1. PREAMBLE

1.1 Introduction of the Project

The "**Eastern Treads Management System**" software covers almost all processes in a retreading of tyres. The main aim of the project is to manage the tyre company. The project is being developed using Visual Studio as its Front End and SQLServer 2012 as its Back End.

The "**Eastern Treads Management System**" is ideal for all tyre companies. This software makes all the functions easier. At present most of the data manipulation is done manually and this has many limitations like lot of manpower and paperwork required and many resources too. Manual manipulation is time consuming.

1.2 Objective of the Project

The "**Eastern Treads Management System**" mainly deals with two retreading methods one is PRECURED and the other is the CONVENTIONAL.PRECURED method means the company itself makes th tyre perfect with the common patterns what they have and customers do not have to do anything on the tyres further. whereas the CONVENTIONAL method means the company retreads a plain tyre on the damages tyres and the customers can make patterns on the tyre upon their choice.

This software mainly deals with 3 users one is the **administrator** of the company, **staff and customer.**

The"**Eastern Treads Management System**" process reduces human errors and increases the efficiency. The main focus of this project is to lessen human efforts. With all these capabilities this project is feasible and can satisfy any requests of the customers.

1.3 Scope of the Project

The "**Estern treads Management System**" is a web application, which is developed with an intention to make the records stored in the database easier.

The "**Estern treads Management System**" is a time-saving and efficient project. The System can use it to efficiently store all the data in a secure database. It is less prone to errors as the program checks the data entered before saving it to database. If it finds any data to be unsatisfactory it shows a warning to the user to correct the error.

It is extremely simple to use and quite powerful at the same time. It takes the load off the staff or admin in the "**Estern treads Management System**". The system is very flexible and changes can be made without much difficulty. The future extension in the system can be made in such a way that addition of new modules can be done without much difficulty. The reconstruction of the system will increase the flexibility of the system.

Our main aim of the project is to make it easy for the customers to do a treading process and do the payment in online. The customer can easily work on the system and the system will give a feedback of the treading process to the customer. That is if the booking is done then the status of booking will be updated. The goals that wish to be achieved are:

1. System manages to save all the customer and staff records accurately and computerized.
2. System manages the booking process in online.
3. The system does the payment in the online.
4. The administrator of the system will see the all functionalities of the system.

2. SOFTWARE REQUIREMENT SPECIFICATION

2.1 System Study

2.1.1 Existing System

The existing system uses manual method for the whole process such manual entry of details such as staffs, tread types, tyre types, etc. This requires a lot of hard work and time consumption to complete the task. This may include human errors. In the existing system, it is difficult to retrieve some particular information. Also, all the records are stored manually and it is a tedious task. As a result the security of these records is always a challenging task. In spite of all the efforts undertaken, the destruction of data may happen often. Hence, the computerization of the system of record maintenance is the only solution to reduce the shortcomings of the existing system.

There are many disadvantages in manual system:

- **Inaccuracies:**

Since all the data entry is done manually, inaccuracies can occur. Cause of error is more in manual system.

- **Modification:**

The transactions are executed in off-line mode, hence on-line data capture and modification is not possible.

- **Expensive:**

The maintenance of this project can be expensive.

Large number of personal hours is required for each and every part of the manual system. We have to buy lot of books, paper for keeping the information. So the existing system is very expensive.

- **Inefficiency:**

The existing system only provides text-based interface, which is not as user-friendly as Graphical user Interface. Inefficiency in the existing system is also caused by the lack of proper communication between the people of the organization.

- **Time and effort:**

Large amount of time and efforts may require. For searching a particular data, we have to search all the paper documents, its corresponding book and find the data.

- **All the searching are done manually:**

In the existing system, all the work like entering the staff details, customer details, payment details etc. is done manually. This is really time consuming.

2.1.2 Proposed System

The proposed system is interactive, highly user friendly and designed exclusively for the "**Eastern Treads Management System**". The system covers almost all the functional areas of the "**Eastern Treads Management System**" such as details of staffs, salaries, tyre types, customers, retreading sales etc. All the operations and activities related to the "**Eastern Treads Management System**" can be carried out efficiently. The project maintains well organized database for storing the resources that they are provided by the client. This helps us to eliminate the entering of invalid data. Most problems of manual system can be solved by this system.

The computerization of the system allows the easy maintenance of the details. Large amount of data can be stored easily. Addition and updating other changes can be done easily. The information can be retrieved with high speed and accuracy. The use of GUI oriented software makes the system user friendly. Since all work is computerized, the calculations are effortless and less time consuming. Speed, accuracy, storage capacity, versatility, automation etc. are the advantages of using a computerized system.

The main purpose behind the proposed system is to provide a comprehensive computerized system, which can capture, collate and analyze the data from these wards and evaluate the impact of the program.

The main advantages of the new system are:

- **Security:**

The software used for this "Eastern Threads Management System" includes the password, so the security is provided. When anyone opens the software it has the provision for entering password. We have to enter the correct password; otherwise we cannot enter into the system. Password is saved in system registry for more security.

- **User Friendly:**

This package is very user friendly because it is easy to maintain and operate. All data entry operations are simple, administrator needs only to enter data and all other operations are performed by the computer.

- **Speed and Accuracy:**

Computerization process increases the speed of all the operations. The manpower is reduced. Instead of doing all operation manually, computer will do it automatically. It also increases the accuracy of all the operations performed.

- **Efficiency and flexibility:**

The flexibility and the efficiency of all the operation in the company is increased because of the computerization. No errors are occurred compared to the manual system

- **Automation:**

The proposed system automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast, and more user friendly. It will reduce the man power and execution time.

2.2 Functional Specifications

The system has 4 modules:

Module 1: Retreading

The company mainly deals with two retreading methods one is PRECURED and the other is the CONVENTIONAL.PRECURED method means the company itself makes the tyre perfect with the common patterns what they have and customers do not have to do anything on the tyres further. Whereas the CONVENTIONAL method means the company retreads a plain tyre on the damages tyres and the customers can make patterns on the tyre upon their choice.

Module 2: Registration

Each user need to register to the system by entering user name and password. If the username and password are correct, then the system will login or it will show error message to the administrator.

Module 3: Booking

The customer can book the order and the staff can record the booking and confirm the bookings. The customer can select the retreading method and continue the booking. The customer can view the rate of the patches and select the pattern type or he/she can do the conventional method.

Module 4: Payment

The payment by the customer is made in two steps first as when the customer comes with the tyre for retreading they are allowed to pay the advance amount and the bill is generated as a report and as when the customer comes for the final sale he pays the remaining amount and for this the final bill is generated as report.

2.3 User Characteristics

This software mainly deals with 3 users.

- **Admin**

The administrator of this company can access every process of this company .The administrator of this company can add the details of staffs, tread type, tyre type etc. The staffs can add the details of customer, sub patterns, retread sales etc. The admin is the all-rounder of the company. He can add the staffs to company and also can add their salary details.

The admin assign the staffs and the staffs uses their username and password to enter the project and they are restricted to access only few process in the company.

- **Staff**

The Staff can record the details of repair patches, patterns, customer details etc. to the system. He/she also can add there details and enquires. Currently the customer, staffs, patches and other details are recorded manually, which is being rectified in the proposed system. Staff can also view customers, update the billing status etc.

- **Customer**

The customer can select one of the retreading method and can book the order. The payment by the customer is made in two steps first as when the customer comes with the tyre for retreading they are allowed to pay the advance amount and the bill is generated as a report and as when the customer comes for the final sale he pays the remaining amount and for this the final bill is generated as report.

2.4 System Specification

2.4.1 Software specification

- Development Configuration
  - ➢ Machine– Windows 10 or above /Linux distros, SQL server management studio.
- Implementation Configuration
  - ➢ Client machine- Windows 10 or above /Linux distros
  - ➢ Server Machine- Windows 10 or above/Linux distros, and SQL server management studio.

2.4.2 Hardware specification

- Development Configuration
  - ➢ Machine (*Minimum Requirement*)
    - Processor –Intel i3, AMD and above version.
    - RAM – 512MB and above.
    - Hardware Device- A monitor with key board and mouse.
    - Hard disk- 100 GB

- Implementation Configuration

  - ➢ Client Machine (*Minimum Requirement*)

    - Hardware Device – A computer with a web browser.
    - RAM – 512 MB and above (Recommended 1GB).
    - Hard Disk – Min 100 GB

3. System Modeling

3.1 Design methodology

System design focuses on the final system and the process by which it is developed. It leads to transition from a user-oriented document which means system proposal a programmer-oriented document. The design methodology followed for this project is the bottom-up design strategy. In this approach, the basic sets of elements are individual modules. Each module is developed individually as a separate project and adds the modules into this as a reference if necessary. The benefit of the design is that it permits the review of the modules during the system development process. We prepare Dataflow diagrams to understand a system in better and simple way.

3.2Data flow diagram

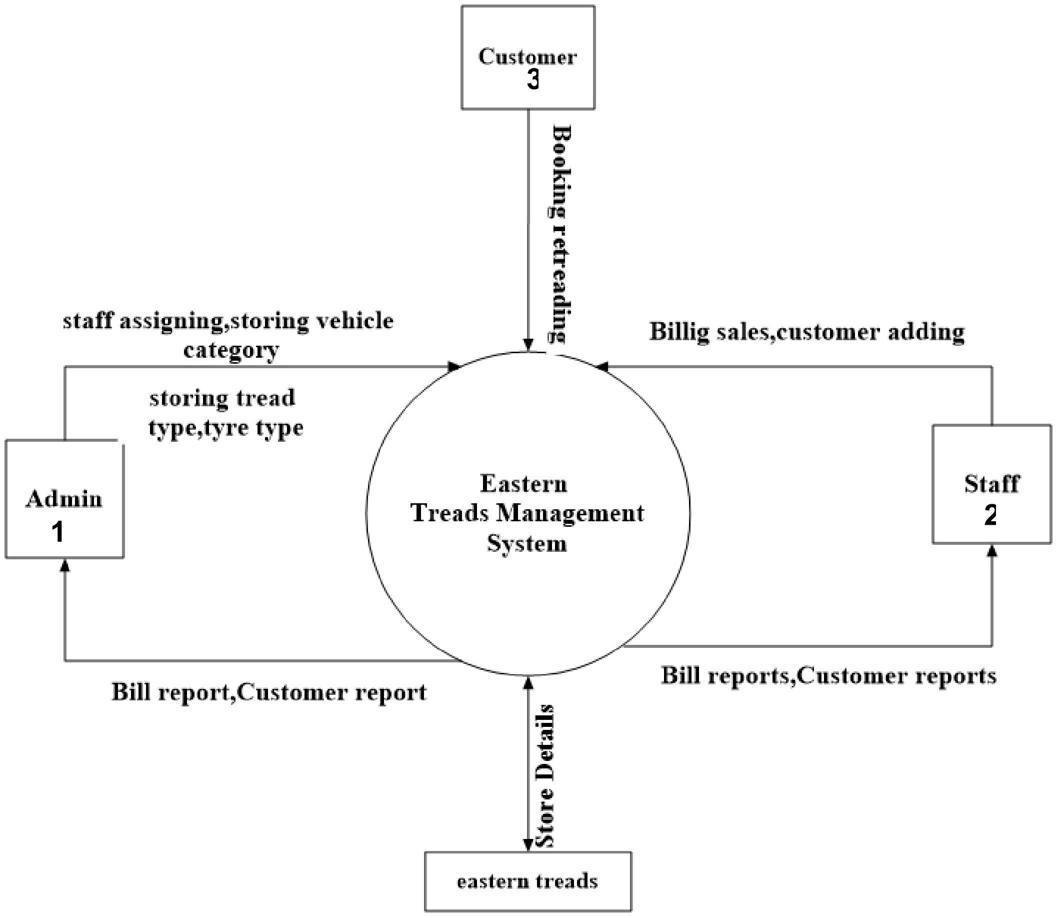
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

3.2.1 Context Level Data flow Diagram

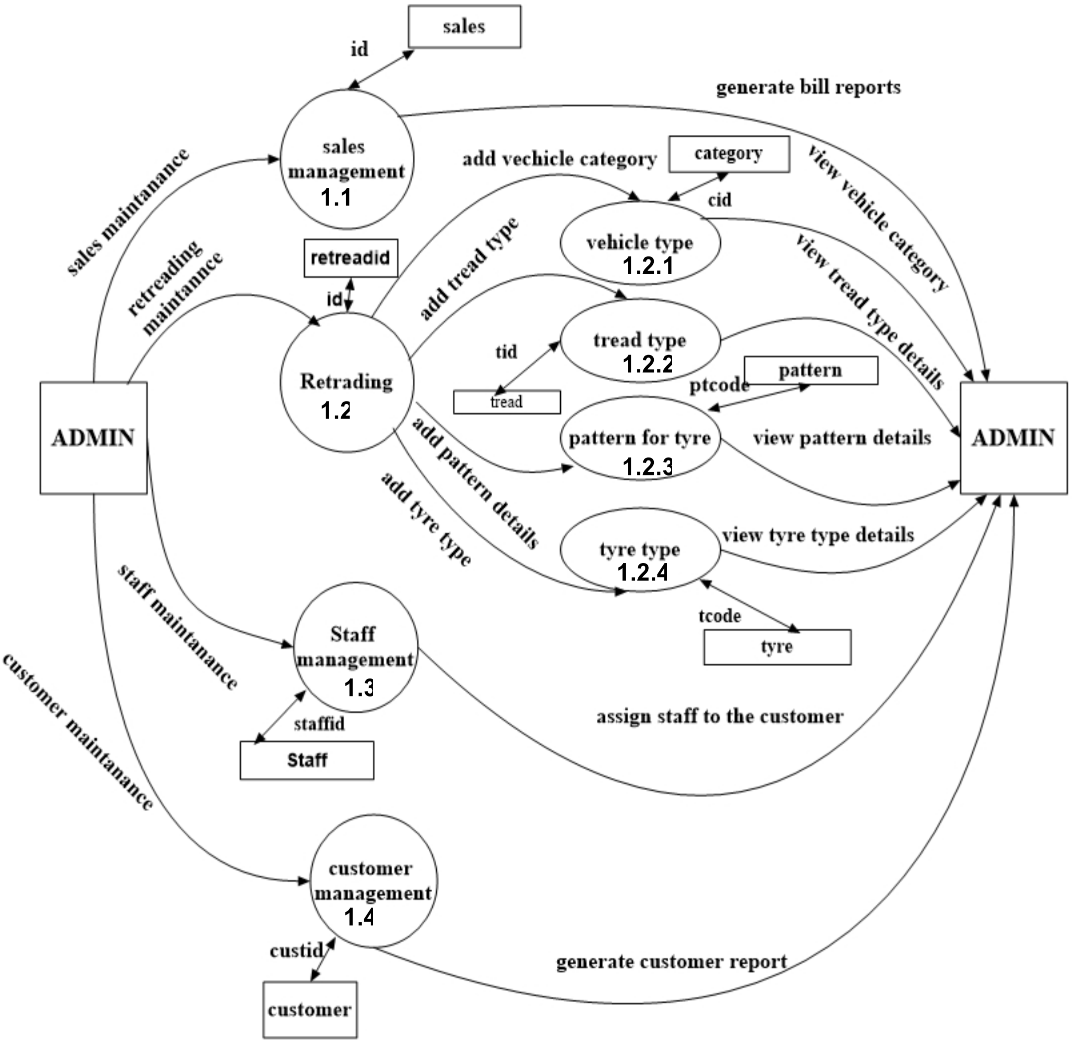
It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs. This process may be repeated at as many levels as necessary until the program at hand is well understood.


Customer
3
Booking retreading
staff assigning,storing vehicle category
storing tread type,tyre type
Billig sales,customer adding
Admin
1
Eastern Treads Management System
Staff
2
Bill report,Customer report
Bill reports,Customer reports
Store Details
eastern treads

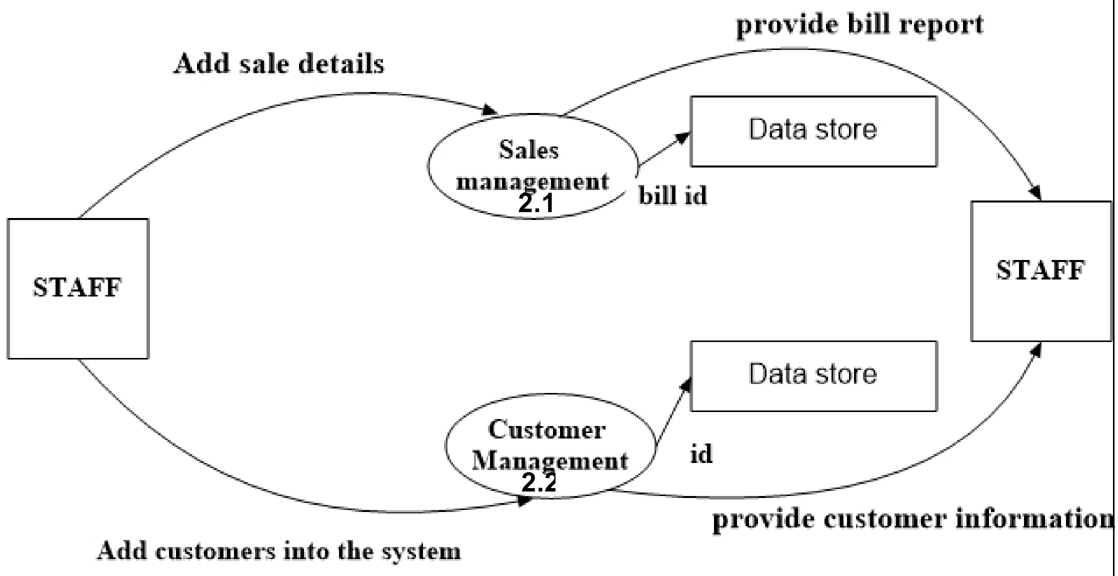
3.2.2 First Level Data Flow Diagram

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into sub processes.

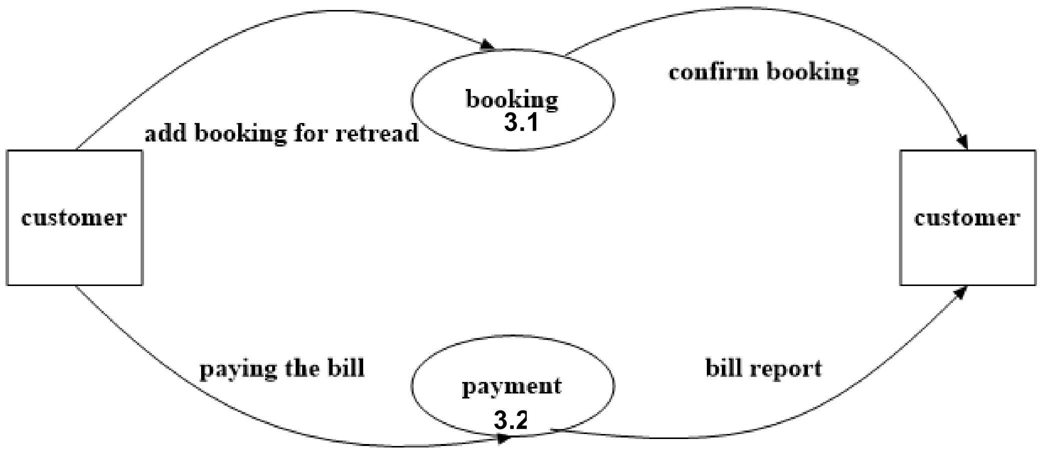
Admin



Staff

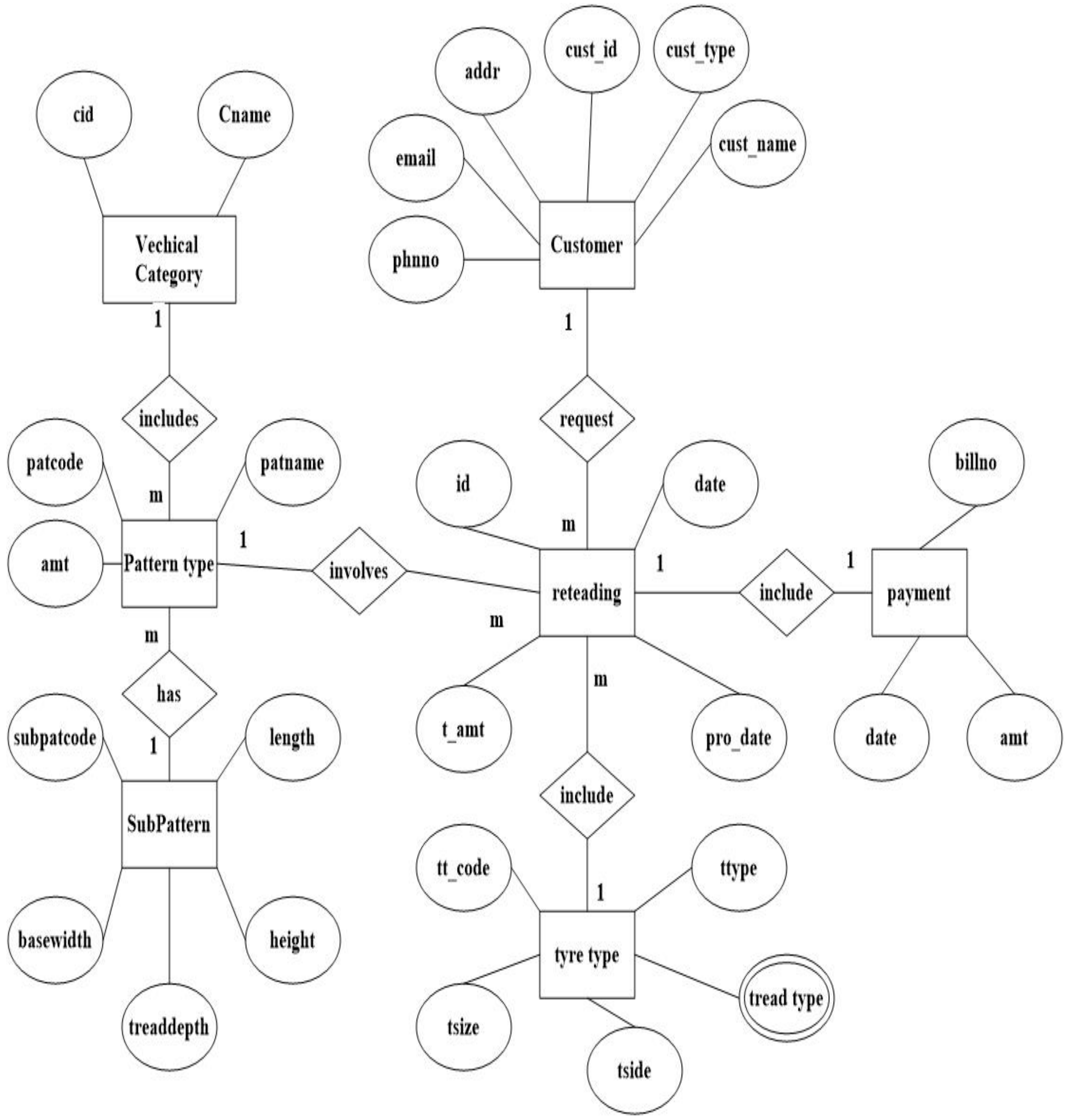


Customer



3.3 ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a Data base. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols that make it unique.



3.4 Table Description

Table no: 1

Category: The table contains data about the vehicle category such as categoryid, category name.

Primary key: cid

Field	Datatype	Description
cid	Varchar(50)	Vehicle category id
Cname	Varchar(50)	Vehicle category name

Table no: 2

Login: The table contains data about the login information such as username, password, usertype,

Primary Key: l_id

Field	Datatype	Description
l_id	int	Login id
Username	Varchar(50)	Username
Password	Varchar(50)	Password
Usertype	Varchar(50)	Usertype

Table no: 3

Pattern: The table contains data about the pattern such as pid, pname, v_id, img and amount

Primary key: pid

Field	Datatype	Description
Pid	int	Pattern id
Pname	Varchar(50)	Pattern Name
Img	Varchar(max)	Pattern Image
v_id	int	Vehicle Type
amount	Varchar(50)	Amount of the Pattern

Table no: 4

Customer: The table contains data about the customer such as name, customerid, address, mail id, contact number and mob

Primary key: custid

Field	Datatype	Description
cid	Varchar(15)	Consumer id
cname	Varchar(15)	Consumer name
Addr	Varchar(15)	Consumer address
City	Varchar(15)	City
Pin	Int	Pin code of customer
Email	Varchar(15)	Mail id
mob	Varchar(15)	Mobile number
Lid	Int	Login id

Table no: 5

Bill: The table contains data about the bill such as billno,date,sales id and amount

Primary key: billno

Field	Datatype	Description
Billno	varchar(50)	Bill no
Date	varchar(50)	Date of the bill produced
Sid	varchar(50)	Sales id
Amt	Int	Amount to be paid

Table no: 6

Sales: The table contains data about the sales of the pattern fields such as sid, trtype, pid, cid, qty, amt, cat_id

Primary key: sid

Field	Datatype	Description
Sid	int	Sales id
trtype	int	Tread type
pid	int	Pattern id
cid	Int	Customer id
Qty	Varchar(50)	Quantity of the tyre
Amt	Varchar(50)	Amount to be paid
Cat_id	Int	Vehicle category

Table no: 7

Tread:the table contain the details of the retreading type such as treadtype and tread name.

Primary key:tid

Field	Datatype	Description
Tid	varchar(50)	Tread id
Tname	varchar(50)	Tread name

Table no: 8

staff: the table contain the details of staff

Primary key:staffid

Field	Datatype	Description
Staffid	varchar(50)	Staff id
Staffname	varchar(50)	Staff name
Desig	varchar(50)	Designation
Gender	varchar(50)	Gender
Email	varchar(50)	Email
Dob	Date	Date of birth
Addre	varchar(50)	Address
Phone	varchar(50)	Phone number
Quali	varchar(50)	Qualification
Joiningdate	Date	Joiningdate

4.0 TESTING

4.1 Introduction

Software Testing is the process of executing a program or system with the intent of finding errors. Testing involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The scope of software testing includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to do and do what it needs to do. Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program.

Unit Testing

A unit is the smallest testable part of an application. Unit testing is a method of testing that verifies the individual units of source code are working properly. Rather than initially testing a program as a whole, testing is first focused on the smaller building blocks of the program. Unit testing eases the task of debugging and provide parallelism to program testing process by giving the opportunity to test multiple modules simultaneously. In this system the validity of fields in which data entered in each form and web form are checked. If the entered data are valid, then only further processing will take place.

Integration Testing

Data can be lost across an interface; one module can have an adverse effect on the other sub functions, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data.

The need for integrated testing is to find the overall system performance. While developing the system, each module is developed individually and integrated with present system. Modules are integrated by adding the module as a reference in other modules.

4.2 Test Cases

No.	Test Data	Cases	Expected result	Actual result	Remark
1.	Login to system	User enter * Invalid username password *Valid Password	Error Message: Invalid password -----	Error Message: Invalid password -----	As expected
2.	Phone number	*Enter phone number more than 10 digits *Valid phone number	Error Message: Incorrect format ----	Error Message: Incorrect format ----	As expected.
3.	Confirm password and password	*user enter different value User enter same value	Error Message: Confirm password and password not same -----	Error Message: Confirm password and password not same -----	As expected
4	Email	*Enter invalid email number for registration *Valid email number	Error Message: Incorrect format ----	Error Message: Incorrect format ----	As expected.

5.0 IMPLEMENTATION DETAILS

5.1 Introduction

Implementation is the stage in the project where the theoretical design is turned into a working system. Implementation is the final and important phase. The most critical stage for achieving a successful new system and forgiving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and only if it is found to working according to the specification. This method also offers the high security since it is implemented after identifying and handling all types of transactions while using the new system. Implementation phase include the training that should be provided for the chosen staff.

5.2 Installation procedure

The software can be installed in the following simple steps. In implementing machine

- Install sql server management studio 2012
- Install visual studio 2015

5.3 Implementation plan

In a direct cut over conversion, the old system is discarded and the new system takes over all at once, it is essentially turning the old system off and turning the new system on. This approach can be the least expensive of the different methods and can occur in the quickest time. A direct cut over conversion may be the only option if the old and new systems cannot co-existing any form. The greatest risk is the impact that errors and failures would have on the organization. The timing of this type of conversion is a key element of its success. The riskiest strategy for new systems installation, the direct cut over conversion can be low cost and the benefits of the new system can be realized without delay.

6.0 CONCLUSION

Eastern Treads Management System is a computerized management system. This management system has been developed to form whole management system including Admin, Staffs and Customer. This system will help the customer to do the treading process in online. This project is web based software that helps in storing, updating and retrieving information through many user-friendly menu driven modules.

In our existing systems all works are done manually. When a customer can do the treading process then he/she will be go to the organization directly and given the tyre to do the treading process. And also all the work in the treading process and customer interaction will be manually. This take time and effort.

In this computerized system, the above drawbacks are avoided. Admin and staff does all the organization process such as add pattern, view the customer, and do the treading process. Customer can now logon to the system and book the treading process they want to give the payment in online then the sales representative will come and collect the tyre and do the treading process and return back to the tyre to the corresponding customer. The system will give the invoice to the customer for the bill payment.

7.0 APPENDIX

7.1 APPENDIX A

Sample Source code/Pseudo code

Login page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

namespace ETMS
{
    public partial class login : System.Web.UI.Page
    {
        dboperation db = new dboperation();
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string un = TextBox1.Text;
            string pass = TextBox2.Text;
            string sql = "select * from login where username='" + un + "' and 
password='" + pass + "'";
            DataTable dt = db.exetable(sql);
            if(dt.Rows.Count>0)
            {
                Session["l_id"] = dt.Rows[0][0].ToString();
                if(dt.Rows[0][3].ToString()=="cust")
                {
                    Response.Redirect("~/customer/welcome.aspx");
                }
                if (dt.Rows[0][3].ToString() == "admin")
                {
                    Response.Redirect("~/admin/welcome.aspx");
                }
                if (dt.Rows[0][3].ToString() == "staff")
                {
                    Response.Redirect("~/staff/welcome.aspx");
                }
            }
            else
            {
                Response.Write("<script LANGUAGE='JavaScript' >alert('Invalid 
Username or Password')</script>");
            }
        }
    }}
```

Add pattern page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

namespace ETMS.admin
{
    public partial class welcome : System.Web.UI.Page
    { dboperation db = new dboperation();
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                db.fillddl("select * from category", DropDownList1, "cname",
"cid");
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            string name = TextBox1.Text;
            string image = FileUpload1.FileName;
            if(FileUpload1.HasFile)
            {
                try
                {
                    FileUpload1.SaveAs(Server.MapPath("~/patternpics/" +
FileUpload1.FileName));
                }
                catch(Exception ex)
                {
                    Response.Write(ex.Message);
                }
            }
            string amt = TextBox2.Text;
            string a= DropDownList1.SelectedItem.Value;
            //Response.Write(a);
            string sql = "insert into pattern1 values('" + name + "','" +image
+ "','" + amt + "','" + a + "')";
            //Response.Write(sql);
            int b = db.exenonquery(sql);
            if (b > 0)
            {
                Response.Write("SUCCESS");
            }
            else
            {
                Response.Write("NOT SUCCESS");
            }
        }

        protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
        {

        }
    }
}
```

Booking treading page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace ETMS.customer
{
    public partial class home : System.Web.UI.Page
    {
        dboperation db = new dboperation();

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {


                db.fillddl("select * from category", DropDownList1, "cname", 
"cid");

                string sql1 = "select * from pattern1 ";
                db.filldlist(sql1, DataList1);
            }
        }

        protected void DataList1_SelectedIndexChanged(object sender, EventArgs 
e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {


        }





        protected void Button2_Click(object sender, EventArgs e)
        {
            string aa = DropDownList1.SelectedItem.Value;
            string sql2 = "select * from pattern1 where v_id='" + aa + "'";
            db.filldlist(sql2, DataList1);
        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            if (DataList1.SelectedIndex != -1)
            {


                try
                {
                    string a = 
((Label)DataList1.SelectedItem.FindControl("Label2")).Text;
                    string pid = 
((Label)DataList1.SelectedItem.FindControl("Label3")).Text;
                    Response.Write(pid);
                    string tid = Session["tid"].ToString();
                    string qty = Session["qty"].ToString();
                    int amt = Convert.ToInt32(a);
                    int Qty = Convert.ToInt32(qty);
                    int tamt = Qty * amt;
                    string quantity = qty.ToString();
                    Session["pid"] = pid.ToString();
                    string lid = Session["l_id"].ToString();
                    string cat_id = DropDownList1.SelectedItem.Value;
                    string sql = "insert into sales values('" + tid + "','" + 
pid + "','" + lid + "','" + quantity + "','" + tamt + "','" + cat_id + "')";
                    int ab = db.exenonquery(sql);
                    if (ab > 0)
                    {
                         Session["date"] = DateTime.Now.ToString("d");

                        Response.Redirect("payment.aspx");
                        Response.Write("success");
                    }
                    else
                    {
                        Response.Write("error");
                    }
                }
                catch (Exception ex)
                {
                    Response.Write(ex.Message);
                }
            }
        }

      protected void DataList1_ItemCommand1(object source, 
DataListCommandEventArgs e)
        {
            DataList1.SelectedIndex = e.Item.ItemIndex;
        }
    }
}
```

Database connection code:

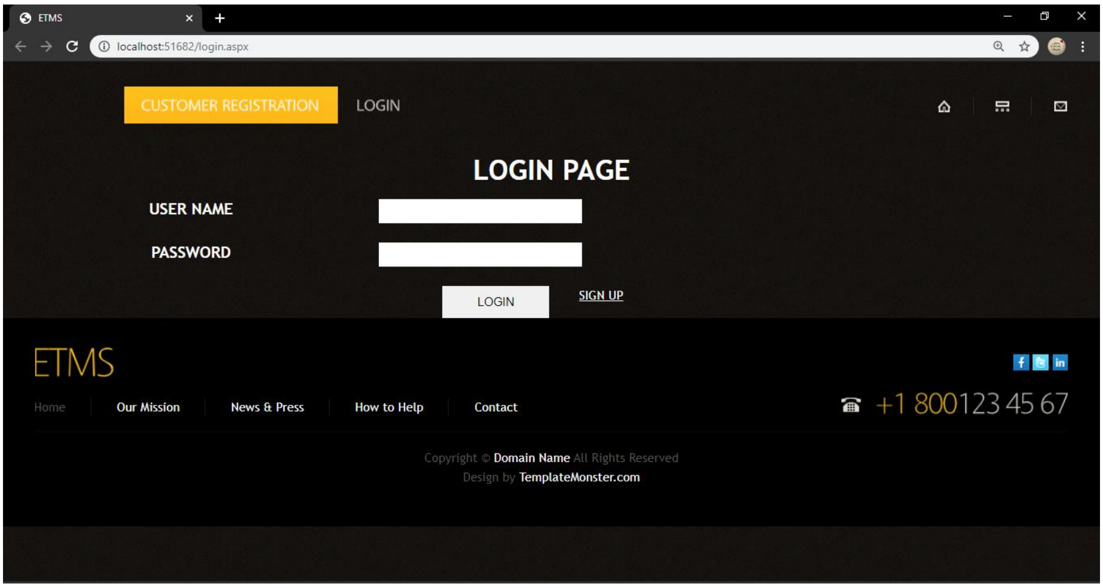
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.SqlClient;
using System.Data;
using System.Web.UI.WebControls;

namespace ETMS
{
    public class dboperation
    {
        public SqlConnection con;
        public SqlCommand cmd;
        public dboperation()
        {
            con = new SqlConnection("Data Source=DESKTOP-86UFVT0;Initial
Catalog=ETMS;Integrated Security=True");
            cmd = new SqlCommand();
            cmd.Connection = con;
        }
        public SqlConnection getcon()
        {
            if (con.State == ConnectionState.Closed)
            {
                con.Open();
            }
            return con;
        }
        public void dbclose()
        {
            con.Close();
        }
        public int exenonquery(string sql)
        {
            getcon();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = sql;
            int i = cmd.ExecuteNonQuery();
            return i;
        }
        public object exescalar(string sql)
        {
            getcon();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = sql;
            object ob = cmd.ExecuteScalar();
            return ob;
        }
        public SqlDataReader exereader(string sql)
        {
            getcon();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = sql;
            SqlDataReader dr = cmd.ExecuteReader();
            return dr;
        }
        public DataSet exedataset(string sql)
        {
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = sql;
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataSet ds = new DataSet();
            da.Fill(ds);
            return ds;
        }
        public DataTable exetable(string sql)
        {
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = sql;
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;
        }
        public void fillgrid(string sql, GridView dv)
        {
            dv.DataSource = exedataset(sql);
            dv.DataBind();
        }
        public void filldlist(string sql, DataList dl)
        {
            dl.DataSource = exedataset(sql);
            dl.DataBind();
        }
        public void fillddl(string sql, DropDownList ddl, string txt, string 
val)
        {
            ddl.DataTextField = txt;
            ddl.DataValueField = val;
            ddl.DataSource = exedataset(sql);
            ddl.DataBind();
            ddl.Items.Insert(0, new ListItem("........select.......", "0"));
        }
        public void fillrbl(string sql, RadioButtonList rbl, string txt, string 
value)
        {
            rbl.DataTextField = txt;
            rbl.DataValueField = value;
            rbl.DataSource = exedataset(sql);
            rbl.DataBind();
        }
    }
}
```

7.2 APPENDIX B

7.2.1 Screenshots

Login page



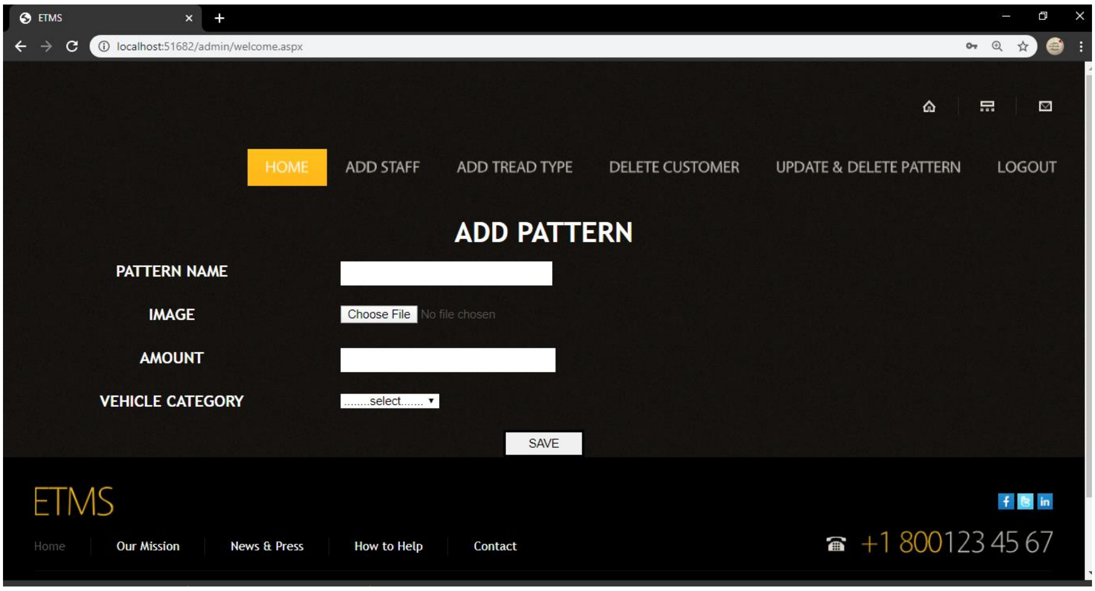
LOGIN PAGE

USER NAME

PASSWORD

Copyright © Domain Name All Rights Reserved
Design by TemplateMonster.com

Add pattern page



ADD PATTERN

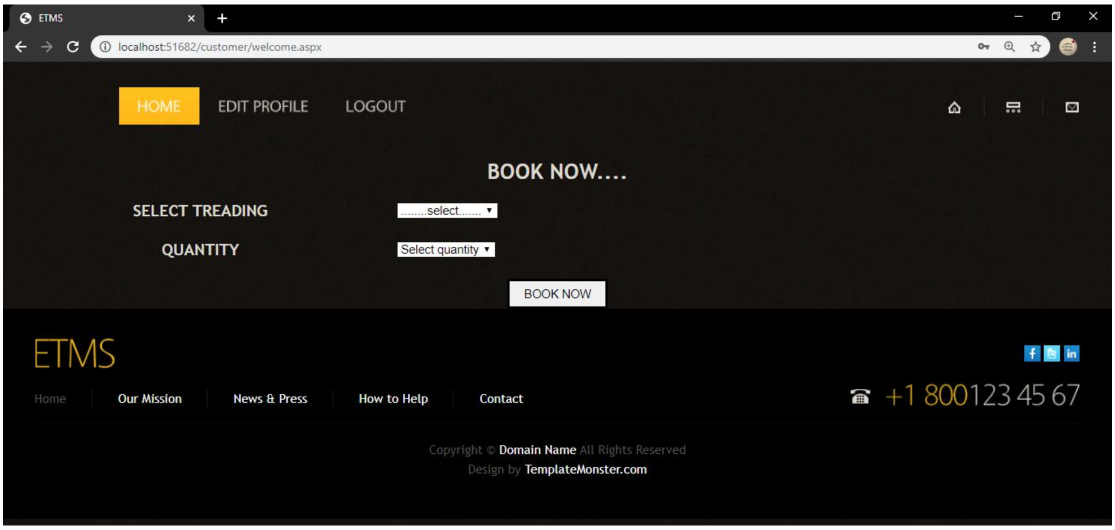
PATTERN NAME

IMAGE Choose File No file chosen

AMOUNT

VEHICLE CATEGORY ....select....... ▾

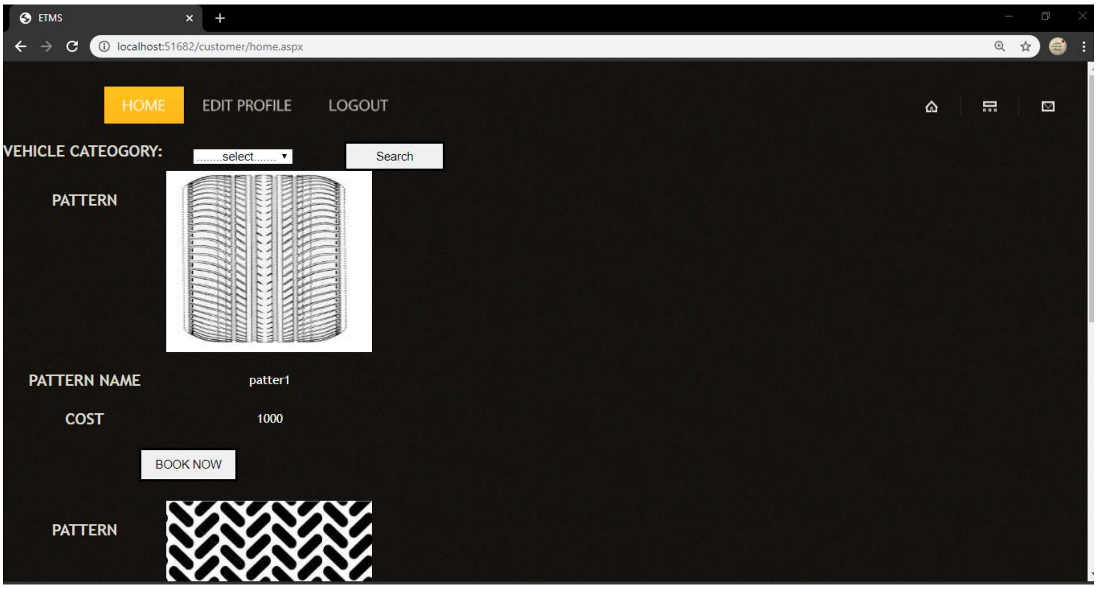
Customer home page



SELECT TREADING ....select.......

QUANTITY Select quantity

Booking page



VEHICLE CATEOGORY: ....select.......

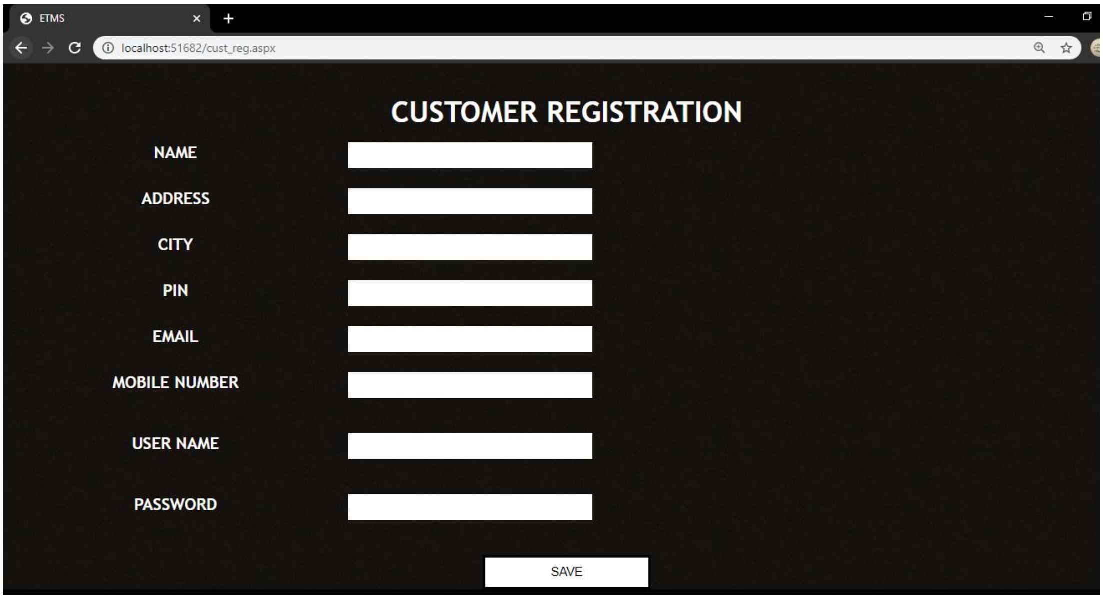
PATTERN

PATTERN NAME patter1

COST 1000

PATTERN

Customer registration



7.2 APPENDIX B

7.2.1 Bibliography

Books Referred

- Elias M. Award.(2005)System Analysis and Design. Galgotia. Second Edition.
- Roger S. Pressman.(2001)Software Engineering. McGraw Hill. Fifth Edition.
- Craig Larman.(2004) Applying UML and Patterns: An Introduction to Object oriented Analysis and Design And the Unified Process Pearson Education. Third Edition.
- Matthew MacDonald Beginning ASP.NET 4.5 in C_ (2012, Apress)

Internet Site

- “Retrieved from Software Developing Life Cycles”(March 2012). http://www.sdlc.ws.
- “ObjectModellingTechnique”(June1999). http://www.omt.org/technology/uml.
- “SoftwareTestingDictionary”(2014). http://www.tutorialspoint.com/sofware_testing/implementation
- “Oriented_Modeling_And_Design” https://www.researchgate.net/publication/2324983_Notes_On_Object\