A
Project Report
On
"TILES SHOP MANAGEMENT SYSTEM"
(B.C.A.-307)
Submitted
In
Partial fulfilment of the requirements for the award
of the degree of
Bachelor of Computer Application-3nd Year
For
Session: 2023-24

At
SAMADHAN COLLEGE BEMETARA
Affiliated to
HEMCHAND YADAV UNIVERSITY, DURG (C. G.)

Guided by
MR. RAJENDRA VERMA

Submitted by
SAURABH SAHU

CERTIFICATE OF APPROVAL

This is to certify that the Project work entitled "**Tiles Shop Management System**" is carried out Mr. **SAURABH SAHU.** A student of BCA-III year at **Samadhan college** is hereby approved as a credible work in the discipline of Computer Science and Application for the award of degree of Bachelor of Computer Application during the year **2023-24** from Hemchand Yadav Vishwavidyalaya, Durg (C.G.).

Head

Computer Science Dept.

CERTIFICATE OF EVALUATION

This is to certify that the Project work entitled "**Tiles Shop Management System**" is carried out **Mr. Saurabh Sahu**, a student of BCA-III year at SAMADHAN COLLEGE, BEMETARA**(C.G.)**, after proper evaluation and examination, is hereby approved as a credible work in the discipline of Computer Science and Application and is done in a satisfactory manner for its acceptance as a requisite for the award of degree of Bachelor of Computer Application during the year **2023-24** from Hemchand Yadav Vishwavidyalaya, Durg (C.G.).

Internal Examiner

External Examiner

TAMANNA TRADERS

By Pass Road, Ward No. 3, Berla, Dist. – Bemetara (C.G.)

Phone : 7987540492

CERTIFICATE

This is to certify that **Mr. Saurabh Sahu**

Entitled **Tiles Shop Management System** the project embodies result of the original work and studies out by himself and the content of the project do not from the basis of the award of any other degree to the candidate or to anybody else.

Place :

Date :

DECLARATION

This is to certify that the Project work entitled "**Tiles Shop Management System**", which is submitted by me in partial fulfilment for the award of degree of Bachelor of Computer Application, **Samadhan College, Bemetara(C.G.)**, comprises the original work carried out by me.

I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full for the award of any other degree or diploma in this Institute or any other Institute or University.

Place : Name :

Date : Roll No. :

ACKNOWLEDGEMENT

We are very delighted on the accomplishment to the project in visual basic, which was very educational and practically beneficial. It is due to his encouragement and persistent motivation that I could extent the scope of project to much useful data processing report. We have specially thankful to MR. RAJENDRA VERMA the SAMADHAN COLLEGE, BEMETARA who helps us by his deep knowledge and practical experience in computer science, rendered all possible in fulfillment of approach. So, system development is the pooling of talents.

SELF CERTIFICATE

: This is to certify that the Practical work entitled "(TILES SHOP)(BCA-307)", which is submitted by me in partial fulfilment for the award of degree of Bachelor of Computer Application 3rd Year during the year 2023-24 from Samadhan College Bemetara, comprises the original work carried out by me.

I further declare that work reported in this project has not been submitted and will not be submitted, either in part or in full for the award of any other degree or diploma in this institute or any other Institute or University.

PLACE: Bemetara NAME: SAURABH SAHU

DATE ROLLNo: 42210457

FORMAT OF THE PROJECT REPORT

- **Abstraction of the project**
- **Objective & scope of the project**
- **Theoretical background of the project**
- **Definition of problem**
- **System analysis and design**
- **System planning (PERT Chart)**
- **Methodology adopted, system implementation front end and back end**
- **Hardware and software details**
- **System maintenance & evaluation**
- **Cost - benefit analysis**
- **Detailed life cycle of the project**
- **ERD**
- **DFD**
- **Input and output design**
- **Database design**
- **Form design**
- **Coding**
- **Process involved**
- **Methodology used for testing**
- **Test report, printout of the code sheet**
- **Conclusion**
- **Soft copy of the project on CD**

ABSTRACTION OF THE PROJECT

- Background And Motivation

Everyone wants to make their home clean and shiny. They can do the cleaning dust through regular cleaning but it does not make it shiny. It needs something else like any upper coating to their walls and floor such that they can be easily maintained and make it clear the way they want it makes the luminous thing for a long time giving it rich look on the whole.

Now where one can buy these tiles and most important thing which design they buy the tiles. This becomes the most difficult task to choose that in a proper manner. Here the customers can be choosing good quality tile according to the user's needs. Tile shop management system deals with managing of a Tile shop which have several branches across the state which also provide whole-sale of the product.

- Project Scope

This system can be implemented so that it is easier for the owner and the manager can analyze financial status and can increase the sales which is directly through the shop and through online so the income will be increased.

OBJECTIVE & SCOPE OF THE PROJECT

In order to computerize any system, we should first define the objectives of the system. In centre automation system the
Objectives are as follows:

- Details with sailing of Tiles and maintaining stock.
- Report generation- monthly, weekly, day-by-day.
- Report keeping customer details of every customer.

The main objective behind our project is to relief the burden on staff n customer to some extent through the use of this package.

Have tried to make this package as user friendly as possible within our limited resource. Much thinking has gone into deciding the methodology to be followed to obtain the optimum design without any failures.

THEORETICAL BACKGROUND OF THE PROJECT

Over the past decade, the Enterprises System (ES) industry has proven to be an enormous growth market. The broad adoption and enormous attention from the business world in ES is considered the most important development in the corporate use of information technology during the 1990s. The ES market has become significant. The last couple of years the attention paid to this phenomenon in literature are slowly increasing again as well.

The Theoretical Background was followed by an analysis. To design a performance measurement tool, first a good analysis of the current situation and the information need is necessary. The analysis focused on the mapping of the typical business process steps taking place within TILES SHOP Management System.

DEFINITION OF PROBLEM

The problem definition is usually set up at the end of the problem analysis phase. The starting point of a problem is the information gathered in the problem analysis stage The problem with earlier system is that it requires more time to implement. And it does not provide any accuracy in determining the total amount of the product to be sold, it also requires enough to do some short paper work to generate a report for that particular product.

The method of earlier way of doing work is very time consuming process. And the result which may be a trouble in away to evaluate selling prices. There was more paper work in earlier method. The Cashier even gets tired by calculating the cost of the tiles. If a person wants to keep track of all the records then it require more space to do so and if he wants to see the previous record it also take more time to finding a record.

SYSTEM ANALYSIS AND DESIGN

System Analysis Is Vast Field OF Study Through that a system analyst puts his thought and searched for the best solution of the problem. This is the stage where we are going to collect facts, diagnosing the problem and using the information to recommended improvement of the system. This project is using a system that gives information about the institute and tells the user about the services provided by them. In this phase we will be dealing in the following aspects:-

- Preliminary Study
- Feasibility Study
- Fact - Findings

- **Preliminary Study :-**

The first step in the system development life cycle preliminary Investigation to determine the feasibility of the system the work Preliminary investigation is to collect the information that helps us to evaluate the merits of the project request and make an information judgment about the feasibility of the proposed project. While working with preliminary investigation once should accomplish the following Objectives: -

- Clarify and understood the project request
- Determine the size of project
- Asses cost and benefits of alternative approaches.
- Determine the technical and operational feasibility.

Here it becomes important that the project should be examined and clarified properly before considering system investigation.

The cost of the project is going to be low because his does not need a number of programmers and a number of software rather this project can be accomplished easily. On technical side this project is feasible because this project did not need a number of professional and team member, thus this project is easy to handle. This project is operational feasible as this do not have any sort of complicated operational to perform. Only a team of two or three person can complete this project.

- **Feasibility Study:-**

Feasibility is the determination of whether or not a project is worth doing the process followed in making these determinations is called feasibility study. This type of study determines id as projects can and should be taken. In conducting the feasibility study, these seven distinct but interrelated feasibility study are considered:-

An initial investigation terminates in a proposal that determines whether an alternative system is feasible.

This is known as feasibility study. Feasibility study can be categorized into three major parts.

Technical Feasibility:-

This is concerned with specifying equipment and software that with successfully satisfy the user requirement. In examining the technical feasibility, the configuration of the System is given importance than the actual hardware: -

- The proposed system has a technical capacity required to hold the data.

- These projects are Efficient and Respond Quickly for various enquiries regardless of number or location of user.
- The system proposed can be expanded easily and efficiently. whenever required.

Operating Feasibility Study:-

The operational feasibility in mainly related to human organizational And political aspect. Regarding the web project we should think that what Changes are brought with the system. What new skills will be required to Existing staff? If not can they be trained in due course of time?

The management of the organization has fully supported us to bring up project and the data security in this project provided by setting up to password procedure so that only the authorized in user can access the system.

Economical Feasibility Study:-

Economical analysis uses technique for evaluating the effectiveness of a proposed system. It determines the benefits and saving that are expected from the proposed system and compares them with cost. If benefits outweigh cost, the system is implemented else alternative system analysis because it is cost that matter more in setting any setting any system.

Our Project is economical feasible as:

- It has computerized, the paper work and also is reduced to way large extends.
- Due to the processing speed of computer, we can extract the desired information in a fraction of second.

METHODOLOGY ADOPTED, SYSTEM IMPLEMENTATION FRONT END

INTRODUCTION OF VISUAL BASIC DOT NET 2022

Visual studio products with same version and in the language-visual C++ and visual basic version 6.0 in English, for example-it can be installed are separately on the same workstation. Such installing is supported by Microsoft. The visual studio 6.0 installing wizard detects if other version of the versions of the 6.0 product line. Have been installed on a developers work station.

Mixing different language versions, point release, or product tiers on the same work station is generally not supported. This means that the installation may, fail, one of more of visual studio 6.0 products may not work even if the installation succeeds, or in the worst case you may not be able to install any of the products completely.

Visual basic is the senior of the family of visual basic products that includes visual basic for application and visual basic scripting edition (v.b.script).while most of the codes that we write in vb. cane shared with application written in v.b for application or vb. script, there is some expectation.

Visual basic for application is a single, common application scripting language and environment that users and developers can leverage across their windows desktop.v.b for application is including in Microsoft office and other Microsoft application. It is also licensed to other software vendor and included in a wide range of products.

In general, code writer in v.b is portable in v.b for application as long as it doesn't referance these elements.

- Visual Basic is a tool that allows you to develop Windows (Graphic User Interface - GUI) applications. The applications have a familiar appearance to the user.

- Visual Basic is event-driven, meaning code remains idle until called upon to respond to some event (button pressing, menu selection...). Visual Basic is governed by an event processor. Nothing happens until an event is detected.

Once an event is detected, the code corresponding to that event (event procedure) is executed. Program control is then returned to the event Processor.

Some Features of Visual Basic

- Full set of objects - you 'draw' the application
- Lots of icons and pictures for your use
- Response to mouse and keyboard actions
- Clipboard and printer access
- Full array of mathematical, string handling, and graphics functions
- Can handle fixed and dynamic variable and control arrays
- Sequential and random access file support
- Useful debugger and error-handling facilities
- Powerful database access tools
- ActiveX support
- Package & Deployment Wizard makes distributing your applications

Simple Visual basic 6.0 is the next version of Visual Basic 5.0 Rather than simply adding sum new features to Visual Basic, Microsoft has reengineered the product to make it easier the ever before to write distributed application such as web and enterprise entire system. Visual Basic 6.0 has two new forms packages (window forms and web forms) a new version of ADO for accessing connected data source; and streamlined language removing legacy required.

BACK END

INTRODUCTION OF MS - ACCESS

The fundamental concept under line access database is that data is stored in tables in your access application tables are comprised of rows and columns of the data much like an excel worksheet. Each table represents a single entity such as a person as product as you want access you will expand considerable designing time and refining the table in your access application. Table design an implementation are two characteristics that distinguish database development for most other activities you may peruse.

In access database is an overall container for the data and associate object it is more than the collection of tables however a database includes many types of object including query from report etc.

Access work a single database at a time as you open access in single database is presented for you to use you may open several copies of access at the same time work with more than one database.

Microsoft access we can create, view, modifies and deletes tables. view, reports, forms, macros, modules. And all type database creation operations we can perform with Microsoft access.

In Microsoft access there are many features. They are:

- Easily create a client/server application a Microsoft access (.adp) is a new type of access file that provides efficient, native mode access to a Microsoft sql, server database through the ole db component architecture. Using an access project, you can easily create a client server application.

- Work with an access project working with a Microsoft access project is very similar to working with an access database. The process of creating forms, reports, data access pages, macros, and modules is the same. Once you connect to an sql server database,

you can view, create, modify, and delete tables, views, stored procedures, and database diagrams using the microsoft sql server design tools.

- Use microsoft data engine (msde) msde is a new technology that provides local data storage compatible with microsoft sql server 7.0. Think of msde as a client/server data engine alternative to the file server microsoft jet database engine. It is designed and optimized for use on smaller computer systems, such as a single user computer or small workgroup server.

upsize data and objects by using the upsizing wizard the upsizing wizard up sizes a microsoft access database (.mdb) to a new or existing Microsoft sql server 6.5 and 7.0 database or new microsoft access project (adp) by upsizing the data and data definitions and migrating database objects.

Database technology is one of the most rapidly growing areas of computers and information science is a computer base is repository for stored data. While database system is a computer based record keeping system, the database management system is software, which is responsible for a layer between the physical database itself and the user of the system. The sliver line is that the data carries all the significant and this data is manifested In to different models physical and logical between which the logical models takes precedence.

MS-ACCESS is the product of MICROSOFT, which is a good, database which all were using, it can be seen that it can store up to 70000 records, it is very useful.

HARDWARE & SOFTWARE DETAILS

HARDWARE DETAILS :-

- **System** : Asus Vivobook
- **Monitor** : 15.6” monitor
- **Cache** : 6 mb
- **Processor** : Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
- **Installed RAM** : 8.00 GB (7.83 GB usable)

MINIMUM HARDWARE REQUIRED FOR IMPLEMENTATION :-

- **Cpu** : pentinum iv processor
- **Monitor** : any
- **Mouse** : any
- **Keyboard :** any
- **Cache** : 512 kb above
- **memory** : 256 mb above

SOFTWARE DETAILS :-

- **Front end** : microsoft visual studio 2022
- **Back end** : Microsoft office access 2007
- **Operating system** : windows 10
- **Edition** : Windows 10 Home Single Language
- **System type** : 64-bit operating system, x64-based processor

DETAILED LIFE CYCLE OF THE PROJECT

E-R DIAGRAM

Entity-Relationship model in software engineering is an abstract Way to describe a database. This article refers to the techniques proposed in Peter Chen's 1976 paper.[1] However, variants of the idea existed previously.[2] and have been devised subsequently such as super type and subtype data entities [3] and commonality relationships.

Also called an Entity-relationship model, a graphical representation of entities and their relationship to each other, typically used in computing in regard to the organization of data within database or information system. An entity is a piece of data is shared between entities.

Entity

An entity is an object or concept about which you want to store information.

Weak Entity

A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attribute alone..

Key attribute

A key attribute is the unique, distinguishing characteristics of the entity. For example, an employee's social security number might be the employee's key attribute.

Multivalued attribute

A multivalve attribute can have more than one value. For exp. An employee entity can have multiple skill values.

Derived attribute

A derived attribute is based on another attribute. For example. An employee's monthly salary is based on the employee's annual salary.

Relationship

Relationship illustrates how two entities share information in the database structure, learn how to draw relationship:

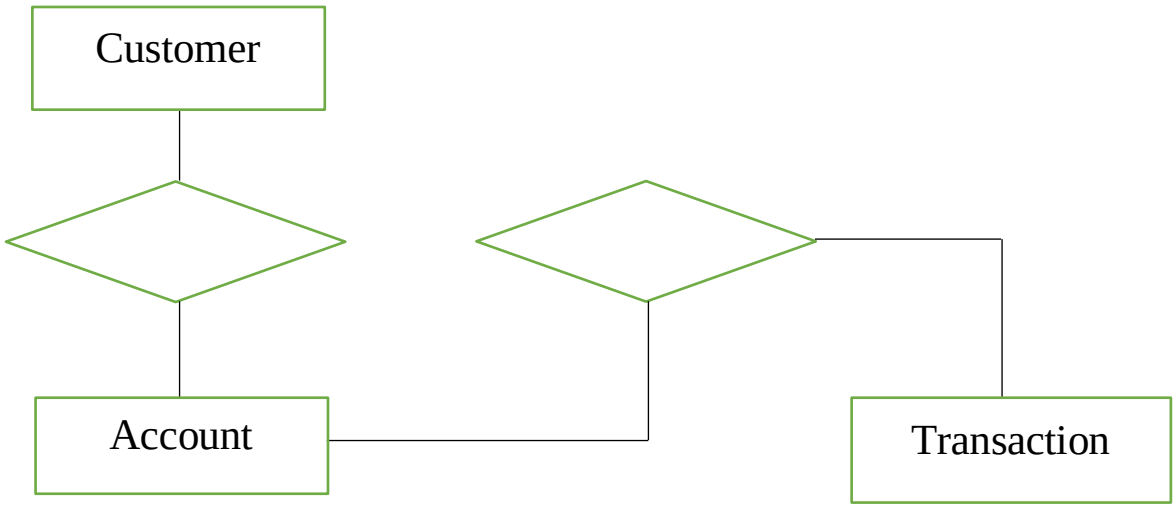
First connect the two entities, then drop the relationship notation on the line.

Cardinality

Cardinality specifies how many instances of an entity relate to one instance of another entity..

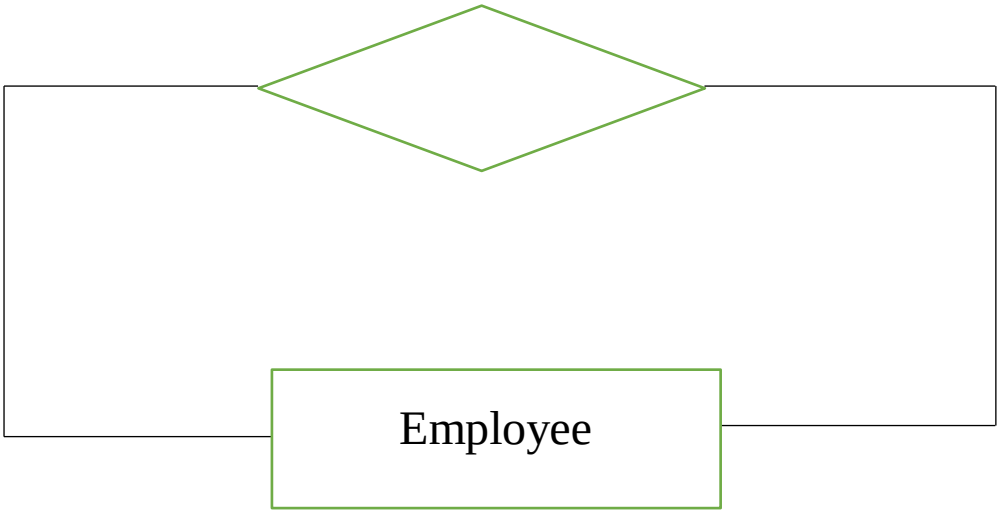
Ordinarily is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describe the relationship as either mandatory or optional. In other words, cardinality

specifies the maximum number of relationships and ordinarily specifies the absolute minimum number of relationships.

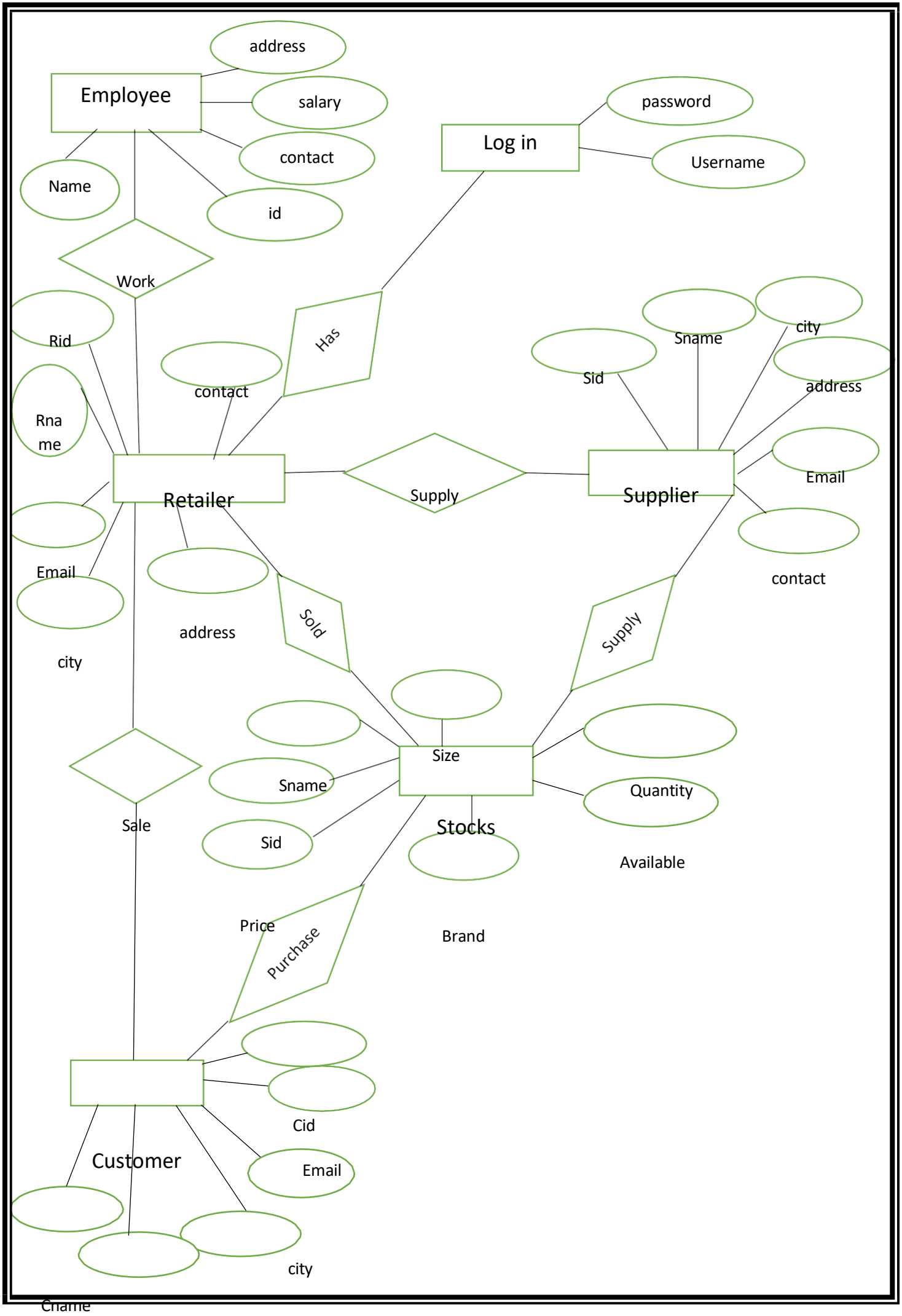


Recursive relationship

In some cases, entities can be self-linked. For example, supervise other employees.



ENTITY - RELATIONSHIP DIAGRAM (E-R DIAGRAM)


address
Employee
salary
contact
Name
id
Work
password
Log in
Username
Rid
Has
Sname
city
Sid
address
contact
Rna
me
Retailer
Supply
Supplier
Email
Email
contact
Sold
Supply
address
city
Size
Sname
Quantity
Sale
Stocks
Sid
Available
Price
Purchase
Brand
Cid
Customer
Email
city
Cname

contact

addre
ss

4
3

DATA FLOW DIAGRAM (DFD)

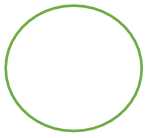
DFD is a graphical representation of data process and files used in a support system.

Data Flow Diagrams are useful tools for analyzing existing systems. Data Flow Diagram is a network that describes flows of data and the processes that changes or transforms the data throughout a system. Data flow diagrams can be expanded to show successive levels of details sufficient.

Expansion should be performed during the initial investigation to be certain that both the analyst and user personnel share a common understanding of the existing system and its data flow.

The different symbols used in the data flow diagram are:-

1. **Circle** — It is used from transfer the input to output

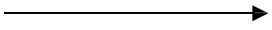


Indicate process that transfer data flow.

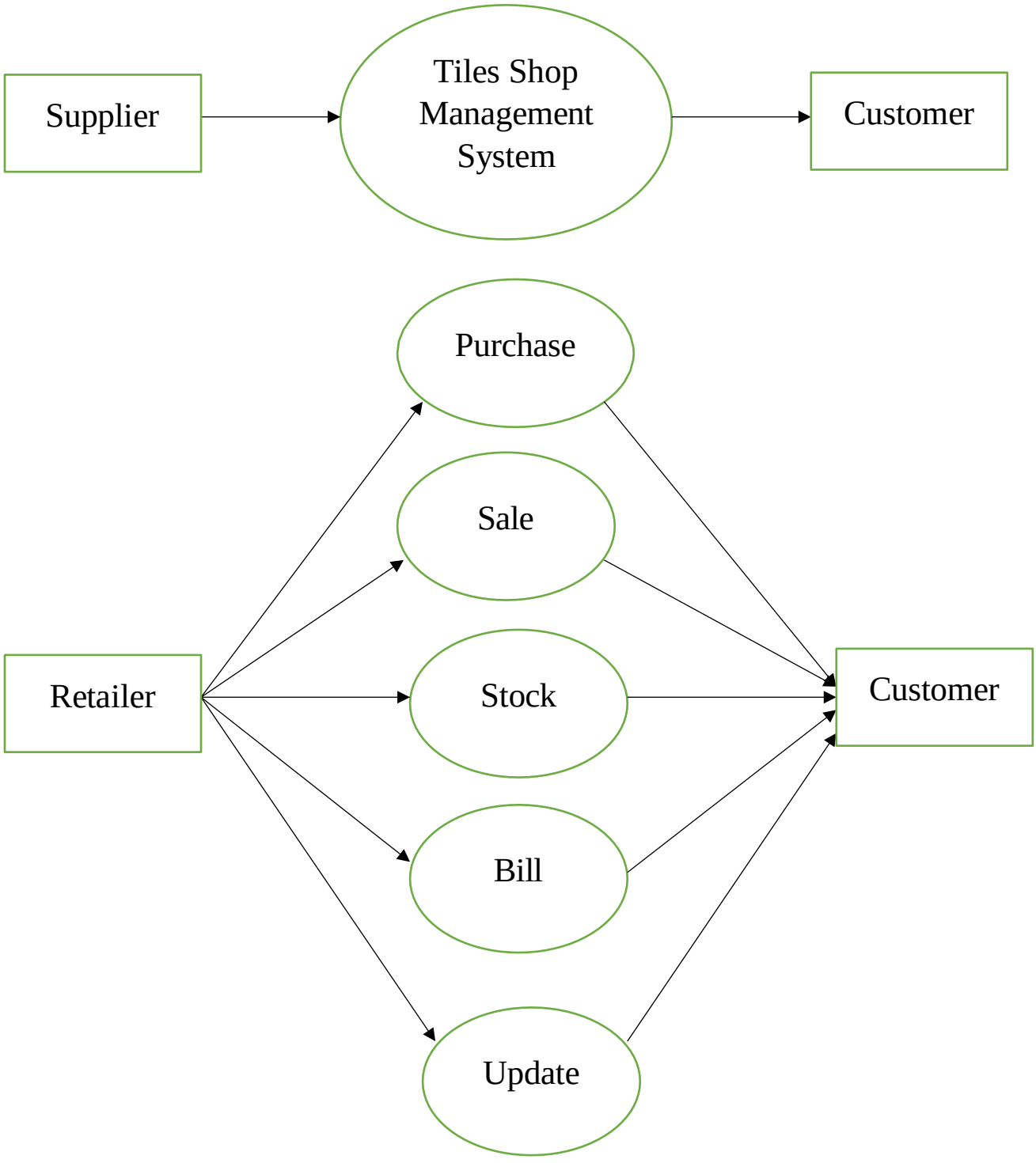
2. **Rectangle with Representation Open End** — when data is in file this symbol is used flow.

3. **Square** — Squares indicates the source of destination.

4. **Arrows** — Arrow are sued to show flow of path from where it is coming and going.

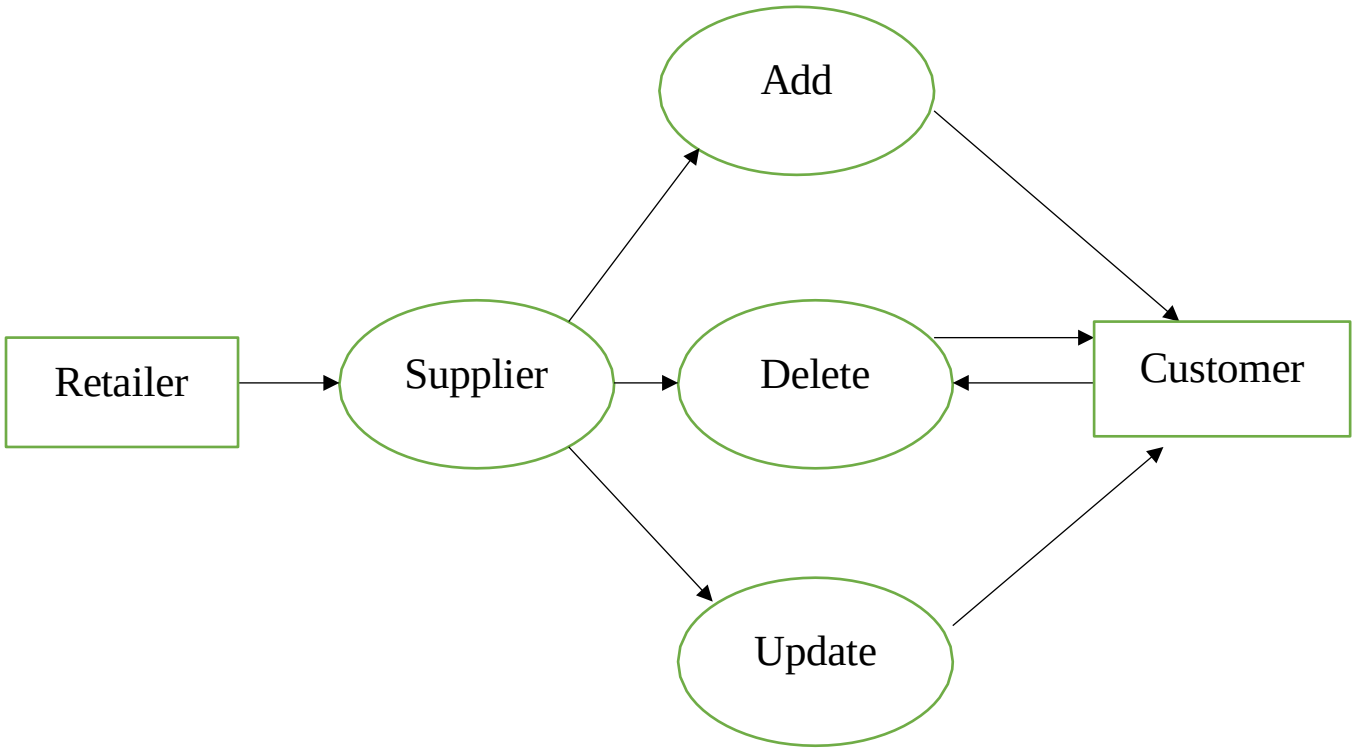


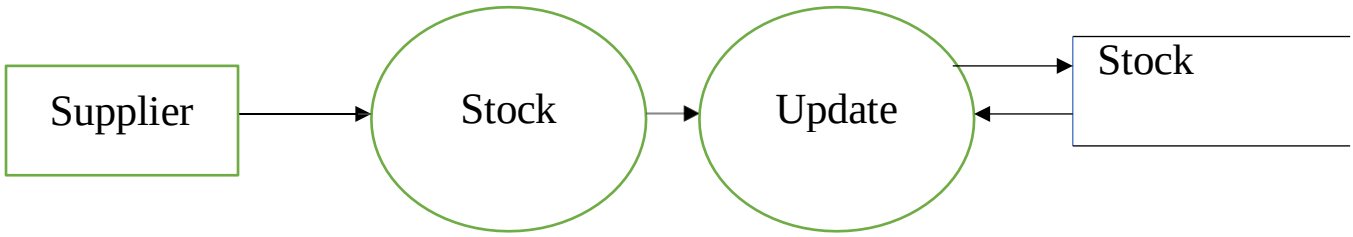
CONTEXT LEVEL DATA FLOW DIAGRAM FOR "TILES SHOP MANAGEMENT SYSTEM"



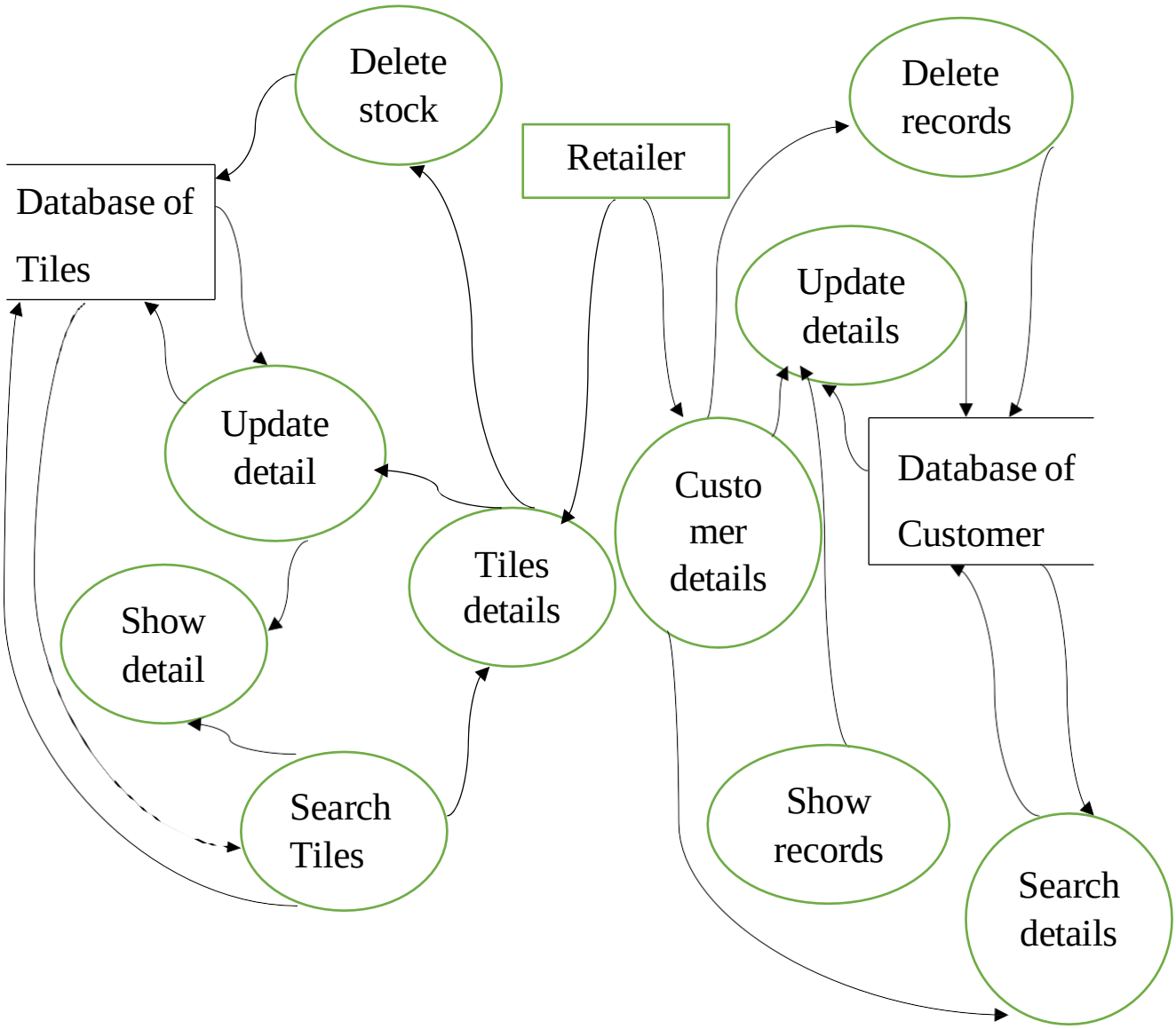
1ST LEVEL DATA FLOW DIAGRAM FOR "TILES SHOP MANAGEMENT SYSTEM"

Level 1.1





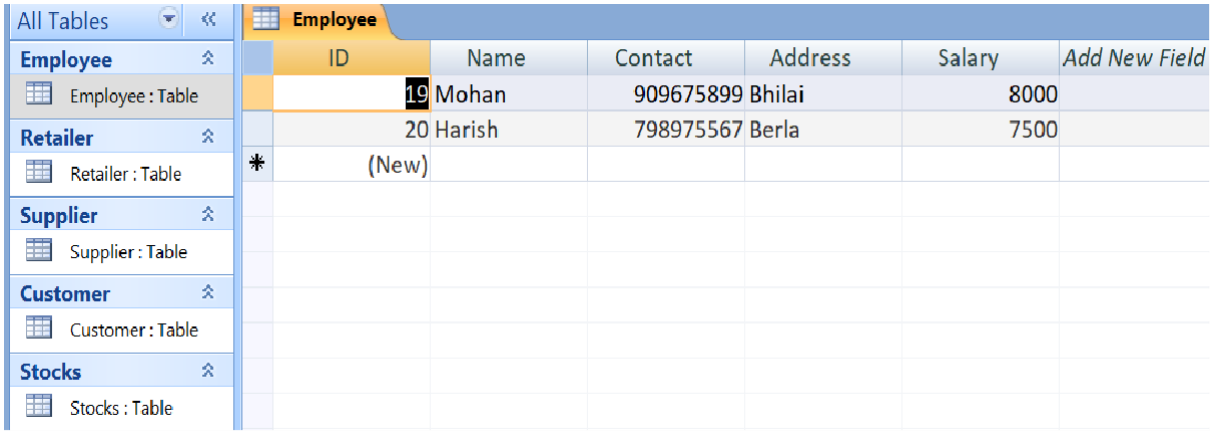
2ND LEVEL DATA FLOW DIAGRAM FOR "TILES SHOP MANAGEMENT SYSTEM"



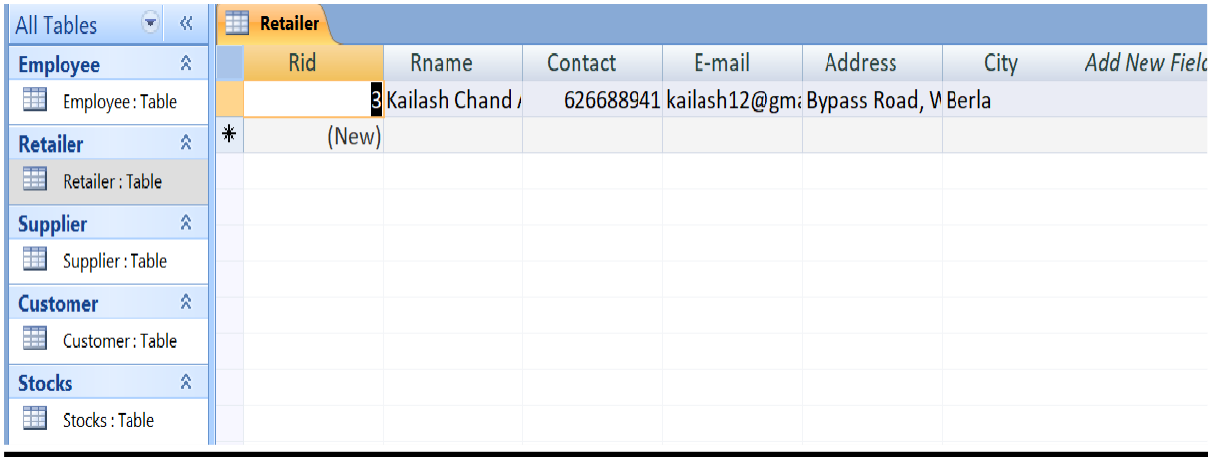
INPUT AND OUTPUT SCREEN DESIGN

DATABASE DESIGN :-

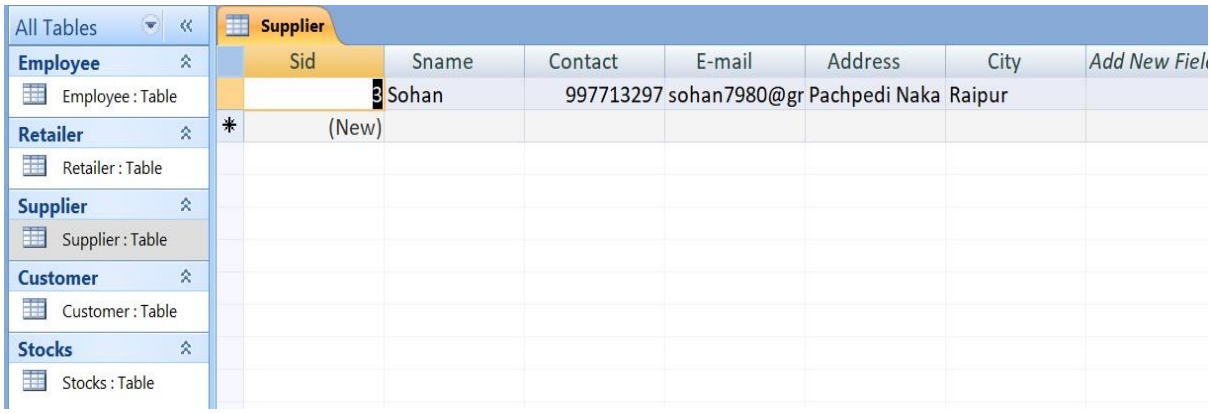
Employee Table



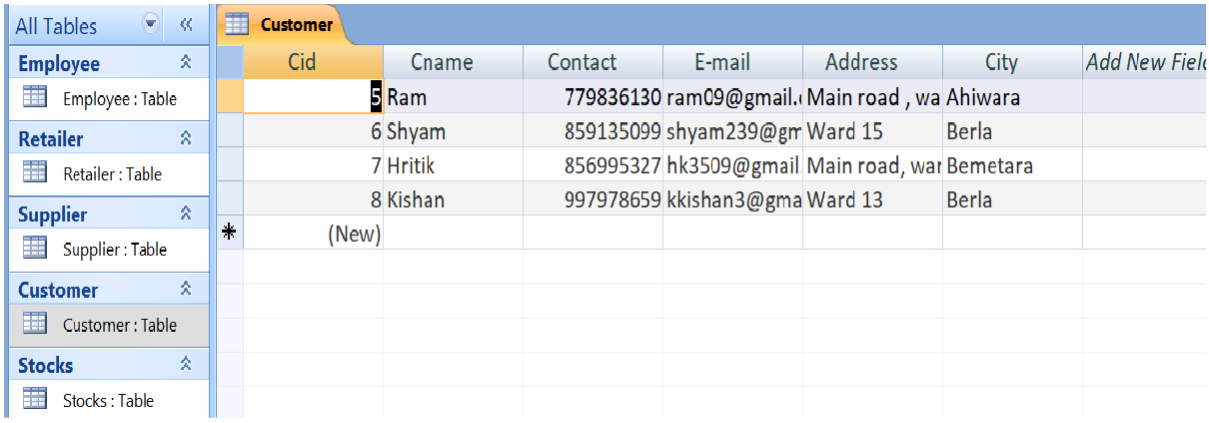
Retailer Table



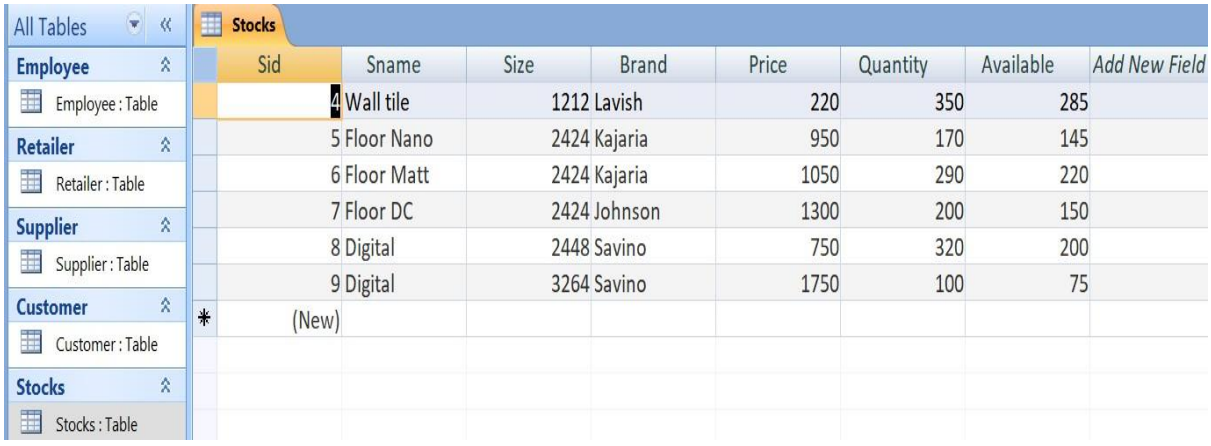
Supplier Table



Customer Table



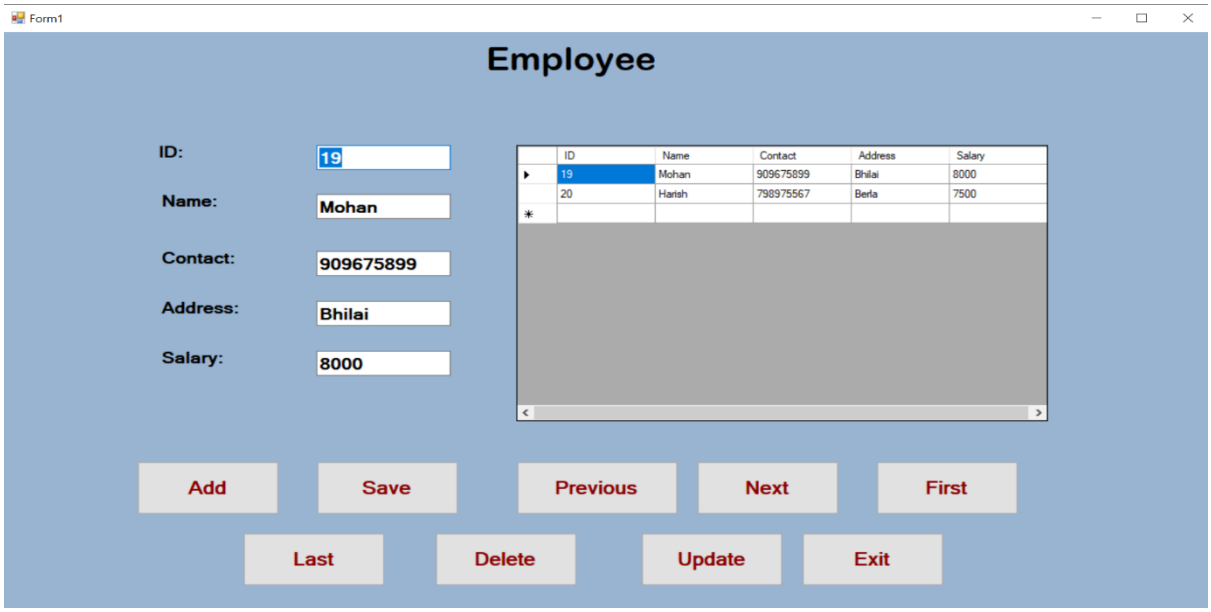
Stocks Table



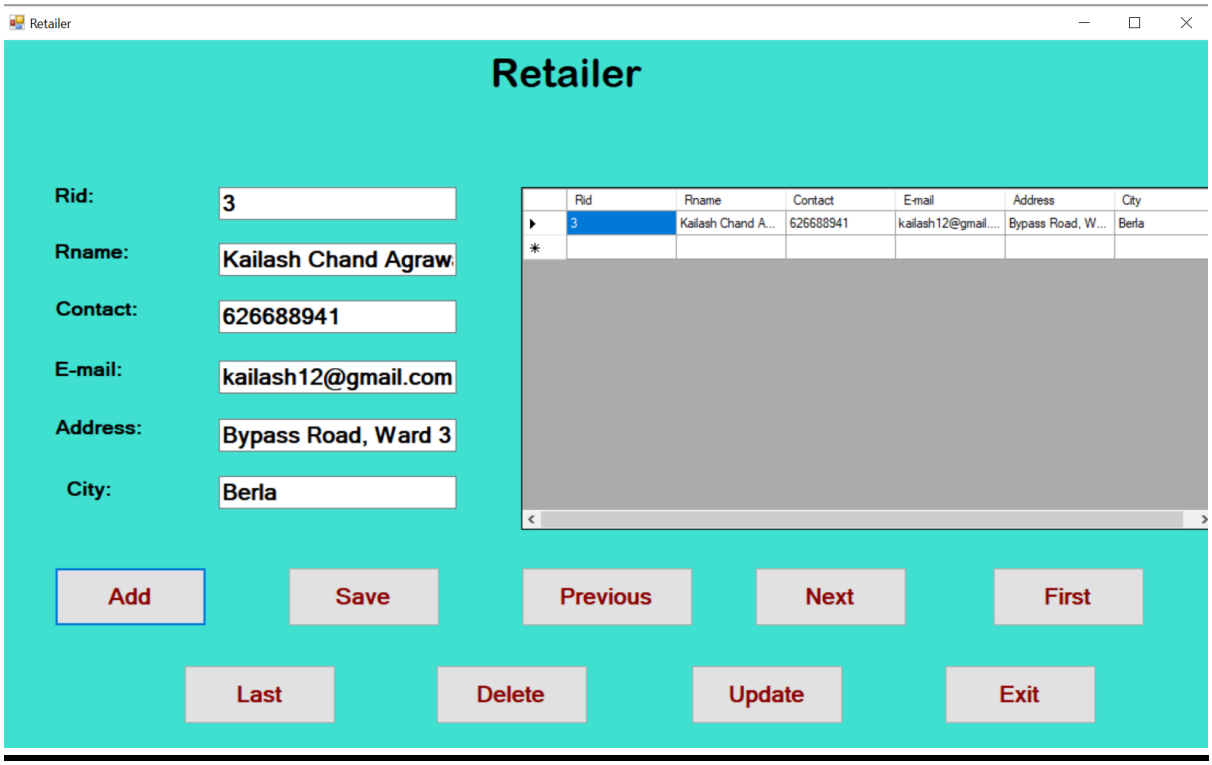
FORM DESIGN :-

Login Form

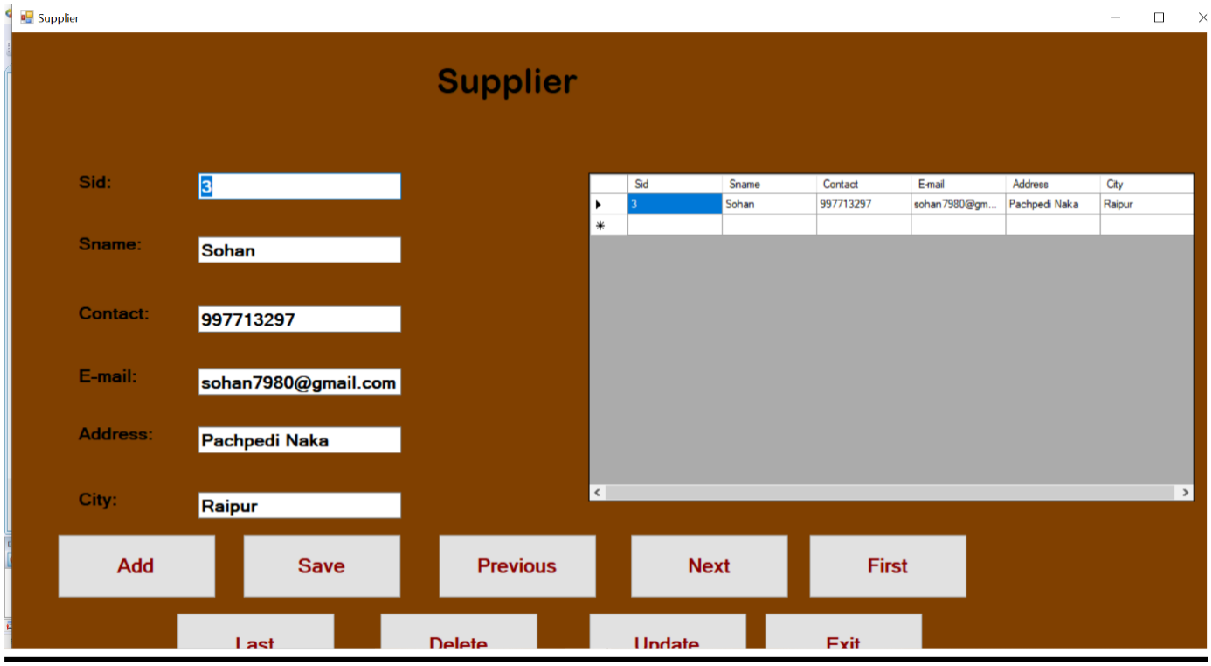
Employee Form



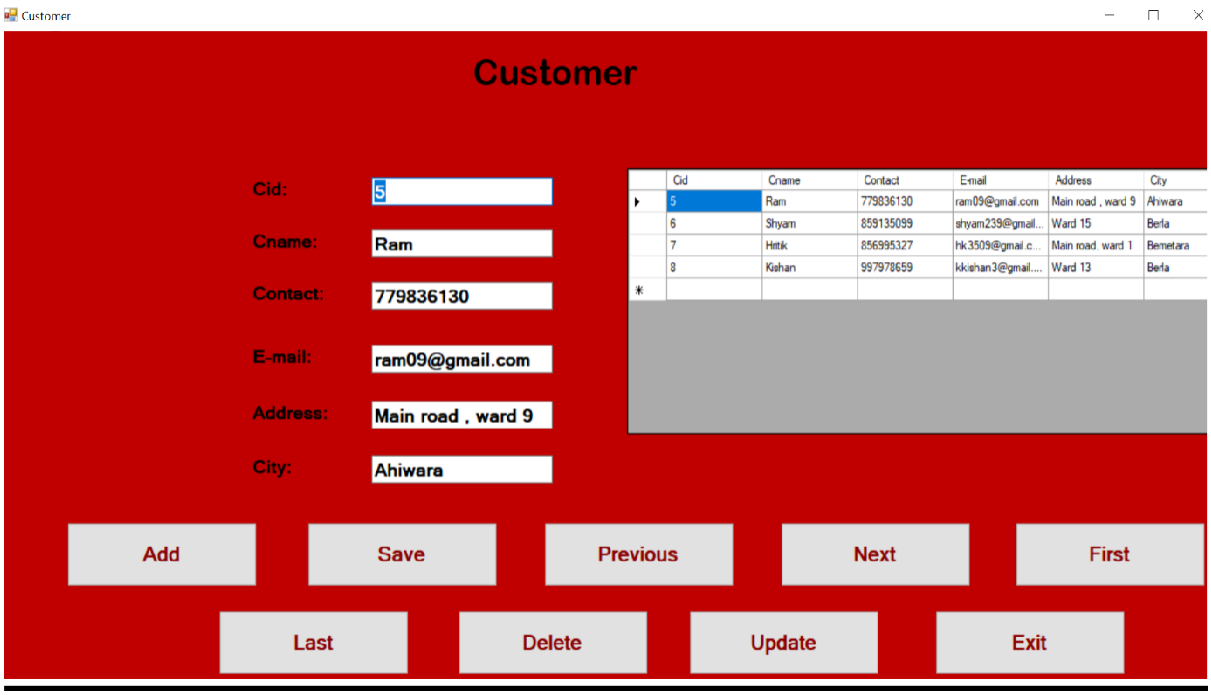
Retailer Form



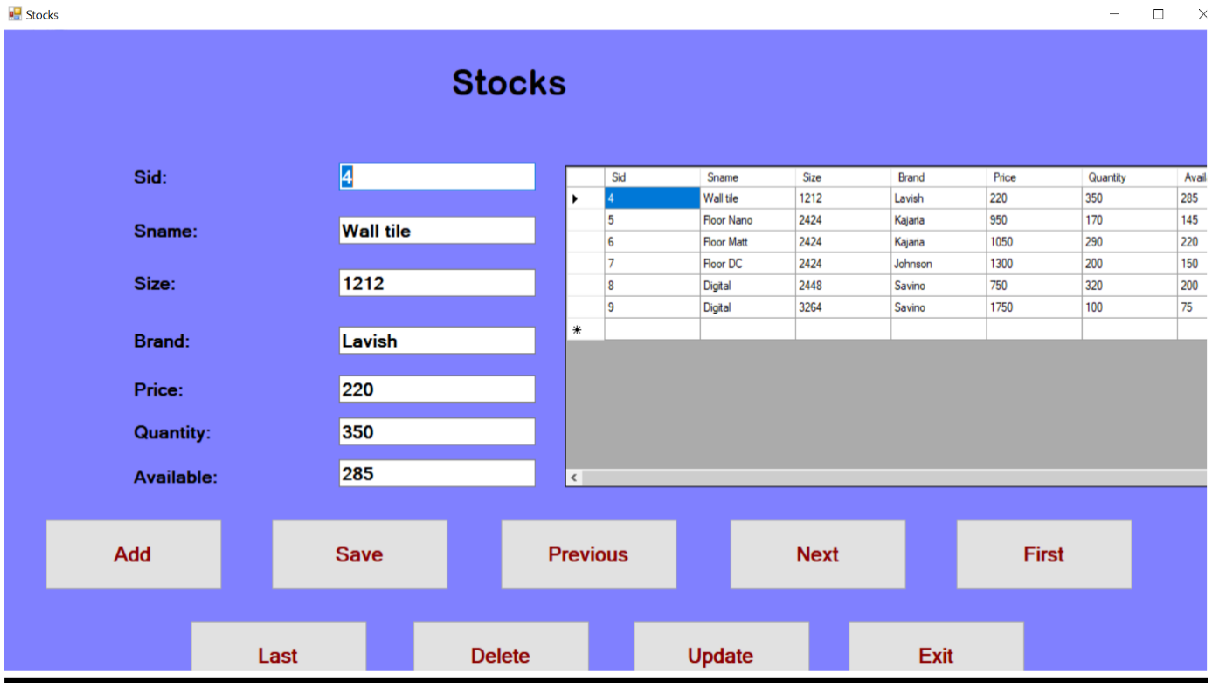
Supplier Form



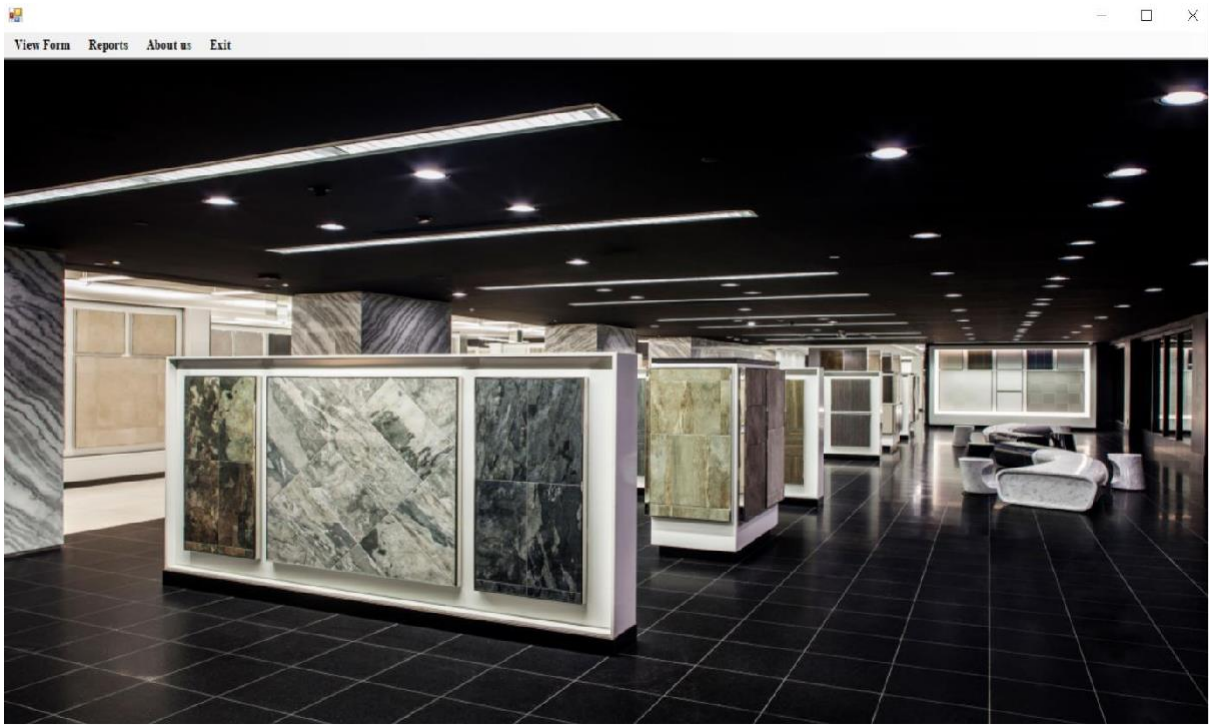
Customer Form



Stocks Form

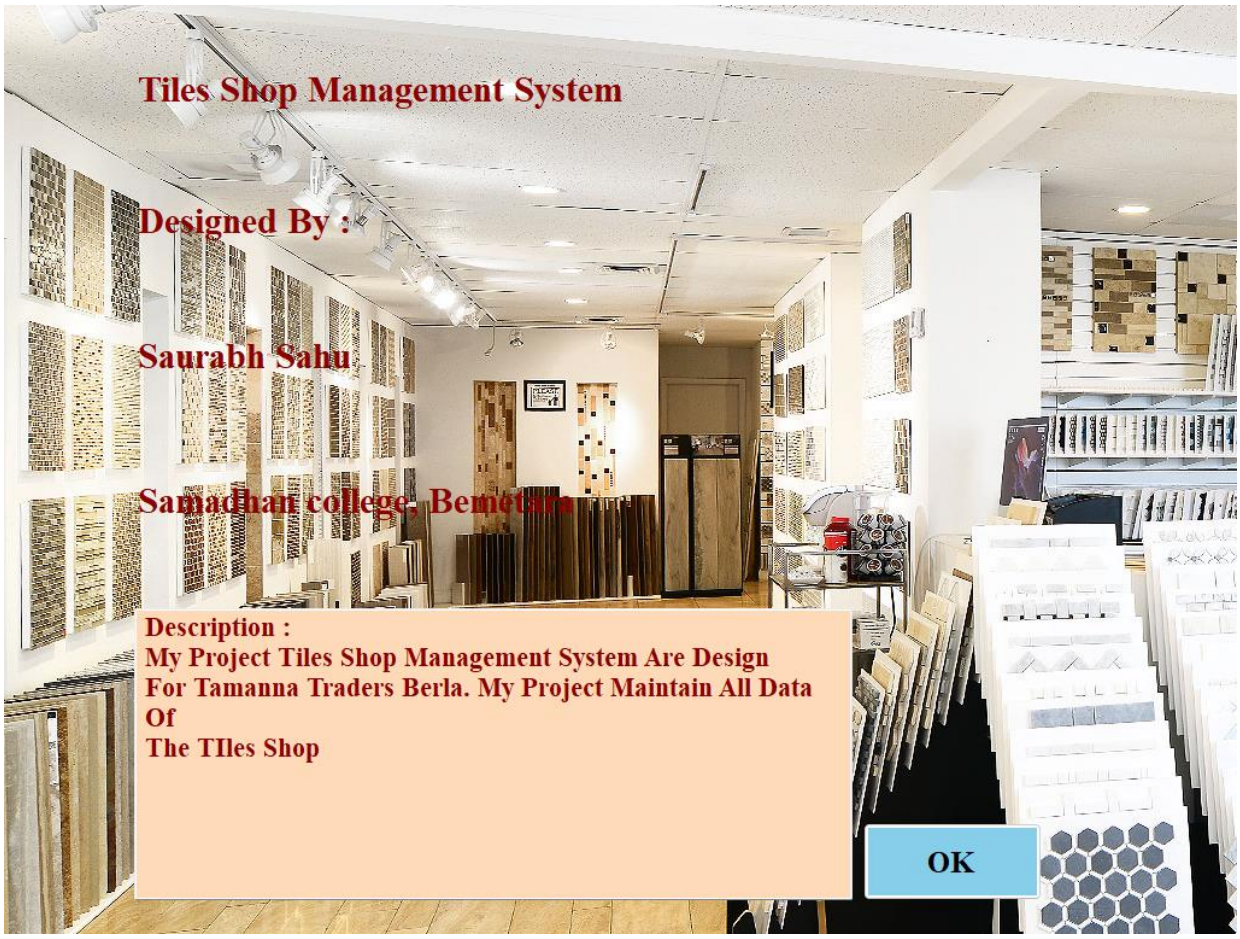


MDI Form



Splash Screen

About


Tiles Shop Management System
Designed By :
Saurabh Sahu
Samadhan college, Bemetara
Description :
My Project Tiles Shop Management System Are Design
For Tamanna Traders Berla. My Project Maintain All Data
Of
The TIles Shop
OK

CODING :-

Login Form Code

```
Public Class LoginForm1

Private Sub OK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles OK.Click
        If (UsernameTextBox.Text = "Piyush" And
PasswordTextBox.Text = "8922") Then
            MsgBox("Login Successfully")
            SplashScreen1.Show()
        Else
            MsgBox("Invailid Username Or Password")

        End If
    End Sub

    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Cancel.Click
        Me.Close()
    End Sub
```

Employee Form Code

```
Imports System.Data.OleDb
Imports System.Net.NetworkInformation

Public Class Employee
    Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\TILES SHOP MANAGEMENT
SYSTEM\TilesProject.accdb"
    Dim connection As OleDbConnection = New OleDbConnection(connectionString)
    Private datatabel As New DataTable
    Private dataAdapter As New OleDbDataAdapter
    Private Sub show_tabel()
        Try
            connection.Open()
            Dim query As String = "Select * from Employee"
            dataAdapter = New OleDbDataAdapter(query, connection)
            dataAdapter.Fill(datatabel)
            DataGridView1.DataSource = datatabel
        Catch ex As Exception
            MessageBox.Show("Error", ex.Message)
```

```
        If connection.State = ConnectionState.Closed Then
            connection.Open()
        End If
    Finally
        connection.Close()
    End Try
End Sub

Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click
    'Me.DataGridView1.Rows.Add(txtid.Text, txtname.Text, contact.Text, txtaddress.Text, txtsalary.Text)

    Dim firstName As String = txtname.Text
    Dim contact As String = txtcontact.Text
    Dim address As String = txtaddress.Text
    Dim salary As String = txtsalary.Text

    ' Save customer information to the database
    Try
        Using connection As New OleDbConnection(connectionString)
            connection.Open()

            Dim query As String = "INSERT INTO Employee (Name, Contact, Address, Salary) VALUES (?, ?, ?, ?)"

            Using cmd As New OleDbCommand(query, connection)

                cmd.Parameters.AddWithValue("@Name", firstName)
                cmd.Parameters.AddWithValue("@Contact", contact)
                cmd.Parameters.AddWithValue("@Address", address)
                cmd.Parameters.AddWithValue("@Salary", salary)

                cmd.ExecuteNonQuery()
            End Using
        End Using

        MessageBox.Show(" saved successfully.")
        show_tabel()
    Catch ex As Exception
        MessageBox.Show("Error: " & ex.Message)
        connection.Close()
    End Try
End Sub

Private Sub Employee_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    show_tabel()
End Sub
End Class
```

Retailer Form Code

```
Imports System.Data.OleDb
Imports System.Net.NetworkInformation

Public Class Retailer
    Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\TILES SHOP MANAGEMENT
SYSTEM\TilesProject.accdb"
    Dim connection As OleDbConnection = New OleDbConnection(connectionString)
    Private datatabel As New DataTable
    Private dataAdapter As New OleDbDataAdapter
    Private Sub show_tabel()
```

```
        Try
            connection.Open()
            Dim query As String = "Select * from Retailer"
            dataAdapter = New OleDbDataAdapter(query, connection)
            dataAdapter.Fill(datatabel)
            DataGridView1.DataSource = datatabel
        Catch ex As Exception
            MessageBox.Show("Error", ex.Message)
            If connection.State = ConnectionState.Closed Then
                connection.Open()
            End If
        Finally
            connection.Close()
        End Try
    End Sub

    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click
        'Me.DataGridView1.Rows.Add(txtid.Text, txtname.Text, contact.Text, txtemail.Text, txtaddress.Text, txtcity.Text)

        Dim firstName As String = txtname.Text
        Dim contact As String = txtcontact.Text
        Dim email As String = txtemail.Text
        Dim address As String = txtaddress.Text
        Dim city As String = txtcity.Text

        ' Save customer information to the database
        Try
            Using connection As New OleDbConnection(connectionString)
                connection.Open()

                Dim query As String = "INSERT INTO Retailer (Rname, Contact, E_mail, Address, City) VALUES (?, ?, ?, ?, ?)"

                Using cmd As New OleDbCommand(query, connection)

                    cmd.Parameters.AddWithValue("@Rname", firstName)
                    cmd.Parameters.AddWithValue("@Contact", contact)
                    cmd.Parameters.AddWithValue("@E_mail", email)
                    cmd.Parameters.AddWithValue("@Address", address)
                    cmd.Parameters.AddWithValue("@City", city)

                    cmd.ExecuteNonQuery()
                End Using
            End Using

            MessageBox.Show(" saved successfully.")
            show_tabel()
        Catch ex As Exception
            MessageBox.Show("Error: " & ex.Message)
            connection.Close()
        End Try
    End Sub

    Private Sub Retailer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        show_tabel()
    End Sub
End Class
```

Supplier Form Code

```
Imports System.Data.OleDb
```

```
Public Class Supplier

    Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\TILES SHOP MANAGEMENT
SYSTEM\TilesProject.accdb;"
    Dim connection As OleDbConnection = New OleDbConnection(connectionString)
    Private datatable As New DataTable
    Private dataAdapter As New OleDbDataAdapter

    Sub Show_Table()
        connection.Open()
        Dim query As String = "Select * from Supplier"
        dataAdapter = New OleDbDataAdapter(query, connection)
        dataAdapter.Fill(datatable)
        SupplierDataGridView.DataSource = datatable
        connection.Close()

    End Sub

    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click
        'Me.SupplierDataGridView.Rows.Add(txtid.Text, txtname.Text, contact.Text, Emailtxt.Text, AddressText.Text, CityText.Text)
        Try
            connection.Open()
            Dim cmd As OleDbCommand = New OleDbCommand("INSERT INTO Supplier(Sname, Contact, E_mail, Address, City)
VALUES (?, ?, ?, ?,?)", connection)


            cmd.Parameters.AddWithValue("@Sname", txtname.Text)
            cmd.Parameters.AddWithValue("@Contact", contact.Text)
            cmd.Parameters.AddWithValue("@E_mail", Emailtxt.Text)
            cmd.Parameters.AddWithValue("@Address", AddressText.Text)
            cmd.Parameters.AddWithValue("@City", CityText.Text)

            cmd.ExecuteNonQuery()

            MessageBox.Show("saved successfully!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information)

            Show_Table()
            If connection.State = ConnectionState.Open Then
                connection.Close()
            End If
        Catch ex As Exception
            MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            If connection.State = ConnectionState.Open Then
                connection.Close()
            End If
        Finally
            If connection.State = ConnectionState.Open Then
                connection.Close()
            End If
        End Try
        If connection.State = ConnectionState.Open Then
            connection.Close()
        End If
    End Sub

    Private Sub Supplier_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Show_Table()
    End Sub
End Class
```

Customer Form Code

```
Imports System.Data.OleDb
Imports System.Net.NetworkInformation
Public Class Customer

    Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\TILES SHOP MANAGEMENT
SYSTEM\TilesProject.accdb;"
    Dim connection As OleDbConnection = New OleDbConnection(connectionString)
    Private datatable As New DataTable
    Private dataAdapter As New OleDbDataAdapter

    Sub Show_Table()
        connection.Open()
        Dim query As String = "Select * from Customer"
        dataAdapter = New OleDbDataAdapter(query, connection)
        dataAdapter.Fill(datatable)
        CustomerDataGridView.DataSource = datatable
        connection.Close()

    End Sub

    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btnsave.Click
        'Me.SupplierDataGridView.Rows.Add(txtid.Text, txtname.Text, contact.Text, Emailtxt.Text, AddressText.Text, CityText.Text)
        Try
            connection.Open()
            Dim cmd As OleDbCommand = New OleDbCommand("INSERT INTO Customer(Cname, Contact, E_mail, Address, City)
VALUES (?, ?, ?, ?,?)", connection)


            cmd.Parameters.AddWithValue("@Cname", CnameTextBox.Text)
            cmd.Parameters.AddWithValue("@Contact", ContactTextBox.Text)
            cmd.Parameters.AddWithValue("@E_mail", E_mailTextBox.Text)
            cmd.Parameters.AddWithValue("@Address", AddressTextBox.Text)
            cmd.Parameters.AddWithValue("@City", CityTextBox.Text)

            cmd.ExecuteNonQuery()

            MessageBox.Show("saved successfully!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information)

            Show_Table()
            If connection.State = ConnectionState.Open Then
                connection.Close()
            End If
        Catch ex As Exception
            MessageBox.Show("Error: " & ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            If connection.State = ConnectionState.Open Then
                connection.Close()
            End If
        Finally
            If connection.State = ConnectionState.Open Then
                connection.Close()
            End If
        End Try
        If connection.State = ConnectionState.Open Then
            connection.Close()
        End If
    End Sub

    Private Sub Customer_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Show_Table()
    End Sub
End Class
```

Stocks Form Code

```
Imports System.Data.OleDb
Imports System.Net.NetworkInformation

Public Class Stocks
    Dim connectionString As String = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=E:\TILES SHOP MANAGEMENT
SYSTEM\TilesProject.accdb"
    Dim connection As OleDbConnection = New OleDbConnection(connectionString)
    Private datatabel As New DataTable
    Private dataAdapter As New OleDbDataAdapter
    Private Sub show_tabel()
        Try
            connection.Open()
            Dim query As String = "Select * from Stocks"
            dataAdapter = New OleDbDataAdapter(query, connection)
            dataAdapter.Fill(datatabel)
            DataGridView1.DataSource = datatabel
        Catch ex As Exception
            MessageBox.Show("Error", ex.Message)
            If connection.State = ConnectionState.Closed Then
                connection.Open()
            End If
        Finally
            connection.Close()
        End Try
    End Sub

    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click




'Me.DataGridView1.Rows.Add(txtid.Text, txtname.Text, contact.Text, txtaddress.Text, txtsalary.Text)

        Dim firstName As String = nametxt.Text
        Dim size As String = sizetxt.Text
        Dim brand As String = brandtxt.Text
        Dim price As String = pricetxt.Text
        Dim quantity As String = quantxt.Text
        Dim avaiable As String = availtxt.Text

        ' Save customer information to the database
        Try
            Using connection As New OleDbConnection(connectionString)
                connection.Open()

                Dim query As String = "INSERT INTO Stocks (Sname, Size, Brand, Price, Quantity, Available) VALUES (?, ?, ?, ?, ?, ?)"

                Using cmd As New OleDbCommand(query, connection)

                    cmd.Parameters.AddWithValue("@Sname", firstName)
                    cmd.Parameters.AddWithValue("@Size", size)
                    cmd.Parameters.AddWithValue("@Brand", brand)
```

```
            cmd.Parameters.AddWithValue("@Price", price)
            cmd.Parameters.AddWithValue("@Quantity", quantity)
            cmd.Parameters.AddWithValue("@Available", avaiable)
            cmd.ExecuteNonQuery()
        End Using
    End Using
    MessageBox.Show(" saved successfully.")
    show_tabel()
  Catch ex As Exception
    MessageBox.Show("Error: " & ex.Message)
    connection.Close()
  End Try
End Sub

Private Sub Stocks_Load(sender As Object, e As EventArgs) Handles MyBase.Load
  show_tabel()
End Sub
End Class
```

MDI Form Code

```
Imports System.Windows.Forms

Public Class MDI

    Private Sub ShowNewForm(ByVal sender As Object, ByVal e As
EventArgs)
        ' Create a new instance of the child form.
```

```
        Dim ChildForm As New System.Windows.Forms.Form
        ' Make it a child of this MDI form before showing it.
        ChildForm.MdiParent = Me

        m_ChildFormNumber += 1
        ChildForm.Text = "Window " & m_ChildFormNumber

        ChildForm.Show()
    End Sub

    Private Sub OpenFile(ByVal sender As Object, ByVal e As
EventArgs)
        Dim OpenFileDialog As New OpenFileDialog
        OpenFileDialog.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments
        OpenFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files
(*.*)|*.*"
        If (OpenFileDialog.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then
            Dim FileName As String = OpenFileDialog.FileName
            ' TODO: Add code here to open the file.
        End If
    End Sub

    Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        Dim SaveFileDialog As New SaveFileDialog
        SaveFileDialog.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments
        SaveFileDialog.Filter = "Text Files (*.txt)|*.txt|All Files
(*.*)|*.*"

        If (SaveFileDialog.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then
            Dim FileName As String = SaveFileDialog.FileName
            ' TODO: Add code here to save the current contents of the
form to a file.
```

```
        End If
    End Sub


    Private Sub ExitToolsStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        Me.Close()
    End Sub

    Private Sub CutToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        ' Use My.Computer.Clipboard to insert the selected text or
images into the clipboard
    End Sub

    Private Sub CopyToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        ' Use My.Computer.Clipboard to insert the selected text or
images into the clipboard
    End Sub

    Private Sub PasteToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        'Use My.Computer.Clipboard.GetText() or
My.Computer.Clipboard.GetData to retrieve information from the
clipboard.
    End Sub



    Private Sub CascadeToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        Me.LayoutMdi(MdiLayout.Cascade)
    End Sub

    Private Sub TileVerticalToolStripMenuItem_Click(ByVal sender
As Object, ByVal e As EventArgs)
```

```
        Me.LayoutMdi(MdiLayout.TileVertical)
    End Sub

    Private Sub TileHorizontalToolStripMenuItem_Click(ByVal
sender As Object, ByVal e As EventArgs)
        Me.LayoutMdi(MdiLayout.TileHorizontal)
    End Sub

    Private Sub ArrangeIconsToolStripMenuItem_Click(ByVal sender
As Object, ByVal e As EventArgs)
        Me.LayoutMdi(MdiLayout.ArrangeIcons)
    End Sub

    Private Sub CloseAllToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        ' Close all child forms of the parent.
        For Each ChildForm As Form In Me.MdiChildren
            ChildForm.Close()
        Next
    End Sub

    Private m_ChildFormNumber As Integer

    Private Sub EmployeeToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
EmployeeToolStripMenuItem.Click
        Employee.Show()

    End Sub

    Private Sub RetailerToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RetailerToolStripMenuItem.Click
        Retailer.Show()

    End Sub
```

```
    Private Sub SupplierToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SupplierToolStripMenuItem.Click
        Supplier.Show()

    End Sub

    Private Sub CustomerToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CustomerToolStripMenuItem.Click
        Customer.Show()

    End Sub

    Private Sub StocksToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
StocksToolStripMenuItem.Click
        Stocks.Show()

    End Sub

    Private Sub MDI_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub AboutUsToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AboutUsToolStripMenuItem.Click
        About.Show()



    End Sub
```

```
    Private Sub ExitToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ExitToolStripMenuItem.Click
        Me.Close()

    End Sub

    Private Sub ReportsToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ReportsToolStripMenuItem.Click


     End Sub

    Private Sub MenuStrip1_ItemClicked(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.ToolStripItemClickedEventArgs) Handles
MenuStrip1.ItemClicked

    End Sub

    Private Sub EmployeeToolStripMenuItem1_Click(ByVal sender
As System.Object, ByVal e As System.EventArgs) Handles
EmployeeToolStripMenuItem1.Click
        EmployeeReport.Show()

    End Sub

    Private Sub RetailerToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RetailerToolStripMenuItem1.Click
        RetailerReport.Show()

    End Sub
```

```
    Private Sub SupplierToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
SupplierToolStripMenuItem1.Click
        SupplierReport.Show()


    End Sub

    Private Sub CustomerToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
CustomerToolStripMenuItem1.Click
        CustomerReport.Show()

    End Sub

    Private Sub StocksToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
StocksToolStripMenuItem1.Click
        StocksReport.Show()

    End Sub
```

Splash Screen Code

```
  Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Timer1.Tick
        ProgressBar1.Value += 1
        If ProgressBar1.Value <= 10 Then
            Label2.Text = "Initializing system... "
        ElseIf ProgressBar1.Value <= 30 Then
            Label2.Text = "Loading all components... "
        ElseIf ProgressBar1.Value <= 50 Then
            Label2.Text = "Integrating Database... "
        ElseIf ProgressBar1.Value <= 70 Then
            Label2.Text = "Please Wait... "
        ElseIf ProgressBar1.Value <= 100 Then
```

```
        Label2.Text = "Welocome To My Tiles Shop... "
        If ProgressBar1.Value = 100 Then
            Timer1.Dispose()
            Me.Hide()
            MDI.Show()
            Me.Close()

        End If


    End If
End Sub
```

PROCESS INVOLVED

Tiles Shop management is an internal business function responsible for managing documents from suppliers. Suppliers represent companies who provide business with economic resources. Economic resources include the direct labor needed to produce consumer service. Companies use this system to ensure suppliers and employees receive payment in a timely manner.

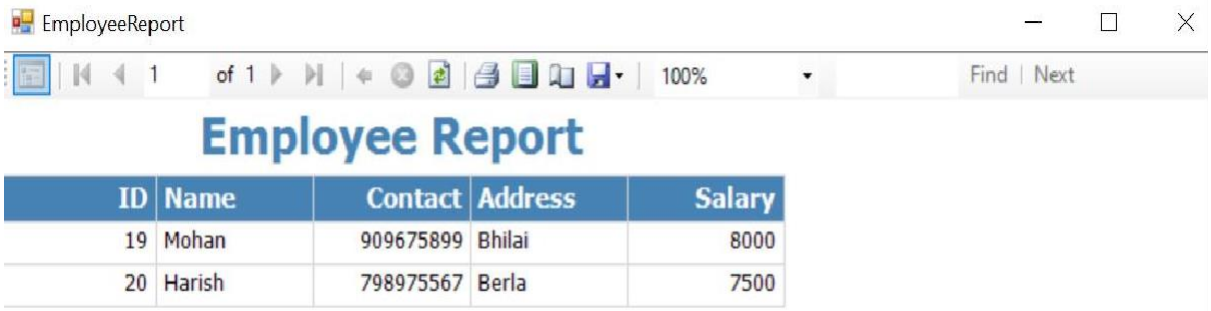
In this system we manage purchase records which represent the tiles shop internal information that provides individuals specific authorization to purchase tiles from suppliers. Mostly two copies of purchase orders usually exist in a business. One is sent to the suppliers and one is retained in the tiles shop's accounting department.

In the tiles shop manager may use electronic copies of purchase orders. Rather than maintaining a paper copy in a file, manager may scan this information into a server database. This allows individuals to review or retrieve the purchase order as needed. Business software can also increase the accuracy of purchase order information.

We develop this software for easy accessing the data of any customer, employee records, purchase record, sales records, stocks etc. And the business owners and managers will develop a system that best supports their back office and accounting functions.

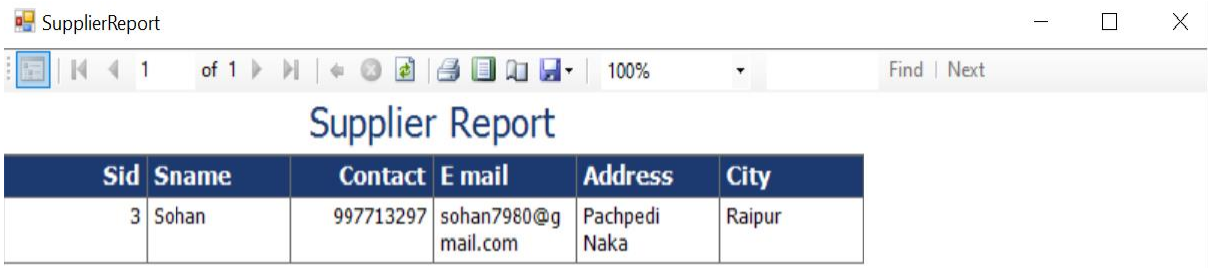
TEST REPORT, PRINTOUT OF THE CODE SHEET

Employee Report



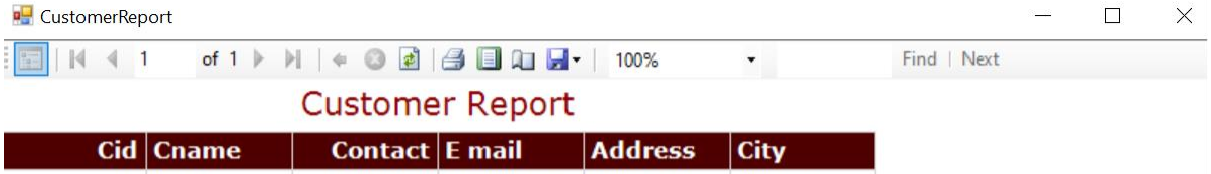
Retailer Report

Supplier Report



Sid	Sname	Contact	E mail	Address	City
3	Sohan	997713297	sohan7980@gmail.com	Pachpedi Naka	Raipur

Customer Report



Cid	Cname	Contact	E mail	Address	City
5	Ram	779836130	ram09@gmail.com	Main road , ward 9	Ahiwara
6	Shyam	859135099	shyam239@gmail.com	Ward 15	Berla
7	Hritik	856995327	hk3509@gmail.com	Main road, ward 1	Bemetara
8	Kishan	997978659	kkishan3@gmail.com	Ward 13	Berla

Stocks Report

Sid	Sname	Size	Brand	Price	Quantity	Available
4	Wall tile	1212	Lavish	220	350	285
5	Floor Nano	2424	Kajaria	950	170	145
6	Floor Matt	2424	Kajaria	1050	290	220
7	Floor DC	2424	Johnson	1300	200	150
8	Digital	2448	Savino	750	320	200
9	Digital	3264	Savino	1750	100	75

CONCLUSION

Now a day's computerization and networking of the existing manual system is going on large scale because of the versatility, speed accuracy and diligence if offers to its users. computer provide a particle means to organize things systematically and economically considering volume of information to be handled and the degree of accuracy in organization, the use of reports can prove to be a blessing. The project in discussion in an attempt all the above said objectives its development was meant to replace the manual system and to achieve, to goal, to maximum accuracy and most efficiently. But like every other system the system might process faults and be called a stepping stone in the run to processes in "TILES SHOP MANAGEMENT".

The Project "TILES SHOP MANAGEMENT" is shows the following attributes.

1. User friendly.
2. Generate the desired purchase and sales report.
3. Proper handling of report of sale n purchase transaction.
4. Daily transaction is very convenient and up to date.
5. Complete update of stock.

There is still the scope of further implementation in this programmed.

REFERENCES

- MSDN online- By Microsoft.
- Visual Basic.NET Complete- BPB Publications, New Delhi.
- The Complete Reference VB.NET –Jeffery R. Shapiro, Tata McGraw Hill.
- Visual Basic.NET Programming Black Book-Steven Holzner by Dreamtech Press.

SOFT COPY OF THE PROJECT ON CD