

**WOLAITA SODO UNIVERSITY**

College of Natural and Computational Science

Department of Computer science and Information Technology

Industrial project

Title: - online vehicle rental system

Prepared by: Group 13

NAME ID No

Ermias Yilma …………………………NSC/R/431/04

Dadhi Fitala …………………………...NSC/R/332/04

Demisew Desalegn ………………….NSC/R/355/04

Advisors: -Hailyie Teklesilassie &

Letebrhan Alemayehu

**Submitted to:**

The department of ***Computer Science and***

***Information Technology,***

Submission date: June 10/2015

Wolaita Sodo Ethiopia

**ONLINE VEHICLE RENTAL SYSTEM**

By. Group 13 Students

Advisor Name1: Letebrhan Alemayehu

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Advisor Name2: Hailyie Teklesilassie

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examiner Name1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examiner Name2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ACKNOWLEDGMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of GOD and many other individuals. We would like to extend our sincere thanks to GOD and all of others.

We are highly indebted to our department teachers especially to our advisors (Mrs Letebrhan Alemayehu and Mr Hailyie Teklesilasie*)* for their guidance and constant supervision as well as for providing necessary information regarding to our project.

*Members of group 13 students!*

Table of Contents

[1. Background of the project: 1](#_Toc421676832)

[1.1 Motivation 1](#_Toc421676833)

[1.2 Mission and vision of organization 1](#_Toc421676834)

[Vision 1](#_Toc421676835)

[Mission 1](#_Toc421676836)

[2. The existing system 1](#_Toc421676837)

[3. Proposed project 2](#_Toc421676838)

[4. Statement of the Problem 2](#_Toc421676839)

[5. Team composition 2](#_Toc421676840)

[6. Objective 4](#_Toc421676841)

[6.1 General objective 4](#_Toc421676842)

[6.2 Specific objective 4](#_Toc421676843)

[7. Scope of the project 4](#_Toc421676844)

[8. Significance of the project 4](#_Toc421676845)

[9. Feasibility of the project 5](#_Toc421676846)

[9.1 Technical feasibility analyses: 5](#_Toc421676847)

[9.2 Operational Feasibility 5](#_Toc421676848)

[9.3 Economic feasibility 5](#_Toc421676849)

[10. Methodology for the Project 6](#_Toc421676850)

[Data collection/ fact finding 6](#_Toc421676851)

[Observation 6](#_Toc421676852)

[Interview 6](#_Toc421676853)

[Analysis and design methodology approach 6](#_Toc421676854)

[Development tools 6](#_Toc421676855)

[Hardware Requirements 6](#_Toc421676856)

[Software Requirements 6](#_Toc421676857)

[11. Estimated budget 7](#_Toc421676858)

[12. Tasks and Time schedule. 7](#_Toc421676859)

[13. Limitation of the project 8](#_Toc421676860)

[14. Risk, assumption and constraint 8](#_Toc421676861)

[Chapter 2 10](#_Toc421676862)

[2 Description of the Existing System 10](#_Toc421676863)

[2.1 introductions to the existing system 10](#_Toc421676864)

[2.2 Players in the existing system 10](#_Toc421676865)

[2.3 Major functions/activities of the existing system 11](#_Toc421676866)

[2.4 Business rule 11](#_Toc421676867)

[2.5 Report generated in the existing system 12](#_Toc421676868)

[2.6 Forms and other documents of the existing system 12](#_Toc421676869)

[2.7 Bottle necks of the existing system 12](#_Toc421676870)

[2.7.1 Performance (response time): 12](#_Toc421676871)

[2.7.2 Input: 12](#_Toc421676872)

[2.7.3 Output 13](#_Toc421676873)

[2.7.4 Security and control 13](#_Toc421676874)

[2.7.5Efficiency 13](#_Toc421676875)

[2.8 Practiced to be preserved 13](#_Toc421676876)

[2.9 Proposed solution for the new system that address problems of the existing system 14](#_Toc421676877)

[2.10 Requirements of the proposed system 14](#_Toc421676878)

[2.10.1 Functional requirement 14](#_Toc421676879)

[2.10.2 Nonfunctional requirement 15](#_Toc421676880)

[Chapter 3 17](#_Toc421676881)

[3. System analysis (modeling of the existing and proposed system) 17](#_Toc421676882)

[3.1 Introduction 17](#_Toc421676883)

[3.2 System Requirement Specification (SRS) 17](#_Toc421676884)

[3.2.1 Use case diagrams 18](#_Toc421676885)

[3.2.2 Use case documentation 19](#_Toc421676886)

[3.2.3 Sequence Diagram 26](#_Toc421676887)

[3.2.4 Activity Diagram 32](#_Toc421676888)

[3.2.5 Analysis level class diagrams (conceptual modeling) 36](#_Toc421676889)

[3.2.6 User Interface Prototyping 37](#_Toc421676890)

[3.2.7 Supplementary specification 38](#_Toc421676891)

[Chapter 4 39](#_Toc421676892)

[System Design 39](#_Toc421676893)

[4.1 Introduction 39](#_Toc421676894)

[4.2 Class Type Architecture 40](#_Toc421676895)

[Class type architecture documentation 41](#_Toc421676896)

[4.3 Class Modeling 41](#_Toc421676897)

[4.4 State chart modeling 42](#_Toc421676898)

[4.5 Collaboration modeling 43](#_Toc421676899)

[4.6 Component Modeling 44](#_Toc421676900)

[4.7 Deployment Modeling 45](#_Toc421676901)

[4.8 User Interface Design 47](#_Toc421676902)

[CHAPTER 5 49](#_Toc421676903)

[Implementation 49](#_Toc421676904)

[5.1 Introduction 49](#_Toc421676905)

[5.2 Hardware software acquisitions 49](#_Toc421676906)

[5.2 User manual preparation 49](#_Toc421676907)

[5.3 Training 50](#_Toc421676908)

[5.4 Installation Process 50](#_Toc421676909)

[Chapter Six 70](#_Toc421676910)

[6. Conclusion and Recommendation 70](#_Toc421676911)

[6.1 conclusions 70](#_Toc421676912)

[6.2 Recommendation 71](#_Toc421676913)

[References 73](#_Toc421676914)

List of tables and figures

[Table1.1 team composition table 3](#_Toc410793685)

[Table 1.2 Table to show hardware cost 7](#_Toc410793686)

[Table 1.3 Timeline 8](#_Toc410793687)

[Table 1.4 risk assumption 9](#_Toc410793688)

[Fig 3.1 use case diagram of the system. 21](#_Toc410793689)

[Table 1: Use Case - Login 22](#_Toc410793690)

[Table 2: Use Case - Reserve Vehicle 22](#_Toc410793691)

[Table 3: Use Case - Rent Registration 23](#_Toc410793692)

[Table 4: Use Case - Vehicle Registration 24](#_Toc410793693)

[Table 5: Use Case - Update Vehicle 25](#_Toc410793694)

[Table 6: Use Case – View Vehicle 25](#_Toc410793695)

[Table 7: Use Case - Update Rent 26](#_Toc410793696)

[Table 8: Use case - Cancel Reservation 26](#_Toc410793697)

[Table 9: Use Case - View Reservation 27](#_Toc410793698)

[Table 10: Use Case - Generate Report 27](#_Toc410793699)

[Table 1: Use Case – Logout 28](#_Toc410793700)

[Figure 1: Sequence Diagram – Login 29](#_Toc410793701)

[Figure 2; Sequence Diagram - Vehicle Reservation 30](#_Toc410793702)

[Figure 3; Sequence Diagram – register ranted Vehicle 30](#_Toc410793703)

[Figure 4; Sequence Diagram – new Vehicle registration 31](#_Toc410793704)

[Figure 5; Sequence Diagram – update Vehicle 32](#_Toc410793705)

[Figure 6; Sequence Diagram – View Vehicle 33](#_Toc410793706)

[Figure 7; Sequence Diagram – Cancel reservation 34](#_Toc410793707)

[Figure 8; Sequence Diagram – update rent 35](#_Toc410793708)

[Figure 9; Sequence Diagram – view reservation 35](#_Toc410793709)

[Figure 10; Sequence Diagram – Generate report 36](#_Toc410793710)

[Figure 3.1: Activity Diagram - Login 37](#_Toc410793711)

[Figure 3.2: Activity Diagram-reserve vehicle 38](#_Toc410793712)

[Figure 3.3: Activity Diagram rent registration 38](#_Toc410793713)

[Figure 3.4: Activity Diagram-Vehicle registration 39](#_Toc410793714)

[Figure 3.5: Activity Diagram-update vehicle 39](#_Toc410793715)

[Figure 3.6: Activity Diagram-view vehicle 40](#_Toc410793716)

[Figure 3.8: Activity Diagram-update rent 40](#_Toc410793717)

[Figure 3.8: Activity Diagram-cancel reservation 41](#_Toc410793718)

[Fig 3.9 class diagram of the system 42](#_Toc410793719)

[Fig 3.10 Structural design of user interface of the system 43](#_Toc410793720)

[Figure 4.1Class type architecture 47](#_Toc410793721)

[Table 4.1 Class documentation 48](#_Toc410793722)

[Figure 4.2 Class modeling 50](#_Toc410793723)

[Figure 4.4 State chart diagrams for login 52](#_Toc410793724)

[Fig 4.5 collaboration modeling for login 53](#_Toc410793725)

[Fig 4.6 collaboration modeling for rent registration 54](#_Toc410793726)

[Fig 4.7 component modeling diagram 55](#_Toc410793727)

[Figure 4.8 deployment diagram 58](#_Toc410793728)

[Figure 4.9 login user interfaces 59](#_Toc410793729)

[Figure 4.10 home page user interfaces 60](#_Toc410793730)

[Figure 4.11 rent registration user interface 61](#_Toc410793731)

**Abstract**

The **online vehicle rental System** is being developed for customers so that they can reserve their vehicles from anywhere. This application takes information from the customers through filling their details. A customer being registered in the website has the facility to book a vehicle which he/she requires.

The proposed system is completely integrated online systems. It automates manual procedure in an effective and efficient way. This automated system facilitates customer and provides to fill up the details according to their requirements. It includes type of vehicle they are trying to hire and location. The purpose of this system is to develop a web based application for the people who can book their vehicles along with requirements from any part of our country.

And the main objective of this proposed project is to develop a web based application that enables the organization to post and communicate information online and to serve customers in many activities like renter can rent the vehicle for weeding, for vacation, for trip and some other activities.

**Chapter 1**

# 1. Background of the project:

In coping with the globalization of information technology, computer had played an important role in an organization or company. Most of the organization used computerized system in handling all their activities regarding with data processing in Order to make the organization operation more efficient. All the manual activities are now can be done using computerized system. Although computerized system had been implemented within an organization, the manual system can still be used as guidance.

Online Vehicle Rental System is a web based application that is developed for existing system in order to replace the current manual way used by the organization. Using the manual way, there are: a lot problems arise that can cause difficulty to the vehicle renter and also lot the, management team of the organization. Hopefully. By developing this system. It willhelp the entire user to overcome the problem faced by them and make the vehicle rental operation easier.

Online vehicle rental system is the most known system in developed countries. Its main goal is to serve customers in many activities. From this the renter can rent the vehicle for weeding, for vacation, for trip and some other activities.

# 1.1 Motivation

The group is motivated in order to solve the problems in manual vehicle rental system and to automate the system.

# 1.2 Mission and vision of organization

## Vision

We will be a growth-oriented vehicle rental organization by efficiently serving value-conscious customers for all of their car rental occasions.

## Mission

We will consistently deliver a quality product, friendly service and great value that make customers confident that Budget is their best vehicle rental choice.

# 2. The existing system

Most of the vehicle rental operations are still being done manually. Therefore, it really causes difficulties to user. Whereas using the manual way, user has to fill in the provided form and give a copy of identification card and license. Generally, difficulties that had been identified in the current system are:

* Difficulties in checking vehicle status whether the vehicle is being used, being repaired or available for rental.
* Customer does not know the detailed information about the vehicles that they had rent.
* Mistakes in giving vehicle to the correct customer due to data redundancies that occurred because of the manual way.
* Keeping track of vehicle is a bit difficult to be done since it is using manual way.
* It is manual.
* It has data redundancy.
* There is time and energy wasting.

The existing system has no such applications and it not give information about online vehicle rental system. And it is not practiced online renting at all. So the defect of the system will be stated below on the statement of problem sections.

# 3. Proposed project

This online vehicle rental system is developed to provide the following services.

1. Customer can reserve a vehicle online form anywhere in the Ethiopia.
2. Every work process activity is done by computer means no need of hardcopy.

3. Everyone can access the system online.

# 4. Statement of the Problem

As we have described before the existing system has the following problems.

Those are:-

* Resource consumption:-it needs resources like man power, coast, and time, documents to fill users or customer’s information and others.
* Data redundancy:-some data’s are duplicated because of the system is manual.
* It is not user friendly:-the manual system is difficult to users to use , access and choose which vehicle they want to rent.
* Difficulty in checking vehicle:-without checking the document the organization can not differentiate the vehicle reserved.
* No more customer satisfaction:-customers cannot reserve the vehicles by their wish

# 5. Team composition

This online vehicle rental system team is composed of three members and each person has its own responsibilities and their own task in the development of this project. The following table illustrates the division of task among the group.

##### Table1.1 team composition table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project title | Online vehicle rental system (OVRS) | | | | |
|  | Name | id number | phone number | Email | Responsibility |
| Ermias Yilma | NCS/R/431/04 | 0920980648 | Ermias2147@gmail.com | designer programmer and leader |
| Dadhi Fitala | NCS/R/332/04 | 0915796258 | Dadi33214@gmail.com | System designer |
| Demisew Desalegn | NCS/R/355/04 | 0916023589 | demis@gmail.com | Conceptual diagram designer |
| Date | 15/03/2007 E.C | | |  | |
| Advisor | Letebrhan Alemayeh  Hailyie Teklesilassie | | |  | |

# 6. Objective

## 6.1 General objective

The general objective of this project is to develop web based online vehicle rental system for existing system in Ethiopia.

## 6.2 Specific objective

The proposed system has the following objectives:-

* Introduce the current system to customers (users).
* To make the manual system computerized/web based.
* To reduce wastage of time.
* To make interactive and user friendly.
* To solve the problems face in the existing system.
* To satisfy customers by saving their time and their interest.
* To serve customers anywhere online.

# 7. Scope of the project

The scope of this project is developing web based system Able to recommend vehicle to be rent by the user based on requirements, provide car catalog for users and it allows the organization to view user information from the data base if they registered or not and updating data base. Customers as well as the organization’s staff will be able to use the system effectively.

# 8. Significance of the project

The project can have the following significances:-

* To increase the owners business by saving time and consumption of materials.
* To increase the customers satisfaction by serving customers everywhere online, saving their time and cost.
* Owners to easily navigate system and manipulate data
* To update information contents to customer and user administrator easily update the information on the system.
* It is user friendly means that is easy to use and customers easily use the system without any difficulties.
* It automates the system by saving time of the organization and customers.

# 9. Feasibility of the project

## 9.1 Technical feasibility analyses:

As we have learned many programming languages such as: c++, Java, object oriented and php we have passed through many mini projects. So we are proudly sure that we can develop this system by referring references, using past mini project experiences, and by contacting our advisor.

## 9.2 Operational Feasibility

The proposed system or the new system is operationally feasible for the process of assessing the degree of which a system solves the business problems or takes advantages of business opportunity.

## 9.3 Economic feasibility

* **Cost benefit analysis**: The system which we are going to develop will have economic benefits. This means the concrete benefit that can be expressed in terms of dollars or birr. So the system proposed to develop will decrease a lot of birr that was expensive to buy the hard copy document material such as paper, pencil, rubber, and so on. It also reduces the loss of data that means if it reduces the loss of data it also reduces the cost of replacing the lost data. For example if the organization have 1000 employees to handle these amount data there must be many people to manage the data and a lot of paper, pen ,and pencil will be bought. But after the system developed the data can be managed with one person and in one computer in very short time.

# 10. Methodology for the Project

## Data collection/ fact finding

### Observation

We will use observation to get information that cannot be gained using the methodologies (Interview and document analysis) such as the behavioral response of the user.

### Interview

We will use interview to gather data from some persons about rental system orally. Because there is no online renting organization. So we interviewed persons to get information about online vehicle rental system.

## Analysis and design methodology approach

There are two types of approaches used to analysis and design. From this we used a detailed object-oriented design for the system design is used. UML is used again for the graphical representation and documentation of the design.

## Development tools

### Hardware Requirements

* Computer any type (Acer ,dell ,hp, Toshiba )
* Digital Camera, CD, Flash disk, printer.

### Software Requirements

* **Operating System**
* Win-XP, Win 7, Window 8
* Microsoft office products such as MS-word, MS- PowerPoint To documentation and presentation.
* **Applications software to be used :**
* Notepad ++
* To write the html5, java script, css, php
* Scripting language(java script)
* To secure the web pages by validation rules
* Style sheet (CSS3)
* Makes the web pages interactive and user friendly
* Coding (PHP)
* To integrate the HTML with database server (WAMP or XAMP
* My SQL Server 2005,2008
* Database part used to store, retrieve data
* Testing (WAMP or XAMP Server)
* To make database functional

# 11. Estimated budget

The estimate budget for the proposed system is **7470** Ethiopian birr

##### Table 1.2 Table to show hardware cost

|  |  |  |  |
| --- | --- | --- | --- |
| ***Material*** | ***Quantity*** | ***Unit price in birr*** | ***Price in birr*** |
| **Rewritable CD** | **2** | **25** | **50** |
| **Paper** | **1ream A4** | **100** | **100** |
| **Pen** | **5** | **4** | **20** |
| **Flash** | **2(4GB)** | **150** | **300** |
| **Computer** | **1** | **4,000** | **4,000** |
| **Printer** | **1** | **3,000** | **3,000** |
| **Total** | | | **7470 birr** |

# 12. Tasks and Time schedule.

Time evaluation is the most important consideration in the development of project. The time schedule required for the development of this project is very important since more development time effect machine time, cost and cause delay in the development of the systems.

Regarding our current source, technology and our ability the time that takes to complete this system successfully will be. (**7 months.)**

##### Table 1.3 Timeline

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tasks | | Months | | | | | | | | |
| No | Steps | Nov1-16 | Nov 17- Dec 02 | De 03-18 | Dec 19-Jan 12 | Feb | Mar | App | May | June |
| 1 | Proposal preparation and Presentation |  |  |  |  |  |  |  |  |  |
| 2 | Data collection and Requirement specification |  |  |  |  |  |  |  |  |  |
| 3 | System analysis |  |  |  |  |  |  |  |  |  |
| 4 | System design |  |  |  |  |  |  |  |  |  |
| 5 | Implementation |  |  |  |  |  |  |  |  |  |
| 6 | Testing |  |  |  |  |  |  |  |  |  |
| 7 | Documentation |  |  |  |  |  |  |  |  |  |
| Total | Overall project time |  |  |  |  |  |  |  |  |  |

# 13. Limitation of the project

Our project is limited to the followings

* The project scope is limited to automate the existing system
* GUI is in English means it not read Amharic.
* It not works without internet.
* It cannot work online payment.
* Without power there is no se rvcie.

# 14. Risk, assumption and constraint

**Risk**

While we are doing our project we encounter different problems.

##### Table 1.4 risk assumption

|  |  |  |
| --- | --- | --- |
| # | Risk | Actions |
| 1 | Computer viruses, computer failures | Backup the files, scanning with anti-viruses and recovering the system |
| 2 | Time when Power and computer lab is not available | Working when the power and computer lab is available, and using our time effectively |

**Assumption**

* We feel that the computer lab is available for us every time we need.
* Backup the files, scanning with anti-viruses and recovering the system will make our working environment free from computer viruses.

**Constraints**

The proposed project is supposed to have the following constraints: -.

* + Time constraint:-The time allotted for the project was not enough.
  + Budget constraint (Inadequate budget.):- Starting from gathering information to do this project and writing proposal we didn’t have enough money and other resources to complete our project

**Chapter 2**

**2 Description of the Existing System**

**2.1 introductions to the existing system**

The system in which our project is applied on is online vehicle rental system. The system is under private ownership. The system works with manager, employer and customers. The workers interact with each other in order to carry out the company’s (organization’s) activities. The system runs almost all its activities manually. It uses paperwork and direct conversation to manage the system. It uses personal management system which requires the manager himself to work on every aspect of the system rather than just managing others.

**2.2 Players in the existing system**

As mentioned earlier the online vehicle rental system consists of three which consist of actors within. Each actor has its own responsibilities.

It’s described as follows:

* **Manager**
  + Responsible for allotting specific job for the employees.
  + Maintain performance check by keeping in touch with the entire employee’s.
  + Conduct business meeting to promote new vehicle sales.
  + Prepare and manage budget.
* **Employees** 
  + Coordinates to increase the quality of services
* Inform the vehicles if the vehicles may be reserved or not for manage
* Assists in Check-in, check-out, reservation confirmation and reservation System
* **Customer**
* Reserve vehicles
* Rent vehicles
* Update rent
* Cancel reservation if they face some troubles.

**2.3 Major functions/activities of the existing system**

In the existing system different activities are accomplished. The activities are;

* ***Input:*** *-* in the existing system there are different input forms which help to customers.
* In the existing system there is no any kind form which helps to *user* to search keywords, and login forms to the administrator to enter his/her username and password because it is manual.
* The input is customer order on face to face contact only.
* ***Process:*** *-* in the existing system only one process activity is accomplished.
* For example the customer can only fill the information’s on a given manual form to rent a vehicle which is not reserved by others.
* ***Output*:** - in the existing system only one output activity is accomplished.
* Finallythe inputted form if valid and available and the user or customer agreed on the agreement then he pays the money to rent vehicle.
* Vehicles reserved, rented and free for rent.
* The output is payment.

**2.4 Business rule**

The following are the major business rule used by the existing system is

* Normally the owner expects customers to make payment before they rent the vehicle.
* But it’s allowed to pay for their expenditure in check-out time.
* Customers that use the vehicles make payment immediately before getting served. This means there is no credit involved for customers who want to pay after their rent.
* No payment is done without manual receipt.
* Service charge is applied on every service given by the system.

**2.5 Report generated in the existing system**

The report is generated from based on the work done. Some of them are explained as follows:

* The vehicles which are rented by the customers.
* The amount of money all receivable and payable accounts.
* The vehicles which are not rent.
* And all the reports are in manual paper or orally.

**2.6 Forms and other documents of the existing system**

In any business documents and forms are the fundamental for the existence of different process but in the existing system there is no form. Because it is working with the help of manual receipt.

All the necessary information about the vehicle rented is filled in receipt and interconnects with the master sheet.

The documents returned by the existing system includes:-

* The existing result in MS-word or spread sheet format this is for rent and reserve vehicle.

**2.7 Bottle necks of the existing system**

**2.7.1 Performance (response time):**

* The performance of the system based on response time is not having a good performance because it is manual.
* And since it is done by man power sometimes the employees can get some trouble at that time customers cannot get performance at better response time so it has no good performance.

**2.7.2 Input:**

The input in the existing system is only customer order but others are not available because;

* data is not correctly captured (gathered)
* Data is not flexible: not easy to meet new information needs from stored data.

**2.7.3 Output**

There is only payment output in the existing system because:-

* Lack of necessary information
* Information that is not timely to its subsequent use

**2.7.4 Security and control**

* In the current system files are exposed to theft, unauthorized modification due to low data security levels and standards.
* Customer’s data is extremely insecure.
* Everyone can access customer’s data on manual form.
* So there is no security at all in the existing system.

**2.7.5****Efficiency**

* The existing system lacks much efficiency because it takes time to perform tasks, the data may be redundantly recorded and data entry procedure may be exposed to errors. The system is not well organized to perform each activities and it is not flexible to change or modify the data whenever needed.

**2.8 Practiced to be preserved**

Means things we are going to take from the existing system in the proposed system. The newly proposed system comes up with new and updated information, lookout and dynamic. From the existing system we will add the following things to our proposed system

* For individual customer there is a unique customer ID card.
* All forms and documents mentioned above must be preserved.
* Using of Vehicle ID, which makes renting of vehicles simple and easy.

**2.9 Proposed solution for the new system that address problems of the existing system**

Our proposed system is a complete new design of the existing system. Depending on the problems identified under section 2.7 the following alternative solutions can be considered to address the problems.

* To increase the owners business by saving time and consumption of materials.
* To increase the customers satisfaction by serving customers everywhere online, saving their time and cost.
* Owners to easily navigate system and manipulate data
* To update information contents to customer and user administrator easily update the information on the system.
* It is user friendly means that is easy to use and customers easily use the system without any difficulties.
* It automates the system by saving time of the organization and customers.

**2.10 Requirements of the proposed system**

**2.10.1 Functional requirement**

Functional requirements are the intended behaviors of the system. This behavior may be expressed as services, tasks or functions that the system is required to perform.

Since we are going to develop a web based application for online vehicle rental system the system will have different functions some of the function are;

* **Login**:- to identify the authorized person to use the system
* **Reserve Vehicle: -** The system shall allow staff to display customers, who rent vehicles.
* **View Reservation:-**the system allows the staff to view reservations of vehicles that which are reserved and which are available for customers.
* **Vehicle Registration**:-the system allows the manager to add new vehicles to the system for customers to choose new vehicle and increase their satisfaction.
* **Rent registration*: -*** the system allows customer to register for renting vehicles .
* **Update Rent:-**the system should allow the staff/employee or manager to update rent where customer wants to add days for use vehicles more than rent registration agreements and to cancel their registration.
* ***View Vehicle*:-**the system should allow customers, staff and others to view vehicles available for rent and which are reserved by customers.
* ***Update Vehicle*:-**the system shall allow the manager to drop or deleted vehicles that have already rented from the database.
* ***Cancel Reservation:***-the system allows the customer to cancel the renting agreement online.
* ***Generate report:-***the system allows manager to generate reports about rented, reserved or free vehicles.

**2.10.2 Nonfunctional requirement**

Non-functional requirement is a [requirement](http://en.wikipedia.org/wiki/Requirement) that specifies criteria that can be used to judge the *operation of a system*, rather than specific behaviors.

Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "[constraints](http://en.wikipedia.org/wiki/Constraints)", "quality attributes", "quality goals" and "quality of service requirements". Qualities, that is, non-functional requirements, can be divided into two main categories:

1. Execution qualities, such as security and usability, which are observable at run time.
2. Evolution qualities, such as [testability](http://en.wikipedia.org/wiki/Software_testability), maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

* ***Performance***

Computer performance is characterized by the amount of useful work accomplished by a computer system compared to the time and resources used.

* ***User interface***

The user interface of our system is displaying the overall information of system which is the homepages.

* And it will be friendly with any user
* It is attractive.
* ***Security***
* The system support user name and password to authenticate.
* The system should give different privilege to protect interfering.
* The system administration must grant and revoke privilege of the user.
* ***Backup and recovery***

The system will back up all its information in to the server of organization server and, the website will also be available in external storage devices like CD disks.

* ***Storage related***

The application that we are going to develop will store all its data in to the ***MYSQL***database which is the backend or panel of the website.

* ***Resources***

To develop our web based application we have used different types of hardware and software resources. Some of the resources are

* ***Hardware***
* Computer, digital camera, paper etc.
* ***Software***
* Operating systems like Window 7/XP
* And different Markup languages like HTML, CSS,
* Scripting and server side languages like PHP, JavaScript.
* Applications WAMP server

**Chapter 3**

**3. System analysis (modeling of the existing and proposed system)**

**3.1 Introduction**

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem Activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through the various processing that the inputs phase through in the organization.

A detailed study of these processes must be made by various techniques like Interviews, Questionnaires etc. The data collected by these sources must be studied to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now, the existing system is subjected to close study and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as a proposal. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is reviewed on user request and suitable changes are made.

**3.2 System Requirement Specification (SRS)**

A well-designed, well-written SRS accomplishes four major goals:

* It provides feedback to the customer. An SRS is the customer's assurance that we have understood the issues or problems to be solved and the software behavior necessary to address those problems.

It decomposes the problem into component parts. The simple act of writing down software requirements in a well-designed format organizes information, places borders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.

* It serves as an input to the design specification. The SRS serves as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirements so that a design solution can be devised.
* It serves as a product validation check. The SRS also serves as the parent document for testing and validation strategies that will be applied to the requirements for verification.

**3.2.1 Use case diagrams**

A *use case diagram is* a set of use cases for a system, the actors of these use cases, the relations between the actors and these use cases, and the relations among the use cases. The UML (unified modeling language) notation for a use case diagram is containing different shapes and each shape has its own meaning.

* An oval represents a use case,
* A man like figure represents an actor,
* A line between an actor and a use case represents that the actor initiates and/or participates in the process.

**Use cases**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case.

Informally speaking, a use case is a story or a case of using a system by some users to carry out a process. A bit more precisely speaking, a *use case* describes the sequence of *events* of some types of users, called *Actors*, using some part of the system functionality to complete a process.

**Actor**

An *actor* represents a coherent set of *roles* that are entities *external* to the system can play in using the system, rather than representing a particular individual. An actor represents a type of *users* of the system or external systems that the system interacts with.

**System use case diagrams**



##### Fig 3.1 use case diagram of the system.

**3.2.2 Use case documentation**

This part describes the pictorially represented use case documentations. As we have drawn the use case diagram for administrator to login and for the search box form, now we are going to define them each and individually (for the login form and for the search box form)

1. **Use-case Login**

##### Table 1: Use Case - Login

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Log in | |
| Actor | Manager/Staff | |
| Description | This use case describes how Staffs to login into the OVRS System. | |
| Precondition | None | |
| Post-condition | If the use case was successful, the actor is now logged into the OVRS system. If not, the system state is unchanged. | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The staff is on the home page to login to the system.   3. The staff enters username and password, Click on Login Button. | 1. The system promotes the staff to enter Username, Password. 2. The system verifies that all the filled have been filled out and valid. 3. The system successfully logged in the system. 4. Use case Exit |
| Alternate course of Action | 6.1 If all fields are not filled out and not matched to the username and password the system notifies the actor a message Verify Username or Password and then goes back or returns to step 4 of basic course of Action to enter again. | |

**2. Use-case Reserve Vehicle**

##### Table 2: Use Case - Reserve Vehicle

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Reserve vehicle | |
| Actor | Customer | |
| Description | This use case permits customers to reserve and make schedule for renting vehicle, based on the availability of the vehicle. | |
| Precondition | Customer wants to reserve a vehicle and reservation details about customer have to be entered. | |
| Post-condition | Customers reserve successfully | |
| Basic course of  Action | **User Action** | **System Response** |
| 1. The customer wants to reserve a vehicle.  2. The customer clicks reservation page.  4. The customer enters the following information customer (full name, ID/Passport No, Country, Mobile number and selects vehicle plate number, Pickup date & return date)  5. The customer clicks reserve button to reserve.  8. The customer accepts the reservation and clicks Accept. | 1. The system prompts the customer to fill a reservation form.   6. The system checks all required information had been filled and the date entered dates are valid  7. The system presents information to accept or decline the rental Agreement.  9. The system shows the customer that the reservation has been completed, and presents the customer a reservation confirmation number.  10. Use case ends. |
| Alternate course of Action | 6.1 If the customer enters invalid date and time, the system goes back to step 4 to enter the valid date and time.  6.1 If the customer fills invalid information, the system goes back to step 4 to enter the invalid field again.  7.1 If the customer declines the agreement, the system displays a message that reservation canceled. | |

1. **Rent Registration**

##### Table 3: Use Case - Rent Registration

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Rent Registration | |
| Actor | Staff | |
| Description | This use case permits to register rental information of the customers and the vehicle that the customer rents. | |
| Precondition | UC-1 | |
| Post-condition | Customer rent information | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The customer wants to take the reserved vehicle.  2. The staff open rent page.  4.The staff enters Full name, Nationality, Country, City, Identification Number, Phone, Plate No, Down Payment, Daily Price, Rent Date, Return Date, Total Rent Day, Total Payment, Refund  5. The staff clicks on rent button. | 3. The system displays a form to be filled out for renting the vehicle.  4. The system prompts to enter the following information.  6. The system verifies that basic fields have been filled out.  7. The system displays successful rent summary  8. Use case Exit. |
| Alternate course of Action | 6.1 If Full name, Nationality, country, City, Id/Passport, Phone, Car Plate No, Down Payment, Price/day, Rent Date, Return date and Total Payment this fields are not filled out system goes back or returns to step 4 of basic course of Action. To fill invalid field. | |

1. **Vehicle Registration**

##### Table 4: Use Case - Vehicle Registration

|  |  |  |  |
| --- | --- | --- | --- |
| Use-Case Name | | Vehicle Registration | |
| Actor | | Staff | |
| Description | | These use case permits staff to register New Vehicles to the system with detail descriptions about the Vehicle such as condition, Model, Brand, fuel type, Number of sits and amount of price per day. | |
| Precondition | | New vehicle Purchased | |
| Post-condition | | New Vehicle information stored successfully. | |
| Basic course of Action | | **User Action** | **System Response** |
| 1. The staff wants to add a new vehicle  2. The staff requests add new vehicle form page.  4. The staff enters the following information in the form.  Vehicle Brand, Vehicle Type, Vehicle Model, Fuel Type, Plate Number, Number of Sits, Condition, Price per day  5. The staff clicks or presses on the save or insert button. | 3. The system response or displays a form to be filled out for vehicle registration.  6. The system verifies that the fields have been filled out correctly.  7. The system displays a successfully stored message to the employee.  8. Use case Exit |
| Alternate course of Action | | 6.1 If all fields are not filled out the system goes back or returns to step 4 of basic course of Action. To fill the invalid or the empty field. | |
| Alternate course of Action | 4.1 If any lists are not selected from the combo box system goes back or returns to step 3 of basic course of Action to select from the combo box. | | |

1. **Update Vehicle**

##### Table 5: Use Case - Update Vehicle

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Update Vehicle | |
| Actor | Staff/manager | |
| Description | This use case permits staff to update or modify vehicle information. | |
| Precondition | UC-1, UC-5, | |
| Post-condition | updated vehicle information | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The employee wants to update vehicle information.  2. Search vehicle by plate number.  4. The staff enters update information of vehicle.  5. The employee click on update button. | 3. The system will display all information about the vehicle.  6. The system successfully updates information in to database.  7. Use case Ends. |
| Alternate course of Action | * 1. If vehicle is not found back to basic course of action 2 | |

1. **View Vehicle**

##### Table 6: Use Case – View Vehicle

|  |  |  |
| --- | --- | --- |
| Use-Case Name | View Vehicle | |
| Actor | Staff , manager and customer | |
| Description | This use case allows staff and customer to view or display all vehicles with their detail description about the vehicle. | |
| Precondition | Vehicle Rent, Reserve | |
| Post-condition | Views all vehicles | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The staff or Customer wants view vehicle.  2. The staff or customer click on view vehicles button. | 3. The system retrieves all information about the vehicles.  4. Use case exit. |
| Alternate course of Action | * 1. If in the database no matched vehicle available or empty goes to Basic course action of 4. | |

**7. Update Rent**

##### Table 7: Use Case - Update Rent

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Update Rent | |
| Actor | Staff and customer | |
| Description | This use case permits employee to update or modify Rent information in case when there is a need for editing | |
| Precondition | Need to Change information | |
| Post-condition | Successful Update Message | |
| Basic course of Action | **User Action** | **System Response** |
| 1. Staff wants to update rent.  2.Open the rent page  3. Search by unique attribute which is given to customer during rent.  5.The Staff update the information  6. Click on update button. | 4. The system displays the rent information.  7. The system validates updated information and saves updated information in to database.  8. Exit use case. |
| Alternate course of Action | 4.1 If match is not found go back to basic course of action 3.  7.1if the entered information is invalid the system back to basic course of action 5 | |

**8. Cancel Reservation**

##### Table 8: Use case - Cancel Reservation

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Cancel a Reservation | |
| Actor | Customer | |
| Description | This use case permits a customer to cancel a reservation. | |
| Precondition | Customer already has reserved and wants to cancel the reservation | |
| Post-condition | Customer successfully cancel a vehicle | |
| Basic course of Action | **User Action** | **System Response** |
| 1. The customer wants to cancel reservation  2. The customer opens reservation page and clicks cancel reservation link  4. The customer enters reservation confirmation number and clicks cancel reservation button. | 1. The system displays a form   5. The system verifies the field has been filled out correctly and checks validity of confirmation number, then popup a message to verify the canceling.  6. The system cancels the reservation and display a message the reservation is canceled.  7.use case Exit |

**9. View Reservation**

##### Table 9: Use Case - View Reservation

|  |  |  |
| --- | --- | --- |
| Use-Case Name | View Reservation | |
| Actor | Staff | |
| Description | These use case allow staff to view or display customer reservation. | |
| Precondition | UC-1 | |
| Post Condition | Display all reservations | |
| Basic Course of Action | **User Action** | **System Response** |
| 1. The staff wants to view reservation. 2. The staff requests the reservation Page. 3. Then on reservation page the employee clicks view button. | 1. The system responds the requested page. 2. The system puts on view or displays all reservation information to the employee. 3. Use case ends |
| Alternate course of Action | * 1. If reservation not found system goes to basic course of action 6. | |

**10. Generate Report**

##### Table 10: Use Case - Generate Report

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Generate Report | |
| Actor | Manager /staff | |
| Description | These use case allow Manager of the organization to generate a report about the renting information of a month. | |
| Precondition | Manager wants to see report | |
| Post Condition | Generate monthly Report Information | |
| Basic Course of Action | **User Action** | **System Response** |
| 1. The Manager wants to generate report. | 2. The system responds the requested page.  3.Use case ends |
| Alternate course of Action | * 1. If the reservation information is empty or not found go to 8. | |

**11. Logout**

##### Table 1: Use Case – Logout

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Log out | |
| Actor | Staff/manager | |
| Description | These use case allow Staff to log out from the system at a time of accomplishing their work. | |
| Precondition | UC-1 | |
| Post Condition | System logs out | |
| Basic Course of Action | **User Action** | **System Response** |
| 1. The Staff or manager wants to log out 2. The Staff or manager clicks the log out button | 1. The system responds to the requested action. 2. The system displays a message that the Staff or manager logged out from the system. 3. Use case Ends |

**3.2.3 Sequence Diagram**

A *sequence diagram* shows an interaction arranged in time sequence. In particular, it shows the instances participating in the interaction by their “lifelines” and the stimuli that they arranged in time sequence. It does not show the associations among the objects.

**1. Sequence Diagram – Login**

****

##### Figure 1: Sequence Diagram – Login

**2. Sequence Diagram – Vehicle Reservation**

##### Figure 2; Sequence Diagram - Vehicle Reservation

1. **Sequence Diagram – Rent Registration**

****

##### Figure 3; Sequence Diagram – register ranted Vehicle

1. **Sequence Diagram – Vehicle Registration**



##### Figure 4; Sequence Diagram – new Vehicle registration

1. **Sequence Diagram – Update Vehicle**



##### Figure 5; Sequence Diagram – update Vehicle

1. **Sequence Diagram – View Vehicle**



##### Figure 6; Sequence Diagram – View Vehicle

1. **Sequence Diagram – Cancel Reservation**



##### Figure 7; Sequence Diagram – Cancel reservation

1. **Sequence Diagram – Update Rent**



##### Figure 8; Sequence Diagram – update rent

1. **Sequence Diagram – View Reservation**



##### Figure 9; Sequence Diagram – view reservation

1. **Sequence Diagram – Generate Report**



##### Figure 10; Sequence Diagram – Generate report

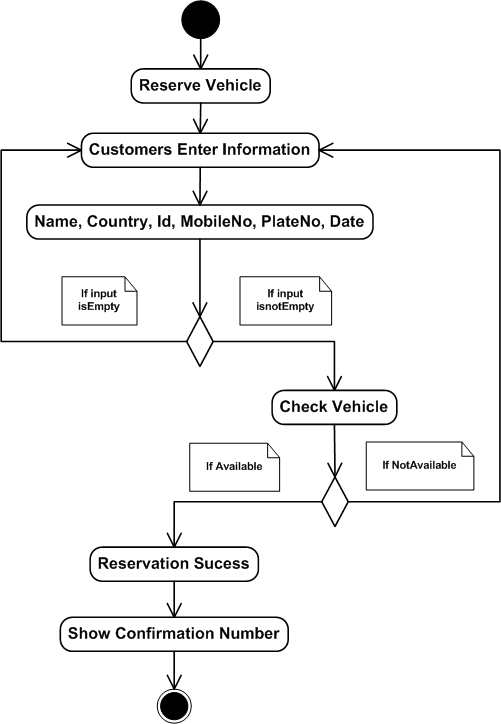
**3.2.4 Activity Diagram**

An activity diagram is a variation of a state machine in which the states represent the performance of actions or sub activities and the transitions are triggered by the completion of the actions or sub activities. It represents a state machine of a procedure itself.

1. **Activity Diagram - Login**

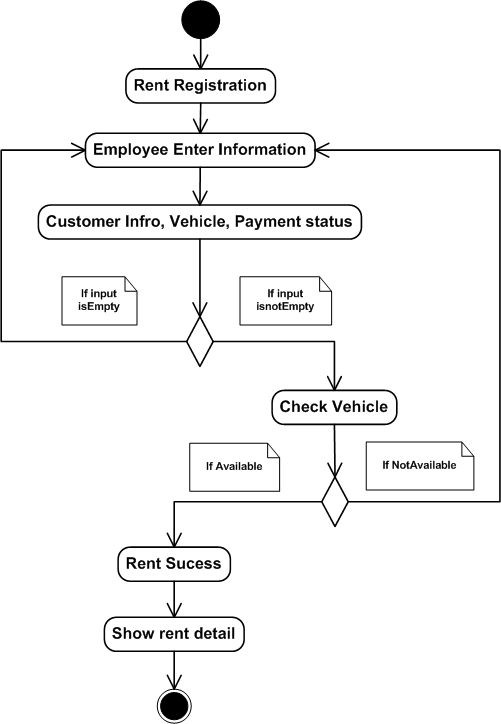
##### Figure 3.1: Activity Diagram - Login

1. **Activity Diagram – Reserve a Vehicle**



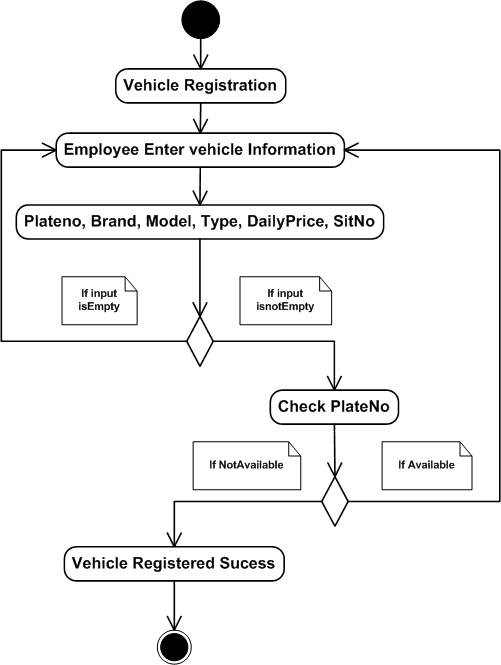
##### Figure **3.2**: Activity Diagram-reserve vehicle

1. **Activity Diagram – Rent registration**



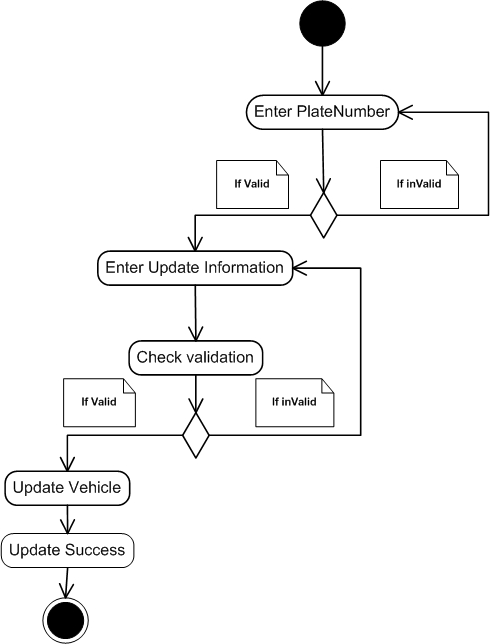
##### Figure **3.3**: Activity Diagram rent registration

1. **Activity Diagram – Vehicle registration**



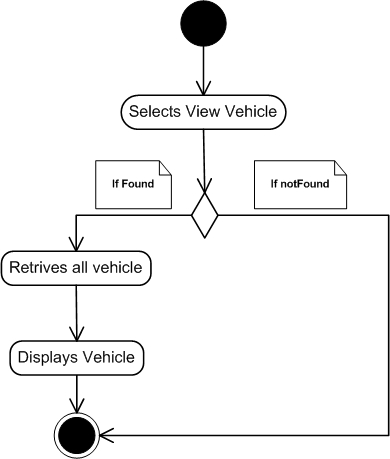
##### Figure **3.4**: Activity Diagram-Vehicle registration

1. **Activity Diagram – Update Vehicle**



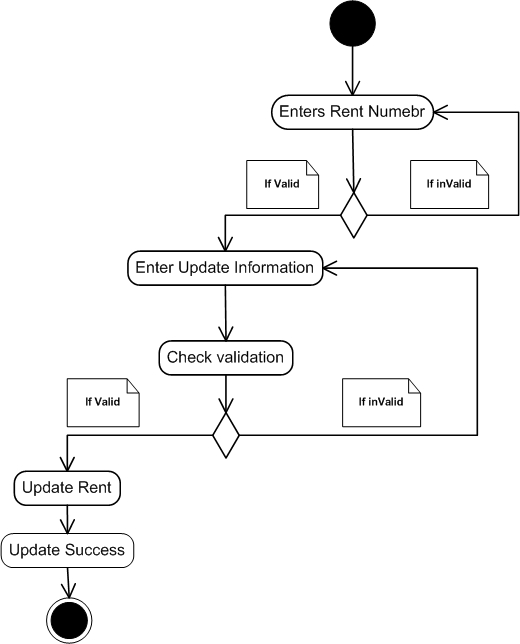
##### Figure **3.5**: Activity Diagram-update vehicle

1. **Activity Diagram – View Vehicle**



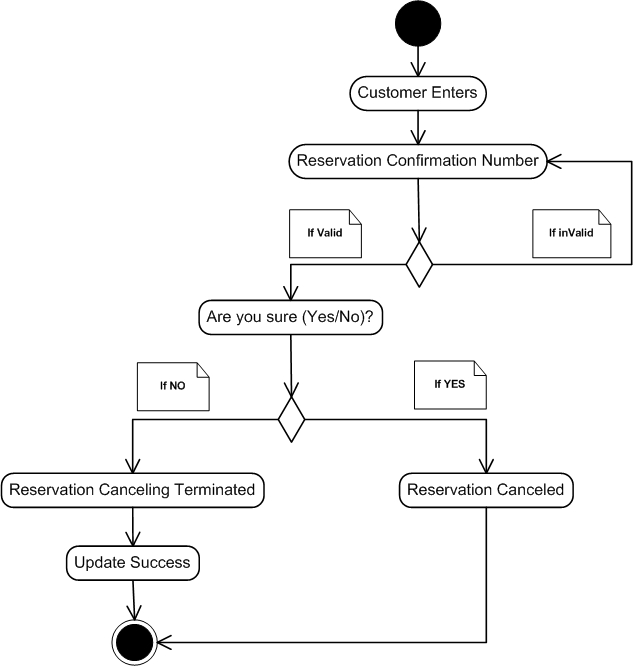
##### Figure **3.6**: Activity Diagram -view vehicle

1. **Activity Diagram – Update Rent**



##### Figure **3.8**: Activity Diagram-update rent

1. **Activity Diagram – Cancel Reservation**



##### Figure **3.8**: Activity Diagram-cancel reservation

**3.2.5 Analysis level class diagrams (conceptual modeling)**

A **class diagram** in the Unified Modeling Language(UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

In the diagram, classes are represented with boxes which contain three parts:

* The top part contains the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

In the design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

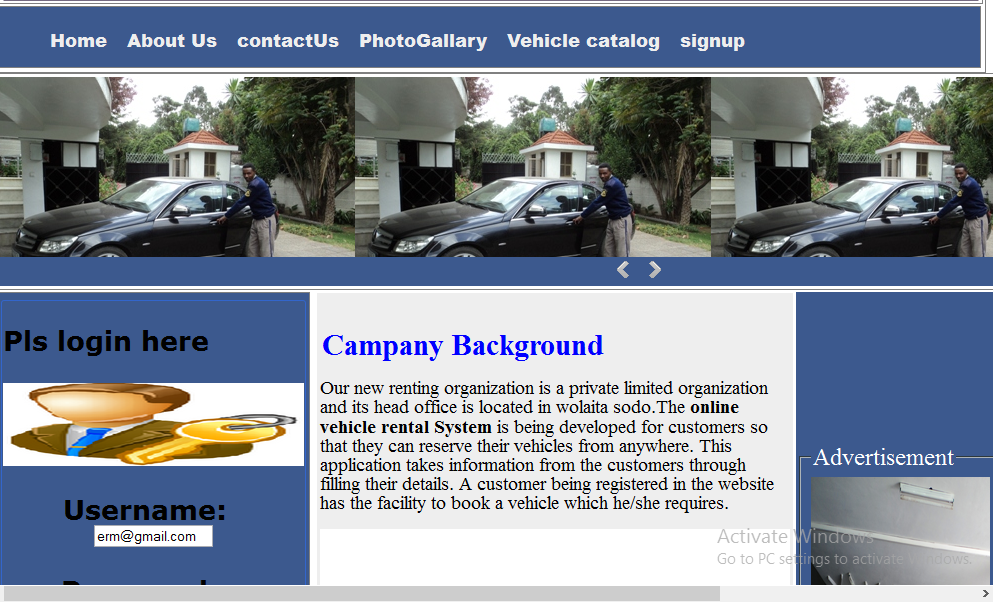
This section discusses classes and their variations, including templates and instantiated classes, and the relationships between classes association and the contents of classes (attributes and operations).Class diagrams show the static structure of the model, in particular, the things that exist (such as classes and types), their internal structure, and their relationships to other things.

**Class diagram for online vehicle rental system.**

##### Fig 3.9 class diagram of the system

**3.2.6 User Interface Prototyping**

In order to capture all the user interface requirements of the user the team member made through analysis and identified the following structural interfaces



##### Fig 3.10 Structural design of user interface of the system

**Note:** The above user interface prototyping has the big hyperlinks and have their some sub links. But because of space we didn’t mention all sub links for hyperlinks.

**3.2.7 Supplementary specification**

It will have additional hyperlinks which were not on the existing system. That the existing system is manual and it not contains much other information.

# Chapter 4

# System Design

# 4.1 Introduction

**What is systems design?**

Systems design is simply the design of systems. It implies a systematic and rigorous approach to design an approach demanded by the scale and complexity of many systems problems.

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems

**The most important thing to be aware of in systems design**

A systems approach to design asks:

* For this situation, what is the system?
* What is the environment?
* What goal does the system have in relation to its environment?
* What is the feedback loop by which the system corrects its actions?
* How does the system measure whether it has achieved its goal?
* Who defines the system, environment, goal, etc.—and monitors it?
* What resources does the system have for maintaining the relationship it desires?
* Are its resources sufficient to meet its purpose?

And finally this chapter mainly focuses on how the system is going to be built. Here Object Oriented Design (OOD) methodology is used for transforming the analysis model into design model. This serves as a blue print for the construction of the software.

The following topics are included in this chapter

* Class type Architecture
* Design Class Modeling
* Class Documentation
* State Chart Diagram
* Collaboration Diagram
* Component Diagram
* Deployment Diagram
* Relational Persistence model
* User Interface design

# 4.2 Class Type Architecture

The class type architecture in this system has three layers such as application, business and data layers. Each layer have their own use application layer provides for user interface and business layer is use to transfer data from application layer to data layer



Figure 4.1Class type architecture

## Class type architecture documentation

|  |  |
| --- | --- |
| Layer | Description |
| User Interface | There are two categories of interface class – user interface (UI) classes that provide people access to your system and system interface (SI) classes that provide access to external systems to your system. |
| Controller  /Process | The process layer implements business logic that involves collaborating with several domain classes or even other process classes. |
| Business  /Domain | This layer implements the concepts pertinent to your business domain such as Student or forums, focusing on the data aspects of the business objects, plus behaviors specific to individual objects. Enterprise Java Bean (EJB) entity classes are a common approach to implementing domain classes within Java. |
| Persistence | [**Persistence layers**](http://www.ambysoft.com/essays/persistenceLayer.html) encapsulate the capability to store, retrieve, and delete objects/data permanently without revealing details of the underlying storage technology. Often implement between your [**object schema and your database schema**](http://www.agiledata.org/essays/drivingForces.html) and there are various available to you. |
| System | System classes provide operating-system-specific functionality for your applications, isolating your software from the operating system (OS) by wrapping OS-specific features, increasing the portability of your application. |

##### Table 4.1 Class documentation

# 4.3 Class Modeling

A Class is a standard UML construct used to detail the pattern from which objects will be produced at run-time. A class is a specification - an object an instance of a class. Classes may be inherited from other classes (that is they inherit all the behavior and state of their parent and add new functionality of their own), have other classes as attributes, delegate responsibilities to other classes and implement abstract interfaces.

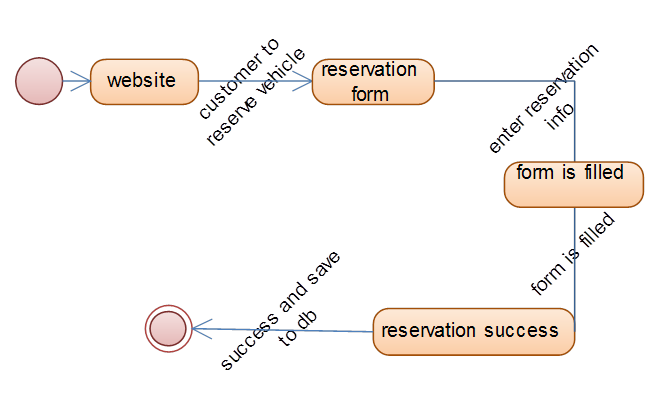
The Class Model is at the core of object-oriented development and design - it expresses both the persistent state of the system and the behavior of the system. A class encapsulates state (attributes) and offers services to manipulate that state (behavior). Good object-oriented design limits direct access to class attributes and offers services which manipulate attributes on behalf of the caller. This hiding of data and exposing of services ensures data updates are only done in one place and according to specific rules - for large systems the maintenance burden of code which has direct access to data elements in many places is extremely high.



##### Figure 4.2 Class modeling

# 4.4 State chart modeling

State chart diagram is used for modeling the dynamic aspects of systems. It is similar to activity diagram. Both activity and state chart diagrams are useful in modeling the lifetime of an object. However, activity diagram shows flow of control from activity to activity; whereas state chart diagram shows flow of control from state to state. State chart modeling is a dynamic modeling technique, one that focuses on identifying the behavior within our system, behavior specified to the instances of a single class. It tries to show different state that an object passes through its life span. However, it is not necessary to build state chart for every class in the system; only state charts of complex objects are necessary to be modeled. State chart diagram enables us to observe the state of complex that simplifies implementationFigure 4.3 State chart diagrams for login



##### Figure 4.4 State chart diagrams for login

# 4.5 Collaboration modeling

Collaboration diagram show the message flow between objects in an OO application, and also imply the basic associations or relationships between classes. The rectangle represent the various objects involves that make up the application, and the line between the classes represents the relationships (association, aggregation, composition, dependencies, or inheritance



##### Fig 4.5 collaboration modeling for login



Fig 4.6 collaboration modeling for rent registration

# 4.6 Component Modeling

Our component modeling describes components of our system such as manager employee and customer how can they interact with each other depending on their functionality, and what is there common function that can share by all the components of our system It can also describe the interaction of our system components with the security control system.

****

##### Fig 4.7 component modeling diagram

# 4.7 Deployment Modeling

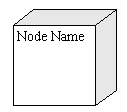
**Deployment modeling documentation**

Deployment diagrams depict the physical resources in a system including nodes, components, and connections.

**Basic Deployment Diagram Symbols and Notations**

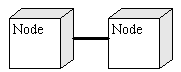
**Component**

A node is a physical resource that executes code components.



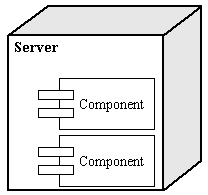
**Association**

Association refers to a physical connection between nodes, such as Ethernet.



**Components and Nodes**

Place components inside the node that deploys them.



***Java Database Connectivity (JDBC)***;-a client may access a database. It provides methods for querying and updating data in a database.

***The Apache JServ Protocol (AJP)*** is a binary protocol that can proxy inbound requests from a web server through to an application server that sits behind the web server.

***Apache*** is a web server application notable for playing a key role in the initial growth of the World Wide Web.

***MySQL*** is a popular choice of database for use in web applications. It is a relational database management system (RDBMS), and ships with no GUI tools to administer.



##### Figure 4.8 deployment diagram

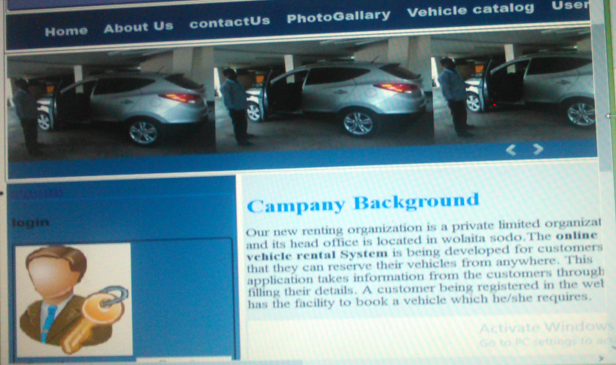
# 4.8 User Interface Design

Since our project deal on developing a web based online vehicle rental system application, it is hard to show all single pages which are found within the website, but for the time being we would like to rough design of the hyperlinks of website.



Figure 4.9 login user interfaces

##### Figure 4.10 home page user interfaces



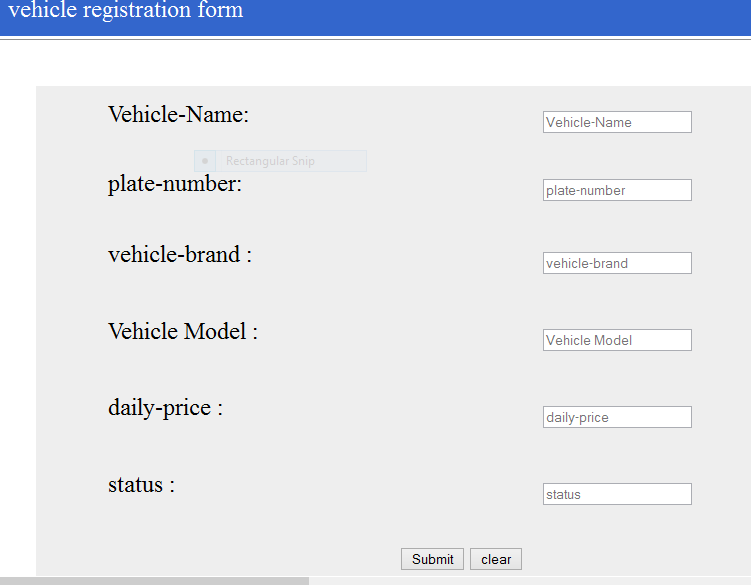


Figure 4.12 vehicle registration user interfaces

# CHAPTER 5

# Implementation

## 5.1 Introduction

In this chapter we mainly focuses on the implementation part, implementation concerned with the type of material (Hardware and software required), techniques to develop the system, algorithm for the system, code samples of the system, data preparation, some testing techniques, startup strategy for the new installed system are briefly described in this part of documentation. We test the interface the website in different systems and it is compatible all systems. System testing is also intended to demonstrate whether the system meets its objective.

## 5.2 Hardware software acquisitions

**Hardware acquisition:** since our project is developing ones organization website. The user should have a computer to browse the website.

**Server computer:** Server computer is capable of efficiently handling as much as possible for the clients accessing to it.

**Software acquisition**

The client should have any browser; such as Google chrome, Opera, or other browser in order to browse the website.

## 5.2 User manual preparation

Since our project is based on development of a website, it does not need user manual for the client. Just being online and browse the website.

In the website the login form found in pages of the website. He/she first need to fill the username and the password correctly in the provided forms. And if he/she becomes the valid user the system will show him/her the back panel of the website. Then by being in the back panel of the website he/she can do anything concerned to the website, such adding, deleting, updating the datas.

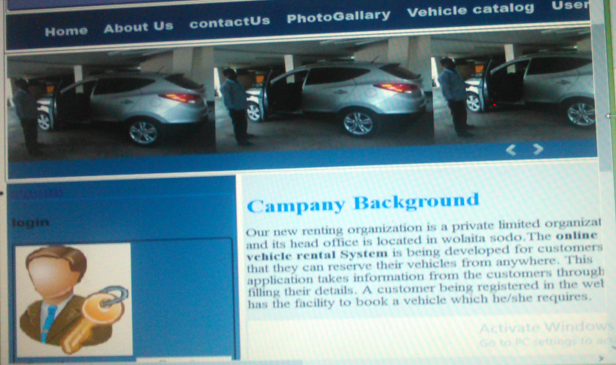
## 5.3 Training

Since our project is based on development of a website, it does not need any sort of training for the client. Because if a person can use an internet he/she have the ability how to browse a website, Just being online and browse the website

## 5.4 Installation Process

Since our project is website development, it doesn’t need to be installed in the client’s computer. But the user should install some browsers such as opera, Google chrome or other browsers to browse the website. But it order make it functional the website should be uploaded to a server. The server may can be the organization’s server or to the ISP server.

Final Testing of the system and coding



HOMRE PAGE

<?php

session\_start();

include("connect.php");

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<script type="text/javascript">

if (document.images) { // Preloaded images

demo1 = new Image();

demo1.src="images/DSC01855.jpg"

demo2 = new Image();

demo2.src="images/DSC01867.jpg"

demo3 = new Image();

demo3.src="images/DSC01899.jpg"

demo4 = new Image();

demo4.src="images/DSC01909.jpg"

demo5 = new Image();

demo5.src="images/DSC01908.jpg"

demo6= new Image();

demo6. src="images/DSC01872.jpg"

demo7=new Image();

demo7.src="images/DSC01865.jpg"

}

function timeimgs(numb) { // Reusable timer

thetimer = setTimeout("imgturn('" +numb+ "')", 2000);

}

function imgturn(numb) { // Reusable image turner

if (document.images) {

if (numb == "7") { // This will loop the image

document["demo"].src = eval("demo7.src");

timeimgs('1');

}

else {

document["demo"].src = eval("demo" + numb + ".src");

timeimgs(numb = ++numb);

}

}

}

</script>

<title>ONLINE VEHICLE RENTAL SYSTEM </title>

<link rel="icon" type="image/ico" href="images/App Icon.ico"/>

<link rel="stylesheet" href="css/newstyle.css" type="text/css" media="all" />

<link rel="stylesheet" href="css/3.css" type="text/css" media="all" />

<link href="themes/4/js-image-slider.css" rel="stylesheet" type="text/css" />

<script src="themes/4/js-image-slider.js" type="text/javascript"></script>

<link href="generic.css" rel="stylesheet" type="text/css" />

</head>

<body onLoad="timeimgs('1');">

<center>

<table style="border:1px solid gray" width="635">

<tr>

<td colspan="3" height="100px" style="border:1px solid gray"><img src="image/logo2.png" height="191" width="980"></td>

</tr>

<tr>

<td colspan="3" height="40px" width="900" style="border:1px solid gray; font-size: 18px; font-family: Verdana, Arial, Helvetica, sans-serif, times-new-roman;" id="dropdown">

<!--Main Menus-->

<li><b><a href="indx.php">Home</a></b></li>

<li><b> <a href="About Us.php">About Us</a></b></li>

<li><b><a href="contact Us.php">contactUs</a></b></li>

<li><b> <a href="photogalary.php">PhotoGallary</a></b></li>

<li><b><a href="view vehicleimg.php">Vehicle catalog</a></b>

<li><b><a href="">Users</a></b>

<ul>

<li><b><a href="login2.php">manager</a></b></li>

<li><b><a href="login2.php">employee</a></b></li>

<li><b><a href="customer.php">customer </a></b></li>

</ul>

<tr>

</tr>

</td>

</tr>

</table>

<table style="border:1px solid gray" width="400">

<tr>

<td width="1" height="100px" colspan="1" valign="top" style="border:0px solid #336699">

<!--Slide show-->

<div id="sliderFrame">

<div id="slider">

<!--<img src="images/product-img1.jpg"-->

<img src="images/DSC0186a.jpg" />

<img src="images/DSCA.jpg" />

<img src="images/DSC01870.jpg">

<img src="images/b1.jpg" />

<img src="images/b2.jpg" />

<img src="images/b3.jpg" />

</div>

<!--Custom navigation buttons on both sides-->

<div class="group1-Wrapper">

<a onClick="imageSlider.previous()" class="group1-Prev"></a>

<a onClick="imageSlider.next()" class="group1-Next"></a>

</div>

<!--nav bar-->

<div style="text-align:center;padding:2px;z-index:2;">

<a onClick="imageSlider.previous()" class="group2-Prev"></a>

<a id='auto' onClick="switchAutoAdvance()"></a>

<a onClick="imageSlider.next()" class="group2-Next"></a>

</div>

<!--End of slide show-->

</td>

</tr>

</table>

<table width="100" height="718" style="border:1px solid gray style="border:1px solid gray; font-size: 18px; font-family: Verdana, Arial, Helvetica, sans-serif, times-new-roman;"">

<tr>

<td height="641" width="14" bgcolor="#3C598E" style="border:1px solid gray; font-size: 18px; font-family: Verdana, Arial, Helvetica, sans-serif, times-new-roman;"id="menu-bar"style="border:0px solid #000000" valign="top">

<!--Side Bar-->

<table width="18">

<tr>

<td>

<li><b><a href="signup.php">signup</a></b></li>

</td>

</tr>

</tr><td valign="top" style="border:1px solid #3366CC;border-radius:3px ">

<form action="indx.php" method="post" onsubmit='return formValidation()'>

<h4>login</h4>

<table align="left" style="border-radius:5px;border:2px solid black;width:150px;height:200px">

<tr><td><font color=red><img src="images/ert.png"><br>

<tr><td><font color=red>\*</font>Email:</td><td><input type="text" name="e\_mail" value="" size="15%" id="txt\_username" placeholder="Email"></input></td></tr>

<tr><td><font color=red>\*</font> Password:</td><td><input type="password" name="pass" value="" size="15%" id="txt\_password" placeholder="Password"></input></td></tr>

<tr><td>&nbsp;</td><td><input type='submit' value='login' style="background-color:gray;color:white" name='submitMain' Onclick="return check(this.form);"/> <input type='reset' value='clear' style="background-color:gray;color:white"/></td></tr>

</table>

</form> </table>

</td>

<!--End Of Side Bar-->

<td width="0"></td>

<td width="2" style="border:0px solid #336699" valign="top">

<table bgcolor="#eeeeee"style="border:0px solid #000000" width="100" height="639">

<tr><td width="100" height="633" valign="top">

<h1><marquee height="40" behavior="slide" direction="left" scrollamount="3" align="center"><a href"#">

<font color="blue"style='font-family:Times New Roman;font-size:30px;line-height:55px;right-align:

justify;padding-left:0px; margin-top:0px;'>

Campany Background</font></marquee></h1>

<p><font color="black" style='font-family:Times New Roman;font-size:19px;line-height:19px;left-align:

justify;padding-left:0px; margin-top:0px; '>

<span class="F">Our new renting organization is a private limited

organization and its head office is located in wolaita sodo.The <strong>online vehicle rental System</strong> is being developed for customers so that they can reserve their vehicles from anywhere. This application takes information from the customers through filling their details. A customer being registered in the website has the facility to book a vehicle which he/she requires.</span></p>

<span class="F"></font></span>

<p align=center><img src="images/ermcar.jpg" height="578" width="470" >

</p>

</td>

</tr>

</table>

</td>

<td width="231" bgcolor="#3C598E"style="border:0px solid #000000" valign="center">

<fieldset style=" border-color:#45565C; text-color:#C39;height:510px;">

<legend><font color="white" size="5px">Advertisement</font></legend>

<img src="images/DSC01857.jpg" height="497" width="179" name="demo" align="center">

</fieldset></font>

</td>

</tr>

<td colspan="3" bgcolor="#000000" height="69" style="border:0px solid #000000"align="center"><p align=center><font color="#ffffff" size="4px" font="font"><a href="face.php"><img src="images/facebook-icon.png" height="35px" width="60px">

<img src="images/youtube.jpg" height="35px" width="60px"></a>&nbsp;copyRight&copy;<font color="green">Online vehicle rental system</font>. Design By <a href="#"><font color="green" >WSU CSIT 4Th year G13 Students</a></font></p>

</table></td>

</tr>

</table>

</body>

</center>

</html>

****

**Codes for login**

<td valign="top" style="border:1px solid #3366CC;border-radius:5px ">

<form action="login.php" method="post" >

<br><br>

<table align="center" style="border-radius:15px;border:1px solid black;width:350px;height:200px">

<tr><td><font color=red><img src="images/ert.png"><br>

<tr><td><font color=red>\* </font>Email:</td><td><input type="text" name="e\_mail" value="" size="20%" id="txt\_username" placeholder="Email"></input></td></tr>

<tr><td><font color=red> \* </font> Password:</td><td><input type="password" name="pass" value="" size="20%" id="txt\_password" placeholder="Password"></input></td></tr>

<tr><td>&nbsp;</td><td><input type='submit' value='login' style="background-color:black;color:white" name='log' Onclick="return check(this.form);"/></td></tr>

<tr><td>&nbsp;</td><td><a href="forget.php">Forgot Your Password ? </a></td></tr></table>

</form>

<?php

if (isset($\_POST['log'])){

$username=$\_POST['e\_mail'];

$password=$\_POST['pass'];

$sql ="SELECT \* FROM user WHERE e\_mail='$username' AND pass='$password'";

$result = mysql\_query($sql);

// TO check that at least one row was returned

$rowCheck = mysql\_num\_rows($result);

$row=mysql\_fetch\_array($result);

$status=$row['status'];

if($row['role']=='Manager'){

if($status==1)

{

$\_SESSION['user\_id']=$row['user\_id'];

echo "<script>window.location='manager.php';</script>";

}

else

{

echo'<p class="wrong"> Your Account is not valid</p>';

echo' <meta content="15;login.php" http-equiv="refresh" />';

}}

else if($row['role']=='employee'){

if($status==1)

{

$\_SESSION['user\_id']=$row['user\_id'];

echo "<script>window.location='employee.php';</script>";

//echo "<script>window.location='employee.php';</script>";

}

else

{

echo'<p class="wrong"> Your Account is not valid</p>';

echo' <meta content="15;login.php" http-equiv="refresh" />';

}}

else if($row['role']=='customer'){

if($status==1)

{

$\_SESSION['user\_id']=$row['user\_id'];

echo "<script>window.location='customer.php';</script>";

}

else

{

echo'<p class="wrong"> Your Account is not valid/p>';

echo' <meta content="15;login.php" http-equiv="refresh" />';

}}

else {

echo'<br>';

echo' <p class="wrong">Check Your username or/and Password!</p>';

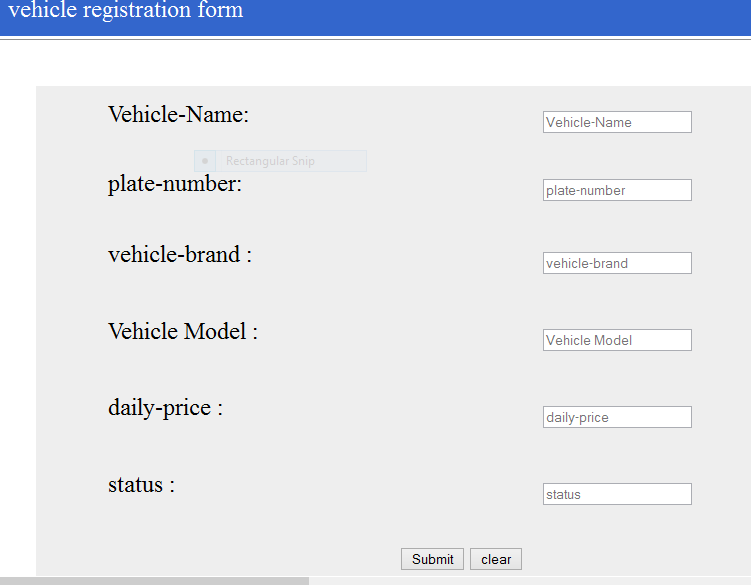
echo' <meta content="15;login.php" http-equiv="refresh" />';

}

}

// mysql\_close($conn);

?>



Code for vehicle registration page

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">

Vehicle-Name:</font></p>

<td><input type="text" name="vehiclename" id="Vehicle-Name" required x-moz-errormessage="Please Enter The Vehicle-Name"placeholder='Vehicle-Name'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">

plate-number:</font></p>

<td><input type="text" name="plate\_no" id="plate-number"required x-moz-errormessage="Please Enter The plate-number" placeholder='plate-number'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">vehicle-brand :</font></p>

<td><input type="text" name="brand" id="vehicle-brand"required x-moz-errormessage="Please Enter The vehicle-brand" placeholder='vehicle-brand'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">Vehicle Model :</font></p>

<td><input type="text" name="model" id="Vehicle Model"required x-moz-errormessage="Please Enter The Vehicle Model" placeholder='Vehicle Model'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">daily-price :</font></p>

<td><input type="text" name="daily\_price" id="daily-price"required x-moz-errormessage="Please Enter The daily-price" placeholder='daily-price'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">status :</font></p>

<td><input type="text" name="status" id="status"required x-moz-errormessage="Please Enter The status" placeholder='status'required></input></td></tr>

<tr><td colspan="4" align="center">

<input type='submit' value='Submit' name='submitMain' Onclick="return check(this.form);" required/>

<input type='reset' value='clear'/></td></tr></table>

</form>

<?php

$con=mysql\_connect('localhost','root','') or die("Faild to connect database".mysql\_error());

$db=mysql\_select\_db('vehiclerental',$con) or die("Database is not connect".mysql\_errno());

$vehiclename=$\_POST['vehiclename'];

$plate\_no=$\_POST['plate\_no'];

$brand=$\_POST['brand'];

$model=$\_POST['model'];

$daily\_price=$\_POST['daily\_price'];

$status=$\_POST['status'];

$query="INSERT INTO vehicleregistration(vehiclename,plate\_no,brand,model,daily\_price,status)";

$query.="VALUES ('$vehiclename','$plate\_no','$brand','$model','$daily\_price','$status');";

$resu=mysql\_query($query) or die(mysqli\_errno());

if($resu)

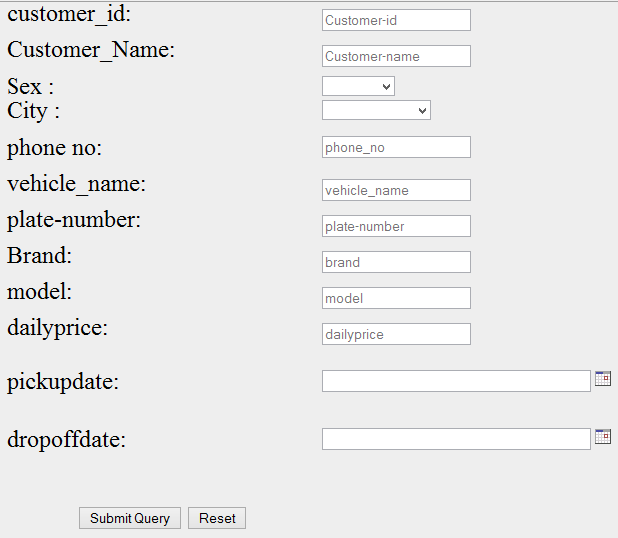
{

echo "THE VEHICLE IS REGISTERED SUCCESSFULLY!!!";

}

mysql\_close($con);

?>



CODES FOR THIS RESERVATION FORM

<form method='POST' action='reservn.php'onsubmit='return formValidation()'> <tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">

customer\_id:</font></p>

<td><input type="text" name="customer\_id" id="customer\_id" required x-moz-errormessage="Please Enter The Customer-id"placeholder='Customer-id'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">

Customer\_Name:</font></p>

<td><input type="text" name="Customer\_Name" id="Customer\_Name" required x-moz-errormessage="Please Enter The Customer-Name"placeholder='Customer-name'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px;

margin-top:0px; '>

<font color=" #ffffFF"size="5px"><font color="#000000"size="5px">Sex :</font></td>

<td><select name="sex" id="sex" placeholder='sex' required>

<option></option>

<option>Male</option>

<option>Female</option>

</select></td></tr>

<p>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</p>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px;

margin-top:0px; '>

<font color=" #ffffFF"size="5px"><font color="#000000"size="5px">City :</font></td>

<td><select name="City" id="gender" placeholder='City' required>

<option></option>

<option>Wolaita Sodo</option>

<option>Arbaminch</option>

<option>Shashamane</option>

<option>Hossana</option>

<option>Hawassa</option>

<option>Adama</option>

<option>Addis</option>

<option>Dire</option>

</select></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><p><font color=" #000000"size="5px">phone no:</font>

<td><input type="text" name="phone\_no" id="phone\_no" required x-moz-errormessage="Please Enter The "placeholder='phone\_no'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">vehicle\_name:</font></p>

<td><input type="text" name="vehicle\_name" id="vehicle\_name"required x-moz-errormessage="Please Enter The vehicle\_name" placeholder='vehicle\_name'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color=" #000000"size="5px">plate-number:</font></p>

<td><input type="text" name="plate\_number" id="plate\_number"required x-moz-errormessage="Please Enter The plate-number" placeholder='plate-number'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color="#000000"size="5px">Brand:</font></p>

<td><input type="text" name="brand" id="brand"required x-moz-errormessage="Please Enter The brand" placeholder='brand'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color="#000000"size="5px">model:</font></p>

<td><input type="text" name="model" id="model"required x-moz-errormessage="Please Enter The model" placeholder='model'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:14px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><font color="#000000"size="5px">dailyprice:</font></p>

<td><input type="text" name="dailyprice" id="dailyprice"required x-moz-errormessage="Please Enter The dailyprice" placeholder='dailyprice'required></input></td></tr>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><p><font color=" #000000"size="5px">pickupdate:</font></td>

<!--Source: http://www.javascriptkit.com/script/script2/tengcalendar.shtml -->

<td><input name="rdate" type=" text" id="rdate" maxlength="10" size="40" value="">

<a href="javascript:NewCal('rdate','MMddyyyy')"> <img src= "Javascript/Tools/Datepicker/cal.gif" width="16" height="16" border="0" alt="Pick a date"></a> </td>

<tr><td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px; margin-top:0px; '><p><font color="#000000"size="5px">dropoffdate:</font></td>

<!--Source: http://www.javascriptkit.com/script/script2/tengcalendar.shtml -->

<td><input name="ddate" type=" text" id="ddate" maxlength="10" size="40" value="">

<a href="javascript:NewCal('ddate','MMddyyyy')"> <img src= "Javascript/Tools/Datepicker/cal.gif" width="16" height="16" border="0" alt="Pick a date"></a> </td>

<tr> <td font style='font-family:Times New Roman;font-size:18px;line-height:18px;left-align:justify;padding-left:70px;

margin-top:0px; '><p align="center"><br />

<input type="submit" />

<input name="Submit2" type="reset" class="style2" value="Reset" />

</p>

<div class="date"> </div>

<p>&nbsp;</p></form>

<?php

$con=mysql\_connect('localhost','root','') or die("Faild to connect database".mysql\_error());

$db=mysql\_select\_db('vehiclerental',$con) or die("Database is not connect".mysql\_errno());

$customer\_id=$\_POST['customer\_id'];

$Customer\_Name=$\_POST['Customer\_Name'];

$sex=$\_POST['sex'];

$City=$\_POST['City'];

$phone\_no=$\_POST['phone\_no'];

$vehicle\_name=$\_POST['vehicle\_name'];

$plate\_number=$\_POST['plate\_number'];

$brand=$\_POST['brand'];

$model=$\_POST['model'];

$dailyprice=$\_POST['dailyprice'];

$rdate=$\_POST['rdate'];

$ddate=$\_POST['ddate'];

$query="INSERT INTO reservation(customer\_id ,Customer\_Name,sex,City,phone\_no,vehicle\_name,plate\_number,brand,model,dailyprice,rdate,ddate)VALUES('$customer\_id','$Customer\_Name','$sex','$City','$phone\_no','$vehicle\_name','$plate\_number','$brand','$model','$dailyprice','$rdate','$ddate')";

$resu=mysql\_query($query);

if(!$resu)

{

echo "YOUR RESERVATION IS NOT SUBMITED SUCCESSFULLY!!!".mysql\_error();

}

else

{

echo"YOUR RESERVATION IS SUBMITED SUCCESSFULLY!!!";

}

mysql\_close($con);

?>

CODE FOR DELETE VEHICLE

<?php

session\_start();

include 'connect.php';

if($log != "log"){

header ("Location: view\_vehicle.php");

}

$ctrl = $\_REQUEST['key'];

$SQL = "DELETE FROM add\_vehicle WHERE Plate\_no = '$ctrl'";

mysql\_query($SQL);

mysql\_close($db\_handle);

print "<script>location.href = 'view\_vehicle.php'</script>";

?>

# Chapter Six

# 6. Conclusion and Recommendation

## 6.1 conclusions

So far we were intended in analyzing the existing system of the ONLINE VEHICLE RENTAL SYSTEM up to proposing our new system that solves the difficulties related to the existing system. To say something on the existing system: it is running almost manually, hence it is highly exposed to the manual related problems, like the productivity of the organization is not effective , the work procedure is susceptible to error, Data is not flexible, Data is not secure as well as it is not well organized. Since the work is performed manually the efficiency of the working system is that much not good generally it degrades the effectiveness and efficiency of the system.

By having this over the existing system our aim was to build a new system that have greater functionality that enhance effectiveness and efficiency related parameters on the system. By keeping in mind that the new system will [have greater usage on]/bring the existing system fully functional.

To achieve our goal [to design new system] the project team has spent all of its time on the project on performing the tasks individually and in group based on the schedule available.

The team has faced many challenges starting from the lab due to electric power loss, lack of resources like time, money and other resources.

## 6.2 Recommendation

While doing this system the team has faced different challenges. But by the cooperation of all the group members and an advisor the team is now able to reach to the final result. Now all the group members recommend to other developers who want to maintain this system, to add some features which are not completed on this system. Among the limitations generating report by selecting id plate number and other online cash or use bank data bases and update vehicles using their plate number.

TODAY IS PASSED TOMMOROW IS NEW DAY WITH NEW UPDATES AND NEW TECHNOLOGY!!

SO

INOVATION IN TECHNOLOGY THAT NEVER ENDS!!

**Acronyms**

1. PHP …………………………....Hypertext Pre processor

2. WAMP………………………….Window Apache MySQL PHP

3. HTML………………………….Hypertext Markup Language

4. CSS……………………………...Cascading Style Sheet

5. SQL……………………………….standard query language

6. MS …………………………...........Microsoft

7. OVRS………………………………Online vehicle rental system

8. SRS…………………………………… System Requirement Specification

# References

1. The web (Internet) particularly

* [www.w3c](http://www.w3c) schools.com
* [www.google.com](http://www.google.com)
* [www.youtube.com](http://www.youtube.com)

www.mekina.com

* (Avis online car rent )www.avis.com
* [www.muskalipman/vbojects/sdlc.pdf](http://www.muskalipman/vbojects/sdlc.pdf)
* Hawk Rent A Car Group, About Hawk Rent A Car, 2008. [Online]
* Kinabalu Rent A Car, [Online]