A PROJECT REPORT ON

AUTOMOBILE SHOPE MANAGEMENT

System BY

Singh Rishikesh Shyamsunder

SEAT NO: - 4018641

UNDER GUIDANCE OF

PROF.HATIM KANORWALA

SUBMITTED TO THE UNIVERSITY OF
MUMBAI FOR THE PARTIAL

FULFILLMENT OF THE REQUIREMENT

FOR QUALIFYING

BACHELOR OF SCIENCE

SEMISTER-VI EXAMINATION

ACADEMIC YEAR 2022-23

DEPARTMENT OF INFORMATION TECHNOLOGY

M.P.S.P. SINGH COLLEGE OF

ARTS, COMMERCE & SCIENCE

BANDRA (EAST), MUMBAI- 400051

Uttar Bharatiya Sangh's

M.P.S.P. Singh College of Arts, Commerce and Science
(Affiliated to University of Mumbai)

DATE: 10TH APRIL 2023

PROJECT CERTIFICATE

This is to certify that the project entitled **Automobile Management System.**
Undertaken at the **M.P.S.P.Singh College of Arts, Commerce & Science** by **Mr.Singh Rishikesh Shyamsunder** Seat no.: **4018641** in partial fulfillments of Bsc.IT degree (Semester VI) Examination had not been submitted for any other examination and does not form part of any other course undergone by the candidate. It is further certified that he has completed all required phases of the project.

(PROJECT GUIDE) (CO-ORDINATOR) (PRINCIPAL)

(INTERNAL EXAMINER) (EXTERNAL EXAMINER)

(COLLEGE SEAL)

ABSTRACT

This project aims at the Introduction to computerized Sales & Service Management.

This software is also developed for daily basis business processes and effectively managing Sales, Purchase and Stock.

This software is designed keeping in mind the user's efficiency & ease of handling and maintenance, as and secured system over centralized data handling and providing with the features to get the complete study and control over the business.

The report depicts the basics logic used for software development long with the Activity diagrams so that logics may be apprehended without difficulty.

For detailed information, screen layouts, provided along with this report can be viewed.

Although this report is prepared with considering the results required these may be across since the project is subjected to future enhancements as per the need of organizations.

ACKNOWLEDGEMENT

In completing this graduate project, we have been fortunate to have help, support and encouragement from many people. We would like to acknowledge them for their cooperation. We would like to thank all the professor of the Bsc.IT department for helping and guiding us through the semester. We also owe a personal debt of gratitude to our parents for being cooperative during the development of this project. Lastly, we would thank God showering his blessings on us.

DECLARATION

We hereby declare that this project entitled **"AUTOMOBILE MANAGEMENT SYSTEM"** is being done by us and not been duplicate or submitted to any other educational institution for the award of any degree, by us. To the best of our knowledge, other than us, no one else has submitted this project anywhere else for any purpose. This project dissertation is submitted as a part of the partial fulfillment of the current curriculum. The complete execution of the same shall be submitted in the next semester.

INDEX

Preliminary investigation

Introduction :

In today's world Computer System is one of the integral part of out life. Computers have made the life and lifestyle of corporate world to common man ease to a great extend.

The use of software in corporate world has made the functioning of the organization smooth and efficient. Due to this the industry is able to keep the track of its progress and maintain the records. These records can be easily accessed and thus help the organization to know it's functioning & track the success & failures of the processes. This helps the organization to grow.

Working :

Our client who owns a automobile showroom , currently has an computerised system which is used to create , maintain, retrieve,update and delete data from the system.

This system used by the client is in form of Excel sheet and Word documents. The Excel sheet is mostly being used to keep track of the inventory, to check howmany products are available , to place a new order is the number of products in stock is less.

Also the word documents are used to generate bills and reports. Bills are used as a proof of sales and the delivery to the product.

Limitations of Current System :

The Current System is a Dos based FoxPro program. Since a dos based broader viewing of product information is impossible. The System does not store any reports of purchase detail except stock. Because of this reason information regarding some reports are maintained manually.

- Information storage of customer, vehicle supplying company, employees details, workshop etc.
- Date storage of customer payments, discounts offered and rates applicable with effective from the predefined date / period.
- Department or user wise relevant screens for type of job / profile handled.
- User access control and rights management.
- Configurable tracking system for internal tracking as well as during the audits.

Introduction to Proposed System :

The Proposed System is managed by the **Visual Studio .Net 2008** which is user friendly Window for every user and for maintaining the database it is managed by the **Miscrosoft Sql 2005.**

Scope Of Proposed System :

The System proposed has many advantages over the current system.

- The proposed system is highly secured, because for login the system it require the password which is an important.
- It provides wide range of search criteria in each window the client is working for better and quicker solution.
- Automatically generates a list of products whose stock level is very low.
- Stores information about regular customer.
- This System can run on the any Windows Xp.
- This System also maintain the Camp Details, whenever the organization is organized the Camps.

Advantages of Proposed System over the Current System :

- The System will provide Security.
- There will be a proper Storing facility.
- This System will be the less time consuming.
- Give better solution as compared to the Current System

Feasibility Study :

The feasibility report studied the feasibility of the project and plays a major role in the analysis of the system. The very decision of whether to design a system or not depends in the feasibility study. The feasibility study can be categorized as follows:- A.

Economical Feasibility.

B. Technical Feasibility.

C. Operational Feasibility.

D. Behavioral Feasibility.

A) Economical Feasibility :-

The economical feasibility of the system is mainly concerned with its financial aspects. It explains whether the project is economically feasible. In other words it determines whether the investment that goes into the implementation of the project is recordable or not. The cost benefit analysis is a commonly used method in evaluating the effectiveness of the system. As the hardware and software are already available and no further investment is to be made in that direction, the only cost involved is that of implementing the system.

The system will be economically feasible considering the following aspects:-

- The system will save a lot of stationary.
- It will save time, which is otherwise wasted in manual process.
- At the time the system will require less manual power as compared to the current system.
- The storage and the handling problem with respect to records and register will be solved.

B) Technical Feasibility:-

It determines whether the technology needed for the proposed system is available and how this technology can be integrated into organization. Technical evaluation must also assess if the existing system can be upgraded to use it. The institute /company must be equipped with the pre-mentioned hardware and software requirements.

c) Operational Feasibility:-

Operational feasibility Is a measure of how well a proposed system solves the problems? There are two aspects of operational feasibility for the system. One is that of technical performance and other is the acceptance. Technical performance determines whether a system can provide correct and timely data required for the institute personal. Computerizing the existing system will ensure that it will provide accurate, timely and up to minute data to the management. The new system will automate the existing manual system and make it user-friendly. The proposed system will cut down unnecessary paperwork and time delay offered by the existing system.

D) Behavioral Feasibility:-

This involves questions such as how much time is available to build the new system, when it can be built, whether it interferes with normal business operations, type and amount of resources required, dependencies, etc. The project's alternatives are evaluated for their impact on the local and general culture. The environmental factors need to be considered and these factors are to be well known.

The Source of system Requirements :

Stakeholders : Stakeholders are all the people who have an interest in success of a new system. Stakeholders are categories into three groups.

User: => who actually use the system on a daily basis.

Clients: => who pay for and own the system

Technical staff: => the people who must ensure that the system operators in the

VARIOUS USER computing environment of the organization.

STAKEHOLDERS

A. BUSINESS OPERATIONS USERS

A. QUERY USER

B. MANAGEMENT USER

C. EXECUTIVE USER

A) Business operation User

Business operation user Are the people who use the system to perform day today operation of an organization that is transaction. Ex. student registration

B) Query User:

Query is request for information for the system or from a database.
Eg. viewing banking transaction.

C) MANAGEMENT USER:

Management is responsible for seeing that company is performing its daily processor efficiently and effectively.

D) EXECUTIVE USER :

EXECUTIVE IS PERSON OR GROUP WHO IS PROVIDING FOUNDING FOR THE PROJECT**.**

Gantt chat

→→→

Date / Phase	May	June	July	Aug	Sep	Oct	Nov	Dec	Jan
Preliminary investigation									
System analysis									
System Design									
System coding									
System uploading									
Future Enhancement									
Reference And Bibliography									

System Analysis

REQUIREMENT ANALYSIS

1. FUNCTIONAL REQUIREMENT :

a) **Strong Data Validation:** There is possibility that user might enter wrong data and wrong data may cause inconsistency to the database and hence to the system. To avoid this, data should be validated whenever entered.

b) **Automatic updating of the database:** After any transaction is performed, it is necessary that the updating should be reflected in the database without any **inconsistency.**

c) **Provide efficiency querying based on user requests:** The major purpose is to generate efficient reports on any user request. This will be done by our query processing system, which should be able to process any combination of queries will be done dynamically at run time depending on the user.

2. EXTERNAL INTERFACE REQUIREMENTS

a) **User friendly interface:** The interface should be developed in such a manner that it is very user friendly, this not only improve interaction but also saves data entry time.

b) **Making well designed forms for capturing data:** The forms for capturing the data should be well-designed

using pop-down menus and drag & drop facilities, which reduce the data entry effort on the part of the user.

3. PERFORMANCE REQUIREMENTS

a) **Security:** All users are not allowed to access the database. Hence there is a need to check authority of every user. Username and Password validation helps to deny unauthorized access to the system. There are 2 main types of users who will be using the software They are:

1) Admin 2) User

Each user is given the specific rights to access the data in Read only, Read Write, Delete.

HARDWARE AND SOFTWARE USED :

The development of mobile sales and service management system requires the following resources

PLATFORM : Windows XP Professional **FRONT**

END: Visual Studio 2008.

BACK END: SQL Server 2005

HARDWARE REQUIREMENTS:

Intel Pentium III 733 MHz or Higher.

256 MB RAM or Higher

SCOPE OF THE PROJECT :

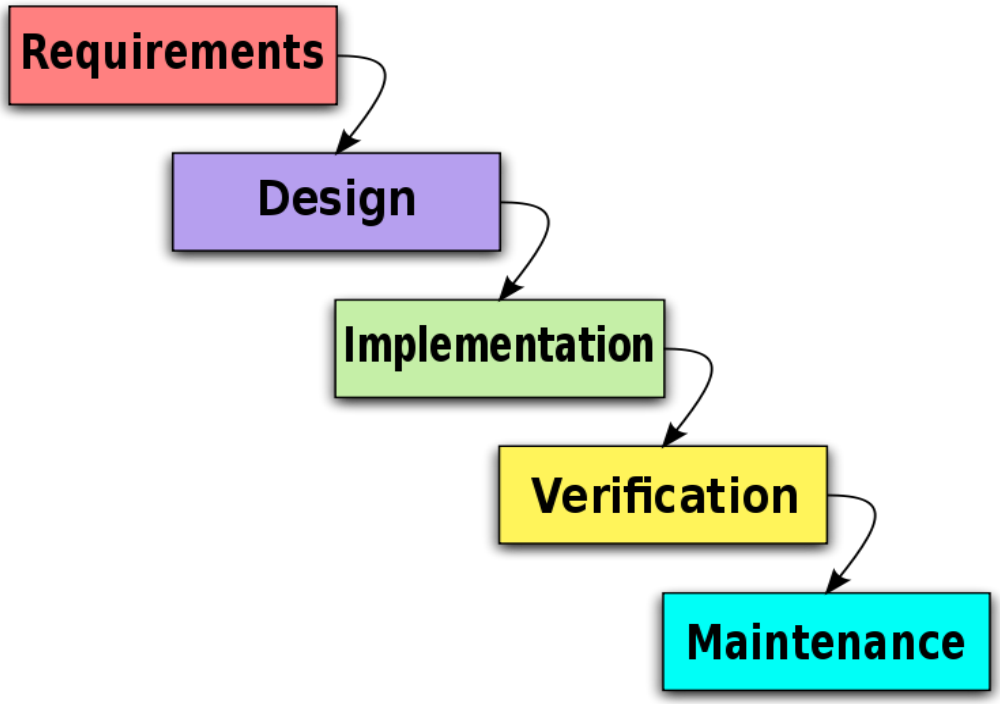
The proposed system for the Automobile showroom will be developed in Visual Studio 2008 and SQL Server 2005 will be used in backend. This new system will allow the users to quickly insert, delete, update and retrieve data from the system. This new system will allow security to the data, by mean of authorizing users. Only those users who have a valid user-id and password can access the system. Also the proposed system will have an MDI form, which will provide a single page control. By mean of the MDI form the user of the system can go to different options of the system from the single page itself. This will provide an easy to use software in term of GUI and also the time of the user will be saved.

OBJECTIVE OF PROJECT :

- Information storage of customer, vehicle supplying company, employees details, workshop etc.
- Date storage of customer payments, discounts offered and rates applicable with effective from the predefined date / period.
- Department or user wise relevant screens for type of job / profile handled.
- User access control and rights management.
- Configurable tracking system for internal tracking as well as during the audits.
- Uploading facility for the information received in soft copy to the system.
- Payment details of the employees as per the work done.

ITERATIVE MODEL :

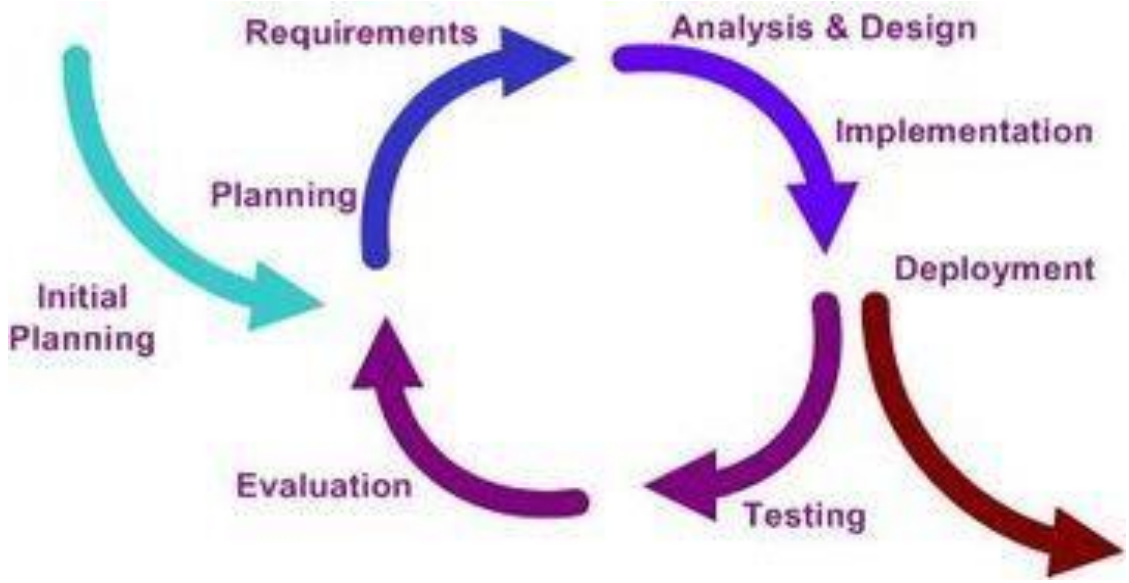
Iterative and Incremental development is a cyclic software development process developed in response to the weaknesses of the Waterfall model. It starts with an initial planning and ends with deployment with the cyclic interaction in between.



The iterative and incremental development is an essential part of the Rational Unified Process. The Dynamic systems development model, Extreme Programming and generally the Agile software development frameworks.

The Basic idea:

The basic idea behind iterative enhancement is to develop a software system incrementally, allowing the developer to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the system.



Learning comes from both the development and use of the system, where possible key steps in the process are to start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving sequence of versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The level of design detail is not dictated by the interactive approach. In a light-weight iterative project the code may

represent the major source of documentation of the system; however, in a mission-critical iterative project a formal software design document may be used. The analysis of an iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, usability, reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

Iterative development:

Iterative development slices the deliverable business value (system functionality) into iterations. In each iteration a slice of functionality is delivered through cross-discipline work, starting from the model/requirements through to the testing/deployment. The unified process groups iterations into phases: inception, elaboration, construction, and transition. Inception identifies project scope, risks, and requirements (functional and nonfunctional) at a high level but in enough detail that work can be estimated.

Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements. Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements. Transition delivers the system into the production operating environment. Each of the phases may be divided into 1 or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their workproduct backlog full.

REQUIRMENT ANALYSIS :

After obtaining the background knowledge, the information on the existing system, its inputs, outputs, costs and other important requirement and features have to be collected and analyzed. The following tools were used for knowing more about the system and gathering more information for developing a new system. For developing a system for an organization one need to acquire necessary information regarding the system. Information plays vital role in an organization collection of information before designing a system solves many problems. Following are the fact finding techniques used.

Interview:

This technique of fact finding was especially helpful for gathering the basic information of the system, Interview being a one-to-one conversation technique was very useful in knowing what the user thinks about the existing system and what he wants in the new one. Using the technique I gathered very useful information like: The drawback of the Existing System. The working habits of the employees of the organization which will help me to build the new system. More inputs on how new system should be. What additional function and utilities are to be added?

The exact logic on how the business takes place in the organization, etc.

Record Inspection:

In order to know the internal working of the organization this technique is essential. This technique helped me a lot to know more about:

- The paper work involved in the organization.
- The forms and documents involved in the system.

- The importance of the documents involved in the organization.
- Their forms and their layouts.

EVENT TABLE

Sr No	Event	Trigger	Source	Activity	Response	Destination
1) 1.	Create Customer	New Customer Details	Admin	Creates Customer	Customer Creation Confirmed	Admin
2.	Update Customer	Request Customer Details for Updating	Admin	Update Customer	Customer Updating Confirmed	Admin
3.	Delete Customer	Request Customer Details for Deleting	Admin	Delete Customer	Customer Deleting Confirmed	Admin
2)	Search Customer	Request Customer Details for Searching	Admin	Search Customer	Customer Search Confirmed	Admin
3) 1.	Create Employee	New Employee Details	Admin	Creates Employee	Employee Creation Confirmed	Admin
2.	Update Employee	Request Employee Details for Updating	Admin	Update Employee	Employee Updating Confirmed	Admin

3.	Delete Employee	Request Employee Details	Admin	Delete Employee	Employee Deleting	Admin
		for Deleting			Confirmed	
4)	Search Employee	Request Employee Details for Searching	Admin	Search Employee	Employee Search Confirmed	Admin
5) 1.	Add Supplier	New Supplier Details	Admin	Add Supplier	Supplier Creation Confirmed	Admin
2.	Update Supplier	Request Supplier Details for Updating	Admin	Update Supplier	Supplier Updating Confirmed	Admin
3.	Delete Supplier	Request Supplier Details for Deleting	Admin	Delete Supplier	Supplier Deleting Confirmed	Admin
6)	Search Supplier	Request Supplier Details for Searching	Admin	Search Supplier	Supplier Search Confirmed	Admin

7)	Add New Model	New Model Details	Admin	Add Model	Model Creation Confirmed	Admin
8)	Delete Model	Request Model Details	Admin	Delete Model	Model Deleting Confirmed	Admin
		for Deleting				
9)	Change Price of Model	Request Model Details for Updating	Admin	Update Model Details	Model Update Confirmed	Admin
10)	Customer Lodges Complaint	Request To Add Complaint	Admin	Customer Complaint Details Added	Complaint Confirmed	Admin
11)	Add New Order Details	Request for Adding New Order Details		Creates New Order Details	Order Details Confirmed	

UML Diagrams

Introduction:

UML stands for Unified Modeling Language. It represents a unification of the concepts and notations presented by the three amigos in their respective books. The goal is for UML to become a common language for creating models of object oriented computer software.

In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

- Class Diagram
- Use-Case Diagrams
- Component Diagram
- Dataflow Diagrams

Use Case Diagram

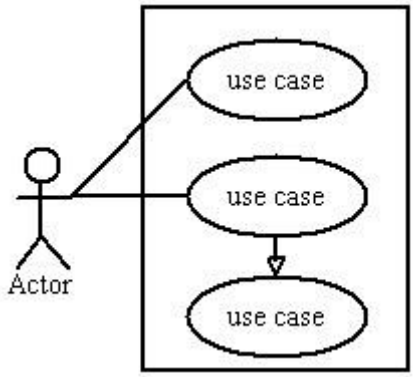
A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

Notations

System

Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

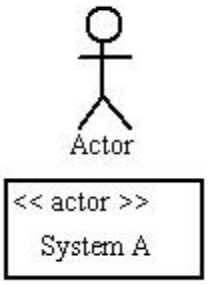


Use Case

Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.

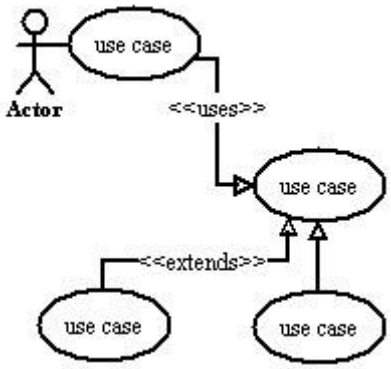
Actors

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

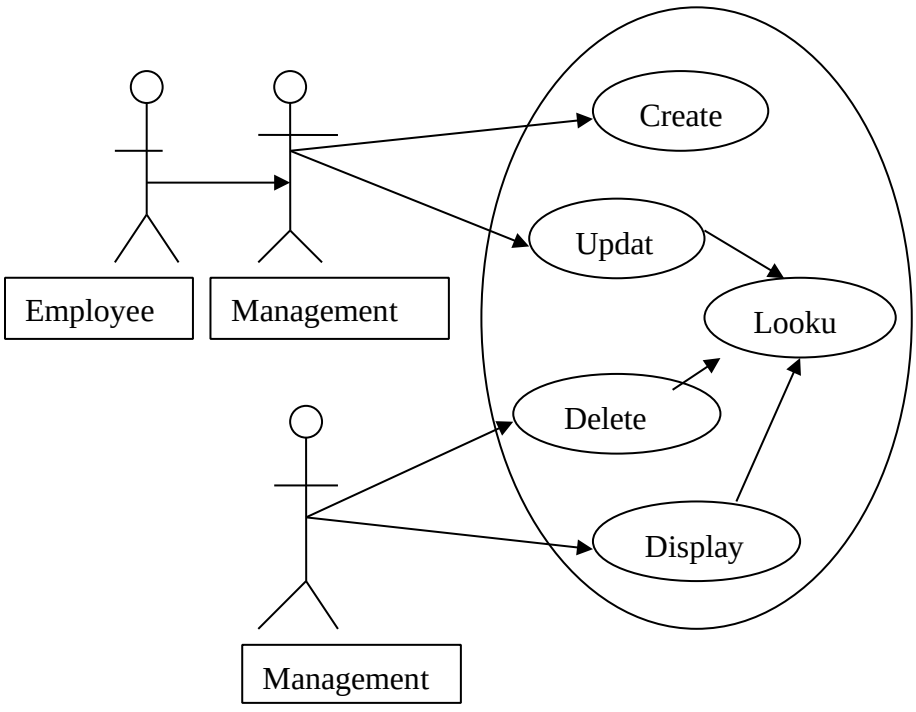


Relationships

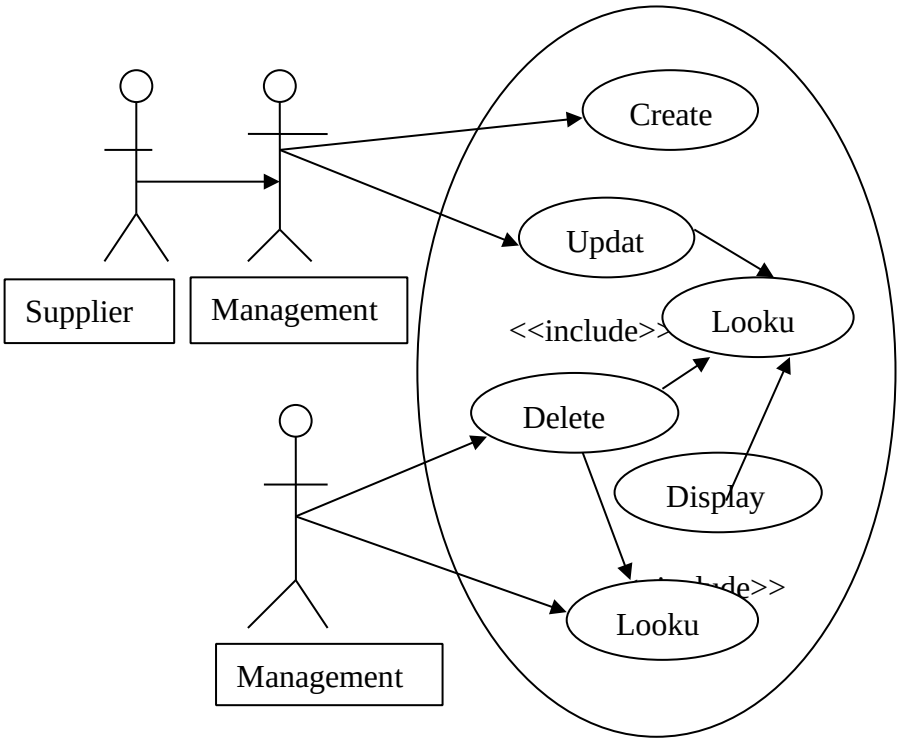
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labelled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case**.**



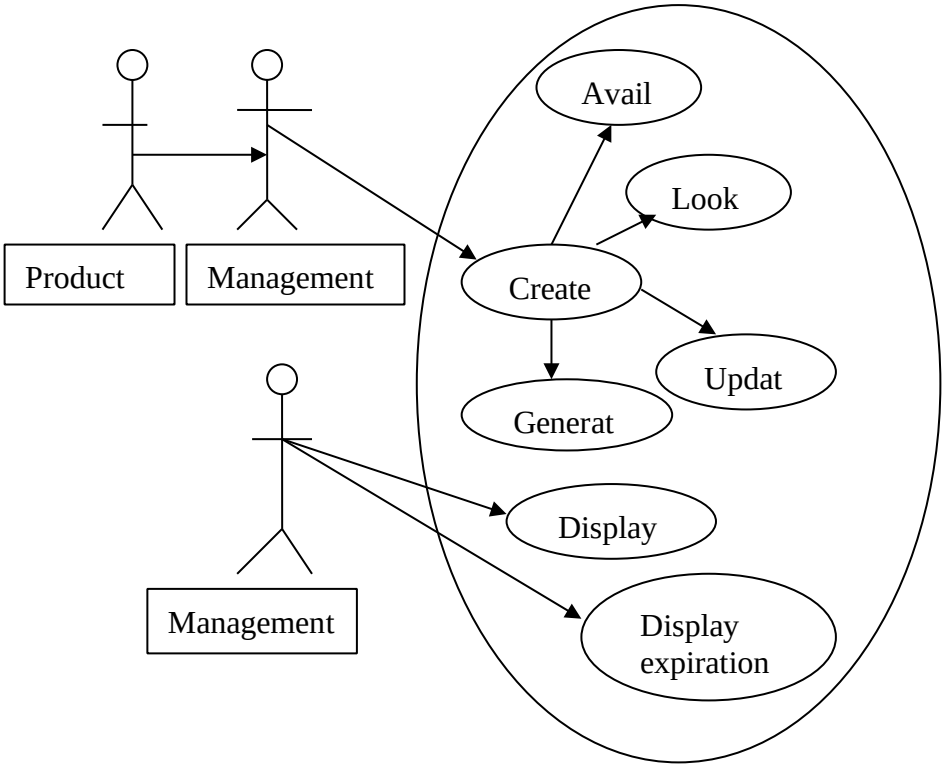
Employee subsystem :



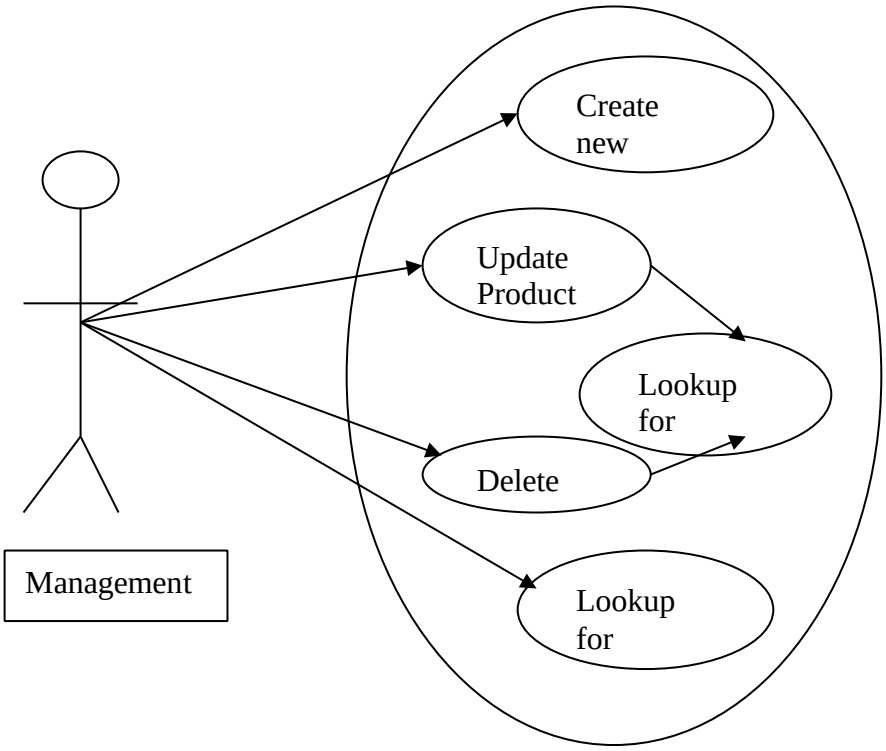
Supplier subsystem :



Product subsystem :



Product detail subsystem :


Create
new
Update
Product
Lookup
for
Delete
Lookup
for
Management

E-R DIAGRAM

To develop a relationship based on the entites present in the system we make use of ERD

Notations

Entity

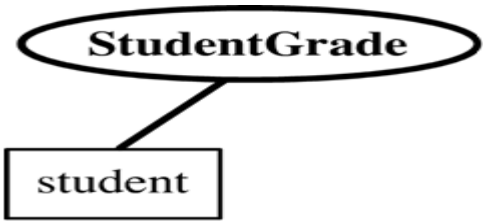
An **entity** is a real-world item or concept that exists on its own. The set of all possible values for an entity is the **entity type**.

eg. ER diagram notation for entity *student*

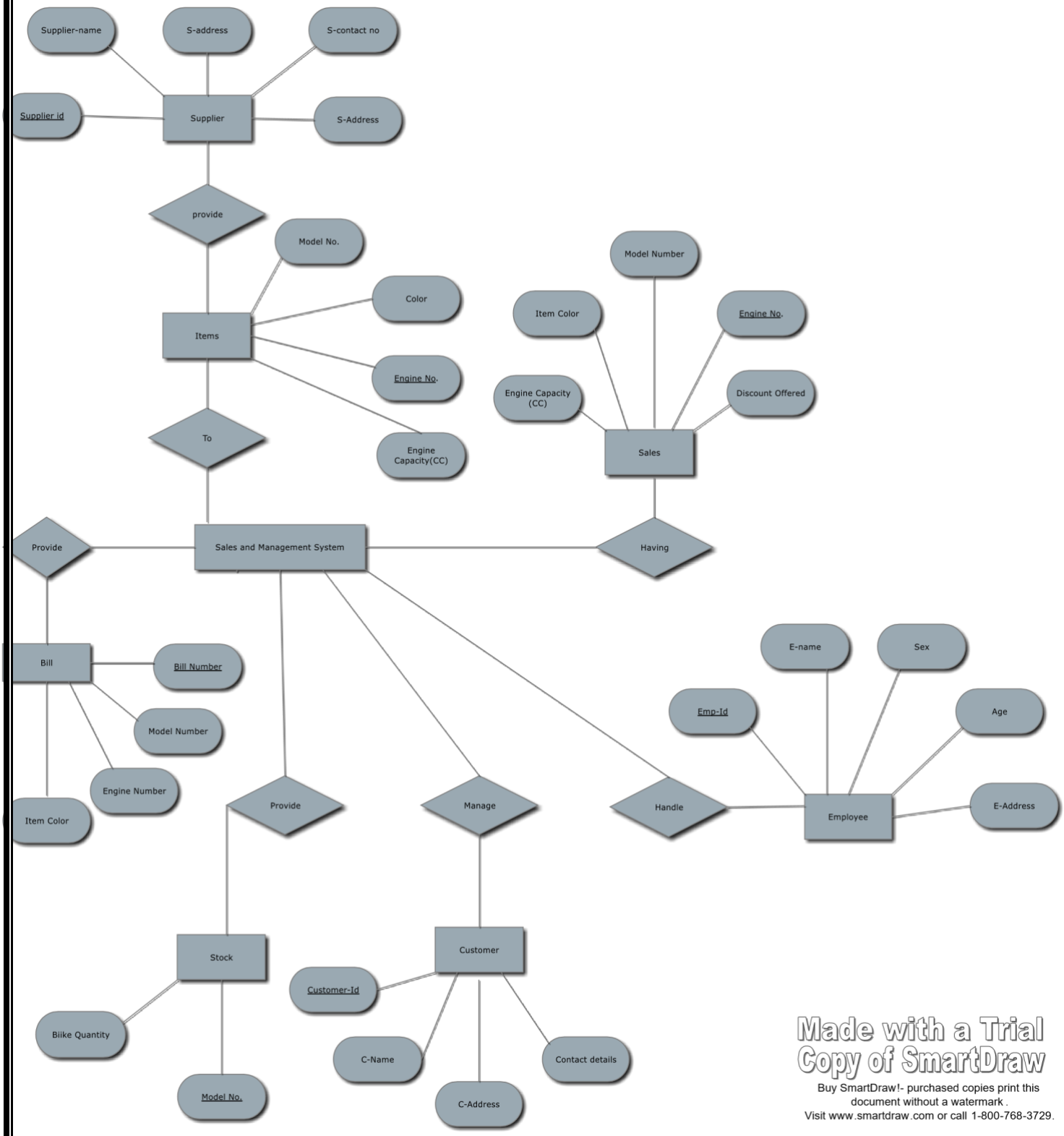
Attribute

An **attribute** of an entity is a particular property that describes the entity. The set of all possible values of an attribute is the **attribute domain**.

eg . ER diagram notation for an attribute domain (*StudentGrade*) of an entity type (*student*)



Entity Relationship Diagram



Activity Diagram

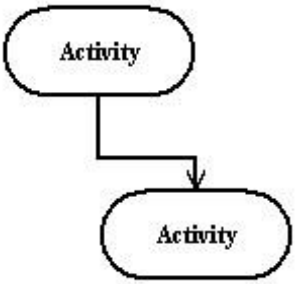
- Diagramdiagram illustrates the dynamic nature of a system by modeling the flow of control from to activity.
- An activity represents an activity operation on some class in the system that results in a change in the state of the system.
- Basic Activity Diagram Symbols and Notations.

Action states

Action states represent the noninterruptible actions of objects. You can draw an action state using a rectangle with rounded corners.

Action Flow

Action flow arrows illustrate the relationships among action states.

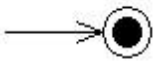


Initial State

A filled circle followed by an arrow represents the initial action state.

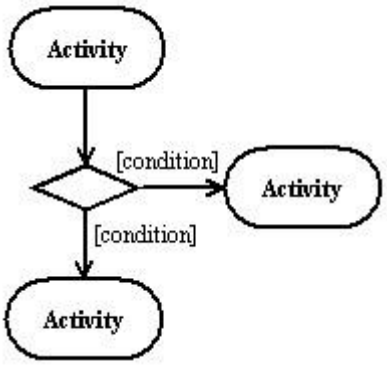
Final State

An arrow pointing to a filled circle nested inside another circle represents the final action state.



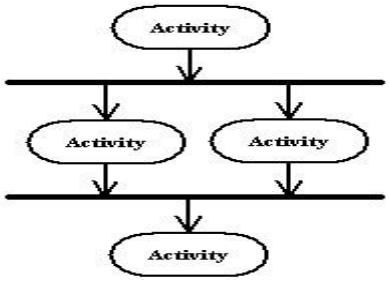
Branching

A diamond represents a decision with alternate paths. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths
"else."



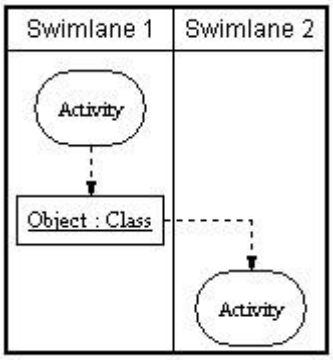
Synchronization

A synchronization bar helps illustrate parallel transitions. Synchronization is also called forking and joining.

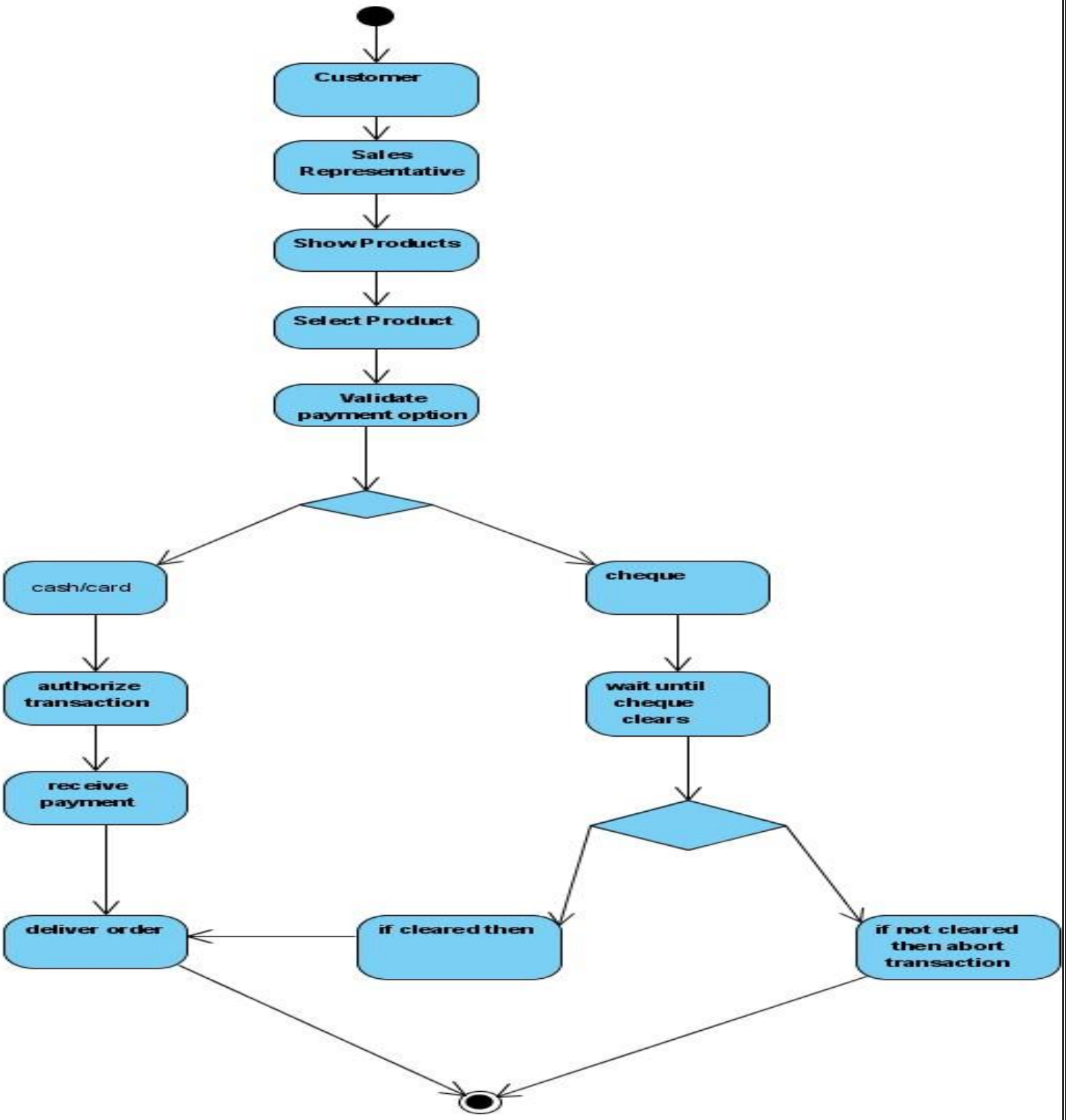


Swim lanes

Swim lanes group related activities into one column**.**



Activity Diagram :



CLASS DIAGRAM

In software engineering, a **class diagram** in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake

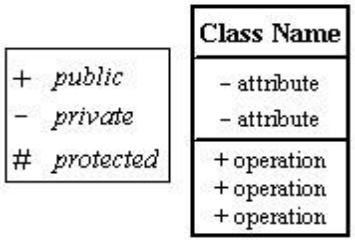
Class Diagram Components

Active Class

Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.

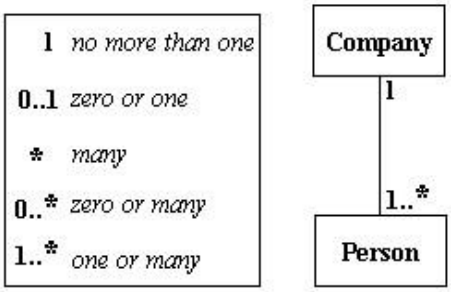
Visibility

Use visibility markers to signify who can access the information contained within a class**.**

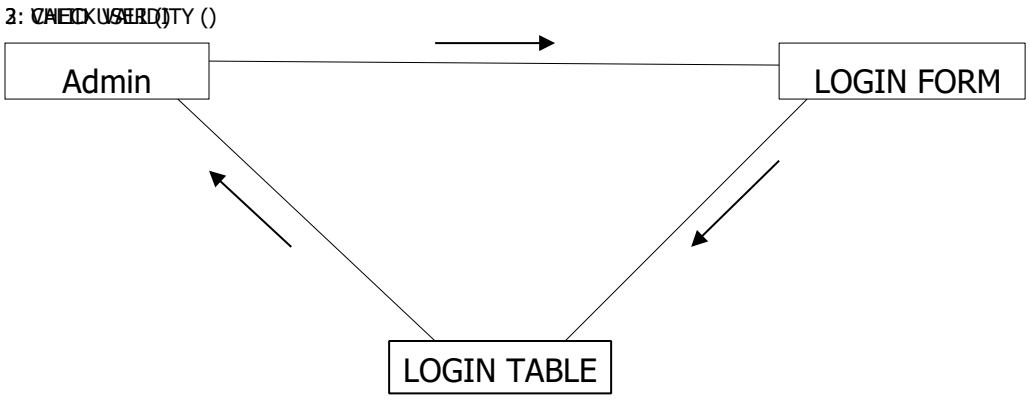


Multiplicity (Cardinality)

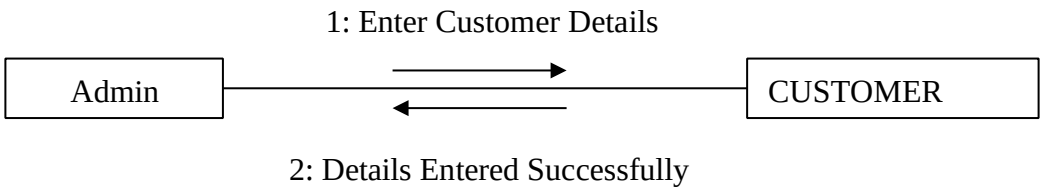
Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class



Collaboration Diagram for Order Details



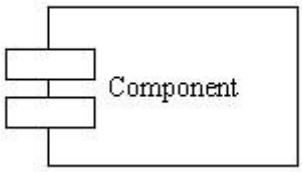
Collaboration Diagram for Customer Details



System design

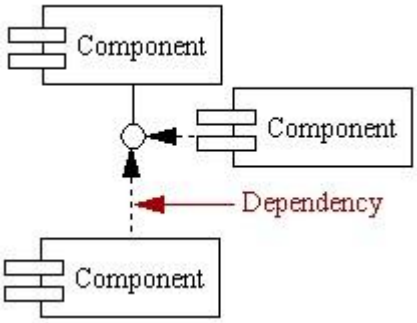
Component

A component is a physical building block of the system. It is represented as a rectangle with tabs.



Dependencies

Draw dependencies among components using dashed arrows.



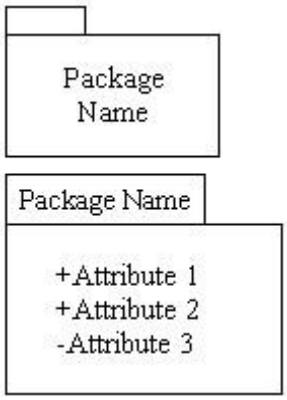
Package Diagram

Package diagrams organize the elements of a system into related groups to minimize dependencies among them.

Notations

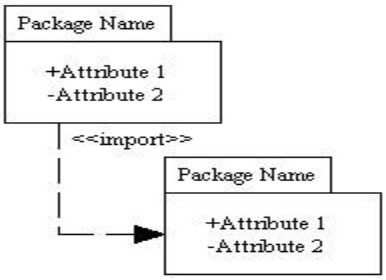
Packages

Use a tabbed folder to illustrate packages. Write the name of the package on the tab or inside the folder. Similar to classes, you can also list the attributes of a package.



Dependency

Dependency defines a relationship in which changes to one package will affect another package. Importing is a type of dependency that grants one package access to the contents of another package**.**



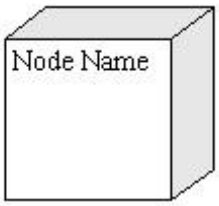
Deployment Diagram

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

Notations

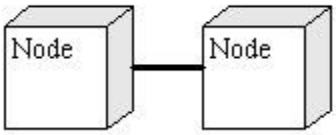
Component

A node is a physical resource that executes code components.



Association

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



Components and Nodes

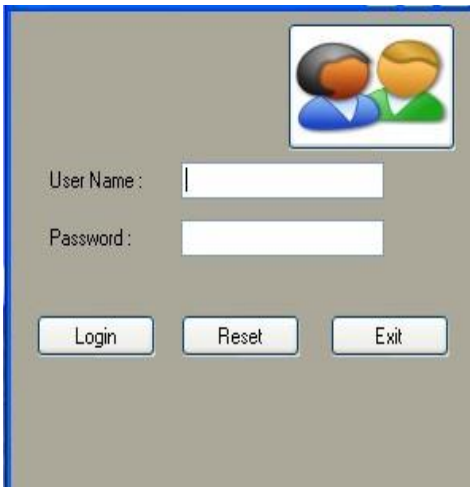
Place components inside the node that deploys them.

System coding

Interface Diagrams (Screen Shots)

Form Design

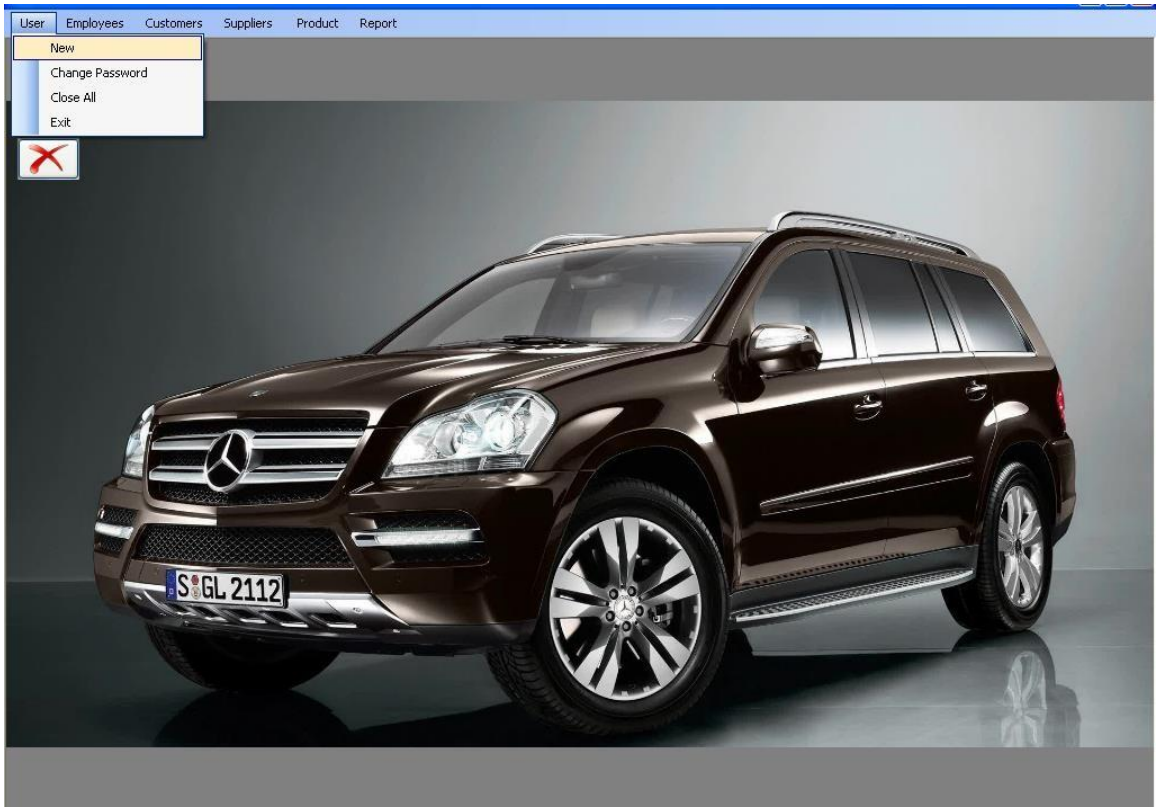
Login Form



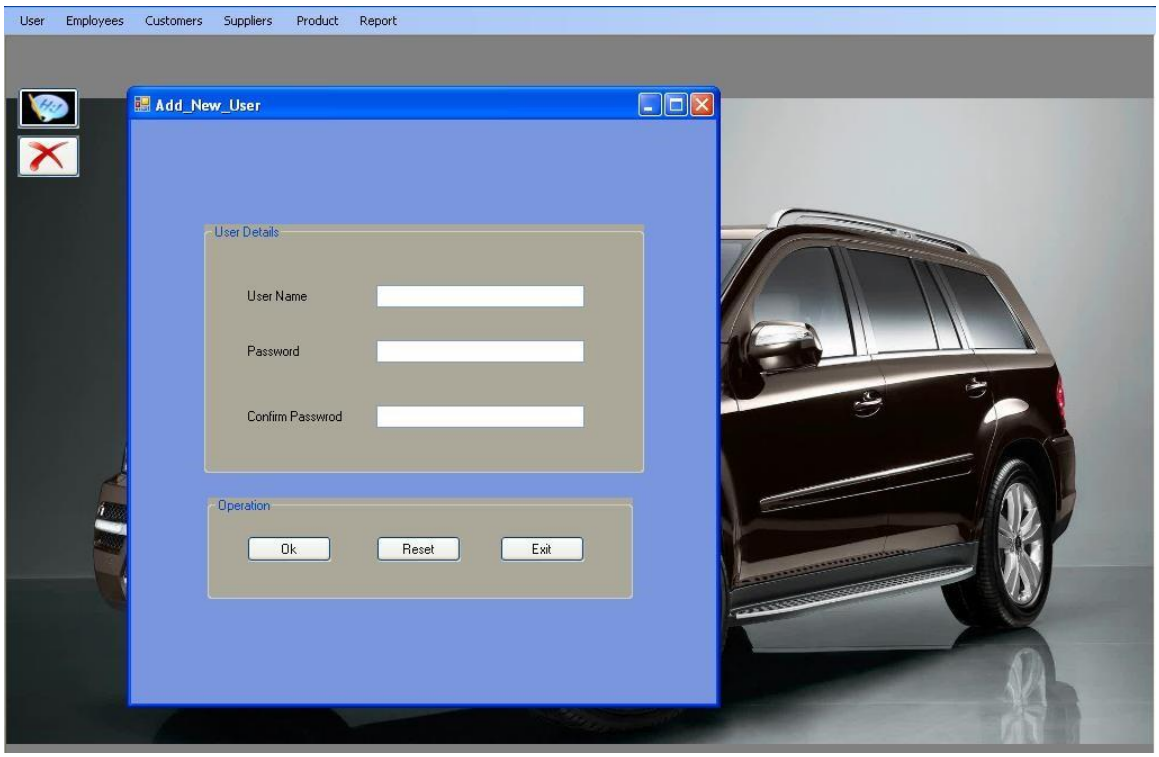
Main Form


User
Employees
Customers
Suppliers
Product
Report
S GL 2112

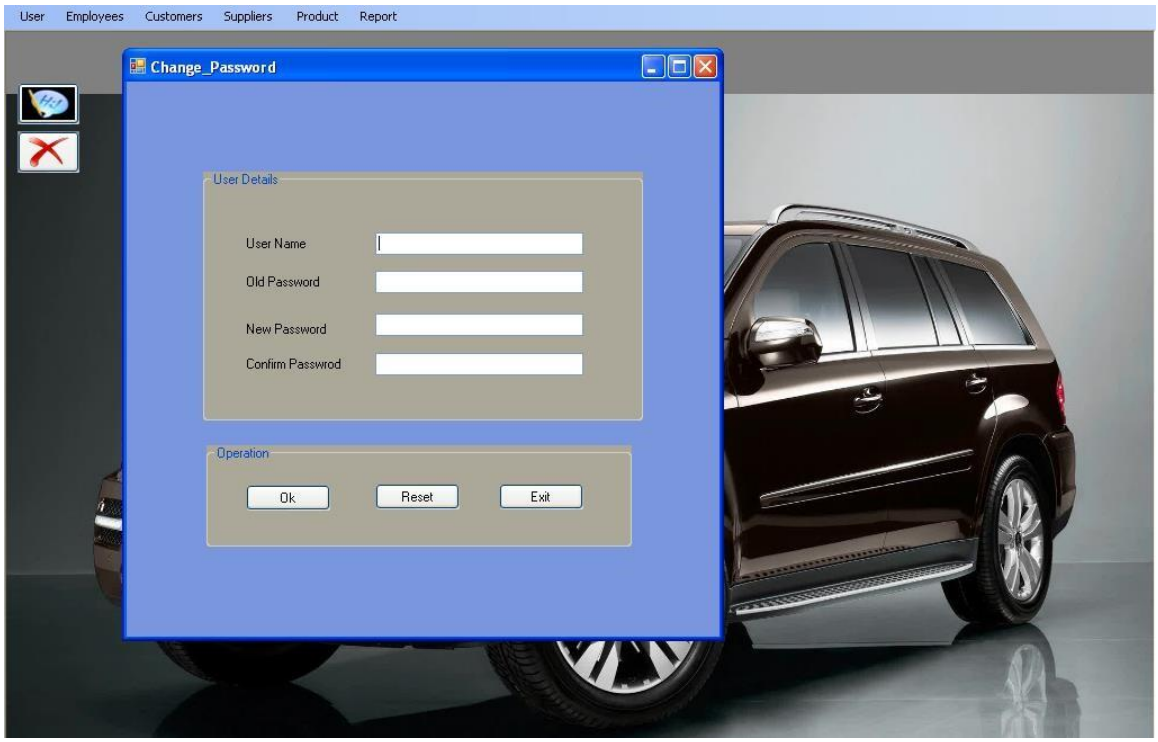
Different Sub menus included in User



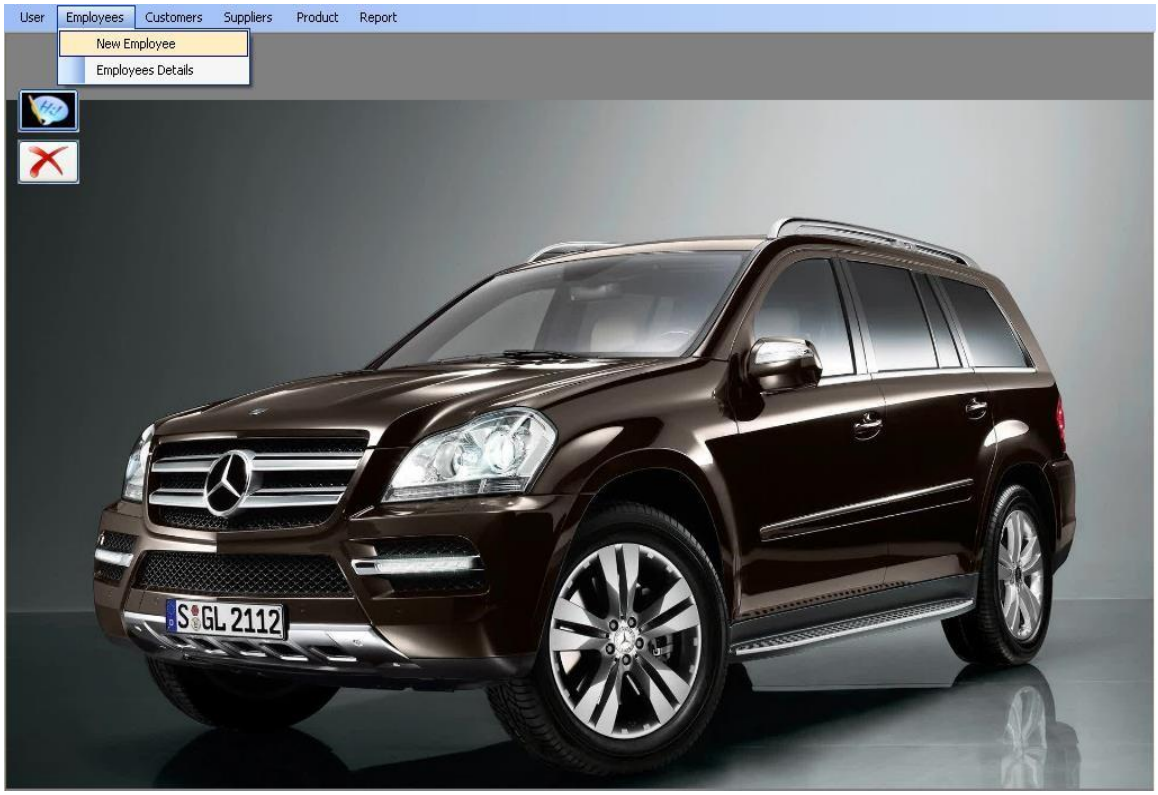
Adding New User



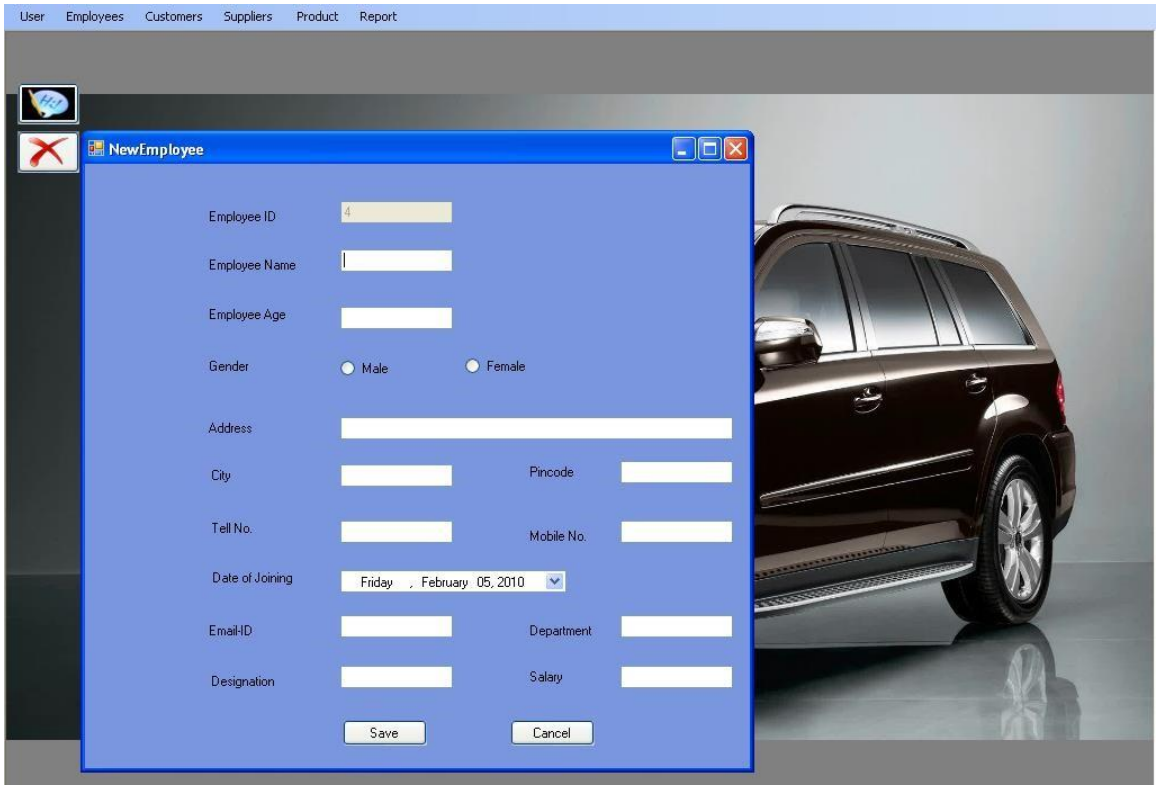
Change Password



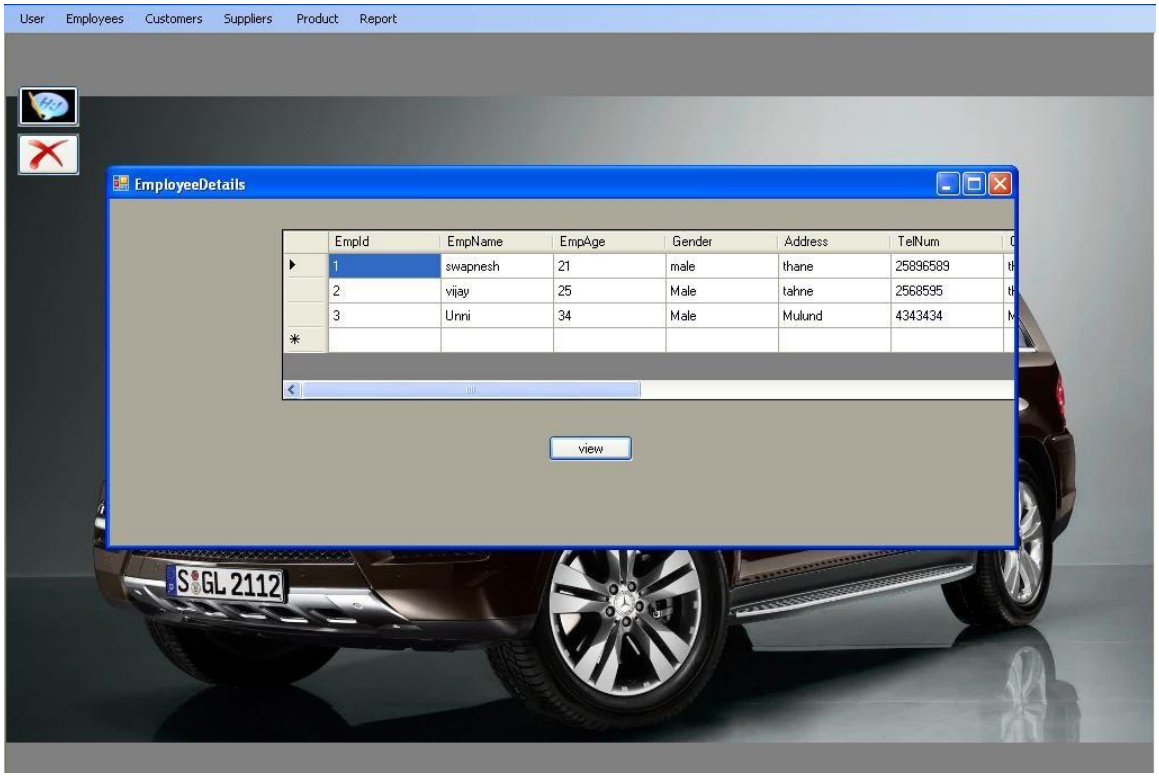
Different Sub menus included in Employees



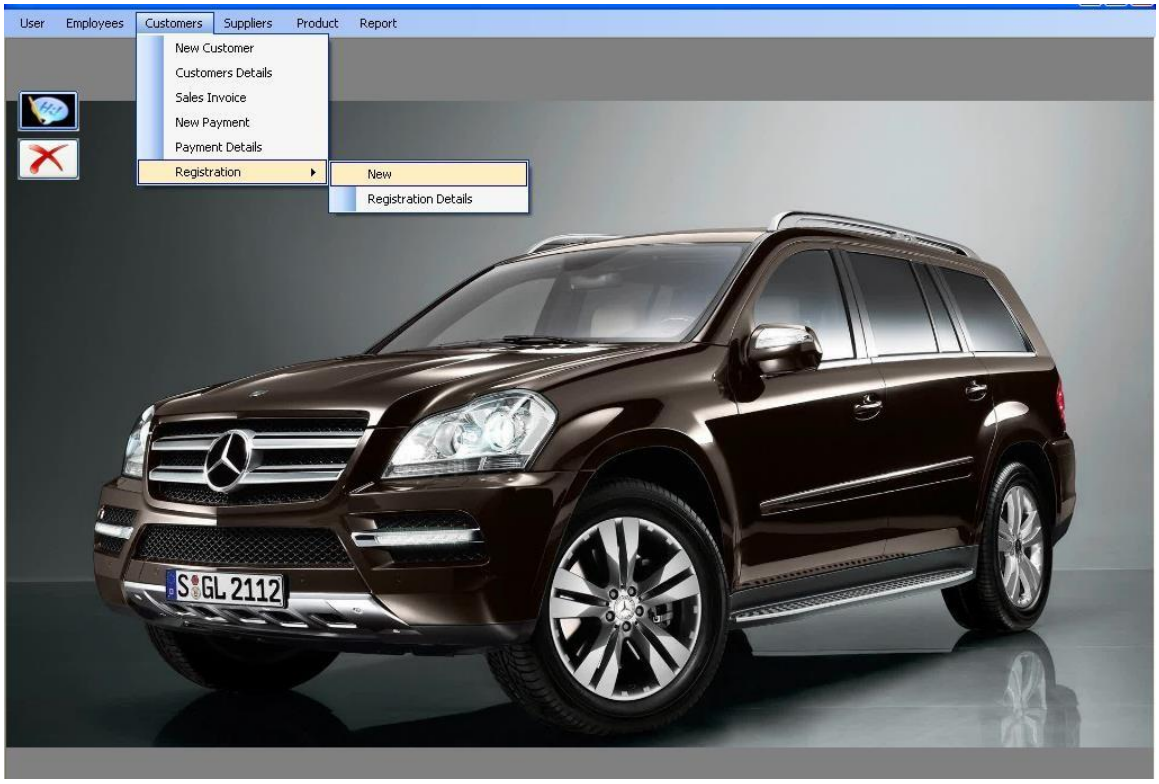
New Employee Form



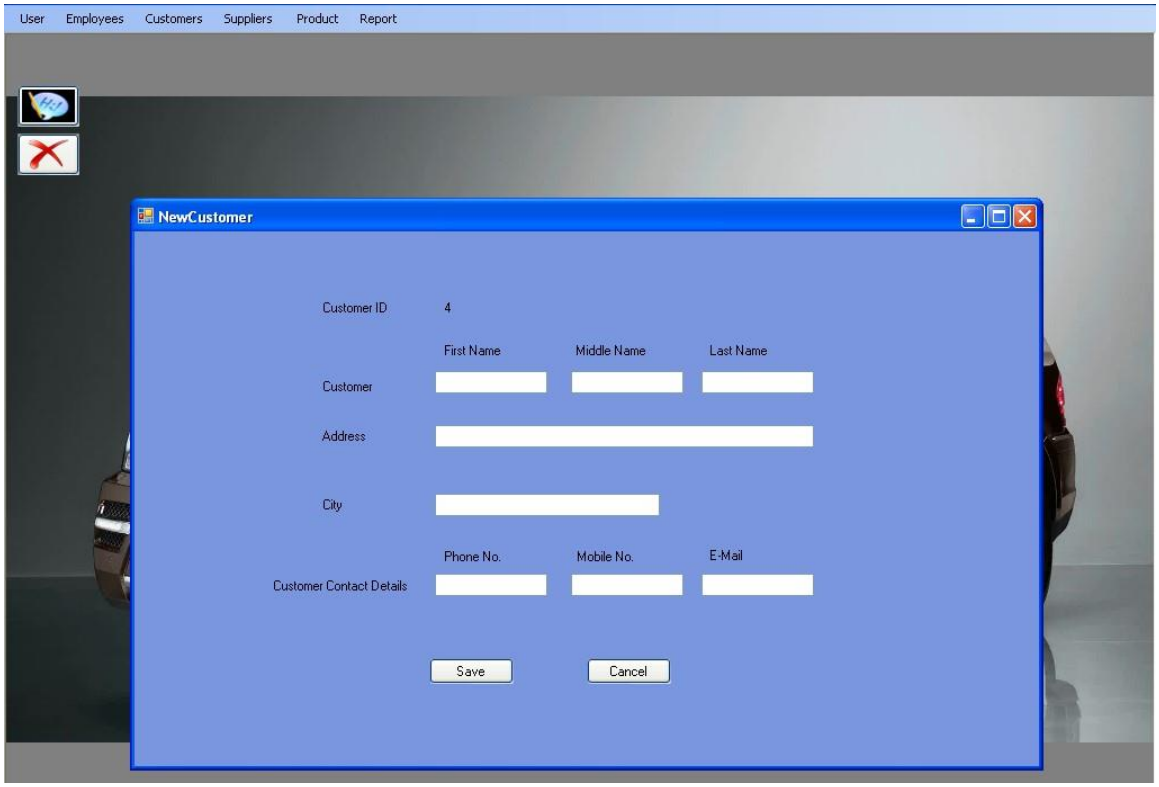
Employee Details Form



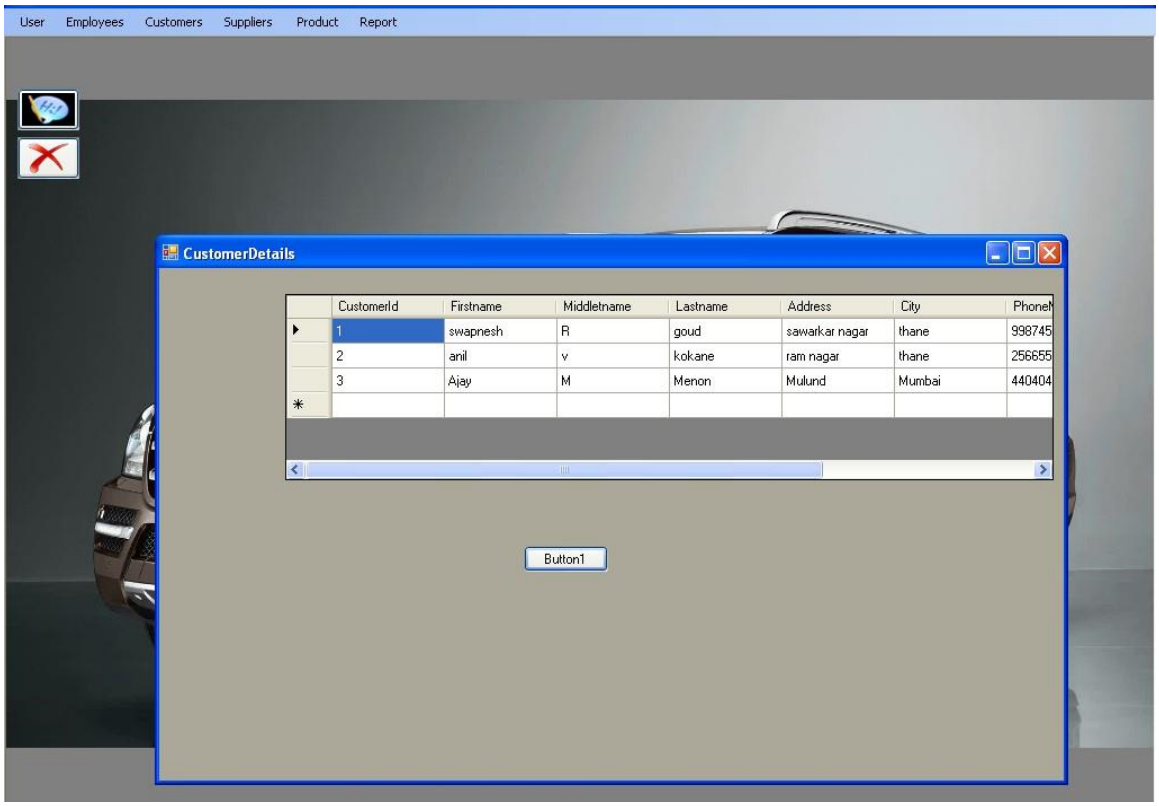
Different Sub menus included in Customers



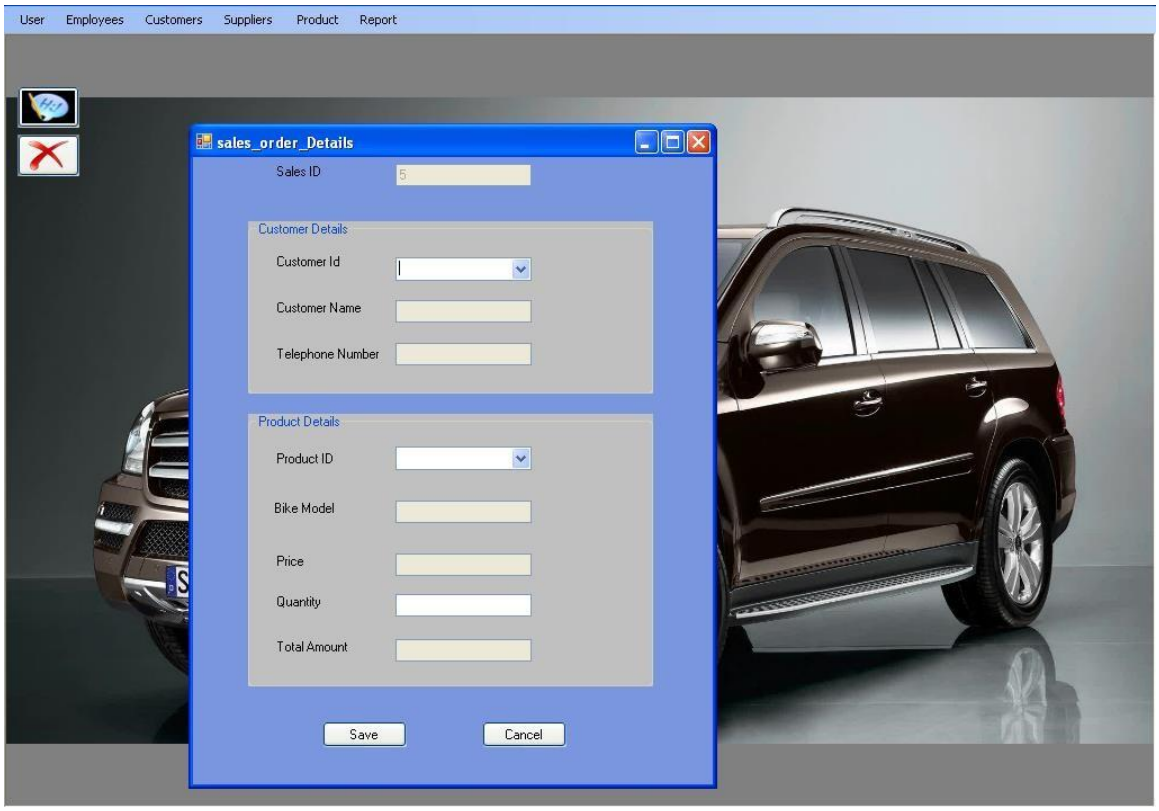
Adding new Customer Form



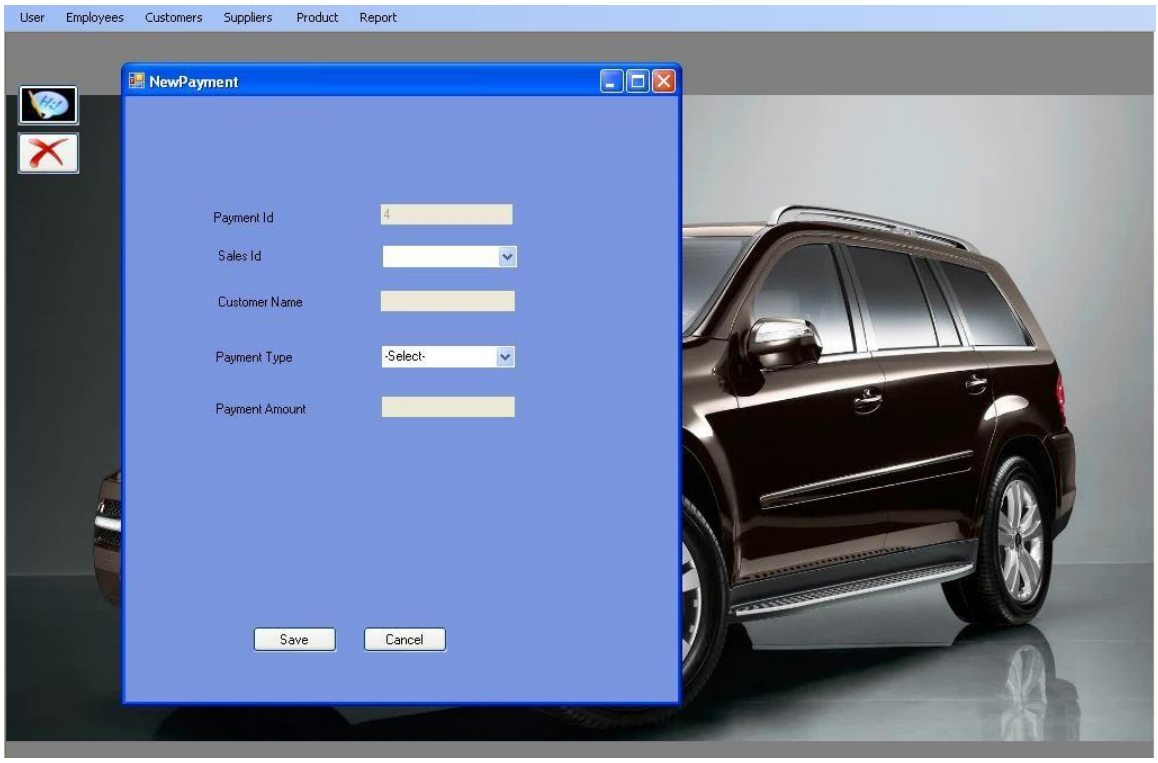
Customers Details From



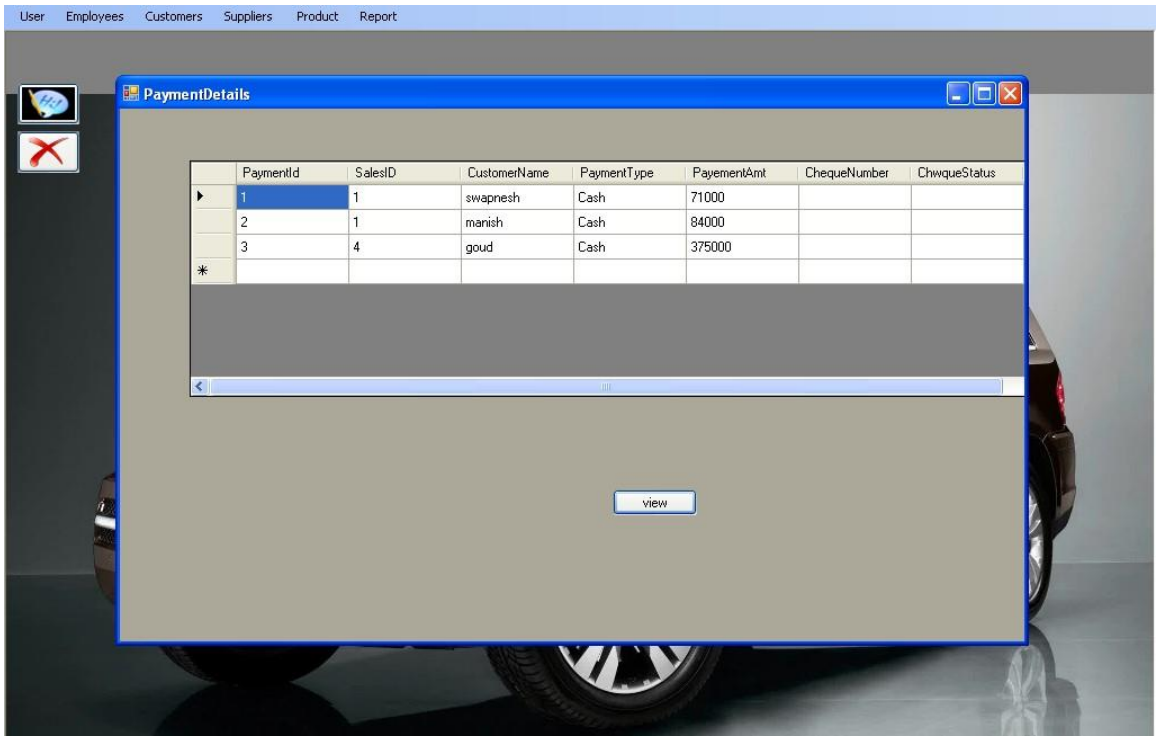
Sales Order Details Form



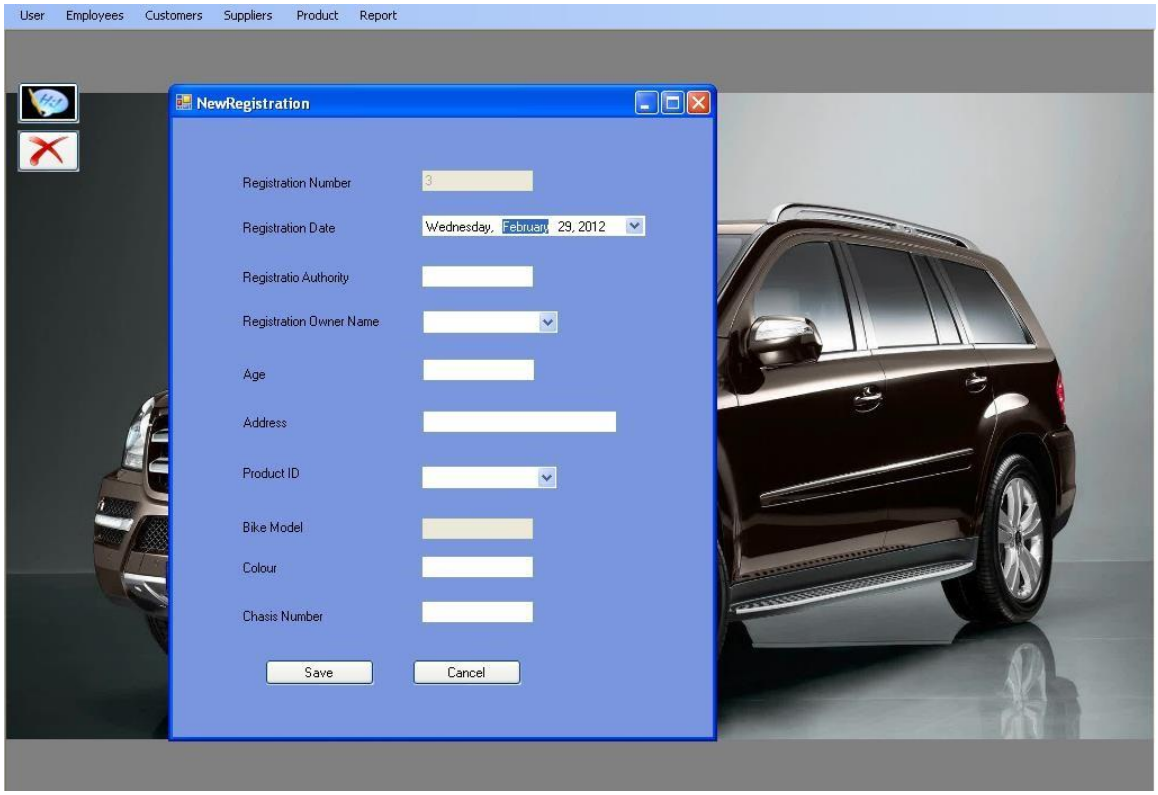
New Payment Form



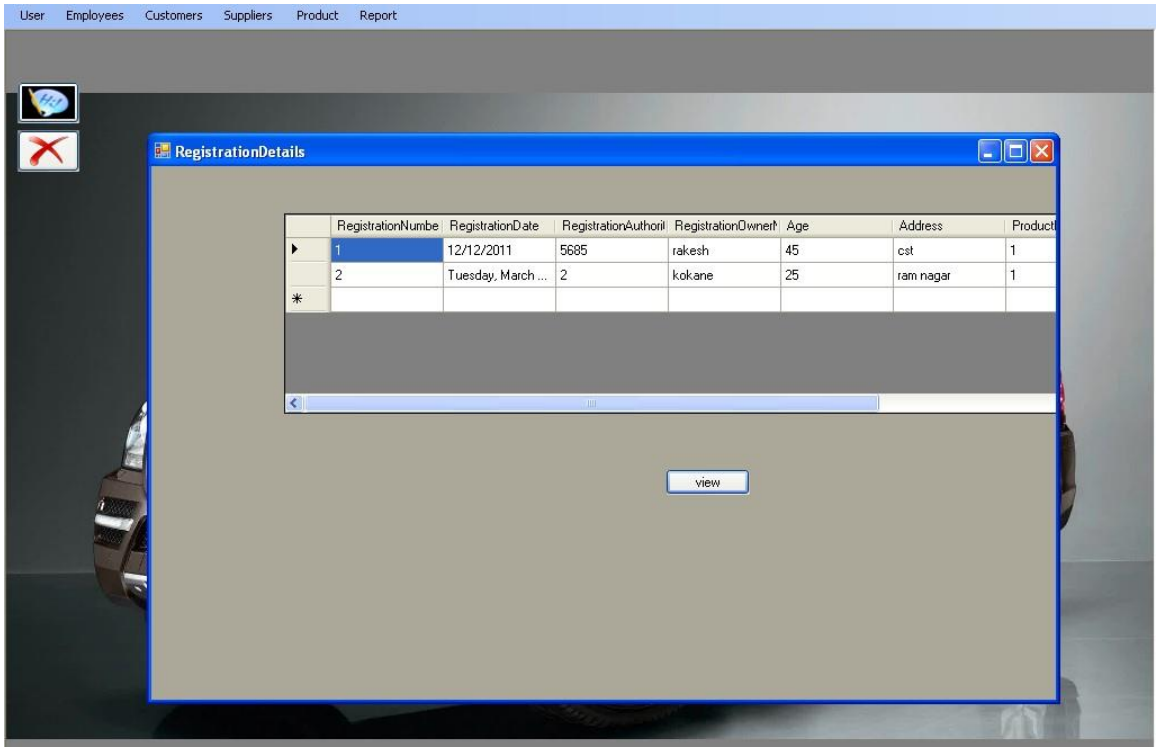
Payment Details Form



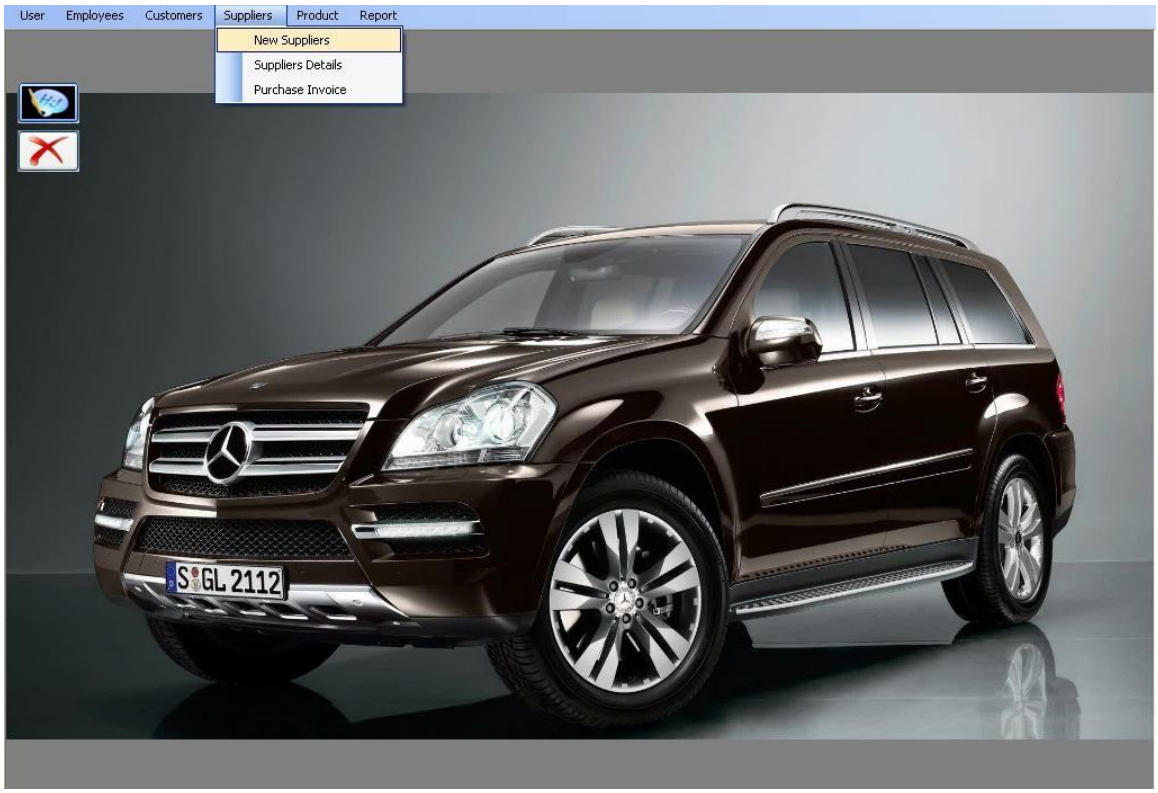
Registration Form



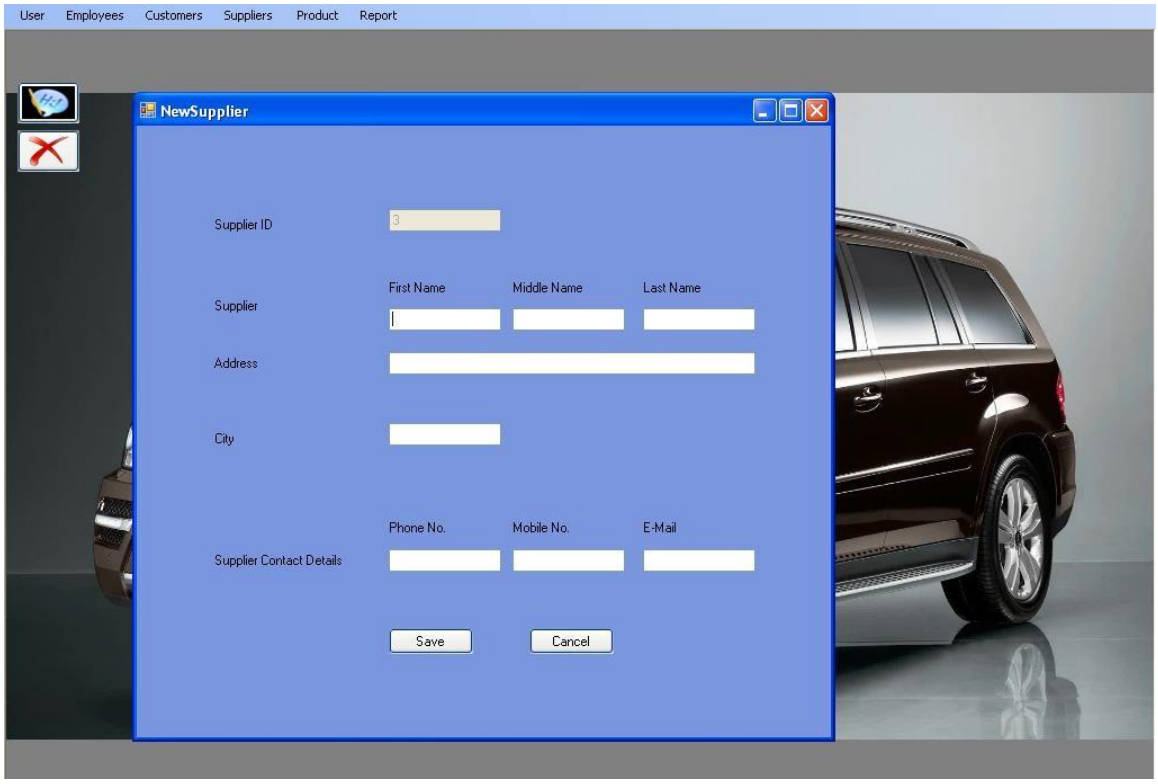
Registration Details Form



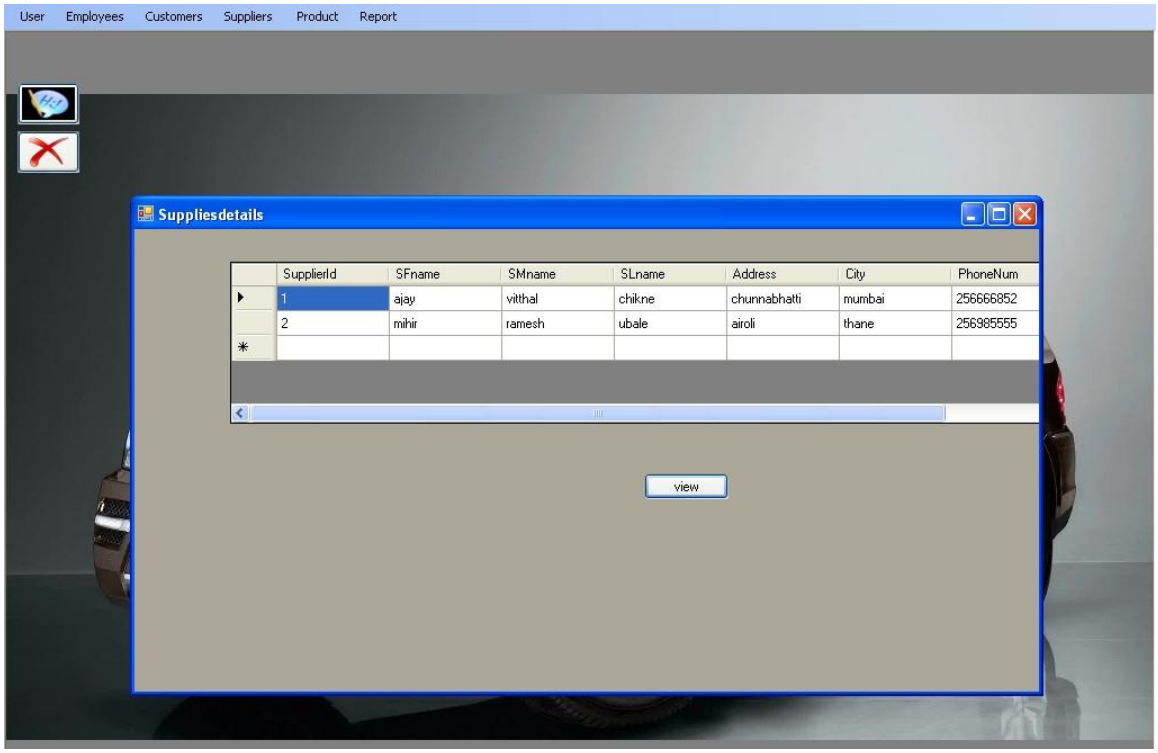
Different Sub menus included in Suppliers



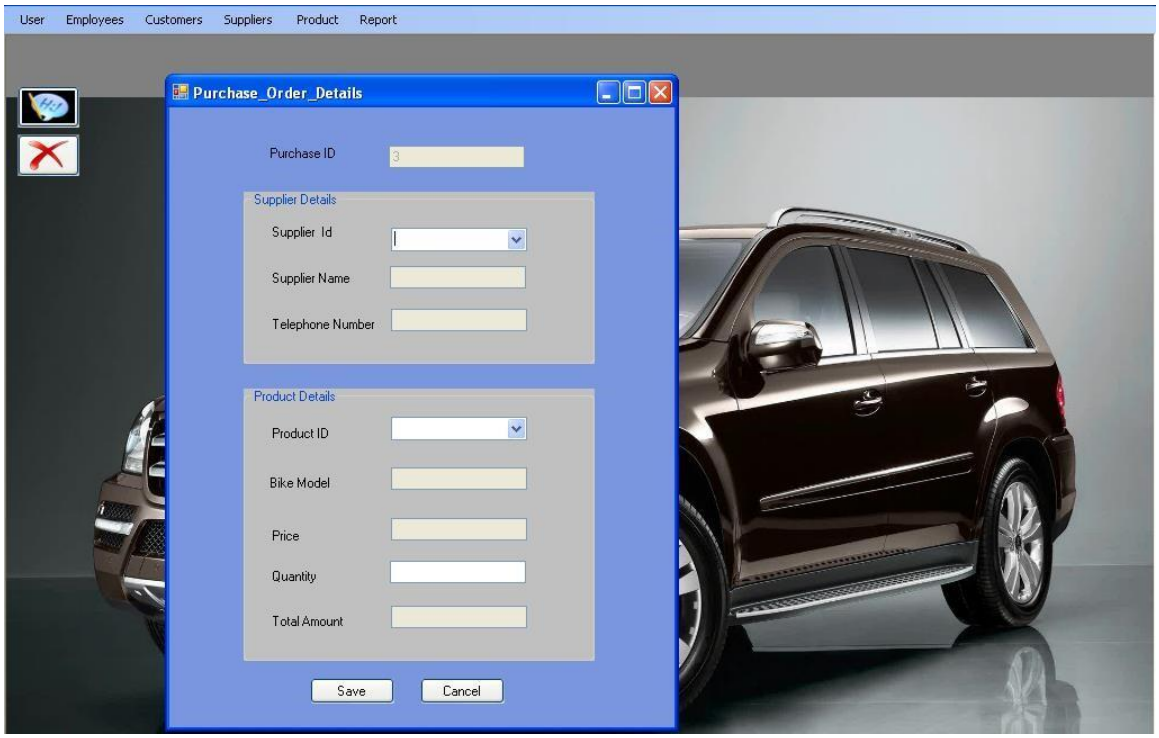
Supplier Form



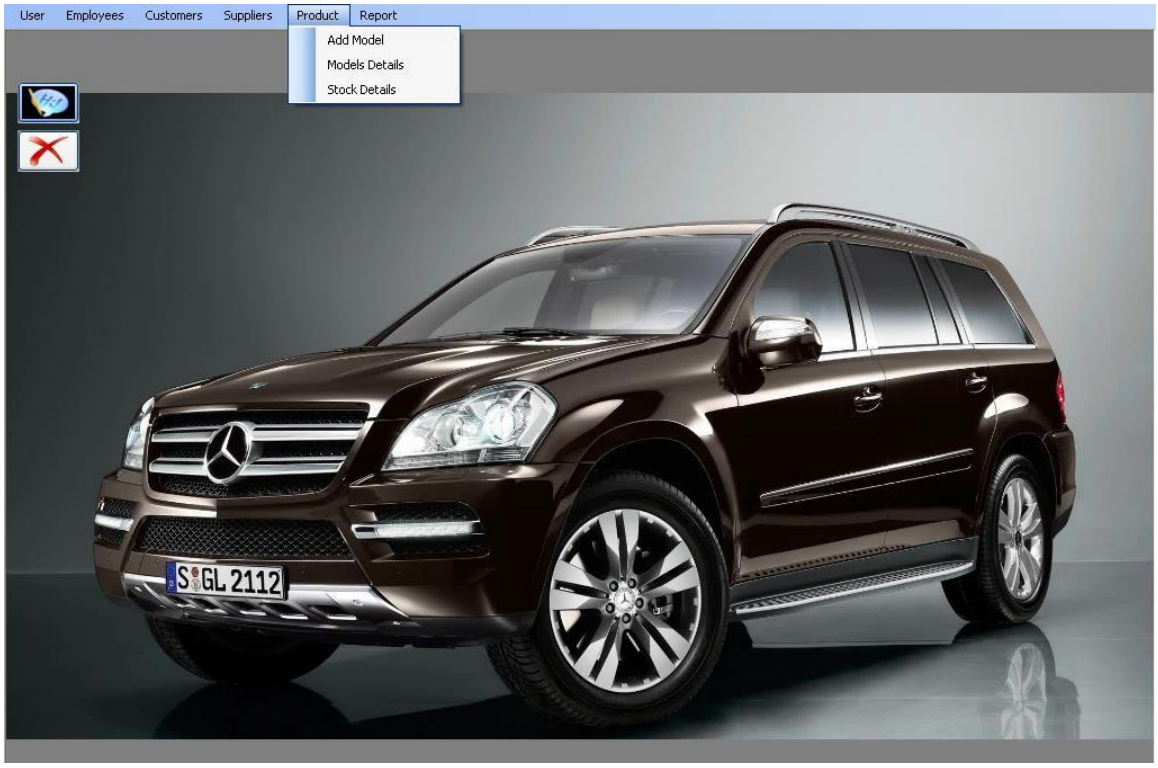
Suppliers Details Form



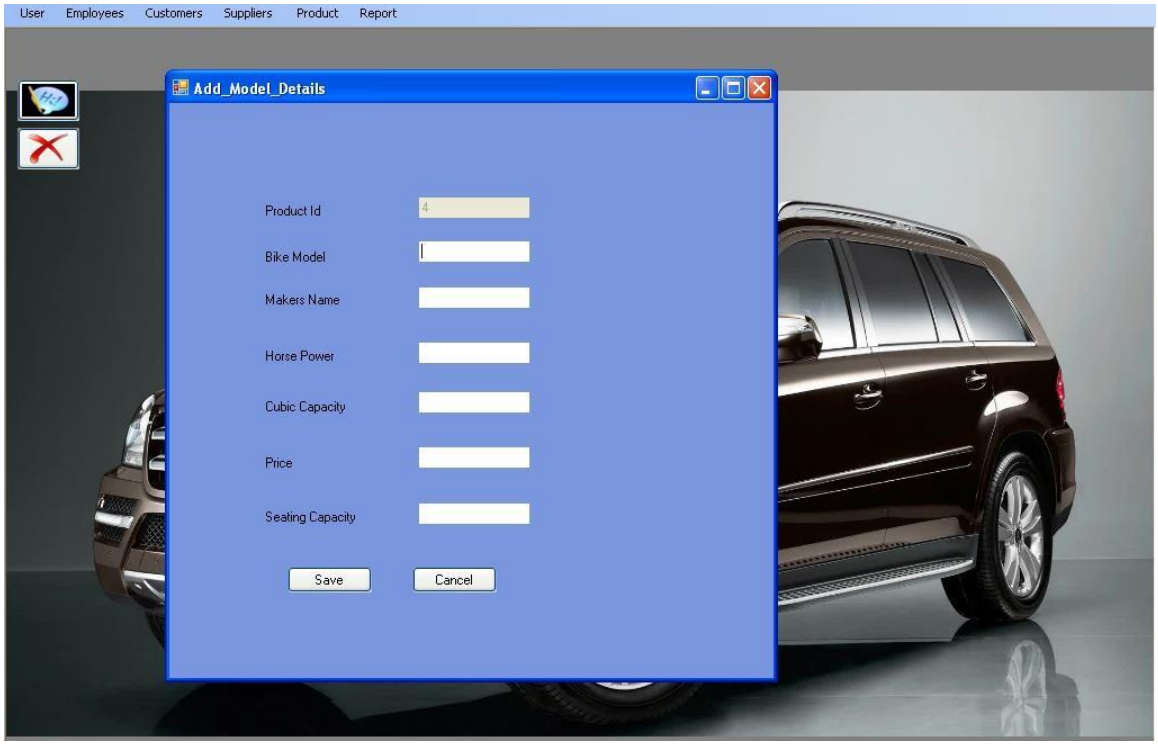
Purchase Order Detail Form



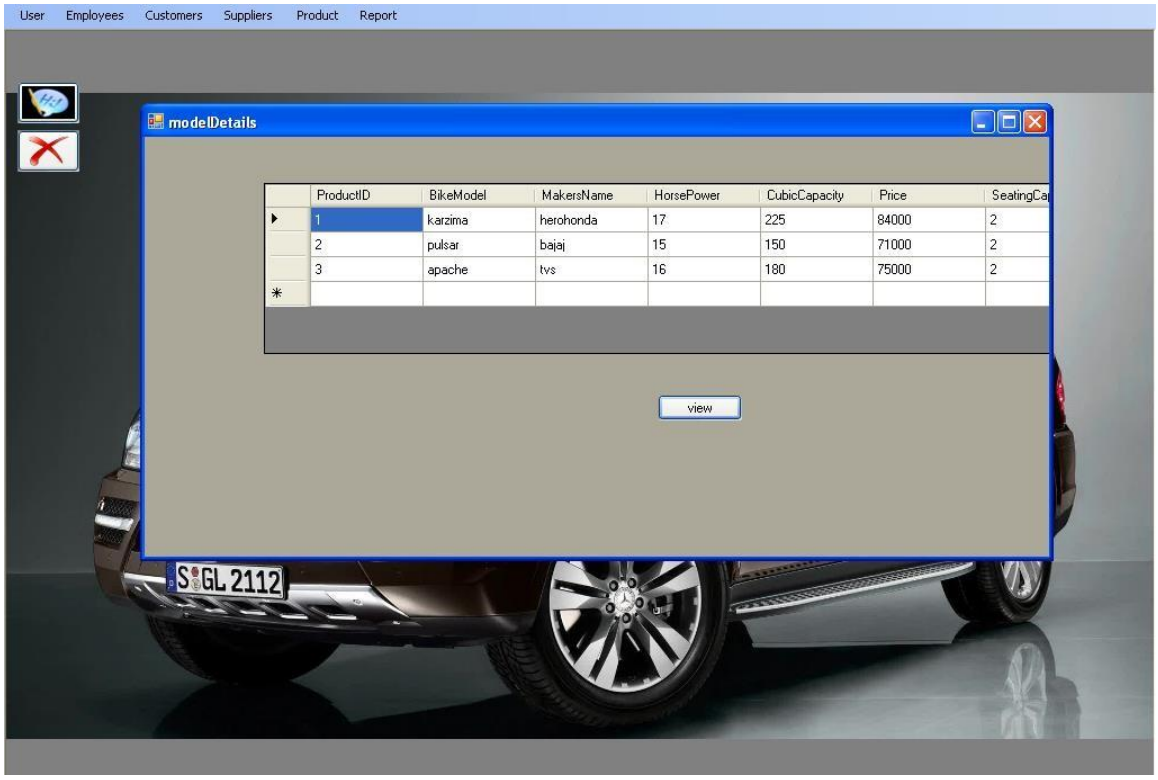
Different Sub menus included in Product



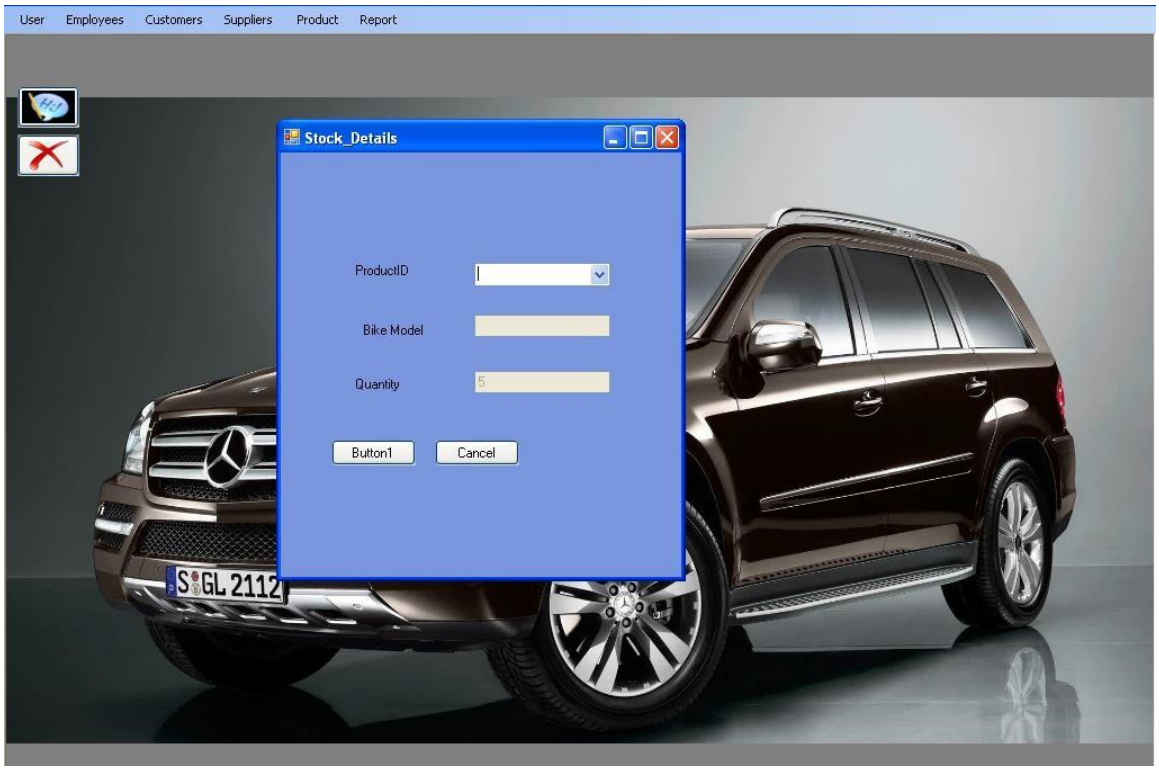
Add New Model Form



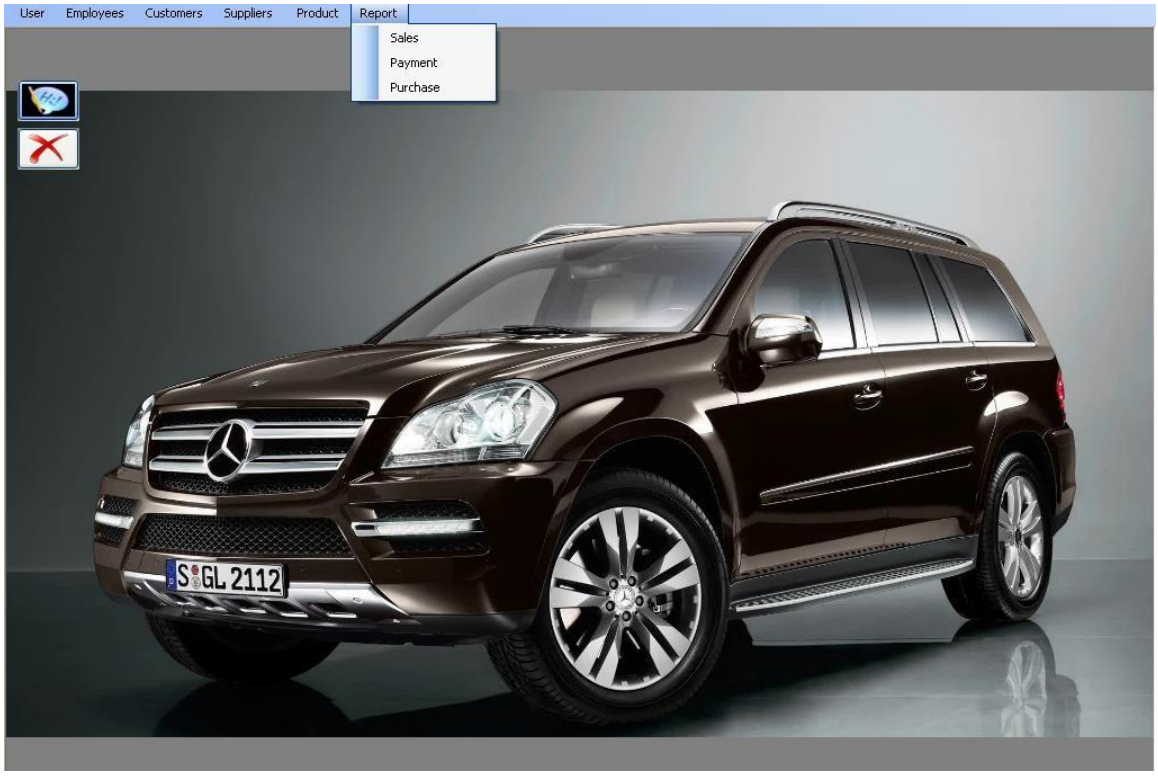
Model Details Form



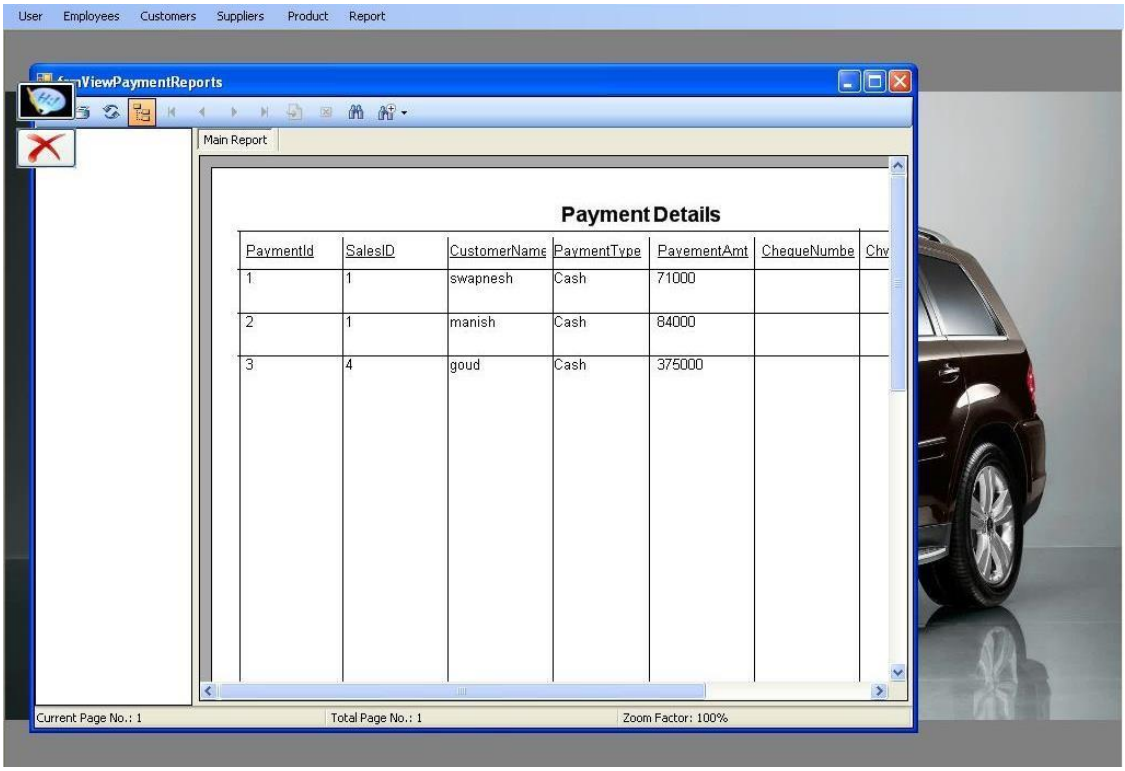
Stock Details Form



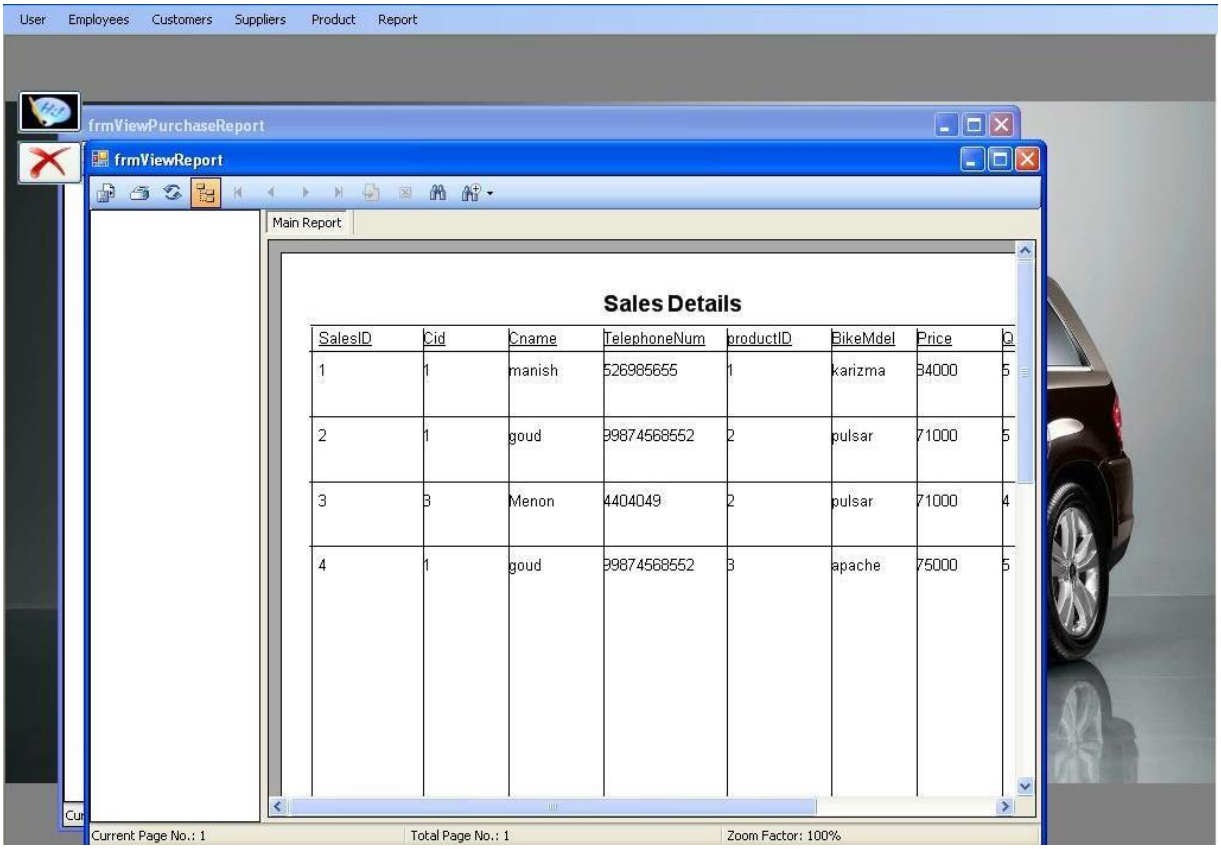
Different Sub menus included in Report



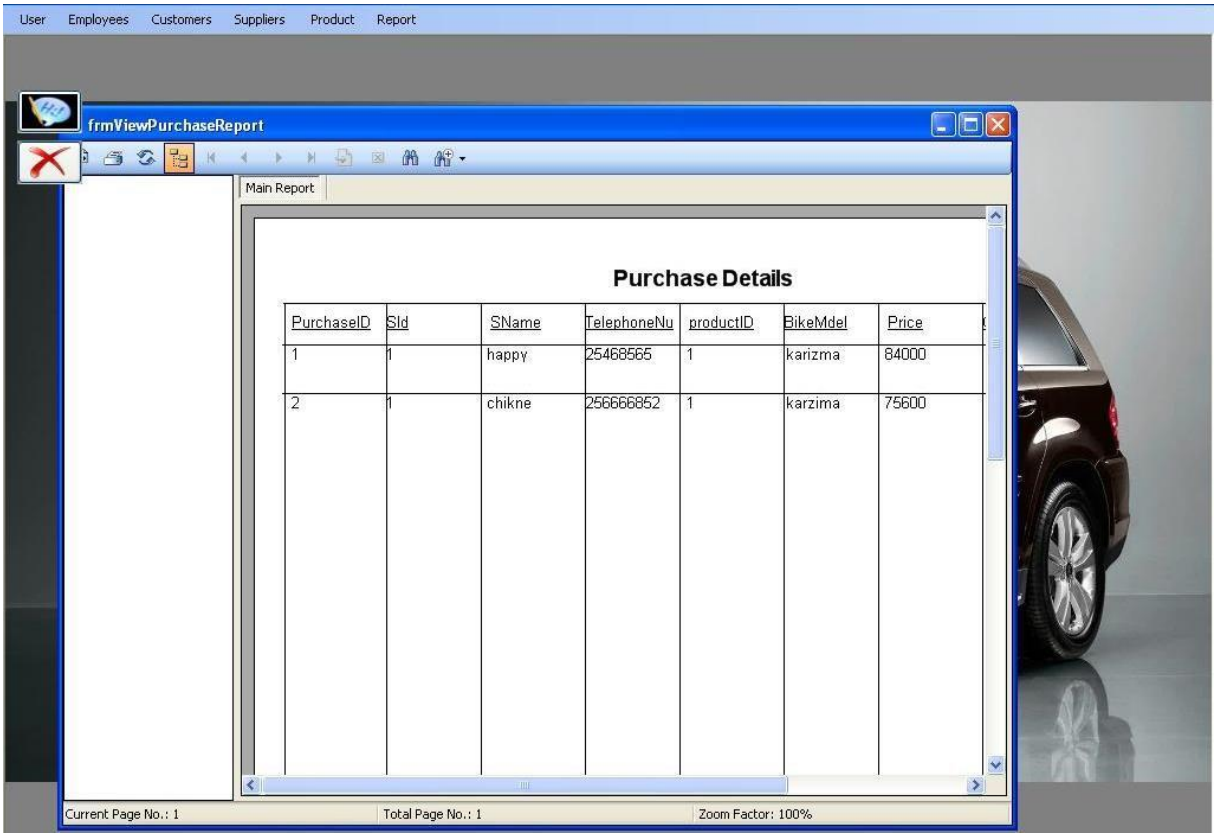
PAYMENT DETAILS



SALES DETAILS



PURCHES DETAILS



Coding of Login Form

```
Imports System.Data.SqlClient.SqlException


Imports System.Data.SqlClient Public
Class Login
   ' Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;Connect Timeout=30;User
Instance=True")
   ' Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;Connect Timeout=30;User
Instance=True;database=AUTOMOBILE")
   Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")

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

   End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
      Dim a, b As String
a = TextBox1.Text        b =
TextBox2.Text        Dim
flag As Integer = 0
      conn.Open()
      Dim cmd As New SqlCommand("select * from login", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
      While (rd.Read())
         If (a = rd(0).ToString.Trim() And b = rd(1).ToString.Trim()) Then
flag = 1              Exit While           Else              flag = 0           End If
      End While

      If (flag = 1) Then
         MsgBox("Login Successfully.", MsgBoxStyle.OkOnly, "Done")
         Me.Hide()
         MDIAMS.Show()
      Else
         MsgBox("User Name or Password may be wrong.", MsgBoxStyle.Critical, "Error")
      End If
conn.Close()

   End Sub

   Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
      TextBox1.Text = ""
      TextBox2.Text = ""
   End Sub
```

```
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
        End
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
conn.Open()
        Dim cmd As New SqlCommand("select * from login where Username='" & TextBox1.Text & "' and
Password='" & TextBox2.Text & "' ", conn)        Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
        While (rd.Read())
            Me.Hide()
            MDIAMS.Show()
        End While
    End Sub
End Class
```

Coding of Main Form

```
Imports System.Windows.Forms

Public Class MDIAMS

    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 ToolBarToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
'    Me.ToolStrip.Visible = Me.ToolBarToolStripMenuItem.Checked
    'End Sub

    'Private Sub StatusBarToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs)
'    Me.StatusStrip.Visible = Me.StatusBarToolStripMenuItem.Checked     '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 MenuStrip_ItemClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ToolStripItemClickedEventArgs) Handles MenuStrip.ItemClicked

    End Sub


    Private Sub NewToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles NewToolStripMenuItem.Click
        Add_New_User.MdiParent = Me
```

```
    Add_New_User.Show()
  End Sub

  Private Sub ChangePasswordToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ChangePasswordToolStripMenuItem.Click
    Change_Password.MdiParent = Me
    Change_Password.Show()
  End Sub

  Private Sub CloseAllToolStripMenuItem_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CloseAllToolStripMenuItem.Click
    For Each ChildForm As Form In Me.MdiChildren
      ChildForm.Close()
    Next
  End Sub

  Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ExitToolStripMenuItem.Click
    Global.System.Windows.Forms.Application.Exit()
  End Sub

  Private Sub UserToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles UserToolStripMenuItem.Click

  End Sub

  Private Sub NewEmployeeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NewEmployeeToolStripMenuItem.Click
    NewEmployee.MdiParent = Me
    NewEmployee.Show()
  End Sub

  Private Sub EmployeesToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

  End Sub

  Private Sub EmployeesDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles EmployeesDetailsToolStripMenuItem.Click
    EmployeeDetails.MdiParent = Me
    EmployeeDetails.Show()
  End Sub

  Private Sub NewCustomerToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NewCustomerToolStripMenuItem.Click
    NewCustomer.MdiParent = Me
    NewCustomer.Show()
  End Sub

  Private Sub CustomersDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles CustomersDetailsToolStripMenuItem.Click
        CustomerDetails.MdiParent = Me
        CustomerDetails.Show()
    End Sub

    Private Sub SalesInvoiceToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SalesInvoiceToolStripMenuItem.Click
        sales_order_Details.MdiParent = Me
sales_order_Details.Show()
    End Sub

    Private Sub NewPaymentToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NewPaymentToolStripMenuItem.Click
        NewPayment.MdiParent = Me
        NewPayment.Show()
    End Sub

    Private Sub PaymentDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PaymentDetailsToolStripMenuItem.Click
        PaymentDetails.MdiParent = Me
        PaymentDetails.Show()
    End Sub

    Private Sub RegistrationToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RegistrationToolStripMenuItem.Click

    End Sub

    Private Sub NewToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NewToolStripMenuItem1.Click
        NewRegistration.MdiParent = Me
        NewRegistration.Show()
    End Sub

    Private Sub RegistrationDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RegistrationDetailsToolStripMenuItem.Click
        RegistrationDetails.MdiParent = Me
        RegistrationDetails.Show()
    End Sub

    Private Sub NewSuppliersToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NewSuppliersToolStripMenuItem.Click
        NewSupplier.MdiParent = Me
        NewSupplier.Show()
    End Sub

    Private Sub SuppliersDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SuppliersDetailsToolStripMenuItem.Click
        Suppliesdetails.MdiParent = Me
        Suppliesdetails.Show()
```

```
    End Sub

    Private Sub PurchaseInvoiceToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PurchaseInvoiceToolStripMenuItem.Click
        Purchase_Order_Details.MdiParent = Me
        Purchase_Order_Details.Show()
    End Sub

    Private Sub AddModelToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AddModelToolStripMenuItem.Click
        Add_Model_Details.MdiParent = Me
        Add_Model_Details.Show()
    End Sub

    Private Sub ModelsDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ModelsDetailsToolStripMenuItem.Click
modelDetails.MdiParent = Me         modelDetails.Show()
    End Sub

    Private Sub StockDetailsToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles StockDetailsToolStripMenuItem.Click
        Stock_Details.MdiParent = Me
        Stock_Details.Show()
    End Sub

    Private Sub SalesToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
Handles SalesToolStripMenuItem.Click
frmViewReport.MdiParent = Me         frmViewReport.Show()
    End Sub

    Private Sub PaymentToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PaymentToolStripMenuItem.Click
frmViewPaymentReports.MdiParent = Me         frmViewPaymentReports.Show()
    End Sub

    Private Sub PurchaseToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PurchaseToolStripMenuItem.Click
frmViewPurchaseReport.MdiParent = Me         frmViewPurchaseReport.Show()
    End Sub

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

    End Sub
End Class
```

Coding of Adding New User

```
Imports System.Data.SqlClient
Public Class Add_New_User
    '               Dim             conn            As              New             SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim
conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        If ((TextBox1.Text = "") Or (TextBox2.Text = "") Or (TextBox3.Text = "")) Then
MsgBox("Details are missing", MsgBoxStyle.Exclamation)
        Else
            If (TextBox2.Text = TextBox3.Text) And (TextBox2.Text <> "") Then
conn.Open()
                Dim cmd As New SqlCommand("insert into login values('" & TextBox1.Text.ToString.Trim() & " ',' "
& TextBox2.Text.Trim() & " ')", conn)
cmd.ExecuteNonQuery()               conn.Close()
                MsgBox("New Account is Created Successfully", MsgBoxStyle.MsgBoxRight)
TextBox1.Text = ""
                TextBox2.Text = ""
                TextBox3.Text = ""
                Me.Close()
            Else
                MsgBox("Password Mismatch", MsgBoxStyle.Critical)
            End If
        End If
    End Sub

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

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
        Me.Close()
    End Sub
End Class
```

Coding of Change Password

```
Imports System.Data.SqlClient
Public Class Change_Password
   'Dim conn As New SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim
conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
   Private Sub Change_Password_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

   End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
      Dim a, b As String
a = TextBox1.Text
b = TextBox2.Text
conn.Open()
      Dim cmd1 As New SqlCommand("select * from login where UserName='" & a & "'", conn)
Dim rd As SqlDataReader        rd = cmd1.ExecuteReader()
      rd.Read()
      If (a = rd(0).ToString.Trim() And b = rd(1).ToString.Trim()) Then
conn.Close()
         If (TextBox4.Text = TextBox3.Text) And (TextBox3.Text <> "") Then
conn.Open()
            Dim s As String = "Delete from login where UserName = '" & TextBox1.Text & " ' "
            Dim s1 As String = "Insert into login values('" & TextBox1.Text & " ','" & TextBox3.Text & " ' )"
            Dim cmd As New SqlCommand(s, conn)
cmd.ExecuteNonQuery()
            Dim cmd2 As New SqlCommand(s1, conn)
            cmd2.ExecuteNonQuery()
            conn.Close()
            MsgBox("Password Changed Successfully", MsgBoxStyle.MsgBoxRight)
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox3.Text = ""
            TextBox4.Text = ""
            Me.Close()
         Else
            MsgBox("Password Mismatch", MsgBoxStyle.Critical)
         End If
      Else
         conn.Close()
         MsgBox("Entered Password is Wrong", MsgBoxStyle.Critical)
      End If
   End Sub

   Private Sub GroupBox2_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
```

```
GroupBox2.Enter

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
        Me.Close()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
    End Sub
End Class
```

Coding of New Employee Form

```
Imports System.Data.SqlClient
Public Class NewEmployee
   'Dim conn As New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\Database\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")     Dim id As Integer
   Private Sub NewEmployee_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
      conn.Open()
      Dim cmd As New SqlCommand("select EmpId from Emp", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
      While rd.Read()
id = rd(0)        End
While
conn.Close()
      id = id + 1
      TextBox9.Text = id
   End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
      If ((TextBox1.Text = "") Or (TextBox2.Text = "") Or (TextBox3.Text = "") Or (TextBox4.Text = "") Or (TextBox6.Text = "") Or (TextBox7.Text = "") Or (TextBox8.Text = "") Or (TextBox10.Text = "") Or (TextBox11.Text = "") Or (TextBox12.Text = "") Or (TextBox13.Text = "")) Then
         MsgBox("Details are Incomplete", MsgBoxStyle.Exclamation)
      Else
         conn.Open()
Dim g, d As String
         If (RadioButton1.Checked = True) Then
g = "Male"           Else
            g = "Female"
End If
         d = DateTimePicker1.Text
         Dim s As String = "Insert into Emp values('" & id & "','" & TextBox8.Text & "','" & TextBox1.Text & "','" & g & "','" & TextBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & d & "','" & TextBox6.Text & "','" & TextBox10.Text & "','" & TextBox11.Text & "','" & TextBox12.Text & "','" & TextBox7.Text & "','" & TextBox13.Text & " ' )"
         Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
         conn.Close()
         Me.Close()
      End If
   End Sub

   Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
      Me.Close()
```

```
    End Sub

    Private Sub TextBox8_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox8.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
          And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
           'Allowed space
           If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
           End If
        End If
        ' Allowed backspace
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub

    Private Sub TextBox4_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox4.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
          And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
           'Allowed space
           If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
           End If
        End If
        ' Allowed backspace
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub

    Private Sub TextBox3_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
e.Handled = True
        End If
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub

    Private Sub TextBox10_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox10.KeyPress
```

```
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
    Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
    e.Handled = True
        End If
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
    e.Handled = False
        End If
      End Sub

      Private Sub TextBox11_KeyPress(ByVal sender As System.Object, ByVal e As
    System.Windows.Forms.KeyPressEventArgs) Handles TextBox11.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
                Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
    e.Handled = True
        End If
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
    e.Handled = False
        End If
      End Sub
    End Class
```

Coding of Employee Details Form

```
Imports System.Data.SqlClient
Public Class EmployeeDetails

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        '           Dim str As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"
        Dim str As String = "Data Source=.;Integrated Security=True;database=AUTOMOBILE"
        Dim con As New SqlConnection(str)
        Dim com As String = "Select EmpId,
EmpName,EmpAge,Gender,Address,TelNum,City,Date,Email,Mobile,Pincode,Department,Designation,Sal
ary from Emp"
        Dim Adpt As New SqlDataAdapter(com, con)
        Dim ds As New DataSet()
        Adpt.Fill(ds, "Emp")
        DataGridView1.DataSource = ds.Tables(0)
    End Sub

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

    End Sub
End Class
```

Coding of Adding new Customer Form

```
Imports System.Data.SqlClient
Public Class NewCustomer
    '           Dim           conn           As           New           SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim
conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")     Dim id
As Integer

    Private Sub NewCustomer_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        conn.Open()
        Dim cmd As New SqlCommand("select CustomerId from Customer", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
        While rd.Read()
id = rd(0)        End
While
conn.Close()        id
= id + 1
        lblid.Text = id
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        If ((TextBox1.Text = "") Or (TextBox2.Text = "") Or (TextBox3.Text = "") Or (TextBox4.Text = "") Or
(TextBox5.Text = "") Or (TextBox6.Text = "") Or (TextBox8.Text = "") Or (TextBox9.Text = "")) Then
MsgBox("Details are Incomplete", MsgBoxStyle.Exclamation)
        Else
            conn.Open()
            Dim s As String = "Insert into Customer values('" & id & "','" & TextBox2.Text & "','" & TextBox1.Text
& "','" & TextBox8.Text & "','" & TextBox9.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" &
TextBox5.Text & "','" & TextBox6.Text & "' )"
Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
            conn.Close()
            Me.Close()
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Me.Close()
    End Sub


    Private Sub TextBox2_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
            Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
```

```
        And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
        Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
        'Allowed space
        If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
        End If
      End If
      ' Allowed backspace
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
  End Sub

  Private Sub TextBox1_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
      If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
        Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
        And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
        Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
         'Allowed space
         If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
         End If
      End If
      ' Allowed backspace
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
  End Sub

  Private Sub TextBox8_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox8.KeyPress
      If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
        Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
        And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
        Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
         'Allowed space
         If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
         End If
      End If
      ' Allowed backspace
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
  End Sub

  Private Sub TextBox3_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
      If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
```

```
        Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
        And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
        Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
         'Allowed space
         If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
         End If
      End If
      ' Allowed backspace
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
   End Sub

   Private Sub TextBox4_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox4.KeyPress
      If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
e.Handled = True
      End If
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
   End Sub

   Private Sub TextBox5_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox5.KeyPress
      If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
e.Handled = True
      End If
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
   End Sub

   Private Sub TextBox6_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox6.KeyPress




   End Sub
End Class
```

Coding of Customers Details From

```
Imports System.Data.SqlClient
Public Class CustomerDetails



    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        'Dim str As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"
        Dim str As String = "Data Source=.;Integrated Security=True;database=AUTOMOBILE"
        ' Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
Dim con As New SqlConnection(str)
        Dim com As String = "Select CustomerId,
Firstname,Middletname,Lastname,Address,City,PhoneNO,Mobile,Email from Customer"
        Dim Adpt As New SqlDataAdapter(com, con)
        Dim ds As New DataSet()
        Adpt.Fill(ds, "Customer")
        DataGridView1.DataSource = ds.Tables(0)
    End Sub

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

    End Sub
End Class
```

Coding of Sales Order Details Form

```
Imports System.Data.SqlClient
Public Class sales_order_Details
    '               Dim              conn              As              New              SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")

    Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")

    Dim id As Integer
    Private Sub sales_order_Details_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        conn.Open()
        Dim cmd As New SqlCommand("select SalesID from Sales", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
        While rd.Read()
id = rd(0)        End
While        id = id +
1
        TextBox7.Text = id
        rd.Close()

        Dim cmd1 As New SqlCommand("select ProductId from Product", conn)
        Dim rd1 As SqlDataReader

        rd1 = cmd1.ExecuteReader()
        While rd1.Read()
            ComboBox2.Items.Add(rd1(0))
        End While
rd1.Close()
        Dim cmd2 As New SqlCommand("select CustomerId from Customer", conn)
Dim rd2 As SqlDataReader        rd2 = cmd2.ExecuteReader()
        While rd2.Read()
            ComboBox1.Items.Add(rd2(0))
        End While
rd2.Close()
conn.Close()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
conn.Open() Dim s As
String = "Insert into Sales
values('" & id & "','" &
ComboBox1.Text & "','" &
TextBox1.Text &
"','" & TextBox2.Text & "','" & ComboBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" &
```

```
TextBox5.Text & "','" & TextBox6.Text & "' )"
Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
      conn.Close()
      Me.Close()
   End Sub

   Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
      Me.Close()
   End Sub

   Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
conn.Open()
      Dim cmd As New SqlCommand("select Lastname,PhoneNO from Customer where CustomerId='" &
ComboBox1.Text & "'", conn)        Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
      If rd.Read() Then
         TextBox1.Text = rd(0).ToString()
         TextBox2.Text = rd(1).ToString()
      End If
conn.Close()
   End Sub

   Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox2.SelectedIndexChanged
conn.Open()
      Dim cmd As New SqlCommand("select BikeModel,Price from Product where ProductID='" &
ComboBox2.Text & "'", conn)        Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
      If rd.Read() Then
         TextBox3.Text = rd(0).ToString()
         TextBox4.Text = rd(1).ToString()
      End If
conn.Close()
   End Sub

   Private Sub TextBox5_Leave(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TextBox5.Leave
      Dim amt, tamt, quan As Double
      Try
         amt  =  TextBox4.Text
quan     =     TextBox5.Text
tamt    =    amt    *    quan
TextBox6.Text = tamt
      Catch ex As Exception
         MsgBox("Please Enter the Quantity", MsgBoxStyle.Critical, "Error")
TextBox5.Focus()
      End Try
   End Sub
End ClASS
```

Coding of New Payment Form

```
Imports System.Data.SqlClient

Public Class NewPayment

    '       Dim       conn       As       New       SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim
conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")     Dim id
As Integer
    Private Sub NewPayment_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        conn.Open()
        Dim cmd As New SqlCommand("select PaymentId from Payment", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
        While rd.Read()
id = rd(0)        End
While
rd.Close()        id =
id + 1
        TextBox5.Text = id

        Dim cmd1 As New SqlCommand("select SalesID from Sales", conn)
Dim rd1 As SqlDataReader        rd1 = cmd1.ExecuteReader()
        While rd1.Read()
            ComboBox1.Items.Add(rd1(0))
        End While
rd1.Close()
conn.Close()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
conn.Open()
        Dim s As String = "Insert into Payment values('" & id & "','" & ComboBox1.Text & "','" & TextBox6.Text
& "','" & ComboBox2.Text & "','" & TextBox1.Text & "','" & TextBox2.Text & "','" & TextBox3.Text & "','" &
TextBox4.Text & "' )"
        Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
        conn.Close()
        Me.Close()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Me.Close()
    End Sub
```

```
  Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
conn.Open()
    Dim cmd As New SqlCommand("select Cname,TotalAmount from Sales where SalesID='" &
ComboBox1.Text & "'", conn)       Dim rd As SqlDataReader       rd = cmd.ExecuteReader()
    If rd.Read() Then
      TextBox6.Text = rd(0).ToString()
      TextBox1.Text = rd(1).ToString()
    End If
conn.Close()
  End Sub

  Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox2.SelectedIndexChanged
    If (ComboBox2.SelectedIndex = 1) Then
      TextBox2.Visible = True
      TextBox3.Visible = True
      TextBox4.Visible = True
      Label5.Visible = True
      Label6.Visible = True
      Label7.Visible = True
    Else
      TextBox2.Visible = False
      TextBox3.Visible = False
      TextBox4.Visible = False
      Label5.Visible = False
      Label6.Visible = False
      Label7.Visible = False

    End If
  End Sub

  Private Sub TextBox6_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox6.KeyPress
    If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
      Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
      And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
      Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
       'Allowed space
       If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
       End If
    End If
    ' Allowed backspace
    If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
    End If
  End Sub

  Private Sub TextBox3_KeyPress(ByVal sender As System.Object, ByVal e As
```

```
System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
          And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
           'Allowed space
           If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
           End If
        End If
        ' Allowed backspace
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub


    Private Sub TextBox2_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
e.Handled = True
        End If
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub
End Class
```

Coding of Payment Details Form

```
Imports System.Data.SqlClient
Public Class PaymentDetails

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        'Dim str As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"
        'Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
        Dim str As String = "Data Source=.;Integrated Security=True;database=AUTOMOBILE"
        Dim con As New SqlConnection(str)
        Dim com As String = "Select PaymentId,
SalesID,CustomerName,PaymentType,PayementAmt,ChequeNumber,ChwqueStatus,BankName from
Payment"
        Dim Adpt As New SqlDataAdapter(com, con)
        Dim ds As New DataSet()
        Adpt.Fill(ds, "Payment")
        DataGridView1.DataSource = ds.Tables(0)
    End Sub

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

    End Sub
End Class
```

Coding of Registration Form

```
Imports System.Data.SqlClient
Public Class NewRegistration
   'Dim conn As New SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim
conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")     Dim id
As Integer

   Private Sub NewRegistration_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
      conn.Open()
      Dim cmd As New SqlCommand("select RegistrationNumber from Registration", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
      While rd.Read()
id = rd(0)        End
While
      id = id + 1
      TextBox1.Text = id
rd.Close()
      Dim cmd1 As New SqlCommand("select ProductId from Product", conn)
Dim rd1 As SqlDataReader        rd1 = cmd1.ExecuteReader()
      While rd1.Read()
         ComboBox2.Items.Add(rd1(0))
      End While
rd1.Close()
      Dim cmd2 As New SqlCommand("select Lastname from Customer", conn)
Dim rd2 As SqlDataReader        rd2 = cmd2.ExecuteReader()
      While rd2.Read()
         ComboBox1.Items.Add(rd2(0))
      End While
rd2.Close()
      conn.Close()

   End Sub

   Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
conn.Open()
      Dim cmd As New SqlCommand("select Address from Customer where Lastname='" &
ComboBox1.Text
& "'", conn)
      Dim rd As SqlDataReader
rd = cmd.ExecuteReader()
      If rd.Read() Then
         TextBox10.Text = rd(0).ToString()
      End If
conn.Close()
   End Sub
```

```
  Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox2.SelectedIndexChanged
conn.Open()
    Dim cmd As New SqlCommand("select BikeModel from Product where ProductID='" &
ComboBox2.Text & "'", conn)        Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
    If rd.Read() Then
      TextBox7.Text = rd(0).ToString()
    End If
conn.Close()
  End Sub

  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
conn.Open()
    Dim s As String = "Insert into Registration values('" & id & "','" & DateTimePicker1.Text & "','" &
TextBox3.Text & "','" & ComboBox1.Text & "','" & TextBox5.Text & "','" & TextBox10.Text & "','" &
ComboBox2.Text & "','" & TextBox7.Text & "','" & TextBox8.Text & "', '" & TextBox9.Text & " ' )"
    Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
    conn.Close()
    Me.Close()
  End Sub

  Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
    Me.Close()
  End Sub

  Private Sub TextBox8_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox8.KeyPress
    If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
      Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
      And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
      Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
       'Allowed space
       If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
       End If
    End If
    ' Allowed backspace
    If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
    End If
  End Sub

  Private Sub TextBox9_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox9.KeyPress
```

```
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
    Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
    e.Handled = True
        End If
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
    e.Handled = False
        End If
      End Sub
    End Class
```

Coding of Registration Details Form

```
Imports System.Data.SqlClient
Public Class RegistrationDetails

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        'Dim str As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"
        'Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
        Dim str As String = "Data Source=.;Integrated Security=True;database=AUTOMOBILE"
        Dim con As New SqlConnection(str)
        Dim com As String = "Select RegistrationNumber,
RegistrationDate,RegistrationAuthority,RegistrationOwnerName,Age,Address,ProductId,BikeModel,Color,
ChasisNumber from Registration"
        Dim Adpt As New SqlDataAdapter(com, con)
        Dim ds As New DataSet()
        Adpt.Fill(ds, "Registration")
        DataGridView1.DataSource = ds.Tables(0)
    End Sub

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

    End Sub
End Class
```

Coding of Supplier Form

```
 Imports System.Data.SqlClient Public Class NewSupplier
   'Dim conn As New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\Database\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")     Dim id As Integer
   Private Sub NewSupplier_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
       conn.Open()
       Dim cmd As New SqlCommand("select SupplierId from Supplier", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
       While rd.Read()
id = rd(0)        End
While
conn.Close()
       id = id + 1
       TextBox7.Text = id
   End Sub

   Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
conn.Open()
       Dim s As String = "Insert into Supplier values('" & id & "','" & TextBox2.Text & "','" & TextBox1.Text & "','" & TextBox8.Text & "','" & TextBox9.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "' )"
Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
       conn.Close()
       Me.Close()
   End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
       Me.Close()
   End Sub


   Private Sub TextBox2_KeyPress(ByVal sender As System.Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
       If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
         Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
         And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
         Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
          'Allowed space
          If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
          End If
       End If
```

```
        ' Allowed backspace
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub


    Private Sub TextBox1_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
          And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
           'Allowed space
           If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
           End If
        End If
        ' Allowed backspace
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub

    Private Sub TextBox8_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox8.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
          And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
           'Allowed space
           If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
           End If
        End If
        ' Allowed backspace
        If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
        End If
    End Sub

    Private Sub TextBox3_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress
        If (Microsoft.VisualBasic.Asc(e.KeyChar) < 65) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 90) _
          And (Microsoft.VisualBasic.Asc(e.KeyChar) < 97) _
          Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 122) Then
           'Allowed space
           If (Microsoft.VisualBasic.Asc(e.KeyChar) <> 32) Then
e.Handled = True
```

```
        End If
      End If
      ' Allowed backspace
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
   End Sub

   Private Sub TextBox4_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox4.KeyPress
      If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
e.Handled = True
      End If
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
   End Sub

   Private Sub TextBox5_KeyPress(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox5.KeyPress
      If (Microsoft.VisualBasic.Asc(e.KeyChar) < 48) _
Or (Microsoft.VisualBasic.Asc(e.KeyChar) > 57) Then
e.Handled = True
      End If
      If (Microsoft.VisualBasic.Asc(e.KeyChar) = 8) Then
e.Handled = False
      End If
   End Sub
End Class
```

Coding of Suppliers Details Form

```
Imports System.Data.SqlClient
Public Class Suppliesdetails

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        'Dim str As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"
        ' Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
        Dim str As String = "Data Source=.;Integrated Security=True;database=AUTOMOBILE"
        Dim con As New SqlConnection(str)
        Dim com As String = "Select SupplierId,
SFname,SMname,SLname,Address,City,PhoneNum,MobileNum,Email from Supplier"
        Dim Adpt As New SqlDataAdapter(com, con)
        Dim ds As New DataSet()
        Adpt.Fill(ds, "Supplier")
        DataGridView1.DataSource = ds.Tables(0)
    End Sub

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

    End Sub
End Class
```

Coding of Purchase Order Detail Form

```
Imports System.Data.SqlClient
Public Class Purchase_Order_Details
   'Dim conn As New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\Database\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")    Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")

   Dim id As Integer
   Private Sub Purchase_Order_Details_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
      conn.Open()
      Dim cmd As New SqlCommand("select PurchaseID from Purchase", conn)
Dim rd As SqlDataReader       rd = cmd.ExecuteReader()
      While rd.Read()
id = rd(0)       End
While
      id = id + 1
      TextBox7.Text = id
rd.Close()
      Dim cmd1 As New SqlCommand("select ProductId from Product", conn)
Dim rd1 As SqlDataReader       rd1 = cmd1.ExecuteReader()
      While rd1.Read()
         ComboBox2.Items.Add(rd1(0))
      End While
      rd1.Close()

      Dim cmd2 As New SqlCommand("select SupplierId from Supplier", conn)
Dim rd2 As SqlDataReader       rd2 = cmd2.ExecuteReader()
      While rd2.Read()
         ComboBox1.Items.Add(rd2(0))
      End While
rd2.Close()
conn.Close()
   End Sub

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
conn.Open()
      Dim s As String = "Insert into Purchase values('" & id & "','" & ComboBox1.Text & "','" & TextBox1.Text & "','" & TextBox2.Text & "','" & ComboBox2.Text & "','" & TextBox3.Text & "','" & TextBox4.Text & "','" & TextBox5.Text & "','" & TextBox6.Text & "' )"
Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
      conn.Close()
Me.Close()    End Sub

   Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
```

```
    Me.Close()
  End Sub

  Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
conn.Open()
    Dim cmd As New SqlCommand("select SLname,PhoneNum from Supplier where SupplierId='" &
ComboBox1.Text & "'", conn)       Dim rd As SqlDataReader       rd = cmd.ExecuteReader()
    If rd.Read() Then
      TextBox1.Text = rd(0).ToString()
      TextBox2.Text = rd(1).ToString()
    End If
conn.Close()
  End Sub

  Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox2.SelectedIndexChanged
    Dim price As Double
conn.Open()
    Dim cmd As New SqlCommand("select BikeModel,Price from Product where ProductID='" &
ComboBox2.Text & "'", conn)       Dim rd As SqlDataReader       rd = cmd.ExecuteReader()
    If rd.Read() Then
      TextBox3.Text = rd(0).ToString()
price = Convert.ToDouble(rd(1))
    End If
    TextBox4.Text = price - (price * 0.1)
    conn.Close()
  End Sub

  Private Sub TextBox5_Leave(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
TextBox5.Leave
    Dim amt, tamt, quan As Double
    Try
      amt  =  TextBox4.Text
quan     =     TextBox5.Text
tamt    =    amt    *    quan
TextBox6.Text = tamt
    Catch ex As Exception
      MsgBox("Please Enter valid the Quantity", MsgBoxStyle.Critical, "Error")
TextBox5.Focus()
    End Try
  End Sub
End ClASS
```

Coding of Add New Model Form

```
Imports System.Data.SqlClient
Public Class Add_Model_Details
    '           Dim           conn           As           New           SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim
conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")     Dim id
As Integer
    Private Sub Add_Model_Details_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        conn.Open()
        Dim cmd As New SqlCommand("select ProductId from Product", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
        While rd.Read()
id = rd(0)        End
While
conn.Close()
        id = id + 1
        TextBox1.Text = id
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        If ((TextBox3.Text = "") Or (TextBox4.Text = "") Or (TextBox5.Text = "") Or (TextBox6.Text = "") Or
(TextBox7.Text = "") Or (TextBox8.Text = "")) Then
            MsgBox("Details are Incomplete", MsgBoxStyle.Exclamation)
        Else
            conn.Open()
            Dim s As String = "Insert into Product values('" & id & "','" & TextBox3.Text & "','" & TextBox4.Text &
"','" & TextBox5.Text & "','" & TextBox6.Text & "','" & TextBox7.Text & "','" & TextBox8.Text & "' )"
            Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
            conn.Close()
            Me.Close()
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Me.Close()
    End Sub
    Private Sub TextBox4_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TextBox4.TextChanged
    End Sub

    Private      Sub      TextBox4_KeyPress(ByVal      sender      As      System.Object,      ByVal      e      As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox4.KeyPress

    End Sub End
Class
```

Coding of Model Details Form

```
Imports System.Data.SqlClient
Public Class modelDetails

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        'Dim str As String = "Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"
        '  Dim conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
        Dim str As String = "Data Source=.;Integrated Security=True;database=AUTOMOBILE"

        Dim con As New SqlConnection(str)
        Dim com As String = "Select ProductID,
BikeModel,MakersName,HorsePower,CubicCapacity,Price,SeatingCapacity from Product"
        Dim Adpt As New SqlDataAdapter(com, con)
        Dim ds As New DataSet()
        Adpt.Fill(ds, "Product")
        DataGridView1.DataSource = ds.Tables(0)
    End Sub

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

    End Sub
End Class
```

Coding of Stock Details Form

```
Imports System.Data.SqlClient

Public Class Stock_Details
    '                Dim                conn                As                New                SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\siddhi\correctedones23rdmarch\Automobile\Automobile\D
atabase\automobile.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")     Dim
conn As New SqlConnection("Data Source=.;Integrated Security=True;database=AUTOMOBILE")
    Private Sub Stock_Details_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        conn.Open()
TextBox1.Text = "8"
        Dim cmd As New SqlCommand("select ProductId from Product", conn)
Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
        While rd.Read()
            ComboBox1.Items.Add(rd(0))
End While        conn.Close()     End
Sub
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
conn.Open()
        Dim cmd As New SqlCommand("select BikeModel from Product where ProductID='" &
ComboBox1.Text & "'", conn)        Dim rd As SqlDataReader        rd = cmd.ExecuteReader()
        If rd.Read() Then
            TextBox2.Text = rd(0).ToString()
        End If
conn.Close()     End
Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Me.Close()
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
conn.Open()
        Dim s As String = "Insert into Stock values('" & ComboBox1.Text & "','" & TextBox2.Text & "','" &
TextBox1.Text & "' )"
        Dim cmd1 As New SqlCommand(s, conn)
cmd1.ExecuteNonQuery()
        conn.Close()
        Me.Close()
    End Sub
    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TextBox1.TextChanged
        TextBox1.Text = "5"
    End Sub
End Class
```

VALIDATION

While creating the software it was assured that validations are proper for each and every field so that it will help the user to fill in the proper input in the field so that he will not make a mistake. Some validations, which are set, are as follows:

- If the user enters any wrong input, a particular field will not allow that input or it will flash an error.
- If user enters higher value in weight or calf, then the text will be cleared and the user will be asked to enter the value.

Process methodology adopted for development

Process methodology adopted for development:

- Before starting the coding for developing the system, analysis of the entire module was done.
- The various diagrams required for smooth functioning of the entire module were done.
- Also, various sketches of how the project would look like were prepared in the design phase.
- Then, those diagrams and sketches were shown to the project leader. Once he approved those layouts, then the implementation phase was started.
- Once the module was coded, though testing was done and bugs were removed.

SYSTEM MAINTENANCE

Standards and guidelines for maintaining the system
Use of the free software enables easy maintenance of the product any person having a minimal knowledge of the vb.net and msAccess can maintain it

In case of the any error occurring during the use of the software the source code of the error page contain track back during the program execution the exact lins causing problems are displayedwhich gives the user the possible idea of the error the corectiveness action can be taken

If the error is due to particular vb codes that error can be commented out
And checked out properly

If the form will have syntax error it will not execute the fprm leasing to the correctness

The provided documentation can lead to better maintence and lead and future enhancements with the software

SOFTWARE TESTING

TESTING

Software testing has a dual function; it is used to establish the presence of defects in program and it is used to help judge whether or not the program is usable in practice. The software testing is used for validation and verification, which ensures that software, conforms to its specification and meets the need of the customer. Software is a critical element of software quality assurance and represents the ultimate review of specification, designs and code generation. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to the customer.

TESTING METHODS

Unit testing:

Unit testing focuses verification effort on the smallest unit of software design the software component or module. In this type of testing the individual modules are tested and verify whether the accurate output is available or not. It can be done in two ways bottom-up or top- down. In bottom-up approach the last module is tested first and then moving upwards towards the first module. Top-down integration testing is an incremental approach to construction of program structure. Modules are integrated by moving down ward through the control hierarchy, beginning with the main control module.

Integration Testing:

When the unit testing is over, all the modules are integrated and tested as a whole. It might be possible that all modules may work individually, but they may not work when we put them together. Data can be lost across the interface, one module can have adverse affect on other or sub functions of another, when combined may not produce desired major function, individually acceptable imprecision may be magnified to unacceptable level; global data structure can present problem. So any system has to be tested this way so that the final output is the desired one.

Validation Testing:

After the integration testing software is completely assembled as a package, interfacing error have been uncovered and corrected, and then validation testing may begin. Validation can be defined in many ways but a simple definition is what a validation succeeds when software functions in a manner that can be reasonably accepted by the company.

System testing:

Any software is only one element of a larger computer based system. Ultimately software is incorporated with other system elements like hardware, people, information and a series of system integration and validation tests are conducted. System testing is actually a series of different test whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

Storage Testing:

The database of the system has to be stored on the database server. So the storage capacity of the database server should be enough to store all the data required for the efficient running of the system.

Maintenance

8 Implementation and Maintenance
8.1 Implementation phase:

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 MAINTENANCE

The maintenance of software is the time period in which the software is software product performs useful work. Maintenance activities involve making enhancement activities to the, adapting product to new environment and correcting problems. Software enhancement may involve providing new functional capabilities, improving user displays and modes of interaction.

Adaptation of software to a new environment may involve moving the software to a different machine. Problem correction involves modification and revalidation of software to correct errors. The four types of maintenance activities are described below:

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.
Corrective maintenance is the most commonly used maintenance approach, but it is easy to see its limitations. When equipment fails, it often leads to downtime in production, and sometime it causes

spreading of damage to other parts. In most cases this is costly business. Also, if the equipment needs to be replaced, the cost of replacing it alone can be substantial. Reliability of systems maintained by this type of maintenance is not known and can not be measured. Therefore, corrective maintenance is carried out on all items where the consequences of failure or wearing out are not significant (less important items) and the cost of this maintenance is not greater than preventive maintenance.

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 requests such as new Capabilities, modifications to existing functions and general enhancements.

Preventive Maintenance:

Preventive maintenance occurs when software is changed to improve future maintainability or to provide a better basic for future enhancements.

CONCLUSION

An attempt is made in all its earnest towards the successful completion of the project. This system was verified with valid as well as with invalid data.

This system is user friendly since it has been developed in Visual Studio 2005, a successful GUI environment. Since the connection can be extended to any database. The control will be more powerful.

Connecting it to any type of database extends the development control. Any suggestions for future development of the system are welcome.

Upgrading the system can be done without affecting the proper functioning of system.

FUTURE SCOPE

The future scope of this project is that by a little modification in it we can easily convert this project into various useful projects they are as follows:

WEB ENABLE PROJECT: the web-enable project means that the online users can access this project. This is a very use full facility provided because it will be convenient for the users to work on line.

MEMBERSHIP: We can add the membership.

Bibliography

Books:

- The Complete Reference Visual Basic.NET
- Murach's Beginning Visual Basic.NET
- Google