

ITE 1942 ICT PROJECT

PROJECT REPORT

**Computer-Based Inventory Management System for Galle
Auto Parts Shop**

Name: Bandara SMKTS

Index: E2046014

Bachelor of Information Technology (External degree)

Faculty of Information Technology

University of Moratuwa

2022

Table of Contents

Table of Contents	i
List of tables.....	iii
List of Figures	iv
1 CHAPTER 01 – INTRODUCTION	1
1.1 Introduction.....	1
1.2 Background and Motivation.....	2
1.3 Problem In Brief	2
1.4 Aim	3
1.5 Objectives	3
1.6 Summary.....	3
2 CHAPTER 02 – RELATED WORK.....	4
2.1 Introduction.....	4
2.2 Auto Lanka Advertising & marketing (Pvt) ltd	4
2.3 Douglas & Sons (Pvt) ltd	5
2.4 Grease Monkey Automobile Parts	6
2.5 Original Equipment Manufacturer (OEM).....	7
2.6 Euro Car Parts	7
2.7 Care Distributers	8
Douglas & Sons (Pvt) ltd	9
Grease Monkey Automobile Parts	9
Euro Car Parts	9
Care Distributers	9
3 CHAPTER 03 – PROPOSED SOLUTION	11
3.1 Introduction.....	11
3.2 Functional Requirements	11
3.2.1 User Registration.....	11
3.2.2 Supplier Registration.....	11
3.2.3 Stock Management.....	11
3.2.4 Order Management	11

3.3	Non- Functional Requirements	12
3.3.1	Requirements for performance.....	12
3.3.2	Requirements for usability	12
3.3.3	Requirements for the User Interface	12
3.4	Flow Chart	13
3.4.1	Admin log In	13
3.4.2	Cashier Log in.....	14
3.4.3	Add product	15
3.4.4	Add Category	16
3.4.5	Add Supplier	17
3.5	Pseudocode	18
3.5.1	Cashier	19
3.5.2	Product	20
3.5.3	Supplier	21
3.6	Inputs.....	22
3.7	Process	22
3.8	Output information.....	23
4	SYSTEM - IMPLEMENTATION.....	24
4.1	INTERFACES	24
4.2	Log In.....	24
4.2.2	Admin Interface	26
4.2.3	Product Interfaces	30
4.2.4	Supplier Interface.....	37
4.2.5	Cashier Log In.....	39
4.2.6	Reports Module.....	45
4.2.7	ER -Diagram	46
4.2.8	Data Base	47
4.2.9	View Systems.....	49
REFERENCES.....		50

List of tables

Table 1 Summary Table.....	9
----------------------------	---

List of Figures

Figure 1 Auto Lanka Spare Parts	4
Figure 2 Douglas & sons (PVT) Ltd.....	5
Figure 3 Grease Monkey Automobile Parts.....	6
Figure 4 Original Equipment Manufacturer.....	7
Figure 5 Euro Car Parts.....	8
Figure 6 Care Distributers.....	9
Figure 7 Flowchart Admin.....	13
Figure 8 Flow Chart.....	13
Figure 9 Flowchart Cashier.....	14
Figure 10 Flowchart Add product.....	15
Figure 11 Flowchart Add Category	16
Figure 12 Flowchart Add Supplier.....	17
Figure 13 Pseudocode	18
Figure 14 Pseudocode	19
Figure 15 Pseudocode	20
Figure 16 pseudocode	21
Figure 17 login Interface.....	24
Figure 18 Admin Interface.....	26
Figure 19 Interface.....	30
Figure 20 Interface.....	32
Figure 21 Interface.....	35
Figure 22 Supplier Interface	37
Figure 23 Cashier Interface.....	39
Figure 24 Interface.....	45
Figure 25 ER Diagram	46
Figure 26 Brand Table	47
Figure 27 Category table.....	47
Figure 28 supplier Table	48
Figure 29 Product table.....	48
Figure 30 View systems.....	49

1 CHAPTER 01 – INTRODUCTION

1.1 Introduction

In the huge development of technology, as long as people still use an automobile, also the need for auto spare parts will continue to be high in demand. This business is more profitable because the replacement of damaged auto parts is inevitable if you want your vehicle to continue to be on the road.

There are so many customers in the whole day to get their necessary parts. We can't fiddle with their time. Because they have to wait for a long time to get their required services. As well as we don't have enough space to keep social distance in the covid pandemic situation. There are so many errors in daily transactions. Newly appointed workers are not familiar with the procedure of the shop. Without checking it is difficult to look for available and unavailable stock. There are so many errors in bills. In the manual system. The books and the papers lose and misplace. There are so many other problems which will be faced by both employees and customers. When we advisably look at these consequences, I thought to give a solution to those problems.

So, this proposed system, the Inventory Management system for the shop will give solutions for those problems. Here we can check the stock and inform the customers whether these parts are available or not. Customers don't want to waste Their time. With the help of this proposed solution, users will be able to check their bills accurately. Furthermore, users will be able can give the exact item / spare parts that he wants. We can check the warranty quickly because it can be added to the system. There won't be any changes in transactions.

1.2 Background and Motivation

Galle Auto Parts is the largest vehicle spare parts seller in the southern province. They are handling their system manually. They are selling spare parts for cars, buses, bikes, and other vehicles. They are trading brakes Pads, signal lights, silencers, headlights, body kits, engine parts, etc. They record their transactions in books. There are many issues with the manual process as employees are changing from time to time, there's a possibility to forget to record the transaction. Employees are also changing from time to time. Sometimes they forget to write the transactions. Even there are issues when calculating their profit at the end of the day, month, year. Hence, the author came up with this proposed solution which will help users to reduce their inventory management challenges.

The above problems can be solved by a newly invented system. In this proposed system, they can verify stock and inform the customers whether the requested parts are available or not. This system provides information on suppliers of the stock, type of stock supplied, the quantity of stock supplied, and date of stock supplied. They can check their bills accurately. They can check the warranties quickly because they can be added to the system. There won't be any changes in transactions. All the data can be saved. There won't be any changes if the employees changed. They can check the monthly profit. They can check the category that most of the parts are sold. Offers can be calculated one by one. Those are the major motivating factors for this project.

It is much more suitable a web-based system to the proposed solution, but it will cost a high budget than the company expected so it will be hard to build a web-based system due to high cost. Although the company suggested the need for an easily controllable system due to the less IT knowledge of the current employees. With regard to the all issues, I proposed a desktop application rather than implementing a web-based system.

1.3 Problem In Brief

With regarding to the current situation of the world, country it may affects for the covid-19 conditions and health instructions due to creating long ques around the shop by increasing touch points. Also, it has a large number of transactions according to that it may increase the paperwork. The requirement of the transaction details to the processes such as analyzing data, decision making ...etc. it has to store the paper works in a different physical storage therefore it may create

some security issues because of the inability of taking backups. All of these problems are affecting to all stake holders of the company.

Furthermore, by designing the system's GUI in a simple and easy-to-use manner, we were able to meet the requirement for trained users. The software programs utilized to fulfill all of the requirements are Visual Studio C#/.NET framework and SQL server. The proposed system is intended to handle as much of the work of the spare parts management as possible. A SQL Server database and a C#-based GUI interface make up the described solution. The GUI frames are used to manage the proposed system's functioning, while the database is used to hold relevant data, as previously indicated.

1.4 Aim

- Aim of this project is to design and develop a computer-based Inventory Management System for Galle Auto Parts Shop.

1.5 Objectives

- All the transactions will be computerized.
- Can keep a backup.
- Can check the monthly income.
- Can check the available stocks and sold-out stocks.
- Can look for items that are highly sold.

1.6 Summary

The Background and Motivation of the proposed system, as well as a brief description of the problem to be solved and the proposed system's aim and objectives, are covered in this chapter.

In the last chapter, I have discussed the introduction in brief about the proposed system, and in chapter 02 it is to be discussed the similar systems and their functionalities as mentioned in websites with some image proofs.

2 CHAPTER 02 – RELATED WORK

2.1 Introduction

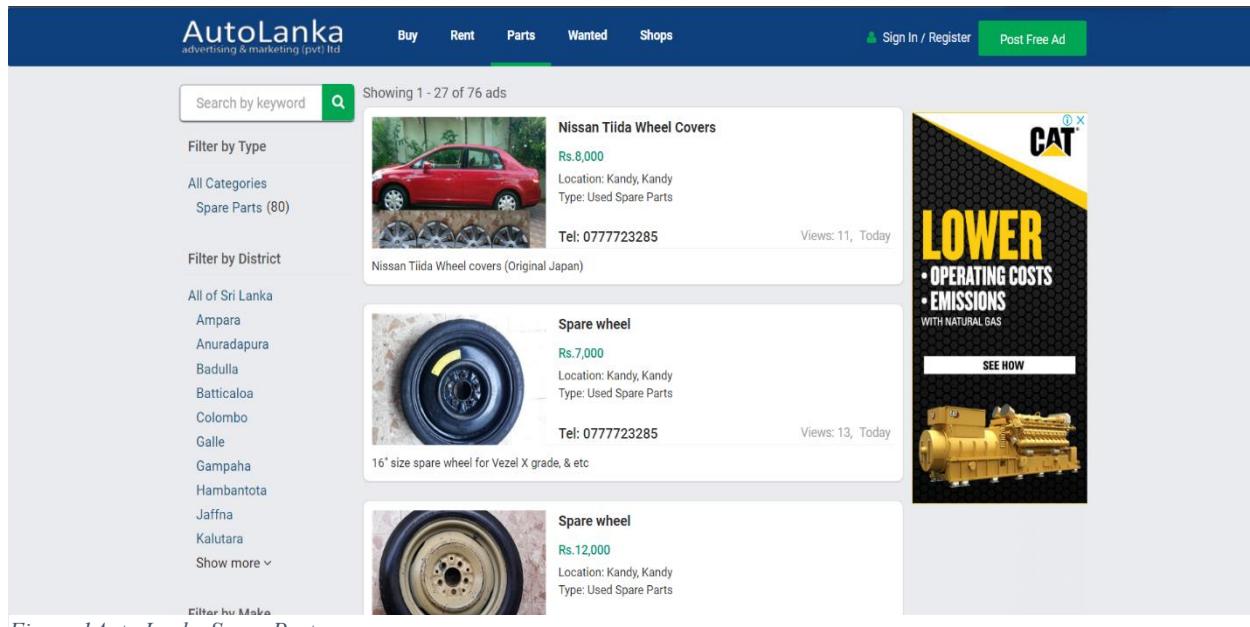
In this chapter I'll wish to describe the similar systems to my proposed system and the company's functionalities in common like login & registration processes, stock handling, online orders and delivery services. But there are some different functionalities differ according to other systems and unique to each.

I choose both four local and two foreign system to have some good idea with regarding their common functionalities and also the differences with unique features.

2.2 Auto Lanka Advertising & marketing (Pvt) ltd

Auto Lanka is an advertising and marketing company in which free service is provided mainly related to buying and selling vehicles. There are five categories in this website, those are Buy, Rent, Parts, Wanted and Shops.

Under the category of Buy, customers can filter vehicles by type, district, and brand(make) while Rent category has provided only filter by type and district. In the category of Parts, less advertisements (75) can be seen compared with other categories while shops category indicates



The screenshot shows the homepage of Auto Lanka. At the top, there is a dark blue header bar with the logo "AutoLanka advertising & marketing (pvt) ltd" on the left, and navigation links for "Buy", "Rent", "Parts", "Wanted", and "Shops" on the right. To the far right of the header are "Sign In / Register" and "Post Free Ad" buttons. Below the header, there is a search bar with the placeholder "Search by keyword" and a magnifying glass icon. To the right of the search bar, it says "Showing 1 - 27 of 76 ads". On the left side, there are two filter sections: "Filter by Type" (with options "All Categories" and "Spare Parts (80)") and "Filter by District" (listing districts from "All of Sri Lanka" down to "Show more"). The main content area displays three ads for "Nissan Tiida Wheel Covers" (Rs. 8,000), "Spare wheel" (Rs. 7,000), and another "Spare wheel" (Rs. 12,000). Each ad includes a small image of the item, its price, location (Kandy, Kandy), type (Used Spare Parts), contact number (Tel: 0777723285), and views (11, 13, 13 respectively). To the right of the ads, there is a large vertical advertisement for Caterpillar equipment with the text "LOWER OPERATING COSTS EMISSIONS WITH NATURAL GAS SEE HOW".

Figure 1 Auto Lanka Spare Parts

the highest number (2033) of advertisements. Other than these services, the details of top sellers and other shops are provided for the clients.

After referring to this website, Search Item and Customer Registration can be considered as functionalities. [1]

2.3 Douglas & Sons (Pvt) ltd

Douglas & Sons (Pvt) Ltd is a reputed company, which started its commercial operations in 1986. This company provides a variety of services including imports, sales, distributions, and awareness programmers. Apart from these, the company is organizing events such as blood donation camps, scholarships programmers and other events of contributions to the society.

In relation to the products, the company provides batteries, tires, auto parts, Agriculture Equipment, Industrial parts, motorcycle, Three wheel and soft toys. They have island wide authorized agents for the products and services. In addition, the DSL company proudly presents the award ceremonies. [2]

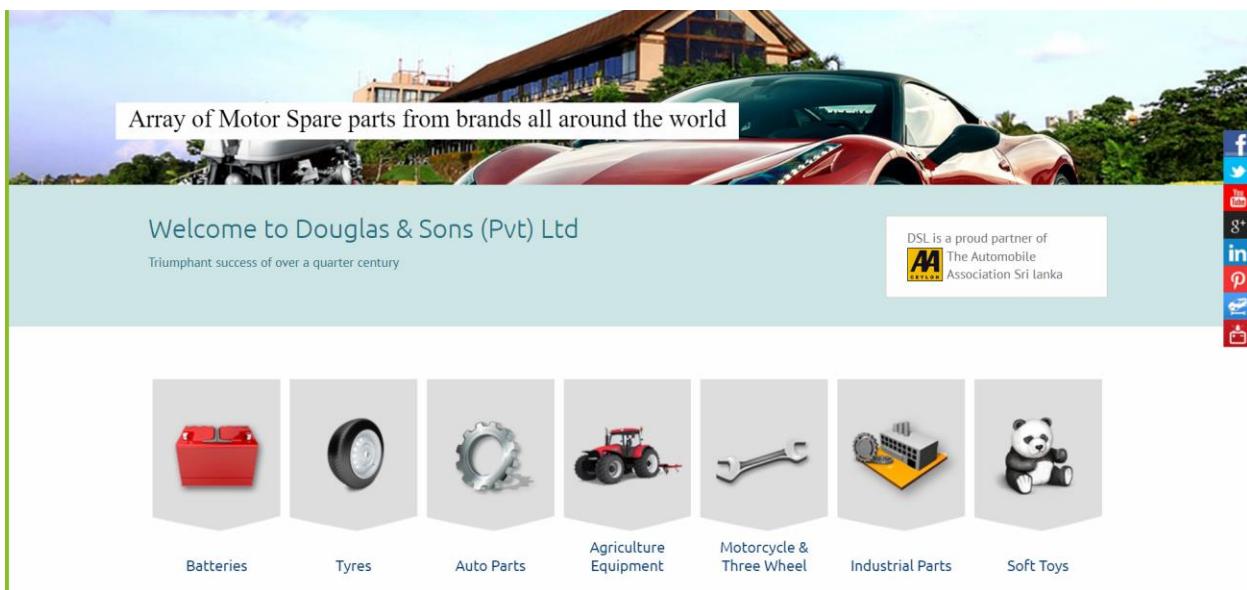


Figure 2 Douglas & sons (PVT) Ltd

2.4 Grease Monkey Automobile Parts

Greasemonkey.lk does away with the hassle of having to go from shop to shop looking for the perfect part and price by giving customers the ability to compare prices and products at their very fingertips! Greasemonkey.lk also provides unbelievable deals and after-sales services as well as product warranties to ensure you and your vehicle are well-taken care of.



**We welcome you to Grease Monkey,
your one-stop online platform for all things automotive!**

From car care products and spare parts to automobile vehicle accessories, Grease Monkey got everything you need



Figure 3 Grease Monkey Automobile Parts

At Greasemonkey.lk, value for money is important, which is why customers can choose from a wide variety of premium internationally recognized brands and top-quality products at competitive prices. Whatever you choose, you can expect nothing but the best and all delivered to your doorstep in just 01 – 03 days!

The system consists of categories which is classified according to a suitable manner and specially they are given a mechanic support with the system. Niece the customer requests a mechanic support via call, text or mail the system schedules a mechanic through the system in an appropriate manner to solve the customer problem easily. [3]

2.5 Original Equipment Manufacturer (OEM)

Oemparts.lk is a research based online automobile spare parts and accessories dealing web store in Sri Lanka. We deliver high quality car care and automobile accessories.

They are not only providing high-quality products, but also the establishment of a sound service system, to provide guidance and assistance for the problems and difficulties encountered with each client. They are focusing to full fill the much-needed online web store vacancy of Sri Lankan automobile industry. [4]

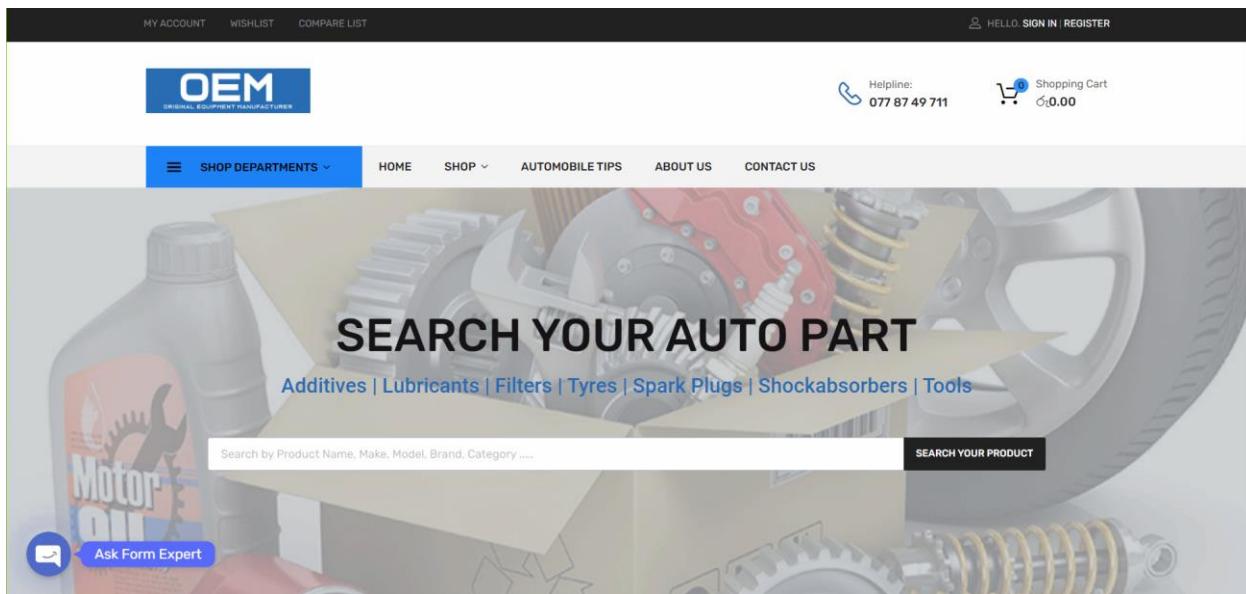


Figure 4 Original Equipment Manufacturer

They are focusing mainly two categories as Oils & fluid and tires rather than the other, but they sell them too. A customer can inquire about the parts and submit the inquire for the better reply. Also, they can ask from an expert with the message box facility.

2.6 Euro Car Parts

They have a web page named Euro Car Parts. It is the largest spare parts shop in Europe. They have a search by option with Keyword, Category, Brand, or Part. They have Spare parts items such as Car parts, Batteries, Oils, Wiper Blades, Bulbs, Accessories, Car covers, Tools, etc. They have free options as free UK delivery over £25 fit it for me .250+ Stores Nationwide, Free click Collect

here. It has a Winter Offer as Winter essentials:45% off parts. Customers can find their car parts easily by entering their number plate number or selecting their vehicle.

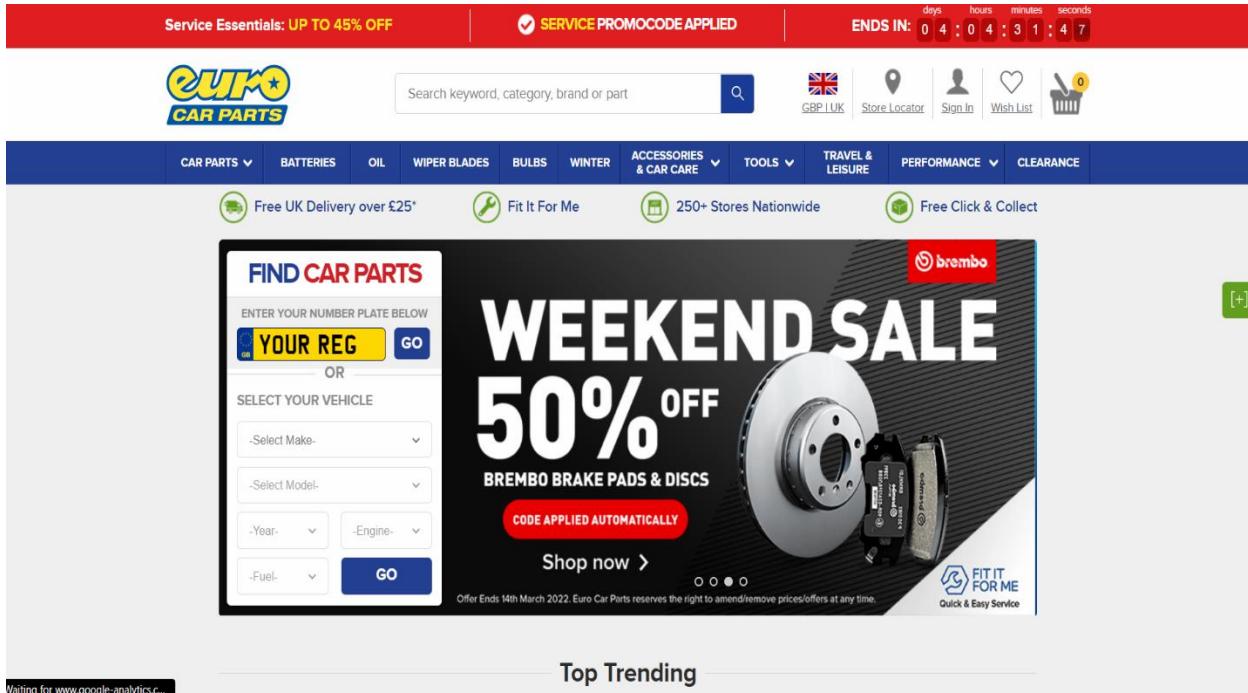


Figure 5 Euro Car Parts

Orders will have been delivered freely in UK. also they given many offers to the visitors via a promo code. Customers can feedback on their services and also they can download the mobile application for a better service. [5]

2.7 Care Distributors

They have a web page named Care Distributors. It is the Largest Spare Parts Shop in Australia. They have Free Shipping on Orders Over \$120. They have products such as workshop Equipment, Tools, Paints, Tapes, Plastic, etc. They have over 50 Brands including fusor, 3M, Sata, and Polymer

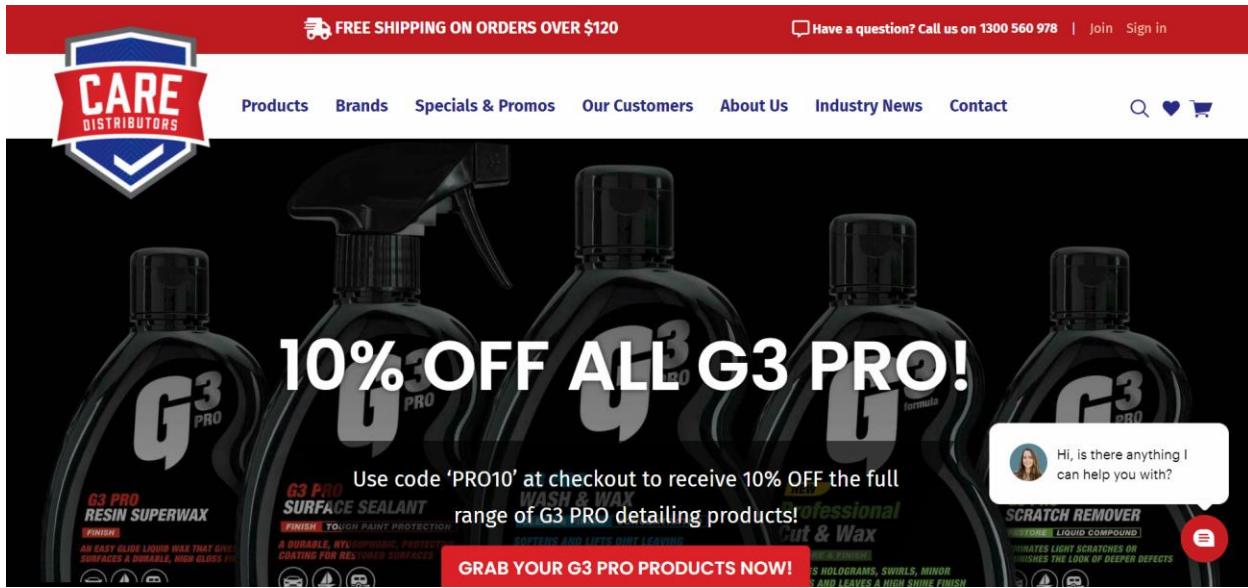


Figure 6 Care Distributors

Customer or the visitor can subscribe their site for all the latest promotions and news within the company. They serve mainly 04 categories such as Automotive, Marine, Industrial and household. Also, they can ask from an expert with the message box facility. [6]

Table 1 Summary Table

Features	Manual System	Proposed System	Auto Lanka	Douglas & Sons (Pvt) ltd	Grease Monkey Automobile Parts	OEM	Euro Car Parts	Care Distributors
			Local					Foreign
Flexible	No	Yes	No	No	Yes	Yes	No	No
Users	No	Yes	Yes	Yes	Yes	Yes	Yes	yes
Categories	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Subcategories	No	Yes	Yes	Yes	Yes	Yes	No	Yes
Search Items	No	Yes	Yes	No	Yes	Yes	Yes	Yes

Supplier Reg	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Delivery	Yes	Yes	No	No	No	Yes	No	Yes

3 CHAPTER 03 – PROPOSED SOLUTION

3.1 Introduction

Now I am in a good knowledge of the proposed system with regarding the similar systems and their all functionalities ...etc. In this chapter I wish to explain briefly about the functional and non-functional requirements of the proposed system, Input output and the processes of the system that I have proposed already and the relevant flowcharts and pseudocodes to the system too.

3.2 Functional Requirements

3.2.1 User Registration

- Delete User
- Activate/Inactivate

3.2.2 Supplier Registration

- Edit Supplier
- Add Supplier
- Delete Supplier

3.2.3 Stock Management

- Add Category
- Edit Category
- Add Brand
- Edit Brand
- Add product
- Edit Product
- Delete Product
- Delete Category
- Delete Brand

3.2.4 Order Management

- Date

- Order number
- Order description

3.3 Non- Functional Requirements

3.3.1 Requirements for performance

- The system must process events at an acceptable rate to allow the user to browse quickly.

3.3.2 Requirements for usability

- When the site is loaded, it should be dispersed among many servers to avoid a large release log.

3.3.3 Requirements for the User Interface

- The user interfaces must have recognized functions. The text in the user interface should be comprehensible and regain proper meaning.

3.4 Flow Chart

3.4.1 Admin log In

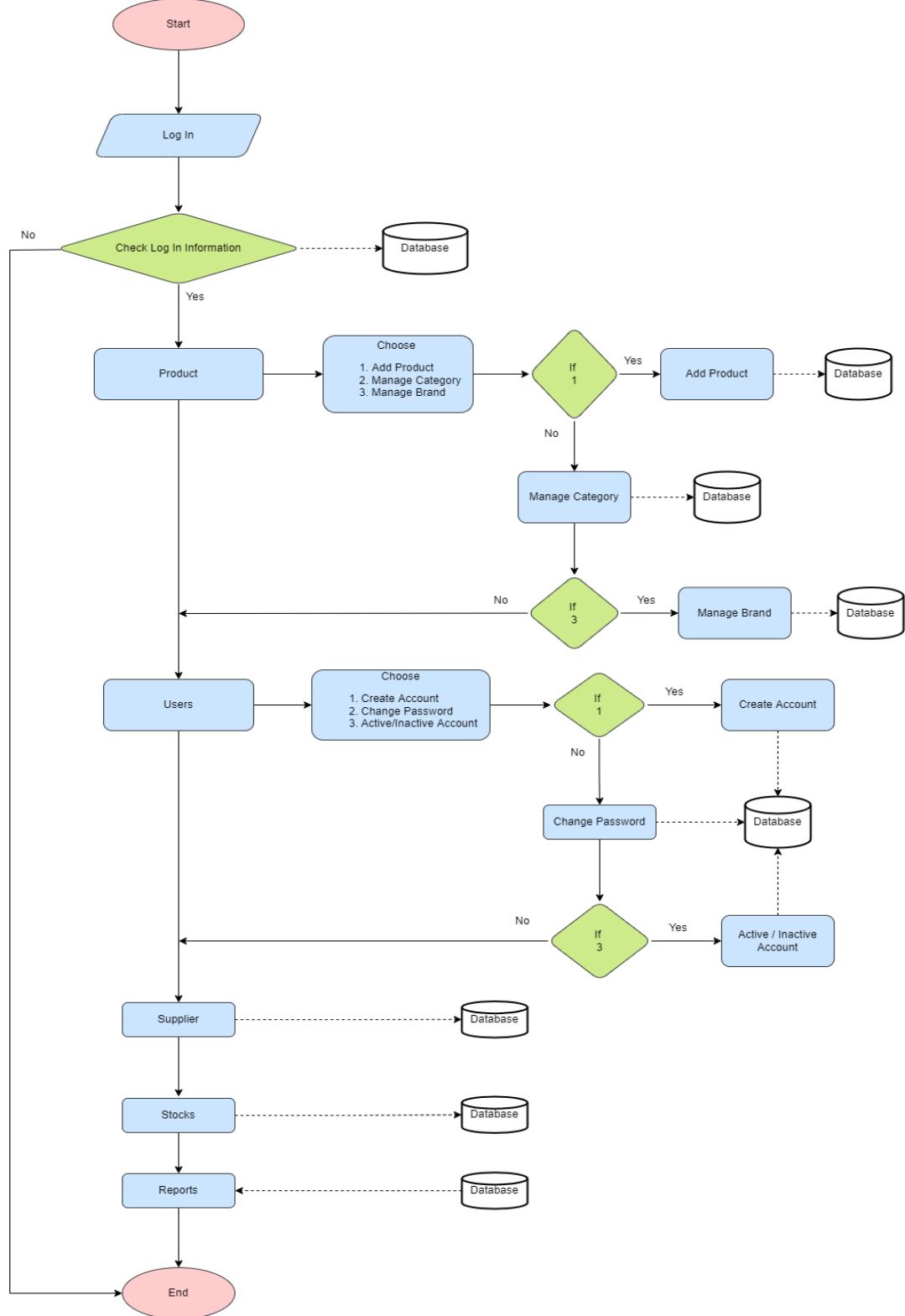


Figure 7 Flowchart Admin

3.4.2 Cashier Log in

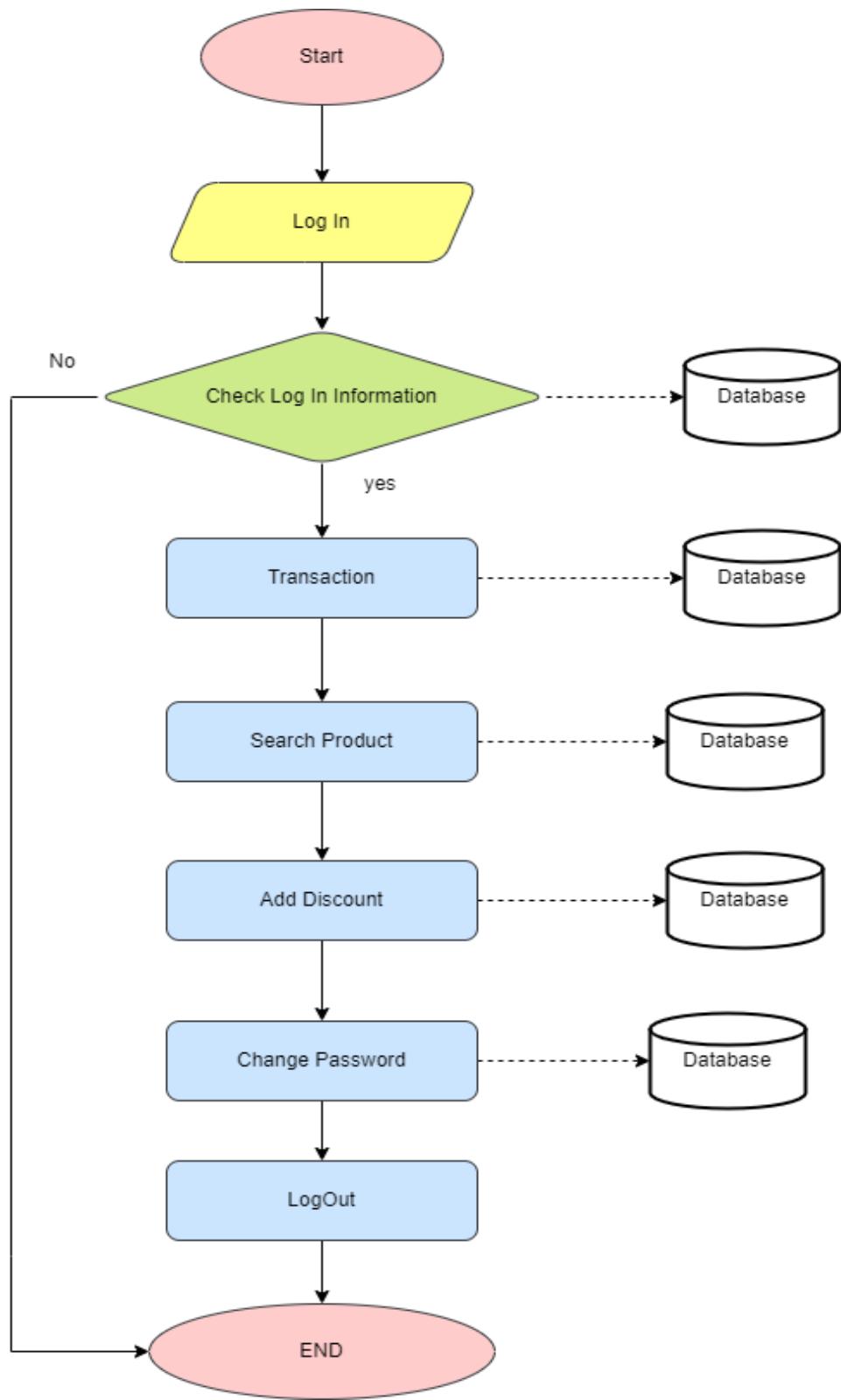


Figure 9 Flowchart Cashier

3.4.3 Add product

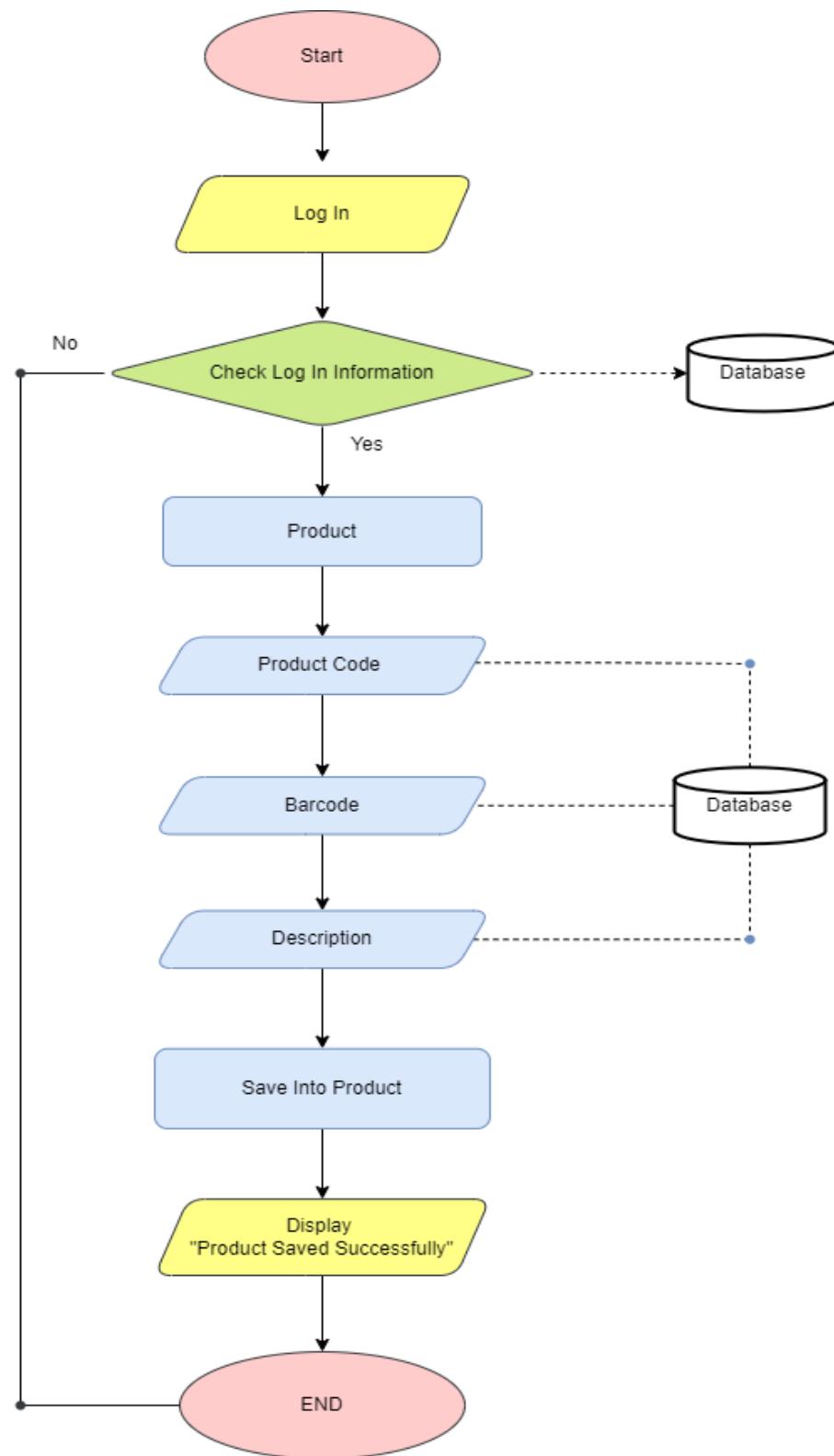


Figure 10 Flowchart Add product

3.4.4 Add Category

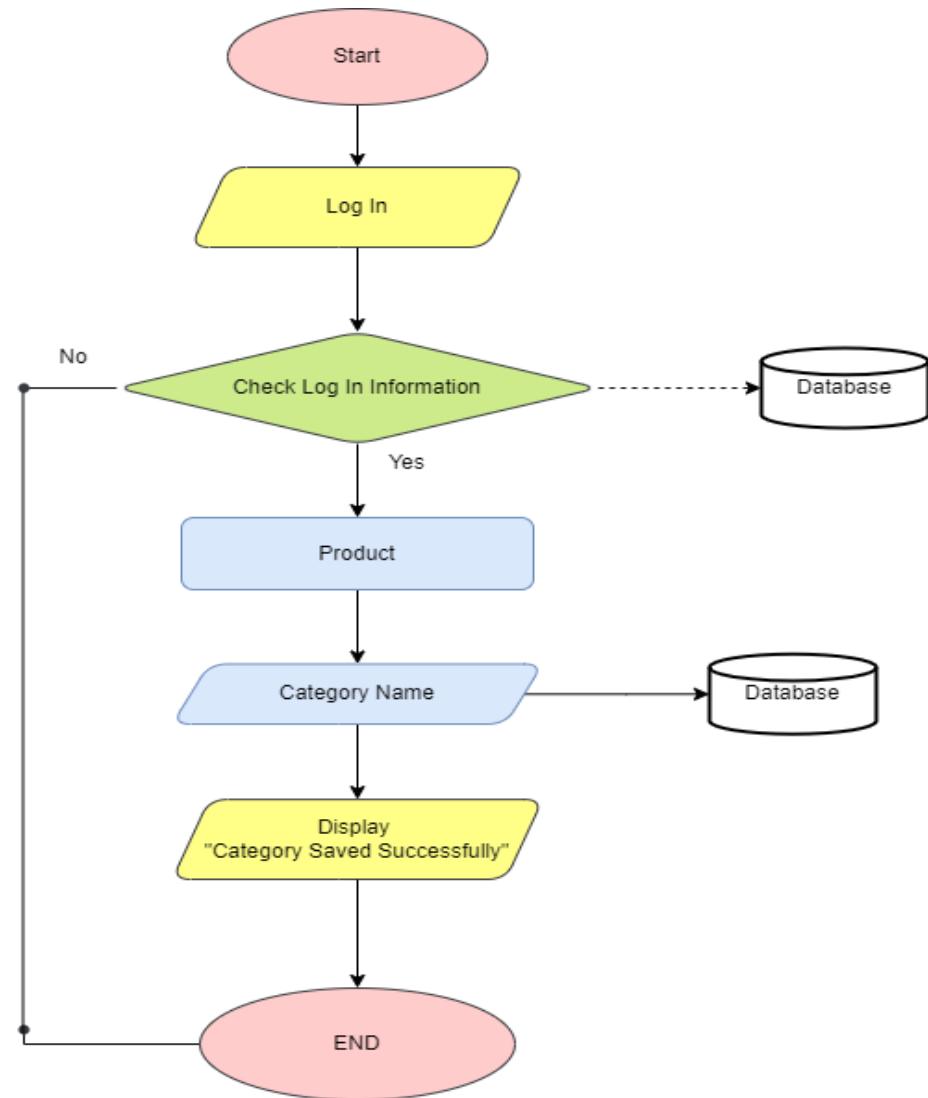


Figure 11 Flowchart Add Category

3.4.5 Add Supplier

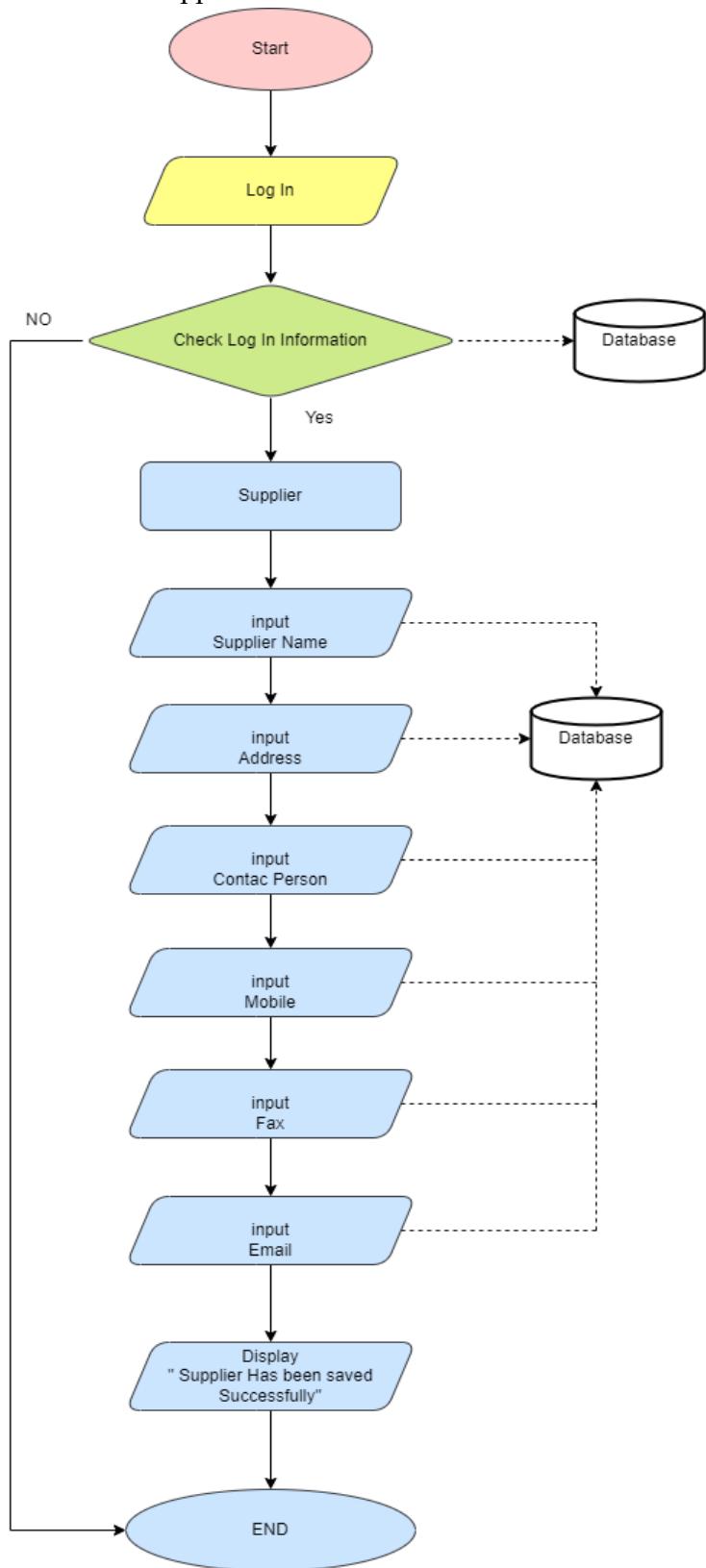


Figure 12 Flowchart Add Supplier

3.5 Pseudocode

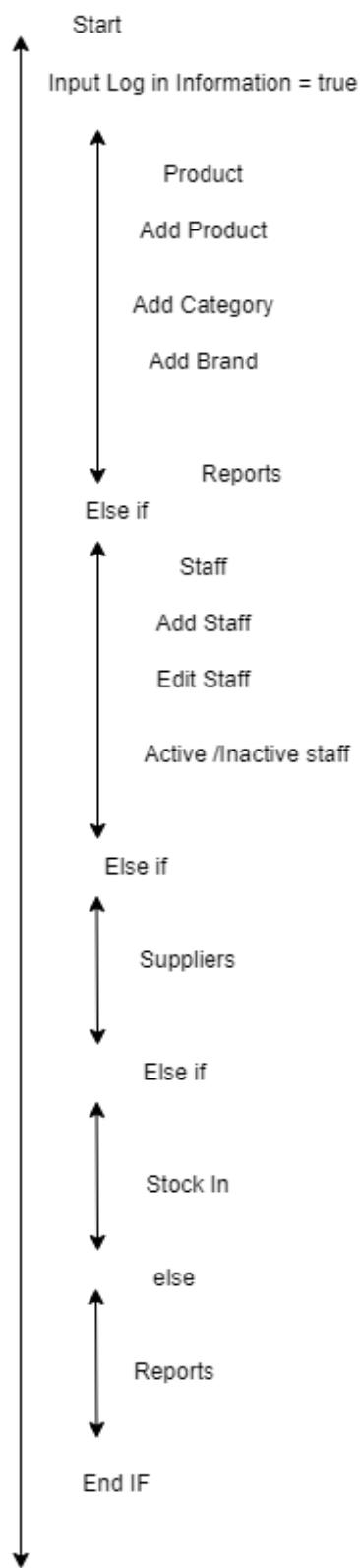


Figure 13 Pseudocode

3.5.1 Cashier

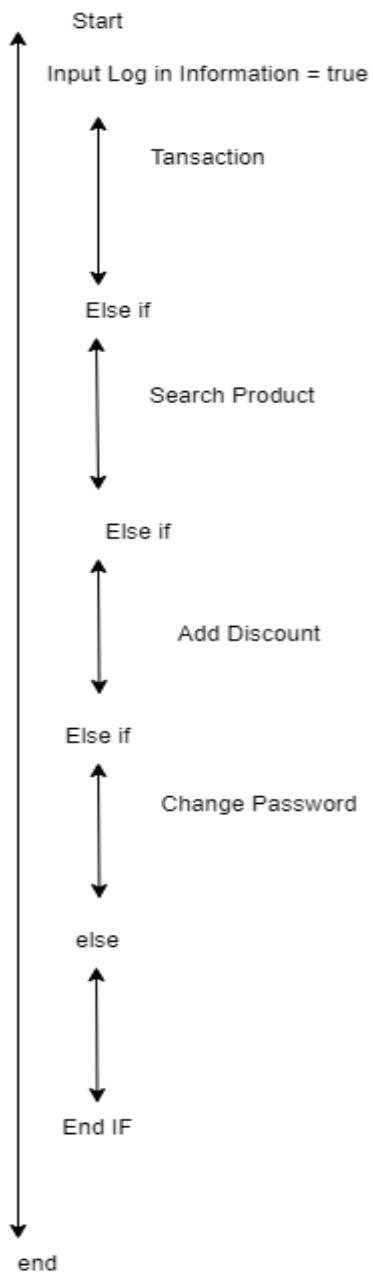


Figure 14 Pseudocode

3.5.2 Product

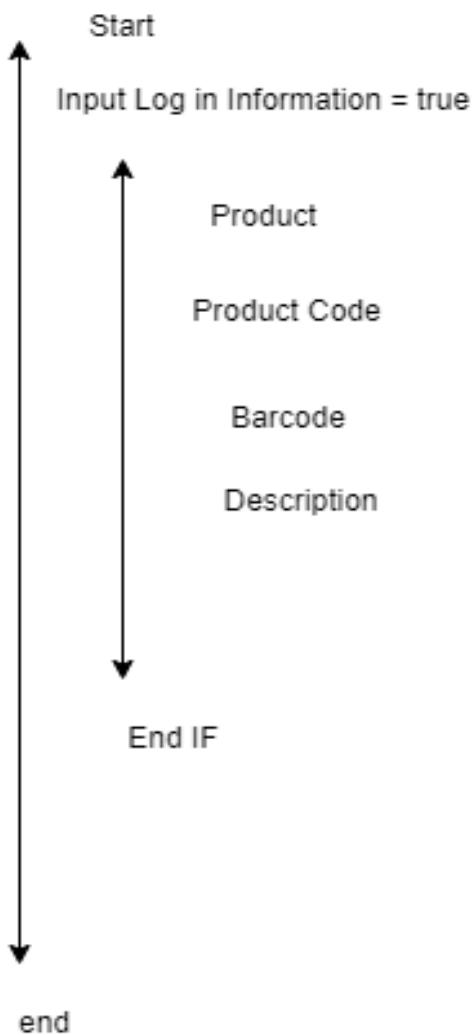


Figure 15 Pseudocode

3.5.3 Supplier

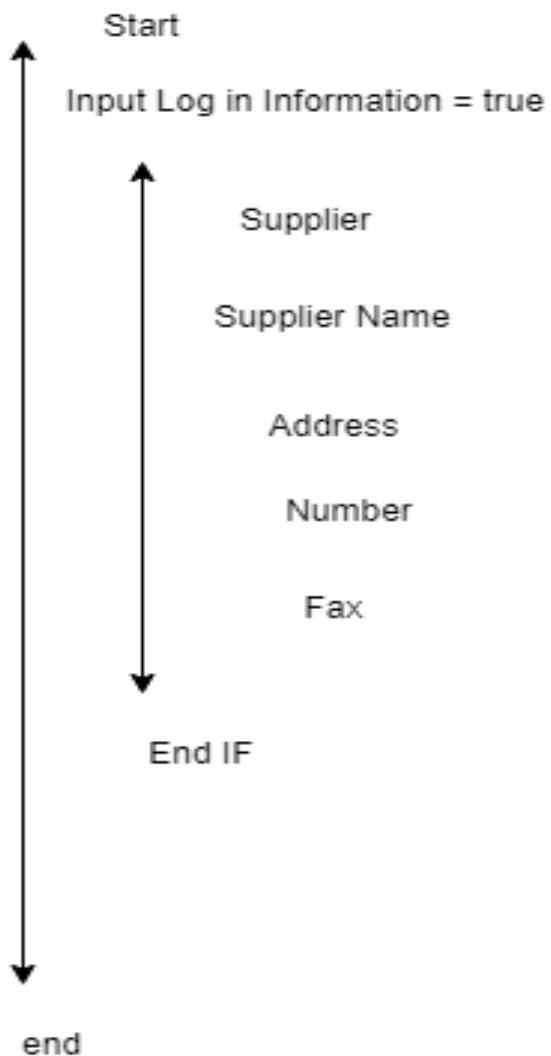


Figure 16 pseudocode

3.6 Inputs

- Staff Name
- Address
- Mobile Number
- Supplier Name
- Mobile Number
- Fax Number
- Address
- Contac Person
- Brand Name
- Date of Bought
- Add Discount
- Category details
- Add Category
- Quantity
- Item code
- Add Items
- Name
- Email
- Username
- Password
- Stock in Date
- Quantity
- Price

3.7 Process

- Handling inventory according to the categories
- Handling out of stock items and categories
- Preparing order details based on customer request
- Generating monthly reports
- Calculating bills
- Sale Date

3.8 Output information

- Most demanding item
- Least demanding item
- Monthly expenses
- Staff profile
- Outstanding staff member
- Available Stocks
- Monthly Profit
- Products

4 SYSTEM - IMPLEMENTATION

4.1 INTERFACES

4.2 Log In

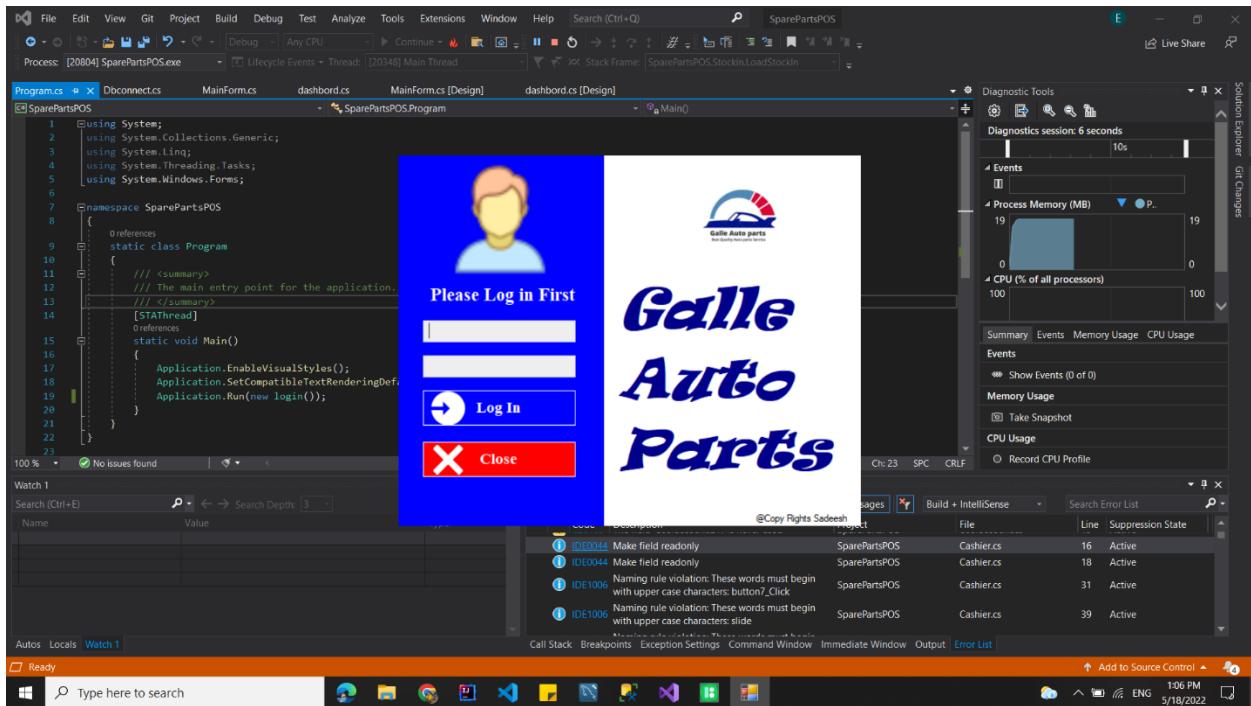


Figure 17 login Interface

4.2.1.1 Log in Code Segment

```
string _username = "", _name = "", _role = "";
try
{
    bool found;
    cn.Open();
    cm = new SqlCommand("Select * From usersTB Where username = @username and password=@password", cn);
    cm.Parameters.AddWithValue("@username", txtlogusername.Text);
    cm.Parameters.AddWithValue("@password", txtlogpasswrds.Text);
    DR = cm.ExecuteReader();
    DR.Read();
    if (DR.HasRows)
    {
        found = true;
        _username = DR["username"].ToString();
        _name = DR["name"].ToString();
        _role = DR["role"].ToString();
        _pass = DR["password"].ToString();
```

```

        _isactive = bool.Parse(DR["isactive"].ToString());

    }
    else
    {
        found = false;
    }
    DR.Close();
    cn.Close();
    if (found)
    {
        if (!_isactive)
        {
            MessageBox.Show("Account is Inactive.Unable to Login", "INACTIVE
ACCOUNT", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
        if (_role=="Cashier")
        {
            MessageBox.Show("Welcome "+_name + " | ", "ACCESS GRANTED",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtlogusername.Clear();
            txtlogpasswrds.Clear();
            this.Hide();
            Cashier cashier = new Cashier();
            cashier.lblUsername.Text = _username;
            cashier.lblName.Text = _name + " " + _role;
            cashier.ShowDialog();
        }
        else
        {
            MessageBox.Show("Welcome " + _name + " | ", "ACCESS GRANTED",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            txtlogusername.Clear();
            txtlogpasswrds.Clear();
            this.Hide();
            MainForm main = new MainForm();
            main.lblUsername.Text = _username;
            main.lblName.Text = _name;
            main.ShowDialog();
        }
    }
    else
    {
        MessageBox.Show("Invalid Username Or Password", "WARNING",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
catch (Exception ex)
{
    cn.Close();
}

```

4.2.2 Admin Interface

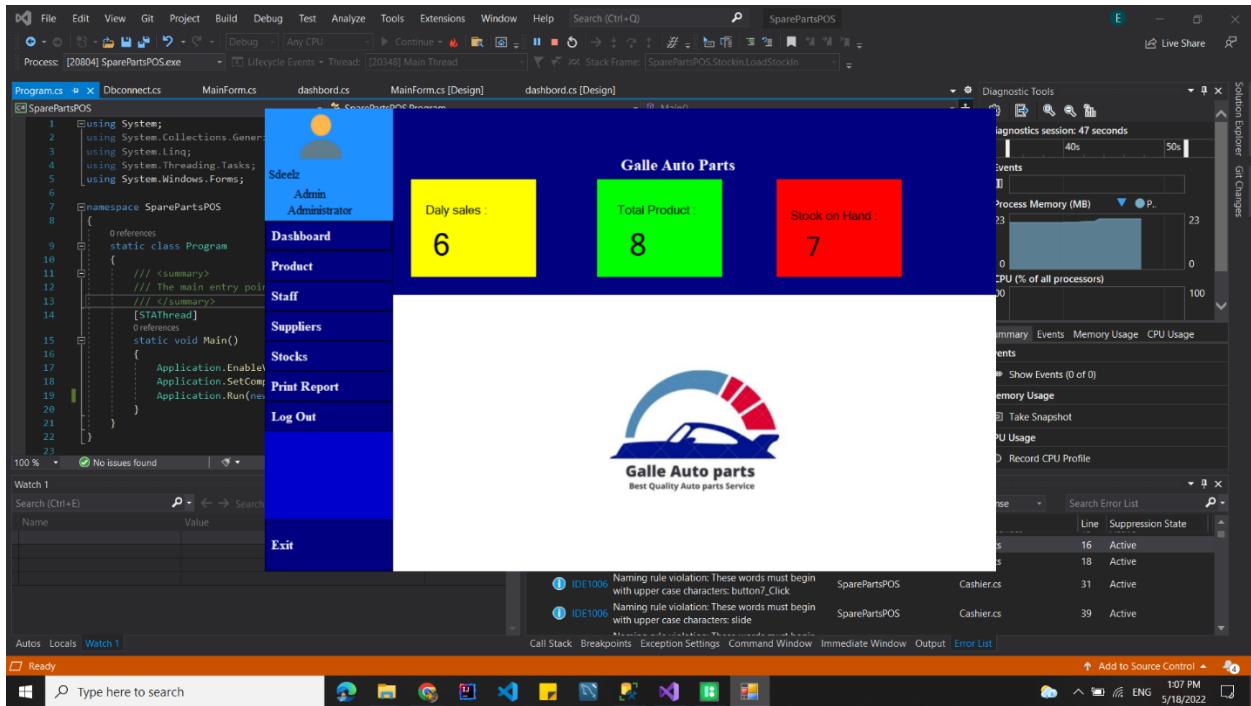


Figure 18 Admin Interface

4.2.2.1 Admin Code Segment

```
public partial class MainForm : Form
{
    SqlConnection cn = new SqlConnection();
    SqlCommand cm = new SqlCommand();
    Dbconnect dbcon = new Dbconnect();
    public MainForm()
    {
        InitializeComponent();
        customizeDesing();
        cn = new SqlConnection(dbcon.myConnection());
        cn.Open();
        //MessageBox.Show("Database is Connected");

    }
    private Form activeForm = null;
    public void openChildForm(Form childForm)
    {
        if (activeForm != null)
            activeForm.Close();
        activeForm = childForm;
        childForm.TopLevel = false;
        childForm.FormBorderStyle = FormBorderStyle.None;
        childForm.Dock = DockStyle.Fill;
        lblTitle.Text = childForm.Text;
        panelmain.Controls.Add(childForm);
        panelmain.Tag = childForm;
    }
}
```

```

        childForm.BringToFront();
        childForm.Show();

    }

#region panel_slide
private void customizeDesing()
{
    panel_Stocks.Visible = false;

}
private void hide_submenu()
{
    if (panel_Stocks.Visible == true)
        panel_Stocks.Visible = false;
}

}

private void show_submenu(Panel submenu)
{
    if (submenu.Visible == false)
    {
        hide_submenu();
        submenu.Visible = true;
    }
    else
        submenu.Visible = false;
}

#endregion panel_slide
private void Form1_Load(object sender, EventArgs e)
{

}

private void button9_Click(object sender, EventArgs e)
{

}

private void button4_Click(object sender, EventArgs e)
{

}

private void button6_Click(object sender, EventArgs e)
{

}

private void button5_Click(object sender, EventArgs e)
{

}

private void button8_Click(object sender, EventArgs e)

```

```

{
}

private void button7_Click(object sender, EventArgs e)
{
}

private void button2_Click(object sender, EventArgs e)
{
    showSubmenu(panel_Stocks);

}

private void btnstff_Click(object sender, EventArgs e)
{
    openChildForm(new Useraccount());
    //showSubmenu(panel_Staff);
}

private void button8_Click_1(object sender, EventArgs e)
{
    openChildForm(new supplier());
    // showSubmenu(panel_customer);
}

private void button10_Click(object sender, EventArgs e)
{
    openChildForm(new Stockin());
    // showSubmenu(panel_transaction);
}

private void btnbrand_Click(object sender, EventArgs e)
{
    openChildForm(new ManagestkFrom());
    hide_submenu();
}

private void btncatctegry_Click(object sender, EventArgs e)
{
    openChildForm(new Category());
    hide_submenu();
}

private void btnaddstk_Click(object sender, EventArgs e)
{
    openChildForm(new Product());
    hide_submenu();
}

private void button14_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Log Out Application ?", "LogOut",
    MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation) == DialogResult.Yes)

```

```

        {
            this.Hide();
            login Log = new login();
            Log.Show();
        }
    }

    private void btnDshbord_Click(object sender, EventArgs e)
    {
        openChildForm(new dashbord());
    }

    private void button15_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Exit Application ?", "EXIT", MessageBoxButtons.YesNo,
MessageBoxIcon.Exclamation) == DialogResult.Yes)
        {
            Application.Exit();
        }
    }

    private void button13_Click(object sender, EventArgs e)
    {
        openChildForm(new dashbord());
    }
}

```

4.2.3 Product Interfaces

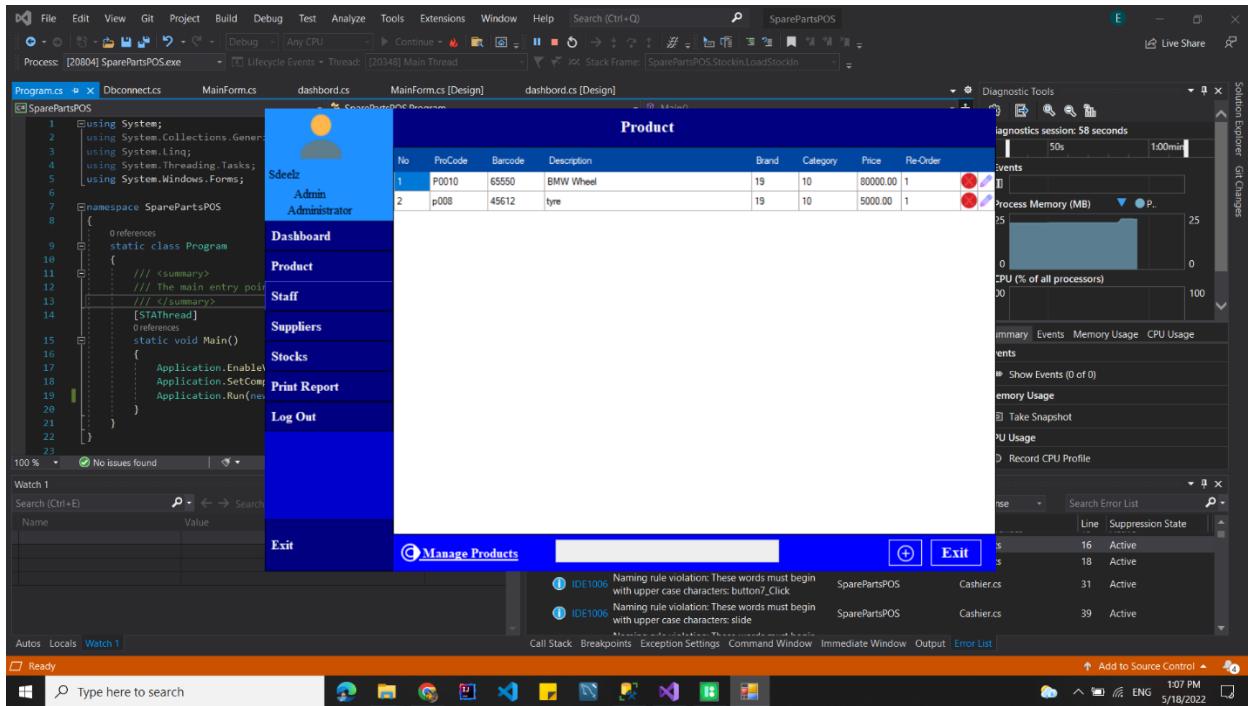


Figure 19 Interface

4.2.3.1 Product Code Segment

```
public partial class Product : Form
{
    SqlConnection cn = new SqlConnection();
    SqlCommand cm = new SqlCommand();
    Dbconnect dbcon = new Dbconnect();
    SqlDataReader DR;
    public Product()
    {
        InitializeComponent();
        cn = new SqlConnection(dbcon.myConnection());
        loadProduct();
    }
    public void loadProduct()
    {
        int i = 0;
        dgvProduct.Rows.Clear();
        cm = new SqlCommand("SELECT
p.ProCode,p.BarCode,p.ProDes,p.BrndID,p.CateID,p.ProPrice,p.ProReordr FROM productTB AS p
INNER JOIN brandTB AS b ON b.ID=p.BrndID INNER JOIN categoryTB AS c ON c.ID =p.CateID
WHERE CONCAT(p.ProDes,p.BrndID,p.CateID) LIKE '%" + txtsearch.Text+"%', cn);
        cn.Open();
        DR = cm.ExecuteReader();
        while (DR.Read())
        {
```

```

        i++;
        dgvProduct.Rows.Add(i, DR[0].ToString(), DR[1].ToString(),
DR[2].ToString(), DR[3].ToString(), DR[4].ToString(), DR[5].ToString(),
DR[6].ToString());

    }
    DR.Close();
    cn.Close();
}

private void btnadd_Click(object sender, EventArgs e)
{
    ProductModule promodule = new ProductModule(this);
    promodule.ShowDialog();
}

private void dgvProduct_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    string colName = dgvProduct.Columns[e.ColumnIndex].Name;
    if (colName == "Edit")
    {
        ProductModule product = new ProductModule(this);
        product.txtproductCode.Text =
dgvProduct.Rows[e.RowIndex].Cells[1].Value.ToString();
        product.txtbarcode.Text =
dgvProduct.Rows[e.RowIndex].Cells[2].Value.ToString();
        product.txtDescrp.Text =
dgvProduct.Rows[e.RowIndex].Cells[3].Value.ToString();
        product.cmbBrand.Text =
dgvProduct.Rows[e.RowIndex].Cells[4].Value.ToString();
        product.cmbCategory.Text =
dgvProduct.Rows[e.RowIndex].Cells[5].Value.ToString();
        product.txtprice.Text =
dgvProduct.Rows[e.RowIndex].Cells[6].Value.ToString();
        product.numOrdr.Value =
int.Parse(dgvProduct.Rows[e.RowIndex].Cells[7].Value.ToString());

        product.txtproductCode.Enabled = false;
        product.btnSave.Enabled = false;
        product.btnupdate.Enabled = true;
        product.ShowDialog();
    }
    else if (colName=="Delete")
    {
        if (MessageBox.Show("Are you want to Delete this Product ?", "Delete Record", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            cn.Open();
            cm = new SqlCommand("DELETE FROM productTB WHERE ProCode Like '" +
dgvProduct[1, e.RowIndex].Value.ToString() + "'", cn);
            cm.ExecuteNonQuery();
            cn.Close();
            MessageBox.Show("Producty has been successfully Deleted.", "Point Of Sales", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
}

```

```

        }
        loadProduct();

    }

    private void txtsearch_TextChanged(object sender, EventArgs e)
    {
        loadProduct();
    }
}

```

4.2.3.2 Product Module

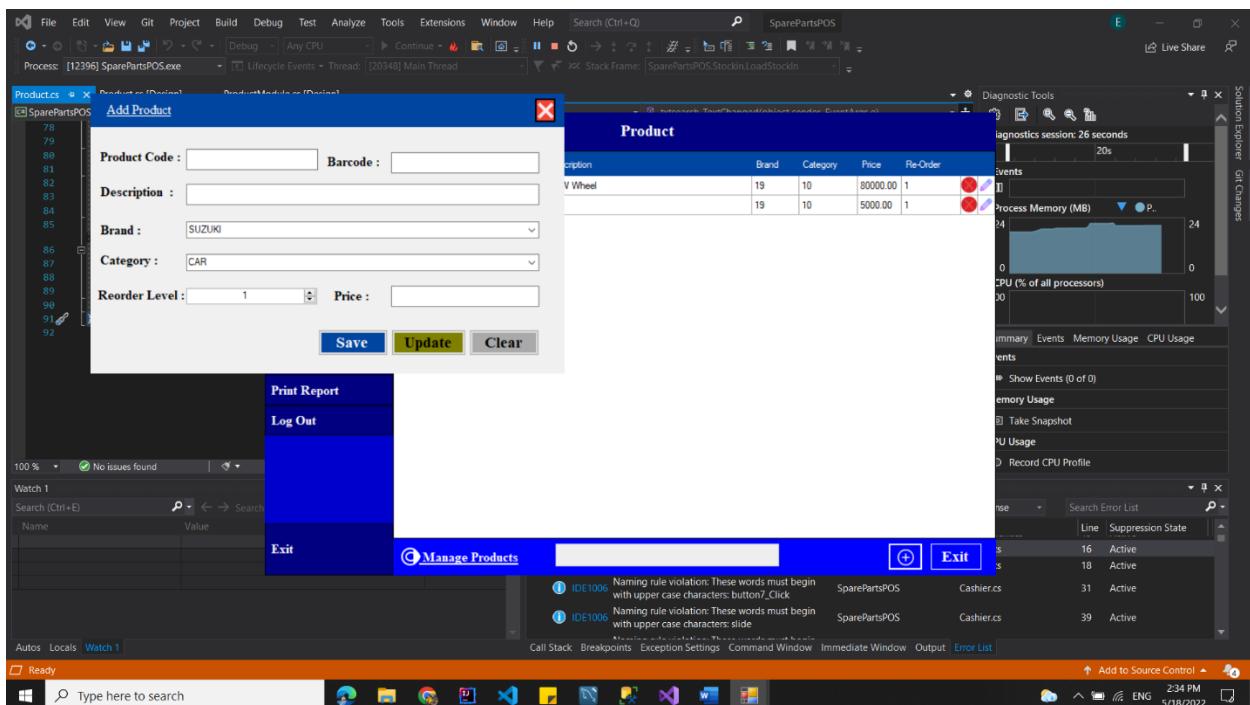


Figure 20 Interface

4.2.3.2.1 Product Module code Segment

```

public partial class ProductModule : Form
{
    SqlConnection cn = new SqlConnection();
    SqlCommand cm = new SqlCommand();
    Dbconnect dbcon = new Dbconnect();
    string stitle = "Point Of Sales";
    Product product;
    public ProductModule(Product pd)
    {
        InitializeComponent();
        product = pd;
    }
}

```

```

        cn = new SqlConnection(dbcon.myConnection());
        loadBrand();
        loadCategory();
    }
    public void loadCategory()
    {
        cmbCategory.Items.Clear();
        cmbCategory.DataSource = dbcon.GetTable("SELECT* FROM categoryTB");
        cmbCategory.DisplayMember = "Category";
        cmbCategory.ValueMember = "ID";
    }
    public void loadBrand()
    {
        cmbBrand.Items.Clear();
        cmbBrand.DataSource = dbcon.GetTable("SELECT* FROM brandTB");
        cmbBrand.DisplayMember = "Brand";
        cmbBrand.ValueMember = "ID";
    }
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void ProductModule_Load(object sender, EventArgs e)
{
}

private void button1_Click(object sender, EventArgs e)
{
    this.Dispose();
}
public void Clear()
{
    txtproductCode.Clear();
    txtDescrp.Clear();
    txtprice.Clear();
    txtbarcode.Clear();
    cmbBrand.SelectedItem = 0;
    cmbCategory.SelectedItem = 0;
    numOrdr.Value = 1;

    txtproductCode.Enabled = true;
    txtproductCode.Focus();
    btnsave.Enabled = true;
    btnupdate.Enabled = false;
}

private void btnsave_Click(object sender, EventArgs e)
{
    try
    {
        if (MessageBox.Show("Are you want to Add this Product ?", "Add product",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {

```

```

        cm = new SqlCommand("INSERT INTO productTB
(ProCode,BarCode,ProDes,BrndID,CateID,ProPrice,ProReordr)VALUES(@ProCode,@BarCode,@ProDes
,@BrndID,@CateID,@ProPrice,@ProReordr)", cn);
        cm.Parameters.AddWithValue("@ProCode", txtproductCode.Text);
        cm.Parameters.AddWithValue("@BarCode", txtbarcode.Text);
        cm.Parameters.AddWithValue("@ProDes", txtDescrp.Text);
        cm.Parameters.AddWithValue("@BrndID", cmbBrand.SelectedValue);
        cm.Parameters.AddWithValue("@CateID", cmbCategory.SelectedValue);
        cm.Parameters.AddWithValue("@ProPrice", double.Parse
(txtprice.Text));
        cm.Parameters.AddWithValue("@ProReordr", numOrdr.Value);
        cn.Open();
        cm.ExecuteNonQuery();
        cn.Close();
        MessageBox.Show("Product has been Successfully Saved", stitle);
        Clear();
        product.loadProduct();

    }

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void btnclose_Click(object sender, EventArgs e)
{
    Clear();
}

private void btnupdate_Click(object sender, EventArgs e)
{
    try
    {
        if (MessageBox.Show("Are you want to Update this Product ?", "Update
Product", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            cm = new SqlCommand("UPDATE productTB SET
Barcode=@Barcode,ProDes=@ProDes,BrndID=@BrndID,CateID=@CateID,ProPrice=@ProPrice,ProReordr=@P
roReordr WHERE ProCode LIKE @ProCode ", cn);
            cm.Parameters.AddWithValue("@ProCode", txtproductCode.Text);
            cm.Parameters.AddWithValue("@Barcode", txtbarcode.Text);
            cm.Parameters.AddWithValue("@ProDes", txtDescrp.Text);
            cm.Parameters.AddWithValue("@BrndID", cmbBrand.SelectedValue);
            cm.Parameters.AddWithValue("@CateID", cmbCategory.SelectedValue);
            cm.Parameters.AddWithValue("@ProPrice", double.Parse(txtprice.Text));
            cm.Parameters.AddWithValue("@ProReordr", numOrdr.Value);
            cn.Open();
            cm.ExecuteNonQuery();
            cn.Close();
            MessageBox.Show("Product has been successfully Updated !", stitle);
            Clear();
            this.Dispose();
        }
    }
}

```

```

        }
        catch (Exception ex)
    {

```

4.2.3.2.2 Category Module

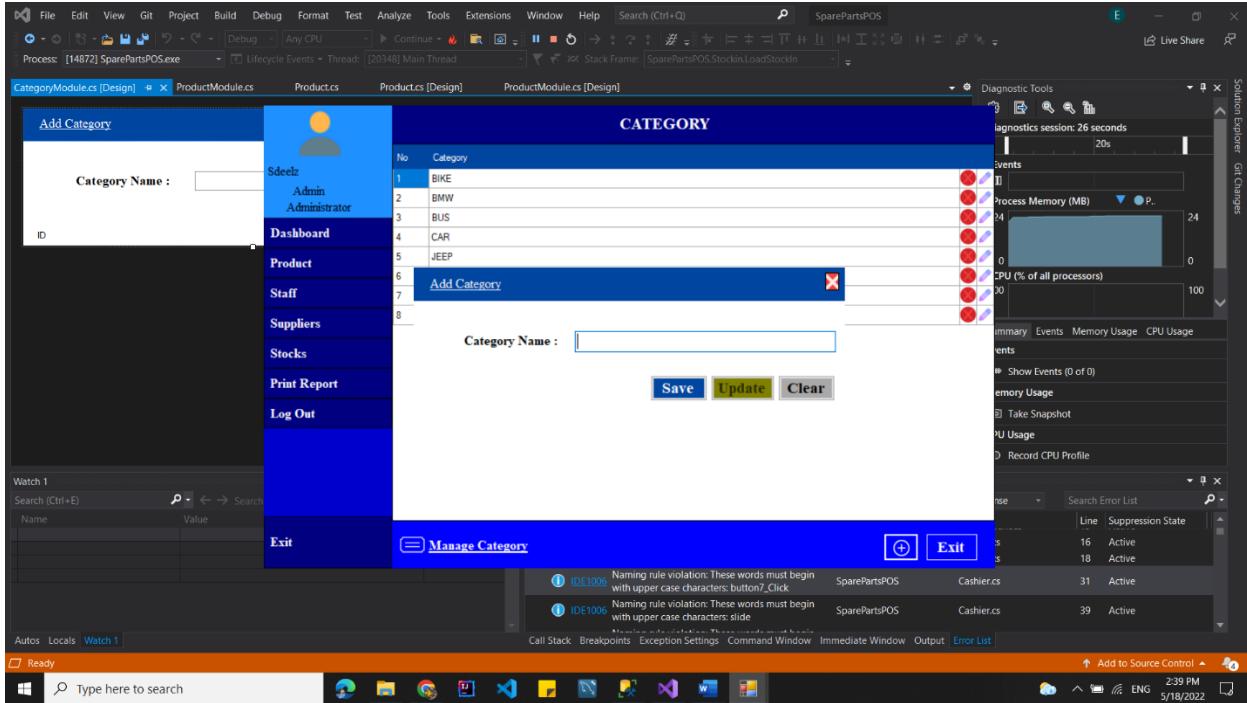


Figure 21 Interface

4.2.3.2.2.1 Category Module Code Segment

```

public partial class CategoryModule : Form
{
    SqlConnection cn = new SqlConnection();
    SqlCommand cm = new SqlCommand();
    Dbconnect dbcon = new Dbconnect();
    Category category;
    public CategoryModule(Category ct)
    {
        InitializeComponent();
        cn = new SqlConnection(dbcon.myConnection());
        category = ct;
    }

    private void CategoryModule_Load(object sender, EventArgs e)
    {

    }
    public void Clear()
    {
        txtCate.Clear();
        txtCate.Focus();
        btnsave.Enabled = true;
    }
}

```

```

        btnupdate.Enabled = false;
    }

    private void btnsave_Click(object sender, EventArgs e)
    {
        try
        {
            if (MessageBox.Show("Are you sure you want to save this Category ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                cn.Open();
                cm = new SqlCommand("INSERT INTO categoryTB
(Category)VALUES(@Category)", cn);
                cm.Parameters.AddWithValue("@Category", txtCate.Text);
                cm.ExecuteNonQuery();
                cn.Close();
                MessageBox.Show("Category has been successful saved.", "Point Of
Sales");
                Clear();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void btnclose_Click(object sender, EventArgs e)
    {
        Clear();
    }

    private void btnupdate_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Are you want to Update this Category ?", "Update
Record", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            cn.Open();
            cm = new SqlCommand("UPDATE categoryTB SET Category =@Category WHERE ID
LIKE'" + lblID.Text + "'", cn);
            cm.Parameters.AddWithValue("@Category", txtCate.Text);
            cm.ExecuteNonQuery();
            cn.Close();
            MessageBox.Show("Category has been Successfully Updated", "Point Of
Sales");
            Clear();
            this.Dispose();
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {

```

4.2.4 Supplier Interface

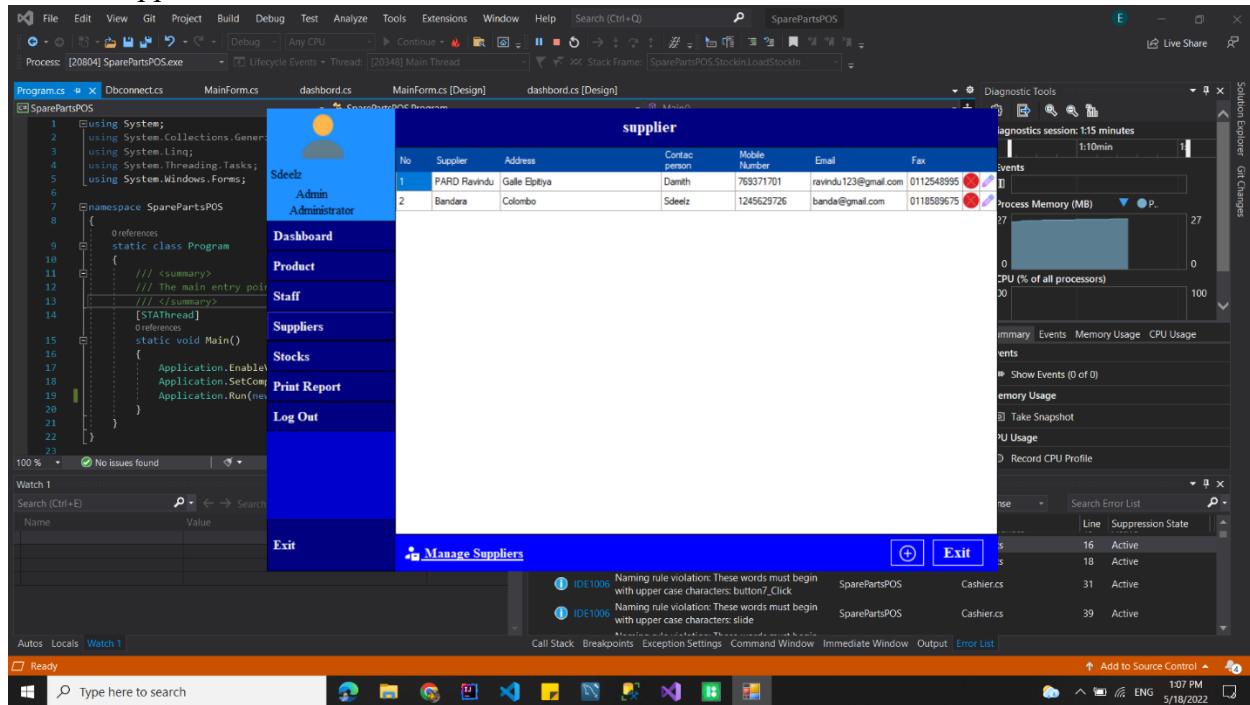


Figure 22 Supplier Interface

4.2.4.1 Supplier Module Code segment

```
public partial class StockModule : Form
{
    SqlConnection cn = new SqlConnection();
    SqlCommand cm = new SqlCommand();
    Dbconnect dbcon = new Dbconnect();
    ManagestkFrom brand;

    public StockModule(ManagestkFrom br)
    {
        InitializeComponent();
        cn = new SqlConnection(dbcon.myConnection());
        brand = br;
    }

    private void btnclose_Click(object sender, EventArgs e)
    {
        Clear();
        //this.Dispose();
    }
    public void Clear()
    {
        txtbrand.Clear();
        btnedit.Enabled = false;
        btnsave.Enabled = true;
        txtbrand.Focus();
    }
}
```

```

    }

    private void btnsave_Click(object sender, EventArgs e)
    {
        try
        {
            if (MessageBox.Show("Are you sure you want to save this brand ?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                cn.Open();
                cm=new SqlCommand("INSERT INTO brandTB (Brand)VALUES(@Brand) ", cn);
                cm.Parameters.AddWithValue("@Brand", txtbrand.Text);
                cm.ExecuteNonQuery();
                cn.Close();
                MessageBox.Show("Record has been successful saved.", "POS");
                Clear();
                brand.loadBrand();

            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Dispose();
    }

    private void btredit_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Are you want to Update this Brand ?", "Update Record", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            cn.Open();
            cm = new SqlCommand("UPDATE brandTB SET Brand =@Brand WHERE ID LIKE'" + lblID.Text + "'", cn);
            cm.Parameters.AddWithValue("@Brand", txtbrand.Text);
            cm.ExecuteNonQuery();
            cn.Close();
            MessageBox.Show("Brand has been Successfully Updated", "POS");
            Clear();
            this.Dispose();

        }
    }

    private void StockModule_Load(object sender, EventArgs e)
    {
    }
}
}

```

4.2.5 Cashier Log In

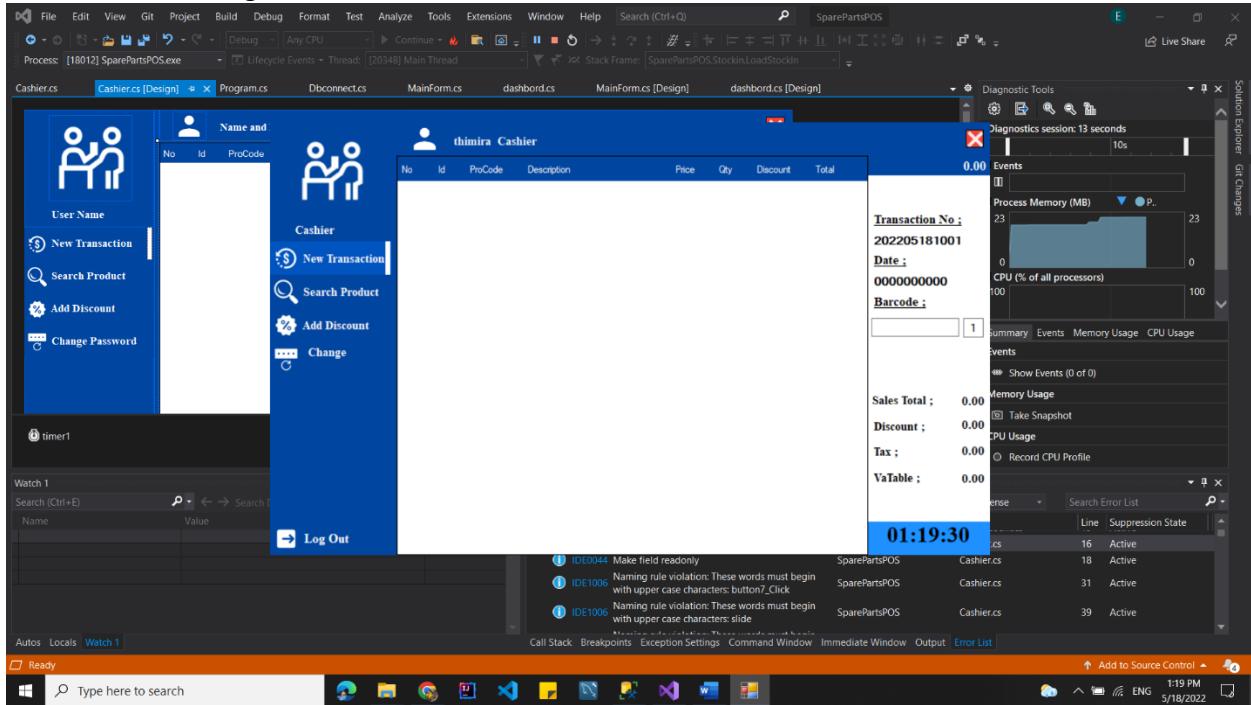


Figure 23 Cashier Interface

4.2.5.1 Cashier Code segment

```
public partial class Cashier : Form
{
    SqlConnection cn = new SqlConnection();
    SqlCommand cm = new SqlCommand();
    Dbconnect dbcon = new Dbconnect();

    int qty;
    string id;
    string price;

    SqlDataReader DR;
    public Cashier()
    {
        InitializeComponent();
        cn = new SqlConnection(dbcon.myConnection());
    }

    private void button7_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Exit this Application?", "Exit Application",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            Application.Exit();
        }
    }
    public void slide(Button button)
    {
        panel7Slide.BackColor = Color.White;
```

```

        panel7Slide.Height = button.Height;
        panel7Slide.Top = button.Top;

    }

#region Button
private void btnnewtrns_Click(object sender, EventArgs e)
{
    slide(btnnewtrns);
    GetTransNo();
}

private void btnnewserchpro_Click(object sender, EventArgs e)
{
    slide(btnnewserchpro);
    LookupProduct lookup = new LookupProduct(this);
    lookup.loadProduct();
    lookup.ShowDialog();
}

private void btnnewdis_Click(object sender, EventArgs e)
{
    Discount dsc = new Discount();
    dsc.lbid.Text = id;
    dsc.txttotalprice.Text = price;
    dsc.ShowDialog();
    slide(btnnewdis);
}

private void btnsttlepay_Click(object sender, EventArgs e)
{
    // slide(btnsttlepay);
}

private void btnclearcrt_Click(object sender, EventArgs e)
{
    // slide(btnclearcrt);
}

private void btndalysale_Click(object sender, EventArgs e)
{
    //slide(btndalysale);
}

private void btnnwchngepw_Click(object sender, EventArgs e)
{
    ChangePassword psw = new ChangePassword();
    psw.ShowDialog();
    slide(btnnwchngepw);
}

private void btnnwlogout_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Log Out Application ?", "LogOut",
    MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation) == DialogResult.Yes)
    {
        this.Hide();
    }
}

```

```

        login Log = new login();
        Log.Show();
    }
    slide(btnnwlogout);
}

private void dgvProduct_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}

private void lbltimer_Click(object sender, EventArgs e)
{
}

#endregion Button
public void Loadcart()
{
    try
    {
        int i = 0;
        double total = 0;
        double discount = 0;

        dgvcashi.Rows.Clear();
        cn.Open();
        cm = new SqlCommand("SELECT c.id,c.ProCode,p.ProDes,c.price, c.disc,
c.total FROM TBCartnew AS c INNER JOIN productTB AS p ON c.ProCode=p.ProCode WHERE
c.transno LIKE @transno and c.status LIKE 'pending'", cn);
        cm.Parameters.AddWithValue("@transno", lblTransNo.Text);
        DR = cm.ExecuteReader();
        while (DR.Read())
        {
            i++;
            total += Convert.ToDouble(DR["total"].ToString());
            discount += Convert.ToDouble(DR["disc"].ToString());
            dgvcashi.Rows.Add(i, DR["id"].ToString(), DR["ProCode"].ToString(),
DR["ProDes"].ToString(), DR["price"].ToString(), DR["qty"].ToString(),
DR["disc"].ToString(), double.Parse(DR["total"].ToString().ToString("#,##0.00")));

        }
        DR.Close();
        cn.Close();
        lblSalestotal.Text = total.ToString("#,##0.00");
        lbldiscount.Text = discount.ToString("#,##0.00");
        GetcartTotal();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "POINT OF SALE");
    }
}

public void GetcartTotal()
{
}

```

```

        double discount = double.Parse(lbldiscount.Text);
        double sales = double.Parse(lblSalestotal.Text) - discount;
        double vat = sales * 0.12;//vat 12%
        double vateble = sales - vat;

        lbltax.Text = vat.ToString("#,##0.00");
        lblvatble.Text = vateble.ToString("#,##0.00");
        lblDisplayTotal.Text = sales.ToString("#,##0.00");
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        lbltimer.Text = DateTime.Now.ToString("hh:mm:ss tt");
    }
    public void GetTransNo()
    {
        string sdate = DateTime.Now.ToString("yyyMMdd");
        string transno = sdate + "1001";
        lblTransNo.Text = transno;
    }

    private void txtBarcode_TextChanged(object sender, EventArgs e)
    {
        try
        {
            if (txtBarcode.Text == string.Empty) return;
            else
            {
                string _pcode;
                double _price;
                int _qty;
                cn.Open();
                cm = new SqlCommand("SELECT * FROM productTB WHERE BarCode LIKE '" + txtBarcode.Text + "'", cn);
                DR = cm.ExecuteReader();
                DR.Read();
                if (DR.HasRows)
                {
                    qty = int.Parse(DR["proQti"].ToString());
                    _pcode= (DR["ProCode"].ToString());
                    _price = double.Parse(DR["ProPrice"].ToString());
                    _qty = int.Parse(txtcqty.Text);

                    DR.Close();
                    cn.Close();
                    //inser to tb cart
                    AddCart(_pcode, _price, _qty);
                }
            }
            DR.Close();
            cn.Close();
        }
        catch (Exception EX)
        {
            cn.Close();
        }
    }
}

```

```

        MessageBox.Show(EX.Message, "POINT OF SALE", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
    }
}

public void AddCart(string _PCODE, double _price, int _qty)
{
    try
    {
        string id = "";
        int cart_qty = 0;
        bool found = false;
        cn.Open();
        cm = new SqlCommand("SELECT * FROM TBCartnew WHERE transno=@transno and
ProCode=@ProCode ", cn);
        cm.Parameters.AddWithValue("@transno", lblTransNo.Text);
        cm.Parameters.AddWithValue("@ProCode", _PCODE);
        DR = cm.ExecuteReader();
        DR.Read();
        if (DR.HasRows)
        {
            id = DR["id"].ToString();
            cart_qty = int.Parse(DR["qty"].ToString());

            found = true;
        }
        else found = false;
        DR.Close();
        cn.Close();

        if (found)
        {
            if (qty < (int.Parse(txtQty.Text) + cart_qty))
            {

                MessageBox.Show("Unabale to Process reaming qty on hand is " +
qty, "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }
            cn.Open();
            cm = new SqlCommand("UPDATE TBCartnew SET qty=(qty+" + _qty + ")Where
id ='" + id + "'", cn);
            cm.ExecuteReader();
            cn.Close();
            txtBarcode.SelectionStart = 0;
            txtBarcode.SelectionLength = txtBarcode.Text.Length;
            Loadcart();
        }
        else
        {
            if (qty < (int.Parse(txtQty.Text) + cart_qty))
            {

                MessageBox.Show("Unabale to Process reaming qty on hand is " +
qty, "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }
        }
    }
}

```

```

        }
        cn.Open();
        cm = new SqlCommand("INSERT INTO TBCartnew
(transno,ProCode,price,qty,sdate,cashier) VALUES
(@transno,@ProCode,@price,@qty,@sdate,@cashier)", cn);
        cm.Parameters.AddWithValue("@transno", lblTransNo.Text);
        cm.Parameters.AddWithValue("@ProCode", _pcode);
        cm.Parameters.AddWithValue("@price", _price);
        cm.Parameters.AddWithValue("@qty", 1);
        cm.Parameters.AddWithValue("@sdate", DateTime.Now);
        cm.Parameters.AddWithValue("@cashier", lblUsername.Text);
        cm.ExecuteNonQuery();
        cn.Close();
        Loadcart();
    }
}
catch (Exception EX)
{
    MessageBox.Show(EX.Message, "POINT OF SALE");
}
}

private void dgvcashi_SelectionChanged(object sender, EventArgs e)
{
    int i = dgvcashi.CurrentRow.Index;
    id = dgvcashi[1, i].Value.ToString();
    price = dgvcashi[7, i].Value.ToString();
}
}
}

```

4.2.6 Reports Module

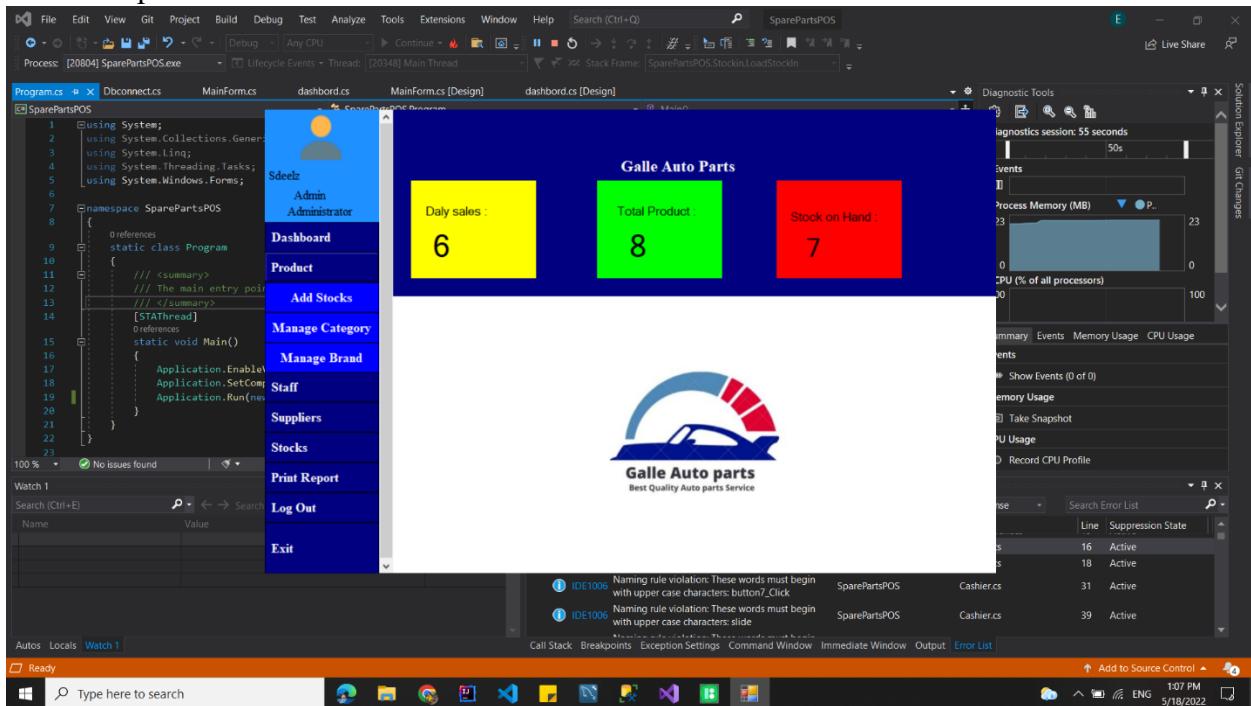


Figure 24 Interface

4.2.6.1 Code segment

```
class Dbconnect
{
    SqlConnection cn = new SqlConnection();
    SqlCommand cm = new SqlCommand();
    private String con;
    string sql;
    public String myConnection()
    {
        con = @"Data Source=DESKTOP-491D699;Initial Catalog=galleParts;Integrated Security=True";
        return con;
    }
    public DataTable GetTable(string query)
    {
        cn.ConnectionString = myConnection();
        cm = new SqlCommand(query, cn);
        SqlDataAdapter adapter = new SqlDataAdapter(cm);
        DataTable table = new DataTable();
        adapter.Fill(table);
        return table;
    }

    public double ExtractData(String sql)
    {
        cn = new SqlConnection();
        cn.ConnectionString = myConnection();
        cn.Open();
```

4.2.7 ER -Diagram

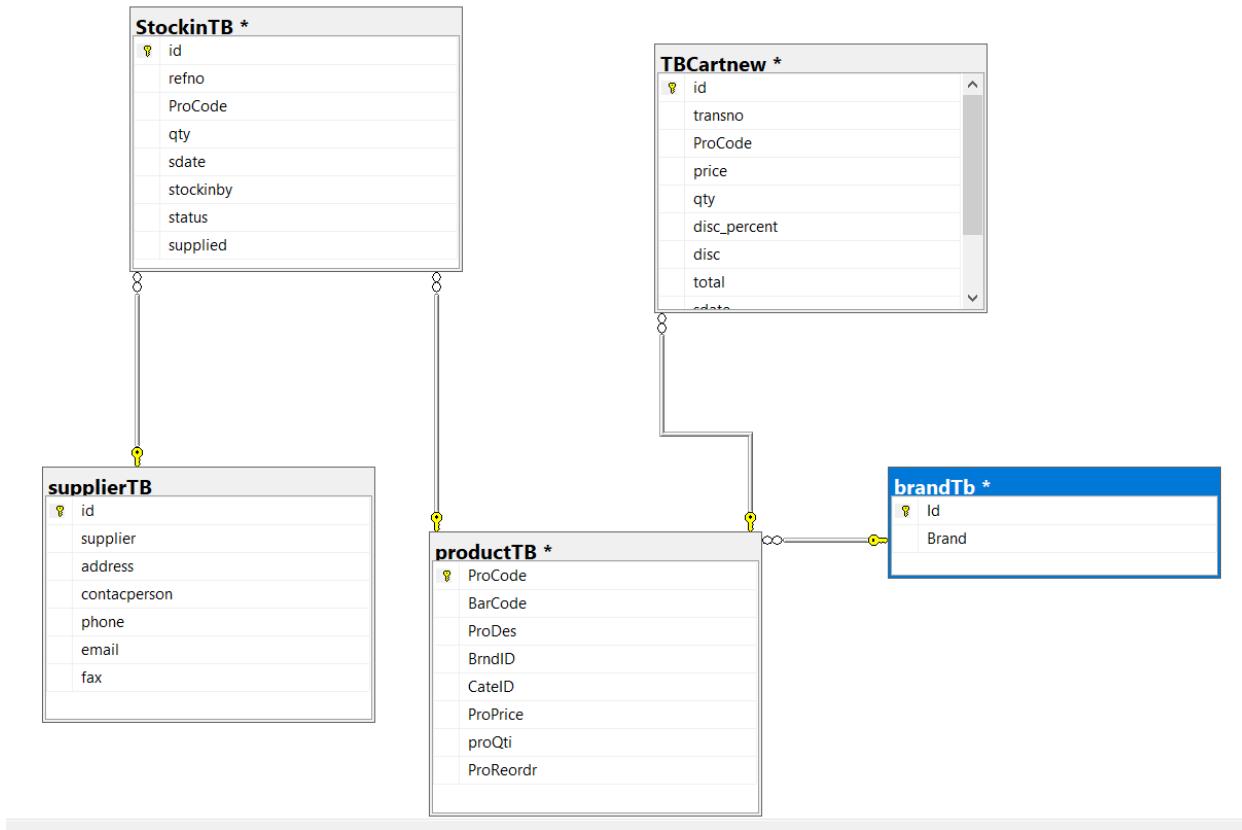
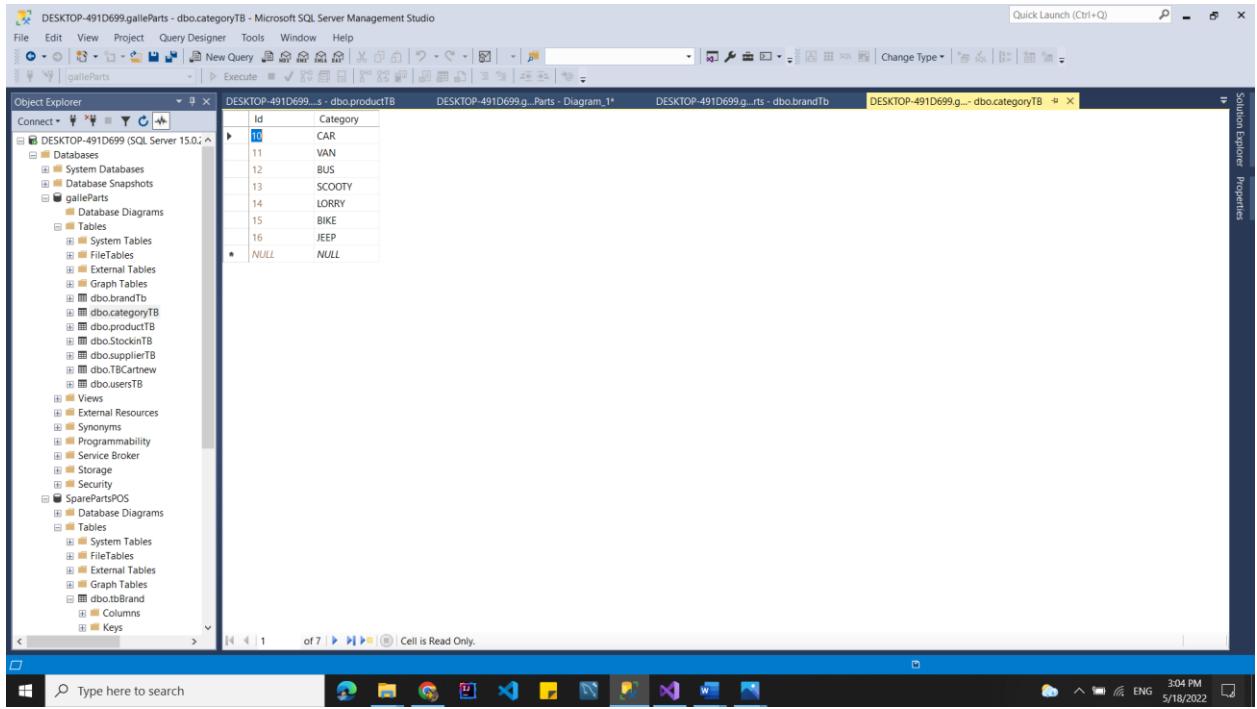


Figure 25 ER Diagram

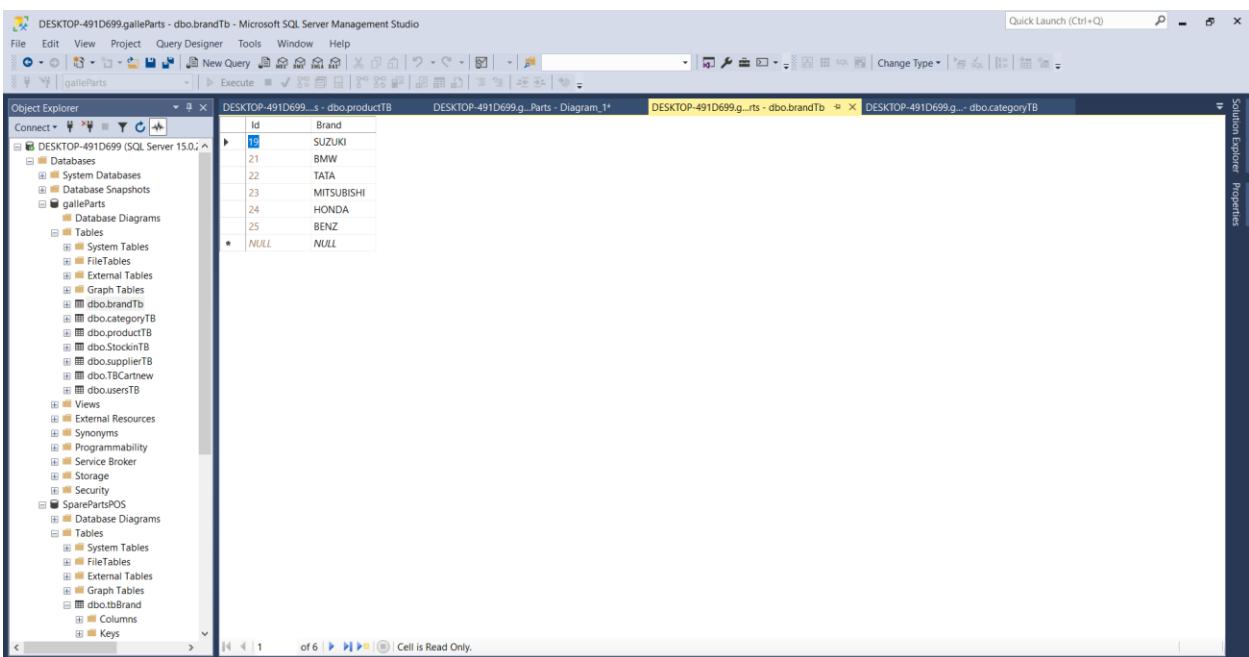
4.2.8 Data Base



The screenshot shows the Microsoft SQL Server Management Studio interface with the 'Category' table selected in the Object Explorer. The table has two columns: 'Id' and 'Category'. The data is as follows:

Id	Category
10	CAR
11	VAN
12	BUS
13	SCOOTY
14	LORRY
15	BIKE
16	JEEP
*	NULL

Figure 27 Category table



The screenshot shows the Microsoft SQL Server Management Studio interface with the 'Brand' table selected in the Object Explorer. The table has two columns: 'Id' and 'Brand'. The data is as follows:

Id	Brand
19	SUZUKI
21	BMW
22	TATA
23	MITSUBISHI
24	HONDA
25	BENZ
*	NULL

Figure 26 Brand Table

DESKTOP-491D699.galleParts - dbo.supplierTB - Microsoft SQL Server Management Studio

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to DESKTOP-491D699.galleParts on SQL Server 15.0. The Object Explorer on the left shows the database structure, including tables like dbo.supplierTB, dbo.productTB, and dbo.brandTB. The main pane displays the data from the dbo.supplierTB table:

	id	supplier	address	contactpers...	phone	email	fax
1	PARD Ravin...	Galle Elpitiya	Damith	769371701	ravindu123...	0112548995	
2	Bandara	Colombo	Sdeelz	1245629726	banda@gm...	0118589675	
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, there are navigation buttons (of 2, back, forward, search) and a note that 'Cell is Read Only'.

Figure 28 supplier Table

DESKTOP-491D699.galleParts - dbo.productTB - Microsoft SQL Server Management Studio

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to DESKTOP-491D699.galleParts on SQL Server 15.0. The Object Explorer on the left shows the database structure, including tables like dbo.productTB, dbo.brandTB, and dbo.categoryTB. The main pane displays the data from the dbo.productTB table:

	ProCode	BarCode	ProDes	BrndID	CatID	ProPrice	proQty	ProReorder
P001	654321	P001-654321	19	3	500.00	10	2	
P0010	65550	BMW Wheel	19	10	80000.00	0	1	
P002	654320	P002-654320	19	3	500.00	0	3	
P003	65429	P003-65429	19	3	800.00	0	3	
P004	65428	P004-65428	20	10	5000.00	0	2	
P005	65427	P005-65427	20	10	6000.00	0	2	
P006	65426	P006-65426	20	10	80000.00	0	2	
p008	45612	tyre	19	10	5000.00	0	1	
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, there are navigation buttons (of 8, back, forward, search) and a note that 'Cell is Read Only'.

Figure 29 Product table

4.2.9 View Systems

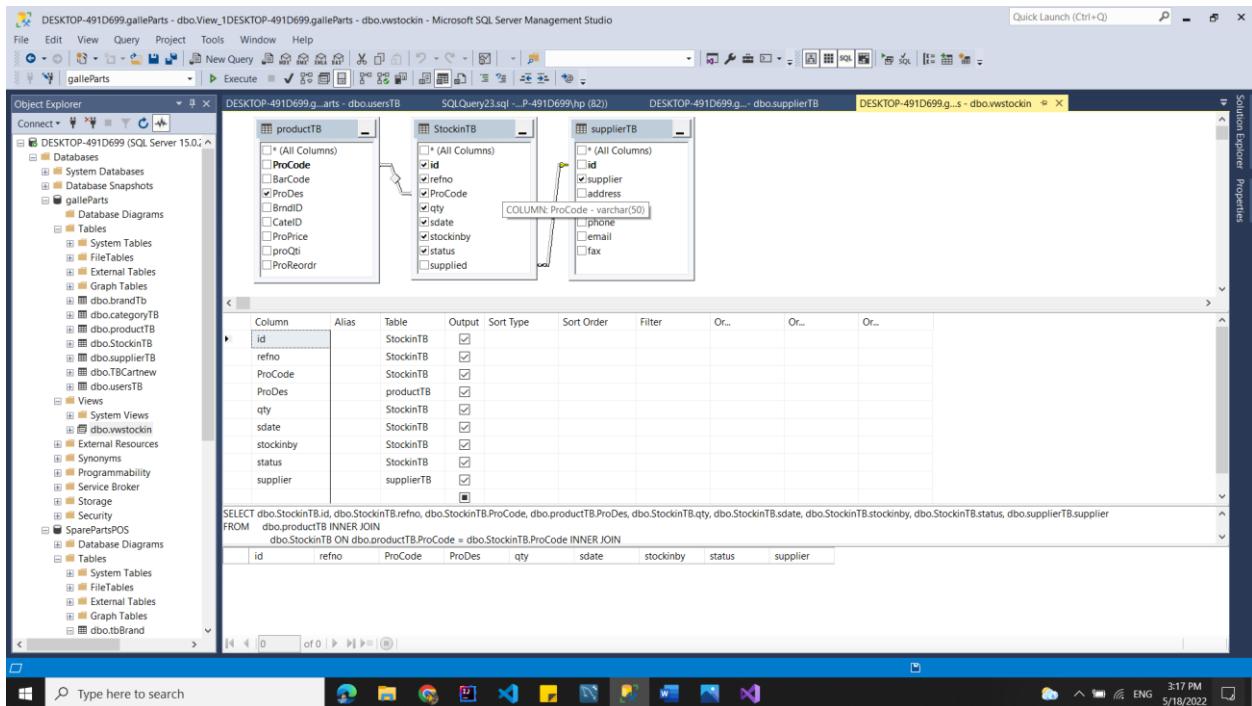


Figure 30 View systems

REFERENCES

- [1] "Auto lanka Advertising & Marketing (pvt) ltd," Auto lanka Advertising & Marketing (pvt) ltd, [Online]. Available: <https://www.auto-lanka.com/>. [Accessed 24 02 2022].
- [2] "Douglas & Sons (pvt) ltd," Douglas & Sons (pvt) ltd, [Online]. Available: <https://www.dsl-autoparts.com/>. [Accessed 25 02 2022].
- [3] "Grease Monkey .lk," Grease Monkey, [Online]. Available: <https://greasemonkey.lk/>. [Accessed 25 02 2022].
- [4] "Original Equipment manufacturer(OEM)," Original Equipment manufacturer, [Online]. Available: <https://oemparts.lk/>. [Accessed 25 02 2022].
- [5] "Euro Car Parts," Euro Car Parts, [Online]. Available: <https://www.eurocarparts.com/>. [Accessed 25 02 2022].
- [6] "Care Distributers," Care Distributers, [Online]. Available: https://caredistributors.com.au/categories/polishes-compounds.html?gclid=Cj0KCQiAmeKQBhDvARIsAHJ7mF4buy1PanpFhRMZF04Z7oHkI6xZ8-rQFPLd7SHqznS5qZxE2bCIv4kaAvL4EALw_wcB. [Accessed 25 02 2022].

