Preface

I would like to thank everyone who has directly or indirectly helped me to do this project. I would like to thank Mr. Bipin Pokhrel CEO of ODDIY eco-international who provided me with his business information. I am grateful towards my computer teacher Tr. Rakesh Chaudhary who taught me programming from basic and guided me through my project. I am also glad towards my classmates and seniors who guided me and helped me in absence or busyness of the teacher.

And finally would also like to thank the Cambridge University for keeping this interesting module in A-levels syllabus through which I could be a real life programmer.

Table of contents

1. Definition, Investigation and Analysis

Definition

About the Client: *OODDIY eco International Pvt. Ltd.*

Ooddiy eco international is a business organization established on 2013 A.D. with an aim to bring international standard luxury products in Nepal to increase the standard of living. It is the authorized dealer of Bio design pools, Magiline pools, volks lift, and IBIJI lifts in Nepal. The raw materials are imported and construction is conducted in the sites. This company is still in its initial phase of business. The organization has not hired any professionals to design its system. Currently the works are being done using common application like MS word, Excel.

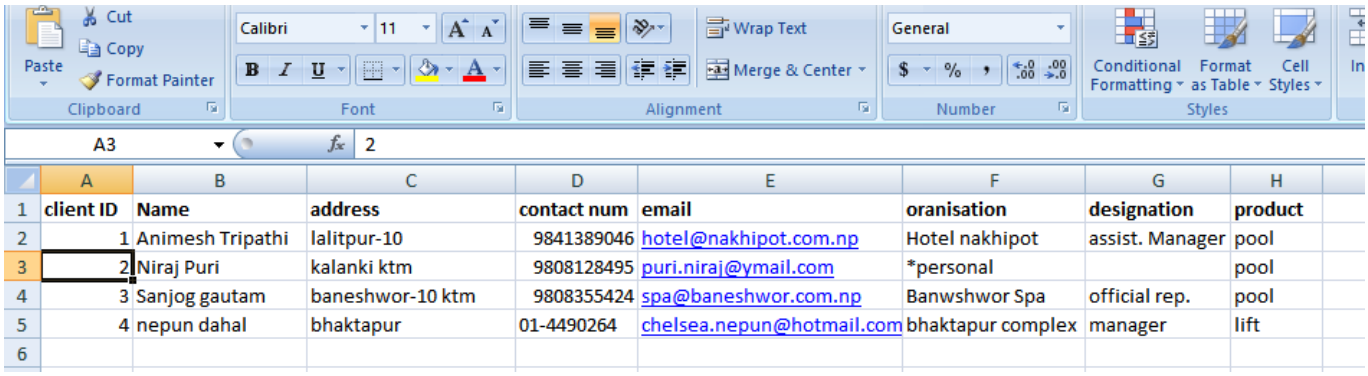
Defining the problem

The problem of the organization seems to be that the ineffective method of generating quotation and keeping its record. Quotation is the initial phase of the business which provides the client with the information about the product (rates, VAT, price, transport) so, it is vital means of communication with the client. The proper study of the past quotations helps the organization to formulate its business plans. So, we have agreed upon making an application that can effectively handle quotation (create, review, edit, and search).

Methods Currently in Use

Quotation is the official and main form of communication with the clients. Quotation generation is regular and day to day task of the organization. The process of creating the quotation was by deleting the fields of previous quotation as a word document and updating the new information. All the Calculations had to be done using a calculator. For future references the quotations were saves as word documents in related folders.

The basic information about the client is recorded in a spreadsheet temporarily. When the deal is finalized the information is used to update the word document file.



	A	B	C	D	E	F	G	H
1	**client ID**	**Name**	**address**	**contact num**	**email**	**oranisation**	**designation**	**product**
2	1	Animesh Tripathi	lalitpur-10	9841389046	hotel@nakhipot.com.np	Hotel nakhipot	assist. Manager	pool
3	2	Niraj Puri	kalanki ktm	9808128495	puri.niraj@ymail.com	*personal		pool
4	3	Sanjog gautam	baneshwor-10 ktm	9808355424	spa@baneshwor.com.np	Banwshwor Spa	official rep.	pool
5	4	nepun dahal	bhaktapur	01-4490264	chelsea.nepun@hotmail.com	bhaktapur complex	manager	lift
6								

Fig: current method of storing client info.

Origin of Data

The data required of produce the program were obtained from the following sources:

Studying current quotation:

The main source of information and the general format was obtained and designed on the basis of the previous produced quotations.

Studying current method:

By studying the current system I was able to find more about the technical part of the project system. Like the format of generating quotation ID, product ID, classifying the clients on the basis of the products they buy.

Email sent by the clients:

Basic information about the clients and the transaction can be obtained from the emails sent and the forms can be designed accordingly

Studying the product types

The organization imports products from various international companies and studying the product manufactured by the companies helps to design the sub-product section and to generate a function that gives the ID values.

Investigation

Method of Investigation: Interview

Before talking about the specifications and requirements I got to know about the background of the organization by short interview with the CEO Mr. Bipin Pokhrel.

When was the Organization established?

We are just in the growing phase. It's just been a year that this organization was established. The Organization was established in the year 2013 AD

What was the motive of the business?

After some market study I saw that Nepal is developing and getting urbanized. I saw the scope of quality swimming pools/Lifts/Elevators in Nepal so it is my main motive to bring international standard products in Nepal.

How is the business nowadays?

The political transition has become a challenge but my marked study was good and I am running a good business.

What are the problems you face while running your business?

There exist problems like labor problems, lack of qualified human resources, political instability and the list goes on.

How many products have you sold yet?

We are still in a beginners phase. During the time of last 6 months we sold 7 products and these products need time to get installed. About one product in a month is a good rate.

How much time do you need to install one product?

It depends upon the product but the international duration to complete one swimming pool is 17 days. In context of Nepal we require about 23 days.

Do you manufacture the Product or import?

I wish we could manufacture but for now we import the product like the lifts come from china. The swimming pool materials are from Italy.

What are your future plans for the organization?

The ultimate goal is to be able to manufacture high quality international standards products and even export to other countries.

What benefits can you get from the system I design?

Communicating with our clients is very important and delicate task. "Quotation Generator" software would save our time and keep a good record of the past quotations which would be easy for us to review our work and plan accordingly.

There is only one person appointed for quotation generation so the software will be used by one user in one computer.

Method of Investigation: Direct Observation

While the co-workers were dealing with the client and were preparing a rough draft for the quotation I got to receive first hand information about the following areas.

1) When client arrives the basic information and contact information is noted on a rough draft. When both parties agree on their deal and the transaction is to happen further details are the recorded on a spread sheet

2) Client may approach the organization form email. Similar process is carried out and the details are updated on the spreadsheet

3) Once the deal is final the final quotation is created by erasing the field of the old quotation in word document and updating it by referring the spreadsheet.

4) The word documents are then stored on the basis of products sold under product folder.

Findings

When sufficient data collected after investigation I came up with some conclusion:

1) The data initially entered into spread sheet could be used later for automatic quotation generation.

2) Erasing the previous quotation is not the professional way of doing the work.

3) During the time of review the quotation can only be searched on the basis if the product bought by the client. As the word documents are arranged in folders based on the products.

4) Workers have to literally go through every quotation if they want to search a quotation by parameter other than product

Analysis

Limitations of the current system:

1) The main drawback in the current system is the inability to search the required quotation on the basis of specific parameter.

2) Currently the organization only has few quotations that look manageable in folders but as the size of the organization grows the files become unmanageable.

3) While backing up the data the entire word documents are copied which occupy unnecessary disk space.

Program Requirements

After studying the current system and methods the following are the areas that my program must cover in order to overcome the limitations.

- Provide a user friendly form-based interface for data entry purpose which contains field for every information need to generate a quotation.
- A database is required which stores the field values permanently form the fields.
- User friendly form that allows the user to search previous quotation through various parameters.
- Establish a connection with the printer that prints out the quotation.
- When one record is deleted it must be removed from all the tables from the database.
- A system with privacy maintenance has to be made so the confidential information can be maintained
- System should allow multiple quotation generation on single date.

System Requirements

Hardware Requirements

The minimum hardware requirements for the system to run effectively are:

- A Pentium 3 processor 700 MHZ or better with a compatible motherboard.
- Minimum RAM of 256 MB
- A hard disk to store and save the files, including the program.
- A low resolution monitor to display the output.
- A mouse and a keyboard for entering the data into the system.
- USB flash drive support/CD-writer/ portable hard drives (for backups)
- An inkjet or laser printer to print reports.

Software Requirements

The minimum software requirements are:

- Operating System (The program runs in all the Windows NT operating system like Windows NT, Windows XP, Windows Vista, etc.)
- .Net framework (3.5 or higher)
- Microsoft Office 2003 or higher with MS Access (database management system) component installed.

Alternative Solution

Manual Solution:

Currently, this method is being implemented by the organization to store all Quotation details. The fields are manually being entered on the word processing.

Computerized Solution:

I. Microsoft Access itself can be used to store information about the Quotation. This method will be much more efficient than storing records manually. Microsoft Access has various useful features like searching a table in the database, filtering records, etc. (only to store the data)

II. A Be-Spoke Software (a software created with the purpose to meet the specific demand of a system) can be developed in order to simplify the problem. It would provide features exactly matching what the system needs.

Pros and cons of the proposed system

Current System

Advantage: No extra knowledge is required about the designed system or MS access.

Disadvantage: The records stored are not stored in a good manner, inefficient search, extra space consuming, manual calculation.

Storing data in MS access

Advantage: The searching part is made easy and it also consumes less space to store the data.

Disadvantage: The data has to be entered twice (in the Access table and changing the fields in word document). The form for data entry is not user friendly. The operator has to learn about the database entity relationships and other access tools.

Using Tailor-made software

Advantage: It is the best solution that the organization can have as it fulfills all the requirements and the data only has to be entered once. All the calculation and field update are done automatically so it is time saving and has less-errors. The operator does not need to learn about the entity relationships or table update as it is done by the system codes.

Disadvantage: The user however has to have a general knowledge about reading access tables. The system itself takes some time to be made. Since it is tailored specifically for particular user it is expensive than other methods.

Feasibility Study

Technical Feasibility:

- The designed program is a simple be-spoke software. Repairs and maintenance is easily available.
- The computers being used for current system are already compatible for new system

Economic Feasibility:

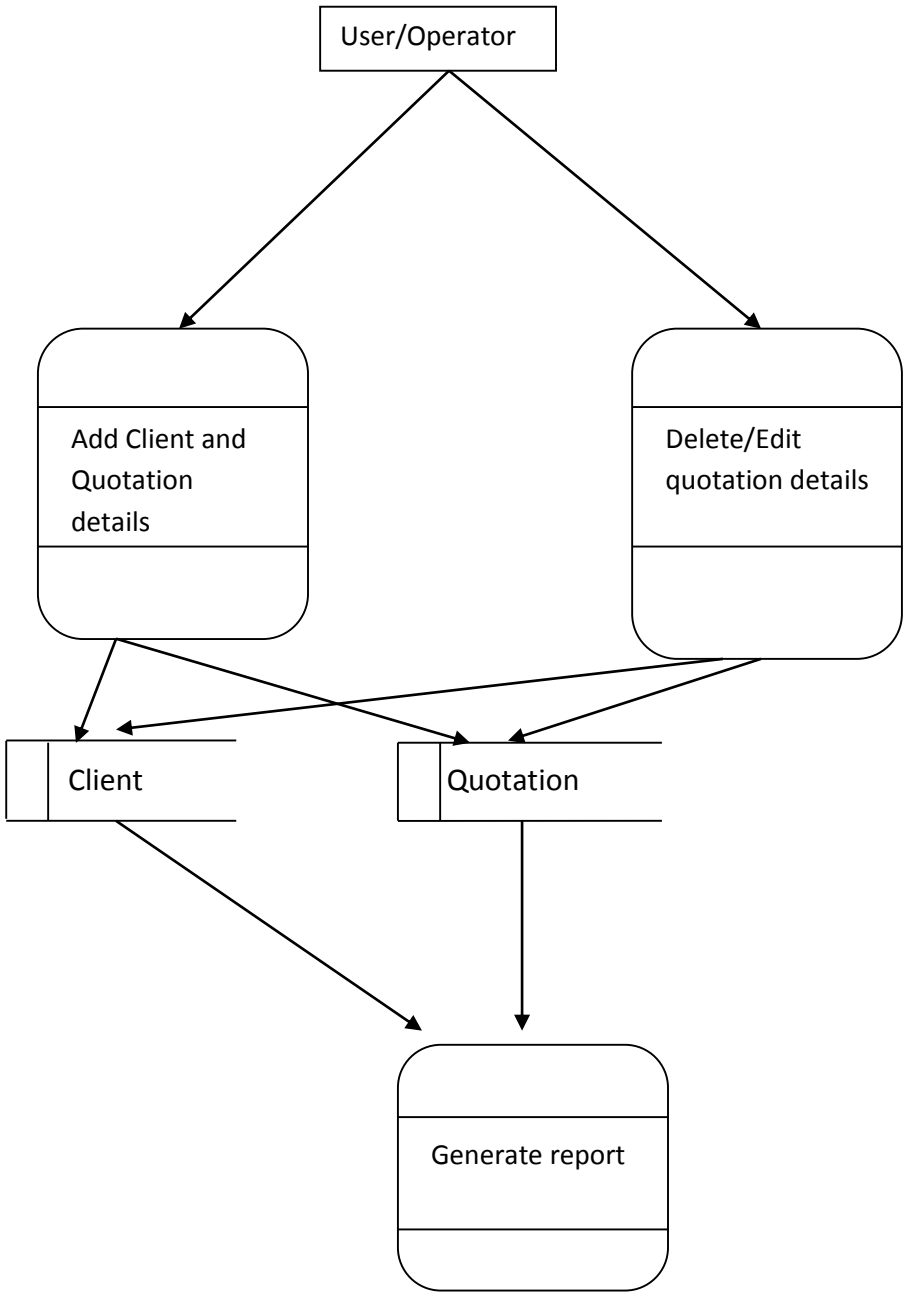
- The program is simple and easily understood so manpower required for its maintenance can be found in cheap.
- All the hardware and software requirements are present so no need for extra cost.

Legal Feasibility:

- The system follows the rule of Data Capture Rights of the country.
- Copyrights of the system must be registered.

Data Flow Diagram

Top Level Data Flow Diagram



2. System Design

Introduction

Entity-Relationship Diagram

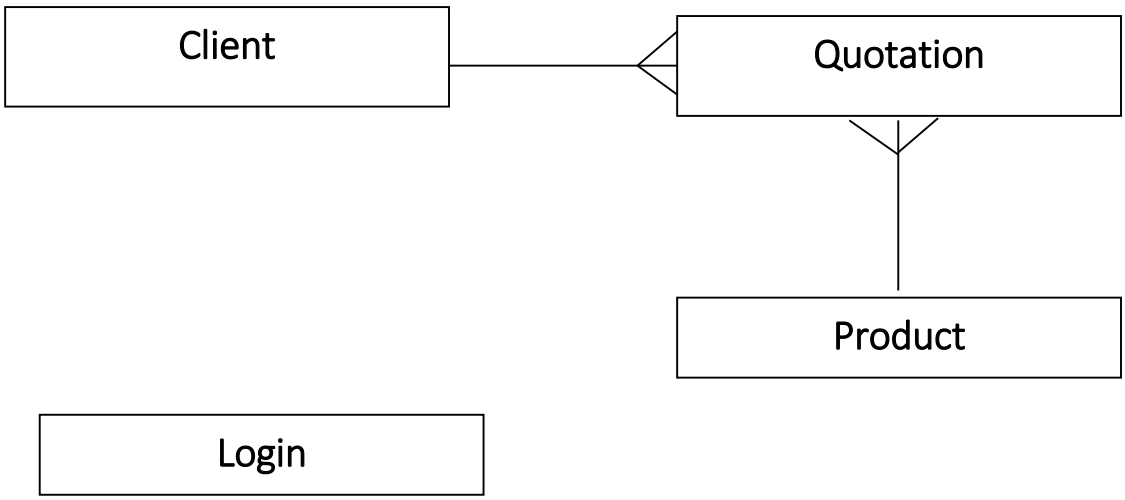


Table Design

Table: Client

Field Name	Data Type	Description
Client_ID	Integer	A unique key to identity the client.
Client_name	String	Name of the Client.
Address	String	Address of the client.
Contact_number	string	Contact number for regular follow ups.
Email	String	The email address of the client.
Organization	String	Organization the client is representing.
Designation	String	Designation of the client in the organization.

Table: Product

Field Name	Data Type	Description
ProductID	String	A unique ID to identify the products.
Product	string	Brand of the product.
Sub-product	string	Type of product.

Table: Quotation

Field Name	Data Type	Description
QutationID	String	A unique key to identify Quotation

ClientID	integer	A unique key to identity the client.
ProductID	string	A unique key to identify product sold.
Product_cost	Currency	The arrival date of the guest.
Transport	Currency	The departure date of the guest.
Custom_VAT	Currency	Government charges
Testing_maintainence	Currency	Post sales services
Quotation_date	Date	Date of the transaction

Table: Quotation

Field Name	**Data Type**	**Description**
QuotationID	String	A unique key to identify the Quotation.
ClientID	Integer	A unique key to identity the client.
SalesID	Integer	A unique key to identify transaction.

Table: User

Field Name	**Data Type**	**Description**
UserName	String	The username of the user.
Password	String	The password for validity

Intended Benefits

- The data can be managed properly due to tables and the relationship between them.
- Product sold can be viewed at a glance in the same table.
- Time saved while entering data and creating backup.
- Details of each section f the quotation viewed easily.

Limitations

System Limitations

- The system is not supported from any online procedure like online payment or online contract.

- The user has to keep the accounting details him/herself since the system only records price of transaction.

- The system does not support credit transaction.

- It can only be used in one computer independently.

Size Estimation of Database

Table: Client

Field Name	Data Type	Size (bytes)
Client_ID	Integer	2
Client_name	String	30
Address	String	30
Contact_number	string	20
Email	String	40
Organization	String	20
Designation	String	20

Total Size (per record) = 162 bytes

Total Records (approximately) = 100

Table: Product

Field Name	Data Type	Size(bytes)
ProductID	String	10
Product	string	15
Sub-product	string	20

Total Size (per record) = 45 bytes

Total Records (approximately) = 50

Table: Quotation

Field Name	Data Type	Size (bytes
QutationID	string	8
ClientID	Number	2
ProductID	Number	2
Product_cost	Currency	2
Transport	Currency	2
Custom_VAT	Currency	2
Testing_maintainence	Currency	2
Sales_date	Date	8

Total Size (per record) = 28 bytes

Total Records (approximately) = 100

Table: User

Field Name	Data Type	Size(bytes)
UserName	String	10
Password	String	10

Total Size (per record) = 20 bytes

Total Records (approximately) = 1

Total Size (of all records) = (162 x 100+ 45 x 50 + 28x100 + 14x100+ 20 x 1)

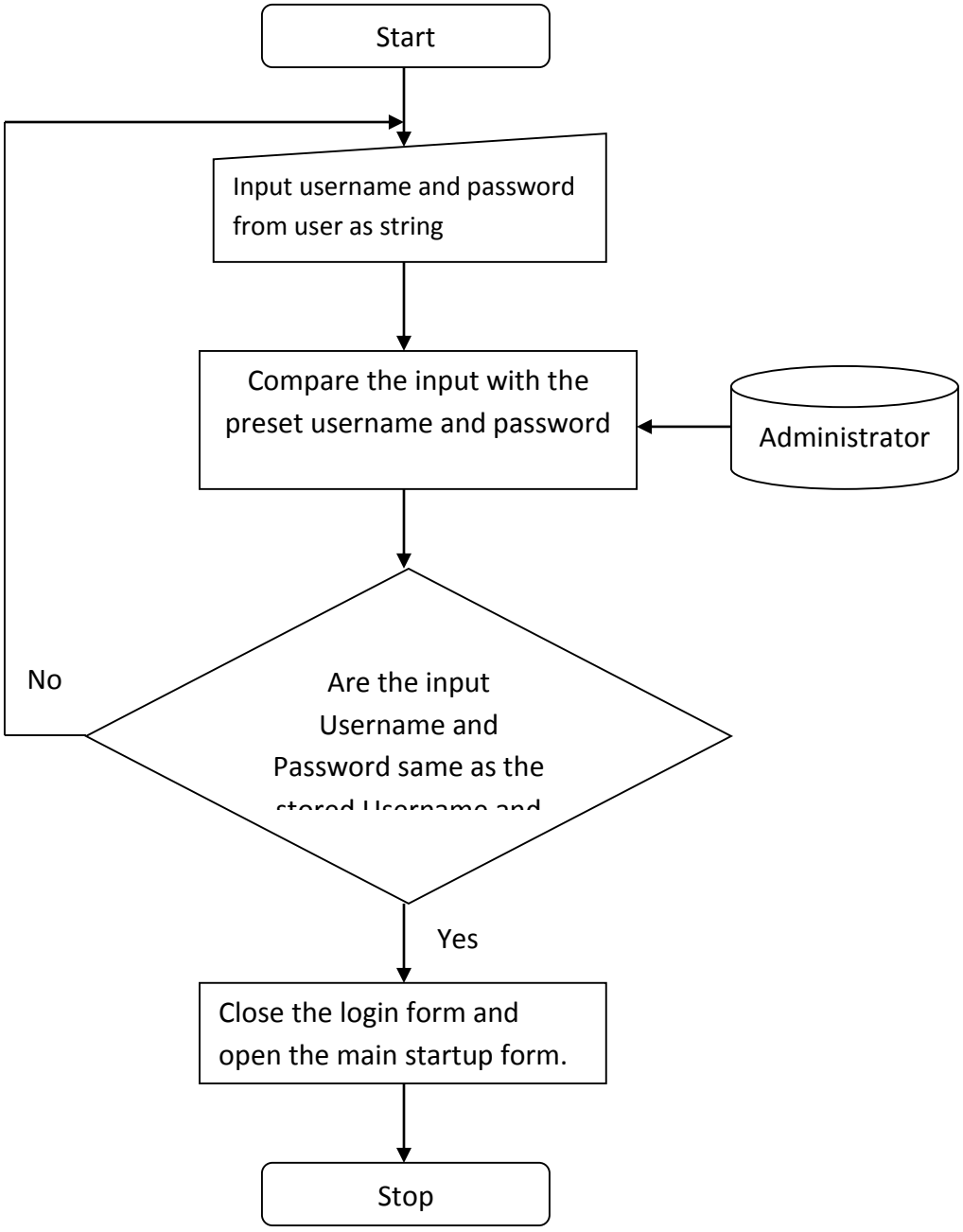
= 22670 bytes

~22 kilobytes (approximately)

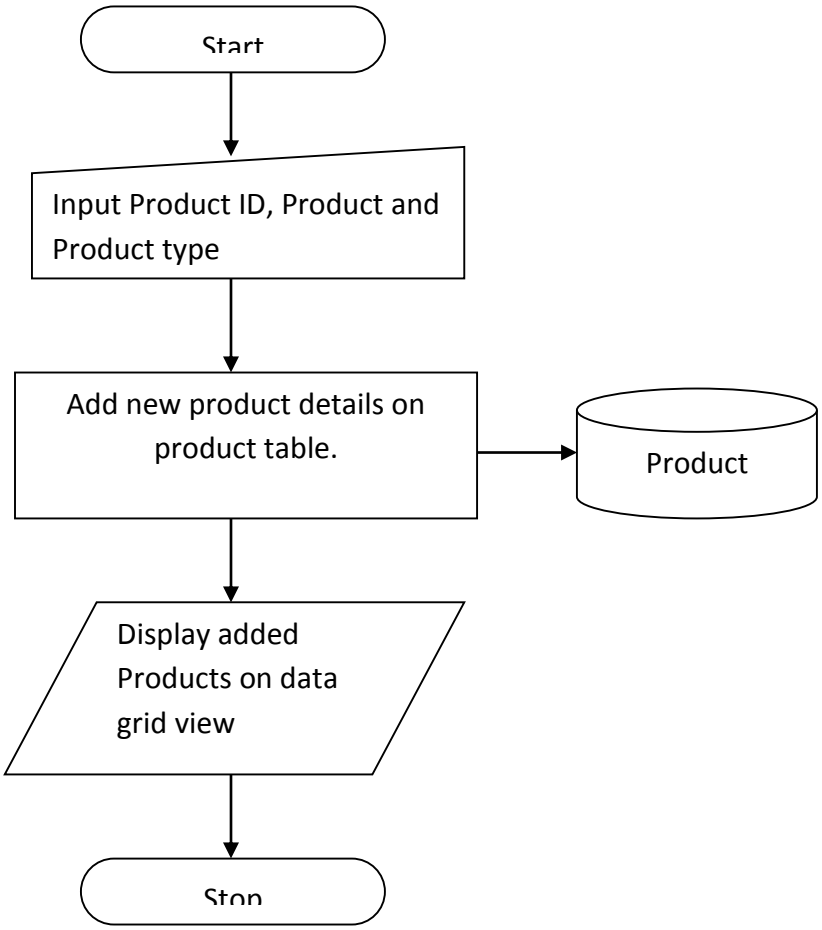
Total size of the database after records have been entered would be approximately **22** Kilobytes. This is the approximated size of the database when 100 clients purchase the products. Assumed there are 50 unique products to choose from. The size of the database keeps on increasing as new clients make the orders or the company decides to increase its contracts and add more products.

Flowcharts

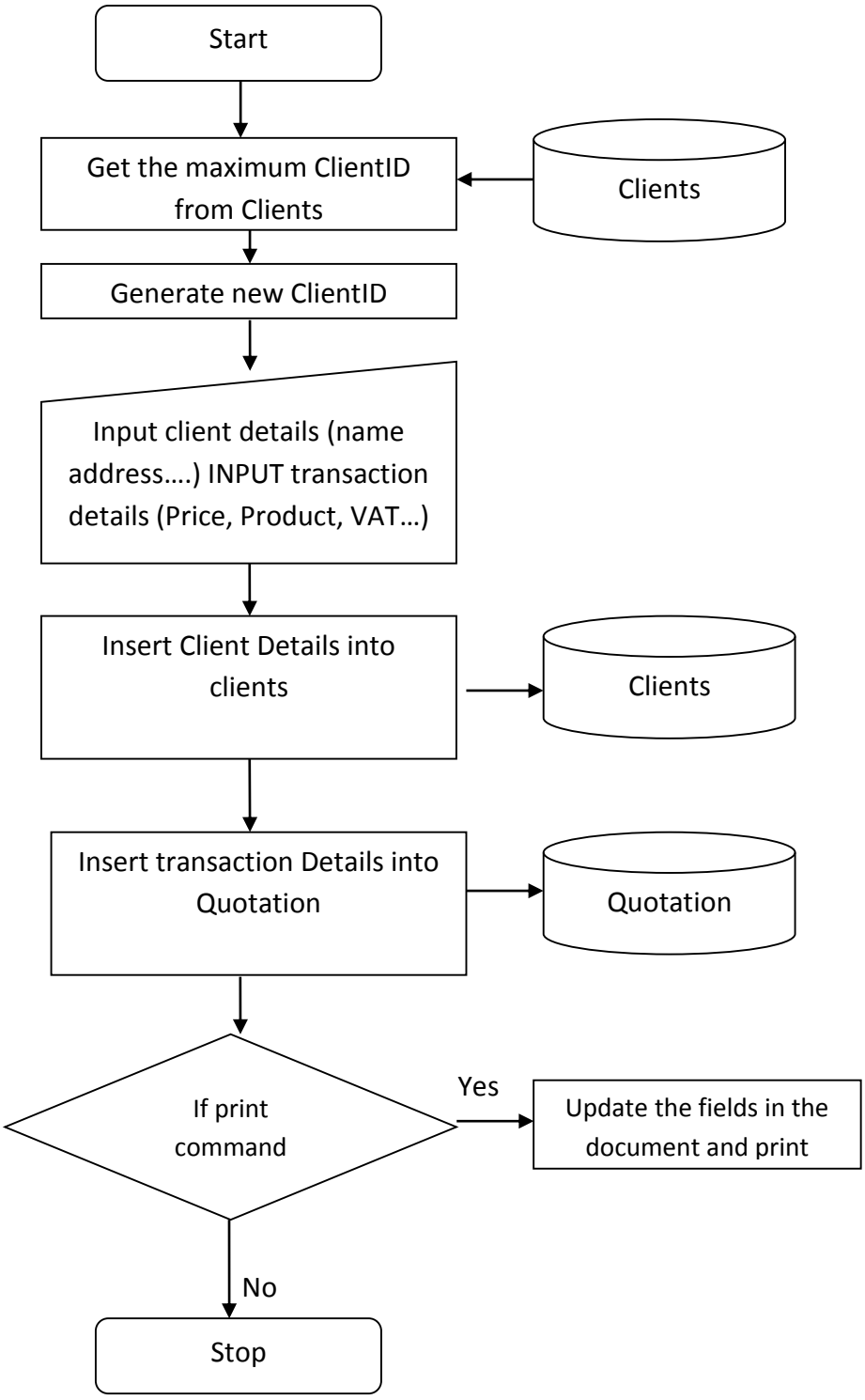
Log In



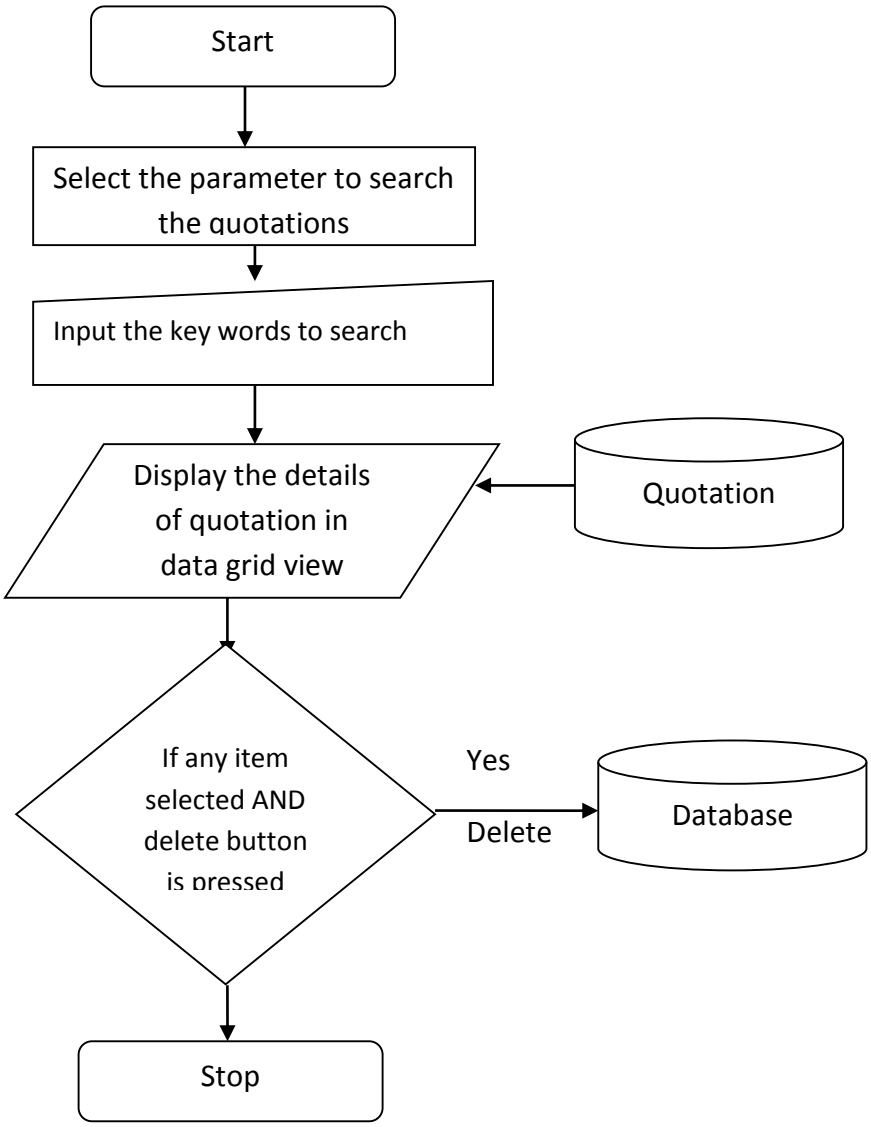
Add New Product (official Data Entry)



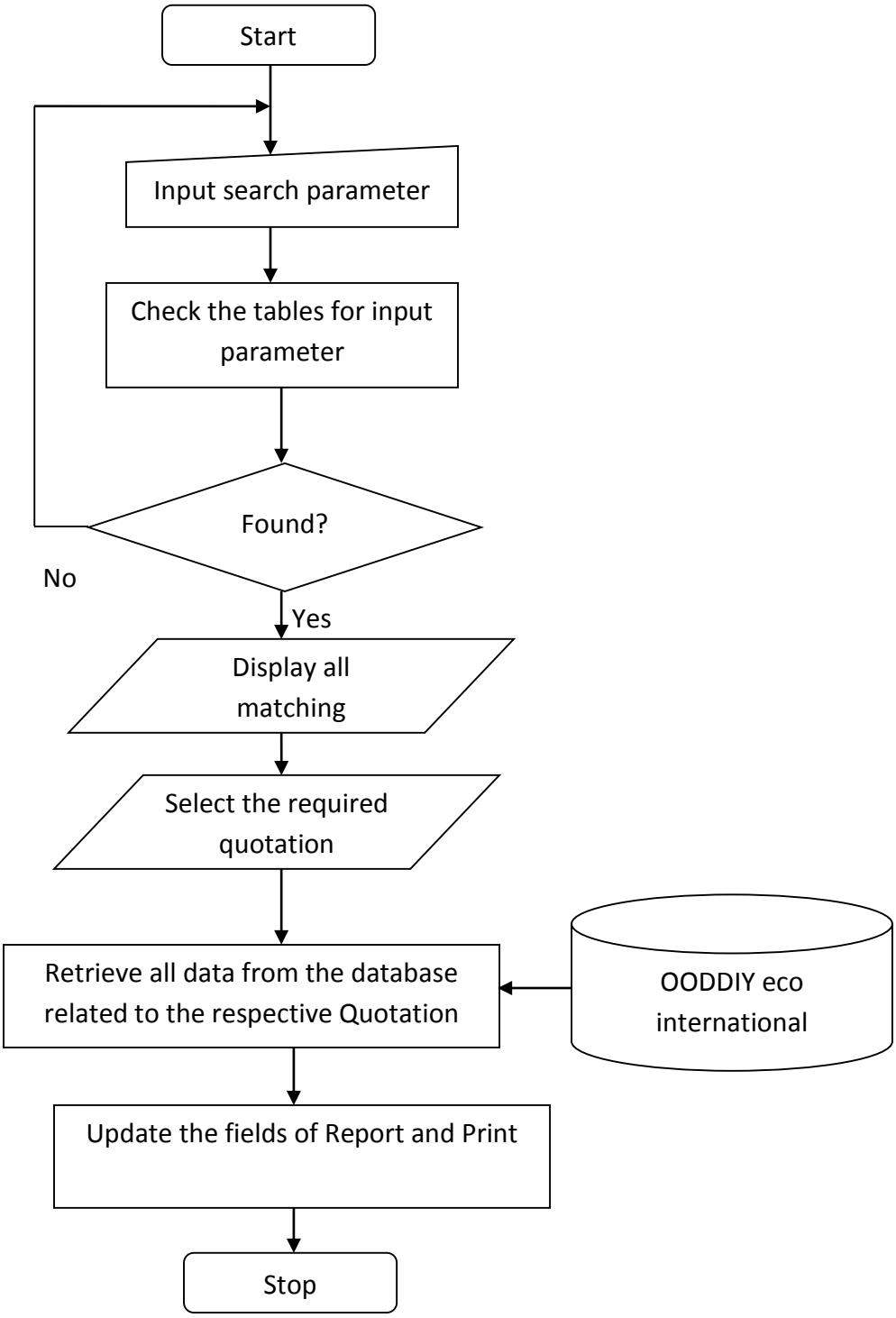
Data Entry and Printing



View/Delete Quotations

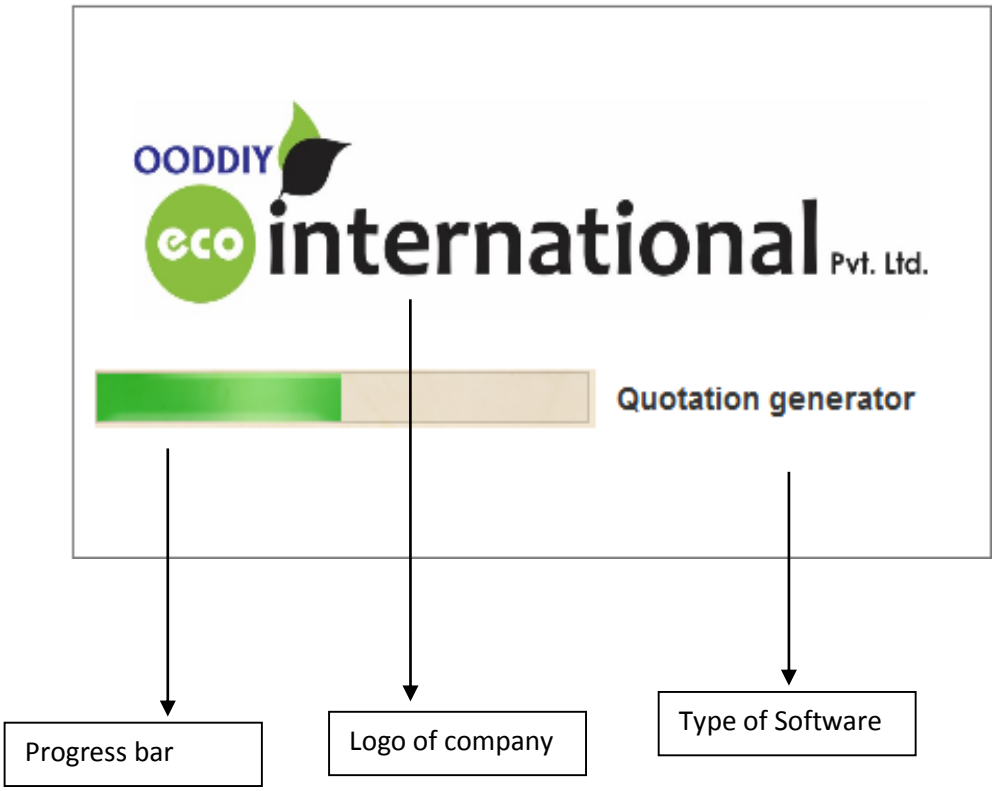


Printing Quotations



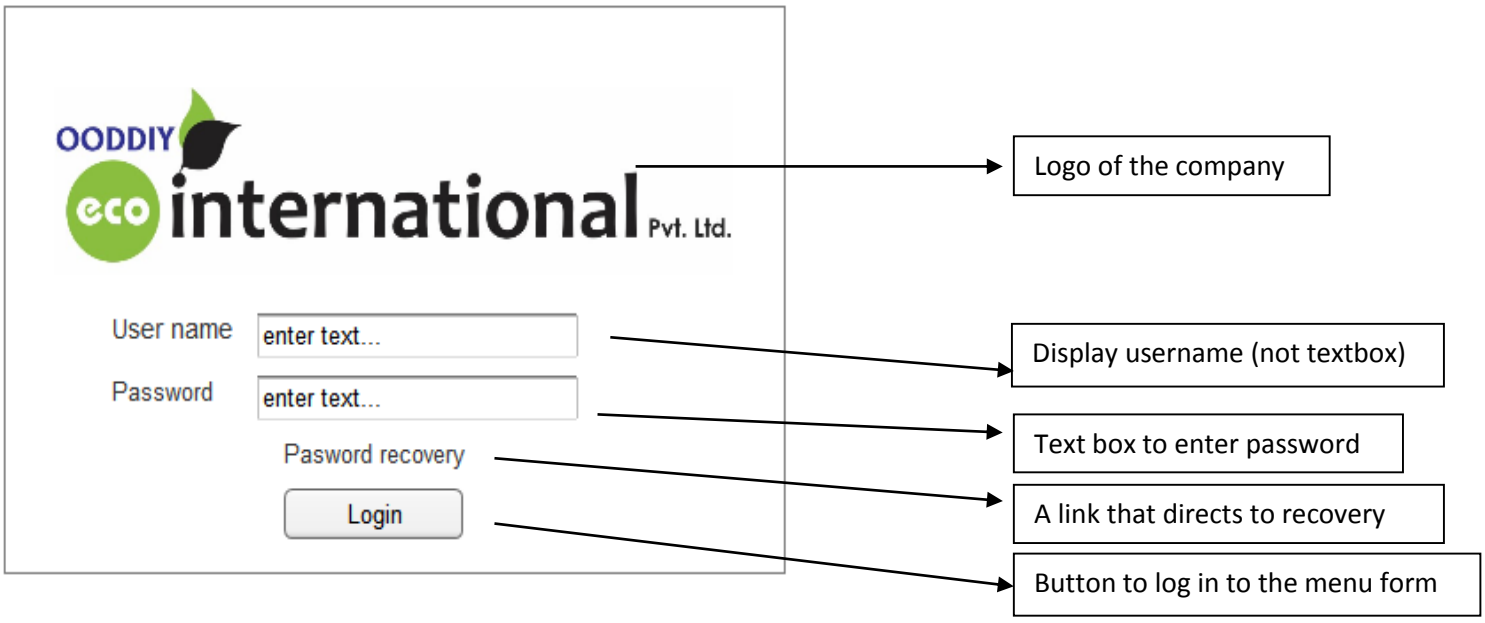
Screen Layout

Splash Screen:



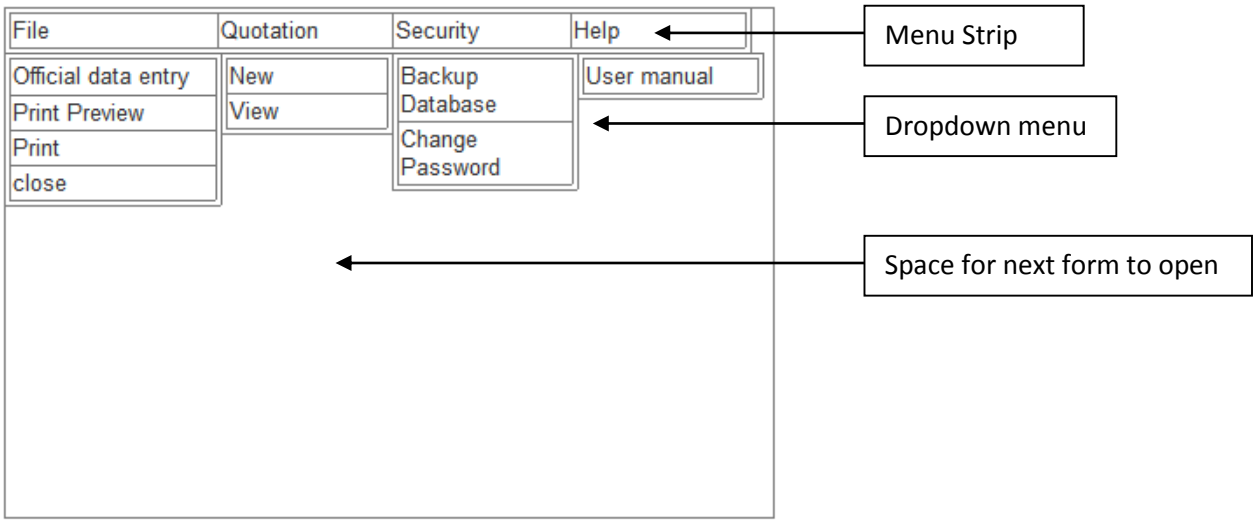
The splash screen is a form initially shown to give the general detail of the company and the software. It makes the Software look professional. The screen appears in the screen for exactly 4s.

Form For Authentication



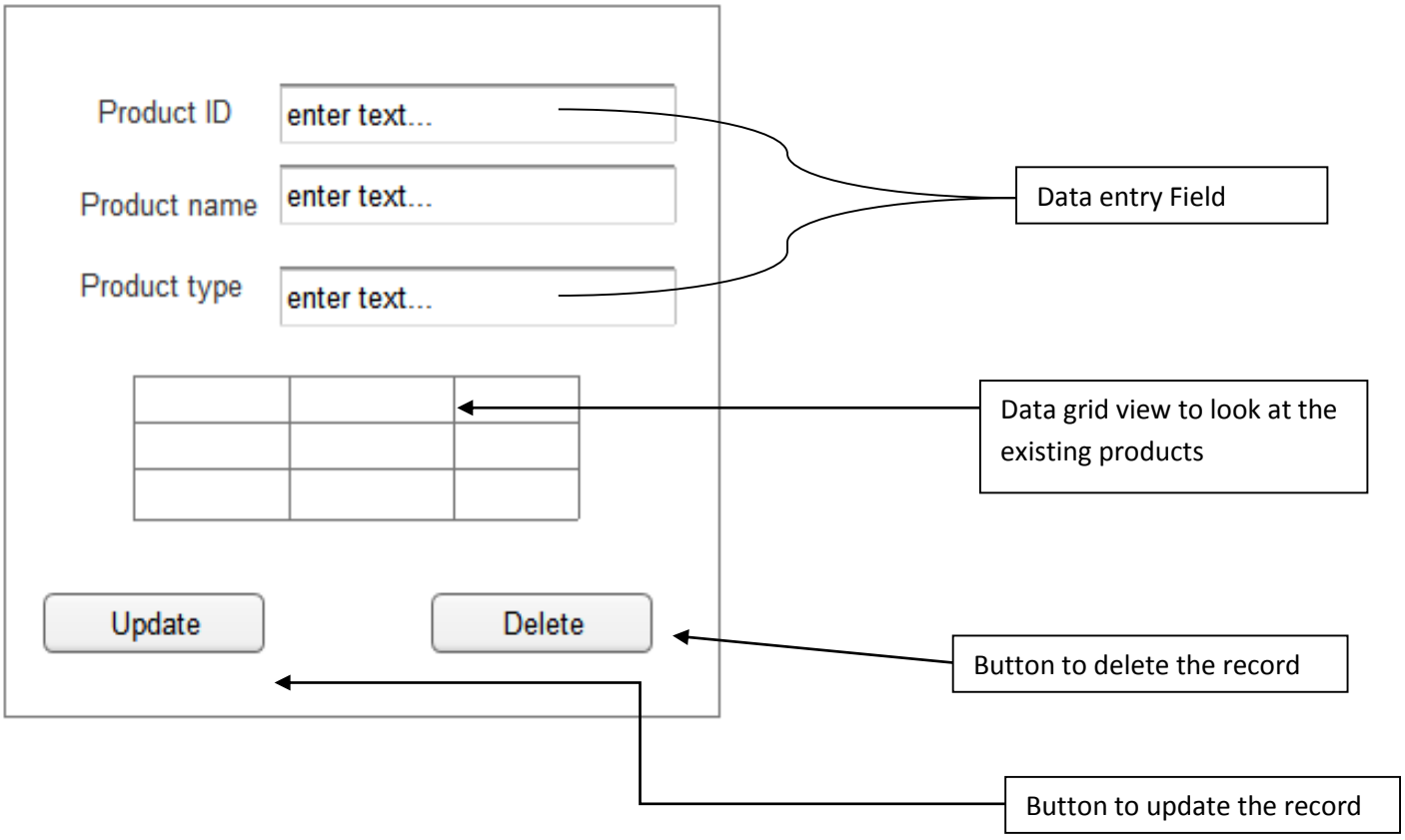
This form will appear just after splash screen. The main purpose is to authenticate the user using the software. Since there is one person hired to do the job there is only one user do there is no textbox for typing username.

Form To View Main Menu:



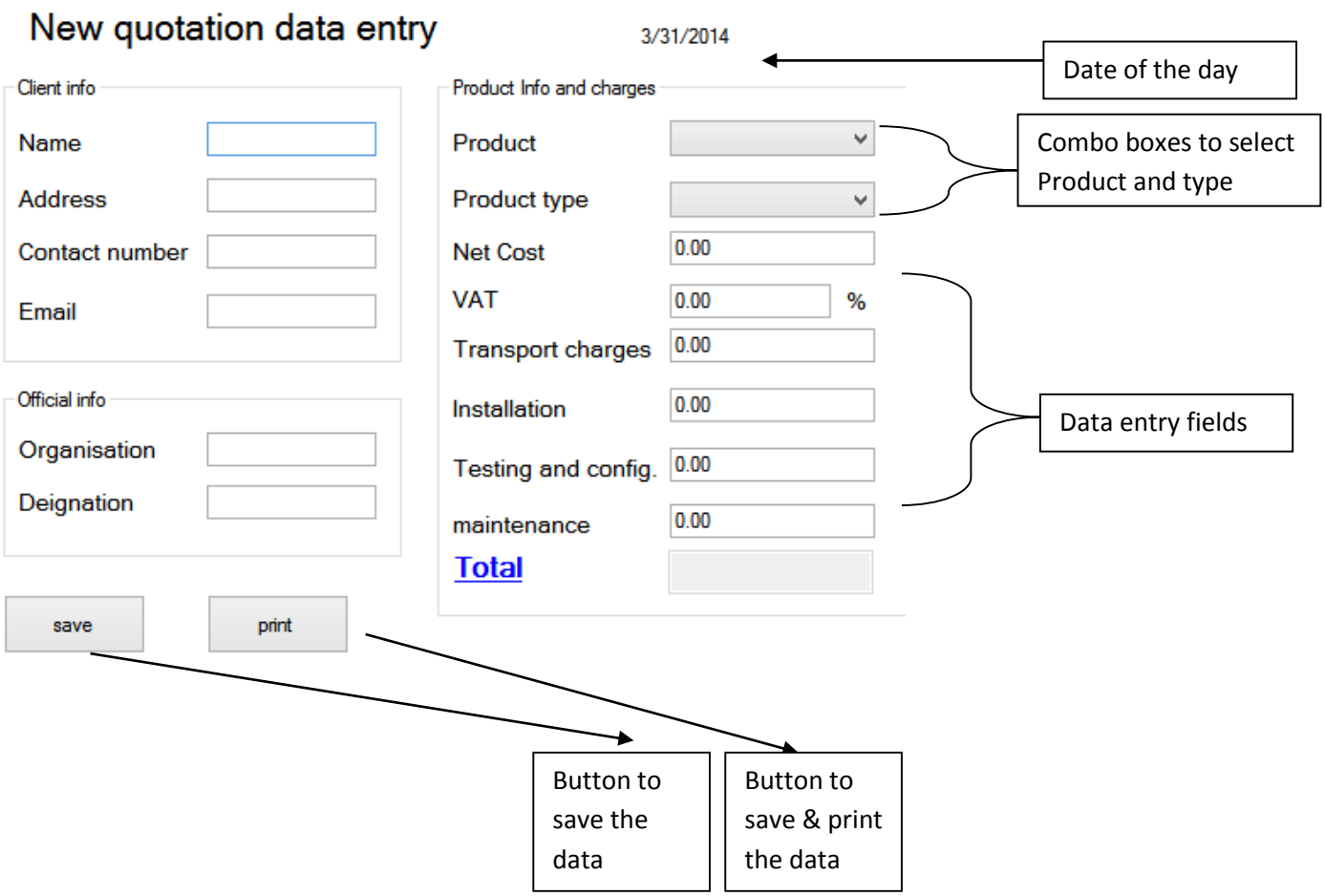
This is the Starting form from which the navigation of all other forms is done. This is the parent form that contains all other forms in it. Every menu strip contains sub menus drop box. By clicking the menu items new form loads on the blank space f the parent form.

Form To Add new Products



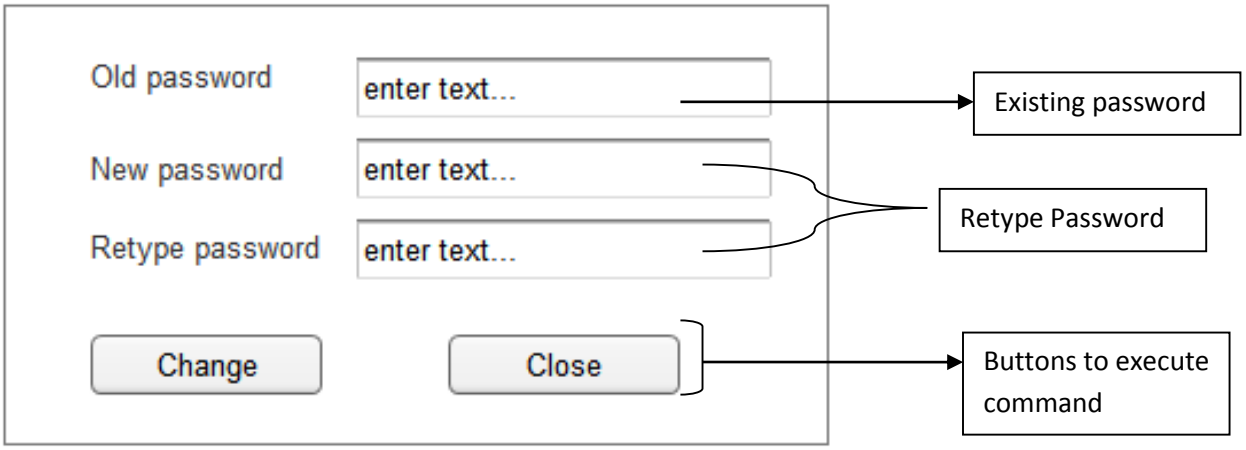
This is the form used when the company decides to add other products. The price of the product is not included as the price changes according to the Exchange rate and is manually entered during the sales.

Form to Add Quotation



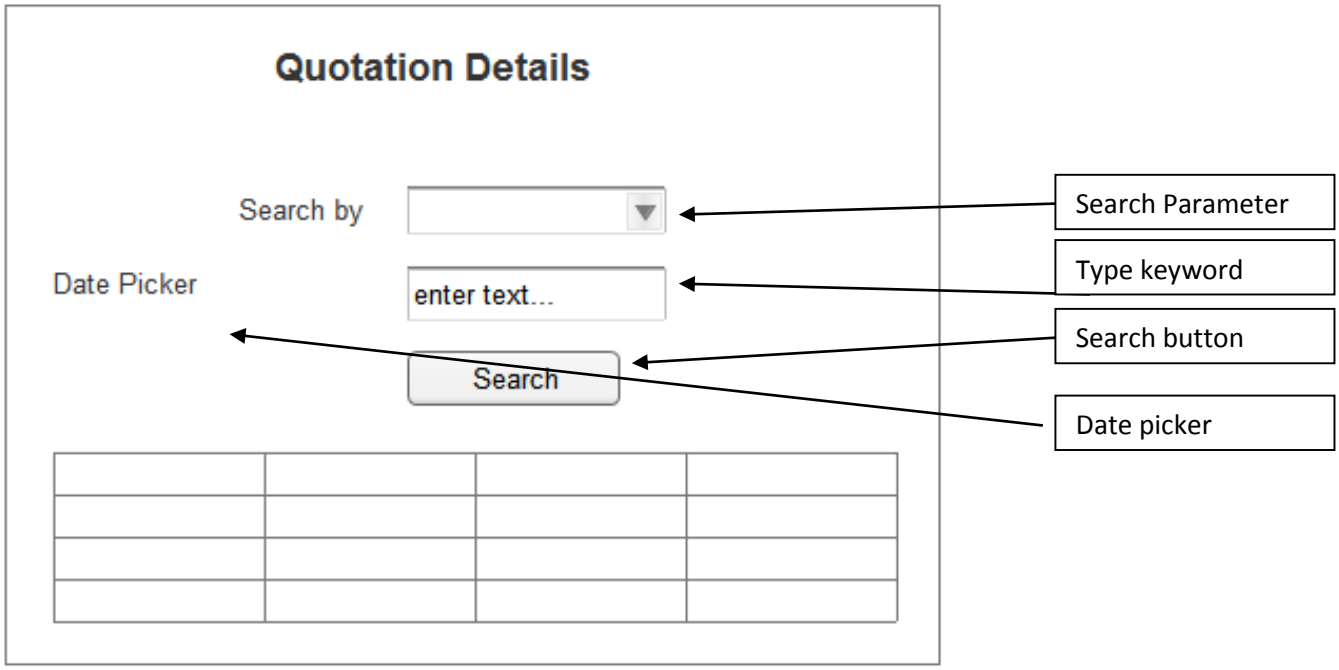
This form is used to enter the data that the quotation requires. It takes all the data necessary and saves, print the quotation which is then given to the client.

Form To change password:



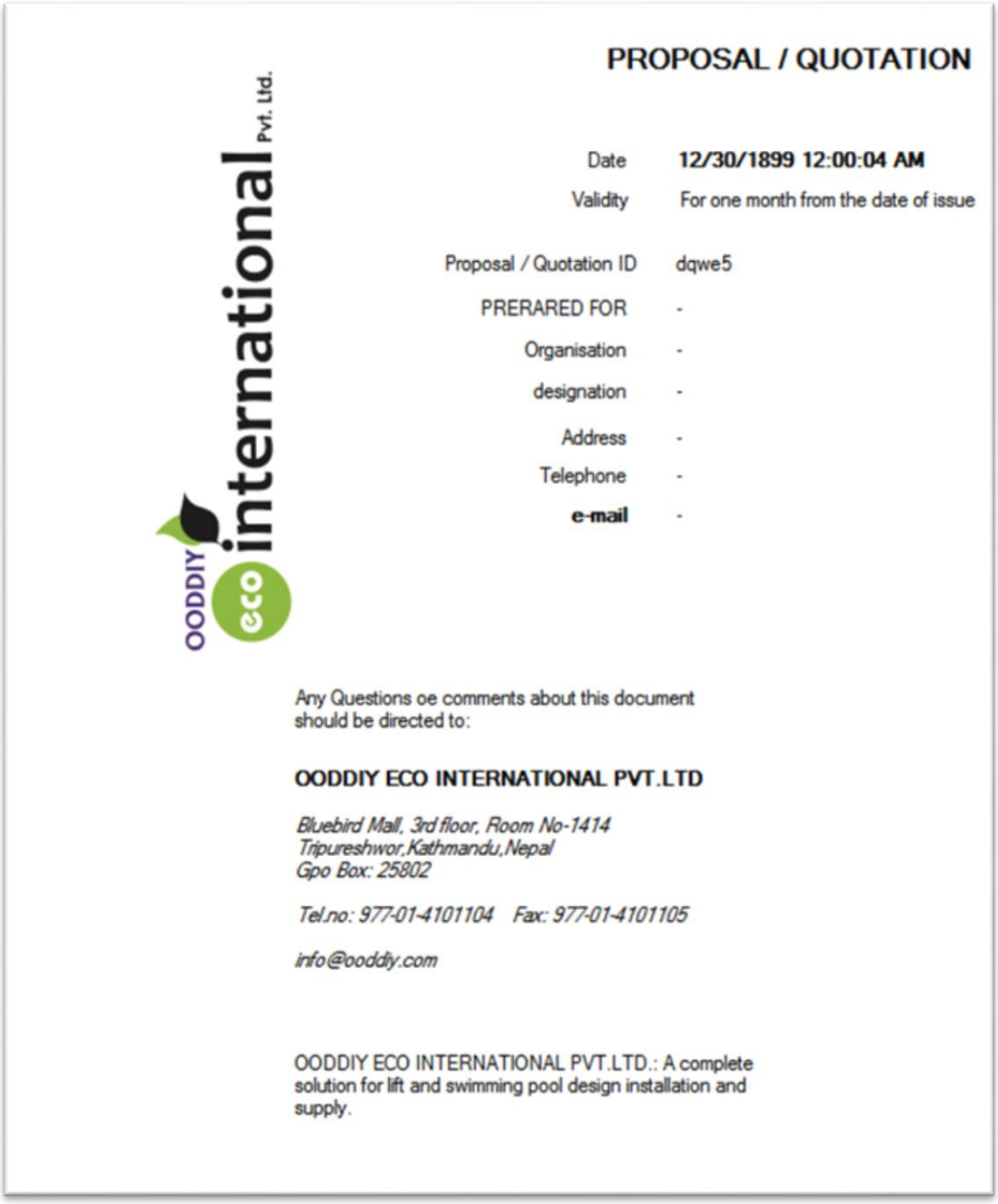
This form allows the user to change the password if the employee is changed or the password is known by the unauthorized personnel.

View Quotation



This form allows the user to view the details of the quotation. The combobox control allows the user to choose a parameter through which the search is carried out. The text box allows typing space to the user and the date picker control allows t search by the date of quotation.

Report layout



PROPOSAL / QUOTATION

OODDIY eco international Pvt. Ltd.

Date	**12/30/1899 12:00:04 AM**
Validity	For one month from the date of issue
Proposal / Quotation ID	dqwe5
PRERARED FOR	-
Organisation	-
designation	-
Address	-
Telephone	-
e-mail	-

Any Questions oe comments about this document should be directed to:

OODDIY ECO INTERNATIONAL PVT.LTD

Bluebird Mall, 3rd floor, Room No-1414
Tripureshwor,Kathmandu,Nepal
Gpo Box: 25802

Tel.no: 977-01-4101104 Fax: 977-01-4101105

info@ooddiy.com

OODDIY ECO INTERNATIONAL PVT.LTD.: A complete solution for lift and swimming pool design installation and supply.

3. Software Development and Programming

Program and Database Specification

The program is a desktop application. The software has been developed in the Integrated Development Environment (IDE) of Visual Studio 2008 using VB.NET programming language. The computer had windows 8 running as the operating system.

The database required to store the data, was created using MS access all the table designing, relationships editing were done through MS access 2007.

Printing of the report was done by Print Form a tool in the visual basic 2008 a .NET component. It prints the form as a page. The labels controls were used to update the text field.

The designing of the mock-up screen were done using “Axure RP pro6.5”.

The report designing, editing of this documentation was done using word processing software Microsoft word 2007

Algorithms

Adding Product details

1. Begin
2. Accept product ID
3. Accept Product name
4. Accept Product type
5. Add the details to database
6. Display the details on the data grid view
7. End

Adding New Quotation

1. Begin
2. Accept client Details
3. Select Product details.
4. Accept Transaction details (cost, transport, installation..)
5. Generate new Quotation ID on the basis of the productID and client ID.
6. Add the details to the database.
7. Add the details to report generating module.
8. End

Viewing Quotation

1. Start
2. Select a parameter to search the details.
3. Add the keyword to search the details
4. Search the details on the database and display on the data grid view
5. Delete button pressed?

   Yes: delete the records.

   No: view the records.
6. End

Print Quotation

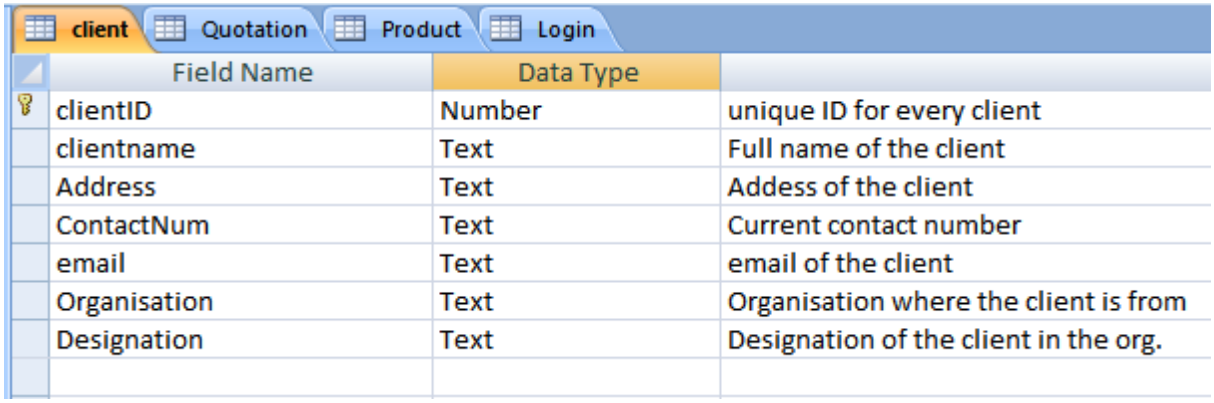
1. Begin
2. Get the details from the data entry form
3. Search in the database for the supplied parameters
4. Are there any details missing?

   Yes: Display “incomplete data”
   No: Go to step 5
5. Display the details on the screen
6. Print Quotation
7. End

Data Layer Tables

Clients

Snapshot of the table in MS Access:



Field Name	Data Type	
clientID	Number	unique ID for every client
clientname	Text	Full name of the client
Address	Text	Addess of the client
ContactNum	Text	Current contact number
email	Text	email of the client
Organisation	Text	Organisation where the client is from
Designation	Text	Designation of the client in the org.

Table Description:

Field Name	**Data Type**	**Field Size (in bytes)**	**Description**
ClientID	Number	Integer (2)	A unique key to identity the Client.
ClientName	Text	String (20)	Name of the Client
Address	Text	String (20)	Address of the client
Contactnum	Text	String (20)	Contact number of the client
Email	Text	String (30)	The email address of the client
Organization	Number	string(20)	Organization client works in
Designation	Number	Byte (1)	Post of the client

ClientID is the primary key.

Example:

clientID = 1

ClientName = Sarun Luitel

Address= New-Baneshwor, Kathmandu

Contactnumber= +977 9840069528

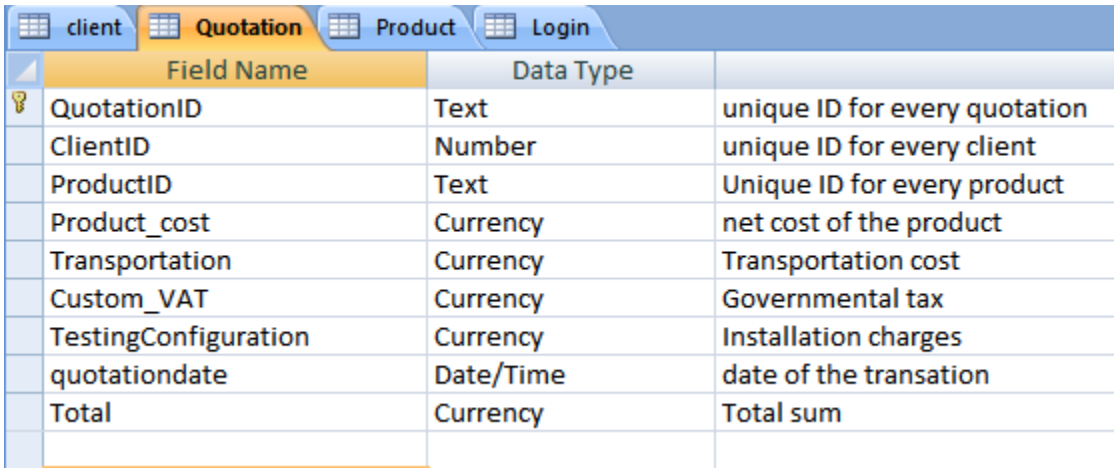
Email = sarunluitel@gmail.com

Organisation =GIHE

Designation = Student

Rooms

Snapshot of the table in Microsoft Office Access:



client | Quotation | Product | Login

Field Name	Data Type	
QuotationID	Text	unique ID for every quotation
ClientID	Number	unique ID for every client
ProductID	Text	Unique ID for every product
Product_cost	Currency	net cost of the product
Transportation	Currency	Transportation cost
Custom_VAT	Currency	Governmental tax
TestingConfiguration	Currency	Installation charges
quotationdate	Date/Time	date of the transation
Total	Currency	Total sum

Table Description:

Field Name	**Data Type**	**Size**	**Description**
QutationID	String	20	A unique key to identify Quotation
ClientID	integer	2	A unique key to identity the client.
ProductID	string	20	A unique key to identify product sold.
Product_cost	Currency	2	The arrival date of the guest.
Transport	Currency	2	The departure date of the guest.
Custom_VAT	Currency	2	Government charges
Testing_maintainence	Currency	2	Post sales services
Quotation_date	Date	4	Date of the transaction

QuotationID is the primary key.

Example:

QutationID - VLKVE12a

ClientID- 1

ProductID -VLKVE

Product_cost- $16,000

Transport- $100

Custom_VAT- 13%

Quotation_date – 2014/3/29

Product

Snapshot of the table in Microsoft Office Access:

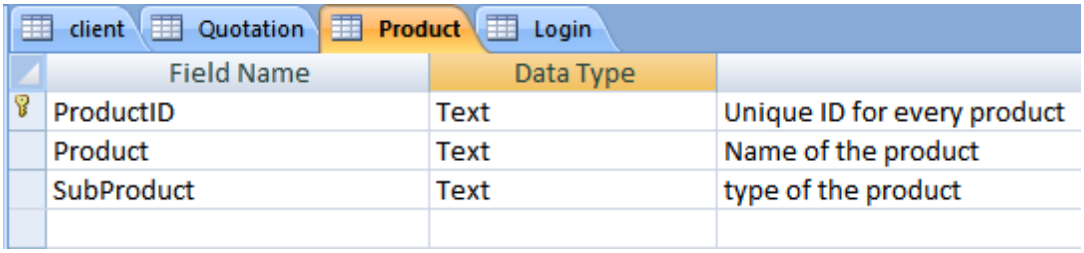


Table Description:

Field Name	**Data Type**	**Field Size (in bytes)**	**Description**
ProductID	String	20	A unique key to identify ProductID
Product	String	40	Name of the product.
Sub Product	String	Date/Time (8)	The arrival date of the guest.

ProductID is the Primary key.

Example:

ProductID- VLKVE

Product- Volks Lift

Sub Product- Home Elevator

Login

Snapshot of the table in Microsoft Office Access:

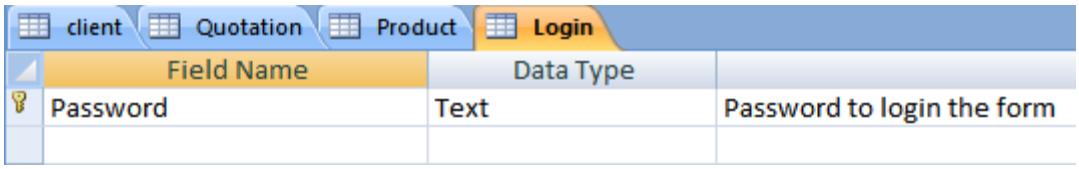


Table Description:

Field Name	Data Type	Field Size (in bytes)	Description
Password	Text	String (35)	Password to login

There is no primary key.

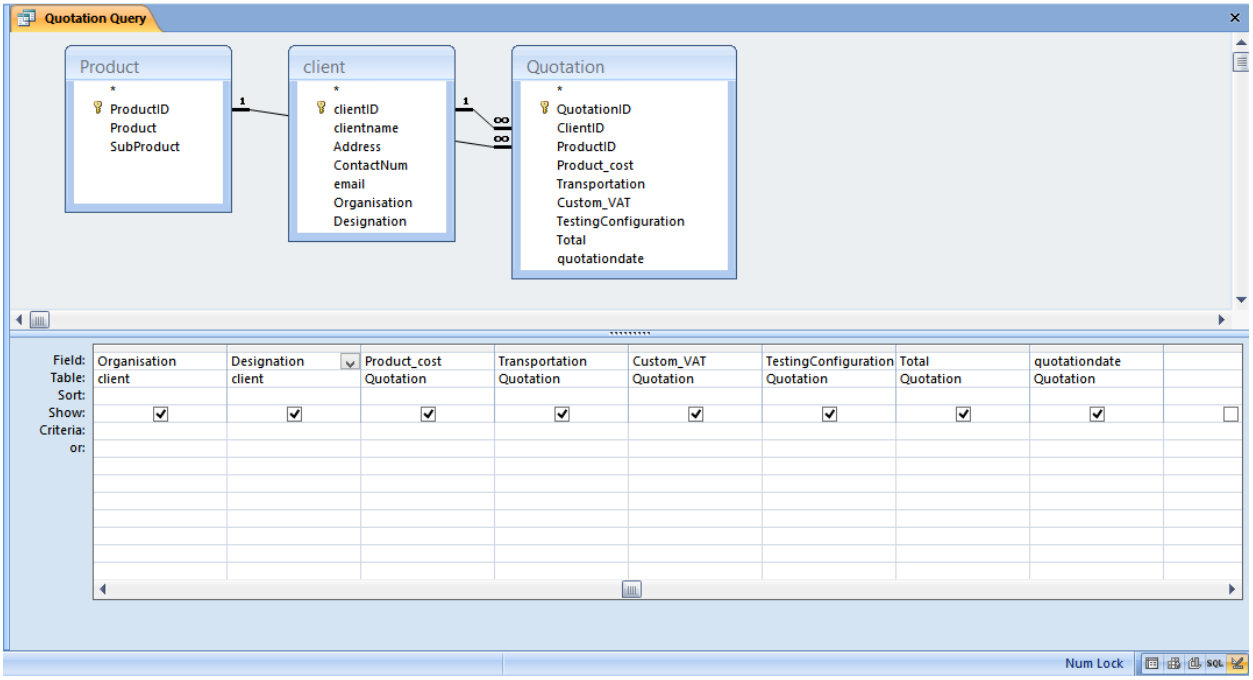
Example:

Password- ECO123@

Query

QueryForReport

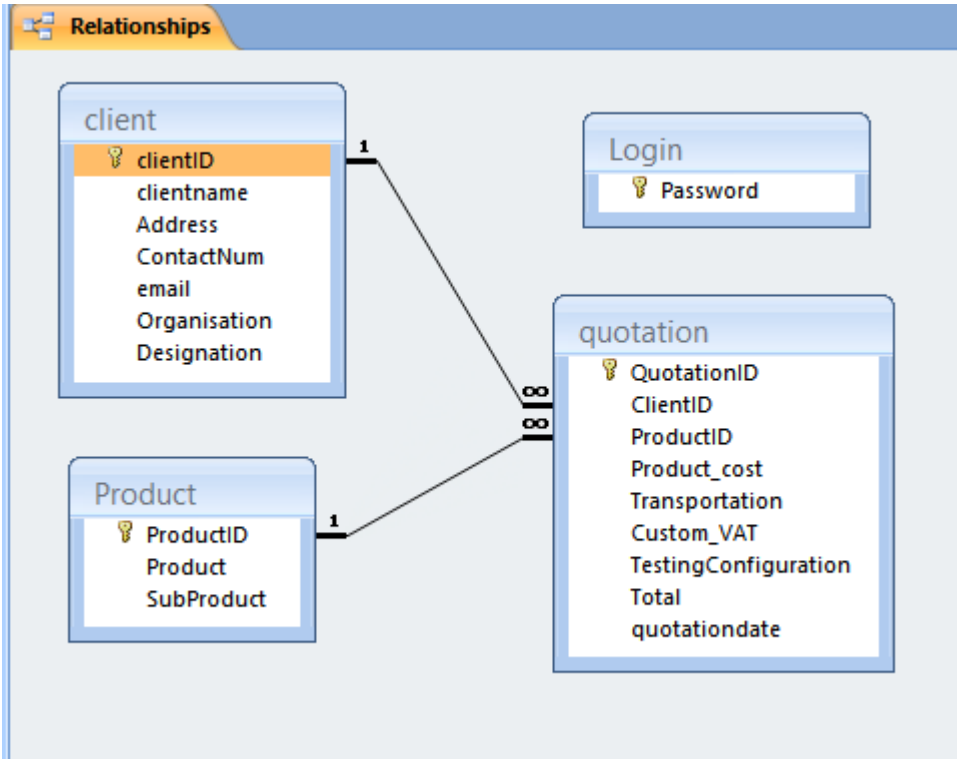
Snapshot of the query in Microsoft Office Access:



Query Description:

- The query QueryForReport extracts data from three tables: Product, Client and Quotation
- QuotationID is extracted from the table Quotation
- ClientID, clientname, Address, Contactnum, email, Organisation, designation is extracted from the Client.
- Product_cost, Transport, Custom_VAT, Testing_configuration, Total, Quotationdate from quotation
- This query is used while generating reports of specified guests.

Relationships Diagram



Variables Used

Variable Name	Variable Type	Used Modules	Description
databasecommand	Object of Class OleDbCommand	Quotation Generator Database Utility Class	It is used to execute a command against the data source.
databaseconnection	Object of Class OleDbConnection	Quotation Generator DB Utility Class	It is used to open connection with the database.
sqlstatement	String	Quotation Generator DB Utility Class	It stores the command to be executed against the data source.
dta	DataTable	Quotation Generator DB Utility Class	It stores the records after the *sqlstatement* has been executed in RetrieveRecords function.
clientname	String	Module, data entry	It stores
ClientID	Integer	Module, data entry	ID of the client
clientaddress	String	Module, data entry	Address of the client

contact	String	Module, data entry	Contact num of the Client
email	String	Module, data entry	Email of the client
clientorg	String	Module, data entry	Organisatin Where the Client works
designation	String	Module, data entry	Client's post.
ProductID	String	Module, data entry	ID of the Prduct
product	String	Module, data entry	Product Company
subproduct	String	Module, data entry	Type of the product.
transport	Integer	Module, data entry	Transportation cost.
installation	Integer	Module, data entry	Installation cost.
testconfig	Integer	Module, data entry	Testing of config charges.
maintenance	Integer	Module, data entry	Maintance cost .

password	String	Module, Login	Password to login.
NetCost	Integer	Module, data entry	Cost of the product
VAT	Bytes	Module, data entry	VAT% given by the government.
quotationid	String	Module, data entry	It stores the quotation id generated
total	Integer	Module, data entry	Sum of all the charges.
Newclient	Integer	Data entry	It stores the new client generated.

Program Listing

Following is the list of codes used to create the program. Some of the features were made using the properties of the respective control itself.

Module

```
Module Modglobal


    Public clientname As String
    Public clientID As Integer
    Public clientaddress As String
    Public contact As String
    Public email As String
    Public designation As String
    Public clientorg As String
    Public orgadd As String
    Public ProductID As String
    Public product As String
    Public subproduct As String
    Public transport As Double
    Public installation As Double
    Public testconfig As Double
    Public maintenance As Double
    Public username As String
    Public password As String
    Public NetCost As Double
    Public VAT As Double
    Public quotationid As String
    Public total As Double

End Module
```

Class

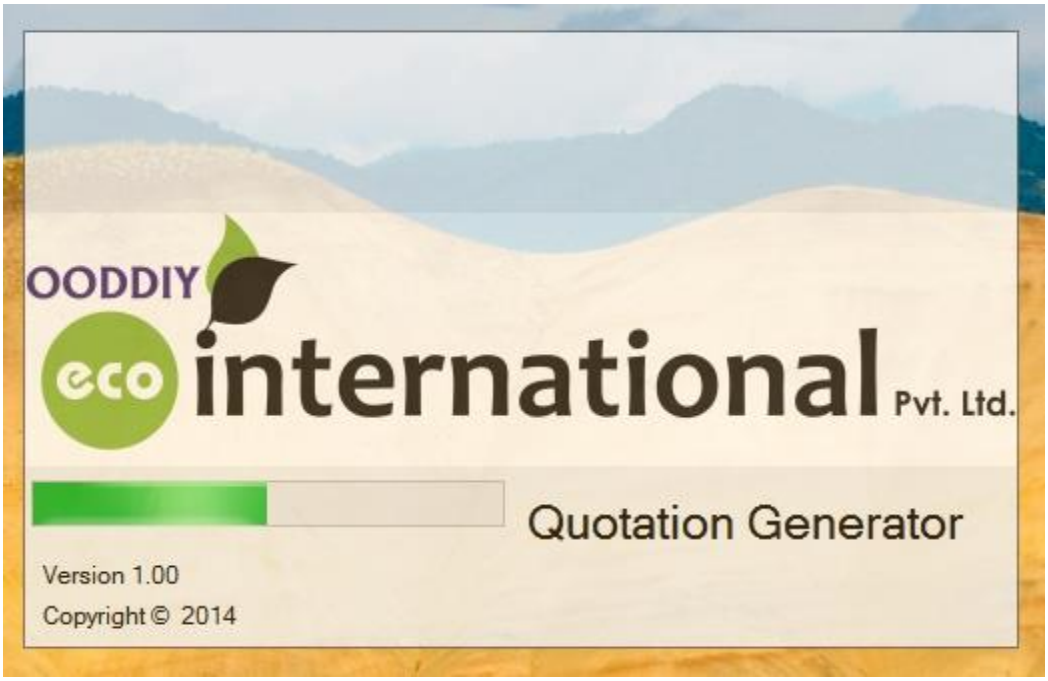
Class- DB utilities

```
      Imports System.Data.OleDb

Public Class DBUtilities

    'declaring globlal variables which are used in the class

    Private DBName As String
    Private StrSql As String

    Dim conn As New OleDbConnection
    Dim cmd As New OleDbCommand

    Public Property SQLStatement() As String
        Get
            Return StrSql
        End Get
        Set(ByVal value As String)
            StrSql = value
        End Set

    End Property



    Private Sub DBConnect()

        'establishing the data base connection

        conn.ConnectionString = "Provider=Microsoft.Ace.Oledb.12.0;Data
Source=" & CurDir() & "\Database1.accdb"
        conn.Open()

    End Sub



    Public Sub DBExecute()
        Call DBConnect()
        Dim cmd As New OleDbCommand()
        cmd.Connection = conn
        cmd.CommandText = StrSql
        cmd.ExecuteNonQuery()
        conn.Close()

    End Sub
```

```
    Public Function GetRecords() As DataTable

   'declaring function to get record in a data table.

        Dim dt As New DataTable()
        Dim cmd As New OleDbCommand()
        Dim da As New OleDbDataAdapter()

        Call DBConnect()
        cmd.Connection = conn
        cmd.CommandText = StrSql

        da.SelectCommand = cmd
        da.Fill(dt)
        conn.Close()
        Return dt

    End Function



    Public Function GetRecordstoCombo(ByVal cbo As ComboBox)
        'declaring a function that returns
        Dim cmd As New OleDbCommand(StrSql)
        Dim reader As OleDbDataReader
        Call DBConnect()
        cmd.Connection = conn
        cbo.Items.Clear()
        reader = cmd.ExecuteReader
        While reader.Read
            cbo.Items.Add(reader.Item(0))
        End While
        conn.Close()
        Return cbo


    End Function


    Public Function GetAValue() As String

        'declaring a function to get a value.

        Dim requiredvalue As String
        Call DBConnect()
        cmd.Connection = conn
        cmd.CommandText = StrSql
        requiredvalue = cmd.ExecuteScalar
        conn.Close()
        Return requiredvalue

    End Function

End Class
```

Forms

Form: Splash Screen




```
Public NotInheritable Class SplashScreen1

    'TODO: This form can easily be set as the splash screen for the
application by going to the "Application" tab
    '  of the Project Designer ("Properties" under the "Project" menu).


    Private Sub SplashScreen1_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        'Set up the dialog text at runtime according to the application's
assembly information.

        'TODO: Customize the application's assembly information in the
"Application" pane of the project
        '  properties dialog (under the "Project" menu).

        'Application title
        If My.Application.Info.Title <> "" Then
            'ApplicationTitle.Text = My.Application.Info.Title
        Else
            'If the application title is missing, use the application name,
without the extension
            ' ApplicationTitle.Text =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.AssemblyName)
        End If
```

```
        'Format the version information using the text set into the Version
control at design time as the
        '  formatting string.  This allows for effective localization if
desired.
        '  Build and revision information could be included by using the
following code and changing the
        '  Version control's designtime text to "Version {0}.{1:00}.{2}.{3}"
or something similar.  See
        '  String.Format() in Help for more information.
        '
        '    Version.Text = System.String.Format(Version.Text,
My.Application.Info.Version.Major, My.Application.Info.Version.Minor,
My.Application.Info.Version.Build, My.Application.Info.Version.Revision)

        Version.Text = System.String.Format(Version.Text,
My.Application.Info.Version.Major, My.Application.Info.Version.Minor)

        'Copyright info
        Copyright.Text = My.Application.Info.Copyright
    End Sub



    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
'counting 4 seonds before startup showsup.
        ProgressBar1.Enabled = True
        ProgressBar1.Value = ProgressBar1.Value + 25
        If ProgressBar1.Value = 100 Then
            frmlogin.Show()
            Me.Close()
        End If

    End Sub



End Class
```

Form: FormLogin


```
Public Class frmlogin

    Private Sub btnlogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnlogin.Click

        Dim dbo As New DBUtilities
        Dim password As String
        dbo.SQLStatement = "select password from login"
        password = dbo.GetAValue()

        'comparing entered password and stored psasword

        If password = txtpass.Text Then
            startup.Show()
            Me.Close()
        Else
            MsgBox("password error")
        End If

    End Sub
```

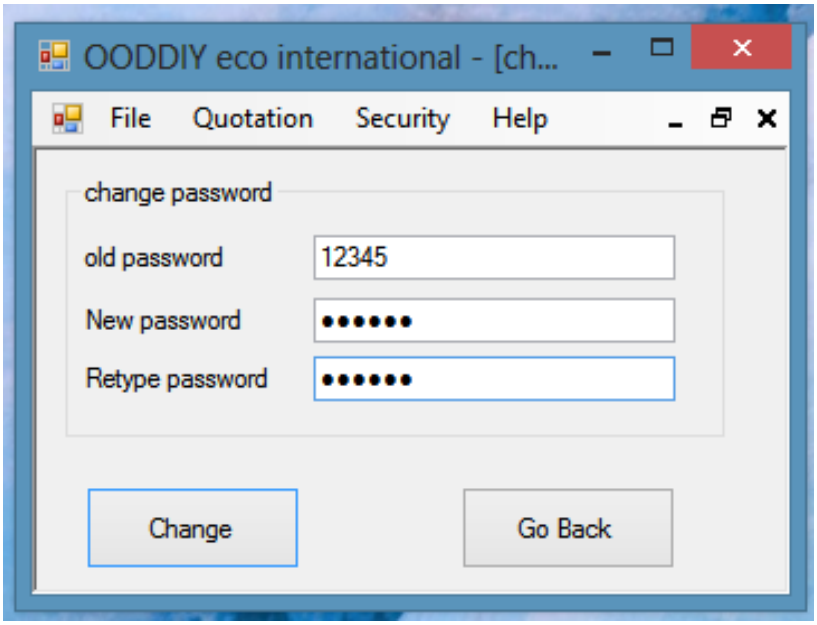
```
    Private Sub LinkLabel1_MouseHover(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LinkLabel1.MouseHover

        'giving hint for initial use

        MsgBox("default Username :ooddiy" & vbCrLf _
        & "default password : ooddiy" & vbCrLf _
        & "Ask authorized person to consult database")


    End Sub

End Class
```

Form: Startup


```
Public Class startup

    Private Sub CloseToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CloseToolStripMenuItem.Click
        Me.Close()

    End Sub


    Private Sub ChangePasswordToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ChangePasswordToolStripMenuItem.Click

        Password_recovery.Show()
        Password_recovery.MdiParent = Me

    End Sub


    Private Sub NewToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NewToolStripMenuItem.Click

        If Dataentry.Visible = True Then
```

```
            Dataentry.BringToFront()

        Else
            Dataentry.MdiParent = Me
            Dataentry.Show()

        End If


    End Sub

    Private Sub ViewToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ViewToolStripMenuItem.Click

        View_quotations.Show()
        View_quotations.MdiParent = Me

    End Sub



    Private Sub OfficialDataEntryToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
OfficialDataEntryToolStripMenuItem.Click

        official_data_entry.Show()
        official_data_entry.MdiParent = Me

    End Sub

Private Sub BackupDatabaseToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
BackupDatabaseToolStripMenuItem.Click

        SaveFileDialog1.ShowDialog()
        System.IO.File.Copy("\Database1.accdb", SaveFileDialog1.FileName)

    End Sub


End Class
```

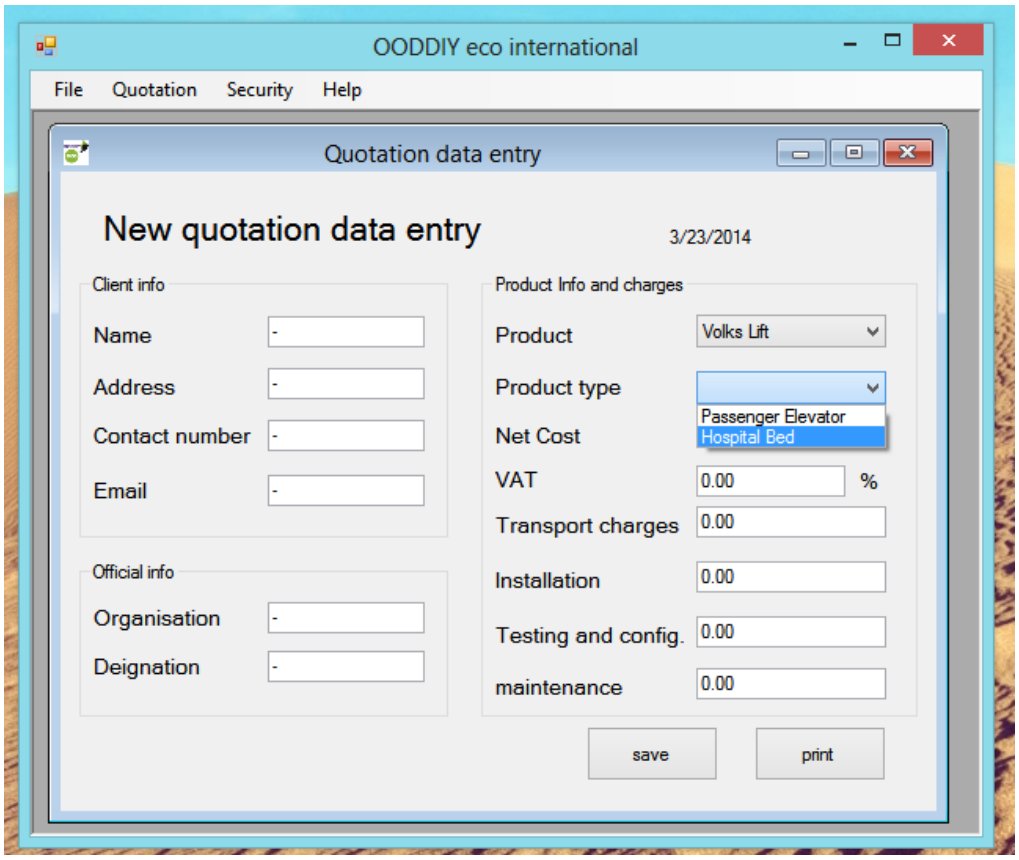
Form: Change Password




```
Public Class Password_recovery

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

        startup.Show()
        Me.Close()

    End Sub


    Private Sub Btnchange_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnchange.Click

        Dim dbo As New DBUtilities
        Dim oldpass As String
        Dim password As String
        dbo.SQLStatement = "select password from login"
        oldpass = dbo.GetAValue()

        ErrorProvider1.Dispose()
```

```
        If txtoldpass.Text = oldpass And txtnewpass.Text = txtrepass.Text And
txtnewpass.Text.Length > 3 Then

            password = txtnewpass.Text

            'deleting the current password from database
            dbo.SQLStatement = "delete * from login"
            dbo.DBExecute()

            'inserting new password into database
            dbo.SQLStatement = "insert into login values ('" & password &
"')"
            dbo.DBExecute()

            MsgBox("password Change Successful")
            txtnewpass.Clear()
            txtrepass.Clear()
            txtoldpass.Clear()


        ElseIf txtnewpass.Text.Length < 3 Then

            ErrorProvider1.SetError(txtnewpass, "Password must atleast be 4
character")
            ErrorProvider1.SetError(txtrepass, "Password must atleast be 4
character")

        ElseIf txtnewpass.Text <> txtrepass.Text Then

            MsgBox("retype new password")
            txtnewpass.Clear()
            txtrepass.Clear()
            txtoldpass.Clear()

        ElseIf txtoldpass.Text <> oldpass Then

            MsgBox("Password Error")
            txtnewpass.Clear()
            txtrepass.Clear()
            txtoldpass.Clear()


        End If


    End Sub


End Class
```

Form: data entry




```
Public Class Dataentry

    Dim prodid As String

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

        lbldate.Text = Today
        Dim Dbo As New DBUtilities
        'adding the values from the database to the combo box.

        Dbo.SQLStatement = "Select distinct Product from Product"
        Dbo.GetRecordstoCombo(cboproduct)

    End Sub
```

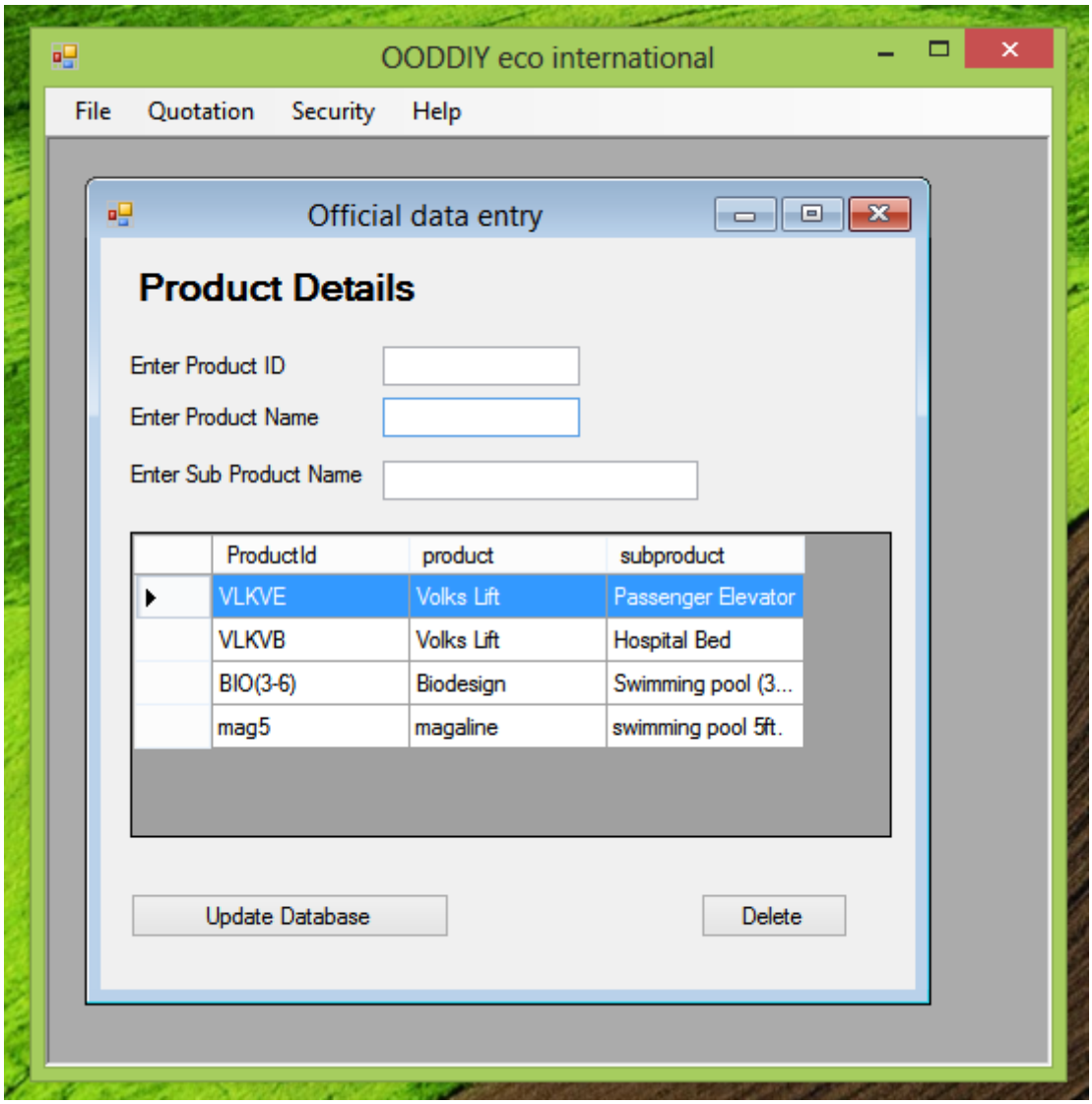
```
    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsave.Click
        'assignning values to variables

        clientname = txtclientname.Text
        clientaddress = txtclientadd.Text
        contact = txtclientcontactno.Text
        email = txtclientemail.Text
        designation = txtclientdesig.Text
        clientorg = txtclientorg.Text
        product = cboproduct.SelectedText
        subproduct = cbosproduct.SelectedText
        transport = txttransport.Text
        installation = txtinstall.Text
        testconfig = txttestconfig.Text
        maintenance = txtmaintain.Text
        NetCost = txtnetcost.Text
        VAT = txtvat.Text
        ProductID = prodid


        Dim dbo As New DBUtilities
        Dim NewIDclient As Integer

        'generating a new client ID

        'calculating the total price
        Dim sum As Double
        transport = txttransport.Text
        installation = txtinstall.Text
        testconfig = txttestconfig.Text
        maintenance = txtmaintain.Text
        NetCost = txtnetcost.Text
        VAT = txtvat.Text
        sum = NetCost + maintenance + transport + installation + testconfig
        total = sum + (sum * VAT) / 100
        txttotal.Text = total

        'selecting existing clientid
        dbo.SQLStatement = "Select Max(ClientID) from Client"
        clientID = dbo.GetAValue()
        NewIDclient = clientID + 1
        quotationid = prodid & NewIDclient

        ErrorProvider1.Dispose()

        If txtclientname.Text = "" Or txtclientname.Text.Length > 20 Or
txtclientname.Text.Length < 4 Then
            ErrorProvider1.SetError(txtclientname, "This field has to be
between 5-20 characters 5-20 characters")

        ElseIf txtclientcontactno.Text = "" Or txtclientcontactno.Text.Length
> 15 Or txtclientcontactno.Text.Length < 7 Then
            ErrorProvider1.SetError(txtclientcontactno, "This field cannot be
blank and between 7-15 characters")
```

```
        ElseIf txtclientemail.Text = "" Or txtclientemail.Text.Length > 30 Or
txtclientemail.Text.Length < 4 Then
            ErrorProvider1.SetError(txtclientemail, "This field has to be
between 5-30 characters")

        ElseIf cboproduct.Text = "" Then
            ErrorProvider1.SetError(cboproduct, "This field cannot be blank")

        ElseIf cbosproduct.Text = "" Then
            ErrorProvider1.SetError(cbosproduct, "This field cannot be
blank")

        ElseIf txtnetcost.Text < 0 Or IsNumeric(txtnetcost.Text) = False Then
            ErrorProvider1.SetError(txtnetcost, "Only positive numbers")

        ElseIf txtinstall.Text < 0 Or IsNumeric(txtinstall.Text) = False Then
            ErrorProvider1.SetError(txtinstall, "Only positive numbers")

        ElseIf txtmaintain.Text < 0 Or IsNumeric(txtmaintain.Text) = False
Then
            ErrorProvider1.SetError(txtmaintain, "Only positive numbers")


        ElseIf txttestconfig.Text < 0 Or IsNumeric(txttestconfig.Text) =
False Then
            ErrorProvider1.SetError(txttestconfig, "Only positive numbers")

        ElseIf txttransport.Text < 0 Or IsNumeric(txttransport.Text) = False
Then
            ErrorProvider1.SetError(txttransport, "Only positive numbers")

        ElseIf txtvat.Text < 0 Or IsNumeric(txtvat.Text) = False Then
            ErrorProvider1.SetError(txtvat, "Only positive numbers")
        Else

            dbo.SQLStatement = "Insert into Client Values(" & NewIDclient &
",'" & clientname & "','" & clientaddress & "','" & contact & "','" & email &
"','" & clientorg & "','" & designation & "')"
            dbo.DBExecute()

            dbo.SQLStatement = "Insert into Quotation Values ('" &
quotationid & "'," & NewIDclient & ",'" & ProductID & "'," & NetCost & "," &
transport & "," & VAT & "," & testconfig & "," & total & "," & lbldate.Text &
")"

            dbo.DBExecute()
            MsgBox("Data addition successful")


            txtclientname.Clear()
            txtclientadd.Clear()
            txtclientcontactno.Clear()
            txtclientemail.Clear()
            txtclientdesig.Clear()
            txtclientorg.Clear()
            txttransport.Text = 0.0
            txtinstall.Text = 0.0
```

```
            txttestconfig.Text = 0.0
            txtmaintain.Text = 0.0
            txtnetcost.Text = 0.0
            txtvat.Text = 0.0
            txttotal.Text = ""
            cboproduct.SelectedIndex = 0
            cbosproduct.SelectedIndex = 0
        End If
    End Sub


    Private Sub cboproduct_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboproduct.SelectedIndexChanged

        cbosproduct.Text = ""
        Dim dbo As New DBUtilities
        dbo.SQLStatement = "Select SubProduct from Product WHERE Product='" &
cboproduct.Text & "'"
        dbo.GetRecordstoCombo(cbosproduct)

    End Sub


    Private Sub cbosproduct_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cbosproduct.SelectedIndexChanged
        'taking product id to create a Qutation id

        Dim dbo As New DBUtilities
        dbo.SQLStatement = "Select ProductID from Product WHERE Product='" &
cboproduct.Text & "' AND SubProduct='" & cbosproduct.Text & "'"
        prodid = dbo.GetAValue()

    End Sub


    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked
        Dim sum As Double


        transport = txttransport.Text
        installation = txtinstall.Text
        testconfig = txttestconfig.Text
        maintenance = txtmaintain.Text
        NetCost = txtnetcost.Text
        VAT = txtvat.Text

        'calculating the total price, adding all the expenses

        sum = NetCost + maintenance + transport + installation + testconfig
        total = sum + (sum * VAT) / 100
        txttotal.Text = total

    End Sub
```

```
    Private Sub btnprint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnprint.Click
        ErrorProvider1.Dispose()
        If txtclientemail.Text = "" Or txtclientemail.Text.Length > 30 Or
txtclientemail.Text.Length < 4 Then
            ErrorProvider1.SetError(txtclientemail, "This field has to be
between 5-30 characters")
        Else

            txtclientname.Clear()
            txtclientadd.Clear()
            txtclientcontactno.Clear()
            txtclientemail.Clear()
            txtclientdesig.Clear()
            txtclientorg.Clear()
            txttransport.Text = 0.0
            txtinstall.Text = 0.0
            txttestconfig.Text = 0.0
            txtmaintain.Text = 0.0
            txtnetcost.Text = 0.0
            txtvat.Text = 0.0
            txttotal.Text = ""
        End If
    End Sub

End Class
```

Official Data Entry

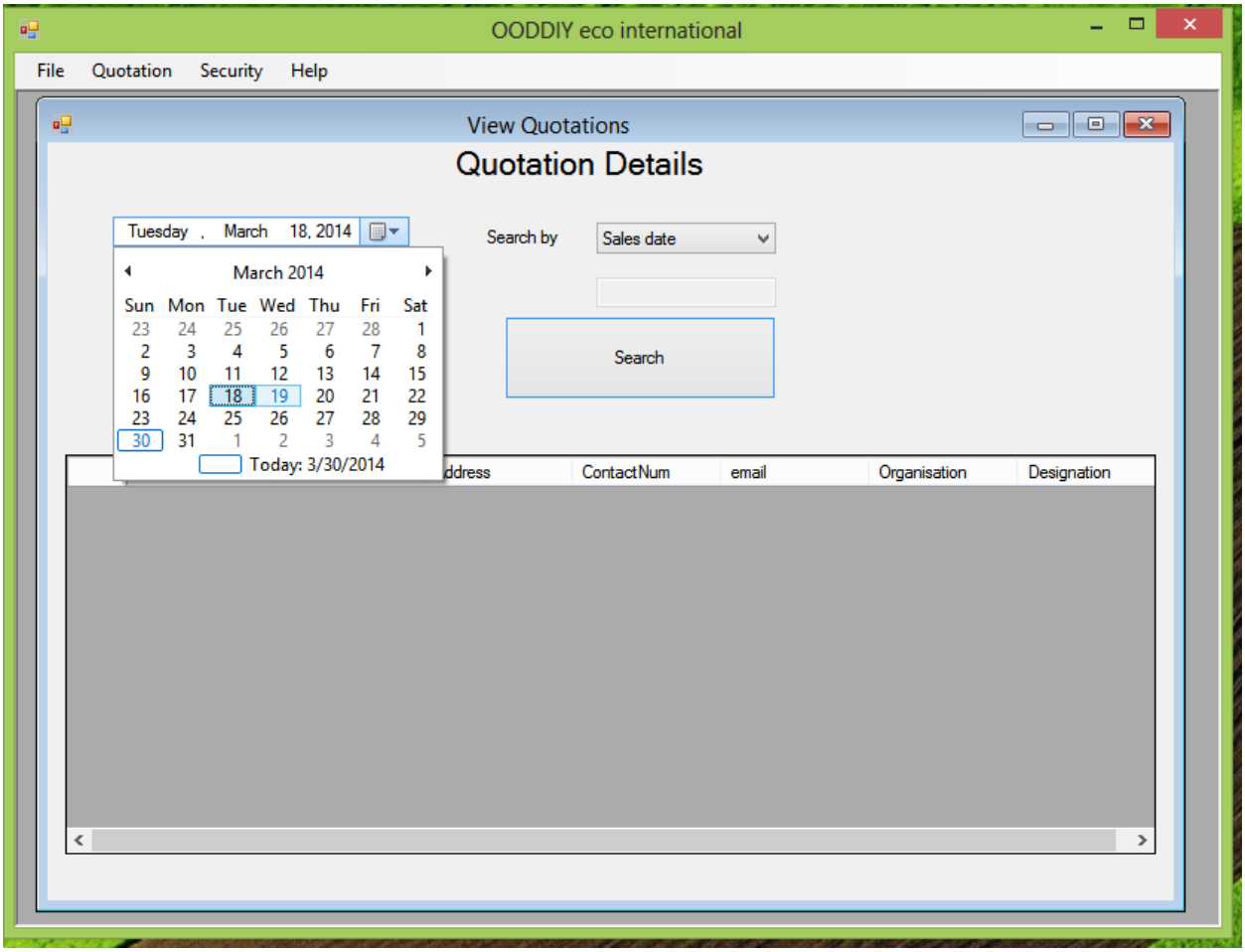



```
Public Class official_data_entry
    Public clickedid As String

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

        Call UpdateGrid()

    End Sub


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

```
        ErrorProvider1.Dispose()
        If txtproductID.Text = "" Then
            ErrorProvider1.SetError(txtproductID, "This field cannot be
blank")
        ElseIf txtproduct.Text = "" Then
            ErrorProvider1.SetError(txtproduct, "This field cannot be left
blank")
        ElseIf txtsubproduct.Text = "" Then
            ErrorProvider1.SetError(txtsubproduct, "This field cannot be left
blank")
        Else
            Dim ProductName, Subproductname As String
            Dim dbo As New DBUtilities
            ProductID = txtproductID.Text
            ProductName = txtproduct.Text
            Subproductname = txtsubproduct.Text
            dbo.SQLStatement = "Insert into product values('" & ProductID &
"','" & ProductName & "','" & Subproductname & "')"
            dbo.DBExecute()
            Call UpdateGrid()
            txtproduct.Text = ""
            txtproductID.Text = ""
            txtsubproduct.Text = ""

        End If



    End Sub

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
gridview.CellContentClick
        'clickedid = DataGridView1.Item(0, DataGridView1.CurrentRow)


    End Sub
    Public Sub UpdateGrid()

        Dim dbo As New DBUtilities
        Dim dta As New DataTable
        dbo.SQLStatement = "Select ProductId,product,subproduct from product"
        dta = dbo.GetRecords
        gridview.DataSource = dta

    End Sub


    Private Sub btndelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndelete.Click

        Dim j As String
        Dim dbo As New DBUtilities

        'prompt to be sure to delete the record
```

```
        If MsgBox("Delete selected record", MsgBoxStyle.YesNo) =
MsgBoxResult.Yes Then

            j = gridview.SelectedCells.Item(0).Value
            dbo.SQLStatement = "delete from product where productID=" & "'" &
j & "'"
            dbo.DBExecute()

            MsgBox("data deleted")

            Call UpdateGrid()
        End If
    End Sub

End Class
```

Form: View Quotation

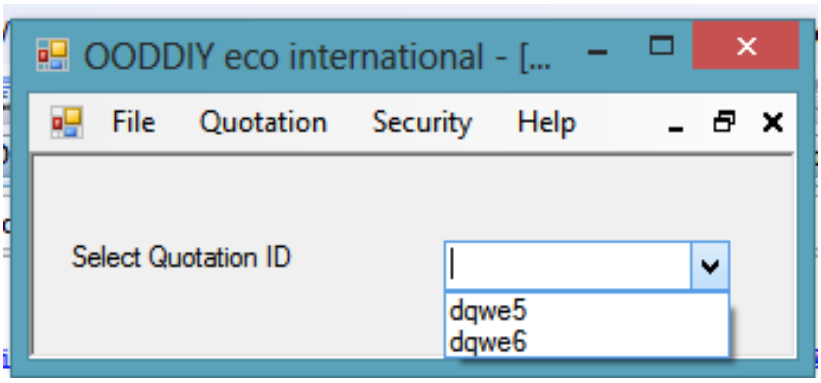



```
Public Class View_quotations

    Public Sub UpdateGrid()

        Dim dbo As New DBUtilities
        Dim dtaclient As New DataTable
        Dim dtaproduct As New DataTable
        Dim dtaquot As New DataTable
        'adding the values into the combo box.

        dbo.SQLStatement = "select * from [Quotation Query]"
        DataGridView.DataSource = dbo.GetRecords

    End Sub
```

```
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs)
        'only enable the textbox when the parameter is selected

        txtsearch.Enabled = True

    End Sub


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

        Dim db As New DBUtilities

        cboparameter.Items.Add("Quotation ID")
        cboparameter.Items.Add("Client ID")
        cboparameter.Items.Add("Client Name")
        cboparameter.Items.Add("Organisation")
        cboparameter.Items.Add("Product ID")
        cboparameter.Items.Add("Sales date")


        Dim dbo As New DBUtilities
        Dim Dt As New DataTable
        Db.SQLStatement = "Select product from product"

        Call UpdateGrid()
        'dbo.GetRecordstoCombo(ComboBox1)

        txtsearch.Enabled = False

    End Sub


    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsearch.Click
        Dim db As New DBUtilities

        'selecting the parameters from the combobox and searching in the
query.

        If cboparameter.Text = "Quotation ID" Then
            db.SQLStatement = "Select * from [Quotation query] where
quotationID=" & txtsearch.Text & "'"
            DataGridView.DataSource = db.GetRecords

        ElseIf cboparameter.Text = "Client ID" Then
            db.SQLStatement = "Select * from [Quotation query] where
ClientID=" & Val(txtsearch.Text)
            DataGridView.DataSource = db.GetRecords

        ElseIf cboparameter.Text = "Client Name" Then
            db.SQLStatement = "Select * from [Quotation query] where
Clientname='" & txtsearch.Text & "'"
            DataGridView.DataSource = db.GetRecords

        ElseIf cboparameter.Text = "Organisation" Then
```

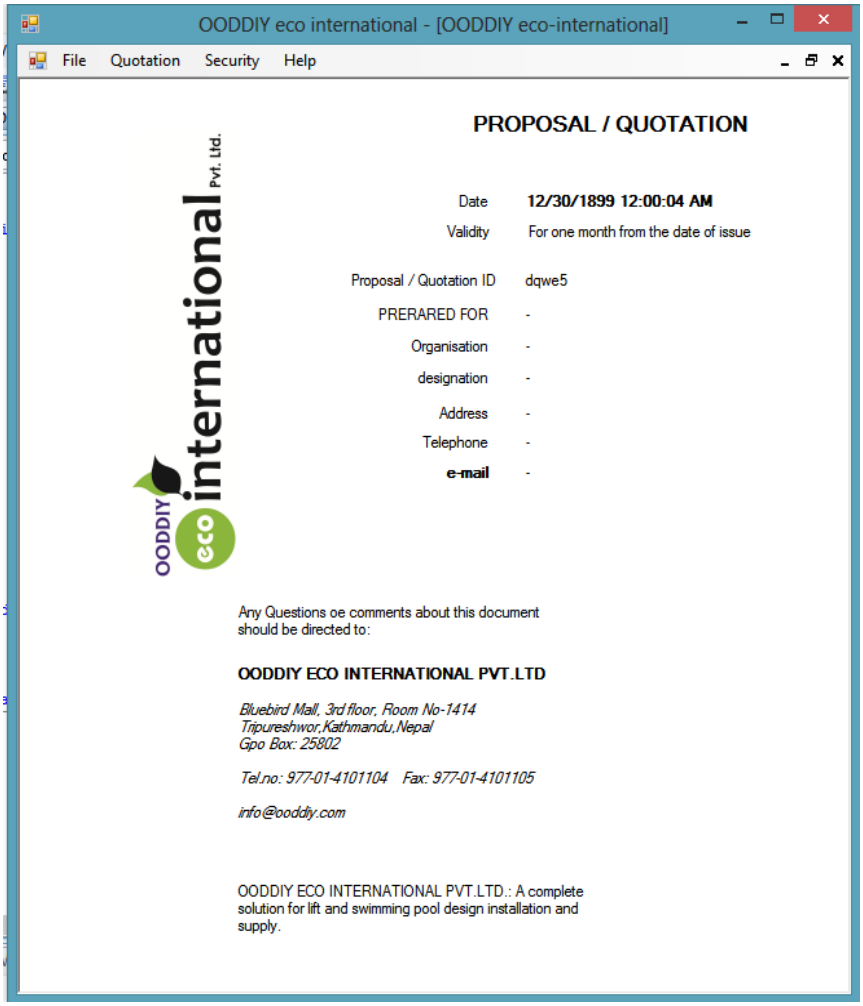
```
            db.SQLStatement = "Select * from [Quotation query] where
organisation='" & txtsearch.Text & "'"
            DataGridView.DataSource = db.GetRecords

        ElseIf cboparameter.Text = "ProductID" Then
            db.SQLStatement = "Select * from [Quotation query] where
productID='" & txtsearch.Text & "'"
            DataGridView.DataSource = db.GetRecords

        ElseIf cboparameter.Text = "sales date" Then
            db.SQLStatement = "select * from Qutation where quotationdate=#"
& dtdate.Value & "#"
            DataGridView.DataSource = db.GetRecords
        End If

    End Sub


    Private Sub cboparameter_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboparameter.SelectedIndexChanged

        'showing the date time picker only if the parameter is sales date.

        If cboparameter.Text = "Sales date" Then
            dtdate.Visible = True
            txtsearch.Enabled = False
        End If

        If cboparameter.Text <> "Sales date" Then
            dtdate.Visible = False
            txtsearch.Enabled = True
        End If

    End Sub


End Class
```

Form: Report generation




```
      Public Class QuotationIDSelector

    Private Sub QuotationIDSelector_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
        Dim Dbo As New DBUtilities

        ' updating a temporary variable with the quotation details

        Dbo.SQLStatement = "Select QuotationID from [Quotation Query]"
        Dbo.GetRecordstoCombo(ComboBox1)
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

        ' opening a new form that contains a print preview layout

        quotationid = ComboBox1.SelectedItem
        QuotationPrint.Show()
        QuotationPrint.MdiParent = startup
        Me.Close()

    End Sub
End Class
```

Form: PrintPreview and print




```
Public Class QuotationPrint

    Private Sub QuotationPrint_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim dbo As New DBUtilities

        'updating the label controls accodring to the data entered.

        Dim dtQuotation As New DataTable
        dbo.SQLStatement = "Select * from [Quotation Query] where QuotationID
='" & quotationid & "'"
        dtQuotation = dbo.GetRecords

        lblquot.Text = dtQuotation.Rows(0)("QuotationID").ToString
        lbldate.Text = dtQuotation.Rows(0)("Quotationdate").ToString
        lblclientname.Text = dtQuotation.Rows(0)("Clientname").ToString
```

```
        lblorg.Text = dtQuotation.Rows(0)("organisation").ToString
        lblemail.Text = dtQuotation.Rows(0)("email").ToString
        lbldesig.Text = dtQuotation.Rows(0)("designation").ToString
        lbladd.Text = dtQuotation.Rows(0)("Address").ToString
        lbltel.Text = dtQuotation.Rows(0)("Contactnum").ToString

        PrintForm1.Print()

    End Sub


  End Class
```

4. Testing Plan and evidence

Validation Test

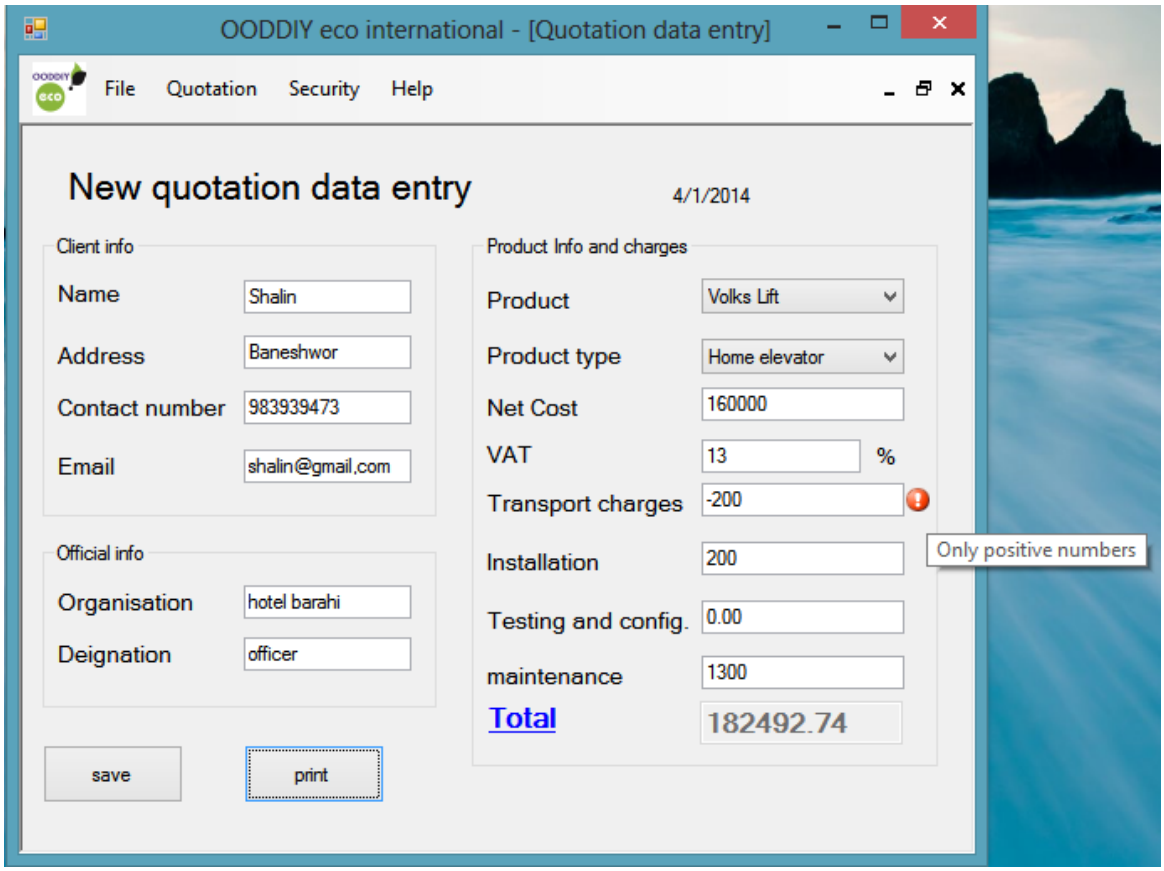
Test Number: 1

Test Objective: Not to allow invalid data to be accepted

Test Condition: Not to accept negative data while accepting price from the user.

Expected Outcome: A warning label (!), should show up indicating the error and a text that specifies the error.

Actual Outcome:



Remark: Test was successful.

Note: Similar logic was used in the programming of other forms as well. To add price of other parameters

Length Test

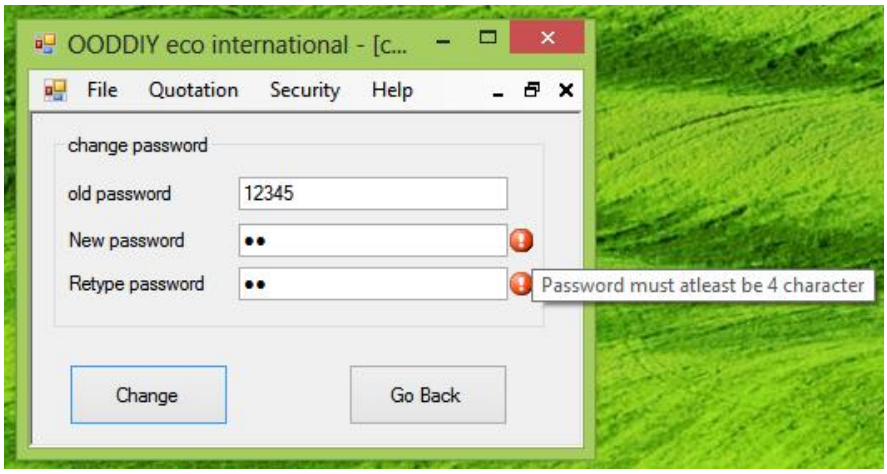
Test Number: 2

Test Objective: to assure the data is entered is appropriate.

Test Condition: to change the password which is only 2 characters long

Expected Outcome: The error signal is shown to the text box containing less character.

Actual Outcome:



Remark: The data is not processed.

Length Test

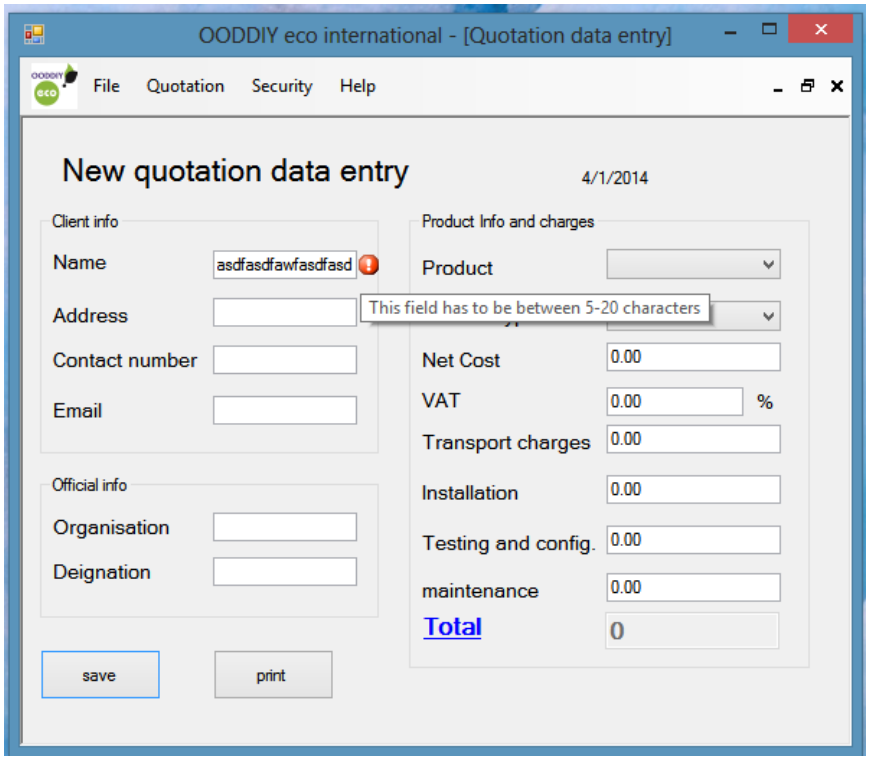
Test Number: 3

Test Objective: Not to allow data having size greater than the specified field size.

Test Condition: The maximum number of characters allowed in client Name is 20. Entering a data having characters greater than 20.

Expected Outcome: A label showing '!' sign in red color must be displayed besides the name text box and data should not be accepted.

Actual Outcome:



Remark: Test was successful.

Note: Similar logic was used in the validating other details as well.

Length Test

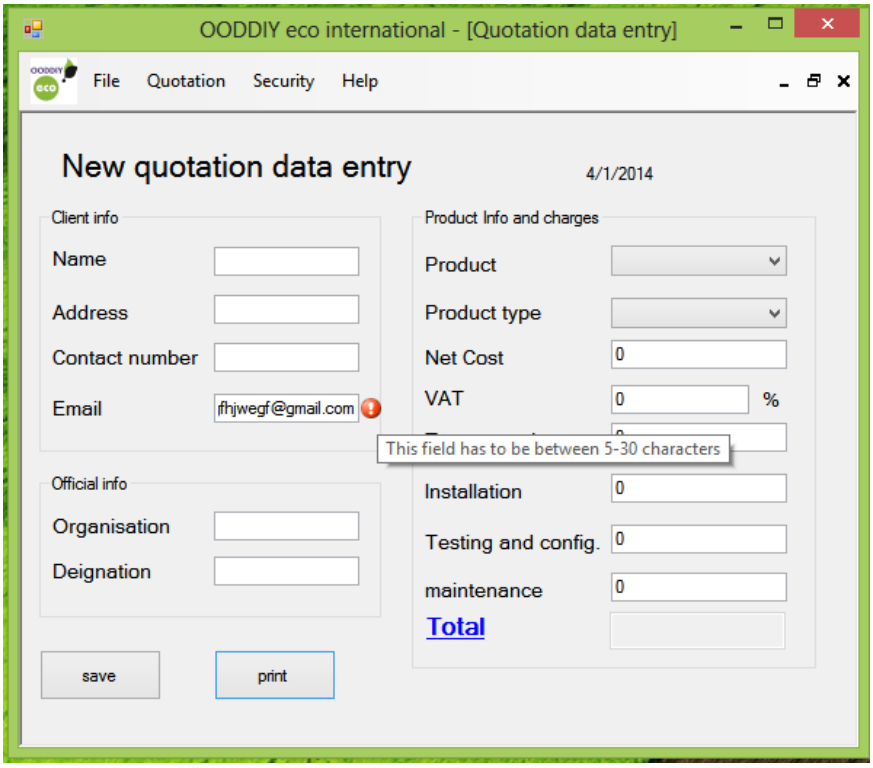
Test Number: 4

Test Objective: Not to allow data having size greater than the specified field size.

Test Condition: The maximum number of characters allowed in Email Address is 30. Entering a data having characters greater than 30.

Expected Outcome: A label showing '!' sign in red color must be displayed besides the Email Address text box and data should not be accepted

Actual Outcome:



Remark: Test was successful.

Note: Similar logic was used in the validating other controls as well. Name, contact number, address has the same field size validation technique.

Null Value Test

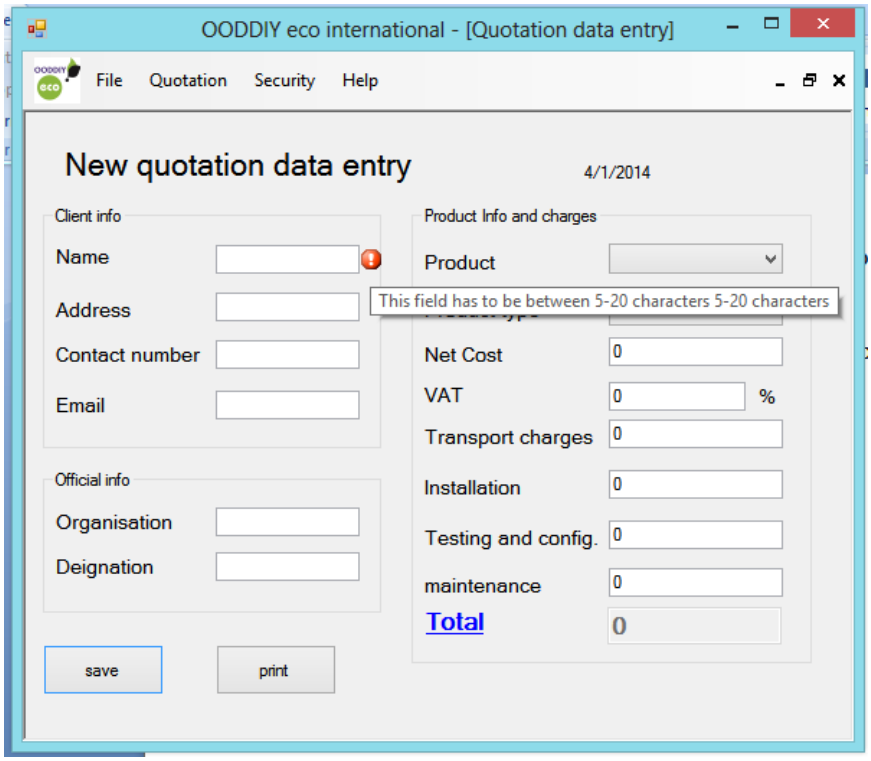
Test Number: 5

Test Objective: Not to allow blank data (null value) to be accepted.

Test Condition: Keeping a null value as client name

Expected Outcome: A label showing '!' sign in red color must be displayed besides the Type client name textbox

Actual Outcome:



Remark: Test was successful.

Note: Similar logic was used in validating other controls as contact number, address which does not allows null value to be processed

Null Value Test

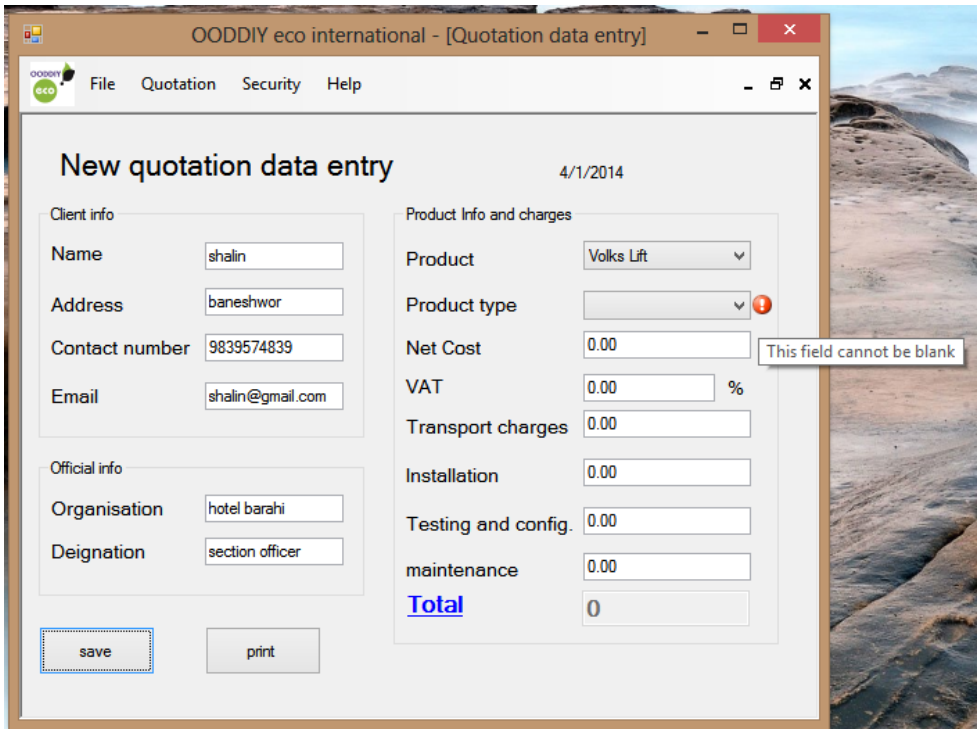
Test Number: 6

Test Objective: to not allow null data to be accepted.

Test Condition: Leaving the combo box empty in the data entry form.

Expected Outcome: An error provider icon that indicates the error.

Actual Outcome:



Remark: Test is successful.

Note: Same logic was used in the validation of other null values.

Functional Testing: Viewing Available Rooms

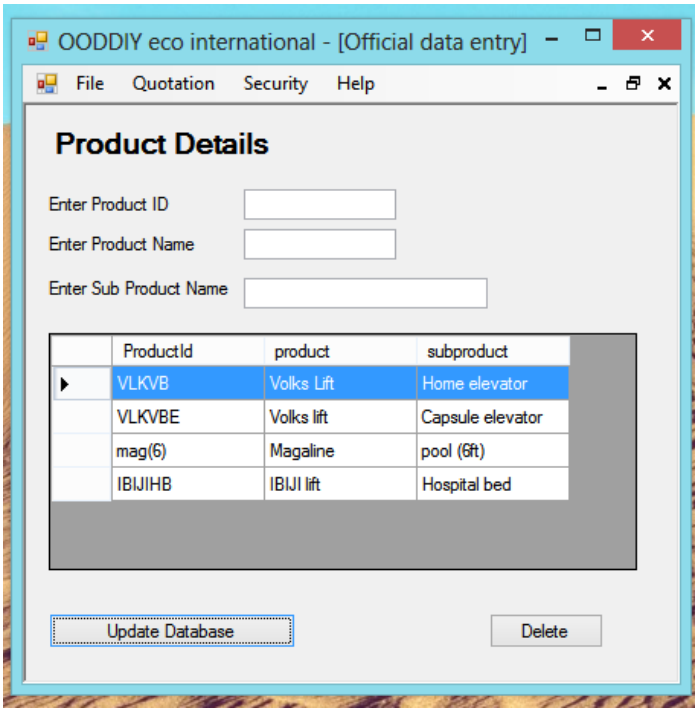
Test Number: 7

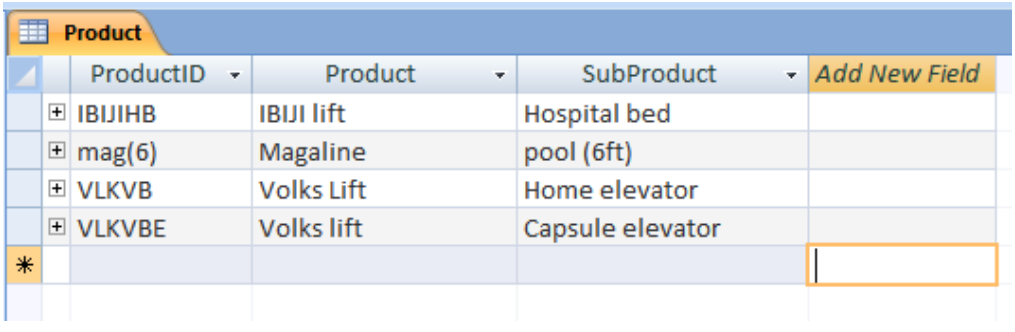
Test Objective: to list all the product details entered.

Test Condition: compare the values in data grid and data base

Expected Outcome: List all details from the database.

Actual Outcome:





Note: Test result successful

Functional Testing: Changing Password

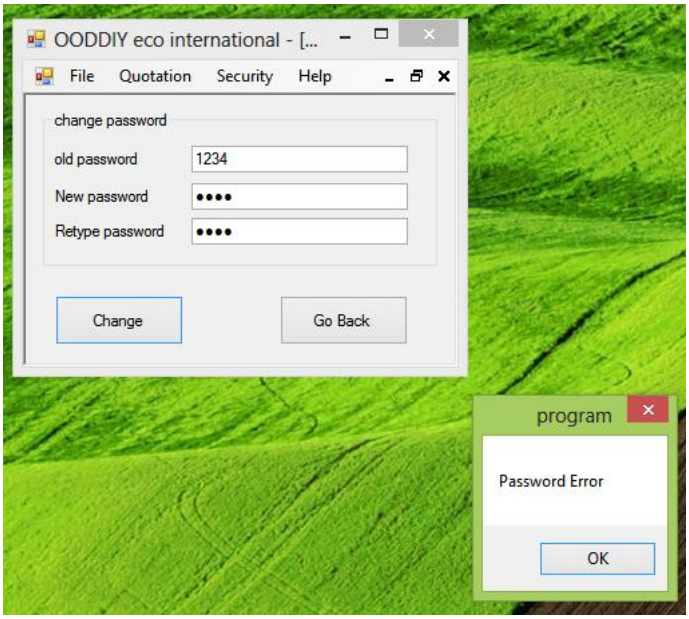
Test Number: 8

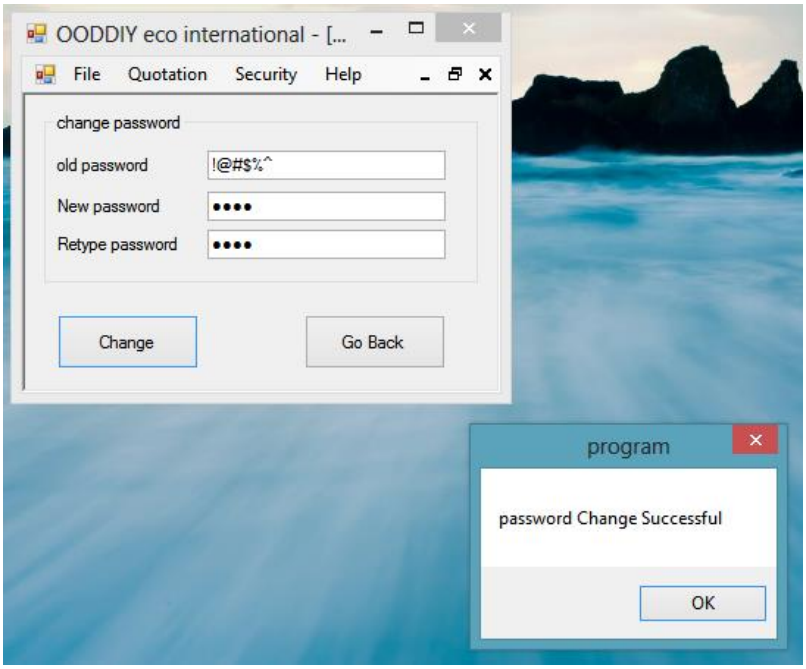
Test Objective: To change the password of the program

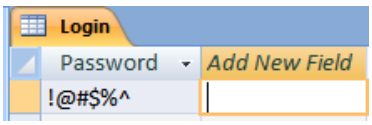
Test Condition: Enter old password and allow change password only if the entered password matches the new password.

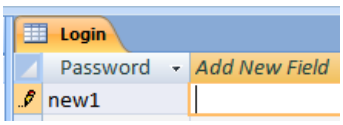
Expected Outcome: An error if the password doesn't match and change password if the password matches.

Actual Outcome:









Remark: The password change was successful the database was updated.

5. Installation

System Adaptation

The software had been designed, debugged, and tested on a windows 8 computer. The system had to be used on windows 7, which showed positive response. Various data testing were done which are shown in section *Testing*.

After a series of data testing were done, the system was finally ready to be installed. The quotation generation was done manually by replacing word document field, I suggested direct implementation. The system was perfectly capable of replacing the current method.

However, Mr. Bipin Pokhrel preferred pilot implementation as he didn't want any risk involved in a delicate task as quotation generator. He wanted to be fully convinced of the new system before using it. Pilot Implementation or Parallel Implementation includes the operator or user of the system to maintain and use both the systems at the same time.

The basic linkage of the forms were shown by me, rest of the information was briefed via user manual.

Implementation

After adding enough test data and some real time work successfully done, Mr. Bipin Pokhrel agreed on using the system permanently. Below is the written approval of the fact,:

The letter of authentication was sent to Cambridge International Examinations.

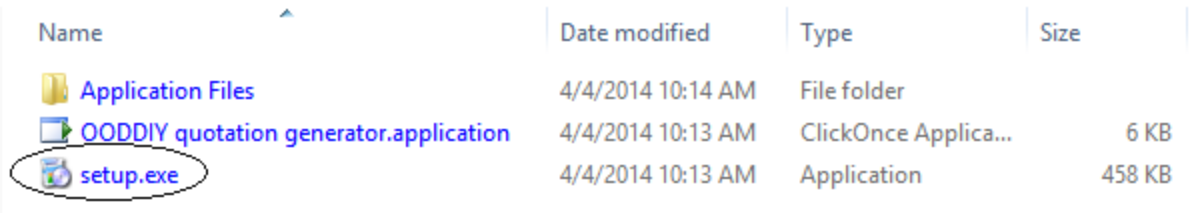
User manual

Sarun Prabhat Luitel

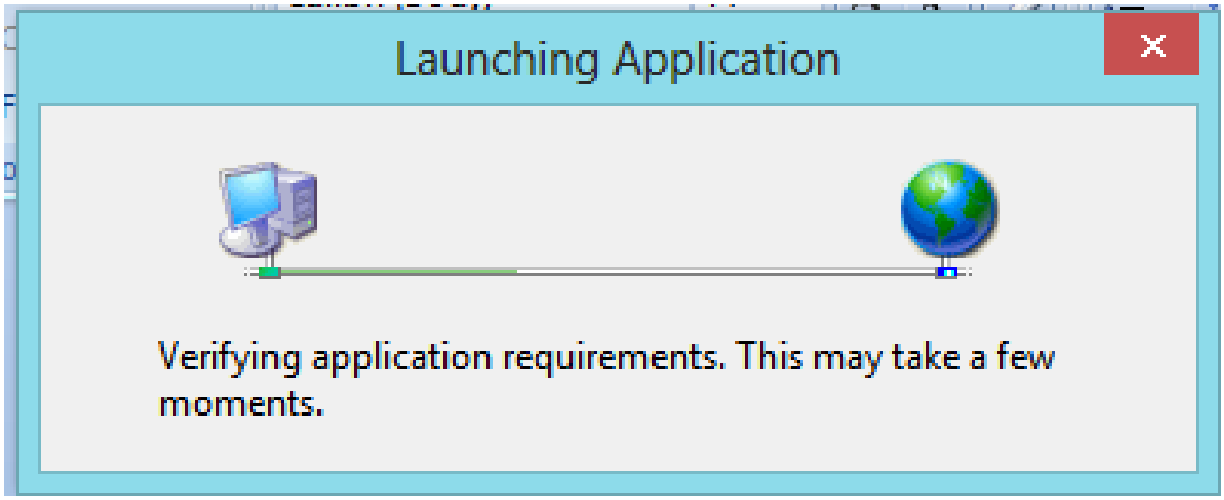
TABLE OF CONTENTS

SYSTEM SETUP

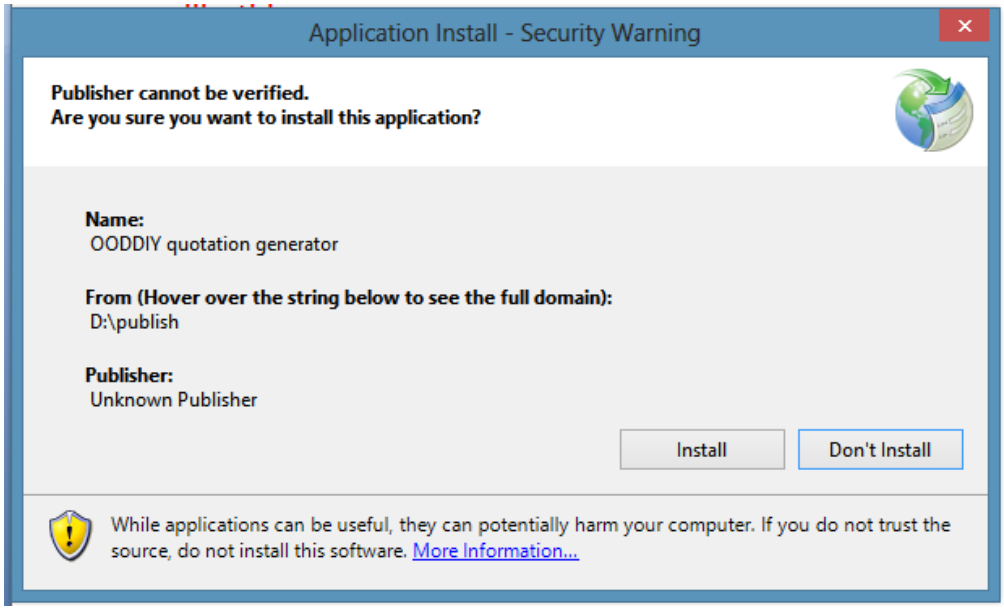
Step 1: Open the setup file to execute the setup task. There will be on screen guidance through the installation process.



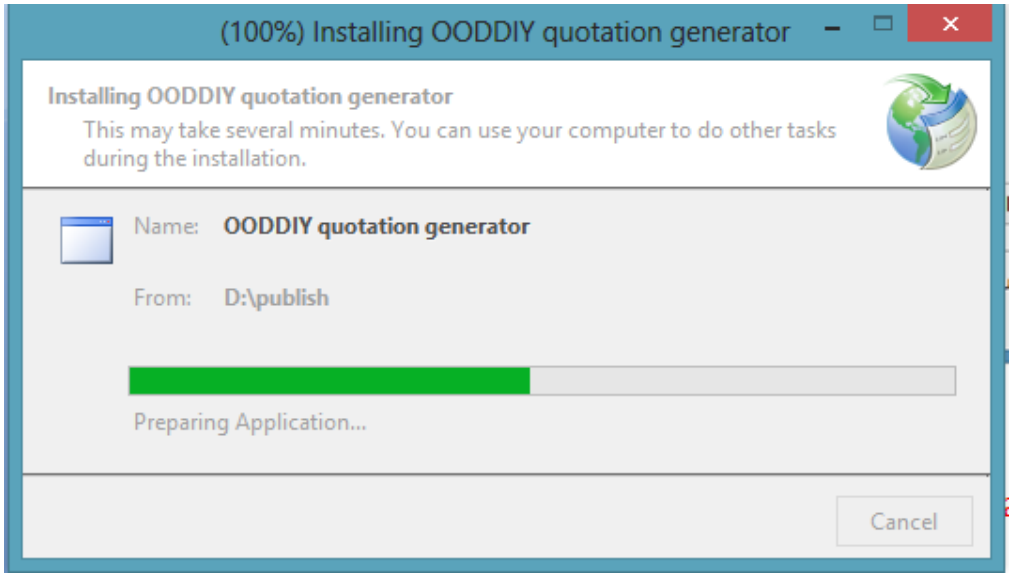
Step 2: Double click the installer file. A screen will appear. The screen looks like this:



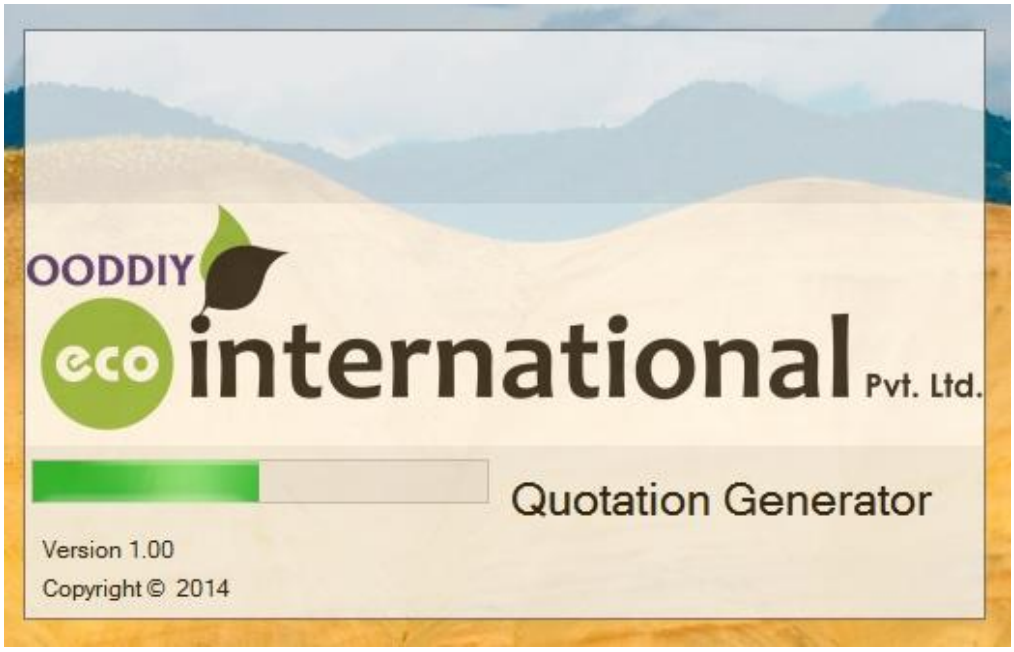
Step 3: The following screen will shortly be followed by the first screen.



Step 4: Click on the install button and the program will be installed:



Step 5: The installation procedure will close and the program will be executed.



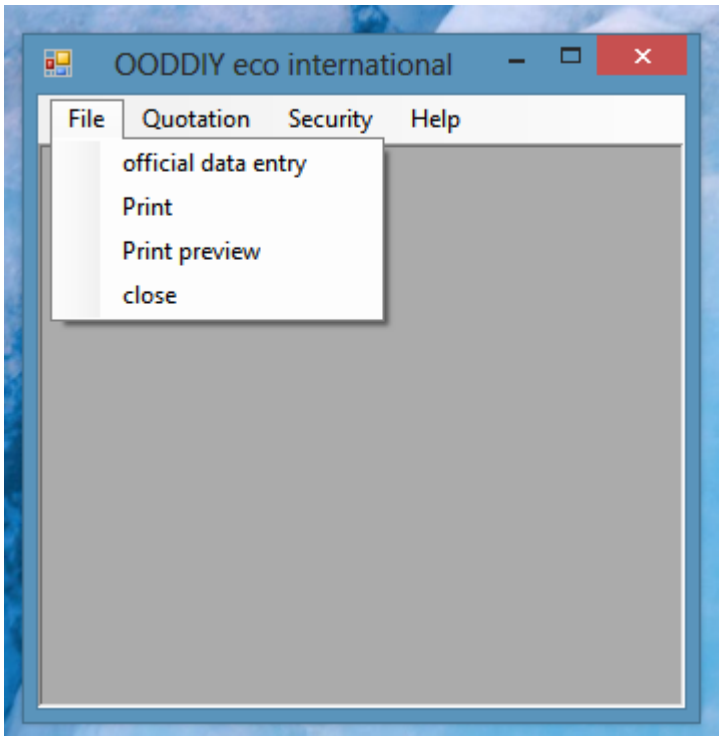
Step 6: Click on the icon using the software after the installation procedure.

HOW TO START WITH THE SYSTEM

Step 1: After clicking the icon there will be a splash screen that will be shown on screen for 4 seconds. The splash screen redirect to the login form.

Step 2: Enter the valid Password provided to you and click on the 'Log In' button.

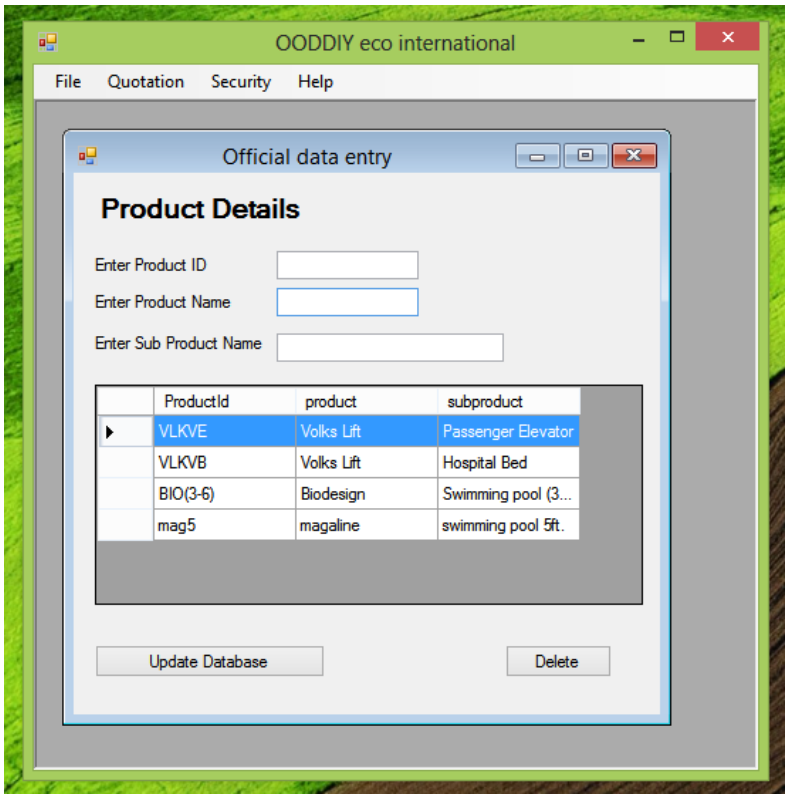
Step 3: After you have been provided valid password you will be redirected to the main menu screen.



Step 4: Choose a menu item from the menu strip.

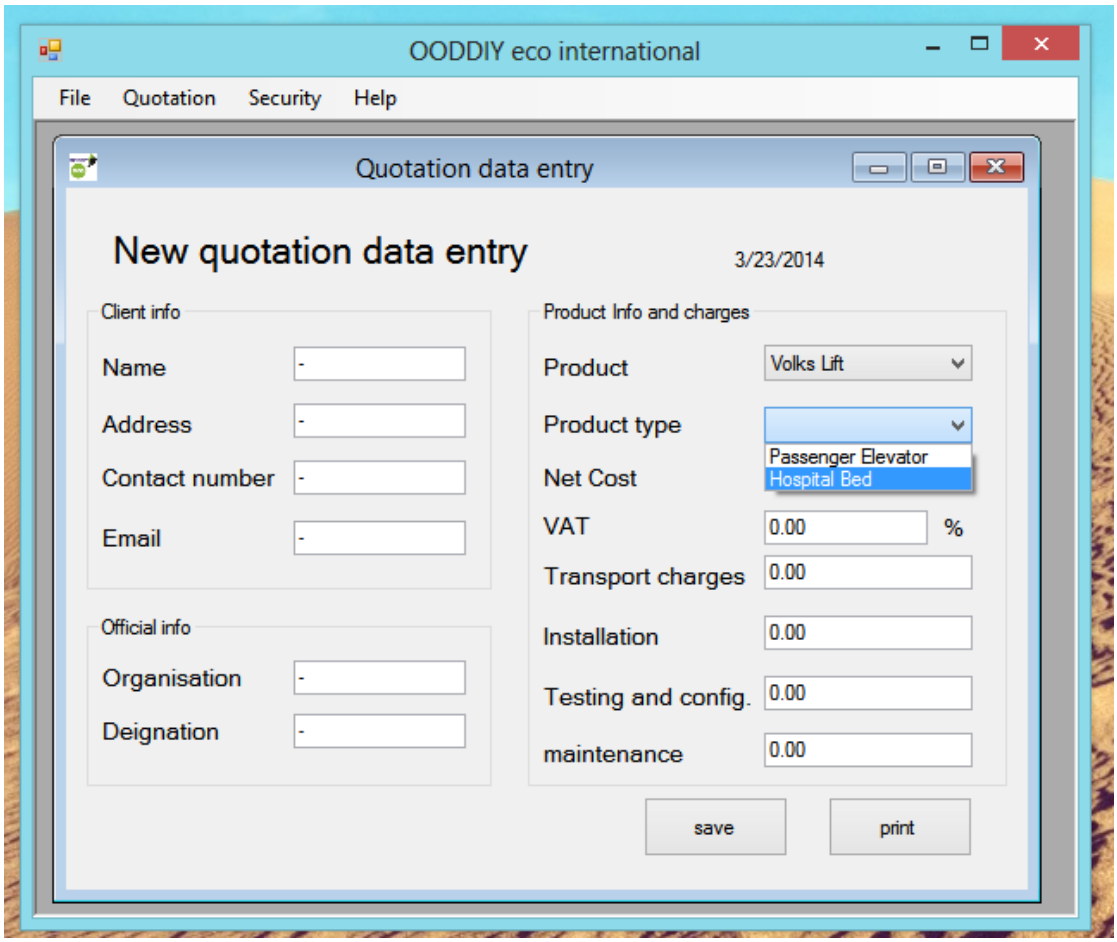
Step 5: to close the program click the [x] on the title bar or close menu under file.

Add product details (official data entry)



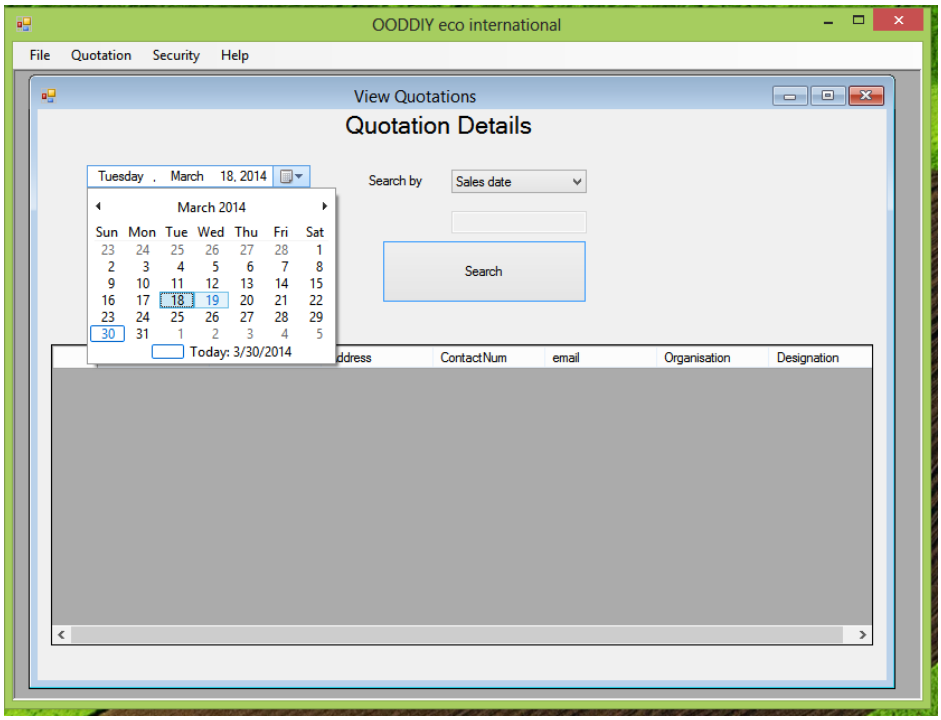
8. Enter product ID
9. Enter Product name
10. Enter Product type
11. Click on update database to update the data to database
12. Click on the data on the screen and click delete to delete the data.

Help for New quotation



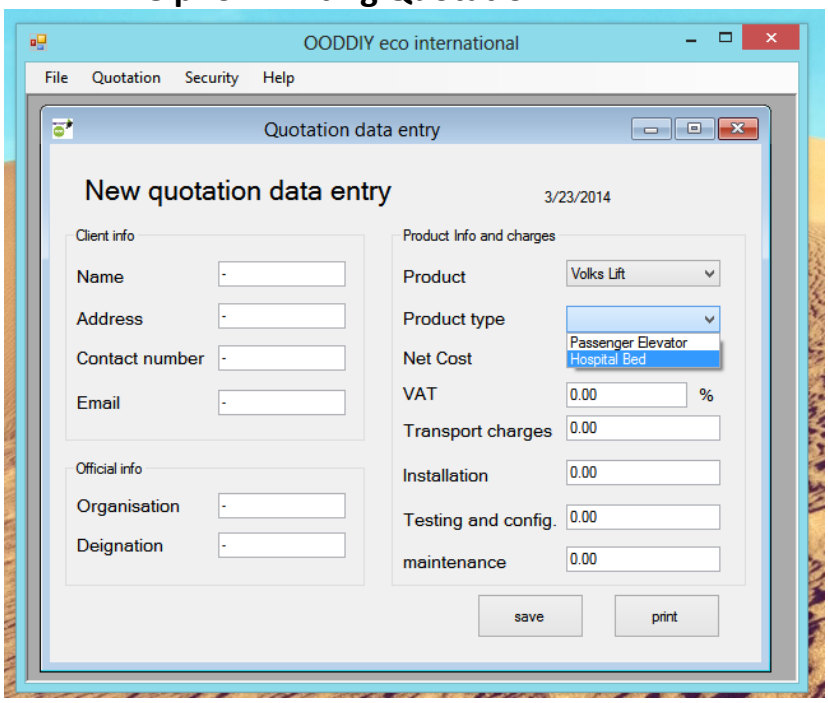
9. Enter client Details
10. Enter Product details.
11. Enter Transaction details (cost, transport, installation..)
12. The Quotation ID will be Generated the basis of the product ID and client ID.
13. Click the save button to save the data to the database.
14. Refer the section *printing* to view printing details.
15. End

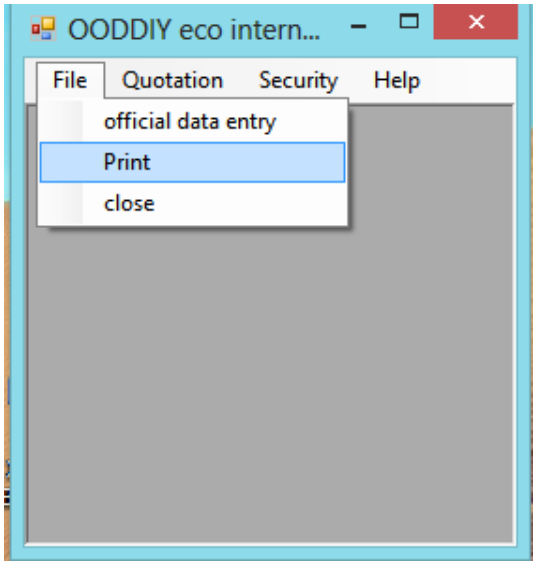
Help for viewing quotation



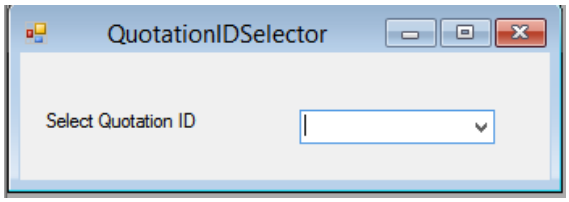
7. Choose a parameter to search the details.
8. Type the key field to search the details
9. The search details are shown on the screen

Help for Printing Quotation





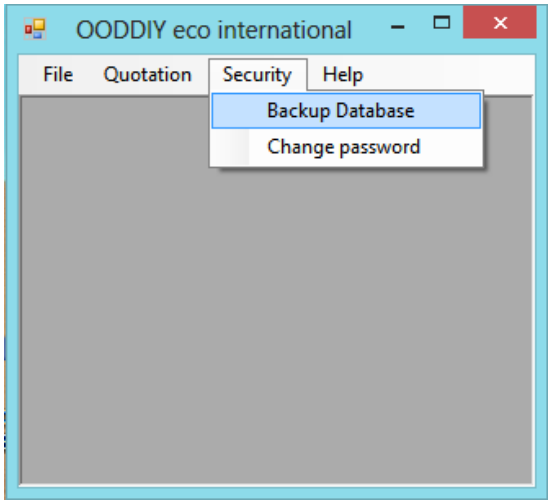
1) Click on the print button from the data entry form or print menu from the main page
2) You will be asked to select quotation Id and the report will be printed.



3) The application will directly communicate with the printer and print one copy of the report. The print screen will be displayed as the print word is being done.

BACKUP OF DATA

Step 1: The backup of the database can be done by clicking the backup database menu under security in the main menu



Step 2: choose the database location to make a backup copy of the database on the desired location.

Technical Documentation

The Technical Documentation has been published as a booklet but it has not been attached with this project report. As it contains the same topics present in this project report. To view the contents of the Technical Documentation, please refer to the section *System Design* and the section *Software Development and Programming.*

Evaluation

Level of Implementation

The system has proven to have met all the requirements mentioned in the section program requirements. Following are the requirements mentioned in section Program requirements along with the solution developed.

1) Provide a user friendly form-based interface for data entry purpose which contains field for every information need to generate a quotation.
-The forms have been successfully been designed and the simplicity in the design is proven useful. The forms are shown in the screen layout section.

2) A database is required which stores the field values permanently form the fields.
   - A database file is created using ms access. The values are accurately being stored on the database.

3) User friendly form that allows the user to search previous quotation through various parameters.
   -The form for search provides parameters which enables the search via various parameters.

4) Establish a connection with the printer that prints out the quotation.
   -The report section deals with the connection with printer and printing documents.

5) When one record is deleted it must be removed from all the tables from the database.
   -In the search form there is the facility to delete the record. Due to the relationship of the tables all the fields gets updated accurately.

6) A system with privacy maintenance has to be made so the confidential information can be maintained
   -The users will be restricted only within the staff having the knowledge of password.

7) System should allow multiple quotation generation on single date.
   -The dates are independent with the quotation and many quotations can be created on the same date.

Client Feedback/Response

Mr. Bipin was happy with the new software. He had no complains about any feature.

- He praised the design for being very simple.
- He was satisfied to get a professional method to generate the quotation.
- The review about the past quotations had been easier.

Possible Program Extensions

Even though the system met the targeted requirement there are still many aspects where the system can be extended in the future to make it more productive. Some areas possible to extension are:

- Payment module can be added so as to handle all the payments of the Clients.
- A feature of linking it to the email or the website so the quotation can be sent via mail.
- The system can include other departments like the employee's department, stock control, expenses track.