LAXMAN DEVRAM SONAWANE COLLEGE

(Affiliated to University Of Mumbai)
2022 – 2023

Project Report On
CAFE MANAGEMENT SYSTEM

Project work submitted in partial fulfilment of the requirements for the award of the degree of
B.Sc. (Computer Science)

SUBMITTED BY
ONASWEE VIVEK JOSHI

PROJECT GUIDE
MS. PRIYANKA SURYAVANSHI

CERTIFICATE

This is to verify that
ONASWEE VIVEK JOSHI

Has satisfactorily carried out the project work on the topic

"**CAFE MANAGEMENT SYSTEM**"
For the T.Y.B.Sc (COMPUTER SCIENCE)
ACADEMIC YEAR
2022- 2023

_____
Project Guide

_____ _____
B.Sc (CS) Co-ordinator External Examiner

INDEX

Sr.No.		Topic
		Acknowledgement
1		**Abstract**
2		**Preliminary Investigation**
	1	Objectives and scope of the system
	2	Description of System
	3	Feasibility Study
	4	Project Analysis
	5	System And Design User Requirement
	6	Gantt Chart
3		**Project Design**
	1	Design Document
	2	Data Flow Diagram
	3	Activity / Registration Diagram
	4	UML
	5	Products
	6	Moderators
	7	Users
	8	SQL(Structured Query Language)
	9	Database Diagram
4		**Screen Layout**
5		**System Implementation**
6		**System Maintenance**
7		**Project Testing**
8		**Conclusion And Biblography**

ACKNOWLEDGEMENT:

We Thank The Almighty For Giving Us The Courage & Perseverance In Completing The Project. This Project Itself Is An Acknowledgement For All Those Who Have Given Us Their Heart-Felt-Co-Operation In Making It A Grand Success.

We Are Greatly Indebted To, Head Of Computer Science, **Mrs.Kanyakumari J Basani** For Providing Valuable Guidance At Every Stage Of This Project Work.

We Are Also Thankful To The Project Coordinator & Guide, **Mrs.Priyanka Suryavanshi** For Extending Their Sincere & Heartfelt Guidance Throughout This Project Work. Without Their Supervision And Many Hours Of Devoted Guidance, Stimulating & Constructive Criticism, This Thesis Would Never Come Out In This Form.

It Is A Pleasure To Express Our Deep And Sincere Gratitude To The Project Guide And Are Profoundly Grateful Towards The Unmatched Help Rendered By Him. Our Special Thanks To All The Lectures Of Information Technology, For Their Valuable Advises At Every Stage Of This Work.

Last But Not The Least; We Would Like To Express Our Deep Sense And Earnest Thanks Giving To Our Dear Parents For Their Moral Support And Heartfelt Cooperation In Doing The Project. We Would Also Like To Thank Our Friends, Whose Direct Or Indirect Help Has Enabled Us To Complete This Work Successfully.

ABSTRACT

The Cafe Management System Aims To Maintain And Manage The Various Products And Processes. It Is Primarily For Shopping In The Database's Core Section The System Gives The Information On The Various Products That Are Available, As Well As Their Current Availability Status. Customers Can Visit And Register With The System By Providing The Required Information. Each Registered Customer Will Get Information About The Availability Of The Products Of The Specific Category They Have Requested For

This Application Allows To Operate The Entire System From A Single Online Interface, Given Them More Power And Flexibility. Selecting Specific Product From Given Categories And Other Process Till Payment And Placing The Order Are Included In This Project

The Admin Can Use The System To Check The Available Products. Customers Can See And Order Their Favourite Products From The Given Specific Categories.Admin Has The Authority To Confirm Or Reject A Customer’s Order. As A Result The System Is Used By The Admin To Control The Activities For The Website.

PRELIMINARY INVESTIGATION

OBJECTIVE AND SCOPE OF THE SYSTEM

Objectives Of The Proposed System

1. To Place Quick Orders
2. To Make Accuracy In Shipping Orders
3. To Provide Proper Information Briefly
4. To Provide Data Security
5. To Provide Huge Maintenance Of Records
6. Flexibility Of Transactions Can Be Completed In Time

After Understanding The Existing System And Understanding The Need For Developing A New System Different People Involved In The Related Activities Have Been Consulted. The Data Needed For The Study Has Been Collected From Website Records.

The System Help The Customer To See Any Product Information In Specific Category With Details At The Click Of A Button. The Record Data Is Maintained And Backed Up Such A Way That Data Is Not Loss. The Speed Of The System Could Also Increased.

Scope Of The Project:

CAFE MANAGEMENT SYSTEM Allows to Uses An Access Database To Store All The Records Of Customers, Products And Purchase Order Information. A Full Fledged System Will Increase Efficiency Through Paper Works. The Complete Integration With Financial Accounting Provides A Full Featured System To Manage Shopping Facility

PROJECT OVERVIEW:

The Entire System Is Divided Into 3 Major Modules

1. **Registration Module**
2. **Customer Module**
3. **Admin Module**

Registration Module:

In This Registration Module It Creates And Maintains All Customer Details, Product Details, Order Placed Details, User Payment Details..

User Module:

In This User Module, It Creates Users, Maintain User Details And Maintain User's Purchase Details.

Payment Module:

In This Payment Module It Creates Express Payments, Qucik Order Place, Secure Payment.

DESCRIPTION OF SYSTEM :

As The Name Specifies “CAFE MANAGEMENT SYSTEM”

For The Past Few Years The Number Of Customers Is Increasing Rapidly.

Thereby The Number Of Orders Is Also Increasing For The Accommodation Of The Customers Purchasing The Products. And Hence There Is A Lot Of Strain On The Peoples Who Are Running This Site.

This Particular Project Deals With The Problems On Managing A Website And Avoids The Problems Which Occur When Carried Manually Identification Of The Drawbacks Of The Existing System Leads To The Designing Of Computerized System That Will Be Compatible To The Existing System With The System

Which Is More User Friendly And More GUI Oriented. We Can Improve The Efficiency Of

The System, Thus Overcome The Drawbacks Of The Existing System.

· Less Human Error

· Strength And Strain Of Manual Labour Can Be Reduced

· High Security

· Data Redundancy Can Be Avoided To Some Extent

· Data Consistency

· Easy To Handle

· Easy Data Updating And Record Keeping

FEASIBILITY STUDY

Feasibility: A Measure Of How Beneficial Or Practical The Development Of Information System Would Be To An Organization Can Be Termed As Project Feasibility.

- Feasibility Should Be Measured Throughout The Life Cycle. The Scope And Complexity Of An Apparently Feasible Project Can Change After The Initial Problems And Opportunities Are Fully Analyzed And After The System Has Been Designed. Thus, A Project That Is Feasible At One Point May Become Infeasible Later.

- The Checkpointsfor Feasibility Study Are: -
  - A Survey Phase Checkpoint
  - A Study Phase Checkpoint.
  - A Definition Phase Checkpoint
  - A Selection Phase Checkpoint.

- Generally There Are Following Areas Of Risk For A New System That Are Considered When Confirming Project Feasibility:

1.Operational Feasibility /Organizational And Cultural Feasibility Are Measure Of How Well The Solution Will Work In The Organization. It Is Also A Measure Of How People Feel About The System Or Project.

2. Technicalfeasibilityis A Measure Of How The Practicality Of A Specific Technical Solution And The Availability Of Technical Resources And Expertise.

3.Schedule Feasibility Is A Measure Of How Responsible The Project Timetable Is.

4. Economic Feasibility Is A Measure Of Cost-Effectiveness Of A Project Or Solution. This Is Often Called A Cost-Benefit Analysis.

PROJECT ANALYSIS

1. Existing System

Present System Is Manual. The Project Metrics Has To Enter All The Details Of Project, Documents, And Tasks. It Also Maintain The Team Information And Also Efforts Estimation. For This Purpose The Organization Maintain The Size Of The Document, Source Code And Update The Information About Team Member's Details Manually. Which Is Much Of Time Consuming Process And More Importantly It Is Error Prone. Limitations Of The Manual System

1. It Is Time Consuming
2. It Leads To Error Prone Results
3. It Consumes Lot Of Manpower To Better Results
4. It Lacks Of Data Security
5. Retrieval Of Data Takes Lot Of Time
6. Percentage Of Accuracy Is Less
7. Reports Take Time To Produce

Hence Computerization Of The Existing System Is Proposed. The New System Completely Removes All Manual Burdens And Provide Efficient On The Entry System.

SYSTEM AND DESIGN USER REQUIREMENT

DEVELOPMENT TOOL USED

SOFTWARE CONFIGURATION

OPERATING PLATFORM	**:**	WINDOWS 10
RDBMS	**:**	SQLSERVER
SOFTWARE	**:**	VS.NET 2015
FRONT END TOOL	**:**	ASP.NET

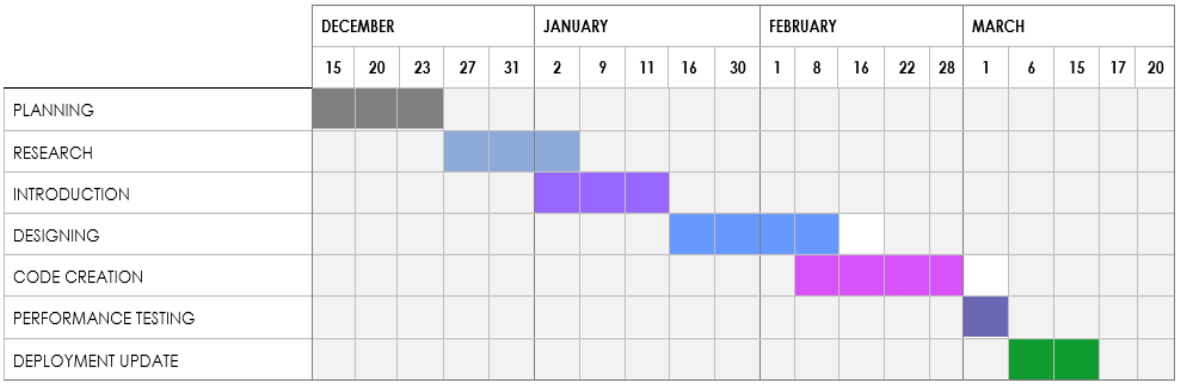
HARDWARE CONFIGURATION

RAM	**:**	**8** GB
HARD DISK	**:**	MINIMUM **500** GB
PROCESSER	**:**	**2.30 GHz**

GANTT CHART

A **Gantt Chart** Is A Type Of Bar Chart That Illustrates A Project Schedule. Gantt Charts Illustrate The Start And Finish Dates Of The Terminal Elements And Summary Elements Of A Project. Terminal Elements And Summary Elements Comprise The Work Breakdown Structure Of The Project.

PROJECT NAME	PROJECT START DATE	PROJECT END DATE
Café Management	December 15	March 15



PROJECT DESIGNING

DESIGN DOCUMENT

- The Entire System Is Projected With A Physical Diagram Which Specifics The Actual Storage Parameters That Are Physically Necessary For Any Database To Be Stored On To The Disk. The Overall Systems Existential Idea Is Derived From This Diagram.

- The Relation Upon The System Is Structure Through A Conceptual ER-Diagram, Which Not Only Specifics The Existential Entities But Also The Standard Relations Through Which The System Exists And The Cardinalities That Are Necessary For The System State To Continue.

- The Content Level DFD Is Provided To Have An Idea Of The Functional Inputs And Outputs That Are Achieved Through The System. The System Depicts The Input And Out Put Standards At The High Level Of The Systems Existence.

DATA FLOW DIAGRAMS:

Data Flows Are Data Structures In Motion, While Data Stores Are Data Structures. Data Flows Are Paths Or 'Pipe Lines', Along Which Data Structures Travel, Where As The Data Stores Are Place Where Data Structures Are Kept Until Needed.

Data Flows Are Data Structures In Motion, While Data Stores Are Data Structures At Rest. Hence It Is Possible That The Data Flow And The Data Store Would Be Made Up Of The Same Data Structure.

Data Flow Diagrams Is A Very Handy Tool For The System Analyst Because It Gives The Analyst The Overall Picture Of The System, It Is A Diagrammatic Approach.

A DFD Is A Pictorial Representation Of The Path Which Data Takes From Its Initial Interaction With The Existing System Until It Completes Any Interaction. The Diagram Will Describe The Logical Data Flows Dealing The Movements Of Any Physical Items. The DFD Also Gives The Insight Into The Data That Is Used In The System I.E., Who Actually Uses It Is Temporarily Stored.

A DFD Does Not Show A Sequence Of Steps. A DFD Only Shows What The Different Process In A System Is And What Data Flows Between Them.

- **PROCESS**:

A Transaction Of Information That Resides Within The Bounds Of The System To Be Module.

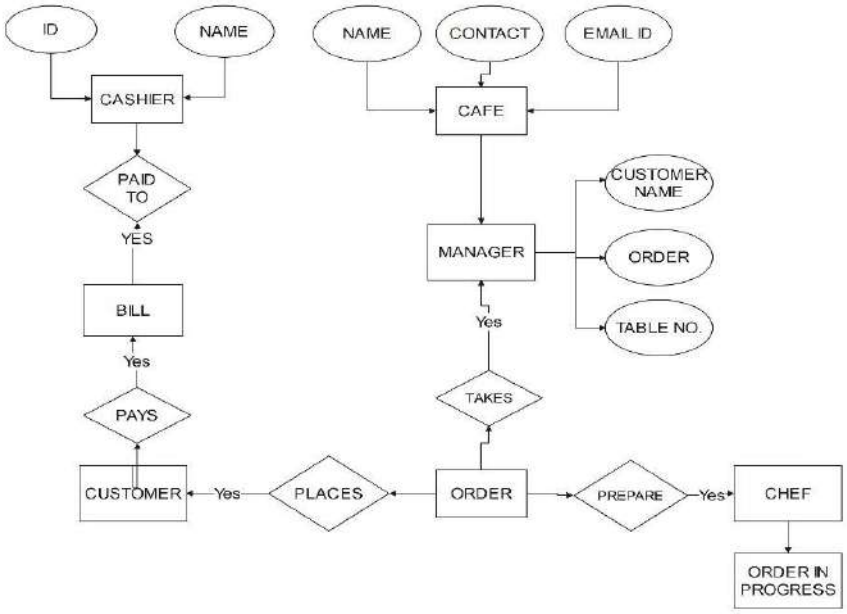
- **DATASTORE**:

A Repository Of Data That Is To Be Stored For Use By One Or More Processes, May Be As Simple As A Buffer Or Queue Or As A Relational Database.

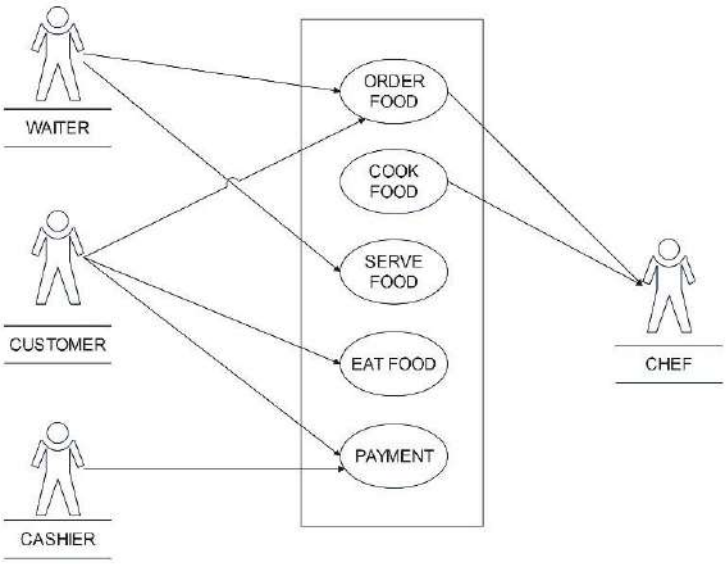
RULES FOR DFD:

- Fix The Scope Of The System By Means Of Context Diagrams.
- Organize The DFD So That The Main Sequence Of The Actions Reads Left To Right And Top To Bottom.
- Identify All Inputs And Outputs.
- Identify And Label Each Process Internal To The System With Rounded Circles. Process Is Required For All The Data Transformation And Transfers. Therefore, Never Connect A Data Store To A Data Source Or The Destinations Or Another Data Store With Just A Data Flow Arrow.
- Do Not Indicate Hardware And Ignore Control Information.
- Make Sure The Names Of The Processes Accurately Convey Everything The Process Is Done.
- There Must Not Be Unnamed Process.
- Indicate External Sources And Destinations Of The Data, With Squares.
- Number Each Occurrence Of Repeated External Entities.
- Identify All Data Flows For Each Process Step, Except Simple Record Retrievals.
- Label Data Flow On Each Arrow.
- Use Details Flow On Each Arrow.
- Use The Details Flow Arrow To Indicate Data Movements.
- There Can’t Be Unnamed Data Flow.
- A Data Flow Can’t Connect Two External Entities.

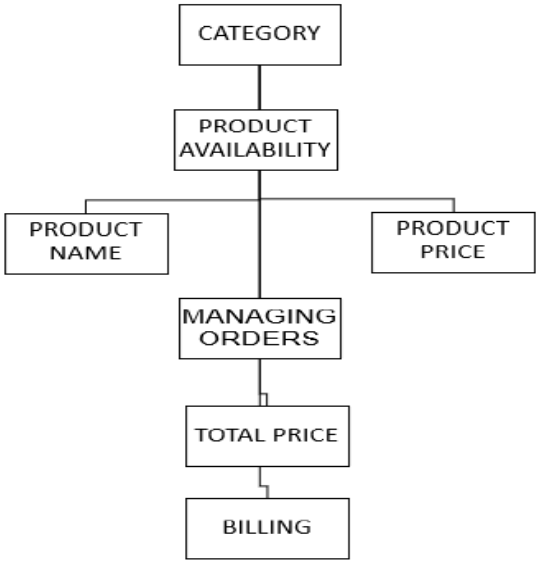
ENTITY RELATIONSHIP DIAGRAM



USE CASE DIAGRAM

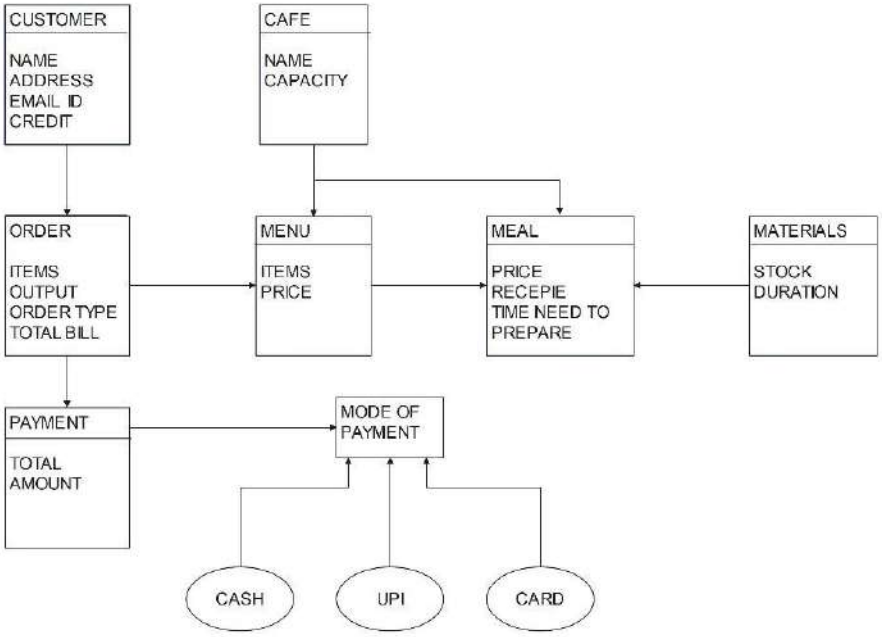


ADMIN ACTIVITY DIAGRAM

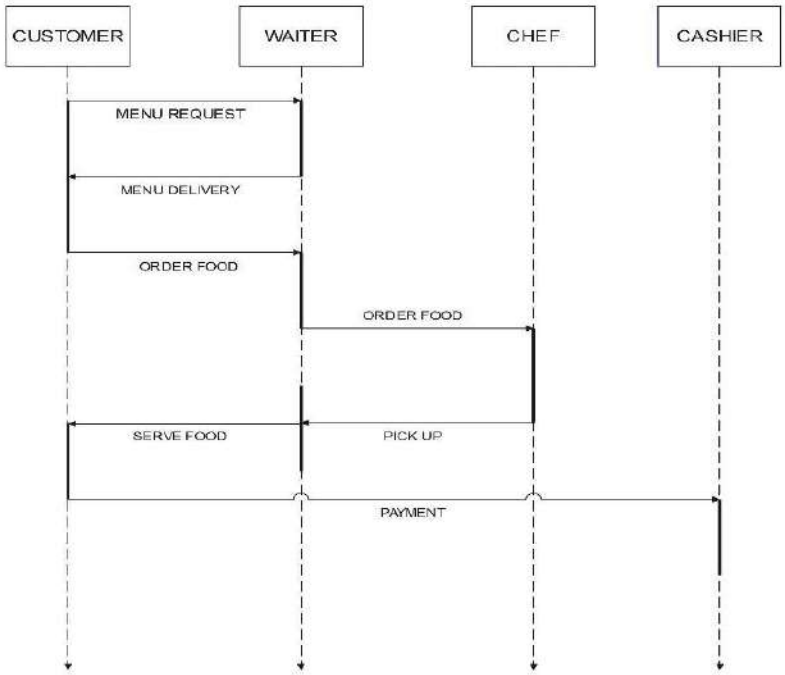


USER ACTIVITY DIAGRAM

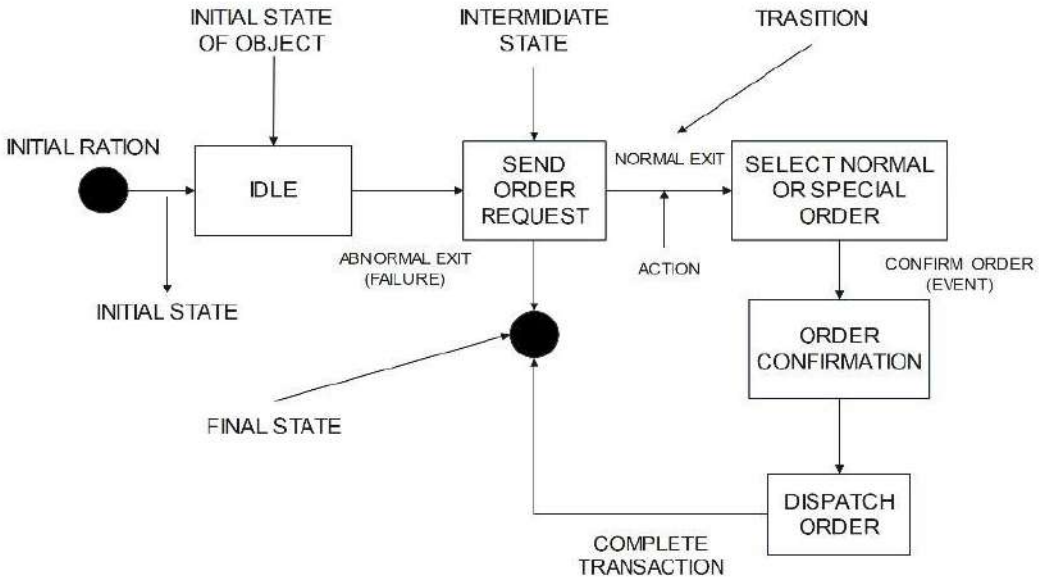
CLASS DIAGRAM



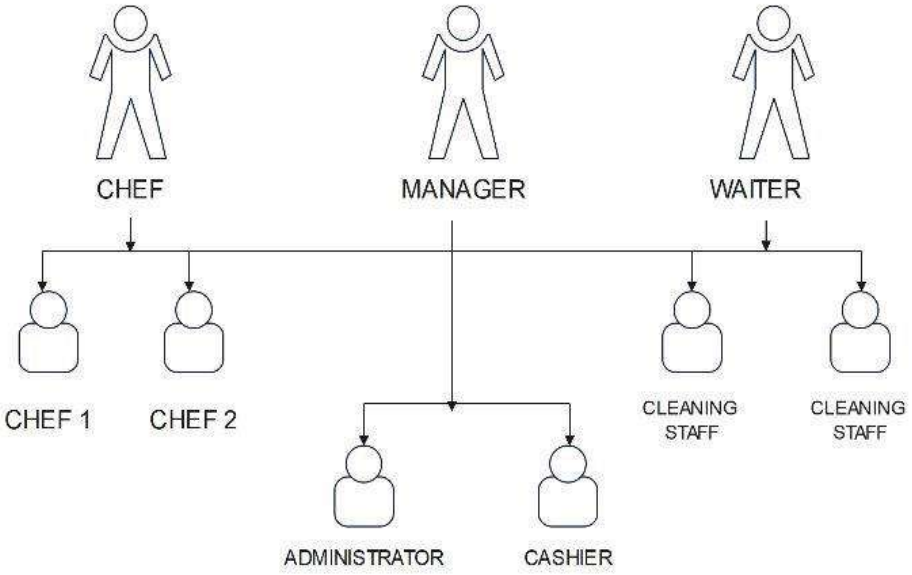
SEQUENCE DIAGRAM



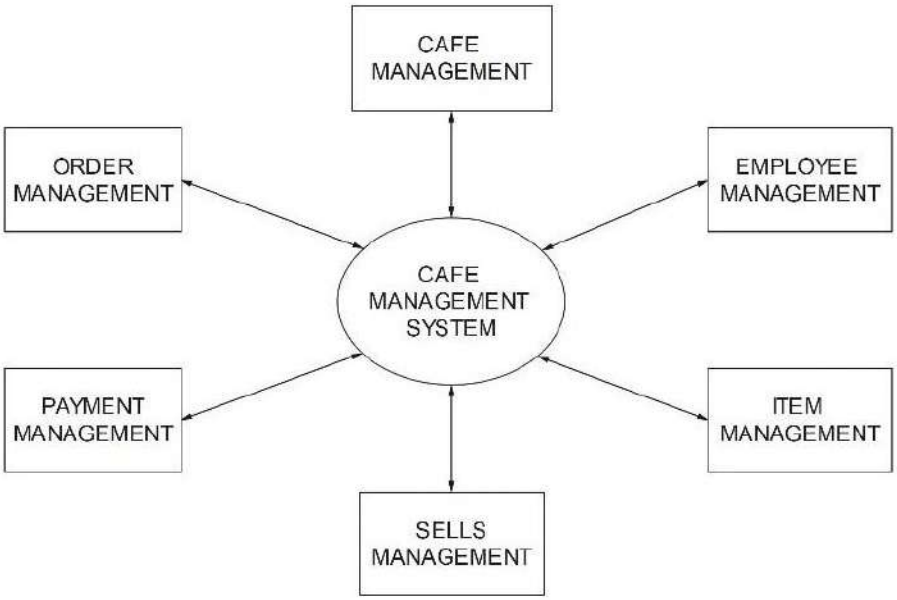
STATE CHART DIAGRAM



OBSERVATION AND FUNCTION OF CAFE MANAGEMENT



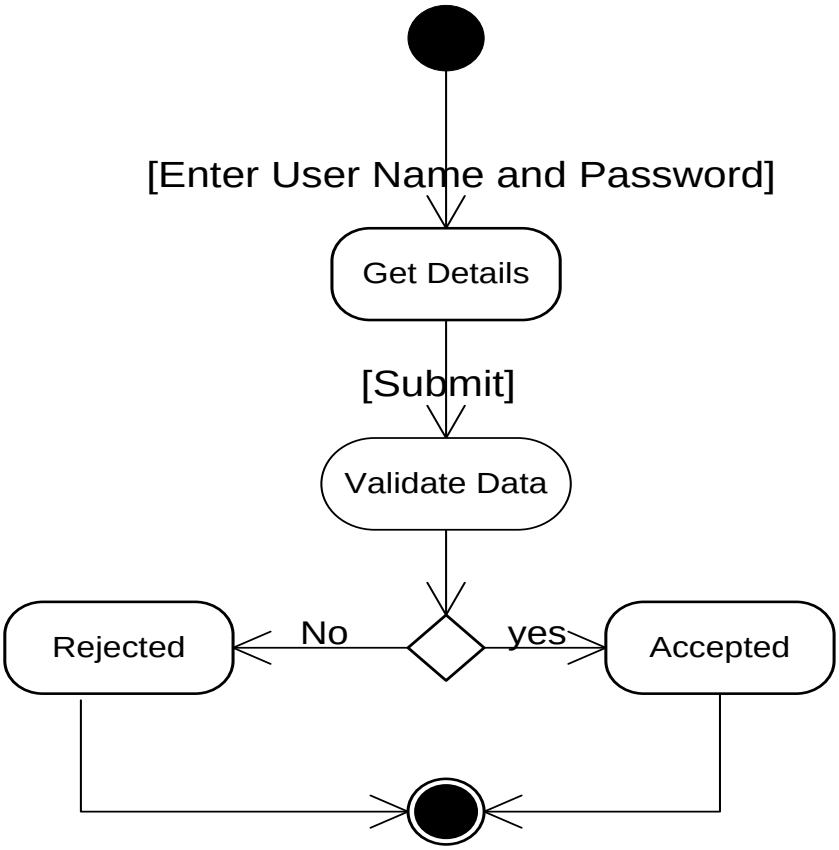
DATA FLOW DIAGRAM



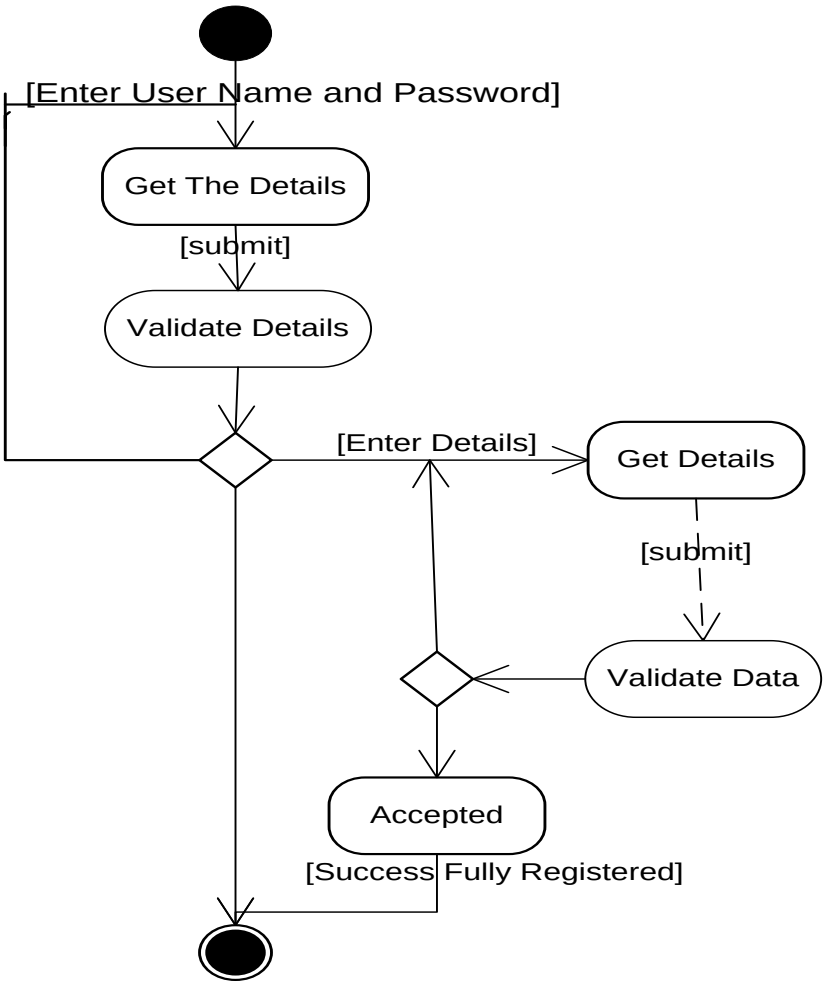
ACTIVITY DIAGRAM:

Activity Diagrams Are Graphical Representations Of Work Flows Of Stepwise Activities And Action Which Support For Choice Interaction And Concurrency

User Login Activity



USER REGISTRATION ACTIVITY



UNIFIED MODELING LANGUAGE DIAGRAMS(UML):

- The Unified Modeling Language Allows The Software Engineer To Express An Analysis Model Using The Modeling Notation That Is Governed By A Set Of Syntactic Semantic And Pragmatic Rules.
- A UML System Is Represented Using Five Different Views That Describe The System From Distinctly Different Perspective. Each View Is Defined By A Set Of Diagram, Which Is As Follows.

User Model View

i. This View Represents The System From The Users Perspective.

ii. The Analysis Representation Describes A Usage Scenario From The End-Users Perspective**.**

Structural Model View

- In This Model The Data And Functionality Are Arrived From Inside The System.
- This Model View Models The Static Structures.

Behavioral Model View

- It Represents The Dynamic Of Behavioral As Parts Of The System, Depicting The Interactions Of Collection Between Various Structural Elements Described In The User Model And Structural Model View

Implementation Model View

- In This The Structural And Behavioral As Parts Of The System Are Represented As They Are To Be Built.

Environmental Model View

In This The Structural And Behavioral Aspects Of The Environment In Which The System Is To Be Implemented Are Represented.UML Is Specifically Constructed Through Two Different Domains They Are

- UML Analysis Modeling, Which Focuses On The User Model And Structural Model Views Of The System.

PRODUCTS

Here Are Many Products Available At Affordable Prices With Premium Quality.

The Products Are Like Beverages, Cakes & Fast Food Snacks. The Food Items Are Prepared Hyginecaly In Professional Way By Best Chefs & Cooks.

Creating This Makes A Good Oppourtunity For The New Start Up Business With Easy Process For All Types Of Food Products As Per Likes

MANAGE PRODUCTS

Add Products -

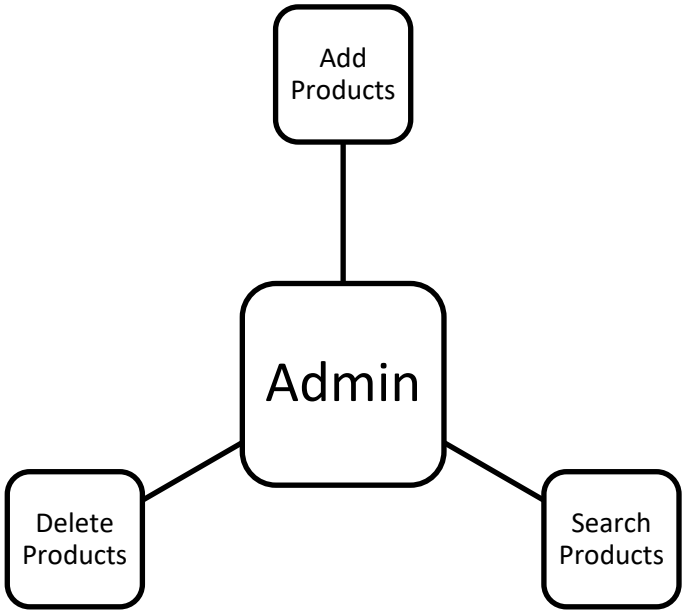
The Project Contains Different Kinds Of Products. The Products Can Be Classified Into Many Different Types And Categories By Name. Admin Can Add New Products Into The Existing System With All Details Including Its Images.

Delete Products -

Admin Can Delete The Products Based On The Stock And Availability Of The Particular Product And Its Quantity.

Search Products –

Admin Has Kept A List Of Available Products On The Site.



MANAGE ORDERS

View Order –

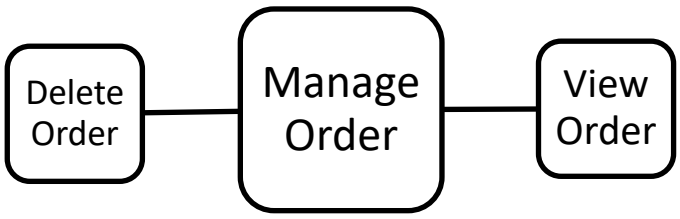
Admin Can View The Order Which Is Generated By The User / Customer.

Admin Can Verify The Details Of The Purchase.

Delete Order-

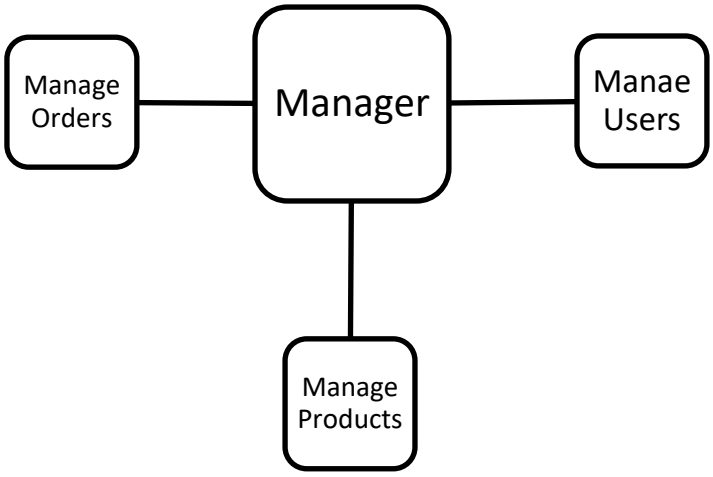
Admin Can Delete The Order From Order List Once It Is Delivered To The Customer.

He Can Also Delete The Order If The Customer Has Canceled The Order.



MANAGER

A Manager Is Considered As A Staff Who Can Manage Orders For The Time Begin. As A Future Update Manager May Give Facility To Add And Manage His Own Products. Managers Can Reduce The Word Load Of Admin. Mow Manager Has All The Privilege An Admin Having Except Managing Other Waiters. He Can Add Products And Users. He Can Also Check The Orders And Edit His Profile.



USERS / CUSTOMERS

A New User Or Customer Will Have To Register In The System By Providing Essential Details In Order To View The Products In The System. The Admin Must Accept A New User By Unblocking Him.

- **LOGIN-**

A User Must Login With His/Her Username And Password To The System After The Registration.

- **VIEW PRODUCTS-**

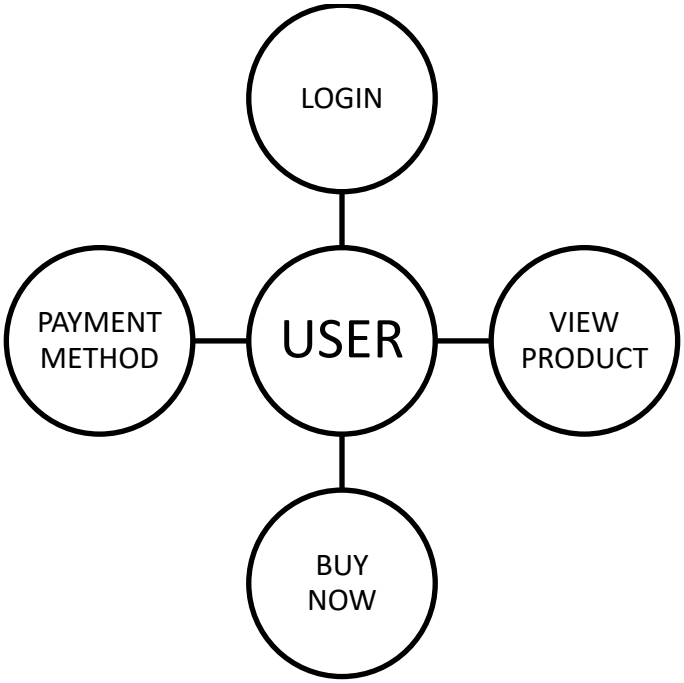
User Can View The List Of Products In Their Names After Successful Login.A Detailed Of A Particular Product With Product Name, Product Details, Product Images, Price Can Be Viewwd By Users

- **PAYMENT METHOD-**

After Buying The Product User Comes To Payment Method. He/She Has To Fill The Appropriate Details As Per Asked. After The Payment Method The Order Is Been Placed And Delivered With In Given Time.

❖ **BUY NOW-**

The User Can Buy The Desired Product From The Specific Category.



SQL(Structured Query Language)

Basically, SQL Stands For **Structured Query Language** Which Is Basically A Language Used By Databases. This Language Allows To Handle The Information Using Tables And Shows A Language To Query These Tables And Other Objects Related (Views, Functions, Procedures, Etc.). Most Of The Databases Like SQL Server, Oracle, Postgresql, Mysql, Mariadb Handle This Language (With Some Extensions And Variations) To Handle The Data.

With SQL You Can Insert, Delete, And Update Data. You Can Also Create, Delete, Or Alter Database Objects.

What Is SQL?

- SQL Stands For Structured Query Language
- SQL Lets You Access And Manipulate Databases
- SQL Became A Standard Of The American National Standards Institute (ANSI) In 1986, And Of The International Organization For Standardization (ISO) In 1987

What Are The Basics Of SQL?

- The Basic Use Of SQL For Data Professionals And SQL Users Is To Insert, Update, And Delete The Data From The Relational Database.
- SQL Allows The Data Professionals And Users To Retrieve The Data From The Relational Database Management Systems.
- It Also Helps Them To Describe The Structured Data.

What Are Tables In SQL?

- Tables Are Database Objects That Contain All The Data In A Database. In Tables, Data Is Logically Organized In A Row-And-Column Format Similar To A Spreadsheet. Each Row Represents A Unique Record, And Each Column Represents A Field In The Record

How Many Types Of Table Are There In SQL?

- There Are Three Types Of Tables: Base, View, And Merged. Every Table Is A Document With Its Own Title, Viewers, Saved Visualizations, And Set Of Data. The Data In Each Type Of Table Has Different Properties.

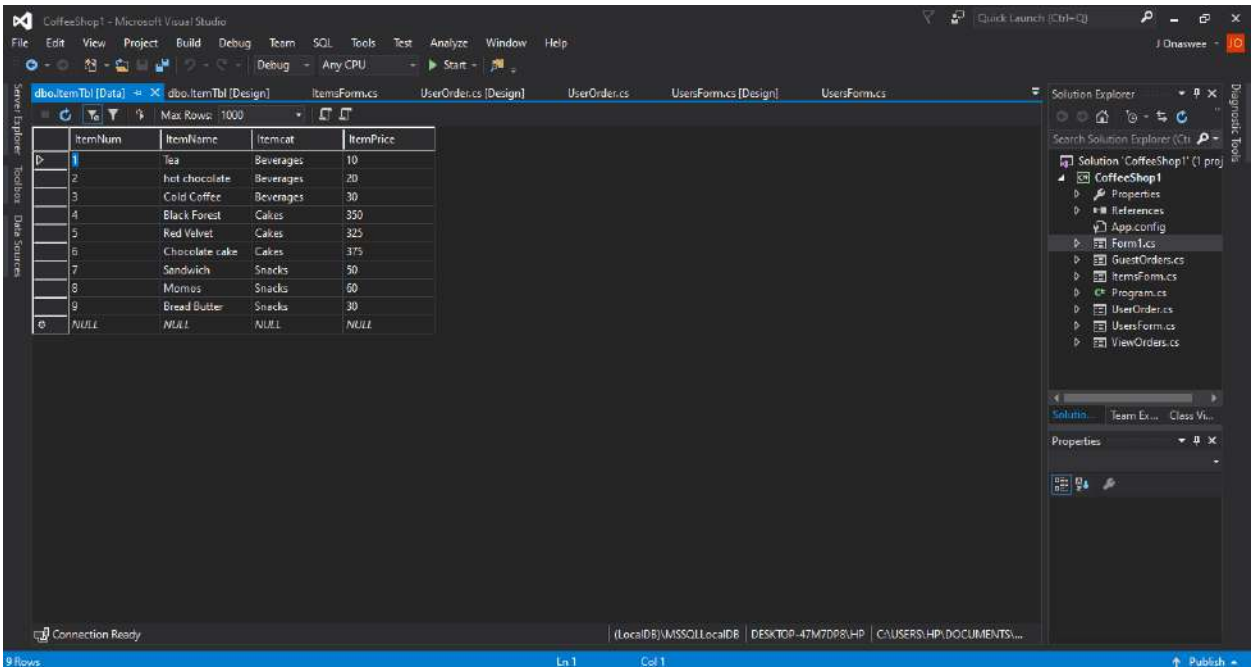
What Is A Table For Data?

- Data Tables Display Information In A Grid-Like Format Of Rows And Columns. They Organize Information In A Way That's Easy To Scan So That Users Can Look For Patterns And Develop Insights From Data. Data Tables Can Contain: Interactive Components (Such As Chips, Buttons, Or Menus)

DATABASE DESIGN

SQL DATABASE

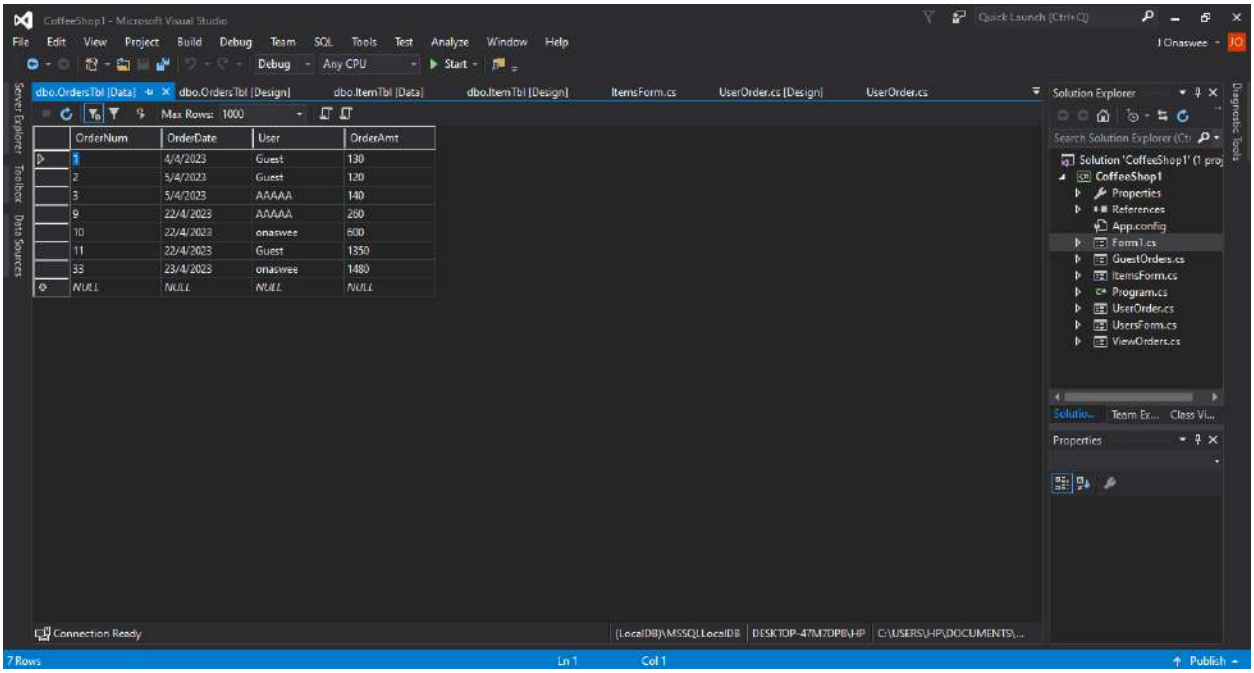
ITEM TABLE DATABASE CONNECTION



ItemNum	ItemName	Itemcat	ItemPrice
1	Tea	Beverages	10
2	hot chocolate	Beverages	20
3	Cold Coffee	Beverages	30
4	Black Forest	Cakes	350
5	Red Velvet	Cakes	325
6	Chocolate cake	Cakes	375
7	Sandwich	Snacks	50
8	Momos	Snacks	60
9	Bread Butter	Snacks	30
NULL	NULL	NULL	NULL

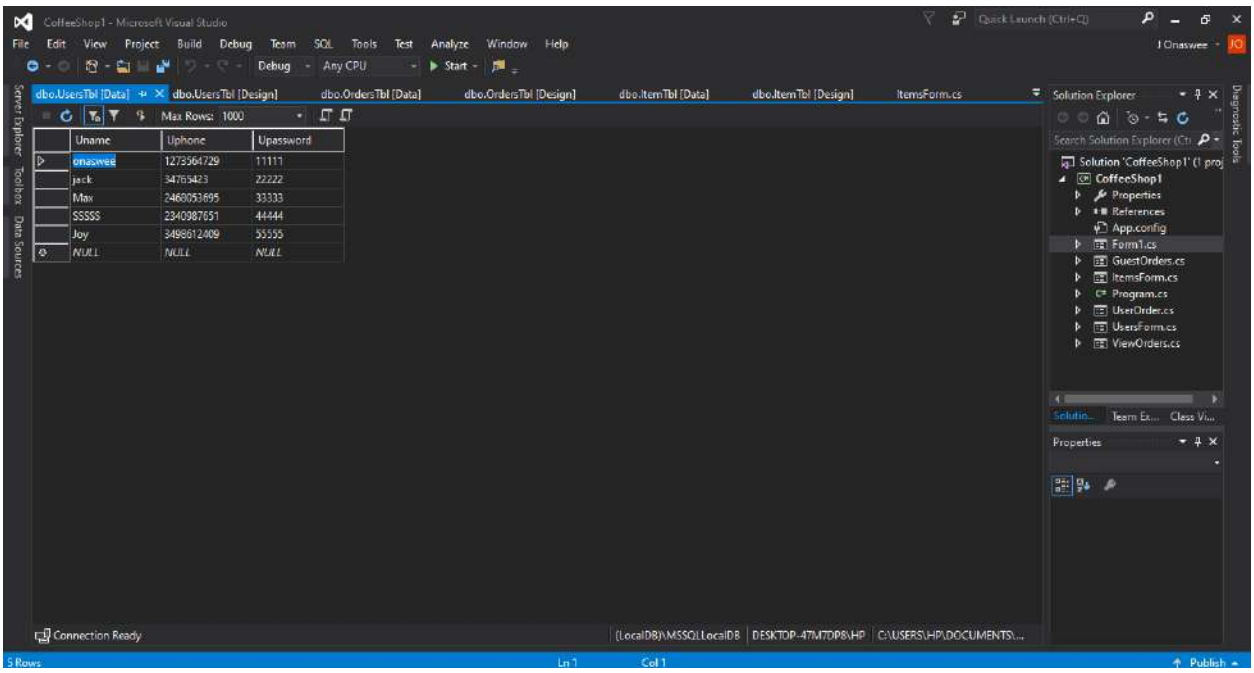
```
CREATE TABLE [dbo].[ItemTbl] (
    [ItemNum]   INT          NOT NULL,
    [ItemName]  VARCHAR (50) NOT NULL,
    [Itemcat]   VARCHAR (20) NOT NULL,
    [ItemPrice] INT          NOT NULL,
    PRIMARY KEY CLUSTERED ([ItemNum] ASC)
);
```

ORDERS TABLE DATABASE CONNECTION




```
CREATE TABLE [dbo].[OrdersTbl] (
    [OrderNum]  INT          NOT NULL,
    [OrderDate] VARCHAR (50) NOT NULL,
    [User]      VARCHAR (50) NOT NULL,
    [OrderAmt]  INT          NOT NULL,
    PRIMARY KEY CLUSTERED ([OrderNum] ASC)
                                  );
```

USER TABLE DATABASE CONNECTION




```
CREATE TABLE [dbo].[UsersTbl] (
    [Uname]     VARCHAR (50) NOT NULL,
    [Uphone]    VARCHAR (50) NOT NULL,
    [Upassword] NCHAR (10)   NOT NULL,
    PRIMARY KEY CLUSTERED ([Upassword] ASC)
);
```

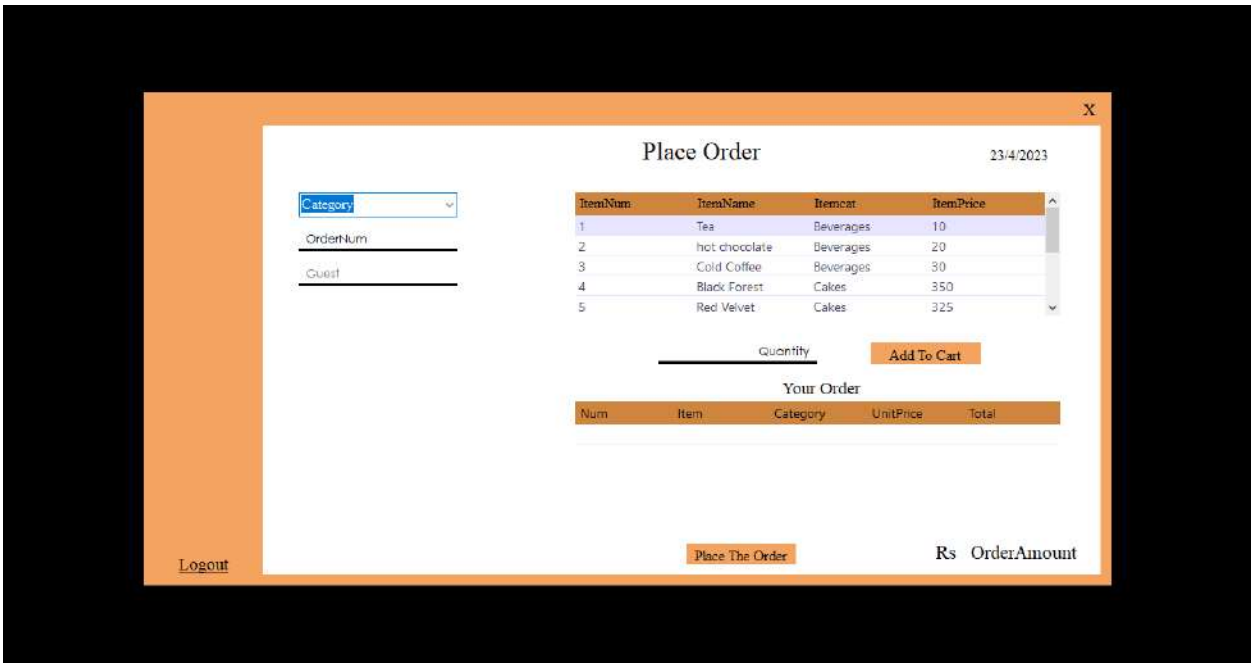
SCREEN LAYOUT

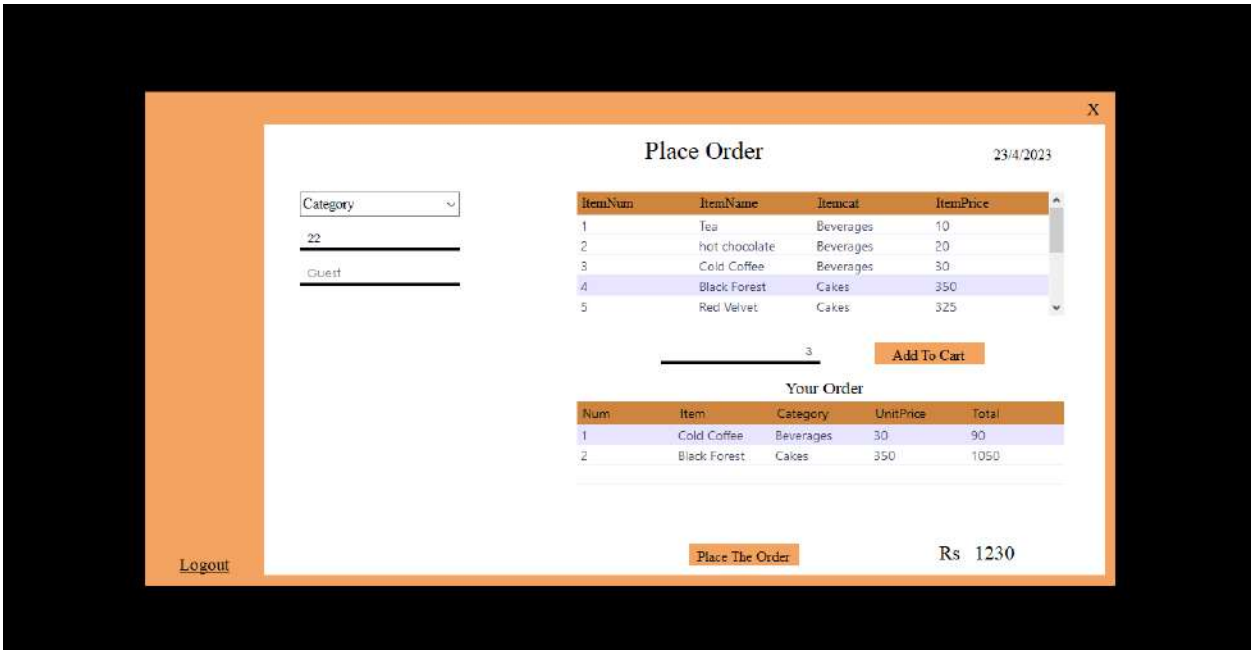
FORM1 PAGE

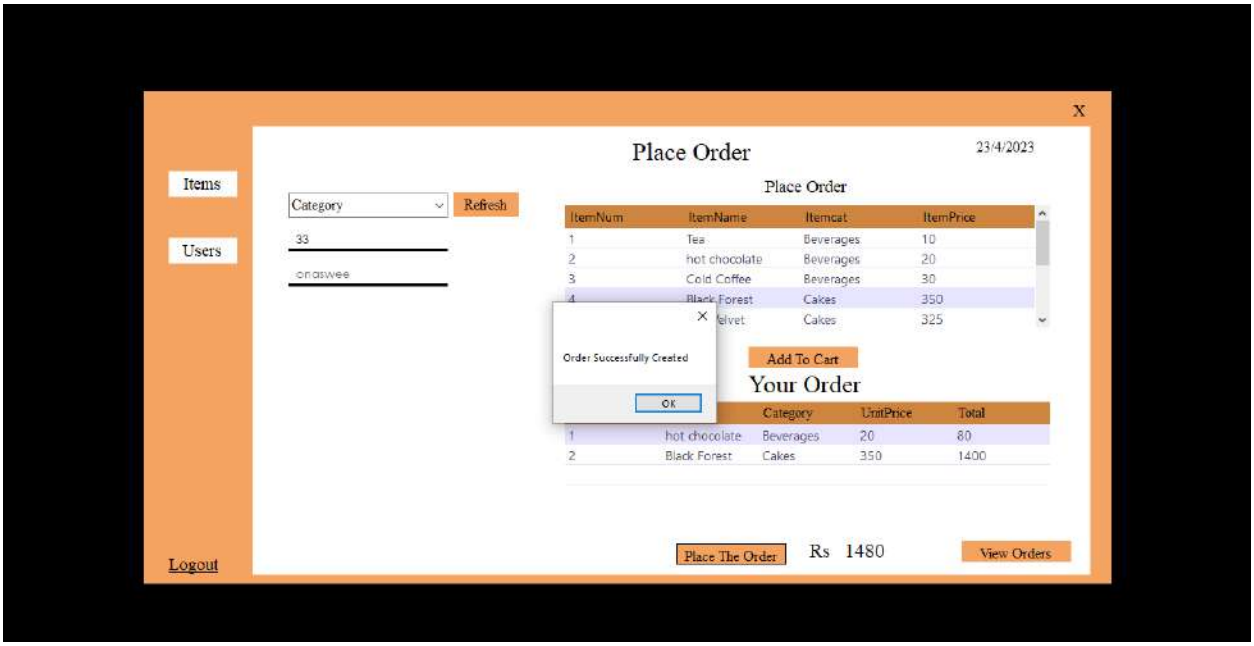

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace CoffeeShop1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        SqlConnection Con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HP\Documents\Coffeesh
opDB.mdf;Integrated Security=True;Connect Timeout=30");

        private void label7_Click(object sender, EventArgs e)
```

```
        {
            Application.Exit();
        }

        private void label4_Click(object sender, EventArgs e)
        {
            this.Hide();
            GuestOrders guest = new GuestOrders();
            guest.Show();
        }
        public static string user;
        private void bunifuThinButton21_Click(object sender, EventArgs e)
        {
            /* UserOrder uorder = new UserOrder();
             uorder.Show();
             this.Hide();
             */
            user = UnameTb.Text;
             if(UnameTb.Text == "" || PasswordTb.Text == "")
            {
                MessageBox.Show("Enter A Username Or Password");
            }
             else
            {
                Con.Open();
                SqlDataAdapter sda = new SqlDataAdapter("select count (*)
from UsersTbl  where Uname='" + UnameTb.Text + "' and Upassword='" +
PasswordTb.Text+ "'", Con);
                DataTable dt = new DataTable();
                sda.Fill(dt);
                if (dt.Rows[0][0].ToString() == "1")
                {
                    UserOrder uorder = new UserOrder();
                    uorder.Show();
                    this.Hide();
                }
                else
                {
                    MessageBox.Show("Wrong Username or Password");
                }
                Con.Close();
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }
    }
}
```

USER ORDER PAGE








```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace CoffeeShop1
{
    public partial class UserOrder : Form
    {
        public UserOrder()
        {
            InitializeComponent();
        }
        SqlConnection Con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HP\Documents\Coffeesh
opDB.mdf;Integrated Security=True;Connect Timeout=30");

        void populate()
        {
            Con.Open();
            string query = "select * from ItemTbl";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
```

```
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            ItemsGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        void filterbycategory()
        {
            Con.Open();
            string query = "select * from ItemTbl where Itemcat = '" +
categorycb.SelectedItem.ToString() + "'";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            ItemsGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        private void label4_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form1 login = new CoffeeShop1.Form1();
            login.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
            ItemsForm Item = new CoffeeShop1.ItemsForm();
            Item.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            this.Hide();
            UsersForm user = new CoffeeShop1.UsersForm();
            user.Show();
        }
        int num = 0;
        int price, total;
        string item, cat;

        private void button1_Click(object sender, EventArgs e)
        {
            if(QtyTb.Text == "")
            {
                MessageBox.Show("What is The Quantity of Item?");
            }
            else if(flag == 0)
            {
```

```
                MessageBox.Show("Select The Product To Be Ordered");
            }
            else
            {
                num = num + 1;
                total = price * Convert.ToInt32(QtyTb.Text);
                table.Rows.Add(num, item, cat, price, total);
                OrdersGV.DataSource = table;
                flag = 0;
            }
            sum = sum + total;
            OrderAmt.Text = ""+ sum;
        }
        DataTable table = new DataTable();
        int flag = 0;
        int sum = 0;

        private void comboBox1_SelectionChangeCommitted(object sender,
EventArgs e)
        {
            filterbycategory();
        }

        private void button5_Click(object sender, EventArgs e)
        {
            populate();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Con.Open();
            string query = "insert into OrdersTbl values(" + OrderNum.Text +
",'" + Datelbl.Text + "','" + SellerName.Text + "'," + OrderAmt.Text + ")";
            SqlCommand cmd = new SqlCommand(query, Con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Order Successfully Created");
            Con.Close();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            ViewOrders view = new ViewOrders();
            view.Show();
        }

        private void label7_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
```

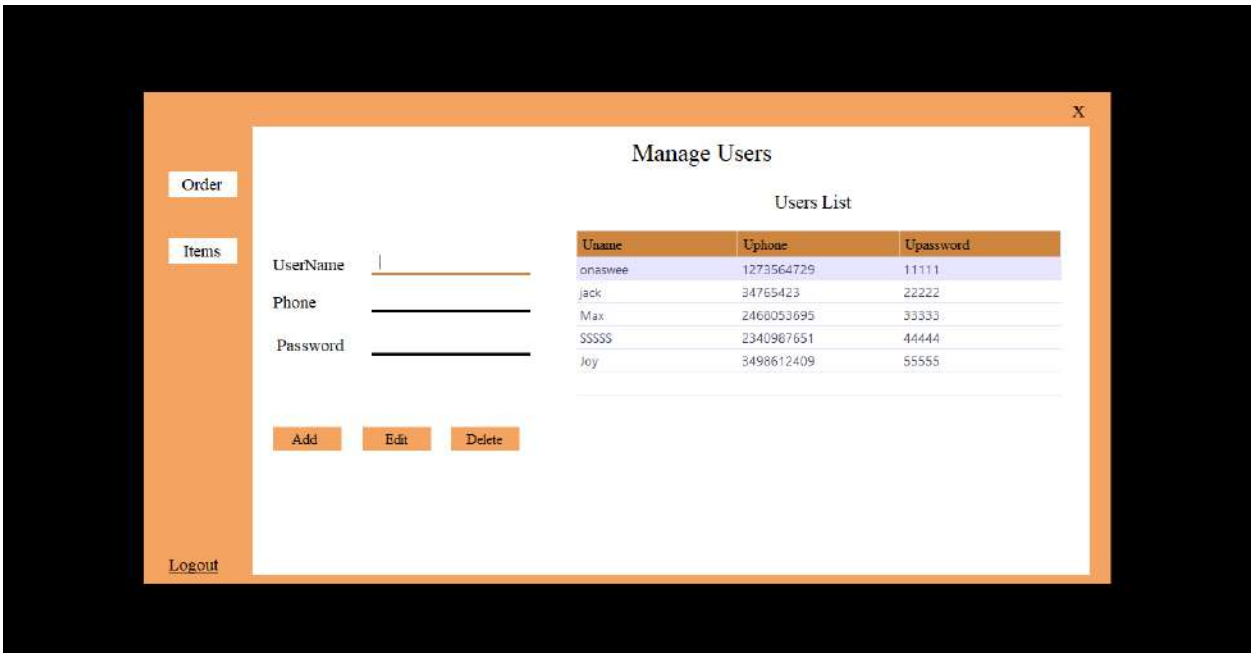
```
        private void categorycb_SelectedIndexChanged(object sender, EventArgs
e)
        {

        }

        private void UserOrder_Load(object sender, EventArgs e)
        {
            populate();
            table.Columns.Add("Num", typeof(int));
            table.Columns.Add("Item", typeof(string));
            table.Columns.Add("Category", typeof(string));
            table.Columns.Add("UnitPrice", typeof(int));
            table.Columns.Add("Total", typeof(int));
            OrdersGV.DataSource = table;
            Datelbl.Text = DateTime.Today.Day.ToString() + "/" +
DateTime.Today.Month.ToString() + "/" + DateTime.Today.Year.ToString();
            SellerName.Text = Form1.user;
        }

        private void ItemsGV_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            item = ItemsGV.SelectedRows[0].Cells[1].Value.ToString();
            cat = ItemsGV.SelectedRows[0].Cells[2].Value.ToString();
            price
=Convert.ToInt32(ItemsGV.SelectedRows[0].Cells[3].Value.ToString());
            flag = 1;
        }
    }
}
```

VIEW ORDER


```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace CoffeeShop1
{
    public partial class ViewOrders : Form
    {
        public ViewOrders()
        {
            InitializeComponent();
        }
        SqlConnection Con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HP\Documents\Coffeesh
opDB.mdf;Integrated Security=True;Connect Timeout=30");

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

```
        void populate()
        {
            Con.Open();
            string query = "select * from OrdersTbl";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            OrdersGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        private void ViewOrders_Load(object sender, EventArgs e)
        {
            populate();
        }
        private void OrdersGV_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            if (printPreviewDialog1.ShowDialog() == DialogResult.OK)
            {
                printDocument1.Print();
            }
        }
        private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
        {
            e.Graphics.DrawString("=====Order Summary=====", new
Font("Abeezee", 20, FontStyle.Bold), Brushes.Red, new Point(200,40));
            e.Graphics.DrawString("=====Order Summary=====", new
Font("Abeezee", 20, FontStyle.Bold), Brushes.Red, new Point(208,70));

e.Graphics.DrawString("Number:"+OrdersGV.SelectedRows[0].Cells[0].Value.ToStr
ing(), new Font("Abeezee", 15,FontStyle.Regular), Brushes.Black, new
Point(120, 135));
            e.Graphics.DrawString("Date:" +
OrdersGV.SelectedRows[0].Cells[1].Value.ToString(), new Font("Abeezee", 15,
FontStyle.Regular), Brushes.Black, new Point(120, 170));
            e.Graphics.DrawString("Seller:" +
OrdersGV.SelectedRows[0].Cells[2].Value.ToString(), new Font("Abeezee", 15,
FontStyle.Regular), Brushes.Black, new Point(120, 205));
            e.Graphics.DrawString("Amount:" +
OrdersGV.SelectedRows[0].Cells[3].Value.ToString(), new Font("Abeezee", 15,
FontStyle.Regular), Brushes.Black, new Point(120, 240));
            e.Graphics.DrawString("=====ThankYou=====", new Font("Abeezee",
20, FontStyle.Bold), Brushes.Red, new Point(208, 340));

        }
    }
}
```

MANAGE USER



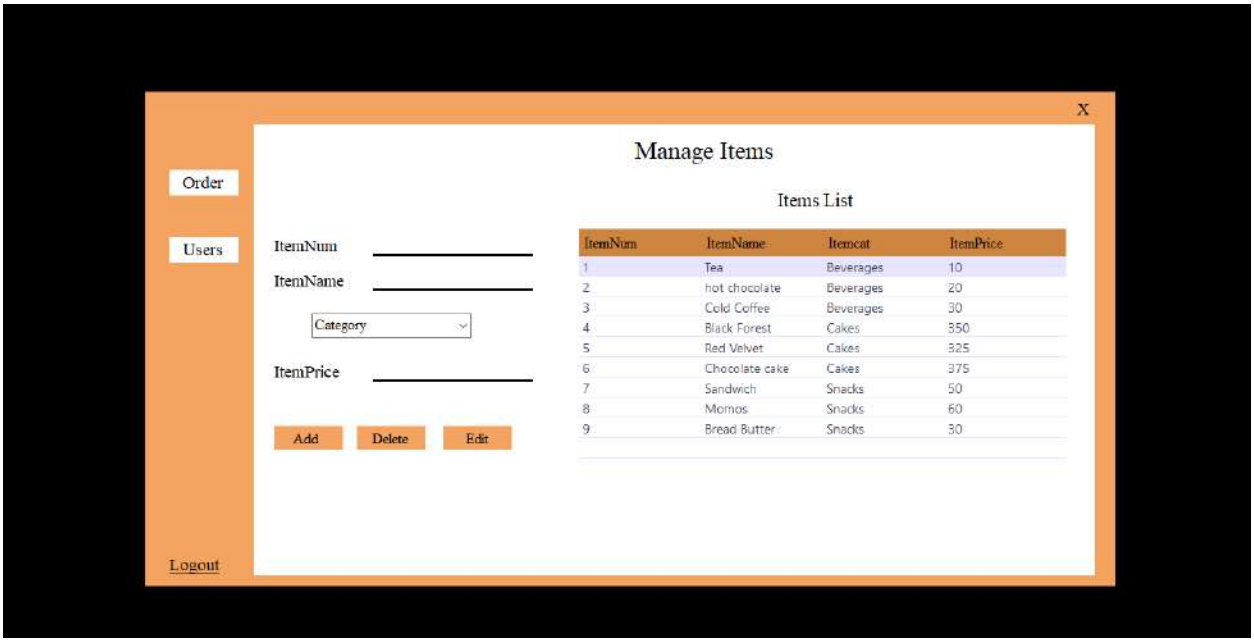

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace CoffeeShop1
{
    public partial class UserOrder : Form
    {
        public UserOrder()
        {
            InitializeComponent();
        }
        SqlConnection Con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HP\Documents\Coffeesh
opDB.mdf;Integrated Security=True;Connect Timeout=30");

        void populate()
        {
            Con.Open();
```

```
            string query = "select * from ItemTbl";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            ItemsGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        void filterbycategory()
        {
            Con.Open();
            string query = "select * from ItemTbl where Itemcat = '" +
categorycb.SelectedItem.ToString() + "'";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            ItemsGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        private void label4_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form1 login = new CoffeeShop1.Form1();
            login.Show();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
            ItemsForm Item = new CoffeeShop1.ItemsForm();
            Item.Show();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            this.Hide();
            UsersForm user = new CoffeeShop1.UsersForm();
            user.Show();
        }
        int num = 0;
        int price, total;
        string item, cat;

        private void button1_Click(object sender, EventArgs e)
        {
            if(QtyTb.Text == "")
            {
                MessageBox.Show("What is The Quantity of Item?");
            }
```

```
            else if(flag == 0)
            {
                MessageBox.Show("Select The Product To Be Ordered");
            }
            else
            {
                num = num + 1;
                total = price * Convert.ToInt32(QtyTb.Text);
                table.Rows.Add(num, item, cat, price, total);
                OrdersGV.DataSource = table;
                flag = 0;
            }
            sum = sum + total;
            OrderAmt.Text = ""+ sum;
        }
        DataTable table = new DataTable();
        int flag = 0;
        int sum = 0;

        private void comboBox1_SelectionChangeCommitted(object sender,
EventArgs e)
        {
            filterbycategory();
        }

        private void button5_Click(object sender, EventArgs e)
        {
            populate();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Con.Open();
            string query = "insert into OrdersTbl values(" + OrderNum.Text +
",'" + Datelbl.Text + "','" + SellerName.Text + "'," + OrderAmt.Text + ")";
            SqlCommand cmd = new SqlCommand(query, Con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Order Successfully Created");
            Con.Close();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            ViewOrders view = new ViewOrders();
            view.Show();
        }

        private void label7_Click(object sender, EventArgs e)
        {
            Application.Exit();
```

```
        }

        private void categorycb_SelectedIndexChanged(object sender, EventArgs
e)
        {

        }

        private void UserOrder_Load(object sender, EventArgs e)
        {
            populate();
            table.Columns.Add("Num", typeof(int));
            table.Columns.Add("Item", typeof(string));
            table.Columns.Add("Category", typeof(string));
            table.Columns.Add("UnitPrice", typeof(int));
            table.Columns.Add("Total", typeof(int));
            OrdersGV.DataSource = table;
            Datelbl.Text = DateTime.Today.Day.ToString() + "/" +
DateTime.Today.Month.ToString() + "/" + DateTime.Today.Year.ToString();
            SellerName.Text = Form1.user;
        }

        private void ItemsGV_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            item = ItemsGV.SelectedRows[0].Cells[1].Value.ToString();
            cat = ItemsGV.SelectedRows[0].Cells[2].Value.ToString();
            price
=Convert.ToInt32(ItemsGV.SelectedRows[0].Cells[3].Value.ToString());
            flag = 1;
        }
    }
}
```

MANAGE ITEMS



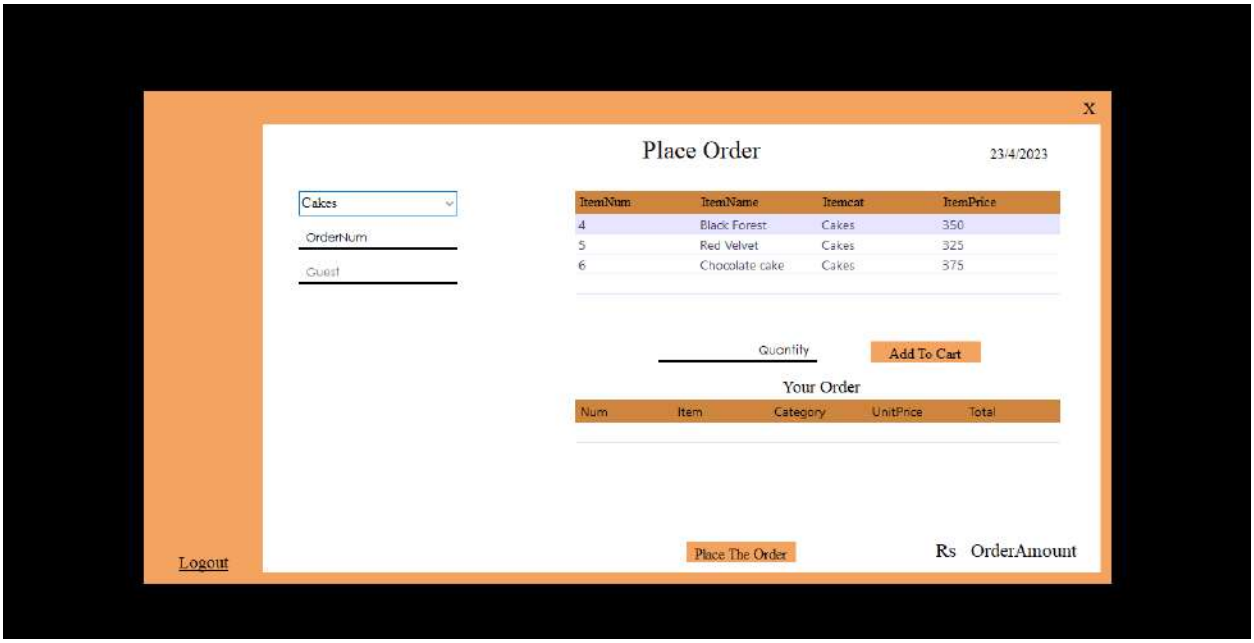

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace CoffeeShop1
{
    public partial class ItemsForm : Form
    {
        public ItemsForm()
        {
            InitializeComponent();
        }
        SqlConnection Con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HP\Documents\Coffeesh
opDB.mdf;Integrated Security=True;Connect Timeout=30");
        void populate()
        {
            Con.Open();
            string query = "select * from ItemTbl";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
```

```
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            ItemsGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        private void ItemsForm_Load(object sender, EventArgs e)
        {
            populate();
        }

        private void panel1_Paint(object sender, PaintEventArgs e)
        {

        }

        private void button4_Click(object sender, EventArgs e)
        {

        }

        private void button3_Click(object sender, EventArgs e)
        {
            this.Hide();
            UserOrder order = new UserOrder();
            order.Show();

        }

        private void label4_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form1 login = new Form1();
            login.Show();
        }

        private void label7_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (ItemNameTb.Text == "" || ItemNumTb.Text == "" || PriceCb.Text
== "")
            {
                MessageBox.Show("Fill All The Data");
            }
            else
            {
```

```
                Con.Open();
                string query = "insert into ItemTbl values(" + ItemNumTb.Text
+ ",'" + ItemNameTb.Text + "','" + CatCb.SelectedItem.ToString() +
"',"+PriceCb.Text+")";
                SqlCommand cmd = new SqlCommand(query, Con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Item Successfully Created");
                Con.Close();
                populate();
            }
        }

        private void ItemsGV_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            ItemNumTb.Text =
ItemsGV.SelectedRows[0].Cells[0].Value.ToString();
            ItemNameTb.Text =
ItemsGV.SelectedRows[0].Cells[1].Value.ToString();
            CatCb.SelectedItem =
ItemsGV.SelectedRows[0].Cells[2].Value.ToString();
            PriceCb.Text = ItemsGV.SelectedRows[0].Cells[3].Value.ToString();
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (ItemNumTb.Text == "")
            {
                MessageBox.Show("Select The Item to be Deleted");
            }
            else
            {
                Con.Open();
                string query = "delete from ItemTbl where ItemNum = '" +
ItemNumTb.Text + "'";
                SqlCommand cmd = new SqlCommand(query, Con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Item Successfully Deleted");
                Con.Close();
                populate();

            }
        }

        private void button5_Click(object sender, EventArgs e)
        {
            if (ItemNumTb.Text == "" || ItemNameTb.Text == "" || PriceCb.Text
== "")
            {
                MessageBox.Show("Fill All The Fields");
```

```
            }
            else
            {
                Con.Open();
                string query = "update ItemTbl set ItemName='" +
ItemNameTb.Text + "',Itemcat='" + CatCb.SelectedItem.ToString() + "' where
ItemPrice =" + PriceCb.Text + ""; ;
                SqlCommand cmd = new SqlCommand(query, Con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("Item Successfully Updated");
                Con.Close();
                populate();
            }
        }
    }
}
```

GUEST ORDER




```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
using System.Data.SqlClient;
namespace CoffeeShop1
{
    public partial class GuestOrders : Form
    {
        public GuestOrders()
        {
            InitializeComponent();
        }
        SqlConnection Con = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\HP\Documents\Coffeesh
opDB.mdf;Integrated Security=True;Connect Timeout=30");

        void populate()
        {
            Con.Open();
            string query = "select * from ItemTbl";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            ItemsGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        void filterbycategory()
        {
            Con.Open();
            string query = "select * from ItemTbl where Itemcat = '" +
categorycb.SelectedItem.ToString() + "'";
            SqlDataAdapter sda = new SqlDataAdapter(query, Con);
            SqlCommandBuilder builder = new SqlCommandBuilder(sda);
            var ds = new DataSet();
            sda.Fill(ds);
            ItemsGV.DataSource = ds.Tables[0];
            Con.Close();
        }
        private void label7_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void panel1_Paint(object sender, PaintEventArgs e)
        {

        }

        private void label4_Click(object sender, EventArgs e)
        {
            this.Hide();
            Form1 login = new CoffeeShop1.Form1();
```

```
            login.Show();
        }

        private void GuestOrders_Load(object sender, EventArgs e)
        {
            populate();
            table.Columns.Add("Num", typeof(int));
            table.Columns.Add("Item", typeof(string));
            table.Columns.Add("Category", typeof(string));
            table.Columns.Add("UnitPrice", typeof(int));
            table.Columns.Add("Total", typeof(int));
            OrdersGv.DataSource = table;
            Datelbl.Text = DateTime.Today.Day.ToString() + "/" +
DateTime.Today.Month.ToString() + "/" + DateTime.Today.Year.ToString();
        }
        int num = 0;
        int price, total;
        string item, cat;
        DataTable table = new DataTable();
        int flag = 0;
        int sum = 0;

        private void ItemsGV_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            item = ItemsGV.SelectedRows[0].Cells[1].Value.ToString();
            cat = ItemsGV.SelectedRows[0].Cells[2].Value.ToString();
            price =
Convert.ToInt32(ItemsGV.SelectedRows[0].Cells[3].Value.ToString());
            flag = 1;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Con.Open();
            string query = "insert into OrdersTbl values(" + OrderNumTb.Text
+ ",'" + Datelbl.Text + "','" + SellerNameTb.Text + "'," +OrderAmt.Text+")";
            SqlCommand cmd = new SqlCommand(query, Con);
            cmd.ExecuteNonQuery();
            MessageBox.Show("Order Successfully Created");
            Con.Close();

        }

        private void button1_Click(object sender, EventArgs e)
        {
             if(QtyTb.Text == "")
            {
                MessageBox.Show("What is The Quantity of Item?");
            }
```

```
            else if(flag == 0)
            {
                MessageBox.Show("Select The Product To Be Ordered");
            }
            else
            {
                num = num + 1;
                total = price * Convert.ToInt32(QtyTb.Text);
                table.Rows.Add(num, item, cat, price, total);
                OrdersGv.DataSource = table;
                flag = 0;
            }
            sum = sum + total;
            OrderAmt.Text = ""+sum;
        }

        private void categorycb_SelectionChangeCommitted(object sender,
EventArgs e)
        {
            filterbycategory();
        }
    }
}
```

PROGRAM.CS CODE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CoffeeShop1
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

SYSTEM IMPLEMETAION

The Following Steps Were Carried Out In Implementation Phase.

- Conduct Training : The Training Was Conducted For The Employees Of The Company To Make Them Familiar With The System.

- Bug Fixing And Documentation: Any Errors That Occurred Were Solved And Documented.

- Install The System: The System Was Then Installed.

SYSTEM MAINTENCE

The Maintenance Of Software Is The Time Period In Which The Software Is Software Product Performes Useful Work. Maintatenance Activites Involve Making Enhancement Activites To The, Adapting Product To New Environment And Correcting Problems. Software Enhancement May Involve Providing New Functional Capabilities, Improving User Display And Modes Of Interaction.

Adaptation Of Software To A New Environment May Involve Moving The Software To A Different Machine. Problems Correction Involves Modification And Revalidation Of Software To Correct Errors. The Four Types Of Maintenance Activites Are Described Below:

- Corrective Maintance
- Preventive Maintenance
- Perfective Maintaenance
- Adaptive Maintenance

Corrective Maintenance:-

Corrective Maintenance Can Be Defined As The Maintenance Which Is Required When An Item Has Failed Or Worn Out, To Bring It Back To Working Order.

Adaptive Maintenance:-

Adaptive Maintenance Is An Activity That Modifies Software To Properly Interface With The Changing Environment.

Perfective Maintenance:-

Perfective Maintenance Is Performed To Satisfy User Request Such As New Capabilites, Modification To Existing Function And General Enhancements.

Preventive Maintenance:-

Preventive Maintanenance Occurs When Software Is Changed To Improved Future Maintainability Or To Provide A Better Basic For Future Enhancements.

PROJECT TESTING

Software Testing Strategies

Testing is a set of activities that can be planned in advanced and conducted systematically. A strategy for software testing must accommodation low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements

There are three types of testing strategies

1. Unit test
2. Integration test
3. Performance test

Unit Testing:

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

Integration Testing:

Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

Performance Testing:

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

CONCLUSION

The **"CAFE MANAGEMENT SYSTEM"** has been successfully completed. The goal of the system is achieved and problems are solved. The package is developed in a manner that it is user friendly and required help is provided at different levels.

The project can be easily used in the process of decision making. Different types of reports can be generated which help the management to take correct decision and reduce the time delay which automatically increases the company's work standards as well as the economical state of the company.

This system never decreases the manpower but helps the development of available manpower and optimizes the manpower by which company's standards and capabilities can be scaled to higher dimensions.

BIBLOGRAPHY

The following books were referred during the analysis and execution phase of the project

- **SQL FOR PROFESSIONALS**

- **Microsoft Visual Studios 2015**

Websites:

- **www.google.com**
- **www.youtube.com**
- **www.w3schools.com**