use Activity Diagram of Expenditures Component	91
5.18	Use Activity Diagram of Owner Area Component	92

5.19	Use Activity Diagram of Decision Support Component	93
5.20	Use Activity Diagram of Support component	94
5.21	Use Activity Diagram of Web Site	95
5.22	Sequence Diagram of Authentication Component	96
5.23	Sequence Diagram of Inventory Component	97
5.24	Sequence Diagram of Sales Component	98
5.25	Sequence Diagram of Manage Stuffs Component	99
5.26	Sequence Diagram of Expenditures Component	100
5.27	Sequence Diagram of Owner Area Component	101
5.28	Sequence Diagram of Dashboard Component	102
5.29	Sequence Diagram of Support Component	103
5.30	Sequence Diagram of Web Site	104
5.31	Class Diagram of Authentication Component	105
5.32	Class Diagram of Inventory Component	106
5.33	Class Diagram of Sales Component	107
5.34	Class Diagram of Manage Stuffs Component	108
5.35	Class Diagram of Expenditures Component	109
5.36	Class Diagram of Owner Area Component	110
5.37	Class Diagram of Dashboard Component	111
5.38	Class Diagram of Support Component	112
5.39	Class Diagram of Web Site	113
5.40	State Chart Diagram of Car Trading Information System	114
5.41	Schema Diagram	115
5.42	Entity Relationship Diagram	116
5.43	0 Level DFD	122
5.44	1 Level DFD	123
7.1	Log In Page	136
7.2	Home Page	137
7.3	Inventory Page	138
7.4	Sales Page	139
7.5	Sales' Invoice Page	140
7.6	Manage Stuff Page	141
7.7	Expenditures Page	142
7.8	Owner Area Page	143
7.9	Income Statement Page	144
7.10	Dashboard Page	145
7.11	Support Page	146
7.12	Home Page	147

7.13 About Page	148
7.14 Inventory Page	149
7.15 Inventory Details Page	150
7.16 Services Page	151
7.17 Contact Page	152

List of Tables

5.1	Accounts	117
5.2	AccountsReceivables	117
5.3	Expenses	118
5.4	Customers	118
5.5	Images	118
5.6	Make	119
5.7	Model	119
5.8	SalesRecord	120
5.9	Stuff	120
5.10	Users	121
5.11	Vehicle	121

Chapter 1

INTRODUCTION

1.1 Introduction

Since the twentieth century, the role of the car has become highly important though controversial. It is used throughout the world and has become the most popular mode of transport in the more developed countries. In developing countries, the effects of the car on society are not as visible, however they are significant.

Having a car is important for people in general because it provides status and the opportunity for personal control and autonomy. In sparsely populated areas, owning a car is even more important, since it provides the only opportunity for travelling long distances due to a lack of public transport. For older people, having more difficulties walking (to the bus stop) and cycling, driving is often the only option for independent mobility. Several studies have found that over 90 percent of older drivers indicate that giving up driving would restrict their independence and mobility. The same drivers expressed anxiety about the poor quality of public transport services. This anxiety seemed to be based on reality because 50 percent of those respondents who already had given up driving felt public transport to be, at least in some measure, inadequate.[1].

Whereas, Bangladesh is a developing country and day after day the demand of personal transportation is growing. In 2018s, around 63 cars sold which is highest in the countrys history and car trading becomes one of the largest business are across the country[2]. By increasing the growth of consumers demand, there be also increasing the competition among traders and in last five years, the competition becomes more complex. Approaches of executing business operations are not

optimized respectively to modern age. Currently none of vehicle manufacturers established production house, but the demand of personal vehicle increased day after day in spite of lack of public transportation service. Currently Japanese public multinational conglomerate corporation Honda Motor Company Ltd. heading to establish production house in Bangladesh, but it'll take time[3]. This is why, the demand of reconditioned vehicles is much higher. Unusual expenses, time consuming processes etc. affects traders policies and strategies and with this competitive market customer relationship, marketing etc. are not as desired as modern day consumers. By analyzing all these scenarios and risk factors, the solution been developed also by focusing technical and financial factors where the traders can execute their business processes faster, smarter and better.

According to Barvida, in mid 2018s, the market size for reconditioned cars is around BDT 5,000cr. Each year the size of the market is increasing by 15 to 20 percent. In our country, around 63 cars are sold every day now the highest in the countrys history with the number of imported cars having increased threefold since the 2012-13 fiscal year. Every year, on average 20,000 of vehicles imported from several countries[2]. and currently it has a huge contribution on our economy from import duty which stands around BDT 5 billion last year[4].

The business chain is comparatively complex in this field of trade. Importer imports used vehicle from countries and mostly sale as dealer to retail sellers. It's rare to see importers sale vehicles directly to customers. Before a vehicle displayed to a customer, it must be reconditioned from local workshops. Reconditioning of a vehicle may performed by importer or retail seller who purchased this, depends on negotiations. Some times, retail sellers also perform as a dealer and sale vehicles to other retail sellers. It's a common scenario we heard that purchasing car from Chittagong can save money of a customer. This happens because of this complex business chain.

It's much interesting that a car shop operates with a huge cash flow with a small area of operation and limited human resources. They purchased vehicles to sale, maintain vehicles which causes various expenditures. They sale vehicle through affiliation, directly or indirectly, profiting around 50,000 to 1,00,000 each car. A few people work in there. In standard approach, the business operates by owner(s), one manager, two or three mechanics etc. It is actually a small sized business with a huge cash flow.

Currently, most of the shops has no digital solution to execute their business operations. Some big shops(kind of brands like HNS, Salsabil Autos etc.) use digital solutions to maintain their business. It may caused by higher price of software systems. From this, most of the shops executes their operations by using registers, which can be defined as "Traditional Hard Paper Based Processing System" which is too old to use in this modern age.

It is clearly seems that, there be a significant contribution of this car trading industry on national economy of Bangladesh. Here, the most interesting thing is day after day, the industry is growing, and it can be state confidently that in this year(2019), the status of the industry is more than previously described status of last year. So, today the car trading actually demands digital solution to execute the business with more optimistic way, which can basically saves car traders' time and money. Because faster business processes can generate revenue faster also. There be a directly proportional relationship between processes' execution and revenue. This is why, day by day the demand is growing on aggregate.

The Car Trading Information Management System is a business information system solution which exactly the combination of Desktop Application and Web Site. Traders will maintain their required business processes like inventory, sales, accounts etc. by using the Desktop Application. The Web Application is mainly focused on building stronger customer relationship and marketing approach. And in this modern age of business, the solution also provides the opportunity of business data analysis which is one of the most important section in any business across the world.

We are confident that, the digital system solution we determined will occur more revenue and profit to any car trader. The solution will reduce existing issues like internal transparency etc. There be no necessity of affiliates or third party to sale specific vehicle. The internal managerial activities becomes faster, smoother and more rich than existing system. The relation between car trader and customer will improve. The solution indirectly works for shop's branding. The more the people knows the name of the shop, potentially profit will increase. It can be state that, the income of car trader will increase upto 40 percent. On aggregate, if all car traders starts using digital solution to execute their business, the industry become bigger in upcoming years. The more the business growth occurs more employment. And also, for the vendor side, the more the solution sales occurs more employment of development and maintenance engineers. This is not only a

project to get profit, but also a large contribution to national economy via aggregate demanded on car sale, taxes and import duty, employment and so forth. We believe the project will bring not only profit to us, but also a satisfaction for being a media for this significant contribution on national economy, which can not be purchased.

1.2 Existing Systems

The term "Existing System" refers that how exactly targeted audiences executes their operations and activities currently without the proposed system. It is too much important to study in spite of identifying their exact demand on proposed system, that what it will perform and how. Currently most of the car traders doesn't use digital solution to operate their business. In this section, we will see all of their business processes and how exactly they operate it.

Currently, most of the shops has no digital solution to execute their business operations. Some big shops(kind of brands like HNS, Salsabil Autos etc.) use digital solutions to maintain their business. It may caused by higher price of software systems. From this, most of the shops executes their operations by using registers, which can be defined as "Traditional Hard Paper Based Processing System" which is too old to use in this modern age. Because of this, several issues formed commonly in car shops. They're-

- Time consuming to maintain too many documents.
- Financial management becomes complex.
- Occurs intangible losses highly.
- Failed to reach customers and depends on affiliates.
- Failed to make business executions transparent.
- Easily formed conspiracy.

Some shops like HNS Automobiles, uses digital solution but it's much old and they are failed to get effective and efficient digital solution in this modern age of technology. It's much hard to accept this reality that with this huge cash flow of a business and there is no effective and efficient software solution to improve and optimize their business processes and boost profit.

A few digital solutions are existing in market. Here we describe some of them.

1.2.1 Prism ERP

This is a generic ERP solution, developed by Divine IT Ltd.[5]. is a platform independent and ready system for any business. Easy to deploy, Easy to Learn, Intuitive and Lightweight System. It is completely integrated with other modules and having the open options to integrate with 3rd party systems. It supports almost every types of business by its presets. There are approximately 30 types of business that are already implemented successfully, and car trading solution is not excluding from them.

PrismERP has versatile core business modules which are capable to manage administrative managerial business propositions entirely centrally[6].

- Cloud based
- High end.
- Cluster supported.
- Supports any database.
- Deployment in any Platform.

1.2.2 VMS-Vehicle Management Solution

The application VMS-Vehicle Management Solution is a ordinated solution, developed by[7]for the problems, generally a Transport manager faced. Have an attentive tour from top to bottom get a complete idea about a Vehicle Management System.

Bdtask has designed the most modern Vehicle Management Software to manage large vehicle and fleet. This system process and analyze every single data of your all Vehicles. It is an online based Software which is really helpful for Bus, Car, taxi, Cover Van Truck operators to manage everyday workflow and performance. It is an online and offline based application, it can be run from anywhere, which is designed by PHP and Codeigniter framework with MySQL. It can be operated from around the world using username and password. In this Vehicle management system, one or more operator can use at a time from anywhere in the world. They can Add, Delete, Edit, any data in the software[8].

Only hard work is not the key of good luck. If it were, then the donkey would

have been the king of jungle but Lion. So, to be successful, you have to be tactful. only an intelligent transportation business owner chose an automation system like vehicle management system to get following benefits:

- To get income and expenses report of all vehicles.
- To get vehicle wise parts uses report.
- To find out driver wise performance based information.
- To ensure best HRM. Such as: Drivers, Helpers, Other employees at a glance Report.
- To set an Alert Center. It will inform about all vehicle taxes, token, insurance, road permit, expiration date.
- To trace all over regular and irregular expenses.
- To apply an organized Inventory management system with regular income and expenses.
- To have an automatic stock register.
- To get an easy traceable stock and product system.
- Expense Reports are particular and timeframe based.

1.2.3 AutoSoft

Developed by a Indo-Japanese company, Mark Soft, is a internationally leading solution for auto traders. AutoSoft Automobile Trading Software is an essential business tool designed for small and medium size companies. AutoSoft is simple and powerful solution and can be installed quickly and affordably. It helps you to reach out to your customers and keep a close eye on the market and your Automobile business 24/7/365 through the Online Car Selling portals[9].

AutoSoft has features to record the Purchase, document, Transport, Auction Sales, Shipping schedule, Export and Internet Sales, Document Tracking, Quotation ProForma invoice, Payment and Manage Websites.

AutoSoft was launched 12 years ago, and has been evolving to meet the new

business needs. It has been implemented successfully by many Small and Big Automobile Trading companies in Japan and other Countries. Here the features of AutoSoft given below:

- Purchase screen provided with multiple functionalities like Auction Purchase, Local Purchase, Auction Purchase along with Auction Sold, Auction Purchase along with Purchase Cancel. For each vehicle unique stock ID generated automatically, Purchase Transaction can be easily retrieved and modified if required, based on the auto generated Transaction Number, Supplier Name and Purchase Date.
- Sales Order screen used for creating sales transactions based on the Loading Port , Vehicle Available Location and Discharge Port and Final Destination. You can add the Vehicle details by selecting from the available stock list or by entering the stock No/Chassis No. Sold Amount can be entered against each vehicle in the grid, In this screen it is also possible to know the shipping status of the each vehicle i.e. whether vehicle shipped or not. Consignee and Notify Party details are entered at the time of sales transaction creation.
- Category, Maker and Type of the Vehicles display the stocks in tree-based structure. It has options to view All Vehicles, Available Vehicles, Shipped Vehicles and Pending Shipment Vehicles and Sales Order only Created Vehicles. The Users can customize the stock view based on selecting/deselecting the one or more properties of the vehicles. It is also possible to search the vehicles from this list based on the Stock No / Chassis No. Stock view can sort, filtered by one or more properties of the Vehicles; The Stock view can be exported to Excel.
- With just a few clicks you can generate reports to gain valuable insight into your business. And with just one click, these reports can be easily exported to Microsoft Excel, Word or PDF. there reports includes purchase list based on supplier and purchase date range, sales reports based on customer and country, shipped vehicle list, recycle reports and stock on hand, export certificate and more.
- This is where you can select list of vehicles you want to show in the Website, enter the sales price of each vehicle, special offer vehicles if any, generate web-stock number and selected stocks can easily updated to website by this screen feature.
- You can easily view list of customers and its associated consignee, notify party details, and customer purchase history and their payment details.

1.3 Motivation

Internal and external factors that stimulate desire and energy in people to be continually interested and committed to a job, role or subject, or to make an effort to attain a goal. Motivation results from the interaction of both conscious and unconscious factors such as the intensity of desire or need, incentive or reward value of the goal, and expectations of the individual and of his or her peers. These factors are the reasons one has for behaving a certain way. An example is a student that spends extra time studying for a test because he or she wants a better grade in the class[10].

Being software engineer, software is a product to us. We never build software for our personal use. It must be sale. So here comes a lot of things before choosing a software project to develop. The idea comes on my mind in a very interesting way. One day when I study about targeted audience and customer demographics of my windows store apps[11] [12], suddenly the idea appears that car traders are comparatively rich audiences and working for them may causes higher profit. But this idea is not enough to initiate the project, because the idea is to build a product which exactly will be a customized solution and partially generic. This is why I started studying to initiate the project. Here is how I performed the study to reach final decision to build the solution.

1.3.1 Targeted Audiences

Firstly we have to choose the targeted audiences for whom we will develop project. In this project, car traders are the targeted audiences. We carefully studied about car traders, their trading style, cash flow etc. and so forth. In car trading, it is a small sized business with huge cash-flow. Car traders are much positive to invest. Things that can improve their business, they are willing to pay. Transaction that performed by car traders is much transparent. In this project, average and small traders are our major targeted audiences.

1.3.2 Demand Analysis

Demand is an economic principle referring to a consumer's desire to purchase goods and services and willingness to pay a price for a specific good or service[13]. Demand analysis is a research done to estimate or find out the customer demand for a product or service in a particular market. Demand analysis is one of the important consideration for a variety of business decisions like determining sales forecasting, pricing products/services, marketing and advertisement spending, manufacturing decisions, expansion planning etc. Demand analysis covers both future and retrospective analysis so that they can analyze the demand better and understand the product/service's past success and failure too[14]. Car traders have desire to get effective digital solution, are willing to pay and they have plan to purchase the service. So, it is clear that there is demand for the solution.

1.3.3 State of the Art

Currently, most of the shops has no digital solution to execute their business operations. Some big shops(kind of brands like HNS, Salsabil Autos etc.) use digital solutions to maintain their business. It may caused by higher price of software systems. From this, most of the shops executes their operations by using registers, which can be defined as "Traditional Hard Paper Based Processing System" which is too old to use in this modern age.

1.3.4 Investments

Whereas, currently the project work will performed as teamwork, direct investments like server purchase, maintenance etc. will not appear, our current development setup is perfect to verify and validate the solution. And this approach will valid for both development and shipment stage. But there be some hidden expenses will occur, but it can be assume that it'll not infeasible to us.

1.3.5 Expected Revenue

There be too many average or small car traders across the country. More than 100s of car traders just executing their business in Dhaka. So, in aggregate, there be a lot of audiences are here and more than 80 percent of them will positive to invest by acquiring the solution which may cause a huge amount of revenue and

that is what exactly we expect.

1.3.6 Strategic Benefits

In previous discussion, if the half of expected subscriber being formed, then it is not hard to be a brand in software industry of Bangladesh. Being a brand, it is easier to increase the business competitively. From external investments, it's possible to step ahead for monopoly market.

From these steps, we identified that working in this project will beneficial and these benefits are the main motivation to initiate the project.

1.4 Objectives

A specific result that a person or system aims to achieve within a time frame and with available resources.[15].[16].

Car Trading Information Management System is a combination of desktop application and web site which exactly covers the entire business processes and activities which is required and feasible.

The most important thing is how feasible the solution is. We can make solution as all as other solution running internationally. But the fact is our targeted audiences, legal issues, business chain, operational execution, customer demographics and so on. The challenge arise in adaptability of targeted audiences. For example, It will totally unrealistic if we offer them big data analysis or traditional accounting system in our proposed system because they have not such qualified personnel and it is hard for them to appoint them in present situation of trade. It can be predict that, the situation will change and one day they can execute their trade with latest business style. But, here another challenge began about validity of proposed solution.

Too many challenges will occur during the development and maintenance of the system but we have to clear about our objective, no matter how or when. Our main objective to develop an effective and efficient solution which is feasible to all targeted audiences, not only for today, but also for future.

In general, objectives are more specific and easier to measure than goals. Ob-

jectives are basic tools that underlie all planning and strategic activities. They serve as the basis for creating policy and evaluating performance. Some examples of business objectives include minimizing expenses, expanding internationally, or making a profit.

The following are the Objective of this project :

- A budget friendly digital system solution.
- Increase operational optimization.
- Increase business transparency.
- Remove affiliation or third-party based business style.
- Effective decision support for senior managerial actions.
- Flexible solution to maintain.
- Increase system reliability.
- Increase customer relationship.
- Upgrading branding and marketing strategy in modern day basis.
- As feasible as possible.
- Keep abreast of the times.

1.5 Contribution

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components. Software development is a process of writing and maintaining the source code, but in a broader sense, it includes all that is involved between the conception of the desired software through to the final manifestation of the software, sometimes in a planned and structured process. Therefore, software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products. [17]

Software can be developed for a variety of purposes, the three most common being to meet specific needs of a specific client/business (the case with custom software), to meet a perceived need of some set of potential users (the case with commercial and open source software), or for personal use (e.g. a scientist may write software to automate a mundane task). Embedded software development, that is, the development of embedded software, such as used for controlling consumer products, requires the development process to be integrated with the development of the controlled physical product. System software underlies applications and the programming process itself, and is often developed separately. [18]

The need for better quality control of the software development process has given rise to the discipline of software engineering, which aims to apply the systematic approach exemplified in the engineering paradigm to the process of software development. [19]

There are many approaches to software project management, known as software development life cycle models, methodologies, processes, or models. The waterfall model is a traditional version, contrasted with the more recent innovation of agile software development. [20]

A software development process (also known as a software development methodology, model, or life cycle) is a framework that is used to structure, plan, and control the process of developing information systems. A wide variety of such frameworks has evolved over the years, each with its own recognized strengths and weaknesses. There are several different approaches to software development: some take a more structured, engineering-based approach to developing business solutions, whereas others may take a more incremental approach, where software evolves as it is developed piece-by-piece. One system development methodology is not necessarily suitable for use by all projects. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project and team considerations. [21]

Here is our contribution of this project :

- This project is for operating car trading business in optimistic way.
- This project helps for ensuring security of business data.
- This project helps for ensuring transparency on business.
- This project helps owner(s) to determine optimistic decision for business

improvement.

- This project helps owner(s) getting informed during his absence.
- This project helps personnels with automated accounting system.
- This project helps visitors getting informed before physically visit the shop.
- This project helping car trader by increasing shop's branding.

1.6 Organaization of Project Report

Now, inform about next chapter of this report:

Show in Chapter-2 This chapter were providing the theoretical background which is related to the project development and make reference to existence of other systems. In this chapter we will be discuss software and programing language. The sources are refer from the book, articles, journals and also sources from internet.

Show in Chapter-3 This project is a web based so we have used technologies and programming language for the development. System analysis and design deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together. System analysts solve business problems through analysing the requirements of information systems and designing such systems by applying analysis and design techniques. This course deals with the concepts, skills, methodologies, techniques, tools, and perspectives essential for systems analysts. The practical component of COMP 361 is object oriented and use-case driven, requiring students to go through the steps of system analysis and design to solve a real-life business problem.

Show in Chapter-4 This chapter will describe methodology and the technique that will be followed by us to develop our application. The methodology is the general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it. These methods, described in the methodology, define the means or modes of data collection or, sometimes, how a specific result is to be calculated.^[4] Methodology

does not define specific methods, even though much attention is given to the nature and kinds of processes to be followed in a particular procedure or to attain an objective. When proper to a study of methodology, such processes constitute a constructive generic framework, and may therefore be broken down into sub-processes, combined, or their sequence changed. A paradigm is similar to a methodology in that it is also a constructive framework. In theoretical work, the development of paradigms satisfies most or all of the criteria for methodology.[6] An algorithm, like a paradigm, is also a type of constructive framework, meaning that the construction is a logical, rather than a physical, array of connected elements.

Show in Chapter-5 Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designers goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

Show in Chapter-6 The primary goal during the Developing Phase is to build the solution components code as well as documentation. Some development work may, however, continue into the Stabilizing Phase in response to testing. The Developing Phase involves more than code development and software developers. The infrastructure is also developed during this phase and all roles are active in building and testing deliverables. The team continues to identify all risks throughout the phase and address new risks as they emerge.

Show in Chapter-7 This chapter will show the Graphical User Interface (GUI) of the system through screen shots and user guides that how will they operate the application. A complete direction for a user to properly run Online Judge.

1.7 Conclusion

The project, observation and the analysis has been done to the existing system in order to make sure that the new system will cover all the function that the existing system does not have and to make sure that the new system is much better than the existing trend to execute a car trading business which will save the time and

money of car trader and boost the profit.

Our goal is to provide a system and service to you. To do this, were here to provide you a desktop application which will use to manage your business issues. A web application which will represent and highlight your business to customers and give them so many opportunities continuously. Were trying to bring the break-point of this traditional system of car trading and take it into digital form. We never compromise with quality and were here to be the witness of your satisfaction.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter were providing the theoretical background which is related to the project development and make reference to existence of other systems. In this chapter we will be discuss software and programing language. The sources are refer from the book, articles, journals and also sources from internet.[22] [23]

A literature review is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and do not report new or original experimental work. Most often associated with academic-oriented literature, such reviews are found in academic journals, and are not to be confused with book reviews that may also appear in the same publication. Literature reviews are a basis for research in nearly every academic field. A narrow- scope literature review may be included as part of a peer-reviewed journal article presenting new research, serving to situate the current study within the body of the relevant literature and to provide context for the reader. In such a case, the review usually precedes the methodology and results sections of the work. Producing a literature review may also be part of graduate and post-graduate student work, including in the preparation of a thesis, dissertation, or a journal article. Literature reviews are also common in a research proposal or prospectus.

2.2 State of the Art

In this current era, transportation stands as a basic need informally and Bangladesh is not excluding from them. In our country, currently non of vehicle manufacturers established production house, but the demand of personal vehicle increased day after day in spite of lack of public transportation service. Currently Japanese public multinational conglomerate corporation Honda Motor Company Ltd. heading to establish production house in Bangladesh, but it'll take time[3]. This is why, the demand of recondition vehicles is much higher.

According to Barvida, in mid 2018s, the market size for reconditioned cars is around BDT 5,000cr. Each year the size of the market is increasing by 15 to 20 percent. In our country, around 63 cars are sold every day now the highest in the countrys history with the number of imported cars having increased threefold since the 2012-13 fiscal year. Every year, on average 20,000 of vehicles imported from several countries[2] and currently it has a huge contribution on our economy from import duty which stands around BDT 5 billion last year[4].

The business chain is comparatively complex in this field of trade. Importer imports used vehicle from countries and mostly sale as dealer to retail sellers. It's rear to see importers sale vehicles directly to customers. Before a vehicle displayed to a customer, it must be reconditioned from local workshops. Reconditioning of a vehicle may performed by importer or retail seller who purchased this, depends on negotiations. Some times, retail sellers also perform as a dealer and sale vehicles to other retail sellers. It's a common scenario we heard that purchasing car from Chittagong can save money of a customer. This happens because of this complex business chain.

It's much interesting that a car shop operates with a huge cash flow with a small area of operation and limited human resources. They purchased vehicles to sale, maintain vehicles which causes various expenditures. They sale vehicle through affiliation, directly or indirectly, profiting around 50,000 to 1,00,000 each car. A few people work in there. In standard approach, the business operates by owner(s), one manager, two or three mechanics etc. It is actually a small sized business with a huge cash flow.

Currently, most of the shops has no digital solution to execute their business operations. Some big shops(kind of brands like HNS, Salsabil Autos etc.) use digital solutions to maintain their business. It may caused by higher price of software systems. From this, most of the shops executes their operations by using registers, which can be defined as "Traditional Hard Paper Based Processing System" which is too old to use in this modern age. Because of this, several issues formed commonly in car shops. They're-

- Time consuming to maintain too many documents.
- Financial management becomes complex.
- Occurs intangible losses highly.
- Failed to reach customers and depends on affiliates.
- Failed to make business executions transparent.
- Easily formed conspiracy.

Some shops like HNS Automobiles, uses digital solution but it's much old and they are failed to get effective and efficient digital solution in this modern age of technology. It's much hard to accept this reality that with this huge cash flow of a business and there is no effective and efficient software solution to improve and optimize their business processes and boost profit.

2.3 Existing Systems

The term "Existing System" refers that how exactly targeted audiences executes their operations and activities currently without the proposed system. It is too much important to study in spite of identifying their exact demand on proposed system, that what it will perform and how. Currently most of the car traders doesn't use digital solution to operate their business. In this section, we will see all of their business processes and how exactly they operate it.

2.3.1 Inventory

The term "Inventory" is an accounting term that refers to goods that are in various stages of being made ready for sale, including:

- Finished goods (that are available to be sold)
- Work-in-progress (meaning in the process of being made)
- Raw materials (to be used to produce more finished goods)

Inventory is generally the largest current asset items expected to sell within the next year a company has[24].

In a car shop, there is the only product they stores is vehicles, whatever it may

be recondition vehicle or used vehicle. In the case of recondition vehicle, it must be imported by shop owner or another importer. Whereas a car has to import from another country, there is too many legal documents, have to maintain. After arriving a car to warehouse from port after customs clearance, it is required to send it to their tied-up workshop to repair damages and furnish from interior and exterior. After fully furnished or conditioned, it takes to the showroom to display.

2.3.2 Sales

A sales process is a set of repeatable steps that a sales person takes to take a prospective buyer from the early stage of awareness to a closed sale. Simply put, it is a potential customers journey from realizing they have a need for a product to making an actual purchase[25].

In a car shop, this is the only process from where trader earns revenue. Here, it is similar to the basic sales process as we all know, but there is some extra activities due to large amount of cash transaction. A typical sales scenario is, let a customer requests for purchase a vehicle. Typically owner or manager acts as a sales man most of the case. He will take some information from customer like NID and ETIN, receives partial payment like more than 1,00,000 Taka via cash or check and register the vehicle as sold and specified unique account by using Engine Number of the vehicle. After all the accounts receivables cleared by user, seller and customer reached a legal deal about payment because of the large amount and then the vehicle is ready to deliver. That time seller also takes some amount of money from customer for vehicle registration from BRTA, because customers are not willing to face this hassles most of the time. After completing registration related paperworks from BRTA, the vehicle will deliver to customer's home.

Another term stands with sales is known as service or warranty. Whereas, vehicles are not brand new, so non of them gives warranty like 3 years/1,00,000km. They usually provides service warranty to customer. It is like, if customer purchase parts and accessories, they will install it free of service charge.

2.3.3 Human Resources

William R. Tracey, in "The Human Resources Glossary," defines Human Resources as: "The people that staff and operate an organization," as contrasted with the financial and material resources of an organization.

A human resource is a single person or employee within your organization. Human

resources refer to all of the people you employ.

Human Resources is also the function in an organization that deals with the people and issues related to people such as compensation and benefits, recruiting and hiring employees, on-boarding employees, performance management, training, and organization development and culture[26].

In a car shop, there seems a little amount of human resource. Mostly in car shops, there be one or more owner(s), one manager, mostly 2-5 mechanics and 1-2 service boys works.

2.3.4 Accounts

It's not possible to operate business without maintaining accounts. The term "Accounts" refers a sequence of activities involving the recording of how cash is received and paid out in a company or organization. The accounting process in business is based on four accounting methods, which are: the accrual method, the consistency method, the prudence method and the going concern method[27].

It's much interesting that mostly in car shops there be no accountant(s). They usually maintain their financial actions by registering some simple ledger balances like revenue, expense, accounts payables and accounts receivables, and sometimes they are irregular to manage those financial transactions.

2.3.5 Customer Relationship

The term "Customer Relationship" refers that the development of an ongoing connection between a company and its customers. The relationship involves marketing communications, sales support, technical assistance and customer service. The relationship is measured by the degree of customer satisfaction through the buying cycle and following receipt of goods or services. See also customer relationship management[28].

This process in a car shop seems comparatively poor in this modern age. Mostly car sales happens via affiliation with owner(s). Unfortunately, there be a huge gap forms between trader and customer. Traders' current strategy of marketing is not sufficient to reach their product to maximum customer. But traders' relation after sale is as usual with customer.

2.4 Supporting Literature

As organizations and technology have become more complex, most large organizations now build project teams that incorporate several analysts with different but complementary, roles. In smaller organizations, one person may play several of these roles. Here we briefly describe these roles and how they contribute to a systems development project. The systems analyst role focuses on the IS issues surrounding the system. This person develops ideas and suggestions for ways that IT can support and improve business processes, helps design new business processes supported by IT, designs the new information system, and ensures that all IS standards are maintained. The systems analyst will have significant training and experience in analysis and design and in programming.

2.4.1 Process of Requirement Analysis

Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design[29].

2.4.2 Database Design

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.[30]

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).[27] A database is usually a fundamental component of the information system, especially in business oriented systems. Thus database design is

part of system development. The following picture shows how database design is involved in the system development lifecycle.

The phases in the middle of the picture (Database Design, Database Implementation) are the phases that you concentrate on in the Database Design course. The other phases are briefly described. They are part of the contents of the Systems Analysis and Design courses, for example.

There are various methods of how the different phases of information system design, analysis and implementation can be done. Here the main tasks or goals are described but no method is introduced.[5] In the conceptual database design phase, the model of the data to be used independent of all physical considerations is to be constructed. The model is based on the requirements specification of the system.

In the logical database design phase, the model of the data to be used is based on a specific data model, but independent of a particular database management system is constructed. This is based on the target data model for the database e.g. relational data model.

In the physical database design phase, the description of the implementation of the database on secondary storage is created. The base relations, indexes, integrity constraints, security, etc. are defined using the SQL language.[31] This is an optional phase. When there is a need for a new database management system (DBMS), this phase is done. DBMS means a database system like Access, SQL Server, MySQL, Oracle.

In this phase the criteria for the new DBMS are defined. Then several products are evaluated according to the criteria. Finally the recommendation for the selection is decided.[29] In this section of this report ,we describe the database design method for the development of Online Judge .

2.4.2.1 Relational DBMS (RDBMS)

The RDBMS (Relational Database Management System) is a system complying to the relational model developed by IBMs E F Cod. It allows the user to construct, modify and administer a relational database. Most of the databases that exist today are an extension of this age-old model. The data is structured in database tables, fields and records. The stored data is allowed to be manipulated using relational operators in Relational Database Management System. SQL is used as the data query language in this system.[32] [33]

Features of RDBMS

The system caters to a wide variety of applications and quite a few of its stand out features enable its worldwide use.[34] [35] The features include:

- First, its number one feature is the ability to store data in tables. The fact that the very storage of data is in a structured form can significantly reduce iteration time.
- Data persists in the form of rows and columns and allows a facility primary key to define unique identification of rows.
- It creates indexes for quicker data retrieval.
- Allows for various types of data integrity like (i) Entity Integrity; wherein no duplicate rows in a table exist, (ii) Domain Integrity; that enforces valid entries for a given column by filtering the type, the format, or the wide use of values, (iii) Referential Integrity; which disables the deletion of rows that are in use by other records and (iv) User Defined Integrity providing some specific business rules that do not fall into the above three.
- Also allows for the virtual table creation which provides a safe means to store and secure sensitive content.
- Common column implementation and also multi user accessibility is included in the RDBMS features.

We used RDBMS because it has following advantage:

- Data is stored only once and hence multiple record changes are not required. Also deletion and modification of data becomes simpler and storage efficiency is very high.
- Complex queries can be carried out using the Structure Query Language. Terms like Insert, Update, Delete, Create and Drop are keywords in SQL that help in accessing a particular data of choice.
- Better security is offered by the creation of tables. Certain tables can be protected by this system. Users can set access barriers to limit access to the available content. It is very useful in companies where a manager can decide which data is provided to the employees and customers. Thus a customized level of data protection can be enabled.

- Provision for future requirements as new data can easily be added and appended to the existing tables and can be made consistent with the previously available content. This is a feature that no flat file database has.

2.4.2.2 Entity Relationship Diagram (ER-Diagram)

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other. It shows the logical structure or flow of data in a database. Entity-Relationship Diagrams are very important in planning a database structure. They hold the basic concepts of the data exchange between different entities in a database. They are also very important in making logical and efficient databases. ER Diagrams consists of entities and their relationship with each other.[31]

- Entity
- Weak entity
- Attribute
- Multi valued attribute
- Derived attribute
- Relationship

Entity

An entity is something that exists as itself, as a subject or as an object, actually or potentially, concretely or abstractly, physically or not. It need not be of material existence. In particular, abstractions and legal fictions are usually regarded as entities. In general, there is also no presumption that an entity is animate, or present. The word is abstract in intention. It may refer, for example, to Bucephalus, the horse of Alexander; to a stone; to a cardinal number; to a language; or to ghosts or other spirits. The word entitative is the adjective form of the noun entity. Something that is entitative is considered in its own right.[24]

Weak entity

A weak entity is an entity that depends on the existence of another entity. In

more technical terms it can be defined as an entity that cannot be identified by its own attributes. It uses a foreign key combined with its attributes to form the primary key. An entity like order item is a good example for this. The order item will be meaningless without an order so it depends on the existence of order. It is represented by double rectangle.[25]

Attribute

An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. An entity can have as many attributes as necessary. Meanwhile, attributes can also have their own specific attributes. Attributes are usually represented in oval shape.[25]

Multi valued attribute

If an attribute can have more than one value it is called a multi valued attribute. It is important to note that this is different to an attribute having its own attributes. For example a teacher entity can have multiple subject values. It is represented by a double ellipse.[36]

2.4.2.3 Mapping Cardinality

Cardinality refers to the number of entity objects on each side of the relationship. In e-r diagram there are four types of mapping cardinalities. For example: a customer can order products one after another.[37]

- One-to-One
- One-to-Many or Many-to-One (dependent on the direction)
- Many-to-One
- Many-to-Many

1. One-to-One

A one-to-one relationship is the simplest relationship between two beans. One entity bean relates only to one other entity bean. For example: a customer can be kept only in one word/cell at a time.[28]

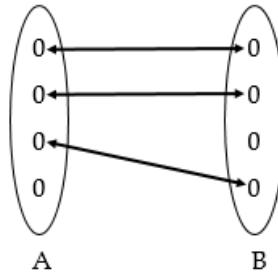


Figure 2.1: One-to-One

2. One-to-Many

In a one-to-many relationship, one object can reference several instances of another. A one-to-many relationship is a type of cardinality that refers to the relationship between two entities (see also entityrelationship model) A and B in which an element of A may be linked to many elements of B, but a member of B is linked to only one element of A. For instance, think of A as mothers, and B as children. A mother can have several children, but a child can have only one biological mother.[38]

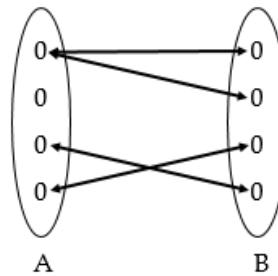


Figure 2.2: One-to-Many

3. Many-to-One

In a many-to-one relationship, many objects can reference one instance of another. A many-to-one relationship is where one entity (typically a column or set of columns) contains values that refer to another entity (a column or set of columns) that has unique values.[39]

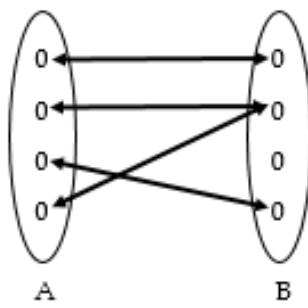


Figure 2.3: Many-to-One

4. Many-to-Many A many-to-many relationship is complex. In a many-to-many relationship, many objects can reference many objects. This cardinality is the most difficult to manage.[40]

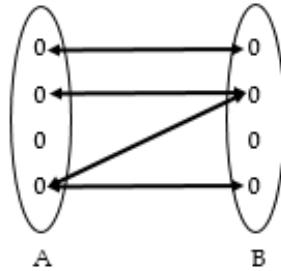


Figure 2.4: Many-to-Many

2.4.2.4 Comparison of Primary Keys to Foreign Keys

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary Key - the field that is unique for all the record occurrences.
- Foreign Key - the field used to set relation between tables.

Foreign Key - the field used to set relation between tables. In this section, the basic structure of the tables composing the database for the project are shown along with information about primary and foreign keys.

2.4.3 Data Flow Diagram

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored[41].

Data flow diagrams became popular in the 1970s in software development. They were first described in a classic text about Structured Design written by Larry Constantine and Ed Yourdon. Yourdon Coad's Object Oriented Analysis and Design (OOA/OOD) was a way of visualizing software systems before UML diagrams[41].

There are essentially two different types of notations for data flow diagrams (Yourdon Coad or Gane Sarson) defining different visual representations for processes, data stores, data flow and external entities. Yourdon and Coad type data flow diagrams are usually used for system analysis and design, while Gane and Sarson type DFDs are more common for visualizing information systems. Visually, the biggest difference between the two ways of drawing data flow diagrams is how processes look. In the Yourdon and Coad way, processes are depicted as circles, while in the Gane and Sarson diagram the processes are squares with rounded corners[41].



Figure 2.5: Data Flow Diagram Notations

2.4.3.1 Process

A process shows a transformation or manipulation of data flows within the system. The symbol used is a rectangular box which contains 3 descriptive elements. Firstly an identification number appears in the upper left hand corner. This is allocated arbitrarily at the top level and serves as a unique reference. Secondly, a location appears to the right of the identifier and describes where in the system the process takes place. This may, for example, be a department or a piece of hardware. Finally, a descriptive title is placed in the centre of the box. This should be a simple imperative sentence with a specific verb, for example 'maintain customer records' or 'find driver'[42].

2.4.3.2 Data Flow

A data flow shows the flow of information from its source to its destination. A data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents[42].

2.4.3.3 Interface

Also known as external entity, is normal for all the information represented within a system to have been obtained from, and/or to be passed onto, an external source or recipient. These external entities may be duplicated on a diagram, to avoid crossing data flow lines. Where they are duplicated a stripe is drawn across the left hand corner, like this[42].

2.4.3.4 Data Store

A data store is a holding place for information within the system. It is represented by an open ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number[42].

2.4.4 Use Case Diagram

In software and systems engineering, a use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language (UML) as an actor) and a system to achieve a goal. The actor can be a human or other external system. In systems engineering, use cases are used

at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in the Systems Modeling Language (SysML) or as contractual statements.

Use case analysis is an important and valuable requirement analysis technique that has been widely used in modern software engineering since its formal introduction by Ivar Jacobson in 1992. Use case driven development is a key characteristic of many process models and frameworks such as ICONIX, the Unified Process (UP), the IBM Rational Unified Process (RUP), and the Oracle Unified Method (OUM). With its inherent iterative, incremental and evolutionary nature, use case also fits well for agile development.

2.4.4.1 Purpose of Use Case Diagram

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view. In brief, the purposes of use case diagrams can be said to be as follows.

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

To understand the use case diagram, here the use case diagram notation represented graphically[43].

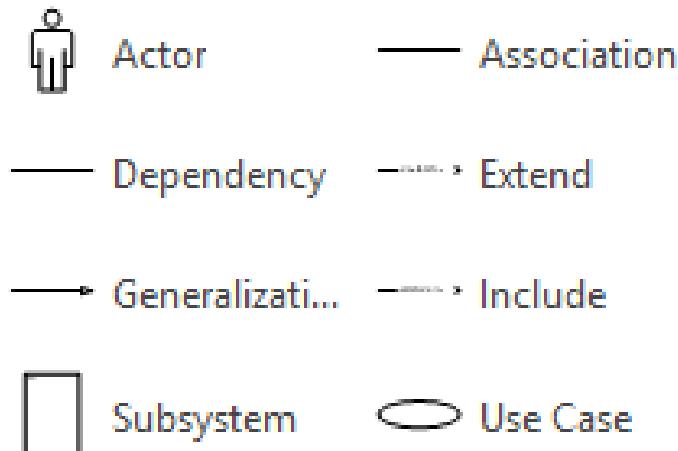


Figure 2.6: Use Case Diagram Notations

2.4.4.2 Use Case

A use case represents a user goal that can be achieved by accessing the system or software application. In Visual Paradigm, you can make use of the sub-diagram feature to describe the interaction between user and system within a use case by creating a sub-sequence diagram under a use case. You can also describe the use case scenario using the Flow of Events editor.[44]

2.4.4.3 Association

Actor and use case can be associated to indicate that the actor participates in that use case. Therefore, an association correspond to a sequence of actions between the actor and use case in achieving the use case.[44]

2.4.4.4 Actor

Actors are the entities that interact with a system. Although in most cases, actors are used to represent the users of system, actors can actually be anything that needs to exchange information with the system. So, an actor may be people, computer hardware, other systems, etc.

Note that actor represents a role that a user can play but not a specific user. So, in a hospital information system, you may have doctor and patient as actors but not Dr. John, Mrs. Brown as actors.[44]

2.4.4.5 System

The scope of a system can be represented by a system (shape), or sometimes known as a system boundary. The use cases of the system are placed inside the system shape, while the actor who interact with the system are put outside the system. The use cases in the system make up the total requirements of the system.[44]

2.4.4.6 Include

An include relationship specifies how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.[44]

2.4.4.7 Extend

An extend relationship specifies how the behavior of the extension use case can be inserted into the behavior defined for the base use case.[44]

2.4.4.8 Dependency

A dependency relationship represents that a model element relies on another model element for specification and/or implementation.[44]

2.4.4.9 Generalization

A generalization relationship is used to represent inheritance relationship between model elements of same type. The more specific model element share the same specification with. the more general the model element but carries more details in extra.[44]

2.4.5 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc[45]. Here the grneral notations of activity diagram given below.

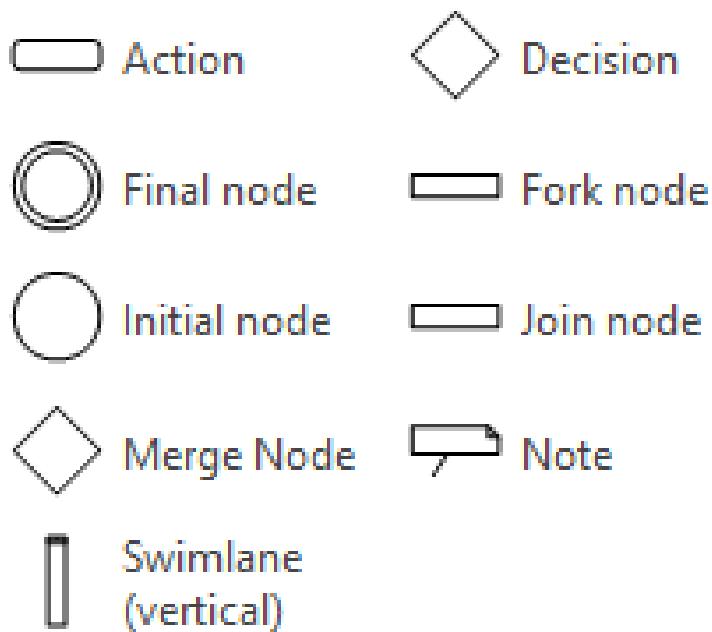


Figure 2.7: Activity Diagram Notations

2.4.5.1 Initial Node

Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point.[46]

2.4.5.2 Activity or Action State

Indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram.[46]

2.4.5.3 Connector Symbol

Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity; once the step is completed, the flow continues with the outgoing arrow.[46]

2.4.5.4 Decision

Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or merging of various flows with the symbol acting as a frame or container.[46]

2.4.5.5 Synchronization

Combines two concurrent activities and re-introduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line.[46]

2.4.5.6 Swim lane

Swimlanes group related activities into one column.[46]

2.4.5.7 Fork

Splits a single activity flow into two concurrent activities. Symbolized with multiple arrowed lines from a join. [46]

2.4.5.8 Note

Allows the diagram creators or collaborators to communicate additional messages that don't fit within the diagram itself. Leave notes for added clarity and specification.[46]

2.4.5.9 Final Node

Marks the end state of an activity and represents the completion of all flows of a process.[46]

2.4.6 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner[47]. Here the general notations of activity diagram given below.

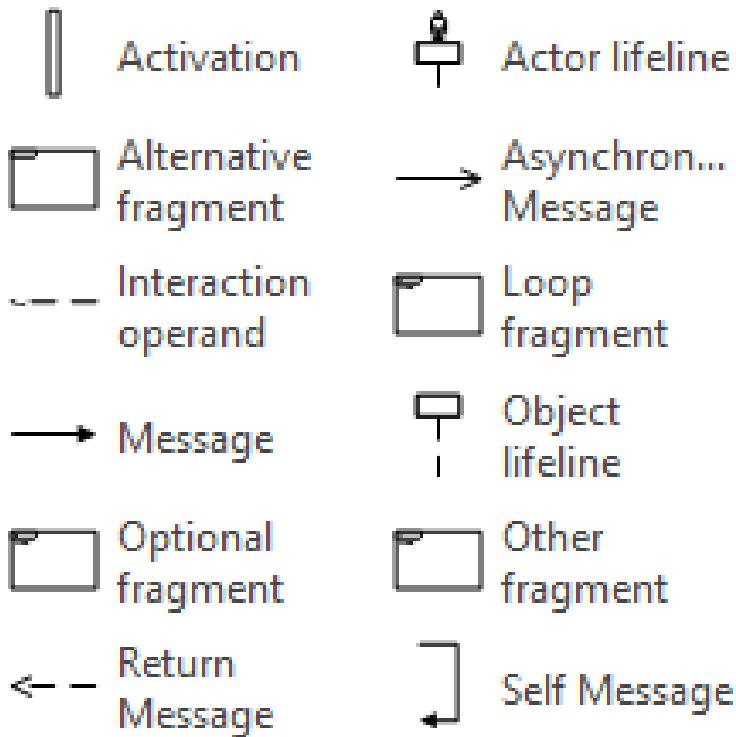


Figure 2.8: Sequence Diagram Notation

2.4.6.1 Class Roles or Participants

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.[48]

2.4.6.2 Activation or Execution Occurrence

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline. .[48]

2.4.6.3 Messages

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.[46]

2.4.6.4 Self Message

A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself. [48]

2.4.6.5 Return Message

A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline. [48]

2.4.6.6 Fragment

Alternative fragments are used to describe conditional activities. The other fragments describes single separated activity where optional fragment used to describe a activity which is not essential.[48]

2.4.6.7 Lifeline

Lifelines are vertical dashed lines that indicate the object's presence over time. [48]

2.4.6.8 Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets. [46]

2.4.7 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram[49]. Here the general notations of class diagram given below.

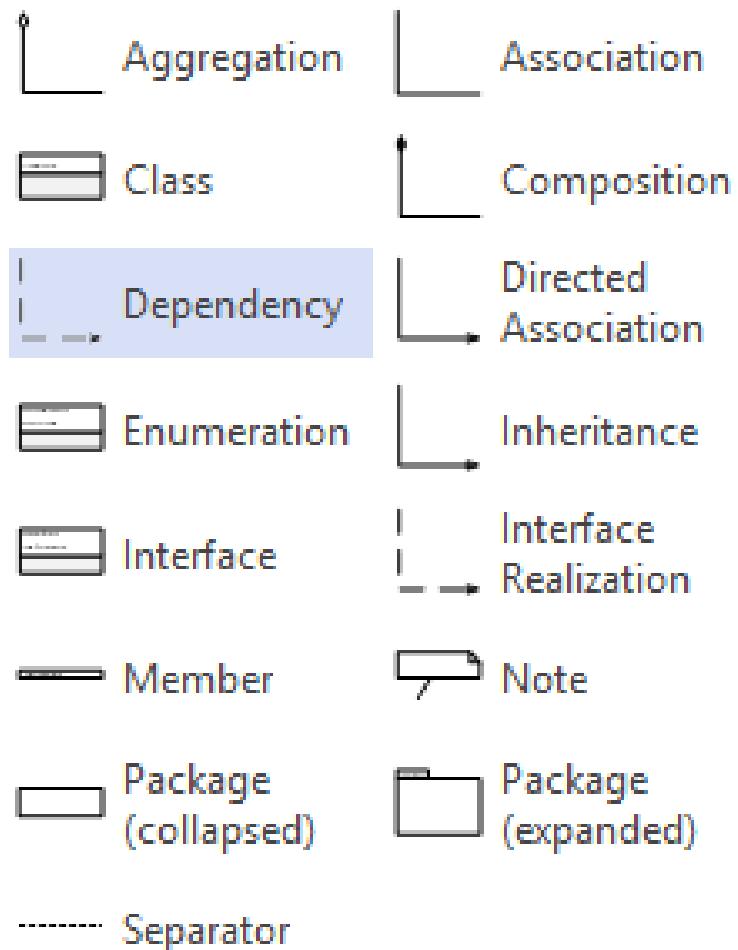


Figure 2.9: Class Diagram Notations

2.4.7.1 Classes

Classes represent an abstraction of entities with common characteristics. Associations represent the relationships between classes. Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition (left-aligned, not bolded, and lowercase), and write operations into the third.[50]

2.4.7.2 Active Classes

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.[50]

2.4.7.3 Visibility

Use visibility markers to signify who can access the information contained within a class. Private visibility, denoted with a - sign, hides information from anything outside the class partition. Public visibility, denoted with a + sign, allows all other classes to view the marked information. Protected visibility, denoted with a `~` sign, allows child classes to access information they inherited from a parent class.[50]

2.4.7.4 Association

Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each other.[50]

2.4.7.5 Multiplicity (Cardinality)

Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class. For example, one company will have one or more employees, but each employee works for just one company.[50]

2.4.7.6 Composition and Aggregation

Composition is a special type of aggregation that denotes a strong ownership between Class A, the whole, and Class B, its part. Illustrate composition with a filled diamond. Use a hollow diamond to represent a simple aggregation relationship, in which the "whole" class plays a more important role than the "part" class, but the two classes are not dependent on each other. The diamond ends in both composition and aggregation relationships point toward the "whole" class (i.e., the aggregation).[50]

2.4.7.7 Generalization

Generalization is another name for inheritance or an "is a" relationship. It refers to a relationship between two classes where one class is a specialized version of another. For example, Honda is a type of car. So the class Honda would have a generalization relationship with the class car.

In real life coding examples, the difference between inheritance and aggregation can be confusing. If you have an aggregation relationship, the aggregate (the whole) can access only the PUBLIC functions of the part class. On the other

hand, inheritance allows the inheriting class to access both the PUBLIC and PROTECTED functions of the superclass.[50]

2.4.8 State Chart Diagram

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Activity diagram explained in the next chapter, is a special kind of a Statechart diagram. As Statechart diagram defines the states, it is used to model the lifetime of an object. Here the general notations of state chart diagram given below[51].

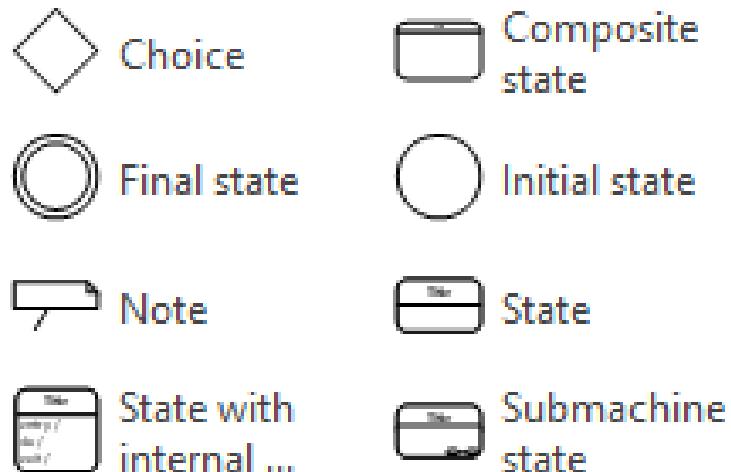


Figure 2.10: State Chart Diagram Notations

2.4.8.1 States

States represent situations during the life of an object. You can easily illustrate a state in SmartDraw by using a rectangle with rounded corners[52].

2.4.8.2 Transitions

A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it. A state can have a transition that points back to itself. [52].

2.4.8.3 Initial State

A filled circle followed by an arrow represents the object's initial state.[52].

2.4.8.4 Final State

An arrow pointing to a filled circle nested inside another circle represents the object's final state. [52].

2.4.8.5 Synchronization and Splitting of Control

A short heavy bar with two transitions entering it represents a synchronization of control. The first bar is often called a fork where a single transition splits into concurrent multiple transitions. The second bar is called a join, where the concurrent transitions reduce back to one. [52].

2.5 The Car Trading Management System Project

A car trading management system is a digital solution of executing car trading business to perform all required business processes based on customized software tools. The system provides several features to make this system effective for stakeholders.

2.5.1 Features

Let's look at the top feature of Car Trading Management System. Not all Car Trading Management Systems are the same features. So here are the most important features that we should provide.

2.5.1.1 Log-in and Access Restriction

A nominated user can log-in to the application for using further functionality. But there be access restriction due to provide information safety. Owner, manager, salesman etc. are not eligible to access all contents. Only owner can access all these stuffs. Other users are only eligible to access all those stuffs according to their designation.

2.5.1.2 Managing Business Processes

Eligible users can manage business processes like inventory information, sales, accounts management and so forth.

2.5.1.3 Analytics and Reporting

Only owner can access this feature. In this section, owner can reported his/her business progresses and evaluations through the desktop application which generates those reports by analyzing stored business data. This section can be use as a Decision Support System(DSS).

2.5.1.4 Marketing and Customer Relation

This section will covered by the Web Application from where consumer find trader for their dream car. By this, business to consumer interaction becomes more flexible and effective for growing the business.

2.5.1.5 Notification

Owner should notified by system after every important activities like a new car added in inventory or a car sold with his/her desired information.

2.5.2 Strategic Benefits

- Flexible to use.
- Extra employment not required.
- Reducing executional time and cost and boosts tangible and intangible profit.
- Help owner to take effective business decisions.
- Transparency among stuffs.
- No need of third party affiliation.
- Smartly handle accounts processes.
- Better customer interaction and easy to provide customer support.
- Modern and efficient approach of marketing.

2.5.3 Disadvantage of Digital System for Car Traders

- System maintenance expenses.
- Need well educated staffs to use this.
- Irregular use can affect analytics and reporting.

2.6 Conclusion

User has got informative basic information about this project. They understand about our application. How it is and why it is better. Information Systems is an academic study of systems with a specific reference to information and the complementary networks of hardware and software that people and organizations use to collect, filter, process, create and also distribute data. An emphasis is placed on an information system having a definitive boundary, users, processors, storage, inputs, outputs and the aforementioned communication networks.

Chapter 3

SYSTEM ANALYSIS & REQUIREMENT

3.1 Introduction

The project is a complete Enterprise Resource Planning(ERP) solution for car traders in Bangladesh. Here, performing system analysis is required in spite of identify end users, customers and what exactly they want the new system and how exactly they want to execute it in their professional life. It's required to perform also for determining an optimistic solution.

System analysis and design deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together. System analysts solve business problems through analysing the requirements of information systems and designing such systems by applying analysis and design techniques. This course deals with the concepts, skills, methodologies, techniques, tools, and perspectives essential for systems analysts. The practical component of COMP 361 is object oriented and use-case driven, requiring students to go through the steps of system analysis and design to solve a real-life business problem.

Systems Analysis and Design (SAD) is an exciting, active field in which analysts continually learn new techniques and approaches to develop systems more effectively and efficiently. However, there is a core set of skills that all analysts need to know no matter what approach or methodology is used. All information systems projects move through the four phases of planning, analysis, design, and implementation; all projects require analysts to gather requirements, model the business needs, and create blueprints for how the system should be built; and all

projects require an understanding of organizational behavior concepts like change management and team building. [53] [54]

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject System Analysis and Design (SAD), mainly deals with the software development activities. [55]

A collection of components that work together to realize some objectives forms a system. Basically there are three major components in every system, namely input, processing and output.[56] In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objective of the system demands that some output is produced as a result of processing the suitable inputs. A well-designed system also includes an additional element referred to as control that provides a feedback to achieve desired objectives of the system. [57]

3.2 System Analysis

The Merriam-Webster dictionary defines system analysis as the process of studying a procedure or business in order to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Another view sees system analysis as a problem-solving technique that breaks down a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose.[57] The field of system analysis relates closely to requirements analysis or to operations research. It is also an explicit formal inquiry carried out to help a decision maker identify a better course of action and make a better decision than she might otherwise have made.[57] System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. [33]

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system. System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle.

In the System Analysis and Design terminology, the system development life cycle also means software development life cycle. [33]

Practitioners of system analysis are often called up to dissect systems that have grown haphazardly to determine the current components of the system. This was shown during the year 2000 re-engineering effort as business and manufacturing processes were examined as part of the Y2K automation upgrades. Employment utilizing system analysis include system analyst, business analyst, manufacturing engineer, system architect, enterprise architect, software architect, etc.

While practitioners of system analysis can be called upon to create new systems, they often modify, expand or document existing systems (processes, procedures and methods). Researchers and practitioners rely on system analysis. Activity system analysis has been already applied to various research and practice studies including business management, educational reform, educational technology, etc. [54]

Systems analysis the process of observing systems for troubleshooting or development purposes. It is applied to information technology, where computer-based systems require defined analysis according to their makeup and design. [33]

In IT, systems analysis can include looking at end-user implementation of a software package or product; looking in-depth at source code to define the methodologies used in building software; or taking feasibility studies and other types of research to support the use and production of a software product, among other things. Systems analysis professionals are often called upon to look critically at systems, and redesign or recommend changes as necessary. Inside and outside of the business world, systems analysts help to evaluate whether a system is viable or efficient within the context of its overall architecture, and help to uncover the options available to the employing business or other party. [22]

Systems analysts are different than systems administrators, who maintain systems day to day, and their roles generally involve a top-level view of a system to determine its overall effectiveness according to its design. [23]

New information systems introduce change to the organization and its people. Leading a successful organizational change effort is one of the most difficult jobs that someone can do. Understanding what to change, knowing how to change it, and convincing others of the need for change require a wide range of skills. These skills can be broken down into six major categories: technical, business, analytical, interpersonal, management, and ethical. Analysts must have the technical skills to understand the organizations existing technical environment, the new systems technology foundation, and the way in which both can be fit into an integrated technical solution. Business skills are required to understand how IT

can be applied to business situations and to ensure that the IT delivers real business value. Analysts are continuous problem solvers at both the project and the organizational level, and they put their analytical skills to the test regularly. Often, analysts need to communicate effectively, one-on-one with users and business managers (who often have little experience with technology) and with programmers (who often have more technical expertise than the analyst does). They must be able to give presentations to large and small groups and to write reports. Not only do they need to have strong interpersonal abilities, but they also need to manage people with whom they work, and they must manage the pressure and risks associated with unclear situations. Finally, analysts must deal fairly, honestly, and ethically with other project team members, managers, and system users. Analysts often deal with confidential information or information that, if shared with others, could cause harm (e.g., dissent among employees); it is important for analysts to maintain confidence and trust with all people. [32]

As organizations and technology have become more complex, most large organizations now build project teams that incorporate several analysts with different, but complementary, roles. In smaller organizations, one person may play several of these roles. Here we briefly describe these roles and how they contribute to a systems development project. The systems analyst role focuses on the IS issues surrounding the system. This person develops ideas and suggestions for ways that IT can support and improve business processes, helps design new business processes supported by IT, designs the new information system, and ensures that all IS standards are maintained. The systems analyst will have significant training and experience in analysis and design and in programming. [33]

The business analyst role focuses on the business issues surrounding the system. This person helps to identify the business value that the system will create, develops ideas for improving the business processes, and helps design new business processes and policies. The business analyst will have business training and experience, plus knowledge of analysis and design.[34]

3.3 Information Gathering

Information gathering is the most initial part of a system analysis. To build an effective and efficient system, it is required to learn and understand the system, how it executes and also about related stuffs like human resources etc. There are several methods to gather information.

- Interactive Method, by which analyst fetch information from audiences and end users through interviewing, etc. Actually this method is direct interac-

tion between analyst and end users.

- Unobtrusive Method, by which analyst observe the system and audiences without directly interacting with them.

We followed both of these method to gather information from targeted audiences.

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. [58] Feasibility analysis (FA, also called feasibility study) is used to assess the strengths and weaknesses of a proposed project and present directions of activities which will improve a project and achieve desired results. The nature and components of feasibility studies depend primarily on the areas in which analyzed projects are implemented.[59] In this section of the interactive teachingl tool we will present theoretical background of feasibility analysis for projects in the field of sustainable development and spatial planning with particular emphasis on the theory and application of geographic information science and technology. Spatial information and its availability as well as analysis and visualization capabilities will, therefore, play an important role in the analyzed projects. Therefore, references made to the INSPIRE directive will be one of the important points of the feasibility study. [36] Feasibility analysis is the process of confirming that a strategy, plan or design is possible and makes sense. This can be used to validate assumptions, constraints, decisions, approaches and business cases. The following are common types of feasibility analysis. [31] Feasibility studies can be used in many ways but primarily focus on proposed business ventures. Farmers and others with a business idea should conduct a feasibility study to determine the viability of their idea before proceeding with the development of a business. Determining early that a business idea will not work saves time, money and heartache later. [30]

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

3.4.1 Technical Feasibility

Technical feasibility study is the complete study of the project in terms of input, processes, output, fields, programs and procedures. It is a very effective tool for long term planning and trouble shooting. The technical feasibility study should most essentially support the financial information of an organization. Encouraging companies to do some strategic thinking around disruptive technologies and taking time to look outside of their own organisations for inspiration and for guidance. This could include a comprehensive literature review to determine the current status of scientific or technical knowledge in the area. An investigation of solutions that may be available from the 3rd level sector. This may include searches of the Knowledge Transfer Ireland web-site and discussions with one or more of the Technology Centres/Gateways in order to identify Research Partners and existing IP if appropriate. Prototype development if appropriate in order to assist in the evaluation of project options. (Note that this scheme is not intended to support the actual development of new products, processes or services). Analysis of the commercial feasibility of the project resources required, risks and potential return on investment. This could include interaction with potential customers. Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out. Technical feasibility is the process of validating the technology assumptions, architecture and design of a product or project. The following are common types of technical feasibility. [59]

3.4.2 Operational Feasibility

Operational feasibility refers to the measure of solving problems with the help of a new proposed system. It helps in taking advantage of the opportunities and fulfills the requirements as identified during the development of the project. It takes care that the management and the users support the project. Operational feasibility refers to the measure of solving problems with the help of a new proposed system. It helps in taking advantage of the opportunities and fulfills the requirements as identified during the development of the project. It takes care that the management and the users support the project. In situations like this, technical feasibility is a matter not only of ensuring the product in question can be built, but that it can be elegantly integrated into existing systems. No one would argue with the importance of this. I've never been in a meeting where someone has said Hey! Enough with the feasibility talk - does it really matter if we can integrate this

into our product or not? [31] But something strange happens when the conversation turns to integrating a product into a company's human systems. How will the product will be adopted culturally and politically? Are employees likely to embrace it? How will it be run? How will it support or conflict with the work and ambitions of internal product teams and IT? Are they for it or against it?

In other words, is it operationally feasible? [58]

This is a perfectly valid line of questioning, but somehow it usually doesn't feel like it's our business. It's political and sticky. Indeed, a client might feel they need tangible, advanced progress in order to get support internally. But every day spent working on something in isolation of the wider ecosystem is a risk. The major goal of this article is to discuss about Operational Feasibility. With Operational Feasibility, we consider whether the current system become implemented using existing manpower and resources or not. To find functional feasibility we determine whether the proposed solution can participate in existing operations and whether the right information in the right time is provided to end users. Operational feasibility of our proposed system is modularized. [59]

3.4.3 Economic Feasibility

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study.[58] [59] Our economists work alongside engineers and scientists and use cutting-edge statistical and programming software and state-of-the-art GIS and GPS technology to provide clients with objective, economic analyses. Examples of the detailed economic analyses we deliver include the economic impact of changes in water supply on agriculture, cost-benefit and economic impact analysis of invasive species, and utility cost of service studies and setting rates. [30] Our environmental economics expertise extends to estimating the dollar value of ecosystem services, evaluating the cost-effectiveness of ecosystem management strategies, determining the community and social benefits of green development practices and policies, and assessing the economic impacts of environmental policies and programs. The key to successfully using environmental information is to utilise rigorous, quantitative frameworks that align the data analysis with the decision makers. Our suite of decision-science services - including business solutions, risk management, ecological modeling, and science communications - can help clients to improve decision making by collecting the right data, analysing it with an innovative approach, interpreting it with the right quantitative tools, and effectively communicating the

results to stakeholders. Our teams work collaboratively to provide the clearest picture of client data and identify ways to utilise this information to maximise opportunities and limit liabilities.

3.5 Result of System Analysis

The result of system analysis is always indicates the problem definitions. After performing all the tasks of analyzing the system, finally we identified some problem currently happening on a car trading business. The definitions of the problems are given below.

3.5.1 Data Store

Storing data on hard paper does not allows personnel easily to access data from hard paper because lots of paper stored in cabinet and it is time consuming for data processing on due time. In a car shop, their be too many documents and it is too hard to any personnel to use and manipulate data from these hard paper documents or registers.

3.5.2 Data Insecurity

Storing data on hard paper does not ensure the security of business data. By following the value of business data, it is much unsafe. It allows bad peoples to form a conspiracy against owner which is much harmful for the business.

3.5.3 Slower Business Processes

In a shop, the absence of digital solution occurs business processes much slower. Currently owners use some computer aided tools like MS Excel but that not as productive as a digital solution. Slower business process occurs some potential loss and most of the time these kind of losses overlooked by personnels.

3.5.4 Lack of Transparency

By using this existing system, it is tough to measure the accurate commission of salesman. If a salesman betrays the owner, it can not be detected instantly

through this system. This is not a rare scenario that salesman sale a car and inform wrong price to owner. So, it is mandatory for trader to ensure transparency on business.

3.5.5 Accounts Issues

Unfortunately but true, a car shop's accounts system performed by only manager and owner and there be no accountant most of the shops. This is why, financial management are not performing in structured and standard way which causes some potential losses. Sometimes personnels are irregular to store business transactions.

3.5.6 Cost Ineffective

By using hard paper on this business, so many unnecessary expenses happens which is not acceptable in smart business management concept. So many papers to purchase and so many storage in room is necessary. It tends to increase several expenses including showroom rental expenses which is performs too much tangible loss in business.

3.5.7 Old Style of Marketing and CRM

Most of the cars sold via affiliations which is not optimistic at this modern age. Here the problem seems that gap between customer and trader is grater and much worrying.

3.5.8 No Decision Support System

By using this existing system, decision makers or owner can not take decisions faster because this system does not support to give business analysis directly. One way to perform this in existing system is, appointing someone to collect data from several documents of business. But this tends to occur loss of around 20-30 percent per year, sometimes more.

3.6 Requirement Analysis

In systems engineering and software engineering, requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.[15] Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Requirement Analysis, also known as Requirement Engineering, is the process of defining user expectations for a new software being built or modified. In software engineering, it is sometimes referred to loosely by names such as requirements gathering or requirements capturing. Requirements analysis encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. [60]

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management. [37]

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users.[38] Requirements analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Requirements can be functional and non-functional.[39]

Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. Historically, this has included such things as holding interviews, or holding focus groups (more aptly named in this context as requirements workshops) and creating requirements lists. More modern techniques include pro-

totyping, and use cases. Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced.[60] Systematic requirements analysis is also known as requirements engineering. It is sometimes referred to loosely by names such as requirements gathering, requirements capture, or requirements specification. The term requirements analysis can also be applied specifically to the analysis proper, as opposed to elicitation or documentation of the requirements, for instance.[37]

3.6.1 System Requirement

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash. [33] System requirements are also known as minimum system requirements. For packaged products, system requirements are often printed on the packaging. For downloadable products, the system requirements are often indicated on the download page. System requirements can be broadly classified as functional requirements, data requirements, quality requirements and constraints. They are often provided to consumers in complete detail. System requirements often indicate the minimum and the recommended configuration. The former is the most basic requirement, enough for a product to install or run, but performance is not guaranteed to be optimal. The latter ensures a smooth operation. [34] Hardware system requirements often specify the operating system version, processor type, memory size, available disk space and additional peripherals, if any, needed. Software system requirements, in addition to the aforementioned requirements, may also specify additional software dependencies (e.g., libraries, driver version, framework version). Some hardware/software manufacturers provide an upgrade assistant program that users can download and run to determine whether their system meets a product's requirements. [54]

3.6.1.1 Non Functional Requirement

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for

implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements. [33] Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of "system shall do [requirement]", an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of "system shall be [requirement]", an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed. [34] Non-functional requirements are often called "quality attributes" of a system. Other terms for non-functional requirements are "qualities", "quality goals", "quality of service requirements", "constraints" and "non-behavioral requirements". [54] A system may be required to present the user with a display of the number of records in a database. This is a functional requirement. How up-to-date [update] this number needs to be, is a non-functional requirement. If the number needs to be updated in real time, the system architects must ensure that the system is capable of updating the [displayed] record count within an acceptably short interval of the number of records changing.

According to the existing system, current trend of executing car trading business and future of this business area[40], we identified these following non-functional requirements for the new system. [35]

- Maintainability : Our application is an action designed to achieve maintainability.
- Re-usability : Our application is an action designed to achieve re-usability.
- Flexibility : Our application is an action designed to achieve flixibility.
- Efficiency : Our application is an action designed to achieve efficiency.
- Reliability : Our application should be reliable.
- Usability : Our application should be useable.
- Consistency: Our application should be consistent.
- Scalability : Our application should be scalable.
- Security : Our application should be highly secured.

3.6.1.2 Functional Requirement

According to ReQtest, a functional requirement simply tells a system to do something specific. On its own, this definition does not really suggest much, but lets take a real life example to provide more clarity. Functional requirements essentially describe what the product must do or the steps it is going to have to take to perform that action. For instance, everytime a customer places an order online, a confirmation email is sent to them. As a general rule, functional requirements are usually written out as shall statements.

When we interacted with car traders, these following requirements identified commonly. [33]

- Manage stock or inventory information.
- Vehicle sales from software.
- Manage stuff or employee information.
- Automated accounts system.
- Decision support system.
- Website to reach out customers.

3.7 Conclusion

Systems analysts are different than systems administrators, who maintain systems day to day, and their roles generally involve a top-level view of a system to determine its overall effectiveness according to its design. We inspected several car shops, interacted with traders, observed the whole system and we analyzed the system by also comparing with modern day business trends and the worldwide car trading approaches. All these functional and non-functional requirements we specified are totally based on their existing technologies, human resources, business model and business chain and also considering the future of this business area in the upcoming age of AI.

Chapter 4

METHODOLOGY

4.1 Introduction

This chapter will describe methodology and the technique that will be followed by us to develop our application. The methodology is the general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it. These methods, described in the methodology, define the means or modes of data collection or, sometimes, how a specific result is to be calculated. Methodology does not define specific methods, even though much attention is given to the nature and kinds of processes to be followed in a particular procedure or to attain an objective.

When proper to a study of methodology, such processes constitute a constructive generic framework, and may therefore be broken down into sub-processes, combined, or their sequence changed.

A paradigm is similar to a methodology in that it is also a constructive framework. In theoretical work, the development of paradigms satisfies most or all of the criteria for methodology. An algorithm, like a paradigm, is also a type of constructive framework, meaning that the construction is a logical, rather than a physical, array of connected elements.

Any description of a means of calculation of a specific result is always a description of a method and never a description of a methodology. It is thus important to avoid using methodology as a synonym for method or body of methods. Doing this shifts it away from its true epidemiological meaning and reduces it to being the procedure itself, or the set of tools, or the instruments that should have been its outcome. A methodology is the design process for carrying out research or the development of a procedure and is not in itself an instrument, or method, or

procedure for doing things.[61]

4.2 Methodology

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.

4.3 Developing the Report Generation and Analytical Functionalities

The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. The systems development lifecycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.

4.4 Use of Object Oriented Analysis & Design

Object-oriented analysis and design (OOAD) is a popular technical approach for analyzing and designing an application, system, or business by applying object-oriented programming, as well as using visual modeling throughout the development life cycles to foster better stakeholder communication and product quality[62].

In the system analysis or object-oriented analysis phase of software development, the system requirements are determined, the classes are identified and the relationships among classes are identified. After the analysis phase, the conceptual model is developed further into an object-oriented model using object-oriented design (OOD). In OOD, the technology-independent concepts in the analysis model are mapped onto implementing classes, constraints are identified, and interfaces are designed, resulting in a model for the solution domain.

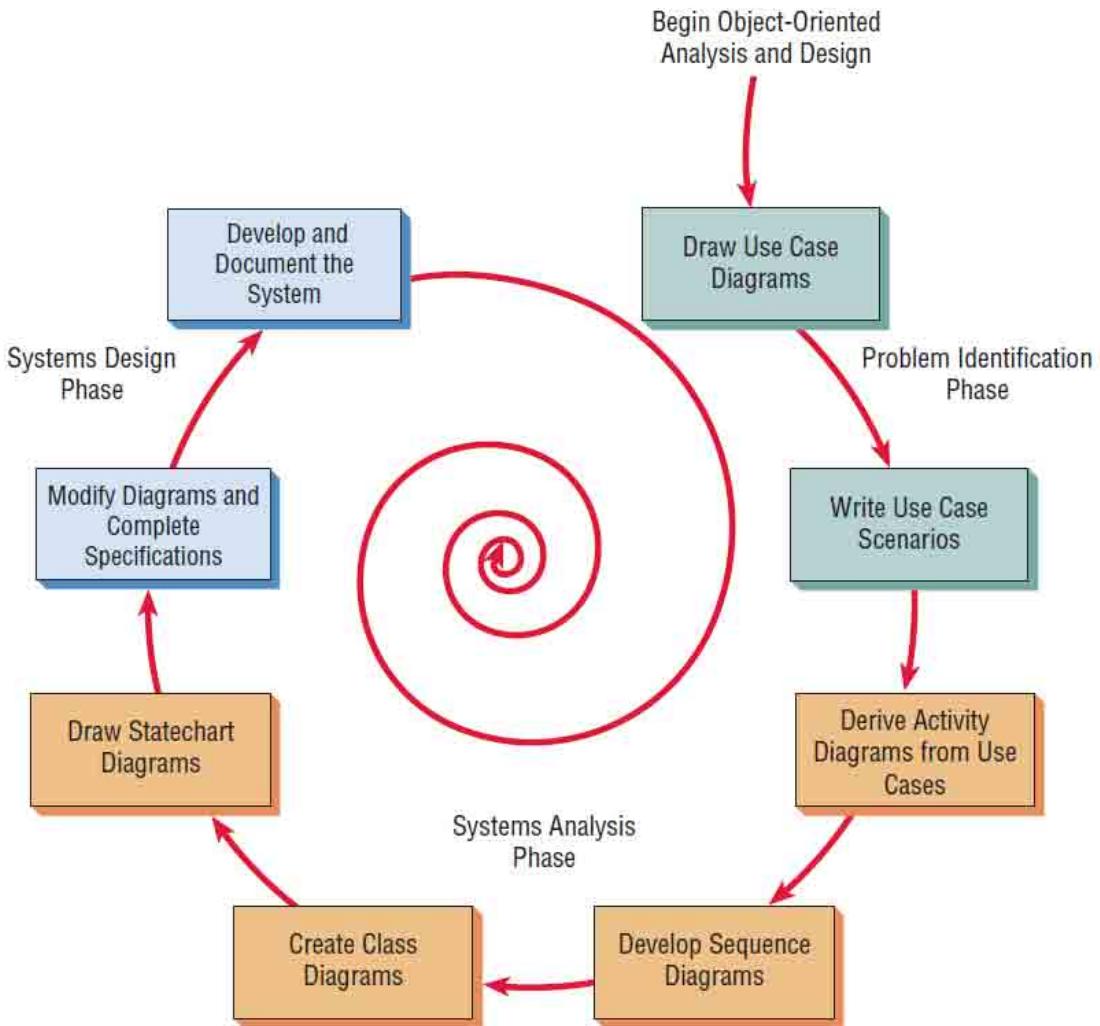


Figure 4.1: Object Oriented Analysis & Development Model

4.4.1 What is Object-Oriented Analysis

To define object-oriented analysis we must first define what we mean by an object. The definition of an object, according to most dictionaries, is 'a tangible, material thing.' Drilling down a bit more to the realm of computer science, an object can be most anything in a programmatic sense, from a variable or data model to a function, class, or method. Moving even deeper into the realm of object-oriented programming, an object is an instance of a thing that typically represents a real world object and has all the same types of characteristics (properties), behaviors (methods), and states (data). When discussing OOAD concepts, an object most closely resembles the object-oriented programming version of an object, in that it is a representation of a real world object with behaviors, characteristics, and states.

With that out of the way, we can define object-oriented analysis (OOA). In short, OOA is an iterative stage of analysis, which takes place during the software development life cycle, that aims to model the functional requirements of the software while remaining completely independent of any potential implementation requirements. To accomplish this task via OOAD practices, an object-oriented analysis will focus everything through the lens of objects. This forces OOA to combine all behaviors, characteristics, and states together into one analysis process, rather than splitting them up into separate stages, as many other methodologies would do.

To accomplish this goal, a typical OOA phase consists of five stages:

- Find and define the objects.
- Organize the objects.
- Describe how the objects interact with one another.
- Define the external behavior of the objects.
- Define the internal behavior of the objects.

For example, a typical implementation of OOA is to create an object model for an application. The object model might describe the names, relationships, behaviors, and characteristics of each object in the system. With this information established for each object, the design process that follows is much simpler[?].

4.4.2 What is Object-Oriented Design

The process of object-oriented design is really just an extension of the object-oriented analysis process that preceded it, except with a critical caveat: the consideration and implementation of constraints. For example, with an analyzed object in hand, such as an object model, we must now consider how that object would actually be designed and implemented, which will often require the application of constraints, such as software or hardware platforms, time and budgetary limitations, performance requirements, developer aptitude, and so forth.

Put another way, the OOD process takes the theoretical concepts and ideas planned out during the OOA stage, and tries to find a way to design and tangibly implement them, usually via code using whatever language and platforms the development team has settled upon. If OOA is the what, then OOD is the how[63].

4.4.3 Advantages and Disadvantages of Object-Oriented Analysis and Design

Since OOAD principles are fundamentally based on real world objects, it's quite easy for everyone on the team to quickly understand what an object name means or how a particular behavior, well, behaves. This makes the overall development life cycle a much smoother process, particularly if your team needs to frequently interact with customers or other non-technical users about the objects and components in the system. In such cases, most people still understand how system components and modelled objects work when they're based on real world objects and ideas.

And everything within OOAD revolves around the concept of objects (specifically, the object-oriented variety), one of the biggest advantages of OOAD is that it encourages planning and development of systems that are truly independent of one another. Just like a class written using object-oriented techniques, all the systems and objects produced during an OOAD development life cycle can be mixed and matched as necessary, since they will ideally be built as completely self-contained entities.[63]

Every object in our real world comes with some advantages and disadvantages and this methodology is not different from it. These followings are the advantages of OOAD. Procedural application can't be developed in this methodology. Whereas this approach of development is too old to use, we can ignore this kind of disadvantage. One thing, it's too hard to develop simple system with this methodology.

4.4.4 Why we choose this methodology

Actually, Object Oriented System Analysis and Design is the most modern, effective and efficient approach for developing a business application at this 21st century. The main reason to choose this is that we developed this system with C Sharp 7.0 programming language which is the most modern, strongly typed and highly managed object oriented programming language. Here some more reasons briefed for which we choose this.

- The problems modeled themselves to classes.
- System can be added gradually, one sub-system at a time.
- Reuse of previous written software is possible.
- Acceptable to tackle the difficult problems first.

- Highly maintainable.
- Easy to understand.
- Management is comparatively optimistic than other methodologies.

4.5 A brief look on the phases of OOSAD

From the figure, we observed that how exactly the process works step by step. Now here we will see what exactly happens on those phases of this development methodology.

4.5.1 Define Use case model

In this phase the analyst identifies the actors and the major events initiated by the actors. Often the analyst will start by drawing a diagram with stick figures representing the actors and arrows showing how the actors relate. This is called a use case diagram and it represents the standard flow of events in the system. Then an analyst typically writes up a use case scenario, which describes in words the steps that are normally performed.

4.5.2 Drawing Activity Diagrams

During the systems analysis phase, the analyst will draw Activity Diagrams, which illustrate all the major activities in the use case. This is the phase where system analyst or system architect actually solves business problems and make logics clear to other stakeholders. It also allows stakeholders to be transparent among them during project development.

4.5.3 Drawing Sequence Diagrams

During the systems analysis phase, the analyst will draw sequence Diagrams, which illustrate all the major activities in the activity diagrams. In addition, the analyst will create one or more sequence diagrams for each use case, which show the sequence of activities and their timing. This is an opportunity to go back and review the use cases, rethink them, and modify them if necessary.

4.5.4 Develop class diagrams

The nouns in the use cases are objects that can potentially be grouped into classes. For example, every automobile is an object that shares characteristics with other automobiles. Together they make up a class. This is in design phase in this life cycle. These class diagrams allows stakeholders to make the project comparatively more reusable and maintainable.

4.5.5 Draw State Chart Diagram

The class diagrams are used to draw statechart diagrams, which help in understanding complex processes that cannot be fully derived by the sequence diagrams. The statechart diagrams are extremely useful in modifying class diagrams, so the iterative process of UML modeling continues.

4.5.6 Modify Diagrams & Complete Specifications

Systems design means modifying the existing system and that implies modifying the diagrams drawn in the previous phase. These diagrams can be used to derive classes, their attributes, and methods (methods are simply operations). The analyst will need to write class specifications for each class including the attributes, methods, and their descriptions. They will also develop methods specifications that detail the input and output requirements for the method, along with a detailed description of the internal processing of the method.

4.5.7 Develop & Document The System

UML is, of course, a modeling language. An analyst may create wonderful models, but if the system isn't developed there is not much point in building models. Documentation is critical. The more complete the information you provide the development team through documentation and UML diagrams, the faster the development and the more solid the final production system.

4.6 Contribution

In Methodology our contribution is we tried to follow the rules and regulations of Object Oriented Analysis & Design methodology. This is rich, smart and modern approach to develop a system and make the system much more effective and

efficient. All the business problems solved by process wise like inventory, sales etc. Each process assumed as component and all diagrams designed component wise in spite of making the design more intellectual to other stakeholders and it boosts scalability and maintainability of the project.

4.7 Conclusion

The methodology we used is the Object Oriented Analysis & Design approach because it is the most suitable method for this project. This method has more advantages when compared with other methods by focusing our objectives. And using of this methodology our project becomes highly maintainable, reliable and quality product. The next chapter covers the design part of this project.

Chapter 5

PROJECT DESIGN

5.1 Introduction

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designers goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software. What is project design in project management? Project design is an early phase of the project where a projects key features, structure, criteria for success, and major deliverables are all planned out. The point is to develop one or more designs which can be used to achieve the desired project goals. Stakeholders can then choose the best design to use for the actual execution of the project.

The project design phase might generate a variety of different outputs which is defined using Unified Modeling Language(UML) according to the development methodology.

5.2 Use Cases

In this project we designed all the use case diagrams by component wise in spite of making it more understandable, also known as UML 1.1. Let's discuss it below.

This is the skyview of the project. Also known as UML 1.0 .

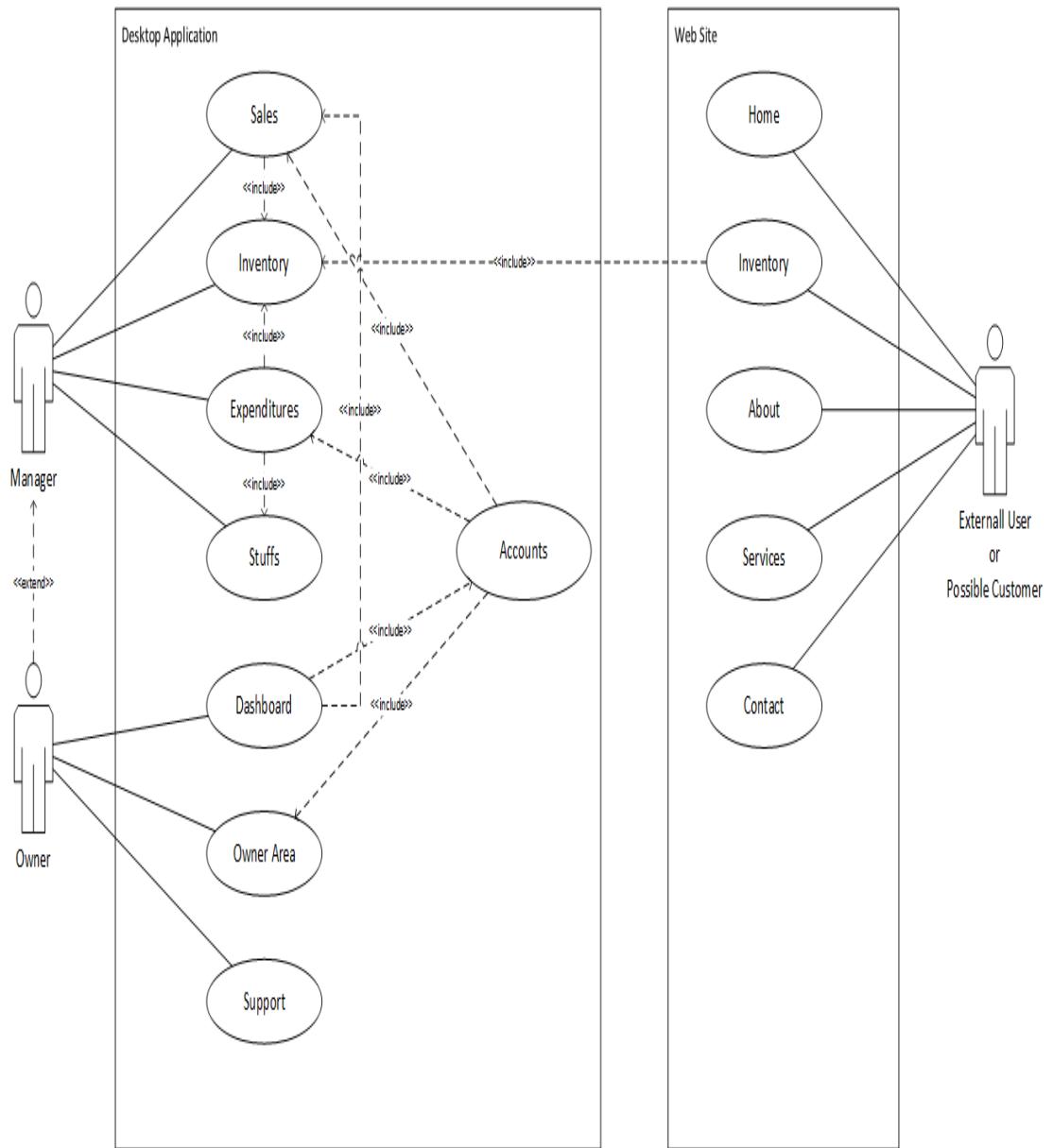


Figure 5.1: Use Case Diagram(Sky View) of the System

This is the use case diagram of authentication component, also known as log in component.

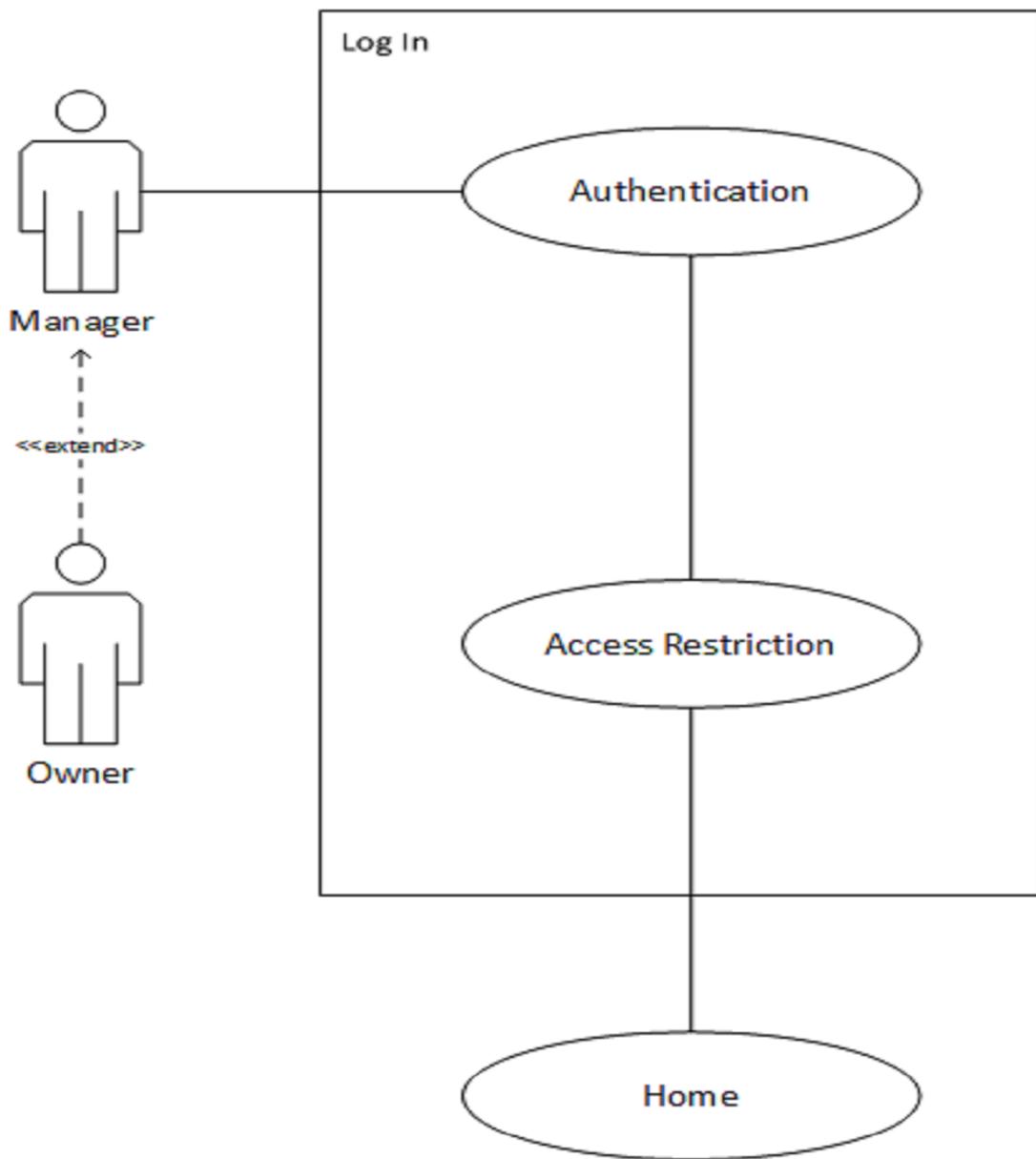


Figure 5.2: Use Case Diagram of Authentication Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_AUTH
Application	Car Trading Information Management System V4.0
Use Case Name	Authentication Component
Use Case Description	In spite of ensuring the application security, business data security, users are required to authenticate themselves. This component looks mostly similar to others by only providing user id and password by user. The authentication event acquires users role in the business or shop and define access restriction for further components usage.
Primary Actor	Owner, Manager
Precondition	None
Trigger	Application Start-up
Basic Flow	This is the first component, which appears after executing the application. The Authentication Request Form seeks users credentials. After providing credentials (user id and password) by user, the Authentication system starts executing and retrieve users role via his/her designation. After completing successful authentication, it triggers an automated event Access Restriction, which defines access restriction for further components usage and navigate to main page of application. Otherwise, it return to the Authentication Request Form and seek for credentials. Any user able to try up to 5 times after authentication failure.
Alternate Flow	⁶⁷ In case of any kind of failure, the component breaks the communication from database.

This is the use case diagram of home navigation component.

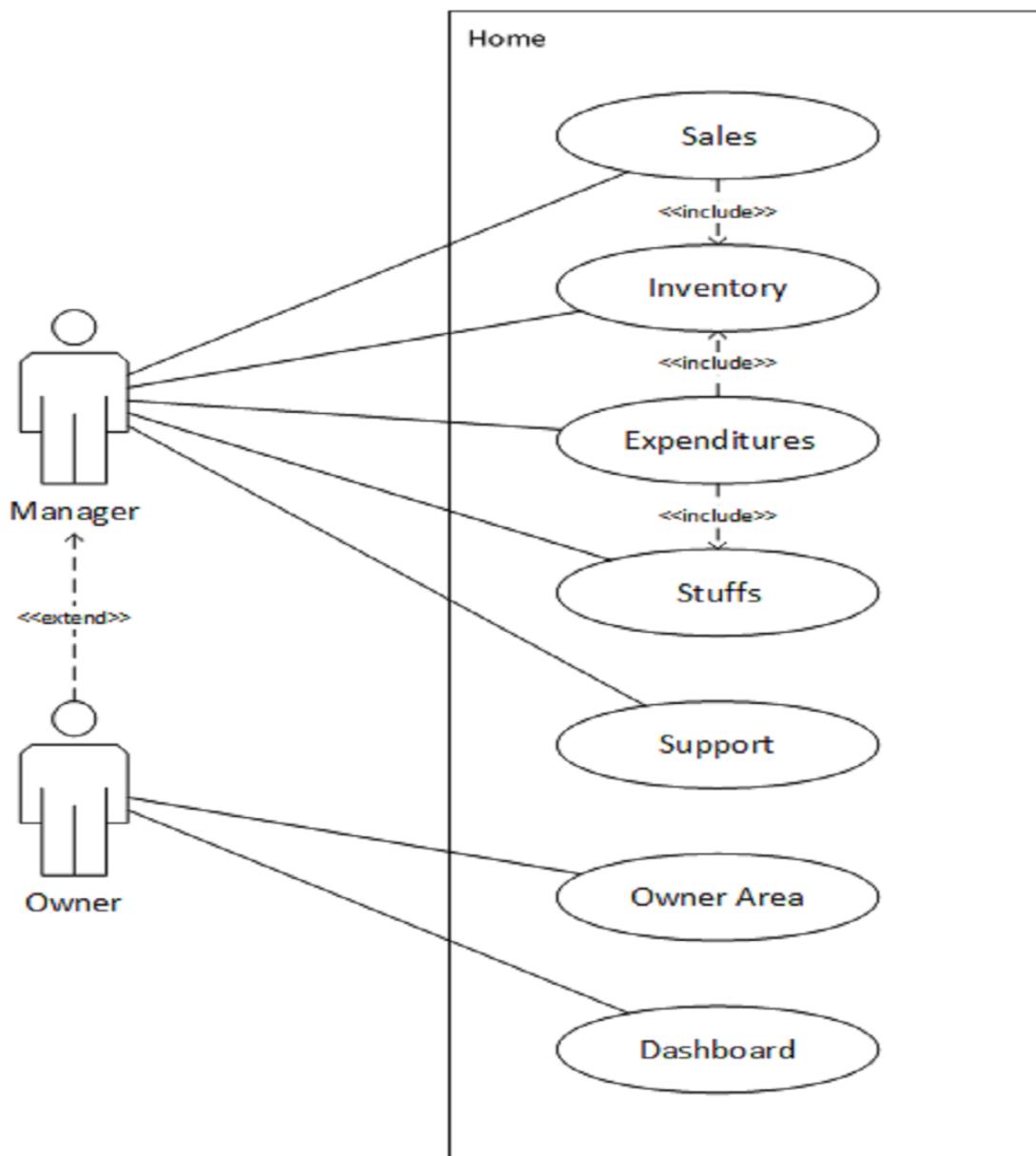


Figure 5.3: Use Case Diagram of Home Navigation Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_HOME
Application	Car Trading Information Management System V4.0
Use Case Name	Home Component
Use Case Description	Home page is the initial page after successful login or authentication. This component allows user to navigate to different components of the system like inventory, sales etc. This can also be called as main menu of the system. According to their business operations, all users are not allowed to use every components. There be some access restriction, but only owner(s) can access all these components.
Primary Actor	Owner, Manager
Precondition	Authorized Personnel
Trigger	Successful log in
Basic Flow	After a successful authentication, user with specific role is able to navigate into this system. This component allows user to navigate to different components of the system. The navigation performed by using user role at car shop also. Only owner(s) can navigate all the components where manager is restricted within inventory, sales, expenditures and manage stuffs components.
Alternate Flow	There is no other action to perform this in spite of making the system secure for the business.

This is the use case diagram of inventory component.

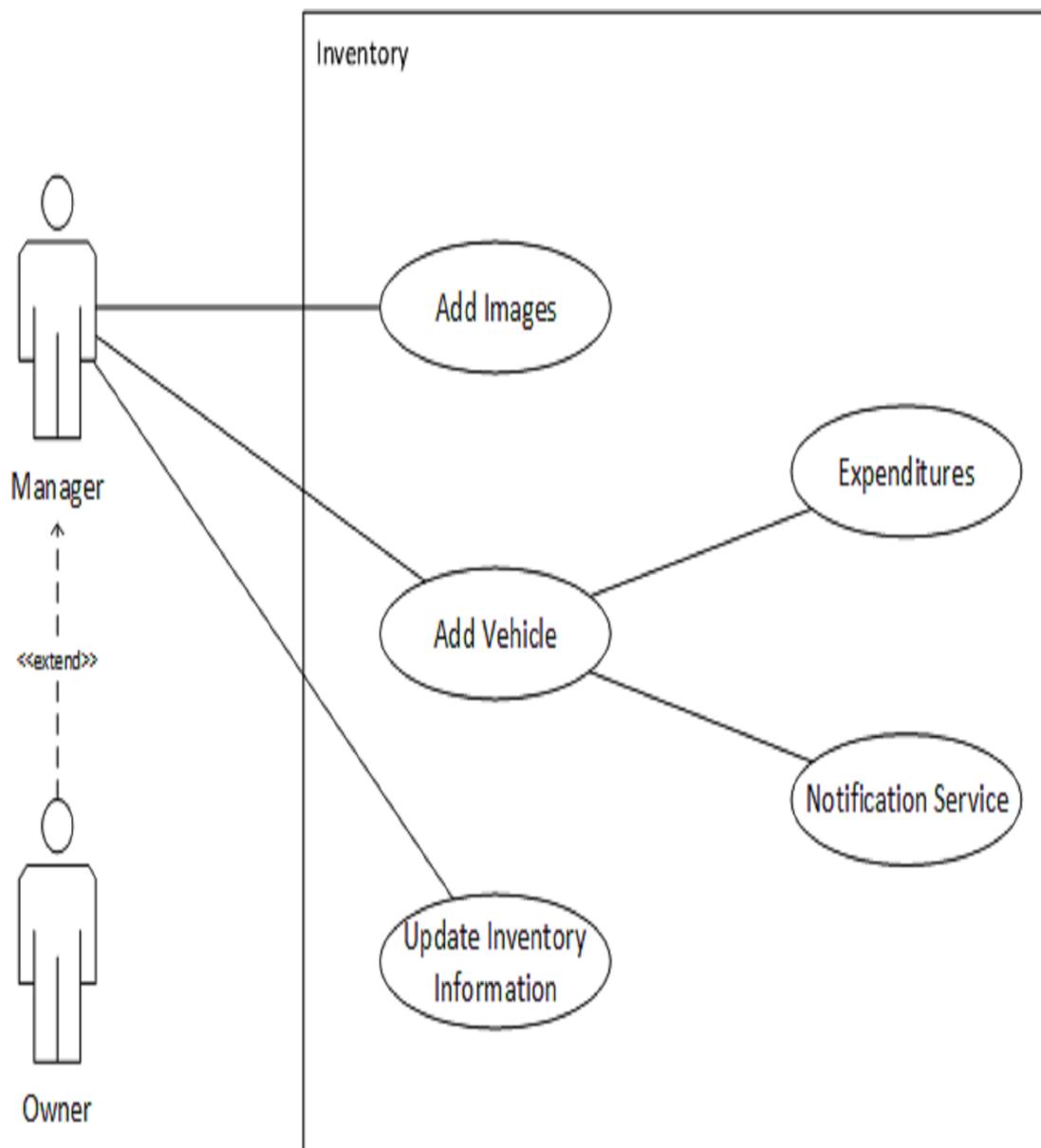


Figure 5.4: Use Case Diagram of Inventory Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_INV
Application	Car Trading Information Management System V4.0
Use Case Name	Inventory Component
Use Case Description	This is the component where product management of a business has performed. In a car shop, the products be import from different countries, mostly Japan. Whereas a vehicle is a fixed asset to customer, so there're too many information to store for legal issues. This component in this system manages vehicle or product information only as per business strategy of a car shop. Changes in inventory record will also notify to owner(s) via email.
Primary Actor	Owner, Manager
Precondition	Authorized Personnel
Trigger	Inventory Button Click
Basic Flow	After appearing inventory page, firstly appears Add Inventory form where user can insert a new vehicle. There be two more tabbed item, Add Images and Update Information. In the page of Add Images, user is able to add vehicle image, one at a time, by using unique engine number of a vehicle. In the Update Information, user is able to make changes in inventory record. Not all inventory attributes of making change allowed due to security issue.
Alternate Flow	In case of any kind of failure, the component ⁷¹ breaks the communication from database with error notifications and return to the user so that threads cannot effect informa-

This is the use case diagram of sales component.

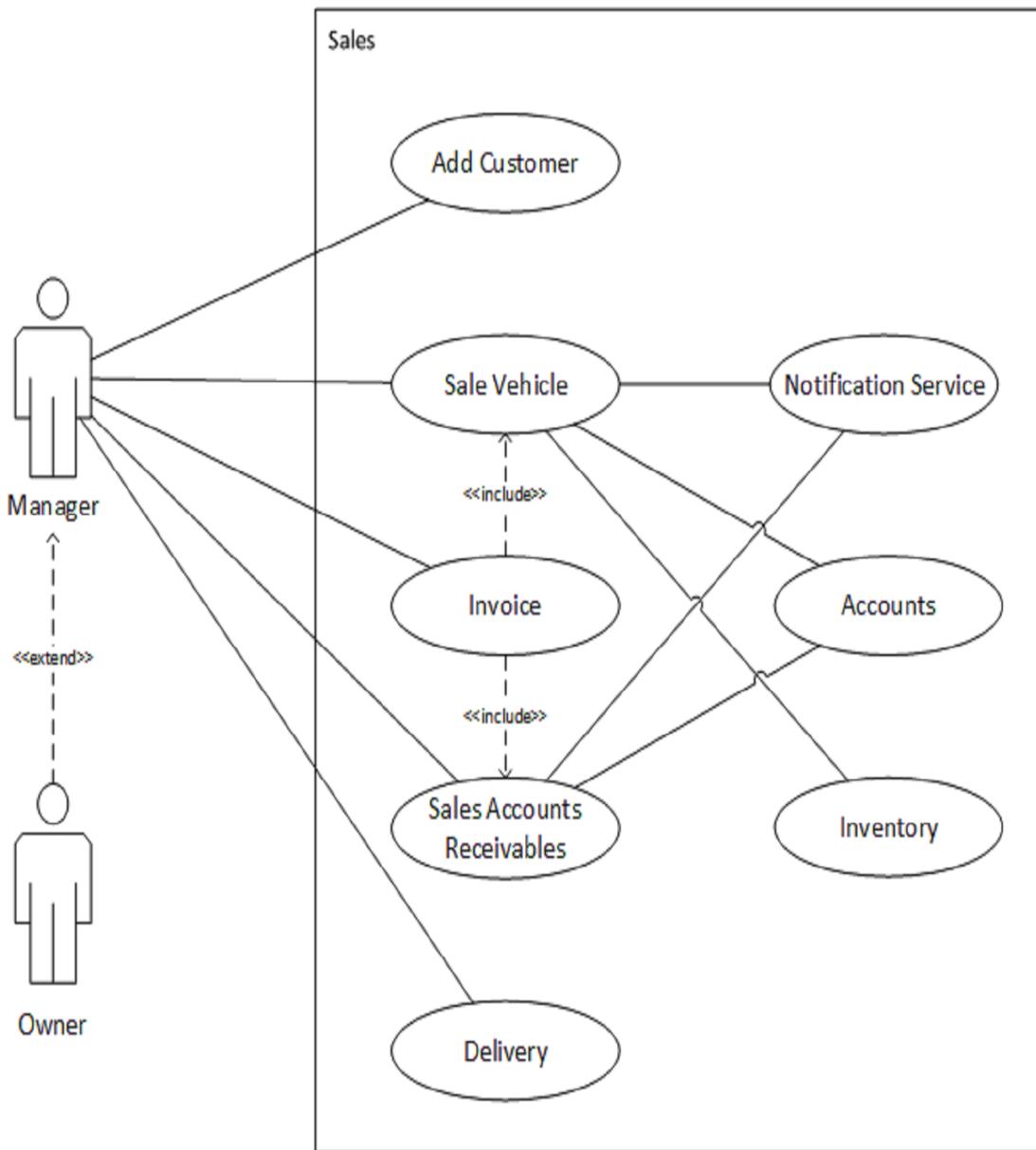


Figure 5.5: Use Case Diagram of Sales Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_SLS
Application	Car Trading Information Management System V4.0
Use Case Name	Sales Component
Use Case Description	Sales process of the shop will perform here as per business operational strategy of a car shop by storing customer information, sales and deliver. This component is also related with accounts and several accounts related action were done yet.
Primary Actor	Owner, Manager
Precondition	Authorized Personnel
Trigger	Sales Button Click
Basic Flow	<p>Firstly in this page, Add Customer tab appears automatically. Here user add required customer information and updates into to customer record. In the tabbed item Sale Vehicle, user simply sale a vehicle by accepting net payment and provide customer an invoice. In the tabbed item Accounts Receivables, user updates vehicle payment after receiving new net payment from the customer and provide a new invoice. All the invoices have transection history for maintaining transparency. When a vehicle has net due of the amount zero, an automatic message appears into user that the vehicle is ready to deliver. After using Deliver Car tabbed item, a document printed twice for legal document support because of the deal. Action performed in sales process will also notify to owner(s) via email.</p>
Alternate Flow	In case of any kind of failure, the component

This is the use case diagram of manage stuff component.

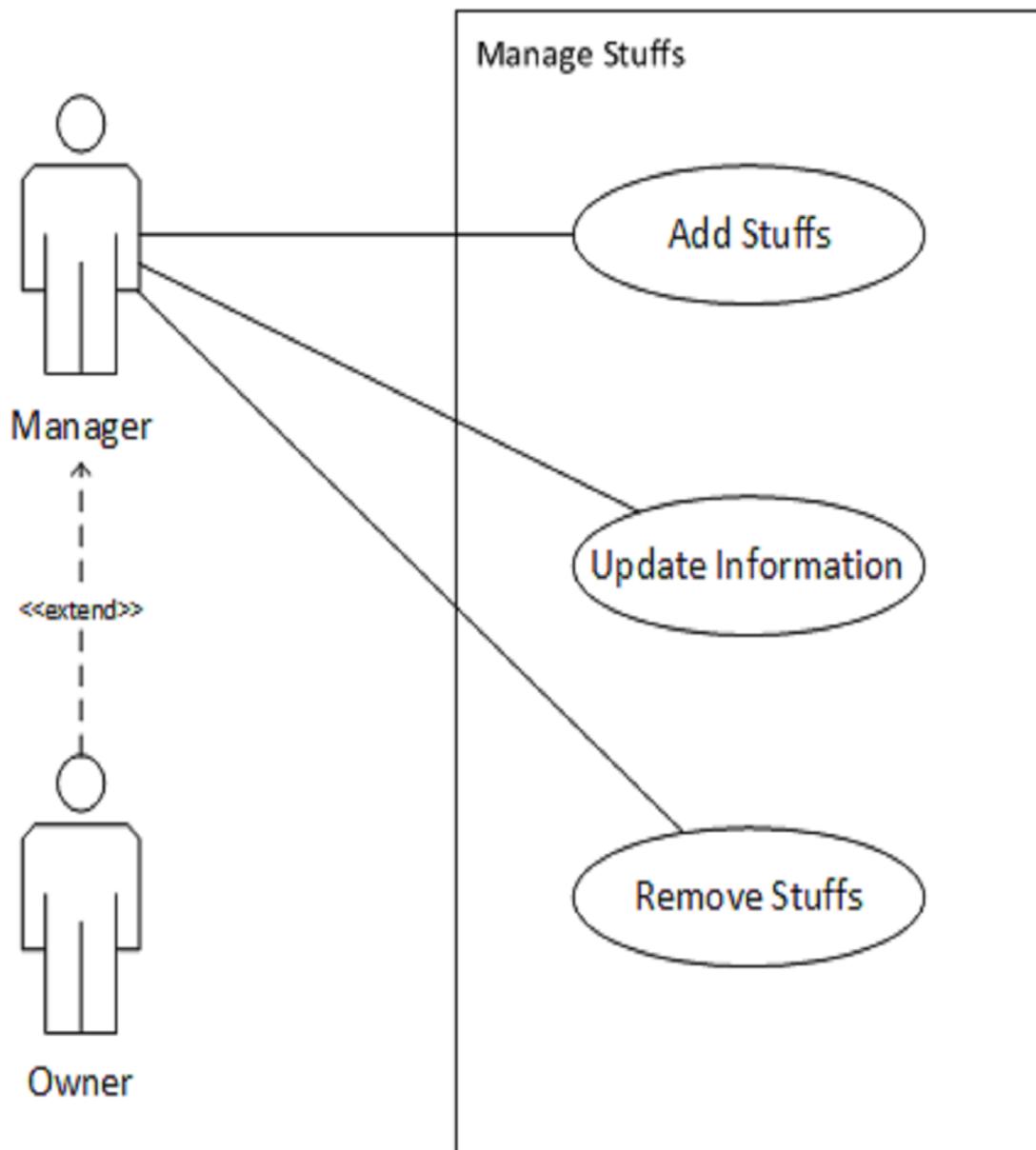


Figure 5.6: Use Case Diagram of Manage Stuff Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_STF
Application	Car Trading Information Management System V4.0
Use Case Name	Manage Stuff Component
Use Case Description	The main objective of the component is to manipulating stuff of employee information. Whereas car business is a smaller area business with huge cash flow, this component remains smaller but important to manage other components like accounts, because they are interrelated.
Primary Actor	Owner, Manager
Precondition	Authorized Personnel
Trigger	Manage Stuff Button Click
Basic Flow	Firstly in this page, Add Stuff tab appears automatically where user can insert new stuff information. User can edit stuff information like designation, salary etc. by using "Update Stuff Information" tabbed item. Note, not all attributes are editable in spite of making the system completely transparent. Finally, to remove a stuff, can be perform from "Remove Stuff" tabbed item.
Alternate Flow	In case of any kind of failure, the component breaks the communication from database with error notifications and return to the user so that threads cannot effect information stored into the database.

This is the use case diagram of expenditures component.

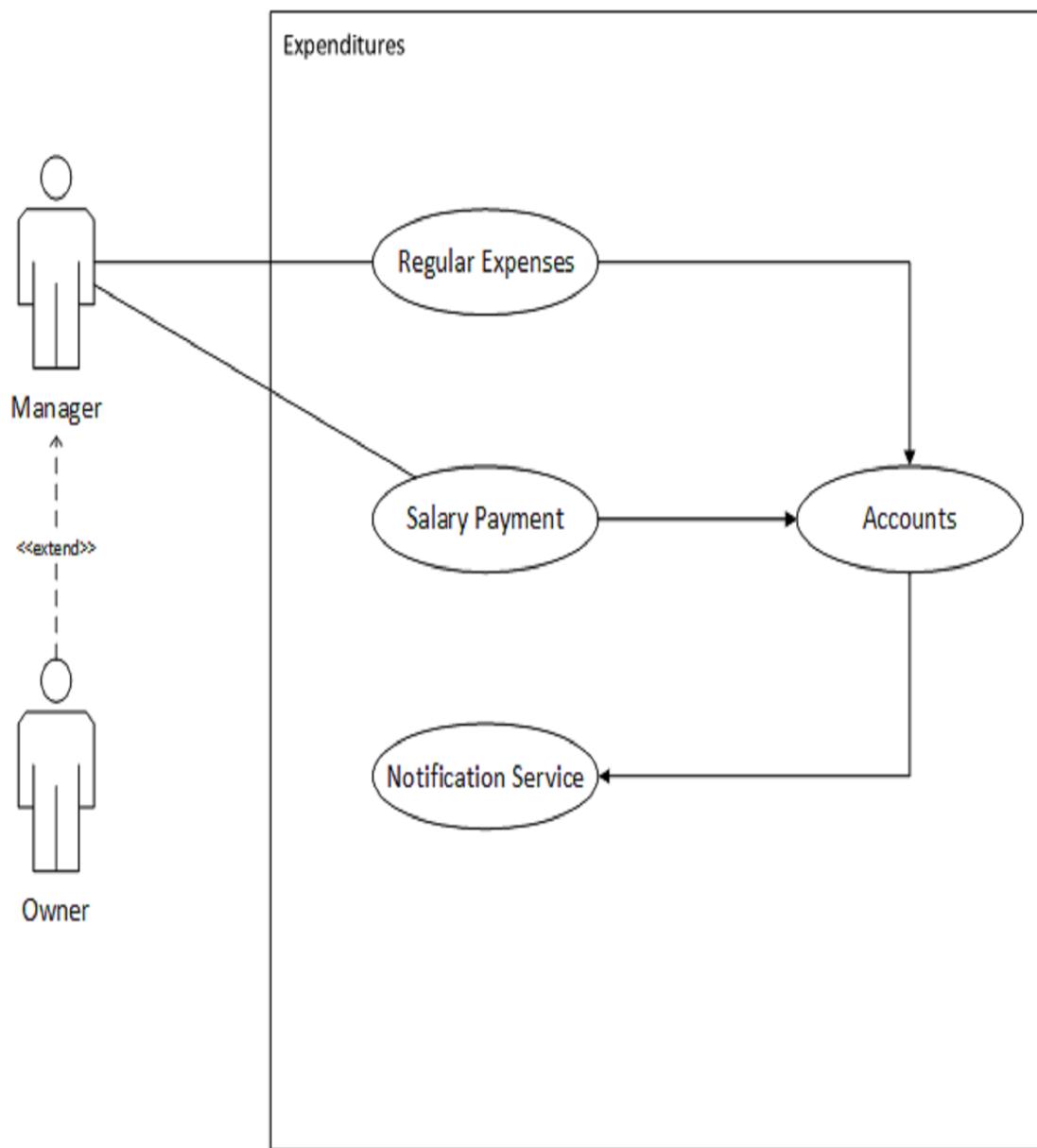


Figure 5.7: Use Case Diagram of Expenditures Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_EXP
Application	Car Trading Information Management System V4.0
Use Case Name	Expenditures Component
Use Case Description	This components allows user to record every expenditures' transactions. Whereas the working personnel are not quite good at managing traditional accounts system, so we derived this component that exactly work as a ledger balance of expenses.
Primary Actor	Owner, Manager
Precondition	Authorized Personnel
Trigger	Expenditures Button Click
Basic Flow	Firstly in this page, Regular Expenses tab appears automatically. Here user updates regular basis expense transections like accessories or fuel purchase of product maintenance. The other tab, Salary Payout enables user to update database after paying stuff salary. This is a monthly basis transection, which performs automatically triggering by pay button click. Changes in expense record will also notify to owner(s) via email.
Alternate Flow	In case of any kind of failure, the component breaks the communication from database with error notifications and return to the user so that threads cannot effect information stored into the database.

This is the use case diagram of decision support or dashboard component.

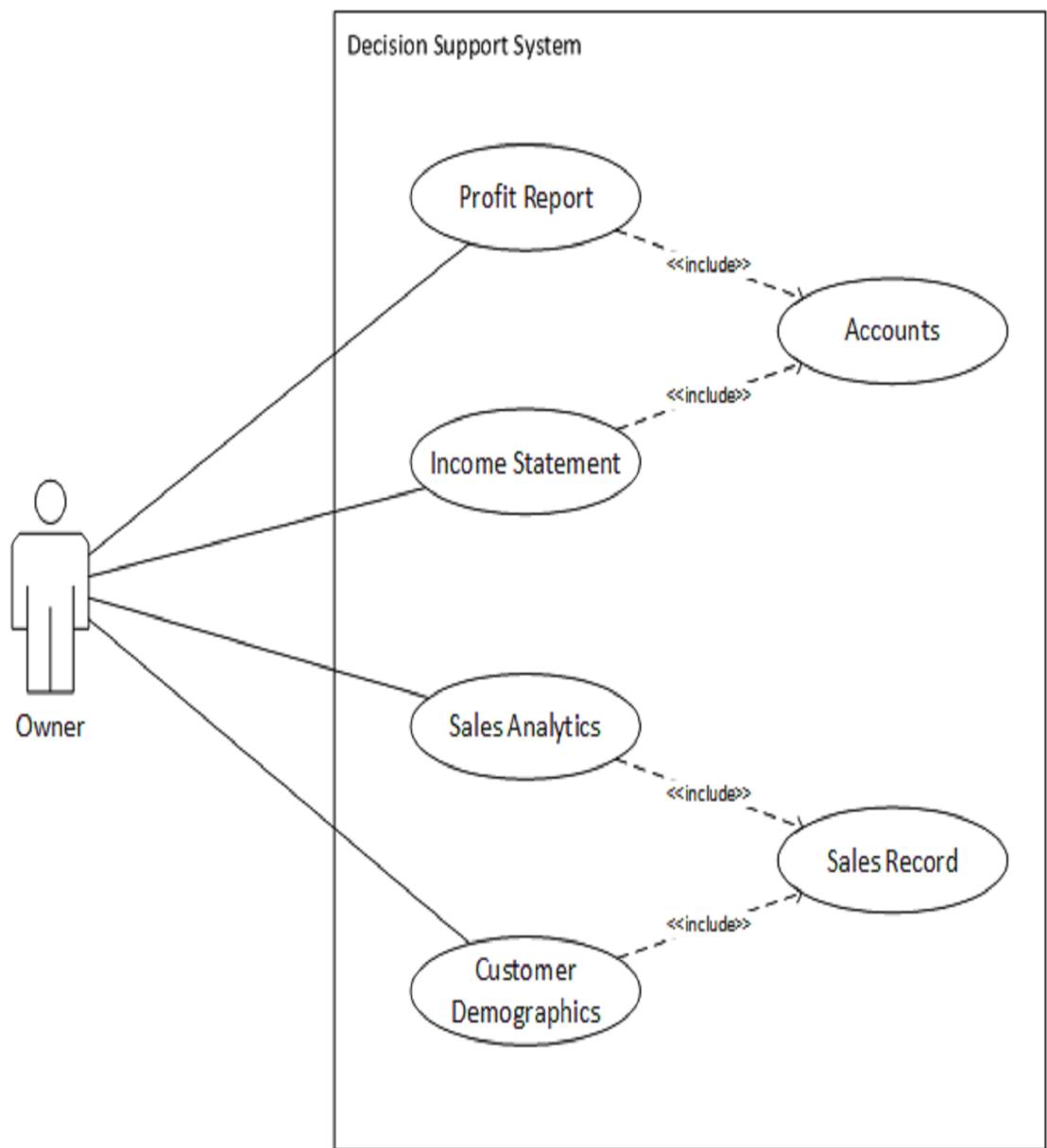


Figure 5.8: Use Case Diagram of Decision Support or Dashboard Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_DSS
Application	Car Trading Information Management System V4.0
Use Case Name	Decision Support or Dashboard Component
Use Case Description	<p>Based on usage, several analytics from business information will represent visually through various charts to help owner to take effective decisions for business growth. It is important for senior managers like owner(s) at this 21st century, to make effective business decision to improve business growth. These visual representation of previously performed actions helps owner(s) to make decision comparatively faster. This component is only for owner(s) use.</p>
Primary Actor	Owner
Precondition	Authorized Personnel and Owner
Trigger	Dashboard Button Click
Basic Flow	<p>In this page, information from sales record and expense record(revenue and expense) manipulated automatically from system and provide these analytics visually like profit growth from the business start-up. Based on sales record, most selling vehicle types and customer demographics will visualize by pie charts. Reports from accounts like "Income Statement" etc. can be retrieve from here.</p>
Alternate Flow	<p>In case of any kind of failure, the component breaks the communication from database with error notifications and return to the user so that threads cannot effect information stored into the database.</p>

This is the use case diagram of owner area component.

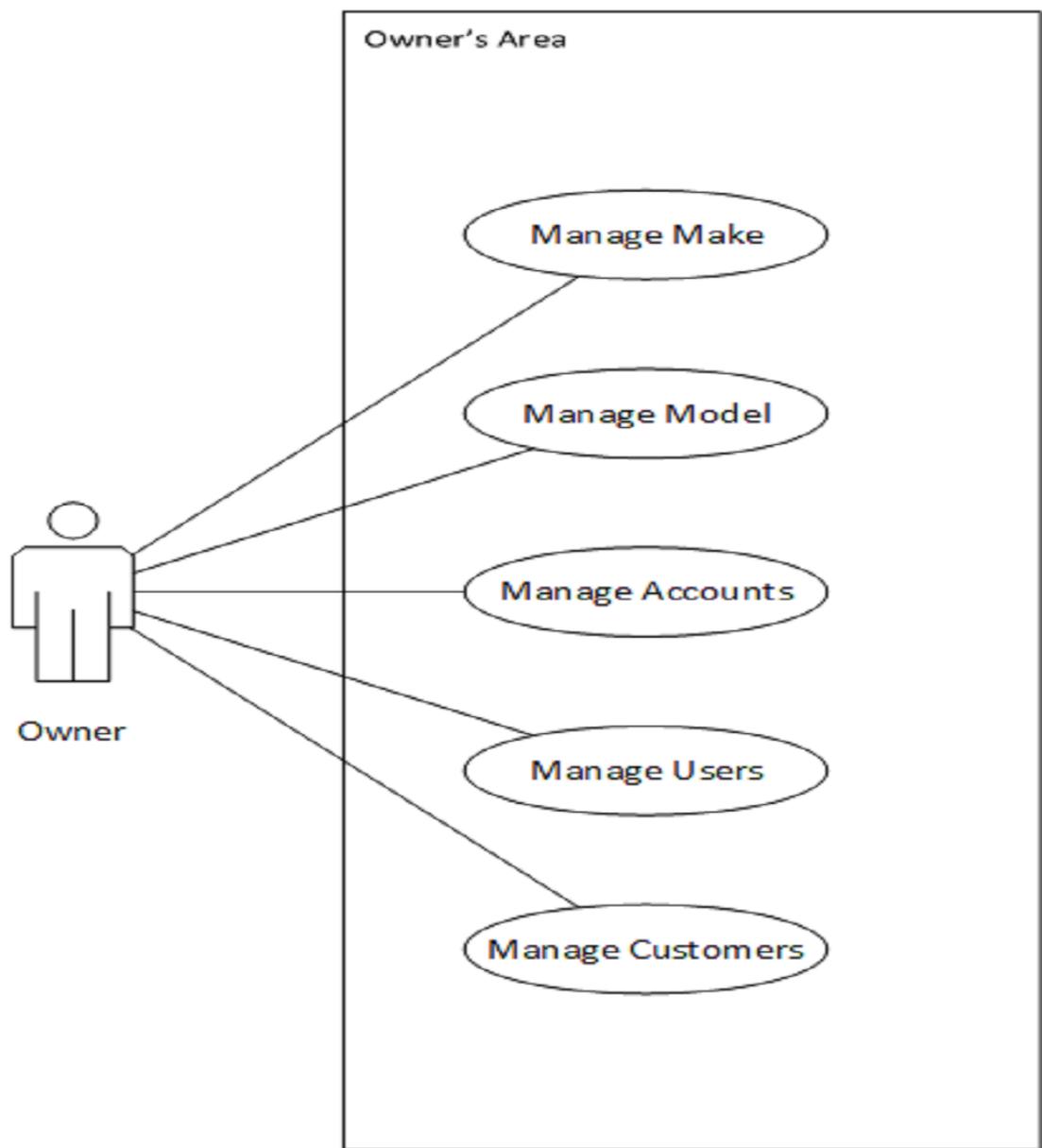


Figure 5.9: Use Case Diagram of Owner Area Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_OA
Application	Car Trading Information Management System V4.0
Use Case Name	Owner Area Component
Use Case Description	All the records can be manipulate through this component except some sensitive attributes in spite of security issue. This component was derived because according to working personnel, mistakes can be happen. So, from this, owner(s) can maintain the system perfectly. This component is only for owner(s) use.
Primary Actor	Owner
Precondition	Authorized Personnel and Owner
Trigger	Owner Area Button Click
Basic Flow	Firstly in this page, Add Make tab appears automatically where user can insert new Make. Similarly, user can insert new Model from Add Model tab. From the Accounts tab, user can trigger automated accounts system to perform accounting from revenue and expense and automatically update the database. There are also some data grids where user is able to make changes of almost every records. These actions are defined here in spite of avoid internal conspiracy.
Alternate Flow	In case of any kind of failure, the component breaks the communication from database with error notifications and return to the user so that threads cannot effect information stored into the database. 81

This is the use case diagram of support component.

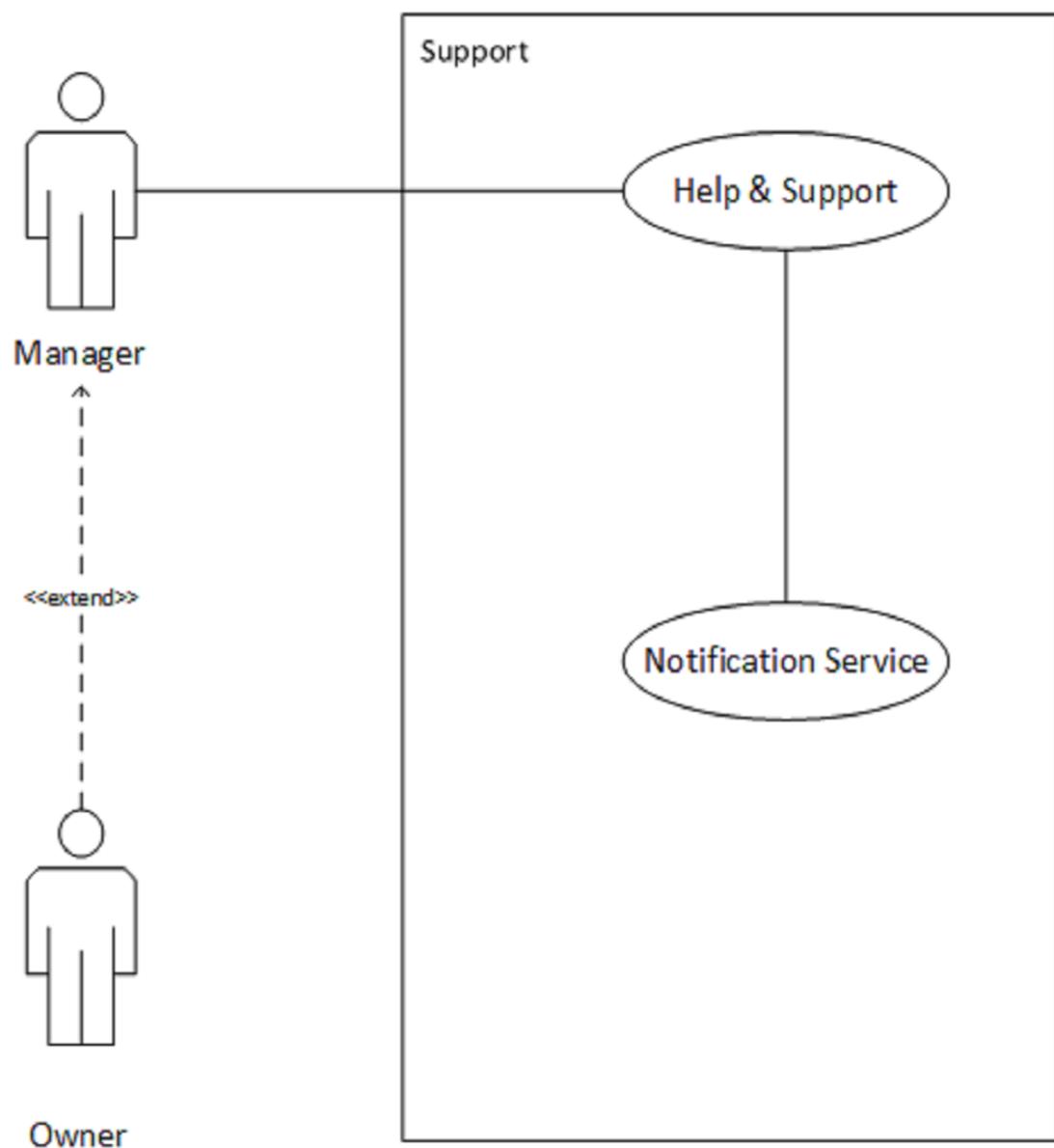


Figure 5.10: Use Case Diagram of Support Component

Use case Elements	Description
Use Case ID	CTIMS_V4_DSK_COMP_SPRT
Application	Car Trading Information Management System V4.0
Use Case Name	Support Component
Use Case Description	This component is derived for ensuring smart communication between car trader and system developer. From the component, trader can notify system developer instantly.
Primary Actor	Owner, Manager
Precondition	Authorized Personnel
Trigger	Support Button Click
Basic Flow	After entering support page, user gets a form where he fills it with a specific issue appeared during system use. After hitting "Send" button, the notification system develop the mail with formalities and send it through owner(s) email to developer's email. Then system developer should take further action against specified issue.
Alternate Flow	In case of any kind of failure of sending notification via email, the process will stop and the attempted message will lost. In this case, system user have to re-write the issue message again and send it.

This is the use case diagram of web site and any visitors can access shop's public information from here.

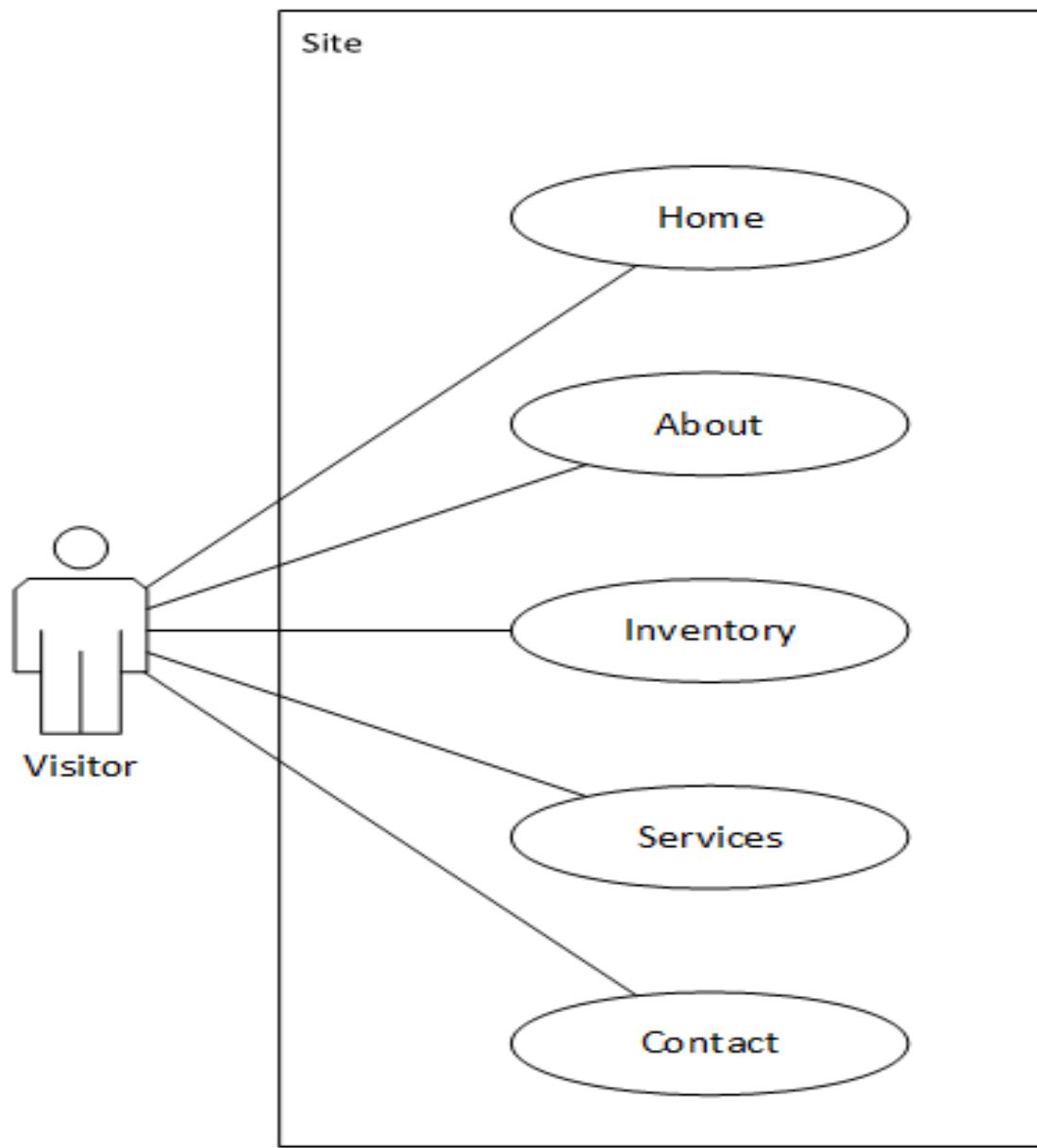


Figure 5.11: Use Case Diagram of Web Site

Use case Elements	Description
Use Case ID	CTIMS_V4_Web
Application	Car Trading Information Management System V4.0
Use Case Name	Web Site
Use Case Description	In spite of reducing the gap between trader and customer, we reached about the decision that in website all the public type information of shop will display. Visitors can browse service information, vehicle information from the website and contact to trader from the website.
Primary Actor	Anyone
Precondition	None
Trigger	carshop.com
Basic Flow	In this component, after hitting the url, the home page will appear and display overview of the car shop. In navigation menu, there be "About Us", "Services" etc are available which are similar static page like home page. In "Inventory" page, visitor can browse vehicle and get detailed information about it. In vehicle details page, there is also a loan calculator which is much supportive to maximum visitor. To contact with trader, simple user can navigate to "Contact" page and send inquiry message to authorized personnel of shop. Here visitor can also get google map support into "Contact" page.
Alternate Flow	There is no alternative flow in this website. If any kind of issue like server failure happens, ⁸⁵ it's assured that the connection with database will break and no business data will harm.

5.3 Activity Diagrams

In this project we designed all the activity diagrams by component wise in spite of making it more understandable, also known as UML 1.1. Let's discuss it below.

This is the activity diagram of authentication component.

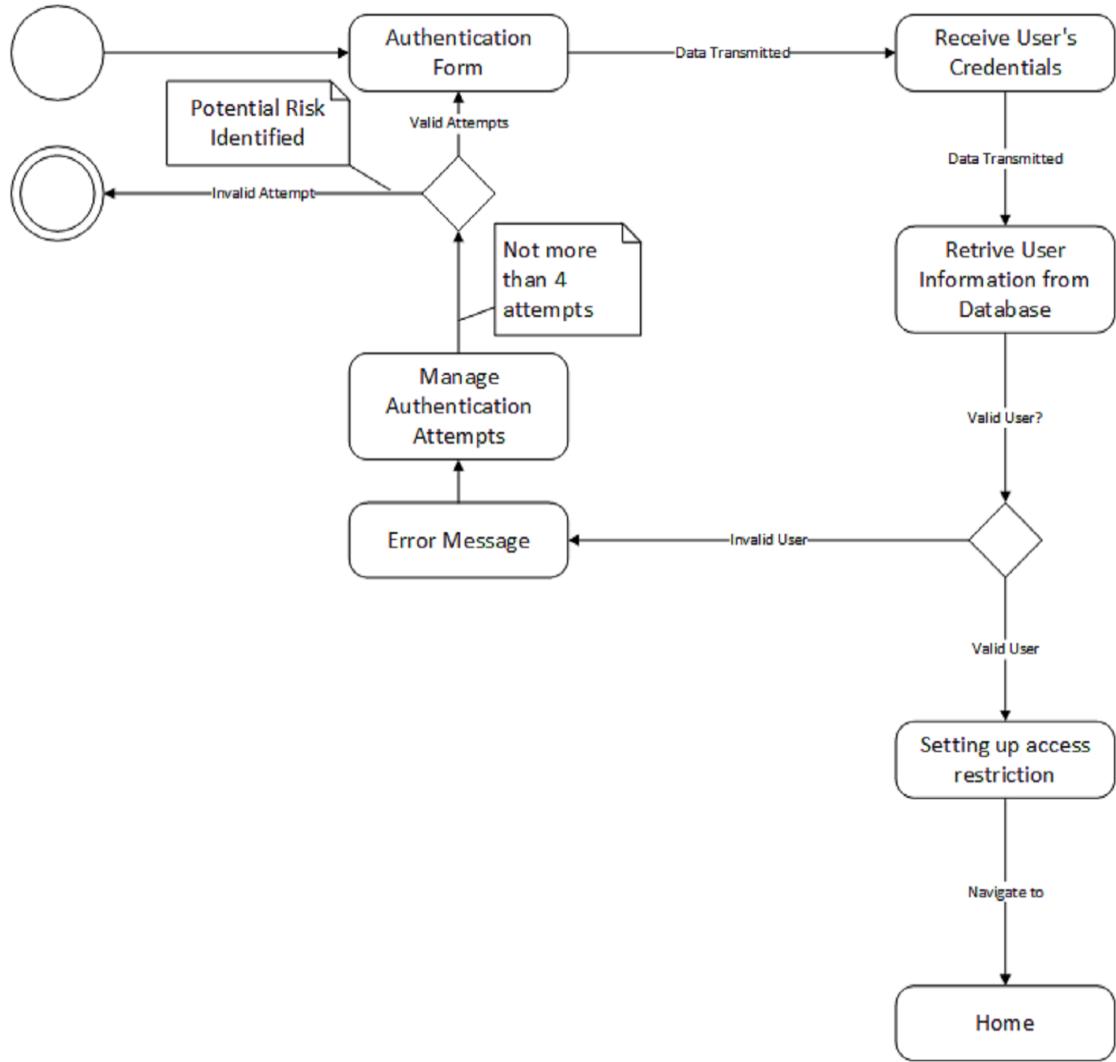


Figure 5.12: Use Activity Diagram of Authentication Component

This is the activity diagram of home navigation component.

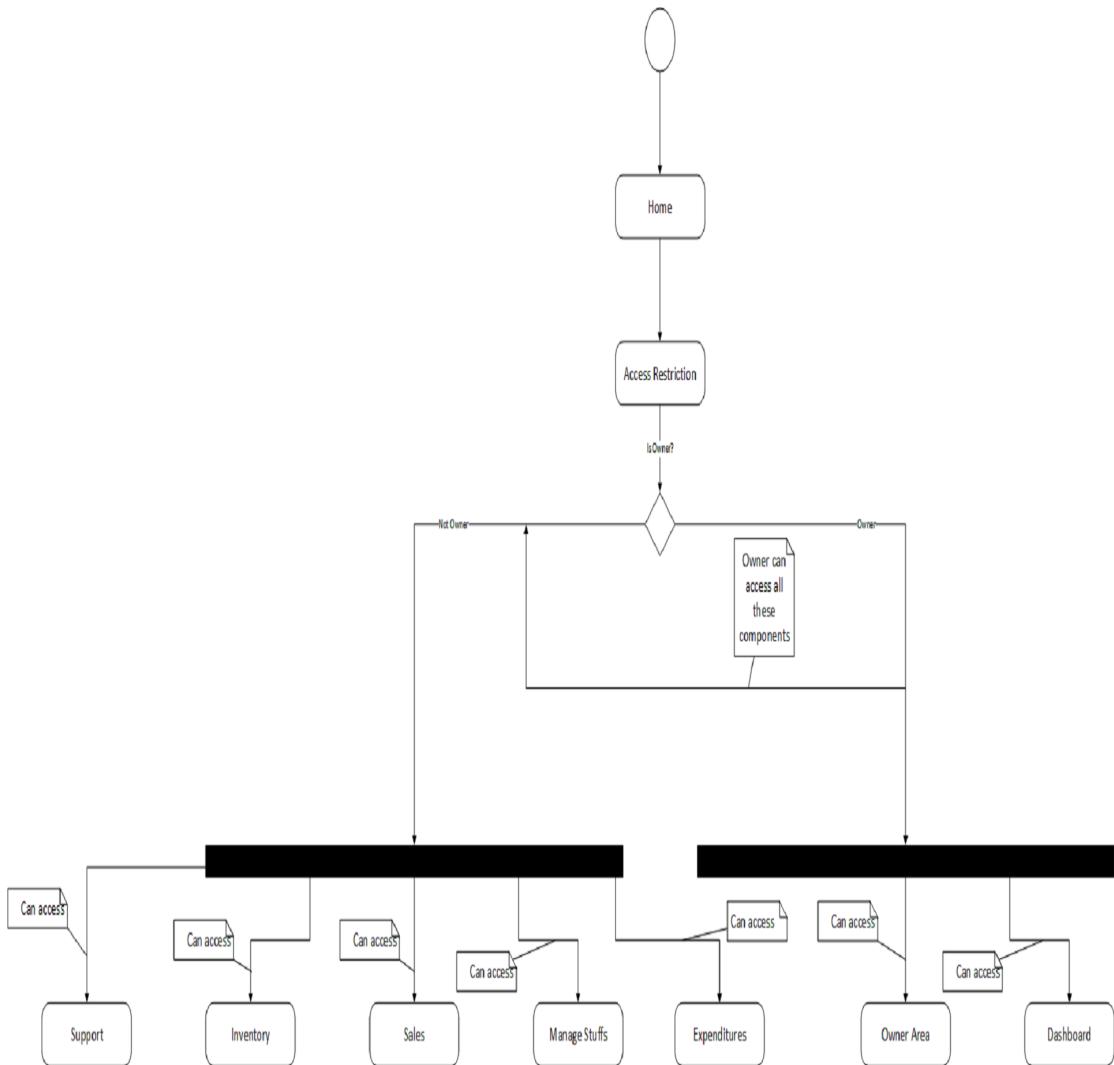


Figure 5.13: Use Activity Diagram of Home Navigation Component

This is the activity diagram of inventory component.

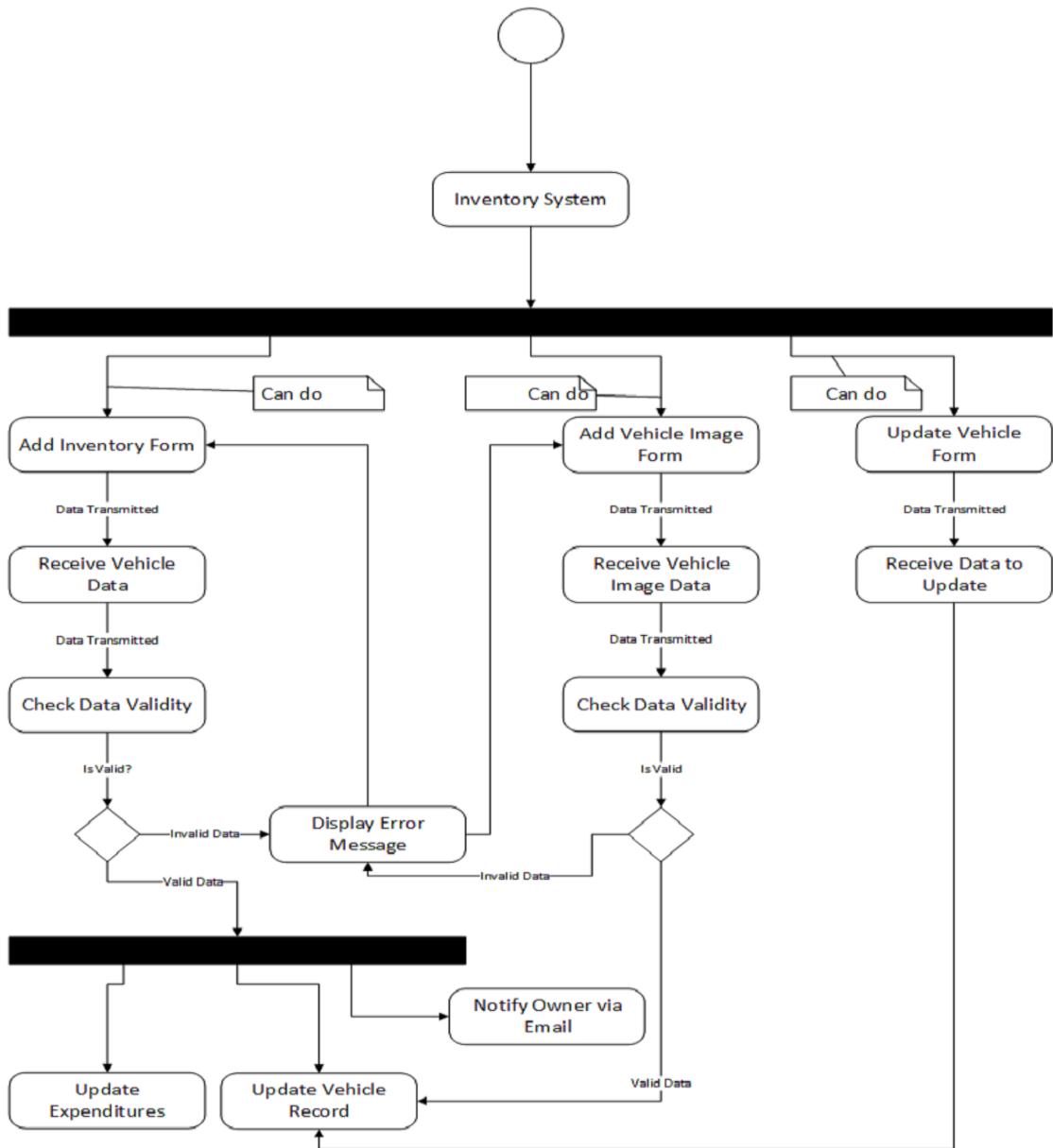


Figure 5.14: Use Activity Diagram of Inventory Component

This is the activity diagram of sales component.

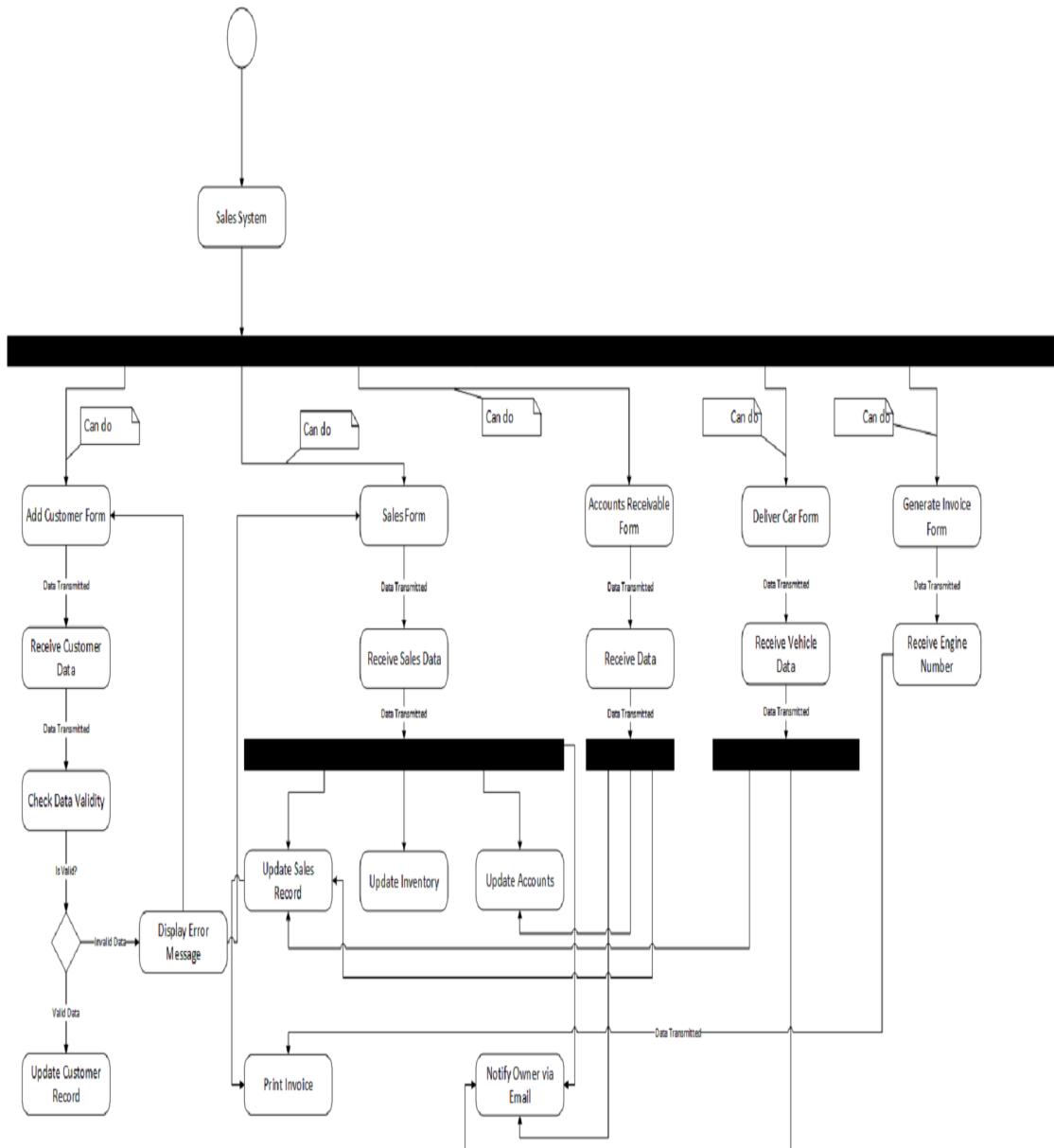


Figure 5.15: Use Activity Diagram of Sales Component

This is the activity diagram of manage stuff component.

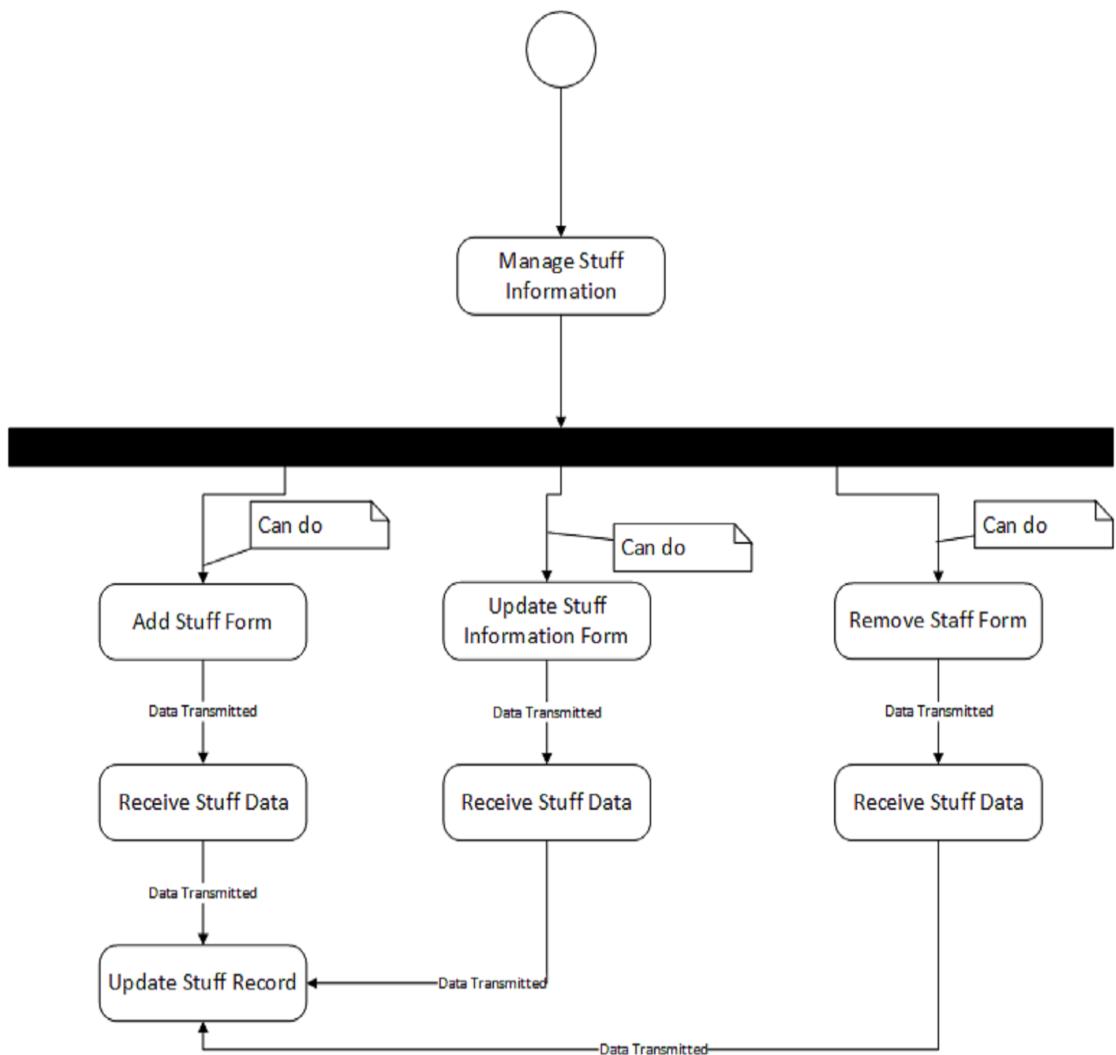


Figure 5.16: Use Activity Diagram of Manage Stuff Component

This is the activity diagram of expenditures component.

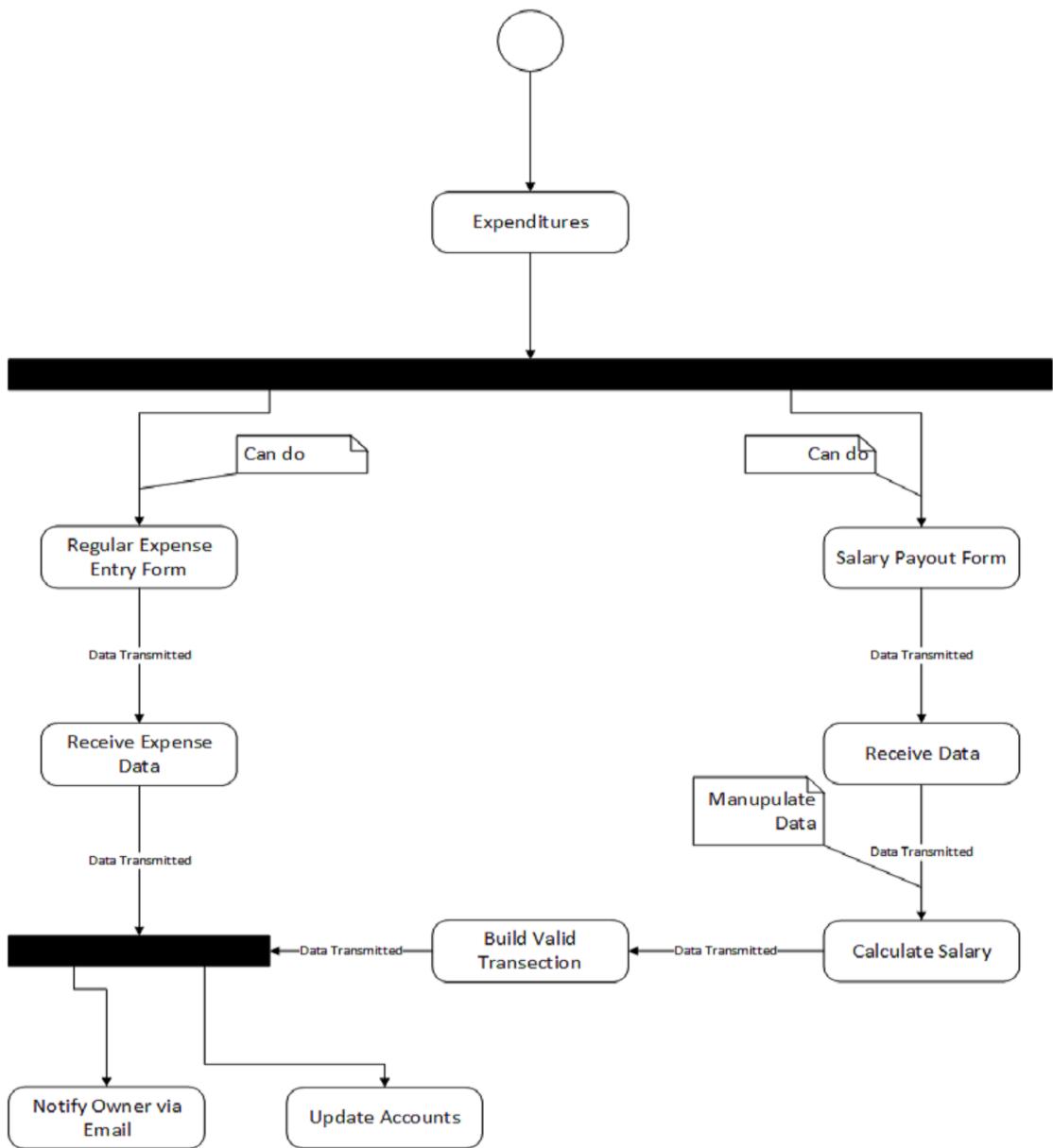


Figure 5.17: Use Activity Diagram of Expenditures Component

This is the activity diagram of owner area component.

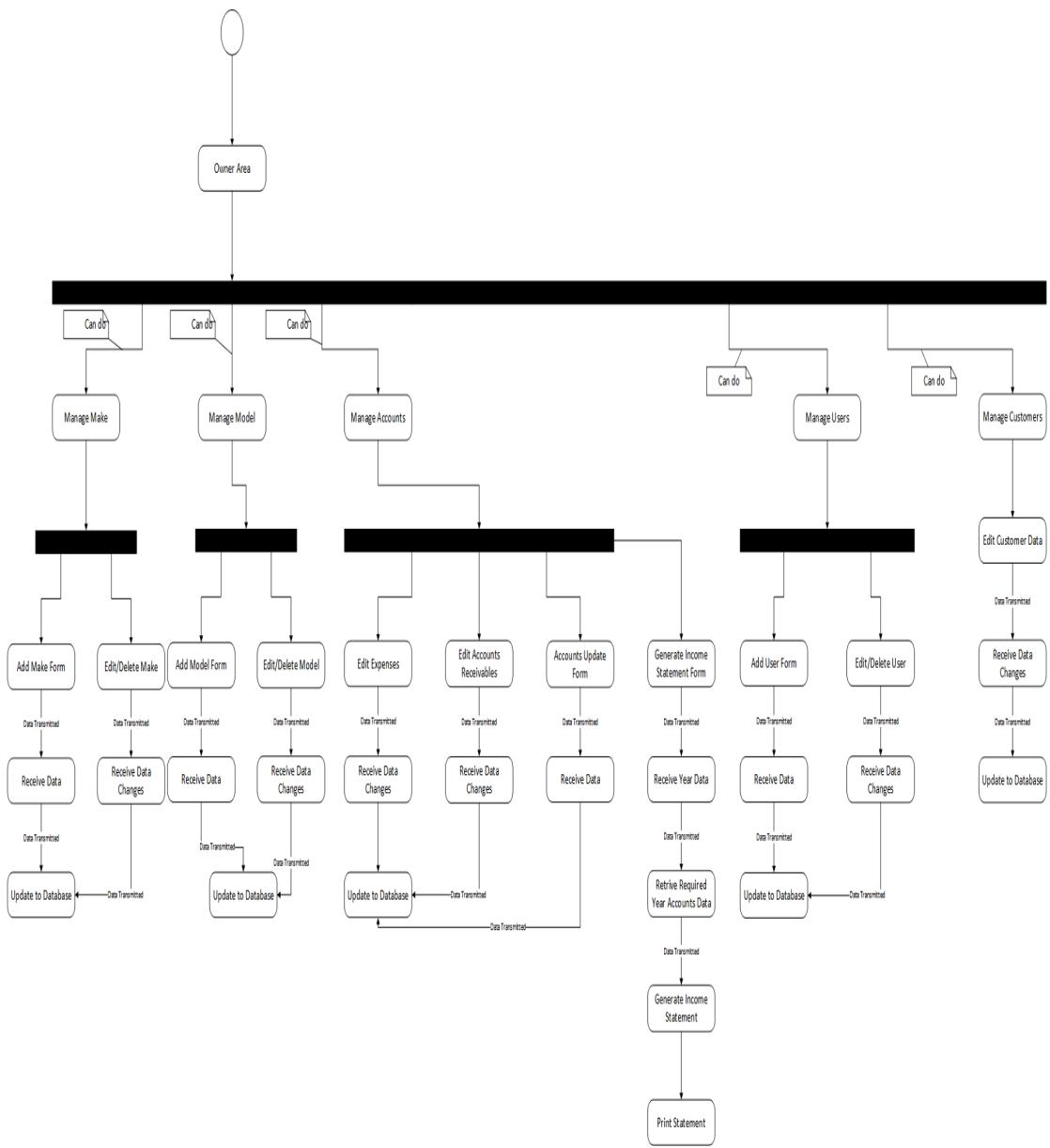


Figure 5.18: Use Activity Diagram of Owner Area Component

This is the activity diagram of decision support or dashboard component.

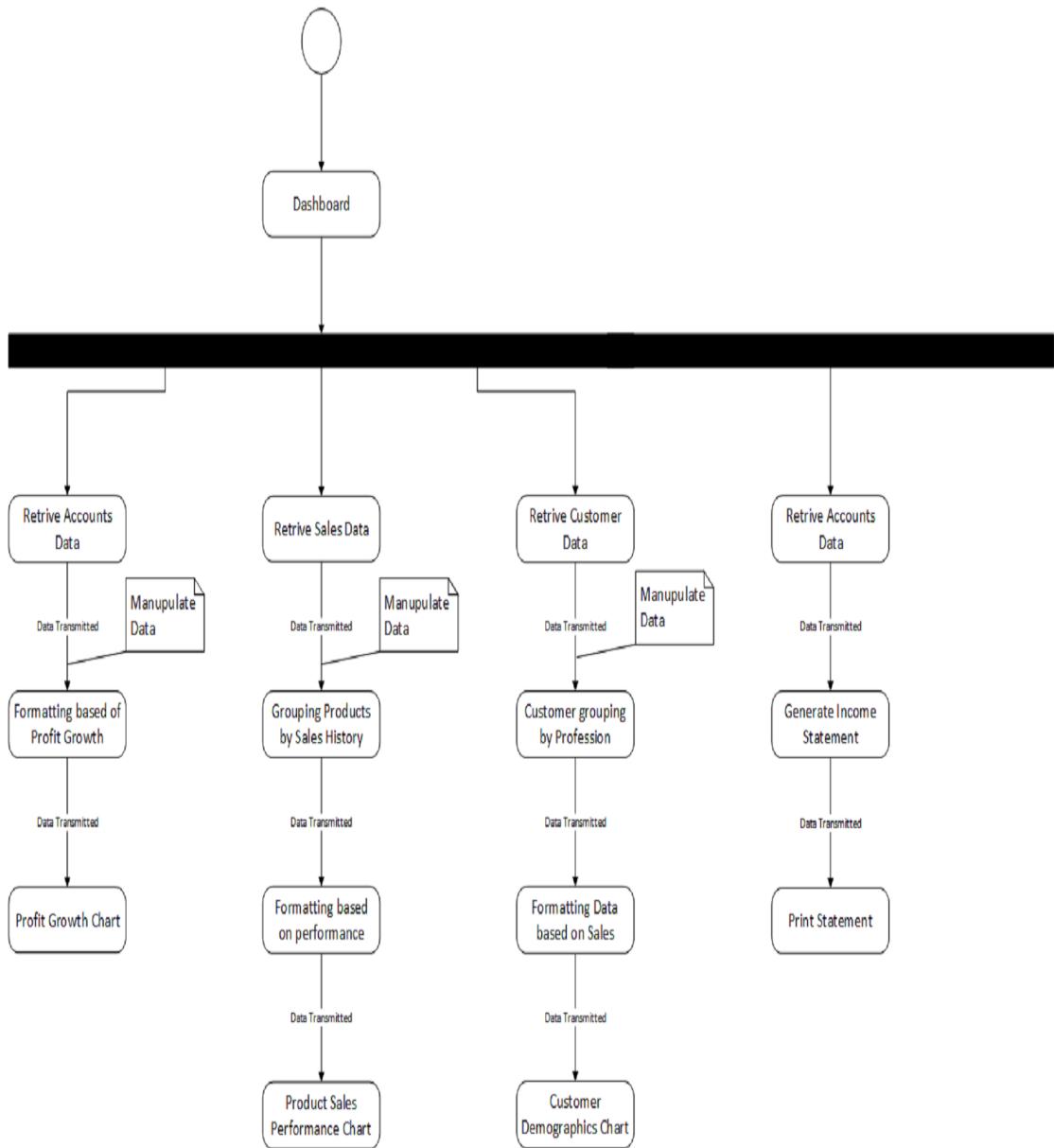


Figure 5.19: Use Activity Diagram of Decision Support Component

This is the activity diagram of support component.

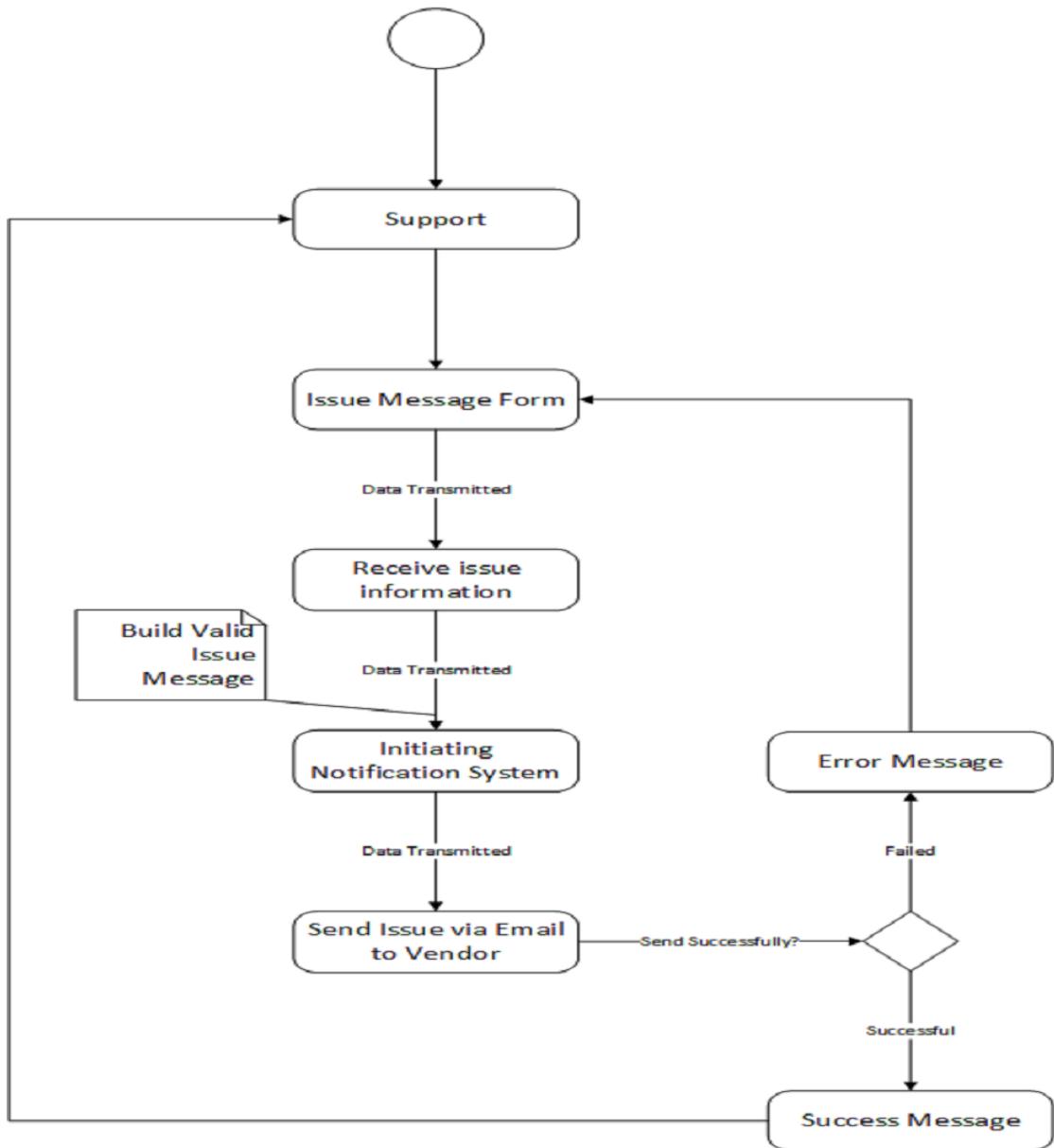


Figure 5.20: Use Activity Diagram of Support component

This is the activity diagram of web site.

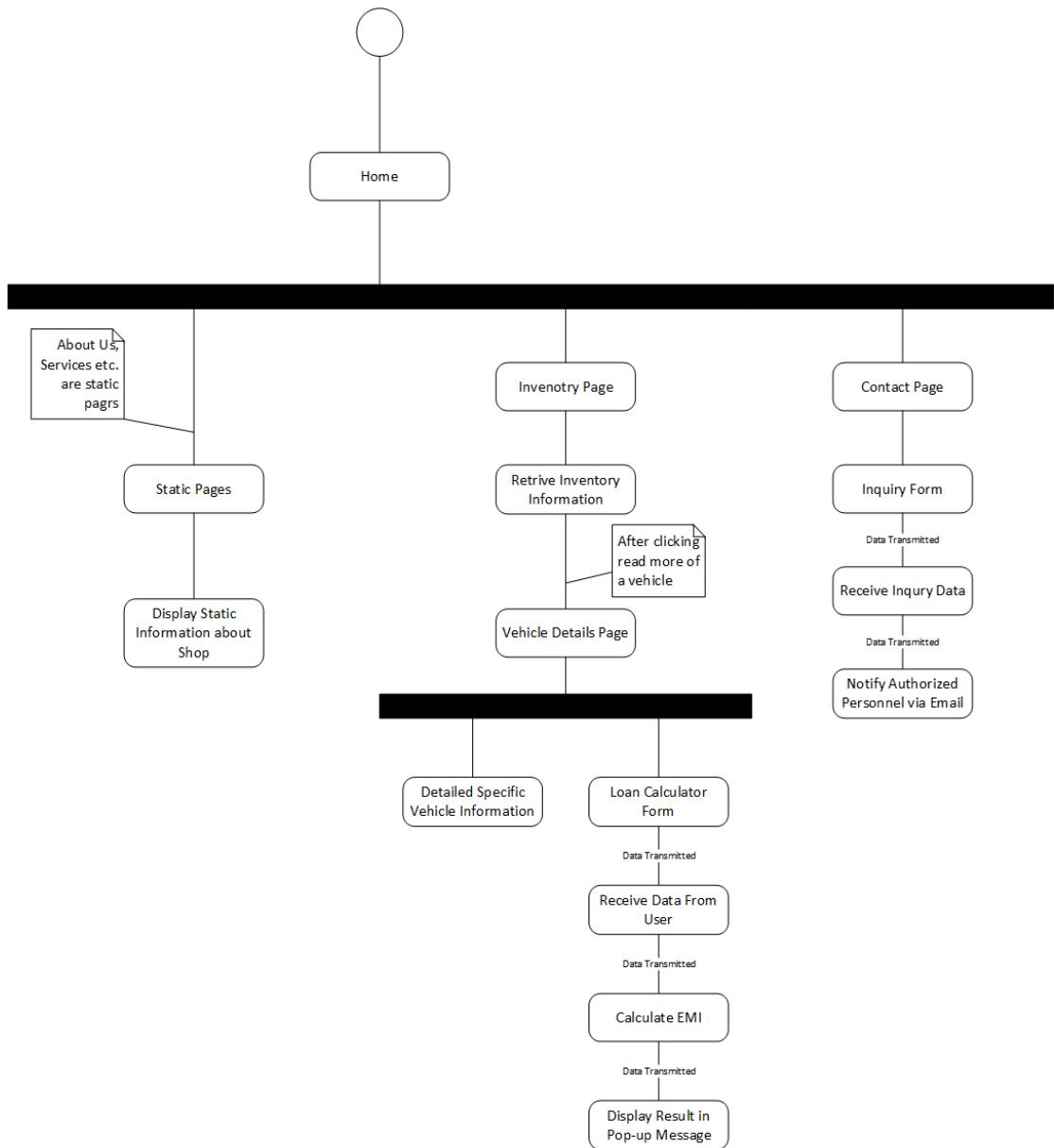


Figure 5.21: Use Activity Diagram of Web Site

5.4 Sequence Diagrams

In this project we designed all the activity diagrams by component wise in spite of making it more understandable, also known as UML 1.1. Let's discuss it below.

This is the sequence diagram of authentication component.

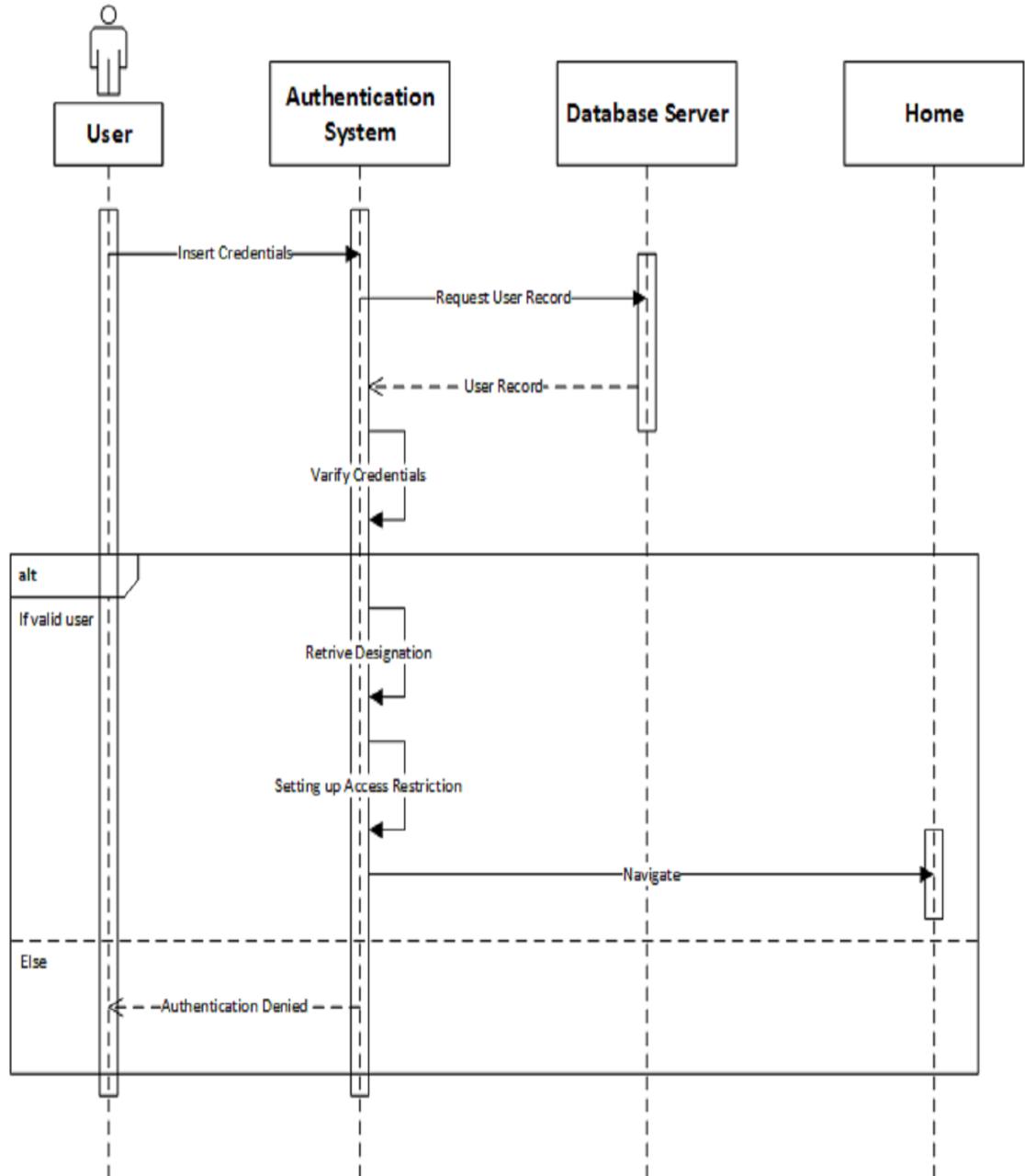


Figure 5.22: Sequence Diagram of Authentication Component

This is the sequence diagram of inventory component.

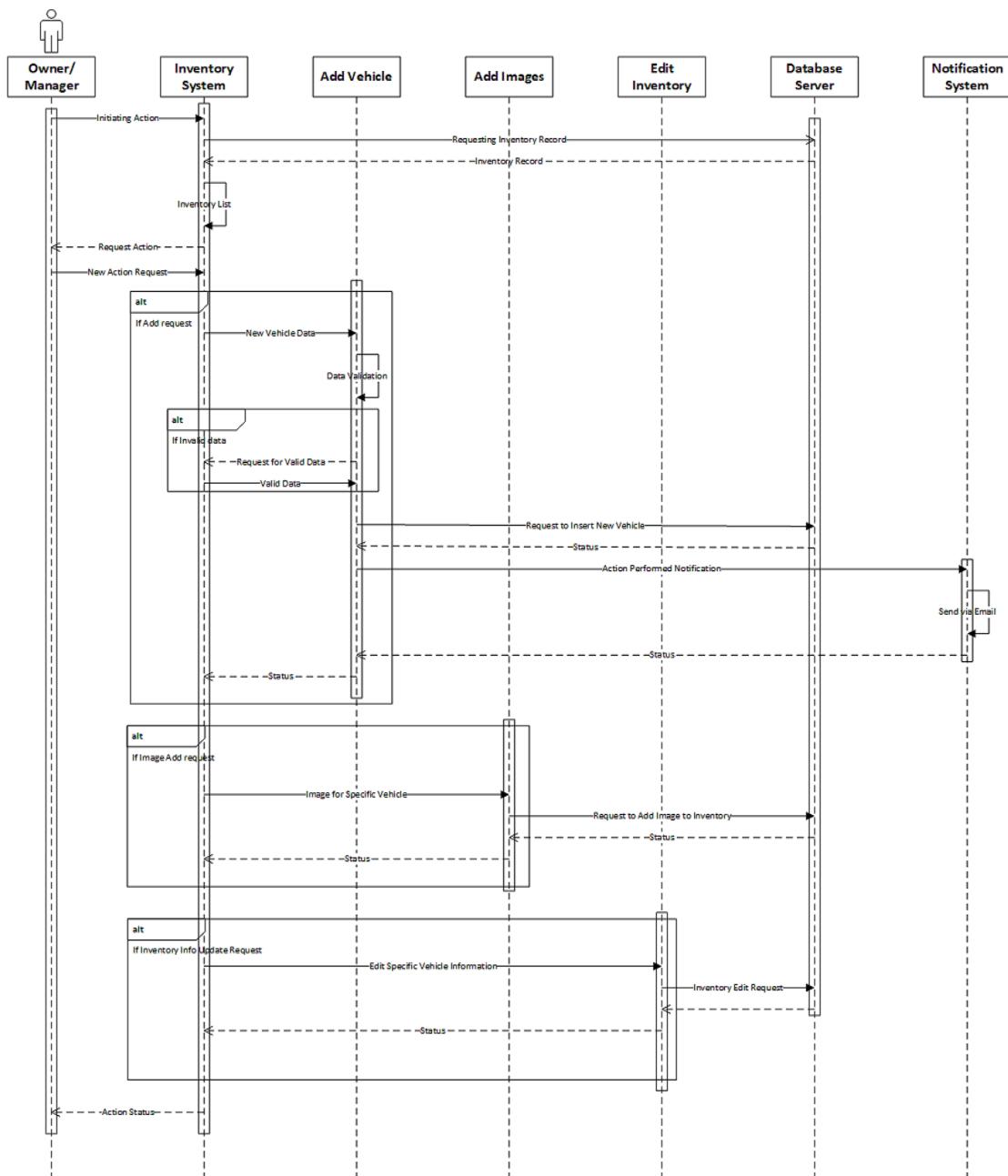


Figure 5.23: Sequence Diagram of Inventory Component

This is the sequence diagram of sales component.

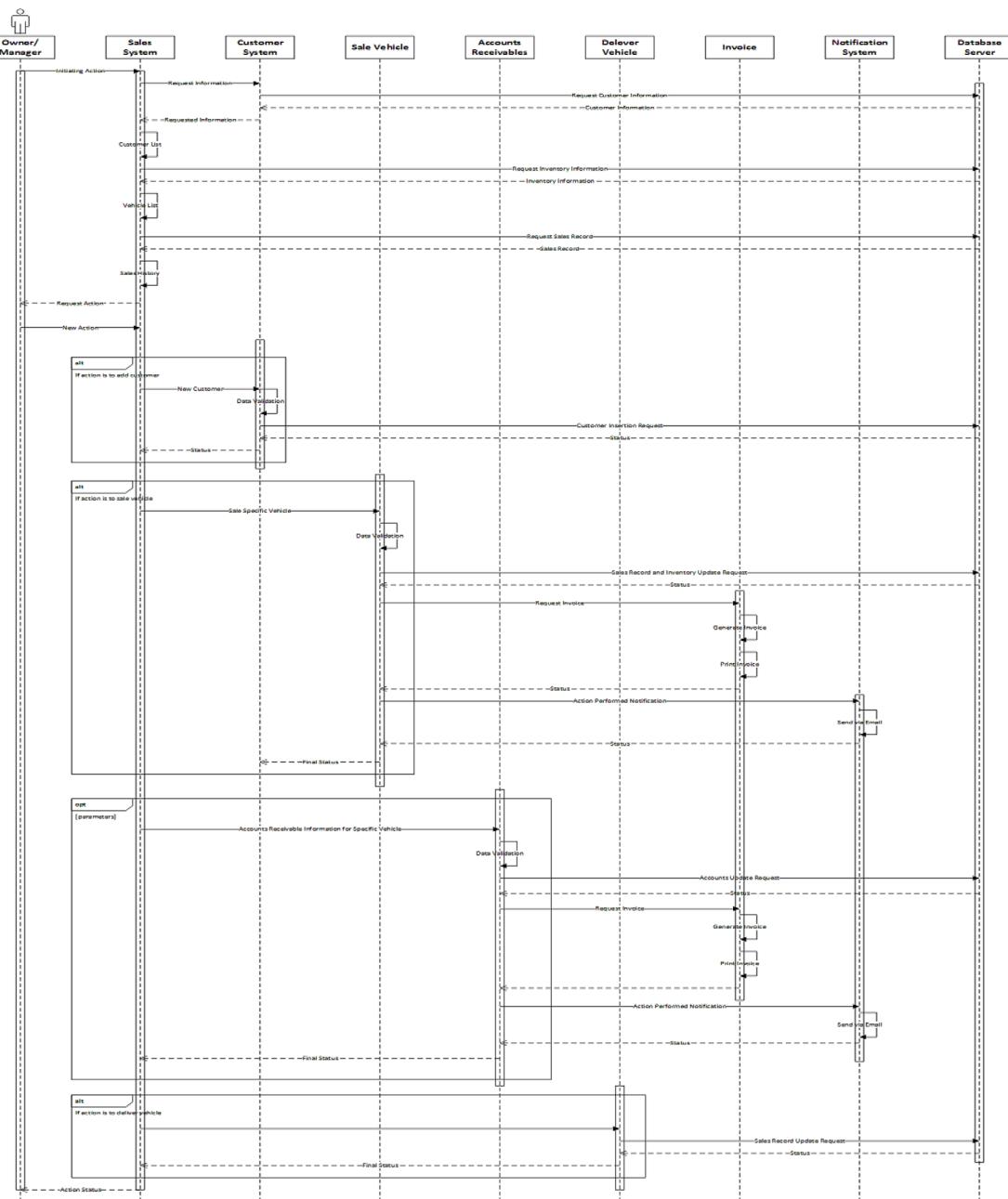


Figure 5.24: Sequence Diagram of Sales Component

This is the sequence diagram of manage stuffs component.

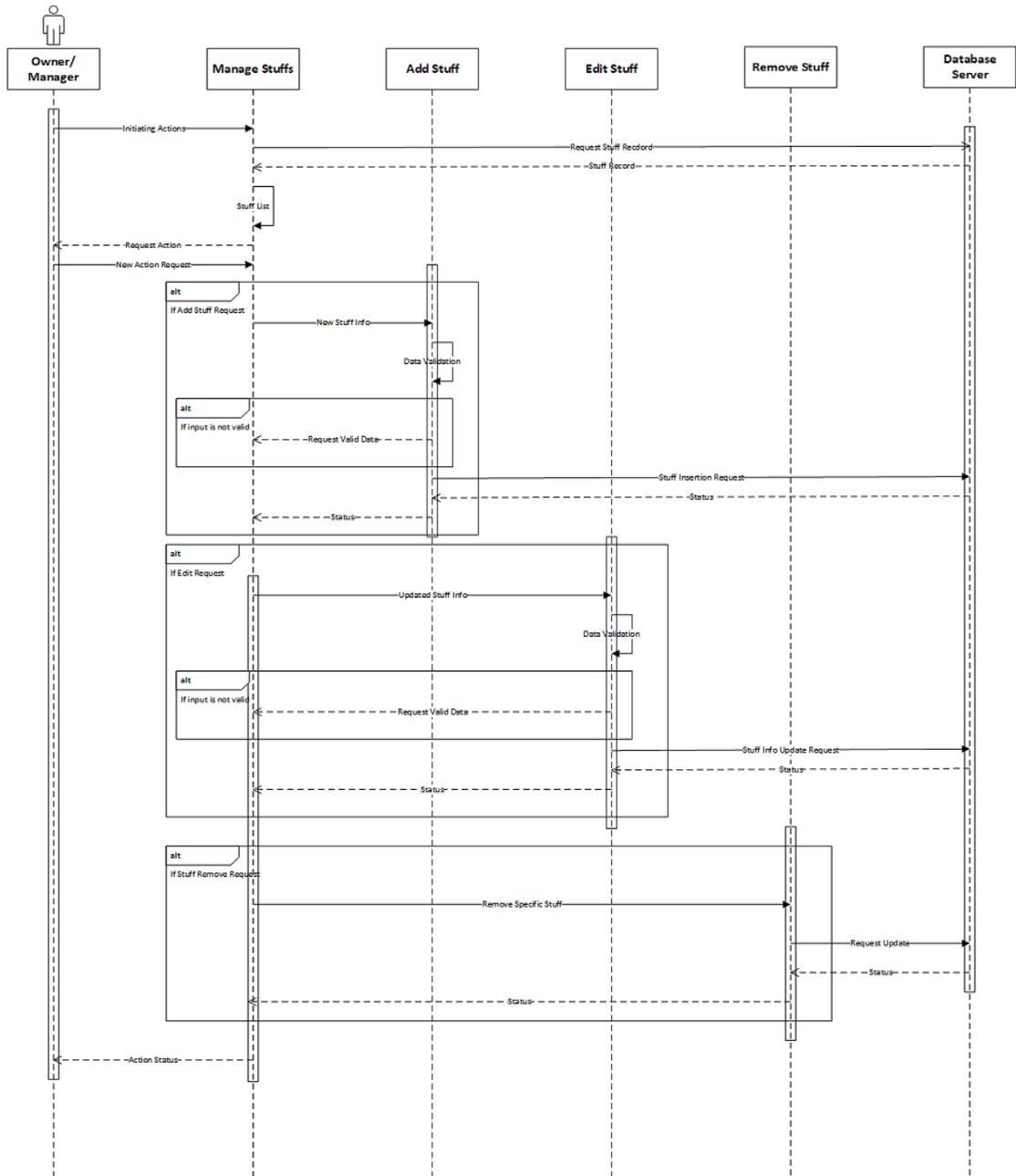


Figure 5.25: Sequence Diagram of Manage Stuffs Component

This is the sequence diagram of expenditures component.

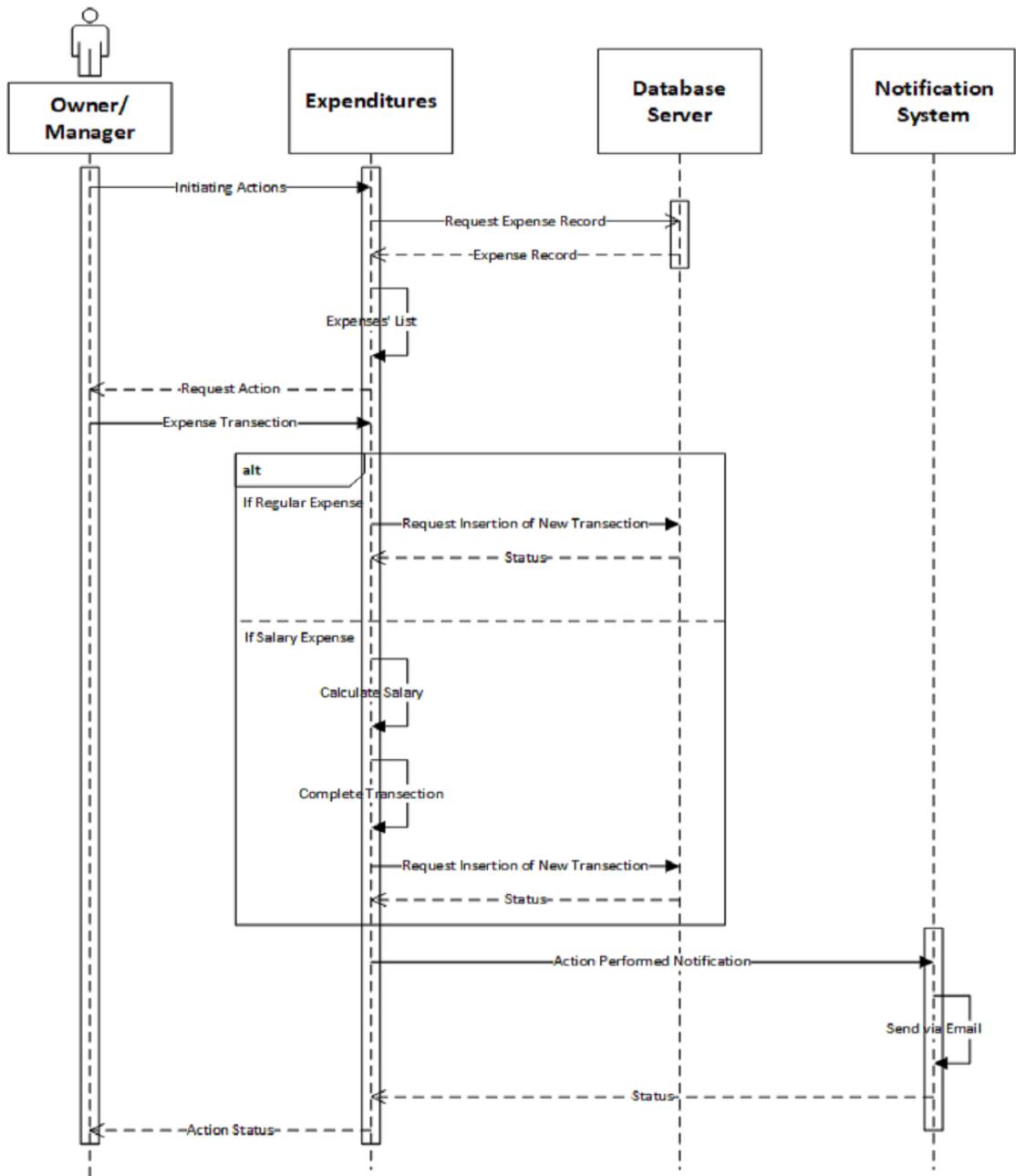


Figure 5.26: Sequence Diagram of Expenditures Component

This is the sequence diagram of owner area component.

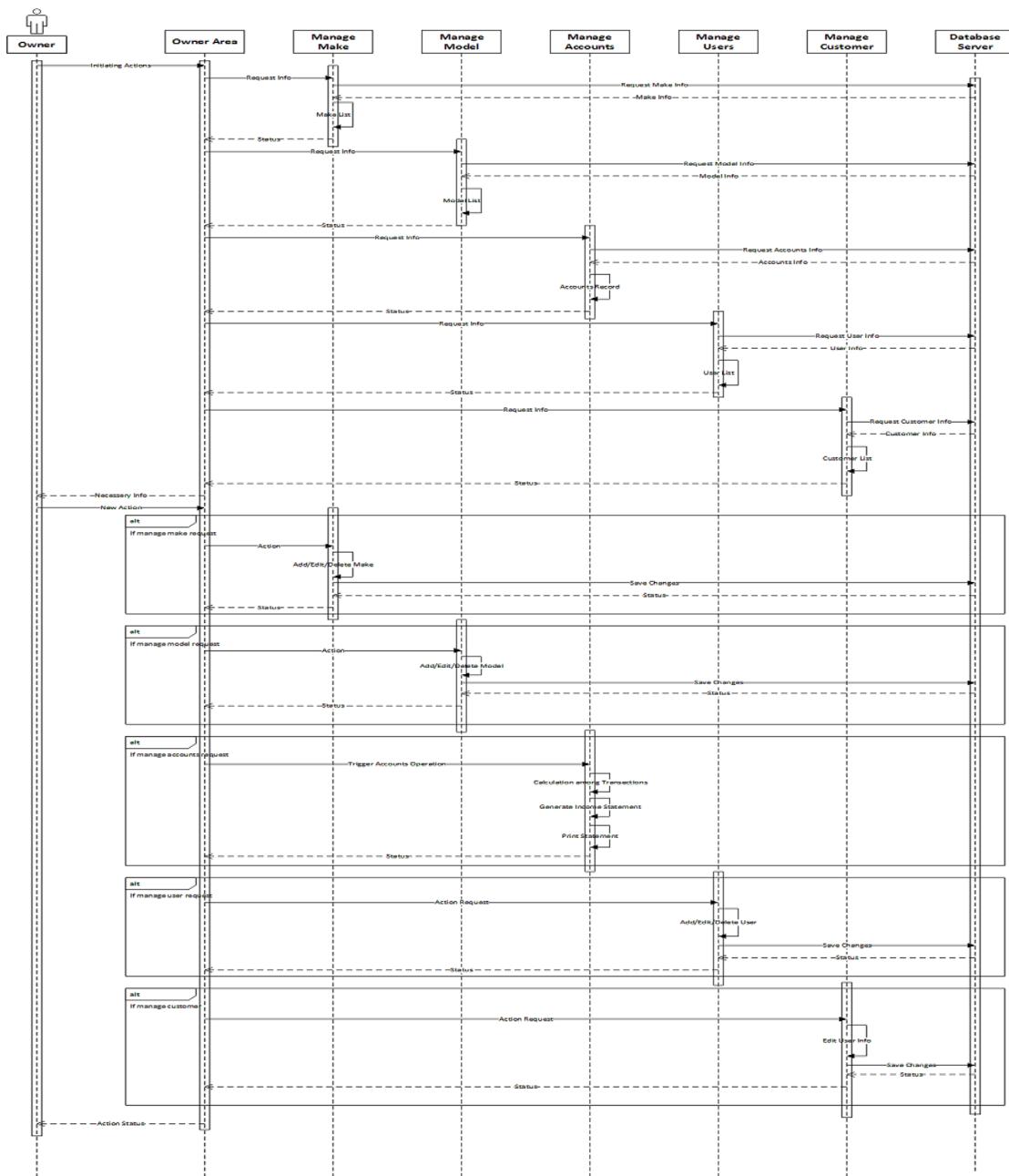


Figure 5.27: Sequence Diagram of Owner Area Component

This is the sequence diagram of decision support or dashboard component.

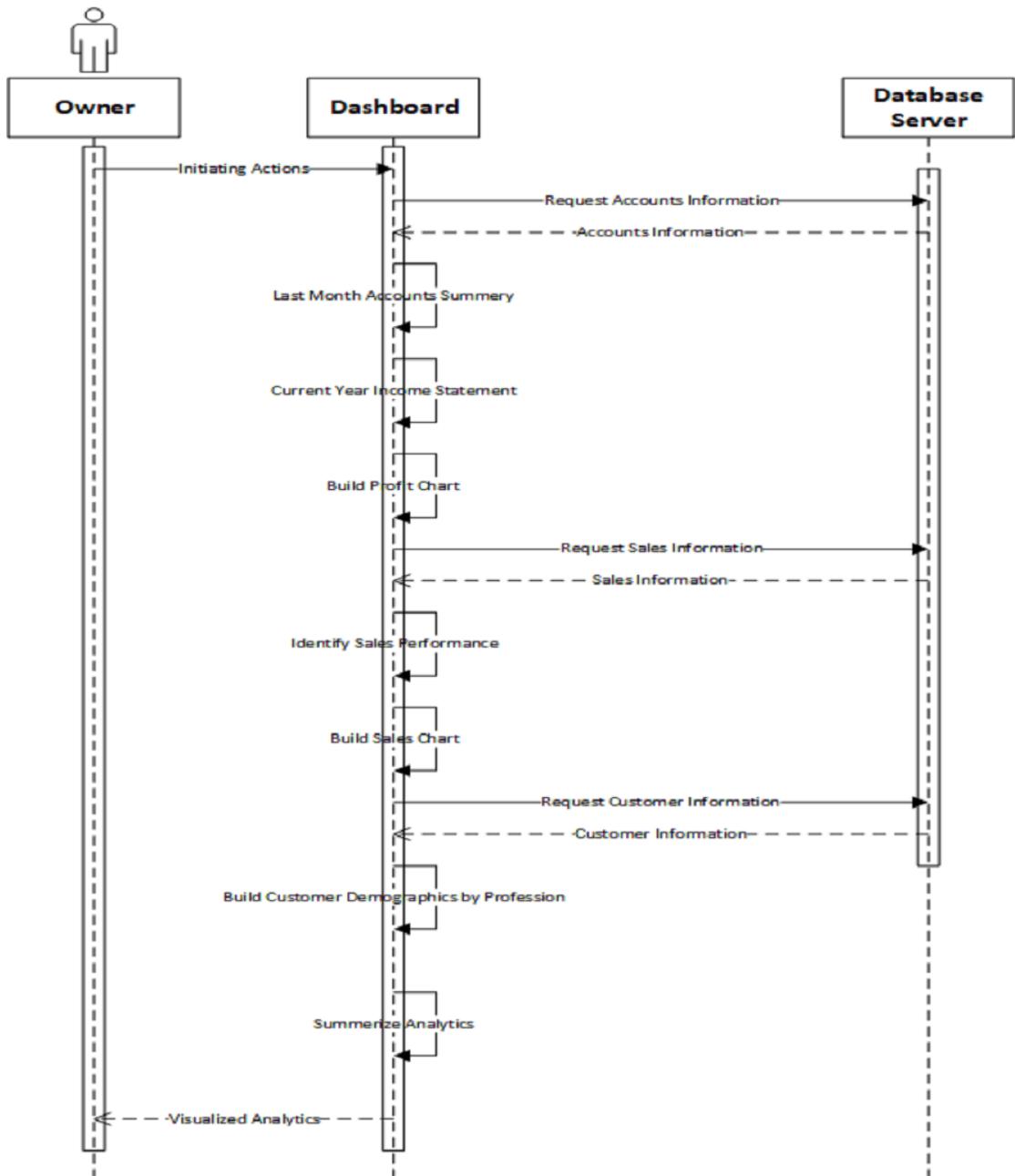


Figure 5.28: Sequence Diagram of Dashboard Component

This is the sequence diagram of support component.

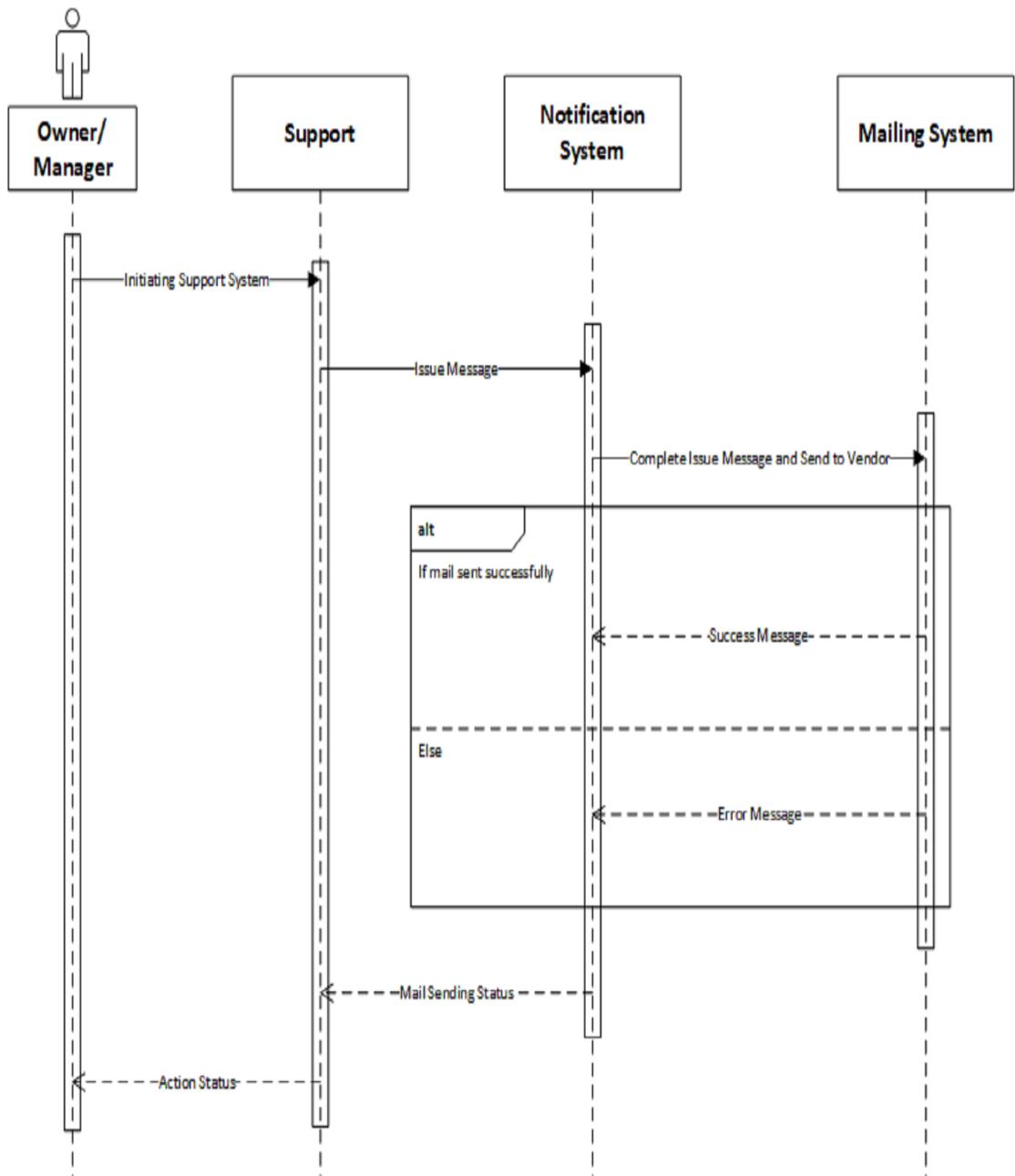


Figure 5.29: Sequence Diagram of Support Component

This is the sequence diagram of web site.

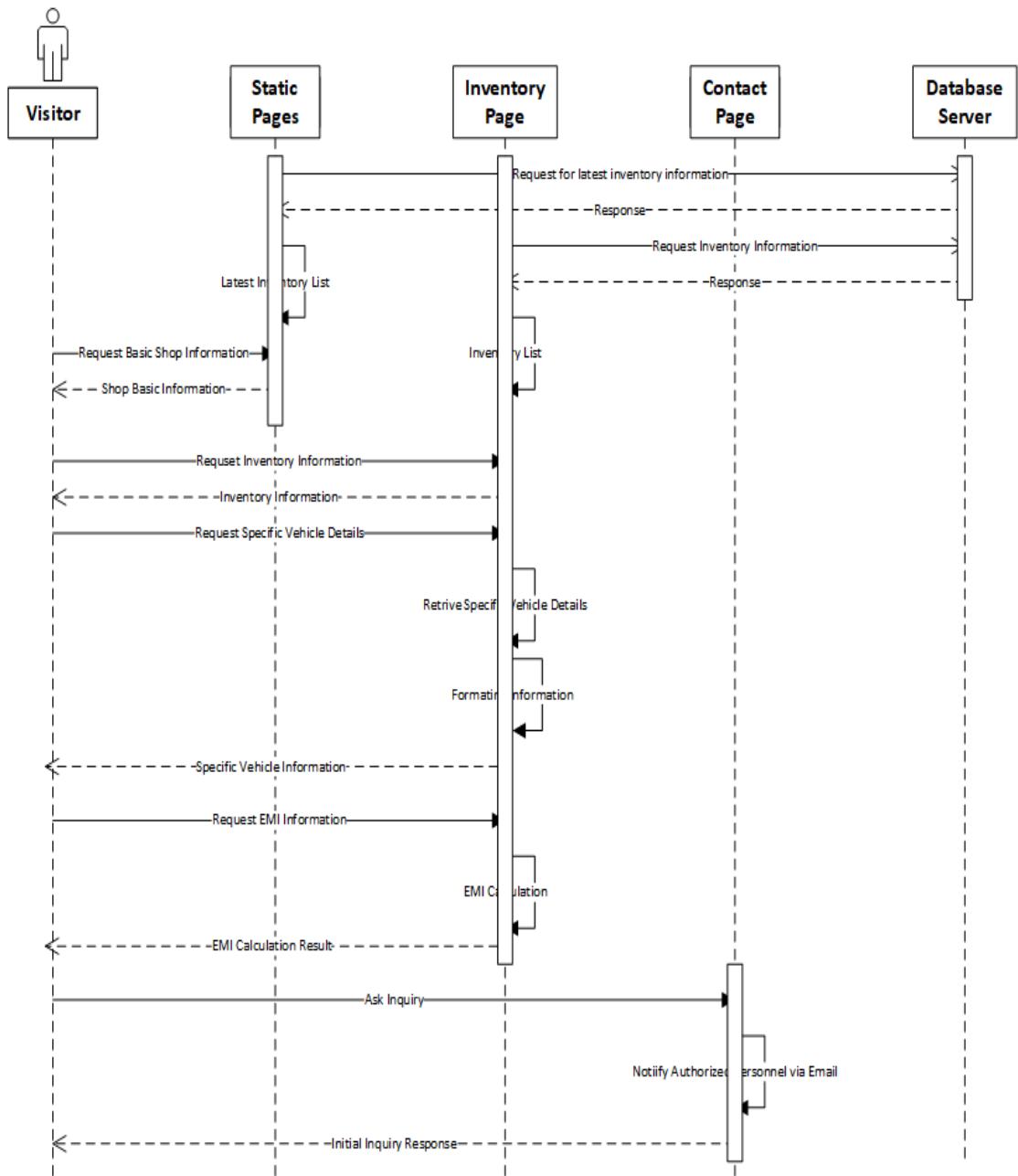


Figure 5.30: Sequence Diagram of Web Site

5.5 Class Diagrams

In this project we designed all the activity diagrams by component wise in spite of making it more understandable, also known as UML 1.1. Let's discuss it below.

This is the class diagram of authentication component.

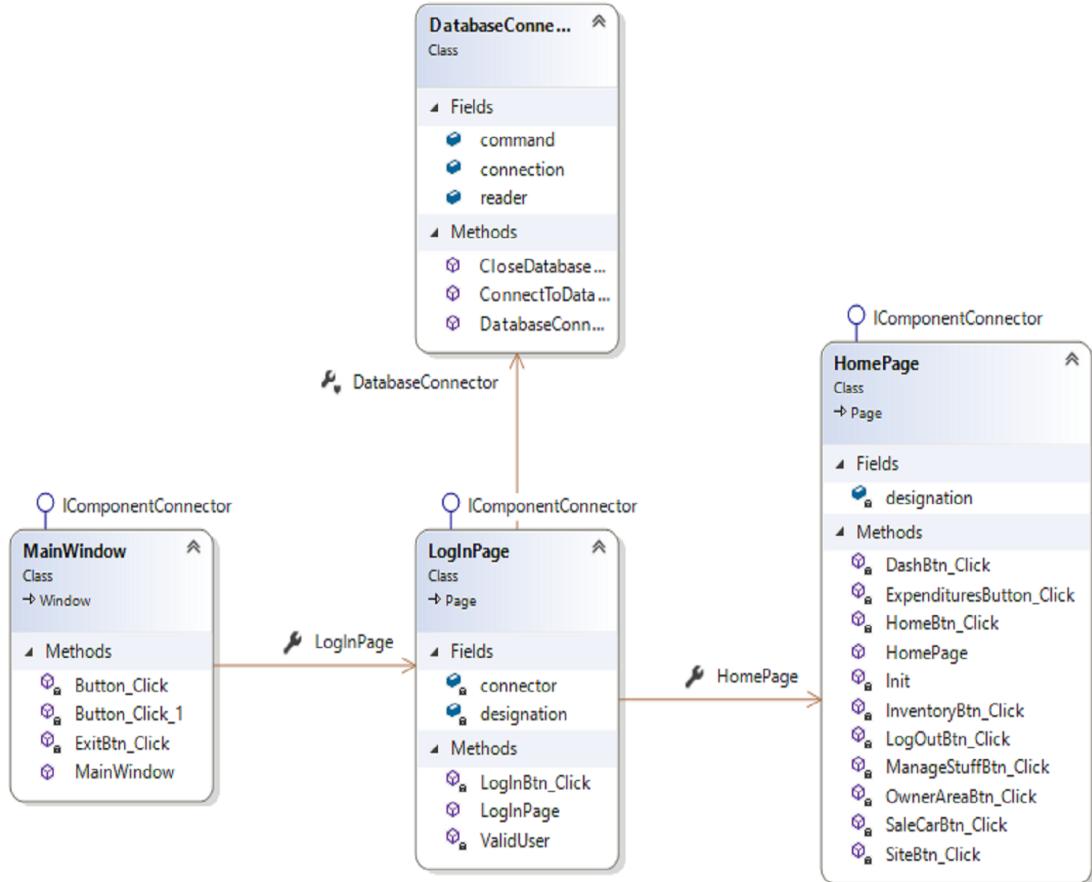


Figure 5.31: Class Diagram of Authentication Component

This is the class diagram of inventory component.

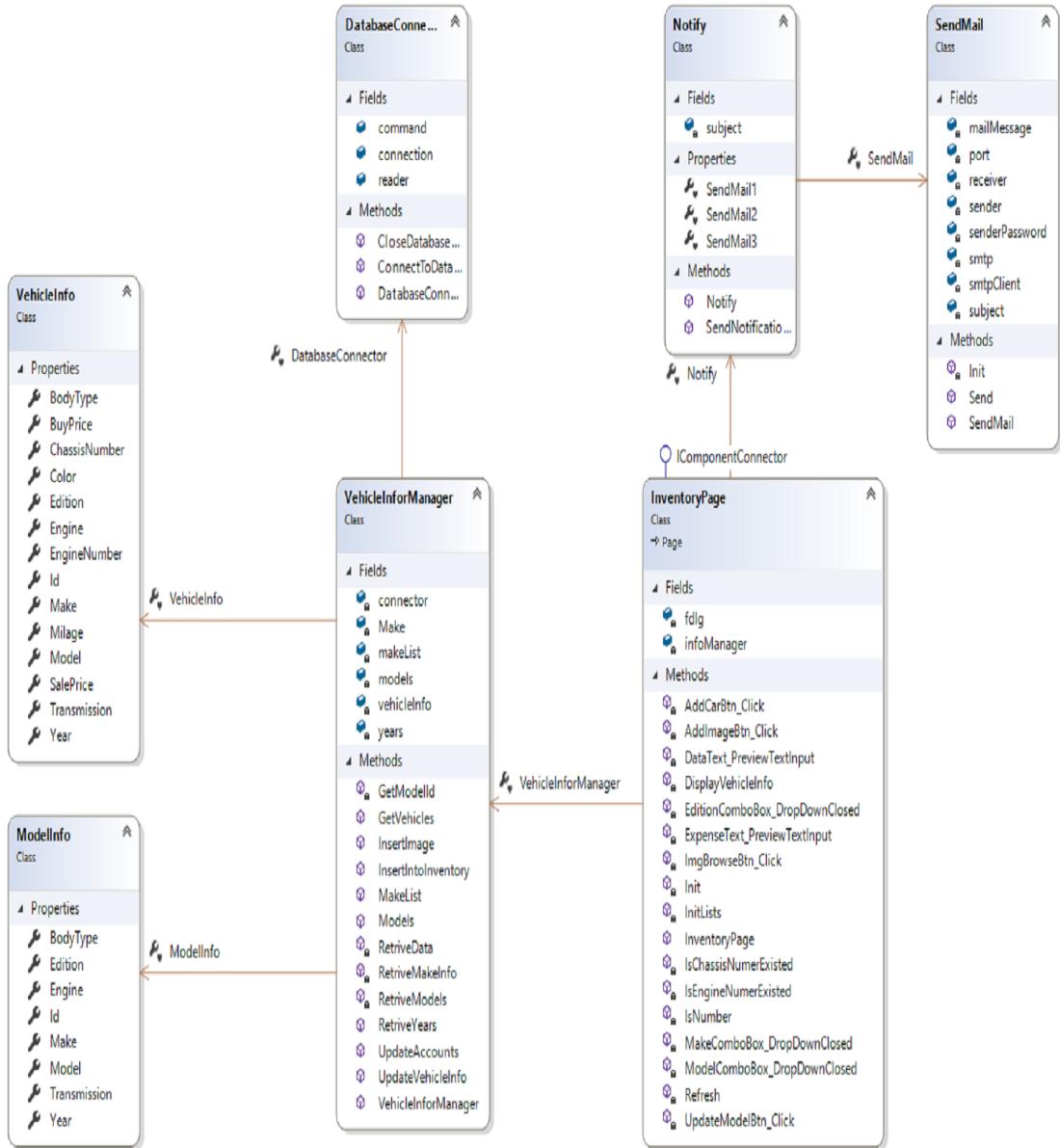


Figure 5.32: Class Diagram of Inventory Component

This is the class diagram of sales component.

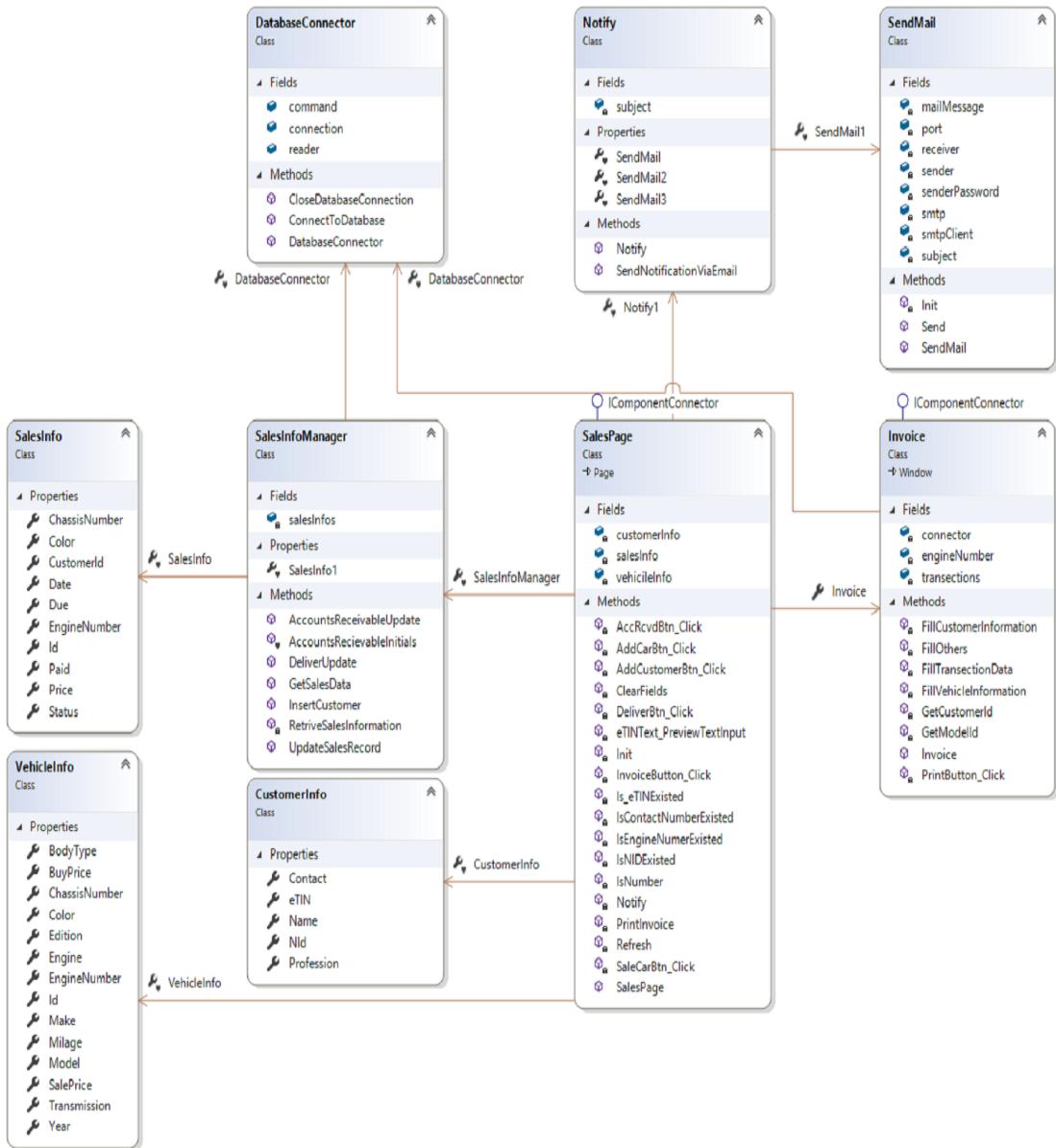


Figure 5.33: Class Diagram of Sales Component

This is the class diagram of manage stuffs component.



Figure 5.34: Class Diagram of Manage Stuffs Component

This is the class diagram of expenditures component.

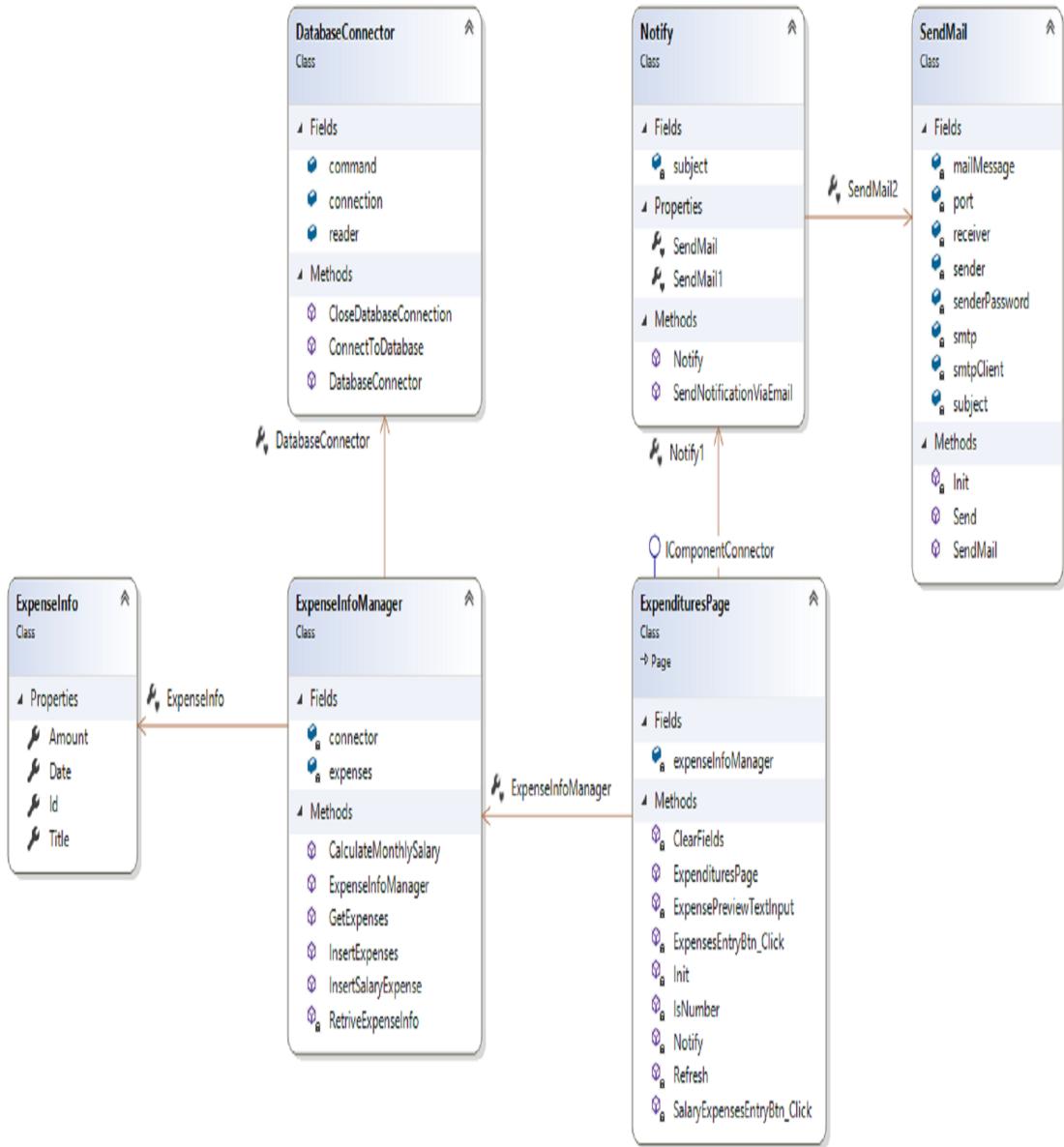


Figure 5.35: Class Diagram of Expenditures Component

This is the class diagram of owner area component.

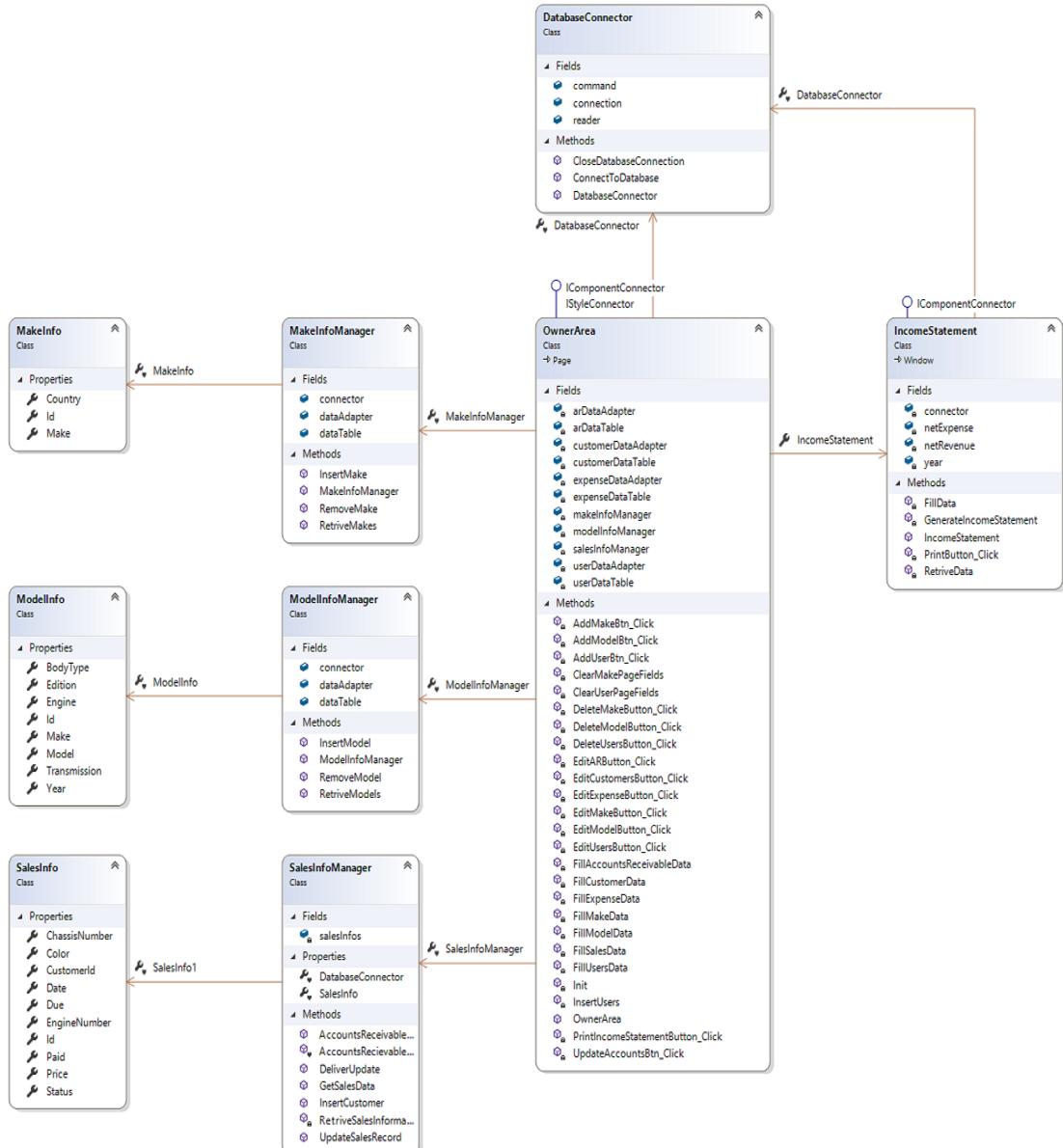


Figure 5.36: Class Diagram of Owner Area Component

This is the class diagram of decision support or dashboard component.

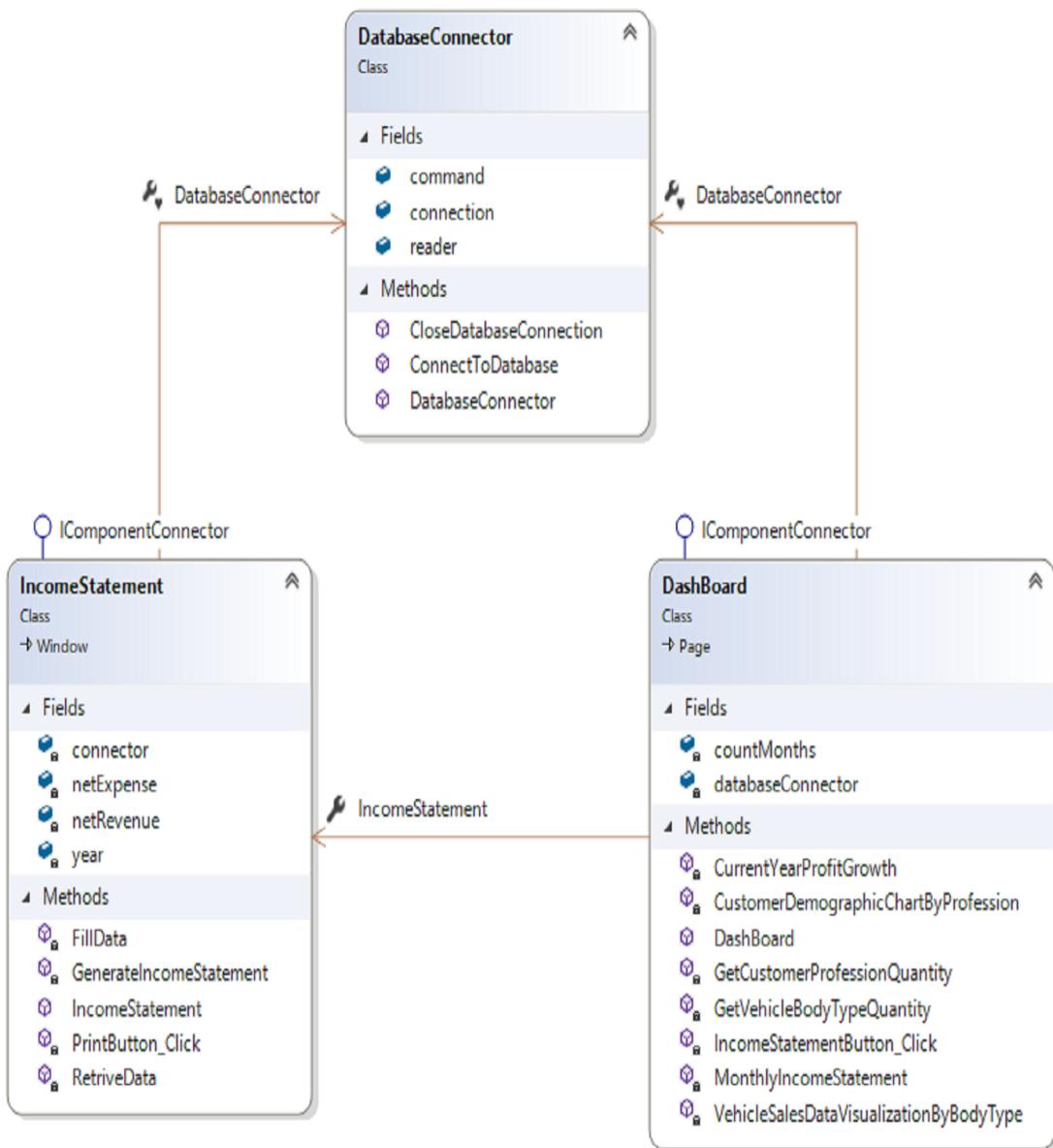


Figure 5.37: Class Diagram of Dashboard Component

This is the class diagram of support component.

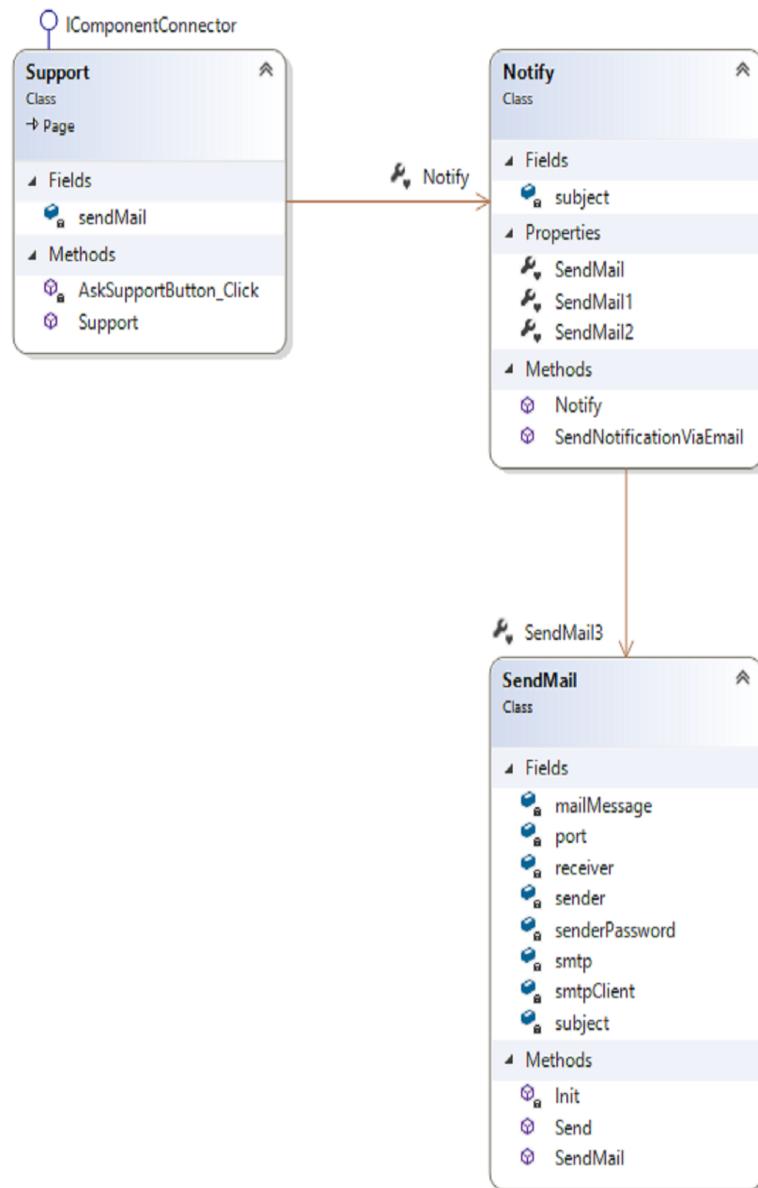


Figure 5.38: Class Diagram of Support Component

This is the class diagram of web site.

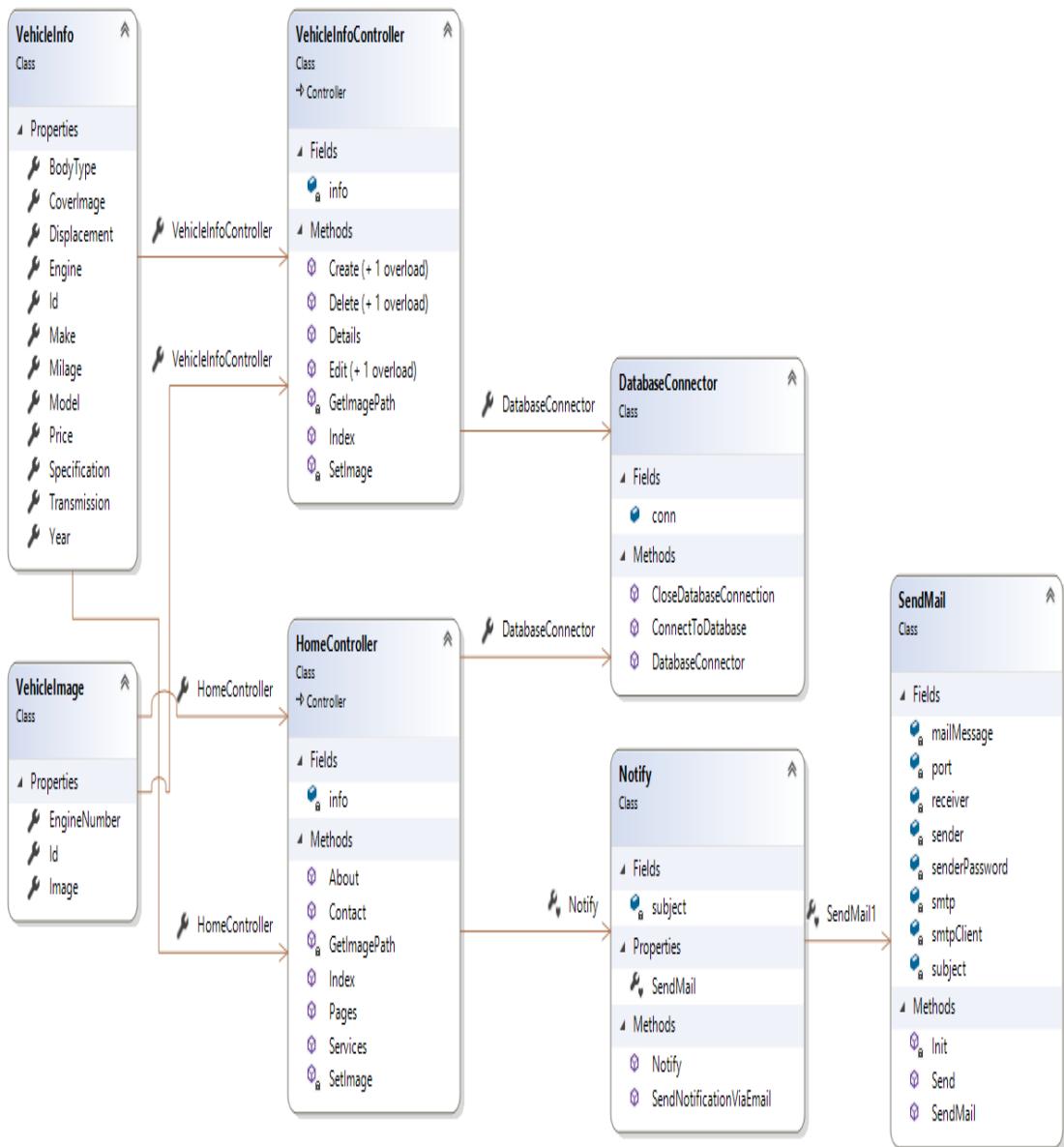


Figure 5.39: Class Diagram of Web Site

5.6 State Chart Diagram

State chart diagram is designed only for the critical state(s) of a system. Here only the accounts system has a critical state. The diagram is given below.

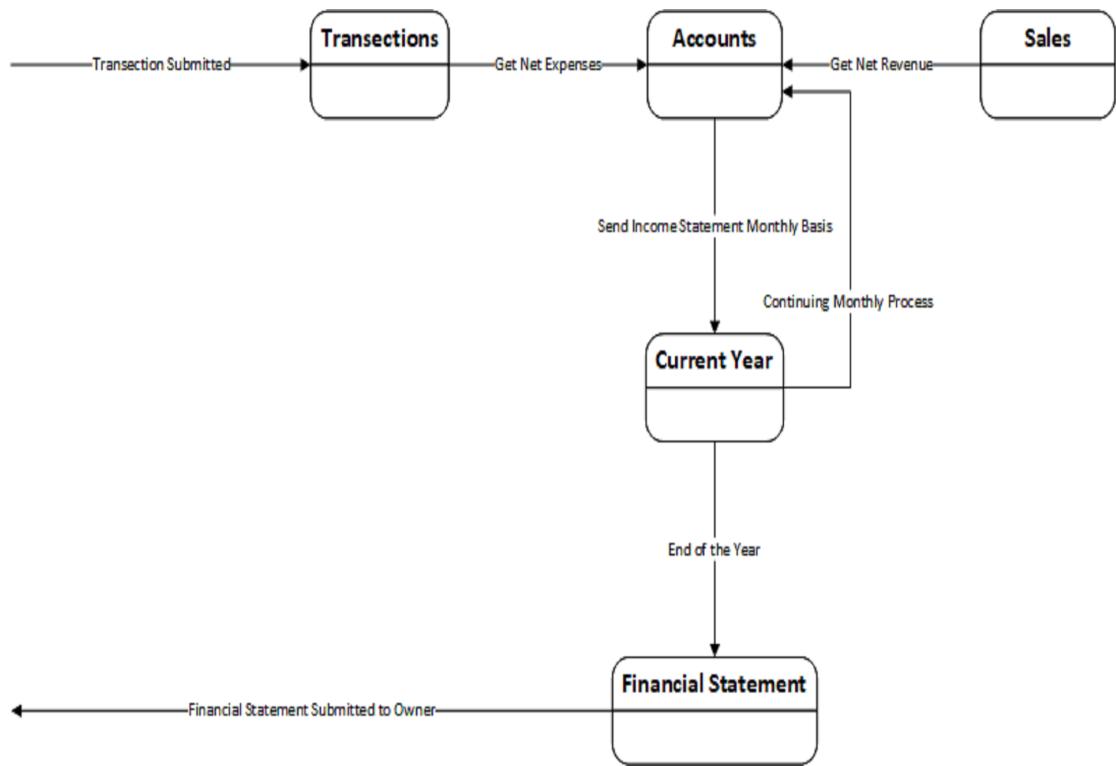


Figure 5.40: State Chart Diagram of Car Trading Information System

5.7 Database Design

5.7.1 Schema Diagram for our database

This is the design of Schema Diagram for the project's database. It is near physical design of the database.



Figure 5.41: Schema Diagram

5.7.2 Entity Relationship Diagram for our database

This is the design of Entity Relationship Diagram (ERD) for this project's database. It can be call as the blue print of the system. This is the logical design of the database.

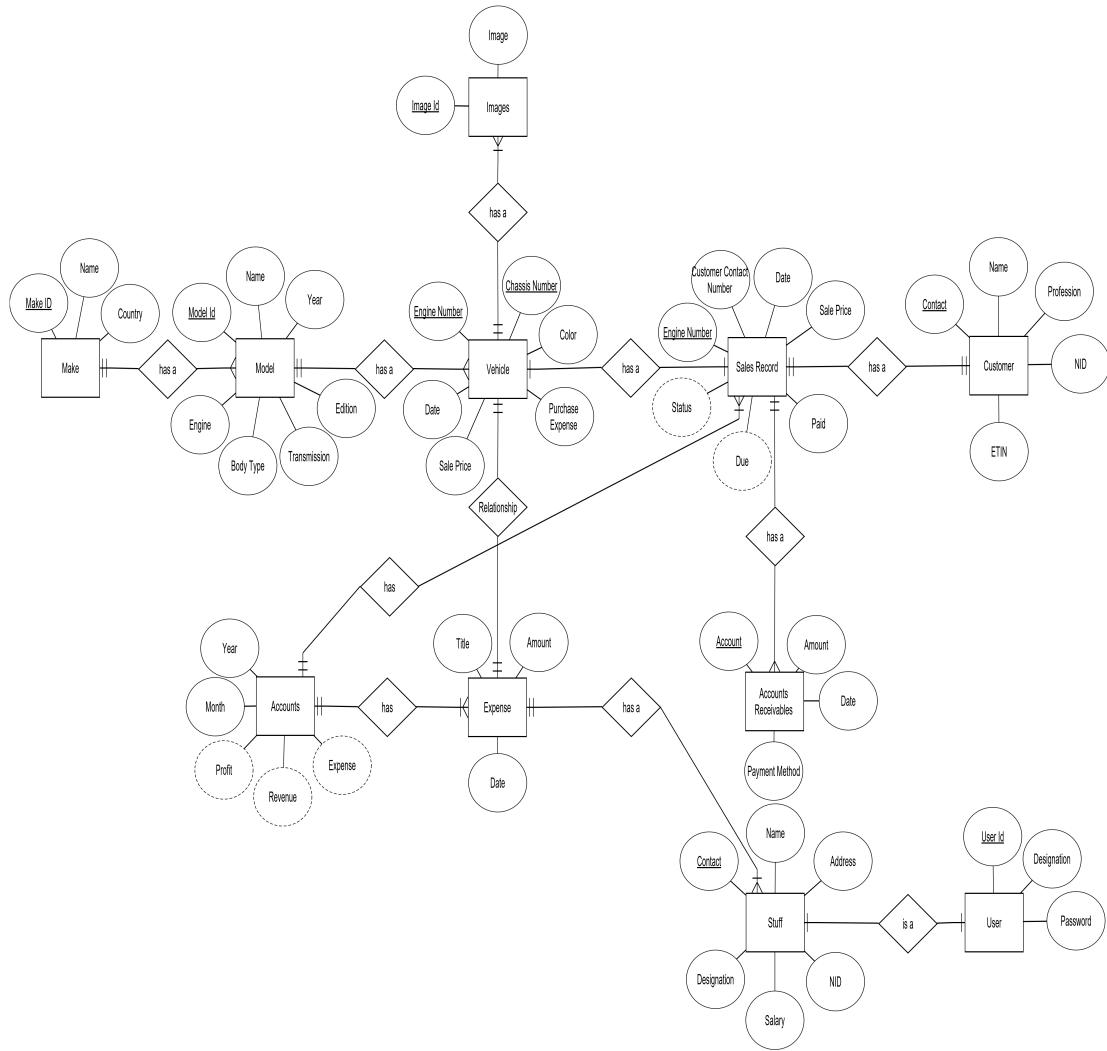


Figure 5.42: Entity Relationship Diagram

5.8 Database Object Design

5.8.1 Accounts

Column Name	Data Type	Nullable
Id	int	False
Month	varchar(50)	True
Year	int	True
Revenue	numeric(18,0)	False
Expense	numeric(18,0)	False
Profit	numeric(18,0)	False

Table 5.1: Accounts

5.8.2 AccountsReceivables

Column Name	Data Type	Nullable
Id	int	False
Account	nvarchar(MAX)	False
Amount	numeric(18,0)	False
Date	date	False
PaymentMethod	varchar(50)	False

Table 5.2: AccountsReceivables

5.8.3 Expenses

Column Name	Data Type	Nullable
Id	int	False
Title	nvarchar(MAX)	False
Amount	numeric(18,0)	False
Date	date	False

Table 5.3: Expenses

5.8.4 Customers

Column Name	Data Type	Nullable
Id	int	False
Contact	nvarchar(50)	False
Name	varchar(MAX)	False
Profession	varchar(MAX)	False
NID	nvarchar(MAX)	True
ETIN	nvarchar(MAX)	True

Table 5.4: Customers

5.8.5 Images

Column Name	Data Type	Nullable
Id	int	False
EngineNumber	nvarchar(MAX)	False
Image	image	False

Table 5.5: Images

5.8.6 Make

Column Name	Data Type	Nullable
Id	int	False
Make	nvarchar(MAX)	False
Country	nvarchar(MAX)	True

Table 5.6: Make

5.8.7 Model

Column Name	Data Type	Nullable
Id	int	False
MakeId	int	False
Name	varchar(MAX)	False
Year	int	True
Edition	nvarchar(MAX)	True
Engine	varchar(50)	False
BodyType	varchar(50)	False
Transmission	varchar(50)	False

Table 5.7: Model

5.8.8 Sales Record

Column Name	Data Type	Nullable
Id	int	False
EngineNumber	nvarchar(MAX)	False
ChassisNumber	nvarchar(MAX)	False
Color	varchar(50)	False
SalePrice	numeric(18,0)	False
Paid	numeric(18,0)	False
Due	numeric(18,0)	False
Status	varchar(50)	False
CustomerContact	nvarchar(50)	False
ModelId	int	False
Date	date	False

Table 5.8: SalesRecord

5.8.9 Stuff

Column Name	Data Type	Nullable
Id	int	False
Name	nvarchar(MAX)	False
Designation	nvarchar(MAX)	False
Salary	numeric(18,0)	False
NID	nvarchar(50)	True
Address	nvarchar(MAX)	True
Contact	nvarchar(50)	True

Table 5.9: Stuff

5.8.10 Users

Column Name	Data Type	Nullable
Id	int	False
Account	varchar(50)	False
Designation	varchar(50)	False
Passowrd	varchar(50)	False

Table 5.10: Users

5.8.11 Vehicle

Column Name	Data Type	Nullable
EngineNumber	nvarchar(MAX)	False
ChassisNumber	nvarchar(MAX)	False
Color	varchar(50)	False
Purchase	numeric(18,0)	False
SalePrice	numeric(18,0)	True
ModelId	int	False
Date	date	False
Description	varchar(MAX)	True
Milage	nvarchar(50)	True

Table 5.11: Vehicle

5.9 Data Flow Diagram

5.9.1 Data Flow Diagram (0 Level)

User and Admin can login our system. And they can use some feature of our software those are: add student, manage accounts, exam, result etc. All those data are store in SQL database[64].

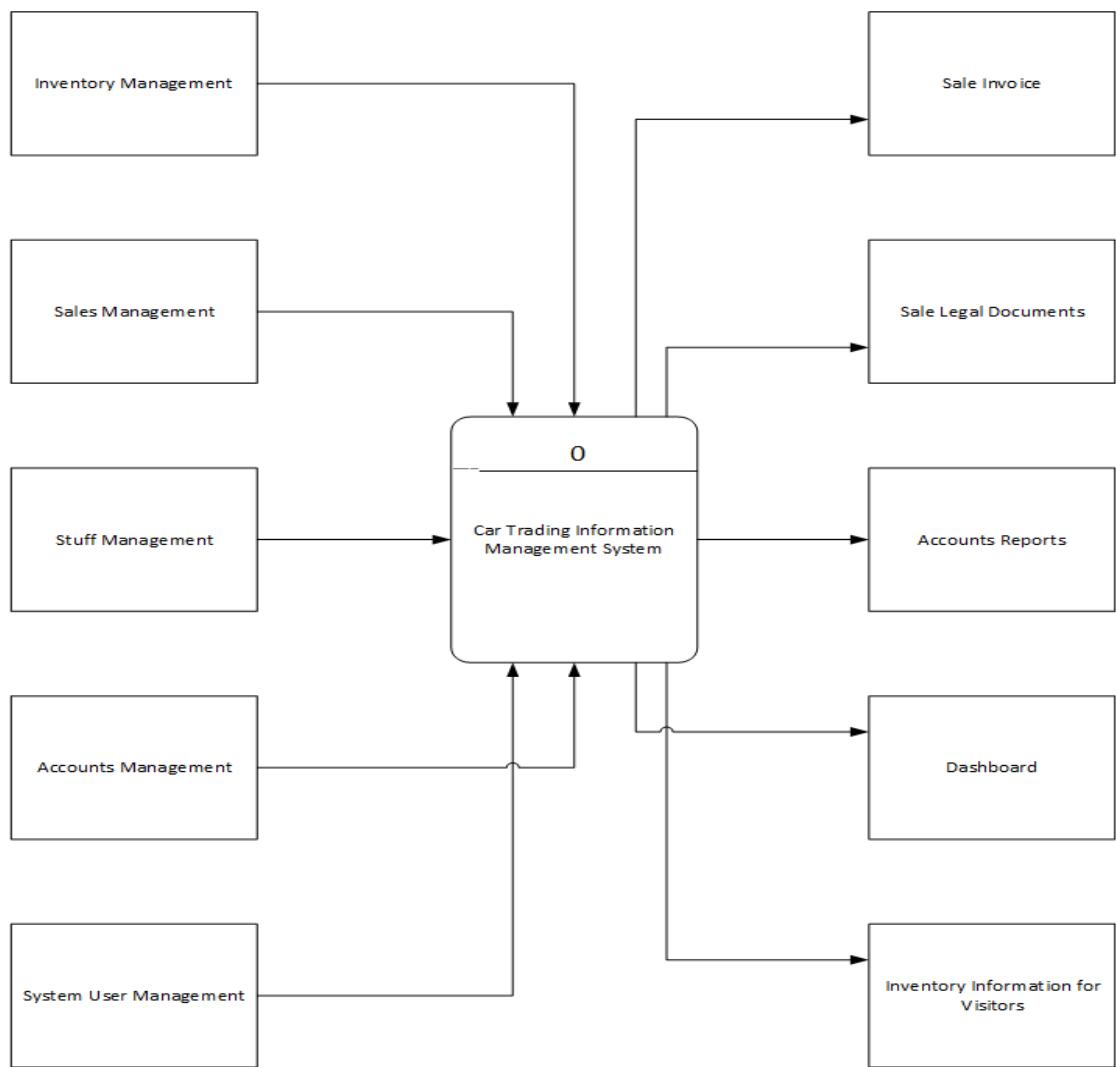


Figure 5.43: 0 Level DFD

5.9.2 Data Flow Diagram (1 Level)

User and admin login in our system, they can view available/unavailable car. If a user choose a car then the user booking the car. Admin view those booking and approve those booking, when admin approve any booking the user got an email that is conformation email[65].

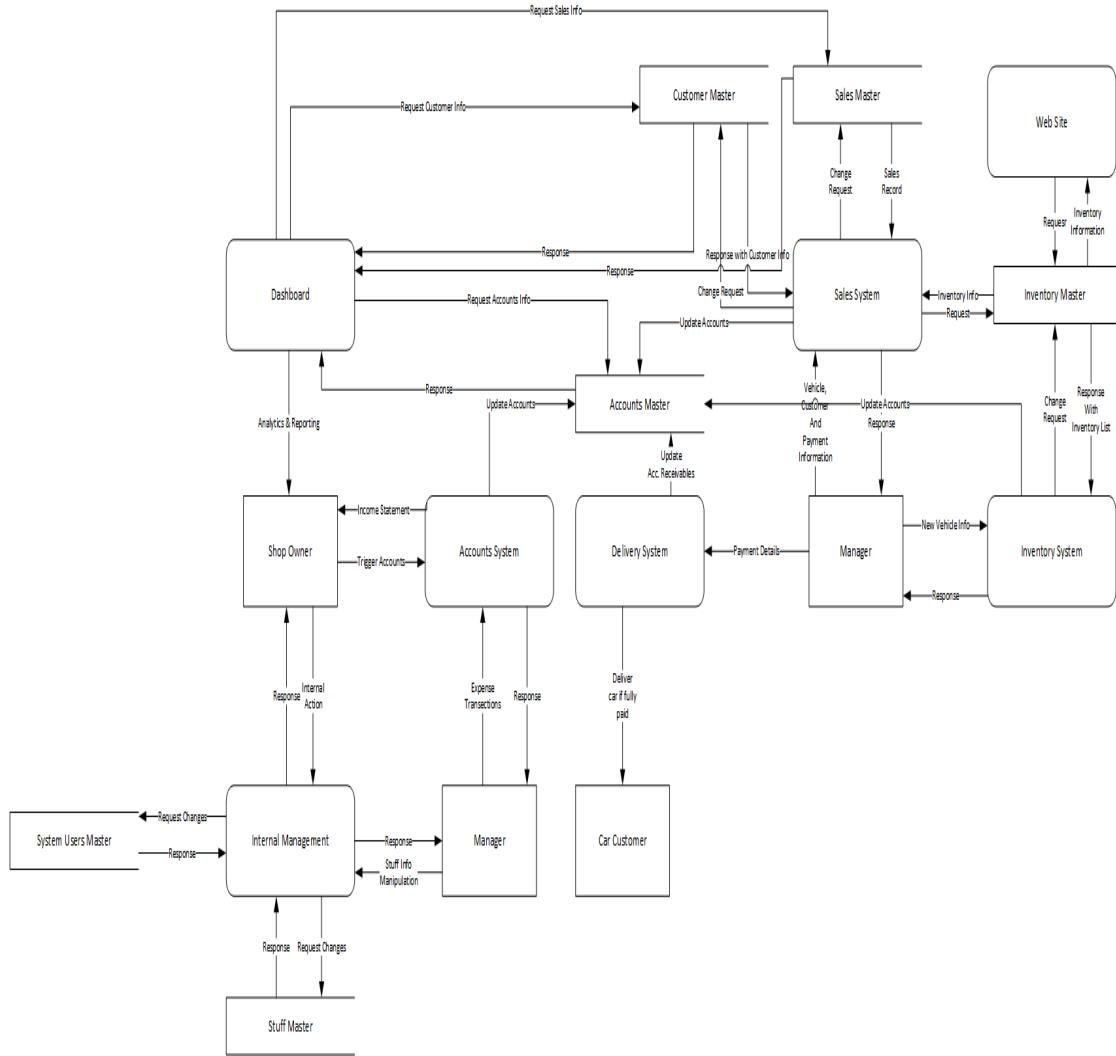


Figure 5.44: 1 Level DFD

5.10 Implementation

The implementation phase involves putting the project plan into action. Its here that the project manager will coordinate and direct project resources to meet the objectives of the project plan. As the project unfolds, its the project managers job to direct and manage each activity, every step of the way. Thats what happens in the implementation phase of the project life cycle: you follow the plan youve put together and handle any problems that come up. The implementation phase is where you and your project team actually do the project work to produce the deliverables. The word deliverable means anything your project delivers. The

deliverables for your project include all of the products or services that you and your team are performing for the client, customer, or sponsor, including all the project management documents that you put together.

5.11 Result Analysis

In any customer projects with more lead time (at least 3 months), RA plays an important role. Results Analysis is to evaluate ongoing unfinished activities, in projects during period-end.

If you look at profit and loss of such ongoing unfinished activities, you will see costs only and therefore your P&L shows loss. If you look at this in the period-end, the projects ongoing activities will have an unfavorable effect on the company's results. Accurate, timely recognition of project profitability, for each project, for every period end, is very important in any company.

SAP Results Analysis is used to show a more realistic view of your ongoing activities by capitalizing the value added so far in the balance sheet.

5.12 Conclusions

In the logical database design phase, the model of the data to be used is based on a specific data model, but independent of a particular database management system is constructed. This is based on the target data model for the database e.g. relational data model.

In the physical database design phase, the description of the implementation of the database on secondary storage is created. The base relations, indexes, integrity constraints, security, etc. are defined using the SQL language. Project design is an early phase of the project where a project's key features, structure, criteria for success, and major deliverables are all planned out. The point is to develop one or more designs which can be used to achieve the desired project goals, in this chapter we try to design Easy Rent system properly.

Chapter 6

DEVELOPMENT

6.1 Introduction

The primary goal during the Developing Phase is to build the solution components code as well as documentation. Some development work may, however, continue into the Stabilizing Phase in response to testing.[47] To ensure context sensitivity though an open, consensus-building dialog among project proponents, reviewers, the public, and other parties. To foster thinking beyond the roadway pavement to achieve the optimum accommodation for all modes. To encourage early planning, public outreach, and evaluation so that project needs, goals and objectives, issues, and impacts can be identified before significant resources are expended. To achieve consistent expectations and understanding between project proponents and those entities who evaluate, prioritize, and fund projects. To ensure allocation of resources to projects that address local, regional, and statewide priorities and needs.[47] The Developing Phase involves more than code development and software developers. The infrastructure is also developed during this phase and all roles are active in building and testing deliverables. The team continues to identify all risks throughout the phase and address new risks as they emerge[?].

6.2 Starting the development cycle

Guidance for this phase introduces and discusses each code component, discusses how to apply the code, and looks at adapting and extending the components to meet the needs of the project requirements.[16]

6.3 Building a proof of concept

Before development, the team does a final verification of the concepts from the designs within an environment that mirrors production as closely as possible.[19]

Typically, the proof of concept is a continuation of some initial development work (the preliminary proof of concept) that occurred during the Planning Phase. The proof of concept tests key elements of the solution on a non-production simulation of the proposed operational environment. The team walks operations staff and users through the solution to validate their requirements.[66]

There may be some solution code or documentation that carries through to the eventual solution development deliverables; however, the proof of concept is not meant to be production-ready. The proof of concept is considered throw away development that gives the team a final chance to verify functional specification content and to address any more issues prior to transitioning into development.[62]

6.4 Developing the solution components

The team develops the solution using the core components and extending them to the specific needs of the solution. The team also develops and conducts unit functional tests to ensure that individual features perform according to specification.[26]

MSF recommends that project teams follow a best practice of performing daily builds with their solution. Building a solution in a form that is executable on a daily basis provides a number of valuable benefits simply by putting different pieces of the code together. A daily build exposes unanticipated design defects and makes diagnosing defects easier.[34]

The daily build should be subjected to as much of the full suite of tests as can be run during the available time. This build validation test pass helps expose integration defects as early as possible. It also allows the team to validate their testing approach and testing infrastructure.[32]

6.5 Testing

The team develops a testing infrastructure and populates it with test cases that help ensure the entire solution performs according to specification. This solution test suite typically incorporates, as a subset, the individual feature tests used by developers in building the solution components.[28]

MSF advocates preparing frequent builds of all the components of the solution for testing and review. This approach is recommended for developing code as well as for builds of hardware and software components. The process of creating interim builds allows a team to find issues early in the development process, which shortens the development cycle and lowers the cost of the project. Daily builds are the practice of assembling all the components working toward the final goal of a solution. This enables the team to determine earlier rather than later that all components will work together. This method also allows the team to add functionality onto a stable build. The idea is to have a shippable product ready at any point in time. In this way, the stability of the total solution is well understood and has ample test data prior to being released into production.[29]

In this project, according to our resources, we were eligible to perform several vital tests. These tests are described below:

6.5.1 White Box Testing

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs. Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system.

This method is named so because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees[67].

In this method of testing, there be several types of testing performed according to our available resources.

6.5.1.1 Component Testing

A component is the lowest unit of any application. So, Component testing; as the name suggests, is a technique of testing the lowest or the smallest unit of any application. Component testing sometimes is also referred to as Program or Module Testing. An application can be thought of a combination and integration of many small individual modules. Before we test the entire system, it is imperial that each component OR the smallest unit of the application is tested thoroughly. In this case, the modules or the units are tested independently. Each module receives an input, does some processing and generates the output. The output is then validated against the expected feature[68]. For example, sales process is a

component of the system, whether it works individually perfectly or not, it can be identified by this kind of testing.

6.5.1.2 Integration Testing

Integration Testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing[69].

For example, the component sales, is interrelated with accounts component and inventory component. After a product sale, it will update both accounts and inventory component including self. This operation happening perfectly or not can be identified by this kind of testing.

6.5.1.3 System Testing

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements[70].

6.5.2 Black Box Testing

BLACK BOX TESTING, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional[71].

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external database access.
- Behavior or performance errors.
- Initialization and termination errors.

We performed this testing to observe end-user uses issues because this method of testing is really effective to identify this. From this, the testing phase has closed and we reached the situation that what exactly we required to implement is successfully performed by all the stakeholders.

6.6 Building the solution

A series of daily, or frequent, builds culminate with major internal builds and signify points where the development team is delivering on key features of the solution. These builds are subjected to some or all of the project test suite as a way of tracking overall progress of the solution and of the solution test suite itself.[63]

6.7 Closing the Developing Phase

When all tests stands successful after implementation and there be no error or issue reports provided by tester then it can be assume that all the required tasks completed successfully and consider it ready to use. Then the team completes all features, delivers the code and documentation, and considers the solution complete, thus entering the approval process for the Scope Complete Milestone[72][66].

6.8 Conclusions

This project development process is complemented by the inclusion of the projects context as a basic design control. Flexibility for determining specific design elements that satisfy the project need, and are responsive to the context of the project, is inherent in the subsequent chapters of this Guidebook. The Developing Phase culminates in the Scope Complete Milestone. At this milestone, all features are complete and the solution is ready for external testing and stabilization. This milestone is the opportunity for customers and users, operations and support personnel, and key project stakeholders to evaluate the solution and identify any remaining issues they need to address before beginning the transition to stabilization and ultimately to release.

Chapter 7

USER MANUAL

7.1 Introduction

If you are running a small business that has a few employees, you understand that it's hard to manage the dynamic nature of doing businesses. Small businesses are not so different from large ones. They contain most of the complexities of a large business along with many other constraints. Small businesses have to communicate with customers, do accounts, pay taxes, do payroll, manage timelines, deliver quality goods and services, answer questions, and keep everyone happy, just like in large businesses.

Large businesses have the advantage of using advanced data systems to manage their process efficiently. Small businesses, on the other hand, typically struggle to keep things organized. They often use a mix of apps like spreadsheets, accounting software, web CRM etc to manage. The problem is, not everyone is on the same page. An ERP changes that.[67] This chapter will show the Graphical User Interface (GUI) of the system through screenshots and user guides that how will they operate the application. A complete direction for a user to properly run Online Judge.

7.2 Hardware requirements

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements:

minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements. A second meaning of the term of System requirements is a generalization of this first definition, giving the requirements to be met in the design of a system or sub-system. Typically an organization starts with a set of Business requirements and then derives the System requirements from there. So the most important hardware requirements are-

7.2.1 Database Server

A computer is a device that can be instructed to carry out arbitrary sequences of arithmetic or logical operations automatically. The ability of computers to follow generalized sets of operations, called programs, enables them to perform an extremely wide Range of tasks.[?]

Configuration:

PROCESSOR / CHIPSET :

CPU - Intel Core i7 (7th Gen) 7500U / 2.7 GHz
Max Turbo Speed - 3.5 GHz
Number of Cores - Dual-Core
Cache - L3 - 4 MB
64-bit Computing - Yes

RAM :

Memory Speed - 2133 MHz
Memory Specification Compliance - PC4-17000
Configuration Features - 1 x 8 GB
Technology - DDR4 SDRAM
Installed Size - 8 GB
Rated Memory Speed - 2133 MHz

MEMORY :

Max Supported Size - 16 GB
Technology - DDR4 SDRAM
Speed - 2133 MHz / PC4-17000 - 2133 MHz
Form Factor - SO-DIMM 260-pin

Slots Qty - 2
Empty Slots - 1

7.2.2 Web Server

A computer is a device that can be instructed to carry out arbitrary sequences of arithmetic or logical operations automatically. The ability of computers to follow generalized sets of operations, called programs, enables them to perform an extremely wide Range of tasks.[?]

Configuration:

PROCESSOR / CHIPSET :

CPU - Intel Core i7 (7th Gen) 7500U / 2.7 GHz
Max Turbo Speed - 3.5 GHz
Number of Cores - Dual-Core
Cache - L3 - 4 MB
64-bit Computing - Yes

RAM :

Memory Speed - 2133 MHz
Memory Specification Compliance - PC4-17000
Configuration Features - 1 x 8 GB
Technology - DDR4 SDRAM
Installed Size - 8 GB
Rated Memory Speed - 2133 MHz

MEMORY :

Max Supported Size - 16 GB
Technology - DDR4 SDRAM
Speed - 2133 MHz / PC4-17000 - 2133 MHz
Form Factor - SO-DIMM 260-pin
Slots Qty - 2
Empty Slots - 1

7.2.3 End User

In product development, an end user (sometimes end-user) is a person who ultimately uses or is intended to ultimately use a product.[1][2][3] The end user stands in contrast to users who support or maintain the product,[4] such as sysops, system

administrators, database administrators,[5] information technology experts, software professionals and computer technicians. End users typically do not possess the technical understanding or skill of the product designers,[6] a fact that is easy for designers to forget or overlook, leading to features with which the customer is dissatisfied.[2] In information technology, end users are not "customers" in the usual sense they are typically employees of the customer.[7] For example, if a large retail corporation buys a software package for its employees to use, even though the large retail corporation was the "customer" which purchased the software, the end users are the employees of the company who will use the software at work.

Configuration of our Computer :

PROCESSOR / CHIPSET :

CPU - Intel Core i7 (7th Gen) 7500U / 2.7 GHz
Max Turbo Speed - 3.5 GHz
Number of Cores - Dual-Core
Cache - L3 - 4 MB
64-bit Computing - Yes

RAM :

Memory Speed - 2133 MHz
Memory Specification Compliance - PC4-17000
Configuration Features - 1 x 8 GB
Technology - DDR4 SDRAM
Installed Size - 8 GB
Rated Memory Speed - 2133 MHz

MEMORY :

Max Supported Size - 16 GB
Technology - DDR4 SDRAM
Speed - 2133 MHz / PC4-17000 - 2133 MHz
Form Factor - SO-DIMM 260-pin
Slots Qty - 2
Empty Slots - 1

7.3 Software requirements

In the software development process, requirement phase is the first software engineering activity. This phase is a user-dominated phase and translates the ideas or views into a requirements document. Note that defining and documenting the user requirements in a concise and unambiguous manner is the first major step to achieve a high-quality product.[80]

7.3.1 Database Server

A server is a computer program that provides a service to another computer programs (and its user). In a data center, the physical computer that a server program runs in is also frequently referred to as a server. That machine may be a dedicated server or it may be used for other purposes as well.

7.3.2 Web Server

A server is a computer program that provides a service to another computer programs (and its user). In a data center, the physical computer that a server program runs in is also frequently referred to as a server. That machine may be a dedicated server or it may be used for other purposes as well.

7.3.3 Client

A client is a computer or a program that, as part of its operation, relies on sending a request to another program or a computer hardware or software that accesses a service made available by a server(which may or may not be located on another computer). For example, web browsers are clients that connect to web servers and retrieve web pages for display. Email clients retrieve email from mail servers. Online chat uses a variety of clients, which vary depending on the chat protocol being used. Multi-player video games or online video games may run as a client on each computer. The term "client" may also be applied to computers or devices that run the client software or users that use the client software.

7.4 Interfaces

7.4.1 Desktop Application

7.4.1.1 Log In Page

This is first page of our application. Everybody can see this page without authentication.

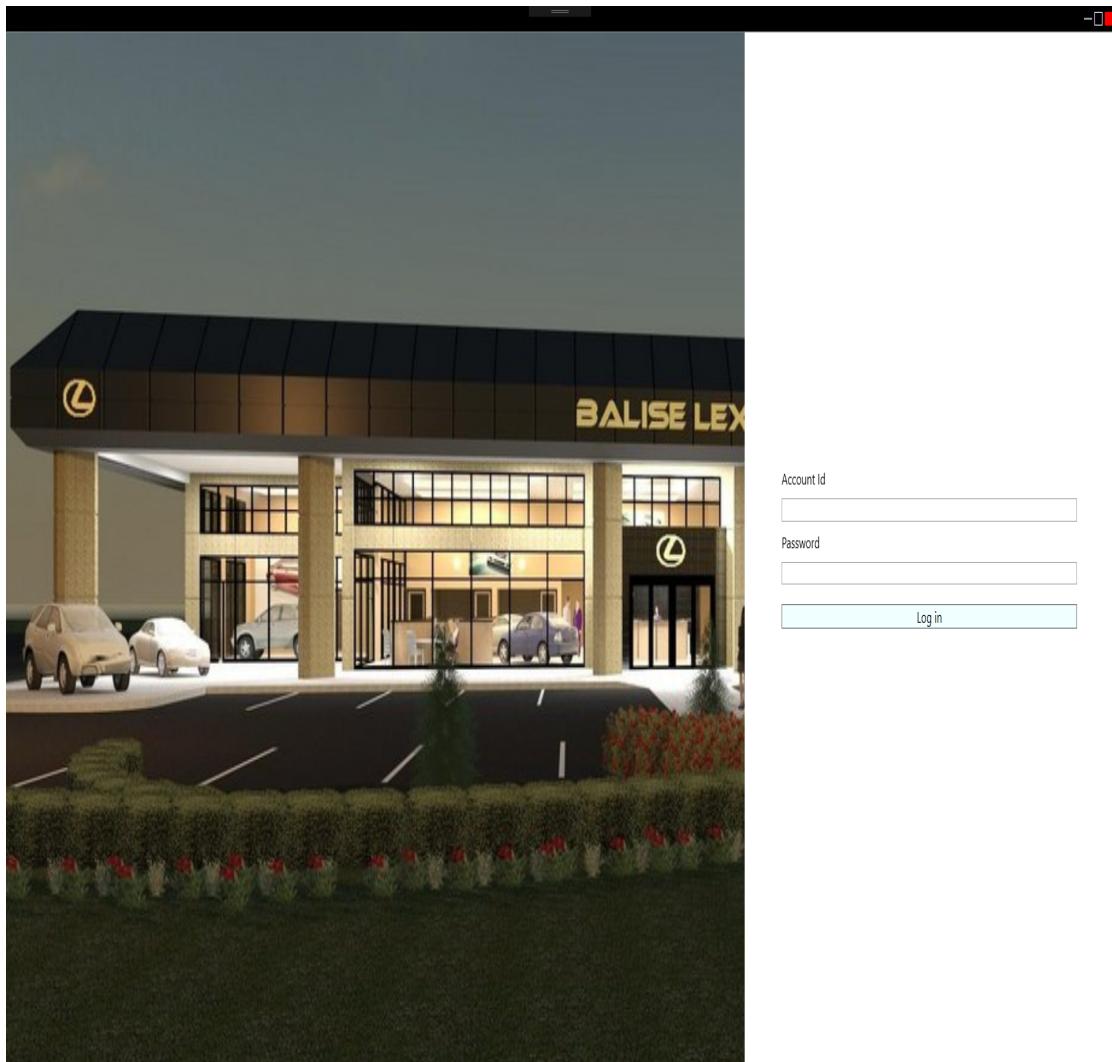


Figure 7.1: Log In Page

7.4.1.2 Home Page

This is first page of our application. Everybody can see this page after successful authentication.

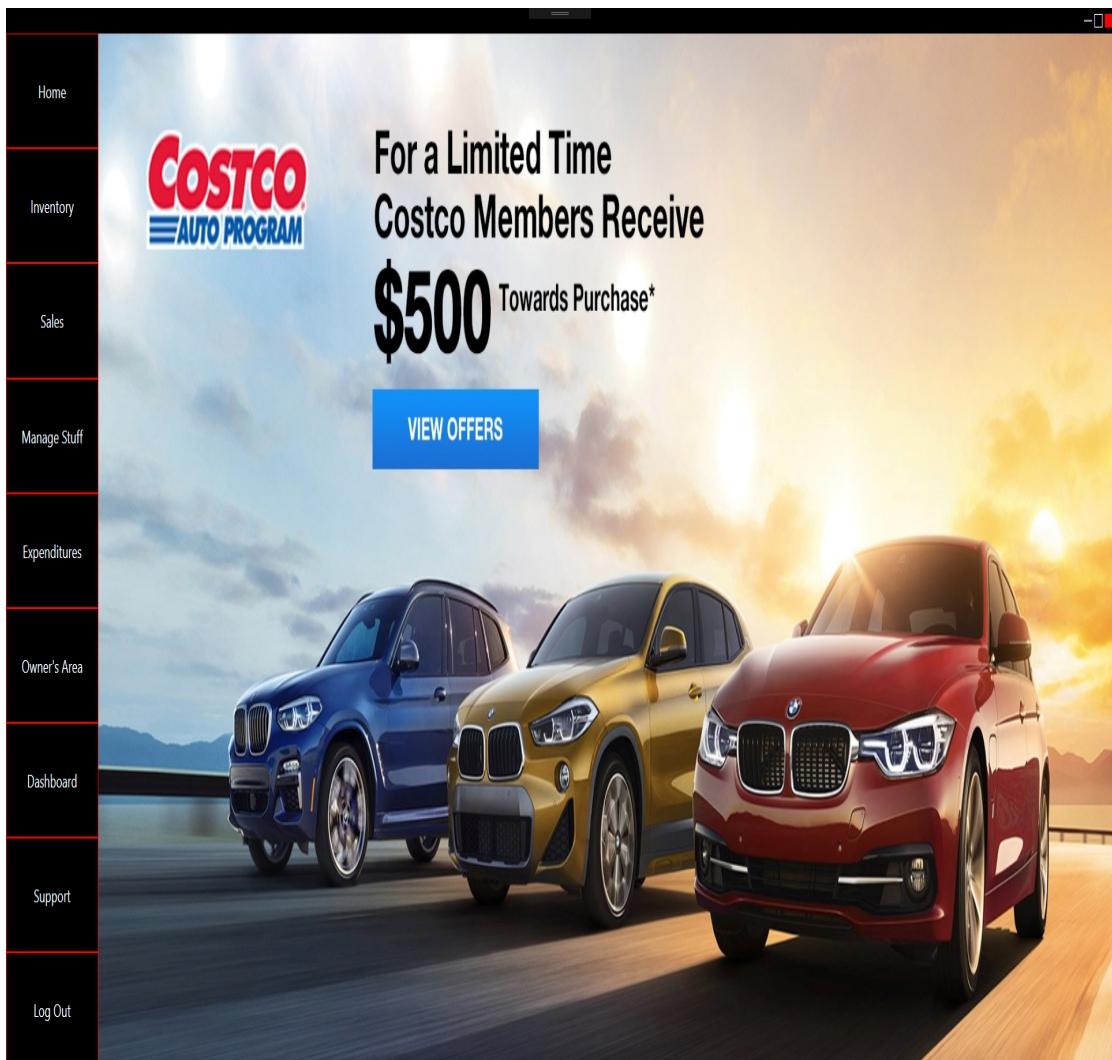


Figure 7.2: Home Page

7.4.1.3 Inventory Section

This is the inventory section where user can insert new vehicle information and manipulate it.

The screenshot shows a web-based application interface for managing vehicle inventory. On the left, there is a vertical sidebar menu with the following items:

- Home
- Inventory
- Sales
- Manage Stuff
- Expenditures
- Owner's Area
- Dashboard
- Support
- Log Out

At the top right of the main content area, there are three buttons: "Add Car", "Add Images", and "Update Information". Below these buttons, there are several input fields and dropdown menus:

- Make: A dropdown menu.
- Model: A dropdown menu.
- Edition: A dropdown menu.
- Year: A dropdown menu.
- Engine Number: An input field.
- Chassis Number: An input field.
- Color: A dropdown menu.
- Mileage: An input field.
- Total Purchase Expense: An input field.
- Sale Price: An input field.
- Date: A date picker input field.
- Description: A large text area for notes.

At the bottom of the main content area, there is a single "Add" button.

The main content area also contains a table displaying vehicle data. The table has the following columns:

ID	Make	Model	Year	Edition	Engine	Body Type	Transmission	Engine Number	Chassis Number	Color	Buy Price	Sale Price	Mileage
17	Toyota	Land Cruiser	2018	ZX	4.5L V8	SUV	Auto-Manual	XCVD-S45U4	SAD7-FD5TW	Black	39850000	40000000	55000 Km
21	Nissan	Sunny	2014		1.5L VVTi	Sedan	Automatic	XCVD-S45U3	SAD7-FD5TV	Black	3237930	3300000	15000 Km
20	Nissan	X-Trail	2014	Gold	2.0L VVTi	SUV	Automatic	XCVD-S45U0	SAD7-FD5TS	Black	4135000	4250000	45000 Km
12	Toyota	Premio	2016	F	1.5L VVTi	Sedan	Automatic	XCVD-S45T8	SAD7-FD5T1	Black	3310000	3400000	20000 Km
14	Toyota	Allion	2016	G	1.5L VVTi	Sedan	Automatic	XCVD-S45T7	SAD7-FD5TP	Black	2875000	2925000	17300 Km
24	Mitsubishi	Pajero	2012	Sport	2.5L D4D	SUV	Manual	XCVD-S45T5	SAD7-FD5TN	Black	3998000	4150000	70000 Km
14	Toyota	Allion	2016	G	1.5L VVTi	Sedan	Automatic	XCVD-S45T6	SAD7-FD5T0	Red Wine	3025000	3100000	12500 Km
23	Mitsubishi	Pajero	2017	G Class	3.0L V6	SUV	Automatic	XCVD-S45T4	SAD7-FD5TM	Black	10368000	10500000	50000 Km
22	Mitsubishi	Outlander	2017	GL	2.4L VVTi	SUV	Automatic	XCVD-S45T3	SAD7-FD5TL	Perl White	4410000	4500000	45000 Km
17	Toyota	Land Cruiser	2018	ZX	4.5L V8	SUV	Auto-Manual	XCVD-S45T0	SAD7-FD5T1	Perl White	39610000	40000000	20000 Km
7	Nissan	Patrol	2017	Nismo	5.6L V8	SUV	Automatic	XCVD-S45S8	SAD7-FD5T1	Perl White	39500000	40000000	52300 Km
8	Toyota	Land Cruiser	2012	VX	4.5L V8	SUV	Automatic	XCVD-S45S7	SAD7-FD5TH	Black	18325000	18750000	45000 Km
12	Toyota	Premio	2016	F	1.5L VVTi	Sedan	Automatic	XCVD-S45S4	SAD7-FD5TE	Red Wine	3420000	3500000	15000 Km
2	Honda	Vezel	2014	X Package	1.5L VVTi	SUV	Automatic	XCVD-S45S2	SAD7-FD5TC	Perl White	2735000	2800000	25000 Km

Figure 7.3: Inventory Page

7.4.1.4 Sales Section

This is sales section where user can entry new customer information, sale vehicle, manage accounts receivables of sales, delivery information etc.

The screenshot shows a Windows application window titled "Sales Page". On the left is a vertical sidebar menu with the following items:

- Home
- Inventory
- Sales
- Manage Stuff
- Expenditures
- Owner's Area
- Dashboard
- Support
- Log Out

The main content area has a header with tabs: "Add Customer", "Sale Car", "Accounts Receivables", "Deliver Car", and "Invoice". Below the tabs is a "Customer Table" tab which is currently selected. The table displays the following data:

Inventory Table										Customer Table			Sales Table		
ID	Make	Model	Year	Edition	Engine	BodyType	Transmission	EngineNumber	ChassisNumber	Color	BuyPrice	SalePrice	Mileage		
17	Toyota	Land Cruiser	2018	ZX	4.5L V8	SUV	Auto-Manual	XCV0-S45U4	SAD7-FD5TW	Black	39850000	40000000	55000 Km		
21	Nissan	Sunny	2014		1.5L VVTi	Sedan	Automatic	XCV0-S45U3	SAD7-FD5TV	Black	3237930	3300000	15000 Km		
20	Nissan	X Trail	2014	Gold	2.0L VVTi	SUV	Automatic	XCV0-S45U0	SAD7-FD5TS	Black	4135000	4250000	45000 Km		
12	Toyota	Premio	2016	F	1.5L VVTi	Sedan	Automatic	XCV0-S45T8	SAD7-FD5TQ	Black	3310000	3400000	20000 Km		
14	Toyota	Allion	2016	G	1.5L VVTi	Sedan	Automatic	XCV0-S45T7	SAD7-FD5TP	Black	2875000	2925000	17300 Km		
24	Mitsubishi	Pajero	2012	Sport	2.5L D4D	SUV	Manual	XCV0-S45T5	SAD7-FD5TN	Black	3998000	4150000	70000 Km		
14	Toyota	Allion	2016	G	1.5L VVTi	Sedan	Automatic	XCV0-S45T6	SAD7-FD5TR	Red Wine	3025000	3100000	12500 Km		
23	Mitsubishi	Pajero	2017	G Class	3.0L V6	SUV	Automatic	XCV0-S45T4	SAD7-FD5TM	Black	10368000	10500000	50000 Km		
22	Mitsubishi	Outlander	2017	GL	2.4L VVTi	SUV	Automatic	XCV0-S45T3	SAD7-FD5TL	Perl White	4410000	4500000	45000 Km		
17	Toyota	Land Cruiser	2018	ZX	4.5L V8	SUV	Auto-Manual	XCV0-S45T0	SAD7-FD5TJ	Perl White	39610000	40000000	20000 Km		
7	Nissan	Patrol	2017	Nismo	5.6L V8	SUV	Automatic	XCV0-S45S8	SAD7-FD5TN	Perl White	39500000	40000000	52300 Km		
8	Toyota	Land Cruiser	2012	VX	4.5L V8	SUV	Automatic	XCV0-S45S7	SAD7-FD5TH	Black	18325000	18750000	45000 Km		
12	Toyota	Premio	2016	F	1.5L VVTi	Sedan	Automatic	XCV0-S45S4	SAD7-FD5TE	Red Wine	3420000	3500000	15000 Km		
2	Honda	Vezel	2014	X Package	1.5L VVTi	SUV	Automatic	XCV0-S45S2	SAD7-FD5TC	Perl White	2735000	2800000	25000 Km		

Below the table, there are several input fields and a button:

- Name:
- Profession:
- eTIN Number:
- NID Number:
- Contact:
- Add:

Figure 7.4: Sales Page

This is the view of invoice and how exactly it appears.

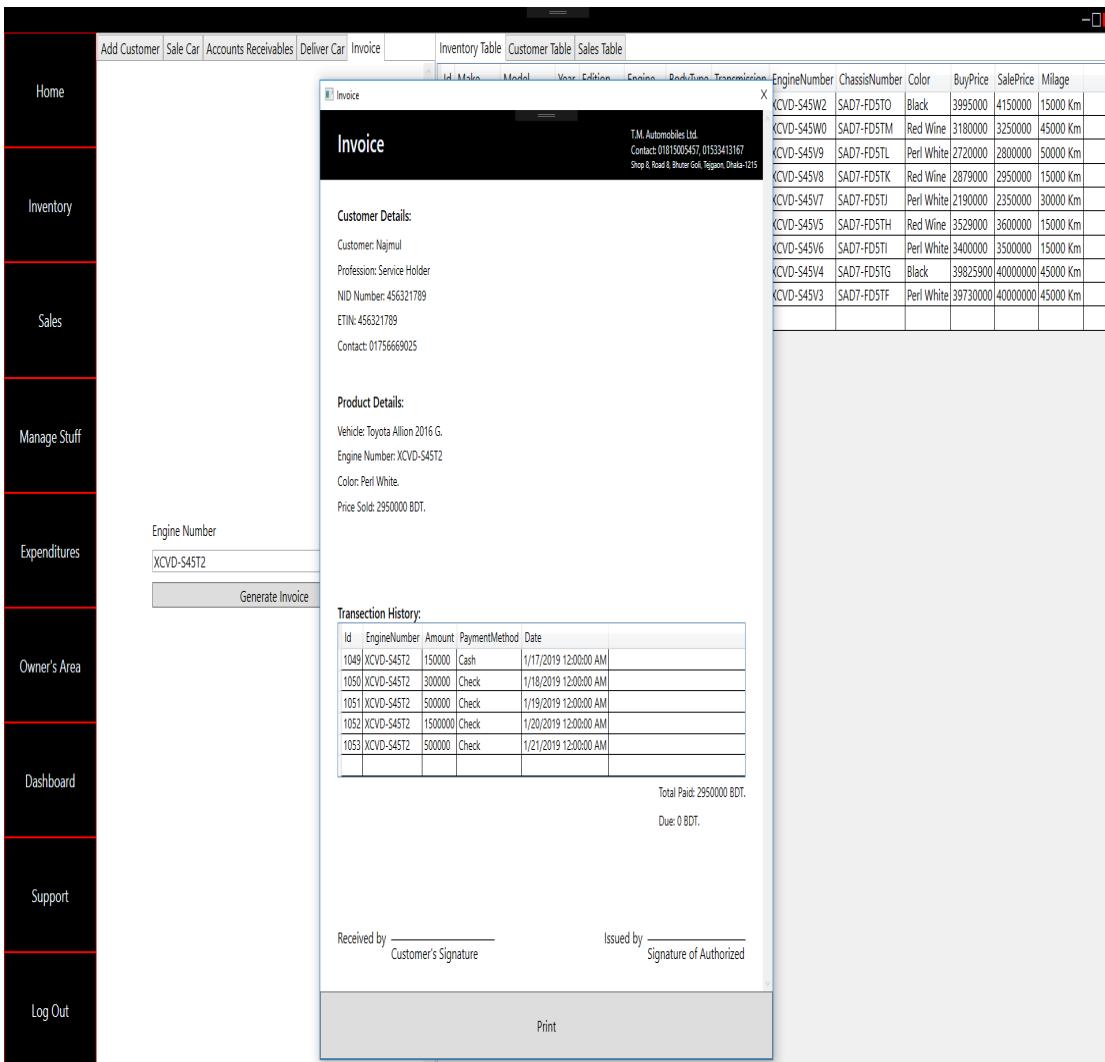


Figure 7.5: Sales' Invoice Page

7.4.1.5 Stuff Section

This is manage stuff section where user can insert newly appointed stuff information and manipulate it.

The screenshot shows a web-based application interface for managing staff. On the left, there is a vertical sidebar menu with the following items:

- Home
- Inventory
- Sales
- Manage Stuff
- Expenditures
- Owner's Area
- Dashboard
- Support
- Log Out

On the right, the main content area is divided into two sections. At the top is a table displaying staff information:

	ID	Name	Designation	Salary	NID	Address	Contact
1	Karim Benzema	Owner	200000	1234567890	Madrid, ES	1234567890	
2	Achraf Hakimi	Manager	57500	1234567890	Madrid, ES	1234567890	
3	Nabil Fekir	Asst. Manager	35600	1234567890	Paris, Fr	1234567890	
4	Alex Bappy	Mechanic	25000	1234567890	Mirpur, Dhaka	1234567889	
5	Najmul	Driver	20000	1234567890	Kazipara, Dhaka	1234567890	
6	Baki	Service Boy	12000	1234567890	Mirpur, Dhaka	1234567890	
7	Reza	Service Boy	12000	1234567890	Mirpur, Dhaka	1234567890	
8	Joy	Service Boy	12000	1234567890	Mirpur, Dhaka	1234567890	

Below the table is a form for adding new staff information:

Form fields (from top to bottom):

- Id:
- Name:
- Designation:
- Salary:
- NID Number:
- Address:
- Contact:
- Add:

Figure 7.6: Manage Stuff Page

7.4.1.6 Expenditures Section

This is expenditure section where user can insert all the expenditure transaction which occurs during operational execution.

The screenshot shows a web-based application interface. On the left is a vertical sidebar menu with the following items: Home, Inventory, Sales, Manage Stuff, Expenditures (which is currently selected and highlighted in blue), Owner's Area, Dashboard, Support, and Log Out. The main content area has two tabs at the top: 'Regular Expenses' and 'Salary Payout' (the latter is also blue, indicating it is selected). Below the tabs is a table with columns 'Id', 'Title', 'Amount', and 'Date'. The table contains numerous rows of expenditure data, such as 'Owners Withdrawals', 'Car Purchased', and 'Sales Share Paid', along with their respective amounts and dates.

	Regular Expenses	Salary Payout	
	Home		
63	Owners Withdrawals	35000	2/7/2019 12:00:00 AM
62	Owners Withdrawals	25000	2/4/2019 12:00:00 AM
61	Salary Expense	374100	2/3/2019 12:00:00 AM
60	Car Purchased	3349000	2/5/2019 12:00:00 AM
59	Car Purchased	39850000	2/5/2019 12:00:00 AM
58	Car Purchased	3250000	2/2/2019 12:00:00 AM
57	Sales Share Paid	10000	1/10/2019 12:00:00 AM
56	Car Purchased	2899000	1/22/2019 12:00:00 AM
55	Car Purchased	3265000	1/22/2019 12:00:00 AM
54	Car Purchased	4135000	1/21/2019 12:00:00 AM
53	Car Purchased	1705000	1/20/2019 12:00:00 AM
52	Car Purchased	3310000	1/19/2019 12:00:00 AM
51	Car Purchased	2875000	1/18/2019 12:00:00 AM
50	Car Purchased	3025000	1/17/2019 12:00:00 AM
49	Car Purchased	3998000	1/17/2019 12:00:00 AM
48	Car Purchased	10368000	1/16/2019 12:00:00 AM
47	Car Purchased	4410000	1/15/2019 12:00:00 AM
46	Owners Withdrawal	20000	1/10/2019 12:00:00 AM
45	Car Purchased	2895000	1/13/2019 12:00:00 AM
44	Car Purchased	4890000	1/14/2019 12:00:00 AM
43	Car Purchased	39610000	1/12/2019 12:00:00 AM
42	Car Purchased	39610000	1/12/2019 12:00:00 AM
41	Car Purchased	39500000	1/11/2019 12:00:00 AM
40	Car Purchased	18325000	1/10/2019 12:00:00 AM
39	Car Purchased	1678000	1/10/2019 12:00:00 AM
38	Car Purchased	3400000	1/9/2019 12:00:00 AM
37	Sales Share Paid	15100	1/10/2019 12:00:00 AM
36	Car Purchased	3420000	1/8/2019 12:00:00 AM
35	Car Purchased	3098500	1/7/2019 12:00:00 AM
34	Car Purchased	2735000	1/5/2019 12:00:00 AM
33	Car Purchased	2898500	1/3/2019 12:00:00 AM
32	Owners Widrawal	50000	1/6/2019 12:00:00 AM
31	Salary Expense	374100	1/2/2019 12:00:00 AM

Figure 7.7: Expenditures Page

7.4.1.7 Owner Area Section

This is owner area section where owner can manage all the data stored in database.

The screenshot shows a web-based application interface titled "Owner Area". On the left side, there is a vertical sidebar menu with the following items: Home, Inventory, Sales, Manage Stuff, Expenditures, Owner's Area (which is currently selected), Dashboard, Support, and Log Out. The main content area is divided into two sections. At the top, there is a navigation bar with links: Add Make, Add Model, Add User, Accounts, Income Statement, Make, Model, Users, Customers, Expense, Accounts Receivables, and Sales Record. Below this, there is a table with columns: ID, Name, and Country. The table contains the following data:

ID	Name	Country	Action
1	Toyota	Japan	Edit Remove
2	Nissan	Japan	Edit Remove
3	Mitsubishi	Japan	Edit Remove
4	Honda	Japan	Edit Remove
5	Hyundai	Korea	Edit Remove
7	KIA	Korea	Edit Remove
11	Ford	USA	Edit Remove
			Edit Remove

Below the table, there is a form with fields for "Make" and "Country", and a "Add" button.

Figure 7.8: Owner Area Page

This is owner area section where owner can get and print income statement.

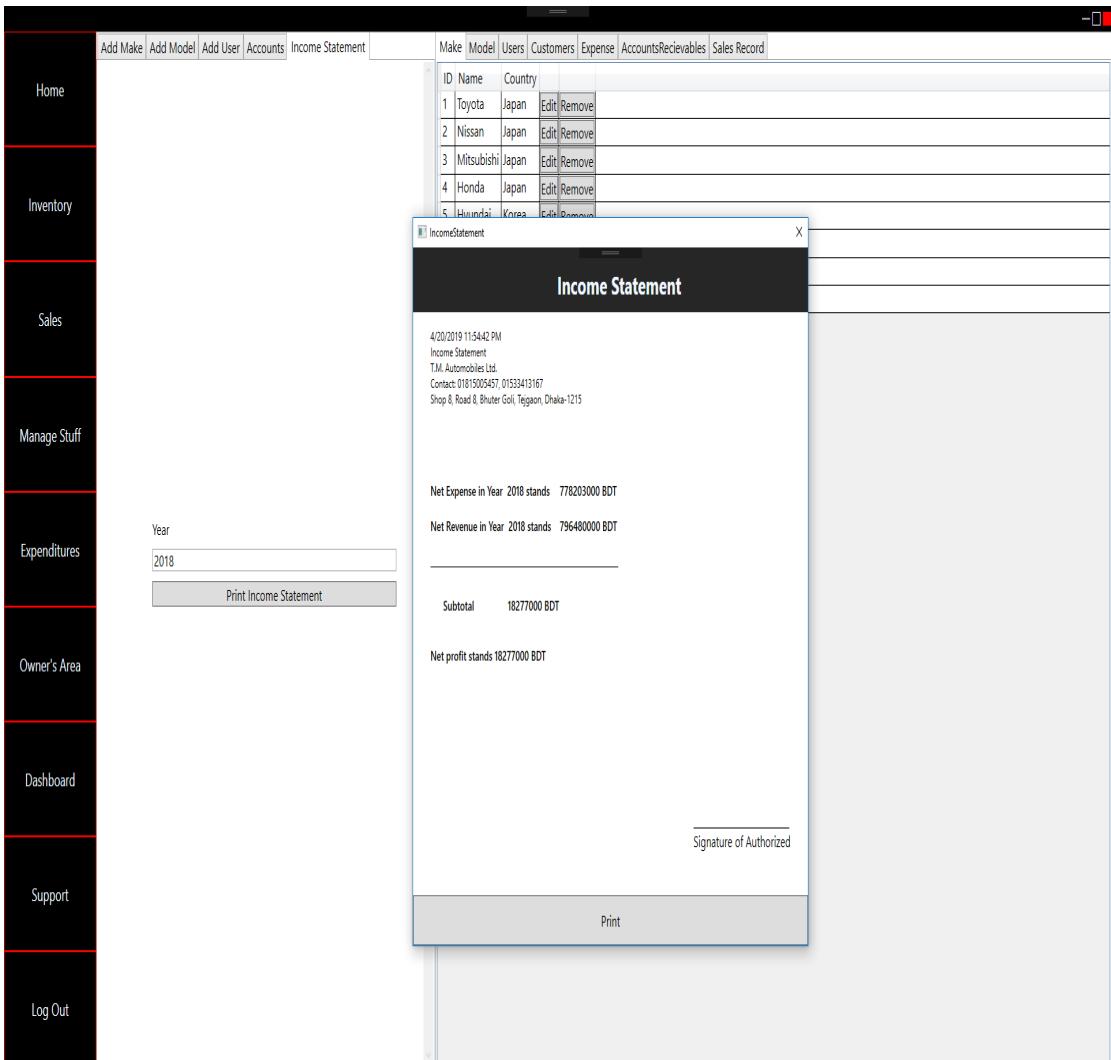


Figure 7.9: Income Statement Page

7.4.1.8 Dashboard Section

This is dashboard section where owner can get performance based analysis which makes easier to take effective decision faster for him.



Figure 7.10: Dashboard Page

7.4.1.9 Support Section

This is support page from where system user can instant notify runtime issues to developer via email.

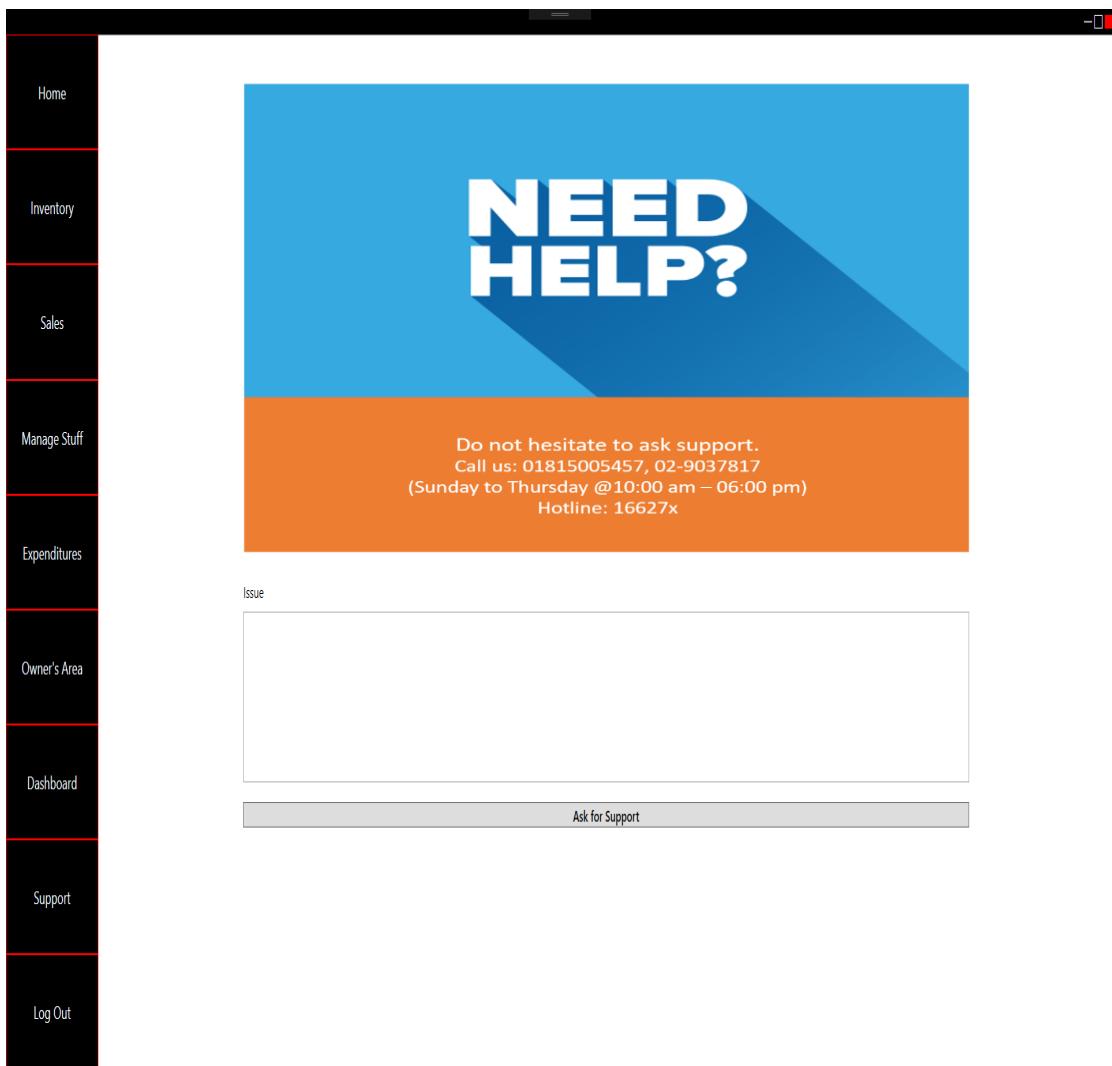


Figure 7.11: Support Page

7.4.2 Web Site

7.4.2.1 Home Page

This is home page of website. Any visitor can access this. Static information displayed here.

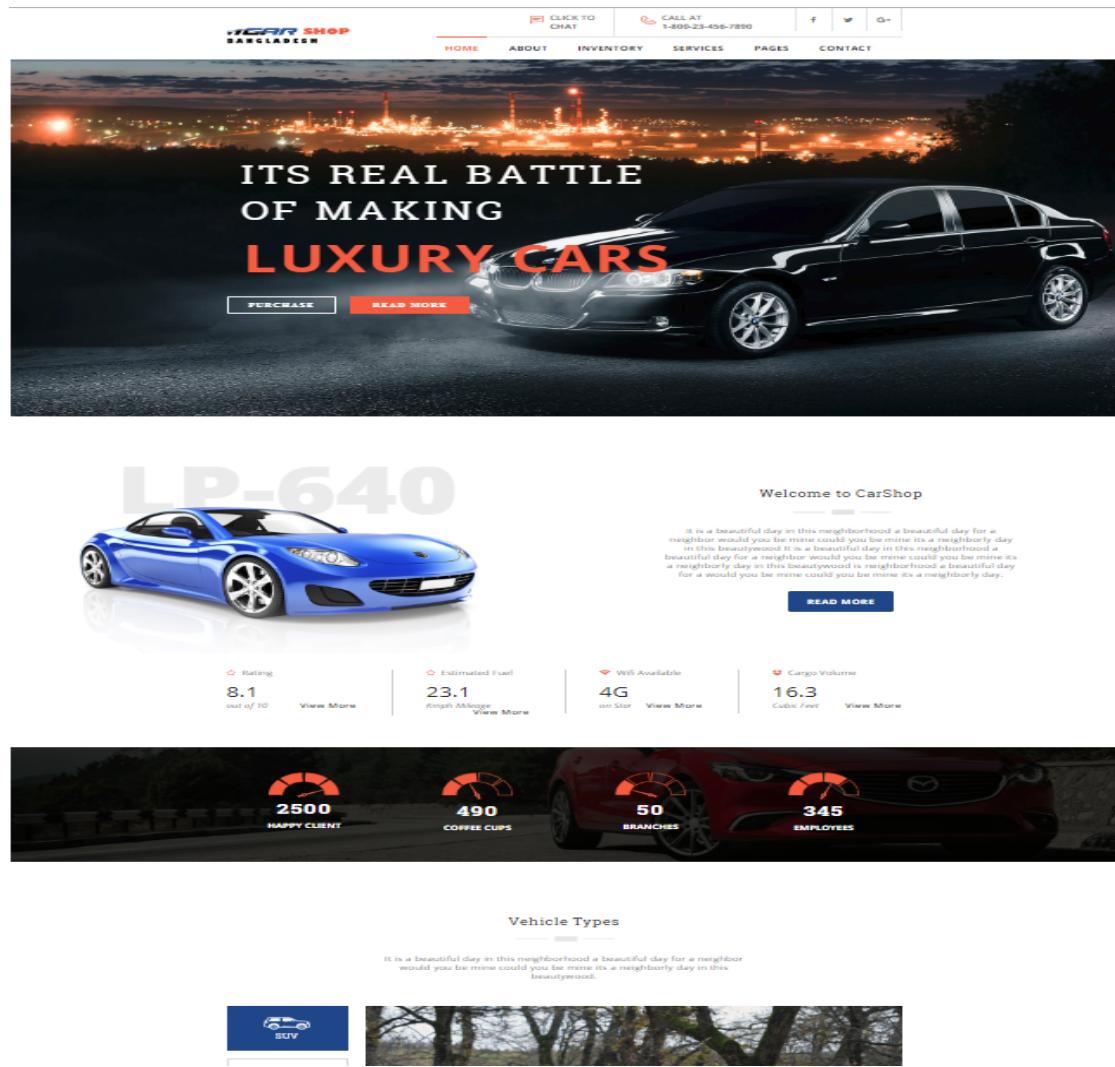


Figure 7.12: Home Page

7.4.2.2 About Page

This is about page of website. Any visitor can access this. Static information displayed here.

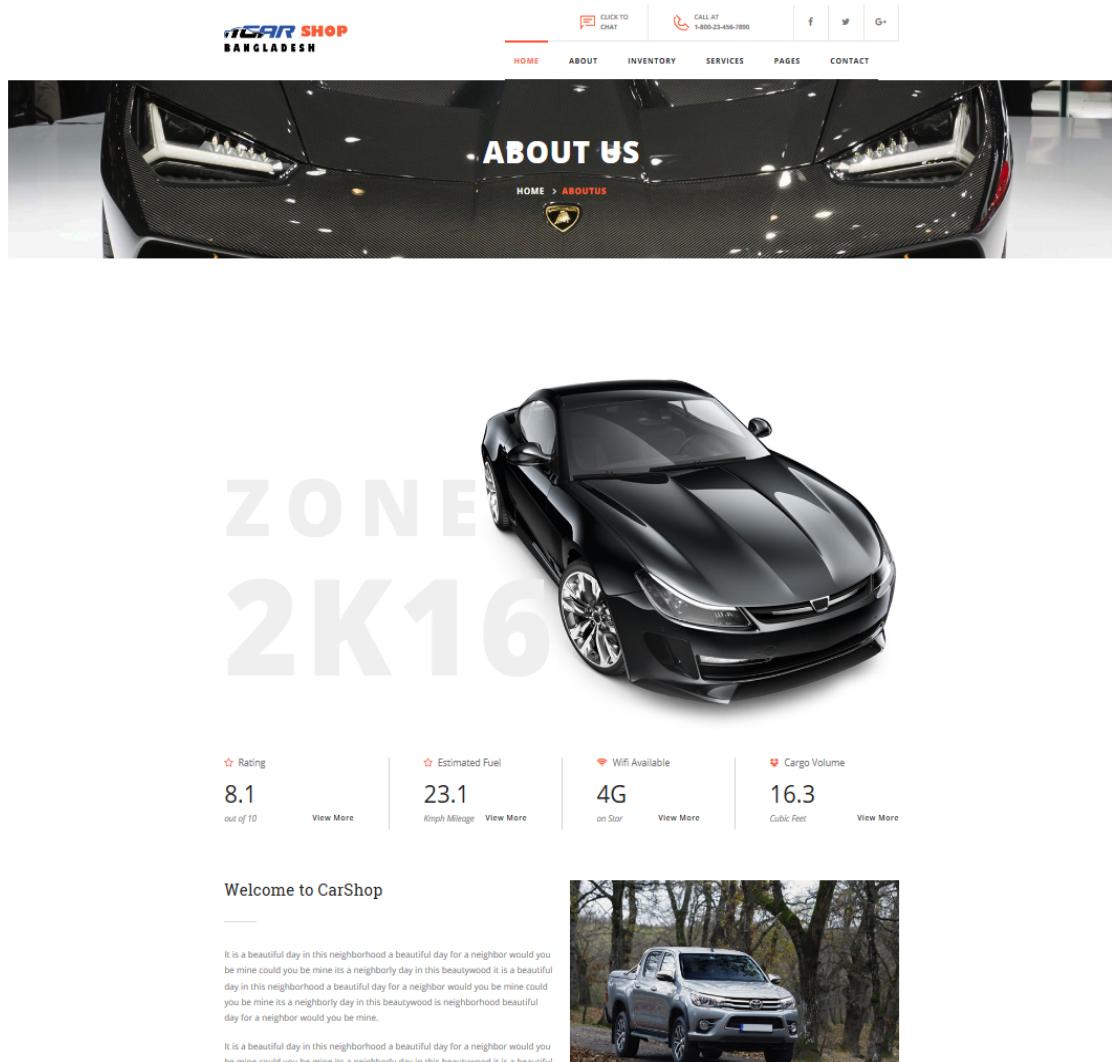


Figure 7.13: About Page

7.4.2.3 Inventory Page

This is inventory page of website. Any visitor can access this. Vehicle information displayed here which retrieved from shop database.

The screenshot shows the inventory page of the iCar Shop Bangladesh website. At the top, there's a navigation bar with links for 'HOME', 'ABOUT', 'INVENTORY' (which is the active page), 'SERVICES', 'PAGES', and 'CONTACT'. There are also social media icons for Facebook, Twitter, and Google+. A banner at the top features a close-up of a car's front grille and headlights. Below the banner, the word 'INVENTORY' is prominently displayed, followed by a breadcrumb trail 'HOME > INVENTORY' and the Lamborghini logo. The main content area displays several car models with their details:

- TOYOTA HI LUX**
2018 | AUTOMATIC
START FROM **BDT 4150000**
[BOOK NOW](#)
- HONDA VEZEL**
2015 | AUTOMATIC
START FROM **BDT 3250000**
[BOOK NOW](#)
- NISSAN**
DRIVE THE
**MOST EXCITING
LINEUP
ON THE ROAD.**
THE 2016 NISSAN ROGUE®
[SHOP NOW](#)
- HONDA VEZEL**
2015 | AUTOMATIC
START FROM **BDT 2800000**
[BOOK NOW](#)
- TOYOTA ALLION**
2016 | AUTOMATIC
START FROM **BDT 2950000**
[BOOK NOW](#)
- MITSUBISHI LANCER**
[BOOK NOW](#)
- NISSAN ALMERA**
[BOOK NOW](#)

Figure 7.14: Inventory Page

7.4.2.4 Inventory Details Page

This is inventory details page of website. Any visitor can access this. Vehicle information displayed here which retrieved from shop database.

The screenshot shows a web page for a car inventory. At the top, there's a navigation bar with the logo 'CAR SHOP BANGLADESH' on the left, and links for 'CLICK TO CHAT', 'CALL AI', '1-800-23-456-7890', and social media icons for Facebook, Twitter, and Google+ on the right. Below the navigation is a large image of a dark-colored Honda Vezel. Overlaid on the image is the word 'INVENTORY' in large white letters, and below it, 'HOME / INVENTORY SINGLE' and the Lamborghini logo. To the right of the car image, there's a section titled 'Specifications' with the following details:

Year:	2015
Make:	Honda
Model:	Vezel
Price:	2800000 BDT
Edition:	X Package
Transmission:	Automatic
Displacement:	1.5L VVTI

Below the specifications is a 'Loan Estimator' section with input fields for 'Loan Amount', 'Loan Terms in Year', and 'Rate of Interest', followed by a 'GET RESULT' button. Further down, there's a section for the 'HONDA VEZEL' with a 'START FROM BDT 2800000' price and a 'VIEW DETAILS >' link. At the bottom, there's a 'SPECIFICATIONS' section containing placeholder text: 'Lorem ipsum dolor sit amet, his epicurei disputationi ea, quem timeam ei cum. Pri te minim graeco assentior. Vocent assueverit ne eos, illum volumus te his. Ad duo cause aliquam.'

Figure 7.15: Inventory Details Page

7.4.2.5 Services Page

This is services page of website. Any visitor can access this. Recent vehicle information displayed here which retrieved from shop database.

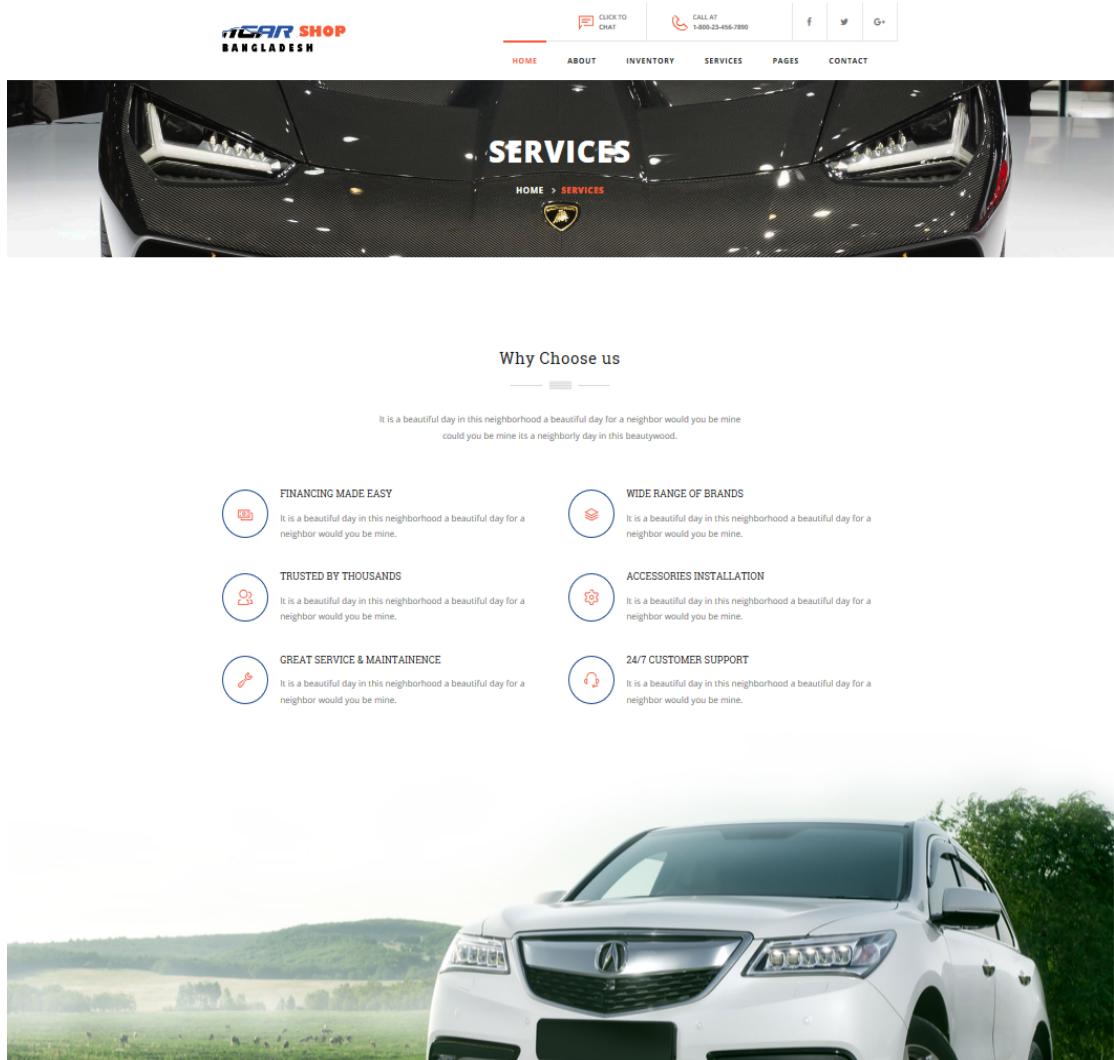


Figure 7.16: Services Page

7.4.2.6 Contact Page

This is contact page of website. Any visitor can access this and eligible to ask for any inquiry.

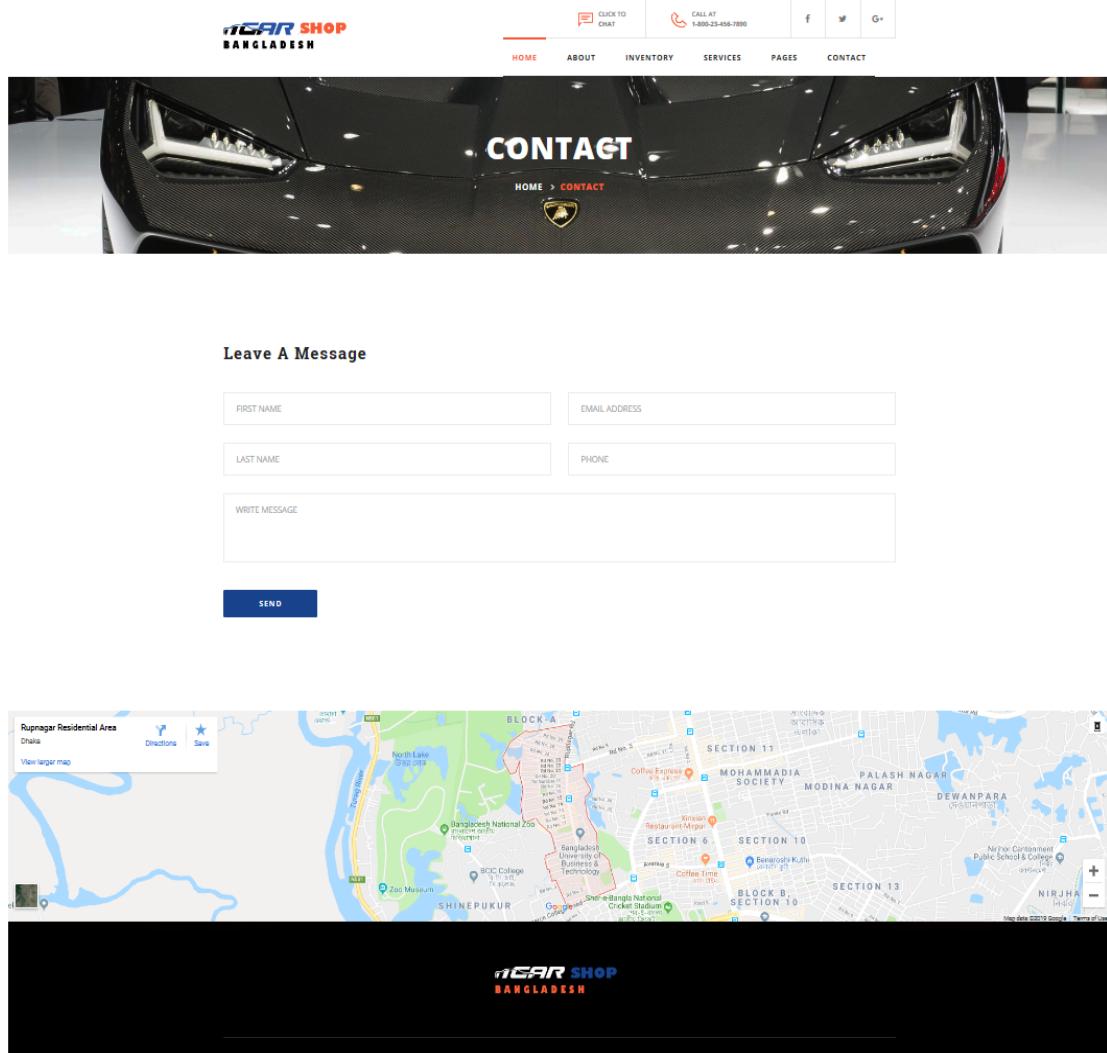


Figure 7.17: Contact Page

7.5 Conclusions

A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system.[1] It is usually written by a technical writer, although user guides are written by programmers, product or project managers, or other technical staff, particularly in smaller companies.[2]

User guides are most commonly associated with electronic goods, computer hardware and software.[3] This represents the Company records for which ERP-Next is setup. With this same setup, you can create multiple Company records, each representing a different legal entity. The accounting for each Company will be different, but they will share the Customer, Supplier and Item records. Now, any user know about this application properly. They use it now easily. This chapter introduced over all interface of our application.

Chapter 8

CONCLUSIONS

8.1 Conclusions

Car Trading Information Management System is a complete Enterprise Resource Planning(ERP) solution, developed according to the common trade pattern of car traders. It is a small sized business with a huge cash flow and the targeted audiences are actually car traders and their internal operational managers. The key deliverables of the solution is a desktop application which allows users to perform internal operation like managing inventory, performing sales etc. The web application is built for visitors in spite of reaching inventory products to possibly maximum peoples. The main challenge was to develop the system as budget friendly as possible. To make this happen, we investigate and analyze the system more and more, finding optimistic solutions to each problem, defining all the functional and non functional requirements as effective as possible. Now, we are confidant about our development. The solution is feasible to car traders. All the features added are completely based on their operational activities and there is no extra or unnecessary features we add, and also we did not exclude any features which seems required to operate their business. The solution does not requires extra resources to traders. Thus, today is the end of the third industrial revolution and the fourth industrial revolution is knocking the door. By considering this also, the architectural design of the solution performed and now we are confidant to claim that the solution we developed is highly durable and it is valid not less than upcoming 25 years.

Our goal is to provide a system and service to you. To do this, were here to provide you a desktop application which will use to manage your business issues. A web

application which will represent and highlight your business to customers and give them so many opportunities continuously. Were trying to bring the breakpoint of this traditional system of car trading and take it into digital form. We never compromise with quality and were here to be the witness of your satisfaction.

8.2 Future Enhancement

The time we are living now is actually the end of the third industrial revolution and the fourth industrial revolution is knocking the door[40]. We stand on the brink of a technological revolution that will fundamentally alter the way we live, work, and relate to one another. In its scale, scope, and complexity, the transformation will be unlike anything humankind has experienced before. We do not yet know just how it will unfold, but one thing is clear: the response to it must be integrated and comprehensive, involving all stakeholders of the global polity, from the public and private sectors to academia and civil society [73].

To survive the solution, updates and modification of the system is highly required. From our initial version of the solution, we also maintain these facts to develop this system. Most of the designs just updated, not completely designed in this project because the design of the solution is highly maintainable and reusable and we do not have to face unusual hassles to any kind of major or minor updates. By following current technological revolution, we have several plans about major updates of the system. Let's have a look on them.

8.2.1 Smart Phone App

In this version of solution, we enables the richest way to develop apps for smart phone so that the gap between trader and consumer remains lesser. We already created web API(Application Program Interface) to build apps in future. You may assume that we are heading for this in near future according to our present activities. Now a days, smart phone has almost every level of people and day by day people becomes more depended onto this. Currently we became confused about its feasibility but there is a big chance that it will happen in upcoming update.

8.2.2 Data Security

Currently in this version, business data stands much secured. But it can not be assumed in this technological revolutionary age that the security of the solution will remain better in upcoming age. This is why, we already concerned about it and thinking about the solution. Currently in this solution, the desktop application and the web site directly hitting database server, and we are thinking about to develop all deliverables by using web APIs of the database server, that means no direct interaction will happen between application and database server. There is also a higher probability to apply this approach in the next version of the solution.

8.2.3 Artificial Intelligence in Car Shop

Big data analysis is a popular activity in modern day business and today or tomorrow, it will also in-demand on a typical car shop. Several hardware and software solution will work together interrelatedly and receives data from each and every activity. For example, an image processing application using cctv camera installed in shop and it will study customers behaviors, speeches etc. and update information to database. From this information another application analyze customers and generate report to authorized personnel, to make effective and optimistic business decision faster and accurately. We are not sure about when we implement this in this solution, but it is clear that when it seems feasible to car traders, it will included to this ERP solution.

8.2.4 Blockchain

This is one of the most recent technology and hot topic in tech world. It seems that the use of blockchain technology is growing rapidly and will much more effective in car trading. Deals done between trader and importer or importer and exporter(from exporting country), always occurs huge amount of money. This is why transactions remains slower. In today's world, slower process occurs slower profit which directly indicates about potential losses. The use of blockchain technology in this business chain makes bigger transaction faster and much more secured and we believe that it brings more transparency in this field of trading. We assure that when the targeted audiences are eligible to adopt this technology, the solution will become blockchain based. It may be in next update or further one.

We can think, we can assume, we can prepare, according to the newest technologies appearing day by day. But practically, we can only provide those deliverables to our targeted audiences which exactly feasible to them. Actually, the future plan not depending on us, it totally depends on our targeted audiences' business needs.

References

References

- [1] The importance of the private car — mobility and transport.
https://ec.europa.eu/transport/road_safety/specialist/knowledge/old/safety_versus_mobility_and_quality_of_life/the_importance_of_the_private_car_en. (Accessed on 05/30/2019).
- [2] Car sales triple in five years — dhaka tribune.
<https://www.dhakatribune.com/business/2018/02/21/car-sales-triple-five-years>. (Accessed on 05/30/2019).
- [3] Honda starts manufacturing in bangladesh — dhaka tribune.
<https://www.dhakatribune.com/business/2018/11/12/honda-starts-manufacturing-in-bangladesh>. (Accessed on 05/31/2019).
- [4] Duty-free car import drives down revenue in fy 18.
<https://thefinancialexpress.com.bd/trade/duty-free-car-import-drives-down-revenue-in-fy-18-1552794256>. (Accessed on 05/31/2019).
- [5] Welcome to divine it limited - your reliable software partner - divine it limited.
<https://www.divineit.net/>. (Accessed on 05/31/2019).
- [6] Platform independent erp — cloud based erp in bangladesh - prismerp.
<https://www.prismerp.net/>. (Accessed on 05/31/2019).
- [7] Bdtask ltd - a leading software company in bangladesh.
<https://www.bdtask.com/>. (Accessed on 05/31/2019).
- [8] Vehicle management system software review — how it works.
<https://www.bdtask.com/vehicle-management-system.php>. (Accessed on 05/31/2019).
- [9] Car trade software trusted by many used car dealers all over the world.
<http://www.cartradesoft.com/>. (Accessed on 05/31/2019).
- [10] ghjhj - david dexters.
<https://www.daviddexters.co.uk/customer-comments/ghjhj/>. (Accessed on 05/31/2019).

- [11] Get - microsoft store. <https://www.microsoft.com/en-us/p/-/9nddkspmbqxf?activetab=pivot%3Aoverviewtab>. (Accessed on 05/31/2019).
- [12] Get numerologics - microsoft store. <https://www.microsoft.com/en-us/p/numerologics/9nblggh3zj7g?activetab=pivot%3Aoverviewtab>. (Accessed on 05/31/2019).
- [13] Demand definition. <https://www.investopedia.com/terms/d/demand.asp>. (Accessed on 05/31/2019).
- [14] Demand analysis definition — marketing dictionary — mba skool-study.learn.share. <https://www.mba-skool.com/business-concepts/marketing-and-strategy-terms/10953-demand-analysis.html>. (Accessed on 05/31/2019).
- [15] Gerald Kotonya and Ian Sommerville. *Requirements engineering: processes and techniques*. Wiley Publishing, 1998.
- [16] Key deliverables in project management: Definition & steps - video & lesson transcript — study.com. <https://study.com/academy/lesson/key-deliverables-in-project-management-definition-lesson-quiz.html>. (Accessed on 05/31/2019).
- [17] Computer programming. https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Computer_programming.html. (Accessed on 05/31/2019).
- [18] Mining an erp system to support introductory - google search. <https://www.google.com/search?q=Mining+an+ERP+System+to+Support+Introductory&oq=Mining+an+ERP+System+to+Support+Introductory&aqs=chrome..69i57.487j0j4&sourceid=chrome&ie=UTF-8>. (Accessed on 05/31/2019).
- [19] Teun Koetsier. On the prehistory of programmable machines: musical automata, looms, calculators. *Mechanism and Machine theory*, 36(5):589–603, 2001.
- [20] What is programming. retrieved 2014-03-03. - google search. <https://www.google.com/search?q=What+is+programming.+Retrieved+2014-03-03.&oq=What+is+programming.+Retrieved+2014-03-03.&aqs=chrome..69i57.399j0j9&sourceid=chrome&ie=UTF-8>. (Accessed on 05/31/2019).
- [21] Computer programming. https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Computer_programming.html. (Accessed on 05/31/2019).

- [22] Systems analysis and design for the global enterprise (book, 2007) [worldcat.org]. <https://www.worldcat.org/title/systems-analysis-and-design-for-the-global-enterprise/oclc/173876739>. (Accessed on 05/31/2019).
- [23] Jeffrey L Whitten and Lonnie D Bentley. System analysis and design for the global enterprise, 2007.
- [24] Inventory definition - what is inventory. <https://www.shopify.com/encyclopedia/inventory>. (Accessed on 05/31/2019).
- [25] What is a human resource? (definition and resources). <https://www.thebalancecareers.com/what-is-a-human-resource-1918144>. (Accessed on 05/31/2019).
- [26] What is a human resource? (definition and resources). <https://www.thebalancecareers.com/what-is-a-human-resource-1918144>. (Accessed on 05/31/2019).
- [27] What is accounting process? definition and meaning - business-dictionary.com. <http://www.businessdictionary.com/definition/accounting-process.html>. (Accessed on 05/31/2019).
- [28] What is customer relationship? definition and meaning - business-dictionary.com. <http://www.businessdictionary.com/definition/customer-relationship.html>. (Accessed on 05/31/2019).
- [29] Requirements analysis - requirements analysis process,techniques. <https://reqtest.com/requirements-blog/requirements-analysis/>. (Accessed on 05/31/2019).
- [30] What is a feasibility study? — ag decision maker. <https://www.extension.iastate.edu/agdm/wholefarm/html/c5-65.html>. (Accessed on 05/31/2019).
- [31] What is feasibility study? definition and meaning - business-dictionary.com. <http://www.businessdictionary.com/definition/feasibility-study.html>. (Accessed on 05/31/2019).
- [32] Chapter 4 - developing phase — microsoft docs. [https://docs.microsoft.com/en-us/previous-versions/tn-archive/bb497041\(v=technet.10\)#EHA](https://docs.microsoft.com/en-us/previous-versions/tn-archive/bb497041(v=technet.10)#EHA). (Accessed on 05/31/2019).
- [33] My library. <https://books.google.com.bd/books?uid=107222170869881241535&hl=en>. (Accessed on 05/31/2019).
- [34] My library. <https://books.google.com.bd/books?uid=107222170869881241535&hl=en>. (Accessed on 05/31/2019).

- [35] Literature review - wikipedia. https://en.wikipedia.org/wiki/Literature_review. (Accessed on 05/31/2019).
- [36] Why a feasibility study is important in project management. <https://www.simplilearn.com/feasibility-study-article>. (Accessed on 05/31/2019).
- [37] Business requirements analysis - project management from mind-tools.com. https://www.mindtools.com/pages/article/newPPM_77.htm. (Accessed on 05/31/2019).
- [38] Filebank answers your records storage and management questions. <https://www.filebank.ca/records-storage-management-faqs/>. (Accessed on 05/31/2019).
- [39] Software requirements analysis with example. <https://www.guru99.com/learn-software-requirements-analysis-with-case-study.html>. (Accessed on 05/31/2019).
- [40] The 4th industrial revolution is here - are you ready? <https://www.forbes.com/sites/bernardmarr/2018/08/13/the-4th-industrial-revolution-is-here-are-you-ready/#6483cdb8628b>. (Accessed on 05/31/2019).
- [41] Data flow diagram - everything you need to know about dfd. <https://www.smartdraw.com/data-flow-diagram/>. (Accessed on 05/31/2019).
- [42] Data flow diagram symbols - create data flow diagrams rapidly with free data flow chart software. <https://www.edrawsoft.com/Data-Flow-Diagram-Symbols.php>. (Accessed on 05/31/2019).
- [43] Use case - wikipedia. https://en.wikipedia.org/wiki/Use_case. (Accessed on 05/31/2019).
- [44] Use case diagram notations guide - visual paradigm. <https://circle.visual-paradigm.com/docs/uml-and-sysml/use-case-diagram/use-case-diagram-notations-guide/>. (Accessed on 05/31/2019).
- [45] Uml - activity diagrams. https://www.tutorialspoint.com/uml/uml_activity_diagram.htm. (Accessed on 05/31/2019).
- [46] Uml activity diagram tutorial — lucidchart. <https://www.lucidchart.com/pages/uml-activity-diagram>. (Accessed on 05/31/2019).
- [47] Sequence diagram - wikipedia. https://en.wikipedia.org/wiki/Sequence_diagram. (Accessed on 05/31/2019).
- [48] Sequence diagrams - what is a sequence diagram? <https://www.smartdraw.com/sequence-diagram/>. (Accessed on 05/31/2019).

- [49] Uml - class diagram. https://www.tutorialspoint.com/uml/uml_class_diagram.htm. (Accessed on 05/31/2019).
- [50] Class diagrams - learn everything about class diagrams. <https://www.smartdraw.com/class-diagram/>. (Accessed on 05/31/2019).
- [51] Uml - statechart diagrams. https://www.tutorialspoint.com/uml/uml_statechart_diagram.htm. (Accessed on 05/31/2019).
- [52] State diagrams - everything to know about state charts. <https://www.smartdraw.com/state-diagram/>. (Accessed on 05/31/2019).
- [53] Literature review - wikipedia. https://en.wikipedia.org/wiki/Literature_review. (Accessed on 05/31/2019).
- [54] Systems analysis and design : Computer science 361 : Courses : Athabasca university. <https://www.athabascau.ca/syllabi/comp/comp361.php>. (Accessed on 05/31/2019).
- [55] System analysis and design overview. https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm. (Accessed on 05/31/2019).
- [56] Systems analysis and design - wikipedia. https://en.wikipedia.org/wiki/Systems_analysis_and_design. (Accessed on 05/31/2019).
- [57] Systems analysis and design - wikipedia. https://en.wikipedia.org/wiki/Systems_analysis_and_design. (Accessed on 05/31/2019).
- [58] Feasibility study - wikipedia. https://en.wikipedia.org/wiki/Feasibility_study. (Accessed on 05/31/2019).
- [59] Feasibility study. <https://www.investopedia.com/terms/f/feasibility-study.asp>. (Accessed on 05/31/2019).
- [60] What is requirements analysis (requirements engineering) ? - definition from whatis.com. <https://searchsoftwarequality.techtarget.com/definition/requirements-analysis>. (Accessed on 05/31/2019).
- [61] Methodology - wikipedia. <https://en.wikipedia.org/wiki/Methodology>. (Accessed on 05/31/2019).
- [62] Amazon.com: Systems analysis and design, 9 ed (9789332547988): Kendall: Books. https://www.amazon.com/Systems-Analysis-Design-9-Ed-dp-933254798X/dp/933254798X/ref=mt_paperback?_encoding=UTF8&me=&qid= (Accessed on 05/31/2019).

- [63] Object-oriented analysis and design: What is it and how do you use it? <https://airbrake.io/blog/design-patterns/object-oriented-analysis-and-design>. (Accessed on 05/31/2019).
- [64] What is a data flow diagram — lucidchart. <https://www.lucidchart.com/pages/data-flow-diagram>. (Accessed on 05/31/2019).
- [65] What is a data flow diagram — lucidchart. <https://www.lucidchart.com/pages/data-flow-diagram>. (Accessed on 05/31/2019).
- [66] What is schema? - definition from techopedia. <https://www.techopedia.com/definition/1242/schema>. (Accessed on 05/31/2019).
- [67] White box testing - software testing fundamentals. <http://softwaretestingfundamentals.com/white-box-testing/>. (Accessed on 05/31/2019).
- [68] What is component testing or module testing (learn with examples). <https://www.softwaretestinghelp.com/what-is-component-testing-or-module-testing/>. (Accessed on 05/31/2019).
- [69] Integration testing - software testing fundamentals. <http://softwaretestingfundamentals.com/integration-testing/>. (Accessed on 05/31/2019).
- [70] System testing - software testing fundamentals. <http://softwaretestingfundamentals.com/system-testing/>. (Accessed on 05/31/2019).
- [71] Black box testing - software testing fundamentals. <http://softwaretestingfundamentals.com/black-box-testing/>. (Accessed on 05/31/2019).
- [72] What is a data flow diagram — lucidchart. <https://www.lucidchart.com/pages/data-flow-diagram>. (Accessed on 05/31/2019).
- [73] The fourth industrial revolution: what it means and how to respond — world economic forum. <https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>. (Accessed on 06/01/2019).

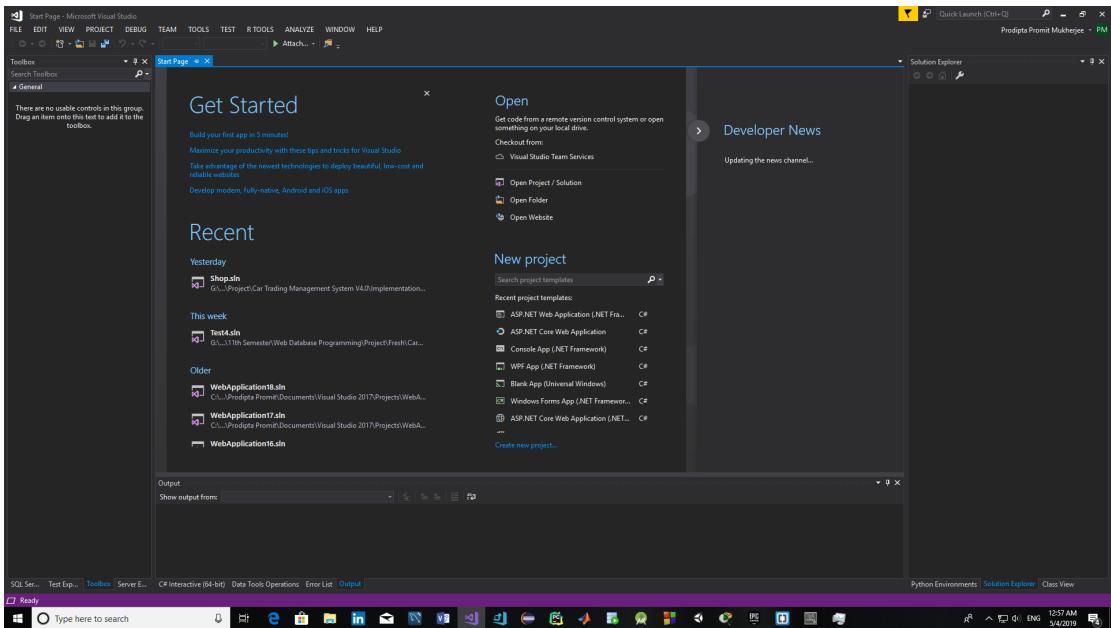
Appendices

Tools

- Microsoft Visual Studio 2017
- JetBrains Resharper
- SQL Server Management Studio
- Microsoft Blend for Visual Studio 2017
- TexWorks
- Adobe Illustrator
- Microsoft Visio 2013

Appendix A

Microsoft Visual Studio 2017



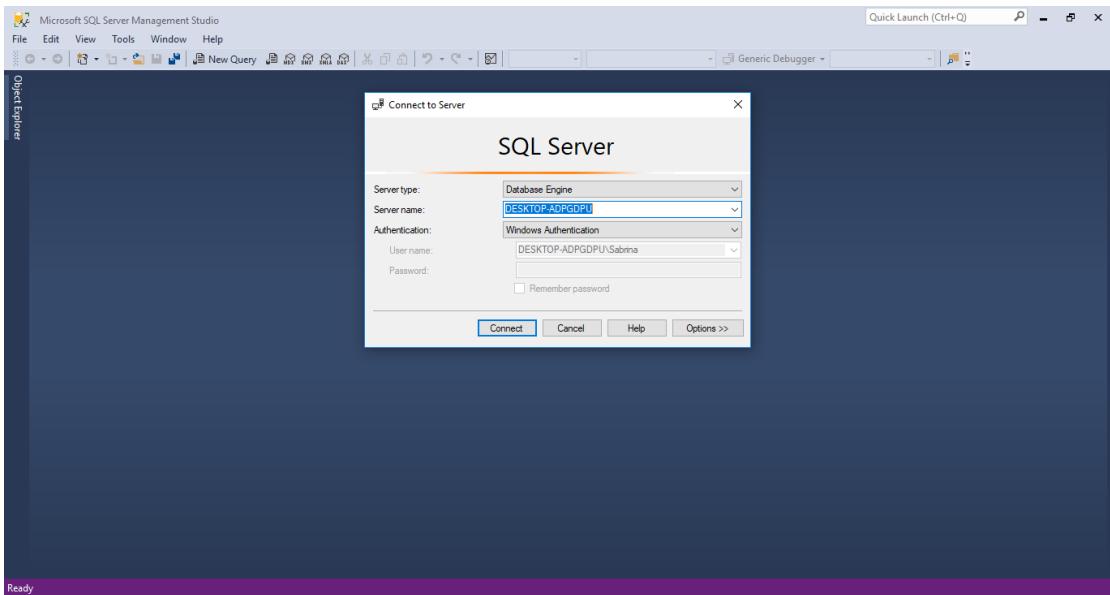
Appendix B

JetBrains Resharper



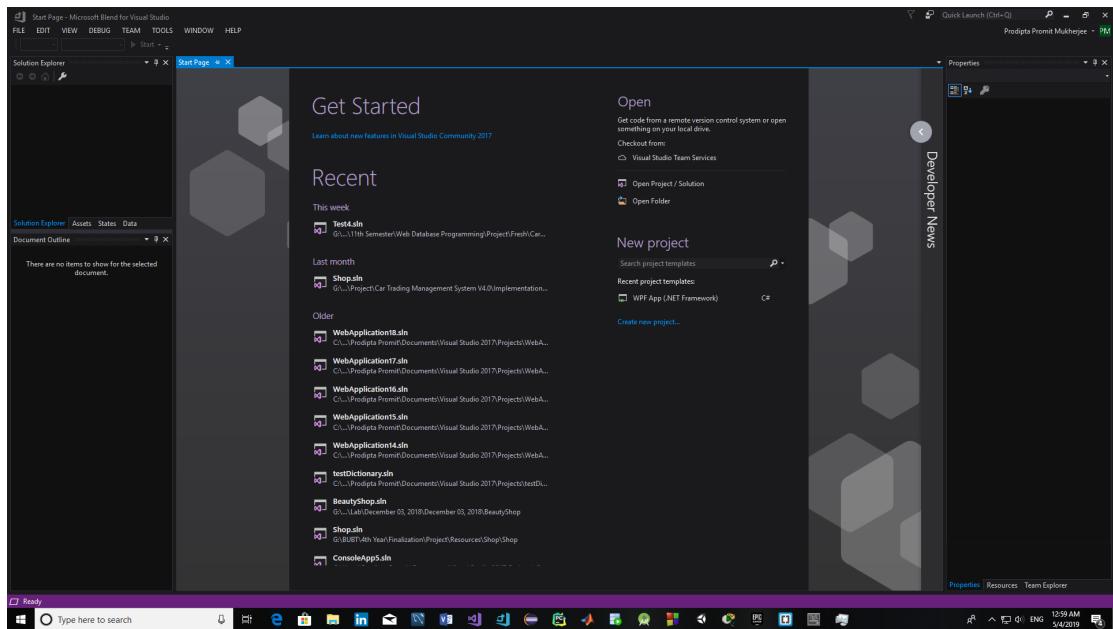
Appendix C

SQL Server Management Studio



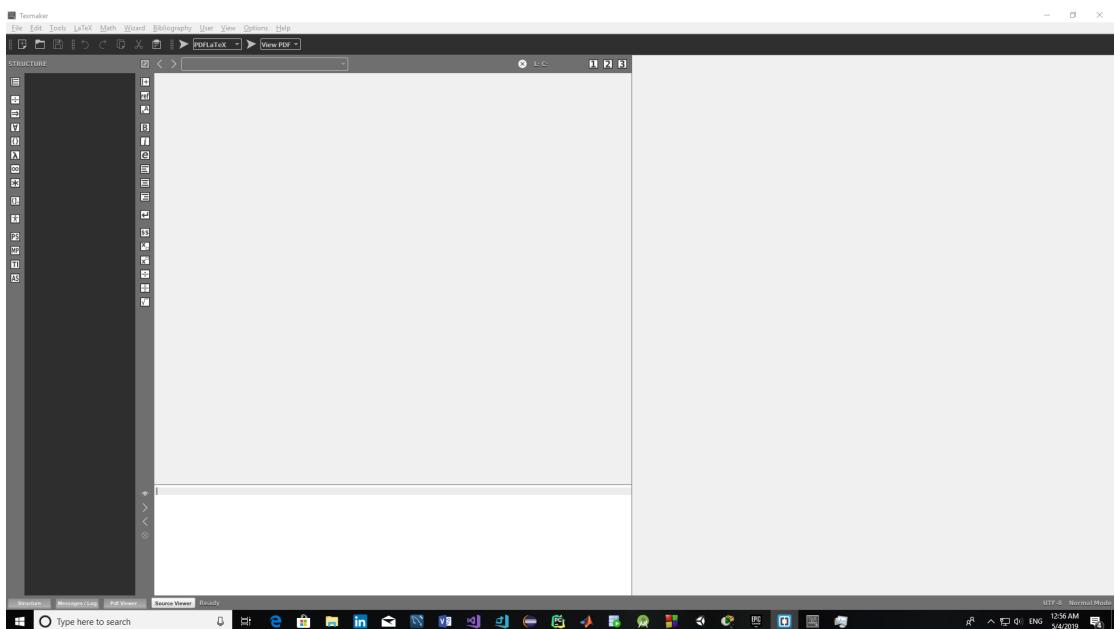
Appendix D

Microsoft Blend for Visual Studio 2017



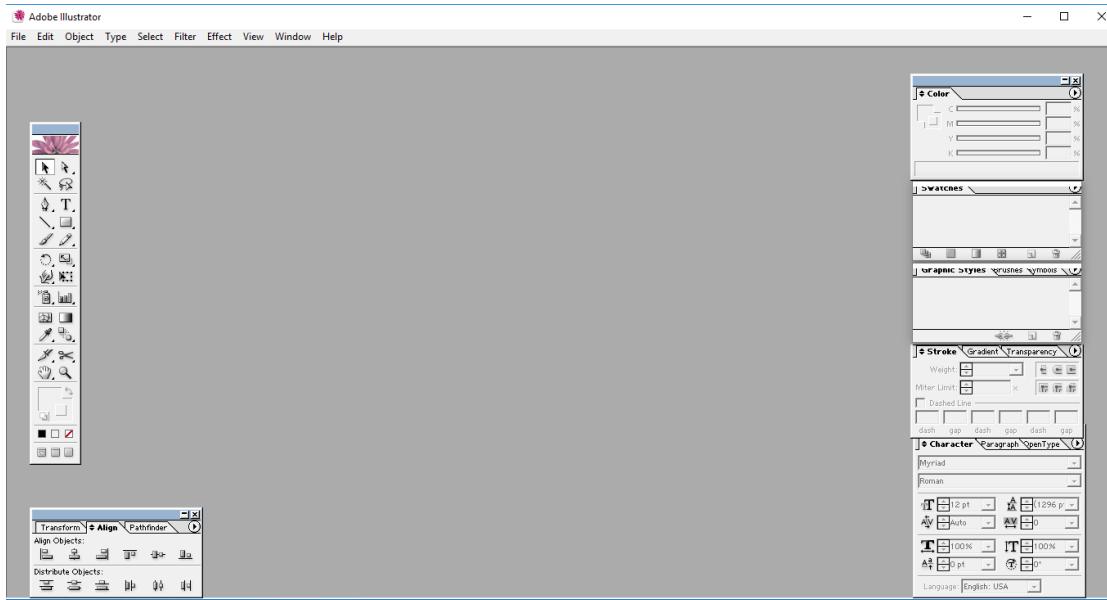
Appendix E

TexWorks



Appendix F

Adobe Illustrator



Appendix G

Microsoft Visio 2013

