

**SYNOPSIS**

**Title: COMPUTER WAREHOUSE**

**OBJECTIVE:**

Computer Warehouse management project in VB.Net is a windows application specially designed to automate quotation printing and billing process in Computer shops. We designed this software in VB programming language. The objective and scope of Project Computer Warehouse Management System is to record the details of various activities of user. It will simplify the task and reduce the paper work.

The primary purpose of a Computer Warehouse Management System is to control the movement and storage of materials within a computer warehouse – you might even describe it as the legs at the end-of-the line, which automates the store. Automated data collection in the Computer warehouse management system would reduce the cost in the labor and increases the accuracy of the data. It increases the effectiveness of the service provided to the customer by reducing the cycle time. Inventory reduction and increased storage capacity are less likely.

In this project, we are using VB.NET 2008 to mainly design the front end of the project and the back end is implemented on MS access. MS access software is used to design back end mainly because we can create relational table employee database system easily using MS access and VB.NET software, which will provide a good graphical user interface to the user of the system.

**List of Modules:**

**Quotation Details:**

Warehouse is a place where materials are stocked. Warehouse deals with wide range of computer spares like monitor, keyboard, processor, drivers, monitors, add-on cards and multimedia products etc. For that, they need several materials and components. As the first step, the customer ask the quotation in computer shop.

**Stock details:**

The user may specify the stock to intend his necessary details of it and can select the required order.

**Customer order details:**

The warehouse department keys-in purchase order and if the stocks are available for the purchaser order, the warehouse department sends the goods to the customer along with the delivery note

**Billing details:**

It is necessary to give the bill for customers that has been purchased from warehouse and to insist the customer to the quality of the computer scarce.

.

**REQUIREMENTS SPECIFICATION**

**Hardware Requirements:**

* Processor : 32 BIT, Pentium-4 or higher
* RAM : 512 MB RAM or higher
* Hard disk : 20 GB (Minimum)
* Monitor : 1024x768(Resolution)
* Keyboard
* Mouse
* Printer

**Software requirements:**

* Operating system : Windows 2000/xp
* Front end : VB.NET 2008
* Back end : SQL server 2008

*Introduction*

**INTRODUCTION**

**Introduction:**

Computerization is basically concerned with the automation of task using computer-based information system. This process with the people involved in doing a certain task, which being replaced by particular machineries. Computerization would also mean less effort exerted, sine manual could be done in shorter span of time.In any organization, computer is a necessity. Computerization to an organizational establishment is important in order to do task easily and efficiently.

A Computer warehouse management system is a software application that supports the day-to-day operations in a computer shop. Computer Warehouse Management programs enable centralized management of tasks such as tracking inventory levels and billing. Computer Warehouse Management is suitable for small computer shops, this system has three main transactions editing items, receiving and selling items. A Computer warehouse managementis a key part of the supply chain and primarily aims to control the movement and storage of materials within a computer warehouse and process the associated transactions.

*Objective*

**OBJECTIVE:**

The main objective of this project is to computerize the Computer Warehouse Management System. Warehouse is a place where materials are stocked. Warehouse deals with wide range of computer spares like monitors, keyboard, processor, drivers, monitors, add-on cards and multimedia products etc. For that, they need several materials and components. As the first step, the Customer asks the quotation in computer shop. If all the terms and conditions are satisfied, it sends the purchase order to the Warehouse shop. The Warehouse department keys-in the purchase order and if the stocks are available for the purchase order, the warehouse department sends the goods to the Customer along with a delivery note

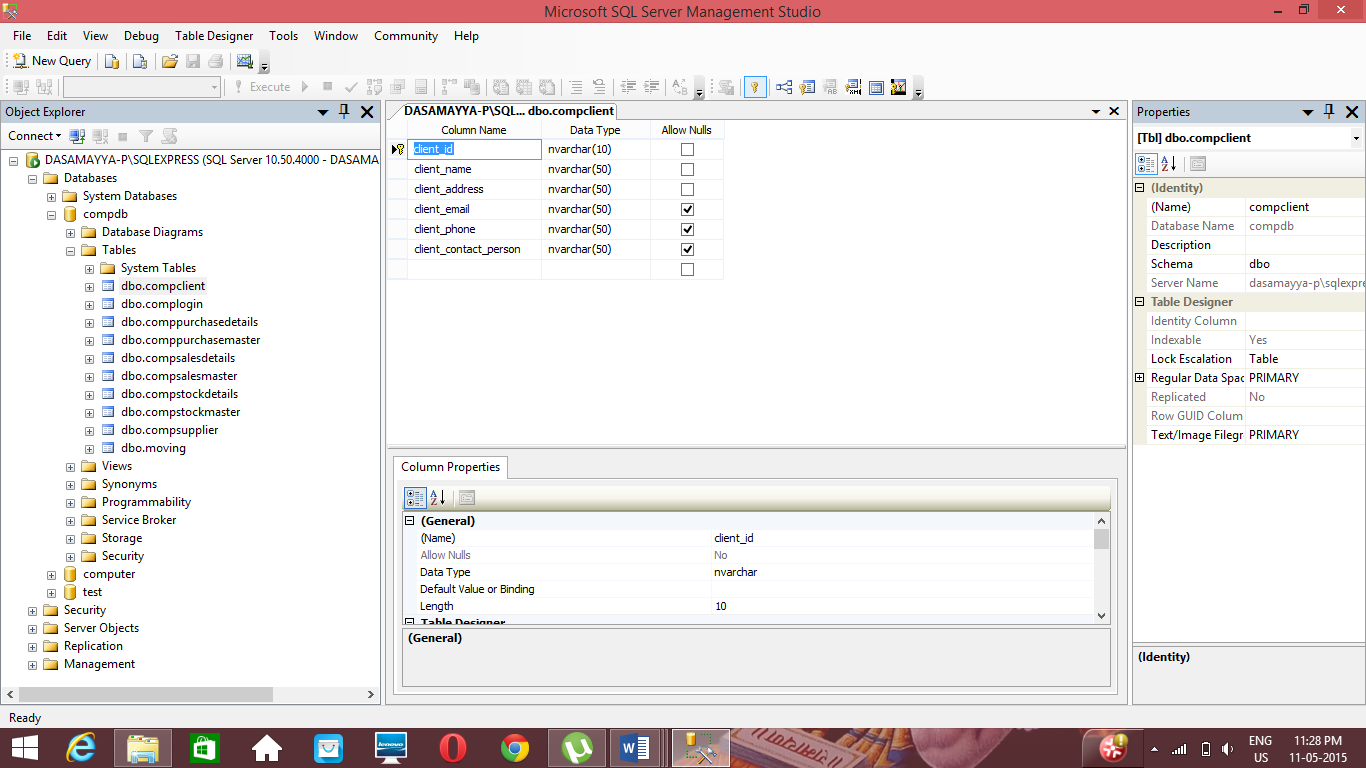
Our project helps the user to prepare the quotation of computer shop. Our application possess automated bill system through which manual errors can be avoided. User can maintain computer shop stock details easily through our application. User can generate the reports of sales and quotation easily to know the status of their business. It is designed specifically to maintain computer warehouse details, stock details, customer order details and billing details. Our computer warehouse management software provides detailed reports and summaries of daily updating information of warehouse. It helps you to keep track the status of computer warehouse activities. Another important facility we have given is monitoring the daily transactions facility. Owner of the computer warehouse could use this facility to check present and past day’s details. Through the help of this project owner can see the daily transaction of computer warehouse in his fingertip.

Computer Warehouse Management can help you manage goods and space more effectively, reduce costs and waste, and gain control over warehouse operations. With access to real-time, accurate inventory data, your warehouse professionals save time locating items or performing physical inventories, sales representatives can keep track on stock availability, and buyers can maintain optimum stock levels while minimizing carrying costs

*Data base design*

**DATA CUSTOMIZATION**

**Clint:**



CREATETABLE [dbo].[compclient]

(

[client\_id] [nvarchar](10)NOTNULL,

[client\_name] [nvarchar](50)NOTNULL,

[client\_address] [nvarchar](50)NOTNULL,

[client\_email] [nvarchar](50)NULL,

[client\_phone] [nvarchar](50)NULL,

[client\_contact\_person] [nvarchar](50)NULL,

CONSTRAINT [PK\_compclient] PRIMARYKEYCLUSTERED

(

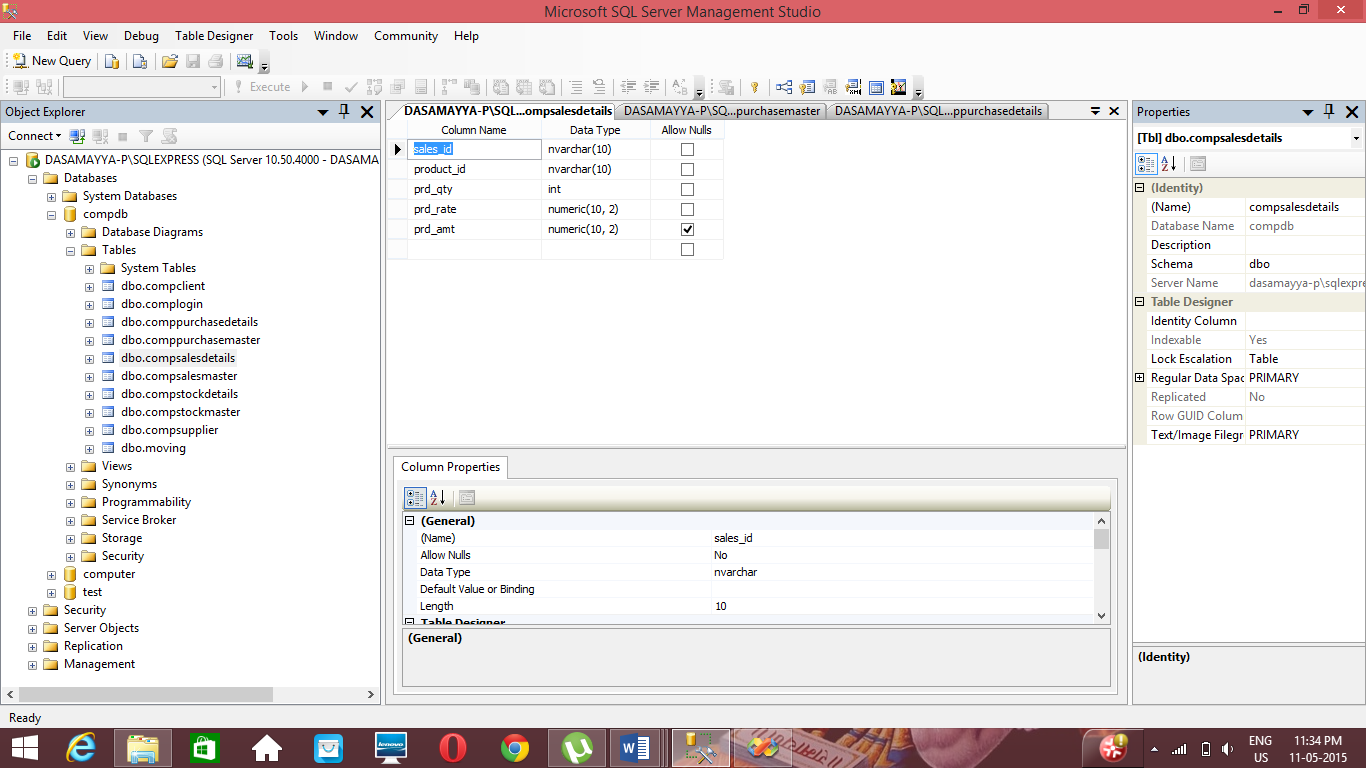
[client\_id] ASC

)WITH (PAD\_INDEX=OFF,STATISTICS\_NORECOMPUTE=OFF,IGNORE\_DUP\_KEY=OFF,ALLOW\_ROW\_LOCKS=ON,ALLOW\_PAGE\_LOCKS=ON)ON [PRIMARY]

)ON [PRIMARY]

GO

Quotation:



CREATETABLE [dbo].[compsalesdetails]

(

[sales\_id] [nvarchar](10)NOTNULL,

[product\_id] [nvarchar](10)NOTNULL,

[prd\_qty] [int] NOTNULL,

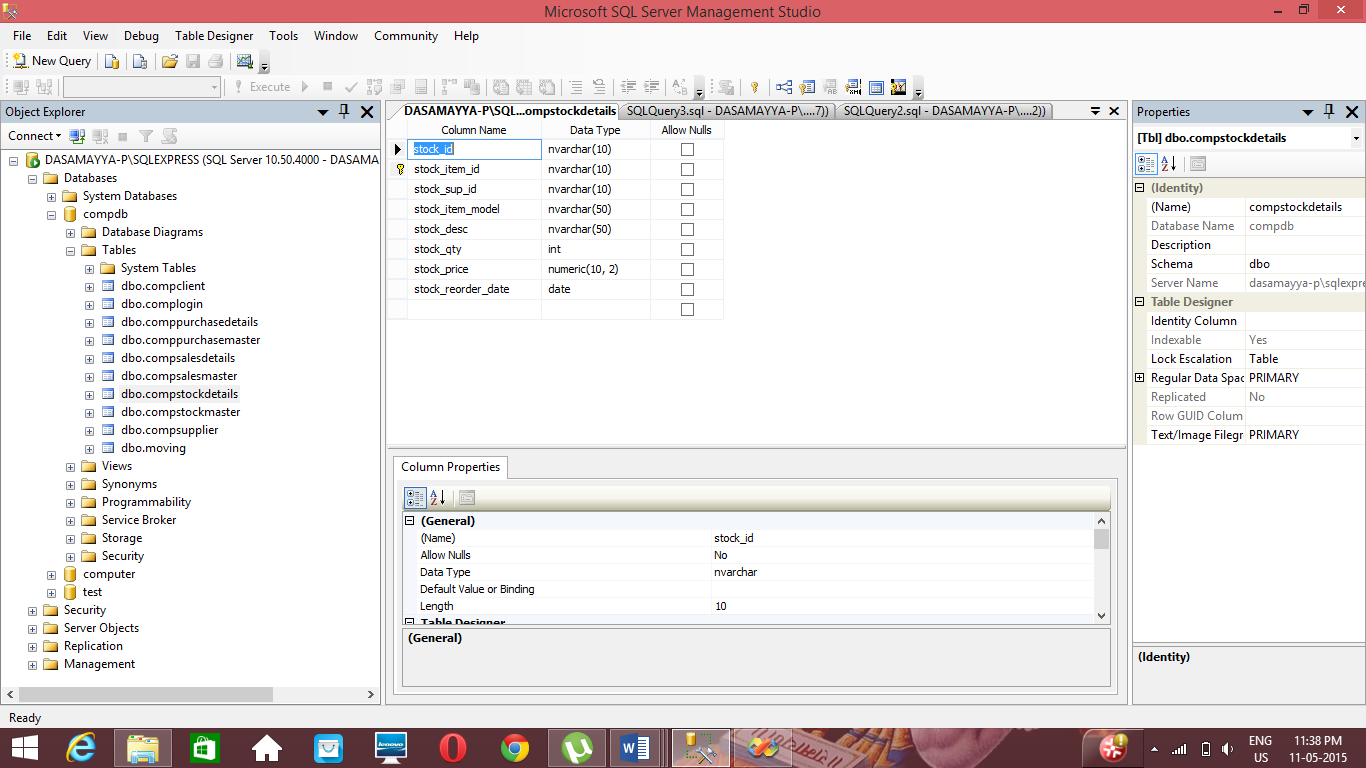
[prd\_rate] [numeric](10, 2)NOTNULL,

[prd\_amt] [numeric](10, 2)NULL

)ON [PRIMARY]

GO

Stock Detail:



CREATETABLE [dbo].[compstockdetails]

(

[stock\_id] [nvarchar](10)NOTNULL,

[stock\_item\_id] [nvarchar](10)NOTNULL,

[stock\_sup\_id] [nvarchar](10)NOTNULL,

[stock\_item\_model] [nvarchar](50)NOTNULL,

[stock\_desc] [nvarchar](50)NOTNULL,

[stock\_qty] [int] NOTNULL,

[stock\_price] [numeric](10, 2)NOTNULL,

[stock\_reorder\_date] [date] NOTNULL,

CONSTRAINT [PK\_compstockdetails] PRIMARYKEYCLUSTERED

(

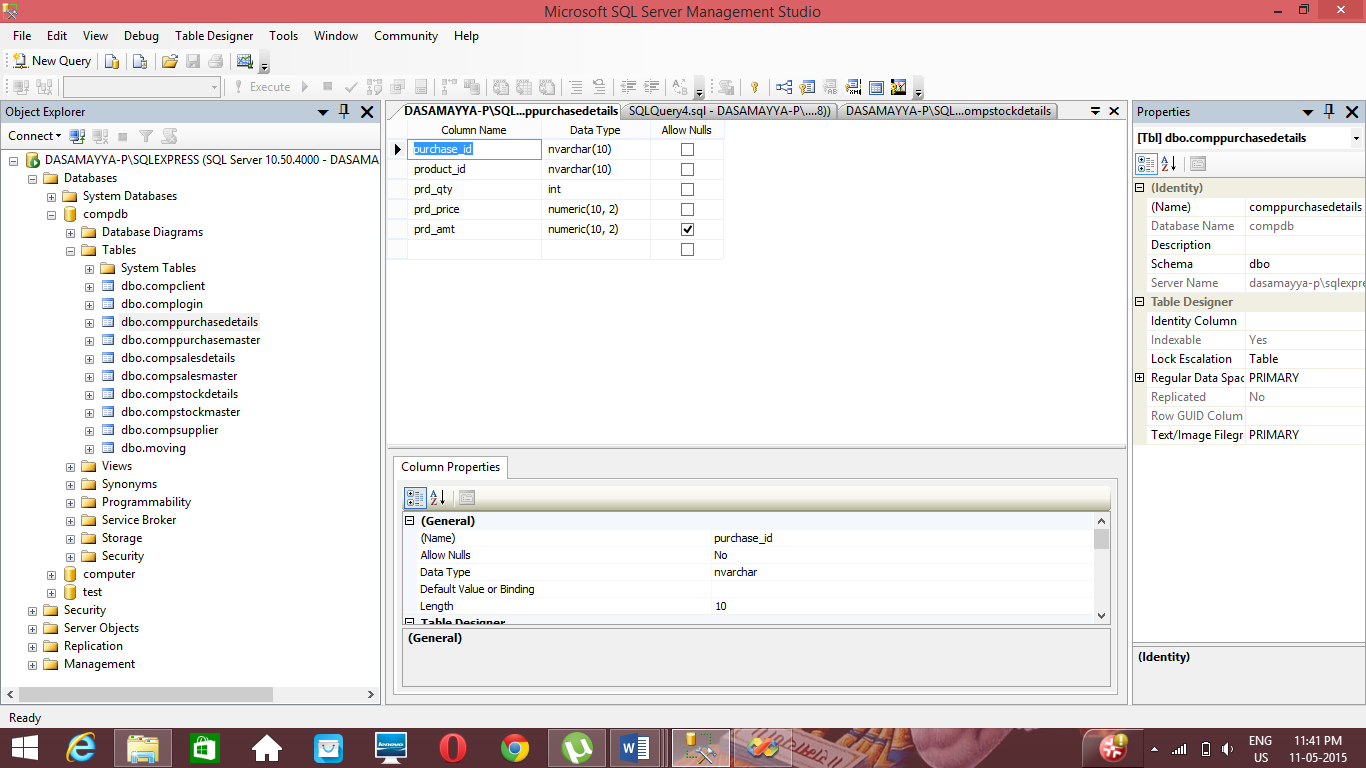
[stock\_item\_id] ASC

)WITH (PAD\_INDEX=OFF,STATISTICS\_NORECOMPUTE=OFF,IGNORE\_DUP\_KEY=OFF,ALLOW\_ROW\_LOCKS=ON,ALLOW\_PAGE\_LOCKS=ON)ON [PRIMARY]

)ON [PRIMARY]

GO

**Bill Details :**



CREATETABLE [dbo].[comppurchasedetails](

[purchase\_id] [nvarchar](10)NOTNULL,

[product\_id] [nvarchar](10)NOTNULL,

[prd\_qty] [int] NOTNULL,

[prd\_price] [numeric](10, 2)NOTNULL,

[prd\_amt] [numeric](10, 2)NULL

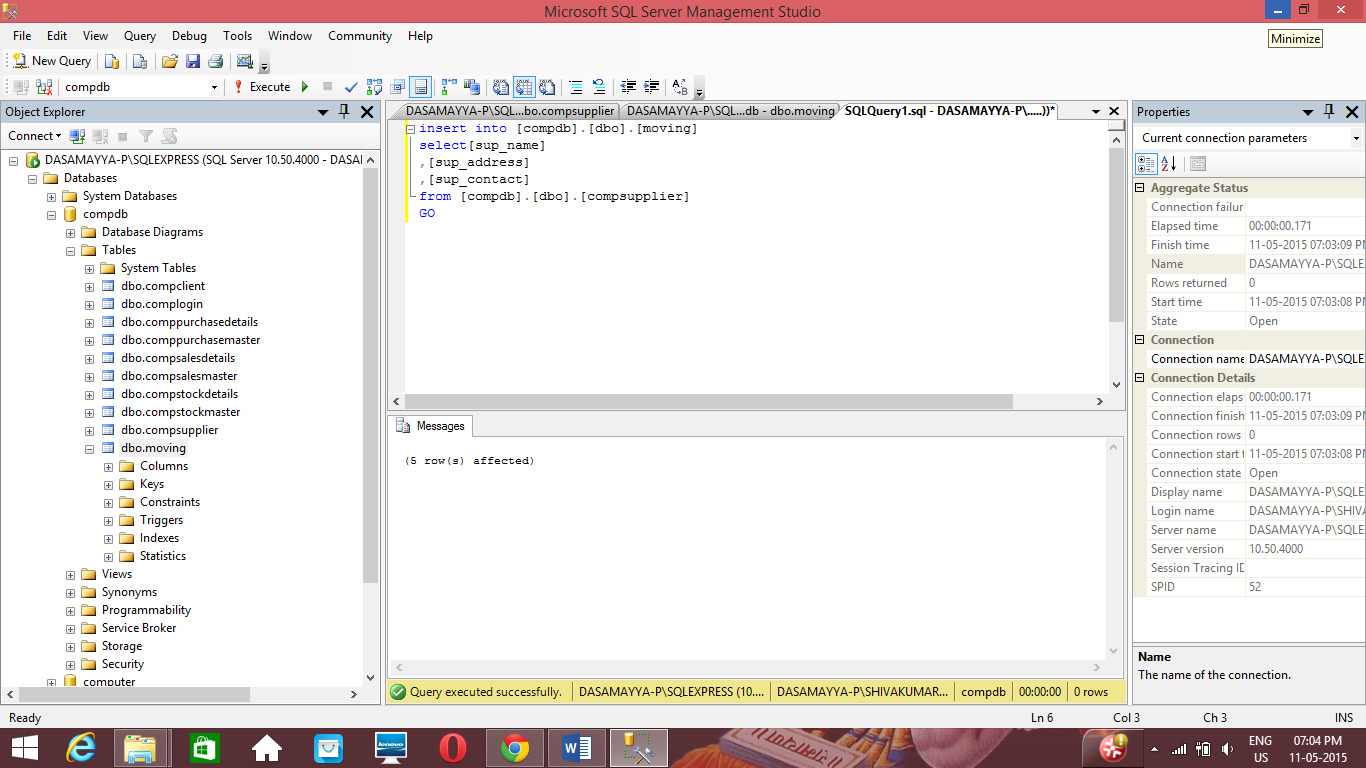
)ON [PRIMARY]

GO

*Moving data*

**Moving Data**

**Moving Data:**



INSERT INTO [compdb].[dbo].[moving]

SELECT [sup\_name]

,[sup\_address]

,[sup\_contact]]

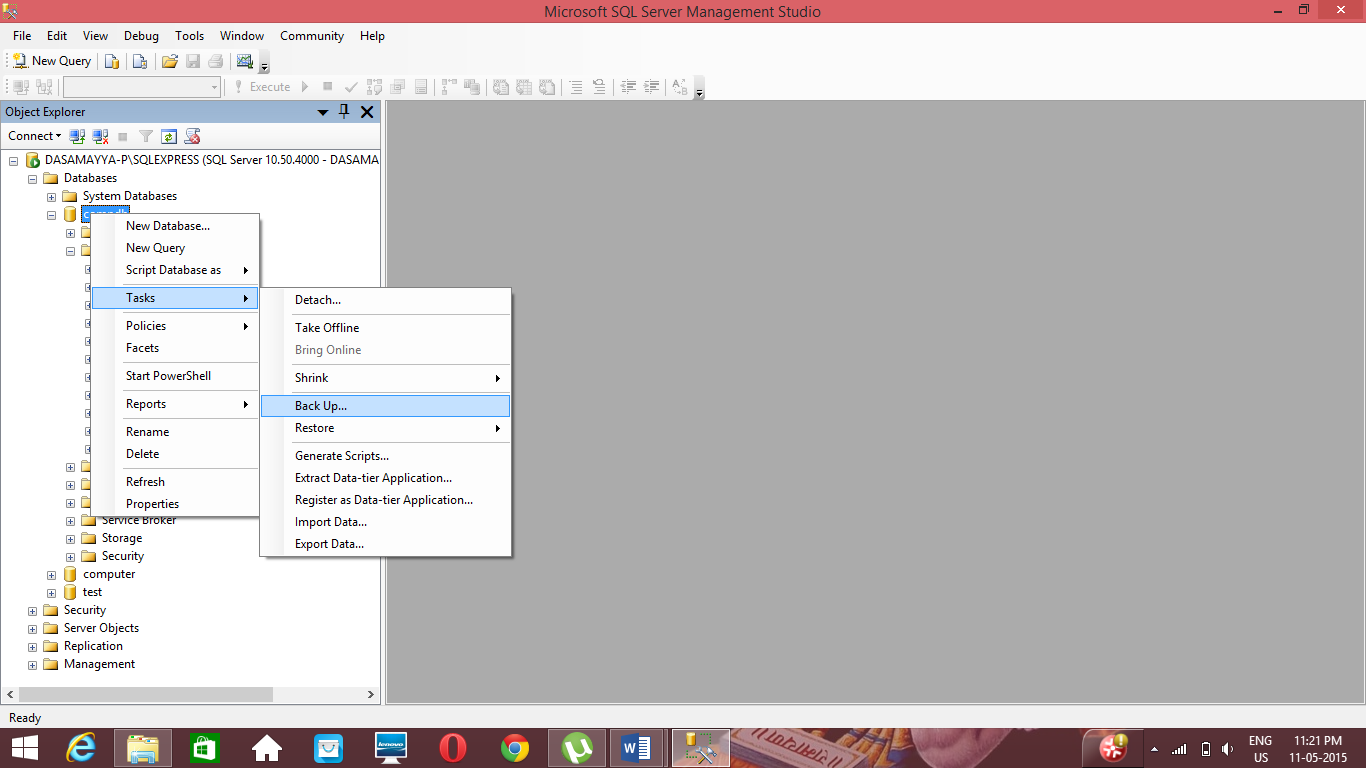
FROM [compdb].[dbo].[moving]

GO

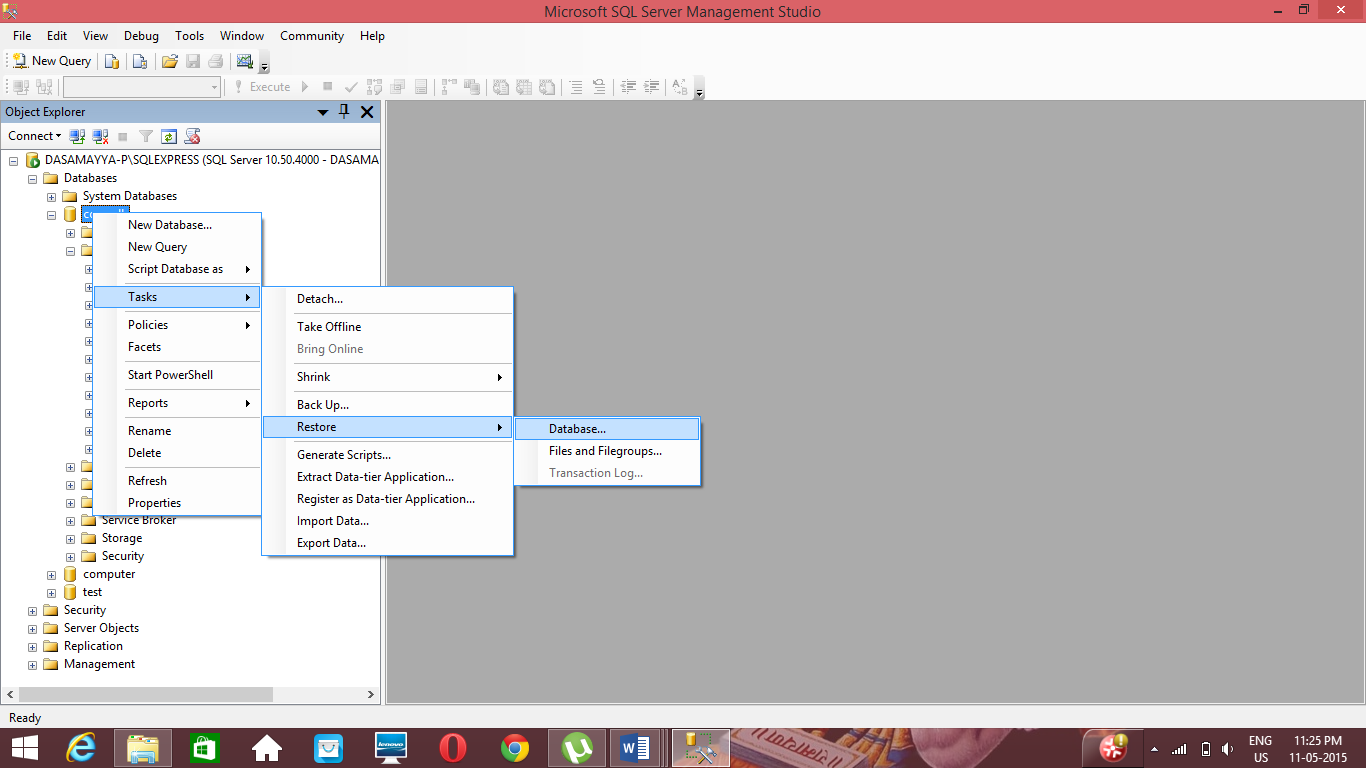
*Recovery*

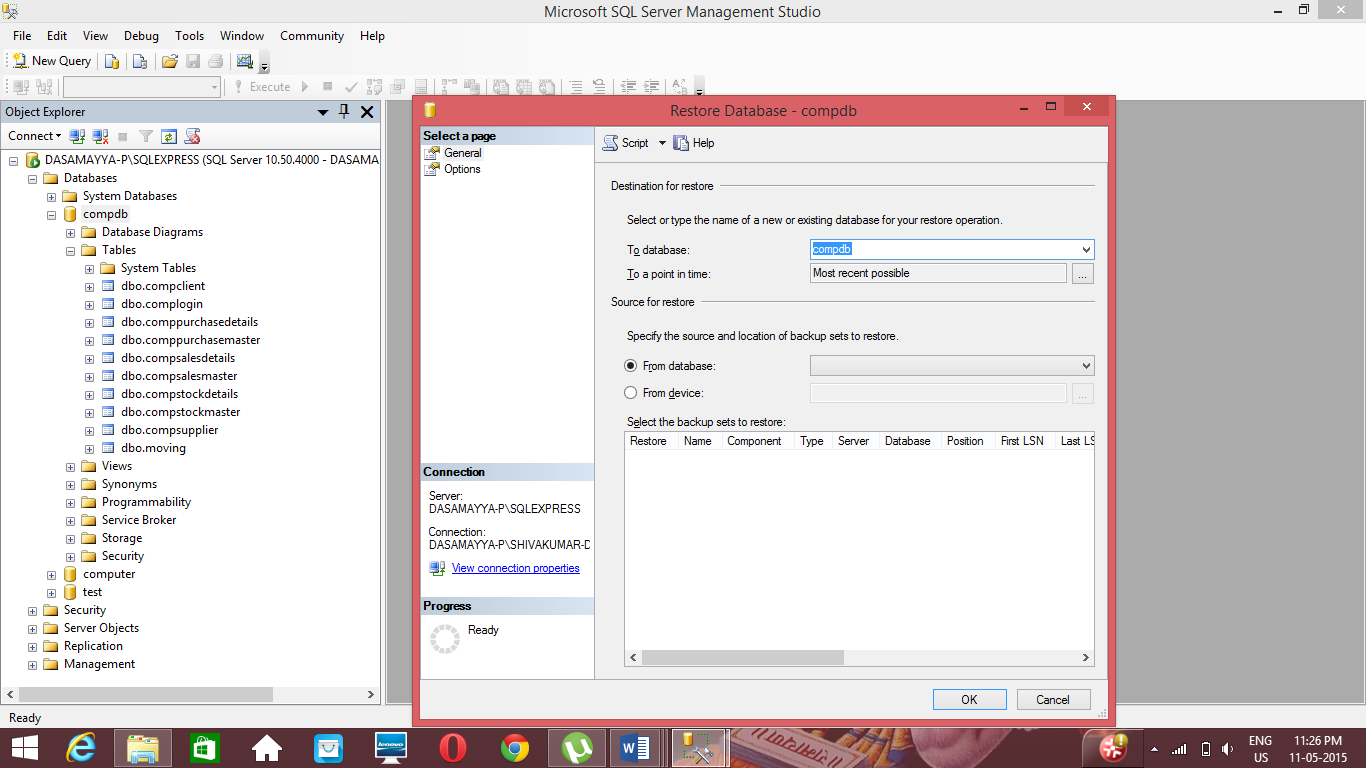
**Recovery**

**Backup:**



**Restore:**





*E-R diagram*

**E-R Diagram**

**Client**

**Purchase**

**Sales**

**Supplier**

**Stock**

**N**

**1**

**N**

Reduce**s**

**Sells**

Adds

Procure

**1**

**1**

N **N**

N

*Tool description*

**Visual Basic 2008**

Visual Basic Express 2012 is the latest version of Visual Basic launched by Microsoft in 2012. Visual Basic Express 2012 is almost similar to Visual Basic Express 2010 and Visual Basic 2008, but it has added many new features. The most distinct difference is that Visual Basic Express 2012 no more comes as a standalone program, it is now integrated with other Microsoft Programming languages C# and C++ in a package called Visual Studio 2012. Further, Visual Studio Express 2012 now come in five editions, they are:

* Visual Studio Express 2012 for Web
* Visual Studio Express 2012 for Windows 8
* Visual Studio Express 2012 for Windows Desktop
* Visual Studio Express 2012 for Windows Phone
* Visual Studio Team Foundation Server Express 2012

Like Visual Basic Express 2010 and Visual Basic Express 2008, Visual Basic Express 2012 is also a full-fledged Object-Oriented Programming (OOP) Language, so it has caught up with other OOP languages such as C++, Java, C# and others.

**Visual Basic 2008 Data Types:**

Visual Basic 2008 classifies the information mentioned above into two major data types, they are the numeric data types and the non-numeric data types.

Numeric Data Types

Numeric data types are types of data that consist of numbers, which can be computed mathematically with various standard operators such as add, minus, multiply, divide and so on. Examples of numeric data types are your examination marks, your height, your weight, the number of students in a class, share values, price of goods, monthly bills, fees and etc. In Visual Basic 2008, numeric data are divided into 7 types, depending on the range of values they can store. Calculations that only involve round figures or data that don't need precision can use Integer or Long integer in the computation. Programs that require high precision calculation need to use Single and Double decision data types, they are also called floating point numbers. For currency calculation, you can use the currency data types. Lastly, if even more precision is requires to perform calculations that involve a many decimal points, we can use the decimal data types

**Non-numeric Data Types**

Nonnumeric data types are data that cannot be manipulated mathematically using standard arithmetic operators. The non-numeric data comprises  text or string data types, the Date data types, the Boolean data types that store only two values (true or false), Object data type and Variant data type

**Visual Studio 2008 Project Page:**

The New Project Page comprises three templates, Visual Basic, Visual C# and Visual C++.We shall select Visual Basic. Visual Basic 2008 offers you four types of projects that you can create. As we are going to learn to create windows Applications, we will select Windows Forms Application.

At the bottom of this dialog box, you can change the default project name WindowsApplication1 to some other name you like, for example, My First Program. After you have renamed the project, click OK to continue.

Controls in Visual Basic 2008 are objects that can be placed on the form to perform various tasks. We can use them to create all kinds of Windows applications. The diagram below shows the toolbox that contains the controls of Visual Basic 2012. They are categorized into Common Controls, Containers, Menus, Toolbars, Data, Components, Printings and Dialogs. At the moment, we will focus on the common controls. Some of the most frequently used common controls are Button, Label, ComboBox, ListBox, PictureBox, Text Box etc.

To insert a control into your form in Visual Basic 2012 IDE, you just need to drag the control from the toolbox and drop it into the form. You can reposition and resize it as you like.

**The Control Properties in Visual Basic 2008:**

Before writing an event procedure for a control in Visual Basic 2008 to response to a user’s input or action, you have to set certain properties for the control to determine its appearance and how it will work with the event procedure. You can set the properties of the controls in the properties window of Visual Basic 2012 IDE at design time or at run time.

**Microsoft SQL Server 2008**

Microsoft SQL Server is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft). As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many [concurrent users](http://en.wikipedia.org/wiki/Concurrent_user). Its primary [query languages](http://en.wikipedia.org/wiki/Query_language) are [T-SQL](http://en.wikipedia.org/wiki/Transact-SQL) and [ANSI SQL](http://en.wikipedia.org/wiki/SQL).

SQL Server 2012 was released on April 2012 and it has started becoming favorite among professionals. Any new product comes from Microsoft the first thing I personally ask myself, is it worth to jump in?. Is it worth to spend customer’s hard earned money to get in to that product?. The way to assess the same is dividing the product features in to   “revolution” and “evolution”. “Revolution” means it’s completely a new thing while “evolution” means there was something already and it has been improvised.

**Features:**

1. **Always On Availability Groups**

This feature takes database mirroring to awhole new level. With always On, users will be able to fail over multiple databases in groups instead of individually. Also, secondary copies will be readable, and can be used for database backups. The big win is that your DR environment no longer needs to sit idle.

1. **Windows Server Core Support**

If you don't know what Windows Server Core is, you may want to come up to speed before Windows 8 (MS is making a push back to the command line for server products). Core is the GUI-less version of Windows that uses DOS and PowerShell for user interaction. It has a much lower footprint (50% less memory and disk space utilization), requires fewer patches, and is more secure than the full install. Starting with SQL 2012, it is supported for SQL Server.

1. **Column store Indexes**

This a cool new feature that is completely unique to SQL Server. They are special type of read-only index designed to be use with Data Warehouse queries. Basically, data is grouped and stored in a flat, compressed column index, greatly reducing I/O and memory utilization on large queries.

**4. User-Defined Server Roles**

DBAs have always had the ability to create custom database role, but never server wide. For example, if the DBA wanted to give a development team read/write access to every database on a shared server, traditionally the only ways to do it were either manually, or using undocumented procedures. Neither of which were good solutions. Now, the DBA can create a role, which has read/write access on every DB on the server, or any other custom server wide role.

**5. Enhanced Auditing Features**

Audit is now available in all editions of SQL Server. Additionally, users can define custom audit specifications to write custom events into the audit log. New filtering features give greater flexibility in choosing which events to write to the log.

**6. BI Semantic Model**

This is replacing the Analysis Services Unified Dimensional Model (or cubes most people referred to them). It's a hybrid model that allows one data model will support all BI experiences in SQL Server. Additionally, this will allow for some really neat text info graphics

**7. Sequence Objects**

For those folks who have worked with Oracle, this has been a long requested feature. A sequence is just an object that is a counter -- a good example of it's use would be to increment values in a table, based a trigger. SQL has always had similar functionality with identity columns, but now this is a discrete object.

**8. Enhanced PowerShell Support** –

Windows and SQL Server admins should definitely start brushing up on their PowerShell scripting skills. Microsoft is driving a lot of development effort into instrumenting all of their server-based products with PowerShell. SQL 2008 gave DBAs some exposure to it, but there are many more in cmdlets in SQL 2012.

**9. Distributed Replay**

Once again this is answer to a feature that Oracle released (Real Application Testing). However, and in my opinion where the real value proposition of SQL Server is, in Oracle it is a (very expensive) cost option to Enterprise Edition. With SQL, when you buy your licenses for Enterprise Edition, you get everything. Distributed replay allows you to capture a workload on a production server, and replay it on another machine. This way changes in underlying schemas, support packs, or hardware changes can be tested under production conditions.

**10. Power View**

You may have heard of this under the name "Project Crescent" it is a fairly powerful self-service BI toolkit that allows users to create mash ups of BI reports from all over the Enterprise.

**11. SQL Azure Enhancements**

These don't really go directly with the release of SQL 2012, but Microsoft is making some key enhancements to SQL Azure. Reporting Services for Azure will be available, along with backup to the Windows Azure data store, which is a huge enhancement. The maximum size of an Azure database is now up to 150G. Also Azure data sync allows a better hybrid model of cloud and on premise solutions

**12. Big Data Support**

The PASS (Professional Association for SQL Server) conference last year, Microsoft announced a partnership with Haddon provider Cloud era. One part of this involves MS releasing a ODBC driver for SQL Server that will run on a Linux platform. Additionally, Microsoft is building connectors for Haddon, which is an extremely popular NoSQL platform. With this announcement, Microsoft has made a clear move into this very rapidly growing space.

SQL 2012 is a big step forward for Microsoft -- the company is positioning itself to be a leader in availability and in the growing area of big data. As a database professional, I look forward to using SQL 2012 to bring new solutions to my clients.

*System analysis*

**Feasibility Study**

Feasibility study is the measure of how beneficial or practical the development of an information system will be to an organization. The Feasibility analysis is a cross life cycle activity and should be continuously performed throughout the system life cycle. Feasibility study lets the developer foresee the future of the project and the usefulness. The study on feasibility is done on the basis of a few factors. They are:

**Operational feasibility:**

By automating the Apartment Money Management System both the administrators and flat owners will feel better than when it was manual. Users will get a very quick service by reducing the manual recordings. Also administrators will feel comfortable by reduction of their work. Recording errors will be reduced and it is easy to handle a large database. Losing of records will also be avoided.

Considering all these factors we can conclude that all the users and end users will be satisfied by the system.

**Technical feasibility:**

The system must be evaluated from the technical point of view. The assessment of this feasibility must be based on an outline design of the system requirement in terms of input, output, programs and procedures.

For the design and development of the system, several software products have been accommodated.

Database design – SQL Server 2008

Interface design – VISUAL BASIC 2008.

This software has enough efficiency in producing the system. Therefore the project is technically feasible.

**Schedule feasibility:**

The duration of time required for the project has been planned appropriately and it is the same as the duration of time expected by the customer. Therefore the product can be delivered to the customer within the expected time duration, satisfying the customer. Hence the project is feasible in scheduling.

**Economic feasibility:**

According to the resources available and the project scheduling process it is estimated that the expenses allocated for the software to be developed, by the customer is sufficient enough. Hence the economical factor has been considered feasible.

**Behavioural Feasibility:**

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

This project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

*Data flow diagram*

**DATA FLOW DIAGRAM**

**Data Flow diagram:**

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).

On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored (all of which are shown on a DFD)

The idea behind the explosion of a process into more process is that understanding at one level of details is exploded into greater detailed at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form, this lead to modular design.

A DFD is known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformation they will become program in system design. So it is the starting point of the design to lowest level of details. A DFD consists of series of bubbles joined by data flows in the system.

**DFD Symbols**

In DFD, there are four Symbols

1. A square-defines a source or destination system data
2. An arrow identified data flow. Its is the pipeline through which the information flow
3. A circle or a bubble represents a process that transforms
4. Incoming data flow into outgoing data flows
5. An open rectangle is a data source, data at rest or a temporary of data

**Process that transforms data flow**

**Source or destination of the data**

**Data flow**

**Process for data store.**

**CONTEXT LEVEL :**

Database

Enter Details Store Data

**User**

Retrieve Data

**LEVEL 1 :**

**User**

Enter Details Verifying

Login

Quotation

Stock

Billing

Reports

Storing

Database

**LEVEL 2 :**

Enter details Verifying

**User**

Login

Stock

Quotation

Billing

Reports

Stock

Login

Insert InsertInsert View

Store Store Store Store

Stock

Quotation

Reports

Billing

*Database design*

**DATABASE DESIGN**

**Client:**

|  |  |
| --- | --- |
| Column Name | Data Type |
| Clid | nvarchar(10) |
| Clname | nvarchar(50) |
| Claddress | nvarchar(50) |
| Clnumber | nvarchar(50) |
| Org\_name | nvarchar(50) |

**Purchase Master:**

|  |  |
| --- | --- |
| Column Name | Data Type |
| Purchase\_id | nvarchar(10) |
| Product\_id | nvarchar(50) |
| Prdqty | nvarchar(50) |
| Prdrate | nvarchar(50) |
| prdamt | nvarchar(50) |

**Sales Master:**

|  |  |
| --- | --- |
| Column Name | Data Type |
| Sales\_id | nvarchar(100 |
| Sales\_date | nvarchar(50) |
| Cust\_id | nvarchar(50) |
| Sales\_amt | nvarchar(50) |

**Stock Master:**

|  |  |
| --- | --- |
| Column Name | Data Type |
| Stock\_id | nvarchar(10) |
| Stock\_type | nvarchar(50) |
| Stock\_name | nvarchar(50) |

**Stock entry:**

|  |  |
| --- | --- |
| Column Name | Data Type |
| Stockid | nvarchar(10) |
| Stock\_item\_id | nvarchar(50) |
| Stock\_sup\_id | nvarchar(50) |
| Stock\_item\_model | nvarchar(50) |
| Stock\_desc | nvarcahr(50) |
| Stock\_qty | Int |
| Stock\_price | numeric(10,2) |
| Stock\_reorder | date |

*Source code*

**SOURCE CODE**

**CODE**

**LOGIN FORM**

Imports System.Data

Imports System.Data.SqlClient

PublicClass frmLogin

PrivateSub cmdLogin\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdLogin.Click

Dim connstr AsNewString("Data Source=DASAMAYYA-P\SQLEXPRESS;Initial Catalog=compdb;Integrated Security = True")

Dim objconn As SqlConnection = New SqlConnection(connstr)

Dim objcmd As SqlCommand = New SqlCommand()

Dim objdatareader As SqlDataReader

Dim flag AsBoolean

objcmd.Connection = objconn

Dim un AsString, pwd AsString

un = Trim(txtusername.Text)

pwd = Trim(txtpassword.Text)

If un = ""Or pwd = ""Then

MessageBox.Show("Please enter username and password")

Else

objcmd.CommandText = "select \* from dbo.complogin where username='"& un &"' and password='"& pwd &"'"

objconn.Open()

objdatareader = objcmd.ExecuteReader()

While objdatareader.Read

flag = True

EndWhile

If flag = TrueThen

MessageBox.Show("Login Successful")

Me.Hide()

MDIParent1.Show()

Else

MessageBox.Show("Invalid credentials")

txtusername.Text = ""

txtpassword.Text = ""

EndIf

objconn.Close()

EndIf

EndSub

PrivateSub cmdExit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdExit.Click

End

EndSub

EndClass

**COMPANY FORM**

Imports System.Text.RegularExpressions

Imports System.Data

Imports System.Data.SqlClient

PublicClass companyform

PrivateSub txtcompname\_Leave(ByVal sender AsObject, ByVal e As System.EventArgs) Handles txtcompname.Leave

Dim cname AsString

cname = txtcompname.Text

IfString.IsNullOrEmpty(cname) Then

MessageBox.Show("Please enter company name")

txtcompname.Focus()

ElseIfNot Regex.Match(cname, "^[a-z,A-Z]\*$").Success Then

MessageBox.Show("Please Enter only ALPHABHET in the company name field")

txtcompname.Clear()

txtcompname.Focus()

EndIf

EndSub

PrivateSub cmdadd\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdadd.Click

Dim connstr AsNewString("Data Source=DASAMAYYA-P\SQLEXPRESS;Initial Catalog=compdb;Integrated Security = True")

Dim objconn As SqlConnection = New SqlConnection(connstr)

Dim objcmd As SqlCommand = New SqlCommand()

objcmd.Connection = objconn

objcmd.CommandText = "Insert into dbo.compsupplier(sup\_id,sup\_name,sup\_address,sup\_contact, sup\_email) values(@sup\_id, @sup\_name,@sup\_address,@sup\_contact,@sup\_email)"

objcmd.Parameters.AddWithValue("@sup\_id", txtcompid.Text)

objcmd.Parameters.AddWithValue("@sup\_name", txtcompname.Text)

objcmd.Parameters.AddWithValue("@sup\_address", txtaddress.Text)

objcmd.Parameters.AddWithValue("@sup\_contact", txtphone.Text)

objcmd.Parameters.AddWithValue("@sup\_email", txtemail.Text)

'objcmd.CommandText = "Insert into dbo.compsupplier values('" & txtcompid.Text & "','" & txtcompname.Text & "','" & txtaddress.Text & "','" & txtphone.Text & "','" & txtemail.Text & "')"

objconn.Open()

objcmd.ExecuteNonQuery()

objconn.Close()

MessageBox.Show("supplier details added")

txtcompid.Text = ""

txtcompname.Text = ""

txtaddress.Text = ""

txtemail.Text = ""

txtphone.Text = ""

EndSub

PrivateSub cmdview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmdview.Click

Dim connstr AsNewString("Data Source=DASAMAYYA-P\SQLEXPRESS;Initial Catalog=compdb;Integrated Security = True")

Dim objconn As SqlConnection = New SqlConnection(connstr)

objconn.Open()

Dim objdataadapter AsNew SqlDataAdapter("select \* from dbo.compsupplier", objconn)

Dim objdataset AsNew DataSet

objconn.Close()

objdataadapter.Fill(objdataset)

DataGridView1.DataSource = objdataset.Tables(0)

MessageBox.Show("supplier details fetched")

EndSub

EndClass

**TESTING**

*Testing*

**SYSTEM TESTING**

**TESTING**

Testing goes through the various stages, during testing the program to be tested has to be executed with a set of test cases, and ha the output of the program for the test case is evaluated to determine if the program is performing as expected. Due to its approach dynamic testing only ascertains the presence of error in the program. The exact nature of error is not usually decided by testing. Testing form is the first in determining error in the program.

Once the programs are tested individually then the system as a whole needs to be tested. During testing, the system is used experimentally to ensure that the software does not fail i.e. it will run according to its specification. The programs executed to check for any syntax or logical error. The error is corrected and test is made to determine whether the program is doing what it is supposed to do.

**Various types of testing**

**Unit testing**

Each component of the system is tested individually. The programmer does the testing. Testing is restrictive in nature i.e. programmer should try to test all individual conditions and see if the program breaks under any circumstance.

**System testing**

This is an integrated form of testing, which focuses on functionality and interface between units and team in a controlled environment does it.

**Acceptance Testing**

This is system testing done by the user of the application the only emphasis is functionally testing as the user is not aware of the technical aspect of the system. The testing is also done in a controlled environment with logging o all error based on the error found in the system, the user has to accept or reject the system.

**Module Testing**

This is an optional form of testing, which is done only for large system, which has a large number of modules.

**Security Testing**

Security testing will be done as a specialized form of testing if there is a high risk exposure in that area. If the risk exposure is not very high, then it can be done as part of the system testing. Typically, security testing would involve trying to break in to the system, trying to execute transactions not allowed to person; to access areas on disk were the user is not allowed.

Testing is vital t the success of the system. If it on. This done successfully, this shows that the parts of the system are working correctly and all the goals are achieved.

**IMPLEMENTATION**

Implementation is used here to mean the process of converting a new or revised system design int operational one; conversion is one aspect of implementation. The other aspect is post implementation review and software and maintenance.

There are three types f implementation:

1. Implementation of a computer system
2. Implementation of a new computer system
3. Implementation of a modified application.

**MAINTAINANCE**

After the system has successfully been implemented maintenance activity may require continuous involvement of the developers. Provision must be made for environmental changes, which may affect either the computer, or other parts of the computer based system: such activity is normally called maintenance. It includes both the improvement of the system functions and the correction of faults that arise during the operation of a system.

Maintenance activity may require the continuing involvement of a large proportion of computer department resources. Maintenance works may arise due to two reasons:

1. Errors that creep up during normal running
2. Request for changes by the service providers. As part of the normal running of the system when errors are found.

This maintenance work will help to ensure that the system works smoothly as predicted in the open environment. Whenever a maintenance work arises, the work has to be properly carried out with proper documentation. This is

to avoid any form of changes in the structure of the system.

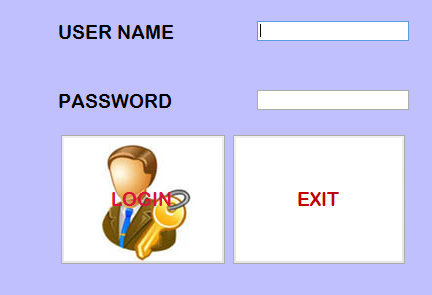
For every maintenance work an amendment notification is to be issued. This notification will have required changes and also authenticated. On the receipt of the amendment notification the amendment

Log is prepared which records these courses of action that has been planned to be taken. It also records the estimated and the actual completion of each activity.

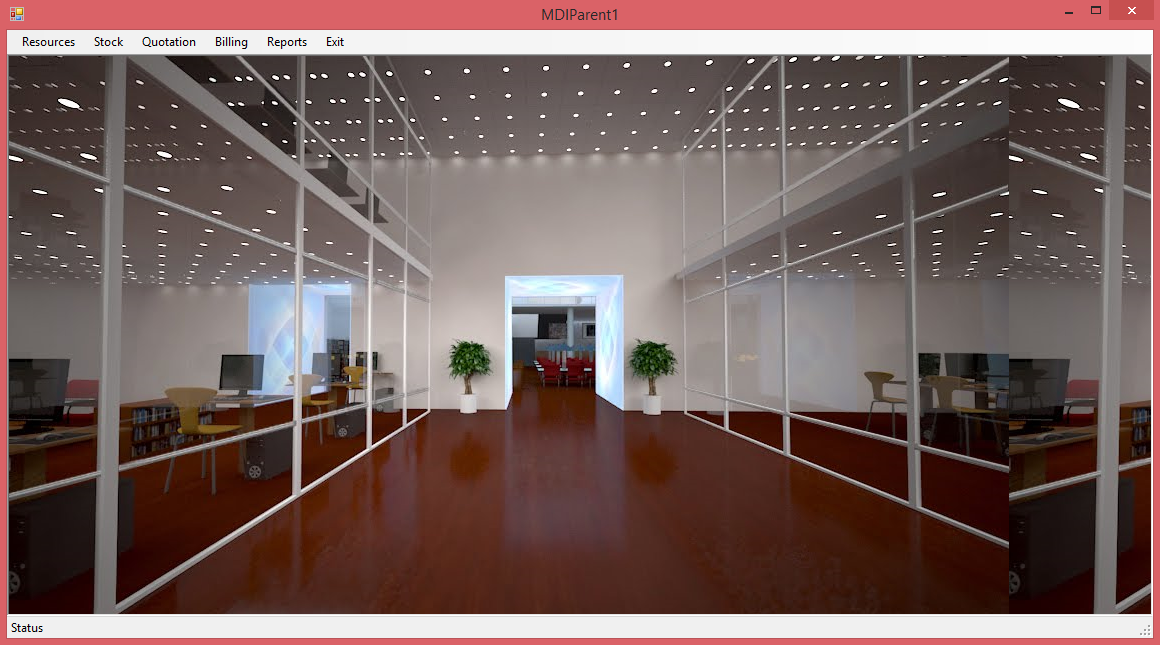
*Snapshots*

**SNAPSHOTS**

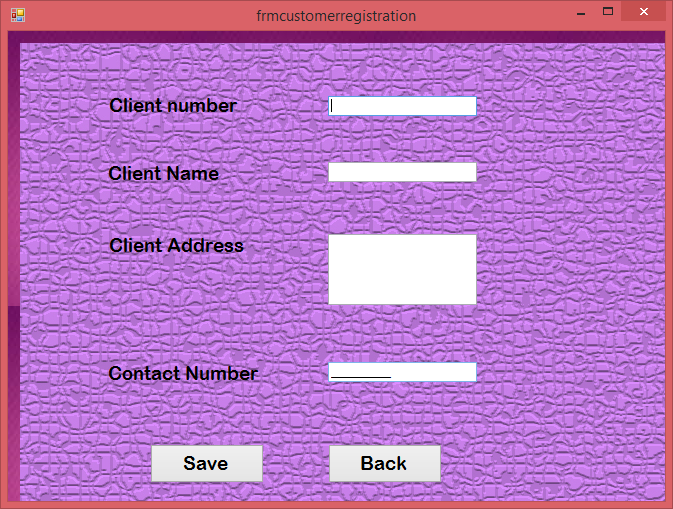
**Login Page:**

****

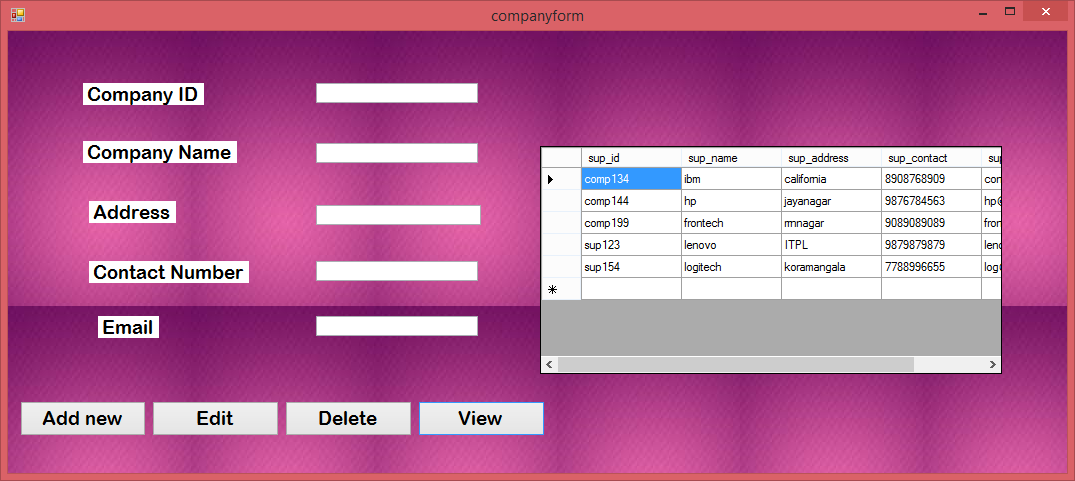
**Main Page:**

****

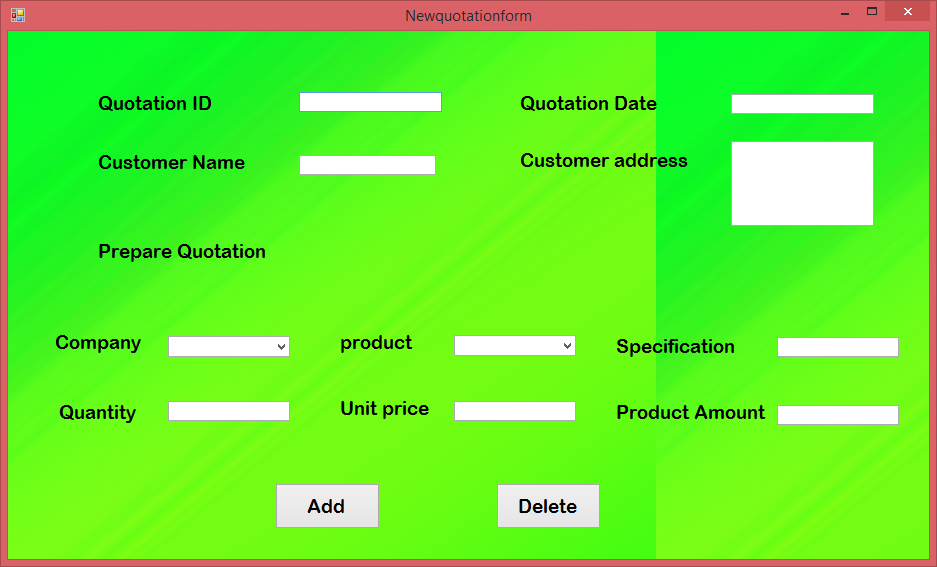
**Client:**



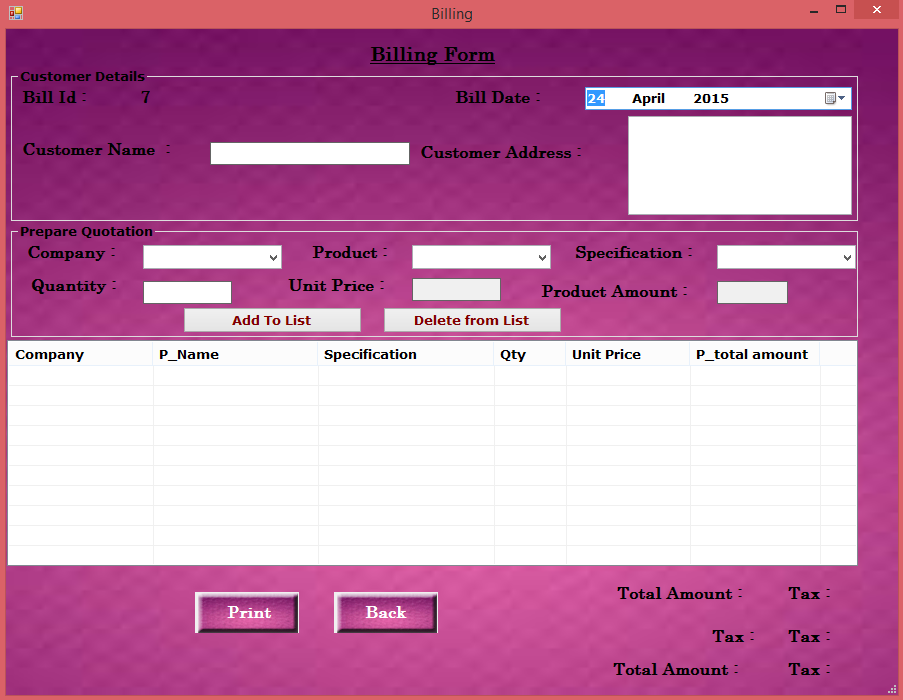
**Company:**



**Quotation:**



**Billing:**

****

*Future Enhancements*

**FUTURE ENHANCEMENT**

As per the computer warehouse needs new application can be integrated without any modification to the existing application and know it is maintaining only computer equipment details so next we try to include with some other electronics goods transaction details by giving online facility. We can also provide barcode facility to improve the productivity of computer shop

**MERITS**

* This project gives detailed overview over the goods in Stock
* Increase in efficiency of storage operations and goods flow within warehouse.
* It improves accuracy of the work of warehouse operators.
* It provides safety and security.
* We have provided user-friendly interface to help the user to operate the system.
* Automated bill facility is provided which avoids the manual errors.
* Quotations can be generated at his fingertip without any errors.

*Conclusion*

# CONCLUSION

This project helps to maintain the all the details of computer sales of computer shop. We hope that this project can be most useful in all Computer warehouse. In shops the user can easily maintain equipment business details easily. It is also easy to calculate the account and update the stock details. Show the details of spars with its respected feature to help shop employs to explain the customers about all the computer equipment’s.

In our project titled “Computer Warehouse Management” using the DBMS MS-Access-2007 as a backend and VB.Net 2008 as front end we have tried to provide many options like Entry forms, View forms, Billing forms and Reports, which are helpful to the owner to give better service to the user.

Since the implementation of a database application by us was related to Computer warehouse management, we could know how real world constraints have to be dealt with and how, any problems that arose could be solved.

*Bibliography*

**Bibliography**

* Beginning SQL Server 2008
* Denise Gosnell
* .NET FRAME WORK
* Matthew A. Stocker & Steven J. Stein
* MSDN CDs
* Visual basic .NET bible
* <http://www.tutorialspoint.com/vb.net/>