GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

eAuction

A Report for the Evaluation 3 of Project 2

Submitted by

KUMAR ADITYA

(1613101345)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

Dr. Tapas Kumar , Ph.D,

Professor

April / may - 2020

GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "eAuction" is the bonafide work of "KUMAR ADITYA (1613101345)" who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
Professor & Dean,

SIGNATURE OF SUPERVISOR

Dr. Tapas Kumar,Ph.D,
Professor
School of Computing Science & Engineering

TABLE OF CONTENTS

1. Problem Definition and scope of project

1.1 Purpose

Today internet become reality and usage of internet become very much popular and there is tremendous increase of online shopping in all over the world. There is so many advantages of online shopping for buyer and seller, and today's most advanced payment gateway can easily handle online transaction easily without any security threat. Seller need not required to establish store to sale his product buyer can bye product as per their convenient from anywhere anytime.

Electronic Shopping does have unique advantages for business Imagine a shop, a showroom or an office that is open 24hrs a day, 7 days a week. Proposed eAuction application is a B2C (business to consumer) application where consumer can bid and buy the product online.

Proposed eAuction application system is end to end solution for online bidding and selling process includes product uploading, dynamic product listing, search, buy, bid and add product to shopping cart, calculate price, calculate shipping cost, payment processing, invoice generation, order tracking, and generation of reports for the management.

This eAuction system can maintain Member account for frequent buyer, it is not compulsory to create account before transaction. It keeps record for all transaction done by the Member, however system does not stored any finical information like credit card number, bank details. All payment processing are done by the payment gateway.

System should have built in security features to handle different security threat like SQL injection, cross scripting, spamming.

This project report describes the software functional and nonfunctional requirements for release 1.0 of the eAuction system. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here is high priority and committed for release 1.0.

1.2 Objective

Online eAuction is complete end to end solution to cover all aspects of online bidding, buying and selling products.

The basic objective of developing this project is:

- Provides complete web site solution, search products, bidding, buying, including shopping cart, product management, Member management, Logical access management.

- All products are categorized by categories, and provide powerful product search facilities; search can be done by using product name, product description.

- Allow user to bid the products before the bid expiry, Highest bidder will won the bid and allow to buy the product

- Add multiple product to shopping card, automatic calculate the shipping price, calculate tax and calculate total amount.

- Can able to store product image with all products.

- Member can list their product for selling on this site

- Member can see the checkout transaction and bidding details in my account page.

- Administrator can also add product or edit product listing

- All administrator option must perform from web interface only includes product management, vendor management and transaction.

- Product Descriptions can include HTML formatting: We know that having complete control over each product description is important, and no two products are alike. System administration can enter product descriptions.

- Member can view Order History, Member one click re-order any prior order, Multiple billing/shipping addresses per Member (similar to Amazon.com style checkout).

- Consumer credit card number must not stored and displayed on site

- System must have transactional data, and all the update and delete operation must be recorded for audit purpose.

- System must able to vendor wise product shipping details with all relevant details.

- System has powerful logical access management in place, each user must be identified by login id and strict password policy is applied to secure the system

- Generate various reports for management like, Total Members by Date/Trend, Member to Purchase Stats (% of visitors that convert, etc), Total sale by date.

1.3 Project Scope

The eAuction System will permit online bidding, buying and selling and maintaining master information and generating various reports. The main users of the project are Member and system Administrator.

From an end-user perspective, the eAuction System Project consists of two functional elements: an enhanced Member module for search products, bid, buy, sell products, manage profile, shopping cart, checkout. And Administration module for Mange product, users, vendors and view the reports.

1.3.1 Member Module

An enhanced interface for Member to registration, edit profile, login, search product, add product to shopping cart, check the transaction, checkout. Following modules pages are available for Member.

CM-1: Home – It is the default page for the site. It shows the new product list. Member can view product category wise or can search product from. All links are available in this page.

CM-2: Login – Member need to login to view his account information and buy the product. If Member forgets his password he can get back old password from Forget password link. New Member can register for this site by click on register link.

CM-3: Register – New Member need to register to buy the product. Type all the details of the Member like email id, name, address, contact details and submit. System validate for email id, it should be unique. In login process Member need to type email id and password. Member should type valid email id because if he forget password old password will be sent to this email address.

CM-4: My Account – It shows the details of currently logged Member details and other links like Edit Profile, Logout, and Change Password.

CM-5: Edit Profile – Member can edit his profile like personal details, address, contact no, display name however Member can not edit email address once register.

CM-6: Change Password – Member can change his password from this link. Member must type his old password to change the password with new password.

CM-7: Logout – By clicking this link user logged out from this site all user session reset to default value.

CM-8: Products – Member can view the products by product category or search product from the database. It display product image, product name, product description, product pricing details. Member can buy product by clicking Buy It link.

CM-9: Shopping Cart – It display the product already added to the cart. User can add product in cart and logout it saves the details. Next time when user logged in it shows the shopping cart again to check out. After check out product from shopping cart removes. Member can edit the quantity of product, remove product from cart or add more product to cart.

CM-10: Check Out – Member can checkout the product from shopping card page. It needs to confirm the shipping address, total amount. Select the Card type. Type the card no, CSV no credit card expiry month and year. System generate unique transaction id for each transaction which can be used to track the transaction. All successful transaction can be viewed in my account page.

CM-11: Buy Product – Member can view the product details and apply for bid or buy the product from this page

CM-11: Sell Product – Member can register their products in this page, while register member can select the selling type whether member wants the product to be sell in fixed price or wants to sell in bidding process. Member enters the buying rate, bid details, upload product image, product details.

1.3.2 Administration Module

Administration module used to enter and edit product details, vendor details, maintain the admin users and other back office users view the reports; The Web-based administration module will include the following features:

AM-1: Login – Login page for the administrator and other system users. All admin users are identified by the user name, password and user type. Admin user can create new user, new product and vendor. Normal user can update details of his own profile however can not add, modify product and vendor data.

AM-2: Users – It show list of admin user, admin user can add new user, edit existing user details.

AM-3: Change Password – Change the current logged user password.

AM-4: Logout – By clicking this link user logged out from this site all user session reset to default value.

AM-5: Vendors – View all the vendor details, edit the vendor details and add new vendor details.

AM-6: Products – View list of product, edit the product details, add new product with name of product, product category, vendor name, product description, product price, tax, name of the product image for best viewing product image must be in (150X150 pixel) and store the image in *prod_image* folder at root directory.

AM-7: Reports – It shows report for the selected date range like no of member registered, Transaction held for the selected date range.

Both of these areas of functionality will be delivered as the first version of the eAuction Application for Hardware Sale is released. Functionality is described in more detail later in this document.

1.4 Technologies

1.4.1 Operating Environment

OE-1: The eAuction web application will operate with the following Web Browsers: Microsoft Internet Explorer version 5.0, 6.0. 7.0

OE-2: The eAuction web application shall operate on a server running the latest versions of IIS (Internet Information Server).

OE-3: The eAuction web application shall permit user access from Internet connection

OE-4: Operating System: Windows 2000. XP

OE-5: Software requirements: SQL Server 2005, .net framework 2.0.

OE-6: Languages used are asp.net using c# and scripting is done using JavaScript.

OE-7: Hardware Requirements: 256(minimum)/512(recommended) MB RAM

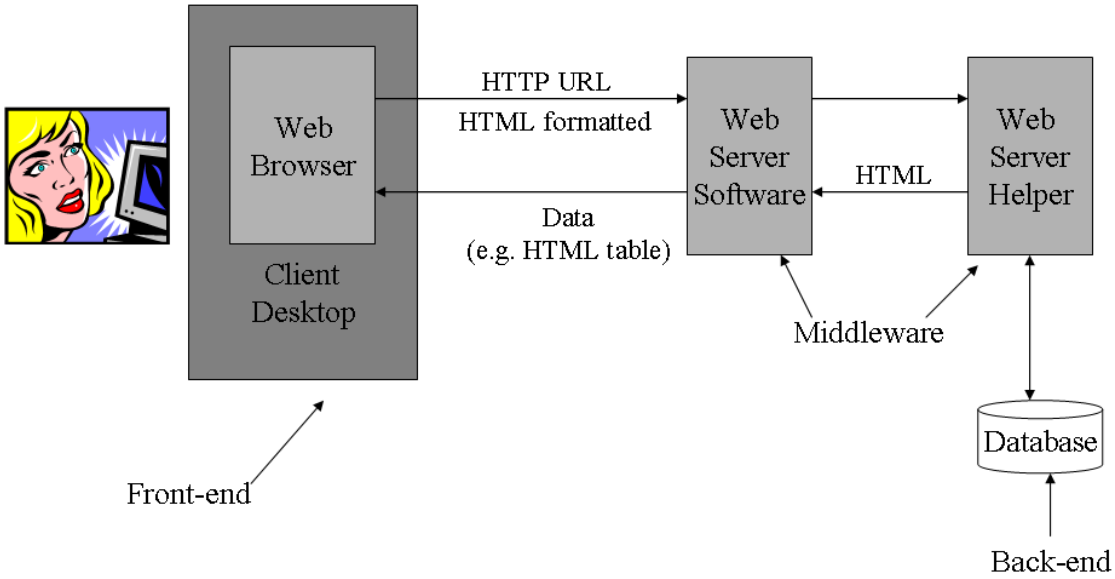
OE-8: Hard disc- nGB depending upon the requirement to store data minimum of 25GB.

1.4.2 Deployment Environment

DE-1: Database Server
OS – Win 2003 Enterprise Server
SQL Server 2005
HDD – Min 10 GB, Recommended 25 GB
RAM – Min 2 GB, Recommended 4 GB
Processor - Pentium Dual Xenon Processor

DE-2: Application Server
OS – Win 2003 Enterprise Server
IIS – Internet Information Server
HDD – Min 5 GB, Recommended 10 GB
RAM – Min 2 GB, Recommended 4 GB
Processor - Pentium Dual Xenon Processor

DE-3: The eAuction web application will operate with the following Web Browsers: Microsoft Internet Explorer version 5.0, 6.0. 7.0.



[eAuction System Architecture]

1.4.3 Development Tools and Technologies

DT-1: ASP.Net

As we need to develop Web Application for eAuction Application. We will used ASP.Net as it is the new Microsoft technology to develop any application which support and integrate other server product for the internet. By ASP.Net we can develop in web application by .NET framework and manage environment with any .NET support language like VB.Net and C#.

DT-2: C#

.NET is built on the Windows Server System to take major advantage of the OS and which comes with a host of different servers which allows for building, deploying, managing and maintaining Web-based solutions. The Windows Server System is designed with performance as priority and it provides scalability, reliability, and manageability for the global, Web-enabled enterprise. The Windows Server System integrated software products are built for interoperability using open Web standards such as XML and SOAP.

.NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it. When developed applications are deployed, those applications will target .NET and will execute wherever .NET is implemented instead of targeting a particular Hardware/OS combination. The components that make up the .NET platform are collectively called the .NET Framework.

The .NET Framework is a managed, type-safe environment for developing and executing applications. The .NET Framework manages all aspects of program execution, like, allocation of memory for the storage of data and

instructions, granting and denying permissions to the application, managing execution of the application and reallocation of memory for resources that are not needed.

The .NET Framework is designed for cross-language compatibility. Cross-language compatibility means, an application written in Visual Basic .NET may reference a DLL file written in C# (C-Sharp). A Visual Basic .NET class might be derived from a C# class or vice versa.

The .NET Framework consists of two main components:

Common Language Runtime (CLR)

Class Libraries

The advantage of C# includes

Powerful Windows-based Applications

Building Web-based Applications

Simplified Deployment

- Powerful, Flexible, Simplified Data Access
- Improved Coding
- Direct Access to the Platform
- Full Object-Oriented Constructs
- XML Web Services
- COM Interoperability

DT-3: SQL Server 2005

EAuction Application uses SQL Server 2005 as storing the data. Microsoft SQL Server 2005 as our database and it has so many features which is ideal for our dot net based application. Features Includes

- Support for Multiple Platforms
- Integration with Windows Back office servers

- Integration with Microsoft .NET Enterprise Servers
- Scalability
- Replication
- Centralized Management
- Reliability
- Automating Tasks

1.4.4 Development Environment

DE-1: 1.Visual Studio 2005 IDE

Most advanced integrated development environment for developing .NET application. Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It allows plug-ins to be added that enhance the functionality at almost every level - including adding support for source control systems (like Subversion and Visual SourceSafe) to adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports languages by means of language services, which allow any programming language to be supported (to varying degrees) by the code editor and debugger, provided a language-specific service has been authored. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), and C# (via Visual C#). Support for other languages such as Chrome, F#, Python, and Ruby among others has been made available via language services which are to be installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Language-

specific versions of Visual Studio also exist which provide more limited language services to the user. These individual packages are called Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

2. Overall Description

2.1 User Characteristics

2.1.1 Member

Able to register, login, edit personal information, shipping details, find the product information with updated price status

2.1.2 System User

View product and vendor details, Update personal details, Generate and view rep.

2.1.3 Administrator

Administrator will have all the access rights. Administrator can create new products, users and view the reports respective access rights to them.

2. 2.Assumptions

1) System User and Administrator communicate with each other via emails.

3. System Features

3. 1.Systern features

3.1.1 .Description:

A web based eAuction application which will use to buy, bid and sale various products over the internet. Proposed eAuction application system is end to end solution for online bidding, buying and selling process includes product updation, dynamic product listing, and search and add product to shopping cart, calculate price, calculate shipping cost, payment processing, invoice generation, order tracking, and generation of reports for the management.

3.1 .2.Constraints

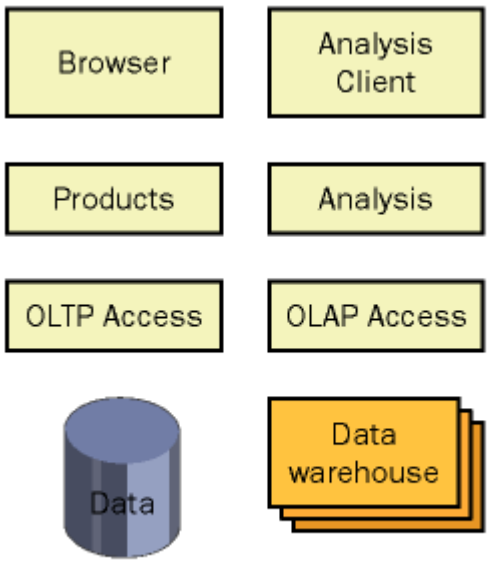
Linking and integration with legacy system for accounting. Integration with vendor and database through Web Services. Connecting to third-party OLAP applications for generating reports. Develop sophisticated system to automate the add multiple payment gateway.

3.1.4 Action/result

Action	Result
Member registers himself	After validation, naive member becomes registered member.
User/Admin Logins via Login page	According to the access rights redirected to the accordance page.
Register new User (Admin)	Redirected to User Details page and User Access page.
Searching Products	Redirected to Product Details page
Add New Product (Admin)	Redirected to Product Info page
Add Product to Shopping Cart	Redirect to Checkout page

4. Requirement Analysis

The requirement analysis outlines the approach the development team will take to meet the goals of the project and provides the basis for proceeding to the planning phase. After identifying the business problem and defining the vision and scope, the team creates the solution concept that explains in general terms how the team intends to meet the requirements of the project.



[Solution concept of eAuction]

For a project to be successful, it is essential that we correctly identify the goals of the project. Project goals can be categorized as follows:

- Business goals
- Design goals

4.1 Business goals

Business goals represent what the organization wants to achieve with the solution. Business goals form the basis for determining the success criteria of the solution. The purpose of defining business goals is to clearly articulate the objectives for the project and to ensure that your solution supports those business requirements. The team needs to determine the best method for identifying the goals and agreeing on them.

Throughout the life of the project, the team makes tradeoffs among resources, schedule, and features. It is important that business goals are prioritized in a way that will allow the

team to have a clear understanding about which ones the Member believes are most important, in case some of the goals cannot be achieved.

For this e-commerce project, business goals might include the following:

- Expand the company's geographic market beyond the current range of physical stores.
- Expand the company's demographic market to include younger consumers who have higher disposable incomes and who shop online with greater frequency than the current Member base.
- Shorten the time to sell products by using more efficient online sites.
- Integrate all suppliers worldwide by using a workflow process, and shorten the order placement and delivery cycle time.

4.2 Design goals

Design goals are similar to business goals in many ways. The difference is that design goals focus more on the attributes of the solution and less on what the solution will accomplish for the business. Design goals address not only what the team wants to accomplish but also what the team is not trying to accomplish with the solution. As with business goals, you need to prioritize design goals so that the team knows which goals must be accomplished, in case the project cannot achieve all of them.

Consider the case of this e-commerce Web site. Some of the design goals for the online shopping cart might include:

- Improve the user experience by reducing page-download wait times to 5 seconds or less.
- Limit dependency on connectivity with the server.
- Reduce the time and level of effort required for a user to complete the online registration.

- The service and all supporting applications must be localized for users worldwide.
- The service must have an availability of 99.99 percent.
- The service cannot lose data.
- The service must permit access only by authorized users.

4.3 Business Requirements

The following preliminary lists are based on initial interviews and study of existing manual system,

The business goal for the application is to support an increase the productivity and complete automation of existing manual or semi automatic eAuction Application. Business requirements are discussed in the Scope section, with the following additional detail:

- Improve the product search facility and Member should get all the information in a second.
- Multiple users must able to bid same product. Members must see the latest bid value.
- System should able to save the shopping card before final checkout.
- Admin user must able to upload product image.
- The Administrator should be able to enter or update master information like product details from web interface only
- System Administrator must able to control the access rights by each user as per requirement.
- The application should support the capability to use multi user environment.
- The Admin user should able to generate all type of reports as and when required by the management.
- All products must be categorized before listing. And system must allow member to select the selling type like fixed price or bid price.

- Every page on the Web site must display a search option with appropriate search controls and a navigation bar on the left side.

- The Member must be able to search for products by using a part of a product description and a price range.
- The home page must display products that are on sale and special offers, and it must include a picture and description of each product.
- A Product Details page must display information about a single product model. It must contain the product name and description, a large picture, and a price range.
- A Current Order page must display the products a Member has ordered and the quantity of each product, and it must include unit and total prices. A Member must be able to change quantities or remove items from this page. A Continue Shopping button must take the Member back to the last instance of the Products page visited.
- A Member Sign-in page must allow registered Members to sign in with their e-mail addresses and passwords. New Members must be able to register from this page.
- A Sign-in Information page must allow Members to input or change their e-mail addresses, passwords, and other personal information.
- An Address Maintenance page must allow Members to create, view, update, and delete billing and shipping addresses in their profiles.
- An Addresses Selection page must allow Members to select or remove billing and shipping addresses for an order.
- An Order Summary page must display an order, and it must include sales tax, shipping cost, and the details of reseller discounts. The Member must be able to change the quantities and remove items.
- An Arrange Payment page must allow Members to use credit cards that are on file in the database or to input new credit card information. Resellers must be able to select payment types and input purchase order numbers.
- An Order Confirmation page must display a summary of the submitted order, and it must include an order date, a confirmation number, and order status.
- An Order Status Lookup page must display a list of all orders entered by the signed-in Member. The Member must be able to select one of the orders so that it can be displayed in the Order Confirmation page.

The system users want to improve their current ability to analyze Member. In particular, they want to focus on identifying their best Members. To enable them to accomplish this goal, they want to extract meaningful data that easily answers the following questions:

- What are the early warning signs of problems?
- Who is my best Member?
- What are my members' issues as groups?
- What type of product are my members wanted?

4.4 User Requirements

User requirements are categorized by user type.

4.4.1 Member

- Able to edit personal information, shipping details.
- Able to find the product information with updated price status.
- Able to apply bid price for selected product
- Able to modify the shopping cart items.
- Able to view previous transaction and bidding details in My Account page.
- Able to add product for selling in the site.

4.4.2 System Users

- View product and vendor details.
- Update personal details
- Generate and view reports.

4.4.3 Administration

- Administrator must able to add new system users, modify details.
- Able to add and update product details.
- Able to add and update vendor details.
- Generate Reports as per requirement.

4.5 Operational requirements

The following requirements provide a high-level view of how the system will run:

- Processor usage should not exceed 80 percent during concurrent uses.
- Backups will occur incrementally throughout the day.
- A full weekly backup is required to WORM drives.
- Ensure that information is easy to access either, and meaningful for the system users and the company.
- Minimize the technical knowledge that system users and Member need to access the data, generate ad hoc queries, search and view books information.
- Any change to information must be reflected immediately, and the changes must be propagated to the search engine so that system users that perform searches see this new information.
- The application should work with the existing communications and networking infrastructure.
- The application should deploy with a minimum of additional operational processes, manual or otherwise.

4.6 System Requirements

These are additional constraints from a system perspective:

- The administrator must be able to monitor everything from the IT department.
- The information must be accessible by everyone in the company in intranet and in internet for the members as per the rights specify.
- All data especially product pricing must be up to date.

4.7 System Constraints

Constraints indicate the parameters to which the final business solution must adhere. They are aspects of the business environment that cannot or will not be changed. Often, these constraints become design goals for the application. If constraints are not identified properly, the project team might design a product that cannot be deployed within the business.

Examples of possible constraints that you should document include:

- Budget limitations
- Characteristics of earlier supporting systems
- Network system architecture
- Security requirements
- Operating systems
- Planned upgrades to technologies
- Network bandwidth limitations
- Maintenance and support agreements and structures
- Knowledge level of development or support staff
- Learning limitations of users

4.8 Behavioral Description

The Member of eAuction system registers himself by simply accessing the system's home page and fills the simple form. When the Member register, he can able to bid, bye or sell the products, search the product and able to add product to shopping cart. Member can view the *my account* page to see the status of order and last shopping cart details.

A System User can view and generate various reports update the profile.

Administrator is the user of the system with administrative rights. Administrator manage product to sale and defines master information like vendors. It creates user and specifies appropriate access rights to them. Monitor the system to view and generate various reports.

5. Nonfunctional Requirements

5.1 Performance Requirements

An application's performance is defined by metrics such as transaction throughput and resource utilization. A user might define an application's performance in terms of its response time. No more than 5-percent degradation in average query response is allowed while all concurrent users are using the system. Processor utilization should not exceed 80 percent during all concurrent users are using the system.

We must define performance requirements before the team proceeds to the developing phase. To define a good performance requirement, we must identify project constraints, determine services that the application will perform, and specify the load on the application.

PR-1: Identifying constraints - Constraints in the project include budget, schedule, infrastructure, and the choice of development tools or technologies. For example, we might need to deploy this eAuction application by a specific date. We might also need to use a specific development tool because the team has expertise in that tool only. We might not be able to design and develop applications that are processor intensive because the client computers do not have adequate hardware. We need to design an application so that it meets its performance goals within the limitations of the constraints. Instead of changing some aspects of a project to improve performance, you can modify aspects of the project that are not constrained to determine how we can improve performance. For example, can the team be trained so that they can create components by using a different tool? Can data access be improved by changing the data access technology?

PR-2: Determining features - The features of this application correspond to use cases and usage scenarios. We can identify the usage scenarios that affect the performance of the application and, for each such scenario, specify what the user does and what the application does in response, including how databases and other system services are accessed. In addition, you need to determine how often each feature will be used. This information can help you create tests for measuring performance that resemble actual usage of the application as closely as possible.

PR-3: Specifying the load - We can specify the load of this eAuction application as the number of clients that will use the application. In addition, we can examine how the load might vary over time. For example, the number of requests for this e-commerce site will be higher during certain times of year. We can use the load to define the performance metrics of this application.

5.2 Availability Requirements

Availability is a measure of how often the application is available to handle service requests as compared to the planned run time. Availability also takes into account repair time because an application that is being repaired is not available for use.

Designing for availability includes anticipating, detecting, and resolving hardware or software failures before they result in service errors, faults, or data corruption, thereby minimizing downtime. To ensure availability, provide multiple routes to application services and data. Use only tested, proven processes (both automated and people-based) that support the application throughout its life cycle.

In addition to unplanned downtime, planned downtime must be reduced. Planned downtime can include maintenance changes, operating system upgrades, backups, or any other activity that temporarily removes the application from service.

Availability of an application also depends on its reliability. For a highly available and reliable application, you need a reliable foundation: good application design, rigorous testing, and certification. Some of the techniques used for designing for availability include:

AR-1: Reduce planned downtime - To avoid planned downtime, use rolling upgrades. For example, to update a component on a server, move the server's resource groups to another server, take the server offline, update the component, and then bring the server online. Meanwhile, the other servers handle the workload, and this application experiences no downtime. You can use this strategy in an application that scales out.

AR-2: Reduce unplanned downtime with clustering - Clustering is a technology for creating high-availability applications. A cluster consists of multiple computers that are physically networked and logically connected using cluster software. By using clustering, a multiple server Web site can withstand failures with no interruption in service. When the active server fails, the workload is automatically moved to a passive server, current client processes are switched over, and the failed application service is restarted automatically. If a resource fails, Members connected to that server cluster might experience a slight delay, but the service will be completed. Cluster software can provide failover support for applications, file and print services, databases, and messaging systems that have been designed as cluster-aware and assigned to a cluster.

AR-3: Use network load balancing - Network load balancing (NLB) is used to distribute traffic evenly across available servers. NLB also helps increase the availability of an application: if a server fails, you can use NLB to redefine the cluster and direct traffic to the other servers. NLB is especially beneficial for e-commerce

applications that link external clients with transactions to data servers. As client traffic increases, you can scale out the Web server farm by adding up to 32 servers in a single cluster. NLB automatically detects server failures and redirects client traffic to the remaining servers, all the time maintaining continuous, unbroken client service.

AR-4: Use redundant array of independent disks (RAID) for data stores. - RAID uses multiple hard disks to store data in multiple places. If a disk fails, the application is transferred to a mirrored data image and the application continues running. The failed disk can be replaced without stopping the application.

AR-5: Isolate mission-critical applications - An application is constantly performing tasks and requesting resources such as network communications, data access, or process threads. Each of these resource requests can affect the performance and availability of applications sharing the same resources. If an application shares these services on the same servers, the workload and throughput characteristics for these servers might change unfavorably. It is recommended that mission-critical applications use dedicated infrastructures and private networks.

AR-6: Use queuing - Queuing enables your application to communicate with other applications by sending and receiving asynchronous messages. Queuing guarantees message delivery; it does not matter whether the necessary connectivity currently exists (with mobile applications, for example). Queuing removes a failure point from the application. Queuing is also a solution for managing peak workloads that can require a lot of hardware. In addition, by increasing the number of routes for successful message delivery, an application can increase the chances for successful and immediate message completion.

Calculation of availability

Measurement Types for Calculating Availability		
Name	Calculation	Definition
Mean Time Between Failure (MTBF)	Hours / Failure Count	Average length of time the application runs before failing
Mean Time To Recovery (MTTR)	Repair Hours / Failure Count	Average length of time needed to repair and restore service after a failure

The formula for calculating availability is:

Availability = (MTBF / (MTBF + MTTR)) × 100

For example, the typical availability requirement for this eAuction application is that the site is available 24 hours a day, 7 days a week. If you assume 1000 continuous hours as a checkpoint, two 1-hour failures during this time period results in availability of:

((1000 / 2) / ((1000 / 2) + 1)) × 100 = (500 / 501) × 100 = .998 × 100 = 99.8%.

A popular way to describe availability is by the nines, for example, three nines for 99.9 percent availability. However, the implication of measuring by nines is often misunderstood. We need to do the arithmetic to discover that three nines (99.9 percent availability) represent about 8.5 hours of service outage in a single year. The next level, four nines (99.99 percent), represents about 1 hour of service outage in a year. Five nines (99.999 percent) represent about 5 minutes of outage per year.

5.3 Reliability Requirement

The reliability of an application refers to the ability of the application to provide accurate results. Reliability and availability are closely related. While availability measures the capacity to handle all requests and to recover from a failure with the least loss of access to the application, reliability measures how long the application can execute and produce expected results without failing. Users bypass unreliable Web sites, resulting in lost revenue and reduced future sales. In addition, the expense of repairing corrupted data increases the cost of application failure. Unreliable systems are also difficult to maintain or improve because the failure points are typically hidden throughout the system. Because of the need no single point failure, automatic failover will be required. In addition, existing disaster recovery and backup plans and procedures must be revised to incorporate the eAuction Application.

To design for reliability, you need to examine the application's expected usage pattern, create a reliability profile, and create a solution that meets the profile. You must examine how a particular service is provided, evaluate failure scenarios, and design preferred alternatives. In addition, you need to consider the application's interactions with other applications.

It is difficult to identify reliability problems and solutions for a system that has not been developed. However, we can begin by analyzing the currently running applications in the organization. Such analysis reveals the failure frequency and distribution, root causes, and possible improvements for existing systems. We can use this information to design a reliable solution.

A reliable solution ensures error-free data input, data transformations, state management, and non-corrupting recovery from any failure conditions. Creating a high-reliability application depends on the entire software development lifecycle, from the planning phase, through development and testing, to deployment and stabilizing. The following tasks can help you create a reliable application:

- Putting reliability requirements in the specification

- Using a good architectural infrastructure
- Including management information in the application
- Using redundancy
- Using quality development tools
- Using reliability checks that are provided by the application
- Implementing error handling
- Reducing the application's functionality instead of completely failing the application

5.4 Scalability Requirement

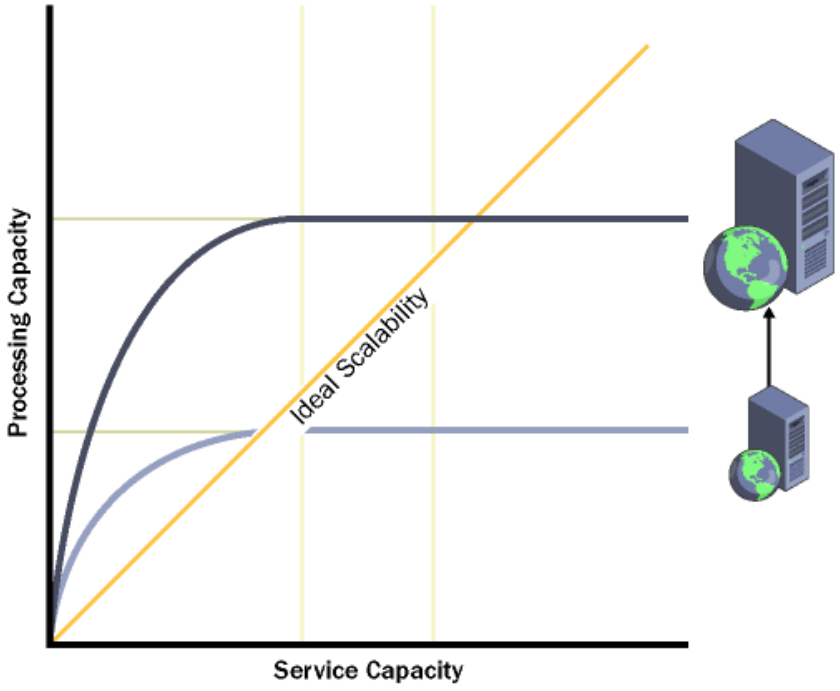
Scalability is defined as the capability to increase resources to produce an increase in the service capacity. In a scalable application, you can add resources to manage additional demands without modifying the application itself.

A scalable application requires a balance between the software and hardware used to implement the application. You might add resources to either software or hardware to increase the scalability of the application. Adding these resources might produce a benefit; however, it could also have a negative or null effect, with the application showing no significant increase in service capacity. For example, you might implement load balancing in an application. This will help only minimally if the application has been written to make synchronous method calls or to retrieve lengthy datasets in response to a user's request.

eAuction Application an average load of 1500 concurrent users after the system is fully operational, and expects that to grow by 25 percent each year for the next five years.

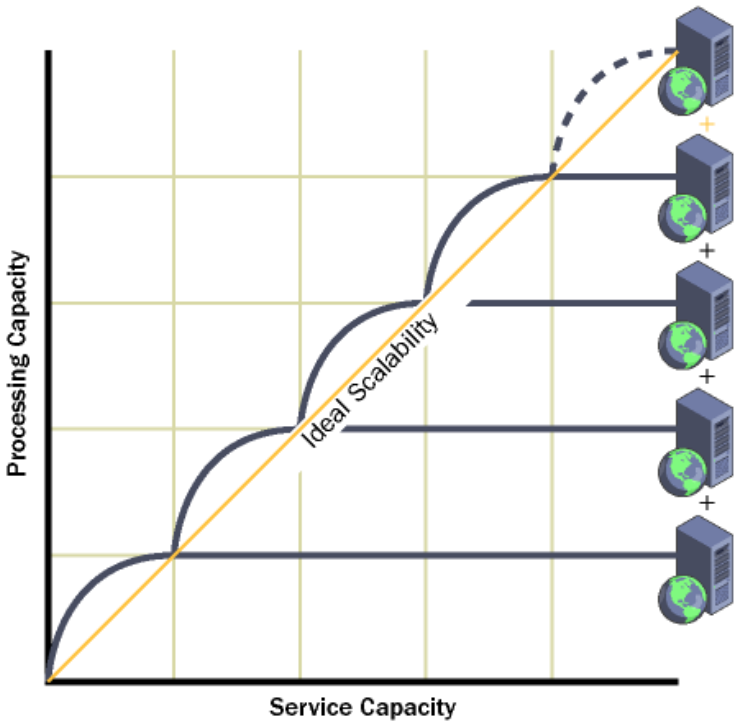
The two most common approaches to scalability are:

> **SR-1: Scaling up** - Refers to achieving scalability by improving the existing server's processing hardware. Scaling up includes adding more memory, more or faster processors, or migrating the application to a more powerful computer. The primary goal of scaling up an application is to increase the hardware resources available to the application. Typically, you can scale up an application without changing the source code. In addition, the administrative effort does not change drastically. However, the benefit of scaling up tapers off eventually until the actual maximum processing capability of the machine is reached.



[Scaling Up]

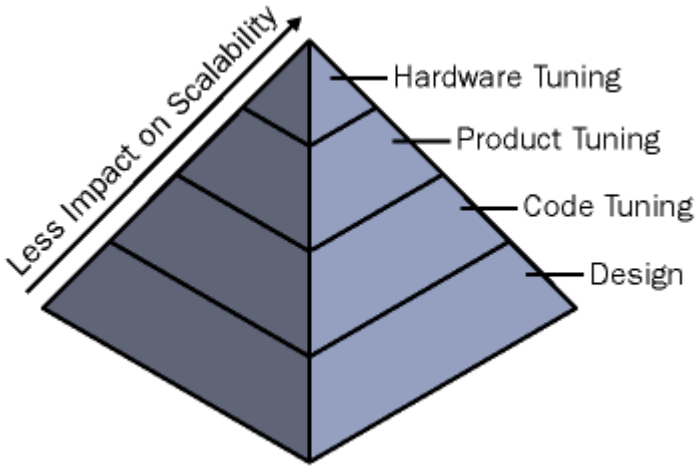
SR-2: Scaling out - Refers to distributing the processing load across more than one server. Although scaling out is achieved by using multiple computers, the collection of computers continues to act as the original device configuration from the end-user perspective. Again, the balance between software and hardware is important. The application should be able to execute without needing information about the server on which it is executing. This concept is called location transparency. Scaling out also increases the fault tolerance of the application.



[Scaling Out]

Good design is the foundation of a highly scalable application. The planning phase has the greatest impact on the scalability of an application.

Bellow figure illustrates the role of design, code tuning, product tuning, and hardware tuning in the scalability of an application. Design has more impact on the scalability of an application than the other three factors. As you move up the pyramid, the impact of various factors decreases. The pyramid illustrates that effective design adds more scalability to an application than increased hardware resources.



[Scalability pyramid]

To design for scalability, we need to following guidelines:

SR-3: Design processes such that they do not wait - A process should never wait longer than necessary. A process can be categorized as synchronous or asynchronous. A synchronous process waits for another process to complete before it continues. Such processes must wait for another process to succeed or fail completely before performing another operation. Applications that implement synchronous processes encounter bottlenecks for resources. These bottlenecks affect both the performance and the scalability of the application. One way to achieve scalability is to implement asynchronous processes. In applications that have asynchronous processes, long-running operations can be queued for completion later by a separate process.

SR-4: Design processes so that processes do not compete for resources - One of the biggest causes of scalability problems is competition for resources such as memory, processor cycles, bandwidth, or database connections. Plan your resource usage to minimize these problems:

- Sequence resource usage to use the most plentiful resources first and the least plentiful resources last.
- Acquire resources as late as possible. The shorter the amount of time a process uses a resource, the sooner the resource becomes available to another process.

SR-5: Design processes for commutability - Two or more operations are called commutative if they can execute in any order and still obtain the same result. Typically, operations that do not involve transactions are commutative. For example, a busy e-commerce site that continuously updates the inventory of its products might experience contention for record locks. To prevent this, each inventory increment and decrement could become a record in a separate inventory transaction table. Periodically, the database could add the rows of this table for each product and then update the product records with the net change in inventory.

SR-6: Design components for interchangeability - An interchangeable component is designed to release its resources, move into a pool managed by a resource manager, and be re-initialized for use by a new client. Design the component so that no client-specific state persists from client to client. In addition, the component should support aggregation and not be bound to a specific thread. Resource pooling schemes such as COM+ component pooling and Open Database Connectivity (ODBC) connection pooling use interchangeable resources. For example, you can use the Component Services Administration tool to enable object pooling, set minimum and maximum pool size, and create timeout settings. For more information.

SR-7: Partition resources and activities - Minimize relationships between resources and activities by partitioning them. This helps you avoid the risk of bottlenecks. Partitioning activities can also ease the load on critical resources such as the processor and bandwidth. For example, using Secure Sockets Layer (SSL) to provide a secure connection results in significant overhead. Therefore, you might decide to use SSL only for pages that require a high level of security and use dedicated Web servers to handle SSL sessions. You can also partition resources and activities by creating many small components rather than a few large components, and by limiting cross-device communication. However, partitioning can make a system more complex. Dividing resources that have dependencies can add significant overheads to an operation.

5.5 Security Requirements

Malicious attackers use various methods to exploit system vulnerabilities to achieve their goals. Vulnerabilities are weak points or loopholes in security that an attacker exploits to gain access to an organization's network or to resources on the network. Some vulnerabilities, such as weak passwords, are not the result of application or software development design decisions. However, it is important for an organization to be aware of such security weaknesses to better protect its systems. Common vulnerabilities of applications include:

SR-1: Weak passwords - A weak password might give an attacker access not only to a computer, but to the entire network to which the computer is connected.

SR-2: Misconfigured software - Often the manner in which software is configured makes the system vulnerable. If services are configured to use the local system account or are given more permissions than required, attackers can exploit the services to gain access to the system and perform malicious actions on the system.

SR-3: Social engineering - A common form of discovering passwords that generally occurs when users are not aware of security issues and can be deceived into revealing their passwords. For example, an attacker posing as a help desk administrator might persuade a user to reveal his or her password under the pretext of performing an administrative task.

SR-4: Internet connections - The default installation of Internet Information Services (IIS) version 5.0 often enables more services and ports than are necessary for the operation of a specific application. These additional services and ports provide more opportunities for potential attacks. For example, modem connections bypass firewalls that protect networks from outside intruders. If an intruder can identify the modem

telephone number and password, the intruder can connect to any computer on the network.

SR-5:Unencrypted data transfer - If the data sent between a server and the users is in clear text, there is a possibility that the data can be intercepted, read, and altered during transmission by an attacker.

SR-6: Buffer overrun - Malicious users probe applications looking for ways to trigger a buffer overrun because they can use a buffer overrun to cause an application or an operating system to crash. They can then find more security weaknesses by reading error messages.

SR-7: SQL injection - SQL injection occurs when developers dynamically build SQL statements by using user input. The attacker can modify the SQL statement and make it perform operations that were not intended.

SR-8: Secrets in code - Many security problems are created when a malicious user is able to find secrets that are embedded in code, such as passwords and encryption keys.

To design a secure eAuction application, we should be familiar with the following principles of security and employ them when creating security strategies:

SR-9: Rely on tested and proven security systems - Whenever possible, we should rely on tested and proven security systems rather than creating your own custom solution. Use industry-proven algorithms, techniques, platform-supplied infrastructure, and vendor-tested and supported technologies. If we decide to develop a custom security infrastructure, validate our approach and techniques with expert auditing and security review organizations before and after implementing them.

SR-10: Never trust external input - We should validate all data that is entered by users or submitted by other services.

SR-11: Assume that external systems are not secure - If our application receives unencrypted sensitive data from an external system, assume that the information is compromised.

SR-12: Apply the principle of least privilege - Do not enable more attributes on service accounts than those minimally needed by the application. Access resources with accounts that have the minimal permissions required.

SR-13: Reduce available components and data - Risk will increase with the number of components and amount of data you have made available through the application, so you should make available only the functionality that you expect others to use.

SR-14: Default to a secure mode - Do not enable services, account rights, and technologies that you do not explicitly need. When we deploy the application on client or server computers, its default configuration should be secure.

SR-15: Do not rely on security by obscurity - Encrypting data implies having keys and a proven encryption algorithm. Secure data storage will prevent access under all circumstances. Mixing up strings, storing information in unexpected file paths, and so on, is not security.

SR-9: Follow STRIDE principles - Each letter in the STRIDE acronym specifies a different category of security threat: spoofing identity, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. These are classes of security vulnerabilities a system needs to protect itself against.

Security Features of .NET Technologies

.NET Web applications implement one or more of the logical services by using technologies such as Microsoft ASP.NET, Enterprise Services, XML Web services, remoting, Microsoft ADO.NET, and Microsoft SQL Server. To create effective security strategies, we need to understand how to fine-tune the various security features within each product and technology area, and how to make them work together.

Authentication

Authentication is the process of discovering and verifying the identity of a user by examining the user's credentials and then validating those credentials against some authority. A variety of authentication mechanisms are used, some of which can be used with .NET Framework role-based security.

Examples of commonly used authentication mechanisms include the operating system, Passport, and application-defined mechanisms, such as NTLM and Kerberos version 5 authentication.

Authorization

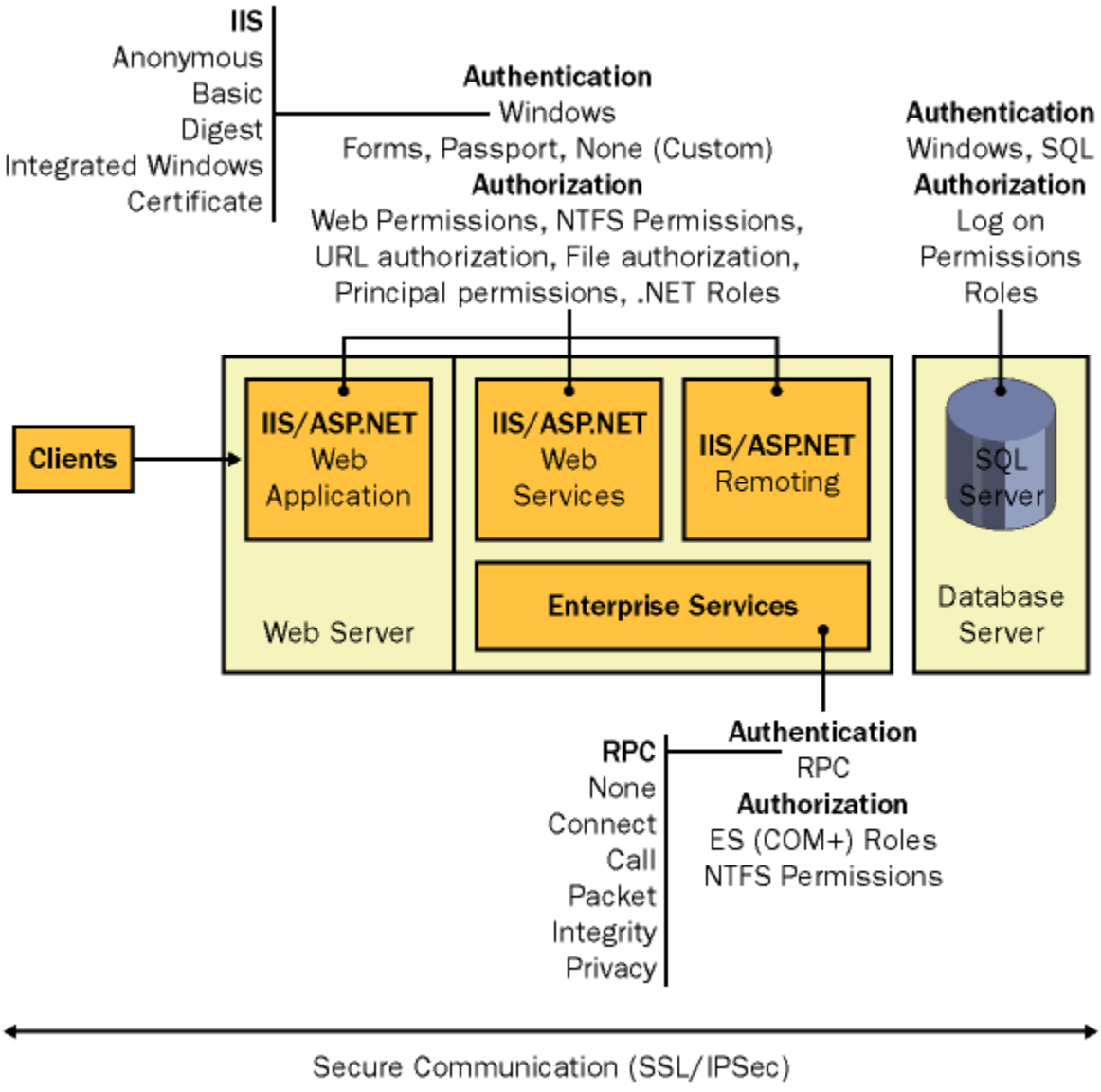
Authorization is the process of determining whether a user is allowed to perform a requested action. Authorization occurs after authentication and uses information about a user's identity and roles to determine what resources that user can access. You can use .NET Framework role-based security to implement authorization.

ASP.NET security

ASP.NET provides a useful tool for application developers to use to create Web pages. When a Web site records a user's credit card information, the file or database that stores such information must be secured from public access. ASP.NET, in conjunction with IIS, can authenticate user credentials such as names and passwords

ADO.NET and SQL Server

ADO.NET provides data access services. It is designed for distributed Web applications, and supports disconnected scenarios. When we build Web-based applications, it is essential that we must use a secure approach to accessing and storing data. ADO.NET and SQL Server provide several security features that can be used to ensure secure data access.



[Security architecture]

5.6 Interoperability

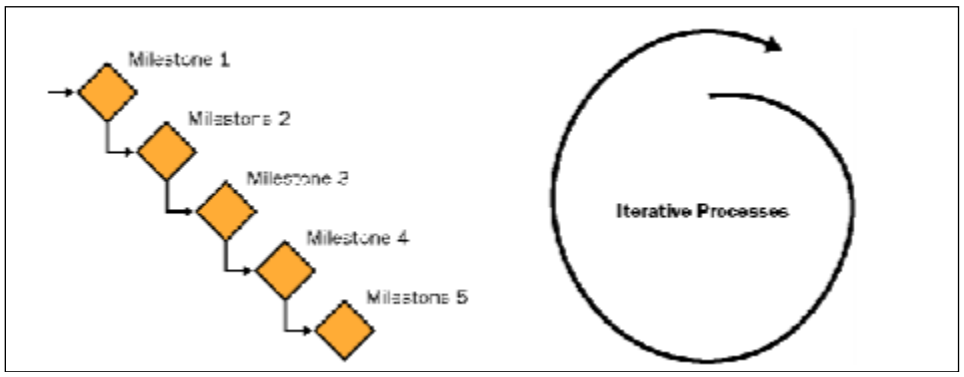
Typically, medium and large organizations have heterogeneous computing environments. For example, many organizations deploy distributed n-tier client/server applications that require access to data or transactions on existing systems. In addition, your application might need to interact with applications that have been developed using proprietary or third-party software. In Version 1.0 of the eAuction Application, there are no requirements for interoperability with other systems.

6. Project Management

6.1 Development Methodologies

What Are Process Models and development methodology?

A process model guides the order of project activities and represents the life cycle of a project. Historically, some process models are static and others do not allow checkpoints. Two such process models are the waterfall model and the spiral model.



[The waterfall model and the spiral model]

These models provide two different approaches to the project life cycle. The preceding illustration shows the waterfall model's cascading checkpoints and the spiral model's circular approach to process.

6.1.1 Waterfall model.

This model uses milestones as transition and assessment points. When using the waterfall model, you need to complete each set of tasks in one phase before moving on to the next phase. The waterfall model works best for projects in which the project requirements can be clearly defined and are not liable to modifications in the future. Because this model has fixed transition points between phases, you can easily monitor schedules and assign clear responsibilities and accountability.

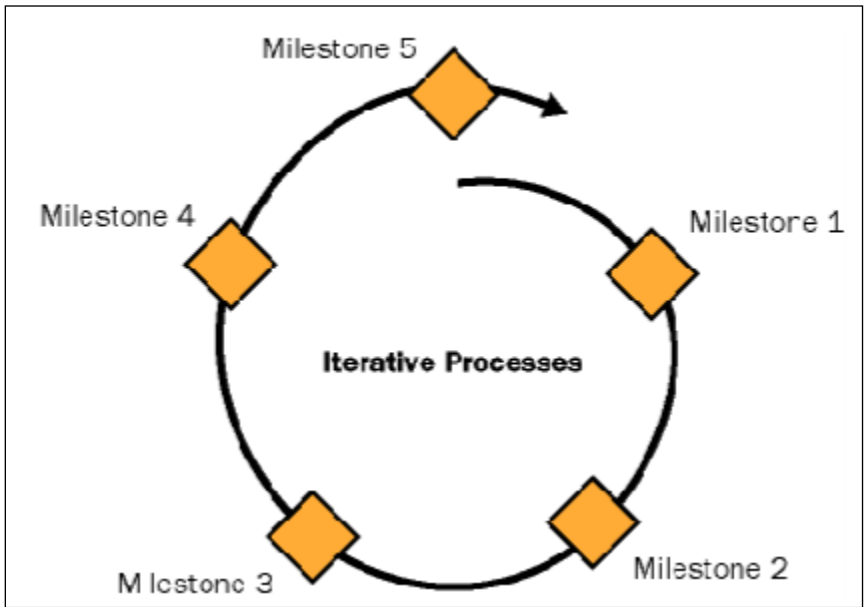
6.1.2 Spiral model.

This model is based on the continual need to refine the requirements and estimates for a project. The spiral model is effective when used for rapid application development of very small projects. This approach can generate great synergy between the development team and the Member because the Member is involved in all stages by providing feedback and

approval. However, the spiral model does not incorporate clear checkpoints. Consequently, the development process might become chaotic.

6.1.3 MSF Process Model:

The MSF – Microsoft Solution Framework Process Model combines the best principles of the waterfall and spiral models. It combines the waterfall model's milestone-based planning and resulting predictability with the spiral model's benefits of feedback and creativity.



[MSF Process Model]

In eAuction devolvement process we have used MSF Process Model. This is a combination of spiral and waterfall process.

6.2 Project Development Life Cycle (PDLC)

The project will completed by this four distinct phases as per MSF Process Model.

6.2.1 Envisioning

Envisioning is gathering the requirement of the project from different sources. Some techniques for gathering information are interviewing, shadowing, user instructions, and prototyping. Creating and identify the project scope. The scope of the project specifies what will and will not be included in the project. In this phase we have created the scope

document. It includes information about the team and project structure, the problem statement, the vision statement, the scope of the project, the solution concept, user profiles, and project goals.

6.2.2 Planning

The planning phase results in the architecture and design of the solution, the plans to accomplish the development and deployment of the solution, and the schedules associated with tasks and resources. There are three design processes in the planning phase: conceptual, logical, and physical design.

6.2.3 Developing and Designing the System

Design of any application is not complete without a way for users to interact with the system. User interaction takes place through the application's presentation layer. The presentation layer is the part of the application that provides a communication mechanism between the user and the business service layer of the system. The most simple presentation layers contain user interface components, such as Windows Forms or ASP.NET Web Forms. For more complex user interactions, you can design user process components to orchestrate the user interface elements and control the user interaction. User interface components display data to users, acquire and validate data from user input, and interpret user gestures that indicate the user wants to perform an operation on the data. Additionally, the user interface should filter the available actions to let users perform only the operations that are appropriate at a certain point in time.

6.2.4 Stabilizing

The purpose of the stabilizing phase is to reduce the risks of releasing the solution to production. A successful stabilizing phase requires that the team make the transition from a mindset focused on building features to one focused on getting the solution to a known state of quality. Deliverables of the deploying phase are operations and support information systems, repository of all versions of documentation and code, and project closeout reports.

7. Estimation and Planning

7.1 Planning

The MSF – Microsoft Solution Framework Process Model describes a generalized sequence of activities for building and deploying enterprise solutions. This process is flexible and can accommodate the design and development of a broad range of enterprise projects. The MSF Process Model is a phase-based, milestone-driven, and iterative model that can be applied to developing and deploying traditional applications, enterprise solutions for eAuction, and Web-distributed applications.

MSF guidance includes disciplines for managing the people, processes, and technology elements that most projects encounter. The three key MSF disciplines are risk management, readiness management, and project management.

7.1 Application of the MSF model in our project:

7.1.1 Envisioning process:

Each phase in the MSF Process Model has interim milestones and a major milestone. Interim milestones are associated with the various activities that are performed in a phase, such as creating a team and creating a vision/scope document. The major milestone indicates that the team can progress to the next phase in the MSF Process Model. For example, the major milestone of the envisioning phase is the vision/scope approved milestone. When the team reaches this milestone, the team can progress to the planning phase of the MSF Process Model. The team creates deliverables for each task in the envisioning phase. Together, these deliverables provide context and direction for the team for the remainder of the project, and communicate the project vision and scope to the Member. The deliverables that the team creates during the envisioning phase include:

- Vision/scope - Problem statements and business objectives, A review of the existing processes, A broad definition of user requirements.
- Project structure - A project structure and process standards for the team to follow
- Risk assessment - A preliminary risk assessment, Plans for mitigating or eliminating the identified risks

7.1.2 Planning process:

During the planning phase, the team determines what to develop and plans how to create the solution. The team prepares the functional specification, creates a design of the

solution, and prepares work plans, cost estimates, and schedules for the various deliverables.

The planning phase involves the analysis of requirements. These requirements can be categorized as business requirements, user requirements, operational requirements, and system requirements. These requirements are used to design the solution and its features and to validate the correctness of the design.

After gathering and analyzing the requirements, the team creates the design of the solution. The team creates user profiles that specify the various users of the solution and their roles and responsibilities. The team then creates a series of usage scenarios. A usage scenario specifies the activity performed by a particular type of user. Therefore, the team needs to create usage scenarios for all user profiles. After creating usage scenarios, the team creates use cases for the usage scenarios. A use case specifies the sequence of steps that a user will perform in a usage scenario.

The planning phase deliverables provide the basis for making future tradeoff decisions. The following deliverables are produced during the planning phase:

- Functional specification
- Risk management plan
- Master project plan and master project schedule

7.1.3 Design Process:

The three design stages are:

- Conceptual design, in which you view the problem from the perspective of the users and business requirements and define the problem and solution in terms of usage scenarios.
- Logical design, in which you view the solution from the perspective of the project team and define the solution as a set of services.
- Physical design, in which you view the solution from the perspective of the developers and define the technologies, component interfaces, and services of the solution.

You document the solution design in the functional specification. The functional specification describes the behavior and appearance of each feature of the solution. It also describes the architecture and the design for all features.

7. 1.4 Development process:
During the developing phase, the project team creates the solution. This process includes creating the code that implements the solution and documenting the code. In addition to developing code, the team also develops the infrastructure for the solution.

The deliverables of the developing phase include:

- Source code and executable files
- Installation scripts and configuration settings for deployment
- Finalized functional specification
- Performance support elements
- Test specifications and test cases

7. 1.5 Stabilization process:
During the stabilizing phase, the team performs integration, load, and beta testing on the solution. In addition, the team tests the deployment scenarios for the solution. The team focuses on identifying, prioritizing, and resolving issues so that the solution can be prepared for release. During this phase, the solution progresses from the state of all features being complete as defined in the functional specification for this version to the state of meeting the defined quality levels. In addition, the solution is ready for deployment to the business

The deliverables of the stabilizing phase are as follows:

- Final release
- Release notes
- Performance support elements
- Test results and testing tools
- Source code and executable files
- Project documents
- Milestone review

7. 1.6 Deployment process:

During this phase, the team deploys the solution technology and site components, stabilizes the deployment, transfers the project to operations and support, and obtains final Member approval of the project. After deployment, the team conducts a project review and a Member satisfaction survey. The deploying phase culminates in the deployment complete milestone.

The deliverables of the deploying phase are as follows:

- Operation and support information systems

    - Procedures and processes
    - Knowledge base, reports, and logbooks
- Documentation repository for all versions of documents and code developed during the project
- A training plan
- Project completion report
    - Final versions of all project documents
    - Member satisfaction data
    - Definition of next steps

6.2 Estimation

6.2. 1 Basic COCOMO

The COnstructive COst MOdel (COCOMO) is an algorithmic Software Cost Estimation Model developed by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics. Constructive Cost Model: It is a hierarchy of estimation models that address: Application composition model: Used during the early stage of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount..

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes

The basic COCOMO equations take the form

Effort Applied = $a_b(KLOC)b^b$ [man-months]

Development Time = $c_b(Effort\ Applied)d^b$ [months]

People required = Effort Applied / Development Time [count]

The coefficients a_b, b_b, c_b and d_b are given in the following table.

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi – Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

E =3.0(KLOC)1.12
= 3.0(5000)^1.12
= 18 person months

D = 2.5E0.35
= 2.5(22)0. 35
= 6.8 months

Therefore No. of persons required =18/6.8=3persons.

6.2.2 Function Point Estimation

A function point is a unit of measurement to express the amount of business functionality an information system provides to a user. Function points are the units of measure used by the IFPUG Functional Size Measurement Method. The IFPUG FSM Method is an ISO recognised software metric to size an information system based on the functionality that is perceived by the user of the information system, independent of the technology used to implement the information system.

The method of measuring the size of an information system and expressing it in a number of function points is called function point analysis (FPA). The method is kept up to date by worldwide cooperating FPA user groups like NESMA and IFPUG. A function point analysis expresses the functional size of an information system in a number of function points (for example: the size of a system is 314 FPs). There are many uses and benefits of function points and the functional size may be used as input into many types of project and organization decisions including determining the:

- Budget for application development or enhancement costs.
- Budget for the annual maintenance costs of the application portfolio.
- Project productivity after completion of the project.
- Software Size for cost estimating.

Function-Oriented Metrics

FP = count_total * [0.65 + 0.01 * sum of F_i]

1. Does the system require reliable backup and recovery=5
2. Are data communications required=4
3. Are there distributed processing functions=2
4. Is performance critical=5
5. Will the system run an existing, heavily utilized operational environment=5
6. Does system requires online data entry=5
7. Does online data entry req. input transaction to be build on multiple screens=3
8. Are master files updated online =4
9. Are I/ps , 0/ps, files and inquires complex=3
10. Is essential processing complex=5
11. Is code reusable=4
12. Are conversion and installation included in design=2
13. Is system supports multiple installations =2
14. Is application designed to facilitate change and ease of use by user=5

sum of **F_i = 54**

Measurement Parameter	Count		Simple	Average	Complex		
Number of User Inputs	12	*	3	4	6	=	36
Number of User outputs	4	*	4	5	7	=	16
Number of User inquires	1	*	3	4	6	=	6
Number of files	5	*	7	10	15	=	35
Number of External interface	3	*	5	7	10	=	15
Count Total							108

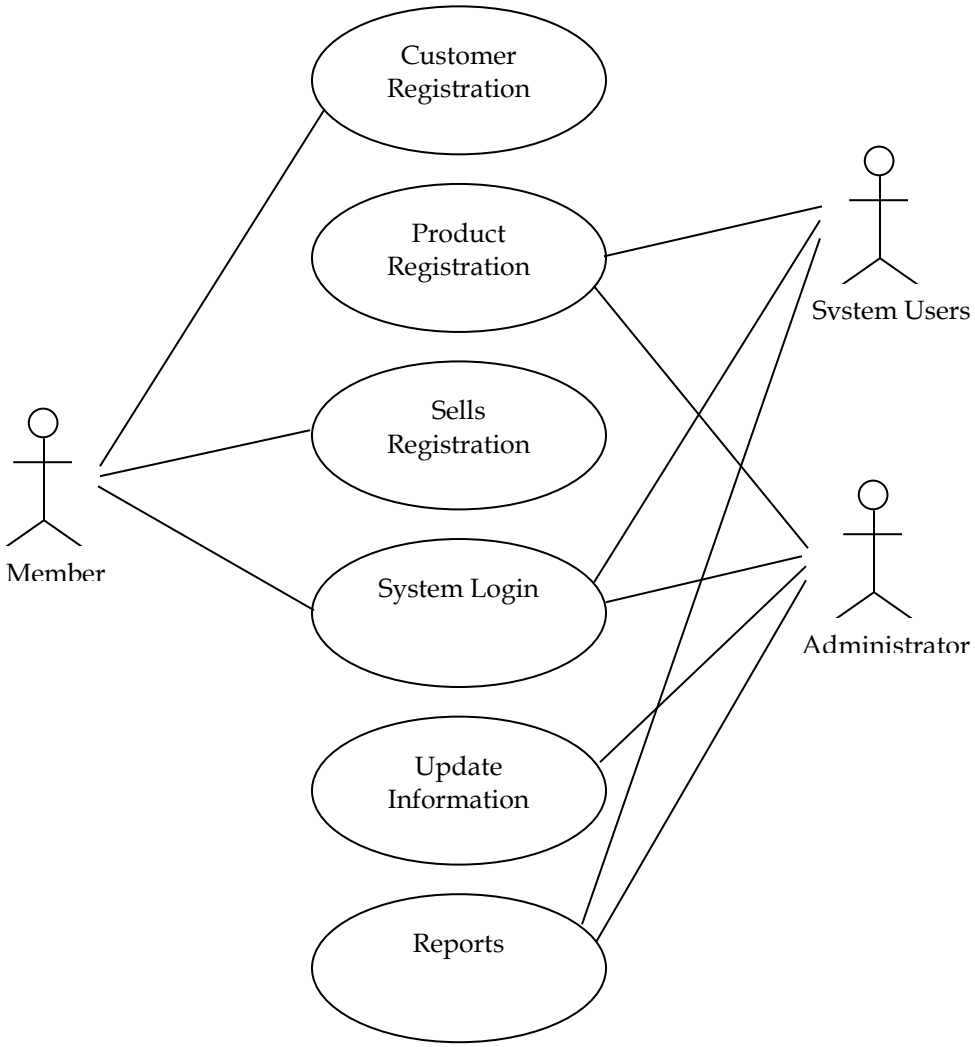
FP = Count_total * [0.65 + 0.01 * sum of F*i*]

FP=108*[0.65 + 0.01 * 54]
FP=129

7. Preliminary Design

7.1 Use Case

eAuction Application Version 1.0 will address the following use cases. The complete usage scenarios will be completed during the information-gathering process. Use cases will be created and prioritized. Selected use cases will be expanded into usage scenarios and features that are derived from both use cases and the usage scenarios, as represented in the following diagram:



[eAuction Usage Scenario – This usage scenario, or scenario for short, describes a real-world example of how one or more people or organizations interact with eAuction

system. It describe the steps, events, and/or actions which occur during the interaction. This Usage scenarios indicating exactly how someone works with the user interface, or reasonably high level describing the critical business actions but not the indicating how they're performed.]

7.2 Specification of actors

The following actors are defined so far in the analysis phase of the eAuction Application development process.

7.2.1 Member

Member	
Element	**Details**
Description	A Member is a client of the eAuction business. This can be an individual or a company.
Examples	A Member add products to shopping card and do checkout and pay cash.

7.2.2 System User

System User	
Element	**Details**
Description	System user is the person in the eAuction application who oversees the sales activity.
Examples	The System User is allowed to view all information in the system. Also his/her own information. System user can view all the reports.

7.2.3 Administrator

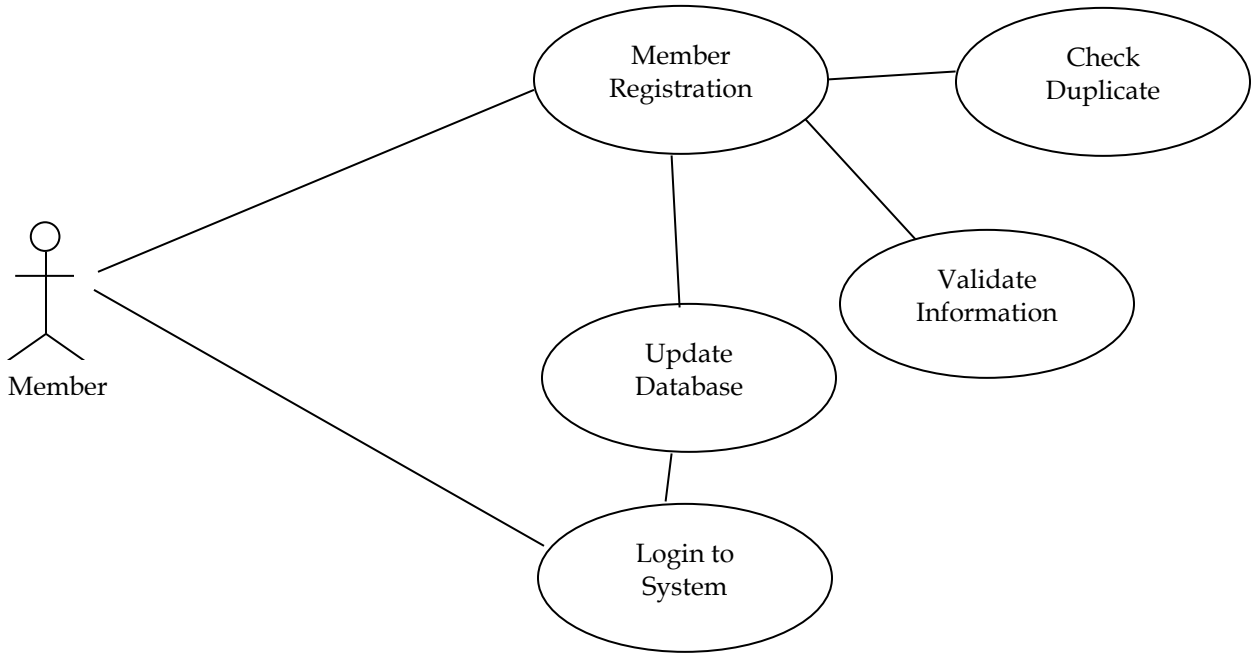
Administrator	
Element	**Details**
Description	The Administrator is the person who can update all the information.
Examples	Administrator update the product price, add new product in the eAuction application.

7.3 Specification of Use Cases

7.3.1 Use Case Member Registration

Member Registration	
Element	**Details**
Actor	Member
Trigger	Member is not registered in the System and the Member wish to register a sale.
Pre Conditions	Member's Email ID is not listed in the system. Member opens the registration page, and the registration page is displayed.
Post Conditions	Member is registered in the system, the Member is logged into the system, and the system menu is displayed.
Normal course	1. Registration form appears on the screen 2. System increment the last registered Member ID by 1 to get the ID for the new Member. 3. Member fill in the Member's information including Name, Adress(es), Phone number(s), E-mail(s), Contact person 4. System update
Alternative courses	3a. Not all mandatory data fields are filled 3a1.Member fills in the missing data fields

Use Case Diagram : Member Registration

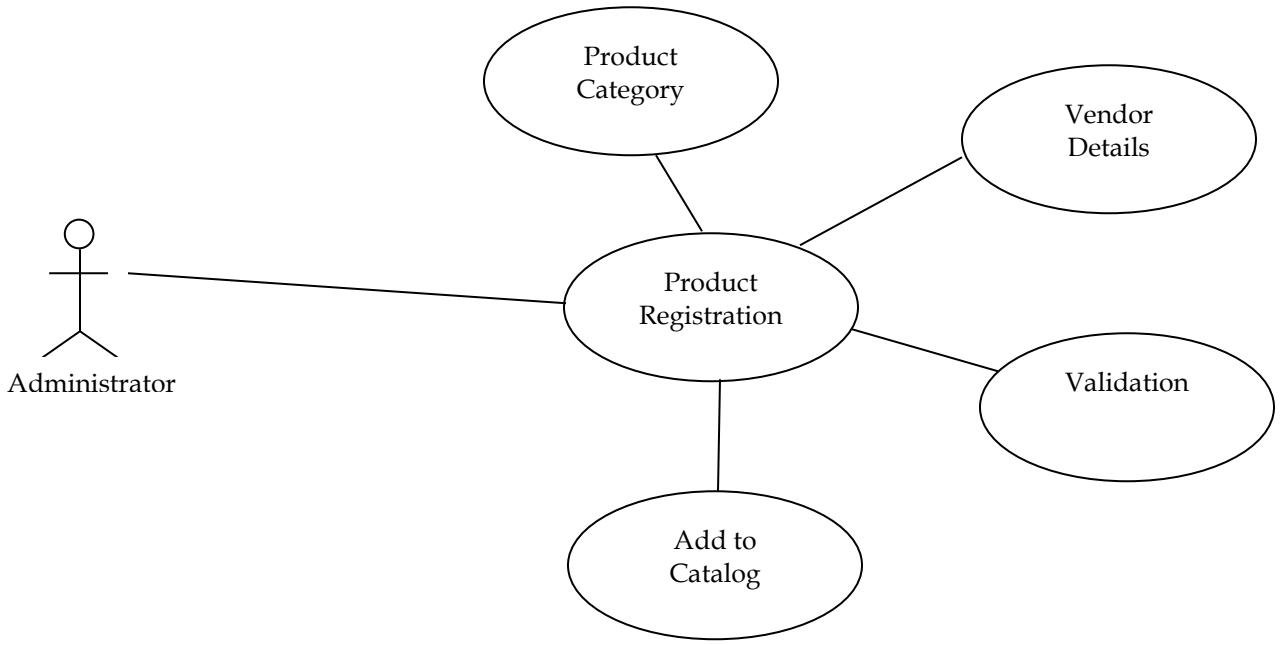


[Use Case: Member Registration – This use case scenario, or scenario for short, describes how Member will registered into eAuction System. It describes the steps, events, and/or actions which occur during the interaction. This Usage scenarios indicating exactly how someone works with the Member registration interface.]

7.3.2 Use Case. Product Registration

Product Registration	
Element	**Details**
Actor	Administrator
Trigger	The vendor have delivered the new product
Pre Conditions	The product is not in the stock, the user is logged into the system, and the system menu is displayed.
Post Conditions	The product is in the stock, the user is logged into the system, and the system menu is displayed.
Normal Event Flow	1. The administrator open the product registration form by choosing the menu item for this action 2. Administrator registers all necessary information about the product in the system. 3. Administrator update the System by confirming the data entered into the registration form.
Variations	3a. Mandatory fields in the registration form are missing 3a1. The system reject the system update with an error message about missing mandatory fields.

Use Case Diagram: Product Registration

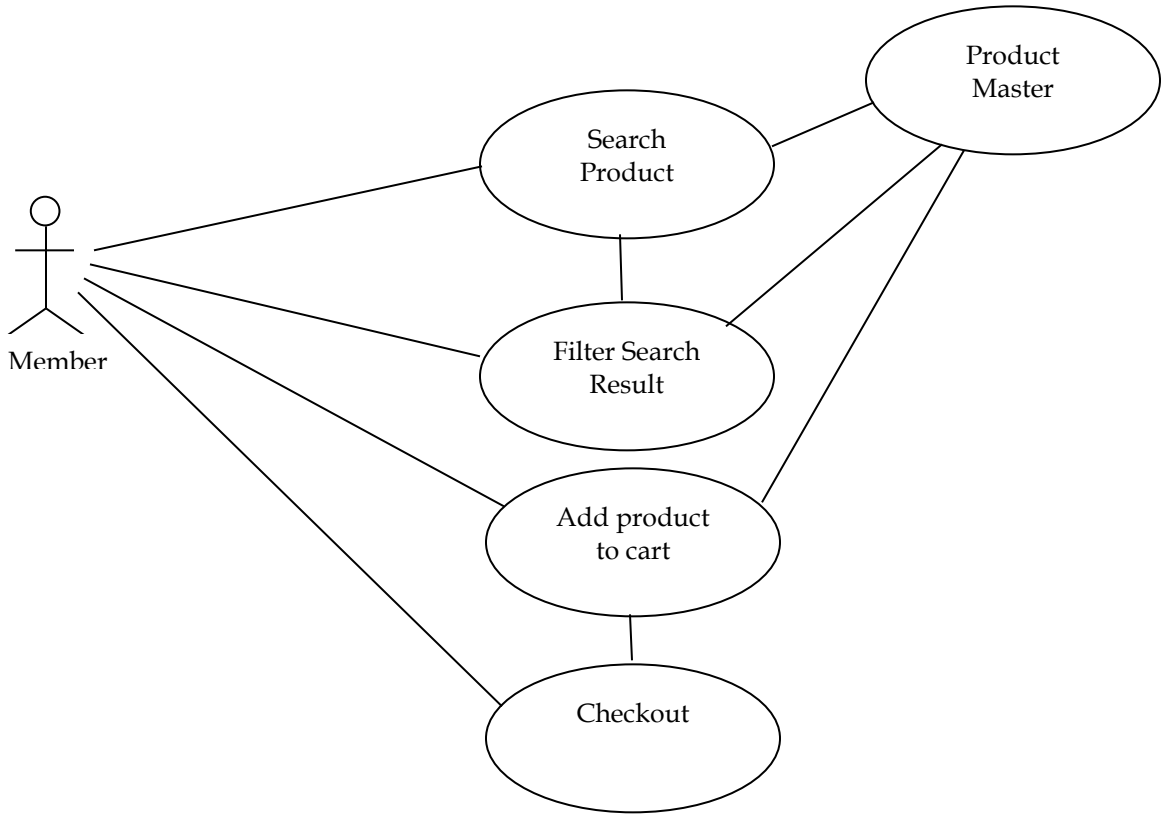


[Use Case: Product Registration – This use case scenario, or scenario for short, describes how administrator will add new product details into eAuction System. It describes the steps, events, and/or actions which occur during the interaction.]

7.3.3 Use Case. Sells Registration

Sells Registration	
Element	**Details**
Actor	Member
Trigger	Member wants to buy product
Pre Conditions	The product is in the stock, the user is logged into the system, and the system menu is displayed.
Post Conditions	The sale is registered, the checkout details are updated, the user is logged into the system, and the system menu is displayed.
Normal event flow	1. Member search for product 2. Member selects the product from list. 3. Add product to shopping cart 4. The system calculate the total of the invoice (product * quantity) 5. Member check out the shopping cart.
Variations	5a.The system check for the payment mode and card details 2a1.The system informs if found mismatch

Use Case Diagram: Sales Registration

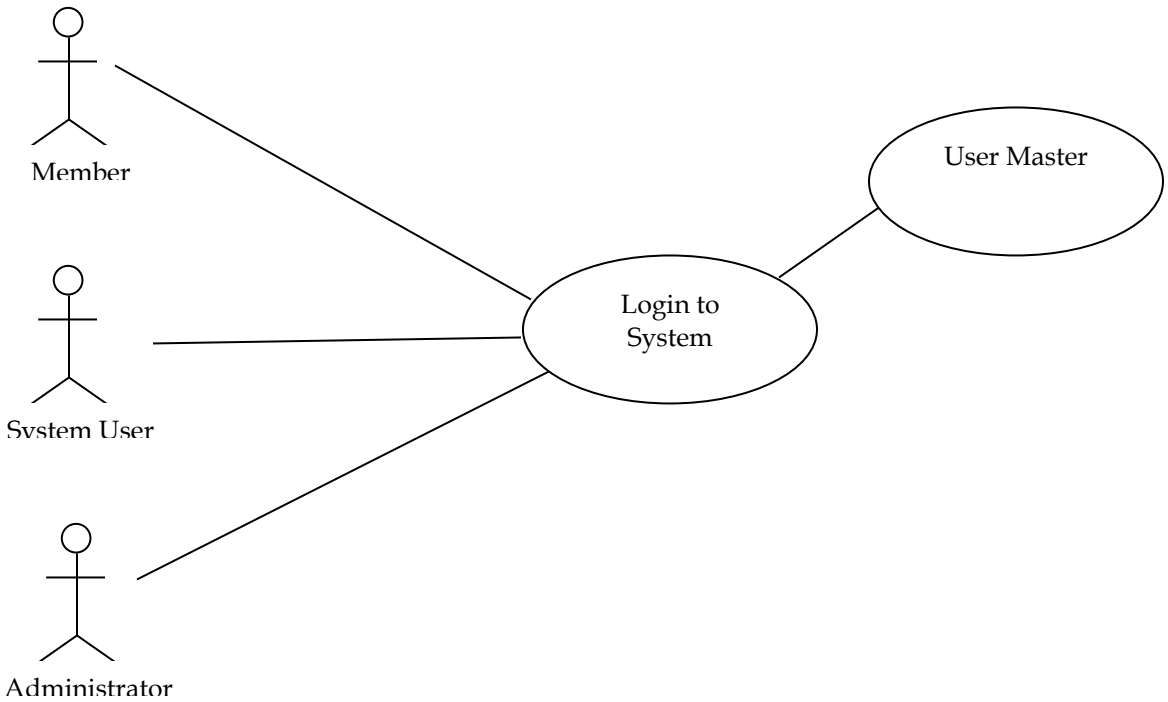


[Use Case: Sells Registration – This use case scenario, or scenario for short, describes how Member will search product, add product to shopping cart and make checkout from eAuction System. It describes the steps, events, and/or actions which occur during the interaction.]

7.3.4 Use Case. System Login

System Login	
Element	**Details**
Actor	Member, System User, Administrator
Trigger	The user wish to start using the system.
Pre Conditions	The user is not logged into the system.
Post Conditions	The user is logged into the system, and the system menu is displayed.
Normal course	1. The user click the link for the eAuction application and a login page appear on the screen. 2. The user types his username and password into the form and press the login button. 3. The system confirms that the user is logged on.
Alternative courses	2a. The user is not a valid user or the user name or the password is mistyped. 2a1. The system reject login with an error message that express wrong login name or password.

Use Case Diagram: System Login

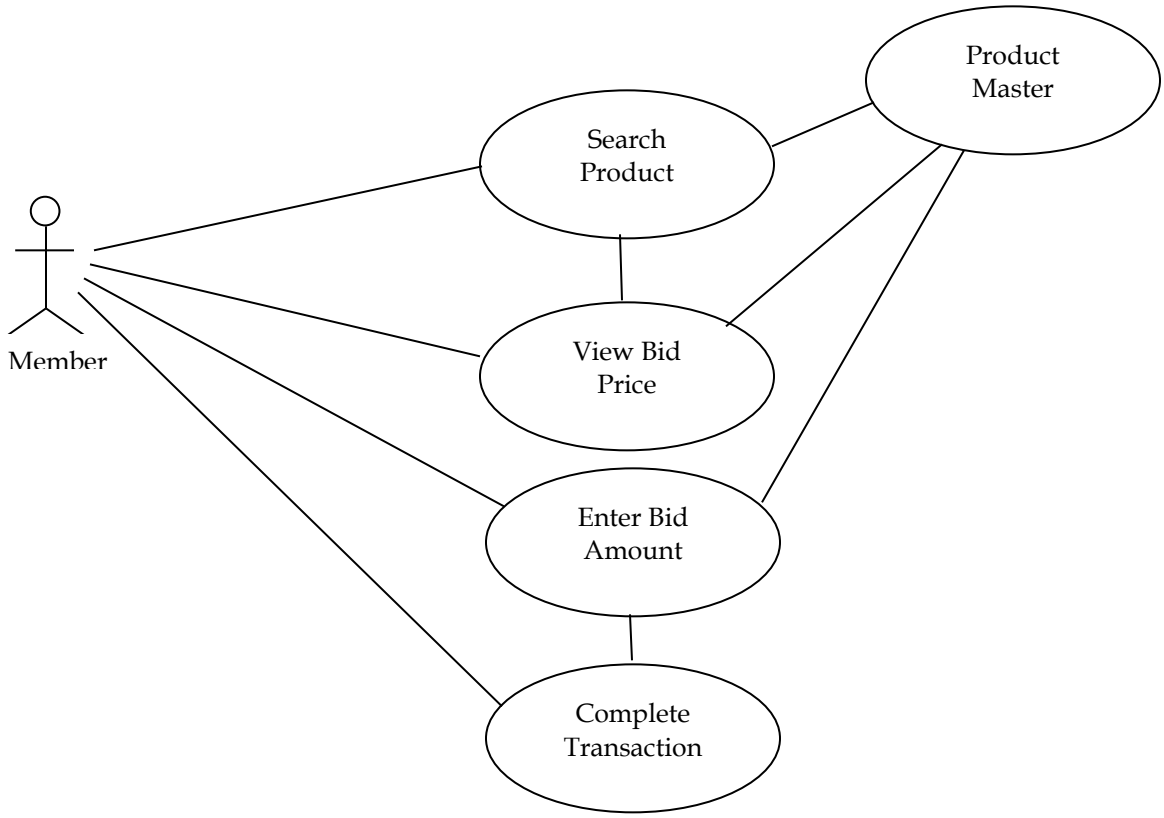


[Use Case: System Login – This use case scenario, or scenario for short, describes how actors will perform login eAuction System. It describes the steps, events, and/or actions which occur during the interaction.]

7.3.5 Use Case. Bidding

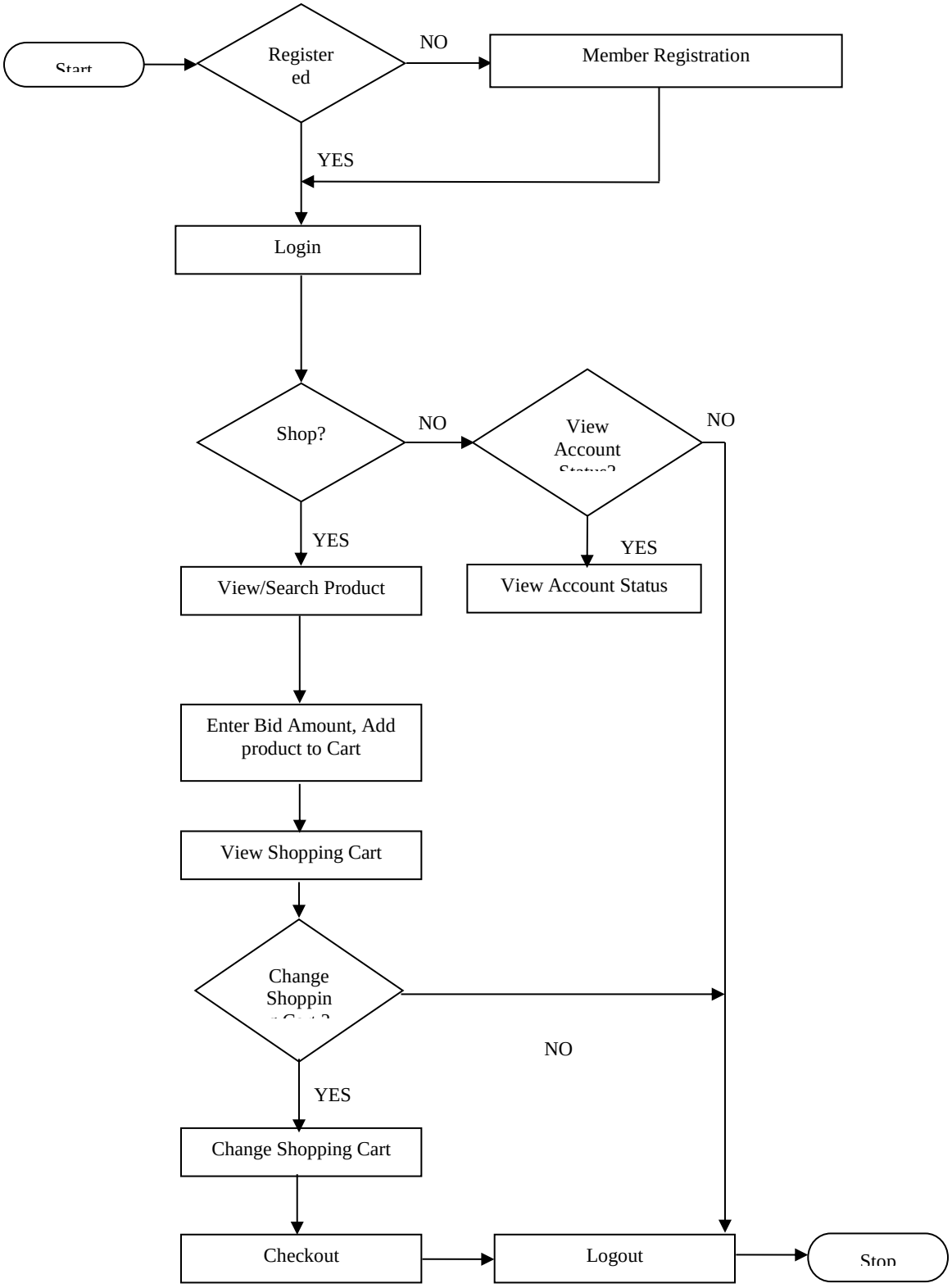
Bidding	
Element	**Details**
Actor	Member
Trigger	Member wants to bid a product
Pre Conditions	The product is in the stock, the user is logged into the system, and the system menu is displayed.
Post Conditions	Bid amount is updated.
Normal event flow	6. Member search for product 7. Member selects the product from list. 8. Member view the earlier bid amount for the product 9. Member enter the bid amount 10. The system update the bid details
Variations	5a.The system check for the biding minimum amount, expiry date 2a1.The system informs if found mismatch

Use Case Bidding

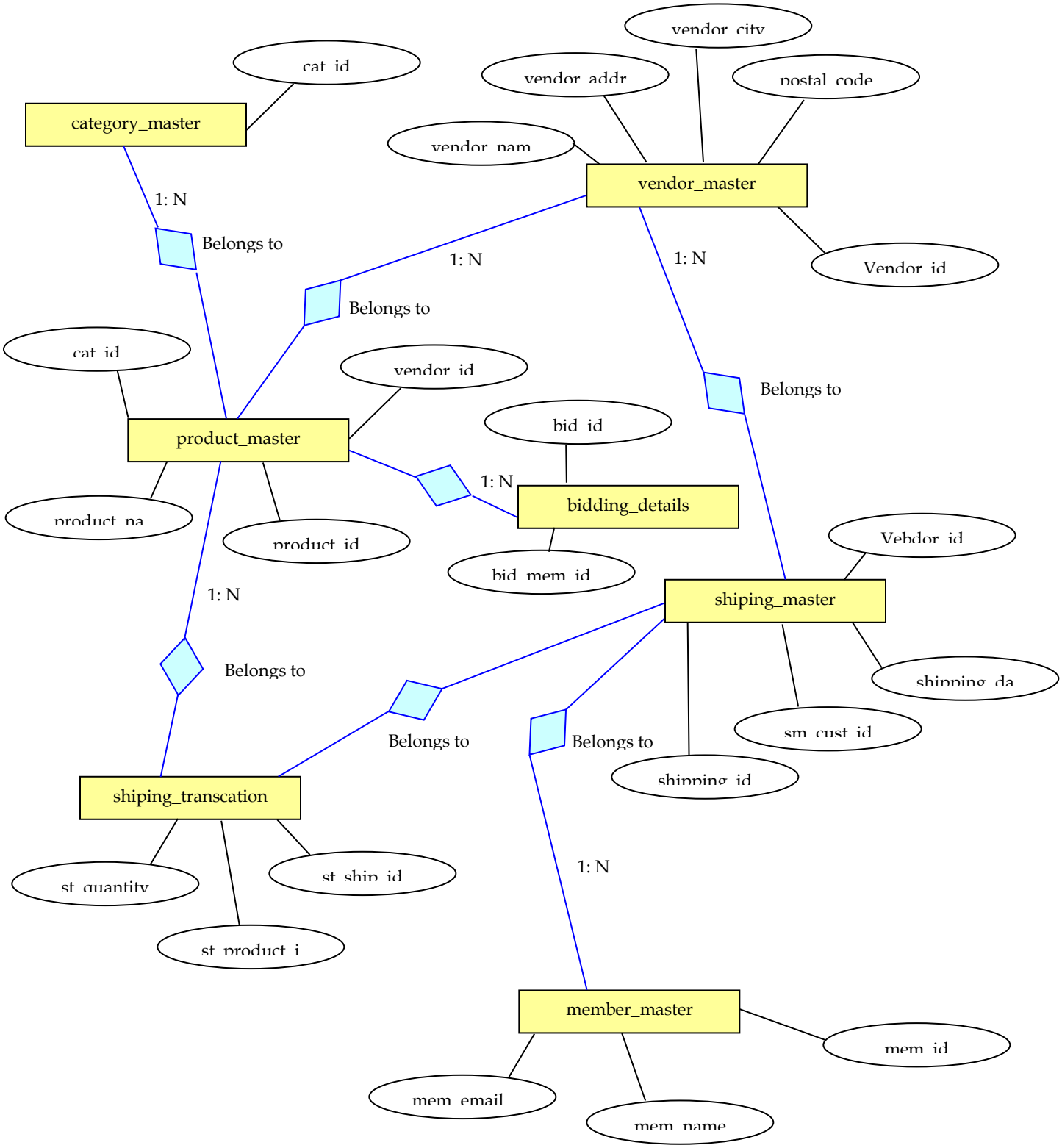


[Use Case: Bidding – This use case scenario, or scenario for short, describes how Member will bid a product in eAuction. It describes the steps, events, and/or actions which occur during the interaction.]

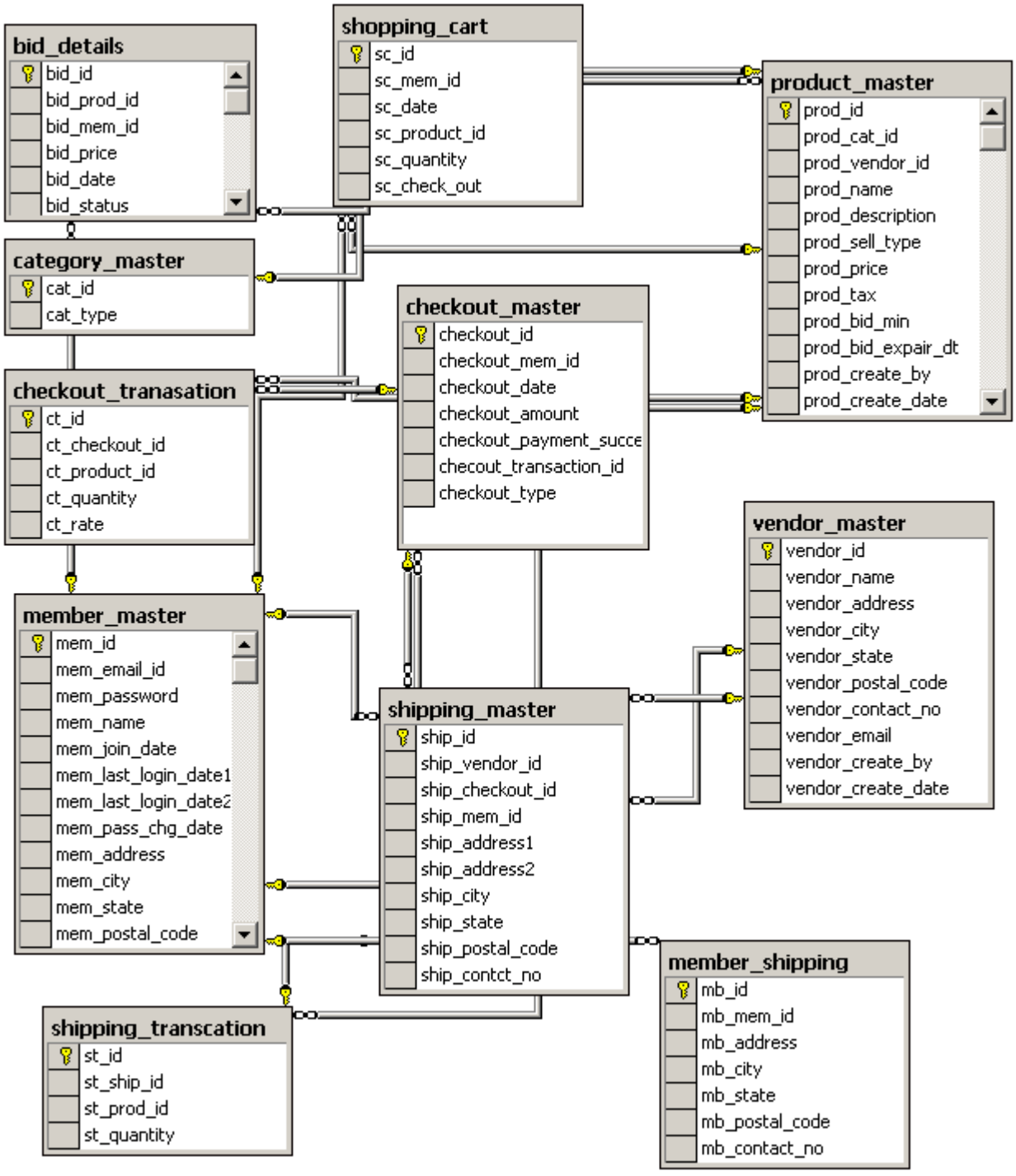
7.4 Process Flow Chart



7.5 ER Diagram



7.6 Schema Diagram



7.7 Data Dictionary

7.7.1 Table: category_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	cate_id	Int	Y	
2	cat_type	Varchar(50)		

[*category_master* table is used to store product category, *cate_id* is the primary key and used in *product_master* table as foreign key]

7.7.2 Table: ceckout_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	checkout_id	Int	Y	
2	checkout_mem_id	Int		member_master(mem_id)
3	checkout_date	Date		
4	checkout_amount	Int		
5	checkout_payment_success	Int		
6	checout_trnsaction_id	Varchar(50)		

[*chekout_master* table is used to store checkout details of the Member it stores record of member details, checkout date, amount and payment processing transaction id, *checkout_id* is the primary key and used in *checkout_transcation* table as foreign key]

7.7.3. Table: checkout_transcation

Sr.	Field Name	Type	Primary Key	Foreign Key
1	ct_id	Int	Y	
2	ct_checkout_id	Int		checkout_master(checkout_id)
3	ct_product_id	Date		product_master(prod_id)
4	ct_quantity	Int		
5	ct_rate	Int		

[*chekout_transcation* table is used to store checkout product details of the Member it stores record of checkout_id, product details, quantity and rate]

7.7.4 Table: Member_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	mem_id	Int	Y	
2	mem_name	Int		
3	mem_user_name	Varchar(50)		
4	mem_password	Varchar(50)		
5	mem_email_id	Varchar(50)		
6	mem_last_login_date1	Date		
7	mem_last_login_date2	Date		
8	mem_pass_chg_date	Date		

[Table *Member_master* is used to store Member details, it stores record of name of the Member, his email id, address and login details. *mem_id* is the primary key and represent for the Member in eAuction database]

7.7.5. Table: member_shipping

Sr.	Field Name	Type	Primary Key	Foreign Key
1	mb_id	Int	Y	
2	mb_mem_id	Int		Member_master(mem_id)
3	mb_address_1	Varchar(150)		
4	mb_address_2	Varchar(150)		
5	mb_city	Varchar(50)		
6	mb_state	Varchar(50)		
7	mb_postal_code	Varchar(6)		
8	mb_contact_no	Varchar(50)		

[Table *member_shipping* is used to store Member shipping details, it stores record of shipping address and contact details.]

7.7.6 Table: product_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	prod_id	Int	Y	
2	prod_cat_id	Int		category_master (cat_id)
3	prod_name	Varchar(50)		
4	prod_description	Varchar(150)		
5	prod_price	Int		
6	prod_tax	Int		
7	prod_create_by	Int		user_master(user_id)
8	prod_create_date	Date		

[Table *product_master* is used to store product details, it stores name of the Member, his product, product category, description, price. *prod_id* is the primary key and represent for the product in eAuction database]

7.7.7 Table: shipping_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	ship_id	Int	Y	
2	ship_vendor_id	Int		vendor_master (vendor_id)
3	ship_checkout_id	Int		checkout_master(check_id)
4	ship_mem_id	Int		Member_master(mem_id)
5	ship_address1	Varchar(100)		
6	ship_address2	Varchar(100)		
7	ship_city	Varchar(100)		user_master(user_id)
8	ship_state	Varchar(100)		
9	ship_postal_code	Varchar(6)		
10	ship_contct_no	Varchar(50)		

[Table *shipping_master* is used to store shipping master data for transaction, it stores vendor and Member details. *ship_id* is the primary key and used in shipping transaction as a foreign key.]

7.7.8 Table: shipping_trnsaction

Sr.	Field Name	Type	Primary Key	Foreign Key
1	st_id	Int	Y	
2	st_ship_id	Int		shipping_master (ship_id)
3	st_prod_id	Int		product_master(prod_id)
4	st_quantity	Int		

[Table *shipping_transaction* is used to store product details for particular shipping.]

7.7.9 Table: shopping_cart_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	sc_id	Int	Y	
2	sc_mem_id	Int		Member_master (mem_id)
3	sc_date	date		
4	sc_check_out	Int		

[Table *shopping_cart_master* is used to store Member shopping details temporarily before checkout, it stores Member details, date and checkout flag. *sc_id* is the primary key and used in *shopping_cart_transcation* as a foreign key.]

7.7.10 Table: shopping_cart_transcation

Sr.	Field Name	Type	Primary Key	Foreign Key
1	sct_id	Int	Y	
2	sct_sc_id	Int		shopping_cart_master (sc_id)
3	sct_product_id	Int		product_master(prod_id)
4	sct_quatity	Int		

[Table *shopping_cart_transaction* is used to store product details for particular shopping cart.]

7.7.11 Table: user_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	user_id	Int	Y	
2	user_name	Varchar(50)		
3	user_password	Varchar(50)		
4	user_type	Varchar(50)		

[Table *user_master* is used to store system user details, it stores login details of user including login id, password and user type. *user_id* is the primary key and used to represent user in eAuction database.]

7.7.12 Table: vendor_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	vendor_id	Int	Y	
2	vendor_name	Varchar(50)		
3	vendor_address_1	Varchar(100)		
4	vendor_address_2	Varchar(100)		
5	vendor_city	Varchar(50)		
6	vendor_state	Varchar(50)		
7	vendor_postal_code	Varchar(6)		
8	vendor_contact_no	Varchar(50)		
9	vendor_email	Varchar(50)		
10	vendor_create_by	int		user_master(user_id)
11	vendor_create_date	Date		

[Table *vendor_master* is used to store vendor details, it stores name of the vendor, address and contact details. *vendor_id* is the primary key and used in *product_master* as a foreign key.]

7.7.12 Table: bid_details

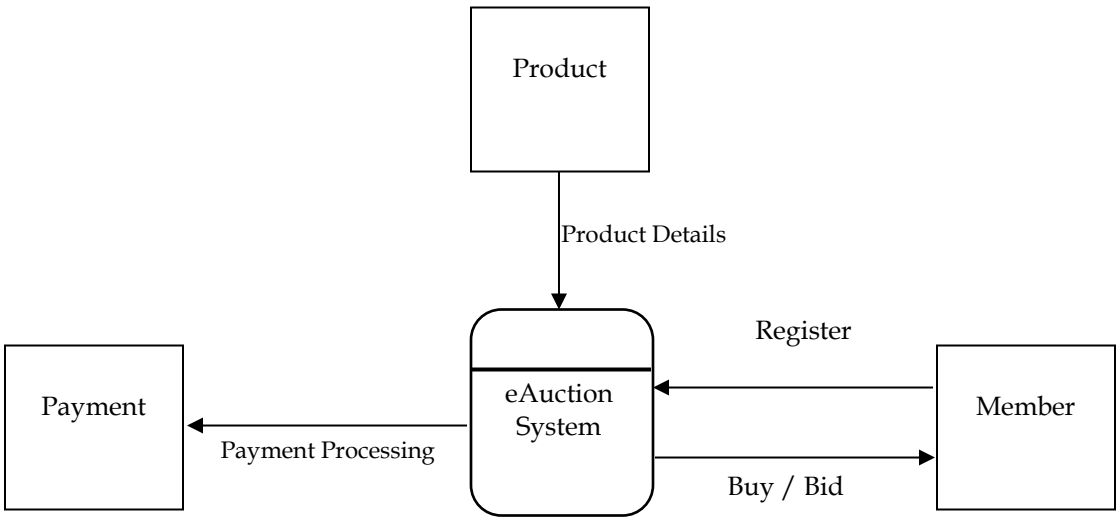
Sr.	Field Name	Type	Primary Key	Foreign Key
1	bid_id	Int	Y	
2	bid_prod_id	Int		Prod_master(pm_id)
3	bid_mem_id	Int		Member_master(mem_id)
4	bid_price	Int		
5	bid_date	datetime		
6	bid_status	Varchar(50)		
7	bid_checkout_status	Int		

[Table *bid_details* is used to store bidding details.]

7.8 Data Flow Diagram DFD

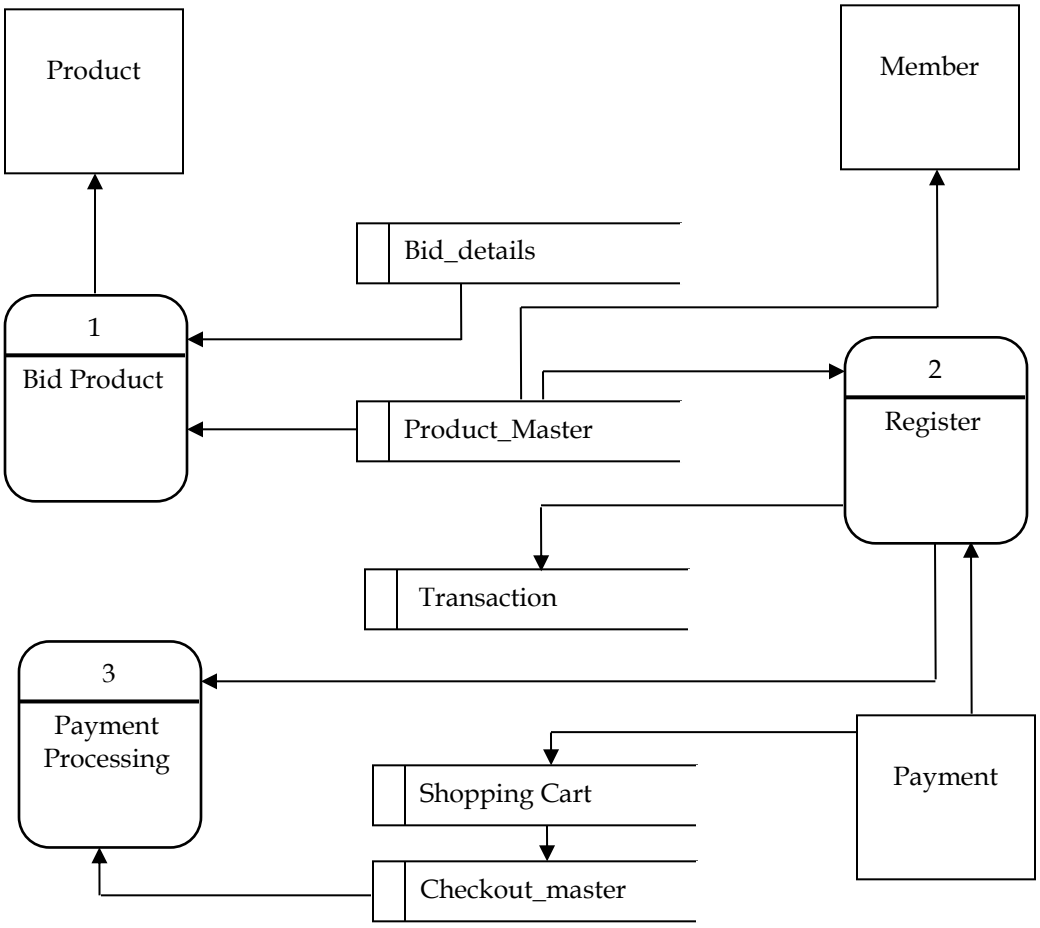
A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

7.8.1 Context Level



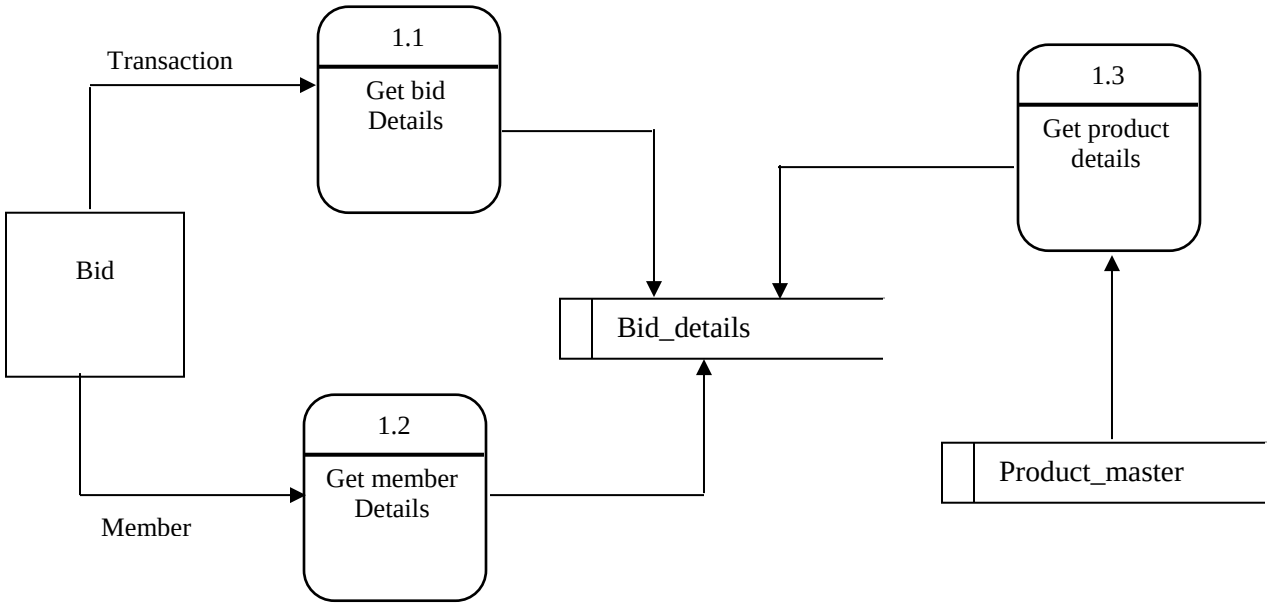
This context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modelled purely in terms of data flows across the system boundary. This context diagram shows the entire eAuction as a single process,

7.8.2 Level 1 (High Level Diagram)



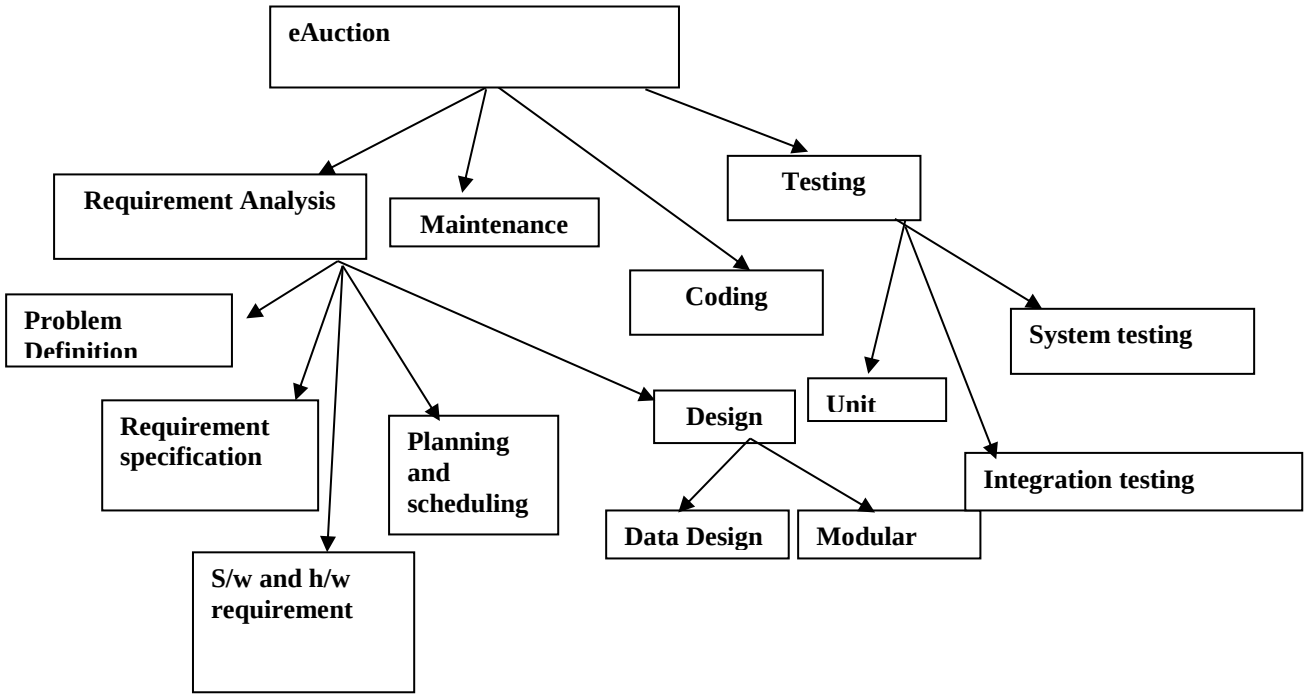
This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major high-level processes of the eAuction system and their interrelation.. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level 1, e.g. the "Buy Product" data flow could be spilt into "Add Product" and "Shopping Cart" and still be valid.

7.8.3 Level 2 (Bid Product)

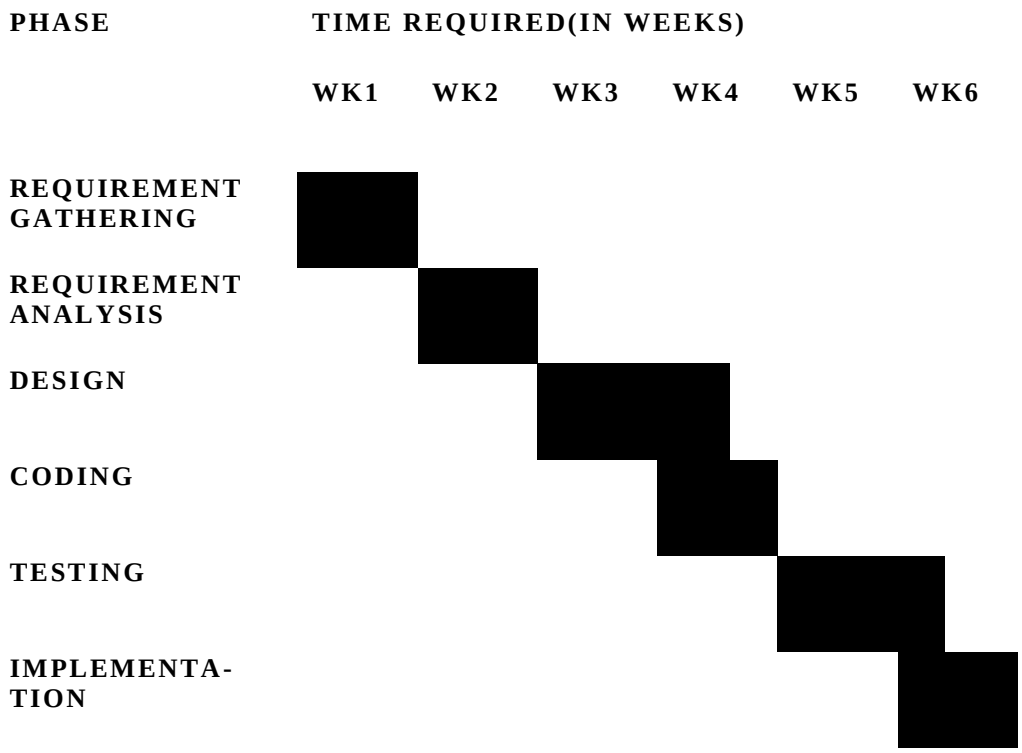


7.9 Scheduling

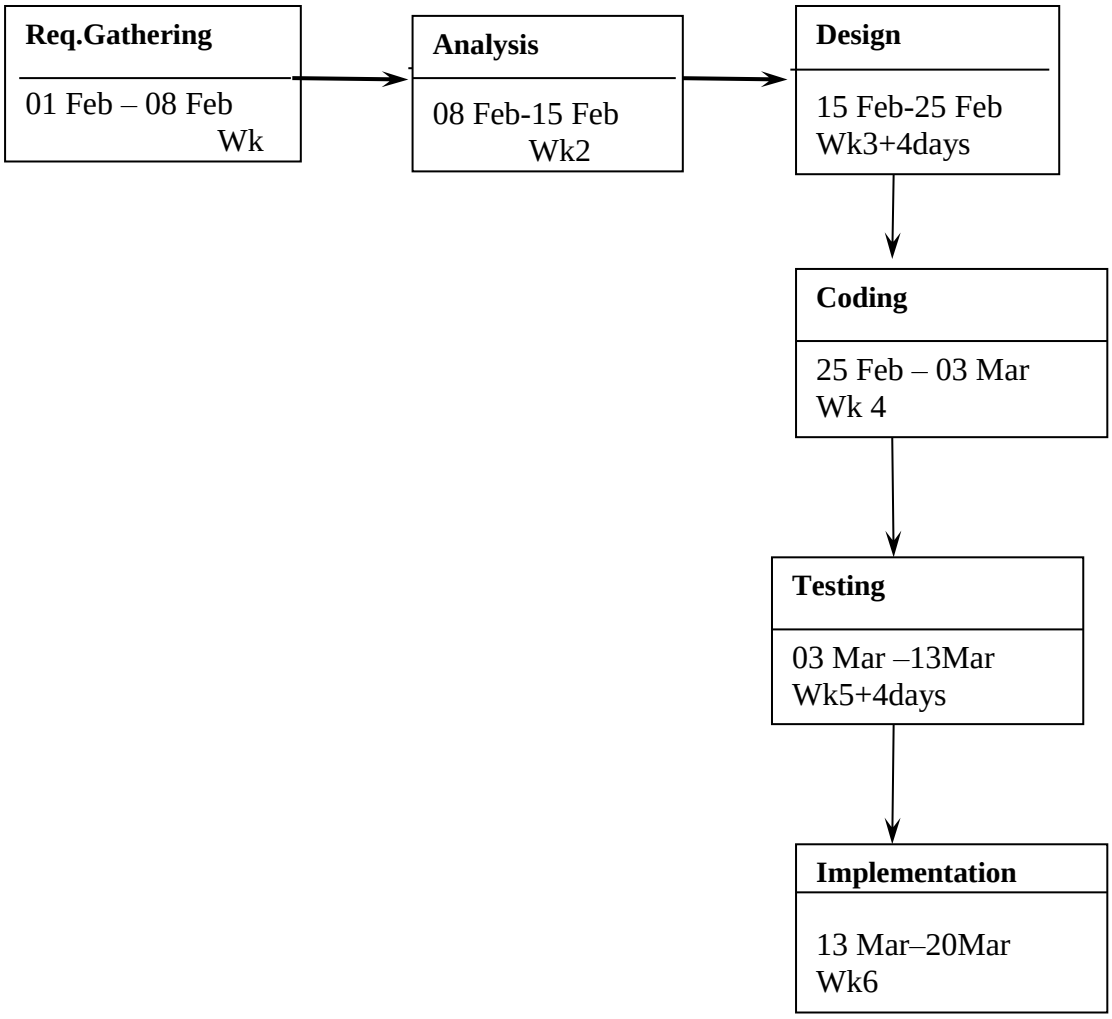
Scheduling of a project can be correlated to prioritizing various jobs with respect to their cost, time and duration. Scheduling can be done with resource constraint or time constraint in mind.



7.9.1 Gantt chart:

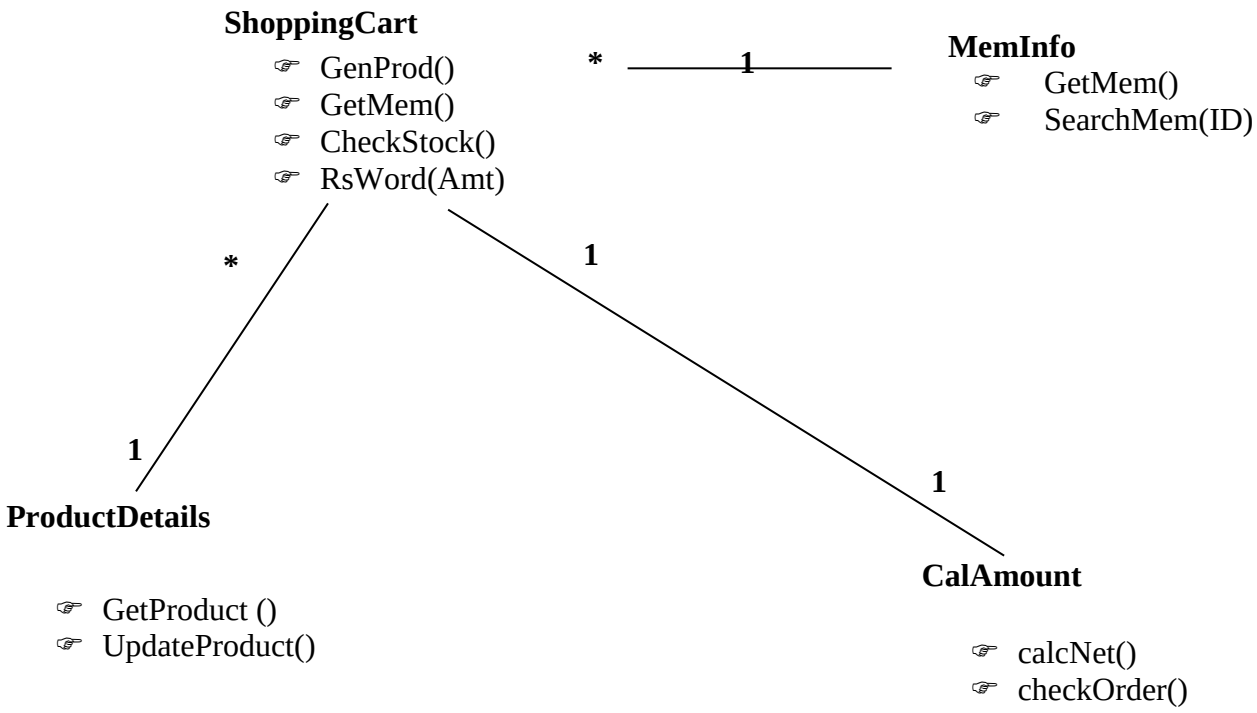


7.9.2 PERT chart:

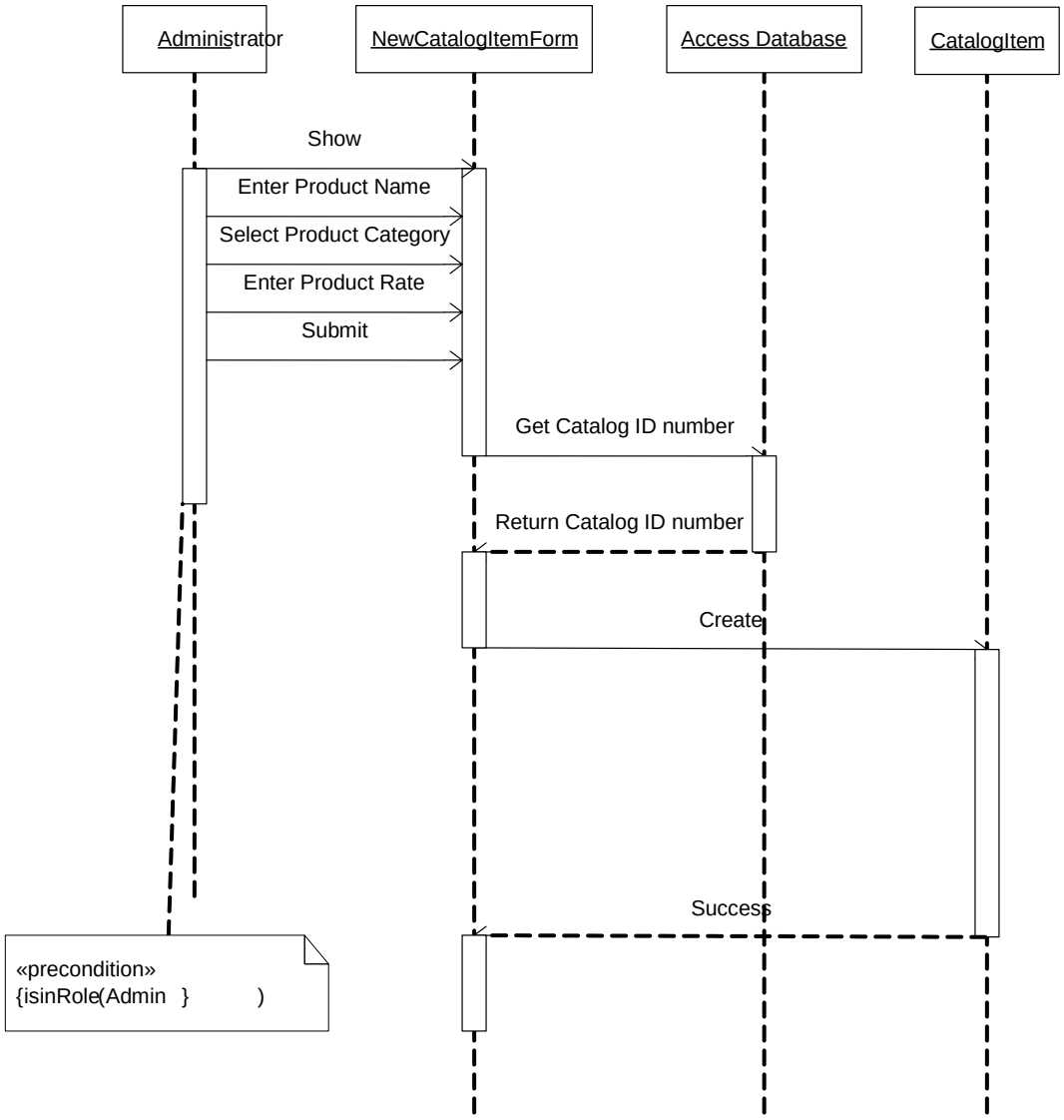


7.10 Analysis

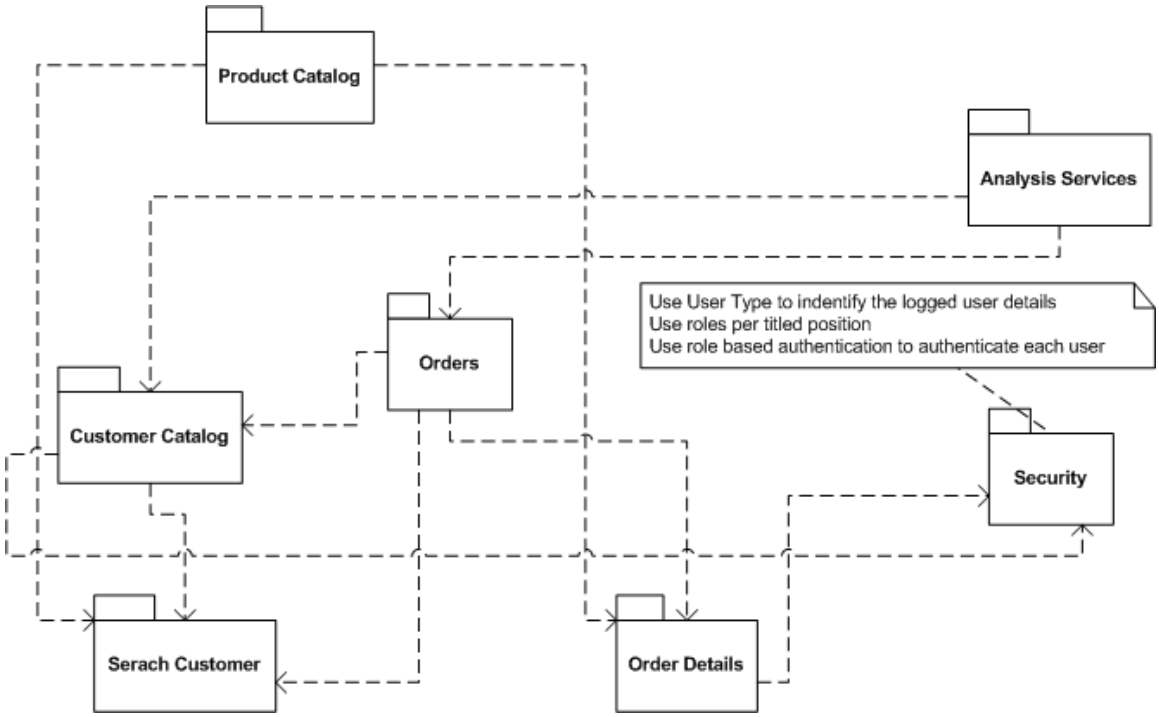
7.10.1 Class Diagram



Sequence Diagram (Add product to catalog)



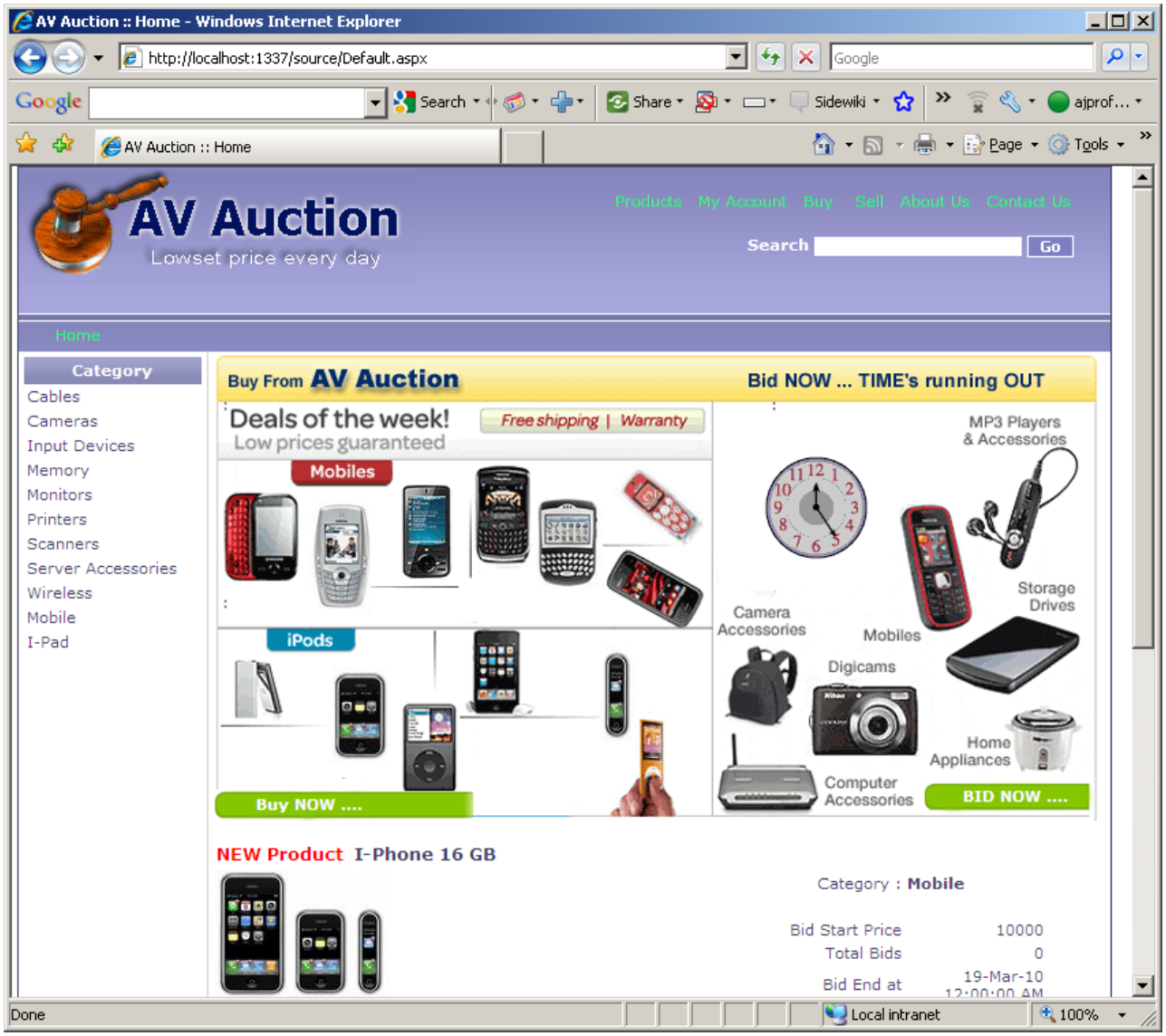
Component Diagram



8 Coding and Web Pages

8.1 eAuction Member Home Page

default.aspx



default.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Web;
```

```
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack ==true )
        {
            return ;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat=" +
dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();

        //get latest product

        string SqlStr;
        SqlStr = "select top 2 p.prod_id, p.prod_sell_type, c.cat_type,
v.vendor_name, p.prod_name, p.prod_description,p.prod_price,p.prod_tax,
p.prod_price+p.prod_tax as total ,p.prod_image,prod_bid_min, prod_bid_expair_dt
" +
        " from product_master p,category_master c, vendor_master v " +
        " where p.prod_cat_id=c.cat_id and p.prod_vendor_id=v.vendor_id  " +
        " and prod_display=1 order by prod_id DESC ";


        SqlDataAdapter da = new SqlDataAdapter(SqlStr, cn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        if (dt.Rows.Count > 0)
        {
```

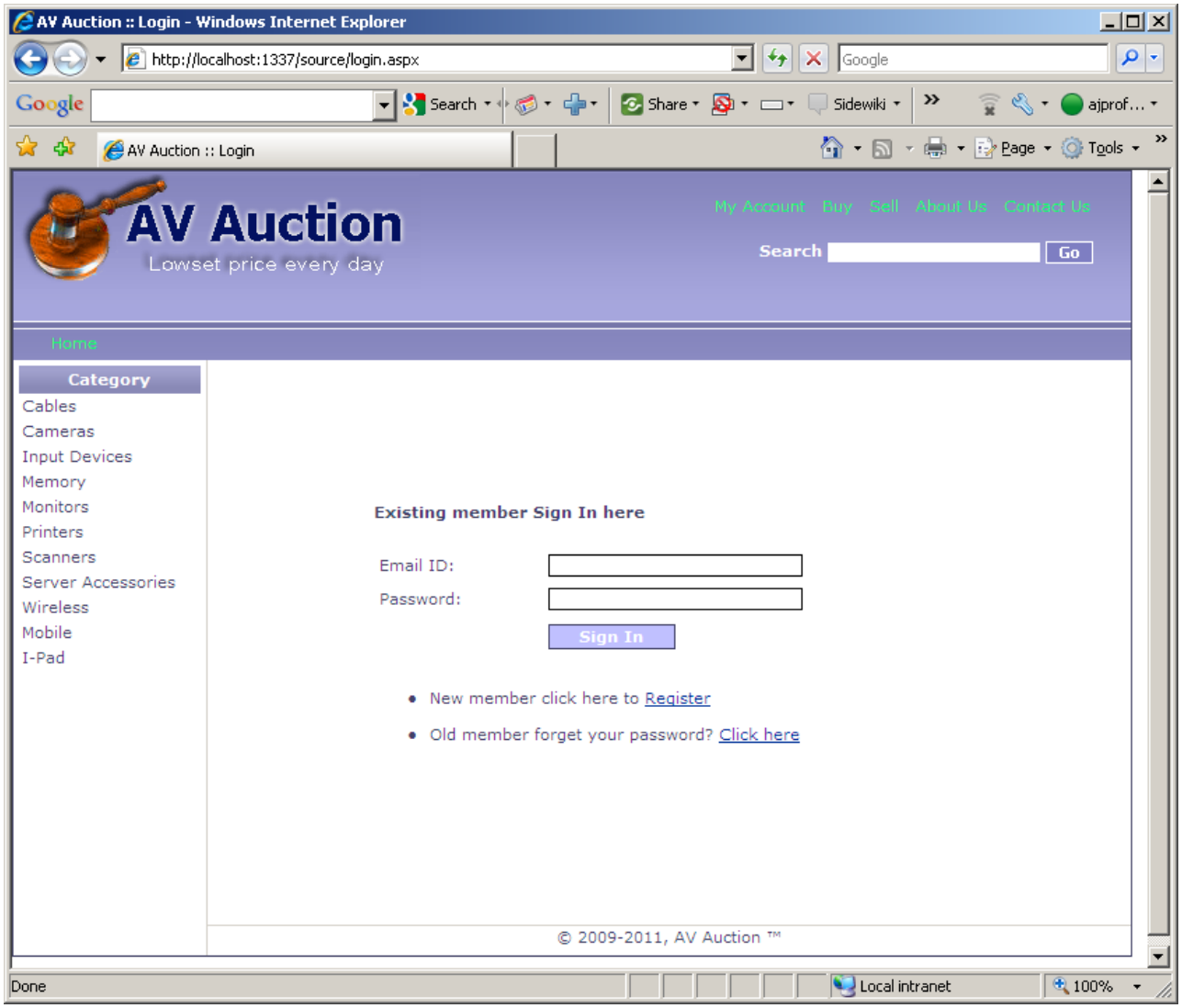
```
            string s1 = "";
            int i;
            for (i = 0; i <= dt.Rows.Count - 1; i++)
            {
                DataRow r;
                r = dt.Rows[i];
                s1 = s1 + "<table border='0' width='95%' id='table__" +
i.ToString() + "' >";
                s1 = s1 + "<tr>";
                s1 = s1 + "<td align='left' valign='top' colspan='2'>";
                s1 = s1 + "<span style='font-size: 10pt'><strong><font
color='red'>NEW Product  </font>" + r["prod_name"].ToString() +
"</strong></span></td>";
                s1 = s1 + "</tr>";
                s1 = s1 + "<tr>";
                s1 = s1 + "<td align='left' valign='top' style='width:
688px'>";
                s1 = s1 + "<img border='0' src='prod_image/" +
r["prod_image"].ToString() + "'></td>";
                s1 = s1 + "<td align='right' valign='top' width='306'>";



                if (r["prod_sell_type"].ToString() == "Buy")
                {
                    s1 = s1 + "<table border='0' id='table2_" + i.ToString() +
"' align='right' style='width: 250px'>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Category : </td>";
                    s1 = s1 + "<td width='100' align='left'><b>" +
r["cat_type"].ToString() + "</b></td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'> </td>";
                    s1 = s1 + "<td width='100'> </td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Price</td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_price"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Tax</td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_tax"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Total</td>";
                    s1 = s1 + "<td width='100' style='border-left-width: 1px;
border-right-width: 1px; border-top-style: solid; border-top-width: 1px;
border-bottom-width: 1px' align='right'>" + r["total"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>";
                    s1 = s1 + "</td>";
                    s1 = s1 + "<td align='right' style='border-top: 1px solid;
border-left-width: 1px; border-bottom-width: 1px; border-right-width: 1px'
width='149'>";
```

```
                    s1 = s1 + "</br><a
href='product_shopping_cart.aspx?cat=add&id=" + r["prod_id"].ToString() + "'>";
                    s1 = s1 + "<img src='images/buy_now.gif'
border='0'></a></td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "</table>";
                }
                else
                {
                    s1 = s1 + "<table border='0' id='table2_" + i.ToString() +
"' align='right' style='width: 250px'>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Category : </td>";
                    s1 = s1 + "<td width='100' align='left'><b>" +
r["cat_type"].ToString() + "</b></td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'> </td>";
                    s1 = s1 + "<td width='100'> </td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Bid Start Price</td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_bid_min"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Total Bids</td>";
                    //get bid count
                    SqlCommand com2 = new SqlCommand("select count(*) from
bid_details where bid_prod_id=" + r["prod_id"].ToString(), cn);
                    string StrC;
                    StrC = com2.ExecuteScalar().ToString();

                    s1 = s1 + "<td width='100' align='right'>" + StrC +
"</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Bid End at </td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_bid_expair_dt"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>";
                    s1 = s1 + "</td>";
                    s1 = s1 + "<td align='right'  width='10'>";
                    s1 = s1 + "</br><a href='bid.aspx?cat=add&id=" +
r["prod_id"].ToString() + "'>";
                    s1 = s1 + "<img src='images/bid_now.gif'
border='0'></a></td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "</table>";
                }

                s1 = s1 + "</td>";
                s1 = s1 + "</tr>";
                s1 = s1 + "<tr>";
                s1 = s1 + "<td align='left' valign='top'
colspan='2'><b>Description:</b><br>";
```

```
                s1 = s1 + r["prod_description"].ToString();
                s1 = s1 + "</td>";
                s1 = s1 + "</tr>";
                s1 = s1 + "</table>";
                s1 = s1 + "</br></br>";
            }


            sp_newproduct.InnerHtml = s1;
        }
        else
        {
            sp_newproduct.InnerHtml = "<b>No products available</b>";
        }

    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
}
```

8.2 eAuction – Member Login Page

login.aspx



login.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```

```
public partial class login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack == true)
        {
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat=" +
dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();

    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        //login and return page call
        if (TxtEmailID.Text == "")
        {
            MyClass.MyAlert(this, "Enter email id.", "123");
            return;
        }
        if (TxtPassword.Text == "")
        {
            MyClass.MyAlert(this, "Enter password.", "123");
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
```

```
        {
            cn.Open();
        }


        SqlDataAdapter da = new SqlDataAdapter("select * from member_master
where mem_email_id='" + TxtEmailID.Text + "' and mem_password='" +
TxtPassword.Text + "'" , cn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        if (dt.Rows.Count == 1)
        {
            //Alert.Show("ok");

            Session["mem_id"] = dt.Rows[0]["mem_id"].ToString();
            Session["mem_name"] = dt.Rows[0]["mem_name"].ToString();
            Session["mem_email_id"] = dt.Rows[0]["mem_email_id"].ToString();
            Session["mem_last_login"] =
dt.Rows[0]["mem_last_login_date2"].ToString();

            SqlCommand com = new SqlCommand("update member_master set
mem_last_login_date2=mem_last_login_date1, mem_last_login_date1=getdate() where
mem_id = " + dt.Rows[0]["mem_id"].ToString(), cn);
            com.ExecuteNonQuery();
            if (Session["ret_url"]==null)
            {
                Response.Redirect("myaccount.aspx");
                //Server.Transfer("myaccount.aspx");
                //this.ClientScript.RegisterStartupScript(this.GetType(),
"Alert", "<script language=\"javaScript\">" + "alert('Login Successfully!');" +
"window.location.href='default.aspx';" + "<" + "/script>");
            }
            else
            {
                Response.Redirect(Session["ret_url"].ToString());
                //this.ClientScript.RegisterStartupScript(this.GetType(),
"Alert", "<script language=\"javaScript\">" + "alert('Login Successfully!');" +
"window.location.href='default.aspx';" + "<" + "/script>");

            }
        }
        else
        {
            MyClass.MyAlert(this, "Can not login, Inavlid email id/password.",
"123");
            return;
        }

    }
}
```

8.3 eAuction – My Account

myaccount.aspx

myaccount.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
```

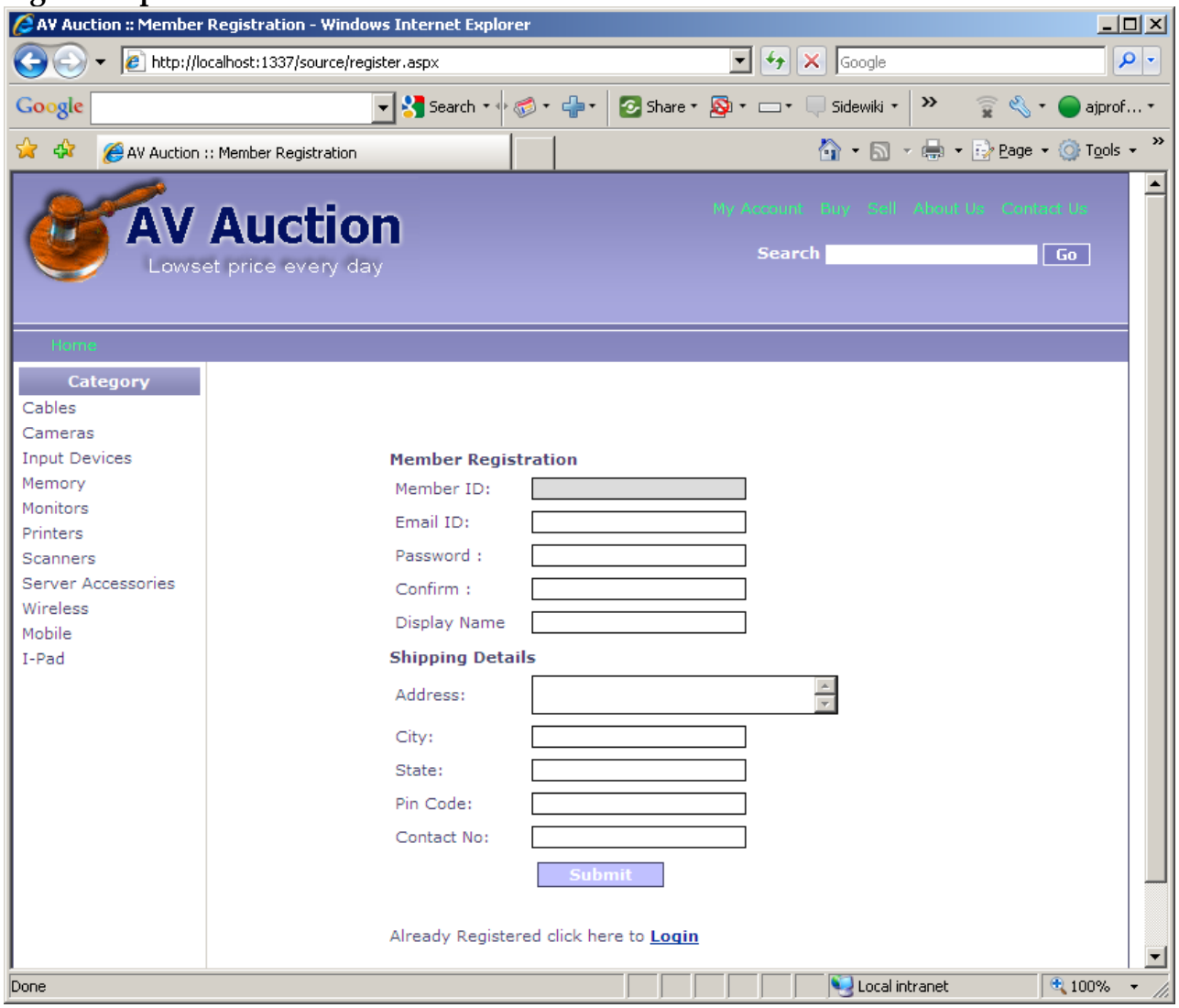
```
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class myaccount : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["mem_id"] == null)
        {
            Session["ret_url"] = "myaccount.aspx";
            Response.Redirect("login.aspx");
        }

        if (Page.IsPostBack == false)
        {
            sp_welcome.InnerHtml = "Welcome back,<strong> " +
Session["mem_name"].ToString() + ".  </strong>Your last login date was
<strong> " + Session["mem_last_login"].ToString() + "<br />";
            string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = connectionInfo;
            if (cn.State == ConnectionState.Closed)
            {
                cn.Open();
            }

            //get category
            string StrCat = "";
            SqlDataReader dr;
            SqlCommand com = new SqlCommand();
            com.CommandText = "GetCategory";
            com.CommandType = CommandType.StoredProcedure;
            com.Connection = cn;
            dr = com.ExecuteReader();
            while (dr.Read() == true)
            {
                StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat="
+ dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
            }

            sp_cat.InnerHtml = StrCat;
            dr.Close();


            //get the transcation
            string StrSql = "";
            StrSql = "select checkout_id as TranID,checkout_date as
[Date],checkout_amount as Amount,checout_transaction_id as [Payment ID]from
checkout_master where checkout_mem_id= " + Session["mem_id"].ToString();
            SqlDataAdapter da = new SqlDataAdapter(StrSql, cn);
            DataTable dt = new DataTable();
            da.Fill(dt);

            sp_tran.InnerHtml =  "Chechout Transcation as on<strong> " +
System.DateTime.Now.ToString() +".      " +
dt.Rows.Count.ToString() + " </strong>Records Found<strong>.</strong>";
```

```
            ViewState["d_t"]=dt;
            Dg1.DataSource =ViewState["d_t"];
            Dg1.DataBind();


            //get the bid transcation

            StrSql = "select checkout_id as TranID,checkout_date as
[Date],checkout_amount as Amount,checout_transaction_id as [Payment ID]from
checkout_master where checkout_mem_id= " + Session["mem_id"].ToString();

            StrSql = " SELECT bid_details.bid_id AS [Bid ID],
product_master.prod_name AS Product, product_master.prod_bid_min AS [Min
Price],  bid_details.bid_price AS [Your Bid Price], bid_details.bid_date AS
[Bidding Date], bid_details.bid_status AS Status " +
                     " FROM  bid_details INNER JOIN product_master ON
bid_details.bid_prod_id = product_master.prod_id and bid_mem_id =" +
Session["mem_id"].ToString() + " order by bid_details.bid_id  desc ";
            SqlDataAdapter da2 = new SqlDataAdapter(StrSql, cn);
            DataTable dt2 = new DataTable();
            da2.Fill(dt2);

            sp_bid.InnerHtml = "Bids as on<strong> " +
System.DateTime.Now.ToString() + ".      " +
dt.Rows.Count.ToString() + " </strong>Records Found<strong>.</strong>";
            ViewState["d_t2"] = dt2;
            Dg2.DataSource = ViewState["d_t2"];
            Dg2.DataBind();


        }
    }
    protected void Dg1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg1.DataSource = ViewState["d_t"];
        Dg1.DataBind();
        Dg1.PageIndex = e.NewPageIndex;
    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
    protected void Dg2_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void Dg2_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowIndex < 0)
        {
            return;
        }
        if (e.Row.Cells[5].Text == "WON")
        {
            string s1 = "";// e.Row.Cells[0].Text;
            s1 = "<font color='red'>WON</font> <a href='checkout_bid.aspx?id="
+ e.Row.Cells[0].Text + "'><font color='black'><u>Check OUT</u></font></a>";
```

```
            e.Row.Cells[5].Text = s1;
        }
    }
}
```

8.4 eAuction – Member Registration and Edit Profile

register.aspx



register.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class register : System.Web.UI.Page
{
```

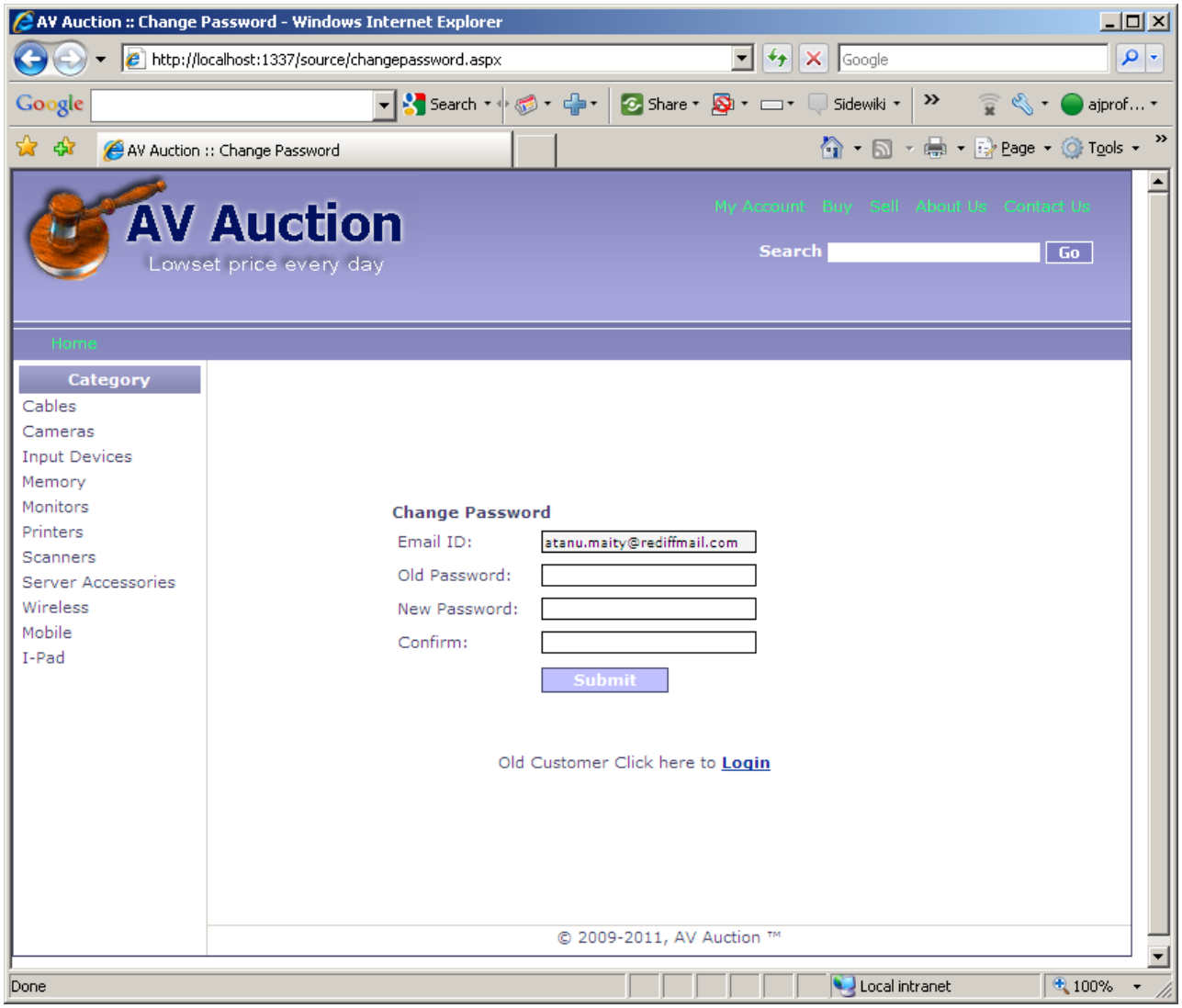
```
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack == true)
        {
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat=" +
dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();


        //if user logged in show data
        if (Session["mem_id"] != null)
        {
            SqlDataAdapter da = new SqlDataAdapter("select * from member_master
where mem_id =" + Session["mem_id"].ToString(), cn);
            SqlCommandBuilder cb = new SqlCommandBuilder(da);
            DataTable dt=new DataTable();
            da.Fill(dt);
            DataRow r;
            if (dt.Rows.Count > 0)
            {
                r = dt.Rows[0];
                TxtMemID.Text = r["mem_id"].ToString();
                TxtEmailID.Text=r["mem_email_id"].ToString() ;
                TxtEmailID.Enabled =false ;
                TxtPassword.Enabled =false ;
                TxtConfirmPassword.Enabled =false ;

                TxtDisplayName.Text = r["mem_name"].ToString();
                TxtAddress.Text = r["mem_address"].ToString();
                TxtCity.Text = r["mem_city"].ToString();
                TxtState.Text = r["mem_state"].ToString();
                TxtPin.Text=r["mem_postal_code"].ToString() ;
                TxtContact.Text=r["mem_contact_no"].ToString() ;
```

```
            }
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        //validation
        if (TxtEmailID.Text == "")
        {
            MyClass.MyAlert(this, "Enter email id.", "123");
            return ;
        }
        if (TxtDisplayName.Text =="")
        {
            MyClass.MyAlert(this, "Enter display name.", "123");
            return;
        }


        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }
        bool ch;
        string s2;
        if (TxtMemID.Text =="")
        {
            s2 = "0";
        }
        else
        {
            s2=TxtMemID.Text;
        }
        SqlDataAdapter da = new SqlDataAdapter("select * from member_master
where mem_id ="+s2, cn);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        DataTable dt=new DataTable();
        da.Fill(dt);
        DataRow r;
        if (dt.Rows.Count <= 0)
        {
            if (TxtPassword.Text   == "")
            {
                MyClass.MyAlert(this, "Enter password.", "123");
                return;
            }
            if (TxtPassword.Text == TxtConfirmPassword.Text )
            {
            }
            else
            {
                MyClass.MyAlert(this, "Confirm password are not match, retype
the password.", "123");
                TxtPassword.Text ="";
                TxtConfirmPassword.Text="";
```

```
                return;
            }

            ch=true ;
            int lastid;
            SqlCommand com = new SqlCommand("select max(mem_id) from
member_master", cn);
            lastid = int.Parse(com.ExecuteScalar().ToString()) + 1;
            r = dt.NewRow();
            r["mem_id"] = lastid;
            r["mem_email_id"] = TxtEmailID.Text;
            r["mem_password"] = TxtPassword.Text;
            r["mem_join_date"]=System.DateTime.Now.ToString();
        }
        else
        {
            ch = false;
            r = dt.Rows[0];
        }
        r["mem_name"] = TxtDisplayName.Text;
        r["mem_address"] = TxtAddress.Text;
        r["mem_city"] = TxtCity.Text;
        r["mem_state"] = TxtState.Text;
        r["mem_postal_code"] = TxtPin.Text;
        r["mem_contact_no"] = TxtContact.Text;

        if (ch == true)
        {
            dt.Rows.Add(r);
        }
        da.Update(dt);

        if (ch == true)
        {
            //MyClass.MyAlert(this, "Registration is completed.", "123");
            //Alert.Show("Registration is completed");
            //Response.Redirect("login.aspx");
            this.ClientScript.RegisterStartupScript(this.GetType(), "Alert",
"<script language=\"javaScript\">" + "alert('Registration Successfully!');" +
"window.location.href='login.aspx';" + "<" + "/script>");
        }
        else
        {
            //MyClass.MyAlert(this, "Profile edit is completed.", "123");
            this.ClientScript.RegisterStartupScript(this.GetType(), "Alert",
"<script language=\"javaScript\">" + "alert('Profile edit is completed!');" +
"window.location.href='myaccount.aspx';" + "<" + "/script>");
            //Response.Redirect("myaccount.aspx");
        }
    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
}
```

8.5 eAuction – Change Password

changepassword.aspx



changepassword.aspx.cs

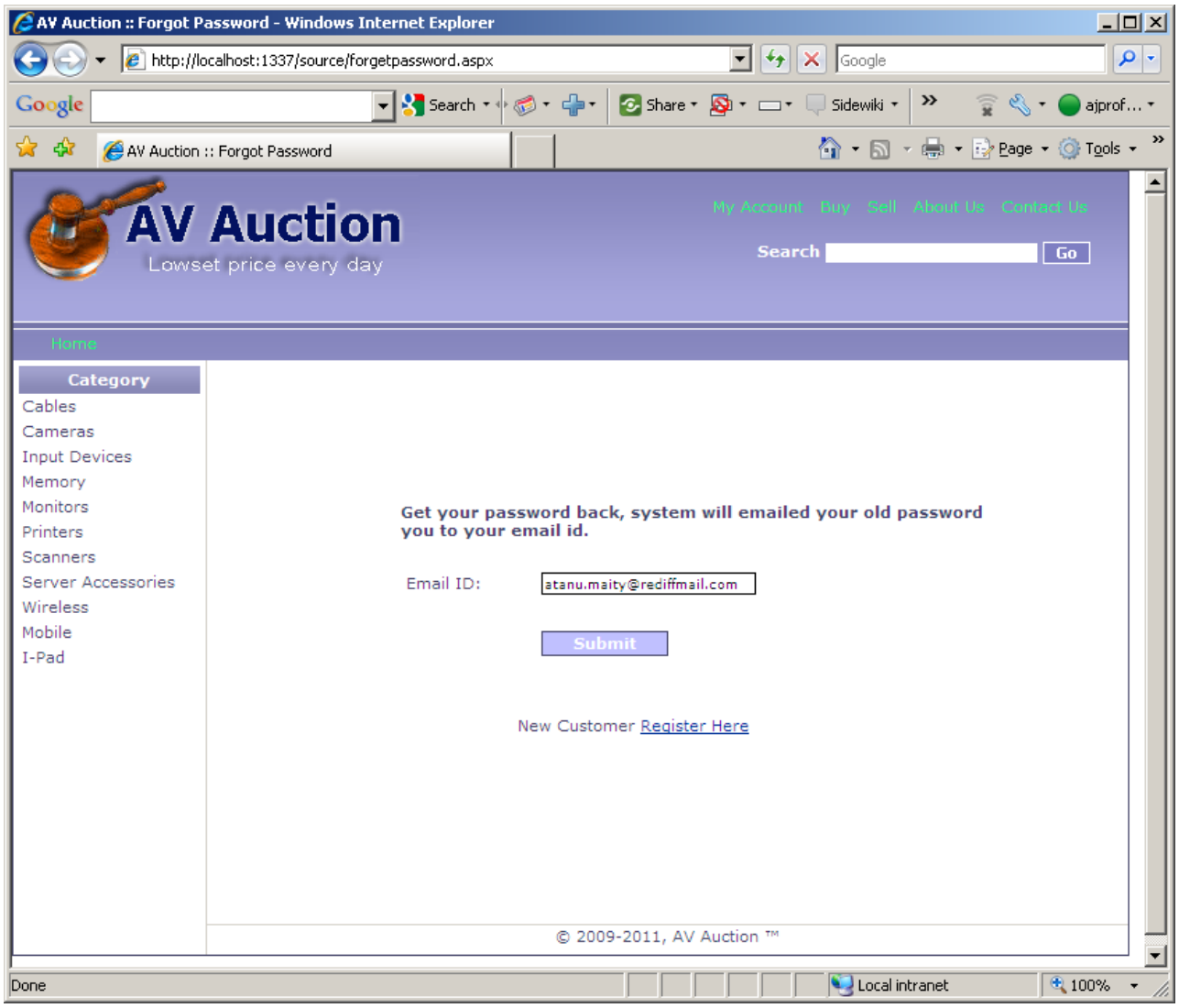
```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```

```
public partial class changepassword : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["mem_id"] == null)
        {
            Response.Redirect("login.aspx");
        }

        if (Page.IsPostBack == true)
        {
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='products.aspx?cat="
+ dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();

        TxtEmailID.Text = Session["mem_email_id"].ToString();
    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        //validation
        if (TxtEmailID.Text == "")
        {
            MyClass.MyAlert(this, "Enter email id.", "123");
            return;
        }
        if (TxtOldPassword.Text == "")
        {
            MyClass.MyAlert(this, "Enter old password.", "123");
            return;
```

```
        }

        if (TxtNewPassword.Text == "")
        {
            MyClass.MyAlert(this, "Enter new password.", "123");
            return;
        }
        if (TxtNewPassword.Text != TxtConfirmPassword.Text)
        {
            MyClass.MyAlert(this, "New and confirm Password are not match.",
"123");
            return;
        }

        //check for old password
        //check login
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;


        DataTable table = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter("select * FROM Member_MASTER
WHERE mem_EMAIL_ID='" + TxtEmailID.Text + "' and mem_password='" +
TxtOldPassword.Text + "'", cn);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        da.Fill(table);
        if (table.Rows.Count == 1)
        {

            DataRow r;
            r = table.Rows[0];
            r["mem_password"] = TxtNewPassword.Text;

            da.Update(table);


            MyClass.MyAlert(this, "Password changes successfully.", "123");


        }
        else
        {
            MyClass.MyAlert(this, "Login Failed, check your email id and
password.", "123");
            return;
        }
    }

}
```

8.6 eAuction – Forget Password

forgetpassword.aspx



forgetpassword.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.Mail;
using System.Web.UI.HtmlControls;
```

```
public partial class forgetpassword : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack == true)
        {
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='products.aspx?cat="
+ dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        SqlDataAdapter da = new SqlDataAdapter("select * from Member_master
where mem_email_id ='"+ TxtEmailID.Text + "'" , cn);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        DataTable dt=new DataTable();
        da.Fill(dt);
        if (dt.Rows.Count == 1)
        {
            Session["forget_email"] = TxtEmailID.Text;
            String StrToEmail;
            String StrMemName;
            MailMessage msgMail = new MailMessage();
```

```
            StrToEmail = dt.Rows[0]["mem_email_id"].ToString();
            StrMemName = dt.Rows[0]["mem_name"].ToString();

            msgMail.To = StrToEmail;

            msgMail.From = "webmaster@avauction.com";
            msgMail.Subject = "Login Password for hardwarestore.com";

            msgMail.BodyFormat = MailFormat.Html;


            String strBody = "";

            strBody = strBody + "";

            strBody = strBody + "<p>Hi " + StrMemName + ",</p>";
            strBody = strBody + "<p>Your password for <a
href='http://www.onlinesale.com'>";
            strBody = strBody + "http://www.hardwarestore.com</a> is </p>";
            strBody = strBody + "<p>" + dt.Rows[0]["mem_password"].ToString() +
"</p>";
            strBody = strBody + "<p>For login to onlinesale.com ";
            strBody = strBody + "<a target='_blank'
href='http://www.hardwarestore.com/dafault.aspx'>";
            strBody = strBody + "click </a> here .</p>";
            strBody = strBody + "<p>From</p>";
            strBody = strBody + "<p>Webmaster</p>";
            strBody = strBody + "<p>hardwarestore.com</p>";
            strBody = strBody + "<hr>";
            strBody = strBody + "<p>It is an auto generated email do not reply
to this email. <br>";
            strBody = strBody + "For any query send email to <a
href='mailto:am@hardwarestore.com'>";
            strBody = strBody + "reply@avauction.com</a> </p>";


            msgMail.Body = strBody;

            SmtpMail.Send(msgMail);

            MyClass.MyAlert(this, "Old password has been send to your email id,
check your mail and login again.", "123");
        }
        else
        {
            MyClass.MyAlert(this, "Email ID not found, unable to get the
password", "123");
            return;
        }
    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("products.aspx?search=" + TxtSearch.Text);
    }
}
```

8.7 eAuction – Buy/Bid

buy.aspx

product.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```

```
public partial class buy : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {


        if (Page.IsPostBack == true)
        {
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat=" +
dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();

        //get latest product

        string qr;
        qr = " ";
        if (Request.QueryString.Count  >0 )
        {
            if (Request.QueryString.Keys[0].ToString() == "cat")
            {
                qr = " and c.cat_type='" + Request.QueryString[0].ToString() +
"' ";
            }
            else
            {
                qr = " and p.prod_name like '%" +
Request.QueryString[0].ToString() + "%' ";
            }
        }
        else
        {
            qr = " ";
```

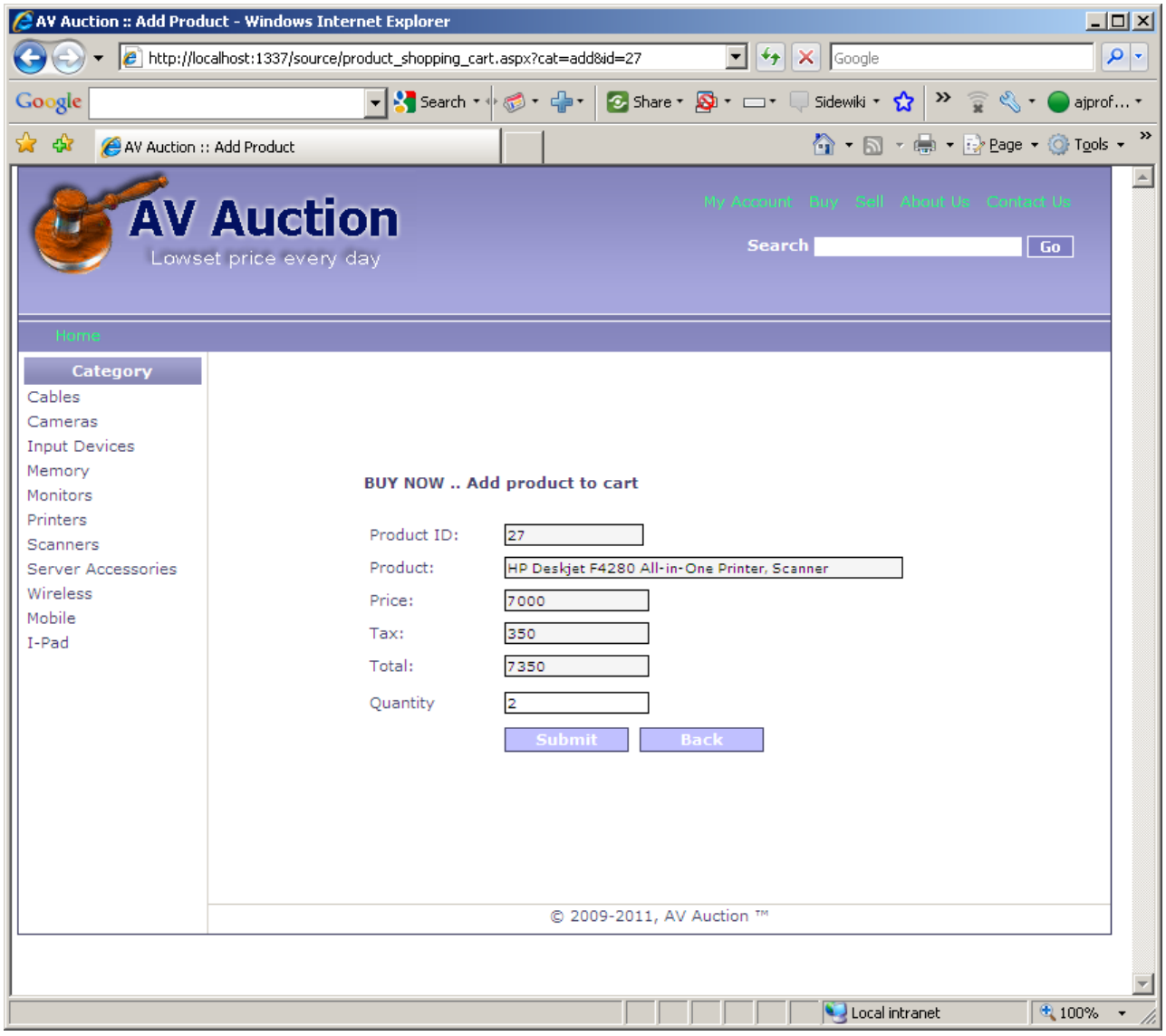
```
        }

        string StrPriceType=" ";
        //if (Request.QueryString.Count > 0)
        //{
        //    StrPriceType = " AND P.prod_sell_type ='Bye' ";
        //}
        //else
        //{
        //    StrPriceType = " ";
        //}

        string SqlStr;
        SqlStr = "select  p.prod_id, p.prod_sell_type, c.cat_type,
v.vendor_name, p.prod_name, p.prod_description,p.prod_price,p.prod_tax,
p.prod_price+p.prod_tax as total ,p.prod_image,prod_bid_min, prod_bid_expair_dt
" +
        " from product_master p,category_master c, vendor_master v " +
        " where p.prod_cat_id=c.cat_id and p.prod_vendor_id=v.vendor_id  " +
        qr + StrPriceType + " and prod_display=1 order by prod_id asc ";


        SqlDataAdapter da = new SqlDataAdapter(SqlStr, cn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            string s1 = "";
            int i;
            for (i = 0; i <= dt.Rows.Count - 1; i++)
            {
                DataRow r;
                r = dt.Rows[i];
                s1 = s1 + "<table border='0' width='95%' id='table__" +
i.ToString() + "' >";
                s1 = s1 + "<tr>";
                s1 = s1 + "<td align='left' valign='top' colspan='2'>";
                s1 = s1 + "<span style='font-size: 10pt'><strong>" +
r["prod_name"].ToString() + "</strong></span></td>";
                s1 = s1 + "</tr>";
                s1 = s1 + "<tr>";
                s1 = s1 + "<td align='left' valign='top' style='width:
688px'>";
                s1 = s1 + "<img border='0' src='prod_image/" +
r["prod_image"].ToString() + "'></td>";
                s1 = s1 + "<td align='right' valign='top' width='306'>";



                if (r["prod_sell_type"].ToString() == "Buy")
                {
                    s1 = s1 + "<table border='0' id='table2_" + i.ToString() +
"' align='right' style='width: 250px'>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Category : </td>";
                    s1 = s1 + "<td width='100' align='left'><b>" +
r["cat_type"].ToString() + "</b></td>";
                    s1 = s1 + "</tr>";
```

```
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'> </td>";
                    s1 = s1 + "<td width='100'> </td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Price</td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_price"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Tax</td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_tax"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Total</td>";
                    s1 = s1 + "<td width='100' style='border-left-width: 1px;
border-right-width: 1px; border-top-style: solid; border-top-width: 1px;
border-bottom-width: 1px' align='right'>" + r["total"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>";
                    s1 = s1 + "</td>";
                    s1 = s1 + "<td align='right' style='border-top: 1px solid;
border-left-width: 1px; border-bottom-width: 1px; border-right-width: 1px'
width='149'>";
                    s1 = s1 + "</br><a
href='product_shopping_cart.aspx?cat=add&id=" + r["prod_id"].ToString() + "'>";
                    s1 = s1 + "<img src='images/buy_now.gif'
border='0'></a></td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "</table>";
                }
                else
                {
                    s1 = s1 + "<table border='0' id='table2_" + i.ToString() +
"' align='right' style='width: 250px'>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Category : </td>";
                    s1 = s1 + "<td width='100' align='left'><b>" +
r["cat_type"].ToString() + "</b></td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'> </td>";
                    s1 = s1 + "<td width='100'> </td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Bid Start Price</td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_bid_min"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Total Bids</td>";
                    //get bid count
                    SqlCommand  com2 = new SqlCommand("select count(*) from
bid_details where bid_prod_id=" + r["prod_id"].ToString(), cn);
                    string  StrC ;
                    StrC =com2.ExecuteScalar().ToString();
```

```
                    s1 = s1 + "<td width='100' align='right'>" + StrC +
"</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>Bid End at </td>";
                    s1 = s1 + "<td width='100' align='right'>" +
r["prod_bid_expair_dt"].ToString() + "</td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "<tr>";
                    s1 = s1 + "<td style='width: 150px'>";
                    s1 = s1 + "</td>";
                    s1 = s1 + "<td align='right'  width='10'>";
                    s1 = s1 + "</br><a href='bid.aspx?cat=add&id=" +
r["prod_id"].ToString() + "'>";
                    s1 = s1 + "<img src='images/bid_now.gif'
border='0'></a></td>";
                    s1 = s1 + "</tr>";
                    s1 = s1 + "</table>";
                }

                s1 = s1 + "</td>";
                s1 = s1 + "</tr>";
                s1 = s1 + "<tr>";
                s1 = s1 + "<td align='left' valign='top'
colspan='2'><b>Description:</b><br>";
                s1 = s1 + r["prod_description"].ToString();
                s1 = s1 + "</td>";
                s1 = s1 + "</tr>";
                s1 = s1 + "</table>";
                s1 = s1 + "</br></br>";
            }


           sp_newproduct.InnerHtml = s1;
        }
        else
        {
            sp_newproduct.InnerHtml = "<b>No products available</b>";
        }

    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
}
```

8.8 eAuction – Add Product to Cart

product_shopping_cart.aspx



product_shopping_cart.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```

```
public partial class product_shopping_cart : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        if (Page.IsPostBack == true)
        {
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        if (Session["mem_id"] == null)
        {

            Session["ret_url"] = "product_shopping_cart.aspx?cat=add&id=" +
Request.QueryString["id"].ToString();
            Response.Redirect("login.aspx");
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat=" +
dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();

        //get shopping products
        SqlDataAdapter da;
        DataTable dt;
        string str1,cat, id;
        cat = Request.QueryString["cat"].ToString();
        id = Request.QueryString["id"].ToString();
        if (cat == "add")
        {
            //query from products
            str1 = "select prod_id,prod_name,prod_price,prod_tax,prod_price+
prod_tax as total from product_master where prod_id=" + id;
            da = new SqlDataAdapter(str1, cn);
            dt = new DataTable();
```

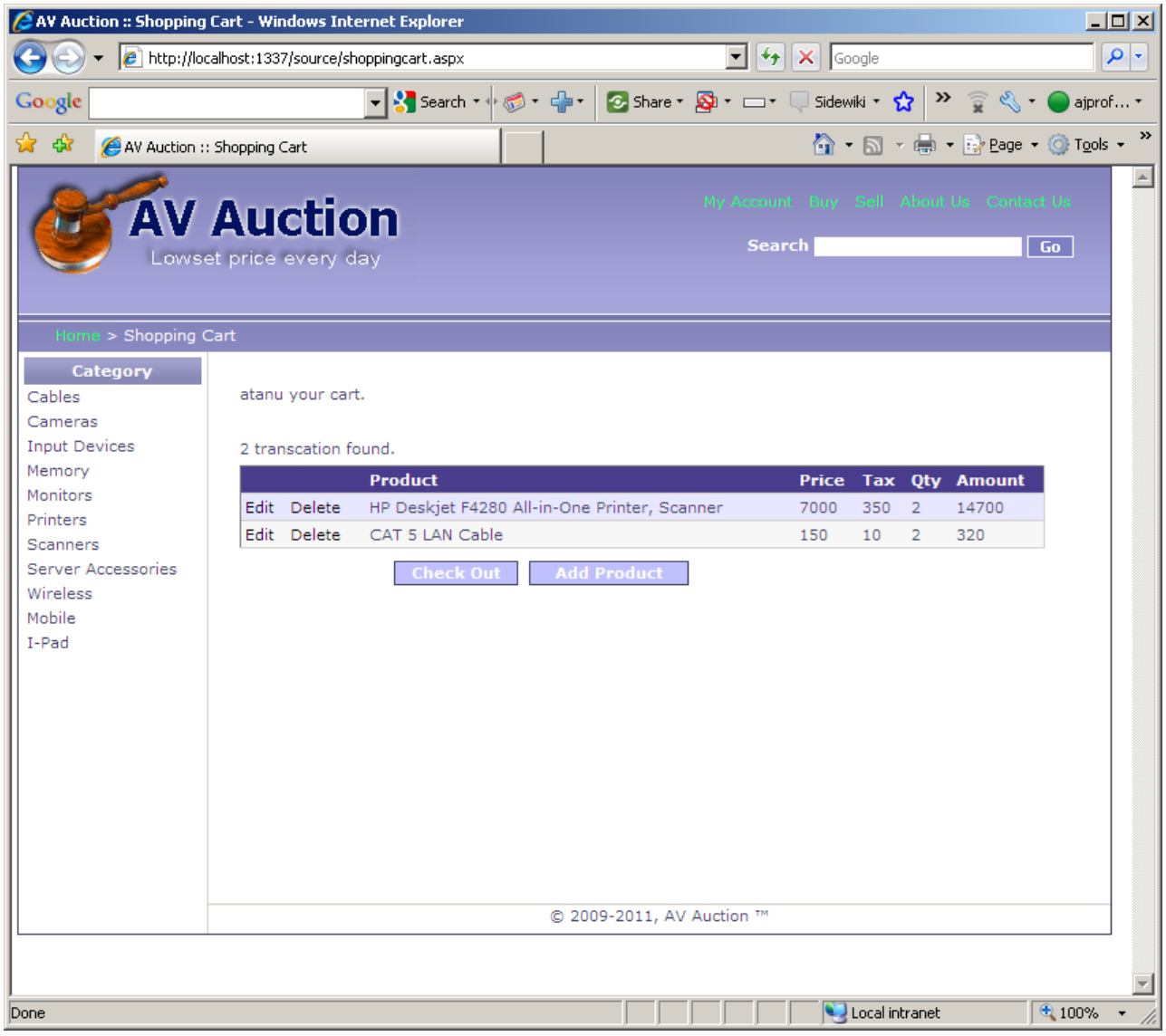
```
            da.Fill(dt);
            TxtProductID.Text = dt.Rows[0]["prod_id"].ToString();
            TxtProduct.Text = dt.Rows[0]["prod_name"].ToString();
            TxtPrice.Text = dt.Rows[0]["prod_price"].ToString();
            TxtTax.Text = dt.Rows[0]["prod_tax"].ToString();
            TxtTotal.Text = dt.Rows[0]["total"].ToString();
            TxtQuantity.Text  = "0";
        }
        else if (cat == "edit")
        {
            //query from shop cart
            str1 = "select prod_id, prod_name, prod_price, prod_tax,
prod_price+ prod_tax as total, sc_quantity from product_master, shopping_cart "
            + "where prod_id=sc_product_id "
            + " and sc_id=" + id;
            da = new SqlDataAdapter(str1, cn);
            dt = new DataTable();
            da.Fill(dt);
            TxtProductID.Text = dt.Rows[0]["prod_id"].ToString();
            TxtProduct.Text = dt.Rows[0]["prod_name"].ToString();
            TxtPrice.Text = dt.Rows[0]["prod_price"].ToString();
            TxtTax.Text = dt.Rows[0]["prod_tax"].ToString();
            TxtTotal.Text = dt.Rows[0]["total"].ToString();
            TxtQuantity.Text  = dt.Rows[0]["sc_quantity"].ToString();
        }
        else if (cat == "delete")
        {


            //del
            com = new SqlCommand("delete from   shopping_cart where sc_id = " +
id, cn);
            com.ExecuteNonQuery ();

            Response.Redirect("shoppingcart.aspx");

        }


    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        if (Session["mem_id"] == null)
        {
            Response.Redirect("login.aspx");
        }

        if (int.Parse(TxtQuantity.Text)<=0)
        {
            MyClass.MyAlert(this, "Enter quantity.", "123");
            return;

        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
```

```
        {
            cn.Open();
        }

        SqlDataAdapter da;
        DataTable dt;
        SqlCommandBuilder cb;

        string str1,cat, id;
        cat = Request.QueryString["cat"].ToString();
        id = Request.QueryString["id"].ToString();
        if (cat == "add")
        {
            int lastid;
            SqlCommand com=new SqlCommand ("select max(sc_id) from
shopping_cart",cn);
            lastid=int.Parse( com.ExecuteScalar().ToString()) +1;

            da = new SqlDataAdapter("select * from shopping_cart where 1=2 ",
cn);
            cb = new SqlCommandBuilder(da);
            dt = new DataTable();
            da.Fill(dt);
            DataRow r;

            r = dt.NewRow();
            r["sc_id"] = lastid;
            r["sc_mem_id"] =int.Parse(Session["mem_id"].ToString());
            r["sc_date"] = System.DateTime.Now.ToString() ;
            r["sc_product_id"] = int.Parse(TxtProductID.Text);
            r["sc_quantity"] = int.Parse(TxtQuantity.Text);
            r["sc_check_out"] = 0;

            dt.Rows.Add(r);
            da.Update(dt);

        }
        else
        {
            da = new SqlDataAdapter("select * from shopping_cart where sc_id =
" + id , cn);
            cb = new SqlCommandBuilder(da);
            dt = new DataTable();
            da.Fill(dt);
            DataRow r;

            r = dt.Rows[0];
            r["sc_quantity"] = int.Parse(TxtQuantity.Text);
            da.Update(dt);
        }


        this.ClientScript.RegisterStartupScript(this.GetType(), "Alert",
"<script language=\"javaScript\">" + "alert('Saved Successfully!');" +
"window.location.href='shoppingcart.aspx';" + "<" + "/script>");
        //Response.Redirect("shoppingcart.aspx");
    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
```

```
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
}
```

8.9 eAuction – Shopping Cart

shoppingcart.aspx



shoppingcart.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```

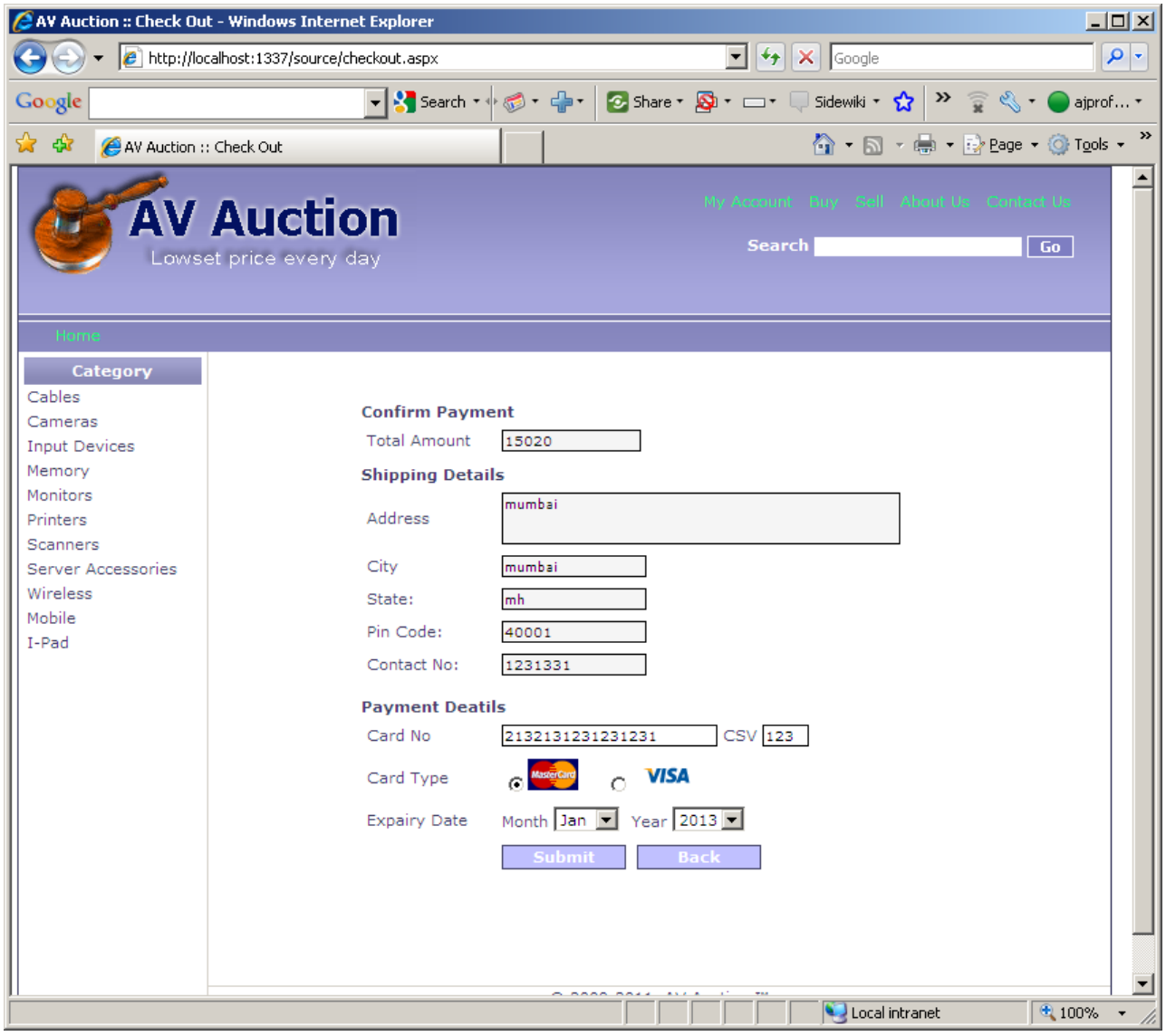
```
public partial class shoppingcart : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        if (Session["mem_id"] == null)
        {
            Session["ret_url"] = "shoppingcart.aspx";
            Response.Redirect("login.aspx");
        }

        if (Page.IsPostBack == true)
        {
            return;
        }

        sp_welcome.InnerHtml  = Session["mem_name"].ToString() + " your cart.";

        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat=" +
dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();

        //get latest product

        string SqlStr;
        SqlStr = "select sc_id as [ ], prod_name as Product, prod_price as
Price, prod_tax as Tax, sc_quantity as Qty, (prod_price+ prod_tax)* sc_quantity
as Amount from product_master, shopping_cart "
        + " where prod_id=sc_product_id and sc_check_out=0 "
        + " and sc_mem_id= " + Session["mem_id"];

        SqlDataAdapter da = new SqlDataAdapter(SqlStr, cn);
        DataTable dt = new DataTable();
        da.Fill(dt);
```

```
        ViewState["g_data"]=dt;
        Dg1.DataSource = ViewState["g_data"];
        Dg1.DataBind();

        sp_tran.InnerHtml = dt.Rows.Count.ToString() + " transcation found.";
    }


    protected void Dg1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg1.DataSource = ViewState["g_data"];
        Dg1.PageIndex = e.NewPageIndex;
        Dg1.DataBind();
    }
    protected void Dg1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowIndex < 0)
        {
            return;
        }
        string s1 = "";// e.Row.Cells[0].Text;
        s1 = "<a href='product_shopping_cart.aspx?cat=edit&id=" +
e.Row.Cells[0].Text + "'><font color=black>Edit</font></a>   <a
href='product_shopping_cart.aspx?cat=delete&id=" + e.Row.Cells[0].Text +
"'><font color=black>Delete</font></a>";
        e.Row.Cells[0].Text = s1;
    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Redirect("buy.aspx");
    }
    protected void Button1_Click1(object sender, EventArgs e)
    {
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }
        SqlCommand com = new SqlCommand("select count(*) from shopping_cart
where  sc_mem_id=" + Session["mem_id"].ToString() + " and sc_check_out=0", cn);
        if (int.Parse (com.ExecuteScalar().ToString())>0)
        {

            Response.Redirect("checkout.aspx");
        }
        else
        {
            MyClass.MyAlert(this, "No product found for checkout.","123");
            return ;
```

```
            }
        }
    }
```

8.10 eAuction – Checkout

checkout.aspx



checkout.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
```

```
using System.Web.UI.HtmlControls;

public partial class checkout : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack == true)
        {
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        //get category
        string StrCat = "";
        SqlDataReader dr;
        SqlCommand com = new SqlCommand();
        com.CommandText = "GetCategory";
        com.CommandType = CommandType.StoredProcedure;
        com.Connection = cn;
        dr = com.ExecuteReader();
        while (dr.Read() == true)
        {
            StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat=" +
dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
        }

        sp_cat.InnerHtml = StrCat;
        dr.Close();


        //if user logged in show data
        if (Session["mem_id"] != null)
        {

            string StrSql;
            StrSql = "select sum((prod_price+ prod_tax)* sc_quantity) as Amount
from product_master, shopping_cart "
            + " where prod_id=sc_product_id and sc_check_out=0 and sc_mem_id="
+ Session["mem_id"].ToString();
            com = new SqlCommand(StrSql, cn);
            TxtAmount.Text = com.ExecuteScalar().ToString();


            SqlDataAdapter da = new SqlDataAdapter("select * from member_master
where mem_id=" + Session["mem_id"].ToString() , cn);
            DataTable dt = new DataTable();
            da.Fill(dt);
            if (dt.Rows.Count>0 )
            {
                DataRow r;
```

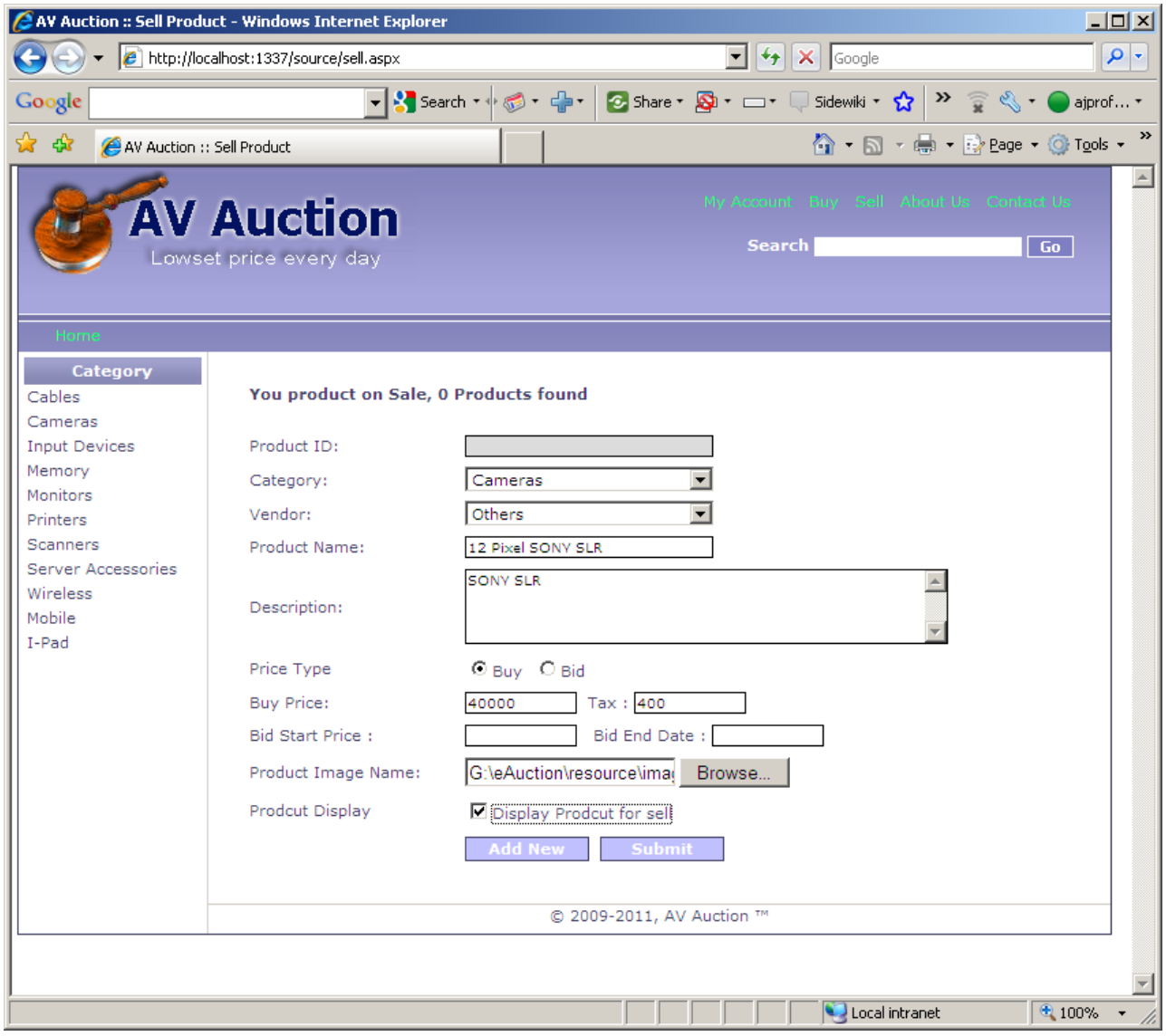
```
            r = dt.Rows[0];
            TxtAddress.Text = r["mem_address"].ToString();
            TxtCity.Text = r["mem_address"].ToString();
            TxtState.Text = r["mem_state"].ToString();
            TxtPin.Text = r["mem_postal_code"].ToString();
            TxtContact.Text = r["mem_contact_no"].ToString();


        }



        DdlMonth.Items.Add("");
        DdlMonth.Items.Add("Jan");
        DdlMonth.Items.Add("Feb");
        DdlMonth.Items.Add("Mar");
        DdlMonth.Items.Add("Apr");
        DdlMonth.Items.Add("May");
        DdlMonth.Items.Add("Jun");
        DdlMonth.Items.Add("Jul");
        DdlMonth.Items.Add("Aug");
        DdlMonth.Items.Add("Sep");
        DdlMonth.Items.Add("Oct");
        DdlMonth.Items.Add("Nov");
        DdlMonth.Items.Add("Dec");


        DdlYear.Items.Add("");
        int i;
        for (i = 2009; i < 2200; i++)
        {
            DdlYear.Items.Add(i.ToString());
        }



    }
}

protected void Button2_Click(object sender, EventArgs e)
{
    Response.Redirect("shoppingcart.aspx");
}
protected void Button1_Click(object sender, EventArgs e)
{
    if (Session["mem_id"] == null)
    {
        Response.Redirect("login.aspx");
        return;
    }

    if (TxtCard.Text.Length !=16)
    {
        MyClass.MyAlert(this, "Enter 16 digit Card No.", "123");
        return;
    }
    if (TxtCSV.Text.Length != 3)
    {
```

```
            MyClass.MyAlert(this, "Enter 3 digit CSV no.", "123");
            return ;

        }

        if (DdlMonth.Text == "")
        {
            MyClass.MyAlert(this, "Select month name from list.", "123");
            return;
        }

        if(DdlYear.Text =="")
        {
            MyClass.MyAlert(this, "Select year from list.", "123");
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }
        int chm, cht, tn;
        SqlTransaction tr;
        SqlCommand com;
        tr = cn.BeginTransaction();
        com = new SqlCommand("select max(checkout_id) from checkout_master ",
cn,tr);
        chm = int.Parse(com.ExecuteScalar().ToString())+1;

        com = new SqlCommand("select max(checout_transaction_id) from
checkout_master ", cn, tr);
        tn = int.Parse(com.ExecuteScalar().ToString()) + 1;

        Session["tn_id"] = tn;
        //update transcation
        //insert into checkout _master
        SqlDataAdapter da = new SqlDataAdapter("select * from checkout_master
where 1=2 ", cn);
        da.SelectCommand.Transaction = tr;
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        DataTable dt = new DataTable();
        da.Fill(dt);
        DataRow r;
        r = dt.NewRow();

        r["checkout_id"] = chm;
        r["checkout_mem_id"] = int.Parse(Session["mem_id"].ToString());
        r["checkout_date"] = System.DateTime.Now.ToString();
        r["checkout_amount"] = TxtAmount.Text;
        r["checkout_payment_success"] = 1;
        r["checout_transaction_id"] =  tn;

        dt.Rows.Add(r);

        da.Update(dt);
```

```
        //insert into checkout_transcation
        SqlDataAdapter da_s = new SqlDataAdapter("select
sc_id,sc_product_id,sc_quantity,prod_price + prod_tax price from
shopping_cart,product_master "
        + " where  sc_product_id=prod_id and sc_mem_id=" +
Session["mem_id"].ToString() + " and sc_check_out=0", cn);
        da_s.SelectCommand.Transaction = tr;
        DataTable dt_s = new DataTable();
        da_s.Fill(dt_s);

        com = new SqlCommand("select max(ct_id) from checkout_tranasation ",
cn, tr);
        cht = int.Parse(com.ExecuteScalar().ToString()) + 1;

        da = new SqlDataAdapter("select * from checkout_tranasation where 1=2
", cn);
        da.SelectCommand.Transaction = tr;
        cb = new SqlCommandBuilder(da);
        dt = new DataTable();
        da.Fill(dt);

        int i;
        for (i = 0; i <= dt_s.Rows.Count - 1; i++)
        {
            r = dt.NewRow();
            r["ct_id"] = cht;
            r["ct_checkout_id"] = chm;
            r["ct_product_id"] = dt_s.Rows[i]["sc_product_id"].ToString();
            r["ct_quantity"] = dt_s.Rows[i]["sc_quantity"].ToString();
            r["ct_rate"] = dt_s.Rows[i]["price"].ToString();

            dt.Rows.Add(r);
            cht = cht + 1;
        }
        da.Update(dt);

        //update shopping cart

        com = new SqlCommand("update shopping_cart set sc_check_out=1 where
sc_mem_id=" + Session["mem_id"].ToString(), cn, tr);
        com.ExecuteNonQuery();

        tr.Commit();

        Response.Redirect("confirmation.aspx");
    }
    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
}
```

8.10 eAuction – Sell Product

sell.aspx



sell.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
```

```
using System.Web.UI.HtmlControls;

public partial class sell : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

        if (Session["mem_id"] == null)
        {
            Session["ret_url"] = "sell.aspx";
            Response.Redirect("login.aspx");
            return;
        }
        if (Page.IsPostBack == false)
        {
            if (Page.IsPostBack == true)
            {
                return;
            }
            string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = connectionInfo;
            if (cn.State == ConnectionState.Closed)
            {
                cn.Open();
            }

            //get category
            string StrCat = "";
            SqlDataReader dr;
            SqlCommand com = new SqlCommand();
            com.CommandText = "GetCategory";
            com.CommandType = CommandType.StoredProcedure;
            com.Connection = cn;
            dr = com.ExecuteReader();
            while (dr.Read() == true)
            {
                StrCat = StrCat + "<tr><td width='126'><a href='buy.aspx?cat="
+ dr["cat_type"].ToString() + "'><font color='#434367'>" +
dr["cat_type"].ToString() + "</font></a></td><tr>";
            }

            sp_cat.InnerHtml = StrCat;
            dr.Close();


            //fill category and vendor
            SqlDataAdapter da = new SqlDataAdapter("select cat_type from
category_master order by cat_type  ", cn);
            DataTable dt = new DataTable();
            da.Fill(dt);
            int i;
            DdlCategory.Items.Add("");
            for (i = 0; i <= dt.Rows.Count - 1; i++)
            {
                DdlCategory.Items.Add(dt.Rows[i]["cat_type"].ToString());
            }
```

```
            da = new SqlDataAdapter("select vendor_name from vendor_master
order by vendor_name  ", cn);
            dt = new DataTable();
            da.Fill(dt);

            DdlVendor.Items.Add("");
            for (i = 0; i <= dt.Rows.Count - 1; i++)
            {
                DdlVendor.Items.Add(dt.Rows[i]["vendor_name"].ToString());
            }

            string SqlStr;
            SqlStr = "select prod_id, prod_name, cat_type , prod_bid_min,
prod_bid_expair_dt, prod_display , vendor_name  "
            + " from product_master, category_master,vendor_master "
            + " where prod_cat_id=cat_id and prod_vendor_id = vendor_id and
prod_create_by =" + Session["mem_id"].ToString() + " order by prod_id ";

            da = new SqlDataAdapter(SqlStr, cn);
            dt = new DataTable();
            da.Fill(dt);

            ViewState["v_table"] = dt;
            Dg1.DataSource = ViewState["v_table"];
            Dg1.DataBind();

            sp_total_prodcuts.InnerHtml = dt.Rows.Count.ToString() + " Products
found";

            TxtProductID.Text = "";
            TxtProductName.Text = "";
            DdlCategory.Text = "";
            DdlVendor.Text = "";
            TxtDescription.Text = "";
            TxtPrice.Text = "";
            TxtTax.Text = "";
            TxtBidEndDate.Text = "";
            TxtBidStartPrice.Text = "";




            if (Request.QueryString.Count > 0)
            {
                string qid;
                qid = Request.QueryString["id"].ToString();

                string SqlStr1;
                SqlStr1 = "select prod_id, prod_name, cat_type , vendor_name,
prod_description, prod_price,prod_tax, prod_bid_min, prod_bid_expair_dt,
prod_image ,prod_display,prod_sell_type "
                + " from product_master, category_master,vendor_master "
                + " where prod_cat_id=cat_id and prod_vendor_id = vendor_id and
prod_id =" + qid;


                SqlDataAdapter da1 = new SqlDataAdapter(SqlStr1, cn);
```

```
                DataTable dt1 = new DataTable();
                da1.Fill(dt1);
                DataRow r;
                if (dt1.Rows.Count > 0)
                {
                    r = dt1.Rows[0];


                    TxtProductID.Text = r["prod_id"].ToString();
                    TxtProductName.Text = r["prod_name"].ToString();
                    DdlCategory.Text = r["cat_type"].ToString();
                    DdlVendor.Text = r["vendor_name"].ToString();
                    TxtDescription.Text = r["prod_description"].ToString();
                    if (r["prod_sell_type"].ToString() == "Buy")
                    {
                        RdBuy.Checked = true;
                        TxtPrice.Enabled = true;
                        TxtTax.Enabled = true;
                        TxtPrice.Text = r["prod_price"].ToString();
                        TxtTax.Text = r["prod_tax"].ToString();

                        TxtBidStartPrice.Enabled = false;
                        TxtBidEndDate.Enabled = false;
                        TxtBidStartPrice.Text = "";
                        TxtBidEndDate.Text = "";
                    }
                    else
                    {
                        RdBid.Checked = true;
                        TxtPrice.Enabled = false;
                        TxtTax.Enabled = false;
                        TxtPrice.Text = "";
                        TxtTax.Text = "";

                        TxtBidStartPrice.Enabled = true;
                        TxtBidEndDate.Enabled = true;
                        TxtBidStartPrice.Text = r["prod_bid_min"].ToString();
                        TxtBidEndDate.Text =
r["prod_bid_expair_dt"].ToString();
                    }

                    if (r["prod_display"].ToString() == "1")
                    {
                        ChkDisplay.Checked = true;
                    }
                    else
                    {
                        ChkDisplay.Checked = false;
                    }

                    sp_link.InnerHtml = "<a href='prod_image\\" +
r["prod_image"].ToString() + "'target='_blank'>View Image</a>";
                }

            }

        }
```

```
}
protected void Button1_Click(object sender, EventArgs e)
{
    TxtProductID.Text = "";
    TxtProductName.Text = "";
    DdlCategory.Text = "";
    DdlVendor.Text = "";
    TxtDescription.Text = "";
    TxtPrice.Text = "0";
    TxtTax.Text = "0";

}
protected void CmdUpdate_Click(object sender, EventArgs e)
{
    if (TxtProductName.Text == "")
    {
        MyClass.MyAlert(this, "Enter product name.", "abc");
        return;
    }

    if (DdlCategory.Text == "")
    {
        MyClass.MyAlert(this, "Select category name.", "abc");
        return;
    }

    if (DdlVendor.Text == "")
    {
        MyClass.MyAlert(this, "Select vendor name.", "abc");
        return;
    }

    if (RdBuy.Checked == true)
    {
        if (TxtPrice.Text.Trim() == "")
        {
            MyClass.MyAlert(this, "Enter buy price.", "abc");
            return;
        }
        if (int.Parse(TxtPrice.Text) <= 0)
        {
            MyClass.MyAlert(this, "Enter buy price.", "abc");
            return;
        }
        if (TxtTax.Text.Trim() == "")
        {
            MyClass.MyAlert(this, "Enter tax amount.", "abc");
            return;
        }
    }
    if (RdBid.Checked == false && RdBuy.Checked == false)
    {
        MyClass.MyAlert(this, "Select price type BID/BUY ?.", "abc");
        return;
    }

    if (RdBid.Checked == true)
    {
```

```
            if (TxtBidStartPrice.Text.Trim() == "")
            {
                MyClass.MyAlert(this, "Enter bid start price.", "abc");
                return;
            }
            if (int.Parse(TxtBidStartPrice.Text) <= 0)
            {
                MyClass.MyAlert(this, "Enter bid start price.", "abc");
                return;
            }
            if (TxtBidEndDate.Text.Trim() == "")
            {
                MyClass.MyAlert(this, "Enter bid date.", "abc");
                return;
            }
        }
        bool ch;
        string qr;
        if (TxtProductID.Text == "")
        {
            ch = true;
            qr = " where 1=2 ";
        }
        else
        {
            ch = false;
            qr = " where prod_id = " + TxtProductID.Text;

        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        int lastid, ven_id, cat_id;
        SqlCommand com = new SqlCommand("select max(prod_id) from
product_master ", cn);
        lastid = int.Parse(com.ExecuteScalar().ToString()) + 1;

        com = new SqlCommand("select cat_id from category_master where
cat_type='" + DdlCategory.Text + "'", cn);
        cat_id = int.Parse(com.ExecuteScalar().ToString());

        com = new SqlCommand("select vendor_id from vendor_master where
vendor_name='" + DdlVendor.Text + "'", cn);
        ven_id = int.Parse(com.ExecuteScalar().ToString());

        SqlDataAdapter da = new SqlDataAdapter("select * from product_master "
+ qr, cn);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        DataTable dt = new DataTable();
        da.Fill(dt);
        DataRow r;
        if (ch == true)
```

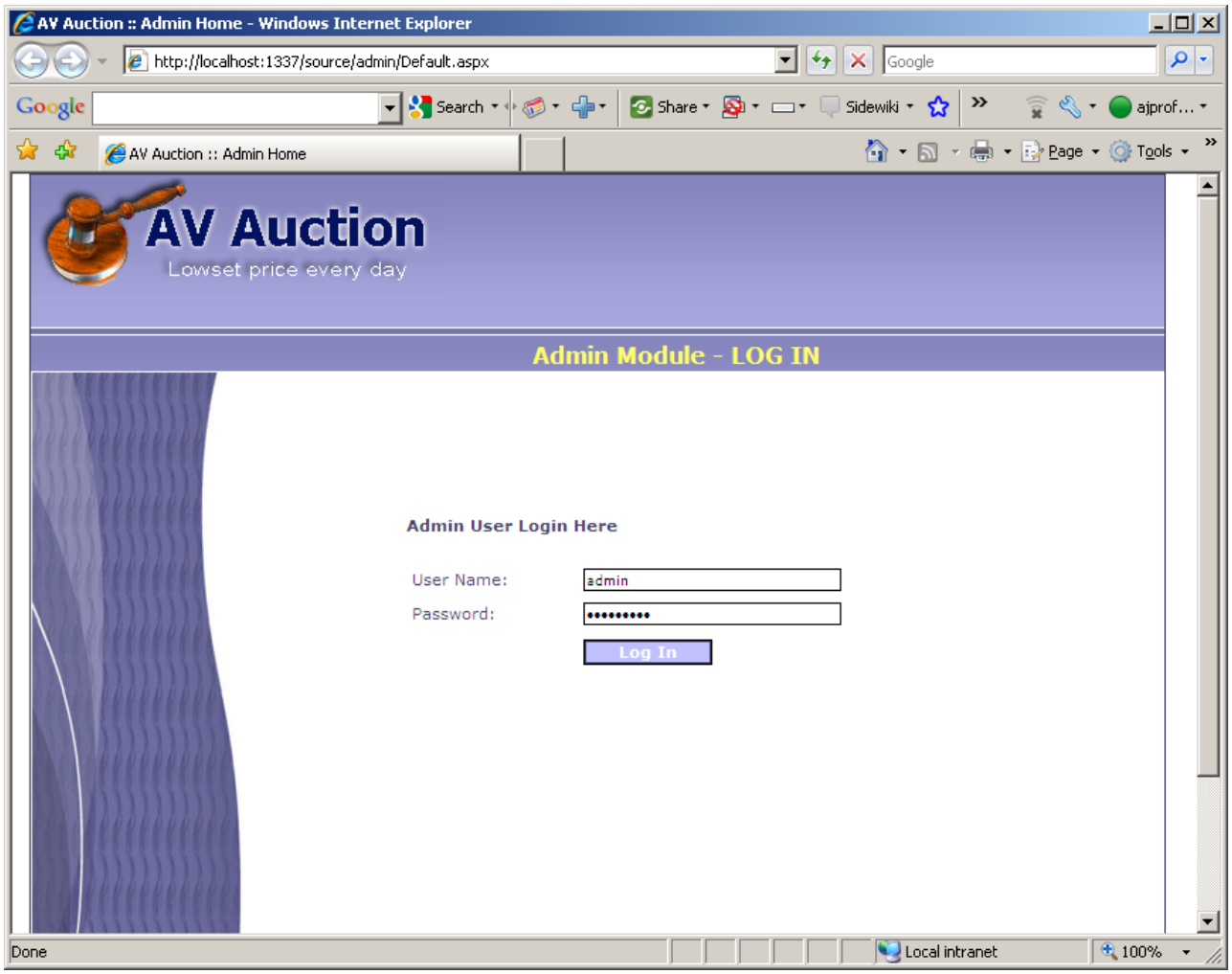
```
{
    r = dt.NewRow();


    r["prod_id"] = lastid;
    r["prod_name"] = TxtProductName.Text;


    r["prod_cat_id"] = cat_id;
    r["prod_vendor_id"] = ven_id;

    r["prod_description"] = TxtDescription.Text;

    if (RdBuy.Checked == true)
    {
        r["prod_sell_type"] = "Buy";
        r["prod_price"] = int.Parse(TxtPrice.Text);
        r["prod_tax"] = int.Parse(TxtTax.Text);
    }
    else
    {
        r["prod_sell_type"] = "Bid";
        r["prod_bid_min"] = int.Parse(TxtBidStartPrice.Text);
        r["prod_bid_expair_dt"] = TxtBidEndDate.Text;
    }
    //r["prod_image"]=TxtProductImage.Text ;

    r["prod_create_date"] = System.DateTime.Now.ToString();
    r["prod_create_by"] = Session["mem_id"].ToString();

    if (ChkDisplay.Checked == true)
    {
        r["prod_display"] = 1;
    }
    else
    {
        r["prod_display"] = 0;
    }
    r["prod_image"] = r["prod_id"].ToString() + ".jpg";
    dt.Rows.Add(r);
}
else
{
    r = dt.Rows[0];
    r["prod_name"] = TxtProductName.Text;


    r["prod_cat_id"] = cat_id;
    r["prod_vendor_id"] = ven_id;

    r["prod_description"] = TxtDescription.Text;

    if (RdBuy.Checked == true)
    {
        r["prod_sell_type"] = "Buy";
        r["prod_price"] = int.Parse(TxtPrice.Text);
        r["prod_tax"] = int.Parse(TxtTax.Text);
    }
```

```
            else
            {
                r["prod_sell_type"] = "Bid";
                r["prod_bid_min"] = int.Parse(TxtBidStartPrice.Text);
                r["prod_bid_expair_dt"] = TxtBidEndDate.Text;
            }


            if (ChkDisplay.Checked == true)
            {
                r["prod_display"] = 1;
            }
            else
            {
                r["prod_display"] = 0;
            }
            r["prod_image"] = r["prod_id"].ToString() + ".jpg";
        }

        if (fu1.HasFile)
        {
            if (fu1.FileName.Substring(fu1.FileName.LastIndexOf(".") + 1, 3) ==
"jpg" || fu1.FileName.Substring(fu1.FileName.LastIndexOf("."), 3) == "gif")
            {
                fu1.PostedFile.SaveAs(Server.MapPath("prod_image") + "\\" +
r["prod_id"].ToString() + ".jpg");
                //ImgLogo.ImageUrl = Server.MapPath("TempImages") + "\\" +
fu1.FileName;
                //name1.ImageUrl = "TempImages/" + fu1.FileName;
            }
        }
        else
        {

        }


        da.Update(dt);

        Response.Redirect("sell.aspx");


    }

    protected void Dg1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg1.DataSource = ViewState["v_table"];
        Dg1.PageIndex = e.NewPageIndex;
        Dg1.DataBind();
    }
    protected void Dg1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowIndex < 0)
        {
            return;
        }
        string s1 = "";// e.Row.Cells[0].Text;
```

```
        s1 = "<a href='sell.aspx?id=" + e.Row.Cells[0].Text + "'><font
color='black'><u>" + e.Row.Cells[0].Text + "</u></font></a>";
        e.Row.Cells[0].Text = s1;
    }
    protected void RdBuy_CheckedChanged(object sender, EventArgs e)
    {
        if (RdBuy.Checked == true)
        {
            TxtPrice.Enabled = true;
            TxtTax.Enabled = true;
            TxtBidStartPrice.Text = "";
            TxtBidStartPrice.Enabled = false;
            TxtBidEndDate.Enabled = false;
            TxtBidEndDate.Text = "";
        }
        else
        {
            TxtPrice.Enabled = false;
            TxtPrice.Text = "";
            TxtTax.Enabled = false;
            TxtTax.Text = "";

            TxtBidStartPrice.Enabled = true;
            TxtBidEndDate.Enabled = true;

        }
    }
    protected void RdBid_CheckedChanged(object sender, EventArgs e)
    {
        if (RdBuy.Checked == true)
        {
            TxtPrice.Enabled = true;
            TxtTax.Enabled = true;
            TxtBidStartPrice.Text = "";
            TxtBidStartPrice.Enabled = false;
            TxtBidEndDate.Enabled = false;
            TxtBidEndDate.Text = "";
        }
        else
        {
            TxtPrice.Enabled = false;
            TxtPrice.Text = "";
            TxtTax.Enabled = false;
            TxtTax.Text = "";

            TxtBidStartPrice.Enabled = true;
            TxtBidEndDate.Enabled = true;

        }


    }

    protected void BtnGo_Click(object sender, ImageClickEventArgs e)
    {
        Response.Redirect("buy.aspx?search=" + TxtSearch.Text);
    }
}
```

8.11 eAuction Admin – Home Page

default.aspx



default.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class admin_Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
```

```
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        //login and return page call
        if (TxtUserName.Text == "")
        {
            MyClass.MyAlert(this, "Enter email id.", "123");
            return;
        }
        if (TxtPassword.Text == "")
        {
            MyClass.MyAlert(this, "Enter password.", "123");
            return;
        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }


        SqlDataAdapter da = new SqlDataAdapter("select * from user_master where
user_name='" + TxtUserName.Text + "' and user_password='" + TxtPassword.Text +
"'", cn);
        DataTable dt = new DataTable();
        da.Fill(dt);
        if (dt.Rows.Count == 1)
        {
            Session["user_id"] = dt.Rows[0]["user_id"].ToString();
            Session["user_name"] = dt.Rows[0]["user_name"].ToString();
            Session["user_type"] = dt.Rows[0]["user_type"].ToString();


            if (Session["admin_ret_url"] == null)
            {
                Response.Redirect("users.aspx");
            }
            else
            {
                Response.Redirect(Session["admin_ret_url"].ToString());
            }
        }
        else
        {
            MyClass.MyAlert(this, "Can not login, Inavlid email id/password.",
"123");
            return;
        }

    }
}
```

8.12 eAuction Admin –Users

users.aspx

users.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class admin_users : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
```

```
    {
        if (Session["user_id"] == null)
        {
            Session["admin_ret_url"] = "users.aspx";
            Response.Redirect("default.aspx");
            return;
        }
        if (Page.IsPostBack == false)
        {
            sp_welcome.InnerHtml = "Welcome back <strong>" +
Session["user_name"].ToString() + "</strong>";
            string ut, id;
            id = Session["user_id"].ToString();
            ut = Session["user_type"].ToString();

            string qr;
            if (ut == "ADMIN")
            {
                qr = " ";
                CmdAddNew.Enabled = true;
                DdlUserType.Enabled = true;
            }
            else
            {
                qr = " where user_id =" + id;
                CmdAddNew.Enabled = false;
                DdlUserType.Enabled = false;
            }

            string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = connectionInfo;
            if (cn.State == ConnectionState.Closed)
            {
                cn.Open();
            }


            DdlUserType.Items.Add("");
            DdlUserType.Items.Add("ADMIN");
            DdlUserType.Items.Add("USER");


            SqlDataAdapter da = new SqlDataAdapter("select user_id,user_name,
user_type, user_create_date from user_master " + qr + " order by user_id  ",
cn);
            DataTable dt = new DataTable();
            da.Fill(dt);

            ViewState["v_table"]=dt;
            Dg1.DataSource = ViewState["v_table"];
            Dg1.DataBind();

            TxtUserID.Text = "";
            TxtUserName.Text = "";
            DdlUserType.Text = "";
```

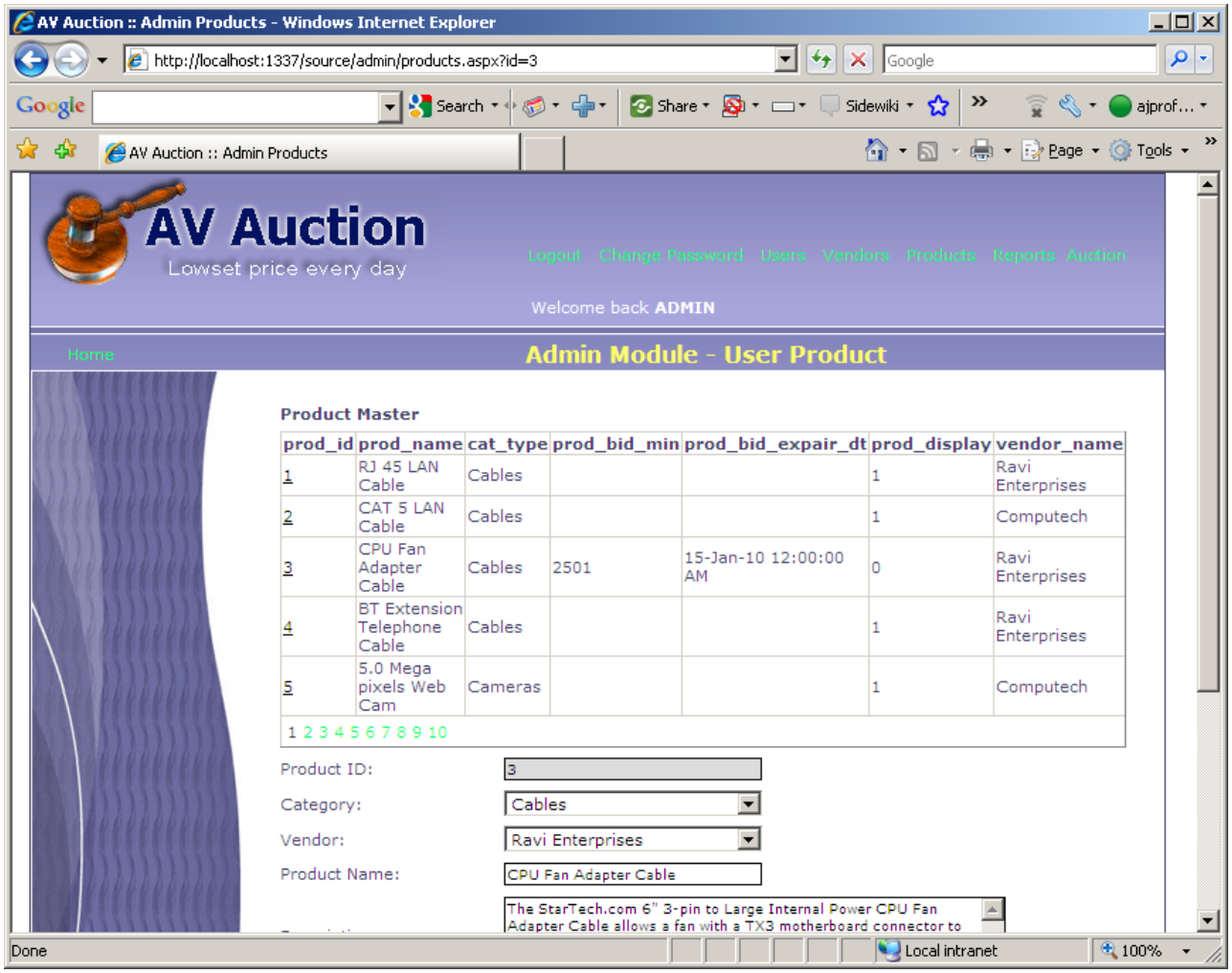
```
            if (Request.QueryString.Count > 0)
            {
                string qid;
                qid = Request.QueryString["id"].ToString();
                SqlDataAdapter da1 = new SqlDataAdapter("select * from
user_master where user_id= " + qid , cn);
                DataTable dt1 = new DataTable();
                da1.Fill(dt1);
                DataRow r;
                if (dt1.Rows.Count > 0)
                {
                    r = dt1.Rows[0];
                    TxtUserID.Text = r["user_id"].ToString();
                    TxtUserName.Text = r["user_name"].ToString();
                    DdlUserType.Text = r["user_type"].ToString();
                }

            }

        }

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        TxtUserID.Text = "";
        TxtUserName.Text = "";
        DdlUserType.Text = "";
    }
    protected void CmdUpdate_Click(object sender, EventArgs e)
    {
        if (TxtUserName.Text == "")
        {
            MyClass.MyAlert(this, "Enter user name.", "abc");
            return;
        }
        if (DdlUserType.Text == "")
        {
            MyClass.MyAlert(this, "Select user type.", "pqr");
            return;
        }

        bool ch;
        string qr;
        if (TxtUserID.Text == "")
        {
            ch = true;
            qr = " where 1=2 ";
        }
        else
        {
            ch = false;
            qr = " where user_id = " + TxtUserID.Text ;

        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
```

```
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        int lastid;
        SqlCommand com = new SqlCommand("select max(user_id) from user_master
", cn);
        lastid =int.Parse(com.ExecuteScalar().ToString())+1;

        SqlDataAdapter da = new SqlDataAdapter("select * from user_master " +
qr   , cn);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        DataTable dt = new DataTable();
        da.Fill(dt);
        DataRow r;
        if (ch==true )
        {
            r = dt.NewRow();
            r["user_id"] = lastid;
            r["user_create_date"] = System.DateTime.Now.ToString();
            r["user_password"] = "123456";
            r["user_name"] = TxtUserName.Text;
            r["user_type"] = DdlUserType.Text;

            dt.Rows.Add(r);
        }
        else
        {
            r = dt.Rows[0];
            r["user_name"] = TxtUserName.Text;
            r["user_type"] = DdlUserType.Text;
        }

        da.Update(dt);

        Response.Redirect("users.aspx");




    }
    protected void Dg1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg1.DataSource = ViewState["v_table"];
        Dg1.PageIndex = e.NewPageIndex;
        Dg1.DataBind();
    }
    protected void Dg1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowIndex < 0)
        {
            return;
        }
        string s1 = "";// e.Row.Cells[0].Text;
        s1 = "<a href='users.aspx?id=" + e.Row.Cells[0].Text + "'><font
color='black'><u>" + e.Row.Cells[0].Text + "</u></font></a>";
```

```
            e.Row.Cells[0].Text = s1;
        }
    }
```

8.13 eAuction Admin – Products

products.aspx



products.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class admin_products : System.Web.UI.Page
{
```

```
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["user_id"] == null)
        {
            Session["admin_ret_url"] = "products.aspx";
            Response.Redirect("default.aspx");
            return;
        }
        if (Page.IsPostBack == false)
        {
            sp_welcome.InnerHtml = "Welcome back <strong>" +
Session["user_name"].ToString() + "</strong>";
            string ut, id;
            id = Session["user_id"].ToString();
            ut = Session["user_type"].ToString();

            string qr;
            if (ut == "ADMIN")
            {

                CmdAddNew.Enabled = true;
                CmdUpdate.Enabled = true;
            }
            else
            {

                CmdAddNew.Enabled = false;
                CmdUpdate.Enabled = false;
            }

            string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = connectionInfo;
            if (cn.State == ConnectionState.Closed)
            {
                cn.Open();
            }

            //fill category and vendor
            SqlDataAdapter da = new SqlDataAdapter("select cat_type from
category_master order by cat_type  ", cn);
            DataTable dt = new DataTable();
            da.Fill(dt);
            int i;
            DdlCategory.Items.Add("");
            for (i = 0; i <= dt.Rows.Count - 1; i++)
            {
                DdlCategory.Items.Add(dt.Rows[i]["cat_type"].ToString());
            }

            da = new SqlDataAdapter("select vendor_name from vendor_master
order by vendor_name  ", cn);
            dt = new DataTable();
            da.Fill(dt);

            DdlVendor.Items.Add("");
            for (i = 0; i <= dt.Rows.Count - 1; i++)
```

```
            {
                DdlVendor.Items.Add(dt.Rows[i]["vendor_name"].ToString());
            }

            string SqlStr;
            SqlStr = "select prod_id, prod_name, cat_type , vendor_name  "
            + " from product_master, category_master,vendor_master "
            + " where prod_cat_id=cat_id and prod_vendor_id = vendor_id order
by prod_id ";

            da = new SqlDataAdapter(SqlStr, cn);
            dt = new DataTable();
            da.Fill(dt);

            ViewState["v_table"] = dt;
            Dg1.DataSource = ViewState["v_table"];
            Dg1.DataBind();

            TxtProductID.Text = "";
            TxtProductName.Text = "";
            DdlCategory.Text ="";
            DdlVendor.Text ="";
            TxtDescription.Text ="";
            TxtPrice.Text ="";
            TxtTax.Text ="";
            TxtProductImage.Text ="";



            if (Request.QueryString.Count > 0)
            {
                string qid;
                qid = Request.QueryString["id"].ToString();

                string SqlStr1;
                SqlStr1 = "select prod_id, prod_name, cat_type , vendor_name,
prod_description, prod_price,prod_tax, prod_image  "
                + " from product_master, category_master,vendor_master "
                + " where prod_cat_id=cat_id and prod_vendor_id = vendor_id and
prod_id =" + qid;


                SqlDataAdapter da1 = new SqlDataAdapter(SqlStr1, cn);
                DataTable dt1 = new DataTable();
                da1.Fill(dt1);
                DataRow r;
                if (dt1.Rows.Count > 0)
                {
                    r = dt1.Rows[0];


                    TxtProductID.Text = r["prod_id"].ToString();
                    TxtProductName.Text = r["prod_name"].ToString();
                    DdlCategory.Text = r["cat_type"].ToString();
                    DdlVendor.Text = r["vendor_name"].ToString();
                    TxtDescription.Text = r["prod_description"].ToString();
                    TxtPrice.Text = r["prod_price"].ToString();
                    TxtTax.Text = r["prod_tax"].ToString();
```

```
                    TxtProductImage.Text = r["prod_image"].ToString();

                }

            }

        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        TxtProductID.Text = "";
        TxtProductName.Text = "";
        DdlCategory.Text = "";
        DdlVendor.Text = "";
        TxtDescription.Text = "";
        TxtPrice.Text = "0";
        TxtTax.Text = "0";
        TxtProductImage.Text = "";
    }
    protected void CmdUpdate_Click(object sender, EventArgs e)
    {
        if (TxtProductName.Text == "")
        {
            MyClass.MyAlert(this, "Enter product name.", "abc");
            return;
        }

        if (DdlCategory.Text == "")
        {
            MyClass.MyAlert(this, "Select category name.", "abc");
            return;
        }

        if (DdlVendor.Text == "")
        {
            MyClass.MyAlert(this, "Select vendor name.", "abc");
            return;
        }

        bool ch;
        string qr;
        if (TxtProductID.Text == "")
        {
            ch = true;
            qr = " where 1=2 ";
        }
        else
        {
            ch = false;
            qr = " where prod_id = " + TxtProductID.Text;

        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
```

```
        {
            cn.Open();
        }

        int lastid,ven_id, cat_id;
        SqlCommand com = new SqlCommand("select max(prod_id) from
product_master ", cn);
        lastid = int.Parse(com.ExecuteScalar().ToString()) + 1;

        com = new SqlCommand("select cat_id from category_master where
cat_type='" + DdlCategory.Text + "'", cn);
        cat_id = int.Parse(com.ExecuteScalar().ToString()) ;

        com = new SqlCommand("select vendor_id from vendor_master where
vendor_name='" + DdlVendor.Text + "'", cn);
        ven_id = int.Parse(com.ExecuteScalar().ToString());

        SqlDataAdapter da = new SqlDataAdapter("select * from product_master "
+ qr, cn);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        DataTable dt = new DataTable();
        da.Fill(dt);
        DataRow r;
        if (ch == true)
        {
            r = dt.NewRow();


            r["prod_id"]= lastid ;
            r["prod_name"] = TxtProductName.Text;


            r["prod_cat_id"] = cat_id;
            r["prod_vendor_id"] = ven_id;

            r["prod_description"]=TxtDescription.Text ;
            r["prod_price"] = int.Parse( TxtPrice.Text );
            r["prod_tax"]= int.Parse( TxtTax.Text );
            r["prod_image"]=TxtProductImage.Text ;

            r["prod_create_date"] = System.DateTime.Now.ToString();
            r["prod_create_by"] = Session["user_id"].ToString();

            dt.Rows.Add(r);
        }
        else
        {
            r = dt.Rows[0];
            r["prod_name"] = TxtProductName.Text;


            r["prod_cat_id"] = cat_id;
            r["prod_vendor_id"] = ven_id;

            r["prod_description"] = TxtDescription.Text;
            r["prod_price"] = int.Parse(TxtPrice.Text);
            r["prod_tax"] = int.Parse(TxtTax.Text);
            r["prod_image"] = TxtProductImage.Text;
```

```
        }

        da.Update(dt);

        Response.Redirect("products.aspx");


    }

    protected void Dg1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg1.DataSource = ViewState["v_table"];
        Dg1.PageIndex = e.NewPageIndex;
        Dg1.DataBind();
    }
    protected void Dg1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowIndex < 0)
        {
            return;
        }
        string s1 = "";// e.Row.Cells[0].Text;
        s1 = "<a href='products.aspx?id=" + e.Row.Cells[0].Text + "'><font
color='black'><u>" + e.Row.Cells[0].Text + "</u></font></a>";
        e.Row.Cells[0].Text = s1;
    }
}
```

8.14 eAuction Admin - Vendors

vendors.aspx

vendors.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class admin_vendors : System.Web.UI.Page
```

```
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["user_id"] == null)
        {
            Session["admin_ret_url"] = "users.aspx";
            Response.Redirect("default.aspx");
            return;
        }
        if (Page.IsPostBack == false)
        {
            sp_welcome.InnerHtml = "Welcome back <strong>" +
Session["user_name"].ToString() + "</strong>";
            string ut, id;
            id = Session["user_id"].ToString();
            ut = Session["user_type"].ToString();

            string qr;
            if (ut == "ADMIN")
            {

                CmdAddNew.Enabled = true;
                CmdUpdate.Enabled = true;
            }
            else
            {

                CmdAddNew.Enabled = false;
                CmdUpdate.Enabled = false;
            }

            string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = connectionInfo;
            if (cn.State == ConnectionState.Closed)
            {
                cn.Open();
            }



            SqlDataAdapter da = new SqlDataAdapter("select
vendor_id,vendor_name,vendor_contact_no, vendor_email from vendor_master order
by vendor_id  ", cn);
            DataTable dt = new DataTable();
            da.Fill(dt);

            ViewState["v_table"] = dt;
            Dg1.DataSource = ViewState["v_table"];
            Dg1.DataBind();

            TxtVendorID.Text = "";
            TxtVendorNAme.Text = "";
            TxtAddress.Text = "";
            TxtCity.Text = "";
            TxtSTate.Text = "";
            TxtPostalCode.Text = "";
```

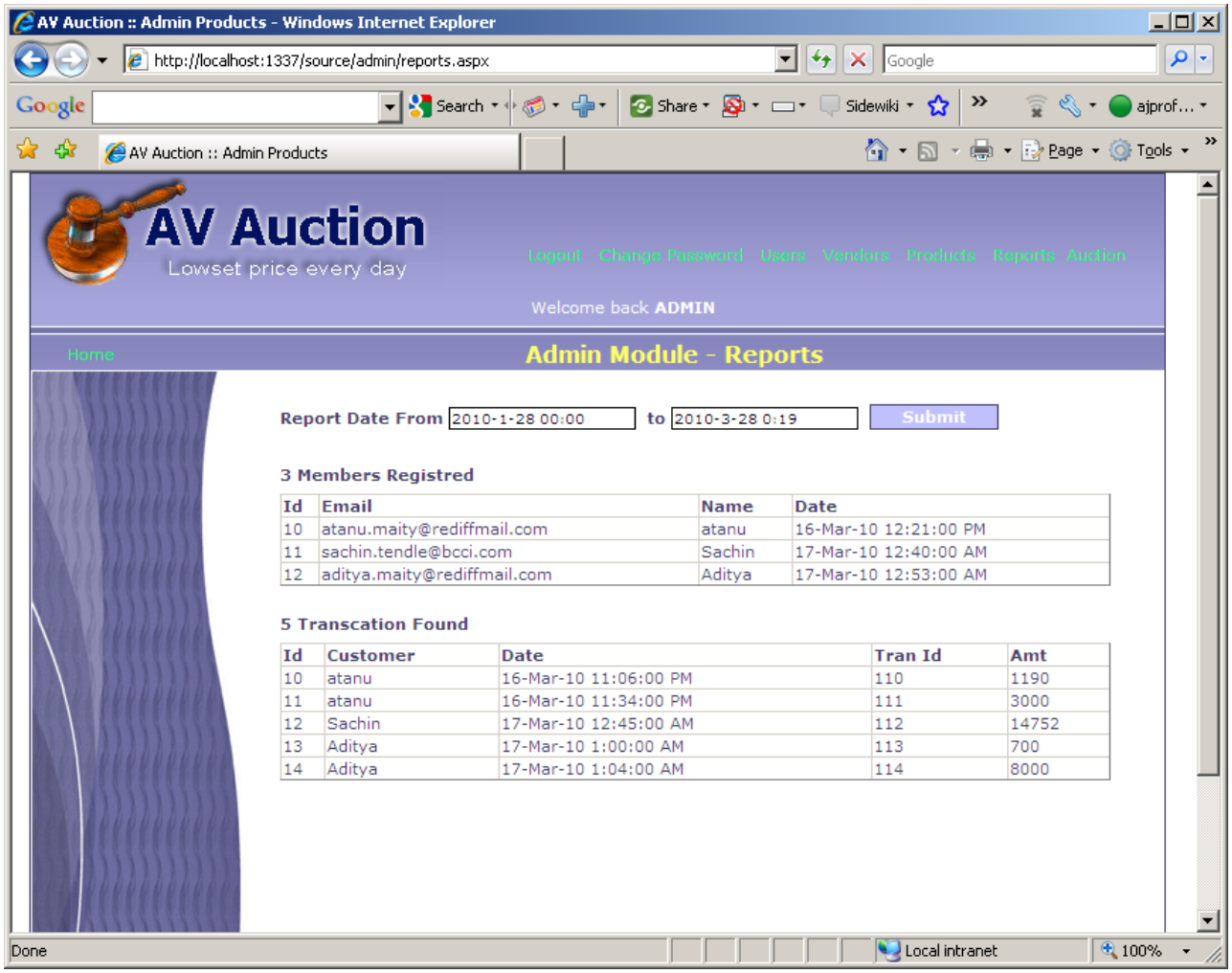
```
            TxtEmailID.Text = "";
            TxtContactNo.Text = "";

            if (Request.QueryString.Count > 0)
            {
                string qid;
                qid = Request.QueryString["id"].ToString();
                SqlDataAdapter da1 = new SqlDataAdapter("select * from
vendor_master where vendor_id= " + qid, cn);
                DataTable dt1 = new DataTable();
                da1.Fill(dt1);
                DataRow r;
                if (dt1.Rows.Count > 0)
                {
                    r = dt1.Rows[0];

                    TxtVendorID.Text = r["vendor_id"].ToString();
                    TxtVendorNAme.Text = r["vendor_name"].ToString();
                    TxtAddress.Text = r["vendor_address"].ToString();
                    TxtCity.Text = r["vendor_city"].ToString();
                    TxtSTate.Text = r["vendor_state"].ToString();
                    TxtPostalCode.Text = r["vendor_postal_code"].ToString();
                    TxtEmailID.Text = r["vendor_email"].ToString();
                    TxtContactNo.Text = r["vendor_contact_no"].ToString();
                }

            }

        }

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        TxtVendorID.Text = "";
        TxtVendorNAme.Text = "";
        TxtAddress.Text = "";
        TxtCity.Text = "";
        TxtSTate.Text = "";
        TxtPostalCode.Text = "";
        TxtEmailID.Text = "";
        TxtContactNo.Text = "";
    }
    protected void CmdUpdate_Click(object sender, EventArgs e)
    {
        if (TxtVendorNAme.Text == "")
        {
            MyClass.MyAlert(this, "Enter user name.", "abc");
            return;
        }


        bool ch;
        string qr;
        if (TxtVendorID.Text == "")
        {
            ch = true;
            qr = " where 1=2 ";
        }
```

```
        else
        {
            ch = false;
            qr = " where vendor_id = " + TxtVendorID.Text;

        }
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        int lastid;
        SqlCommand com = new SqlCommand("select max(vendor_id) from
vendor_master ", cn);
        lastid = int.Parse(com.ExecuteScalar().ToString()) + 1;

        SqlDataAdapter da = new SqlDataAdapter("select * from vendor_master " +
qr, cn);
        SqlCommandBuilder cb = new SqlCommandBuilder(da);
        DataTable dt = new DataTable();
        da.Fill(dt);
        DataRow r;
        if (ch == true)
        {
            r = dt.NewRow();
            r["vendor_id"] = lastid;
            r["vendor_name"]=TxtVendorNAme.Text;
            r["vendor_address"]=TxtAddress.Text ;
            r["vendor_city"]=TxtCity.Text ;
            r["vendor_state"]=TxtSTate.Text ;
            r["vendor_postal_code"]=TxtPostalCode.Text  ;
            r["vendor_email"]=TxtEmailID.Text  ;
            r["vendor_contact_no"]=TxtContactNo.Text  ;
            r["vendor_create_date"] = System.DateTime.Now.ToString();
            r["vendor_create_by"] = Session["user_id"].ToString();

            dt.Rows.Add(r);
        }
        else
        {
            r = dt.Rows[0];
            r["vendor_name"] = TxtVendorNAme.Text;
            r["vendor_address"] = TxtAddress.Text;
            r["vendor_city"] = TxtCity.Text;
            r["vendor_state"] = TxtSTate.Text;
            r["vendor_postal_code"] = TxtPostalCode.Text;
            r["vendor_email"] = TxtEmailID.Text;
            r["vendor_contact_no"] = TxtContactNo.Text;
        }

        da.Update(dt);

        Response.Redirect("vendors.aspx");
```

```
    }

    protected void Dg1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg1.DataSource = ViewState["v_table"];
        Dg1.PageIndex = e.NewPageIndex;
        Dg1.DataBind();
    }
    protected void Dg1_RowDataBound(object sender, GridViewRowEventArgs e)
    {
        if (e.Row.RowIndex < 0)
        {
            return;
        }
        string s1 = "";// e.Row.Cells[0].Text;
        s1 = "<a href='vendors.aspx?id=" + e.Row.Cells[0].Text + "'><font
color='black'><u>" + e.Row.Cells[0].Text + "</u></font></a>";
        e.Row.Cells[0].Text = s1;
    }
}
```

8.15 eAuction Admin - Reports

reports.aspx



reports.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class admin_reports : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
```

```
    {
                if (Session["user_id"] == null)
        {
            Session["admin_ret_url"] = "reports.aspx";
            Response.Redirect("default.aspx");
            return;
        }
        if (Page.IsPostBack == false)
        {
            sp_welcome.InnerHtml = "Welcome back <strong>" +
Session["user_name"].ToString() + "</strong>";
            string ut, id;
            id = Session["user_id"].ToString();
            ut = Session["user_type"].ToString();


            string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = connectionInfo;
            if (cn.State == ConnectionState.Closed)
            {
                cn.Open();
            }

            string s1;
            s1 = System.DateTime.Now.Year.ToString() + "-" +
System.DateTime.Now.Month.ToString() + "-" + System.DateTime.Now.Day.ToString()
+ " 00:00";
            TxtDateFrom.Text = s1;

            s1 = System.DateTime.Now.Year.ToString() + "-" +
System.DateTime.Now.Month.ToString() + "-" + System.DateTime.Now.Day.ToString()
+ " " + System.DateTime.Now.Hour.ToString() + ":" +
System.DateTime.Now.Minute.ToString();
            TxtDateTo.Text=s1;

            string SqlStr;
            SqlStr = "select mem_id as [Id], mem_email_id as [Email], mem_name
as [Name], mem_join_date as [Date] from Member_master "
            + " where mem_join_date between '" + TxtDateFrom.Text + "' and '" +
TxtDateTo.Text + "' order by mem_id ";

            SqlDataAdapter da = new SqlDataAdapter(SqlStr, cn);
            DataTable dt = new DataTable();
            da.Fill(dt);

            sp_mem.InnerHtml = dt.Rows.Count.ToString() + " Members Registred";
            ViewState["c_table"] = dt;
            Dg1.DataSource = ViewState["c_table"];
            Dg1.DataBind();


            SqlStr = "select checkout_id as [Id], mem_name,checkout_date as
[Date], checout_transaction_id as [Tran Id], checkout_amount as [Amt] "
            + "from checkout_master,Member_master where checkout_mem_id= mem_id
"
```

```
            + "and  checkout_date   between '" + TxtDateFrom.Text + "' and '" +
TxtDateTo.Text + "' order by mem_id ";

            da = new SqlDataAdapter(SqlStr, cn);
            dt = new DataTable();
            da.Fill(dt);

            sp_tran.InnerHtml = dt.Rows.Count.ToString() + " Transcation
Found";
            ViewState["t_table"] = dt;
            Dg2.DataSource = ViewState["t_table"];
            Dg2.DataBind();

        }
    }
    protected void CmdUpdate_Click(object sender, EventArgs e)
    {
        string connectionInfo =
ConfigurationManager.AppSettings["ConnectionInfo"];
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = connectionInfo;
        if (cn.State == ConnectionState.Closed)
        {
            cn.Open();
        }

        string SqlStr;
        SqlStr = "select mem_id as [Id], mem_email_id as [Email], mem_name as
[Name], mem_join_date as [Date] from Member_master "
        + " where mem_join_date between '" + TxtDateFrom.Text + "' and '" +
TxtDateTo.Text + "' order by mem_id ";

        SqlDataAdapter da = new SqlDataAdapter(SqlStr, cn);
        DataTable dt = new DataTable();
        da.Fill(dt);

        sp_mem.InnerHtml = dt.Rows.Count.ToString() + " Members Registred";
        ViewState["c_table"] = dt;
        Dg1.DataSource = ViewState["c_table"];
        Dg1.DataBind();


        SqlStr = "select checkout_id as [Id], mem_name  as [Member],
checkout_date as [Date], checout_transaction_id as [Tran Id], checkout_amount
as [Amt] "
        + "from checkout_master,Member_master where checkout_mem_id= mem_id "
        + "and  checkout_date   between '" + TxtDateFrom.Text + "' and '" +
TxtDateTo.Text + "' order by mem_id ";

        da = new SqlDataAdapter(SqlStr, cn);
        dt = new DataTable();
        da.Fill(dt);

        sp_tran.InnerHtml = dt.Rows.Count.ToString() + " Transcation Found";
        ViewState["t_table"] = dt;
        Dg2.DataSource = ViewState["t_table"];
        Dg2.DataBind();
    }
```

```
    protected void Dg1_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg1.DataSource = ViewState["c_table"];
        Dg1.PageIndex = e.NewPageIndex;
        Dg1.DataBind();
    }
    protected void Dg2_PageIndexChanging(object sender, GridViewPageEventArgs
e)
    {
        Dg2.DataSource = ViewState["t_table"];
        Dg2.PageIndex = e.NewPageIndex;
        Dg2.DataBind();
    }
}
```

9. **Test Plan**

Introduction

This document describes the user acceptance test plan for the eAuction Application. The complete test strategy for the eAuction Application is to perform the following kinds of tests, in sequence:

1. **Component testing** of each component that makes up the eAuction Application
2. **Integration testing** of the eAuction Application, to ensure the correct interworking of its components
3. **Validation testing** of the eAuction Application, to ensure that it works correctly in a pseudo-live environment
4. **User acceptance testing** of the eAuction Application, to ensure that its function is acceptable to its users

Acceptance testing is the last set of tests to be performed before the application goes officially live.

9.1 Test Scope

The scope of the user acceptance testing covers:

- Version 1 of the eAuction Application
- User-facing functionality defined by a set of use cases
- Administrator-facing functionality defined by a set of use cases

The aim of the testing is to determine how well the application meets its functional requirements from the perspective of the user, and to identify any issues so they can be resolved. Also, the testing serves to compile a set of test data and results that can be used during subsequent test cycles, to test for non-regression of the software in later releases or after the application is in maintenance.

Working practices might vary from user to user and are considered outside the scope of the testing.

9.2 Test Strategy

The basis of user acceptance testing is that other tests were completed successfully, so the application and its required infrastructure are considered to be stable and reliable. Acceptance testing concentrates on the application from the user's perspective, that is, how the application is used and whether it meets the necessary quality criteria.

Change requests will be sent to the development team as the actionable documentation. Change criteria will be determined by the Test team and the Development team prior to the beginning of testing. For instance, criteria may include *impact to desired functionality, amount of code impacted by proposed change,* and *design required by proposed change*. The tester will evaluate the criteria. The test lead will determine Change Required or not. Once a bug has been determined as Change Required, the bug report will be translated into a Change Request and passed on to development.

The Member of the acceptance testing is the System Users, Supervisor, Manager and MIS Executive for eAuction Application. The progress of the acceptance testing will be reported to the Member, together with any issues that are discovered and their planned resolutions. Sign-off of the tests, and therefore the acceptance of the application, will be performed by the Member or a selected representative.

9.3 Preconditions

The following items are required before testing can take place:

- A complete and coherent functional specification of the eAuction Application expressed as use cases and usage scenarios
- A complete and validation-tested release of eAuction Application, delivered according to the delivery plan
- An agreed-upon procedure for dealing with any anomalies that are discovered during the testing process
- A set of test specifications describing how each functional area of the eAuction Application is to be acceptance tested
- An implemented test environment for the testing
- Sufficient, suitable resources to carry out the testing
- Available standards for the acceptance testing

9.4 Test Priorities

During testing of the eAuction Application, the following qualities will be tested in order of priority:

- Functionality—whether the required functions are available and working as expected
- Usability—how user-friendly and intuitive the eAuction Application is
- Security—how well-protected and guaranteed corporate and user data is
- Performance—whether the response times are within acceptable limits
- Customization—how straightforward it is to use the application in new, unpredicted ways

9.5 Test Techniques

The following techniques will be applied:

- Scripted tests—sequences of user interactions (based on the use case and usage scenarios) using predefined data sets against predicted results
- Unscripted tests—based on scripted tests, the tester tries to modify the scenarios to explore what-if possibilities
- Penetration tests—scripted tests to attempt unauthorized entry into the system
- Usability checklists—tests to determine the complexity of interactions
- Performance statistics—generation of performance information to check against desired performance criteria

9.6 Test Organization

Roles and Responsibilities

The following roles are defined:

- QA lead/test manager—responsible for planning and ensuring the smooth running of the test process
- Tester—carries out the tests according to the test plan, and then reports the results
- Product manager—ensures that the tests are carried out successfully from a user perspective
- Project sponsor/client—acts as main stakeholder, and ensures that the needs of the Member community as a whole are considered
- Test support—provides technical assistance, such as test environment configuration, and non-technical assistance, such as methodological support

Weekly team meetings will be held involving the test manager, testers, and product managers. At these meetings, the progress of the testing process will be reported, any issues will be discussed, and actions will be agreed upon.

9.7 Deliverables

The following deliverables will be expected from the user acceptance testing process:

- Test plan—this document, together with any updates that have occurred during the testing process
- Change requests—any bugs, defects, or other changes required to the eAuction Application as a result of the testing process
- Weekly reports—progress reports to enable the status of the testing process to be determined
- Completion report—a report to be signed off by the Member, to signify the successful completion of the user acceptance testing

9.8 Test Environment

Hardware and Software

The test environment will consist of:

Server

A single Intel-based computer running:

- Microsoft Windows
- eAuction Application components

Client Workstations

Two Intel-based client laptop computers, each running:

- Microsoft Windows XP Professional
- Microsoft Office
- Internet Explorer 6 or greater

The following additional hardware will be required:

- One laser printer to print reports
- One color printer (laser or inkjet) to print screen dumps
- One CD-ROM drive to enable clean installation of the eAuction Application
- Networking connectivity to permit interconnection of the server, clients.

9.9 Testing Automation Software

No testing automation software packages are selected at present.

9.10 Application Configuration

The following user accounts will be configured on the server:

- System Administrator
- System Users 1
- System Users 2
- Member 1
- Member 2

9.11 Test Management

Tests shall be managed according to the corporate test management standards, which cover:

- Conduct of tests
- Reporting of test results
- Defect tracking and resolution
- Configuration management of the test environment
- Configuration control of test deliverables.

9.12 Testing Schedules

The user acceptance testing schedules are shown in the project structure document and resulting Gantt charts.

9.13 Threats to Testing

Potential threats to the testing process are as follows:

- **Insufficient resources available for testing.** Testing resources have been seconded from the development departments, whose time is at a premium. Mitigation: ensure department heads apply a high priority to the testing of the eAuction Application.
- **Availability of sales personnel for testing.** The test team should be overseen by at least one administrator.

10. Future Enhancement

This project was developed to fulfill user requirement; however there are lots of scope to improve the performance of the eAuction Application in the area of user interface, database performance, and query processing time. Etc.

So there are many things for future enhancement of this project. The future enhancements that are possible in the project are as follows.

- Linking and integration of any legacy system for accounting.
- Integration with vendor database through Web Services
- Connection to third-party OLAP applications
- Electronic Data Interchange (EDI) system between purchasing department and their vendors
- In the area of data security and system security.
- Provide more online tips and help.
- To optimize the query which is embedded in the system.

11. Bibliography

11.1 Websites

Following websites are referring to create this project reports.

- http://www.google.com
- http://www.microsoft.com
- http://www.programmer2programmer.net
- http://www.codeproject.com
- http://www.asp.net
- http://www.asp123.com
- http://www.wikipedia.org

11.2 Books

Following books and ebook are used to complete this project reports.

- Mastering C# (Paperback)
- SQL Server Bible (Paperback)
- .NET Black Book (Paperback)
- Professional C#, 2nd Edition (Paperback)
- Professional ASP.NET (Paperback)
- MCAD/MCSD Self-Paced Training Kit: Developing Web Applications with Microsoft® Visual Basic® .NET and Microsoft Visual C#® .NET, Second Edition
- MCAD/MCSE/MCDBA Self-Paced Training Kit: Microsoft SQL Server 2000 Database Design and Implementation, Exam 70-229, Second Edition