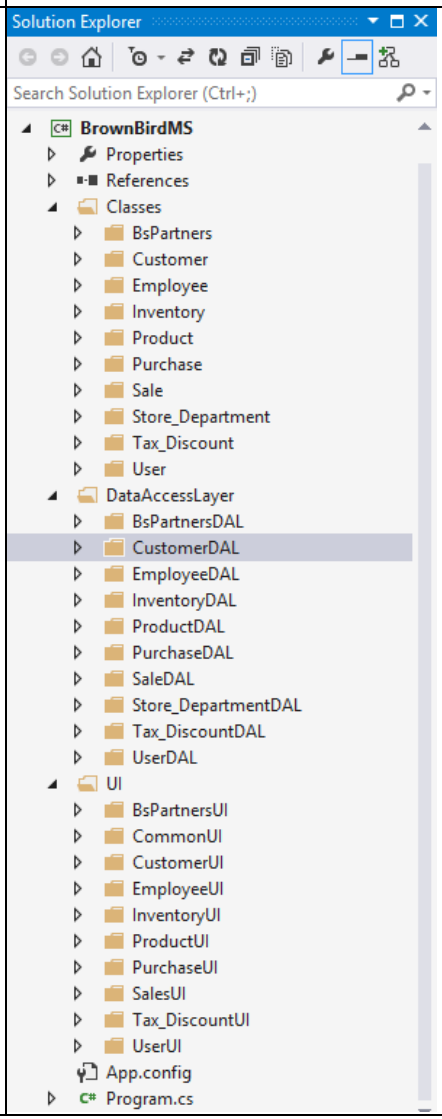
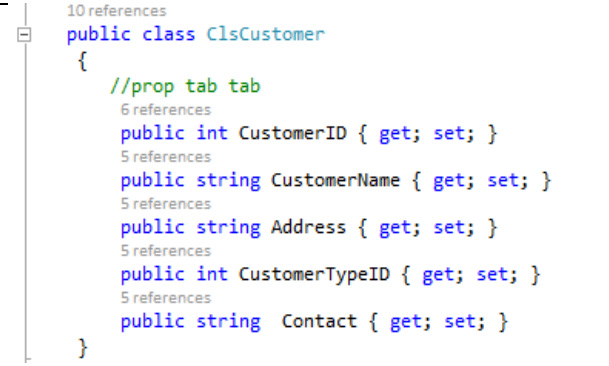
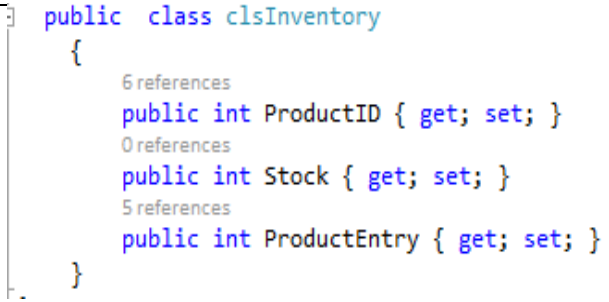
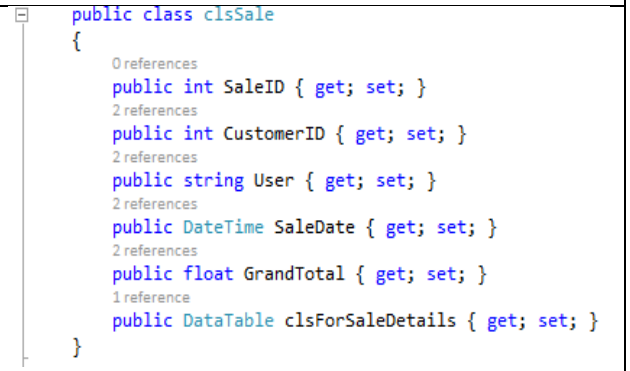
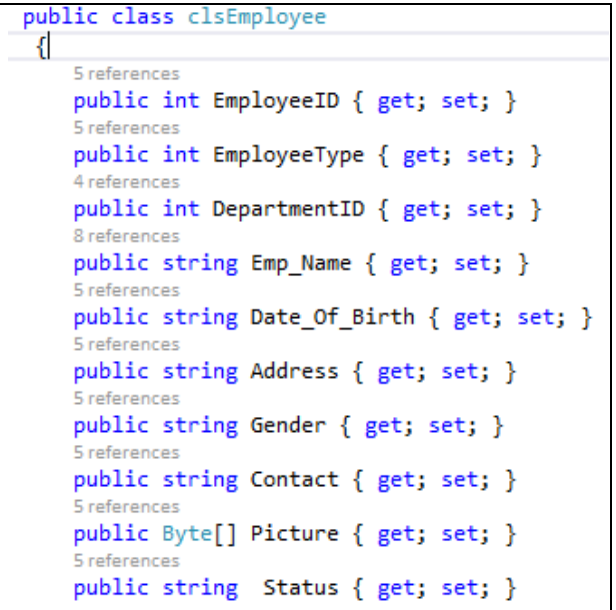
S.N.	Name	Purpose of use of the component	Screen shots
1.	**Whole Solution file structure**	The image at right is screen shot of the file tree for the whole solution. It shows required properties classes, UI files, Data access layer class and other files. It is structure of required solution to be implemented. Each component of the sytem like sales, purchase, costumer, employee etc. has their own separed, UI and classes. This makse solution more effieciet, durable and maintainable.	

Properties Structure			
2.	Customer	This class contains properties requried to calculate, read and write data from customer table. Each properties will be used for inserting, updating or deleting customer data. To save new customer, values for different column of customer table will be taken form UI and saved them temporarly in clscustomer. Then, with help of DAL it will be saved in the customer table. Other operations will done similarly.	
3.	Inventory	Like for customer class, clsInventory contains properties required to delete, update and insert into Inventory table. Data to be inserted or data taken from inventroy table can be saved temporarly in clsInventory. This makes development process much easier.	
4.	Sale	Use of clsSale is similar to other properties classes. It contains, saleID, CustomerID, User, SaleDate, Grand Total and datatable for clsSaleDetails.	

```
10 references
public class ClsCustomer
{
    //prop tab tab
    6 references
    public int CustomerID { get; set; }
    5 references
    public string CustomerName { get; set; }
    5 references
    public string Address { get; set; }
    5 references
    public int CustomerTypeID { get; set; }
    5 references
    public string  Contact { get; set; }
}
```

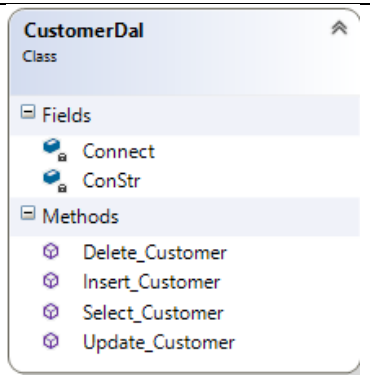
```
public  class clsInventory
{
    6 references
    public int ProductID { get; set; }
    0 references
    public int Stock { get; set; }
    5 references
    public int ProductEntry { get; set; }
}
```

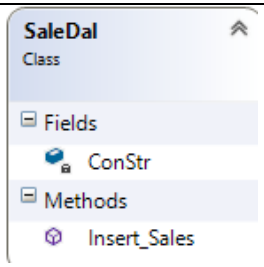
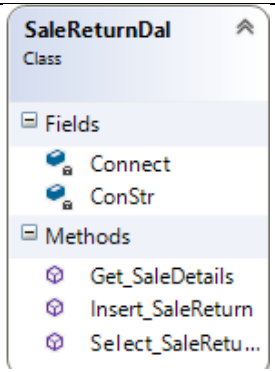
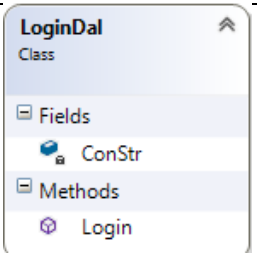
```
public class clsSale
{
    0 references
    public int SaleID { get; set; }
    2 references
    public int CustomerID { get; set; }
    2 references
    public string User { get; set; }
    2 references
    public DateTime SaleDate { get; set; }
    2 references
    public float GrandTotal { get; set; }
    1 reference
    public DataTable clsForSaleDetails { get; set; }
}
```

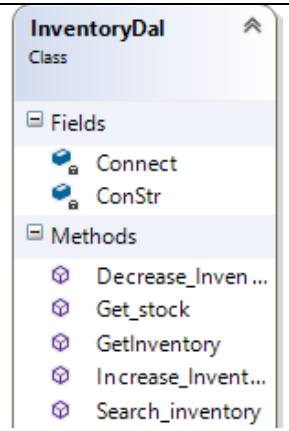
5.	Employee	clsEmployee is properties class for employee table. It contains all the properties required to perform read, update or delete operatation on employee table. To save image of employee in the employee table first image data is converted into array if byte denoted as byte[].	

```
public class clsEmployee
 {
     5 references
     public int EmployeeID { get; set; }
     5 references
     public int EmployeeType { get; set; }
     4 references
     public int DepartmentID { get; set; }
     8 references
     public string Emp_Name { get; set; }
     5 references
     public string Date_Of_Birth { get; set; }
     5 references
     public string Address { get; set; }
     5 references
     public string Gender { get; set; }
     5 references
     public string Contact { get; set; }
     5 references
     public Byte[] Picture { get; set; }
     5 references
     public string  Status { get; set; }
```

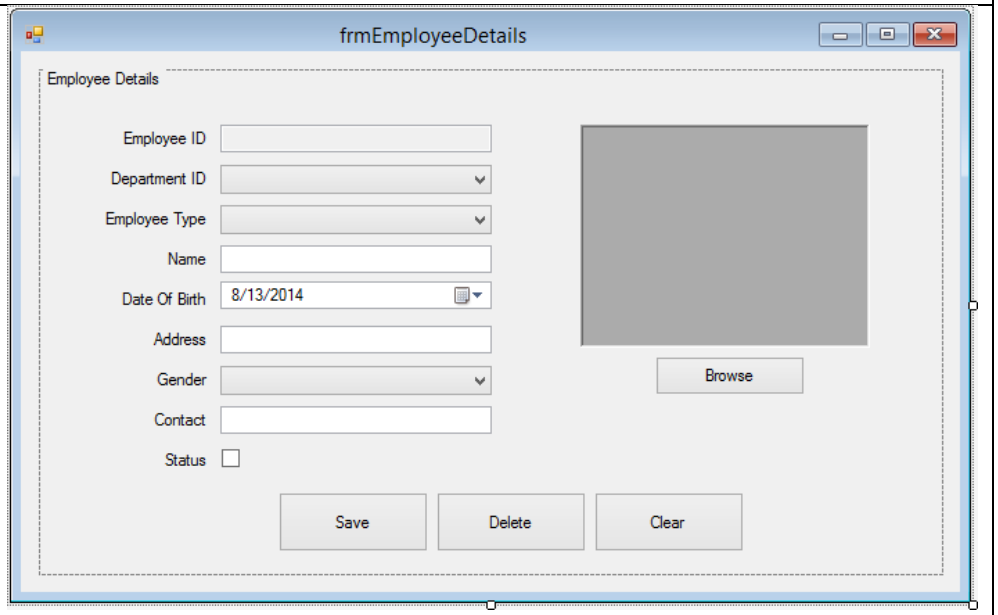
Structure for Data Access Layer (DAL): Data Access Layer contains classes which contains separate functions to perform read, insert, update, delete operation in the table. When the specified event is triggered, specified function is executed.

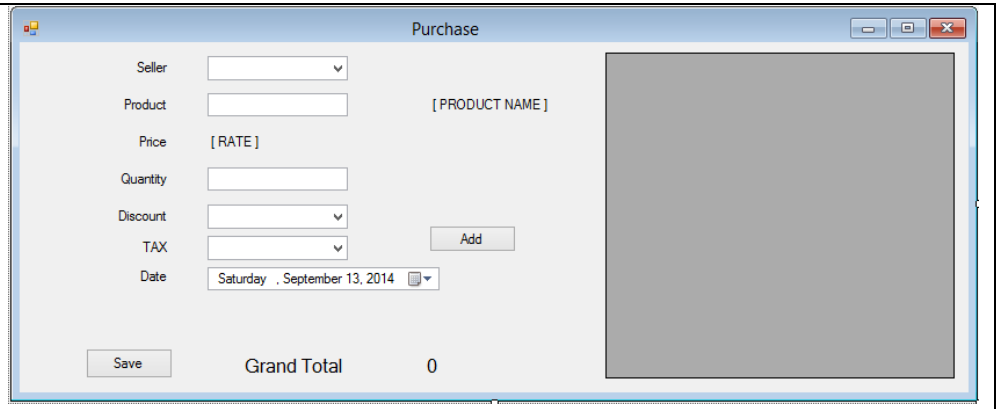
6.	Customer	CustomerDal contains several function (methods) and variables as shown in image at right. Variables connect and ConStr are used for database connection and other methods are for as name suggests delete, insert, select and update. When certain event is triggered, for instance, delete button click event, it calls delete_customer which executes codes inside to delete certain specified row from table customer.	

7.	Sales	SaleDal is used for inserting sales information into sales table. Data from UI first saved temporarly in clsSale (properties class for sale) then data is inserted into the database.	
8.	SaleReturnDal	Values for each properties of clsSaleReturn (properties class) is taken from UI and then with help of SaleReturnDal inserting, deleting, updating or selecting operation is done.	
9.	Login	When user provides username and password LoginDal perform select operation on User table, if values provided matches value in user table, it perform successful login. Else returns false value which allows UI to generate error message.	

10.	Inventory	This DAL class contains several methods required to perform inventory operations. When sale is done InventoryDal has method than can be called to decrease the product stock as per sale number. Similarly, When purchased, stock in inventory increases. It also has method to filter the result as per productID.	

Structure for User Interface (UI): User Interface is very important part of system development. It is first this client notices. For this particular solution, Forms are used. Various user controls are used to make solution simple and user friendly.

11.	Employee: frmEmployeeDetails contains various user controls such as combo box, buttons, text box, image box, data time picker etc. This allows user to provide employee details in respective fields. When user clicks save button, values in each fields first gets saved in clsEmployee then executes insert function from EmployeeDal. If textbox employeeID already has some value it save button executes update method instead save method. Similar process happens when delete button is press. Browse button allows to browse and select image file.	

12.	Purchase: frmPurchase allows user to perform purchase operation. It contains data grid view which shows product to be purchase. Grand total displays total cost of each product to be purchased.	
13.	Sale Return: frmSaleReturn form provide UI form sale return feature of the system. When saleID and ProductID is provided gridview displays sold product. Once save button is clicked after providing all field's values, it perform insert operation in SaleReturn table and increase the returned product number in the inventory stock.