**INTRODUCTION**

  Reliable and accurate global and domestic transportation of goods is a vital aspect in maintaining a sound and robust economic state.  Everything from wheat to luxury automobiles is transported through cargo containers.  A 1997 Commodity Flow Survey estimated that over $7 Trillion dollars worth of goods traveled via inter modal cargo containers. In this dependence by industry and government on this method of transportation, there is often little ability to accurately track location and status of various containers.  Companies in the transport sector try to find ways of increasing their efficiency and reducing costs.  Customers often demand a tracking service, because they want to know where their shipments are and when to expect them.  Current systems that are in use are limited by the high cost of equipping mobile units or containers, or the limited functionality to meet the needs of the shipping/transport organizations.

Through this online tracking facility you can get real time status information on your shipment. All you have to do is select the cargo, verify or change the prefix of your shipment number, enter the given number, click on the Submit button, and this system will automatically retrieve and display the current status of your cargo

What has been proposed, and is the goal of this project, is to develop a system in which the problems of current systems can be addressed.

Placement Cell is a web application that provides a platform for candidates seeking job and the employers to share their needs. They enter the job details such that job title, vacancies, last date of application, etc. Then they enter the question paper. It includes the questions, the choices and the correct answer. After the exam has been written by the seekers, the result can be viewed such that the top rankers for the particular exam. In addition, the provider can see the resume details of the particular job applicant. The seekers enter into the web site, create their logins and enter the resume details.

Then they may search for the job and apply for the desired job. A job registration id is generated and responded to the seeker. During the exam, he should produce that id so that the corresponding company’s question paper is displayed. There is no time limitation for the attending the exam. After the exam is completed, the result is displayed immediately. He may also view the exam results written earlier. Thus the site acts as the intermediate to both the companies and the job seekers.

The candidates seeking job (referred as job seekers now onwards) can perform following operations:

* Register with the web site.
* Add frequently used searches to their favorites list.

The employers can perform following operations:

* Register with the web site.
* Enter profile of their company.
* Post one or more job postings.
* Add resumes to their favorites list.

**Objective of the project:**

1. Develop online application which should be running 24/7 without interruption
2. The application should be used by multiple user at a time
3. The application should provide the security to the user so that there personal information cannot be stolen by other people which may harm there security
4. The application should be useful to the organization as well as it should be usefull to the potential job seekers
5. The application must be easy to use with simple graphical user interface

**Scope of the Project:**

1. The application involves the uploading the information of the company which in need of the potential candidates who can work in there company. The application has information of the job requirements which involves the specific technical as well as the work experience details.
2. The application revolves around the company information and the candidates details who is seeking job in the company.

**SYSTEM ANALYSIS**

**SYSTEM ANALYSIS**

As in any other system development model, system analysis is the first phase of development in case of Object Modeling too. In this phase, the developer interacts with the user of the system to find out the user requirements and analyses the system to understand the functioning.

Based on this system study, the analyst prepares a model of the desired system. This model is purely based on what the system is required to do. At this stage the implementation details are not taken care of. Only the model of the system is prepared based on the idea that the system is made up of a set of interacting objects. The important elements of the system are emphasized.

**SYSTEM DESIGN**

System Design is the next development stage where the overall architecture of the desired system is decided. The system is organized as a set of sub systems interacting with each other. While designing the system as a set of interacting subsystems, the analyst takes care of specifications as observed in system analysis as well as what is required out of the new system by the end user.

as the basic philosophy of object-oriented method of system analysis is to perceive the system as a set of interacting objects, a bigger system may also be seen as a set of interacting smaller subsystems that in turn are composed of a set of interacting objects. while designing the system, the stress lies on the objects comprising the system and not on the processes being carried out in the system as in the case of traditional waterfall model where the processes form the important part of the system.

**EXISTING SYSTEM:**

The present system that exists is one of manual management that can keep track of information in a limited manner. As the manual management of data requires the users to maintain the registers which in turn is not the feasible solution for any cargo systems.

PROPOSED SYSTEM:

The proposed application is software which is used to maintain the details of the cargo’s, which are used to transport the materials from one source to destination. By using the application one can easily track the materials like the material status ie where the material is, when the material will arrive, what is the cargo details ect.

Advantages:

1. One can track the business details like product to be delivered and on what date it’s supposed to be delivered to the customer. In turn it leads to the attachment of customer and hence business can be extended.
2. The application gives the instant details about the material which are transported using the cargo’s
3. The application is secured and proactive

**Modules:**

1. **Transportation details:** This module will keep the information of transportation where the materials are being transported.
2. **Cargo faculty details:** Cargo facility module will give the details of the cargo facility in different cities.
3. **Cargo type:** This will provide the information about kind of cargo we are using.
4. **Shipment details:** This module provides the shipment details, information like source, destination, cargoID ect
5. **Goods details:** This goods detail contains the details of goods which are being transported by our cargos.
6. **Admin module:** Admin module will take care of administrative work.

**System Requirement Specification:**

Front End: HTML, JAVASCRIPT

Language: ASP.NET, C#.Net

Back end: SQl Server

# Hardware Requirements:

Hard Disk: 80GB

RAM : 1GB

ASP.NET

**FEASIBILITY REPORT**

**FEASIBILITY STUDY**

Analysis and evaluation of a proposed project to determine if it is technically feasible, is feasible within the estimated cost, and will be profitable. Feasibility studies are almost always conducted where large sums are at stake. Also called feasibility analysis.

The main difficulty with this You have probably noticed that the same web page may look different depending on what browser you are using and even what version of the browser. In some cases a web page will not work properly unless you upgrade to the latest version of a particular browser. Likewise a web page may work fine with an older browser but not a newer one.

Now consider the company that deploys several browser-based applications. And then suddenly one of those applications comes out with an upgrade that requires an upgrade to the current standard company browser. If the company decides to upgrade the browser, there is a likely consequence that some features of at lease one of the browser-based applications will not work with the new browser. This leaves the company paralyzed. Do they upgrade the browser and risk breaking some of the other browser-based applications? Do they allocate extensive resources to testing the deployed browser-based applications to see if they will still work with the new browser before deploying it? Or do they stay entrenched on old technology?

Operational

Technical

Economical

Feasibility

**TECHNICAL FEASIBILITY**

Also the developers of browser-based applications have to make sure their user interface works with multiple browsers and versions of those browsers. This means it takes more time to develop and test each new feature, and every time a new version of a browser comes out this problem becomes worse. It also means that it takes more time and is more expensive to implement new features in browser-based systems. Consequently, web client systems will eventually overtake browser-based competitors with either lower price or better functionality or both.

The time slot could be 100 milliseconds. If *job1* takes a total time of 250ms to complete, the Round-Robin scheduler will suspend the job after 100ms and give other jobs their time on the CPU. Once the other jobs have had their equal share (100ms each), *job1* will get another allocation of [CPU](http://en.wikipedia.org/wiki/CPU) time and the cycle will repeat. This process continues until the job finishes and needs no more time on the CPU.

Modelling of actual finish time, while feasible, is computationally intensive. The model needs to be substantially recomputed every time a packet is selected for transmission and every time a new packet arrives into any queue.

**OPERATIONAL FEASIBILITY:**

Lastly there is the issue of performance. Web-based applications work by sending data over the internet or intranet. This mode of communication is relatively slow compared to network speeds and when the database becomes large there will be performance problems with many web-based applications. However, it is easy for a web-client application to solve performance problems caused by data transmission simply by caching data on the client computers. Browser based applications can do some caching too, however, the cached data is generally stored in RAM and lost when the browser is closed. PR-Tracker on the other hand uses a cached database to improve performance. When PR-Tracker is closed the cache remains intact. Consequently, there are many actions PR-Tracker Web Client can do in a split second that may take minutes with a browser-based application or may even be impossible.

**ECONOMICAL FEASIBILITY:**

The proposed system won’t be that much economically costly as this application does not require any hardware part and interfacing with that. Only one web server you need which you will get very easily and economically.

#### 

#### OBJECT ORIENTED METHODOLOGY

Object oriented technology is based on a few simple [concepts](http://www.essortment.com/object-oriented-methodology-43698.html) that, when combined, produce significant improvements in software construction. Unfortunately, the basic concepts of the technology often get lost in the excitement of advanced features and advantageous features. The basic characteristics of the OOM are explained ahead.

**Characteristics of Object Oriented Technology:**

*\* Identity*

*\* Classification*

*\* Polymorphism*

*\** [*Inheritance*](http://www.essortment.com/object-oriented-methodology-43698.html)

**Identity**

The term Object Oriented means that we organize the software as a collection of discrete objects. An object is a software package that contains the related [data](http://www.essortment.com/object-oriented-methodology-43698.html) and the procedures. Although objects can be used for any purpose, they are most frequently used to represent real-world objects such as products, customers and sales orders. The basic idea is to define software objects that can interact with each other just as their real world counterparts do, modeling the way a system works and providing a natural foundation for building systems to manage that business.

**Classification**

In principle, packaging data and procedures together makes perfect sense. In practice, it raises an awkward problem. Suppose we have many objects of the same general type- for [example](http://www.essortment.com/object-oriented-methodology-43698.html) a thousand product objects, each of which could report its current price. Any data these objects contained could easily be unique for each object. Stock number, price, storage dimensions, stock on hand, reorder quantity, and any other values would differ from one product to the next. But the [methods](http://www.essortment.com/object-oriented-methodology-43698.html) for dealing with these data might well be the same. Do we have to copy these methods and duplicate them in every object?

**Polymorphism**

Polymorphism is a Greek word meaning ¡§many forms¡¨. It is used to express the fact that the same message can be sent to many different objects and interpreted in different ways by each object. For example, we could send the message "move" to many different kinds of objects. They would all respond to the same message, but they might do so in very different ways. The move operation will behave differently for a window and differently for a chess piece.

**Inheritance**

Inheritance is the sharing of attributes and operations among classes on a hierarchical relationship. A class can be defined as a generalized form and then it specialized in a subclass. Each subclass inherits all the properties of its superclass and adds its own properties in it. For example, a car and a bicycle are subclasses of a class road vehicle, as they both inherits all the qualities of a road vehicle and add their own properties to it.

Object-Oriented development requires that object-oriented techniques be used during the analysis, and implementation of the system. This methodology asks the analyst to determine what the objects of the system are, how they behave over time or in response to events, and what responsibilities and relationships an object has to other objects. Object-oriented analysis has the analyst look at all the objects in a system, their commonalties, difference, and how the system needs to manipulate the objects.

###### Object Oriented Process

The Object Oriented Methodology of Building Systems takes the objects as the basis. For this, first the system to be developed is observed and analyzed and the requirements are defined as in any other method of system development.

The basic steps of system designing using Object Modeling may be listed as:

System Analysis

System Design

Object Design

Implementation

**Object Design**

In this phase, the details of the system analysis and system design are implemented. The Objects identified in the system design phase are designed. Here the implementation of these objects is decided as the data structures get defined and also the interrelationships between the objects are defined.

Let us here deviate slightly from the design process and understand first a few important terms used in the Object-Oriented Modeling.

As already discussed, Object Oriented Philosophy is very much similar to real world and hence is gaining popularity as the systems here are seen as a set of interacting objects as in the real world. To implement this concept, the process-based structural programming is not used; instead objects are created using data structures. Just as every programming language provides various data types and various variables of that type can be created, similarly, in case of objects certain data types are predefined.

For example, we can define a data type called pen and then create and use several objects of this data type. This concept is known as creating a class.

**Class**: A class is a collection of similar objects. It is a template where certain basic characteristics of a set of objects are defined. The class defines the basic attributes and the operations of the objects of that type. Defining a class does not define any object, but it only creates a template. For objects to be actually created instances of the class are created as per the requirement of the case.

**Abstraction**: Classes are built on the basis of abstraction, where a set of similar objects are observed and their common characteristics are listed. Of all these, the characteristics of concern to the system under observation are picked up and the class definition is made. The attributes of no concern to the system are left out. This is known as abstraction.

The abstraction of an object varies according to its application. For instance, while defining a pen class for a stationery shop, the attributes of concern might be the pen color, ink color, pen type etc., whereas a pen class for a manufacturing firm would be containing the other dimensions of the pen like its diameter, its shape and size etc.

**Inheritance**: Inheritance is another important concept in this regard. This concept is used to apply the idea of reusability of the objects. A new type of class can be defined using a similar existing class with a few new features. For instance, a class vehicle can be defined with the basic functionality of any vehicle and a new class called car can be derived out of it with a few modifications. This would save the developers time and effort as the classes already existing are reused without much change.

Coming back to our development process, in the Object Designing phase of the Development process, the designer decides onto the classes in the system based on these concepts. The designer also decides on whether the classes need to be created from scratch or any existing classes can be used as it is or new classes can be inherited from them.

##### Advantages of Object Oriented Methodology

Object Oriented Methodology closely represents the problem domain. Because of this, it is easier to produce and understand designs.

The objects in the system are immune to requirement changes. Therefore, allows changes more easily.

Object Oriented Methodology designs encourage more re-use. New applications can use the existing modules, thereby reduces the development cost and cycle time.

Object Oriented Methodology approach is more natural. It provides nice structures for thinking and abstracting and leads to modular design.

## Disadvantages of Object Oriented Methodology

As new software development technologies are adopted, an enthusiasm develops as is often the case, and many expectations are brought to the surface. Like an empty glass, developers and managers may fill it up not knowing whether the technology will, in fact, deliver what it is supposed to. Lack of understanding and unrealistic expectations seem to be common denominators. So there is a delay of their proper use. These problems can be minimized, even avoided, by having a clear understanding of the problems and pitfalls that programmers face when adopting these technologies.

# OBJECT MODELING TECHNIQUE (OMT)

The Object Modeling Technique (OMT) software engineering methodology [#!omt!#]   is another well known example of a software engineering methodology. The OMT software engineering methodology deals with object-oriented development in the analysis and design phases.

The analysis phase starts with a problem statement which includes a list of goals and a definitive enumeration of key concepts within a domain. This problem statement is then expanded into three views, or models: an object model, a dynamic model, and a functional model. The object model represents the artifacts of the system. The dynamic model represents the interaction between these artifacts represented as events, states, and transitions. The functional model represents the methods of the system from the perspective of data flow. The analysis phase generates object-model diagrams, state diagrams, event-flow diagrams, and data-flow diagrams. The analysis phase is now complete.

The system design phase follows the analysis phase. Here the overall architecture is established. First the system is organized into subsystems which are then allocated to processes and tasks, taking into account concurrency and collaboration. Then persistent data storage is established along with a strategy to manage shared-global information. Next, boundary situations are examined to help guide trade-off priorities.

The object design phase follows the system design phase. Here the implementation plan is established. Object classes are established along with their algorithms with special attention to the optimization of the path to persistent data. Issues of inheritance, associations, aggregation, and default values are examined.

The OMT software engineering methodology is sequential in the sense that first comes analysis, followed by design. In each phase, a cyclical approach is taken among the smaller steps. The OMT is very much like the Booch methodology where emphasis is placed on the analysis and design phases for initial product delivery. Both the OMT and Booch do not emphasize implementation, testing, or other life cycle stages.

**Omt has proposed three main types of models**:

* ***Object model*** : The object model represents the static and most stable phenomena in the modeled domain. Main concepts are classes and associations, with attributes and operations. Aggregation and generalization (with multiple inheritance) are predefined relationships
* ***Dynamic model*** : The dynamic model represents a state/transition view on the model. Main concepts are states, transitions between states, and events to trigger transitions. Actions can be modeled as occurring within states. Generalization and aggregation (concur-rency) are predefined relationships
* ***Functional model*** : The functional model handles the process perspective of the model, corresponding roughly to data flow diagrams. Main concepts are process, data store, data flow, and actors

**LANGUAGE OVERVIEW**

**ASP.NET**

Open your favorite text editor, Notepad will suffice, and paste the following HTML

code. Save the document as nowhere.html. Next, open the HTML file in your browser

of choice. The following HTML code will set up a perfectly clear HTML form that sends

information into the ether:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

<title>HTML to Nowhere</title>

</head>

<body>

<form>

Please enter your name:<br/>

<input type="text" name="username"/>

<p/>

Press the button to receive all of your hopes and dreams:<br/>

<input type="submit" value="Dream Button" />

</form>

</body>

</html>

When you test the code after you enter your name and click the Dream Button (a thinly

disguised submit button), your output will look just like Figure 1-1.

Without some kind of mechanism to capture and store the information or pass it on

to somewhere else, all you have is an HTML white elephant. What ASP.NET 3.5 offers

is a way to store, retrieve, and process the information. Other than storing cookies on the

visitor’s computer, you can’t do too much with HTML when it comes to controlling the

state of your data.

Your Browser Is a Thin Client

Your browser’s main job is requesting pages from the server and displaying what the

server has processed. Using HTTP as a transfer protocol, your browser parses (interprets)

the HTML code it gets from the server, but otherwise does little processing. Because your

HTTP client (browser) leaves most of the processing work to the server, it is considered a

thin client. There’s nothing wrong with a thin client, and it can process JavaScript. With

plug-ins, which virtually all browsers have built in, a thin client can also process certain kinds of files such as SWF (compiled Adobe Flash files), Java Applets, ActiveX Controls, as well as other files requiring that the browser have compatible plug-ins. For the most part, though, the thin client model is one where the server does the processing, and your browser’s job is to display the contents it gets from the server.

A Protocol Without a Country: Stateless HTTP

In addition to being a thin client, your HTTP client browser is stateless. As soon as a

web page reaches your computer, the connection between your browser and the server is broken. The browser does not remember the last page—it does not hold state. As soon as the next page arrives, it does not remember the last page.

You may be thinking that your cache holds lots of previous pages and that your

browser’s history feature remembers previous pages. That’s not what retaining state

means. You cannot use the data and information in your cache or the browser’s history as states to use with the current page in memory. What is in active memory is the web page that the server has sent; the state of the previous page is not there. As soon as you load a page, the previous page is kicked out, and the new page is placed there. Just as the information you place in a text input box is sent to silicon oblivion when you click the Submit button, knowledge of the previous page (its state) is gone when the new page arrives. A well-organized web site may appear to maintain state as the links on pages connect to a set of related pages, but that is an illusion that the web designer has crafted by good planning.

**ASP.NET 3.5 as an Alternative to CGI**

Microsoft’s alternative to a Common Gateway Interface (CGI) is ASP.NET, now in

version 3.5. As a unified web platform, ASP.NET provides what you need to develop

applications that hold state and use the information that you put into the HTML form.

Instead of sending form information into a vacuum when you click a submit button, your data goes where it can be stored temporarily or permanently. Usually, when we think of saving state, we imagine writing the data to a storage device like a hard drive. Using a word processor, every time you save your file, you save its state. Using ASP.NET, you can do the same thing with information from anyone who uses your web application. This allows you to build applications where the information entered can be stored for use with either the next HTTP request or with a whole set of data entered by users all over the world—that’s quite a feat compared with saving state in your word processor file. ASP.NET’s state-management facilities provide you with the tools that you need to control state. You do not necessarily want to save all states of a web page, but you certainly want to save the state of data entered by users and perhaps the URL of a page. Having state management allows you to do this.

Microsoft’s web server, Internet Information Services (IIS), uses the Internet Server

API (ISAPI) to make function calls instead of using CGI scripts. By using ISAPI,

developers can create web-based applications that execute faster and have greater

extensibility than CGI, among other advantages. At the lowest level, ASP.NET interfaces with IIS through an ISAPI extension. However, this book focuses on the high-level interaction with ISAPI in the form of ASP.NET and code written in C# (pronounced “C sharp”) that use ASP.NET. So rather than having to deal with the fine-grained, low-level communications, ASP.NET allows

you to write your scripts in C#. Another way of looking at ASP.NET is as a requestprocessing engine that takes incoming requests and sends them to a point where you can attach your C# script to process the request.

So while we are not going to spend time dwelling on the low-level operations, you can rest assured that those operations are handled in an efficient manner. Using managed code, the Microsoft name for code that executes under the management of the .NET framework,

an application is executed by a virtual machine rather than by your own processor. Both C# and Visual Basic.NET (VB.NET) are languages for creating managed code that is efficiently run in the .NET environment.

**From Client Side to Server Side**

As you saw in Figure 1-2, all that the web browser does is make requests to the server and receive web pages in the form of HTML. The browser takes the HTML and constructs a page for viewing on your browser. For the most part, that’s what will continue to occur when you adopt ASP.NET. The main difference is that by writing and executing serverside code, you can generate HTML that effectively handles dynamic states so that you can use and reuse a given state. Figure 1-3 shows the general flow when an ASPX file on a Microsoft server receives a call from the client.

In looking for an example, we need look no further than the original example of the

HTML page with the form. With a server-side program to catch the data that is sent to

the server, lots of processes on the server are possible. Suppose the user enters the name,

Willie B. Goode. The server-side file can use the property name, username, to extract a

value. In this case, it would extract the name Willie B. Goode and do something with it. It

might run a SQL script to store the name in a database, compare it with a password, or it

could pass information about Willie B. Goode back to the browser in HTML.

To get an idea of the differences and similarities between HTML and ASP.NET, we

can take the HTML form and add a calculated response from the server. In this next

example, you will see two scripts. One is the ASP.NET script and the other is the C# code

using a code behind file to serve as the event engine for the application. The form tag

includes a name and the code

runat="server"

What distinguishes the web hobbyist from the professional developer is the ability to store and retrieve data from a server over the Internet. Some readers may have made that step with PHP or Perl, and for them the journey has already begun. For others, ASP.NET 3.5 is just the next step in the growth and development of ASP.NET. For them, much will be familiar and some will be very new. If the transition is from ASP.NET using Visual Basic to ASP.NET with C#, then you will find even more new, and the transition to C# is going to be easier than many imagine. Still others are making the first step into the realm of server-side programming, so just about everything about ASP.NET 3.5 and C# 3.0 is new.For those using ASP.NET for the first time, the good news is that the transition to the server side could hardly be easier. Everything in this book is set up to learn ASP.NET 3.5 and C# 3.0 using Visual Studio 2008. Using the tools built into Visual Studio 2008, you will quickly learn that most of what needs to be done can be accomplished by dragging controls into a visual editor. And with the “code behind” method, C# is added in a separate file, so when editing code, you will see a clear separation—the C# 3.0 code comes “behind” the ASP.NET 3.5 code. For the most part, though, you need only a minimum of coding skills in either ASP.NET, which feels a lot like HTML, or C#, which has many features recognizable from JavaScript. However, make no mistake about it, C# 3.0 is a full-fledged coding language with the power of any good object-oriented programming (OOP) language. Most of the C# you need is simply working with functions and subroutines called by ASP.NET events. As a result, learning C# is quite simple, and you’ll get a lot of help from Visual Studio 2008 coding tips and built-in IntelliSense. However, if you wish not to use Visual Studio 2008, all of the code for both the ASP.NET and C# is provided as well. (You can program it all using Notepad if you like!) You can find a free Express version of Visual Studio 2008 at www.micr osoft.com/express/download/, and it has much of the functionality of the full-fledged version. Likewise, you will find an Express Edition of SQL Server 2005, and with them both you can learn ASP.NET 3.5 and C# 3.0 on a budget while using a powerful development tool.

The main use of ASP.NET in conjunction with C# and Structured Query Language (SQL) is to store user input in a database and get it back again. If you’ve ever dealt with forms in HTML, you may know how frustrating it is to build a web site with data entry that cannot be stored. In fact, without some kind of storage facility and the tools required to place the data in storage, HTML forms have very limited use. However, not only can you use plain HTML forms, ASP.NET 3.5 has some web controls that are almost identical to HTML forms except they have far more functionality. In no time, you’ll be working with tables and databases to store, fetch, change, and delete a wide variety of data. In addition to the HTML-like controls, a whole other set of web controls is available to work with data and create objects that HTML cannot. For example, if your web site needs a calendar, all you have to do with ASP.NET 3.5 is add a calendar control by dragging it from the Toolbox and into the editor. Then you can use the Calendar properties to perform other functions, like reminding you of your upcoming anniversary. (Getting that right can be worth the price of this book!)

All in all, my hope is that you will enjoy working with ASP.NET 3.5 and C# 3.0 as much as I have. Not only is it very easy to develop rich interactive web sites using ASP.NET, but in conjunction with C# 3.0, the toolset is also a very powerful one. With it, you have the best of all worlds.

In this tutorial we will explore this wonderful free tool along with its advantages especially for new web developers. After downloading and installing the Visual Web Developer 2005 express edition form this link, run it and you will get this start up page as shown in figure1 below.



As you see, it nearly seems like Visual Studio 2005. If you clicked the "File" menu you will see the menu item "New Web Site ...", you can not create other types of applications except web applications. As Visual Studio you will have a toolbox, a properties window, a solution explorer window, a debugging menu, and so on. All these items give you the ability to control your application and visually designing it in a simple WYSIWYG drag and drop interface.

**Visually Design Your Data**

Visual Web Developer provides a set of controls dialogs and wizards to help you in communicating with a certain database or data source. First of all you can install SQL Server 2005 Express Edition while you are installing the Visual Web Developer. SQL Server Express Edition is a lightweight database server designed for smaller load web sites. Now let's browse what Visual Web Developer provides for ease of using manipulating and requesting data from a data source.

Data Source Configuration Wizard provides a step by step wizard to help you connecting to a data source, whether this source is an SQL database, an Access database, an XML database, and more. The wizard dynamically pulls the tables along with their fields from the data source allowing you to select the fields you want to display.

  
Figure 6 - Data Source Configuration Wizard

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language. The ASP.NET SOAP extension framework allows ASP.NET components to process SOAP messages.

**CHARECTERISTIC:**

### Pages

ASP.NET web pages or webpage, known officially as "web forms", are the main building block for application development.[[8]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-macdonald63-7) Web forms are contained in files with an ".aspx" extension; these files typically contain static ([X](http://en.wikipedia.org/wiki/XHTML)) [HTML](http://en.wikipedia.org/wiki/HTML) markup, as well as markup defining server-side Web Controls and User Controls where the developers place all the required static and dynamic content for the web page. Additionally, dynamic code which runs on the server can be placed in a page within a block <% -- dynamic code -- %>, which is similar to other web development technologies such as [PHP](http://en.wikipedia.org/wiki/PHP), [JSP](http://en.wikipedia.org/wiki/JavaServer_Pages), and [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages). With ASP.NET Framework 2.0, Microsoft introduced a new code-behind model which allows static text to remain on the .aspx page, while dynamic code remains in an .aspx.vb or .aspx.cs file (depending on the programming language used).[[9]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-QuickstartAspNet-8)

### Code-behind model

Microsoft recommends dealing with dynamic program code by using the code-behind model, which places this code in a separate file or in a specially designated script tag. Code-behind files typically have names like *MyPage.aspx.cs* or *MyPage.aspx.vb* while the page file is*MyPage.aspx* (same filename as the page file (ASPX), but with the final extension denoting the page language). This practice is automatic in[Microsoft Visual Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio) and other [IDEs](http://en.wikipedia.org/wiki/Integrated_development_environment). When using this style of programming, the developer writes code to respond to different events, like the page being loaded, or a control being clicked, rather than a procedural walk through of the document.

*ASP.NET'*s code-behind model marks a departure from Classic ASP in that it encourages developers to build applications with [separation of presentation and content](http://en.wikipedia.org/wiki/Separation_of_presentation_and_content) in mind. In theory, this would allow a web designer, for example, to focus on the design markup with less potential for disturbing the programming code that drives it. This is similar to the separation of the controller from the view in [Model–View–Controller](http://en.wikipedia.org/wiki/Model%E2%80%93View%E2%80%93Controller)(MVC) frameworks.

### Directives

A directive is special instructions on how ASP.NET should process the page.[[10]](http://en.wikipedia.org/wiki/ASP.NET#cite_note-msdnMicrosoftSyntax-9) The most common directive is <%@ Page %> which can specify many things, such as which programming language is used for the server-side code.

#### Examples

<%@ Page Language="C#" %>

<!DOCTYPE html PUBLIC "---//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">

protected void Page\_Load(object sender, EventArgs e)

{

lbl1.Text = DateTime.Now.ToLongTimeString();

}

</script>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title>Sample page</title>

</head>

<body>

<form id="form1" runat="server">

<div>

The current time is: <asp:Label runat="server" id="lbl1" />

</div>

</form>

</body>

</html>

The above page renders with the Text "The current time is: " and the <asp:Label> Text is set with the current time, upon render.

**C#.NET**

**Basic .NET or C#.**

Most of the samples in this book were written by hand, without the help of Visual Studio .NET. That’s not a knock on Visual Studio .NET; it’s evidence of my belief that learning is best accomplished by coding and not by having someone else code for you. Once you understand what goes into a Windows form or a Web form or a Web service, you’ll find Visual Studio .NET an able partner in helping to create them. Writing applications the old-fashioned way first will increase your depth of understanding and better prepare you to work in an environment in which tools shoulder part of the load for you.

C# (see section on name, pronunciation) is an object-oriented programming language developed by Microsoft as part of the .NET initiative and later approved as a standard by ECMA and ISO.

Anders Hejlsberg leads development of the C# language, which has a procedural, object-oriented syntax based on C++ and includes aspects of several other programming languages (most notably Delphi and Java) with a particular emphasis on simplification.

C# principal designer and lead architect at Microsoft is Anders Hejlsberg. His previous experience in programming language and framework design (Visual J++, Borland Delphi, and Turbo Pascal) can be readily seen in the syntax of the C# language, as well as throughout the Common Language Runtime (CLR) core.

C# is intended to be a simple, modern, general-purpose, object-oriented programming language.

**.NET Framework 1.1**

This is the first major .NET Framework upgrade. It is available on its own as a redistributable package or in a software development kit, and was published April 3, 2003. It is also part of the second release of Microsoft Visual Studio .NET (released as Visual Studio .NET 2003). This is the first version of the .NET Framework to be included as part of the Windows operating system, shipping with Windows Server 2003.

**Why C#?**

Many people believed that there was no need for a new programming language. Java, C++, Perl, Microsoft Visual Basic, and other existing languages were believed to offer all the functionality needed.

C# is a language derived from C and C++, but it was created from the ground up. Microsoft started with what worked in C and C++ and included new features that would make these languages easier to use. Many of these features are very similar to what can be found in Java. Ultimately, Microsoft had a number of objectives when building the language. These objectives can be summarized in the claims Microsoft makes about C#:

C# is simple.

C# is modern.

C# is object-oriented.

 In addition to Microsoft's reasons, there are other reasons to use C#:

 C# is powerful and flexible.

C# is a language of few words.

C# is modular.

C# will be popular.

C# Is Simple

C# removes some of the complexities of languages such as Java and C++, including the removal of macros, templates (in the form of Generic in C# 2.0), multiple inheritance, and virtual base classes. These are all areas that cause either confusion or potential problems for C++ developers.

**What is C#?**

C# (pronounced C sharp) is a new programming language introduced by Microsoft with the Microsoft .NET framework. It was first created in the late 1990's as part of Microsoft’s.NET strategy. It is a new language free of backward compatibility and a bunch of new, exciting and promising features.

It is an Object Oriented Programming language, which at its core, and has similarities with Java, C++ and VB. The main brains behind C# were Anders Hejlsberg and Scott Wiltamuth. However, many other personalities like Rob Howard, Scott Guthrie were also involved behind C# and the .NET Framework.

 In fact, C# combines the power & efficiency of C++, simple & clean Object Oriented design of Java, and code simplification of Visual Basic. Like Java, C# also does not allow multiple inheritance and use of pointers (in safe and managed code) while it does provide garbage memory collection at runtime, type and memory access checking.

But, contrary to java, C# keeps the different useful concepts of C++ like operator overloading, enumerations, pre-processor directives, pointers (in unmanaged and un-safe code), function pointers (in the form of delegates), also promises to have template support (with the name of generics) in next versions. Like VB it also supports the concepts of properties (context sensitive accessor to fields).

In addition to this, C# comes up with some new/exciting features like reflections, attributes, marshalling, remoting, threads, streams, data access with ADO.NET, etc. C# programming language is designed from the scratch keeping in mind the Microsoft.Net environment. MS.Net (and thus C#) programs runs on top of the Common Language Runtime (CLR), which provides the runtime support to them.

C# has been standardized by ECMA (European Computer Manufactures Association)

Microsoft originally released beta versions of .NET framework to get feedback about the product from developers and customers. Based on the feedback, received from them for the first beta, Microsoft released second beta in early 2001. Finally, they released final versions of C# under the name, .NET Framework 1.0 in 2002 and that of 1.1 in 2003.

With each release, the product shipped with improved features for developing secured and scalable applications along with high quality documentation. Like C++ and Java, C# is an object oriented programming language. In fact all .NET languages are object oriented since they all have the .NET Framework built into them.

**HTML**

Learn basic HTML to create your Web site. Basic HTML is not hard to learn. Learning HTML is probably the single most important thing you'll ever do if you want to create your own personal Web site. You must [learn HTML](http://personalweb.about.com/library/glossary/bldef-html.htm) to design good pages because it is the language in which Web site on the Internet are based on.

Web sites are a great way to show off something in your life. Basic HTML will let you show the world whatever it is you want to show them on your Web site. Adding colors, changing text size and including pictures on your Web site are just a few of the things you can do when you learn basic HTML.

To learn basic HTML you must keep in mind that it's simply a series of letters that are abbreviations of what they actually stand for. For example, H1 stands for a heading for a paragraph that is the first of five sizes and BR is a line break.

One important thing to remember, while you learn basic HTML, is that on a Web page the HTML tags must come in a certain order and most HTML tags must have a start and end tag for the browser to recognize the command. An end tag is just simply the same as the start tag except that it starts with the **/** symbol. A heading would look like this**<H1>Heading Here</H1>**. There is a start tag,**H1**, the heading, and an end tag, **/H1**.

Learn basic HTML the right way and remember that the tags must come in a certain order. The basic structure of a web page is as follows:

**<HTML>**

**<HEAD>** Here you can put important document information such as frames, language, and special instructions.

**<TITLE>**Title of your page.**</TITLE>**

**</HEAD>**

**<BODY>** Put your story, pictures, links, and everything else here.

**<H1>**Heading of your paragraph.**</H1>**

**<P>**This is where you enter the text of your document.**</P>**

This is how to write a link:  
**<A HREF="**http://www.nameofpage.com**">**Title or what you want to say.**</A>**

**</BODY>**

**</HTML>**

You can also enter a link into the middle of a sentence. If I wanted to say "[Microsoft](http://www.microsoft.com/en/us/default.aspx)has some wonderful demos." It would look like this:

**<A HREF="**http://www.microsoft.com/en/us/default.aspx**">**Microsoft **</A>**has some wonderful demos to help you learn basic HTML.

You not only can create links to other pages but you can also create links to another place on the same page. If I wanted you to go back to the beginning of this article I would say something like "[Go Back](http://personalweb.about.com/library/weekly/aa052498.htm#Basic)" and by clicking on the hyperlink you would go back to the beginning. This one is a little more difficult because there are two parts to it. First of all you create your link:

**<A HREF="**nameofdocument**#There">**Go Back**</A>**

The word "There" is the word I want to take you back to so now I must go to that word and create and anchor so the link I just created knows where to go:

**<A NAME="**There**">There</A>**

When you learn basic HTML you can write your Web site in the text editor that comes with Windows, programs such as [NoteTab](http://www.notetab.com/) and [Arachnophilia](http://www.arachnoid.com/arachnophilia/), or one that comes with your web browser. Either way if you just follow the rules from when you learned basic HTML everything will turn out just fine.

The CSS visual formatting model is a way of laying out documents on visual media (such as a computer screen or a printed page). By all accounts, it is a very simple formatting model, and the publishing world has seen much more complicated ones. Still, it is not something that is simple to explain or grasp in a few words.

Various CSS properties are used to control how a document will appear in the context of this formatting model. You have seen most of them in the examples used so far, but I haven't explained their usage because I have to explain the formatting model first.

As I said, the formatting model is a pretty complicated affair. As if that wasn't enough, though, the implementations that exist, most notably in Microsoft Internet Explorer and Netscape Navigator, are respectably problematic and horrible. Explorer does a pretty good job of sticking to the model, and will behave sensibly with a few exceptions that can be dealt with simply. Navigator, on the other hand, has its own concept of what the various properties should do.

Instead of going through the theory and observing the bugs in Explorer and Navigator along the way, as I have done in the tutorials so far, I will explain the theory, that is how the CSS specification explains formatting should be done, in this tutorial, and deal with the implementations in the next one. I hope this makes the whole process easier to understand.

## What Is a Tag?

A tag is a method of formatting HTML documents.

With tags you can create italic or bold characters, make things blink, and can control the color and size of the lettering.Tags can be "nested". This means that you can make something bold and italic and green and blinking by simply surrounding the previous tag with the next.

Tags can be used to insert pictures and graphics.

Tags can be used to create bulleted lists like this one, or numbered lists.

Tags look something like this: <b> chosen text </b> .

All tags use the < and > (less-than and greater-than symbols) to signal the browser. These are located above the comma and period keys.

Within a tag, capitalization doesn't matter. <FONT> is the same as <font> or <Font> . I usually don't bother capitalizing when I use tags. Below, I have used capital letters when I felt that the number "1" and the letter "l" might be confused.

It is essential to always close the tags! If not, the formatting will contaminate everything that follows it. But don't panic, closing tags is very easy, and even the most experienced surfers sometimes forget.

**SQL SERVER 2008**

**Structured Query Language (SQL)**

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational databases; the most common is SQL. Standards for SQL have been defined by both the American National Standards Institute (ANSI) and the International Standards Organization (ISO). Most modern DBMS products support the Entry Level of SQL-92, the latest SQL standard (published in 1992).

**Client/Server Architecture**

Microsoft® SQL Server™ is designed to work effectively in a number of environments:

As a two-tier or multitier client/server database system

As a desktop database system

Client/Server Database Systems

Client/server systems are constructed so that the database can reside on a central computer, known as a [server](mk:@MSITStore:D:\D%20Drive\e-books1\sqlbol.chm::/html/gloss01.htm#_server_gloss01), and be shared among several users. Users access the server through a client or server application:

In a two-tier client/server system, users run an application on their local computer, known as a [client](mk:@MSITStore:D:\D%20Drive\e-books1\sqlbol.chm::/html/gloss01.htm#_client_gloss01), that connects over a network to the server running SQL Server. The client application runs both business logic and the code to display output to the user, and is also known as a thick client.

In a multitier client/server system, the client application logic is run in two locations:

The thin client is run on the user’s local computer and is focused on displaying results to the user. The business logic is located in server applications running on a server. Thin clients request functions from the server application, which is itself a multithreaded application capable of working with many concurrent users. The server application is the one that opens connections to the database server and can be running on the same server as the database, or it can connect across the network to a separate server operating as a database server. This is a typical scenario for an Internet application. For example, a server application can run on a Microsoft Internet Information Services (IIS) and service thousands of thin clients running on the Internet or an intranet. The server application uses a pool of connections to communicate with a copy of SQL Server. SQL Server can be installed on the same computer as IIS, or it can be installed on a separate server in the network. Having data stored and managed in a central location offers several advantages:

Each data item is stored in a central location where all users can work with it. Separate copies of the item are not stored on each client, which eliminates problems with users having to ensure they are all working with the same information. Business and security rules can be defined one time on the server and enforced equally among all users.

This can be done in a database through the use of constraints, stored procedures, and triggers. It can also be done in a server application.

A relational database server optimizes network traffic by returning only the data an application needs. For example, if an application working with a file server needs to display a list of the names of sales representatives in Oregon, it must retrieve the entire employee file. If the application is working with a relational database server, it sends this command:

SELECT first\_name, last\_name

FROM employees

WHERE emp\_title = 'Sales Representative'

  AND emp\_state = 'OR'

The relational database sends back only the names of the sales representatives in Oregon, not all of the information about all employees.

SQL is the acronym for Structured Query Language. This language is used for developing database related applications. Before developing any application of database it is necessary to understand what is Database?

**Database:**

A database can be defined as the collection of tables of relative Application. For example if you are developing an application for maintaining the information of an university then you have to create database with name STUDENT after words you can create any number of tables related to that Student.

Now a table is nothing but the combination of Rows & Columns. The columns are known as the Attributes or Properties of the table and Columns are known as Tuples, which are the actual values of the respective column headers.

Attributes

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Tuples Structure of a Table

Fig. The structure of a Table

**QUERIES**

The queries can be defined as statements, which can be used to develop the database related applications. There are different queries in SQL some major queries are explained below.

CREATE DATABASE Database\_Name

This query will create a database with the name given in place of Database\_Name.

Ex: CREATE DATABASE Student

Now you have to enter your database by using the following query.

USE Database\_Name

Ex: USE Student

Now developer can create the tables related to university in this database. To create the tables the following query can be used.

CREATE TABLE Table\_Name (Column\_Name DATATYPE)

Here Table\_Name and Column\_Name are any valid variable names and datatype should any of the following.

Integer-> Represented as int, and can be used if the column is going to hold whole numbers.

Real Numbers-> Represented as float, and can be used if the column is going to hold fractional numbers.

Characters-> Represented as varchar, and can be used if the column is going to hold characters or strings. If you are mentioning the datatype of a column as varchar then it also necessary to mention the space within the bracket, which indicates how many characters a column should store.

Ex: CREATE TABLE student (STD\_No int, Emp\_Name varchar (10), marks float)

After the creation of tables we can insert the values in that table by using the following query.

INSERT INTO Table\_Name VALUES (Values)

If the datatype of a column is int or float then you can write the Numbers directly but if datatype is varchar then the values should be enclosed within the single quotations

Ex: INSERT INTO student (1,’R.R. Sharma’, 20)

Apart from above mentioned queries SQL has many more queries as

Mentioned below.

Update -> Used to update the contents of a table

Alter -> To alter the table i.e., You can add the columns or remove the columns

Drop -> to drop tables or databases

Delete -> to delete contents of the table.

**SYSTEM DESIGN**

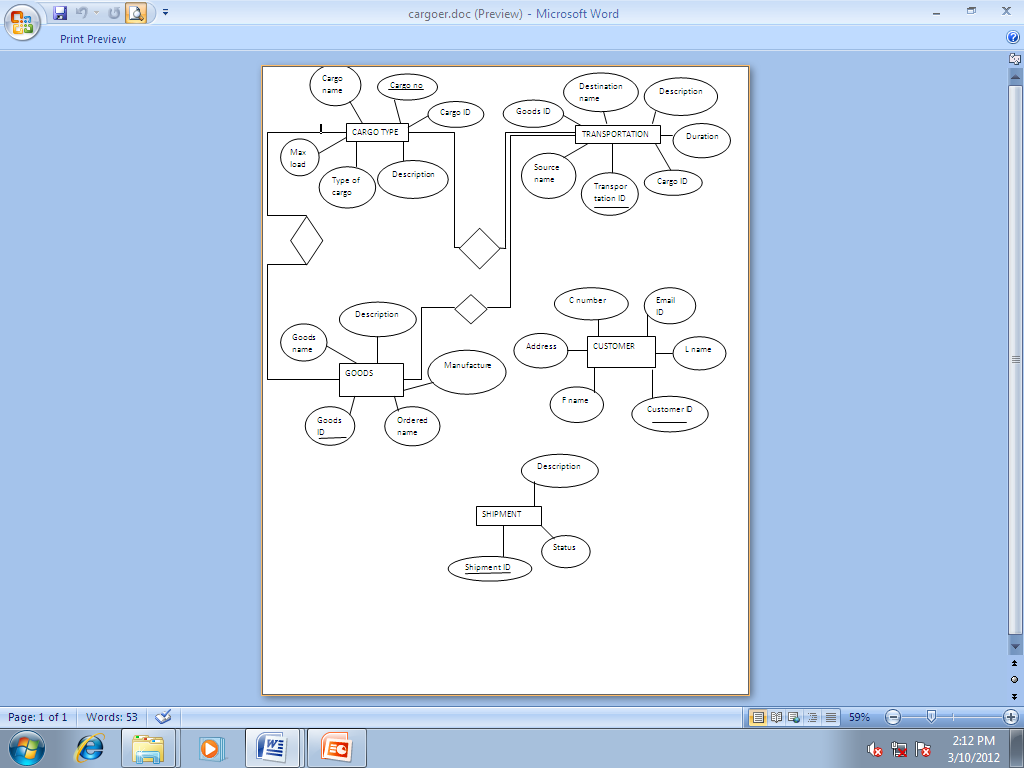
**Data Base Design**

**E R DIAGRAM**

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system). DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Visualization) of [data processing](http://en.wikipedia.org/wiki/Data_processing) (structured design).

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

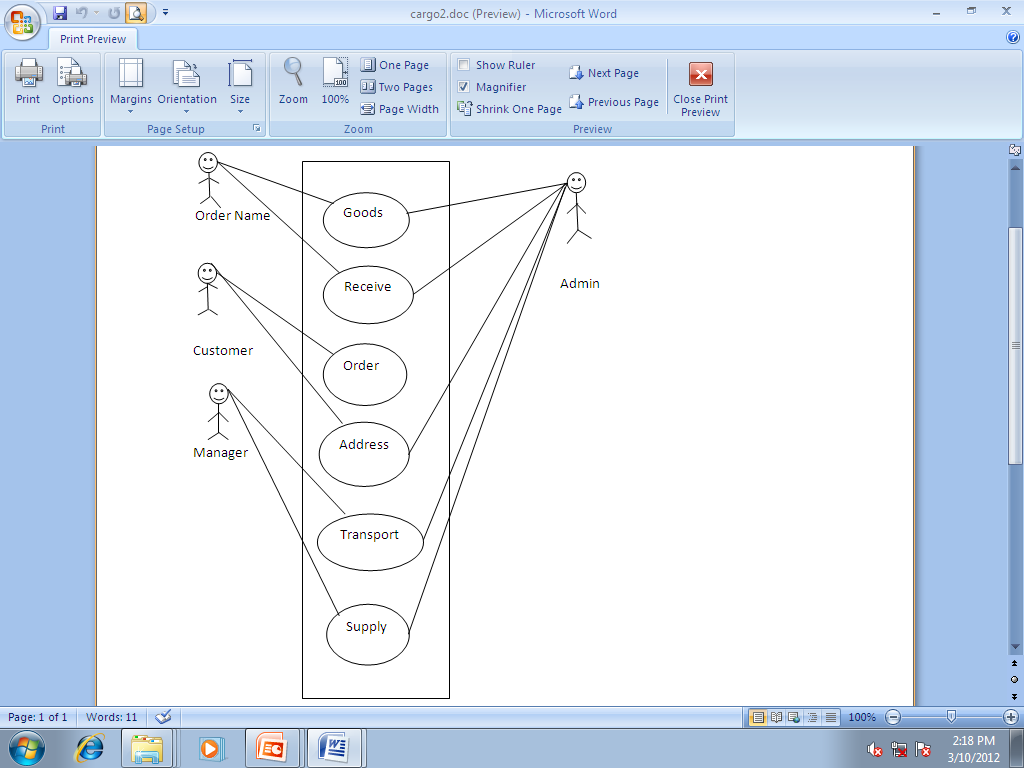
A DFD provides no information about the the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a [flowchart](http://en.wikipedia.org/wiki/Flowchart), which shows the flow of control through an algorithm, allowing a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



DFD:

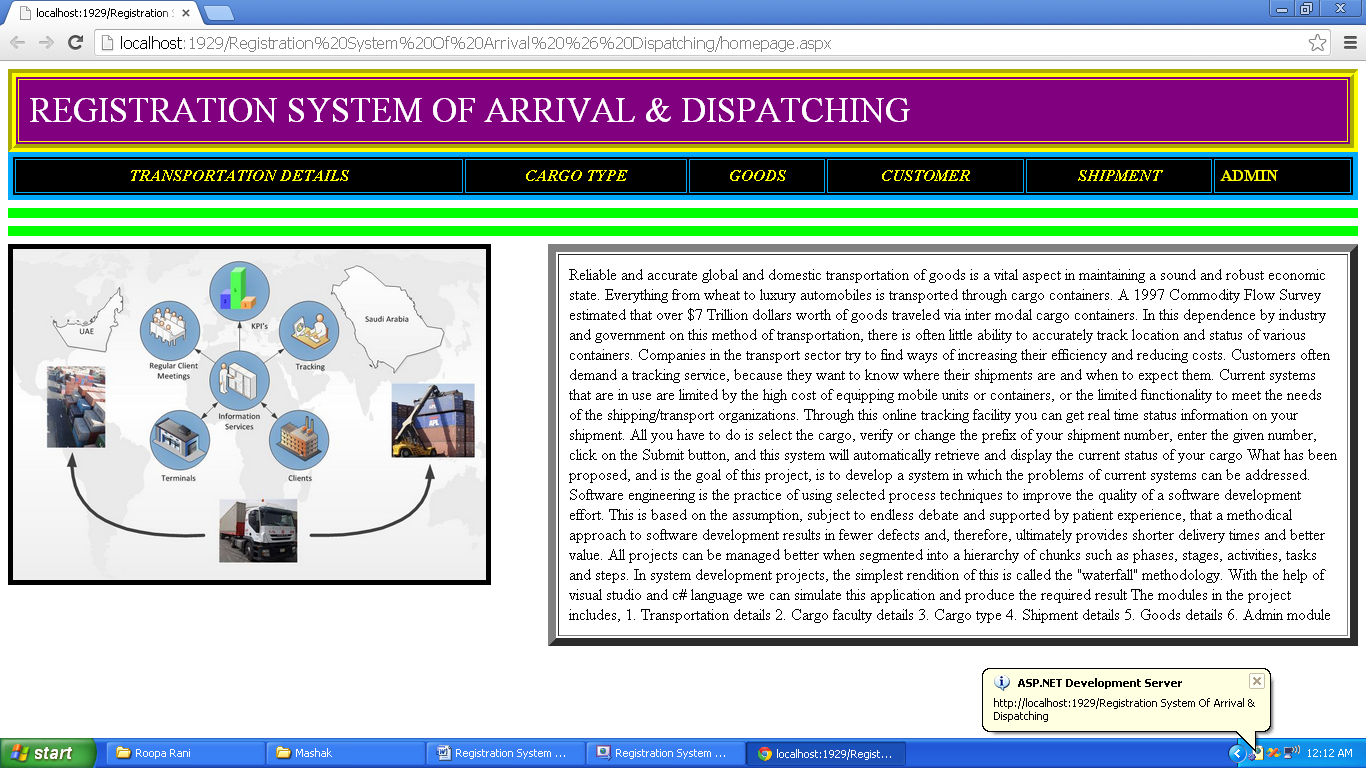


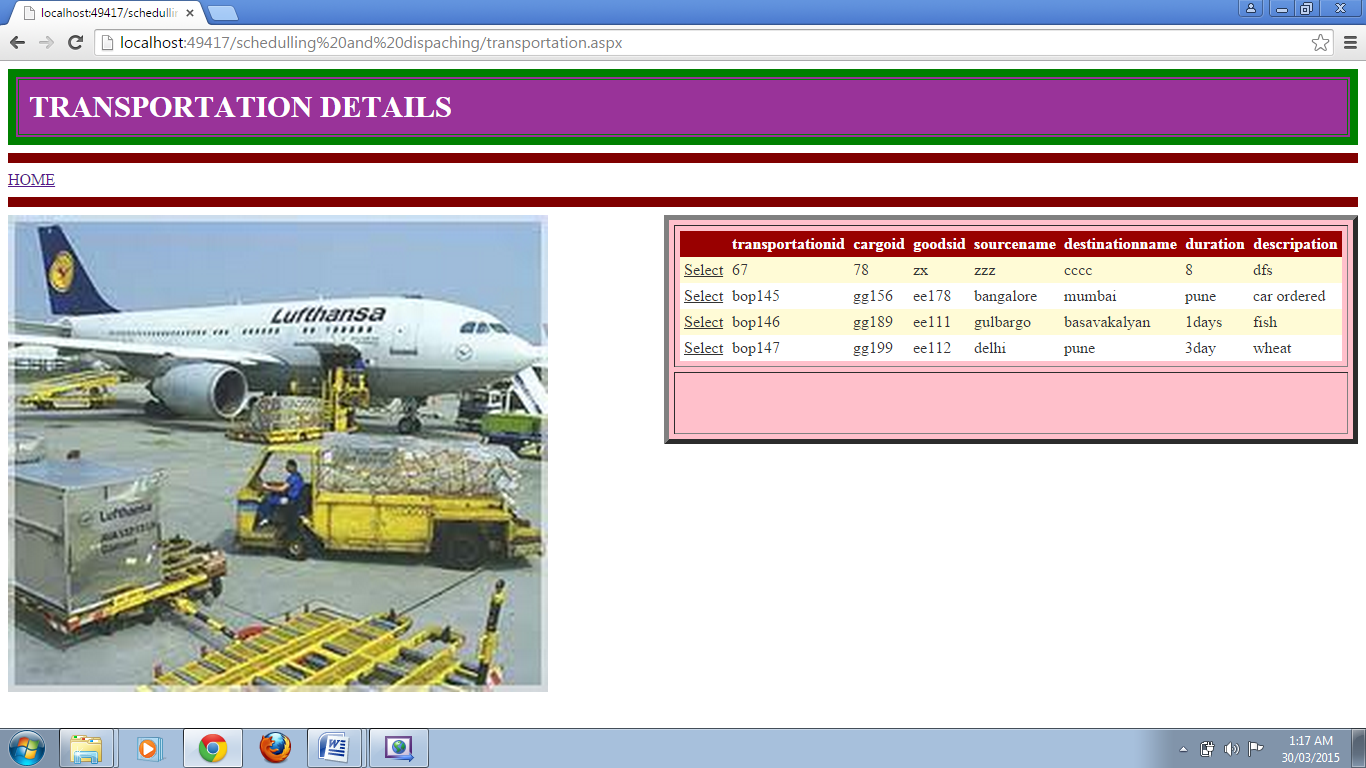
USER CASE DIAGRAM:



**OUTPUT**

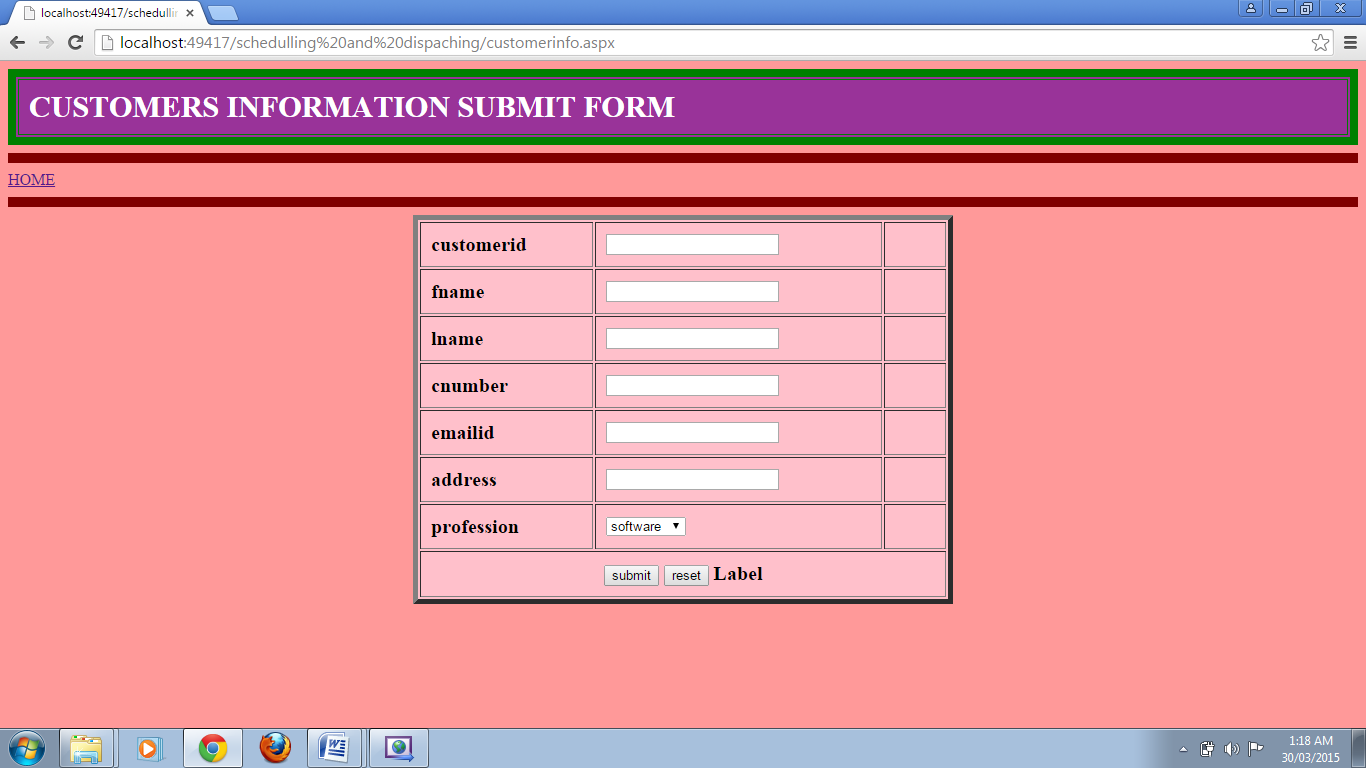
**HOME PAGE**

****

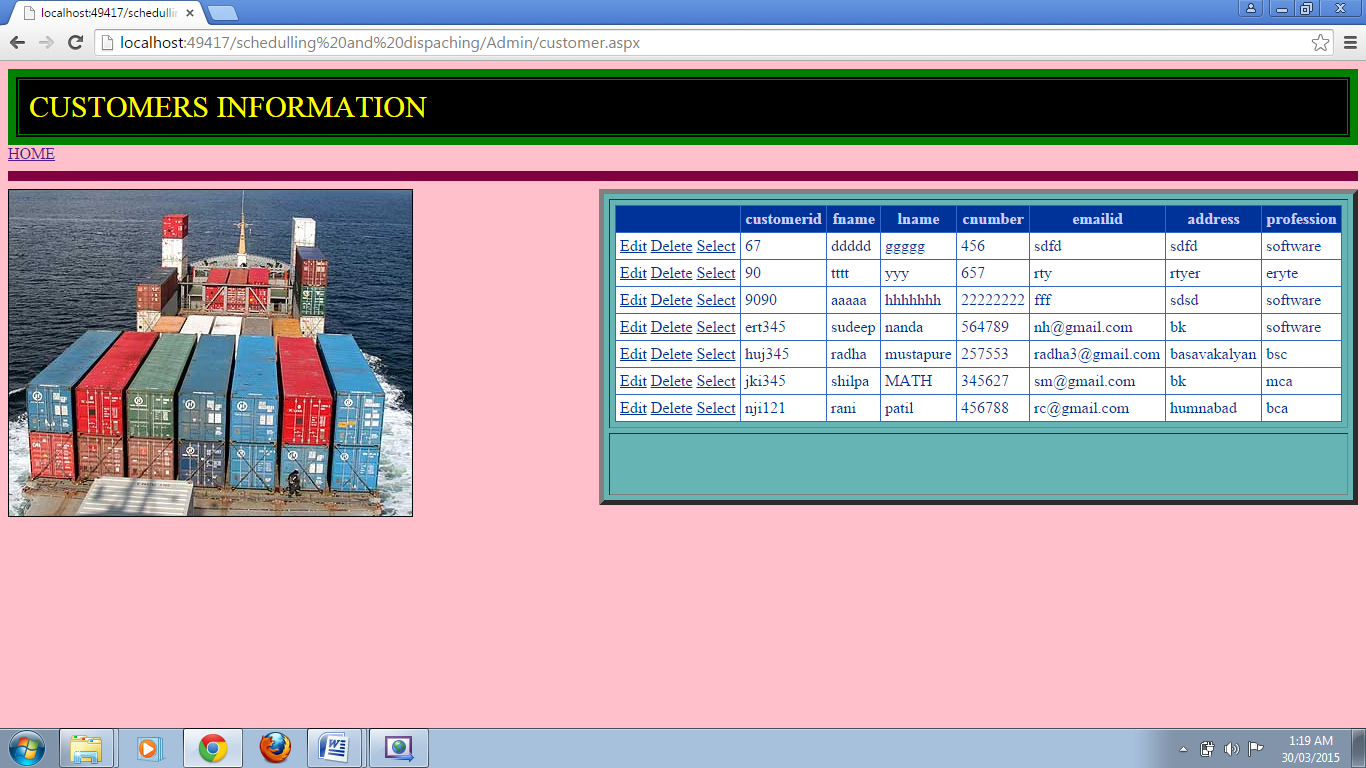
****

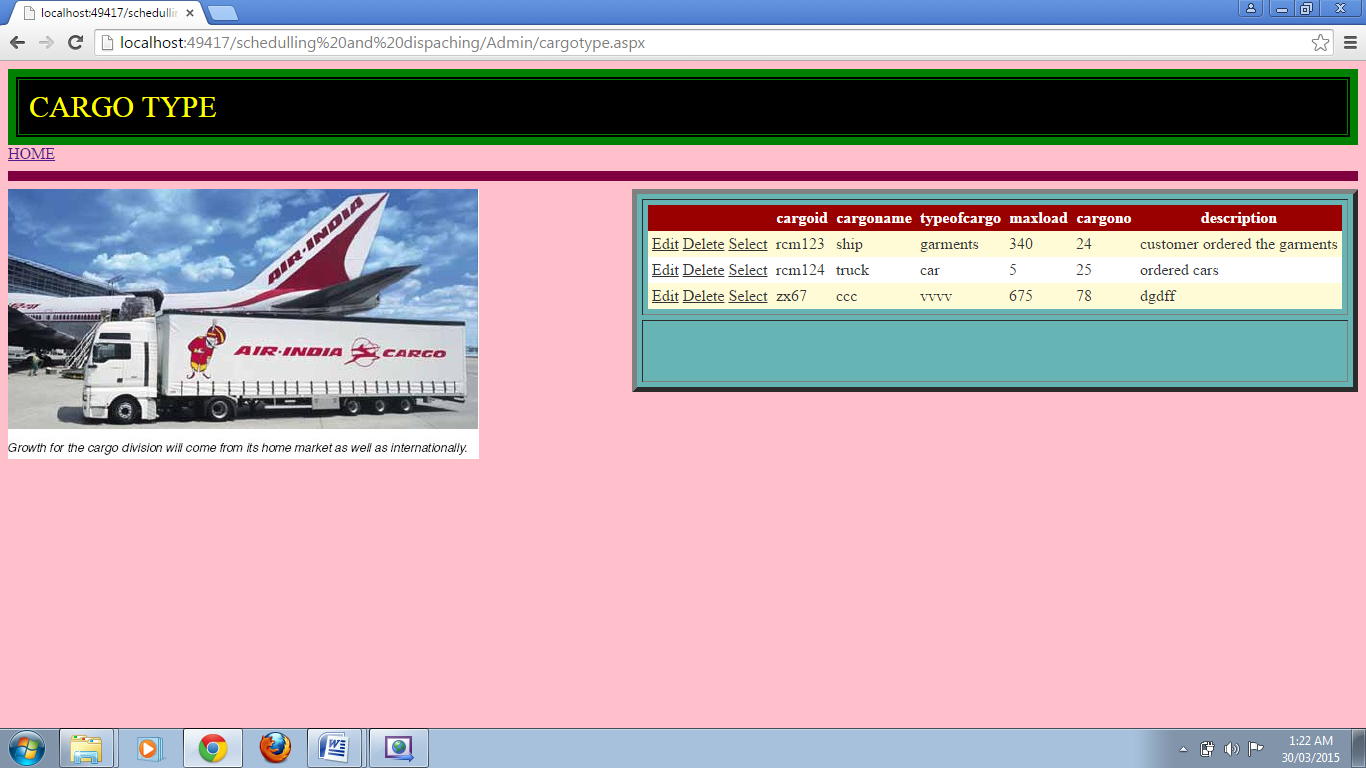
****

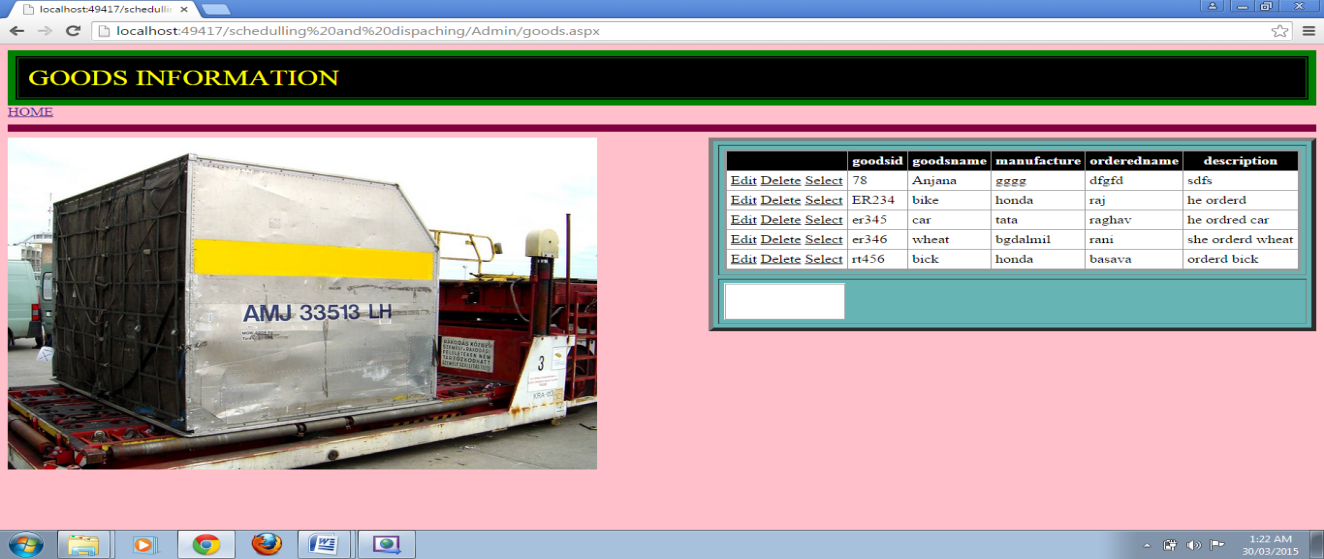
****

****

****

****

****

****

**SYSTEM TESTING & IMPLEMENTATIONS**

**UNIT TESTING**

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Then the web form level testing is made. For example storage of data to the table in the correct manner.

In the company as well as seeker registration form, the zero length username and password are given and checked. Also the duplicate username is given and checked. In the job and question entry, the button will send data to the server only if the client side validations are made.

The dates are entered in wrong manner and checked. Wrong email-id and web site URL (Universal Resource Locator) is given and checked.

**INTEGRATION TESTING:**

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus the system testing is a confirmation that all is correct and an opportunity to show the user that the system works.

**VALIDATION TESTING:**

The final step involves Validation testing, which determines whether the software function as the user expected. The end-user rather than the system developer conduct this test most software developers as a process called “Alpha and Beta Testing” to uncover that only the end user seems able to find.

The compilation of the entire project is based on the full satisfaction of the end users. In the project, validation testing is made in various forms. In question entry form, the correct answer only will be accepted in the answer box. The answers other than the four given choices will not be accepted.

**MAINTENANCE:**

The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In our project the process can be added without affecting other parts of the system.

Maintenance plays a vital role. The system liable to accept any modification after its implementation. This system has been designed to favour all new changes. Doing this will not affect the system’s performance or its accuracy.

In the project system testing is made as follows:

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Then the web form level testing is made. For example storage of data to the table in the correct manner.

In the form, the zero length username and password are given and checked. Also the duplicate username is given and checked. The client side validations are made.

The dates are entered in wrong manner and checked. Wrong email-id is given and checked.

This is the final step in system life cycle. Here we implement the tested error-free system into real-life environment and make necessary changes, which runs in an online fashion. Here system maintenance is done every months or year based on company policies, and is checked for errors like runtime errors, long run errors and other maintenances like table verification and reports.

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.

The application is implemented in the Internet Information Services 5.0 web server under the Windows 2000 Professional and accessed from various clients.

**Software testing** is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.[[1]](http://en.wikipedia.org/wiki/Software_testing#cite_note-0) Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding [software bugs](http://en.wikipedia.org/wiki/Software_bug) (errors or other defects).

Software testing can also be stated as the process of validating and verifying that a software program/application/product:

1. meets the business and technical requirements that guided its design and development;
2. works as expected; and
3. can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

Different software development models will focus the test effort at different points in the development process. Newer development models, such as [Agile](http://en.wikipedia.org/wiki/Agile_software_development), often employ [test driven development](http://en.wikipedia.org/wiki/Test_driven_development) and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed.

**White box testing** is when the tester has access to the internal data structures and algorithms including the code that implement these.

**Types of white box testing**

The following types of white box testing exist:

* [API](http://en.wikipedia.org/wiki/Application_programming_interface) testing (application programming interface) - testing of the application using public and private APIs
* [Code coverage](http://en.wikipedia.org/wiki/Code_coverage) - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
* [Fault injection](http://en.wikipedia.org/wiki/Fault_injection) methods - improving the coverage of a test by introducing faults to test code paths
* [Mutation testing](http://en.wikipedia.org/wiki/Mutation_testing) methods
* [Static testing](http://en.wikipedia.org/wiki/Static_testing) - White box testing includes all static testing

**Test coverage**

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important [function points](http://en.wikipedia.org/wiki/Function_points) have been tested.[[21]](http://en.wikipedia.org/wiki/Software_testing#cite_note-20)

Two common forms of code coverage are:

* Function coverage, which reports on functions executed
* Statement coverage, which reports on the number of lines executed to complete the test

They both return a [code coverage](http://en.wikipedia.org/wiki/Code_coverage) [metric](http://en.wikipedia.org/wiki/Software_metric), measured as a [percentage](http://en.wikipedia.org/wiki/Percentage).

[[edit](http://en.wikipedia.org/w/index.php?title=Software_testing&action=edit&section=17)]**Black box testing**

Main article: [Black box testing](http://en.wikipedia.org/wiki/Black_box_testing)

Black box testing treats the software as a "black box"—without any knowledge of internal implementation.

**Specification-based testing**: Specification-based testing aims to test the functionality of software according to the applicable requirements.[[22]](http://en.wikipedia.org/wiki/Software_testing#cite_note-21) Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Specification-based testing is necessary, but it is insufficient to guard against certain risks.[[23]](http://en.wikipedia.org/wiki/Software_testing#cite_note-22)

**Advantages and disadvantages**: The black box tester has no "bonds" with the code, and a tester's perception is very simple: a codemust have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not. On the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when (1) a tester writes many test cases to check something that could have been tested by only one test case, and/or (2) some parts of the back-end are not tested at all.

**CODE EFFICIENCY**

**MEASURES OF CODE EFFICIENCY**

The code is designed with the following characteristics in mind.

**Uniqueness**: The code structure must ensure that only one value of the code with a single meaning is correctly applied to a give entity or attribute.

**Expandability**: The code structure are designed for in a way that it must allow for growth of it’s set of entities or attributes, thus providing sufficient space for the entry of new items with in each classification.

**Conciseness**: The code requires the fewest possible number of positions to include and define each item.

**Uniform size and format**: Uniform size and format is highly desirable in mechanized data processing system. The addition of prefixes and suffixes to the root code should not be allowed especially as it is incompatible with the uniqueness requirement.

**Simplicity:** The codes are designed in a simple manner to understand and simple to apply.

**Versatility:** The code allows modifying easily to reflect necessary changes in conditions, characteristics and relationship of the encoded entities. Such changes must result in a corresponding change in the code or coding structure.

**Sortability**: Reports are most valuable for user efficiency when sorted and presented in a predetermined format or order. Although data must be sorted and collaged, the representative code for the date does not need to be in a sortable form if it can be correlated with another code that is sortable.

**Stability:** Codes that do not require to be frequently updated also promote use efficiency. Individual code assignments for a given entity should be made with a minimal likelihood of change either in the specific code or in the entire coding structure.

**Meaningfulness**: Code is meaningful. Code value should reflect the characteristics of the coded entities, such as mnemonic features unless such a procedures results in inconsistency and inflexibility.

**Operatability:**  The code is adequate for present and anticipated data processing both for machine and human use. Care is taken to minimize the clerical effort and computer time required for continuing the operation.

**CONCLUSION**

**.** The cargo tracking system makes the process of tracking the cargo trucks easier and efficiently making the process secured and robust.

Through this online tracking facility you can get real time status information on your shipment. All you have to do is select the cargo, verify or change the prefix of your shipment number, enter the given number, click on the Submit button, and this system will automatically retrieve and display the current status of your cargo

What has been proposed, and is the goal of this project, is to develop a system in which the problems of current systems can be addressed.

**BIBLIOGRAPHY**

### [Mastering Visual C#.Net](http://books.google.co.in/books?id=9uT5p5hxXUkC&printsec=frontcover&dq=C%23.net+books&hl=en&sa=X&ei=QHxSU4yLJ8mUrAffkIEg&ved=0CDcQ6AEwAA)

### [Practical Database Programming With Visual C#.NET](http://books.google.co.in/books?id=2phfFKM8hyoC&printsec=frontcover&dq=C%23.net+books&hl=en&sa=X&ei=QHxSU4yLJ8mUrAffkIEg&ved=0CD4Q6AEwAQ)

### [Programming In C#, 3E](http://books.google.co.in/books?id=JUNqMPlZ4yMC&printsec=frontcover&dq=C%23.net+books&hl=en&sa=X&ei=QHxSU4yLJ8mUrAffkIEg&ved=0CGEQ6AEwBw)

### OOP with Microsoft Visual Basic.NET and Microsoft Visual C# .NET step by step

### [Essential ASP. NET with Examples in C#](http://books.google.co.in/books?id=DUwx_H3G0AEC&printsec=frontcover&dq=ASP.net+books&hl=en&sa=X&ei=rHxSU7aLAYrGrAf42oGQAg&ved=0CEIQ6AEwAQ)

### [Professional ASP.NET 3.5 SP1 Edition: In C# and VB](http://books.google.co.in/books?id=o7m0u0OJ_VEC&printsec=frontcover&dq=ASP.net+books&hl=en&sa=X&ei=rHxSU7aLAYrGrAf42oGQAg&ved=0CGAQ6AEwBg)

### [Beginning ASP.NET 2.0 with C#](http://books.google.co.in/books?id=4NSXpJFita0C&printsec=frontcover&dq=ASP.net+books&hl=en&sa=X&ei=rHxSU7aLAYrGrAf42oGQAg&ved=0CHUQ6AEwCQ)

* + 1. [**http://www.wikipedia.com**](http://www.wikipedia.com)
    2. [**http://www.asp.net**](http://www.asp.net)
    3. [**http://www.c#.net**](http://www.c#.net)