Shri Shivaji Science & Art College ,Chikhli
Department of Computer Science
Academic Year 2021-2022
Project Review – I

Date – 15/03/2022

Group	Student Name	Signature	Project Remark
Group 1	1. Dipali Bhaskar Gadge	Dgadge	Good
	2. Kiran Vithoba Solanki	K.V.Solanki	
Group 2	✓3. Shubhangi Sahebrao Mutthe		Very Good.
	✓4. Vaishnavi Vasantrao Jumde	V.V.Jumde	
Group 3	5. Mayur Gajanan Umre	MUmre	Excellent
	6. Rishikesh Vijay Chechare	Chechare	
	7. Vishal Prabhakar Dhare		

15/03/2022.

Head
Dept. Of Computer Science
Shri Shivaji Sc. & Arts College
Chikhli, Dist :- BULDANA (M.S.)

Shri Shivaji Science and Arts College, Chikhli
Department of Computer Science
M.Sc II Year-(Sem IV)
Academic Year 2021-22

Practical Submission

Sr. No.	Student Name	Practical Name	
		LAB VII	LAB VIII (Project)
01	DIPALI BHASKAR GADGE	Bgadge	Sign & Gesture Recognition System using Image processing.
02	KIRAN VITHOBA SOLANKI	K.V. Solanki	—〃—
03	MAYUR GAJANAN UMRE	MUmre	Bidding
04	RISHEKSH VIJAY CHECHARE	Chechare	Bidding
05	SHUBHANGI SAHEBRAO MUTTHE		Hotel Booking & Management System
06	VAISHNAVI VASANTRAO JUMDE	V.V. Jumde	—〃—
07	VISHAL PRABHAKAR DHARE		Bidding

Place – Chikhli
Date - 15/03/2022.

Dr.A.B. Kadam
Head of The Department

Head
Dept. Of Computer Science
Shri Shivaji Sci. & Arts College
Chikhli, Dist :- BULDANA (M.S.)

SANT GADAGEBABA AMARVATI UNIVERSITY, AMARAVTI

Shri Shivaji Science & Arts College Chikhli, Dist. Buldhana(M.S.)

A PROJECT REPORT ON

“BIDDING”

Submitted by

Rishikesh V. Chechare

Mayur G. Umre

Vishal p. Dhare

M.Sc Computer Science

Guided By

Dr. A.B.Kadam

2021-2022

Shri Shivaji Science & Arts College Chikhli

Department of Computer Science

Certificate

|| ज्ञानम् परम ध्येयम् ||

This is to certify that Mr. ***Rishikesh Vijay Chechare*, Mayur G. Umre, Vishal P. Dhare, *M.Sc (Computer Science)II Year*** has satisfactorily completed **Project** entitled **"Online Bidding Management System "** for the partial fulfilment of Master in Computer Science Sant Gadage Baba Amravti University, Amaravti for academic year 2021-2022.

Project Guide

Dr.A.B.Kadam 27/05/2022.

Head of Department

Dr.A.B.Kadam 27/05/22.

Head
Dept. Of Computer Science
Shri Shivaji Sci. & Arts College
Chikhli, Dist :- BULDANA (M.S.)

External Examiner

DECLARATION

I hereby that the present project work in the form entitled "**Online Bidding management System**", is an original work carried out by me under the guidance of **Dr.A.B. Kadam** , **H.O.D, Department of Computer Science**, Shri Shivaji Science & Arts College Chikhli for the submission of degree of Master in Computer Science to Sant Gadge Baba Amravti University,Amaravti. I further declare that to the nest of my knowledge the project does not contain any part of any work, which has been submitted for the award of any Degree, either in this University or any other University without proper citation.

I further declare that the material referred form other sources has duly acknowledged in the Project.

ACKNOWLEDGEMENT

We avail this to express out deep sense of gratitude and whole hearted thankful to our guidance **DR.A.B.Kadam** sir for this valuable guidance, suggestion and encouragement to embark this project.

We are also thankful to our Honorable principle **Dr.O.S. Deshmukh** who for providing us that kind of Opportunity.

We are also thankful to our Computer Science department staff Professors **Dr. Mukti Jadhav, Dr.S.S.Gaikwad, Mr.S.D.Burkul and Mr. S.A.Chavan & P.S. Sonune (Lab Attendant)** to help us in this Project complication.

Yours Faithfull
Rishikesh v. Chechare
Mayur G. Umre
Vishal P. Dhare

Mumre

ABSTRACT

Biddings are among the oldest economic institutions in place. They have been used since antiquity to sell a wide variety of goods, and their basic form has remained unchanged. In this dissertation, we explore the efficiency of common Biddings when values are interdependent- the value to a particular bidder may depend on information available only to others-and asymmetric. In this setting, it is well known that sealed-bid Biddings do not achieve efficient allocations in general since they do not allow the information held by different bidders to be shared.

Typically, in a Bidding, say of the kind used to sell art, the Bidding sets a relatively low initial price. This price is then increased until only one bidder is willing to buy the object, and the exact way this is done varies. In my model a bidder who drops out at some price can “reenter” at a higher price.

With the invention of E-commerce technologies over the Internet the opportunity to bid from the comfort of one’s own home has seen a change like never seen before. Within the span of a few short years, what may have begun as an experimental idea has grown to an immensely popular hobby, and in some cases, a means of livelihood, the online Bidding gathers tremendous response every day, all day. With the point and click of the mouse, one may bid on an item they may need or just want, and in moments they find that either they are the top bidder or someone else wants it more, and you’re outbid! The excitement of a Bidding all from the comfort of home is a completely different experience.

The levels of comfort may rise soon but the rules to be followed remain the same. In fact, may rise with the new technologies.

Society cannot seem to escape the criminal element in the physical world, and so it is the same with online Biddings. This is one area wherein a question can be raised as to how safe online Biddings.

INDEX

Chapter 1
Introduction

INTRODUCTION

we are developing a software for online-Bidding. online-Bidding is known by several names, including ‘electronic reverse bid Auctions’, ‘reverse auctions’ or simply ‘e-Auctions’. Online Bidding is a group which is based for auction. If you want to something sells by Bidding then you post that on website. It just a selling of products. This project user can do a bid on products. The main objective of the e-Bidding process must be to obtain best value and the highest price. It cannot be possible to achieve best value outcomes whilst the focus remains on price. There are two categories of persons one is customers and second one is vendor. Vendor can sell his products on this website and customer will purchase it. Product will be given to those customers who put a high bid price on product to purchase. One user can do a more than one bid on products. There are two categories of persons one is customers and second one is vendor. Both have their own registration form. There is a common login page for vendors and for users but when both login in, it would be easily found out that ether it is a vendor or a user. There are two home pages i.e., one page behaves different if vendor login then show the vendor’s menu or if customer will login then customer menu will show. Customers should have an appropriate knowledge of the e- Bidding subject its market, its market price. In this website anyone wants to sell products will have to register first then a unique id is given to the registered users. After registered user have to give details of their products like (price of product, at what price he/she wants to start the bid, image of the product) if he/she wants to sell his/her product .

Product will be given to that customer who put a high bid price on product to purchase. One user can do a more than one bid on products. After completion of bidding process all the bids are check and the product will give to that customer for purchase who has a high bid price. The bidding exercise will have a specified opening and closing time. The E-Bidding will close when no new bids are received, and the closing time has expired. If, however, a new (lower) bid is received just before the Scheduled closing time, the allocated bidding time will be extended.

If user have problem related and wanted to know that what is e-Bidding they can contacts the experts through contact details given on website. Online Bidding is loaded with features and functionality that allows you to create a Bidding that is powerful and useful for buyers and sellers.

Chapter 2
About Project

ABOUT PROJECT

Bidding:

A Bidding is a sale in which a seller presents his product on a public platform/ forum. The selling price in a Bidding is determined by the bids made by interested buyers. The price they bid is based on their own valuation of, and need for, the product. The product is sold to the highest bidder. A potential buyer participates by bidding on an item that a seller has listed. The person who has offered the highest bid at close of Bidding wins the right to purchase the item at that price.

Bid:

A bid is the amount of money proffered for an item that has been put on sale in a Bidding. The bidder competes with other potential buyers, keeping in mind that the buyer with the highest bid is obliged to complete the purchase with the seller. In other words, your bid tells other buyers, "I want to buy this item at this price."

Win:

You win a Bidding by placing the winning (read highest) bid and obtaining the item on Bidding. The person with the highest bid is the winner of the Bidding. Now that you're familiar with how a Bidding generally works.

PROBLEM IDENTIFICATION

At the end of every Bidding, you would receive an email (if your bid is the highest at that point of time) with your seller's contact details. You would then need to contact your seller to arrange for the delivery and complete the deal offline.

A bidder can know about the item he/she won in a Bidding once the Bidding of an item you have bid on has ended. You will receive an auto-generated email if your bid was the highest and the reserve price (if any) had been met. The mail will include item details and the seller's contact details. The seller too will receive an email providing them with your contact details. You will need to contact your seller to inspect the goods and arranged delivery and payment. Once the Bidding is ended, both the seller and the buyer are notified through email and given each other's contact details. They need to contact each other and work out payment as per the terms of payment mentioned by the seller on the item page.

PRELIMINARY PLANNNG

Existing System

The transaction between the buyer and the seller can be carried out manually if the buyer and seller are in a reachable distance. In other case the transaction is carried out over the net via E-commerce Secured transaction systems.

Society cannot seem to escape the criminal element in the physical world, and so it is the same with online Biddings.

One of the concepts that the online Biddings were started with was the idea that people could police themselves, but those with the intent to take advantage of others have found the means to bypass this mild safeguard as well. It was envisioned that with the opportunity to give 'feedback' on each other as buyers and sellers, honesty would remain intact. However, some have found a way to inflate their ratings which makes them appear trustworthy to do business with.

There are steps to take to reduce the risk of being scammed by a less-than-honest individual in an online transaction, and the first step is to deal only with reputable establishments. At least if there is a problem with the transaction, whether as the buyer or the seller, one stands a better chance of resolving the problem satisfactorily for everyone. So do your homework as to where it is safe to do business. Establish the top price that you are willing to give, keeping in mind how much more than the bid price you will have to give to cover shipping, insurance and tax. If possible, establish the identity of whom you are dealing with. Have they been operating with this Bidding service for an acceptable length of time? Do they have any comments about them, either positive or negative? These things must be considered to protect oneself from the project.

Proposed System and its need

The opportunity to shop from the comfort of one’s own home has seen a change like never seen before. Within the span of a few short years, what may have begun as an experimental idea has grown to an immensely popular hobby, and in some cases, a means of livelihood, the online Bidding gathers tremendous response every day, all day. With the point and click of the mouse, one may bid on an item they may need or just want, and in moments they find that either they are the top bidder or someone else wants it more, and you’re outbid! The excitement of a Bidding all from the comfort of home. One can have a positive experience doing business on an online Bidding, and each party can leave satisfied with their ‘deal’, whether they are the buyer, the seller, or the conveyance that brought it all together.

Chapter 3
Objective of the Project

Objectives of the System

The process of an online Bidding is much the same as a live Bidding. This means that users place bids for items, and the goods get sold to the highest bidder. You are notified through email on the status of your bids, which is when you place a bid, when you've been outbid and when you've won an item.

To bit for an item, there is a bidding form through which you may place bids on the item. To bid on an item, enter your bid amount. While entering your bid, you need to consider the bid increment. The bid increment is the amount by which each bid increases. The seller sets this amount. The bid amount should be one bid increment or more above the current leading bid specified on the item page.

Automatic Bids and manual bids need not be exact multiples of the bid increment amount. Bids will only have to be one increment or more above the current bid to be accepted.

Example:

Bid increment amount - Rs. 20

Lets say the starting price is Re.1

Current highest bid is Rs. 21

Now in order to bid: -You can bid at anywhere above or equal to the current highest bid amount for your bid to be accepted, i.e. you can bid at anywhere between Rs. 21 to Rs. 41 (or more) for your bid to be accepted. -This means that you can bid at say Rs. 23 and the next current minimum bid will get adjusted to Rs. 43 (since the Bid Increment level is set to 20).

If you wish to place a higher bid, you may do so. For eg. - You can place a bid of Rs. 45. This would incorporate the current bid at Rs. 21 and the minimum bid increment of Rs. 20. The remaining Rs. 4 is the additional amount you have bid. Therefore, the user who bids after you would have to bid at least Rs. 65

(Rs. 45 of your bid and Rs. 20 of the bid increment)

The amount of the next minimum acceptable bid will be displayed in the "Bid Amount per item" text box.

To find more information about the product the description of that product must be referred. A description of every item is put up on the item page. This way you can acquire more information about the item. If it is convenient for you and the seller, you can also arrange to personally inspect the item.

To bid on multiple items is a special Bidding where a seller has more than one quantity of the item he or she wants to sell/bid. The seller selects the starting bid amount and indicates how many of these items are available for the Bidding.

A Bidding is a multiple Bidding or not can be known by checking the quantity box of the item. If there is number of quantities displayed is more than 1, then it is a MQA (Multiple Quantity Bidding)

After a Bidding:

At the end of every Bidding, you would receive an email (if your bid is the highest at that point of time) with your seller's contact details. You would then need to contact your seller to arrange for the delivery and complete the deal offline.

A bidder can know about the item he/she won in a Bidding once the Bidding of an item you have bid on has ended. You will receive an auto-generated email if your bid was the highest and the reserve price (if any) had been met. The mail will include item details and the seller's contact details. The seller too will receive an email providing them with your contact details. You will need to contact your seller to inspect the goods and arrange delivery and payment. Once the Bidding is ended, both the seller and the buyer are notified through email and given each other's contact details. They need to contact each other.

Chapter 4
Feasibility of Study

FEASIBILITY STUDY

Technical Feasibility

There are very few situations in which Biddings cannot be performed due to existing market conditions (e.g., in the case of a monopoly). It is more likely that the combination of different (legitimate) interests of the internally involved parties will lead to a situation in which a Bidding would seem inappropriate. By means of thorough and systematic preparation with all responsible internal parties involved, we help establish a framework in which a Bidding can be carried out, allowing the realization of the underlying savings potential. At the same time, we make sure that the company's main sourcing targets are reached. Many of our clients initially thought that commitment to the outcome of a Bidding would reduce their flexibility. They very soon concluded that systematic preparation only shifts the required flexibility to a position closer to the beginning. If a certain result is not to be supported, we make sure - by systematic preparation and careful Bidding design - that it will never occur.

Economic Feasibility:

Economic Feasibility means the cost of understanding project should less cost than the existing system. Electronic mailing system is economically feasible, because it reduces the expenses in the system.

Operational Feasibility:

Operational Feasibility means users should support the project. It should not cause any problems to users after implementing. Electronic mail was fully supported by users. So, Electronic mailing system is operationally feasible.

Chapter 5
System Analysis & Design

System Analysis

Analysis of Present System

The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as SQL Server. The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. The SQL server was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using the c#.net technologies. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations. The database connectivity was planned to use the latest “Sql Connection” technology provided by Microsoft Corporation. The authentication and authorization were crosschecked at all the relevant stages.

Analysis of System Requirements

Object Oriented Analysis:

Analysis is the process of extracting the needs of a system and what the system must do to satisfy user requirements. The goal of Object-Oriented analysis is to first understand the domain of the problem and the systems responsibilities by understanding how the users or will use the system. OOA process consists of the following steps:

1. Identify the actors.
2. Develop a simple business process model using UML activity diagram.
3. Develop the use case.
4. Develop interaction diagram.
5. Identify classes.

Object Oriented Design:

Unified approach combines the high level models of Ram Baugh and Jacobson's analysis and interaction diagrams. Booch's Object diagrams, and Ram Baugh's Domain models and finally following the Jacobson's life cycle model can produce designs that are traceable across requirements, analysis, design, coding and testing. OOD

Process consists of:

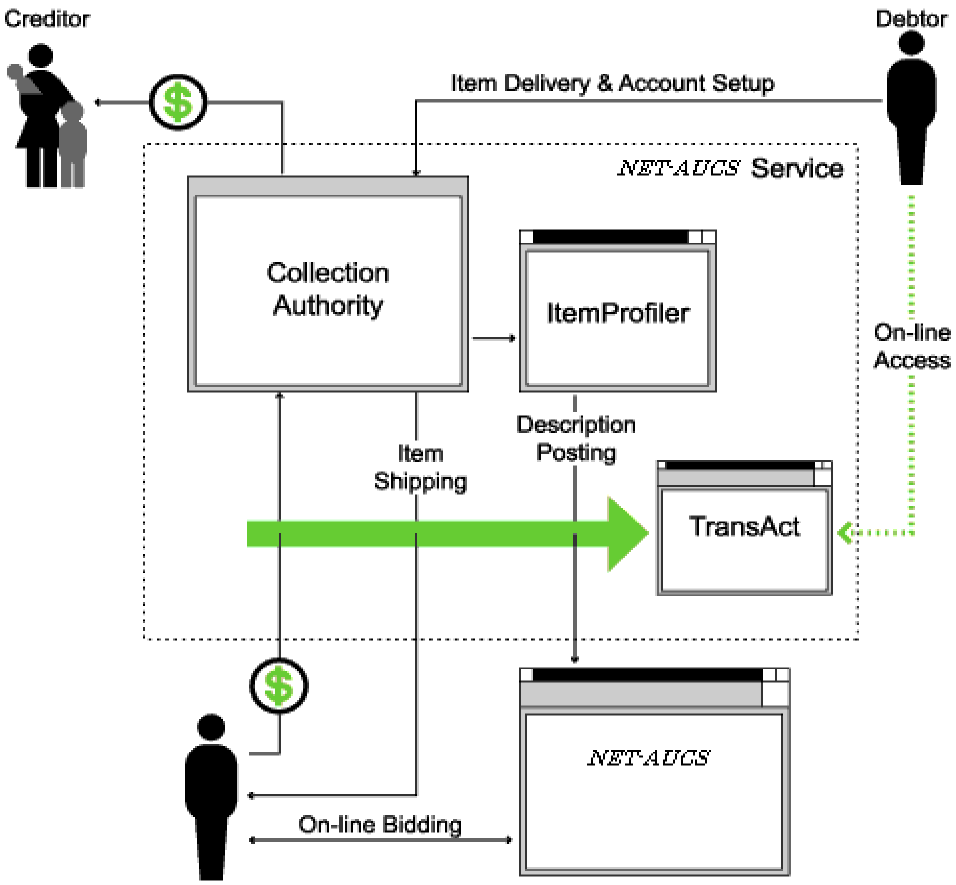
1. Designing, classes, their attributes, methods, associations, structures, and protocols, apply design axioms.
2. Design the access layer
3. Design and Prototype user interface.
4. Iterate and refine the design.

System Design

People tend to frequent seized property Biddings because goods can be purchased at a low price relative to market value. This is good for the buyer, but not for the debtor whose budget and credit record are dependent on this money. A limited audience who will not pay the full value of the items up for bidding attends the Biddings. The body appointed to seize and Bidding property operates it.

When debtors must liquidate their assets, they must first open an account with the Collection Authority. The Collection Authority is housed inside the sheriff's office or the organization responsible for public Biddings of seized property and assets. It is a combination of customer service counters and an inventory/shipping warehouse. Seized goods and items brought in by a debtor are inventoried at the Collection Authority where daily shipments are made for the items sold. Once the item is sold, the buyer makes the payment to the Collection Authority, which then transfers the balance owed to the creditor, minus shipping costs and a percentage charged by the Collection Authority.

All items going through site, either seized or brought in by the debtor, use the Item Profiler to create a standard description page to post items on the on-line Bidding site. The page layout includes digital images representing the orthographic views of the item plus a text section detailing its make, model, year, history, etc. with an additional line worded by the debtor.



The item’s description is logged under the appropriate category on the Bidding site. People shopping at this site can view items and bid on the goods similar to any other item listed on the site. The only discernible difference is the consistent layout between the posted items and a line attached by the Collection Authority stating that the item is being sold to resolve a debt.

TransAct records all of the items posted for Bidding by item number and account number, and records the sales made on these items. Beyond the bookkeeping value of these records, summary and status statements can be provided for the debtor to present to the court if a question of compliance arises; an account number enables the system to summarize the debtor’s TransAct ion to show remaining debt to the creditor as well as prove to the court that an effort is being made to pay the debt.

Scenario

Open Multithreaded Transactions form an advanced transaction model that provides features for controlling and structuring not only accesses to objects, as usual in transaction systems, but also threads taking part in transactions. The model allows several threads to enter the same transaction in order to perform a joint activity. It provides a flexible way of manipulating threads executing inside a transaction by allowing them to be forked and terminated, but it restricts their behavior in order to guarantee correctness of transaction nesting and isolation among transactions. In addition, transactions are exception handling contexts, and the model therefore provides forward and backward error recovery. In this paper we show that the model is indeed powerful, and that a complex application, i.e. an online Bidding system, can be designed and implemented in a very elegant way.

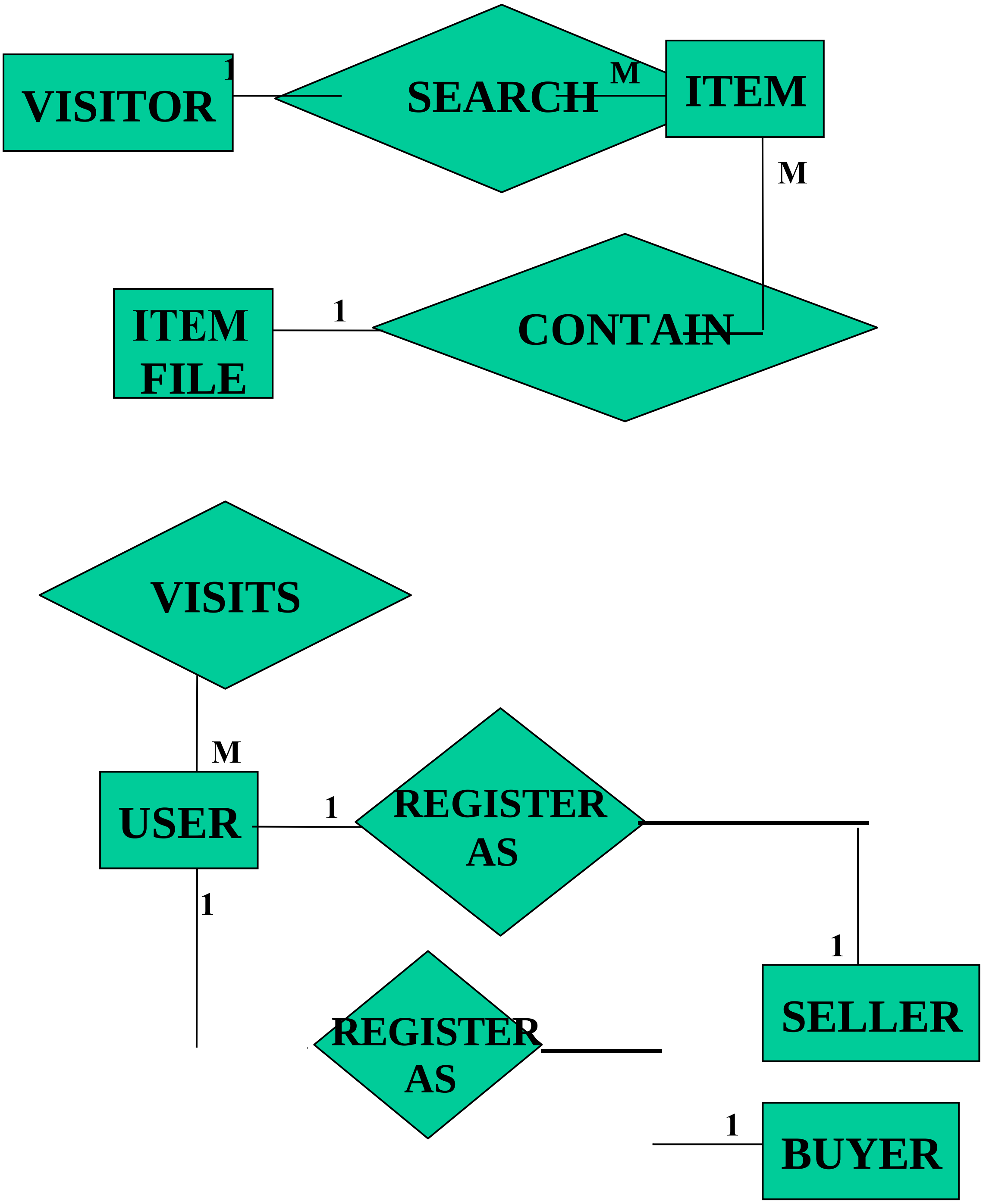
Chapter 6
E-R Diagram & Data Flow Diagram

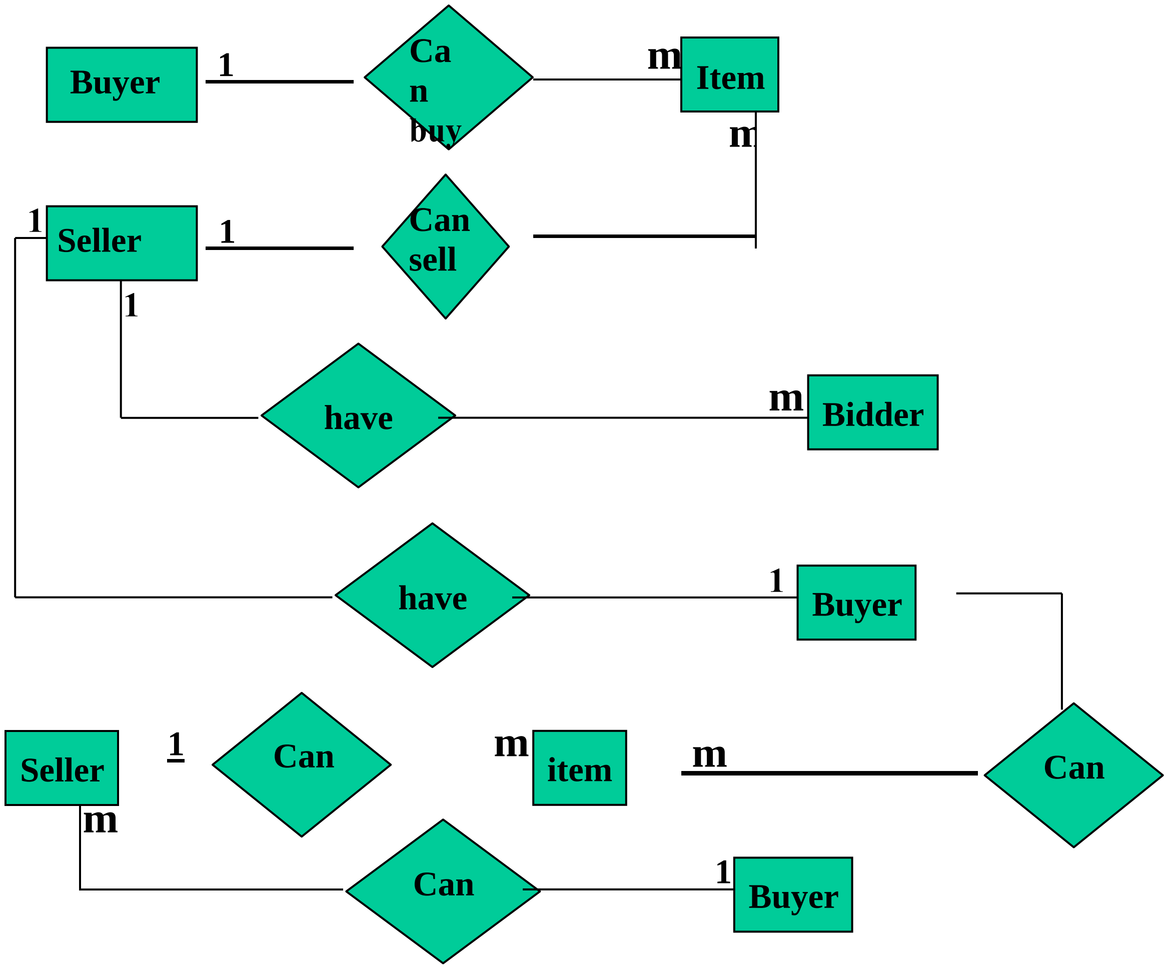
E-R Diagrams:

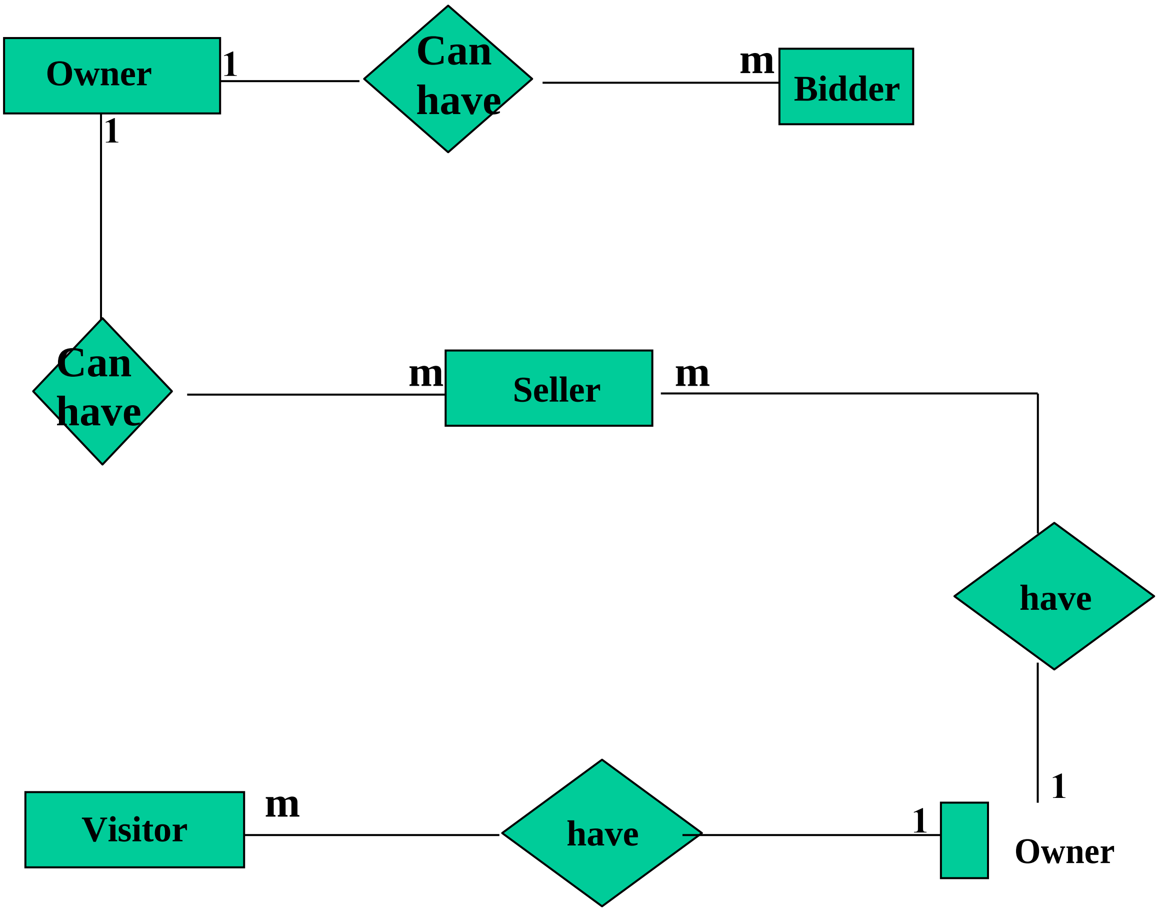
E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

The symbols used in E-R diagrams are:

SYMBOL	PURPOSE
▭	Represents Entity sets.
⬭	Represent attributes.
◇	Represent Relationship Sets.
――	Line represents flow


VISITOR
1
SEARCH
M
ITEM
M
ITEM FILE
1
CONTAIN
VISITS
M
USER
1
REGISTER AS
1
1
SELLER
REGISTER AS
1
BUYER


Buyer
1
Ca
n
buy
m
Item
m
1
Seller
1
Can
sell
1
have
m
Bidder
have
1
Buyer
Seller
1
Can
m
item
m
Can
m
Can
1
Buyer


Owner
1
Can have
m
Bidder
1
Can have
m
Seller
m
have
1
Visitor
m
have
1
Owner

Data Flow Diagrams:

The data flow diagram is used for classifying system requirements to major transformation that will become programs in system design. This is starting point of the design phase that functionally decomposes the required specifications down to the lower level of details. It consists of a series of bubbles joined together by lines.

Bubbles: Represent the data transformations.

Lines: Represents the logic flow of data.

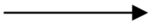
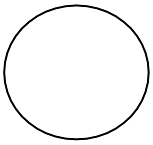
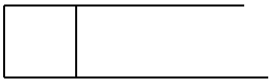
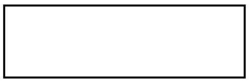
Data can trigger events and can be processed to useful information. System analysis recognizes the central goal of data in organizations. This dataflow analysis tells a great deal about organization objectives are accomplished.

Dataflow analysis studies the use of data in each activity. It documents this finding in DFD's. Dataflow analysis give the activities of a system from the viewpoint of data where it originates how they are used or hanged or where they go, including the stops along the way from their destination. The components of dataflow strategy span both requirements determination and system's design. The first part is called dataflow analysis.

As the name suggests, we didn't use the dataflow analysis tools exclusively for the analysis stage but also in the designing phase with documentation.

NOTATIONS USED IN DATA FLOW DIAGRAMS

The logic dataflow diagrams can be drawn using only four simple notations i.e., special symbols or icons and the annotation that associates them with a specific system. Since the choice of notation we follow, does not affect impede or catalyze the system process; we used three symbols from YOURDON notation and one from Gain and Sarson notation as specified below.

Element References	symbols
Data Flow Process	
Process	
Data Store	
Source or Sink	

Description:

Process: describes how input data is converted to output Data

Data Store: Describes the repositories of data in a system

Data Flow: Describes the data flowing between process, Data stores and external entities.

Sources: An external entity causing the origin of data.

Sink: An external entity, which consumes the data.

Context Diagram:

The top-level diagram is often called a “*context diagram*”. It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

Types of data flow diagrams

DFDs are two types

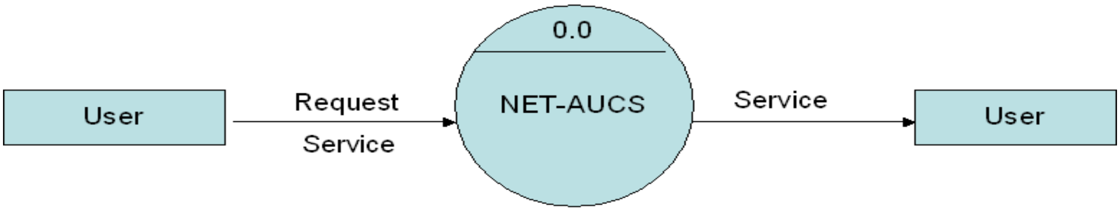
1. Physical DFD

Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensure that the current system has been clearly understood. Physical DFDs shows actual devices, departments, people etc., involved in the current system

2. Logical DFD

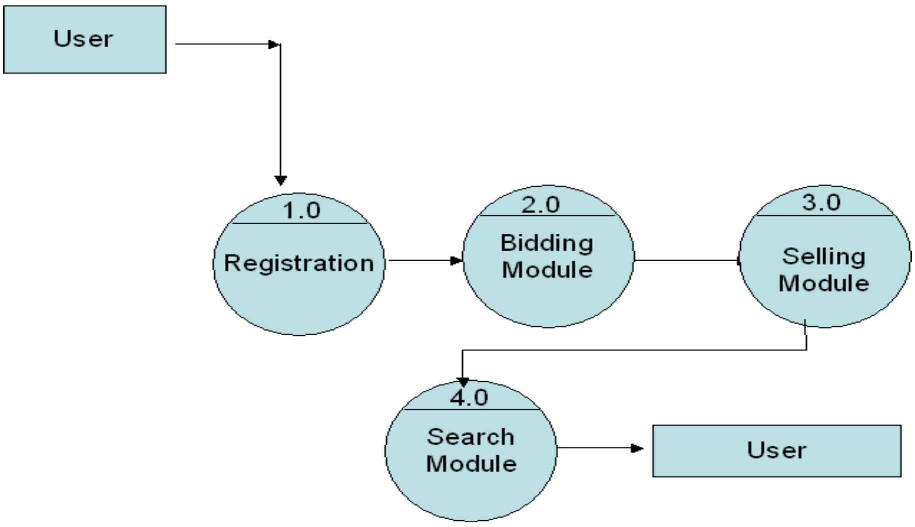
Logical Dads are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts.

NET-AUCS

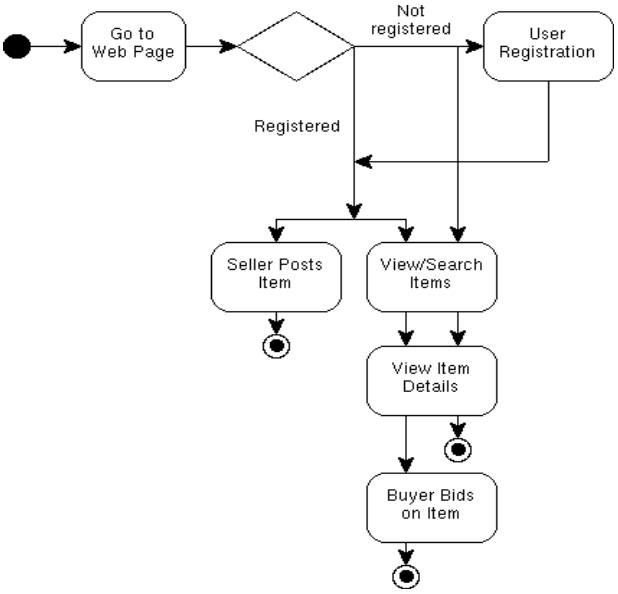


Context Diagram (Zero Level Diagram).

Level 1 Diagram.



FlowChart


Go to Web Page
Not registered
User Registration
Registered
Seller Posts Item
View/Search Items
View Item Details
Buyer Bids on Item

Chapter 7
System Implementation

SYSTEM IMPLEMENTATION

Choosing a Language:

ASP.NET

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. We can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that we can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables us to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft ActiveX controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

Features Of Commen Runtime CLR

The Common Language Runtime (CLR) is a layer between an application and the operating system it executes on. The common language runtime simplifies an application's design and reduces the amount of code developers need to write because it provides a variety of execution services that include memory management, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

The key benefit of the CLR is that it transparently provides these execution services to all applications, regardless of what programming language they're written in and without any additional effort on the part of the developer.

The CLR is also responsible for compiling code just before it executes. Instead of producing a binary representation of our code, as traditional compilers do, .NET compilers produce a representation of our code in a language common to the .NET Framework: Microsoft Intermediate Language (MSIL), often referred as IL. When our code executes for the first time, the CLR invokes a special compiler called a Just In Time (JIT) compiler, which transforms the IL into executable instructions that are specific to the type and model of our system's processor. Because all .NET languages have the same compiled representation, they all have similar performance characteristics. This means that a program written in Visual Basic.NET can perform as well as the same program written in Visual C++.NET.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

The Common Type System (CTS) is a component of the CLR and provides a common set of data types, each having a common set of behaviors. In Visual Basic, for example, the String data type maps to the CTS System.String class. Therefore, if a Jscript.NET client needs to communicate with a component implemented in VB.NET, the client doesn’t have to do any additional work to exchange information because it’s using a type common to both Jscript.NET and VB.NET. The CTS eliminates many interoperability problems that exist outside .NET.

.NET programming languages take advantage of the CTS by enabling developers to use their language’s built-in data types, the .NET compilers convert the native data types’ into their equivalent CTS types at compile time. Developers can also use CTS types at compile time.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

The runtime can be hosted by high-performance, server-side applications, such as Microsoft SQL Server and Internet Information Services (IIS). This infrastructure enables us to use managed code to write our business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

.NET Framework Class Library:

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which our own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

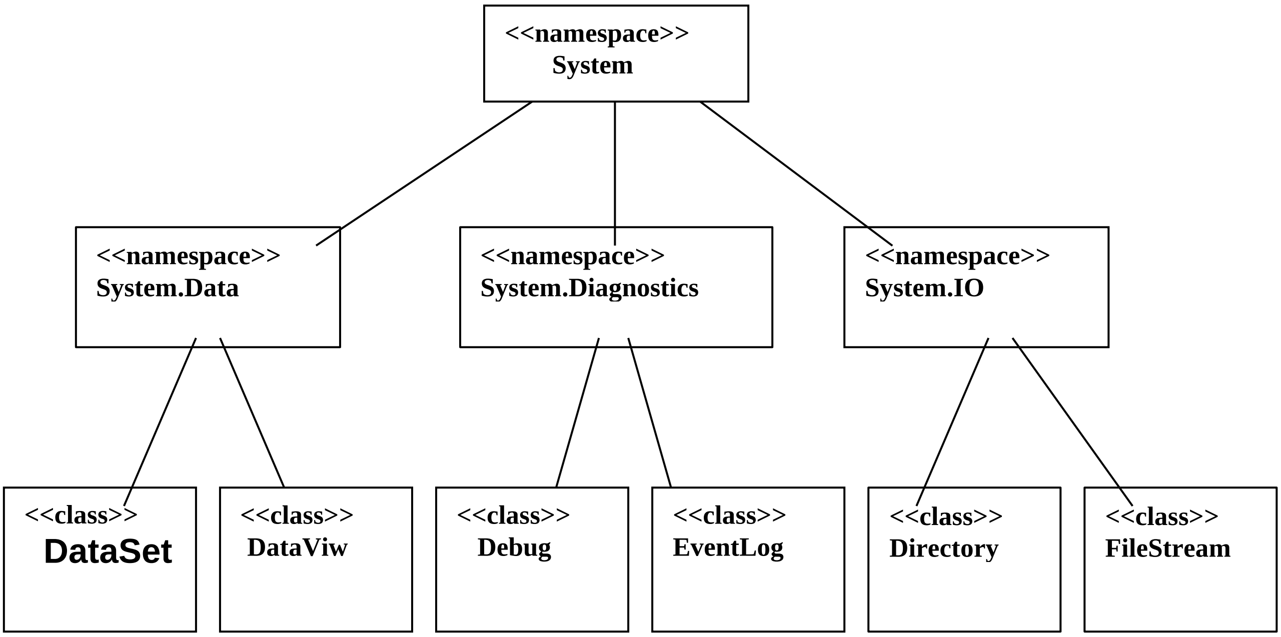
For example, the .NET Framework collection classes implement a set of interfaces that we can use to develop our own collection classes. Our collection classes will blend seamlessly with the classes in the .NET Framework.

To make .NET Class Library easier to work with and understand, it is divided into *namespaces.* The root namespace of the .NET Class Library is called ‘System’, and it contains core classes and data types, such as Int32, Object, Array and Console. Secondary namespaces reside within the ‘System’ namespace.

Examples of nested namespaces include the following:

- System.Diagnostics: Contains classes for working with the Event Log
- System.Data: Makes it easy to work with data from multiple data sources (System.Data.Oledb resides within this namespace and contains the ADO.NET classes)
- System.IO: Contains classes for working whit files and data streams.

The below figure illustrates the relationship between some of the major namespaces in the .NET Class Library.



Organization of the .NET Class Library

The benefits of using the .NET Class Library include a consistent set of services available to all .NET languages and simplified deployment, because the .NET Class Library is available on all implementations of the .NET Framework.

The .NET Framework types enable us to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, we can use the .NET Framework to develop the following types of applications and services:

- Console applications.
- Scripted or hosted applications.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.
- XML Web services.
- Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If we write an ASP.NET Web Form application, we can use the Web Forms classes.

Client Application Development:

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft Visual Basic. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. We can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

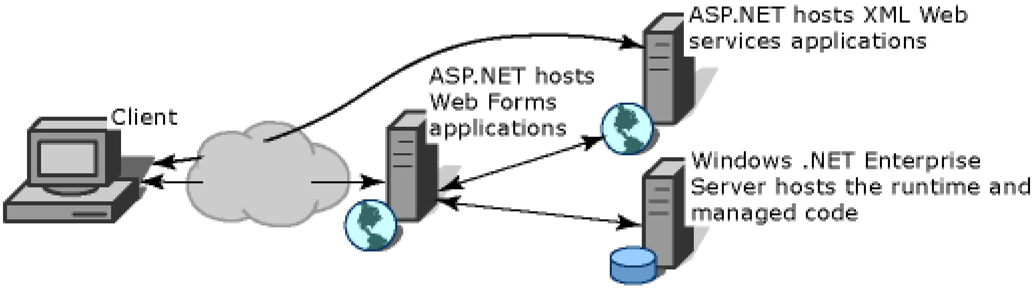
Unlike ActiveX controls, Windows Forms controls have semi-trusted access to a user's computer. This means that binary or natively executing code can access some of the resources on the user's system (such as GUI elements and limited file access) without being able to access or compromise other resources. Because of code access security, many applications that once needed to be installed on a user's system can now be safely deployed through the Web. Our applications can implement the features of a local application while being deployed like a Web page.

Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows our custom managed code to control the behavior of the server. This model provides us with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while our application logic executes through the managed code.

Server-side managed code


ASP.NET hosts XML Web services applications
ASP.NET hosts Web Forms applications
Client
Windows .NET Enterprise Server hosts the runtime and managed code

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use

IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If we have used earlier versions of ASP technology, we will immediately notice the improvements that ASP.NET and Web Forms offers. For example, we can develop Web Forms pages in any language that supports the .NET Framework. In addition, our code no longer needs to share the same file with our HTTP text (although it can continue to do so if we prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

For example, the Web Services Description Language tool included with the .NET Framework SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that our application can use to become a client of the XML Web service. The source code can create classes derived from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although we can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate our development efforts with the .NET Framework.

If we develop and publish our own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables us to focus on the logic of our service, without concerning ourself with the communications infrastructure required by distributed software development.

Like Web Forms pages in the managed environment, our XML Web service will run with the speed of native machine language using the scalable communication of IIS.

ACTIVE SERVER PAGES.NET (ASP.NET):

ASP.NET is a unified Web development platform that provides the services necessary for developers to build enterprise-class Web applications. While ASP.NET is largely syntax compatible with ASP, it also provides a new programming model and infrastructure for more secure, scalable, and stable applications. We can feel free to augment our existing ASP applications by incrementally adding ASP.NET functionality to them.

ASP.NET is a compiled, .NET-based environment; we can author applications in any .NET compatible language, including Visual Basic .NET, C#, and JScript .NET. Additionally, the entire .NET Framework is available to any ASP.NET application. Developers can easily access the benefits of these technologies, which include the managed common language runtime environment, type safety, inheritance, and so on.

ASP.NET has been designed to work seamlessly with WYSIWYG HTML editors and other programming tools, including Microsoft Visual Studio .NET. Not only does this make Web development easier, but it also provides all the benefits that these tools have to offer, including a GUI that developers can use to drop server controls onto a Web page and fully integrated debugging support.

Developers can choose from the following two features when creating an ASP.NET application, Web Forms and Web services, or combine these in any way they see fit. Each is supported by the same infrastructure that allows us to use authentication schemes, cache frequently used data, or customize our application's configuration, etc.

- Web Forms allows us to build powerful forms-based Web pages. When building these pages, we can use ASP.NET server controls to create common UI elements, and program them for common tasks. These controls allow us to rapidly build a Web Form out of reusable built-in or custom components, simplifying the code of a page.
- An XML Web service provides the means to access server functionality remotely. Using Web services, businesses can expose programmatic interfaces

to their data or business logic, which in turn can be obtained and manipulated by client and server applications. XML Web services enable the exchange of data in client-server or server-server scenarios, using standards like HTTP and XML messaging to move data across firewalls. XML Web services are not tied to a particular component technology or object-calling convention. As a result, programs written in any language, using any component model, and running on any operating system can access XML Web services.

Each of these models can take full advantage of all ASP.NET features, as well as the power of the .NET Framework and .NET Framework common language runtime. These features and how we can use them are outlined as follows:

- The ASP.NET object model has changed significantly from ASP, making it more structured and object-oriented. Unfortunately this means that ASP.NET is not fully backward compatible; almost all existing ASP pages will have to be modified to some extent in order to run under ASP.NET. In addition, major changes to Visual Basic .NET mean that existing ASP pages written with Visual Basic Scripting Edition typically will not port directly to ASP.NET. In most cases, though, the necessary changes will involve only a few lines of code.
- Accessing databases from ASP.NET applications is an often-used technique for displaying data to Web site visitors. ASP.NET makes it easier than ever to access databases for this purpose.
- ASP.NET provides a simple model that enables Web developers to write logic that runs at the application level. Developers can write this code in the global. sax text file or in a compiled class deployed as an assembly. This logic can include application-level events, but developers can easily extend this model to suit the needs of their Web application.
- ASP.NET provides easy-to-use application and session-state facilities that are familiar to ASP developers and are readily compatible with all other .NET Framework APIs.
- For advanced developers who want to use APIs as powerful as the ISAPI programming interfaces that were included with previous versions of ASP,

ASP.NET offers the IHttpHandler and IHttpModule interfaces. Implementing the IHttpHandler interface gives us a means of interacting with the low-level request and response services of the IIS Web server and provides functionality much like ISAPI extensions, but with a simpler programming model. Implementing the IHttpModule interface allows us to include custom events that participate in every request made to our application.

- ASP.NET takes advantage of performance enhancements found in the .NET Framework and common language runtime. Additionally, it has been designed to offer significant performance improvements over ASP and other Web development platforms. All ASP.NET code is compiled, rather than interpreted, which allows early binding, strong typing, and just-in-time (JIT) compilation to native code, to name only a few of its benefits..
- Writing custom debug statements to our Web page can help immensely in troubleshooting our application's code. However, it can cause embarrassment if it is not removed. The problem is that removing the debug statements from our pages when our application is ready to be ported to a production server can require significant effort. ASP.NET offers the TraceContext class, which allows us to write custom debug statements to our pages as we develop them. They appear only when we have enabled tracing for a page or entire application. Enabling tracing also appends details about a request to the page, or, if we so specify, to a custom trace viewer that is stored in the root directory of our application.
- The .NET Framework and ASP.NET provide default authorization and authentication schemes for Web applications. We can easily remove, add to, or replace these schemes, depending upon the needs of our application.
- ASP.NET configuration settings are stored in XML-based files, which are human readable and writable. Each of our applications can have a distinct configuration file and we can extend the configuration scheme to suit our requirements.

ACTIVEX DATA OBJECTS.NET (ADO.NET):

ActiveX Data Objects for the .NET Framework (ADO.NET) is a set of classes that expose data access services to the .NET programmer. ADO.NET provides a rich set of components for creating distributed, data-sharing applications.

ADO.NET is the main data access system and protocol that Visual Basic.net uses. It uses a disconnected data architecture, which means that the data we work with is just a copy of the data in the actual database.

To store the data we work with in our application, we use datasets, which represent a sort of data cache of records. The data in the dataset is usually a much-reduced version of what is in the database. While we are doing so, we remain disconnected from the database.

Datasets are really just passive containers for data. To actually get data from a database and write it back, we use data adapters. A data adapter contains the instructions for populating a single table in the dataset and updating the corresponding table in the database. The instructions are methods that encapsulate either SQL commands or references to stored procedures.

The dataset is a container; it is filled by SQL commands or stored procedures executed from a data adapter. If we want to see the latest changes made by other users, we can refresh the dataset by calling the FILL method.

ADO.NET uses XML as the format for transferring data. Similarly, if data needs to be saved, which Microsoft calls “persisting,” it is stored as XML, the data description language developed by the World Wide Web Consortium (W3C).

FEATURES OF ADO.NET:

The following are the key features offered by ADO.NET.

- Interoperability
- Maintainability
- Programmability
- Performance
- Scalability

Interoperability:

ADO.NET uses XML for transmitting datasets among components and across tiers. Any component that is capable of reading the XML format can process the data. It is not necessary for the receiving component to be an ADO.NET component. However, the important point to be considered is that the receiving component should be capable of accepting the XML file formatted as a dataset.

Maintainability:

After the application is deployed on a server, there might be a need for changes in application. These changes to improve the performance of the application may affect the system resources, resulting in higher response time for the user. As a solution to these problems, the application might need to undergo architectural changes by adding tiers. The transformation between tiers becomes easier if the original application is implemented in ADO.NET using datasets. In ADO.NET, the communication between tiers is relatively easy; because the tiers can transmit data through XML formatted datasets.

Programmability:

ADO.NET model uses typed programming, so the programming language itself recognizes the types of things that are important to users.

Also the errors in syntax caused by misspelling are detected at compile time rather than at run time.

Performance:

In ADO, while transmitting data across tiers using COM marshalling in the form of disconnected Record Sets, the values must be converted to data types that are recognized by COM. This results in poor performance. On the other hand, ADO.NET is designed to use the disconnected data architecture, which in turn is easier to scale because it reduces the load on database. Thus, in the ADO.NET model, everything is handled at the client side, which in turn improves performance.

Scalability:

The web based, data-centric applications require multiple users to access data simultaneously. Applications that use resources, such as database connections and database locks cannot support more users to access data simultaneously, because eventually the user demand for the limited resources will exceed their supply. Because ADO.NET uses disconnected data access, applications do not retain database locks or active database connections for long durations. Hence, ADO.NET accommodates scalability.

SQL Server 2000 Architecture

Microsoft® SQL Server™ 2000 product meet the data storage requirements of the largest data processing systems and commercial Web sites and provides easy-to-use data storage services.

The data storage needs of a modern corporation or government organization are very complex. Some examples are:

- Online Transaction Processing (OLTP) systems.
- Organisations like Human resources planning, manufacturing resources planning, and inventory.
- Independent Software Vendors (ISVs) must be able to distribute data storage capabilities.

Features of SQL Server 2000

Microsoft® SQL Server™ 2000 features include:

- **Internet Integration.**

The SQL Server 2000 database engine includes integrated XML support. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications.

- **Scalability and Availability.**

SQL Server 2000 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.

- **Enterprise-Level Database Features.**

The SQL Server 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity.

- **Ease of installation, deployment, and use.**

SQL Server 2000 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites.

- **Data warehousing.**

SQL Server includes tools for extracting and analyzing summary data for online analytical processing. SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

Relational Database Components

The database component of Microsoft® SQL Server™ is a Structured Query Language (SQL)–based, scalable, relational database with integrated Extensible Markup Language (XML) support for Internet applications.

Database:

A database is similar to a data file in that it is a storage place for data. A database does not present information directly to a user; the user runs an application that accesses data from the database and presents it to the user in an understandable format.

Scalable:

SQL Server supports having a wide range of users access it at the same time. An instance of SQL Server includes the files that make up a set of databases and a copy of the DBMS software.

Structured Query Language:

To work with data in a database, you have to use a set of commands and statements (language) defined by the DBMS software.

Extensible Markup Language:

XML is the emerging Internet standard for data. XML documents can be easily processed by the Hypertext Markup Language, which is the most important language for displaying Web pages.

Chapter 8
S/W & H/W Environment

S/W and H/W Environment:

SOFTWARE CONFIGURATION

OPERATING PLATFORM	**:**	WINDOWS 10
RDBMS	**:**	SQL SERVER 2018
SOFTWARE	**:**	Visual Studio 2022
FRONT END TOOL	**:**	Asp.Net

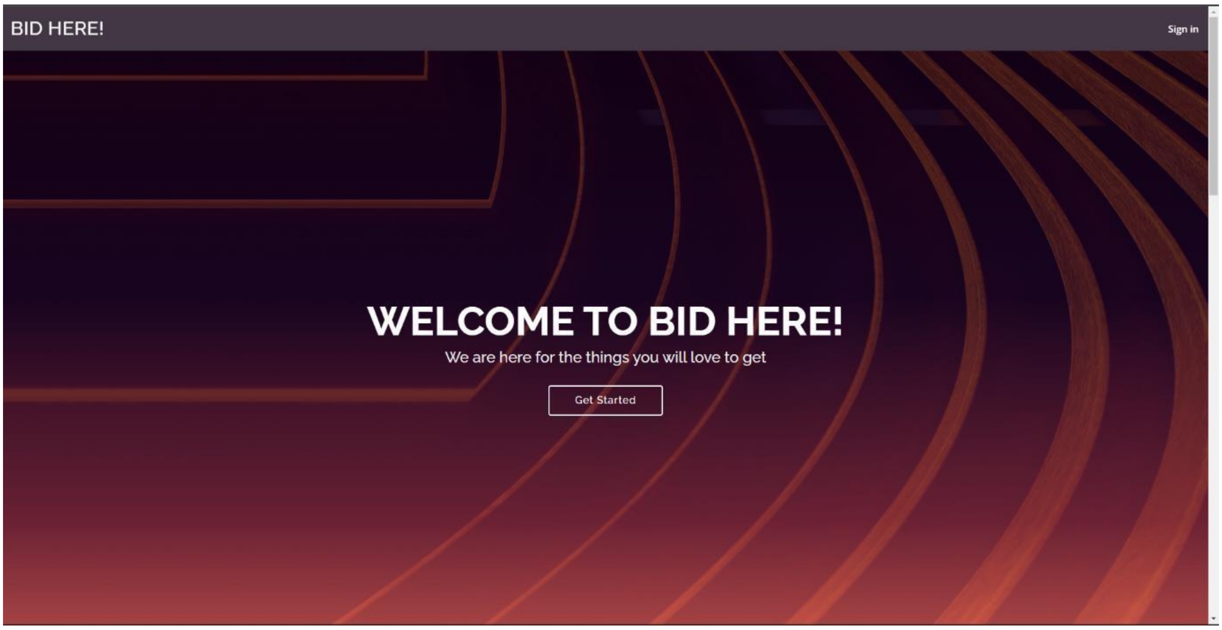
HARDWARE CONFIGURATION

RAM	**:**	16GB
HARDDISK	**:**	1 TB

Chapter 9
Outputs

Output

User Interface:



BID HERE!

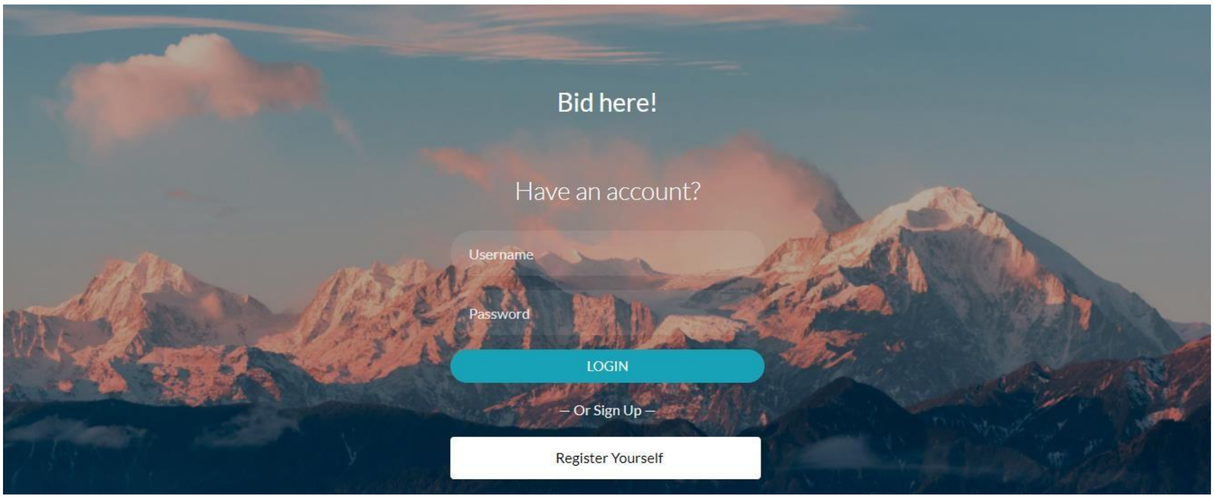
Sign in

WELCOME TO BID HERE!

We are here for the things you will love to get

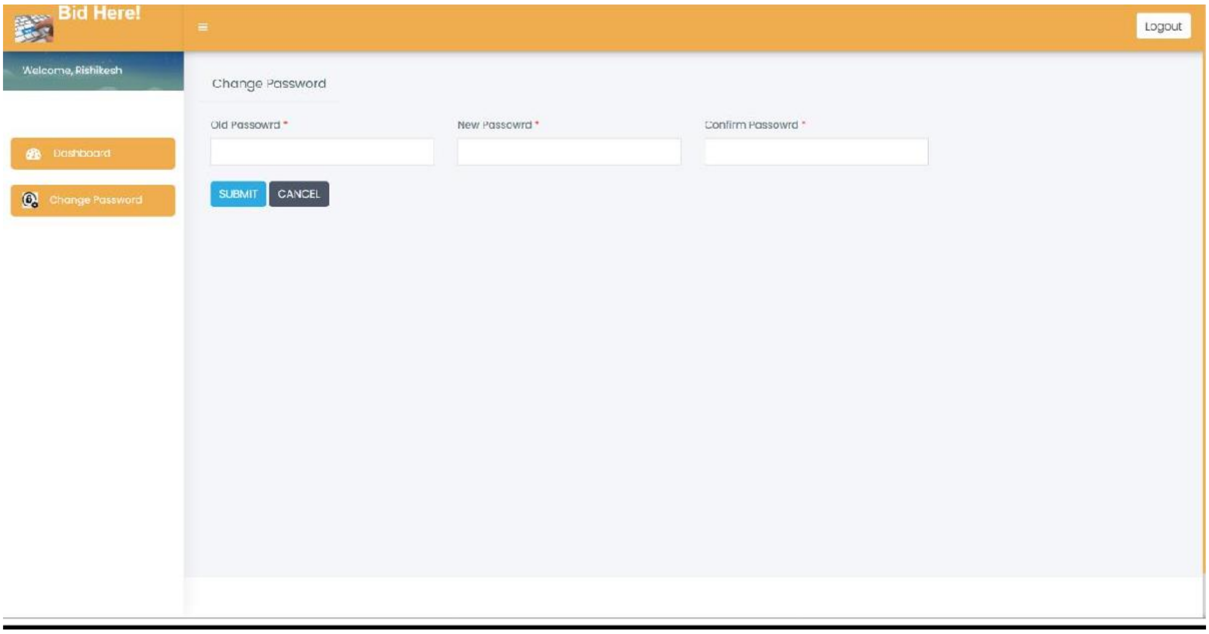
This page is default page of Online auction website. User will get all details about our website. User will redirect to Login page once he/she will click on button “Sign In”.

Login page for vendor seller and admin:



This page have options like it shows the list of products in which a user can bid or set Bidding, user can login from this page with the username and password given to him by registration. If a new user wants to do a Bidding or bid he can sign up from the sign up link.

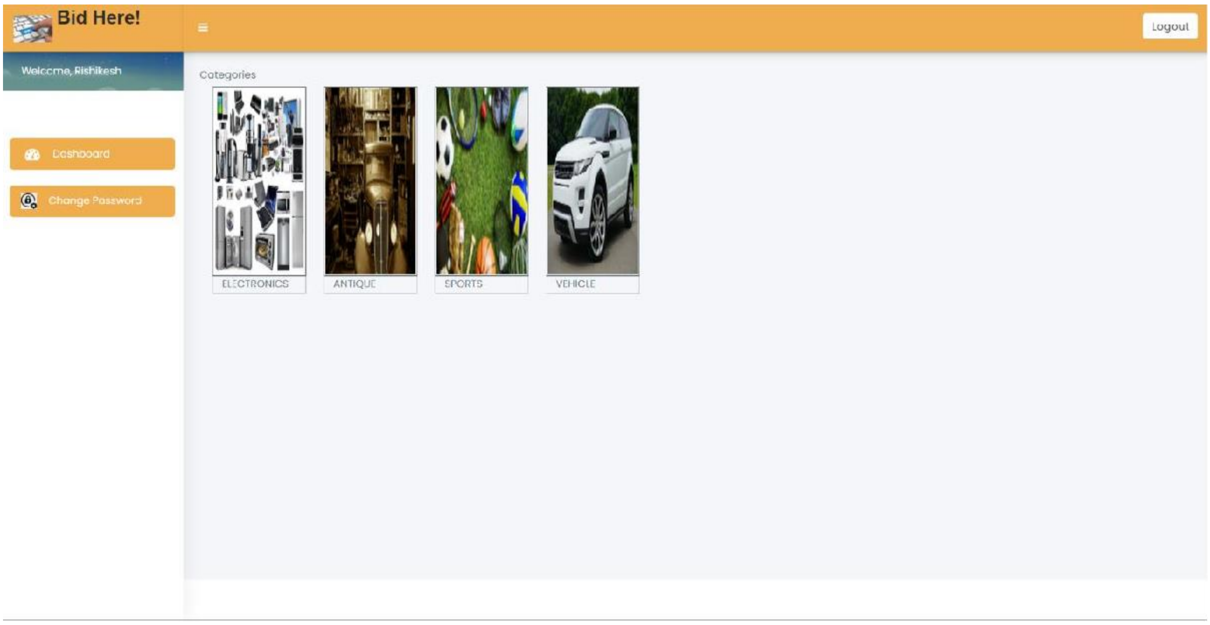
Password Recovery



If a user forgot his password he/she can retrieve his password from this page.

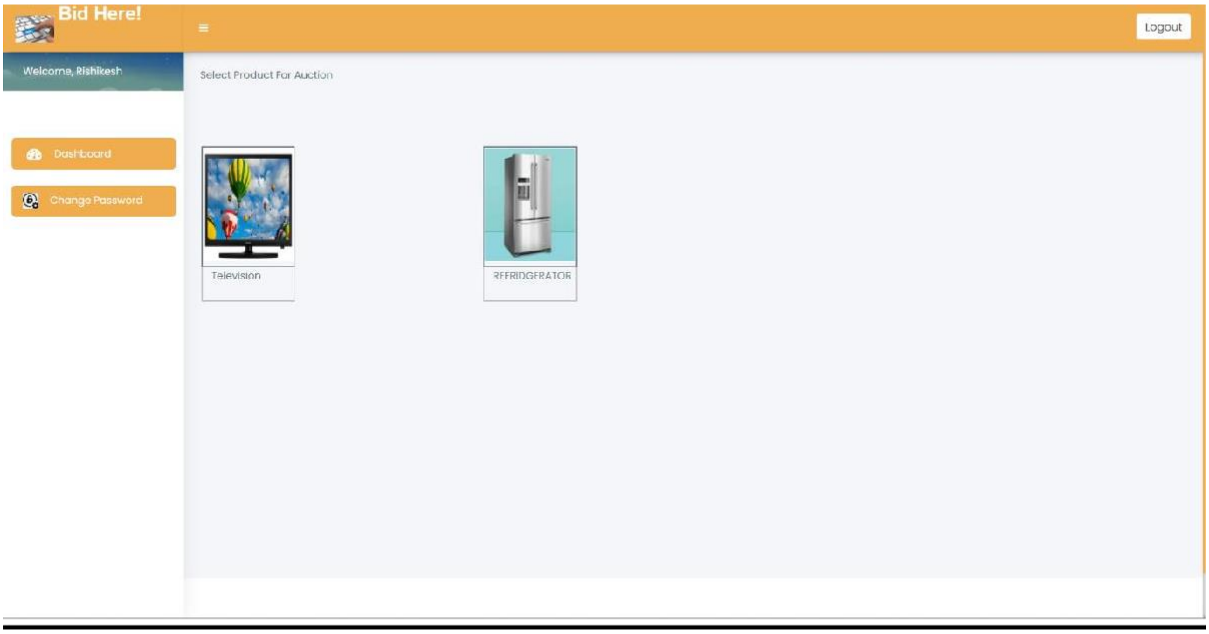
He has to answer his security question and enter his username both information should be correct if he/she wants to retrieve his/her password after clicking on submits.

Vendor Account



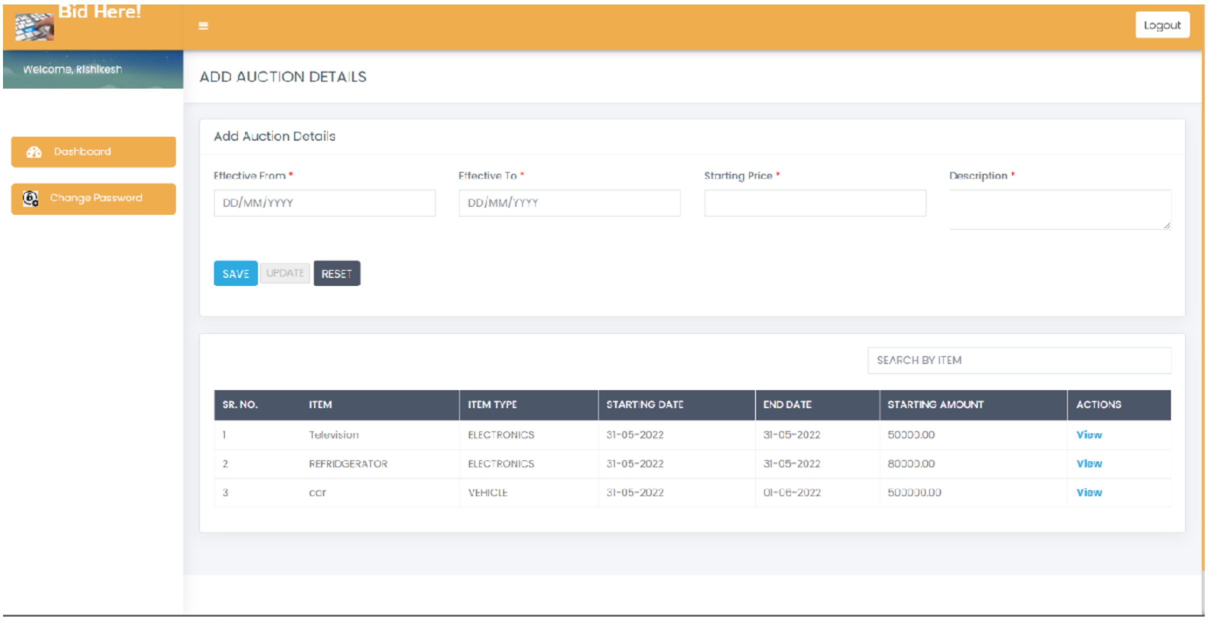
This is vendor account page after vendor login this page will show this page have all details of user his mobile no bank account details address etc. he can edit his details and he can start Bidding from the given categories by clicking on it. All Biddings by current user will also shows in this page.

Add Bidding page



When a user click on categories of product all products will show in this page in that categories .like a user clicks on electronics then all related products to electronics will shows in this page user can click on these items and then start Bidding.

Add Bidding Details



ADD AUCTION DETAILS

Add Auction Details

Effective From * DD/MM/YYYY

Effective To * DD/MM/YYYY

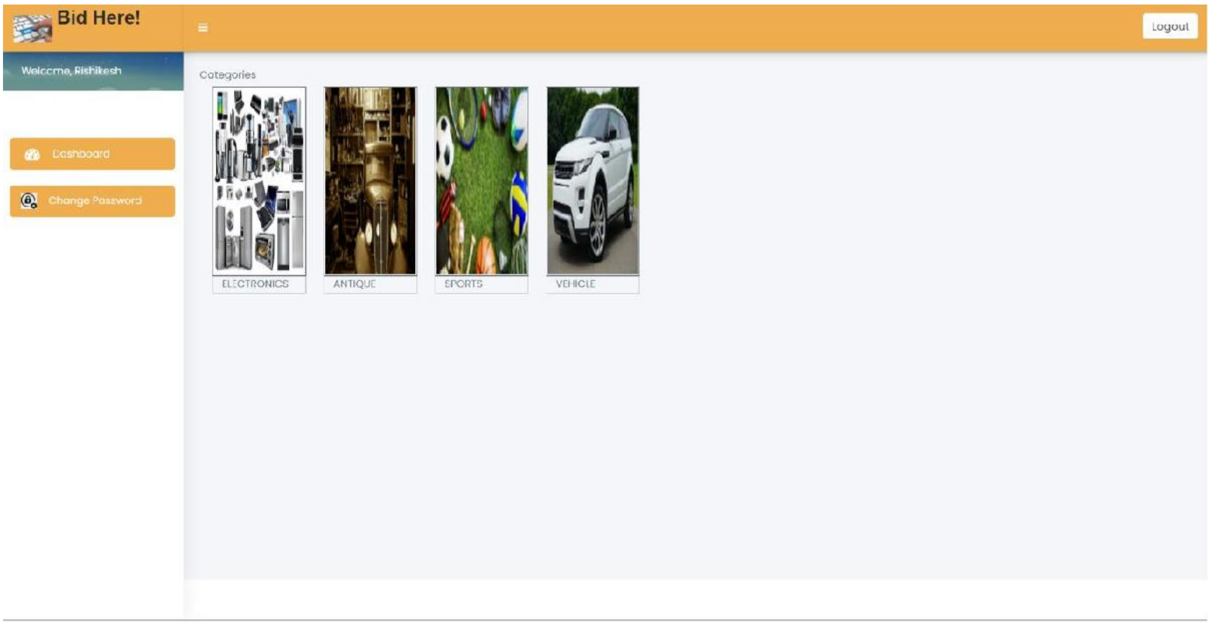
Starting Price *

Description *

SR. NO.	ITEM	ITEM TYPE	STARTING DATE	END DATE	STARTING AMOUNT	ACTIONS
1	Television	ELECTRONICS	31-05-2022	31-05-2022	50000.00	View
2	REFRIDGERATOR	ELECTRONICS	31-05-2022	31-05-2022	80000.00	View
3	car	VEHICLE	31-05-2022	01-06-2022	500000.00	View

After selecting the item for bidding user will add details of Bidding from this page like starting date ending date of the Bidding and starting price for bidding and description of his product.

Admin Page



This page opens after admin’s login from this page side menus have all the options for the admin , admin can add items delete existing items and updates items.

Chapter 10
System Testing

SYSTEM TESTING

In the test phase various test cases intended to find the bugs and loop holes exist in the software will be designed. During testing, the program to be tested is executed with a set of test cases and the output of the program is performing as it is expected to.

Often when we test our program, the test cases are treated as " throw away" cases. After testing is complete, test cases and their outcomes are thrown away. The main objective of testing is to find errors if any, especially the error uncovered till the moment. Testing cannot show the absence of defects it can only show the defects that are a set of interesting test cases along with their expected output for future use.

Software testing is crucial element and it represents at the ultimate review of specification design and coding. There are black box testing and glass box testing. When the complete software testing is considered Back box attitudes to the tests. That is concluded predicted on a close examination of procedural detail.

The software is tested using control structures testing method under white box testing techniques. The two tests done under this approach. One condition testing to check the Boolean operator errors, Boolean variable errors, Boolean parenthesis errors etc. Loop testing to check simple loops and tested loops.

Faults can be occurred during any phase in the software development cycle. Verification is performed on the output in each phase but still some fault. We likely to remain undetected by these methods. These faults will be eventually reflected in the code. Testing is usually relied upon to detect these defaults in addition to the fault introduced during the code phase .For this, different levels of testing are which perform different tasks and aim to test different aspects of the system.

Chapter 11

Coding

Coding

Sign in page:

```
<!doctype html>
<html lang="en">
<head>
    <title>Bid here!</title>
    <meta  charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" href="css/style.css">
</head>
<body class="img js-fullheight" style="background-image: url(images/bg.jpg);">
    <section class="ftco-section">
        <div class="container">
            <div class="row justify-content-center">
                <div class="col-md-6 text-center mb-5">
                    <h2 class="heading-section">Bid here!</h2>
                </div>
            </div>
            <div class="row justify-content-center">
                <div class="col-md-6 col-lg-4">
                    <div class="login-wrap p-0">
                        <h3 class="mb-4 text-center">Have an account?</h3>
                        <form action="#" class="signin-form" runat="server">
                                                            <div class="form-group">
                                                                            <%--<input type="text" class="form-control"
placeholder="Username" required>--%>
                          <asp:TextBox ID="txtUserName" runat="server" class="form-control"
placeholder="Username"></asp:TextBox>
                                                             </div>
                           <div class="form-group">
                             <%--<input id="password-field" type="password" class="form-control" placeholder="Password"
required>--%>
                    <asp:TextBox ID="txt_pwd" runat="server" class="form-control" placeholder="Password"
TextMode="Password"></asp:TextBox>
                             <%--<span toggle="#password-field" class="fa fa-fw fa-eye field-icon toggle-password"></span>--
%>
                           </div>
                           <div class="form-group">
                              <%--<button type="submit" class="form-control btn btn-primary submit px-3">Sign In</button>--
%>
                    <asp:Button ID="btnlog" OnClick="btnlog_Click" runat="server" Text="Login" class="btn btn-info btn-
block margin-top-10" />
                           </div>
                          <%-- <div class="form-group d-md-flex">
                              <div class="w-50">
                                                                                                                         </div>
                                                                                                                         <div class="w-50 text-md-right">
                                                                                                                                        <a href="#" style="color:
#fff">Forgot Password</a>

                                                                                                                         </div>
                           </div>--%>
                         </form>
                        <%-- <p class="w-100 text-center">&mdash; Or Sign In With &mdash;</p>
                        <div class="social d-flex text-center"><a href="#" class="px-2 py-2 mr-md-1 rounded"><span class="ion-logo-facebook
                        mr- 2"></span>Facebook</a>
                            <a href="#" class="px-2 py-2 ml-md-1 rounded"><span class="ion-logo-twitter mr-
2"></span>Twitter</a>
                        </div>--%>
                    </div>
                </div>
            </div>
        </div>
    </section>

    <script src="js/jquery.min.js"></script>
    <script src="js/popper.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/main.js"></script>

</body>
</html>
```

Chapter 12
Conclusion

CONCLUSION

The efficiency of any system designed to suit an organization depends cooperation during the implementation stage and also flexibility of the system to adopt itself to the organization. “Net-Aucs” has been developed to overcome the problems with traditional Bidding systems.

As evidence of the success of this mission, there are millions of items listed each day in thousands of different categories. There are items for almost any interest that one could imagine, from sheet music to automobiles to hand tools to real estate. And the variety doesn’t stop there. Need a computer? One may find it listed in the proper category, in any configuration from very old and obsolete to the latest greatest machine available. What about antiques? One can find an antique quilt that is up for highest bid, or maybe an old violin, whose beautiful tones have enchanted many though its years. Tickets. Maybe a ticket to the next concert of ones favorite artist or play production. One can even find that special bottle of wine, some aged, exotic cheese, and the perfect ‘mood’ music for that special occasion.

Chapter 13
References

REFERENCES:

[1] Hu Wenyan, Alvaro Bolivar, “Online Auctions Efficiency: A Survey of eBay Auctions”, Alternate Track: Industrial Practice and Experience, 2008.

[2] Charu C. Aggarwa, Philip S. Yu, “Online Auctions: There can be only one”.

[3] Xiling Cui, Vincent S. Lai and Connie K.W. Liu “Consumer Behaviour in Online Auctions: A Review”, Electronic Markets Vol. 18 No.4.

[4] Chuan-Hoo Tan, Hock-Hai Teo, Heng Xu, “Online Auction: The Effects of Transaction Probability and Listing Price on A Sellers Decision Making Behaviour”, Electron Markets (2010) 20:6779.

[5] Liang Zhang, Na Li, “Multi-Agent Negotiation System in Online Auction”, IEEE, Second International Conference on Communication Systems, Networks and Applications, 2010.

[6] Shuangke Wu, Yanjiao Chen, Qian Wang, Minghui Li, Cong Wang, Xiangyang Luo, “Cream: A Smart Contract Enabled Collusion-Resistant e Auction”, IEEE, Transactions on Information Forensics and Security, 2018.

[7] Hamid-Reza Ghasemi, Gholam-Reza Mohammadi, “Architecture - oriented approach for detecting fraud in the online auction”, IEEE, 8th International Conference on e-Commerce with focus on e-Trust, 2014.

[8] Zhang Jie, Zhang Yaping, “Research on Duration and Bid Arrivals in eBay online Auctions in the Internet”, IEEE, 2011.

[9] Benjamin J. Ford, Haiping Xu and Iren Valova, “A Real-Time Self-Adaptive Classifier for Identifying Suspicious Bidders in Online Auctions”, Published by Oxford University Press on behalf of The British Computer Society, 2012.

[10] Janhavi Baikerikar, Vaishali Kavthekar, Esmond Dsouza, Steffie Fernandes, Mureil Dsouza, “Hammer Down-An Online BiddingApplication”, IEEE, 2017