

## **Table of Contents**

Table of Contents .....	1
Task 1 .....	7
Introduction .....	7
SOLID principle .....	7
Single Responsibility Principle .....	7
Open/ Close Principle .....	7
Liskov Substitution Principle .....	8
Interface Segregation Principle .....	8
Dependency Inversion Principle .....	8
Characteristics .....	8
Interoperability .....	8
Language Independence .....	9
Common Runtime Engine .....	9
Base Class Library .....	9
Simplified Deployment .....	9
Security .....	9
Portability .....	10
Features .....	10
Class Libraries .....	10
Custom Object Libraries .....	10
Managed and Unmanaged Applications .....	10
Summary .....	11
References .....	12
Task 2 .....	13
Various Languages .....	13
C# .....	13

Visual Basic (VB.Net).....	13
Critical comparison .....	14
Development Environments.....	14
Windows Web Form .....	14
ASP.Net Web Form.....	14
Web Services.....	15
NET framework versions.....	15
Conclusion .....	15
References.....	16
Critical evaluation of various components .....	17
Common Languages Runtime (CLR) .....	17
Common Types System (CTS) .....	18
Common Languages specification (CLS) .....	19
Base Class Library/Framework Class Library (FCL) .....	19
Summary .....	20
References.....	20
Task 4 (2.1, M2).....	21
Introduction.....	21
Flowchart .....	21
Context diagram.....	22
Use case Diagram .....	22
Zero Lever DFD.....	23
Organization Requirement and Program Design .....	24
Summary .....	24
Task 5.....	25
Introduction.....	25
File Structure .....	25

Properties Structure .....	25
Structure for Data Access Layer (DAL) .....	27
User Interface (UI) .....	28
Web Forms .....	28
Schema Diagram .....	30
Potential Delivery Environment .....	31
Interaction .....	31
Data Exchange.....	31
Summary .....	31
Task 6.....	33
Implementation of UI.....	33
Home .....	33
Graphic User Interface .....	33
Source.....	33
About.....	34
Source.....	34
Sale.....	34
Source.....	35
Inventory .....	35
Source.....	35
Department.....	35
Source.....	36
New Tax .....	36
Source.....	36
Customer .....	37
Source.....	37
Implementation of Event handling.....	37

---

Event.....	37
Event Handling.....	38
Button Click Event Handling .....	38
Text Changed Event Handling .....	38
Implementation of Control Structures .....	39
If statement.....	39
For statement .....	39
Switch statement .....	40
Error Handling .....	40
Try Catch and Finally.....	40
Summary .....	41
Task 7.....	42
Introduction.....	42
Unit Test.....	42
Stress Test .....	42
Integrated Test.....	42
Compatibility Test.....	42
Critical Analysis of Test results.....	43
Critical Review .....	43
Introduction .....	43
Body .....	43
Strengths.....	43
Limitations .....	44
Summary .....	44
Task 8.....	45
Feedback .....	45
Independent Feedback .....	45

Feedback Given By: .....	45
<b>Project Description</b> .....	45
Checklists .....	46
Feedback Evaluation .....	47
Evaluation of the Feedback:.....	47
Recommendations for future consideration: .....	48
Move to E-commerce site .....	48
Improve UI with DevCraft .....	48
Add password recovery feature.....	48
Add History keeping feature .....	49
Reference .....	49
Task 9.....	50
Introduction.....	50
General Helps.....	50
Navigating Through Pages.....	50
Visit about Page .....	50
Visit Contact Us .....	50
Operational Helps .....	51
Registering New User .....	51
Login .....	52
Logout .....	52
Customer Management .....	52
Add New Customer.....	52
Edit Customer.....	53
Delete Customer.....	54
Discount Management .....	54
Add New Discount.....	54

Modify Discount .....	54
Delete Discount .....	55
Sales Management .....	55
Inventory Management .....	56
Additional Help.....	57
Task 10.....	58
Introduction.....	58
Technology Used .....	58
Choice of Tools.....	59
Visual Studio .....	59
SQL Server.....	59
Design .....	60
Master Page .....	61
CSS.....	62
Task Breakdown structure .....	62
Gantt chart.....	62
Data and information Collection.....	62
Product deployment .....	63
Domain Name .....	63
Publishing the Application.....	63
Maintenance Techniques .....	64
Note.....	65

## **Task 1**

**Discuss** the principles, characteristics and features of programming using .NET Framework. [1.1, M1]

### **Introduction**

.Net framework is object oriented based software development environment developed by Microsoft. It is windows based framework where software can be developed and executed. TECHNOPEdia (n.d.) notes, .Net is Microsoft's competition to Java. Like any programming environment, .Net has its features and characteristics. This document thoroughly discusses principles, various features as well as characteristics of .Net framework.

### **SOLID principle**

.Net Framework follows various principles of object oriented design and programming. There are five principles for class design and they are together famously known as SOLID. According to Carr (2010) acronym **SOLID** was first introduced by Robert Cecil Martin. While programming a software, code can become fragile and difficult for reuse. These rigid and fragile code are very difficult to maintain and modify when required. Programming done following SOLID principle produces code that are flexible and easier for reuse. All five principles of SOLID are separately discussed here.

#### **Single Responsibility Principle**

This principle states that a class should change for one reason only, Nagel (2010). In other words one class should have only one reason to change. If there are more than one reasons (let's say two) to change a class then functionality of that class should be split into two classes. Hence each class now will have only one reason to change. In simple worlds, each class should have only one purpose. Single Responsibility Principle offers various benefits such as less complex code, code reusability, better code maintenance and testing.

#### **Open/ Close Principle**

Basically, open/closed principle says entities such as modules, class and functions of a system should be designed in such way that adding new function can be achieved without modifying existing code. Instead of that, new functionality can be achieved via writing new codes. These new codes will be used by existing code (Csaba, 2014). The core theme of this principle is to allow programming code in such way that does not require to change every time requirement changes.

### Liskov Substitution Principle

Sanaulla (2011) notes that according to a principle written by Barbara Liskov in 1988, Liskov Substitution Principle states “*Methods that use references to the base classes must be able to use the objects of the derived classes without knowing it*”. In simpler words, this principle says if A is subset of B then object of B can be replace by object of A without making any negative impact into the system and causing errors. A subtype should be able to substitute base type.

### Interface Segregation Principle

According to Nemes (2013) ISP states users should not be forced to be contingent on interfaces that they do not use. In more general words, ISP says users should not be obligated to use methods that are not useful to them. For example, let’s say an interface is written to get user information about name, address, email address and class and batch. Now if same interface is used for getting teacher information use of batch attribute makes no sense.

User does not have to provide information about teacher’s batch as it makes no sense but application forces user to provide teacher’s batch information. To avoid any complication, codes can be written to set batch as zero for all teachers. But this still violates ISP. To avoid this, user should not be forced to use teacher class created using interface targeted for student. Teacher class should be concreted using sensible interface.

### Dependency Inversion Principle

Gervasio (2012) writes according to Dependency Inversion Principle (DIP) states that *high level module should not depend on low level module rather they both should depend on abstraction*. Other statement of the principle says “*abstraction should not depend on details but details should be based on abstraction*.” This principle basically states that concrete classes should be created based on interfaces. Creating classes directly can create complicated situation.

### Characteristics

Programming using .Net Framework exhibits various features and characteristics. One of the key benefits is it allows programmers to use same coding skillset to develop application of different kind and environment. Below here some of the key characteristics are discussed.

### Interoperability

Microsoft (n.d.) states, interoperability enables programmers to take advantage of both managed and unmanaged code. Interoperability is characteristic of .Net framework programming which allows different .Net language understand each other and work in same project together. Though



code from C# (.cs) and Visual Basic (.vb) cannot work together directly. Interoperability is achieved via through dynamic link library (DLL). This allows a team of programmers with skills of different .Net programming language work together.

### Language Independence

One of the key feature of the .Net framework is language independency. Which means programmers has choice of using any .net language such as C#, VB.net and C++ to develop program. Common language specification (CLS) one of the core component of .Net framework sets basic features for the .net languages. In order to support .Net environment, a language needs to follow CLS. One of the other key component of .Net framework which enables language independency is Common type system (CTS). This basically set rules for languages which must be followed by languages in order the become .net language.

### Common Runtime Engine

CLR (Common language runtime) is component of .NET framework that supervise execution of all .NET programs. It provides runtime environment for the programs based on .NET framework. With CLR all .NET program is passed through similarly memory management, security mechanism and exceptional handling.

### Base Class Library

Base class library is collection of managed types that provides fundamental functionality of the .NET framework. As any .NET framework based programming languages must support BCL, programmer has language independency and can choose language of their own preference. Types are logically grouped into Namespaces.

### Simplified Deployment

Before the development of .NET framework, installing a new application was tricky task as it could replace or remove component required for previously installed applications. This could cause older application stop working properly. .NET framework ensures that component required by previously installed applications are not interfered.

### Security

According to Sabbadin (2006.) .NET framework features simple yet flexible security mechanism that is identity-based. This offers control over security of application and its components. .NET framework security mechanism offers some level of security over vulnerabilities of system that could be exploit by malicious programs.

### Portability

Programming with .NET framework development of a cross-platform application is very much possible. One of the big example of this is Microsoft's implementation of framework in windows, mobile windows and Xbox platforms.

### Features

#### Class Libraries

Microsoft (n.d.) writes, class library is included in .NET framework and is library of various classes, value type and interfaces. Programmer can use class library to get access to functionalities of system. Programmer specifies the required class to be used and instantiate object of that required class. This makes life easier for programmer as they do not need to recode whole class and just use already written code.

#### Custom Object Libraries

.Net framework allows creating custom object libraries. Unlike base class libraries, custom object library consists of custom value types, custom interfaces and custom classes. Such custom class first compiled as DLL file which later can be used by referencing. .NET framework provide programmers features to use custom built library to achieve requirement of the system development.

#### Managed and Unmanaged Applications

Codes developed within .NET framework is known as managed code. Such codes are written using any .NET framework programming language such as C# or VB.NET or C++. Managed code is converted into intermediate language. Then executed to system. CLR offers memory management, security mechanism etc. to the managed code.

Unmanaged code is code that is written outside of .NET framework. Programming languages such as C++ can be used for writing this kind of program. A program to access low level of operating system is good example of such program. Mehra (2009) writes, unmanaged code is executed with help of wrapper classes. Codes wrapped with COM callable wrapper (CCW) is converted to Microsoft intermediate language whereas codes wrapped with Runtime callable wrapper (RCW) are executed directly.

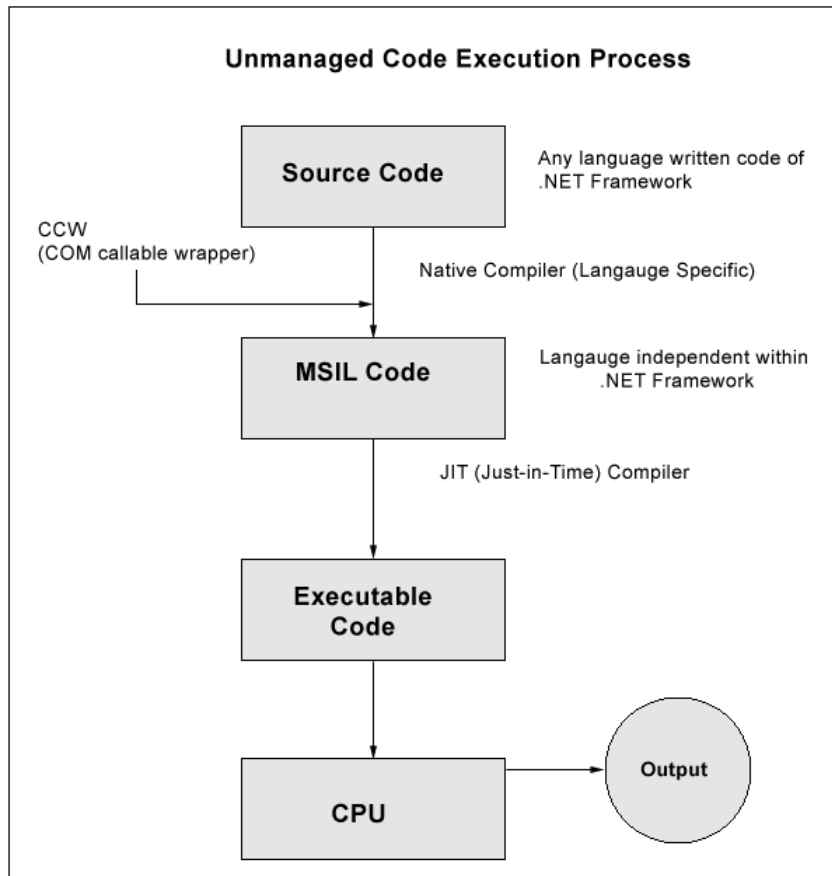


Figure 1 Execution of Unmanaged code.

### **Summary**

This paper discusses principle of programming using .NET framework. Various characteristics and features are thoroughly discussed in this document. Discussion shows that .Net framework allows programmers develop system using different choices of languages. As .Net is Language Independent programmer can choose language of their choice to develop system. Here language must follow Common language specification (CLS) to support .Net. This report also discusses that fact that .NET framework provide supports for both managed and unmanaged system. Additionally, class libraries, security, portability, custom object, deployment of system while programming in .NET framework is also discussed.

## **References**

- Carr (2010) the SOLID Principles [online] Available: <http://www.blackwasp.co.uk/SOLID.aspx> Accessed [4/2/2015]
- Csaba, P. (2014) SOLID: Part 2 - The Open/Closed Principle [online] Available: <http://code.tutsplus.com/tutorials/solid-part-2-the-open-closed-principle--net-36600> Accessed [4/2/2015]
- Gervasio, A. (2012) the Dependency Inversion Principle [online] Available: <http://www.sitepoint.com/dip-intro/> Accessed [4/2/2015]
- Managed code and unmanaged code in .NET
- Mehra, P. (2009) Managed code and unmanaged code in .NET [online] Available: <http://www.c-sharpcorner.com/uploadfile/puranindia/managed-code-and-unmanaged-code-in-net/> Accessed [4/2/2015]
- Microsoft (n.d.) Class Library [online] Available: <https://msdn.microsoft.com/en-us/library/d11h6832%28v=vs.71%29.aspx> Accessed [4/2/2015]
- Microsoft (n.d.) Interoperability (C# Programming Guide) [online] Available: <https://msdn.microsoft.com/en-us/library/ms173184.aspx> Accessed [4/3/2015]
- Nagel, C. (2010) Professional C# 4 and .Net 4. Indianapolis, IN: Wiley Pub.
- Nemes, A. (2013) SOLID design principles in .NET: the Interface Segregation Principle [online] Available: <http://dotnetcodr.com/2013/08/22/solid-design-principles-in-net-the-interface-segregation-principle/> Accessed [4/3/2015]
- Sabbadin, E. (n.d.) .NET Framework features security mechanism [online] Available: <http://searchsoftwarequality.techtarget.com/tip/.NET-Framework-features-security-mechanism> Accessed [4/3/2015]
- Sanaulla, M. (2011) SOLID – Liskov Substitution Principle [online] Available: <http://www.javacodegeeks.com/2011/11/solid-liskov-substitution-principle.html> Accessed [4/3/2015]
- TECHNOPEdia (n.d.) .NET Framework (.NET) [online] Available: <http://www.techopedia.com/definition/3734/net-framework-net> Accessed [4/3/2015]

## **Task 2**

**Critically compare** different types of .net framework architectures. [1.2]

**Consider:** Various types of languages, Web forms/Windows forms/ Window mobile application, .net frame work versions under different version of windows.

### **Various Languages**

#### **C#**

C# is simple, type safe and object oriented programming language. This language drives from C and C++ and borrows some key concepts from programming languages like Java. It is great software development tool that abandoned features of C++ like multiple inheritance and brought new feature like garbage collection which provides automatic memory optimization. According to TUTORIALSPPOINT (n.d.) Anders Hejlsberg and his team developed C# during development of .NET framework. This language is extensively used for game, mobile, desktop and web application development. Example of C# coding structure is shown below.

```
using System;

namespace Hello {
    public class HelloWorld {
        public static void Main(string[] args) {
            string name = "C#";

            // See if an argument was passed from the command line
            if (args.Length == 1)
                name = args[0];

            Console.WriteLine("Hello, " + name + "!");
        }
    }
}
```

#### **Visual Basic (VB.Net)**

Visual Basic .Net programming language is based on Basic and Visual basic programming language and developed by Microsoft. It is modern and object oriented language that has power of .NET framework. Although this language is easy to learn due to easy syntax, it cannot be underestimated as it can program powerful application. According to Yagulasamy (2005) visual basic IDE compiles vb.net code in background which helps small programs but as program grows this feature slows down IDE and can become frustrating. Example of VB.Net coding structure is shown below.

```
Imports System

Namespace Hello
Class HelloWorld
    Overloads Shared Sub Main(ByVal args() As String)
        Dim name As String = "VB.NET"

        'See if an argument was passed from the command line
        If args.Length = 1 Then name = args(0)

        Console.WriteLine("Hello, " & name & "!")
    End Sub
End Class
End Namespace
```

### Critical comparison

Even though both C#.NET and VB.NET language share some mutual features, both has some unique features too. One of the clear difference between two languages is their syntax. C# follows C-Style syntax while VB.Net follows Basic-Style syntax. Hence for Visual Basic 6.0 or Basic developer it is easier to switch to VB.Net similarly, for C and C++ programmers it's easier to switch to C#.

C# has powerful features that are not available in VB.Net like handling unmanaged code. Both language uses common language runtime (CLR).

### Development Environments

#### Windows Web Form

Windows Forms is the structure generator for applications and is similar to the structures generator of Visual Basic 6. Visual Basic developers will be acquainted with the look and feel of Windows Forms, yet this similitude is to a great extent corrective and there are considerable contrasts in the impel-mentation. Windows Forms is new to Visual C++ developers, who previously needed to code each part of a program's graphical client interface.

#### ASP.Net Web Form

ASP.NET Web Forms is a piece of the ASP.NET web application structure and is incorporated with Visual Studio. It is one of the four programming models we can use to make ASP.NET web applications, the others are ASP.NET MVC, ASP.NET Web Pages, and ASP.NET Single Page Applications.

Web Forms are pages that our client request utilizing their program. These pages can be composed utilizing a mix of HTML, customer script, server controls, and server code. At the point when

clients ask for a page, it is ordered and executed on the server by the structure, and after that the system produces the HTML markup that the program can render. An ASP.NET Web Forms page presents data to the client in any program or customer gadget.

### Web Services

A Web Service is a product program that uses XML to trade data with other programming by means of basic web conventions. In a basic sense, Web Services are a path for cooperating with articles over the Internet. It is a web application which is fundamentally a class comprising of systems that could be utilized by different applications. It additionally takes after a code-behind construction modeling like the ASP.Net pages, despite the fact that it doesn't have a client interface.

### NET framework versions

Over the years, Microsoft has published several version of .NET framework. To most recent version of .Net are discussed and compare below here.

.NET framework 3.5 and 4.0 are two adaptations of the Microsoft .NET structure. Microsoft dependably concocts different applications and systems so that the application improvement is more exceptional and upgraded. Microsoft .NET Framework is a structure which is intended for Windows working framework. It has a vast library and backings different programming dialects. It likewise bolsters interoperability and NET library is accessible to all programming dialects which are upheld by .NET. In year 2007, .NET 3.5 was discharged which had included more components that .NET 2.0 and .NET 3.0 couldn't stay in the business for long time as it has different issues. In any case, .NET 4.0 was discharged in April 2010.

Some of the key difference between these two frameworks are as follows:

- Web installer in .NET 4 are less than 1 MB
- .NET 4 allows direct method for accessing data whereas there is no direct method in .NET 3.5
- .NET 4 is improved version of .NET 3.5

### Conclusion

This paper critically discussed and compare different types of .Net framework architecture. Different development environment in .NET framework, windows from, web from, web services were compare. Additionally, two different .net language was compared. And finally various .Net framework versions were critically compared.

**References**

- TUTORIALSPPOINT (n.d.) *C# - Overview* [Online] Available:  
[http://www.tutorialspoint.com/csharp/csharp\\_overview.htm](http://www.tutorialspoint.com/csharp/csharp_overview.htm)
- Yagulasamy, A. (2005) *Complete Comparison for VB.NET and C#* [Online] Available:  
<Http://www.codeproject.com/Articles/9978/Complete-Comparison-for-VB-NET-and-C>  
Accessed [4/5/2015]



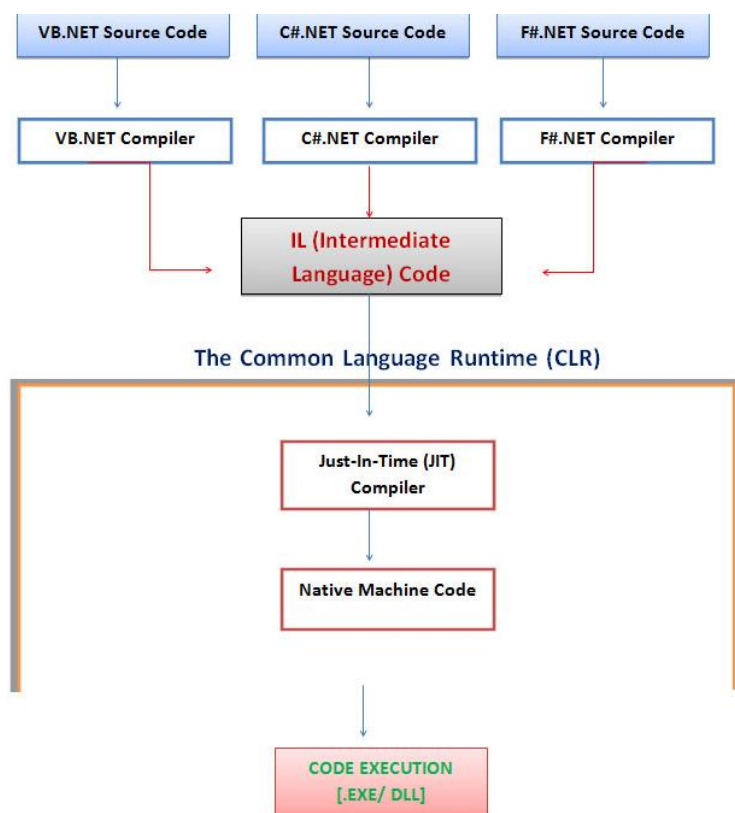
### Task 3

**Critically evaluate** the components that support the .NET framework. [1.3]

#### Critical evaluation of various components

##### Common Languages Runtime (CLR)

Common Languages Runtime (CLR) is runtime provided by .NET framework. Basically this provides .net programs environment to run. As the name suggests CLR provides common runtime to all programming languages supported by .net framework. For example, in past VB required VBRunxxx.dll but with VB.net individual runtime is not required.



**Figure 2 Representation of HOW .net and CLR works**

<http://resources.infosecinstitute.com/net-framework-clr-common-language-runtime/>

Different programming languages can be used to develop a program, regardless of languages used in source code CLR converts them to intermediate language known as CIL (common intermediate language), Janssen (n.d.). As shown above figure 1, program is first converted into MSIL (Microsoft intermediate language) then JIT (Just in time) compiler transforms them to bytecode. This bytecode is later converted into native code which is understood by machine.

According to JVP (n.d.) CLR provides memory management and thread management hence, programmer does not required to be worry on that. CLR is a great component supported by .net framework which allows programmer to use any supported language they know to solve the problem. This makes .net framework flexible as programmer can choose any language they prefer to develop the system. Here are some of the key benefits of CLR.

### Benefits of CLR

1. Ensures language Interoperability
2. Provides Automatic memory management
3. It supports both famous programing paradigm i.e.: procedural, object oriented
4. Supports custom attributes

### Common Types System (CTS)

Common Types System also referred as CTS is standard followed .net framework while declaring and using types. This component enables cross languages integration. According to Chauhan (2013), CTS defines rule that all .net compatible languages must follow while writing objects to ensure it can interact with each other regardless of what language is used. For example, Both VB and C# uses int32 from CTS though they have integer Data Type and int Data type respectively.

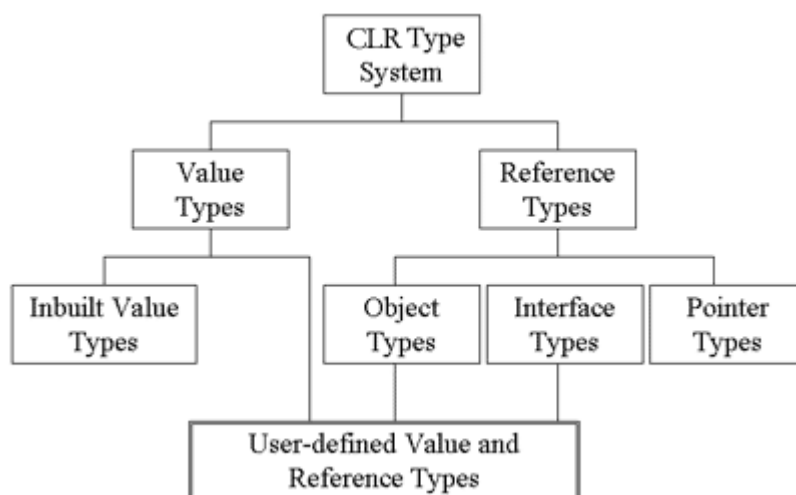


Figure 3 Common Types System Source: <https://i-msdn.sec.s-msft.com/dynimg/IC173135.gif>

As shown in figure 2, Common Types System supports two categories of types:

### Value Type

This type contains data and are allocated in stack. Value type can be built in, user defined (defined in source code) or enumeration where value is stored and numeric but represented label.

### References Type

This type contains reference to the values memory location. It can be of interface type, pointer type and self-describing type.

### Common Languages specification (CLS)

Common Languages specification is a set of basic language features defined. It is rules that defines a subset of the Common Type System, MSDN (n.d.). This means rules that applies to CTS are also applied to CLS. But in addition, CLS contains strict set of rules that every .net supported languages must follow. It helps in cross language debugging and cross language inheritance.

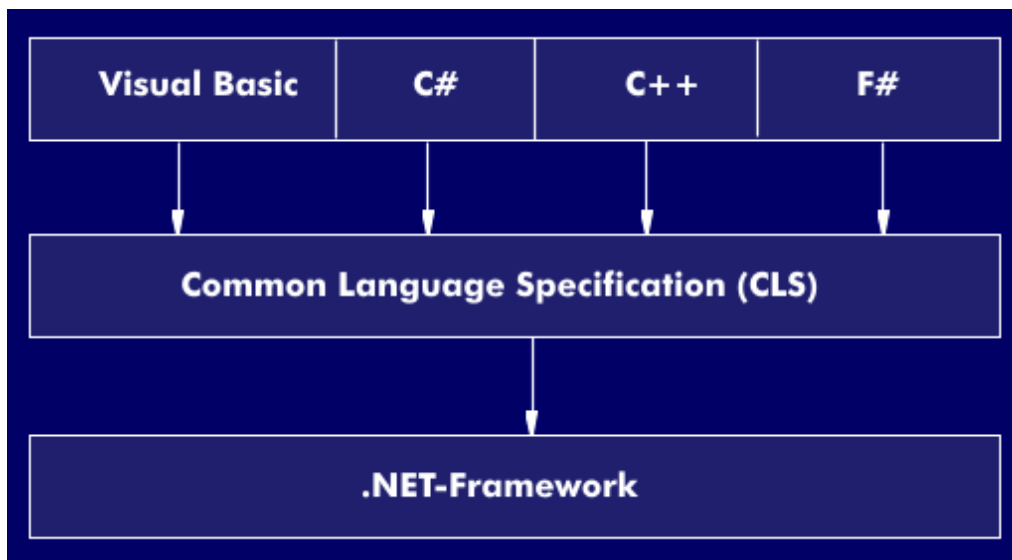


Figure 4CLS

Source: <http://3.bp.blogspot.com/-eSeqmpNMsCU/Tw1RmwV1IwI/AAAAAAAAAHs/6okb-CakKVk/s1600/common-language-specification-spec+india.png>

CLS is basically a guideline a language should follow so that it can understand .net framework and work with other .net languages. Figure 3 show how different languages go through CLS standard. It is key component of .NET Framework to provide language independency.

### Base Class Library/Framework Class Library (FCL)

Base Class Library is collection of reusable types including interfaces, classes, and data types included in the .NET Framework. It provides key functionality of .Net Framework. It is standard library used in source code by all .Net languages. BCL follows object oriented paradigm where it

encapsulates large number of features such as read, write, and database connection etc. making things easier for programmer.

Function and features provided by BCL can be used by any .net language. Framework Class Library (FCL) is superset of BCL. It provides other larger features of .NET such as ADO.NET, WINFORM, and ASP.NET etc.

### **Summary**

This document critically evaluated various components that supports .NET Frameworks. Evaluation of Common Language Runtime (CLR) showed that it is essential component that provides runtime environment to the program written using .net language. It also help to enhance performance through memory management. Similarly, evaluation of Common Type System (CTS) showed it is another essential component that ensures all .net program can communicate with each other regardless of selection of language. Meanwhile, CLS ensures languages follow some basic guideline so that it can take advantage of .NET Framework.

Framework Class Library and Base Class Libraries are predefined classes and types that programmer can import and use to solve the problem. Summary of this document is .NET Framework is great technology that has many components that enables user to choose language of their choice and still utilize great features of .NET Framework.

### **References**

- Chauhan (2013) Understanding Relationship between CTS and CLS <http://www.dotnet-tricks.com/Tutorial/netframework/J7N9181013-Understanding-Relationship-Between-CTS-and-CLS.html> Accessed [4/6/2015]
- Janssen (n.d.) Common Language Runtime (CLR) [Online] Available: <http://www.techopedia.com/definition/5225/common-language-runtime-clr> Accessed [4/6/2015]
- JVP (n.d.) Common Language Runtime (CLR) [Online] Available: <http://whatis.techtarget.com/definition/Common-Language-Runtime-CLR> Accessed [4/6/2015]
- MSDN (n.d.) Common Language Specification [Online] Available: <https://msdn.microsoft.com/en-us/library/vstudio/12a7a7h3%28v=vs.100%29.aspx> Accessed [4/6/2015]

**Task 4 (2.1, M2)**

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






**Introduction**

Before implementing the proposed system for Bhat-Bhateni it is essential to design event driven programming solution analyzing the given scenario. To design the plans, various tools and techniques can be utilized. This document utilizes such tools to develop system design for the system.

**Flowchart**

A flowchart is a formalized graphic representation of a rationale arrangement, work or assembling procedure, association outline, or comparable formalized structure. The reason for drawing a flowchart diagram is to provide people reference point for project management.

Flowcharts use straightforward geometric images and bolts to characterize connections. In this chart, for instance, the starting or end of a system is described to by an oval. A procedure is spoken to by a rectangle, a choice is described to by a diamond and an I/O procedure is spoken to by a parallelogram. The Internet is described by a cloud.

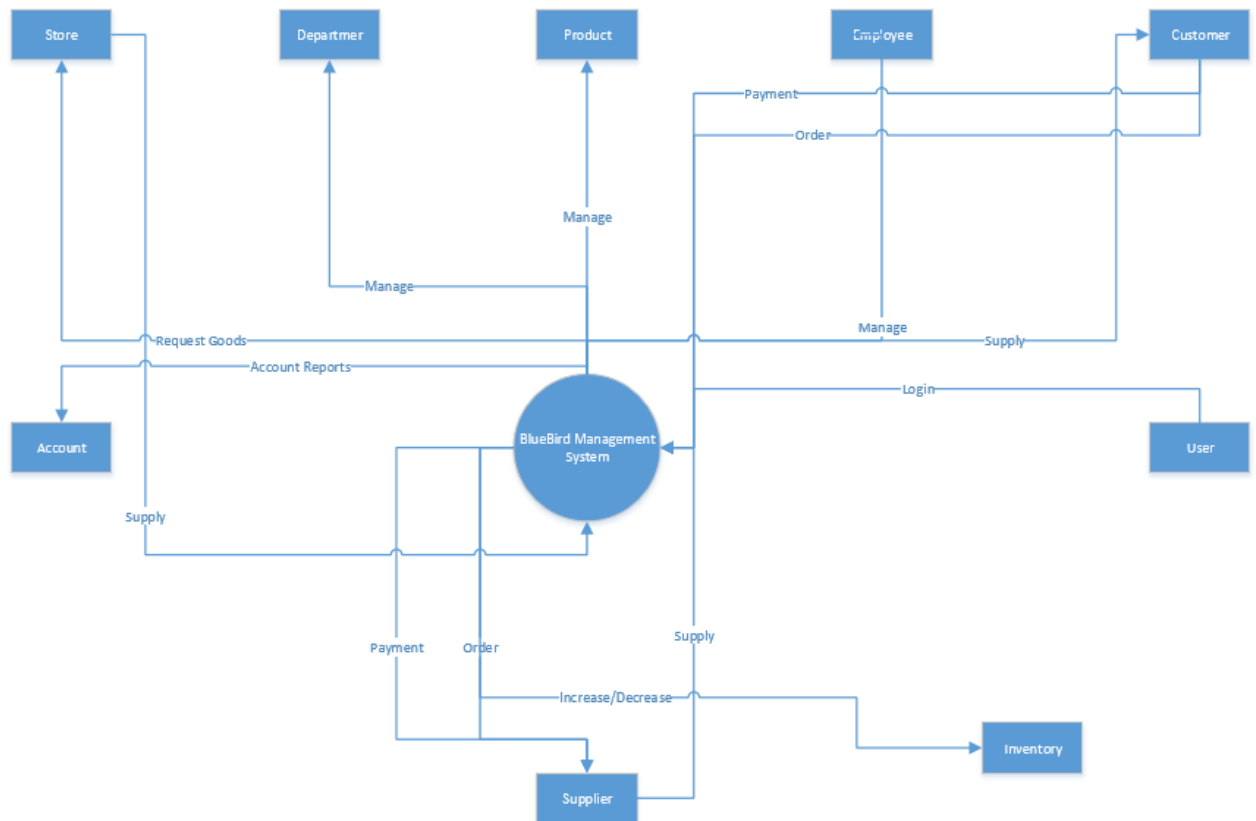
Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

**Figure 5**Symbols used in flowchart

Flowchart design for proposed system for Bhat-Bhateni super market will enable better communication as flowchart is very simple and understandable by both technical and non-technical people. Flowchart will help project planning as it represents flow of the project. It provides an easy way of communication because any other person besides the programmer can understand the way they are represented. Additionally, diagram will help programmer to understand overall view of the product to be developed and understand how different modules are related with each other. Flowchart diagram for proposed system is draw below in separate page.

### **Context diagram**

Context diagram is fundamental structure of project. This graphical representation shows connection of elements of project with main process. A context diagram very high level overview of the system. It describes system's core process and external entities project. Context diagram for Bhat-Bhateni software will allow analyst to understand the overview and develop lower level Data flow Diagram. Context diagram for proposed software application is drawn table below.

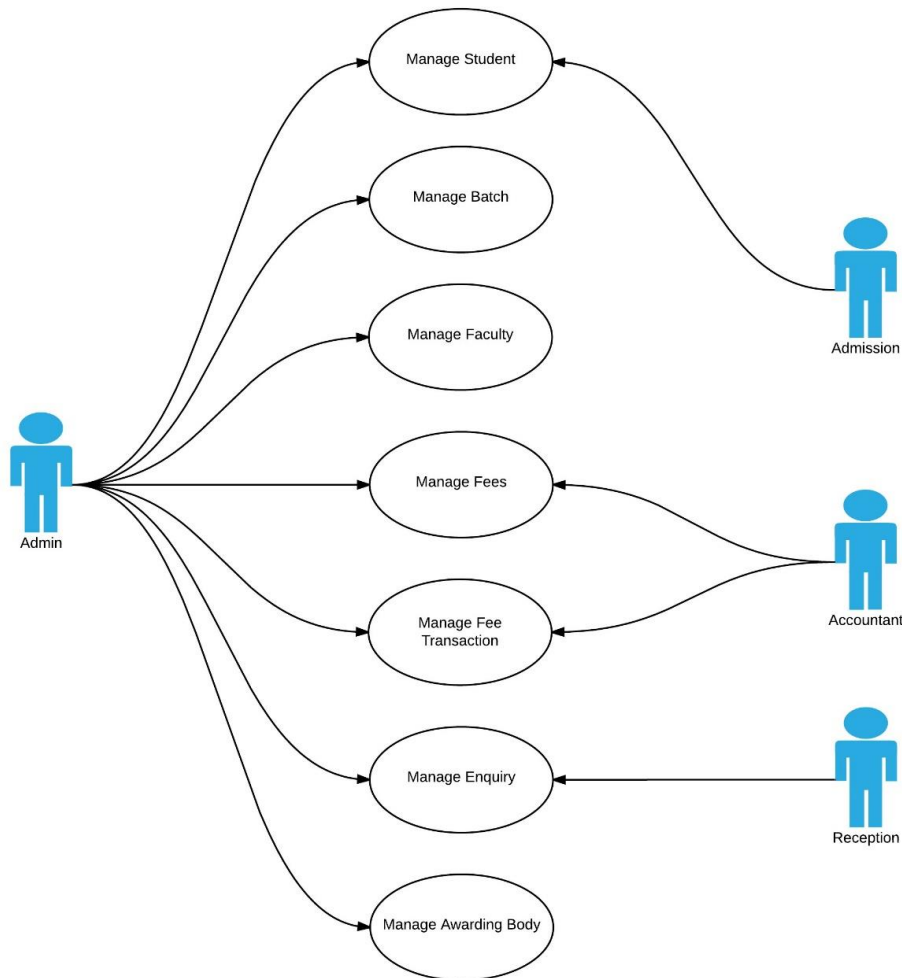


**Figure 6 Context diagram for Bhat-Bhateni supermarket software**

In given context diagram for Bhat-Bhateni, external entities such like store, department, product, employee, customer etc. are linked with main process. This will help analyze various aspect for the software.

### **Use case Diagram**

A use case is a procedure utilized as a part of framework investigation to distinguish, clear up, and sort out framework prerequisites. In this connection, the expression "framework" alludes to something being created or worked, for example, a mail-request item deals and administration Web website. Use case diagram for propose system is designed below.



**Figure 7 Use case diagram for proposed system**

In given use case diagram above, boundaries of the proposed system graphically described. It also describes possible actors and their roles in the system. Administrator will have access to every module of system where as receptionist will use system to manage enquiry, accountant will use system for fee management and admission department staff will use system to manage students. This diagram describes relationship between use cases and potential actors.

### **Zero Level DFD**

Zero level diagram is more detailed form for context diagram. It is first level diagram that demonstrates the data flow inside a software. This enables developer to structure his/her development approach. 0 level diagram for proposed system is prepare below.

Prepared 0 level DFD for the system describes various top level processes and their relationship with external entities. It demonstrates how different processes interact with each other. Prepared diagram enables programmer understand dataflow in the system.

### **Organization Requirement and Program Design**

Table underneath legitimizes that choice of outline fulfill the necessity of the framework.

SN.	Organizational Requirement	Feature in software to satisfy the requirements
1.	Keep up Information about Employee, clients, items	System has dedicated feature page to allow management of clients, employees, products.
2.	Control sales and purchase	Designed system has sales and purchase management feature.
3.	Manage Information about tax/discounts	Tax and discount information can be modified and new discount policies can be created
4.	Manage inventory and Transaction Log	System has inventory checking feature
5.	Have Admin/User Login Feature	System has login feature where user will have access to authorized features only.

### **Summary**

This paper prepared design for the proposed system using several tools and techniques. To understand top level overview of the system, context diagram tool is utilized. Furthermore, to analyzed and study flow of application, flow chart tool is used. Designed data flow diagram analyzes flow of data in the system. And finally use case diagram was prepared to understand various actors and their relation with use cases. This paper also justified how proposed system is appropriate for solving given problem.



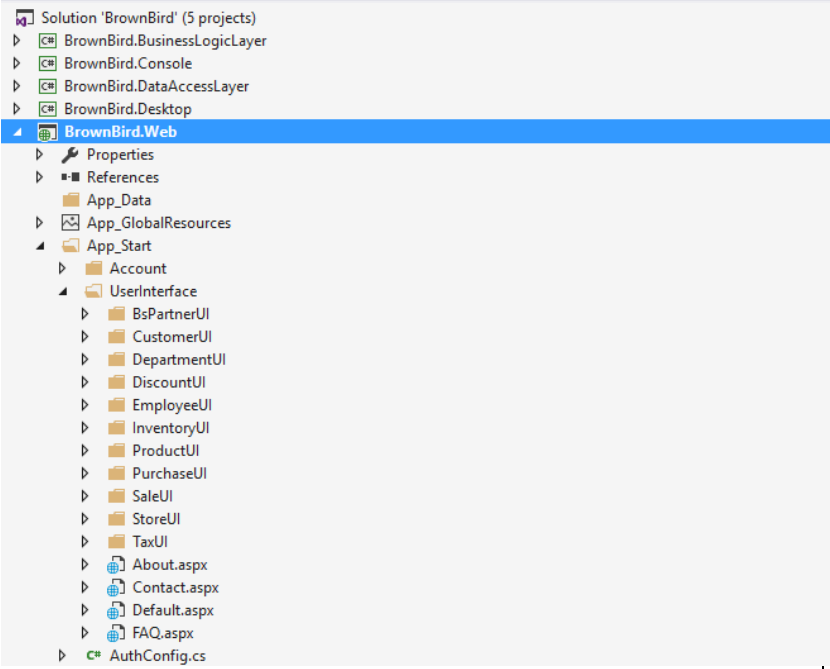
## **Task 5**

**Explain** the components and data and file structures required to implement a given design and **evaluate** potential delivery environment and interaction. [2.2, 2.3 D1]

### **Introduction**

To implement given design various components, tables, UI components, data structures are need to be identified. These components includes, Web forms, class, properties and database tables etc. This document explains required components to implement solution for the given problem. Additionally, this document also evaluates potential deployment platform and interaction for the solution.

### **File Structure**

S.N	Name	Screen shots
1.	<b>Solution File Structure</b>	 <p>Image above shows required file structure for the solution. UI, Data layer and business logic are separated. Files are structured in such way to make solution management efficient ,durable and maintainable.</p>

### **Properties Structure**

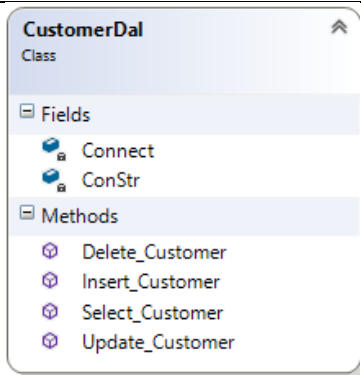
Properties are special type of class members which has get and set method to access and modify them. Read and write is done through get and set method. For Bhat Bhateni Department store number of properties are required. At least one each for equivalent data table.

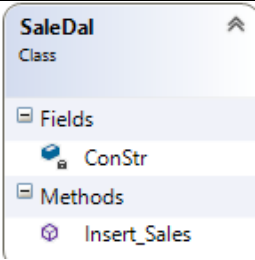
2.	Customer	<pre> 10 references public class ClsCustomer {     //prop tab tab     6 references     public int CustomerID { get; set; }     5 references     public string CustomerName { get; set; }     5 references     public string Address { get; set; }     5 references     public int CustomerTypeID { get; set; }     5 references     public string Contact { get; set; } } </pre>
<p>This class contains properties required to calculate, read and write data from customer table. Properties in class will be used for inserting, updating or deleting customer data. To save new customer, value for each attributes of table will be taken from web forms and temporarily stored in customer property. Later it will be stored to database via Data Access Layer.</p>		
3.	Inventory	<pre> public class clsInventory {     6 references     public int ProductID { get; set; }     0 references     public int Stock { get; set; }     5 references     public int ProductEntry { get; set; } } </pre>
<p>Like customer class, clsInventory contains properties required to delete, update and insert into Inventory table. Data to be inserted or data taken from inventory table can be saved temporarily in clsInventory. This makes development process much easier and secure.</p>		
4.	Sale	<pre> 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; } } </pre>
<p>Use of clsSale is similar to other properties classes. It contains, saleID, CustomerID, User, SaleDate, Grand Total and datatable for clsSaleDetails.</p>		

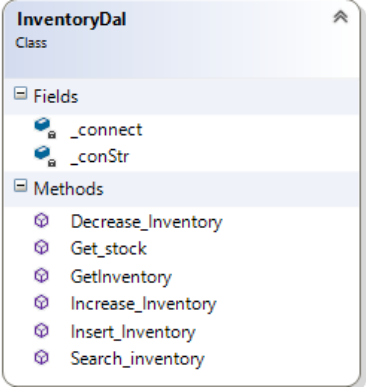
5.	Employee	<pre> public class clsEmployee {     5 references     public int EmployeeID { get; set; }     5 references     public int EmployeeType { get; set; }     4 references     public int DepartmentID { get; set; }     8 references     public string Emp_Name { get; set; }     5 references     public 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; } </pre>
<p>clsEmployee is properties class for employee table. It contains all the properties required to perform read, update or delete operation on employee table. To save image of employee in the employee table first image data is converted into array of byte denoted as byte[].</p>		

#### Structure for Data Access Layer (DAL)

To perform data related actions, solution has separate layer called Data Access Layer. To perform read, insert, update, delete operation in the table DAL has separate method in each of the classes. Each class target different object of the database. When an event is triggered, for example customer save button is clicked, DAL stores information in database using customer properties class and customer save method.

6.	Customer	
<p>CustomerDal contains several function (methods) and variables as shown in image above. 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 specified row from table customer.</p>		

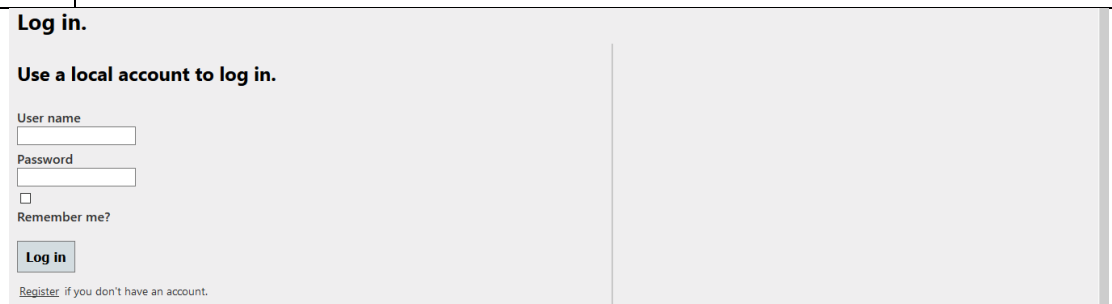
7.	Sales	
<p>SaleDal is used for inserting sales information into sales table. Information from Webform is first saved temporarily in clsSale (properties class for sale) then data is inserted into the sale table in database.</p>		

10.	Inventory	
<p>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.</p>		

### User Interface (UI)

#### Web Forms

For the given solution, GUI is designed using web forms. Various web pages required for solution is listed here below.

11	<p><b>Login:</b></p> <p>Login page has UI controls to take user credential information. Username and password textboxes are place above submit button.</p>
	

## 11. Employee:

Employee webpage contains table that displays employees working in the department store. It has Create New Employee link that opens webpage to insert new employee information.

**BhatBhateni Super Market** [Register](#) [Log in](#)

Home About Transaction Store Management Tax Stakeholders Product Contact

**Employee Management**

[Create New Employee](#)

Update	Delete	EmployeeID	Type	DepartmentID	Emp_Name	Date_Of_Birth	Address	Gender	Contact	Status
<a href="#">Update</a>	<a href="#">Delete</a>	1	Full Time	4	Ashi	1/24/1990 12:00:00 AM	Balkumari	Male	983123123123	Active

## 12. Purchase:

Purchase webpage has various UI controls such as combo box, textbox, calendar, grid view etc. This page would allow user to perform purchase operation.

**Purchases Management**

Customer:

Product:

Price:

Quantity:

Discount Mode:

Tax Mode:

Date:

[Add to Cart](#)

ProductID	ProductName	Price	Quantity	Tax(%)	Discount(%)	Total
2	Football	500	500	13	10	254250

## 13. Add Department

Department web page has various UI controls such as textbox, dropdown etc. and will allow user to add new department.

**BhatBhateni Super Market** [Register](#) [Lo](#)

Home About Transaction Store Management Tax Stakeholders Product Contact

**Department Management**

Store:

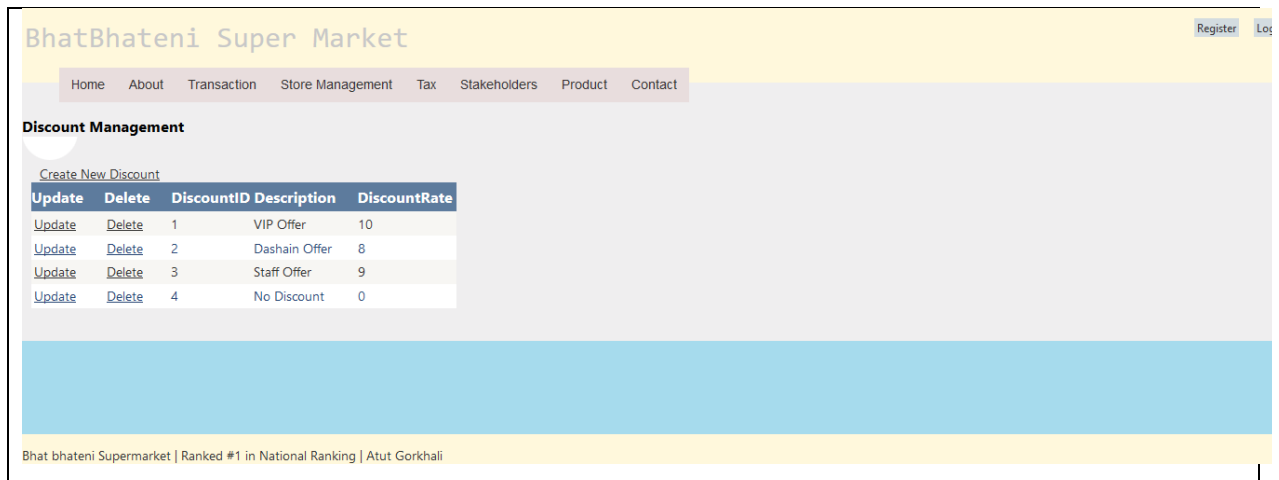
Department Name:

Description:

[Save](#)

## 13. Discount

Discount page would display discount information in database. Page would also consist links to pages related to created, modifying and deleting discount information.



### Schema Diagram

Database requires for storing various information such as users, products, sales, purchase etc. Schema diagram is drawn below to describe relationship between entities of database. It is visual representation of entities and attributes of database. A schema is the structure behind information association. It is a visual representation of how table connections empower the schema's fundamental mission business rules for which the database is made.

Schema diagram for proposed system is developed in next page which allows database programmer to understand database requirement of the system. Conversion from schema to database is not complicated task. Hence, schema helps to provide database solution for the proposed system. Schema diagram of the proposed system consists of tables like users, roles, sales, purchase, product, customer, employee etc. and their attributes are described as well. Schema diagram also describes relation between various tables.

## **Potential Delivery Environment**

Web applications really envelop every one of the applications that speak with the client through HTTP. This incorporates light applications like Flash recreations, online notes, schedules, etc., and additionally more serious applications that have utilization HTTP. Developed system will be web based system which means user will access the application through HTTP. Browser will require to run this application. User will be provided with one common domain name to access the application. Developed system is developed in such manner using OOP that with few change management, it can be delivered to other environment as well.

If project sponsor, i.e.: Bhat-Bhateni department store wants system to be delivered as desktop application, existing codes can be reused to provide delivery for desktop. Actually this is the beauty of C# programming language and OOP. It allows programmer to develop multi-platform system.

## **Interaction**

### **Data Exchange**

The prominence of XML for information trade on the World Wide Web has a few reasons. Above all else, it is firmly identified with the prior benchmarks Standard Generalized Markup Language (SGML) and Hypertext Markup Language (HTML), and all things considered a parser kept in touch with backing these two dialects can be effortlessly stretched out to bolster XML too. For instance, XHTML has been characterized as an organization that is formal XML, yet saw effectively by most (if not all) HTML parsers. This prompted brisk appropriation of XML backing in web programs and the tool chains utilized for producing pages. In developed system, XML is used for storing database location information.

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkId=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=neutral, PublicKeyT
  </configSections>
  <appSettings>
    <add key="ChartImageHandler" value="storage=file;timeout=20;dir=c:\TempImageFiles\" />
  </appSettings>
  <connectionStrings>
    <add name="ConnStr" connectionString="Data Source=(local);Initial Catalog=BrownBird;Integrated Security=SSPI;" providerName="System.Data.SqlClient"/>
    <add name="BrownBirdConnectionString" connectionString="Data Source=(local);Initial Catalog=BrownBird;Integrated Security=True" providerName="System.Data.SqlClie
  </connectionStrings>
  <system.web>
    <httpHandlers>
      <add path="Reserved_ReportViewerWebControl.ashx" verb="*" type="Microsoft.Reporting.WebForms.HttpHandler, Microsoft.Reporting.WebForms, Version=11.0.0.0, Full
```

## **Summary**

This paper identified and explained components and data and file structure that are required to implement proposed system. File structure for system development, database structure, Required

UI design for the system has been explained. Additionally, potential delivery environment for the project and project interaction has been evaluated. Evaluation of the system shows developed project will be web application and XML will be used for data exchange.



## Task 6

**Implement** the .NET programming solution based on the prepared design specification. [3.1]

**Implement** event handling using control structures to meet the design algorithms. [3.2 D2]

**Identify** and implement opportunities for error handling and reporting [3.3]

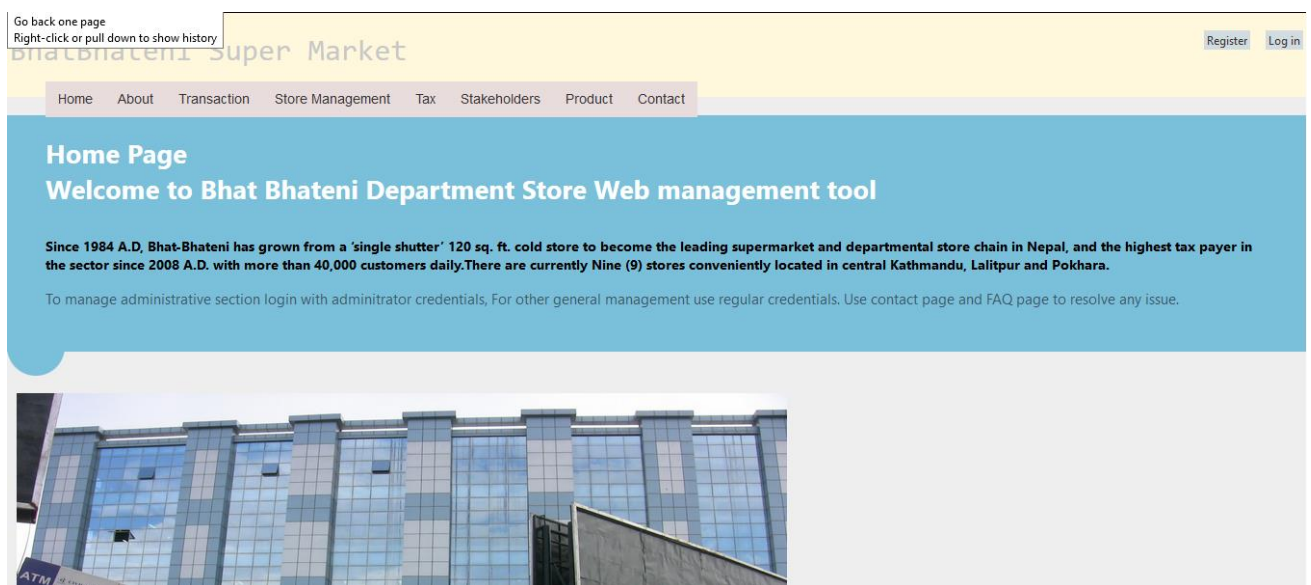
**Make** effective use of an Integrated Development Environment (IDE) including code and screen templates. [3.4]

## Implementation of UI

The graphic element of system is developed using various web form controls. Each designed pages are saved as web page with extension .aspx.

### Home

#### Graphic User Interface



### Source

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="BrownBird.Default" %>
D:\BrownBirdAtut\BrownBird.App_Start\UserInterface\Default.aspx
<asp:ContentPlaceHolder ID="FeaturedContent">
    <section class="featured">
        <div class="content-wrapper">
            <hgroup class="title"><h1><%: Title %></h1>
            <div>
                <h1>Welcome to Bhat Bhateni Department Store Web management tool</h1>
            </div>
            </hgroup>
            <p>
                <h6>
                    Since 1984 A.D, Bhat-Bhateni has grown from a 'single shutter' 120 sq. ft. cold store to become the leading supermarket
                    and departmental store chain in Nepal, and the highest tax payer in the sector since 2008 A.D. with more than 40,000 customers daily.
                    There are currently Nine (9) stores conveniently located in central Kathmandu, Lalitpur and Pokhara.
                </h6>
            </p>
            <p>
                To manage administrative section login with administrator credentials, For other general management use regular credentials. Use contact page a
            </p>
        </div>
    </section>
</asp:ContentPlaceHolder>
```

## BhatBhateni Super Market

[Home](#) [About](#) [Transaction](#) [Store Management](#) [Tax](#) [Stakeholders](#) [Product](#) [Contact](#)

Bhat-Bhateni offers a full range of 120,000 products from almost 1000 local and international suppliers, including a wide range of groceries, fresh fruits and vegetables; a broad range of leading international liquor, toiletries and cosmetics brands; and an extensive choice of kitchenware, clothing, sports, toys and electrical items. Furthermore, both premises include excellent value jewellery stores that offer a wide range of gold and silver ornaments in both traditional and modern styles.

Source

Sale

[Home](#)
[About](#)
[Transaction](#)
[Store Management](#)
[Tax](#)
[Stakeholders](#)
[Product](#)
[Contact](#)

## Date \_\_\_\_\_

**Add to Cart**

**Sale**

50850

## Source

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Sales.aspx.cs" Inherits="BrownBird.UserInterface.SaleUI.Sales" %>

<asp:Content ID="Content2" ContentPlaceHolderID="FeaturedContent" runat="server">
    <h3>
        Sales Management
    </h3>
</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
    <div id="MainContent" class="dynamic">
        <div id="UIControl">
            <table style="width: 100%;">
                <tr>
                    <td class="auto-style1">
                        <asp:Label ID="Label2" runat="server" Text="Customer"></asp:Label>
                    </td>
                    <td class="auto-style2">
                        <asp:DropDownList ID="cmbCustomer" runat="server" CssClass="dynamic" Height="17px" Width="128px">
                        </asp:DropDownList>
                    </td>
                </tr>
            </table>
        </div>
    </div>
</asp:Content>
```

## Inventory

BhatBhateni Super Market			Register	Log in
<a href="#">Home</a> <a href="#">About</a> <a href="#">Transaction</a> <a href="#">Store Management</a> <a href="#">Tax</a> <a href="#">Stakeholders</a> <a href="#">Product</a> <a href="#">Contact</a>				
ProductID	ProductName	Stock		
1	Chocolate	9354		
2	Football	20000		
7	Adidas Football 5' Black	-10		

## Source

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="Inventory.aspx.cs" Inherits="BrownBird.UserInterface.I
<%@ Register assembly="Microsoft.ReportViewer.WebForms, Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91" namespace="Microsoft.Reporti
<%@ Register assembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" namespace="System.Web.UI.DataVis
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="FeaturedContent" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
    <asp:GridView ID="GridView1" runat="server" AllowSorting="True" AutoGenerateColumns="False" CellPadding="4" DataKeyNames="ProductID" DataSourceID="s
        <AlternatingRowStyle BackColor="White" ForeColor="#284775" />
        <Columns>
            <asp:BoundField DataField="ProductID" HeaderText="ProductID" ReadOnly="True" SortExpression="ProductID" />
            <asp:BoundField DataField="ProductName" HeaderText="ProductName" SortExpression="ProductName" />
            <asp:BoundField DataField="Stock" HeaderText="Stock" SortExpression="Stock" />
        </Columns>
        <EditRowStyle BackColor="#999999" />
        <FooterStyle BackColor="#50789D" Font-Bold="True" ForeColor="White" />
        <HeaderStyle BackColor="#50789D" Font-Bold="True" ForeColor="White" />
        <PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center" />
        <RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
        <SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333" />
        <SortedAscendingCellStyle BackColor="#E9E7E2" />
        <SortedAscendingHeaderStyle BackColor="#506C8C" />
```

## Department

BhatBhateni Super Market			Register	Log in
<a href="#">Home</a> <a href="#">About</a> <a href="#">Transaction</a> <a href="#">Store Management</a> <a href="#">Tax</a> <a href="#">Stakeholders</a> <a href="#">Product</a> <a href="#">Contact</a>				
Add New Department				
Update	Delete	DepartmentID	DepartmentID1 Name	Description
Update	Delete	4	4	Marketing This is Marketing Department of Kalanki Bhatbhateni.

## Source

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="DepartmentView.aspx.cs" Inherits="BrownBird.U
D:\BrownBird\Atut\BrownBird\App_Start\UI\DepartmentUI\DepartmentView.aspx>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="FeaturedContent" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
<asp:HyperLink ID="HyperLinkNewDepartment" runat="server">
<a href="DepartmentDetail.aspx">Add New Department</a>
</asp:HyperLink>
<asp:GridView ID="GridViewDepartmentList" runat="server" OnRowUpdating="GridViewDepartmentList_RowUpdating" OnSelectedIndexChanged="GridVie
<AlternatingRowStyle BackColor="White" ForeColor="#284775"/>
<Columns>
<asp:ButtonField CommandName="Update" HeaderText="Update" Text="Update"/>
<asp:ButtonField CommandName="Delete" HeaderText="Delete" Text="Delete"/>
</Columns>
<EditRowStyle BackColor="#999999"/>
<FooterStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White"/>
<HeaderStyle BackColor="#5D7B9D" Font-Bold="True" ForeColor="White"/>
<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center"/>
<RowStyle BackColor="#F7F6F3" ForeColor="#333333"/>
<SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"/>
<SortedAscendingCellStyle BackColor="#E9E7E2"/>
```

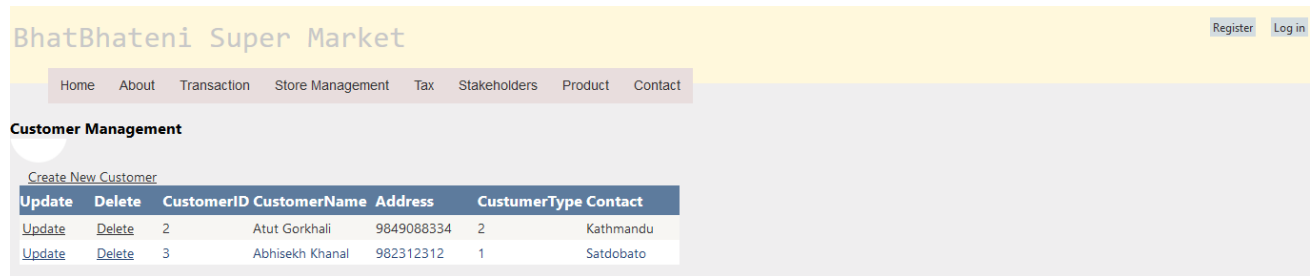
## New Tax

The screenshot shows the 'New Tax' page of the BhatBhateni Super Market. The page has a yellow header with the site name and 'Register' and 'Log in' links. A navigation bar contains links for Home, About, Transaction, Store Management, Tax, Stakeholders, Product, and Contact. The 'Tax' section is active, displaying a form with three input fields: 'Tax ID', 'Description', and 'Rate'. A 'Save' button is located below the 'Rate' field. The footer of the page states 'Bhat bhateni Supermarket | Ranked #1 in National Ranking | Atut Gorkhali'.

## Source

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="TaxDetail.aspx.cs" Inherits="BrownBird.TaxUI.TaxDetail" %>
D:\BrownBird\Atut\BrownBird\App_Start\UI\TaxUI\TaxDetail.aspx>
<style type="text/css">
<.auto-style2 { text-align: right; }
</style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="FeaturedContent" runat="server">
<p>
<asp:Label ID="TaxIDLabel" runat="server" Text="Tax ID">
</p>
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
<table style="width: 100%;">
<tr>
<td class="auto-style2">
<asp:Label ID="TaxIDLabel" runat="server" Text="Tax ID">
</td>
<td>
<asp:TextBox ID="TaxIDTextBox" runat="server" ReadOnly="True" Visible="False">
</td>
<td>
<asp:Label ID="TaxIDLabel" runat="server" Text="Tax ID">
</td>
</tr>
</table>
```

## Customer



## Source

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="CustomerView.aspx.cs" Inherits="BrownBird.UserInterfac
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="FeaturedContent" runat="server">
<h3>Customer Management</h3>
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="MainContent" runat="server">
<asp:HyperLink ID="HyperLinkNewCustomer" runat="server">
<a href="CustomerDetail.aspx">Create New Customer</a>
</asp:HyperLink>
<asp:GridView ID="GridViewCustomerList" runat="server" OnRowUpdating="GridViewCustomerList_RowUpdating" OnSelectedIndexChanged="GridViewCustomerList
<AlternatingRowStyle BackColor="White" ForeColor="#284775"/>
<Columns>
<asp:ButtonField CommandName="Update" HeaderText="Update" Text="Update"/>
<asp:ButtonField CommandName="Delete" HeaderText="Delete" Text="Delete"/>
</Columns>
<EditRowStyle BackColor="#999999"/>
<FooterStyle BackColor="#5D789D" Font-Bold="True" ForeColor="White"/>
<HeaderStyle BackColor="#5D789D" Font-Bold="True" ForeColor="White"/>
<PagerStyle BackColor="#284775" ForeColor="White" HorizontalAlign="Center"/>
<RowStyle BackColor="#F7F6F3" ForeColor="#333333"/>
<SelectedRowStyle BackColor="#E2DED6" Font-Bold="True" ForeColor="#333333"/>
```

## Implementation of Event handling

While utilizing a PC, if client snaps mouse or squeeze console catch or communicate with touch screen, PC reaction to the movement done by client. This is the thing that event driven programming is all about. The framework must have some event perceive system and in addition mechanism handling such event. Event driven programming has attributes like event audience (listener), handler and trigger that demonstrations event recognizer, handler and responder. To develop web application for Bhat-Bhateni department store, program should utilize various events and event handling features provided by operating system.

## Event

Event is client activities, for example, mouse snap, and key press, information through sensors to program, for example, scanner and messages from different projects. There are different sorts of event, for example, mouse event, console event, HTML object event, Form event and User control Events.

In developed system some of the fundamental client produced events are mouse snap, mouse double tap, structure burden, mouse move, key down, key up center, leave and so on.

## Event Handling

This describes method that will handle an action when the event is activated. It is segments of code that handles the activities done after event triggers. It is the real code that runs when a trigger happens. Various event handling in developed system is shown below here.

### Button Click Event Handling

When button is clicked, codes in event handles is executed. Button click event is used enormous in the developed system: example, add student, add to cart, perform sale etc. Example below shows event handler code for perform sale button.

```
protected void ButtonAddtoCart_Click(object sender, EventArgs e)
{
    var productId = int.Parse(txtProductID.Text.Trim());
    var productName = LabelProductName.Text.Trim();
    var price = float.Parse(LabelPrice.Text.Trim());
    var quantity = int.Parse(TextBoxQuantity.Text.Trim());
    var discountRate = float.Parse(cmbDiscount.SelectedValue);
    var taxRate = float.Parse(cmbTax.SelectedValue);
    var total = (price*quantity);
    var tax = (total*(taxRate/100));
    total += tax;
    var discount = (total*(discountRate/100));
    total -= discount;
    //show grandtotal
    var grandTotal = float.Parse(LabelGrandTotal.Text);
    grandTotal = grandTotal + total;
    LabelGrandTotal.Text = grandTotal.ToString();
    //add rows to datatable
    _dTable.Rows.Add(productId, productName, price, quantity, taxRate, discountRate, total);

    //show dtPurchase to gridview
    GridViewCart.DataSource = _dTable;
    GridViewCart.DataBind();

    //clear for next add
    ClearUi();
}
```

### Text Changed Event Handling

Like button click event, text change event handler code is executed when text change occur in textbox. Codes in example below is utilized in sale page.

```
protected void txtProductID_TextChanged(object sender, EventArgs e)
{
    var value = txtProductID.Text.Trim();

    if (value == "")
    {
        LabelProductName.Text = "";
        LabelPrice.Text = "";
        return;
    }

    var p = _productBl.GetProduct(int.Parse(value));
    LabelProductName.Text = p.ProductName;
    LabelPrice.Text = p.Price.ToString();
}
```

## **Implementation of Control Structures**

In C# programming, control structure alludes to condition and circling structure. These structures are utilized as a part of programming to execute programming rationales. With help of these structures designer accomplish business necessities. Different control structures utilized as a part of planned Bhat-Bhateni department store framework is recorded underneath here in this document.

### **If statement**

IF-else if permits to check condition and control stream of system. Word IF is trailed with check condition. If condition is valid, it executes statements within IF block. Otherwise, articulations from else square is get executed. In developed system, if and else-if statements are utilized to control flow of the code. Certain statement is executed is check condition is true. And other statements are executed is condition is false. Example below shows if partner is deleted it will show success message else it will show error message.

```
if (bsPartnersBl.Delete_BsPartners(bsPartnersobj))
{
    LabelMessage.Text = "BsPartners Deleted successfully!";
    LoadGridView();
}
else
{
    LabelMessage.Text = "Unsuccessfull!";
}
```

### **For statement**

Looping is rehashing square of code until the characterized condition matches. It is exceptionally basic yet valuable system. In current project for loop is utilized to add items to the cart while buying or selling.

```

for (var i = 0; i < _dataTable.Rows.Count; i++)
{
    var productId = int.Parse(_dataTable.Rows[i][0].ToString());
    var productName = _dataTable.Rows[i][1].ToString();
    var price = float.Parse(_dataTable.Rows[i][2].ToString());
    var quantity = int.Parse(_dataTable.Rows[i][3].ToString());
    var taxRate = int.Parse(_dataTable.Rows[i][4].ToString());
    var discountRate = int.Parse(_dataTable.Rows[i][5].ToString());
    var total = float.Parse(_dataTable.Rows[i][6].ToString());
    var sld = new ClsPurchaseDetails();
    sld.ProductId = productId;
    sld.Quantity = quantity;
    sld.TotalPrice = total;
    sld.DiscountRate = discountRate;
    sld.TaxRate = taxRate;
    sld.PurchaseId = purchaseId;
    _classInventory.ProductId = productId;
    _classInventory.ProductEntry = quantity;
    var x = _inventoryBl.Increase_Inventory(_classInventory);
    var y = _purchaseDetailsBl.Insert_Purchases(sld);
    ifsuccess = w && x && y;
}

```

### Switch statement

A switch statement permits a variable to be tried for fairness against a rundown of qualities. Every quality is known as a case, and the variable being exchanged on is checked for every switch case. Different block of code is executed according to the value. Example below shows switch statement implementation in developed project.

```

switch (e.which) {
    case 37: this.prev(); break
    case 39: this.next(); break
    default: return
}

```

### Error Handling

Error handling alludes to the suspicion, identification, and determination of programming, application, and correspondences errors. One of the characteristics of such tools is to handle the error in a way that it prevents system termination and provide alternative flow to the program. In developed system, Try-Catch and finally tool has been utilized.

### Try Catch and Finally

Try segment tries to execute codes inside. If there is an error it block of code, flow of programming is direct to catch section, if there isn't any error, all statements in try block is executed and then flow is directed to finally block. In a condition where try section has error then catch section catches the error and executes codes in catch segment. After execution of codes inside catch section, flow of program directed to finally section. This means finally section is called in both scenario, with error and without error. Code below is used in Implemented system to tackle unexpected error.



```
try
{
    String query = "SELECT BS_patnerType,Description FROM tblBsPatnersType";
    SqlCommand cmd = new SqlCommand(query, _connect);
    SqlDataAdapter dAdapter = new SqlDataAdapter(cmd);
    _connect.Open();
    dAdapter.Fill(dtable);
}
catch (Exception ex)
{
    throw ex;
}
finally
{
    _connect.Close();
}
```

### **Summary**

This paper documented the implementation of proposed Bhat-Bhateni management system. UI of the designed has been developed utilizing Web form technology and static html as source. Implementation of UI is well documented in this paper. To implement various feature of system to achieve business requirement, event handling technologies has been utilized by the system. Events like mouse click and text change is used in system. Additionally, system uses various control structure such as if statement, for statement and switch statement while writing program.

Developed system utilizes error handling technology such as try-catch-finally to tackle with error in the program. This helps program to use alternative flow in case of error, preventing the system termination. This document makes effective use of IDE to manage and develop all modules of the system.

## **Task 7**

**Critically** review and test a .NET programming solution and analyze actual test result against expected results to identify discrepancies. [4.1, 4.2]

### **Introduction**

Designed .NET software plan for Bhat-Bhateni supermarket has been implemented. Now before deployment of the developed system it is necessary to carryout tests. Test will allow developer to study the performance of developed system. It enables to understand if all part of application functioning correctly or not. To analyze the test results, various test methodologies has been applied. This paper documents result obtained during testing process. Additionally, this paper also perform critical review of the system.

### **Unit Test**

Unit test portrays testing of littlest testable piece of use. Smallest unit of the system is tested in unit testing. Different module of system is tested to find bugs, and check modules functionality. In test carried out developed system, various units such as sales, add to cart, delete customer, add customer, update customer, etc. were tested. Test results has been documented as test log for further analysis.

### **Stress Test**

Stress testing is the procedure of deciding the capacity of a framework to take a shot at critical conditions (extreme) condition. For instance, it tests, the result when expansive number of information is procedures at single time. For stress testing, validation test has been performed in developed system. Test is performed by passing invalid emails, string as number of products etc.

### **Integrated Test**

Integrated test is utilized for checking the association between distinctive segments of programming and hunt of imperfections. For Example, it tests integration of one module with another. To perform Integration test on Bhat-Bhateni application, integration between application and database, integration with master page and integration of inventory with sale/purchase is tested.

### **Compatibility Test**

It is non-functional test which checks framework behavior in diverse conditions, for example, distinctive working framework, diverse equipment or diverse client. As developed system is browser based application, system was made to run on various web browsers to check it compatibility with different browsers.

Test document for all tests i.e.; unit, integration, compatibility and stress is provided beneath here.

### **Critical Analysis of Test results**

This paper previously carried out various test works on developed .Net system for Bhat-Bhateni supermarket. Unit test verified functioning of small units of the system. Unit test showed all units of the systems are functioning well. Tested units included, functioning of dropdown box, sale button, add to cart function and customer saving feature etc. For compatibility testing, system was tested in chrome, Mozilla and new Microsoft edge. In all browsers, system functioned normally.

Then integration test was carried out. In this test integration with master page, integration with database was tested. It showed all pages inherits master page layout and database is functioning well too. And finally, validation test was performed as stress testing which showed system is well validated. All tests has been carried out successfully.

### **Critical Review**

#### **Introduction**

Bhat-Bhateni supermarket is renowned department store located in various major location within capital city of the country. Since Bhat-Bhateni is using Microsoft excel to manage information about sales, purchases, suppliers, customers, employees etc. Bhat-Bhateni now had planned to develop automated system to store and maintain all transactions information.

This document critically reviews various aspects of developed solution for Bhat-Bhateni. After analyzing problems of the department store a .NET framework based web portal solution was designed. Developed solution is based on the designed using various tools and technologies such as use case diagram, flowchart, context diagram and DFD diagram.

#### **Body**

Developed system is focused on satisfying the requirement of Bhat-Bhateni department store. It has various features like product management, user management, customer management etc. that specifically manages various components of department store. Each sales or purchase information is stored in database through the solution which helps to keep track of inventory, transaction history etc. Strengths and limitations of the system are review here below.

#### **Strengths**

- System has key features that targets requirement of the supermarket

- System is developed using .NET framework which means it can run smoothly on any windows based computer. This will allow supermarket to use existing systems to run new application
- Developed solution has authentication system that helps preventing unauthorized access
- Simple operational method that will help users
- Developed solution is tested and maintained.

#### Limitations

- Solution has very simple UI which may fail to impress users. Application could have been more interactive.
- System is developed using .NET framework which may prevent supermarket from moving to other operating system than windows.

#### Summary

Programming is done in .NET framework to develop solution for Bhat-Bhateni supermarket. This solution has separate modules that manages sales, purchases information of the supermarket. Additionally, developed application also maintains information about products, users, customers etc. Like any solution, this application does have some strength as well as some limitations. It offers user individual webpage for each action. This simple approach makes whole system really simple to use. User can browse through webpages to perform their tasks. To summarize this document, development solution will help Bhat-Bhateni department store to automate record keeping process.

**Task 8**

**Evaluate** independent feedback on a developed event driven programming solution and make necessary recommendations for improvement. [4.3, D3]

**Feedback**

Feedback is process in which the impact or yield of an activity is "returned" to adjust the following activity. Criticism is fundamental to the working and survival of every single project. As a two-way stream, criticism is innate to all cooperation, whether human-to-human, human-to-machine, or machine-to-machine. In an organizational setting, feedback from a group/individual is taken over a product to analyze them and make necessary modification. For example, a customer feedback is customer's reaction to a product or organization's service.

As proper feedback system is key to project success, feedback on developed system was taken from co-worker (Nabin K. Singh) to understand its limitations and positive aspects. To get feedback, various data collection techniques such as interview, questionnaire, survey was utilized. This paper documents the feedback provided by Mr. Nabin and that feedback is evaluated independently.

<b><u>Independent Feedback</u></b>	
<b>Project Title: Management System For Bhat-Bhateni Department Store using .NET Framework</b>	
<b>Feedback Given By:</b> <b>NABIN K. SINGH</b>	<b>Signature:</b> <b>Date: 4/20/2015</b>
<b><u>Project Description</u></b> Bhat-Bhateni supermarket was dealing with its day by day forms in Microsoft exceed expectations yet because of it impediment, department store was confronting a few issues. This project Plan, Design and Implement and Test solution using .NET framework to automate and record transaction and other information	
<b>Important Features of the Project</b>	Following are some of the key features of developed solution: <ol style="list-style-type: none"> <li>1. Solution has User authentication system, only administrator are allowed to perform transaction</li> <li>2. Allows to store and maintain department, employees, customer information</li> <li>3. Perform sales and purchase transactions</li> </ol>

	4. Create and maintain various types of discounts and tax system 5. Add and maintain products and product types 6. Inventory Information page to display product stock	
<b>Checklists</b>	<b>YES / NO</b>	<b>Comment</b>
Have identified objects and data and file structures required to design the system?	Yes	Has identified required tables, file structures, objects
Have designed UI and database?	Yes	Has designed Web form UI and database
Does the system has been implemented in align with the proper design?	Yes	Developed solution is based on designed pseudo-code, use-case diagram, context diagram.
Have been critically tested and reviewed the system or not?	Yes	Developed solution has been critically tested using various types of tests and reviewed
Does the system fulfill the requirements of the clients?	Partially Yes	Developed solution satisfies most of the client's requirements though there are some limitations
Have developed the onscreen help form to assists the user of the computer program?	Yes	User documentation is created to provide guideline to the users
<b>Significance of the Project</b>	While giving feedback following positive points are noted: <ul style="list-style-type: none"> <li>Developed system has successfully matched system design and user requirements.</li> <li>Application UI is simple and fast</li> <li>System is thoroughly tested</li> <li>Only admin are allowed to use critical features such as sales/purchase</li> <li>Passwords are in saved in database in encrypted form</li> <li>Navigating through pages are simple</li> </ul>	
<b>Project Limitation</b>	While giving feedback for the developed system, following limitation are noted:	

	<ul style="list-style-type: none"> <li>• User interface is simple and has classic but it get boring as could have been better with modern look</li> <li>• System is not usable for general visitors as it does not give information about company, its events and activates</li> <li>• It does not have history keeping for sales and purchase</li> <li>• System does not have password recovery system.</li> </ul>
--	---

### **Feedback Evaluation**

Feedback Evaluated By:

**ATUT GORKHALI**

Signature:

Date: 4/30/2015

### **Evaluation of the Feedback:**

Developed system has very simple User interface. As system UI is designed using Web form technology, it offers limited UI options. Furthermore, system has minimal use of interactive contents to make system load faster. As it is targeted for organizational works only, UI does not play huge role. Though UI can be improved using modern tools and techniques.

Developed system is developed to be used as management system for Bhat-Bhateni. It should automate the transaction and inventory system. Which implemented system has accomplished successfully. System requirement provided by Bhat-Bhateni does not signifies involvement of general visitor. Through, for general visitor web application has about page which provides basic information about supermarket and contact us page allows general visitor to connect with organization.

Implemented system has features like sales transaction and purchase transaction. Product inventory is programmed to get updated every time successful transaction happens. Stock size in inventory increases when product is purchased and stock decreases when that product is sold to customer. As feedback states above, system does not have transaction history tracking mechanism. Having history keeping for transaction will allow organization to study its transaction as track any inaccuracy is account sheet.

User or administrator requires proper combination of username and password to access the system. User are allowed to access authorized features only. In long run, there can be scenario when a user forgets his/ her password or ever administrator can forget password. As passwords are saved after encryption due to security reasons, view database using Microsoft server also

does not solve the issue. Most of the web based system uses password recovery feature that allows password resetting. Lack of this feature can create critical complications and can have impact on productivity by jeopardizing services.

#### **Recommendations for future consideration:**

This document has analyzed feedback provided by Mr. Nabin k. Singh on developed system. His feedback included some positive as well as some negative reaction. Developer of the system then evaluated the given feedback. Now it is essential to give feedback for the future reflecting the evaluation of feedback.

##### **Move to E-commerce site**

Developed system does not prioritize general visitor in current system. But converting developed system into e-commerce site will address general visitor with great deal as well as it will improve business benefits of the system. Developed system will work together with recommended e-commerce system and will work as back bone of the new system. As Current system already has product management and customer management system, new pages will be required to list products for sale. Visitor will able to browse through products and order product of their choice. Moving to e-commerce will improve the value of the developed system.

##### **Improve UI with DevCraft**

As by default, visual studio offers limited UI controls to design web application, third party tools like DevCraft can be utilized. It offers modern UI libraries, great performance and modern look to the system (DevCraft, n.d.). With DevCraft, it is easier to customize the system. It also offers support and maintenance for the system.

##### **Add password recovery feature**

System should add password recovery feature. For this an email address can be asked during user registration. And when user asks for password recovery, system should send password to provided password. This will improve security of the system as well as help to avoid complications.



**Add History keeping feature**

As system does not has history keeping system to display transaction history. To add this feature, developer need to add web page that will display relevant data from sales/purchase table. This will help user to get necessary transaction information from the past.

**Reference**

DevCraft (n.d.) *DevCraft* [Online] Available: <http://www.telerik.com/devcraft> Accessed [4/29/2015]

## **Task 9**

Create user documentation or user manual for the developed event driven program solution. [4.4]

### **Introduction**

This document serves as user guideline for developed .NET application. Reader will get step guides for using all feature of the system. This guideline manual is divided into two parts. **General help** would provide fundamental guideline for the system whereas **Operational help** would provide more in-depth system operation guideline.

### **General Helps**

#### **Navigating Through Pages**

To navigate through pages of the web application, click the page link from the navigation menu.

#### **Visit about Page**

- To go to about page click on about menu from navigation bar [Figure 1]

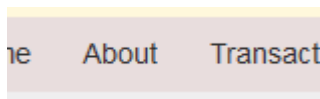


Figure 8 About Menu from Navigation bar

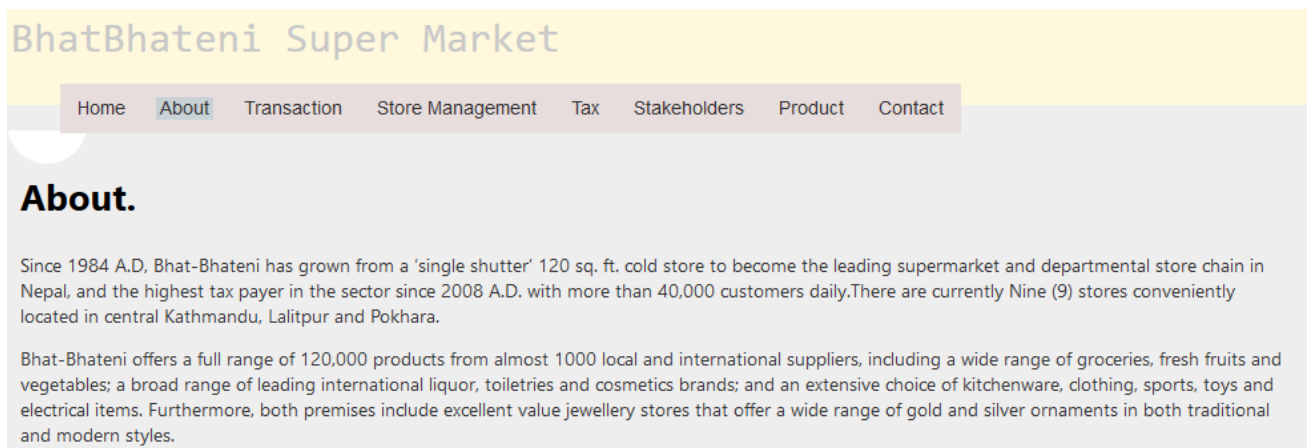


Figure 9 about Page

#### **Visit Contact Us**

- To go to about page click on contact us menu from navigation bar [Figure 3]

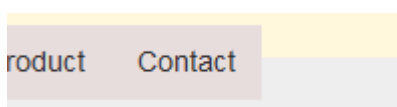


Figure 10 contact us menu

## Contact Us Our Contact Information

**Bhat-Bhateni Head Office, Bhat-Bhateni, Kathmandu**

**Ms. Nabina Moktan**

Tel: 977-1-4419181, 4413825

Email: [customer@bbsm.com.np](mailto:customer@bbsm.com.np)

**Bhat-Bhateni Maharajgunj, Narayan Gopal Chowk, Kathmandu**

**Ms. Laxmi Shrestha/Rejina Pradhan**

Tel: 977-1-4016130, 4016131, 4016132

Email: [customer@bbsm.com.np](mailto:customer@bbsm.com.np)

Figure 11 contact us page

### Operational Helps

Operational help segment would provide guideline to utilize features of the web application such as sales, customer management, inventory system etc.

### Registering New User

- Click the Register Link to go to user registration page
- Fill the form with requirement information [Figure 5]
- Click on submit button
- Done

### **Register.** Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

User name

Email address

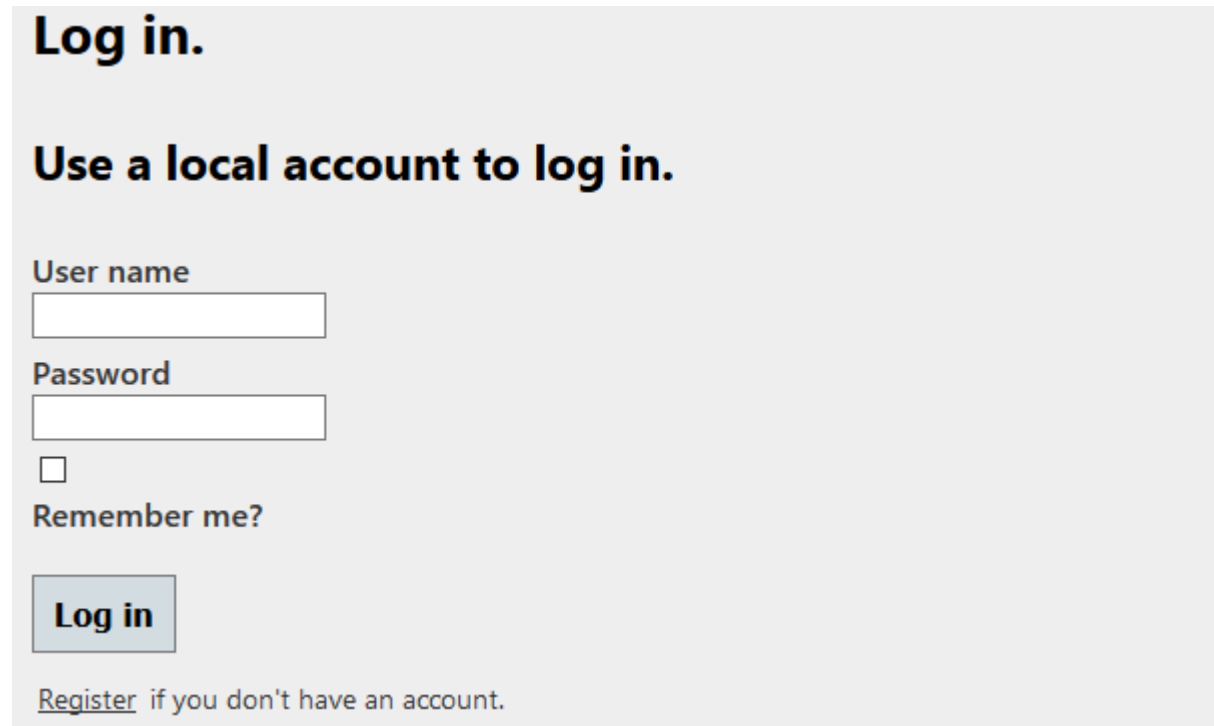
Password

Confirm password

Figure 12 User registration form

### Login

- To log into the system first click on login link
- On the login page, fill up the login form with right credential for authentication [Figure 6]
- If you are authorized user, you will get access to the system.



**Log in.**

**Use a local account to log in.**

User name

Password

☐

**Remember me?**

**Log in**

[Register](#) if you don't have an account.

Figure 13 Login form

### Logout

- To logoff from logged in system, simply click on logout link for the top right corner [Figure 7]



Hello, admin ! **Log off**

Figure 14system logout link

## **Customer Management**

### Add New Customer

- Go to Customer page first by clicking customer from navigation menu
- Customer is located under stakeholders section of the navigation [Figure 8]
- Click on add new customer link to open new customer page [Figure 9]
- Fill the requirement information and save

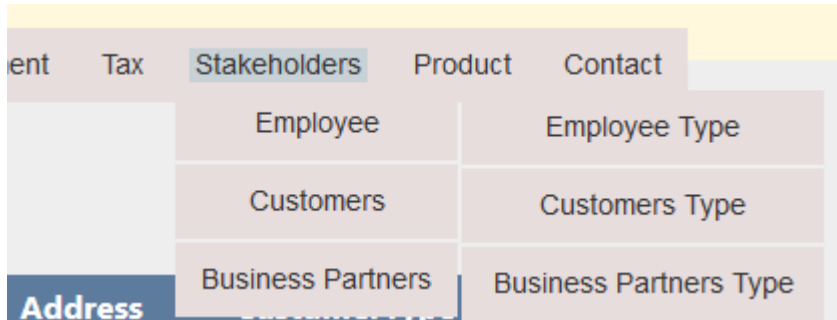


Figure 15 Customer menu under stakeholders

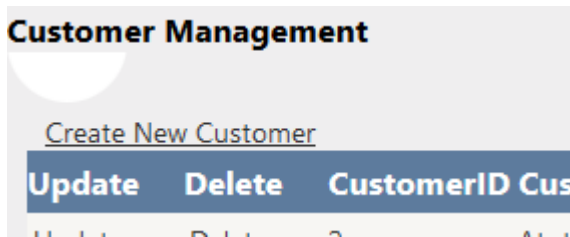


Figure 16 create new customer

### Edit Customer

- First go to customer management page [follow new customer guideline if you have problem with this step]
- Click on Update customer form table [Figure 10]
- Modify the customer information and save. [Figure 11]

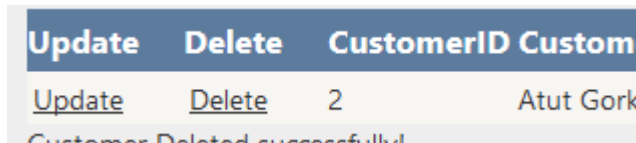


Figure 17 Edit Customer Information

Customer Name

Contact

Address

Customer Type

Figure 18 Customer Modify page

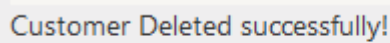
### Delete Customer

- First go to customer management page [follow new customer guideline if you have problem with this step]
- Click on delete form the table [Figure 12]
- Delete is done. [Figure 13]



Delete	CustomerID	CustomerName
<a href="#">Delete</a>	2	Atut Gorkhali

Figure 19 Delete Customer



Customer Deleted successfully!

Figure 20 Delete success message

### Discount Management

#### Add New Discount

- Go to discount page first by clicking discount from navigation menu
- Customer is located under Tax section of the navigation [Figure 14]
- Click on create new discount [Figure 15]
- Provide the necessary information →click save→Done

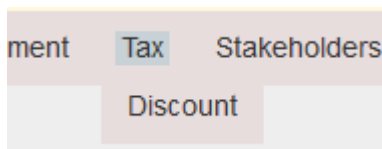


Figure 21 Discount Menu

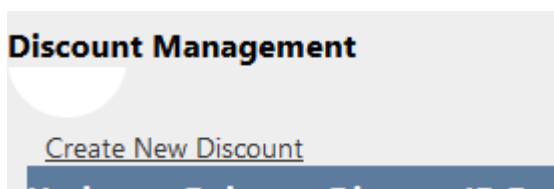


Figure 22 Link to add new discount

#### Modify Discount

- First go to discount management page
- Click on Update discount form table [Figure 16]

- Modify the customer information and save. [Figure 17]

Update	Delete	DiscountID	Description
<a href="#">Update</a>	<a href="#">Delete</a>	1	VIP Offer
<a href="#">Update</a>	<a href="#">Delete</a>	3	Staff Offer
<a href="#">Update</a>	<a href="#">Delete</a>	4	No Discour

Figure 23 Modify Discount

Discount ID

Description

Rate

Figure 24 save Updates

### Delete Discount

- First go to discount management page
- Click on delete form the table [Figure 18]
- Delete is done. [Figure 19]

Update	Delete	DiscountID	Description
<a href="#">Update</a>	<a href="#">Delete</a>	1	VIP Offer
<a href="#">Update</a>	<a href="#">Delete</a>	3	Staff Offer
<a href="#">Update</a>	<a href="#">Delete</a>	4	No Discour

Figure 25 Delete discount

Discount Deleted successfully!

Figure 26 Succesfull delete

### Sales Management

- Go to Sale page first by clicking Sale from navigation menu which is located under
- Sale is located under Transaction section of the navigation [Figure 20]
- Fill the necessary information [Figure 21]

- Add product into cart
- Complete the sale by clicking Sale button [Figure 21] → Done

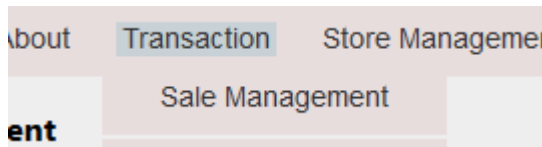


Figure 27 Sale Menu

**Sales Management**

Customer:

Product:

Price: 90 Chocolate

Quantity:

Discount Mode:

Tax Mode:

Date:

ProductID	ProductName	Price	Quantity	Tax(%)	Discount(%)	Total
2	Football	500	500	13	10	254250

**Add to Cart**

Figure 28 Sales management Page

**Sale**

**Grand Total** 254250

### Inventory Management

- Go to Inventory page first by clicking Inventory from navigation menu
- Inventory is located under Store management section of the navigation [Figure 22]



Figure 29 Inventory Menu Page



Home	About	Transaction	Store Management	Tax	Stakeholders	Product	Contact
ProductID	ProductName	Stock					
1	Chocolate	9354					
2	Football	20000					

Figure 30Inventory Page

### **Additional Help**

This document provided guideline for most of the features of the developed system. In case of further assistance, please contact support team via email to [admin@Bhat-Bhateni.com](mailto:admin@Bhat-Bhateni.com).

**Task 10**

**Create** documentation for the support and maintenance of computer program. [4.5, M3]

**Introduction**

This document serves as guidance for maintenance and support for developed “Bhat-Bhateni Department Store using .NET Framework”. This system is developed to provide automated service to Bhat-Bhateni supermarket. To provide required support to technical and maintenance team, this document covers, design layout of the system, choice of tools, task break down approach, product deployment techniques and maintenance techniques for the system.

**Technology Used**

To develop system for supermarket using .NET framework, ASP.Net is utilized. ASP.NET permits programmer to utilize a full featured programming language, for example, C# (purported C-Sharp) or VB.NET to assemble web applications effortlessly. ASP.Net is .Net framework based system that offers all benefits of .Net technologies. Benefits of using ASP.Net to develop proposed system are as follows:

1. ASP.NET still renders HTML

Internet still has data capacity restrictions and everyone does use internet with same capacity and same web browser. To tackle these issues plain HTML plays huge role. . This implies pages won't look very as astounding as a completely fledged application running under Windows, however with a touch of expertise and inventive energy, you can make some fairly astonishing web applications with ASP.NET.

2. Object Oriented Programming on the Internet

ASP.Net is OOP based language. This allows programmer to develop neat and well managed code structure web application.

3. XML support

ASP.Net offers support for XML. XML in developed system is used for storing location for libraries and database.

4. Class Libraries

ASP.Net offers large options of class libraries. Programmer can utilize these libraries to achieve requirement of the system.

## Choice of Tools

To developed proposed system several tools and technologies has been utilized. Those tools are listed here.

### Visual Studio

Visual Studio .NET is a comprehensive arrangement of development technology from Microsoft for building ASP Web system, XML Web system, desktop applications, and versatile applications. .Net languages uses integrated development environment, which permits them to share apparatuses and encourages in the production of blended language solution. Visual studio allows programmer to take full advantage of .Net technology. Furthermore, Visual studio allows to manage program file structure, find errors and debug the system.

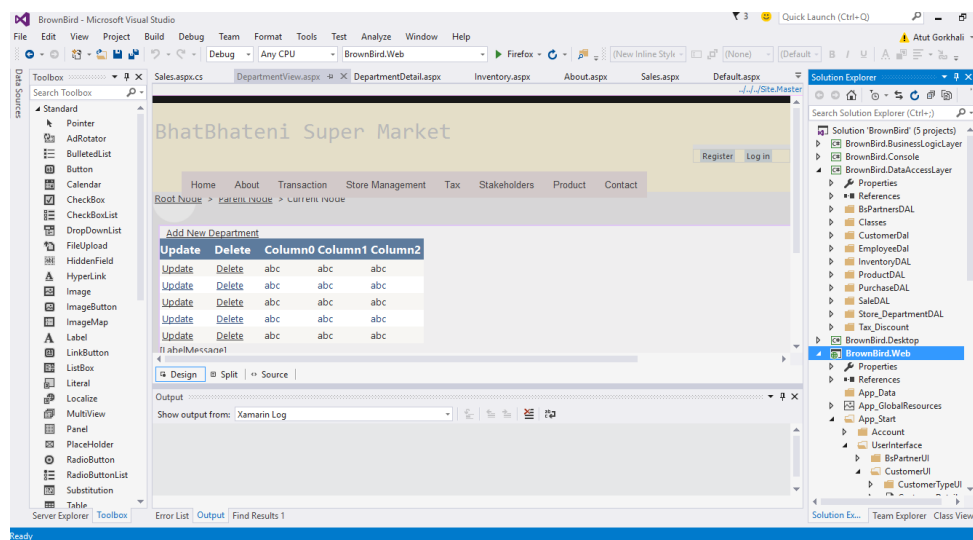


Figure 31 Visual studio

### SQL Server

Microsoft SQL Server is an application used to make PC databases for the Microsoft Windows group of server working frameworks. Microsoft SQL Server gives an environment used to create databases that can be gotten to from workstations, the Internet, or other media, for example, an individual advanced right hand (PDA). Database for developed system is designed and maintained using Microsoft SQL server 2008.

## Design

To design user interface of the developed system static HTML is utilized. Programmer can utilize any HTML, JavaScript or CSS. You can likewise have structures, pictures, recordings, JavaScript - anything that programs can deal with. To methods are utilized while designing UI for the system.

### 1. ASP.Net UI source

This method is accessed by clicking source tab in IDE. This allows programmer to add/modify necessary static HTML code. Figure 3 demonstrates how ASP.Net utilized HTML rendering.

### 2. Web Form

With web form programmers can easily developed web application just by dragging and dropping UI control offered by IDE to the webpage.

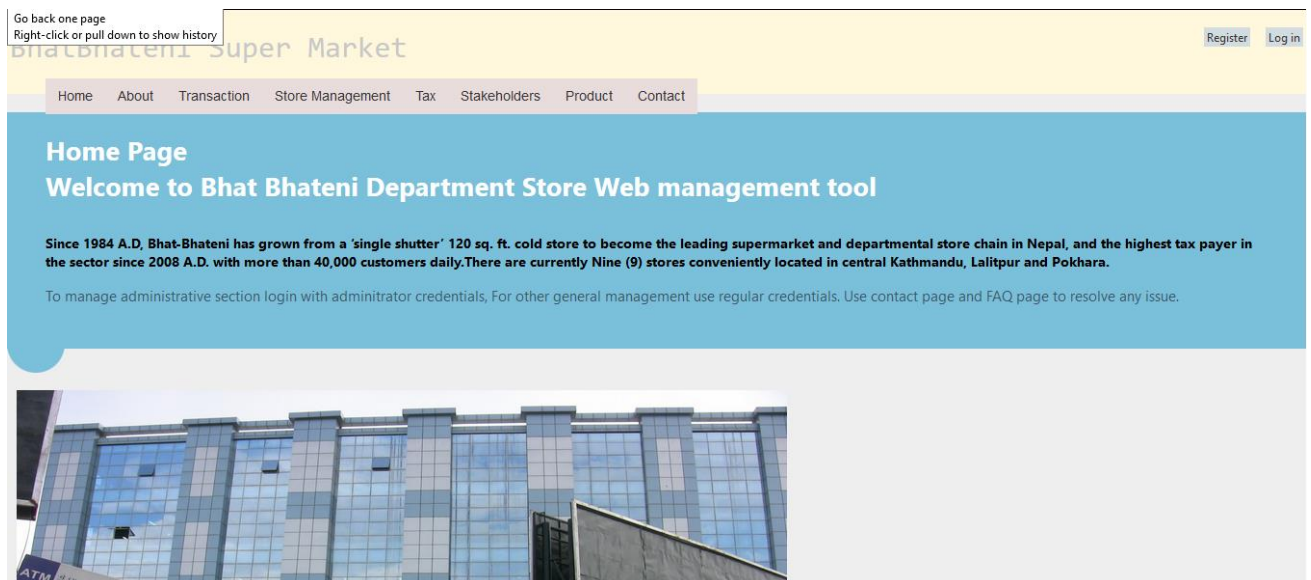


Figure 32UI for the developed system

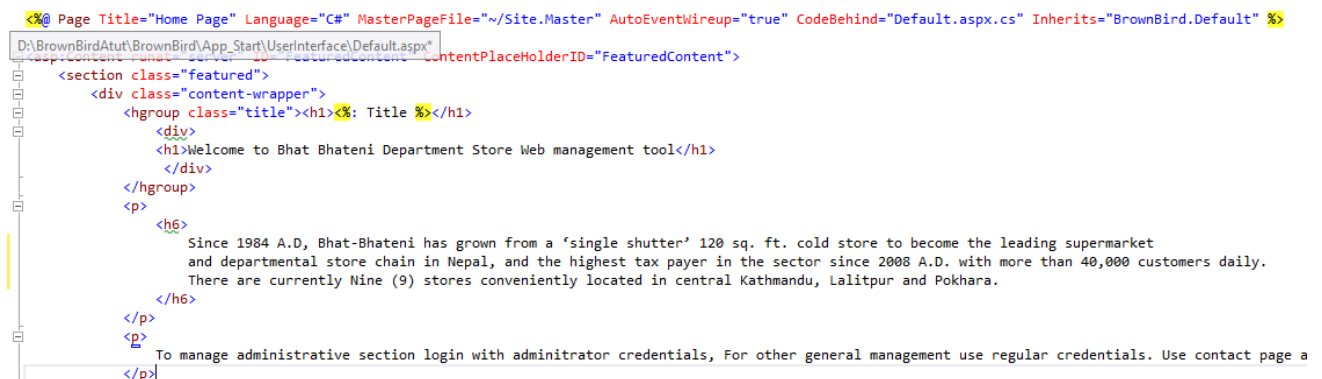


Figure 33ASP.NET source code for the UI

## Master Page

Gives us an approach to make normal arrangement of UI components that are required on numerous pages of our site.

Content page utilize UI elements of master page while rendering as shown in figure 4.

Developed .Net system uses master page technology to provide common UI looks for most of the page. To access the master page from contents pages following codes are used.

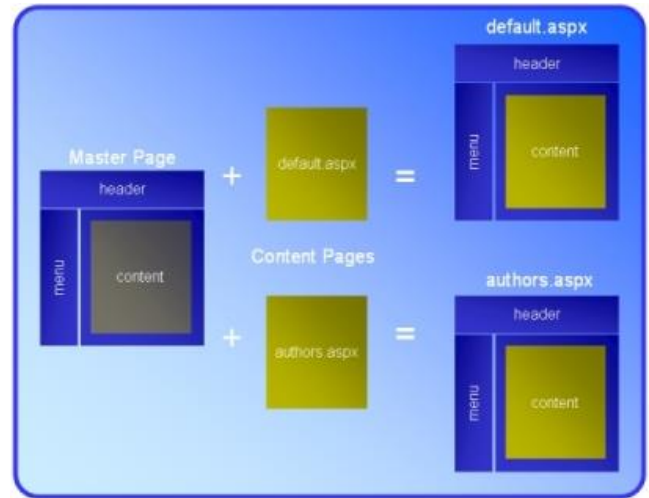


Figure 34 Master page and content page (Source: www.asp.net)

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
```

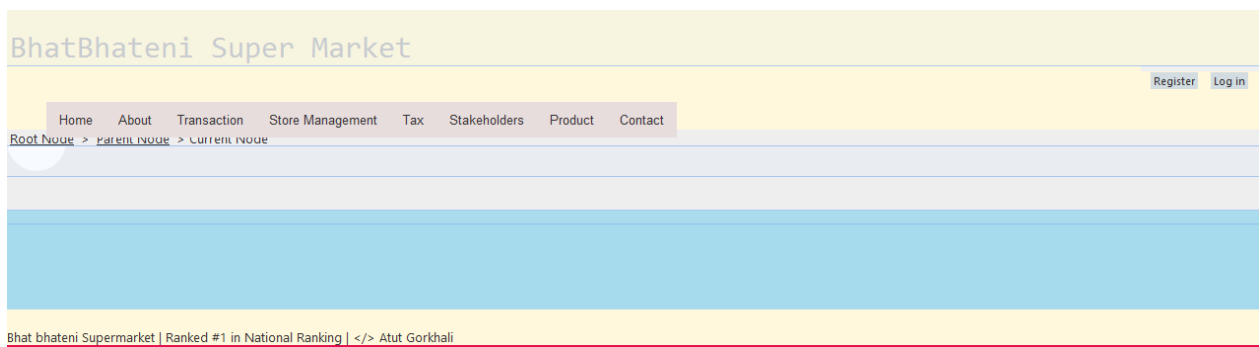


Figure 35 Master page for developed system

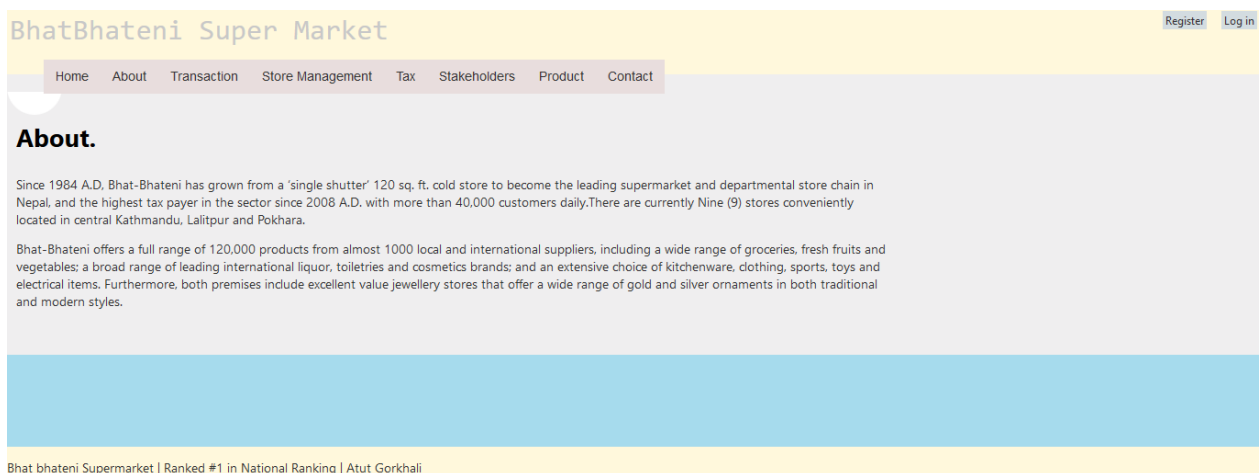


Figure 36 Web page utilizing master page

## CSS

While designing UI for the system Cascaded style sheet is utilized.

```
@font-face {
    font-family: Yanone Kaffeesatz;
    src: url('../fonts/YanoneKaffeesatz-Regular.eot');
    src: local("Yanone Kaffeesatz"), url(
'../fonts/YanoneKaffeesatz-Regular.ttf');
}

@font-face {
    font-family: News Cycle;
    src: url('../fonts/NewsCycle-Regular.eot');
    src: local("News Cycle"), url('../fonts/NewsCycle-Regular.ttf');
}

html {
    height: 100%;
}
```

## Task Breakdown structure

To properly manage plan, design and implement developed system, project is broken into various phases. Task breakdown in project administration is venture of deteriorating the work into littler and reasonable errands. Developed system for Bhat-Bhateni is likewise accomplished by separating the task into littler stages. To depict these stages Gantt chart for the task is drawn.

## Gantt chart

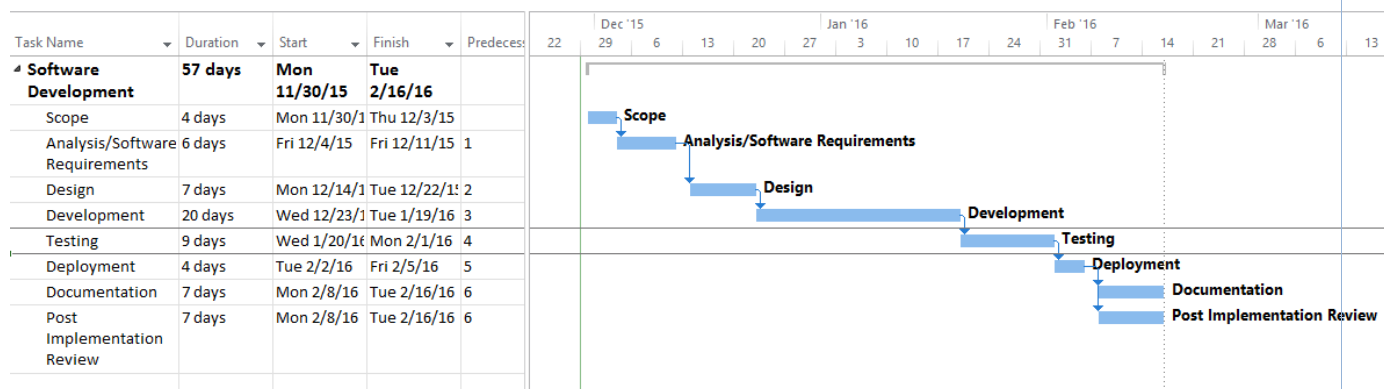


Figure 37 Gantt chart for developed system

## Data and information Collection

Various fact finding methods has been utilized to gather information while preparing Feedback document. Those methods are listed here.

1. Interview

2. Questionnaire
3. Feedback form/Surveys

To collection information from developed system, web application uses web form. Information provided by user through form is stored in database system.

User name  
 The user name field is required.

Email address  
 The email address field is required.

Password  
 The password field is required.

Confirm password  
 The confirm password field is required.

**Register**

Figure 38 Data collection using web form

## **Product deployment**

### **Domain Name**

Domain name is unique name that recognizes the developed web application site Domain name can be bought online from different sources. Initial step of application deployment is to buy a suitable area name for the site. For these take after the strides recorded underneath.

- Visit area domain vender locales, for example, godaddy.com, register.com or domain.com and so forth.
- Check for accessible domain names that matches the prerequisite of site.
- Complete the purchase

United States - English | Sign In | Register | My Account | Cart is empty

GoDaddy  
It's go time

All Products Domains Websites Hosting & SSL Get Found Email & Tools Support

Find a domain Domain Search Bulk Domain Search Transfer Domains New Domain Extensions Domain Privacy

**DOMAIN SEARCH**

Find your perfect name

Enter a domain name  Search Domain

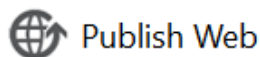
✓ MORE PEOPLE TRUST GoDaddy with their domains than anyone else.  
 ✓ WORLD CLASS 24/7 help building your online business.  
 ✓ BIG SAVINGS over other registrars.

Figure 39 Purchase domain name from Go Daddy

## **Publishing the Application**

Developed system is published using FTP publishing service provided by Visual studio. These procedure were followed.

1. Go to Build menu
2. Select Publish
3. Go to Connection Tab and select FTP method from drop down [Figure 10]
4. Fill the FTP connection Form
5. Next→ Publish



Profile **BrownBird \***

Connection

Settings

Preview

Publish method: File System

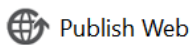
Web Deploy

Web Deploy Package

FTP

Target location: File System

Figure 40 Select FTP method



Profile **BrownBird \***

Connection

Settings

Preview

Publish method: FTP

Server: e.g. ftp://www.contoso.com

Site path: e.g. wwwroot

☒ Passive mode

User name:

Password:

☐ Save password

Destination URL: e.g. http://www.contoso.com

Validate Connection

< Prev Next > Publish Close

Figure 41 FTP connection form

## Maintenance Techniques



To maintain this system technician should follow these important directors in order to perform system maintenances properly.

- To assist users with system operation follow user guideline
- Developed system does not have database backup system, to avoid complications, create database backup using SQL server on daily.
- For some reason if data is lost, restore backed up database using SQL server
- Modify CSS and HTML using Visual studio to modify UI design
- To change Database Location, modify Web.Config.

**Note**

This document provided detailed guidance for support and maintenance. For further support, mail to [support@Bhat-Bhateni.com](mailto:support@Bhat-Bhateni.com)