Table of Contents

Task 1 (1.1, P1, M1)

Prepare a report discussing the principles, characteristics and features of event driven programming. Consider the various characteristics like event handler, listener, trigger, functions, etc. Along with features like flexibility, suitability for GUI environments, simplicity of application, ease of development and access, also discuss various programming languages and the development environments for event driven programming.

Introduction

In past, computer users had to enter the precise codes to initiate a process. This type of technology required larger amount of trainings as well as time just to run computer properly. Computer users had to remember these codes (commands) hence computers were difficult to operate and off-course was not user friendly. Thanks to development of event driven programming approach, computers operation is easier and faster now.

According to Python school (N.D.) event driven is such programming method in which instructions are not executed in order as defined by programmer but executes according to event generated by user. It is such object oriented approach in which control flow of program depends upon the occurrence of external input/event/activity. GUI based windows, games and applications are possible due to event driven programming. This report discusses principles, characteristics, features and advantages of event driven programming system.

Principles of Event Driven Programming

While using a computer, if user clicks mouse or press keyboard button or interact with touch screen, computer response to the activity done by user. This is what event driven programming is all about. The system must have some event recognize mechanism as well as event handling mechanism to responding the event generated by user, writes Weiss (N.D.). These can be understood as basic principle of event driven programming.

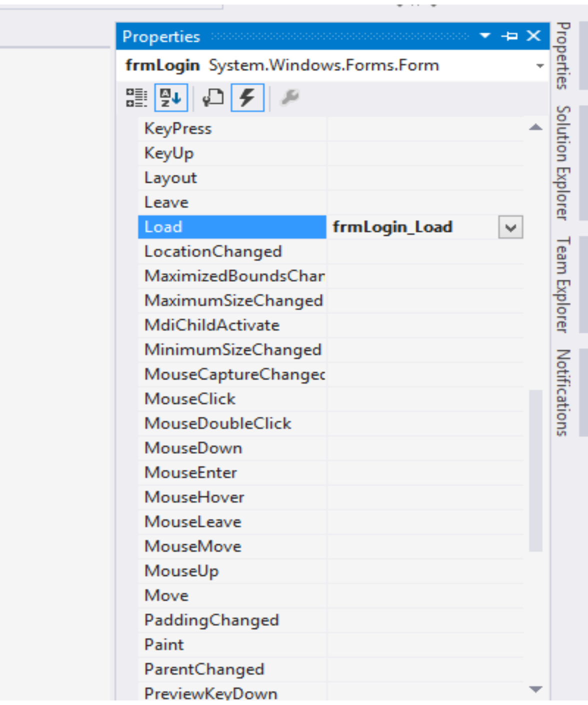
Event driven programming has characteristics like event listener, handler and trigger that acts event recognizer, handler and responder. Below, characteristics shown in event driven programming is briefly discussed.

Characteristics

Event

Event in computing is signal input to the system. According to Princeton (N.D.) event is user actions such as mouse click, key press, input through sensors to program such as barcode reader and messages from other programs. There are various types of events such as mouse events, keyboard events, HTML object events, Form events and User Interface Events.

In event driven programming some of the basic user generated events are mouse click, mouse double click, form load, mouse move, key down, key up focus, leave etc.



frmLogin System.Windows.Forms.Form

KeyPress	
KeyUp	
Layout	
Leave	
Load	frmLogin_Load
LocationChanged	
MaximizedBoundsChar	
MaximumSizeChanged	
MdiChildActivate	
MinimumSizeChanged	
MouseCaptureChangec	
MouseClick	
MouseDoubleClick	
MouseDown	
MouseEnter	
MouseHover	
MouseLeave	
MouseMove	
MouseUp	
Move	
PaddingChanged	
Paint	
ParentChanged	
PreviewKeyDown	

Figure 1 Some Events

Event Handler

This represents the function that will handle an event when the event is triggered. According to hank (2012) it is sections of code that denote the actions of event triggers within the coding structure. It is the actual code that runs when a trigger occurs.

```
Private void Form1_Load (object sender, EventArgs e)
    {
       MessageBox. Show ("Form Load Successful!");
    }
```

Event Handler

Above example is created in C# which denotes when form load event is triggered, it will show messages saying "Form Load Successful".

Listener/ Event Loops

Event loop or also known as listeners, waits for event to be fired. Nothing happen until no-event is triggered. For example when an application starts, it loads and listener waits for user to trigger event via mouse, keyboard or other input peripheral. Listener is associated with event source like mouse whereas handler is associated with event like DoubleClick.

Trigger Function

Trigger function are used for determining which event handler code should run after event triggered, Burrows (2013). These are used along with object names to specify which event handlers should run. Object has range of trigger function, for each possible event. For example, a button named btnSave has many event handlers for triggers such as mouse hover, mouse click, double click etc.

Forms

Forms are one of the most important characteristics of event driven programming. It is user interface that user see when program loads. Form contains controls such as buttons, text boxes, grid views etc. Form itself has events such as form load, form activate etc.

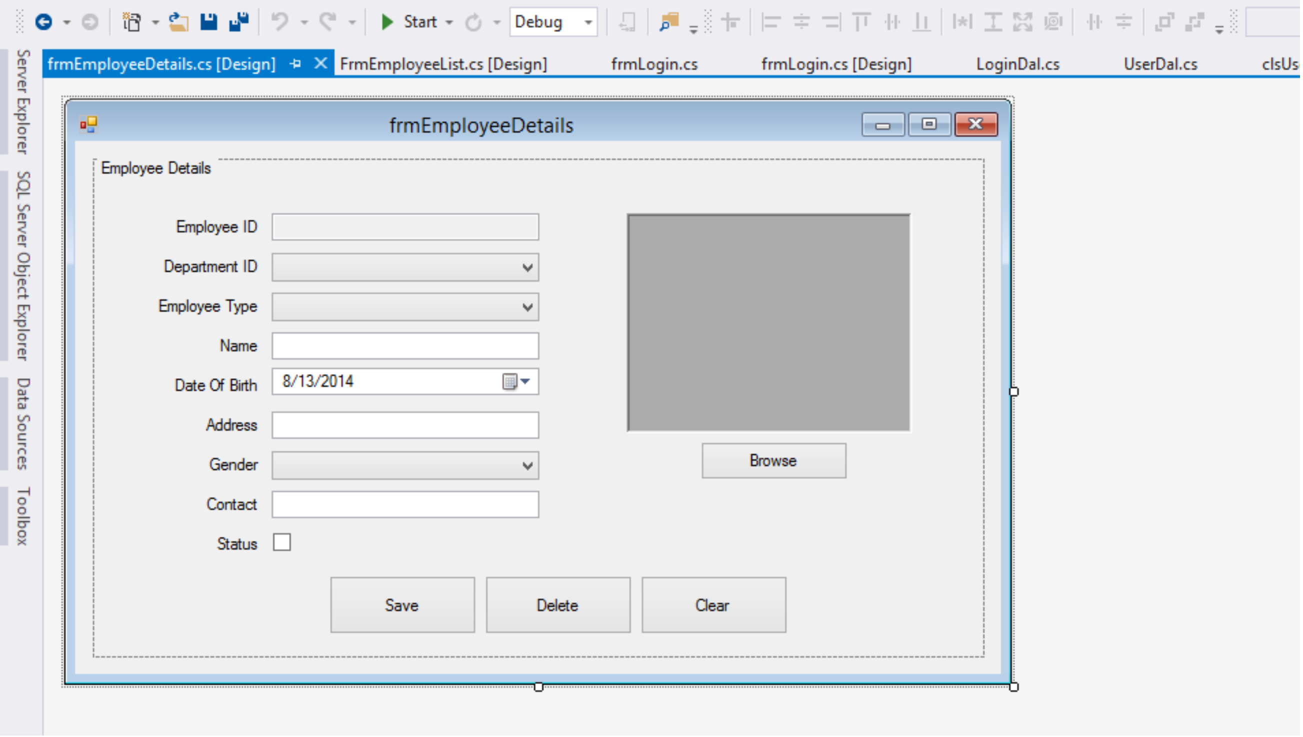


Figure 2 Windows Form Using Event Driven Programming

Features

Flexibility

Event Driven Programming is based in object oriented programming, programmers can have enormous benefits from OOP. They can place their code anywhere and can start from anywhere. Objects in program can have good control over different events. This makes event driven programming much more flexible.

Suitability for GUI environments

Characteristics like form makes event driven programming extremely suitable of GUI. Programmer has advantage of using windows library to apply windows controls in their system. Event commands are available for each control objects such as mouse click, mouse hover, enter click etc. for buttons, picture box, text box etc. This helps development of GUI system to be easier and efficient.

Ease of development

Development of system using event driven approach is so easy compared to other programming methods. Since programmer need to focus on one even at a time and can start programming from any event development process is very simple. As it's based on OOP, this also makes development easier. Unlike traditional approaches, programmer can run and test the system after working on required event handler rather than completing whole system to test it.

```
       public bool Insert_CustType(ClsCustomerType cT)
       {
           bool Result=false;
            try
            {
                String query = "Insert into tblCustomerType(Description) values
(@Description)";
                SqlCommand Cmd = new SqlCommand(query, Connect);
                Cmd.Parameters.AddWithValue ("@Description", cT.CustomerType);
                Connect.Open ();
                int rows= Cmd.ExecuteNonQuery ();
                   if (rows>0)
                       Result=true;
                else
                       Result =false;
            }
            catch (Exception ex)
            {
                throw ex;
            }
            finally
            {
                Connect.Close();
            }
           return Result;
       }
```

Figure 3 Event Driven is OOP based, hence programming is easier

Potential Portability

Programs developed in event driven is very easy to alter when require in future. Programmer can add/remove functions, add more forms, and alter how it reacts to certain events. It also provides facility that allows different teams to work on same project without any problem. Here, one team work on certain form and other teams work one other.

Simplicity of Application

Since programmers have ability to design visually, it saves whole lots of time of development. Developer can design system as per user's requirement making it simple and user friendly.

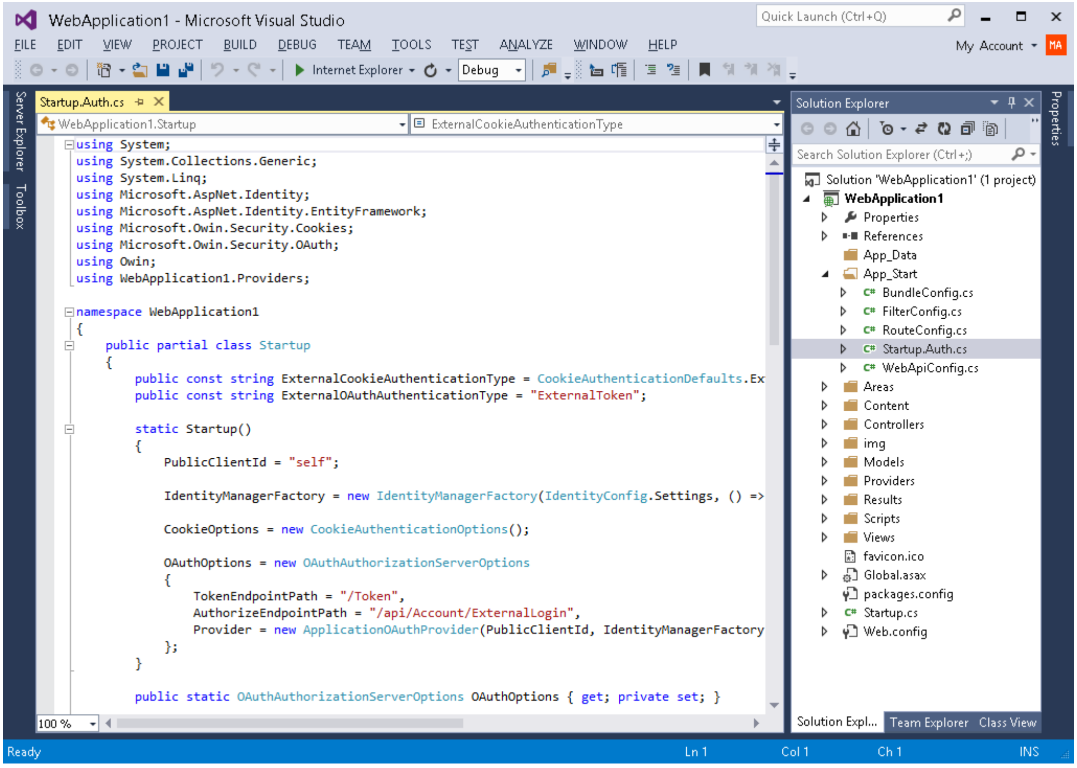
Development Languages and Environments

C# and Visual Studio .Net Environment

C# is simple, modern and object oriented programming language. C# is used to develop web applications as well as applications for windows and mobiles. It is well managed and easy to lean language that is based on event driven programming.

Visual studio provides developing platform for c# and other programming languages such ass C++, F#, J# etc.

C# is a good event driven programming language which provides much control options to programmer. Visual studio generated lots of codes itself and programmers only need to code even handlers.



```
using System;
using System.Collections.Generic;
using System.Linq;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin.Security.Cookies;
using Microsoft.Owin.Security.OAuth;
using Owin;
using WebApplication1.Providers;

namespace WebApplication1
{
    public partial class Startup
    {
        public const string ExternalCookieAuthenticationType = CookieAuthenticationDefaults.Ex
        public const string ExternalOAuthAuthenticationType = "ExternalToken";

        static Startup()
        {
            PublicClientId = "self";

            IdentityManagerFactory = new IdentityManagerFactory(IdentityConfig.Settings, () =>

            CookieOptions = new CookieAuthenticationOptions();

            OAuthOptions = new OAuthAuthorizationServerOptions
            {
                TokenEndpointPath = "/Token",
                AuthorizeEndpointPath = "/api/Account/ExternalLogin",
                Provider = new ApplicationOAuthProvider(PublicClientId, IdentityManagerFactory
            };
        }

        public static OAuthAuthorizationServerOptions OAuthOptions { get; private set; }
```

Figure 4 Programming in Visual Studio using C#

Reference

- Burrows, R. (2013) *key-features-of-event-driven-programming* Available: http://reece-burrows-media.blogspot.com/2013/11/key-features-of-event-driven-programming.html Accessed [8/29/2014]
- Hank, A. (2012) *key-features-of-event-driven-programming* Available: http://prezi.com/wv4ovm2xatna/key-features-of-event-driven-programming/ Accessed [8/29/2014]
- Princeton (n.d.) *Event Driven programming* Available : http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Event-driven_programming.html Accessed [8/20/2014]
- Python school (n.d.) *Event Driven Programming* Available: http ://www.pythonschool.net/eventdrivenprogramming/ Accessed [8/20/2014]
- Weiss, S. (n.d.) *Chapter 6 Event Driven Programming* Available: http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/unix_lecture_notes.php Accessed [8/21/2014]

Task 2 (2.1, M2)

Design an event driven programming solution to a given problem in scenario.

Introduction

To design event driven programming solution for given problem in scenario, several tools can be used to plan the solution. These tools help to develop structure for the system.

Flowchart

Flowchart is symbolic representation of flow of a system. Each step contains short description and connected with help of arrow to show the direction of process. It has terminator, process, decision, data and document and their relation. Flowchart of the system is drawn below in diagram1.

Context diagram

Context diagram is basic structure of system. This diagram demonstrates relation of different entities that interact with different processes. Context diagram of this system is drawn in diagram2.

0-level DFF diagram

Zero level diagrams are first level diagram that represents the flow of data inside system. 0 level diagrams for developed solution is drawn in digram3.

Organization Requirement and Program Design

Table below justifies that selection of design satisfy the requirement of the system.

SN.	**Organizational Requirement**	**Feature in software to satisfy the requirements**
1.	Maintain Information about Employee, customers, products	To manage information about various stakeholders of department store, this software will have management feature for each category.
2.	Manage sales and purchase	The developed system will have mechanism that will allow user to manage sales and purchase record.
3.	Manage Sales Return And Purchase Return	This system will also allow user to keep record of sale or purchase return. Product in inventory will be managed automatically.
4.	Manage inventory and Transaction Log	This system will have separated section to manage history logs and inventory.

5.	Have Admin/User Login Feature	In the startup of the software, user name and password will be asked. Screen layout, and privileges will be according to user type.

Summary

In this report, several tools are used to design event driven programming for the given problem. Zero level diagram, context diagram and flowchart diagram is drawn in three separate pages. This report also discussed how designed system is appropriate for given problem in the scenario.

Task 3 (2.2, D1)

Identify the screen components and data and file structures required to implement a given design.

Criteria definition

To implement solution for given problem of brown bird super market, various screen components are required to be identified. These components include forms, controls, class, properties, tables and various other elements. Identification of these components and data and file structure should be able to satisfy the criteria. Criteria are listed below:

1. System should match software requirement such as sales, purchase, sales return etc.
2. System's files should be well managed
3. Identified components should able to develop simple and user friendly solution

Properties

In event driven programming with C#, properties refer to a member who provides mechanism for read, write and compute the value. It has get and set accessors. If properties do not have set Accessor, it is read only. In current solution for brown bird supermarket properties are used in various areas which are shown in later part of the report.

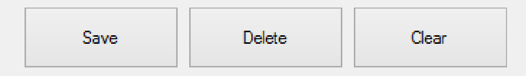
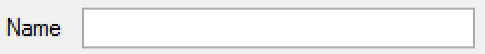
Form

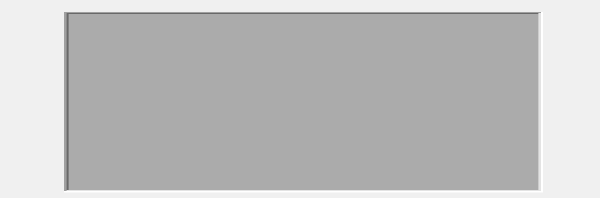
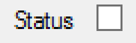
Form is that mechanism which provides users platform to use the system. In case of current solution, as it is to be developed on windows based system, form is user interface similar to windows. It is collection of several controls. Implementation of Form in the solution will make the system more users friendly, attractive and easy to develop.

Controls

Controls are those components, user will directly interact with. Several examples of controls are grid view, textbox, button etc.

Identified required controls for the solution is shown in the table below.

S.N.	Control Name	Image
1.	Button	
2.	Textbox	

3.	Grid view	
4.	Check box	
5.	Combo box	

Schema Diagram:

Schema diagram is a visual representation of relation between different data tables in the system. Schema diagram also displays all the data tables in the database of system which helps developer to design database for the system as per planning.

Schema diagram for the system to be developed is drawn in page below.

Screen Component structure for the solution to be developed is listed below.

Evaluation of Identified Components and Criteria

After identification of screen components, data and file system for the solution to be developed, this document critically evaluates the finding with the criteria set for the evaluation.

1. System should match software requirement such as sales, purchase, sales return etc.

Specific class, properties, UI is identified for each features such as sale, purchase, customers etc. These components are essential for satisfying the software requirement of the solution.

2. System's files should be well managed

Screen Component identified are web managed due to the fact they are well placed in different directories, classes etc. All UI forms are placed inside UI directory. Similar, all class files are place inside class folder. Class for variable is place in different folder. This technique helps to manage the development phase properly.

3. Identified components should able to develop simple and user friendly solution

The identification of screen components such as buttons, grid view, text box etc. helps to develop user friendly solution. Text box is used for getting data from user; grid view is used for displaying the data in data table. Similarly, button is another component identified for the solution which in event driven programming can be used for even triggering. These identified screen components are capable of developing simple solution that can match system requirement of the solution.

Conclusion

Different screen components and data and file structure is identified in this document. For the validity of the finding, different criteria were set. After identification, the finding was critically evaluated with the set criteria. During the evaluation it is found that finding of components satisfies the set criteria. This evaluation justifies that validity of result has be evaluated using defined criteria.

Task 4 [3.1] [3.2 D2] [3.3]

Implement the event driven programming solution based on the prepared design specification.

- **Implement event handling using control structure to meet the design algorithms.**
- **Identify and implement opportunities for error handling and reporting.**

Introduction

To implement event driven solution (BrownBirdMS) for the given problem, several events, control structure, looping mechanism and error handling were used in the system. This report describes which, how and why various components are implemented in the system to meet design algorithm.

Project Management

For proper management of the project, whole project is divided into different phases. Different phase has different set of task to be done. These phases are:

1. Planning

In this phase basic planning for the solution is done. The basic requirement for the system is analyzed.

2. Analysis

In this phase detailed analysis is done. Information is collected to find user needs, solution requirement etc.

3. Design

In this phase, layout for the system is done.

4. Implementation

In this phase, codding, testing, debugging is done.

5. Deployment and Maintenance

In this phase solution is distributed to the client. Further maintenance is done after user testing.

Gantt chart

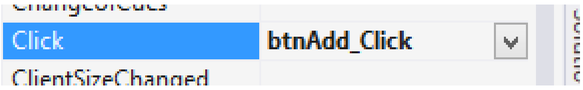
Gantt chart is graphical representation of decomposing the work into smaller and practicable tasks along with the time scale. To represent different phase of project and time scale of each phase, Gantt chart for the system is drawn below.

Events

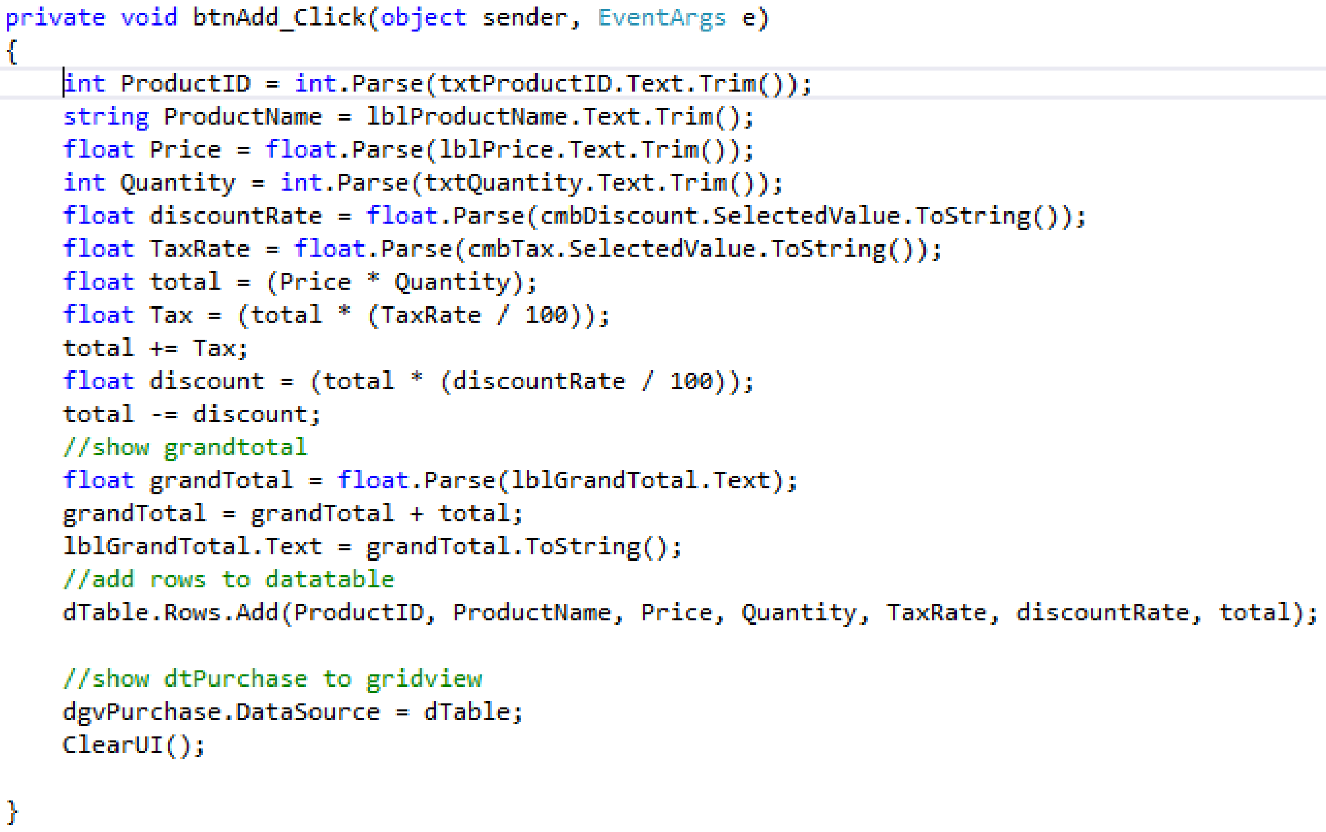
To make the solution efficient and user intractable, various events are implemented in the system. These events are implemented as per system requirement and suitability in the scenario. Proper use of these events helps to reduce unnecessary software operation time.

Button Click Event

Button click event is generated when user clicks on button. It is very simple user control which allows user to perform required action with just click of a button.



Click	btnAdd_Click
ClientSizeChanged	



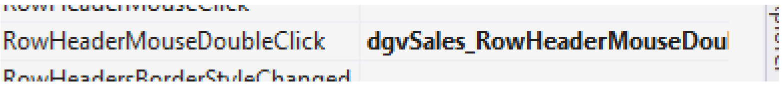
```
private void btnAdd_Click(object sender, EventArgs e)
{
    int ProductID = int.Parse(txtProductID.Text.Trim());
    string ProductName = lblProductName.Text.Trim();
    float Price = float.Parse(lblPrice.Text.Trim());
    int Quantity = int.Parse(txtQuantity.Text.Trim());
    float discountRate = float.Parse(cmbDiscount.SelectedValue.ToString());
    float TaxRate = float.Parse(cmbTax.SelectedValue.ToString());
    float total = (Price * Quantity);
    float Tax = (total * (TaxRate / 100));
    total += Tax;
    float discount = (total * (discountRate / 100));
    total -= discount;
    //show grandtotal
    float grandTotal = float.Parse(lblGrandTotal.Text);
    grandTotal = grandTotal + total;
    lblGrandTotal.Text = grandTotal.ToString();
    //add rows to datatable
    dTable.Rows.Add(ProductID, ProductName, Price, Quantity, TaxRate, discountRate, total);

    //show dtPurchase to gridview
    dgvPurchase.DataSource = dTable;
    ClearUI();

}
```

Grid view Events

Figure 5 Event Handler for button Click Event (Example)

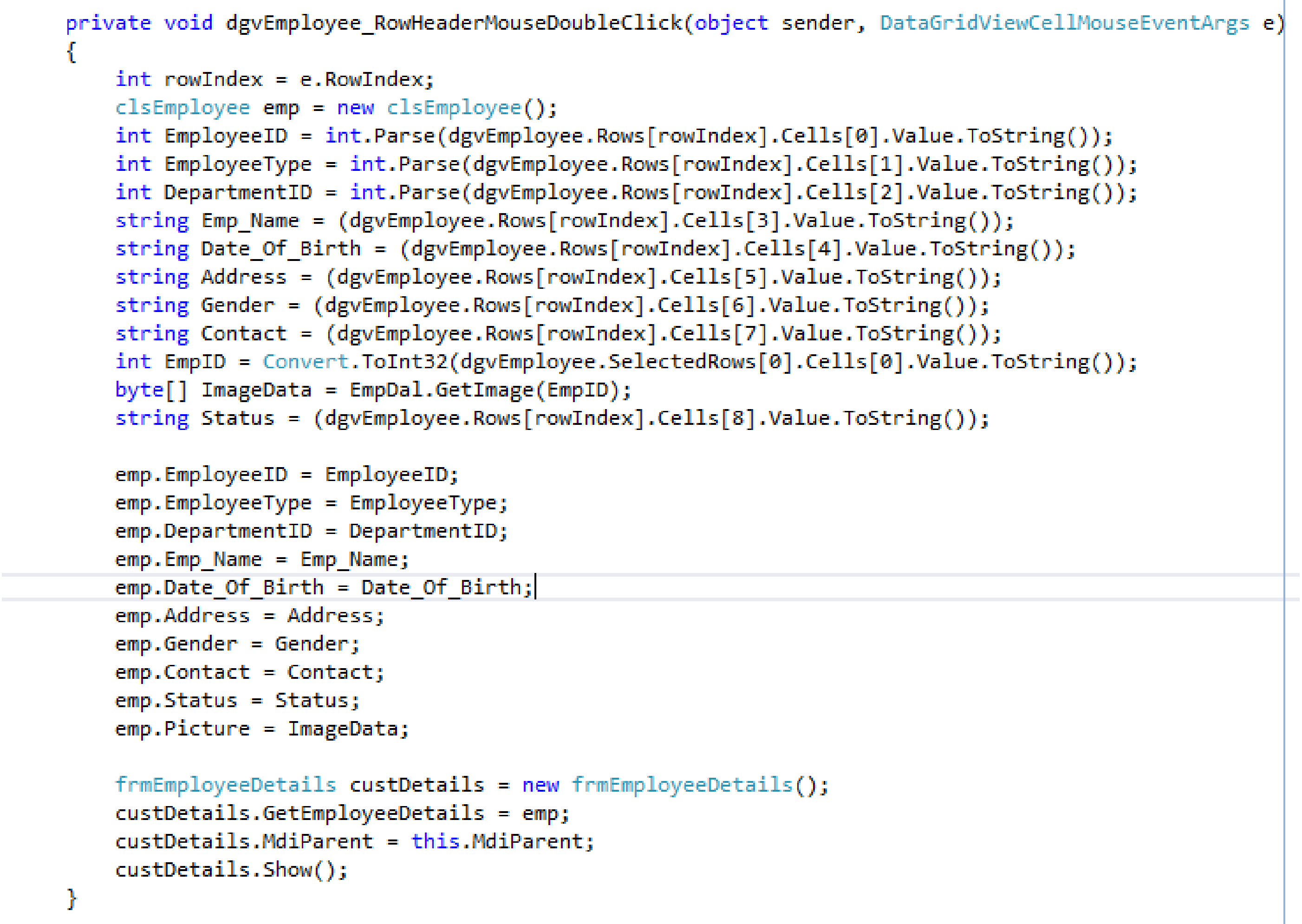
Grid vie' is very useful component in event driven programming with C#. Grid view supports various events. In this system grid view cell click, row header click and row header double click events are used.

Row Header Double Click



RowHeaderMouseDoubleClick	dgvSales_RowHeaderMouseDou

This event is triggered when user double click (left) instantly on the row header of table in grid view. This event is used for loading tabular data in UI components and cancelling sale/purchase in this system.

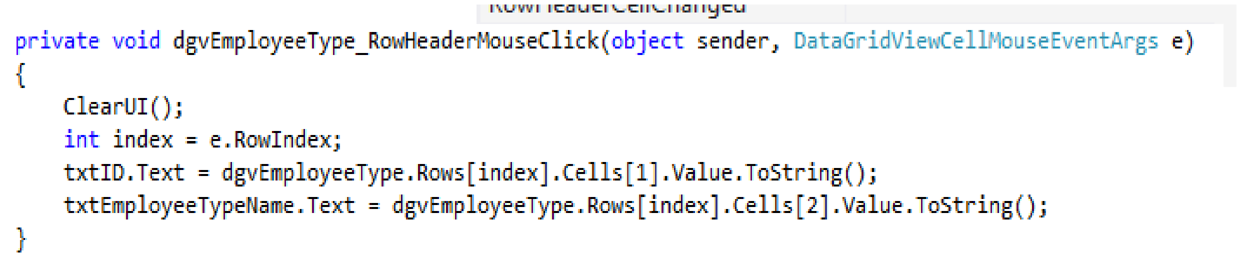


```
    private void dgvEmployee_RowHeaderMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)
    {
        int rowIndex = e.RowIndex;
        clsEmployee emp = new clsEmployee();
        int EmployeeID = int.Parse(dgvEmployee.Rows[rowIndex].Cells[0].Value.ToString());
        int EmployeeType = int.Parse(dgvEmployee.Rows[rowIndex].Cells[1].Value.ToString());
        int DepartmentID = int.Parse(dgvEmployee.Rows[rowIndex].Cells[2].Value.ToString());
        string Emp_Name = (dgvEmployee.Rows[rowIndex].Cells[3].Value.ToString());
        string Date_Of_Birth = (dgvEmployee.Rows[rowIndex].Cells[4].Value.ToString());
        string Address = (dgvEmployee.Rows[rowIndex].Cells[5].Value.ToString());
        string Gender = (dgvEmployee.Rows[rowIndex].Cells[6].Value.ToString());
        string Contact = (dgvEmployee.Rows[rowIndex].Cells[7].Value.ToString());
        int EmpID = Convert.ToInt32(dgvEmployee.SelectedRows[0].Cells[0].Value.ToString());
        byte[] ImageData = EmpDal.GetImage(EmpID);
        string Status = (dgvEmployee.Rows[rowIndex].Cells[8].Value.ToString());

        emp.EmployeeID = EmployeeID;
        emp.EmployeeType = EmployeeType;
        emp.DepartmentID = DepartmentID;
        emp.Emp_Name = Emp_Name;
        emp.Date_Of_Birth = Date_Of_Birth;
        emp.Address = Address;
        emp.Gender = Gender;
        emp.Contact = Contact;
        emp.Status = Status;
        emp.Picture = ImageData;

        frmEmployeeDetails custDetails = new frmEmployeeDetails();
        custDetails.GetEmployeeDetails = emp;
        custDetails.MdiParent = this.MdiParent;
        custDetails.Show();
    }
```

Figure 6 Event Handler for Row Header Double Click (Example

Row Header Mouse Click

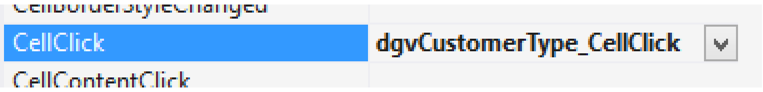
Similar to Row Header Double mouse click this event requires single click to be triggered. In this solution, this event is used for getting data from grid view to various UI components.



```
    private void dgvEmployeeType_RowHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)
    {
        ClearUI();
        int index = e.RowIndex;
        txtID.Text = dgvEmployeeType.Rows[index].Cells[1].Value.ToString();
        txtEmployeeTypeName.Text = dgvEmployeeType.Rows[index].Cells[2].Value.ToString();
    }
```

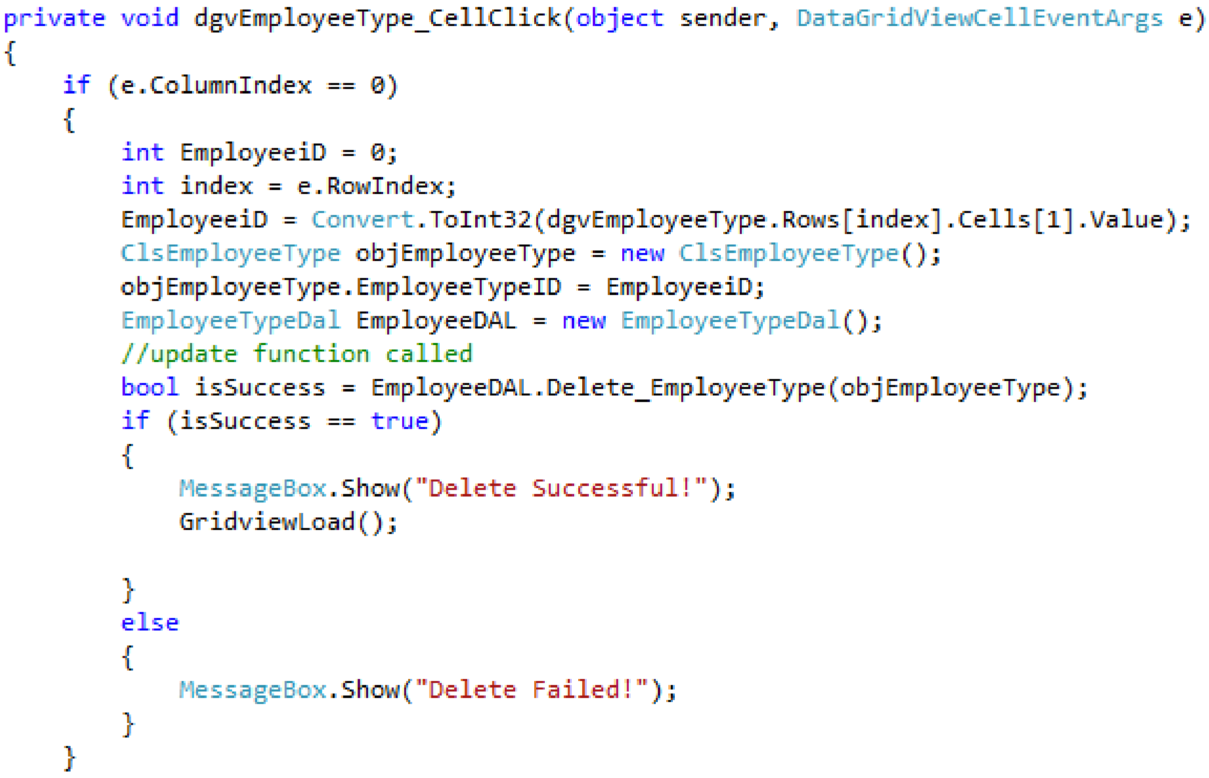
Figure 7: Event Handler for Row Header Mouse Click (Example)

Data grid view cell click



CellBorderStyleChanged
CellClick | dgvCustomerType_CellClick
CellContentClick

Cell Click event is triggered when user click inside the cell of table in grid view. It is also a useful event used in this system to detect if user clicked delete button inside grid view, if clicked it execute codes that deletes data from table correspond to that delete button's row number.

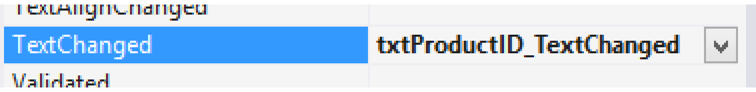


```
private void dgvEmployeeType_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == 0)
    {
        int EmployeeiD = 0;
        int index = e.RowIndex;
        EmployeeiD = Convert.ToInt32(dgvEmployeeType.Rows[index].Cells[1].Value);
        ClsEmployeeType objEmployeeType = new ClsEmployeeType();
        objEmployeeType.EmployeeTypeID = EmployeeiD;
        EmployeeTypeDal EmployeeDAL = new EmployeeTypeDal();
        //update function called
        bool isSuccess = EmployeeDAL.Delete_EmployeeType(objEmployeeType);
        if (isSuccess == true)
        {
            MessageBox.Show("Delete Successful!");
            GridviewLoad();

        }
        else
        {
            MessageBox.Show("Delete Failed!");
        }
    }
```

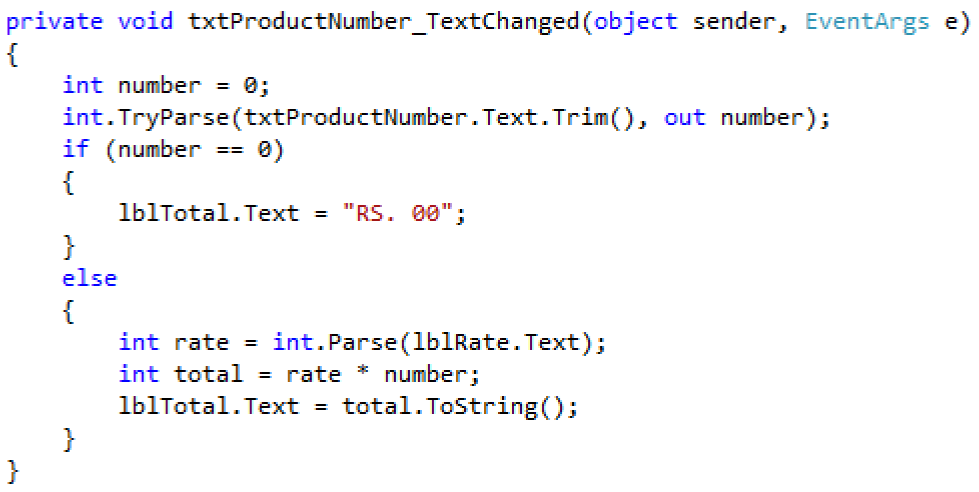
Figure 8 Event Handler for Grid View cell click event (Example)

Text Change Event

This event is triggered when user changes text inside a textbox. Event is triggered every time a character is either added or deleted from the text box. This event is used for searching, filtering features in this solution.



TextAlignChanged
TextChanged | txtProductID_TextChanged
Validated



```
private void txtProductNumber_TextChanged(object sender, EventArgs e)
{
    int number = 0;
    int.TryParse(txtProductNumber.Text.Trim(), out number);
    if (number == 0)
    {
        lblTotal.Text = "RS. 00";
    }
    else
    {
        int rate = int.Parse(lblRate.Text);
        int total = rate * number;
        lblTotal.Text = total.ToString();
    }
}
```

Figure 9: Event Handler for Text Change Event

Menu stripe Event

Menu stripe is user control which generates menu bar in the form. It consist columns that link to different forms. Menu stripe supports several basic events. For this solution mouse single click and double click event is utilized.

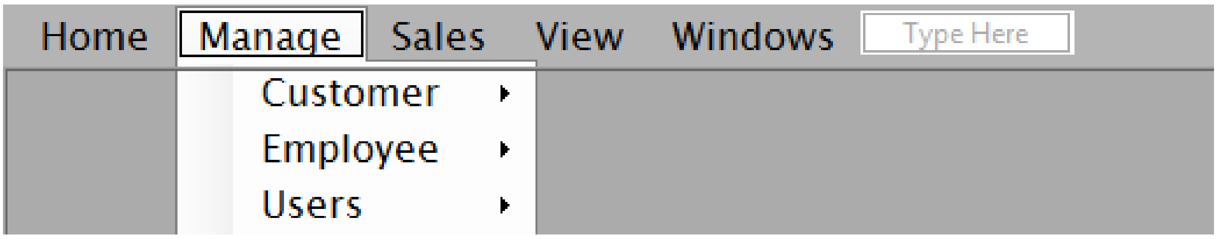
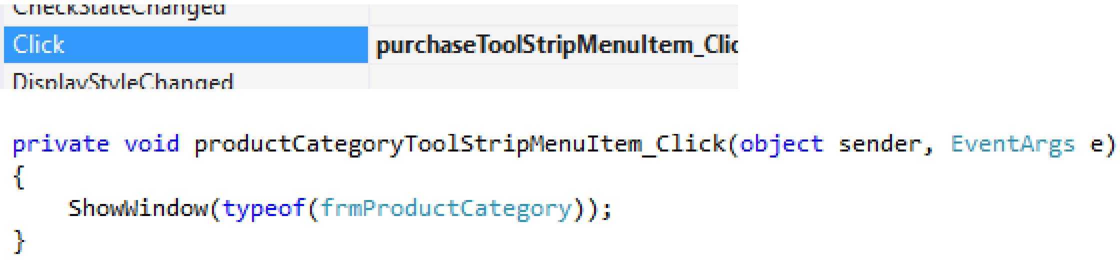


Figure 10 Menu Stripe

Mouse click event

This event is triggered when user single click on menu stripe. This event is used in this system for showing different forms as required by user.



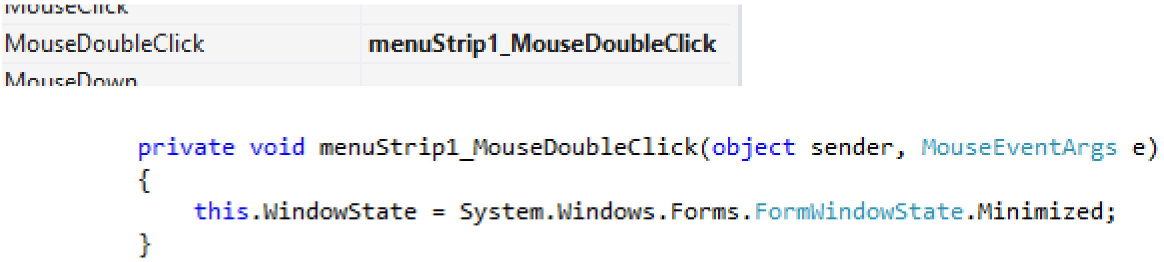
```
CheckStateChanged
Click                   purchaseToolStripMenuItem_Clic
DisplayStyleChanged
```

```
private void productCategoryToolStripMenuItem_Click(object sender, EventArgs e)
{
    ShowWindow(typeof(frmProductCategory));
}
```

Figure 11 Event Handler for Menu Stipe Click event

Mouse double click event

Similar to single mouse click event but it requires double mouse click to be triggered. This event is used in system for when user double click on menu stripe, it minimizes the solution.



MouseClick	
MouseDoubleClick	menuStrip1_MouseDoubleClick
MouseDown	

```
private void menuStrip1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    this.WindowState = System.Windows.Forms.FormWindowState.Minimized;
}
```

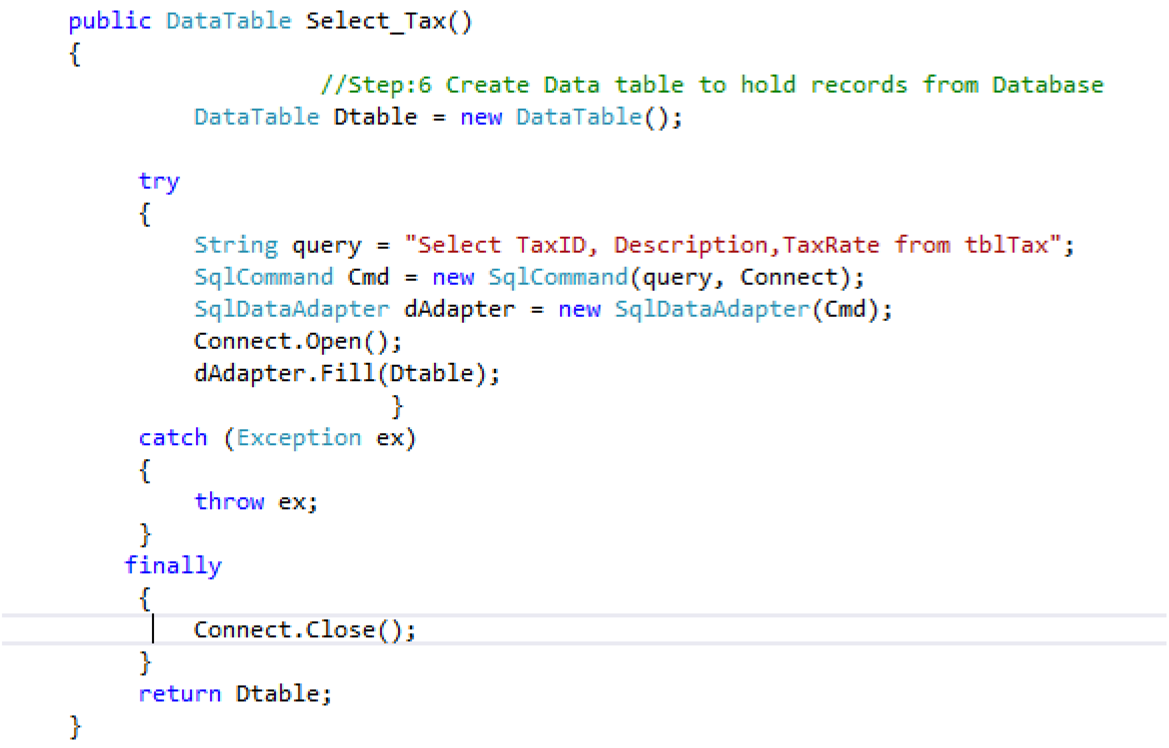
Figure 12 Event Handler for Menu stripe double mouse click event

Exceptional handling

Exceptional handling mechanism in programming helps to identify the error and help to reduce them. It also provides alternation to flow to the code when exceptional case arises. Brown Bird Management System (BrownBirdMS) has try-catch-finally mechanism as exceptional handler.

Try-catch-finally

In try-catch-finally mechanism, it has 3 different blocks; try section tries to execute codes within. If there is an error it direct the flow of code direct to catch block, if there isn't any error it finishes codes within the try block and direct the flow to finally section. In case of error occurrence, catch part executes codes within and directs the flow towards finally. Hence Even if there is error occur or not, codes within finally section always gets executed.



```
public DataTable Select_Tax()
{
                  //Step:6 Create Data table to hold records from Database
         DataTable Dtable = new DataTable();

     try
     {
         String query = "Select TaxID, Description,TaxRate from tblTax";
         SqlCommand Cmd = new SqlCommand(query, Connect);
         SqlDataAdapter dAdapter = new SqlDataAdapter(Cmd);
         Connect.Open();
         dAdapter.Fill(Dtable);
                       }
     catch (Exception ex)
     {
         throw ex;
     }
    finally
     {
         Connect.Close();
     }
     return Dtable;
}
```

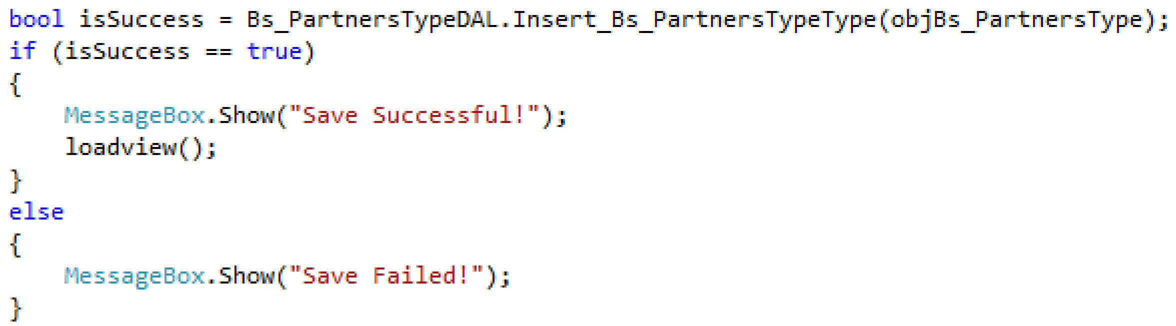
Figure 13 Implementation of Exceptional handling (Try-Catch-Finally) in the solution

Control Structure

Control Structure provides alternative for different cases. It allows defining what to be done and when to be done. For example it lets to decide, if condition is this what should be done and no condition is something else, what should be done in that case.

If-else

In BrownBirdMS there is use of if-else condition which allows to check the result and if result in not as required what action to be taken, or what action to be taken in different conditions. Codes below demonstrate use of If-Else condition where it is checking Boolean value stored in isSuccess. If value is positive it displays success message, if value of isSuccess is false, it displays unsuccessful message.



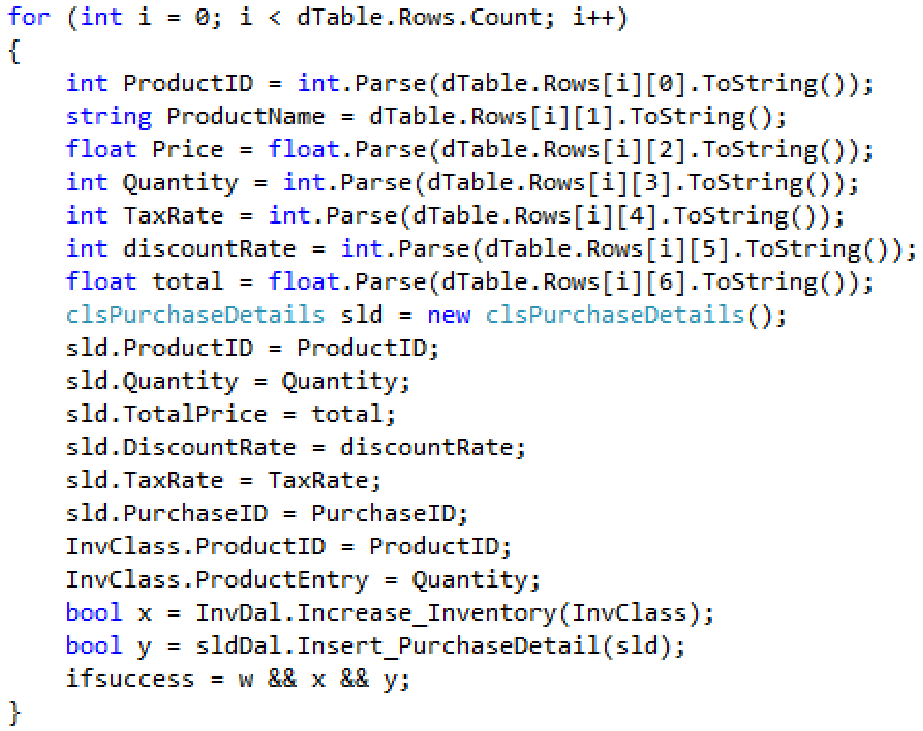
```
bool isSuccess = Bs_PartnersTypeDAL.Insert_Bs_PartnersTypeType(objBs_PartnersType);
if (isSuccess == true)
{
    MessageBox.Show("Save Successful!");
    loadview();
}
else
{
    MessageBox.Show("Save Failed!");
}
```

Figure 14 Use of Control Mechanism (IF-Else) in the system

Loops

Looping is repeating block of code until the defined condition matches. It is very simple yet useful mechanism. In current solution for brow bird department store, for loop is used to add products to the cart while purchasing or selling.

For

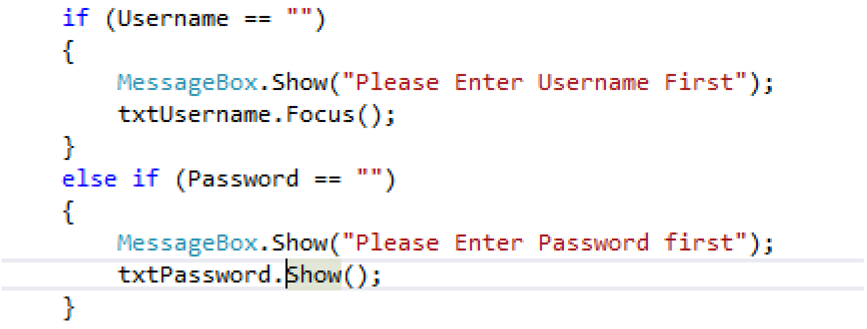


```
for (int i = 0; i < dTable.Rows.Count; i++)
{
    int ProductID = int.Parse(dTable.Rows[i][0].ToString());
    string ProductName = dTable.Rows[i][1].ToString();
    float Price = float.Parse(dTable.Rows[i][2].ToString());
    int Quantity = int.Parse(dTable.Rows[i][3].ToString());
    int TaxRate = int.Parse(dTable.Rows[i][4].ToString());
    int discountRate = int.Parse(dTable.Rows[i][5].ToString());
    float total = float.Parse(dTable.Rows[i][6].ToString());
    clsPurchaseDetails sld = new clsPurchaseDetails();
    sld.ProductID = ProductID;
    sld.Quantity = Quantity;
    sld.TotalPrice = total;
    sld.DiscountRate = discountRate;
    sld.TaxRate = TaxRate;
    sld.PurchaseID = PurchaseID;
    InvClass.ProductID = ProductID;
    InvClass.ProductEntry = Quantity;
    bool x = InvDal.Increase_Inventory(InvClass);
    bool y = sldDal.Insert_PurchaseDetail(sld);
    ifsuccess = w && x && y;
}
```

Figure 15: Implementation of looping statement in solution

Validation

Validation is process of insuring only required data is passed by user. For example in age field, validation blocks user to input alphabetical character. Validation process insures, only supported data type is passed to respective field. If user presses delete button without selecting any employee, it blocks user to delete employee. Similarly validation process is done in textbox event, click events, double click events etc. Below demonstrate use of validation in BrownBirdMS which block user to provide blank username or password when trying to login. It helps to prevent unnecessary error and complexity.



```
if (Username == "")
{
    MessageBox.Show("Please Enter Username First");
    txtUsername.Focus();
}
else if (Password == "")
{
    MessageBox.Show("Please Enter Password first");
    txtPassword.Show();
}
```

Figure 16: User of Validation in BrownBirdMS

Task 5 (4.1, 4.2)

Critically review and test the event driven programming solution and analyze actual test results against expected results to identify discrepancies.

Unit Test

Unit test describes testing of smallest testable part of application. For example save employee, update employee, delete employee are tested in current solution if they are working as the way they are expected to work.

Stress Test

Stress testing is the process of determining the ability of a system to work on unfavorable conditions (extreme) condition. For example, it tests, the outcome when large number of data is processes at single time. It test system in extreme number of data, users etc.

Integrated Test

Integrated test is used for checking the interaction between different components of software and search of defects. For Example, it tests if product is purchased does it increase the stock value in inventory table.

Compatibility Test

It is non-function test which checks system behavior in different conditions such as different operating system, different hardware or different user. This test clarifies if system runs or operates properly and do not care about its input/output.

Each of the tests done in system along with test result is shown in pages below.

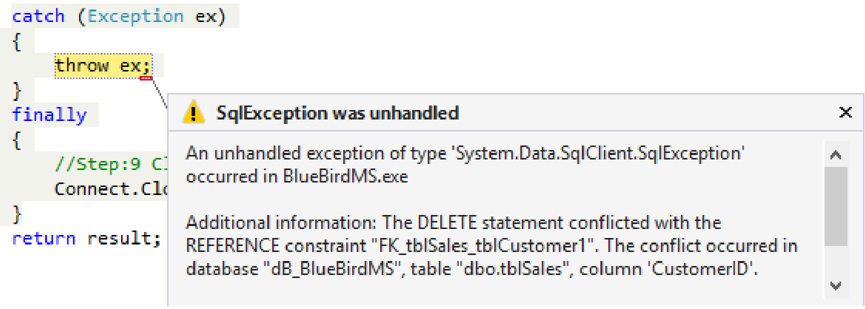
Critical Review of Test results

Unit Testing

For the unit test of the developed system, different units such as delete, update, sales, purchase, sale return etc. actions in system is tested. During tests, actual output during test and expected output were compared to achieve the test result.

Testing of Save and Update function in Customer form showed output result exactly as expected output which is successful save and update in database. Hence no error was found in that table. But during the test of delete function in customer form, result was not as per expected result. The error message popped up during testing of function during test stated, deleting is not possible due to the conflict of same customer id in sales table.

Similarly, testing of other functions such as sales, sales return, and inventory filter was performed. This test gave output similar to expected output.



```
catch (Exception ex)
{
    throw ex;
}
finally
{
    //Step:9 Cl
    Connect.Clc
}
return result;
```

SqlException was unhandled

An unhandled exception of type 'System.Data.SqlClient.SqlException' occurred in BlueBirdMS.exe

Additional information: The DELETE statement conflicted with the REFERENCE constraint "FK_tblSales_tblCustomer1". The conflict occurred in database "dB_BlueBirdMS", table "dbo.tblSales", column 'CustomerID'.

Figure 17 Error message during testing

Integrated Testing

Integration test on the developed solution is based on integration with database. Test is done by checking increment or decrement of product store number in inventory table. For testing, numbers in inventory is recorded before and after execution of sales function. Same was done for purchase function.

Test showed product is decreasing in inventory exactly same amount as the sale is done. Also, number of product is increasing exactly by the number of product purchased.

Stress Test

For stress testing large data was inserted into solution’s data table. Random data generated in generatedata.com was inserted into customer and product table. Data were inserted successfully which showed system can work on stressful condition.

Compatibility Testing

For Compatibility testing solution was run under two different environment and tested how they behave in different working condition. For that, solution has executed with two different users. ADMIN user displayed all feature of the system. This showed ADMIN has full authorization in the solution while non-admin user showed only few and basic features in the system. This showed non-admin user has access to only basic features.

Compatibility test showed two different users have access to two different levels of features. Test result highlights that system performs different and properly in different environment.

Task 6 (4.3, D2)

Evaluate independent feedback on a developed event driven programming solution and male necessary recommendation for improvement.

Scenario Requirement

Brown Bird Department Store is a supermarket that sells products of several categories such as electronics, gift, cloths etc. and located at various places of Kathmandu. Department store was managing its daily processes in Microsoft excel but due to it limitation, Brown bird was facing several issues. They needed an automated system to manage their daily process. The developed system has several key features such as:

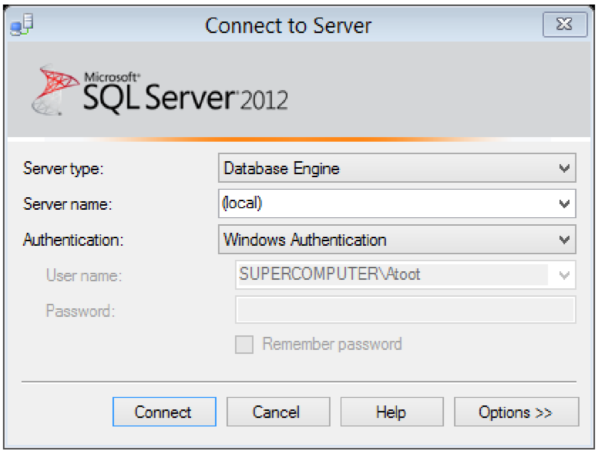
- Customer, Employee and Product Management System
- Sales/Purchase Management
- Inventory Management
- User Authentication Control
- Sales/Purchase History

This report evaluates independent feedback on developed event driven solution. To make evaluation various criteria are taken into consideration such as security, User Interface, scope management and system development cost management.

Security

Developed system (BrownBirdMS) has taken good care of security. It uses strongly typed parameterized queries. That blocks attacker performing SQL injection. It also has user authentication system, user need to provide user name and password before using the system. Only admin has full access to the system other user has limited privilege and allowed to perform much specified tasks only. Additionally, for security of data, system for confirmation before deleting row from table, which gives warning before deleting something unintentionally.

Password provided by user is encrypted before being saved into database. In addition to that, SQL server has another security level should anyone tries to access database through SQL server. Unless system is attacked by highly skilled attacker, it is very much secure.



Server type: Database Engine
Server name: (local)
Authentication: Windows Authentication
User name: SUPERCOMPUTER\Atoot
Password:
□ Remember password

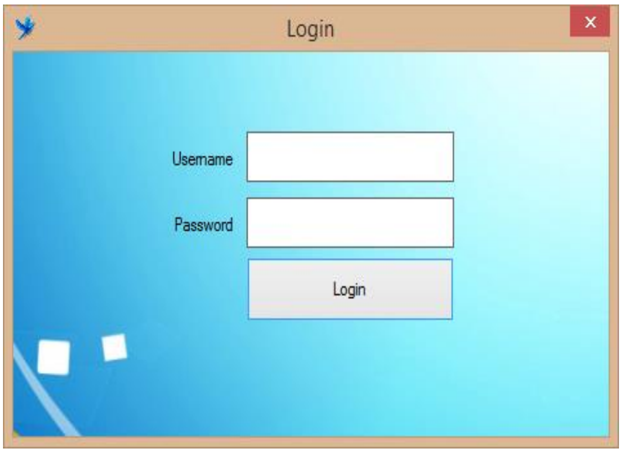


Figure 18 SQL Server Login

Figure 19 Login Feature for the BrownBirdMS

Limitation and Feedback

After the evaluation of the system, it is found that BrownBirdMS has no back up feature. It is only possible to take back up through SQL server. But it lacks its own backup process which can cause issue if something goes wrong in daily process.

Implementation of Back up feature is system will allow data to be safer. In case of data lost, backed up file can be restored easily. Automated backup can be additional feature.

User Interface

BrownBirdMS is developed in visual studio using C Sharp Win Forms technology. Windows GDI+ (GDIPLUS) libraries are used to develop UI. This technology allows developer to develop windows compatible simple, user friendly and interactive software.

Event driven programming makes this system much comfortable and productive. Performing tasks such as sales, purchase, customer management, inventory search are very simple. Use of data grid view helps to displays tabular data easily. In addition to that, if client find something to be changed in UI, designer can easily change it.

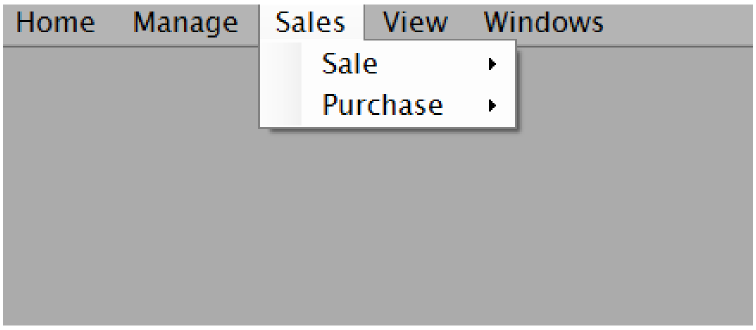
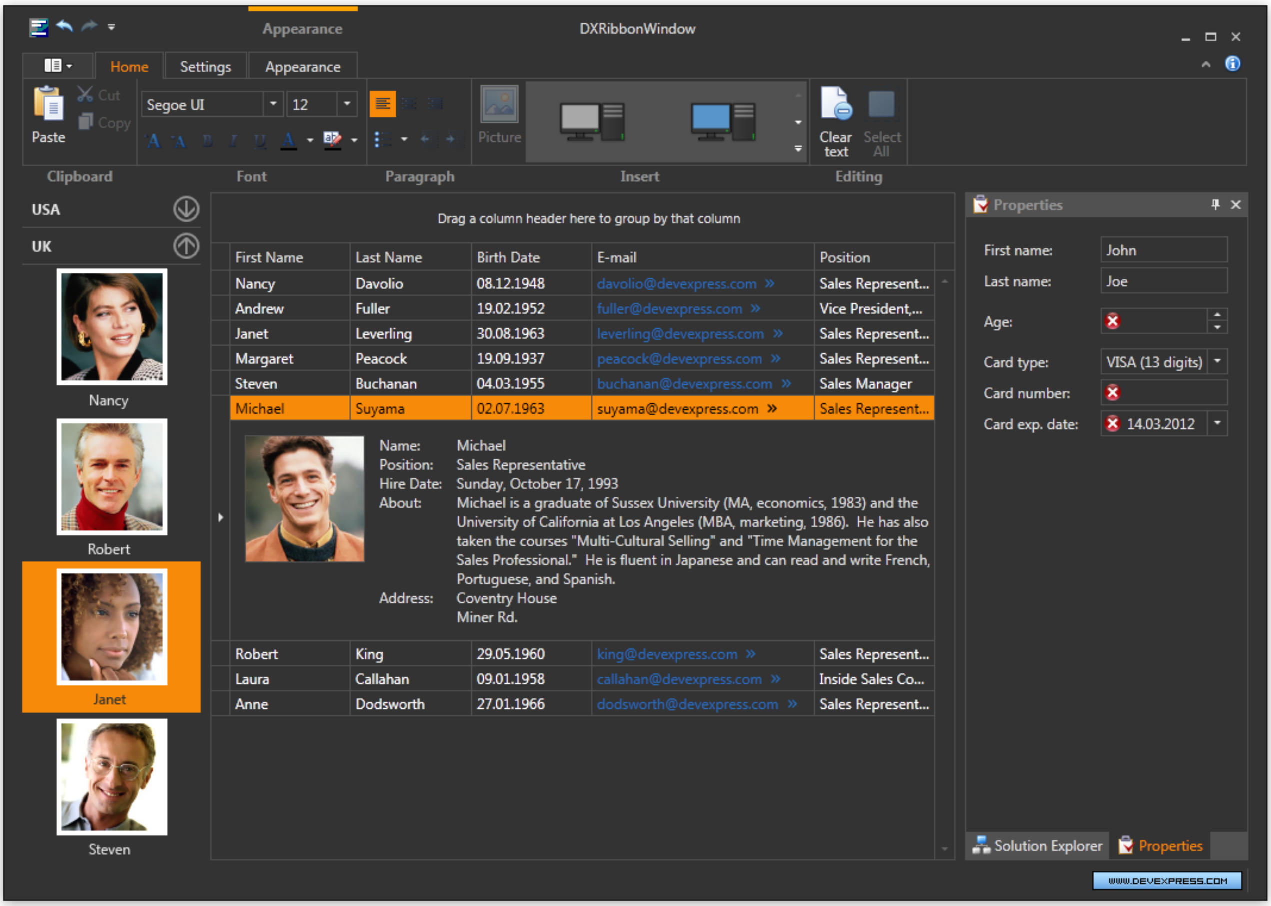


Figure 20 UI OF the developed Solution (BrownBirdMS)

Limitation and Feedback:

Despite having attractive user interface, it still haven't got modern rich look. Additionally, when loading large data into grid view it can hamper solution's performance. If modern technology such as WPF with C# is used to develop such event driven solution, UI would be more attractive and rich in look. WPF has UI virtualization feature that help to maintain system performance even during loading large database. But before developing system in WPF it is necessary to keep in mind that it does not support windows 2000 or earlier versions.



Drag a column header here to group by that column

First Name	Last Name	Birth Date	E-mail	Position
Nancy	Davolio	08.12.1948	davolio@devexpress.com »	Sales Represent...
Andrew	Fuller	19.02.1952	fuller@devexpress.com »	Vice President,...
Janet	Leverling	30.08.1963	leverling@devexpress.com »	Sales Represent...
Margaret	Peacock	19.09.1937	peacock@devexpress.com »	Sales Represent...
Steven	Buchanan	04.03.1955	buchanan@devexpress.com »	Sales Manager
Michael	Suyama	02.07.1963	suyama@devexpress.com »	Sales Represent...
Robert	King	29.05.1960	king@devexpress.com »	Sales Represent...
Laura	Callahan	09.01.1958	callahan@devexpress.com »	Inside Sales Co...
Anne	Dodsworth	27.01.1966	dodsworth@devexpress.com »	Sales Represent...

Name: Michael
Position: Sales Representative
Hire Date: Sunday, October 17, 1993
About: Michael is a graduate of Sussex University (MA, economics, 1983) and the University of California at Los Angeles (MBA, marketing, 1986). He has also taken the courses "Multi-Cultural Selling" and "Time Management for the Sales Professional." He is fluent in Japanese and can read and write French, Portuguese, and Spanish.
Address: Coventry House
Miner Rd.

Properties

First name: John
Last name: Joe
Age: ☒
Card type: VISA (13 digits)
Card number: ☒
Card exp. date: ☒ 14.03.2012

Figure 21 UI rich Application Developed with WPF (Source: www.devexpress.com)

Scope Management

Scope management refers to analysis and management of potential impact of system to its potential consumer, stakeholders. BrownBirdMS has several direct and indirect stakeholders that will be affected by the implementation of this solution.

- Scope for Department Store

  Implementation of event driven solution will directly and instantly affect department store and its employees. Daily tasks such as data entry, calculation, payment etc. will be much easier and quicker. This will help to generate more revenue as it can now serve more customers in same time cost. It will also help to reduce expenses on large man power as its automated system and require less man power.

- Scope for Customer

  Customer will also directly and indirectly affected by the system. Payment process will take much less time and it will save time. Since department store's revenue is increased, they can expect various discount offers, sale packages etc.

- Scope for Society

  This system has indirect scope in the society. It will help to develop IT oriented minded customers, employees and other stakeholders, which will eventually help to develop IT oriented society.

Limitations and Recommendation

Even though this solution has high scope for its stake holders there are still some areas improving which can bring greater scope for its stake holders. Some of the key limitation that is holding this system to get higher score is:

1. System does not store cash information about company
2. Does not support swipe credit/debit card

To improve these limitations solution can add more features that can store cash information about the company. Additionally more mechanism needs to be added in the system that can accept the debit/credit card information. For these upgrades first thing need to be done is software should be a web supported solution as direct transaction is done through bank.

Software Development Life Cycle

To develop and implement software system properly, it has to go through process called Software development life cycle.

Cost Management

During this independent feedback evaluation, cost management is another topic that is taken into consideration. To calculate total cost of the system, expenses done in each of phase of the cycle are calculated.

S.N.	Task	Work Force Rate	Total Hour Required	Sub Total

1.	Planning	5$ /HR	20 HR	100$
2.	Analysis	10$ HR	15 HR	150$
3.	Design	6$ HR	10 HR	60$
4.	Implementation	4$ HR	200 HR	800$
5.	Maintenance	100$/YR	-	100$/YR
Total				1110$+100/YR

Feedback

This system is developed by a single person. This makes development phase longer and more expensive. If more skilled workers get involved into this project, development can be much cheaper and would take much lesser time to complete.

Quality Control

After full evaluation of the system, there were some limitation were found in the system. To reduce or eliminate those limitations, feedback and suggestions were given. But this system still lacks proper quality testing and assurance. To make this solution up to market standard, it has to go through Quality Control (QC). QC ensures the quality of the system and its features (wisegeek (n.d.). It tests the system and gives feedback to developer, who works on improving the system and deploy defect less system. Quality Control itself has two modes.

1. **Quality Assurance**

   Quality Assurance (**QA**) is a method to prevent error or defects in developed system and ensure smooth delivery to target. It is used for ensuring system is mistake less. Performing QA for BrownBirdMS will ensure system is deficiencies free when delivered to its client.

2. **Quality Check**

   Quality check is similar to QA but not exact same. This process ensures system is up to specified set of quality of client's requirements. This process checks if product is as per client's demand or not when they get the product in their hand.

Hence, Quality Control for developed event driven programming will assure client will get error less, up to market standard and as per client's requirement.

Reference

- Wisegeek (n.d.) *What is quality control* Available: **http://www.wisegeek.org/what-is-quality-control.htm** Accessed [9/30/2014]

Task 7 (4.4)

Create onscreen help to assist the users of a computer program.

This document will serve as onscreen help for system operator of this solution. This document guides user to perform various general as well as transaction and management tasks. This help document is divided into 3 major categories according to task to be performed.

General Help

Staring Up Solution

1. Double Click on solution logo from desktop (Fig:1)
2. Wait for few seconds to splash form auto close (Fig:2)
3. When logon page is displayed, provides credentials.(Fig:3)
4. Press Enter key or click on “Login”.
5. Main Page starts, done.

Fig: 1

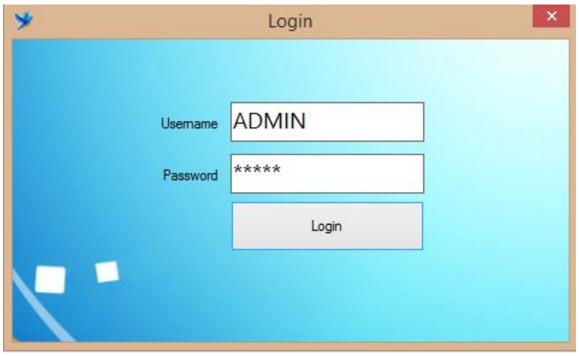


Fig: 3

Fig: 2

Exit Solution

1. Click on Home Menu form main page.
2. Click on Exit. (Fig:4)

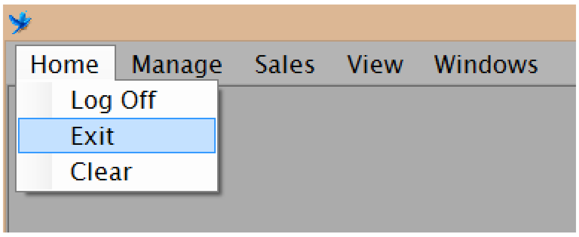


Fig: 4

Login Off

1. Click on Home Menu
2. Click on log off. (Fig:5)

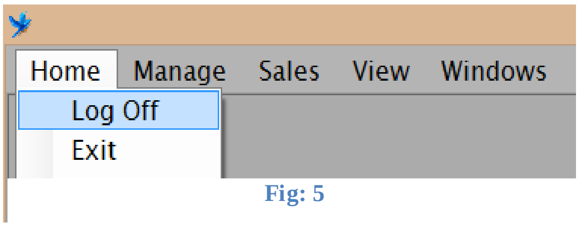


Fig: 5

Manage opened windows

To manage opened several pages follow these steps:

1. Click on Windows
2. Click on required style to manage opened windows accordingly. (Fig:6, Fig:8)

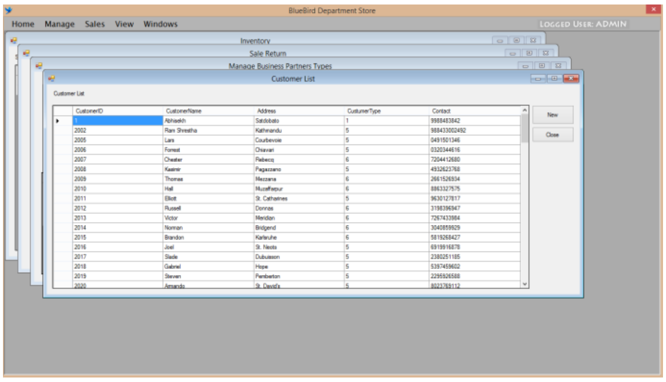


Fig: 6 Managed with Cascade Style

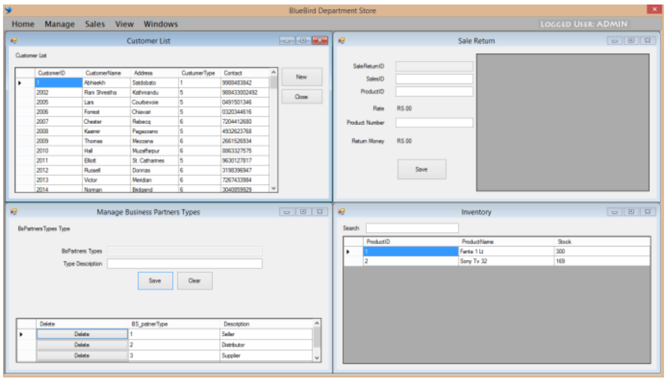


Fig: 8 Managed with Vertical Style

Management Help

View Customer

1. Click on Manage menu and go to customer. (Fig:7)

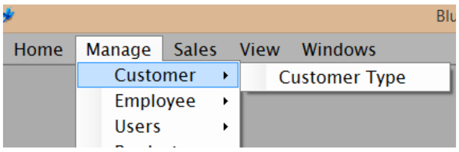
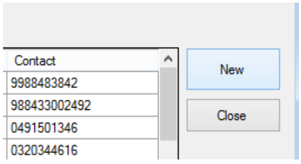


Fig: 7

Save

1. Click on New from View Customer form (Fig:9)
2. Provide required information.
3. Click on Save. Once success message displayed, click ok. Done. (Fig:10)

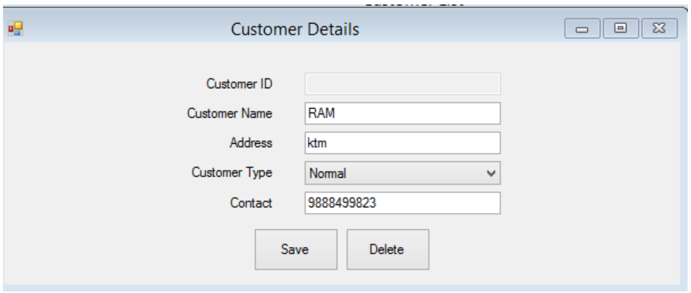


Contact
9988483842
988433002492
0491501346
0320344616

Fig: 9

Update

1. From View customer form, double click on row header of customer to be updated. (Fig: 11)
2. Edit the required information and click on save.
3. Once update successful message



Customer ID

Customer Name RAM

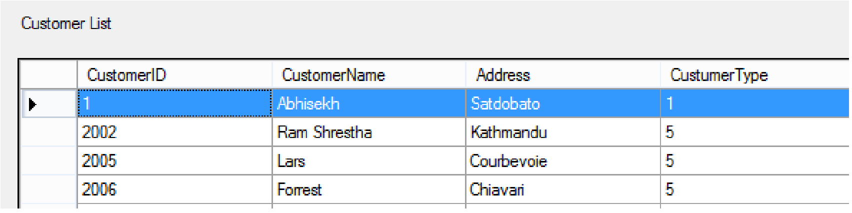
Address ktm

Customer Type Normal

Contact 9888499823

Fig: 10

displayed, click ok. Done.



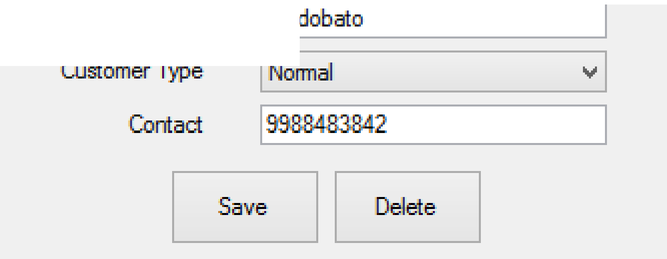
Customer List

	CustomerID	CustomerName	Address	CustumerType
▶	1	Abhisekh	Satdobato	1
	2002	Ram Shrestha	Kathmandu	5
	2005	Lars	Courbevoie	5
	2006	Forrest	Chiavari	5

Fig: 11

Delete

1. From View customer form, double click on row header of customer to be deleted.
2. Click on delete button in customer details form. Ok. Done. (Fig:12)



dobato

Customer Type: Normal

Contact: 9988483842

Fig: 12

Customer Type

1. Click on Manage
2. Then click on Customer→ Customers Type (Fig:13)

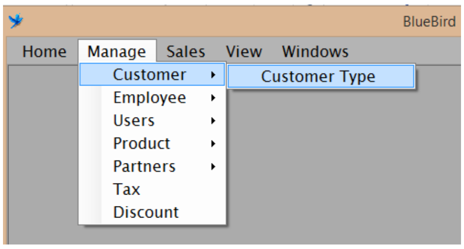
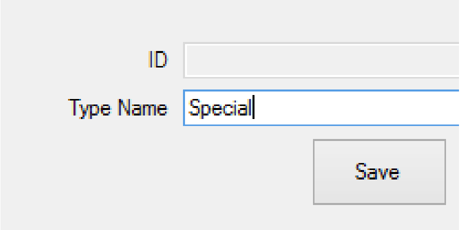


Fig: 13

Save New Customer Type

1. Provide required information
2. Left click on save button. (Fig:14)
3. Save successful message appears. Click on OK. Done



ID:

Type Name: Special

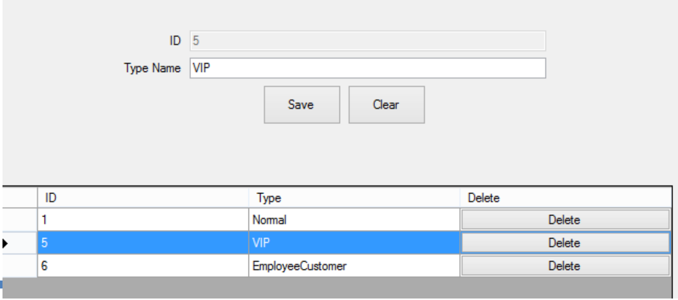
Fig: 14

Update Customer Type

1. Left click on row header of the customer type to be updated in data grid view. (Fig:15)
2. This loads grid view row data into UI controls.
3. Make necessary changes in customer type description.
4. Click on save button to update the selected customer type.
5. Done

Delete Type

1. Click on delete button from grid view to delete required row. (Fig:17)



ID: 5

Type Name: VIP

	ID	Type	Delete
	1	Normal	Delete
▶	5	VIP	Delete
	6	EmployeeCustomer	Delete

2. Done

Fig: 15

Clearing UI controls

1. Click on Clear button. (Fig:16)

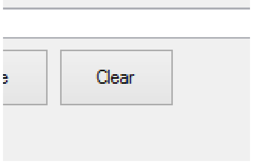
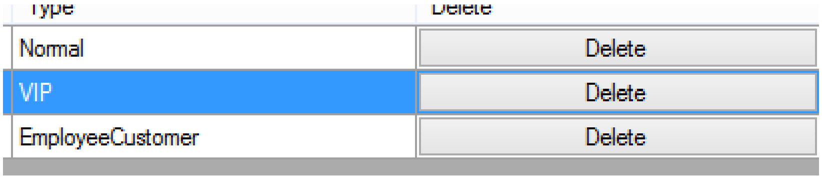


Fig: 16



Type	Delete
Normal	Delete
VIP	Delete
EmployeeCustomer	Delete

Fig: 17

Tax

1. Click on Manage menu
2. Click on Tax (Fig:18)

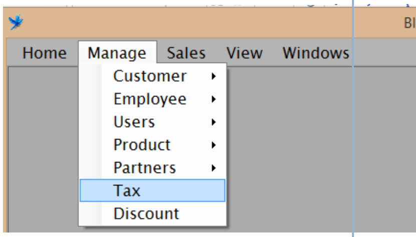
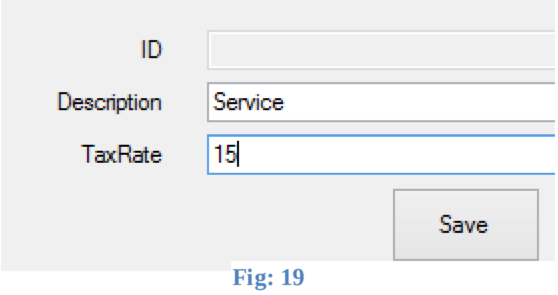


Fig: 18

Save Tax

1. Provide required information (Fig:19)
2. Click on save button. Done



ID

Description Service

TaxRate 15

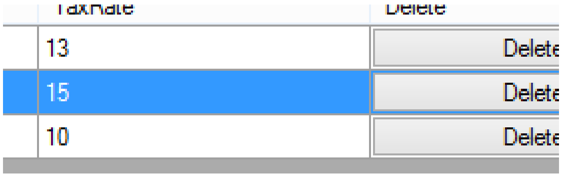
Fig: 19

Delete

1. Click on delete button from grid view to delete required row.
2. Done (Fig:21)

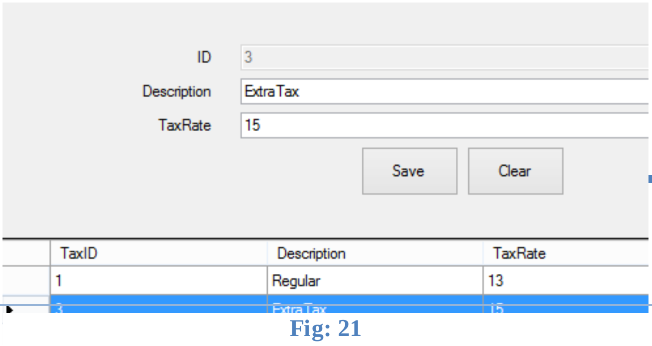
Update

1. Left click on row header of the Tax to be updated in data grid view.
2. This loads grid view row data into UI controls.
3. Make necessary changes in tax information.
4. Click on save button to update the selected tax. (Fig:20)
5. Done



TaxRate	Delete
13	Delete
15	Delete
10	Delete

Fig: 20



ID 3

Description ExtraTax

TaxRate 15

TaxID	Description	TaxRate
1	Regular	13
3	ExtraTax	15

Fig: 21

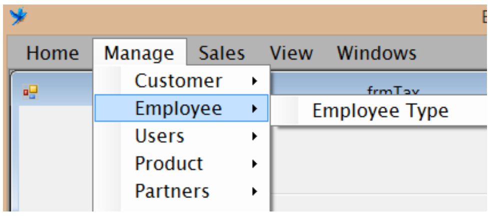


Fig: 22

Employee

1. Go to Manage menu
2. Click on employee (Fig:22)

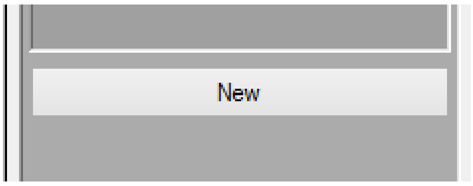
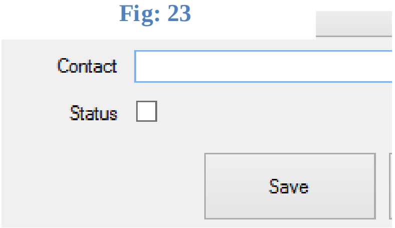


Fig: 23

Save

1. Click on new (Fig:23)
2. Provide required information.
3. Click on Save
4. Done



Contact
Status □

Fig: 24

Delete

1. Double click on row header of employee to be deleted.
2. In new form, click on delete button. (Fig:25)

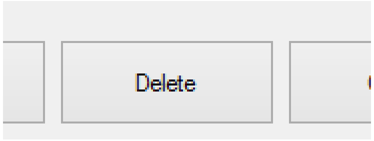
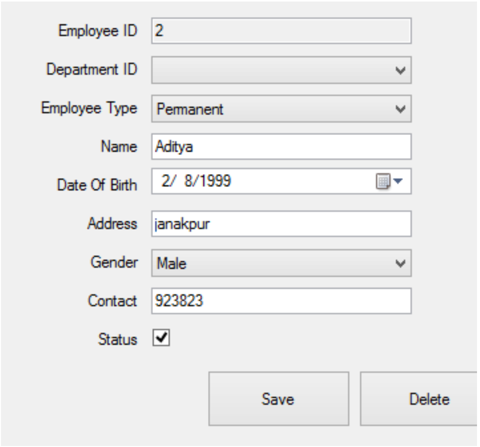


Fig: 25

Update

1. Double click on row header of employee to be deleted.
2. Edit required information.
3. Click on save button to update. (Fig:26)
4. Done.

To perform other management tasks in the system please follow similar steps as above. Similar steps are Valid for form listed below.



Employee ID 2
Department ID
Employee Type Permanent
Name Aditya
Date Of Birth 2/ 8/1999
Address janakpur
Gender Male
Contact 923823
Status ☑

Fig: 26

- Tax→ Discount Form
- Customer Type→ Employee Type/ Product Type/ User

Type/User/Partners Type

- Customer→ Product/Partners

Transaction Help

Sales

1. Go to Sales menu.
2. Click on Sales

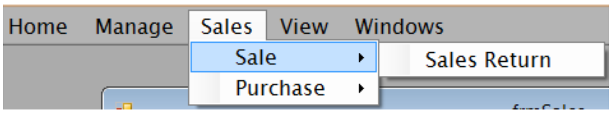
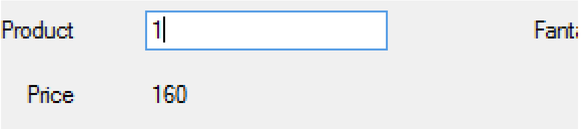


Fig: 27

Search Product

1. Type to product ID to filter the product. (Fig:28)

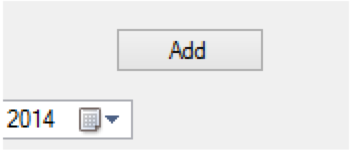

Product 1 Fant

Price 160

Fig: 28

Add to cart (grid view)

1. Search required product
2. Fill other required fields such as date, customer id, item number to be sold etc.
3. Click on add button, product will added to grid view(cart) (Fig:29)
4. Once UI control fields are cleared automatically, add another products as per requirements



2014

Fig: 29

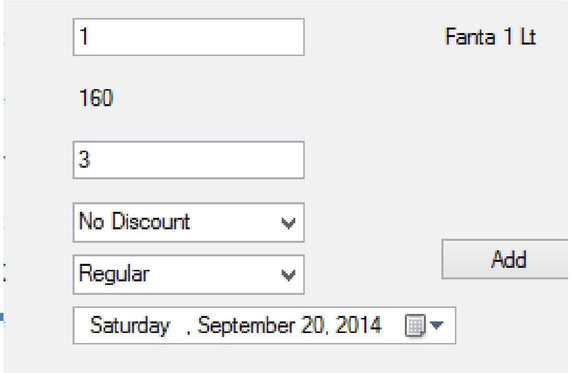
Save

1. Once all required products are added to grid make sure, customer and Date is correct,
2. Click on save button to finish the transaction. (Fig:30)
3. Click ok on success message. Done.

Fig: 30

Editing/Deleting item from cart (grid view)

1. If item added in grid view need to be modified or deleted, double click on row header of required item. (Fig:32)
2. This will delete that row form grid view and load data inside UI control fields.
3. To delete Clear the field or to update edit as per requirement and click on add button. (Fig:31)
4. Done.


1 Fanta 1 Lt

160

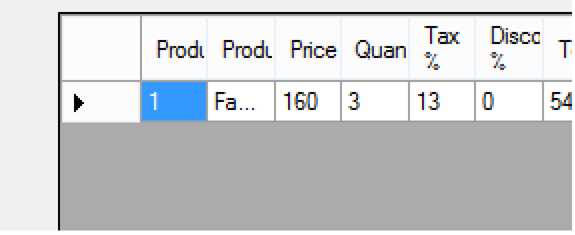
3

No Discount

Regular

Saturday , September 20, 2014

Fig: 31



	Produ	Produ	Price	Quan	Tax %	Disco %	T
▶	1	Fa...	160	3	13	0	54

Fig: 32

Sales Return

1. Go to Sales Menu
2. Then click on sales->sale return (Fig:33)
3. Done

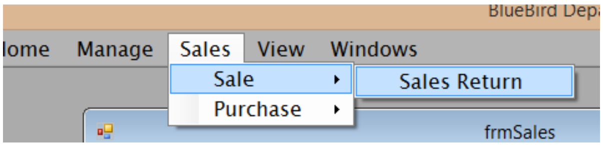
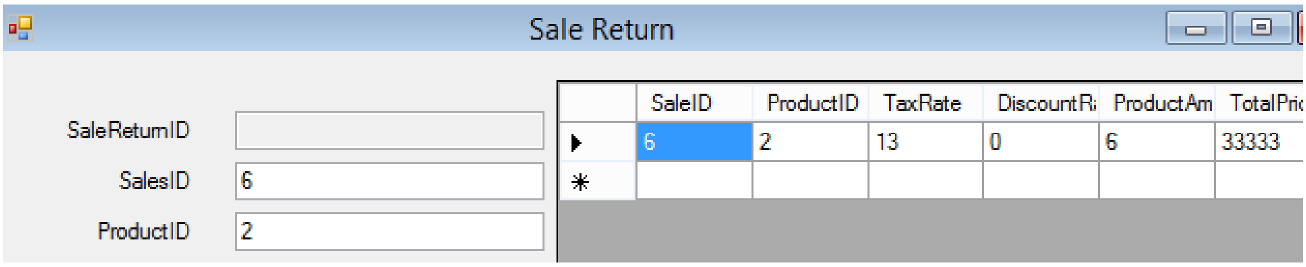


Fig: 33

Search sold Product

1. To search required product to be returned, type sale id in the textbox labeled with sale id.
2. Now type product id in textbox labeled with product id.
3. Grid view will display the sold product (Fig:34)



Sale Return

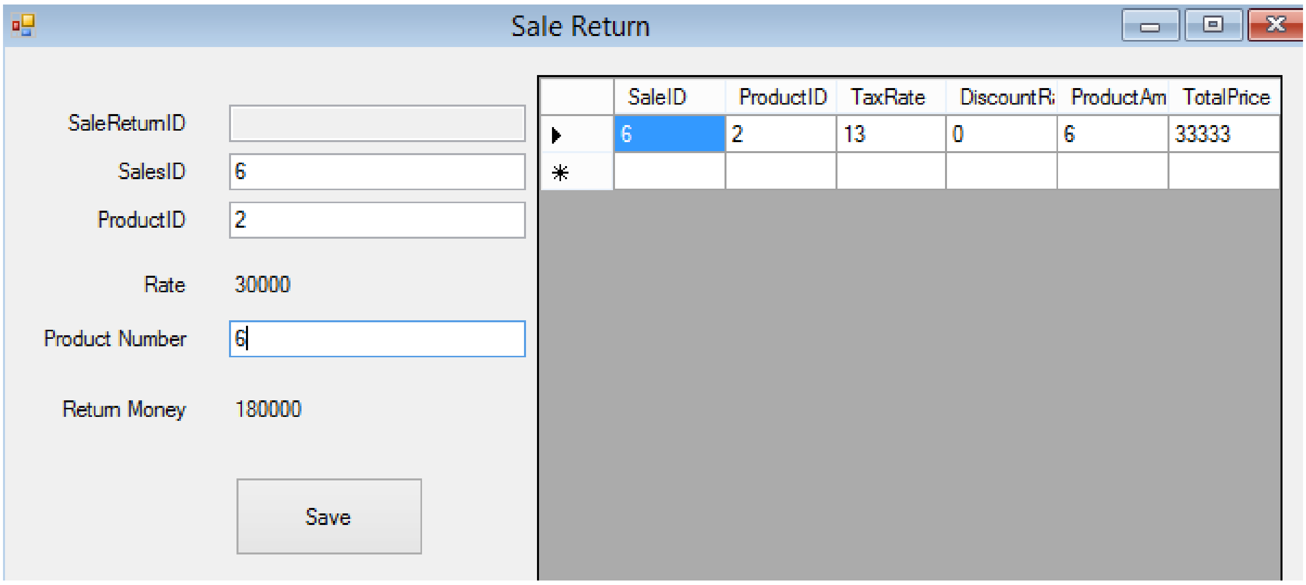
SaleReturnID	
SalesID	6
ProductID	2

	SaleID	ProductID	TaxRate	DiscountR	ProductAm	TotalPric
▶	6	2	13	0	6	33333
*						

Fig: 34

Save

1. Search the product
2. Enter remaining details
3. Click ok save button to finish sales return.(Fig:35)
4. Done.



SaleReturnID

SalesID 6

ProductID 2

Rate 30000

Product Number 6

Return Money 180000

	SaleID	ProductID	TaxRate	DiscountR	ProductAm	TotalPrice
▸	6	2	13	0	6	33333
*						

Fig: 35

Inventory

1. Click on View Menu
2. Click on Inventory. Done (Fig:36)

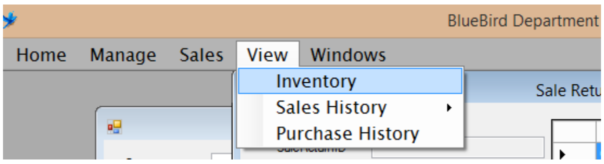
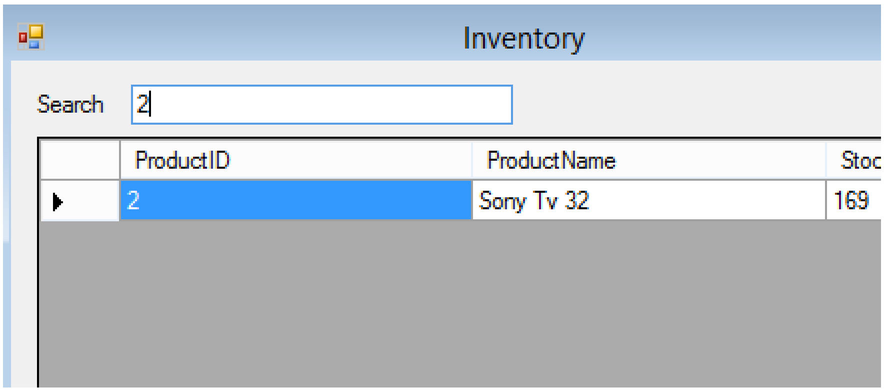


Fig: 36

<u>Filter</u>

1. Once inventory form opens, type the Product ID to filter required filter.
2. Done. (Fig:37)



Search 2

	ProductID	ProductName	Stoc
▸	2	Sony Tv 32	169

Fig: 37

To perform other transaction tasks in the system please follow similar steps as above. Similar steps are Valid for form listed below.

- Sales→ Purchase
- Sale Return→ Purchase Return
- Inventory→ Sales history/Purchase History/Sales Return history/ Purchase Return history/

This document provided guidance for performing several tasks and also listed tasks that can be done by performing similar actions. If user still finds issue with performing certain task in the system please contact IT department or email at support@atut.net.

Task 8 [4.5, M3]

Create documentation for support and maintenance of a computer program.

Introduction:

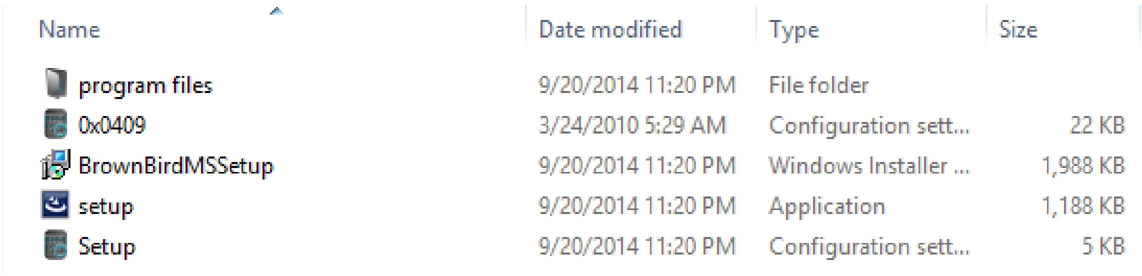
This document provides guidance for support and maintenance task of the BrownBirdMS. BrownBirdMS software is developed for Brown Bird Department store to help them with their daily process. To support and maintain this software IT technician can follow this document as guidance. This documentation covers support for layout design, choice of tools, product deployment and maintenance techniques.

Product Deployment

To deploy product first download setup file from official site or buy it from store. For installation process follow these steps.

1. Make sure system has required .Net Framework
2. Double click on Setup.exe and follow the screen steps.
3. Finish installation.
4. Open SQL Server
5. Restore back up file provided inside Setup folder.
6. Locate app.config file form installation location.
7. Edit connection string to match with the system.

8. Done.



Name	Date modified	Type	Size
program files	9/20/2014 11:20 PM	File folder	
0x0409	3/24/2010 5:29 AM	Configuration sett...	22 KB
BrownBirdMSSetup	9/20/2014 11:20 PM	Windows Installer ...	1,988 KB
setup	9/20/2014 11:20 PM	Application	1,188 KB
Setup	9/20/2014 11:20 PM	Configuration sett...	5 KB

Fig: 38

Choice of tools

To develop this system, different tools were used. Below choice of different tools and their purpose are listed.

1. Visual Studio 2013: To develop event driven solution in C# for given problem, visual studio is used. Visual studio 2013 provides better features and development experience.
2. SQL server 2012: For database development and management SQL server is used.

Data and information collection

1. Users are asked to provide necessary information and then specified task is done as per user instructions.
2. If client require additional field to be saved in database follow these steps.
    - Start SQL Server
    - Provide credentials and browse through the required table.
    - Add new column as per requirement. Save table.
    - Start system solution in visual studio.
    - Modify, Data Access layer, Property class and UI
    - Build the solution. Done.

Designing layout

This system is designed using C# windows form. Hence, designing and modifying layout is very simple. To modify UI style or position of UI control, these steps below should be followed is guidance.

1. Open Solution in Visual studio
2. Open required form to be modified

3. Modify the position of controls or add new control from tools as required.
4. Build the solution. Done

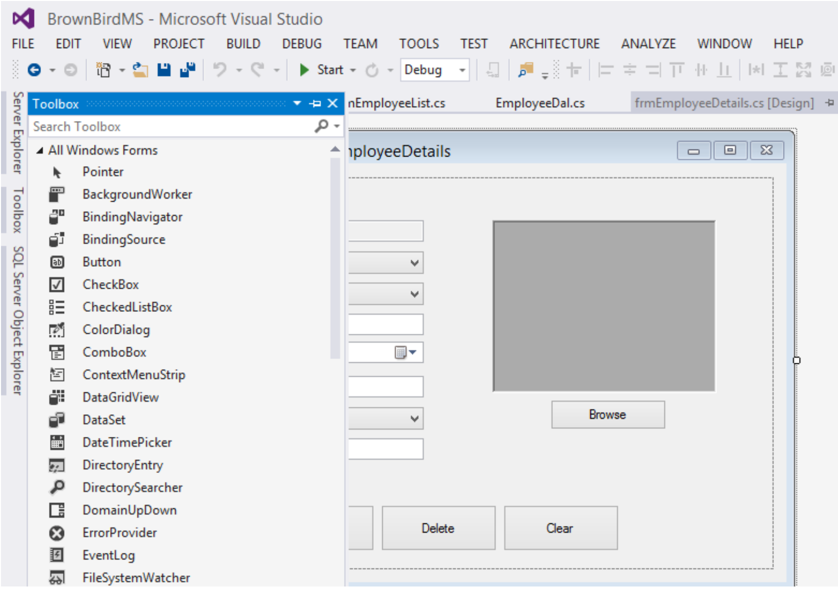


Fig: 39

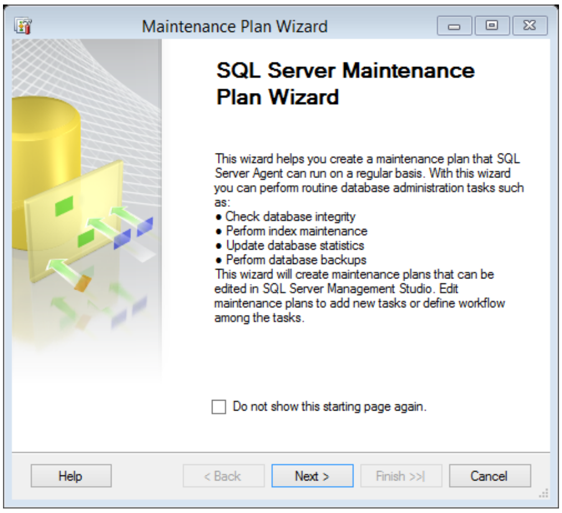
Maintenance

1. Read User Manual for operation helps of the system.
2. Since, this solution does not have its inbuilt data backup feature, Follow these steps to setup auto backup feature manually.

- Start SQL Server
- Right Click on Maintenance Plan
- Choose wizard
- Click next, in new Page Provide Plan name, Change schedule type to full back up.
- Next, Fill required fields, in database choose required database only (dB_BlueBirdMS).
- Click Next, Then Finish.

3. If for some reason, data is lost restore database manually from latest backup file.

Above in this document, supporting and maintaining the system in several situations is covered. If necessary please contact IT administration or mail at support@atut.com.



Maintenance Plan Wizard

SQL Server Maintenance Plan Wizard

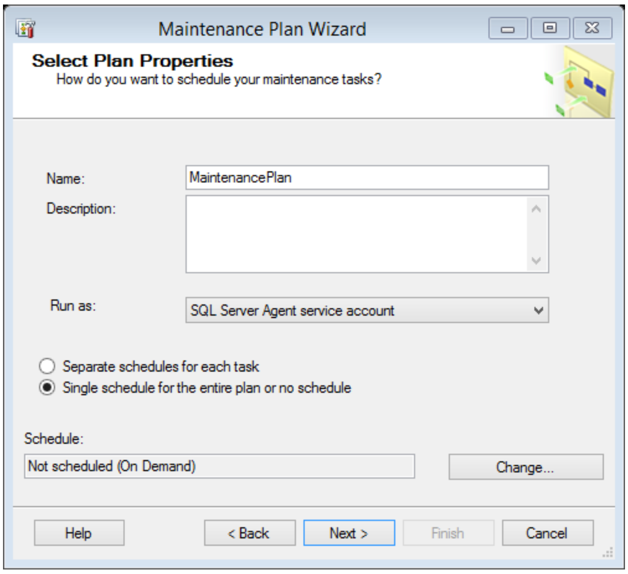
This wizard helps you create a maintenance plan that SQL Server Agent can run on a regular basis. With this wizard you can perform routine database administration tasks such as:

- Check database integrity
- Perform index maintenance
- Update database statistics
- Perform database backups

This wizard will create maintenance plans that can be edited in SQL Server Management Studio. Edit maintenance plans to add new tasks or define workflow among the tasks.

□ Do not show this starting page again.

Fig: 3



Maintenance Plan Wizard

Select Plan Properties

How do you want to schedule your maintenance tasks?

Name: MaintenancePlan

Description:

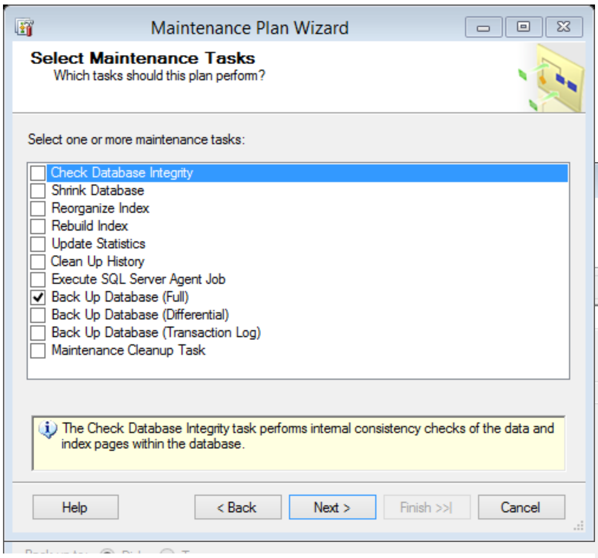
Run as: SQL Server Agent service account

○ Separate schedules for each task

● Single schedule for the entire plan or no schedule

Schedule: Not scheduled (On Demand)

Fig: 4



Maintenance Plan Wizard

Select Maintenance Tasks

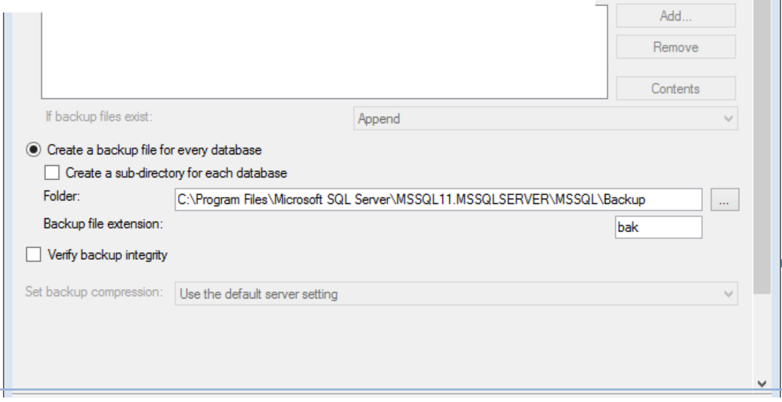
Which tasks should this plan perform?

Select one or more maintenance tasks:

- □ Check Database Integrity
- □ Shrink Database
- □ Reorganize Index
- □ Rebuild Index
- □ Update Statistics
- □ Clean Up History
- □ Execute SQL Server Agent Job
- ☑ Back Up Database (Full)
- □ Back Up Database (Differential)
- □ Back Up Database (Transaction Log)
- □ Maintenance Cleanup Task

The Check Database Integrity task performs internal consistency checks of the data and index pages within the database.

Fig: 5



If backup files exist: Append

● Create a backup file for every database

□ Create a sub-directory for each database

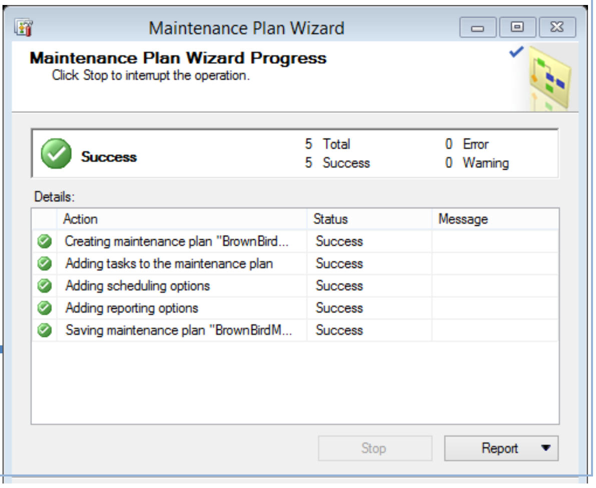
Folder: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup

Backup file extension: bak

□ Verify backup integrity

Set backup compression: Use the default server setting

Fig: 6



Maintenance Plan Wizard

Maintenance Plan Wizard Progress

Click Stop to interrupt the operation.

Success

5 Total 0 Error

5 Success 0 Warning

Details:

Action	Status	Message
Creating maintenance plan "BrownBird...	Success	
Adding tasks to the maintenance plan	Success	
Adding scheduling options	Success	
Adding reporting options	Success	
Saving maintenance plan "BrownBirdM...	Success	

Fig: 7